UNIVERSITI TEKNOLOGI PETRONAS (UTP)

LINUX VIRTUAL SERVER CLUSTERING: PRISM

By

Muhammad Afif Bin Idham

13196

FYP2 Dissertation

BACHELOR OF TECHNOLOGY (Hons)

(INFORMATION & COMMUNICATION TECHNOLOGY)

Universiti Teknologi PETRONAS

Table of Contents

ACKNOWLEDGEMENT

Alhamdulillah, all the praise and gratitude is only for Allah, with His blessing and guidance has enabled me to successfully complete this Final Year Project smoothly for the past 2 years here in Universiti Teknologi Petronas (UTP). With that, I would also like to express my special gratitude to UTP for providing me the chance and facilities to conduct this project for the last 2 semesters in partial fulfilment for Bachelor of Information and Communication Technology. This project has opened my mind with many useful knowledge and experiences as I go through the hardships of completing this task while preparing me for the real working environment later after graduation.

My honest appreciation goes to my supervisor, Dr Lukman Bin Ab Rahim, who gave support to me throughout the hard times completing project with high commitment and passion. He always reminds me of the ethics of working and not to give up when there is hardship but instead face them with calm and prayers. He also gave a lot of insights and knowledge required for me to complete this entire project.

Finally, special thanks to everyone involved in the process of completing of my final year project, either directly or indirectly. Thank you.

CERTIFICATION DECLARATION

I hereby certify that this report and the project it describes is my original work and is not taken from the work of others save and to the extent that such work has been cited and acknowledged in the report.

Student:

Signed:_________ Student No:______________ Date:_________

Supervisor:

Signed:_________ Date:_________

ABSTRACT

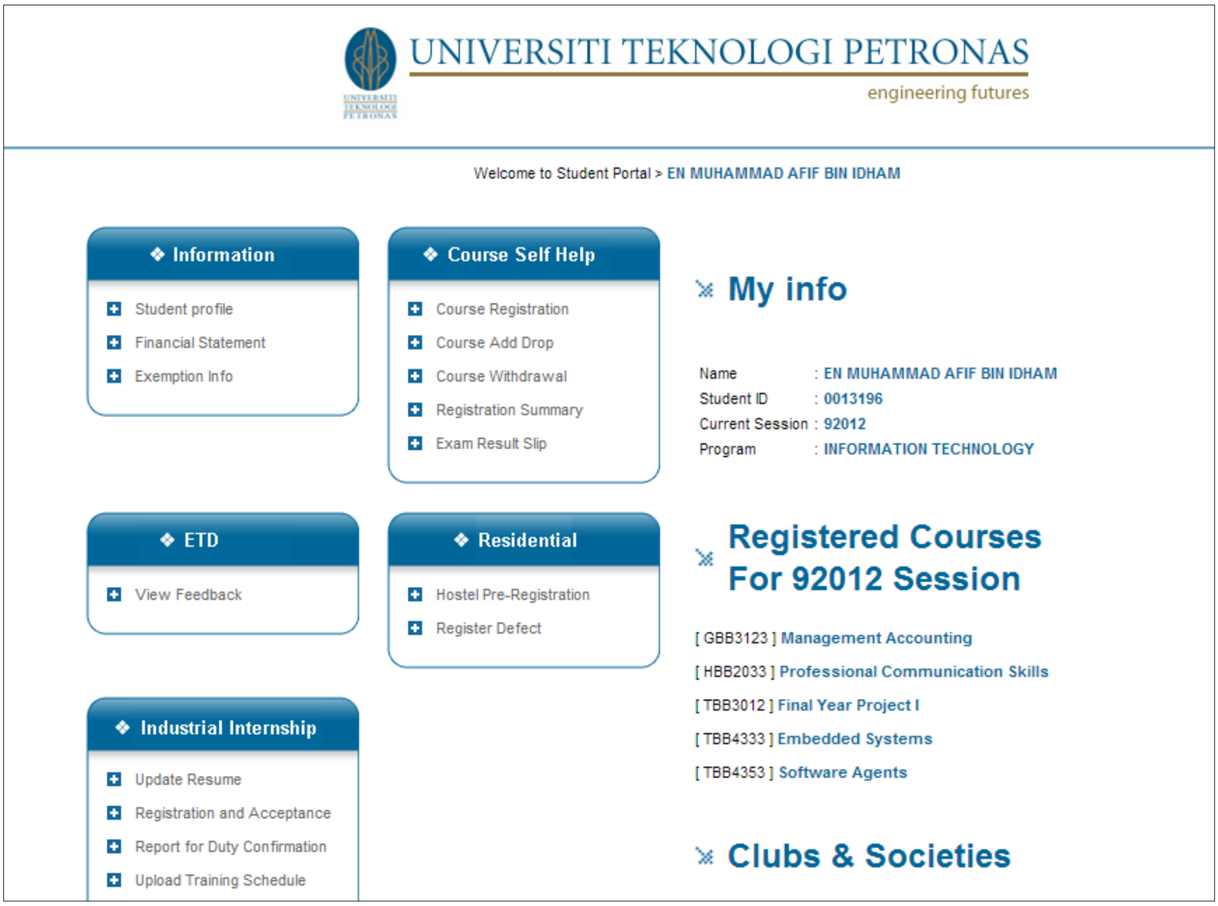
The common problem of Universiti Teknologi Petronas (UTP) Student Portal (PRISM) service failure especially during student registration period urged the needs of the existence of a solution. Basically, the idea is where a virtual server clustering architecture is developed using Linux to support the existing system by improving availability with active redundancy. Thus, the main objective of the project is to improve overall availability of PRISM and the said solution is known to increase the availability up to 99% server run time. The cluster will not only hold the prism website but it would also integrate the entire student registration database inside which allow a wider view of the PRISM service status and easier database access for the UTP staffs. The whole project will be based on the Waterfall model development cycle where each step is done thoroughly before proceeding to the next phase.

LIST OF FIGURES

1.0 Introduction

1.1 Background of Study

1.1.1 UTP Student Portal (PRISM)



Welcome to Student Portal > EN MUHAMMAD AFIF BIN IDHAM

My info

Name	: EN MUHAMMAD AFIF BIN IDHAM
Student ID	: 0013196
Current Session	: 92012
Program	: INFORMATION TECHNOLOGY

Registered Courses For 92012 Session

[GBB3123] Management Accounting
[HBB2033] Professional Communication Skills
[TBB3012] Final Year Project I
[TBB4333] Embedded Systems
[TBB4353] Software Agents

Figure 1.1.1 – UTP Student Portal (PRISM)

The UTP Student Portal or also known as PRISM is the main focus and is the main background of the project. This portal is server to provide students a medium to register their course for every new semester, review taken course and obtaining the exam result slip from previous semester.

Furthermore, PRISM also handles the application for Industrial Training, residential hostel pre-registration and hold student profile for all UTP students. All of this data is stored inside UTP itself and is maintained by the ITMS department.

1.1.2 Linux Virtual Server (LVS)

The Linux Virtual Server (LVS) clustering is a software tool that directs network connections to multiple servers that share their workload, providing a basic framework for building a highly scalable and highly available network services using a large cluster of commodity servers providing service inside the internal network which is the whole of UTP local area network (LAN). This framework will then be used to hold and run the PRISM service and its student database.

Moreover, the cluster will then directs network connections to the different virtual servers created using load balancing according to the scheduling algorithms to eliminate the problem of server burden and single point of failure. Also, parallel services to the cluster will appear as a virtual service on a single IP address. This means student interact with the cluster via PRISM as if it were a single server as usual, where they are not affected by the implementation of the cluster and do not need modification in the method of accessing this service as the cluster itself is transparent to the users.

1.2 Problem Statement

The UTP Student Portal (PRISM), responsible for providing an interface for the students to complete their course registration for each semester is facing a common seasonal problem, a situation where the servers are unable to cope with the high demand for the service and huge number of simultaneous access to the service resulting in failure of the whole service.

Students are always complaining where they are unable to complete their registration within the set period due to service being unavailable and slowed down frequently. The potential number of students that a server must support will dramatically increase during the registration period where it could reach up to thousands of simultaneous student connection and yet no solution can be seen from UTP until now. Moreover, UTP should worry how their Student Portal will cope with the similar situation in the future since the number of total students in UTP will further increase from the new intakes.

1.3 Objectives and Scope of Study

To address the above issues, basically setting up and implement the Virtual Linux Server Clustering is one of the best solutions to the above requirements. By implementing this, our main aim is to:

- Improve the overall availability of the PRISM service by eliminating single point of failure (SPOF) in the current system
- Improve reliability of PRISM servers using load balancing and applying redundancy to enhance availability.

In order to create the system architecture, the scope of study would be to develop a clustered framework for the UTP Student Portal to act as a reliable platform for student registration

1.4 Feasibility and Relevancy Study

This project is totally feasible to be executed in UTP simply because the whole project can run on totally no cost. Simply because the reduce cost of pre-production procedures such as simulation and testing is free since Linux is a free software which improve the total cost of production and also testing can be done in a personal computer environment using VMware.

In a world with rapid pace of technological enhancement, this LVS project will still be relevant since the IPv6 support for Internet Protocol Virtual Server (IPVS) was included in the Linux kernel, which means implementation of IPv6 (which will sure to be implemented in our near future to replace the old IPv4) in our University Environment will not affect the network structure of UTP and no modification is needed.

2.0 Literature Review

2.1 Availability Issue in the Business World

“In any important business services, rapid response and high availability are critical”

By: Wensong and Wenzhuo.(2003)

In the business world where client accessing the system all the time, availability is the criteria that are most stressed on by the customers in looking for their preferred service which offers the **best access experience** to complete their tasks. This shows that availability issues shouldn’t be taken lightly and drastic measures are needed if the service’s downtime is frequent or else the company image could be **tarnished**, resulting in loss of revenue.

Failure to keep high rate of service availability has always been an enormous issue especially in the IT business world. Even huge business firms such as Continental Airlines, the London Stock Exchange, eBay, AT&T, and even Microsoft have experienced high-profile downtime of their service which resulted in receiving huge press coverage for all the wrong reasons. For most, the outages resulted in lost revenue and even tarnished their company image and reputation. For example in January 2001, Microsoft suffered a three-day outage of many of its Web sites because of a domain name service (DNS) configuration error, a common cause of failure for many enterprise Web sites. This shows that even the most-sophisticated IT environments are vulnerable to service failure and its consequences are severe.

A research conducted named “Often the weakest link in Business Availability” Gartner (2001) shows that mission-critical application service downtime caused by technology failure should be minimized at all cost. But for big company to actually address this matter, it is not a cheap matter. In “Making Smart Investments to reduce unplanned downtime” Donna (1999) also agrees that due to the complexity of today’s IT infrastructure and applications makes high-availability systems management extremely difficult and the effect of it is equally painful to the company.

One of the simplest solutions to these availability issues would be using the LVS Clustering method. Although LVS claimed to be a new technology in Malaysia, it is proven to be an effective and safe method in the current world as it is also preferred and implemented on huge websites such as astro.com and Wikipedia.com. Certain Malaysian Government Body such as Jabatan Pengangkutan Jalan (JPJ) is also benefiting from this infrastructure. This approach will still be viable to any company towards the future technology since the IPv6 support for Internet Protocol Virtual Server (IPVS) was included in the Linux kernel on November 2, 2008 by Julius Volz and Vince Busam from Google Inc.

2.2 Linux Virtual Server (LVS) Type

In "Linux Virtual Server Clusters: Build highly-scalable and highly-available network services at low cost" Wensong and Wenzhuo (2003) stated that the system architectures should be designed so that it fits the most for their client business. Thus, there are three main types of LVS to consider based on each specific system's customer requirement which are:

- LVS - DR: Direct Routing

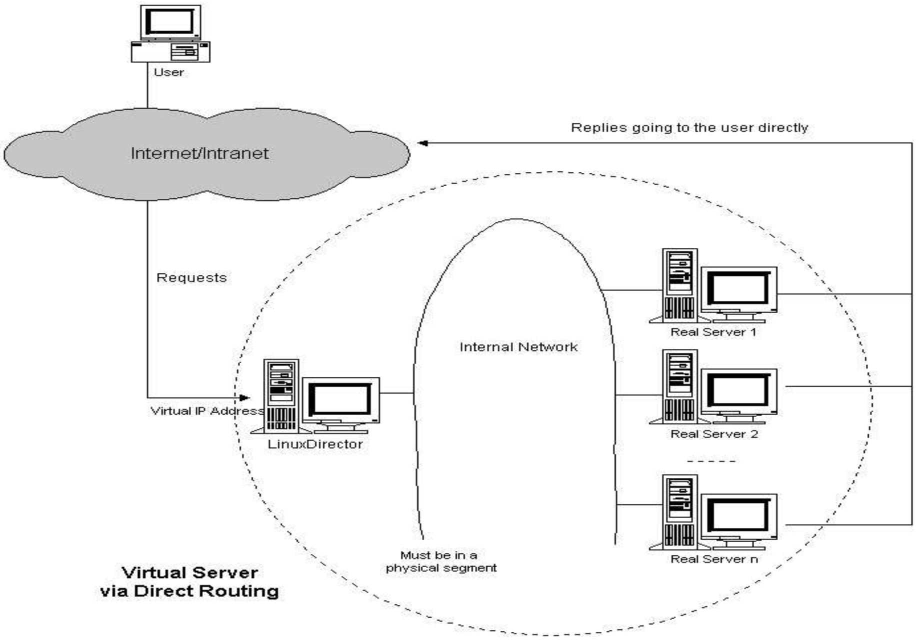


Figure 2.2.1 – Example of LVS via Direct Routing

The load balancer receives the request and forwards the packet directly to a waiting real server to process and the response from the real server is sent directly to the client. The IP packet is not modified, so the real servers must be configured to accept traffic for the virtual server's IP address. Performance-wise this method gives the best result, but it has a certain restriction which is all the servers must be on the same network subnet and they have to be able to share the same router with no other routing devices in between them. Preferred if all the components are connected in the same LAN segment.

- LVS - TUN: Tunneling

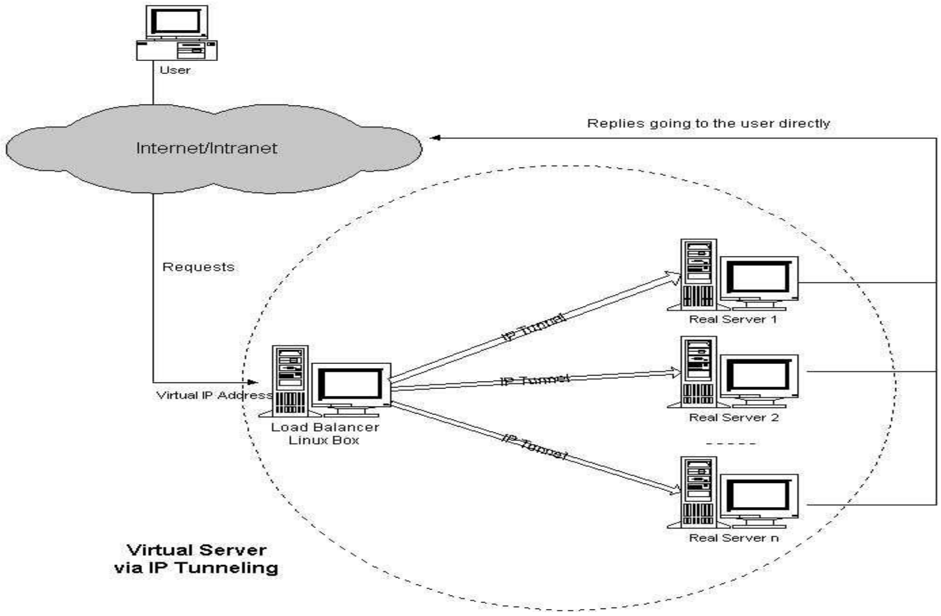


Figure 2.2.2 – Example of LVS via IP Tunneling

Tunneling removes the restriction of all servers must be on the same network. Similar to the direct routing approach, but in tunneling the packets are encapsulated and sent directly to the real servers once the load balancer receives them. With this feature, load balancing between multiple datacenters is made possible. This is preferred if the server nodes are geographically located apart from each other.

- LVS - NAT: Network Address Translation (NAT)

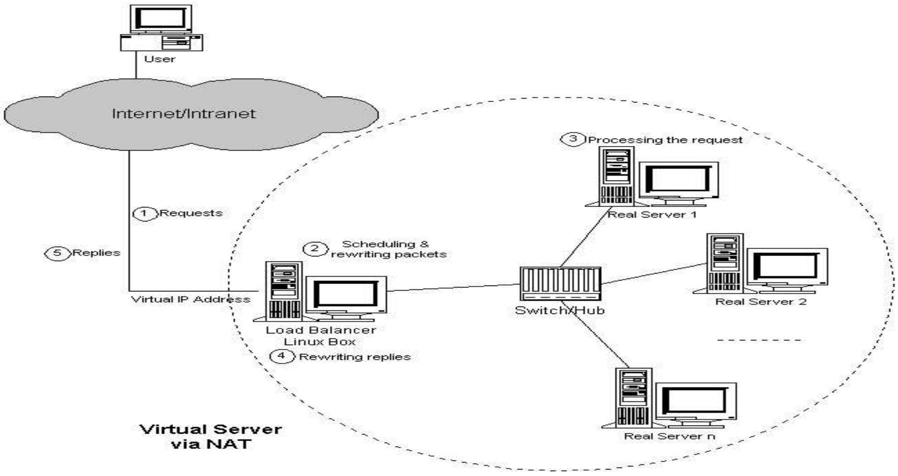


Figure 2.2.3 – Example of LVS via NAT

In this configuration, the incoming requests are rewritten so that they will be transported correctly in a NAT environment. This puts a bigger burden on the load balancer as it must rewrite the requests quickly while still keeping up with how much work is being done by each web server. By doing so, this design yields the least performance and scaling of all of the implementation options but nevertheless, this installation method is the simplest out of the 3 and will be more time-efficient to implement.

2.3 Linux Virtual Server (LVS) Configuration

There are 2 types of configuration that can be used to set up the basic framework of the cluster which are: (in the case of 2 servers in the cluster)

i. Active-Active Server Configuration

- Both of the servers in the cluster are active at the same time. The load balancer will distribute the load equally to all available servers. This configuration is mainly used to improve performance and it doesn’t improve reliability and availability since both of the servers are active, if any of them stops for any reason, all the processes and transaction of the downed server will be lost.

ii. Active-Passive Server Configuration

- Only 1 server of the cluster is active and runs all the services at one time and the other will be on stand-by until the active server goes down. The heartbeat service that monitor the status of the servers will automatically fail-over (transfer) all the resource from the active server to the stand-by server if it cannot get any reply from the active server (server down). Thus, all on-going transaction and processes will be continued in the other server instead of being completely lost. This configuration focuses on improving service availability and reliability instead of performance.

3.0 Methodology

3.1 Software Development Life Cycle (SDLC) technique

Although this not a software development project, but the steps involved is similar and can be implemented. To set up the whole service providing system, the waterfall model is used to construct the virtual clustering development process. Since this model is the most common one, it is easy to understand and easy to use. In a waterfall model, each phase must be done thoroughly before moving on to the next phase. This will ensure that all phases are done according to UTP's specific requirement. Any requirement changes should be made entirely during requirement gathering before proceeding with the design phase.

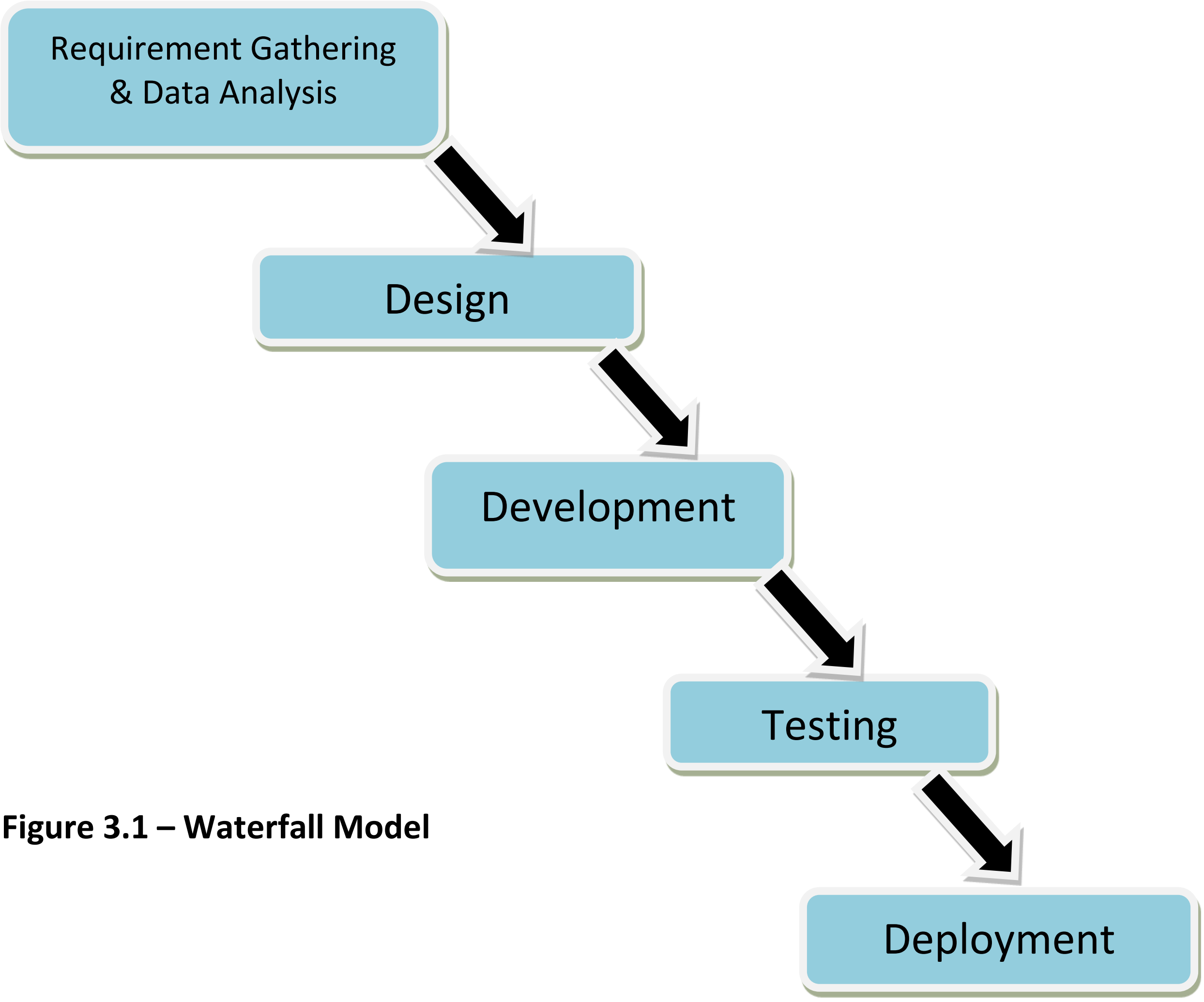


Figure 3.1 – Waterfall Model

Phase 1: Requirement Gathering & Data Analysis

- Before deciding to develop the product, problem statements need to be identified so that objectives of the project can be set up.
- In this phase, all the necessary data are gathered and analyzed to develop the product. Knowledge on how the Linux Virtual Server Clustering works and how to develop it is needed before starting the project. The requirement of the client which is UTP in this case needs to be specified according to their specification. Only then then the system can be developed.

Phase 2: Design

- After data have been thoroughly analyzed, the cluster design can be started by designing the architecture and the resources associated to each server must be identified and defined according to the requirements that have been discussed earlier. The PRISM prototype is also designed in this phase using Joomla 2.6.

Phase 3: Development

- This phase will concentrate on producing the products so that it can operate and function effectively. A prototype is made as a sample of the original product. The system is first configured and developed using the VMware and PRISM test page before porting it to the real server in UTP after finishing testing phase.

Phase 4: Testing

- During testing phase, any defects on the system are identified and fix to ensure the product effectiveness and whether it meets the client expectation and solve their problems. The architecture is tested with the Server Stress Test to check whether the overall availability of the PRISM service is improved compared to before.

Phase 5: Deployment

- After successfully passed the testing phase and it is confirmed error-free, the system and its configuration are implemented to the UTP servers. Technical support and a manual along with the product release will be provided to assist the maintenance staff regarding the architecture to maintain and troubleshoot the system.

3.2 Project Strategy/Activity

I. Requirement gathering
II. Conduct research and study on subject
III. Design Virtual Server Architecture
IV. Phase 1(Prototype): Develop virtual machines and Linux in VMware.
V. Install and configure PRISM services.
VI. Phase 2: Move/Set up to mainframe (UTP server's environment).
VII. Configure/Set up identical services to the test build.
VIII. Test connection and services (Availability Test/Server Stress Test).
IX. Prepare Final Report and Technical Paper.

3.3 Project Planning.

Activities\Week	1	2	3	4	5	6	7	8	9	10	11	12	13
Finding Supervisor	x												
Research on propose title		x	x										
Propose final chosen title			x										
Requirement Gathering				x	x	x	x						
Planning system design							x	x	x	x			
Project Testing										x	x		
Analyze data												x	x

Figure 3.3.1 – Gantt chart (FYP1)

Activities\Week	1	2	3	4	5	6	7	8	9	10	11	12	13
Development	x	x	x	x	x	x	x	x					
Cluster Design and Components	x	x	x										
Develop PRISM website prototype				x	x	x	x	x					
Integrate Apache Server with DB2.						x	x	x					
System Testing									x	x			
Cluster Server Stress Testing									x	x			
Review complete prototype with supervisor											x		
Maintenance/ Prepare Technical Paper												x	x

Figure 3.3.2 – Gantt chart (FYP 2)

3.4 Research Methodology

- Presentation and Documentation sourced from internet. Example: The article "Linux Virtual Server Clusters: Build highly-scalable and highly-available network services at low cost" written by Wensong Zhang and Wenzhuo Zhang.
- Linux Virtual Server Documentation Website - http://www.linuxvirtualserver.org/Documents.html
- IBM Redbooks on Linux Virtual Server with system Z.
- LVS deployment website: http://www.linuxvirtualserver.org/deployment.html

3.5 Tools/Software

- VMware Server 2.0.2 – downloaded free from http://downloads.vmware.com/d/details/server202/dCpiZCVqdGJkZWRq
- SUSE Linux Enterprise Server (SLES) 11 SP1 & SUSE Linux Enterprise High Availability (HA) Extender – can be downloaded in http://download.novell.com/index.jsp
- Wireshark 1.6.4 – tracing packets in network level.
- Joomla 2.6. – Develop the prototype website to simulate the PRISM website.

4.0 Discussion

4.1 Initial Cluster Design

Based on the Linux Virtual Server (LVS) designs that have been stated in the Literature Review section, it can be concluded that the LVS-Direct Routing structure and the Active-Passive Server Configuration are the best to be implemented in this project simply because it offers the best in terms of availability enhancement but one important aspect that should be noted is that when the cluster service uses this design, the system can only be accessed from a local network with a similar subnet, which means the PRISM service can only be accessed by the student only using the UTP Local Area Network and not from any external network/internet. However, this also means that there is added security to the whole system in terms of breach of access from any external network personnel.

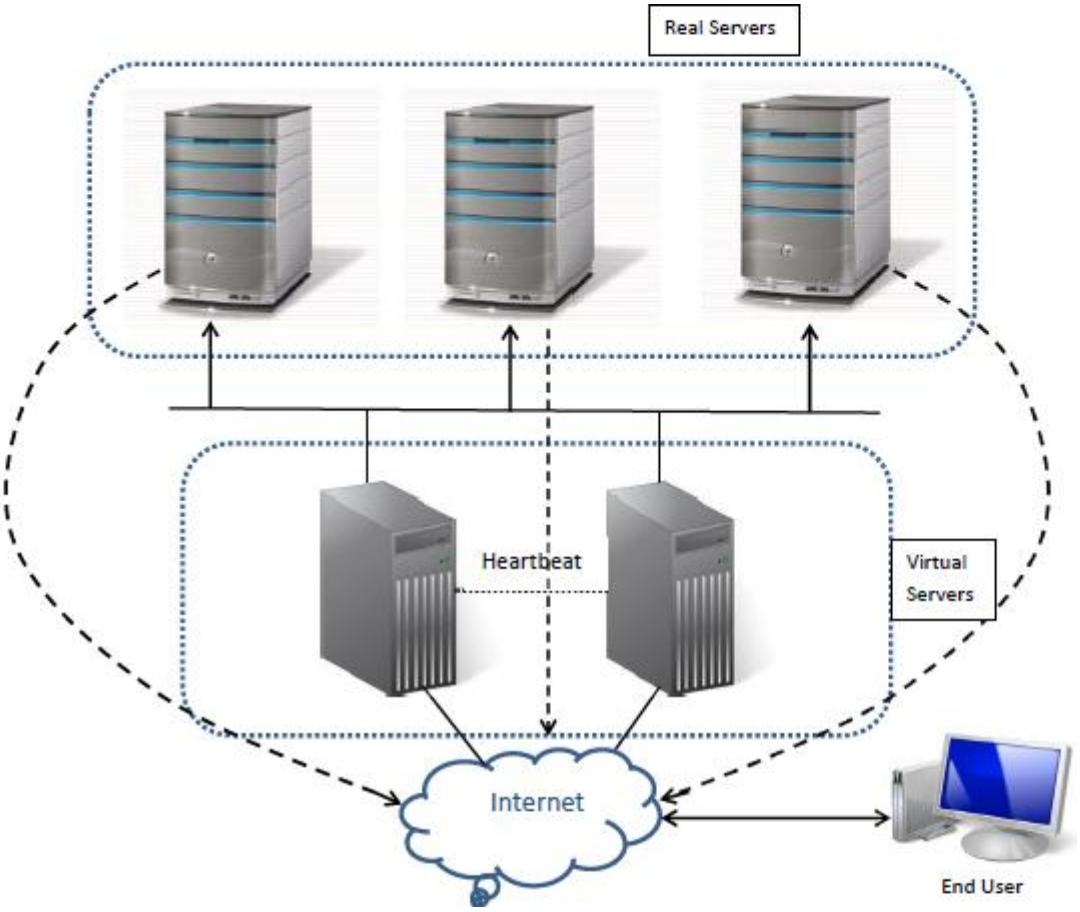


Figure 4.1.1 – Initial Cluster Design (Direct Routing)

4.2 Process Flow – Direct Routing LVS

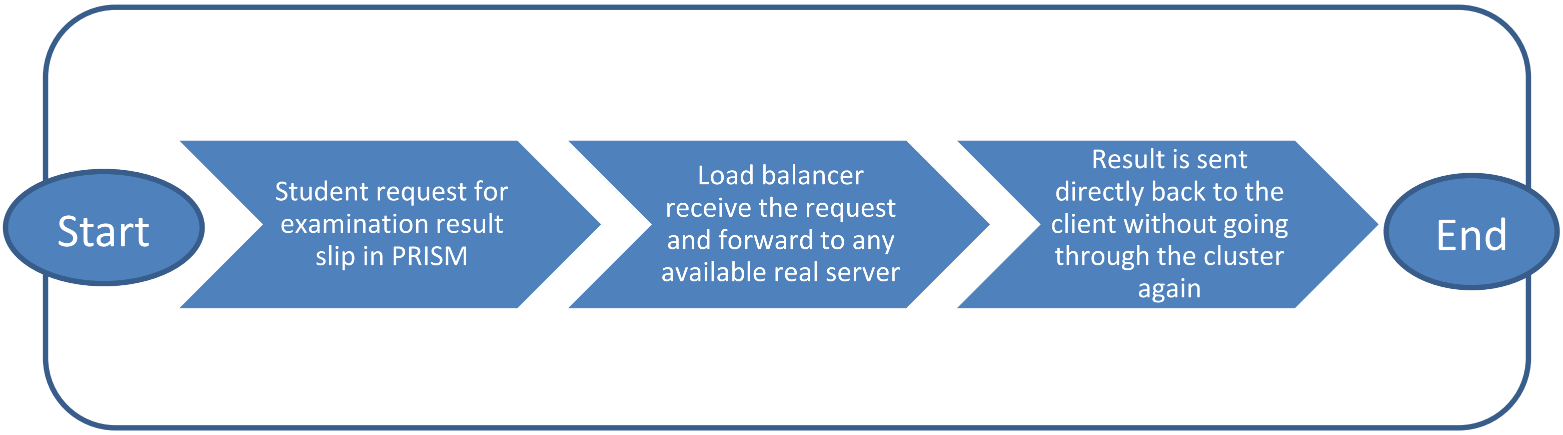

Start
Student request for examination result slip in PRISM
Load balancer receive the request and forward to any available real server
Result is sent directly back to the client without going through the cluster again
End

Figure 4.2.1 – Process Flow of cluster service

4.3 Five Node Cluster Configuration

To ease the development process later, I have define the name and IP Address for each of the servers in the system together with the Ethernet segment IP attached to each server (Real/Virtual). Note that the server's entire Ethernet IP configuration is set in the same subnet to adhere to the requirement of the direct routing design.

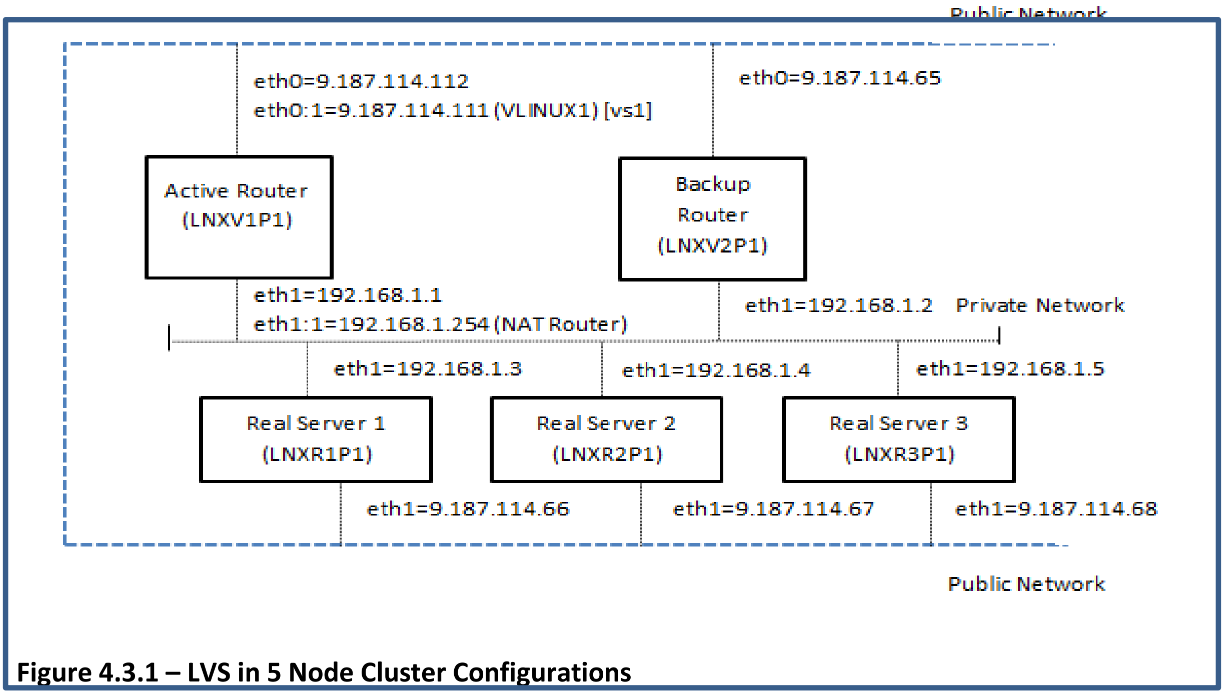


Figure 4.3.1 – LVS in 5 Node Cluster Configurations

Referring to the diagram above, a simple table created below giving more detailed information and better understanding of the server's characteristics that is going to be developed.

Server' Hostname/Alias	IP Address(Virtual/Physical)	Services Enabled
Linux Server 1 (LNXV1P1)	192.168.0.109v	IP Load Balancer, Heartbeat
Linux Server 2 (LNXV2P1)	192.168.0.110v	IP Load Balancer, Heartbeat
Linux Server 3 (LNXR1P1)	192.168.0.211p	Apache, SBD, DB2
Linux Server 4 (LNXR2P1)	192.168.0.212p	Apache, SBD, DB2
Linux Server 5 (LNXR3P1)	192.168.0.213p	Apache, SBD, DB2

Figure 4.3.2 – Table: Detailed Information on the list of servers

4.4 Service Running in Cluster

Service	
Name	Description
Heartbeat	The heartbeat portion is brought up between the 2 virtual servers where the cluster configuration will be done. The purpose of the heartbeat is to determine the status or health of the other nodes in the cluster by exchanging message in a time interval. If a node fail to respond in the interval, the node will be considered dead and brought out of the cluster service until it respond back to the heartbeat service.
Storage Based Death (SBD) Device	The SBD Device function is to protect the integrity of data in a shared storage by preventing uncoordinated concurrent access to the data storage. Prevent Split Brain Scenario: network or software malfunction causing several coordinators to be elected at the same time that can cause corruption of data.
Clustered Samba Server	A multiple file server that provides shared file and print services, a standard on nearly all distributions of Linux and is commonly included as a basic system service. Enables interoperability between Linux/Unix servers and Windows clients. Example: accessing Linux folders from windows.
LVS IP Load Balancing	Load Balance all workloads equally among all active real server in the cluster. Monitored by Ldirectord, load balancer daemon that runs on the virtual server. All requests will be forwarded from the load balancer to the real servers that are providing the service.
Clustered Apache Server	Web Server with high availability to hold the dummy PRISM website. Uses virtual IP address monitored by heartbeat. Must have the main web page to be displayed in all the nodes. If one of the nodes goes down, the IP address and the apache resource will failover to the other nodes.

Steps to set up the resources:

- Install and activate the resources manually
- Set up the resources to the cluster
- Register the resources to the Pacemaker GUI for monitoring
- Set constraints and order for all resources.

4.5 Requirement Gathering: Problem Statement

In order to check whether the UTP Student Portal poor availability problem truly exists, a simple interview was conducted among 30 ICT Final Year Students to check if they have encountered this problem before throughout their stay in UTP. The results are as shown:

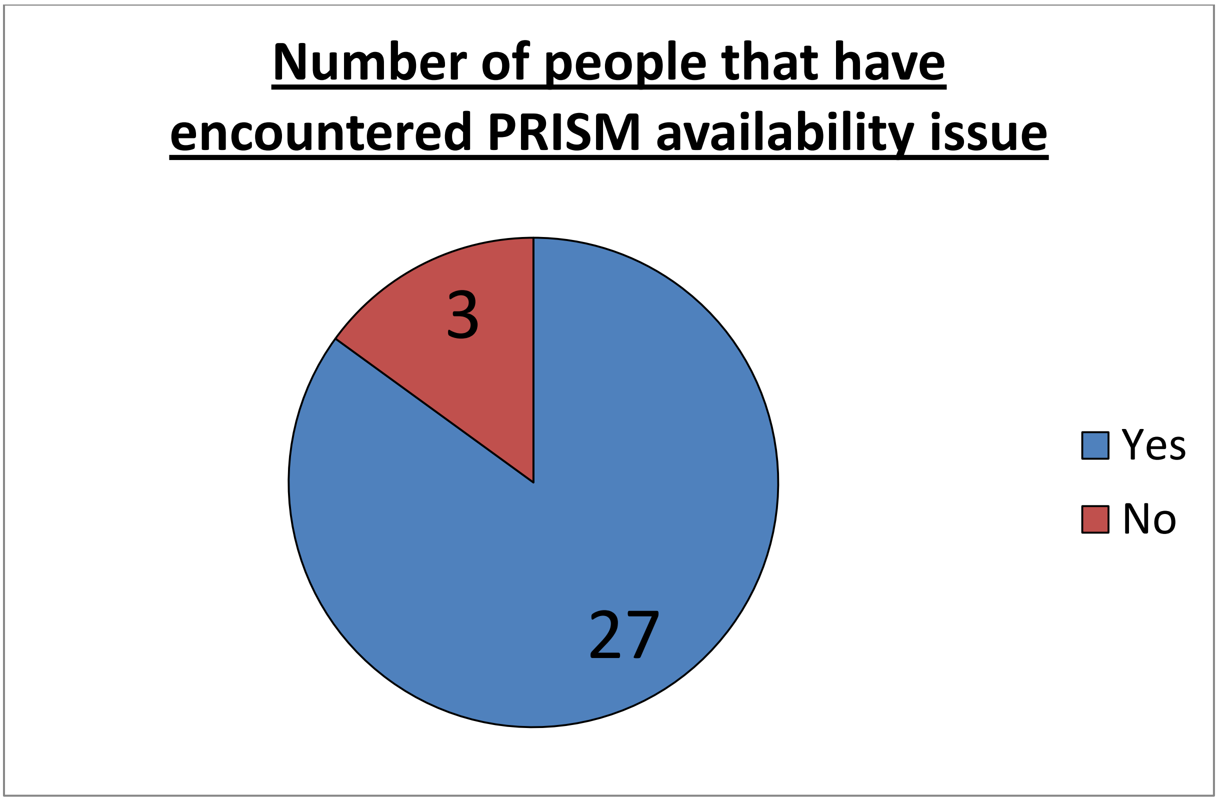


Figure 4.5.1 – Interview Result (Pie Chart)

From the result, 85% of the respondent agreed that they have encountered the PRISM poor availability issue during course registration or when obtaining their final year result. This shows that this problem truly exists and still it is unsolved.

5.0 Recommendation (FYP1)

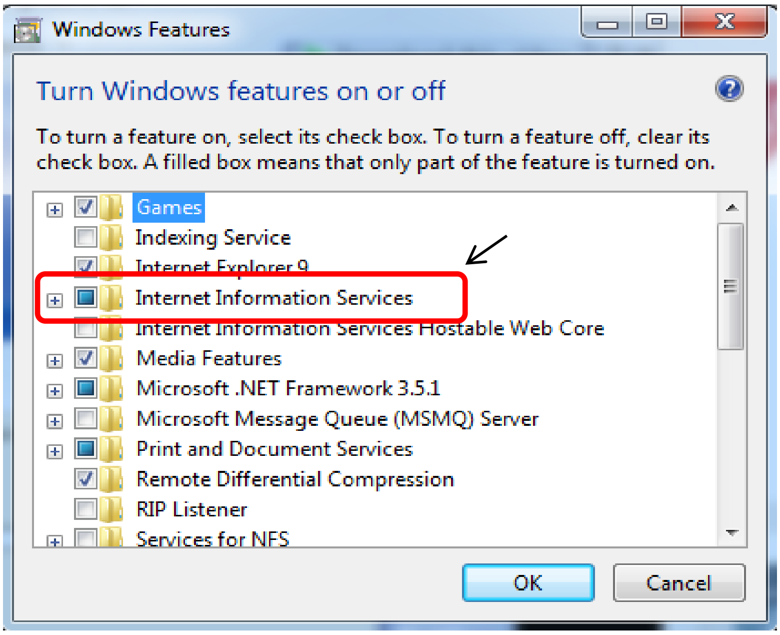
5.1 Next Step

- Continue research on LVS architecture and how to bring up cluster services.
- Create a Prototype website to simulate the current PRISM webpage with integration to database inside the cluster.
- Discuss and keep updated with the supervisor's advice during development.
- Gather feedback and comments from student after implementation.
- Prepare a technical paper for future maintenance of the cluster system.
- Create a Linux Script to test the Apache Servers and rectify that the overall project objective is obtained, improved availability of PRISM.

6.0 VMware Environment Setup Steps

6.1 VMware Installation Steps

- Register for an account on http://www.vmware.com/ to allow download and obtain serial number.
- Download VMware Server 2.0.2 from VMware official download site http://downloads.vmware.com/d/details/server202/dCpiZCVqdGJkZWRq
- Enable Internet Information Services on Windows before executing setup program or else it would fail.



Windows Features

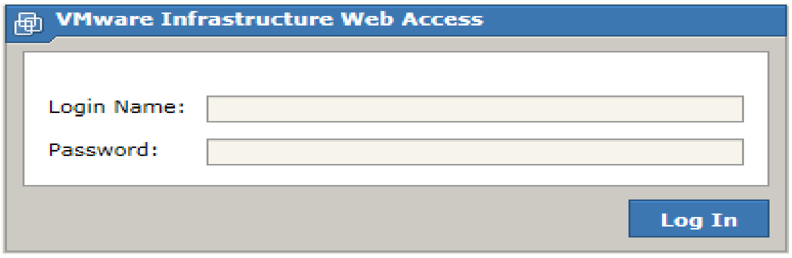
Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ☑ Games
- □ Indexing Service
- ☑ Internet Explorer 9
- ■ Internet Information Services
- □ Internet Information Services Hostable Web Core
- ☑ Media Features
- ■ Microsoft .NET Framework 3.5.1
- □ Microsoft Message Queue (MSMQ) Server
- ■ Print and Document Services
- ☑ Remote Differential Compression
- □ RIP Listener
- □ Services for NFS

Figure 6.1.1 – Enabling Internet Information Services on Windows

- Run VMware Setup, specify setup directory and wait for the installation to finish.
- Open VMware Server Home Page that is automatically placed on the desktop.



VMware Infrastructure Web Access

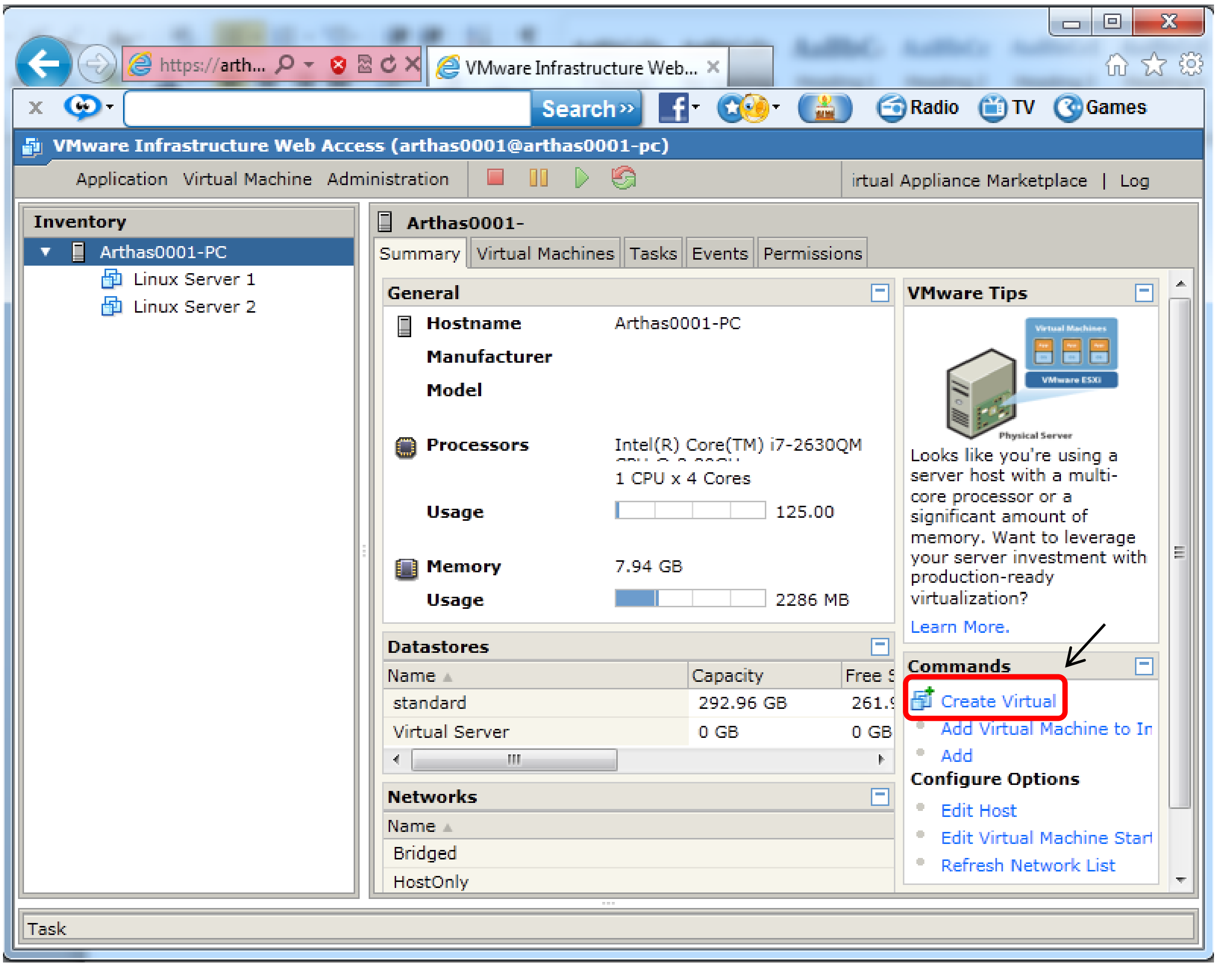
Login Name:

Password:

Figure 6.1.2 – VMware Server Login Page

- Fill the login form just like in Figure 6.1.2 using the username and password of the current computer user account.

- Once logged in, create 2 virtual machines to place the SUSE Linux Enterprise Server 11(SLES11) as shown in Figure 6.1.3.



General

Hostname: Arthas0001-PC
Manufacturer:
Model:
Processors: Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz
1 CPU x 4 Cores
Usage: 125.00
Memory: 7.94 GB
Usage: 2286 MB

Looks like you're using a server host with a multi-core processor or a significant amount of memory. Want to leverage your server investment with production-ready virtualization?
Learn More.

Datastores

Name ▲	Capacity	Free S
standard	292.96 GB	261.9
Virtual Server	0 GB	0 GB

Networks

Name ▲
Bridged
HostOnly

Figure 6.1.3 – Creating Virtual Machines inside the VMware

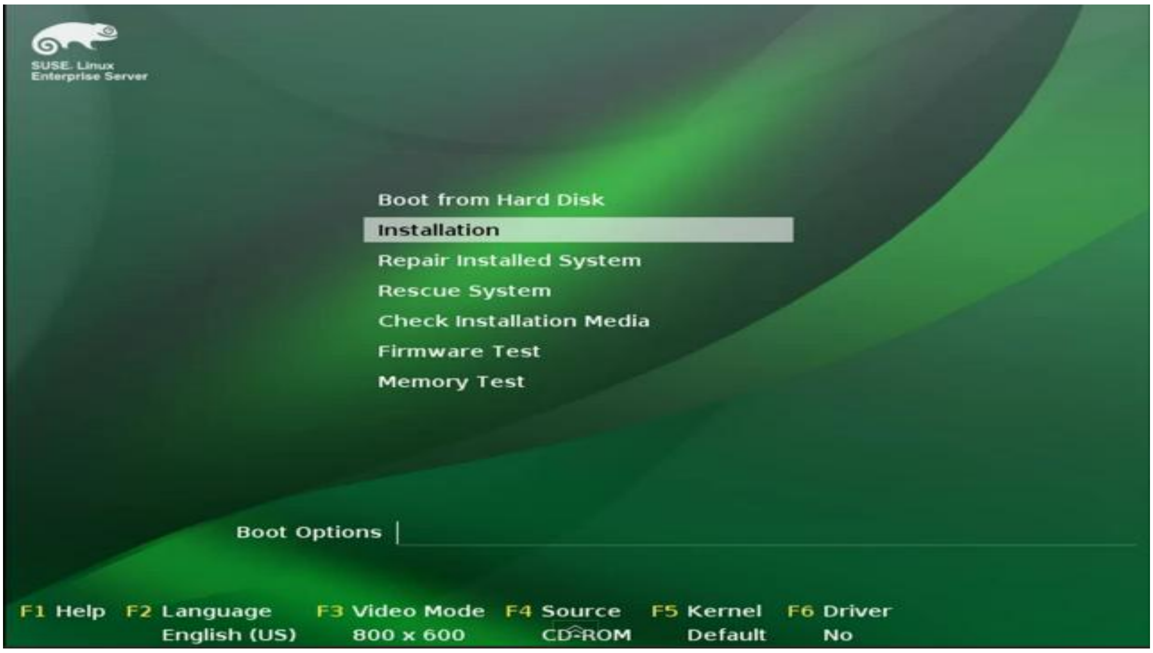
- Follow the Virtual Machine(VM) installation steps according to self-specifications:
  - Name: Linux Server 1 & Linux Server 2
  - Datastore: Standard(automatically generated)
  - Operating System: Linux OS. Version: SUSE Linux Enterprise Server(32-Bit)
  - Memory Size: 512MB
  - Processor Count: 2
  - Virtual HD size: 290GB
  - Network Connection: Bridged
  - Host Media: ISO image(SLES-11-SP1-DVD-i586-GM-DVD1.iso)

    place the ISO image on your Datastore first

- Add USB Controller, Finish installation.(repeat step for 2nd VM)

6.2 SLES 11 Installation Steps

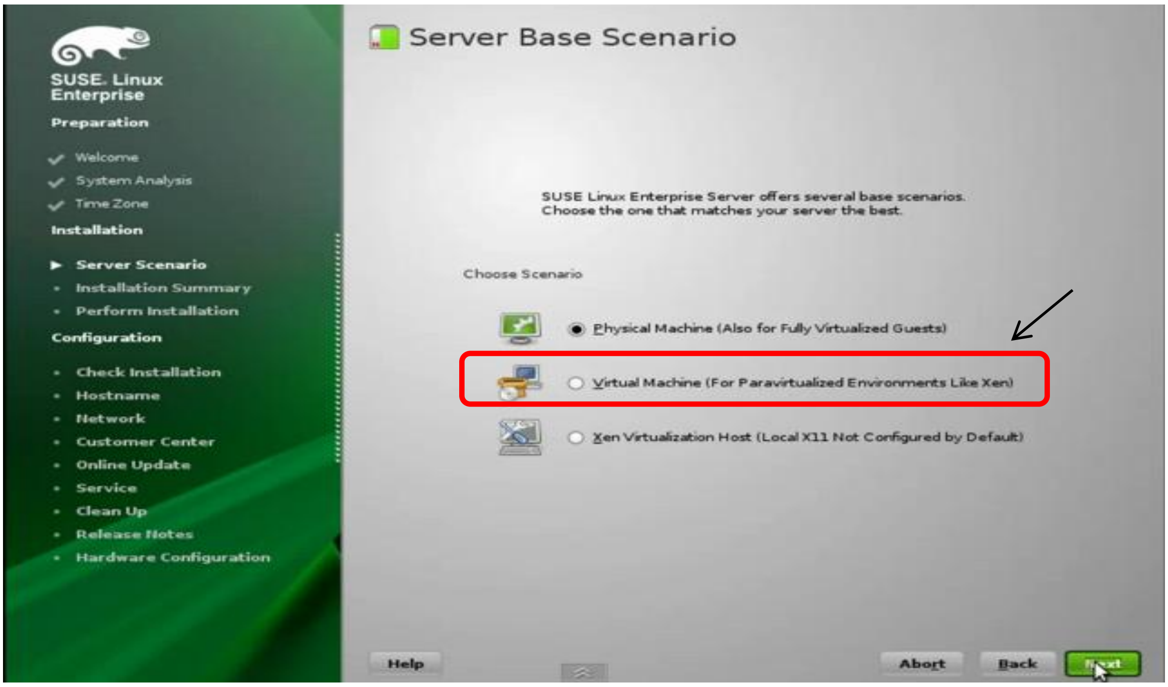
- Power up any of the two VMs and open the console, SLES 11 ISO will be automatically booted up. Repeat all the steps later for the other VM.



F2 Language English (US)
F3 Video Mode 800 x 600
F4 Source CD-ROM
F5 Kernel Default
F6 Driver No

Figure 6.2.1 - Boot screen for SLES 11

- Enter the installation option as shown in Figure 6.2.1 to start or use check installation media option to be sure.
- Proceed through the preparation phase and fill it with your preferences.



Server Base Scenario

SUSE Linux Enterprise Server offers several base scenarios. Choose the one that matches your server the best.

Choose Scenario

● Physical Machine (Also for Fully Virtualized Guests)

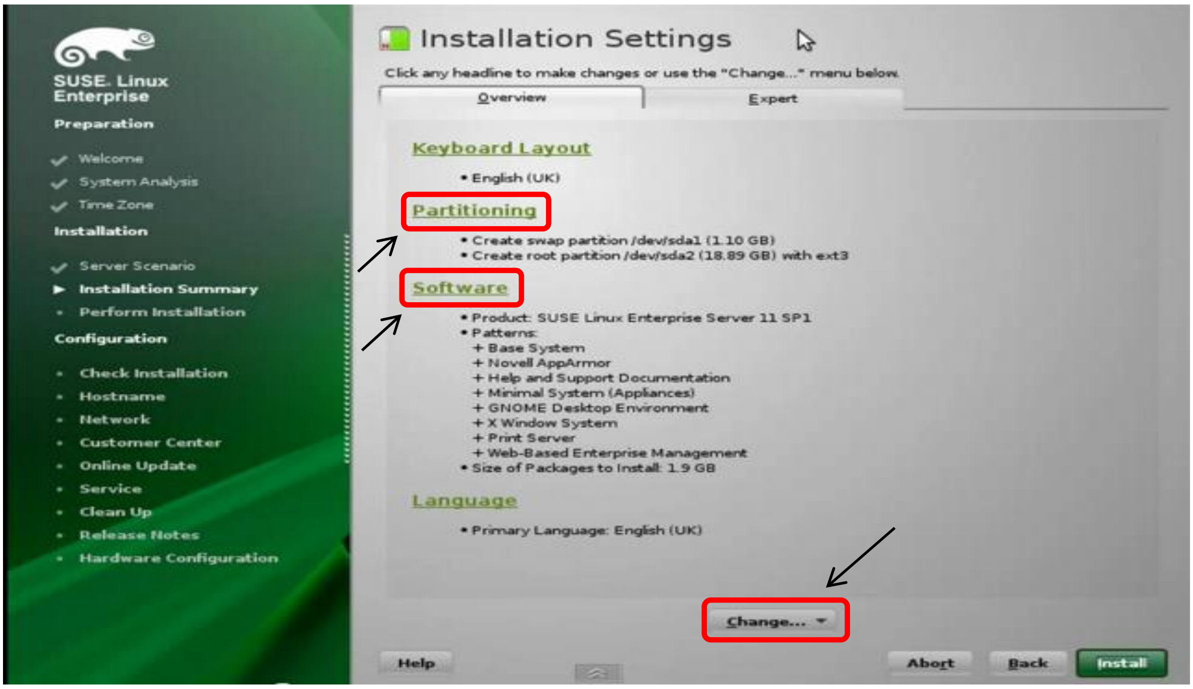
○ Virtual Machine (For Paravirtualized Environments Like Xen)

○ Xen Virtualization Host (Local X11 Not Configured by Default)

Figure 6.2.2 - First section of the installation phase

- (*) Pick the 2nd option, Virtual Machine (For Paravirtualized Environments Like Xen) as the OS will be installed on a VM.

- Click Next to open the installation summary and any changes is possible using the drop down menu or clicking on any headlines as shown in figure 6.2.3.



Installation Settings

Click any headline to make changes or use the "Change..." menu below.

Keyboard Layout

- English (UK)

Partitioning

- Create swap partition /dev/sda1 (1.10 GB)
- Create root partition /dev/sda2 (18.89 GB) with ext3

Software

- Product: SUSE Linux Enterprise Server 11 SP1
- Patterns:
  - + Base System
  - + Novell AppArmor
  - + Help and Support Documentation
  - + Minimal System (Appliances)
  - + GNOME Desktop Environment
  - + X Window System
  - + Print Server
  - + Web-Based Enterprise Management
- Size of Packages to Install: 1.9 GB

Language

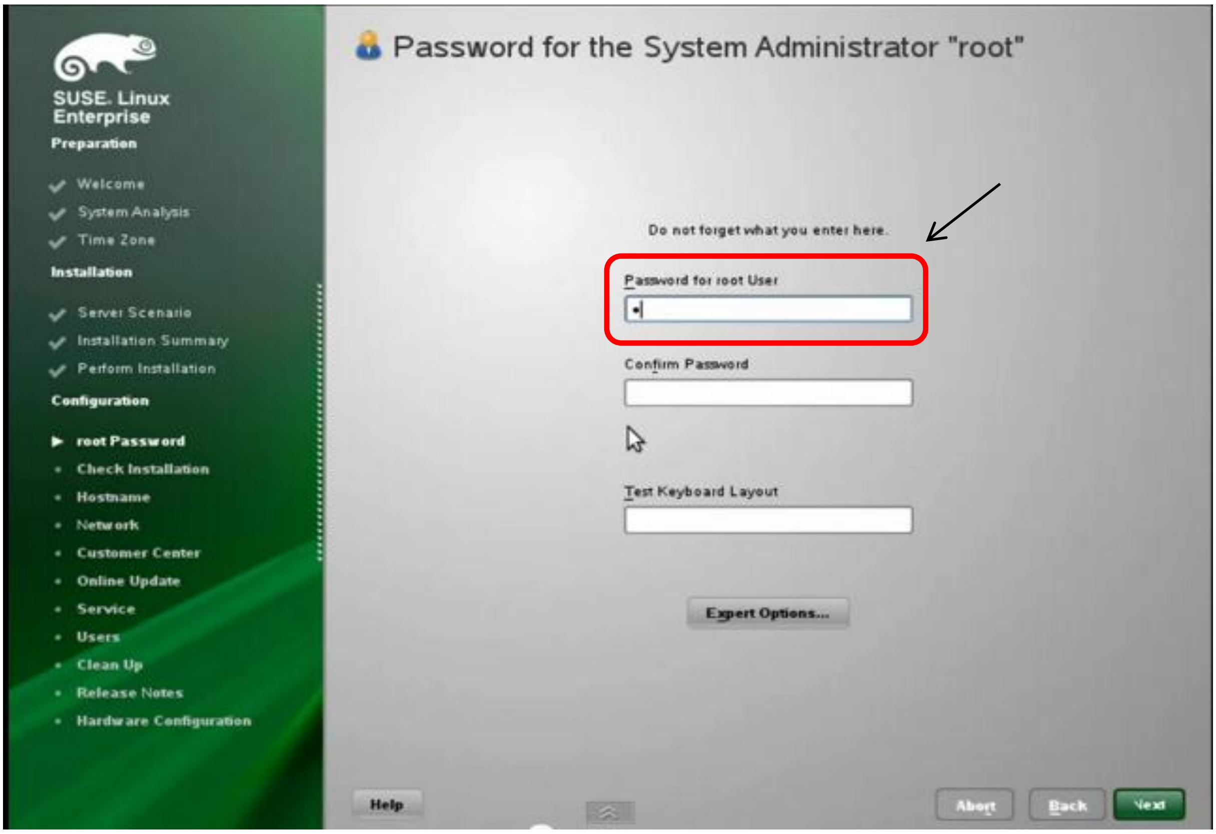
- Primary Language: English (UK)

Figure 6.2.3 - SLES 11 Installation Overview

- After all changes, press the install button. It will take a while to perform the installation approximately 15-20 minutes depending on your computer capability.

- Once completed, user is prompted to enter the root password and confirm it like in Figure 6.2.4

- Do not forget this password, as root profile is important for future installation steps and certain actions can only be done while logging in the root account.



Password for the System Administrator "root"

Do not forget what you enter here.

Password for root User

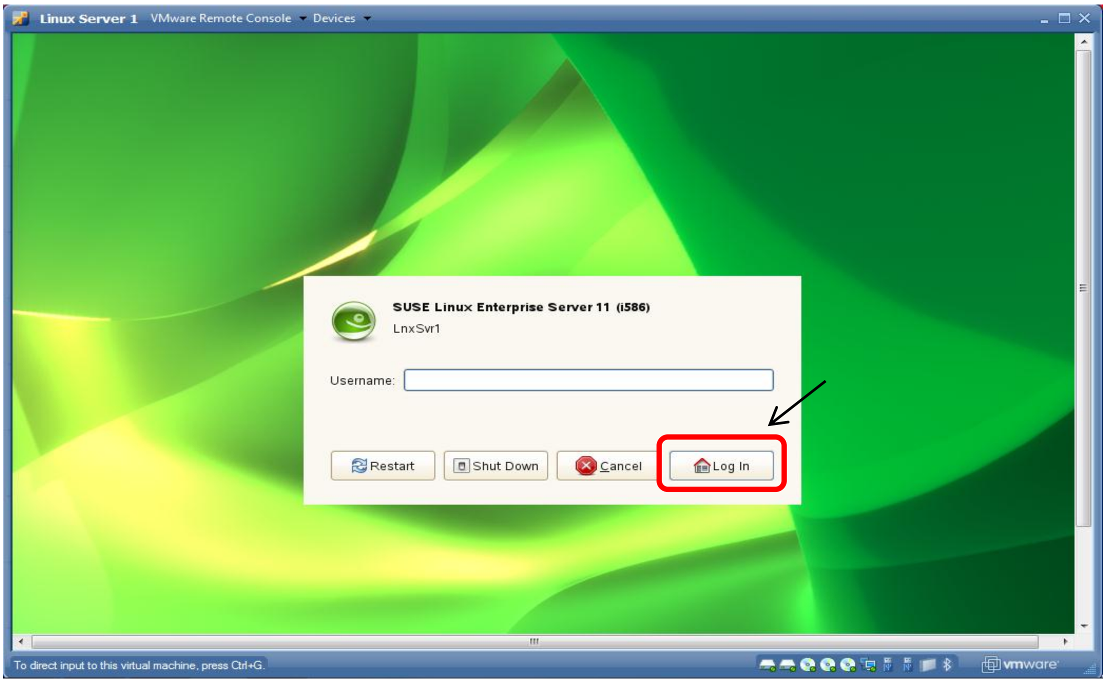
Confirm Password

Test Keyboard Layout

Figure 6.2.4 – Assigning password to System Administrator (root)

- The fields on the next steps can be modified and changed later after installation according to user situation thus it is not so important to fill it perfectly now.
  - Assign Hostname and Domain name
  - Network Configuration: Network Card Setup, Network Services
  - User Authentication Method: Local(Create another local user profile)
  - Hardware configuration (use following configuration after auto detect)
  - Untick the box “Clone this system for AutoYasST”

- After selecting finish, the system starts and the graphical login screen is displayed (see Figure 6.2.5) where you can log in using the user account you created during the installation or using the root account.



Linux Server 1
VMware Remote Console
Devices
SUSE Linux Enterprise Server 11 (i586)
LnxSvr1
Username:
Restart
Shut Down
Cancel
Log In
To direct input to this virtual machine, press Ctrl+G.
vmware

Figure 6.2.5 - SLES 11 Login Screen

6.3 High Availability (HA) Extension Installation Steps

- The High Availability extension for the SUSE Linux is stored inside an .iso file that needs to be added first into the Virtual Machine before it can access it for installation.

- Add the installation media by locating the HA extension ISO in the VM's Datastore and mount them to the VM's disk drive as shown in Figure 6.3.1. (* can only be done while the virtual machine is stopped *)

- The next menu will prompt for the Host Media. Choose the Use an ISO option and locate the directory of the ISO in the Datastore to complete the mounting.

- Power up the VM and login to the OS as system administrator (root). Right Click the screen and select ‘Open in Terminal’ to open the Linux Command Line Interface.

- Then, open the YaST GUI using the command “yast2 &”. Following that, select Software > Software Repositories, add all the required DVD Disk Image like in Figure 6.3.2.

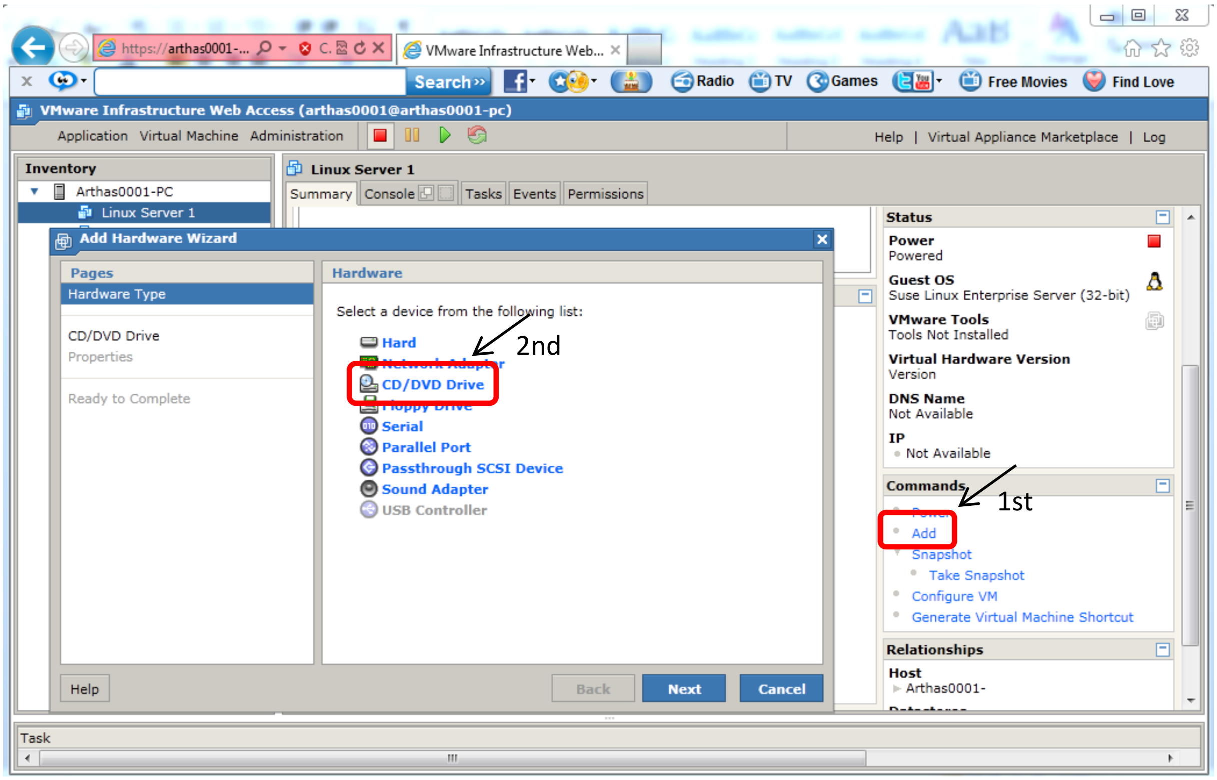


Figure 6.3.1 – Adding DVD Drive to Virtual Machine

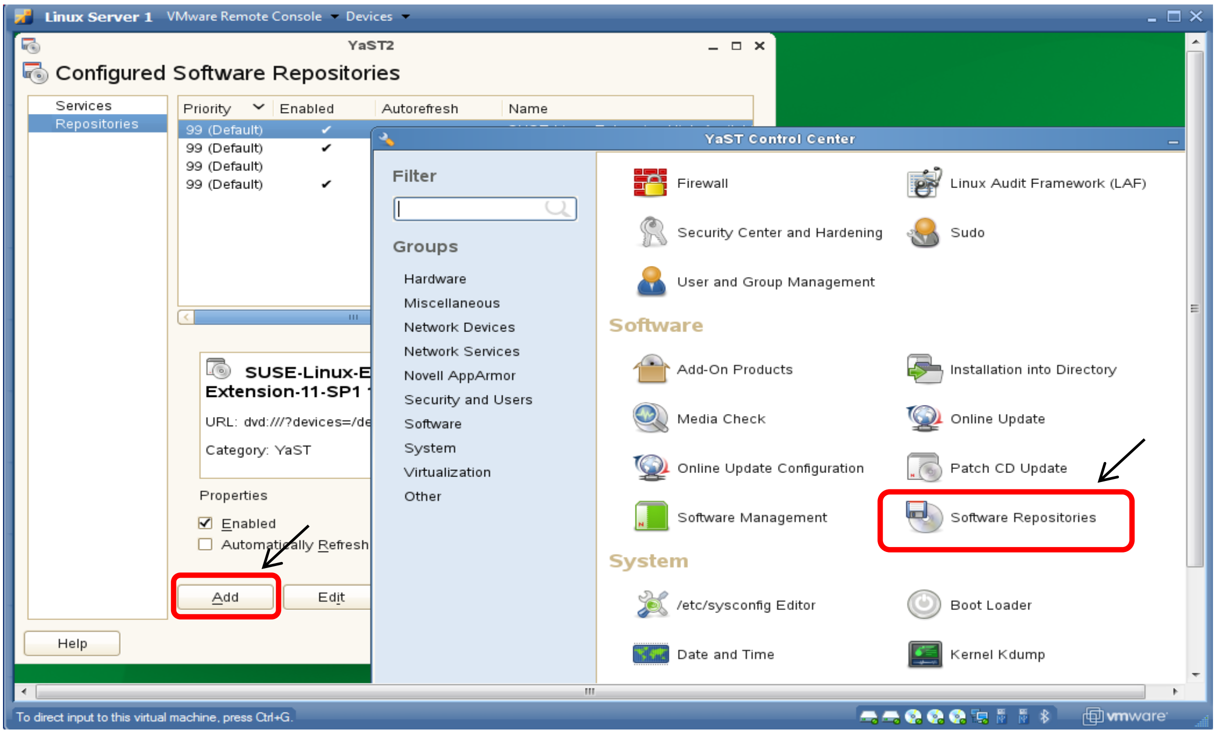
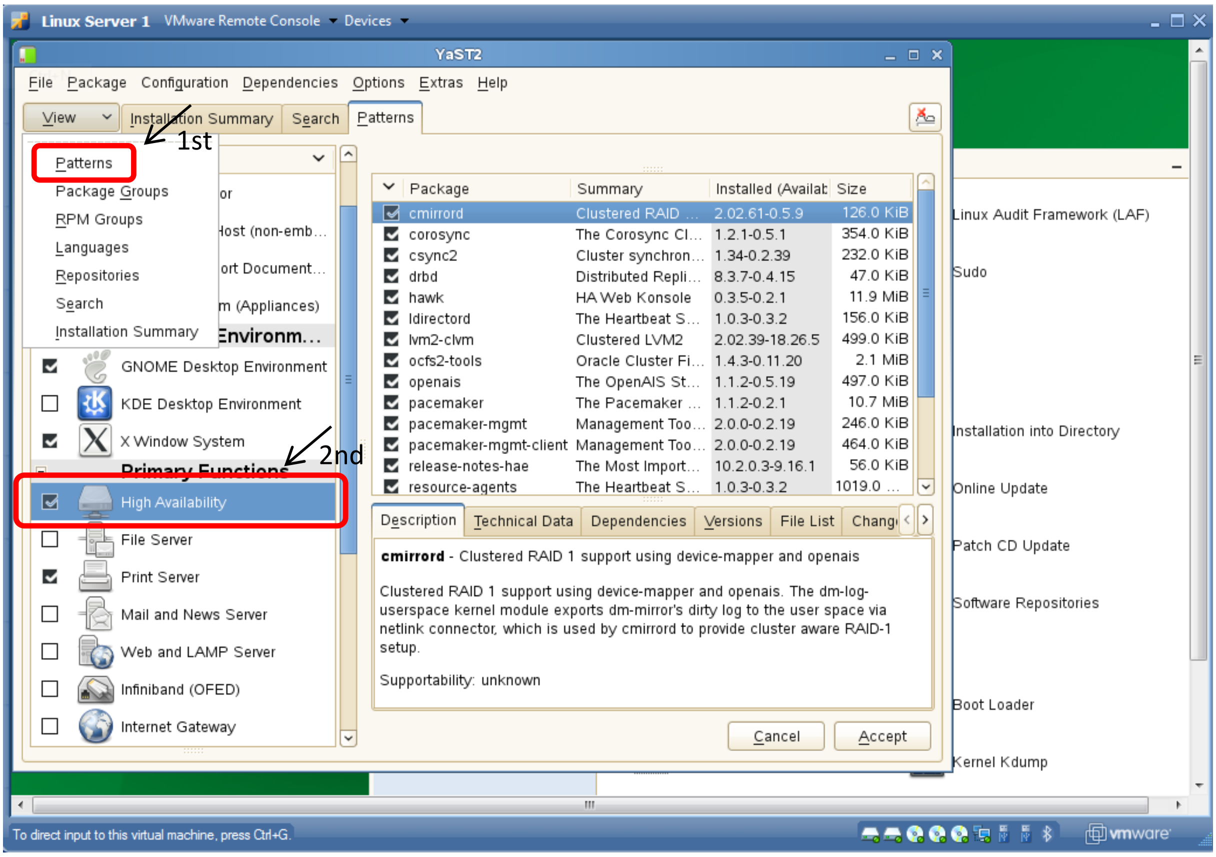


Figure 6.3.2 – Configure Software Repositories for Linux HA Extension

- The next step is to enter the Software Management under Software in YaST to start installation. Select View>Patterns to open the software patterns installation. Locate and tick the High Availability segment as shown in Figure 6.3.3.



	Package	Summary	Installed (Availat	Size
☑	cmirrord	Clustered RAID …	2.02.61-0.5.9	126.0 KiB
☑	corosync	The Corosync Cl…	1.2.1-0.5.1	354.0 KiB
☑	csync2	Cluster synchron…	1.34-0.2.39	232.0 KiB
☑	drbd	Distributed Repli…	8.3.7-0.4.15	47.0 KiB
☑	hawk	HA Web Konsole	0.3.5-0.2.1	11.9 MiB
☑	ldirectord	The Heartbeat S…	1.0.3-0.3.2	156.0 KiB
☑	lvm2-clvm	Clustered LVM2	2.02.39-18.26.5	499.0 KiB
☑	ocfs2-tools	Oracle Cluster Fi…	1.4.3-0.11.20	2.1 MiB
☑	openais	The OpenAIS St…	1.1.2-0.5.19	497.0 KiB
☑	pacemaker	The Pacemaker …	1.1.2-0.2.1	10.7 MiB
☑	pacemaker-mgmt	Management Too…	2.0.0-0.2.19	246.0 KiB
☑	pacemaker-mgmt-client	Management Too…	2.0.0-0.2.19	464.0 KiB
☑	release-notes-hae	The Most Import…	10.2.0.3-9.16.1	56.0 KiB
☑	resource-agents	The Heartbeat S…	1.0.3-0.3.2	1019.0 …

cmirrord - Clustered RAID 1 support using device-mapper and openais

Clustered RAID 1 support using device-mapper and openais. The dm-log-userspace kernel module exports dm-mirror's dirty log to the user space via netlink connector, which is used by cmirrord to provide cluster aware RAID-1 setup.

Supportability: unknown

Figure 6.3.3 – Installing High Availability Extension in Software Management screen.

- Click Accept and the installation process will take over. Restart the VM for the installation changes to take effect.

6.4 Transfer Configurations to all Nodes

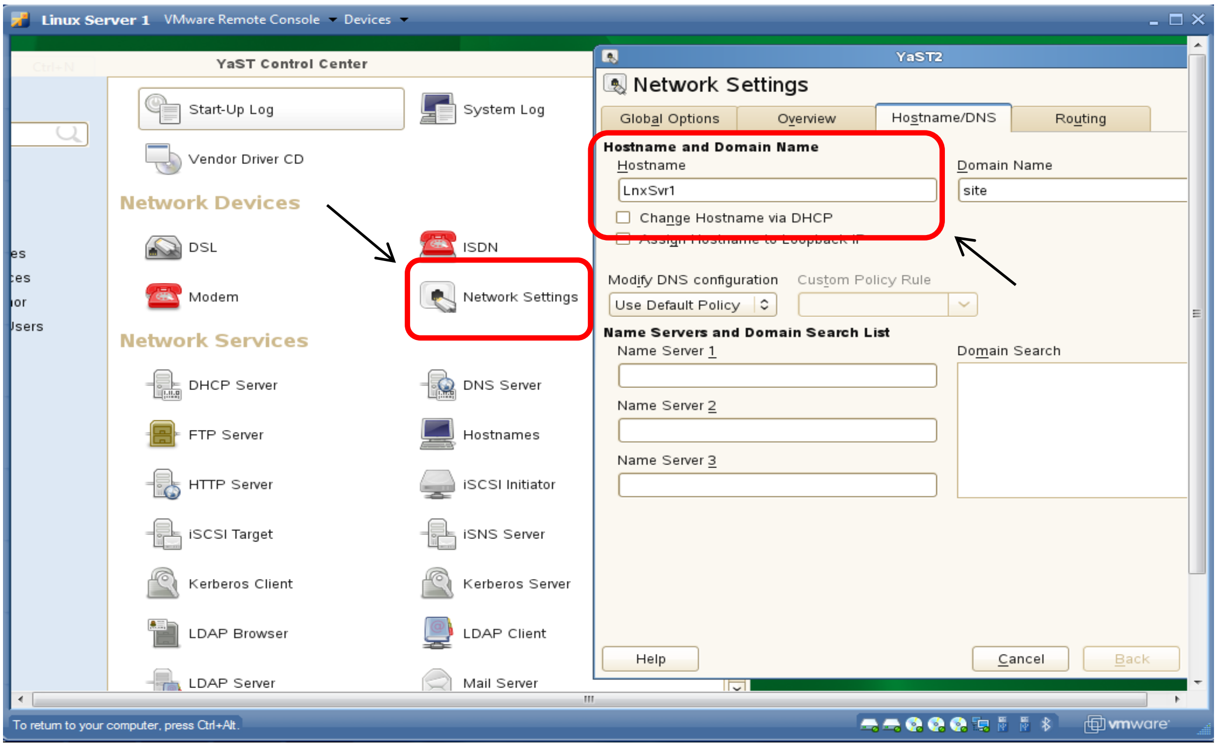
- This step is required so that both of the Linux Servers have the same configuration and that they are aware of the existence of one another using their IP Address so that they can communicate inside the cluster.

- There are 2 ways of synchronizing the configuration files of the servers inside the cluster. Instead of copying the resulting configuration files to all nodes manually, use the csync2 tool for replication across all nodes in the cluster. Csync2 can handle any number of hosts, sorted into synchronization groups.

- Firstly, we need to define the hostname for all the nodes so that Csync2 can recognize them using their exact name. In the YaST Network Devices Module, go to Networks

Setting and change the node's hostname in the HostName/DNS tab as shown in Figure 6.4.1.

- You can either assign your own Hostname (recommended) or use the function Change Hostname via DHCP if you are using DHCP to configure it.



Network Settings

Hostname and Domain Name

Hostname: LnxSvr1

Domain Name: site

☐ Change Hostname via DHCP

☐ Assign Hostname to Loopback IP

Modify DNS configuration: Use Default Policy

Custom Policy Rule:

Name Servers and Domain Search List

Name Server 1:

Name Server 2:

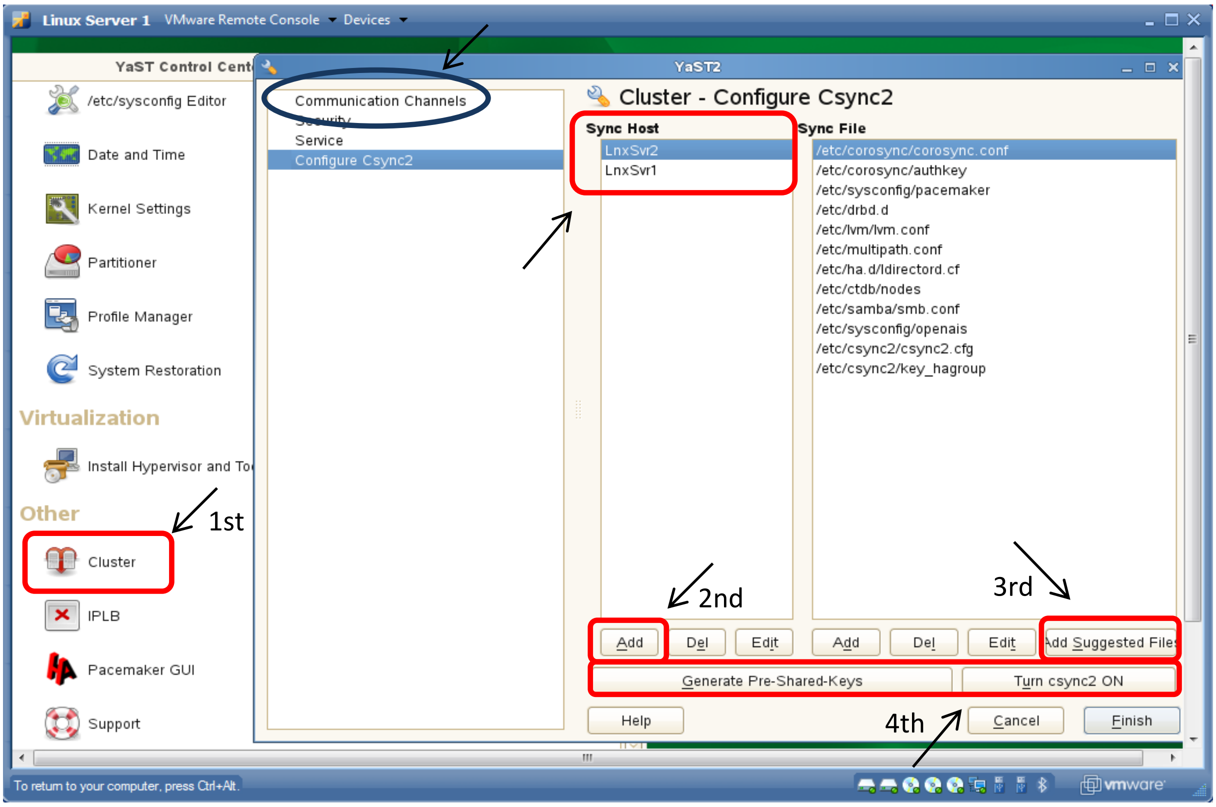
Name Server 3:

Domain Search:

Figure 6.4.1 – Configure Node's Hostname for Csync2

- Next, in the YaST cluster module, switch to the *Csync2* category. Fill your network preferences such as default IP Address and communication media type inside the communication channel tabs shown inside the blue circle in Figure 6.4.2.

- To specify the synchronization group, click Add in the Sync Host group and enter the local hostnames of all nodes in your cluster or follow the steps shown in Figure 6.4.2.



Cluster - Configure Csync2

Sync Host
LnxSvr2
LnxSvr1

Sync File
/etc/corosync/corosync.conf
/etc/corosync/authkey
/etc/sysconfig/pacemaker
/etc/drbd.d
/etc/lvm/lvm.conf
/etc/multipath.conf
/etc/ha.d/ldirectord.cf
/etc/ctdb/nodes
/etc/samba/smb.conf
/etc/sysconfig/openais
/etc/csync2/csync2.cfg
/etc/csync2/key_hagroup

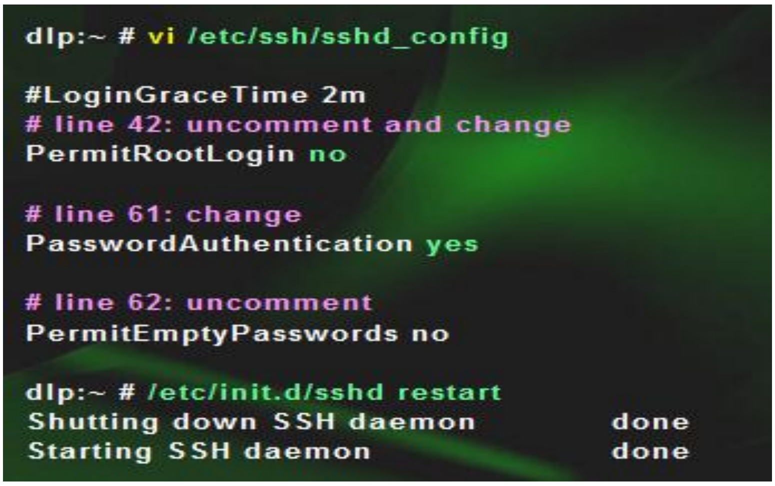
Figure 6.4.2 – Csync2 Customization & Adding Host/Node

- Click Generate Pre-Shared-Keys to create a key file for the synchronization group. Only with this key that they can recognize the members of the group. The key file is written and located to /etc/csync2/key_hagroup. After it has been created, it must be copied manually to all members of the cluster.

- To populate the Sync File list with the files that usually need to be synchronized among all nodes, click Add Suggested Files. Finally, activate Csync2 by clicking Turn Csync2 On. This will also start Csync2 automatically at boot time.

- However, not all files are automatically synchronized with the Csync2 function. Some files that are needed to be manually copied to all other nodes are */etc/csync2/csync2.cfg and /etc/csync2/key_hagroup.*

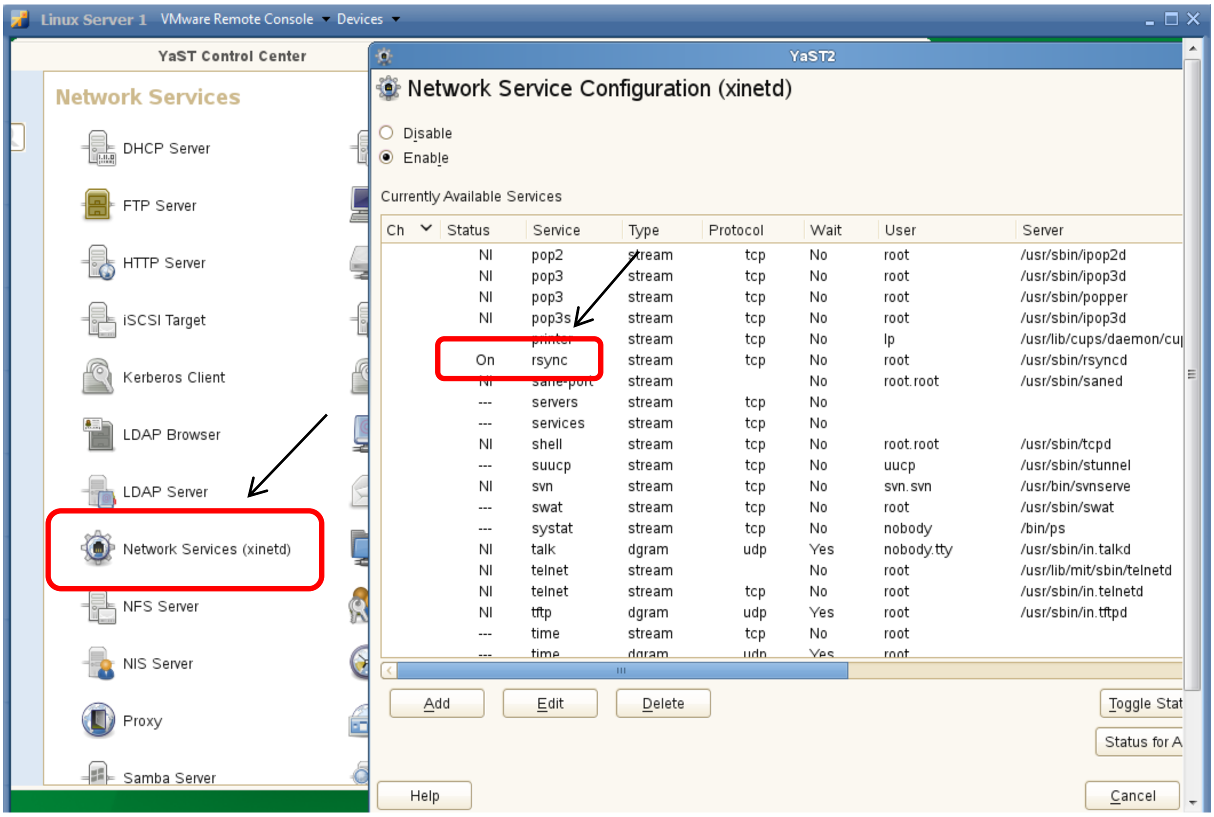
- In order to do so, we need to firstly modify our SSH configuration that is located inside */etc/ssh/sshd_config* as shown in Figure 6.4.3 to allow password authentication so that client computers can access them.



```
dlp:~ # vi /etc/ssh/sshd_config

#LoginGraceTime 2m
# line 42: uncomment and change
PermitRootLogin no

# line 61: change
PasswordAuthentication yes

# line 62: uncomment
PermitEmptyPasswords no

dlp:~ # /etc/init.d/sshd restart
Shutting down SSH daemon                done
Starting SSH daemon                     done
```

Figure 6.4.3 - SSH Configuration Change

- Enable Rsync in Network Services (xinetd) on both nodes. Screen showed in Figure 6.4.4.



Network Service Configuration (xinetd)

○ Disable
● Enable

Currently Available Services

Ch	Status	Service	Type	Protocol	Wait	User	Server
	NI	pop2	stream	tcp	No	root	/usr/sbin/ipop2d
	NI	pop3	stream	tcp	No	root	/usr/sbin/ipop3d
	NI	pop3	stream	tcp	No	root	/usr/sbin/popper
	NI	pop3s	stream	tcp	No	root	/usr/sbin/ipop3d
		printer	stream	tcp	No	lp	/usr/lib/cups/daemon/cu
	On	rsync	stream	tcp	No	root	/usr/sbin/rsyncd
	NI	sane-port	stream		No	root.root	/usr/sbin/saned
	---	servers	stream	tcp	No		
	---	services	stream	tcp	No		
	NI	shell	stream	tcp	No	root.root	/usr/sbin/tcpd
	---	suucp	stream	tcp	No	uucp	/usr/sbin/stunnel
	NI	svn	stream	tcp	No	svn.svn	/usr/bin/svnserve
	---	swat	stream	tcp	No	root	/usr/sbin/swat
	---	systat	stream	tcp	No	nobody	/bin/ps
	NI	talk	dgram	udp	Yes	nobody.tty	/usr/sbin/in.talkd
	NI	telnet	stream		No	root	/usr/lib/mit/sbin/telnetd
	NI	telnet	stream	tcp	No	root	/usr/sbin/in.telnetd
	NI	tftp	dgram	udp	Yes	root	/usr/sbin/in.tftpd
	---	time	stream	tcp	No	root	
	---	time	dgram	udp	Yes	root	

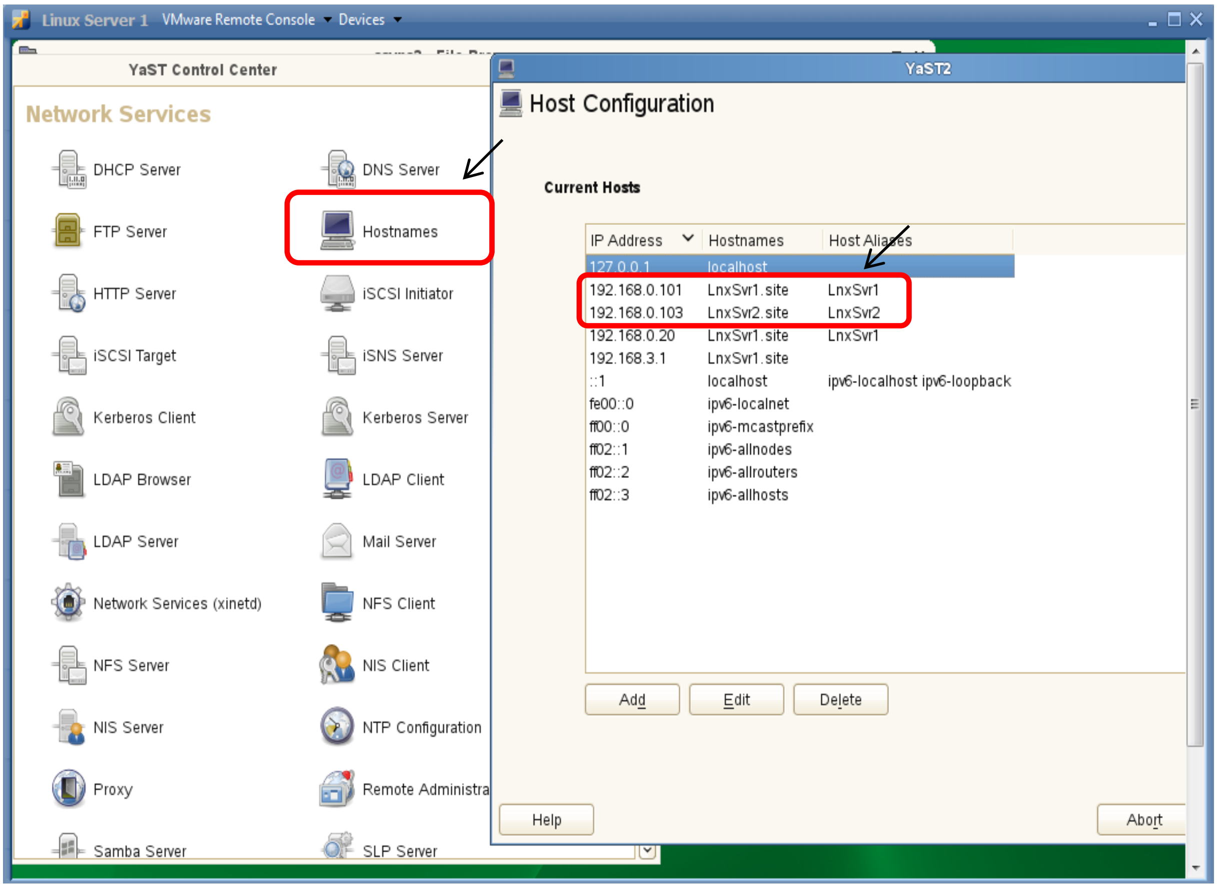
Figure 6.4.4 – Enabling Rsync Services.

i. Disable Firewall on both nodes as unexpected errors can occur. Execute any of the following lines in the terminal.

   - Push(server) - *rsync –av /etc/csync2 root@192.168.0.103:/etc * if you are using destination IP* or

- Pull(client) – *rsync –av –e ssh root@192.168.0.101:/etc/csync2 /etc * if you are using server IP**

ii. Use *csync2 -xv* to synchronize all the files at once. Make sure that csync2 service is running in all nodes by checking the network service configuration. In case of *unable to 'resolve peername'* error. Check the hostnames configured in both nodes and include each node's exact IP Address and hostnames as shown in Figure 6.4.5.



IP Address	Hostnames	Host Aliases
127.0.0.1	localhost	
192.168.0.101	LnxSvr1.site	LnxSvr1
192.168.0.103	LnxSvr2.site	LnxSvr2
192.168.0.20	LnxSvr1.site	LnxSvr1
192.168.3.1	LnxSvr1.site	
::1	localhost	ipv6-localhost ipv6-loopback
fe00::0	ipv6-localnet	
ff00::0	ipv6-mcastprefix	
ff02::1	ipv6-allnodes	
ff02::2	ipv6-allrouters	
ff02::3	ipv6-allhosts	

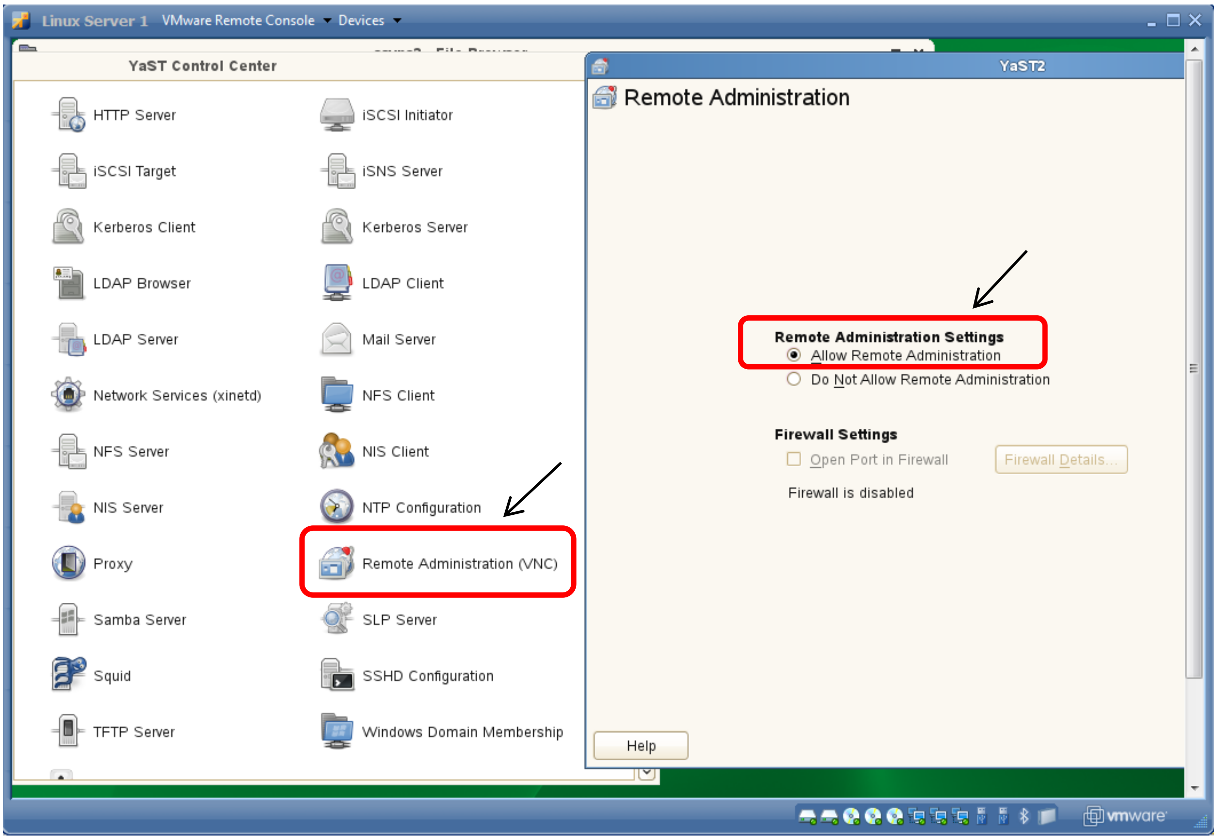
Figure 6.4.5 – Hostname Configuration

6.5 Administration Settings

6.5.2 Setup remote administration using VNC viewer or PuTTY

- To allow remote administration for the linux machine instead of direct accessing some steps are needed.

- With SSH already configured in the previous step, the next step is to enable the remote configuration settings as shown in Figure 6.5.1.1.



Remote Administration

Remote Administration Settings

● Allow Remote Administration

○ Do Not Allow Remote Administration

Firewall Settings

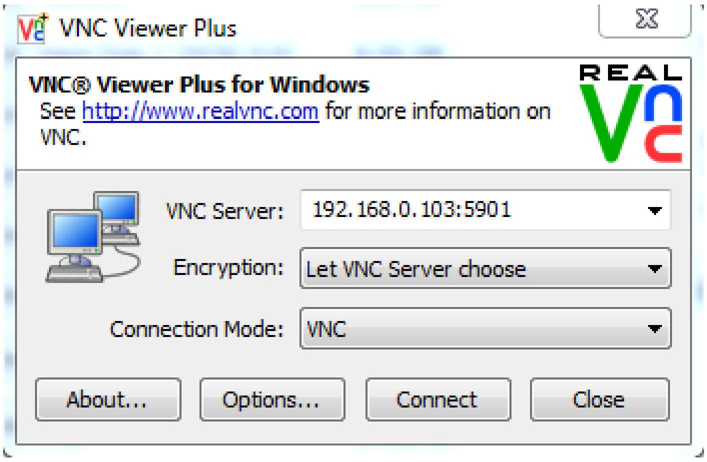
□ Open Port in Firewall

Firewall is disabled

Figure 6.5.1.1 – Enabling Remote Administration

- Disable firewall before running starting remote access as it can cause problems. Two methods of remote administration:

  i. Download free VNC viewer from its official website. Run the program with target machine's IP Address and default port 5901. Example in Figure 6.5.1.2.



VNC® Viewer Plus for Windows

See http://www.realvnc.com for more information on VNC.

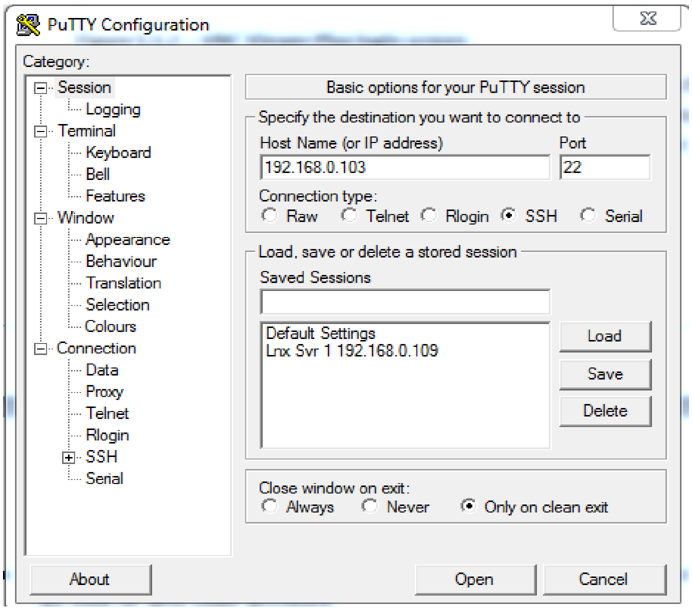
VNC Server: 192.168.0.103:5901

Encryption: Let VNC Server choose

Connection Mode: VNC

Figure 6.5.1.2 – VNC Viewer Plus login screen

ii. Or download PuTTY, a free telnet/SSH client from its official website www.putty.org. Input the targeted Linux’s IP Address and default port 22 or save the configuration for future use as shown in Figure 6.5.1.3.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.0.103

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
Lnx Svr 1 192.168.0.109

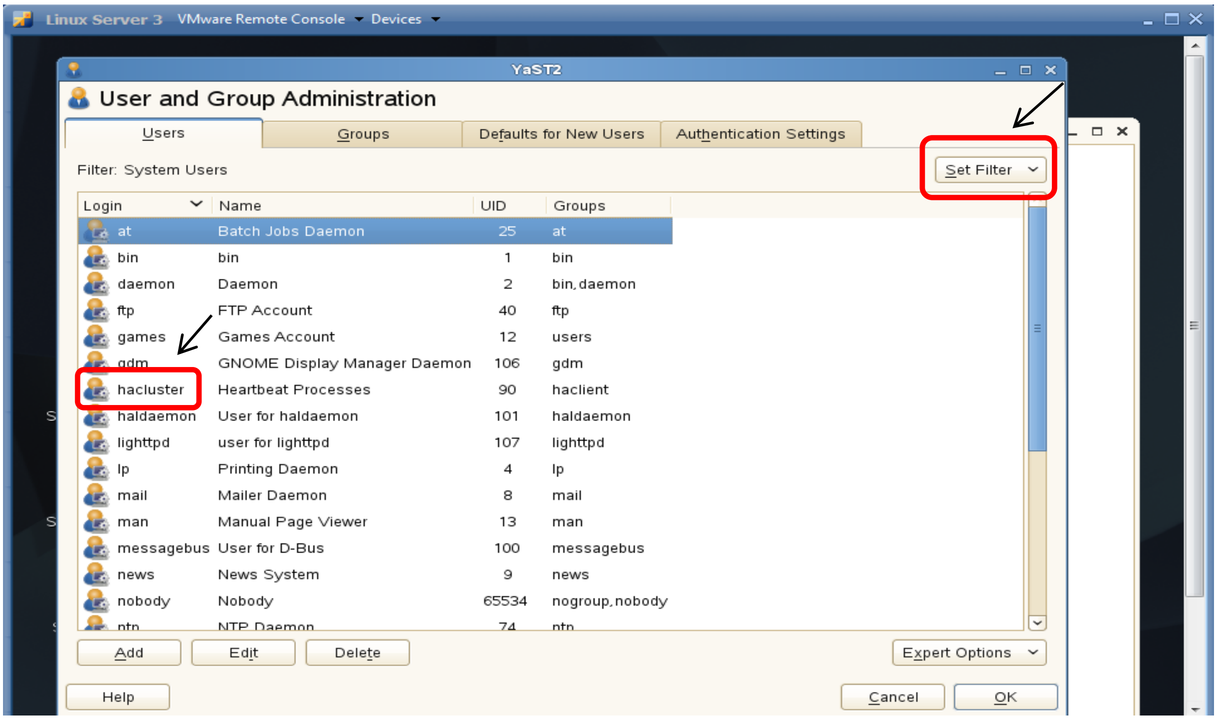
Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 6.5.1.3 – PuTTY Login Screen

- If successful, a screen appears requesting username and password for the Linux. Login as usual as root or any user account.

6.5.2 Using the Pacemaker GUI

- Before managing the cluster resources using the pacemaker GUI, a password for the user, *hacluster* need to be set. From the yast2, go to user and group administration section.
- Set the filter list to **system user**, locate the *hacluster* user and double click on it to set the password as shown in figure 6.5.2.1



Filter: System Users

Login	Name	UID	Groups
at	Batch Jobs Daemon	25	at
bin	bin	1	bin
daemon	Daemon	2	bin,daemon
ftp	FTP Account	40	ftp
games	Games Account	12	users
gdm	GNOME Display Manager Daemon	106	gdm
hacluster	Heartbeat Processes	90	haclient
haldaemon	User for haldaemon	101	haldaemon
lighttpd	user for lighttpd	107	lighttpd
lp	Printing Daemon	4	lp
mail	Mailer Daemon	8	mail
man	Manual Page Viewer	13	man
messagebus	User for D-Bus	100	messagebus
news	News System	9	news
nobody	Nobody	65534	nogroup,nobody
ntp	NTP Daemon	74	ntp

Figure 6.5.2.1 – Set password for Hacluster user

- After bringing up the cluster using rcopenais start, the pacemaker GUI is now accessible using the password with hacluster as the user in the login screen.

6.6 Adding SBD Device/Enable STONITH

- There are two kinds of fencing: resource level and node level.
- Using the resource level fencing the cluster can make sure that a node cannot access one or more resources. One typical example is a SAN, where a fencing operation changes rules on a SAN switch to deny access from a node.

- The resource level fencing may be achieved using normal resources on which the resource we want to protect would depend. Such a resource would simply refuse to start on this node and therefore resources which depend on it will be unrunnable on the same node as well.

- The node level fencing makes sure that a node does not run any resources at all. This is usually done in a very simple, yet brutal way: the node is simply reset using a power switch. This may ultimately be necessary because the node may not be responsive at all. The node level fencing is our primary subject below to complement our Apache Server.

6.6.1 Creating a shared disk for SBD Device

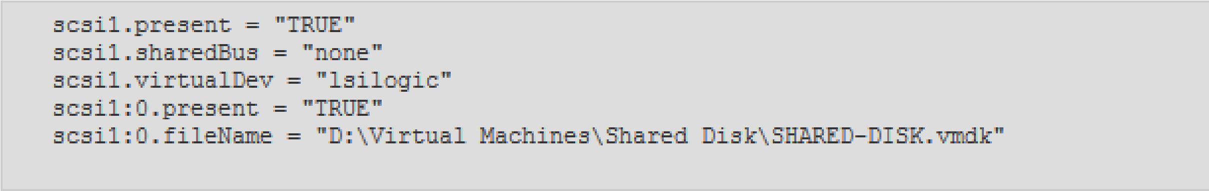
- Locate the folder Virtual Machines in your pc and create a new folder inside of it. Name the new folder as Shared Disk where the split brain detection (sbd) device settings will be stored.

- Back to the VMware, create a new virtual disk and direct it to the Shared Disk folder earlier. Assign 8GB to the disk and change the disk’s mode to Independent-Persistent and Adapter SCSI 1: Device 0. Take note the name of the device for referencing later.

- Now attach it to the other machine by using the same steps but selecting existing virtual harddisk and give the correct path to the disk on the other VM.

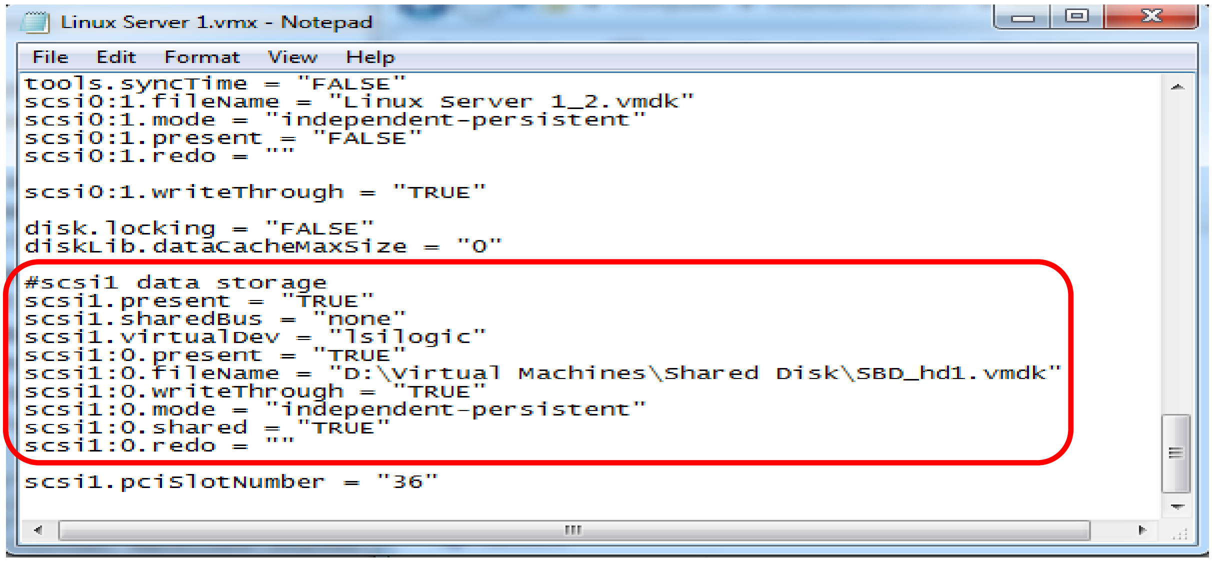
- This entire configuration is stored in the vmx file of your virtual machine. Some addition is needed to the file thus creating a backup is necessary.

- Go to the bottom of the vmx file until the lines as shown in Figure 6.6.1.1 and insert scsi(device number).shared = “TRUE” until similar to what is shown in Figure 6.6.1.2.



```
scsi1.present = "TRUE"
scsi1.sharedBus = "none"
scsi1.virtualDev = "lsilogic"
scsi1:0.present = "TRUE"
scsi1:0.fileName = "D:\Virtual Machines\Shared Disk\SHARED-DISK.vmdk"
```

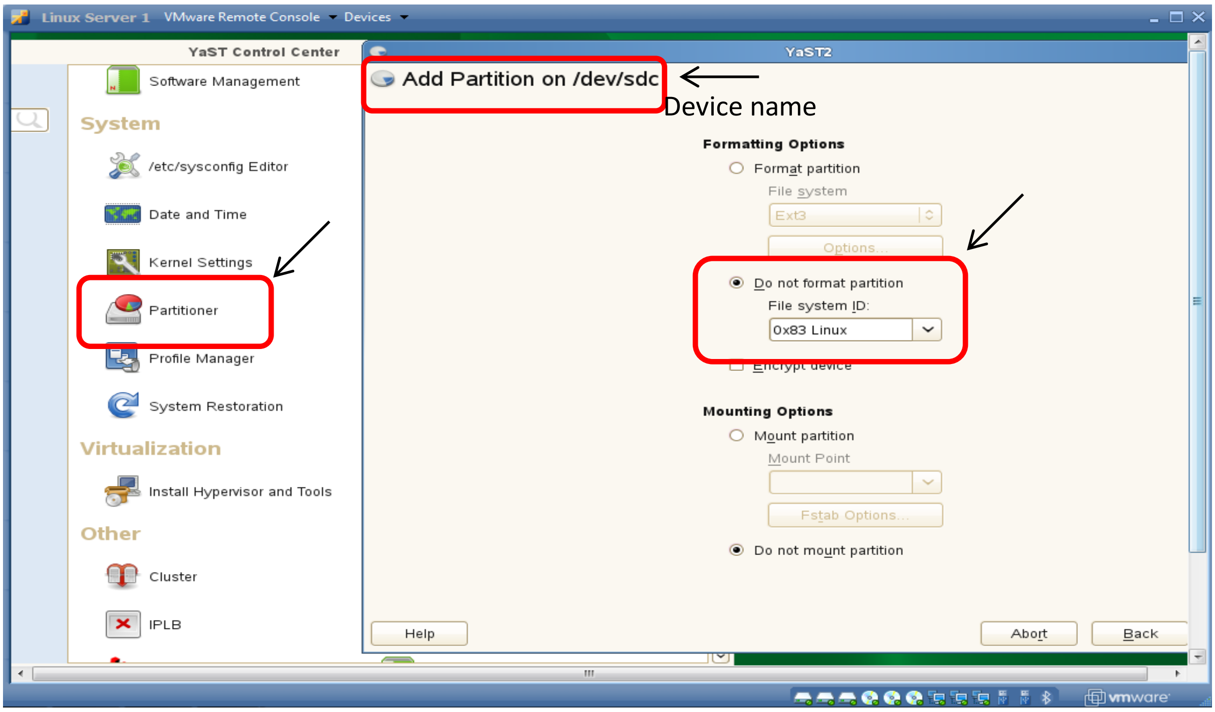
Figure 6.6.1.1 – Initial Configuration in VM’s vmx file



```
tools.syncTime = "FALSE"
scsi0:1.fileName = "Linux Server 1_2.vmdk"
scsi0:1.mode = "independent-persistent"
scsi0:1.present = "FALSE"
scsi0:1.redo = ""

scsi0:1.writeThrough = "TRUE"

disk.locking = "FALSE"
diskLib.dataCacheMaxSize = "0"

#scsi1 data storage
scsi1.present = "TRUE"
scsi1.sharedBus = "none"
scsi1.virtualDev = "lsilogic"
scsi1:0.present = "TRUE"
scsi1:0.fileName = "D:\Virtual Machines\Shared Disk\SBD_hd1.vmdk"
scsi1:0.writeThrough = "TRUE"
scsi1:0.mode = "independent-persistent"
scsi1:0.shared = "TRUE"
scsi1:0.redo = ""

scsi1.pciSlotNumber = "36"
```

Figure 6.6.1.2 – Changes needed to the vmx file

- If an error still persist during the VM startup, change the previous line to disk.locking = “FALSE”.
- Start up the VM, go to module YaST > System > Partitioner. Choose the disk with the sbd files > Add partition > Primary partition > Custom Size > Do not format partition: File system ID:0x83 Linux > Finish as shown in Figure 6.6.1.3.



Add Partition on /dev/sdc

Formatting Options

○ Format partition

File system: Ext3

Options...

● Do not format partition

File system ID: 0x83 Linux

□ Encrypt device

Mounting Options

○ Mount partition

Mount Point

Fstab Options...

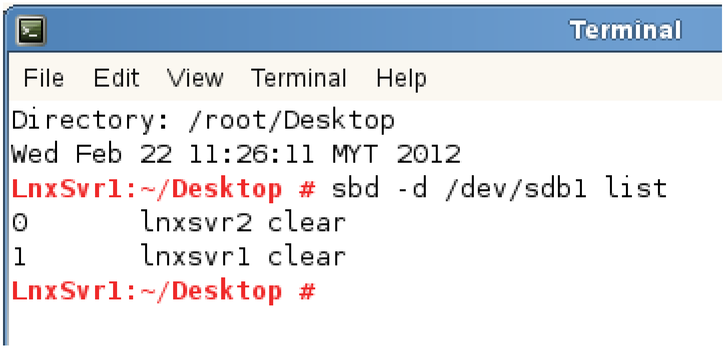
● Do not mount partition

Figure 6.6.1.3 – Partition the shared disk for Linux use.

- Click Finish and the shared disk are ready.

6.6.2 Setting up SBD Device

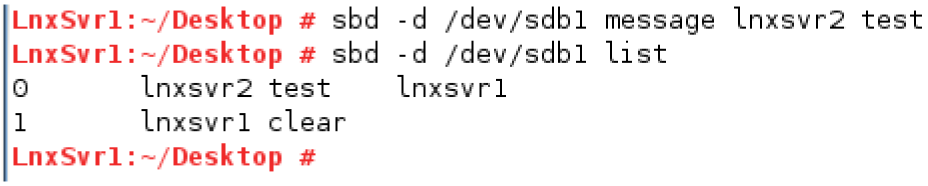
- Initialize the SBD device with the following command:
  ‘*sbd –d /dev/SBD create’* (SBD is the name of the device)
- After that, to start the SBD for testing and configuration, use the command ‘rcopenais start’.
- Testing the sbd once it runs by executing the command ‘sbd –d /dev/SBD list’ where the command will dump the node slots and their current messages from the SBD device. The result should be similar to Figure 6.6.2.1.



```
Directory: /root/Desktop
Wed Feb 22 11:26:11 MYT 2012
LnxSvr1:~/Desktop # sbd -d /dev/sdb1 list
0       lnxsvr2 clear
1       lnxsvr1 clear
LnxSvr1:~/Desktop #
```

Figure 6.6.2.1 – sbd List Result

- Try sending a test message to one of the nodes with *‘sbd –d /dev/SBD message nodea test’*. If the SBD is indeed up and running, the node will acknowledge the receipt of the message in the system logs as shown in Figure 6.6.2.2.



```
LnxSvr1:~/Desktop # sbd -d /dev/sdb1 message lnxsvr2 test
LnxSvr1:~/Desktop # sbd -d /dev/sdb1 list
0       lnxsvr2 test    lnxsvr1
1       lnxsvr1 clear
LnxSvr1:~/Desktop #
```

Figure 6.6.2.2 – Running sbd Service Confirmation

- Setup the software watchdog by adding the line modprobe softdog to /etc/init.d/boot.local.

- Next, make the OpenAIS init script start and stop SBD, create the file /etc/sysconfig/sbd and add the following lines:
  SBD_DEVICE=”/dev/sdb1”
  # the next line enables the watchdog support
  SBD_OPTS=”-W”

- Transfer the above config, */etc/sysconfig/sbd* to other nodes as well using Rsync or Csync2 before proceeding and ensure that SBD has started on all nodes by executing rcopenais restart or rcopenais start.

- To complete the SBD setup, it is necessary to activate SBD as a STONITH/fencing mechanism in the CIB as follows:

  - Using command line interface.

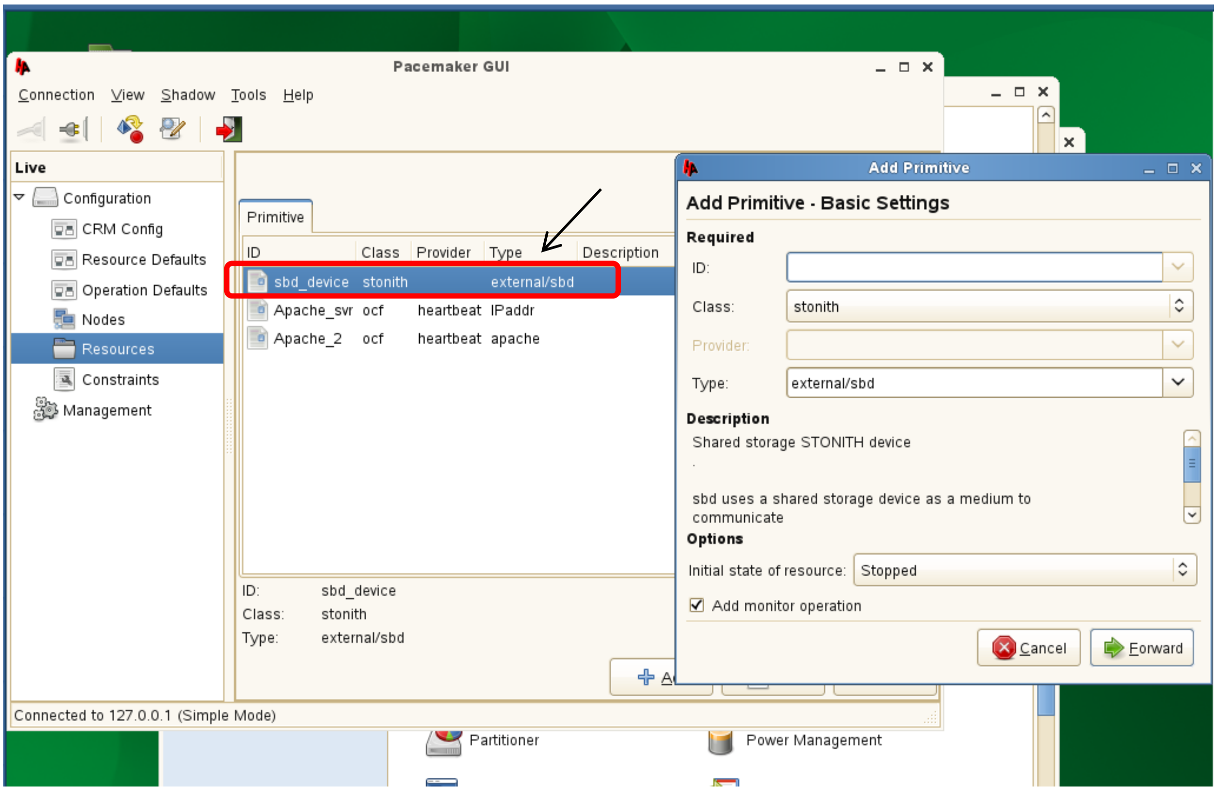
```
-  'crm configure'

-  'crm(live)configure# property stonith-enabled="true"'

-  'crm(live)configure# property stonith-timeout="30s"'

-  'crm(live)configure# primitive stonith_sbd stonith:external/sbd params
sbd_device="/dev/SBD"'

- 'crm(live)configure# commit'

-'crm(live)configure# quit'
```

  - Using Pacemaker GUI (refer Figure 6.6.2.1)

Add resource type: primitive

Class: stonith

Type: external/sbd



ID	Class	Provider	Type	Description
sbd_device	stonith		external/sbd	
Apache_svr	ocf	heartbeat	IPaddr	
Apache_2	ocf	heartbeat	apache	

ID: sbd_device
Class: stonith
Type: external/sbd

Add Primitive - Basic Settings

Required

ID:
Class: stonith
Provider:
Type: external/sbd

Description

Shared storage STONITH device

sbd uses a shared storage device as a medium to communicate

Options

Initial state of resource: Stopped

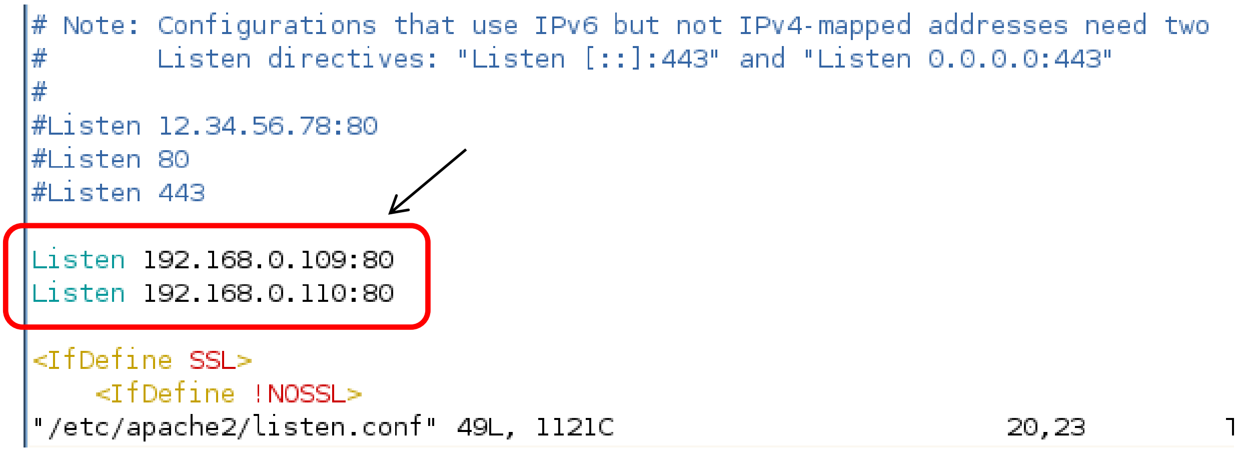
☑ Add monitor operation

Figure 6.6.2.1 – Added stonith Resource

- Set the device block used for SBD in the resource parameter. Cleanup the resource in crm_gui to start it.
- By default, the stonith service will reside in the 1st node, if that node fails. The service will be failover to the 2nd node and if the 1st node goes back online, the service will failback to node 1 because it has a higher priority.

6.7 Adding clustered apache2 server service.

- This is the main resource in our whole project, the Apache Server to hold the PRISM website. Firstly, install HTTP service (apache2) in all nodes in the cluster from the network services in yast2.
- Modify the listen directive in */etc/apache2/listen.conf* using vi function by adding Listen 192.168.0.110:80 (Shared IP Address) for both nodes. Refer Figure 6.7.1.



```
# Note: Configurations that use IPv6 but not IPv4-mapped addresses need two
#       Listen directives: "Listen [::]:443" and "Listen 0.0.0.0:443"
#
#Listen 12.34.56.78:80
#Listen 80
#Listen 443

Listen 192.168.0.109:80
Listen 192.168.0.110:80

<IfDefine SSL>
    <IfDefine !NOSSL>
"/etc/apache2/listen.conf" 49L, 1121C                          20,23          1
```

Figure 6.7.1 – Added line in /etc/apache2/listen.conf

- For starter before putting in the PRISM website prototype, create a dummy html file in /srv/www/htdocs as a default main page for the apache server when the IP Address is called. Name it index.html.
- For the floating IP Address to work, append the following line to */etc/sysctl:* *net/ipv4/ip_nonlocal_bind=1*
- In the terminal, change directory to /srv: cd /srv and execute the following command: *Chmod –R 755 www* to fully enable the www service.
- Log in to pacemaker GUI. First, define a shared IP address resource to the web server corresponding to the IP Address in listen.conf file. Steps similar in Figure 6.6.2.1

Add resource type: primitive
Class: ocf
Provider: heartbeat
Type: IPaddr

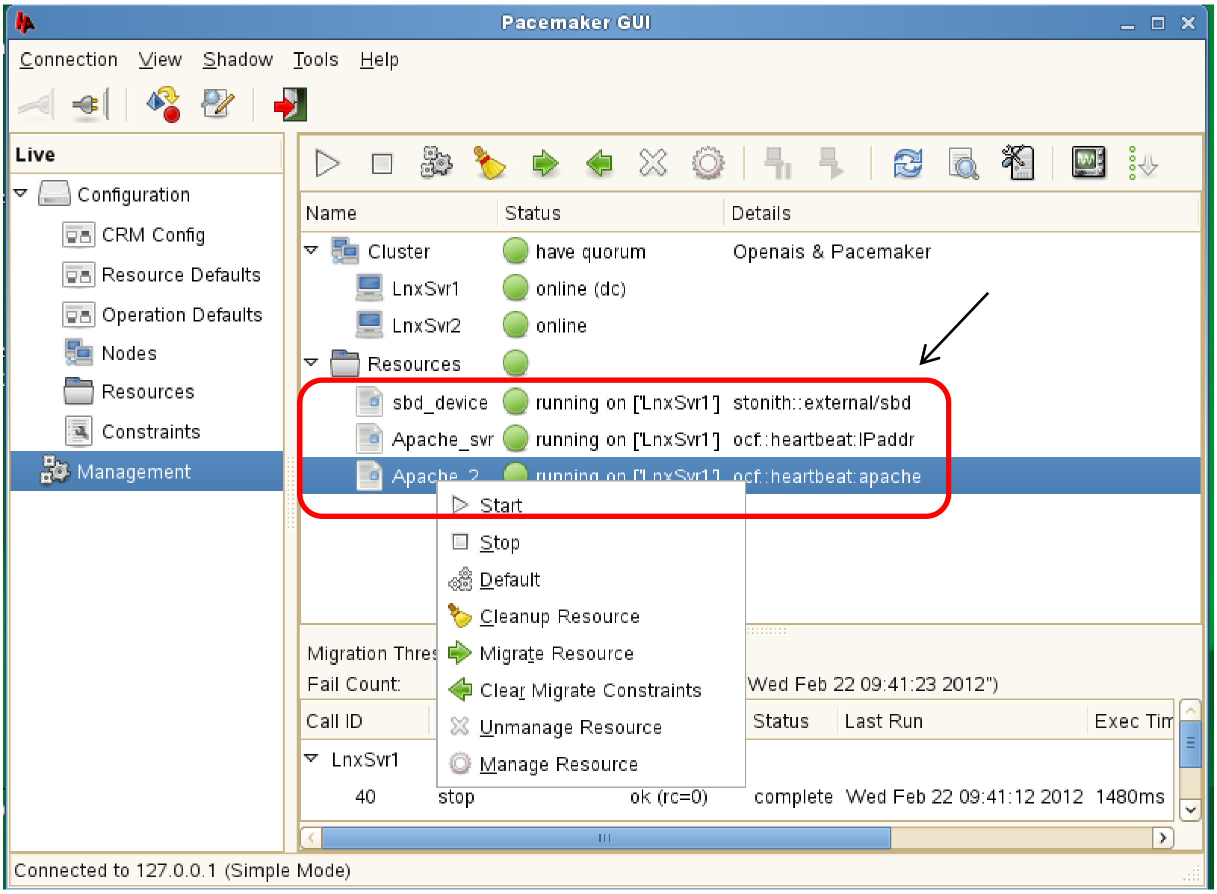
- The 2nd item will be the apache resource and the server will be ready to use.

Add resource type: primitive
Class: ocf
Provider: heartbeat
Type: apache

- For a well configured cluster, the apache service resource and the IP address resource need to be bound so that the Apache is always running on the same node where its IP address is running. Create a co-location in the constraints tab>Apache resource>with IP Address> Score:INFINITY.

- We also need to make sure the Apache is always started on a particular node only after its IP address has been activated, else the web server might not work. Create an ‘Orders’ rule in the constraints tab>Type:order> IPaddress resource> before: Apache resource.

- Now the service is ready to be started. Right click on both resources and click ‘start’.



Name	Status	Details
Cluster	have quorum	Openais & Pacemaker
LnxSvr1	online (dc)	
LnxSvr2	online	
Resources		
sbd_device	running on ['LnxSvr1']	stonith::external/sbd
Apache_svr	running on ['LnxSvr1']	ocf::heartbeat:IPaddr
Apache_2	running on ['LnxSvr1']	ocf::heartbeat:apache

Migration Thres

Fail Count: Wed Feb 22 09:41:23 2012")

```
Call ID                                 Status    Last Run                  Exec Tim
LnxSvr1
  40        stop              ok (rc=0) complete  Wed Feb 22 09:41:12 2012  1480ms
```

Connected to 127.0.0.1 (Simple Mode)

Figure 6.7.2 – Starting apache and its IP address resource

- Test the servers by inserting the shared IP address in any web browser from any node and see whether it shows the index.html that was created earlier. This shows that indeed the server is running as shown in Figure 6.7.3.

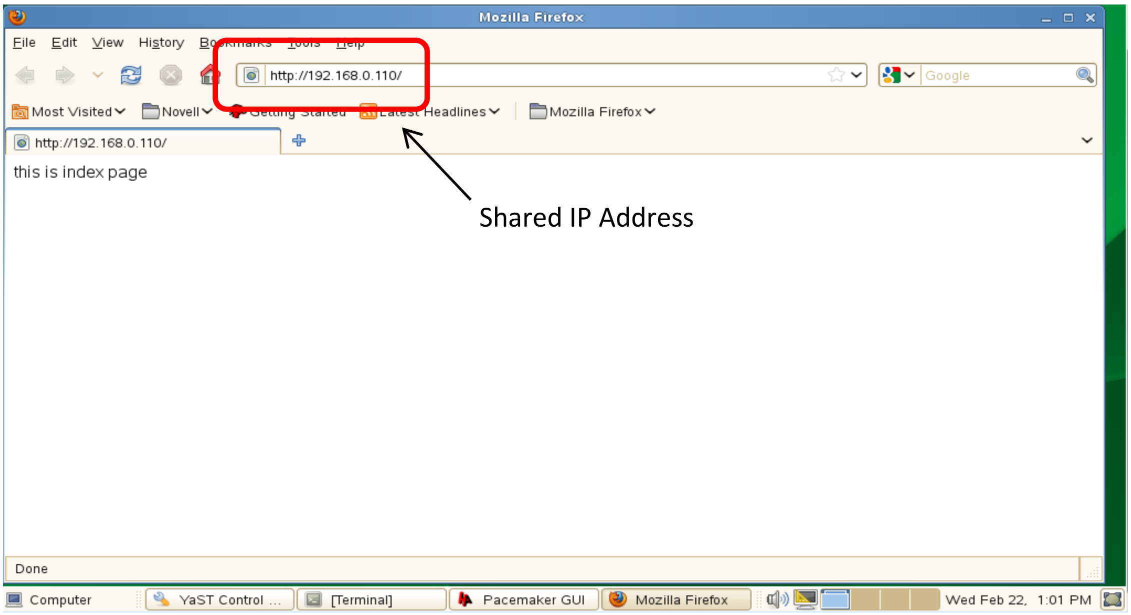

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.0.110/
Google
Most Visited Novell Getting Started Latest Headlines Mozilla Firefox
this is index page
Shared IP Address
Done
Computer YaST Control ... [Terminal] Pacemaker GUI Mozilla Firefox
Wed Feb 22, 1:01 PM

Figure 6.7.3 – Index Page from Apache Server

6.8 Adding clustered samba file server service.

- Some files are needed to be transferred from our host Windows system to the Virtual Machine to complete the project such as the prototype website of the PRISM and the Python coding for further testing. In order to do so, a medium is needed between the our local machine and the Virtual Machine inside the VMware, this is where the samba file server takes part as it enable the SUSE Linux to do specific task.

- Before that, some resources and configurations are needed for us to set up the clustered samba server as shown below and since this is a clustered service environment.

6.8.1 Configuring DLM and O2CB Resources

- The configuration consists of a base group that includes several primitives and a base clone. Both base group and base clone can be used in various scenarios afterwards for both OCFS2 and cLVM.

- Log in to pacemaker GUI, three important resources need to be defined which are: dlm, cLVM and o2cb. Create a group named base-group and cloned with name: base-clone. The configured resource will appear like in Figure 6.8.1.1

- Distributed Lock Manager (DLM)

```
Add resource type: primitive
Class: ocf
Provider: pacemaker
Type: controld
```

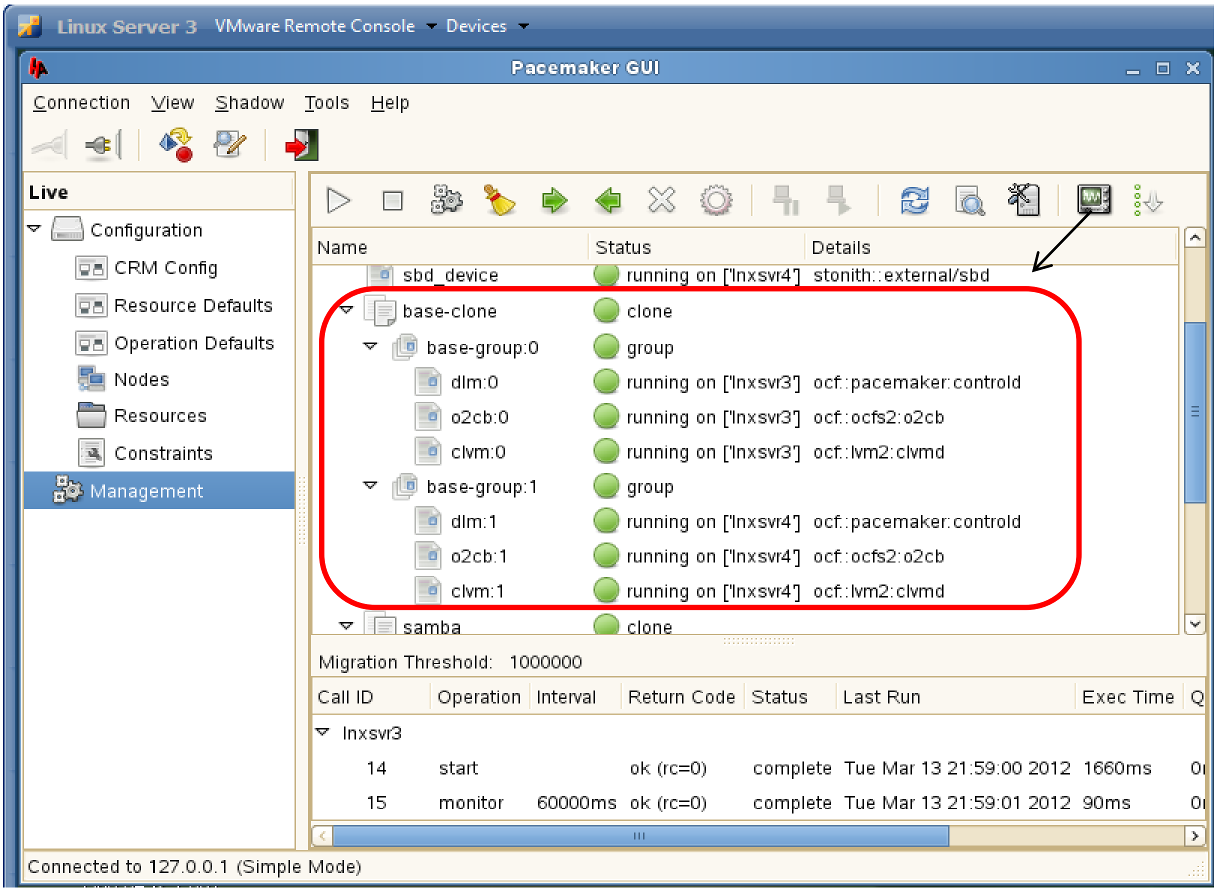
- O2CB (can only run where a DLM resource is already running)

```
Add resource type: primitive
Class: ocf
Provider: ocfs2
Type: o2cb
```

- Clustered Logical Volume Manager (cLVM)

```
Add resource type: primitive
Class: ocf
Provider: lvm2
Type: clvmd
params: daemon_timeout: 30sec
```

- The resources need to be running before formatting the logical volume with ocfs2 later.



Name	Status	Details
sbd_device	running on ['lnxsvr4']	stonith::external/sbd
base-clone	clone	
base-group:0	group	
dlm:0	running on ['lnxsvr3']	ocf::pacemaker:controld
o2cb:0	running on ['lnxsvr3']	ocf::ocfs2:o2cb
clvm:0	running on ['lnxsvr3']	ocf::lvm2:clvmd
base-group:1	group	
dlm:1	running on ['lnxsvr4']	ocf::pacemaker:controld
o2cb:1	running on ['lnxsvr4']	ocf::ocfs2:o2cb
clvm:1	running on ['lnxsvr4']	ocf::lvm2:clvmd
samba	clone	

Migration Threshold: 1000000

Call ID	Operation	Interval	Return Code	Status	Last Run	Exec Time	Q
lnxsvr3							
14	start		ok (rc=0)	complete	Tue Mar 13 21:59:00 2012	1660ms	0
15	monitor	60000ms	ok (rc=0)	complete	Tue Mar 13 21:59:01 2012	90ms	0

Connected to 127.0.0.1 (Simple Mode)

Figure 6.8.8.1 – Overview of base-clone resources

6.8.2 Creating a mounted file system, ocfs2

- Prepare a new shared disk to install with ocfs2 file system with steps similar to section 6.6.1 Creating a shared disk for SBD Device.
- Initialize the SCSI partition as physical volume (PV) by executing

  *pvcreate /dev/sdc1 *follow the name of your own disk**
- Next, create a cluster-aware volume group (VG) named *vg1* consisting of the PVs you just initialized.

  vgcreate --clustered y vg1 /dev/sdc1
- Add a logical volume inside the volume group created by executing:

  *lvcreate --name vg1 -L 4G lv1 *create logical volume,lv1 inside volume group,vg1 with size 4GB**

- Format the logical volume with ocfs2 with the following line. (make sure the resources in 6.8.1 Configuring DLM and O2CB Resources is running before proceeding)

  *mkfs.ocfs2 –N 2 /dev/vg1/lv1 *maximum number of nodes:2**

- Test the LV with manual mounting:
  - Activate the volume group: *vgchange –a y*
  - Mount the LV in any directory: *mount /dev/vg1/lv1 /mnt/shared/samba*
  - In case of version issue resulting in mount error, update the ocfs2 filesystem resource in both nodes to update the cluster stack to pacemaker by issuing: *fsck.ocfs2 /dev/vg1/lv1*

- Now, create 2 resources to activate the volume group and mounting the ocfs2 volume using the pacemaker named vg1 and ocfs2-1 respectively. The two resource need to be grouped and cloned as shown in Figure 6.8.2.1.

- vg1

```
Add resource type: primitive
Class: ocf
Provider: heartbeat
Type: LVM
volgrpname: vg1
```

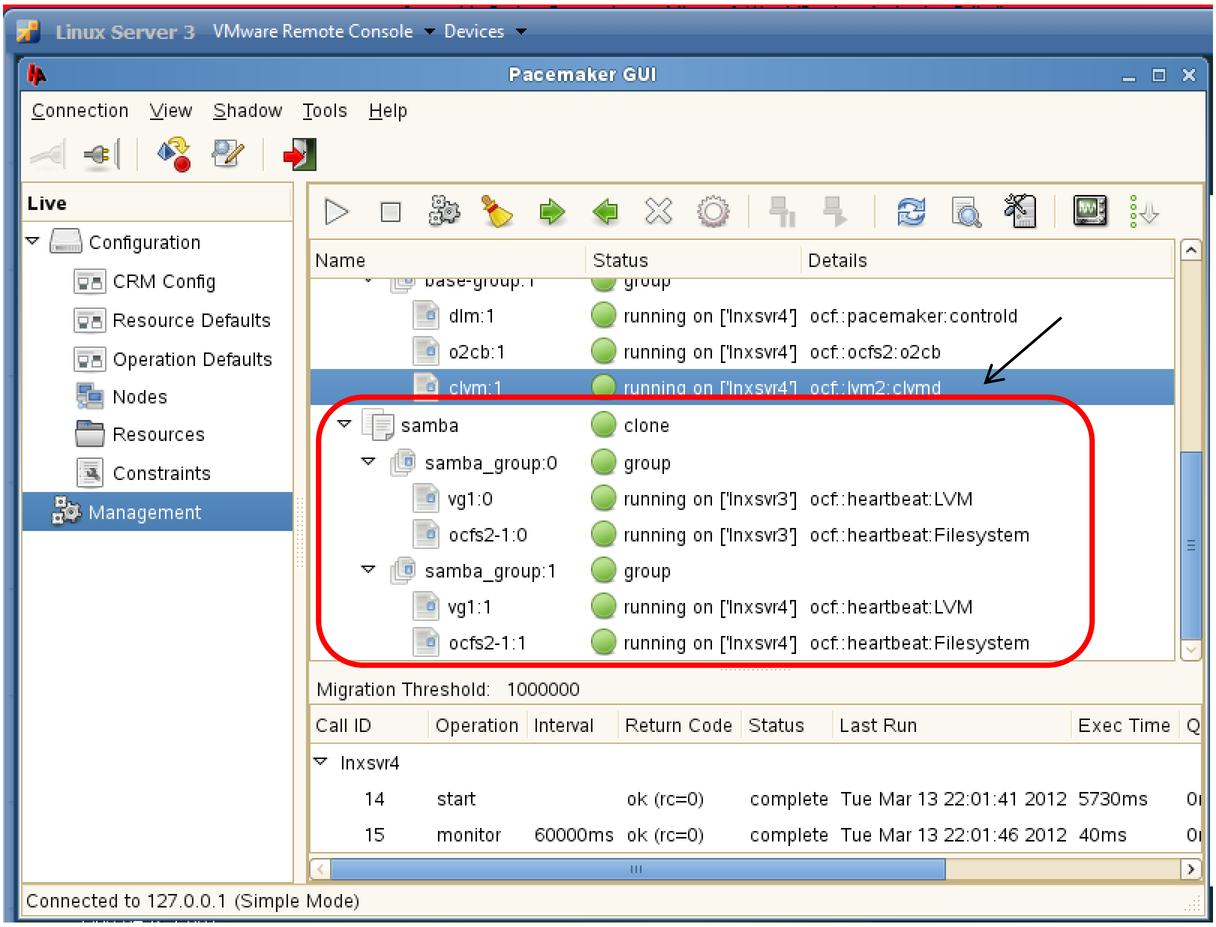
- ocfs2-1

```
Add resource type: primitive
Class: ocf
Provider: heartbeat
Type: Filesystem
Device: /dev/vg1/lv1
Directory: /mnt/shared/samba
fstype: ocfs2
Options: acl
Start Delay: 30sec
```

* the directory must be available on all nodes *

- You need to configure a fencing device (covered in the previous step). Without a STONITH mechanism (like external/sbd) in place the configuration will fail.

- In case of resource not running, try to manually mount the filesystem 1st as stated previously and determine the problem before bringing up the resource again.



Name	Status	Details
base-group:1	group	
dlm:1	running on ['lnxsvr4']	ocf::pacemaker:controld
o2cb:1	running on ['lnxsvr4']	ocf::ocfs2:o2cb
clvm:1	running on ['lnxsvr4']	ocf::lvm2:clvmd
samba	clone	
samba_group:0	group	
vg1:0	running on ['lnxsvr3']	ocf::heartbeat:LVM
ocfs2-1:0	running on ['lnxsvr3']	ocf::heartbeat:Filesystem
samba_group:1	group	
vg1:1	running on ['lnxsvr4']	ocf::heartbeat:LVM
ocfs2-1:1	running on ['lnxsvr4']	ocf::heartbeat:Filesystem

Migration Threshold: 1000000

Call ID	Operation	Interval	Return Code	Status	Last Run	Exec Time	Q
lnxsvr4							
14	start		ok (rc=0)	complete	Tue Mar 13 22:01:41 2012	5730ms	0
15	monitor	60000ms	ok (rc=0)	complete	Tue Mar 13 22:01:46 2012	40ms	0

Connected to 127.0.0.1 (Simple Mode)

Figure 6.8.2.1 – Overview of samba-clone resources

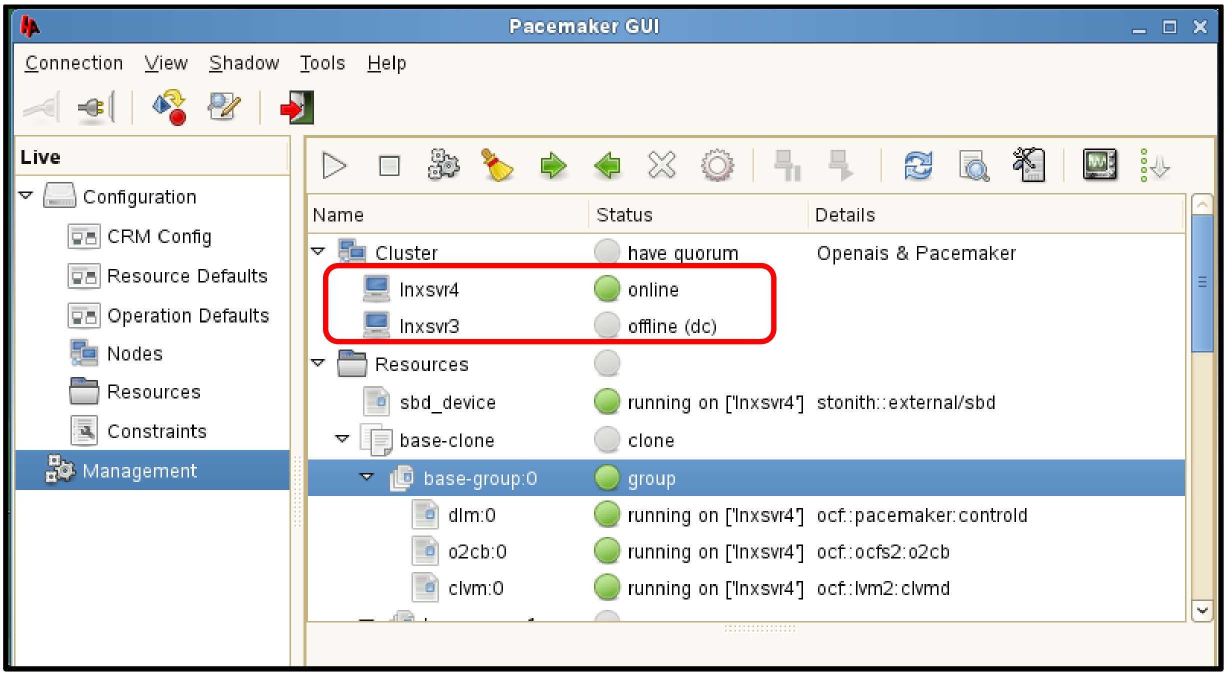
- Create colocation and order so that base-clone is running 1st before these resource started because it depends on it.

- With the samba server online, we can now port the PRISM prototype (created from the page source of the original PRISM website) prepared in the Windows local machine to both of the Linux's Apache Server just to simulate a close to similar situation of the real life PRISM.

7.0 Results

Important: The Linux Virtual Server Clustering in this project is just a simulation created in a virtual environment inside the VMware including the Web Servers which supposed to be real servers and is not an actual implementation to the real UTP Student Portal website. The main idea of this project is to prepare a working simulated cluster as a base for the PRISM. For an actual implementation the same exact steps included in the final report will just need to be reproduced in a real environment with real web servers and creating the virtual cluster in one of the available servers.

7.1 Network Monitoring Tool (Pacemaker GUI)



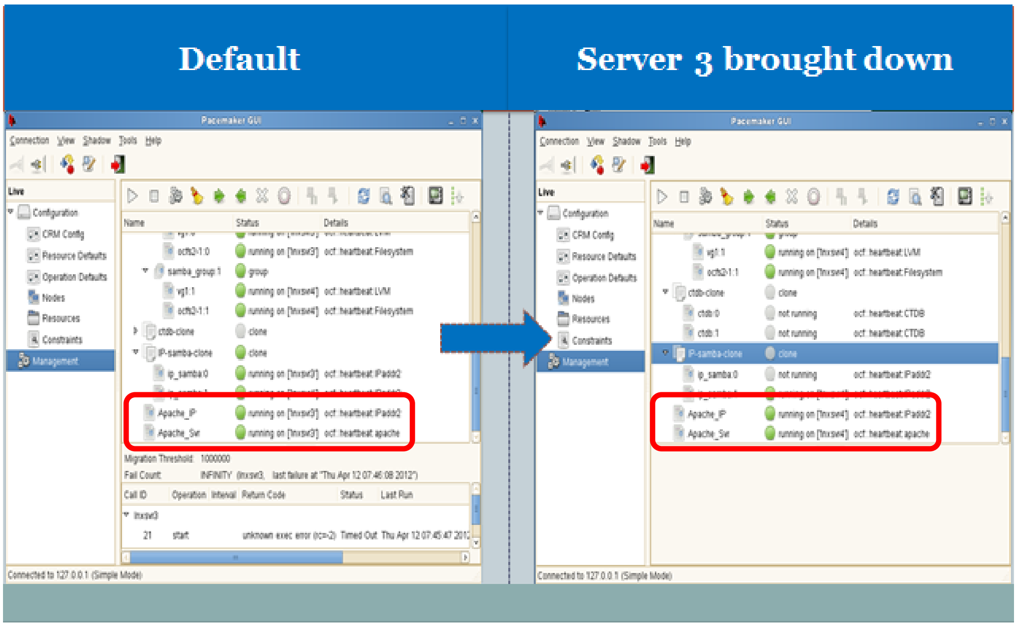
Name	Status	Details
Cluster	have quorum	Openais & Pacemaker
lnxsvr4	online	
lnxsvr3	offline (dc)	
Resources		
sbd_device	running on ['lnxsvr4']	stonith::external/sbd
base-clone	clone	
base-group:0	group	
dlm:0	running on ['lnxsvr4']	ocf::pacemaker:controld
o2cb:0	running on ['lnxsvr4']	ocf::ocfs2:o2cb
clvm:0	running on ['lnxsvr4']	ocf::lvm2:clvmd

Figure 7.1.1 – Pacemaker GUI – Server Status

- The Pacemaker GUI that is used to monitor the resources of the cluster, status of the servers and can also be used to simulate a manual shut down of the Apache Server that is running to simulate a server downtime in real life.

- If the cluster is working as intended, the heartbeat service will automatically failover the Apache Resource from the fallen server into the Linux Server that is on standby, making sure that the resource is still alive and continue providing the service.

- Any transaction or process from the downed server (LnxSvr3) will not be lost but instead will be continued to run at the newly active server (LnxSvr4) as shown in Figure 7.1.2.



```
Name              Status                 Details
  ocfs2-1:0       running on [lnxsvr3]   ocf::heartbeat:Filesystem
samba_group:1     group
  vg1:1           running on [lnxsvr4]   ocf::heartbeat:LVM
  ocfs2-1:1       running on [lnxsvr4]   ocf::heartbeat:Filesystem
ctdb-clone        clone
IP-samba-clone    clone
  ip_samba:0      running on [lnxsvr3]   ocf::heartbeat:IPaddr2
Apache_IP         running on [lnxsvr3]   ocf::heartbeat:IPaddr2
Apache_Svr        running on [lnxsvr3]   ocf::heartbeat:apache

Migration Threshold:  1000000
Fail Count:           INFINITY (lnxsvr3,  last failure at "Thu Apr 12 07:46:08 2012")

Call ID   Operation  Interval  Return Code              Status     Last Run
lnxsvr3
  21      start                unknown exec error (rc=2) Timed Out Thu Apr 12 07:45:47 2012
```

```
Name              Status                 Details
  vg1:1           running on [lnxsvr4]   ocf::heartbeat:LVM
  ocfs2-1:1       running on [lnxsvr4]   ocf::heartbeat:Filesystem
ctdb-clone        clone
  ctdb:0          not running            ocf::heartbeat:CTDB
  ctdb:1          not running            ocf::heartbeat:CTDB
IP-samba-clone    clone
  ip_samba:0      not running            ocf::heartbeat:IPaddr2
Apache_IP         running on [lnxsvr4]   ocf::heartbeat:IPaddr2
Apache_Svr        running on [lnxsvr4]   ocf::heartbeat:apache
```

Figure 7.1.2 – Apache Resource fail-over from manual server shut down.

7.2 Simulated Environment (Python Scripting)

- Instead of manually shut down of the servers for no particular reason, I have created a script in Python to simulate a number of simultaneous users, N is accessing the PRISM Service inside the cluster for N period of time (minutes) to create a simulated situation of a number of UTP Students trying to open and use the PRISM website in real life to a point that the service is unable to comprehend the burden to the point that the server crashed.

But again creating this simulated environment will just create a situation similar to the real-life situation but it will not produce result that can presented with actual data or graph nor can it be tabulated. The important aspect is to prove that the cluster is indeed working and the best way to prove it is by using the Pacemaker GUI as stated before.

8.0 Conclusion

Linux Virtual Server Clustering, a method to improve the PRISM as a reliable and always available service to the UTP Students. Using the Linux Virtual Server to maintain and manage the PRISM, theoretically it will provide higher availability and can accommodate higher number of students accessing the system simultaneously. Other services can also be set up inside the University internal network with the Linux wide range of service settings.

In this documentation, although this is just a simulation of the cluster itself, the base and steps required is there for it to be implemented in the actual environment of the PRISM. Hopefully, this effort can help to provide a better Student Portal for the future students of UTP and at the same time help to raise the name of UTP as a technological based university among others.

As a conclusion, if this project can serve as a future plan to UTP in order to improve the university in terms of the technological level of internal system implementation, the project will provide a lot of benefit to the University and the students. This means that the project will not stop here and could be exploited to bring more improvement to the University's system with the LVS huge potential overall.

REFERENCES

1. Alvin Chan (2011, April 13). High-Availability Web Server Cluster Presentation. Retrieved from: http://www.slideshare.net/avinc/ha-systemfirst-presentation?src=related_normal&rel=10067798

2. Linux Wiki (2011, March 8). Installation of Linux on z/VM. Retrieved from: http://wiki.linuxvm.org/wiki/Installation_on_z/VM

3. Donna Scott (2001, July 3). NSM: Often the Weakest Link in Business Availability. Retrieved from: http://www.gartner.com/id=334197

4. Wensong Zhang and Wenzhuo Zhang (2003). Linux Virtual Server Clusters: Build highly-scalable and highly-available network services at low cost. Retrieved from: http://www.linuxvirtualserver.org/

5. Server World. (Copyrighted: 2007 - 2012). Password Authentication Setting - SLES 11. Retrieved from: http://www.serverworld.info/en/note?os=SUSE_Linux_Enterprise_Server_11&p=ssh

6. man0070 (Youtube User Account: 2010, September 18). How to set up a simple Samba Server. Retrieved from: http://www.youtube.com/watch?v=Nu2RNJ7_VcE

7. Tanja Roth, Thomas Schraitle (2012, May 7). SUSE Linux Enterprise High Availability Extension: High Availability Guide. Retrieved from: http://doc.opensuse.org/products/draft/SLE-HA/SLE-ha-guide_sd_draft/index.html

8. Brightstorm (2010, April 11). Interesting Challenge getting SSH outside to my VMware guest. Retrieved from: http://www.unix.com/linux/147802-interesting-challenge-getting-ssh-outside-my-vmware-guest.html

9. Bob Rankin (2010, January 22). Simple Guide to the Vi - Linux Editor. Retrieved from http://lowfatlinux.com/linux-editor-vi.html
10. Dirk Taggesell (2010, April 7). How to build a heart beat cluster. Retrieved from: http://blog.taggesell.de/index.php?/archives/83-How-To-Build-A-Heartbeat-Cluster.html

11. Shaheer Badar (2010, February 3). Configure Apache services on SUSE Linux Enterprise Server 11. Retrieved from: http://infotechinspiration.blogspot.com/2010/02/configure-apache-services-on-suse-linux.html

12. Davesimons (2007, January 5). How to Setup and Configure Apache Web Server in SLES 10. Retrieved from: http://www.novell.com/communities/node/1346/how-setup-and-configure-apache-web-server-sles-10

13. Dell Linux (2011, June 15). Set up an OCFS2 Cluster Filesystem. Retrieved from: http://linux.dell.com/wiki/index.php/Set_up_an_OCFS2_cluster_filesystem

14. Erikw (2008, November 19). Building a VMware Shared Disk Steps. Retrieved from: http://www.symantec.com/connect/articles/building-vmware-shared-disk

15. Michael Schwartzkopff (2011, November 30). Guide: Dual Primary DRBD + OCFS2. Retrieved from: http://www.clusterlabs.org/wiki/Dual_Primary_DRBD_%2B_OCFS2

16. Joanne Daudier (2013, March 13). Beginner Tutorial for Python Scripting. Retrieved from: http://wiki.python.org/moin/BeginnersGuide/Programmers