

Anastasiia Dudnik

# Creating a high-availability cluster with two physical servers and virtual machines

Bachelor's Thesis  
Information Technology Degree programme

2017



South-Eastern Finland  
University of Applied Sciences

| Author (authors)  | Degree                 | Time                              |
|---|------------------------|-----------------------------------|
| Anastasiia Dudnik   | Information Technology | May 2017                          |
| <b>Title</b>  |                        |                                   |
| Creating a high-availability cluster with two physical servers and virtual machines.  |                        | 60 pages<br>7 pages of appendices |
| <b>Commissioned by</b>  |                        |                                   |
| Mizar Ltd.  |                        |                                   |
| <b>Supervisor</b>   |                        |                                   |
| Matti Juutilainen   |                        |                                   |
| <b>Abstract</b>   |                        |                                   |
| <p>Many companies use cluster systems for their work. It provides a failover, high-availability and greater performance, than just a regular server system. This thesis work provides the information how to implement and configure a high-availability cluster environment with use of two physical servers and virtual machines. Also, it describes server networks, virtualization technology, storage sharing technology, backup technology and clustering.</p> <p>To achieve the objectives following methods were used: two server nodes were combined into one cluster system, shared storage was attaching to it and virtual machines were created, which in future acted as separate servers. Also, migration and live migration of virtual machines were implemented and backup/restore system was configured. To implement live migration in the cluster, easy backup/restore system and to achieve resource optimization, virtualization technology and shared storage were used.</p> <p>As the result, high-availability environment was implemented. It provides optimization of resources, easy migration and copying of VMs. And also, live migration inside the cluster and easy backup/restore system was configured.</p> <p>This thesis work was written for the Mizar Ltd, placed in Russia, Petrozavodsk. The practical part was done with cooperation of company's IT employees and implemented on firm's hardware for its needs. With high-available cluster system, downtime of application, in case of failure, is decreased and sensitive applications stay online maximized time and there is no need for company customers and users to wait for application availability.</p> |                        |                                   |
| <b>Keywords</b>   |                        |                                   |
| Cluster, High-availability, Virtualization, Shared storage, Proxmox VE  |                        |                                   |

## CONTENTS

|     |                                       |    |
|-----|---------------------------------------|----|
| 1   | INTRODUCTION .....                    | 5  |
| 2   | STRUCTURE OF NETWORK .....            | 6  |
| 3   | SERVERS AND THEIR TYPES.....          | 10 |
| 4   | SHARED STORAGE .....                  | 11 |
| 4.1 | RAIDs .....                           | 11 |
| 4.2 | SAN .....                             | 13 |
| 4.3 | NAS .....                             | 14 |
| 4.4 | NAS-SAN hybrid .....                  | 15 |
| 5   | VIRTUALIZATION .....                  | 15 |
| 5.1 | Virtualization in server network..... | 16 |
| 5.2 | Virtual machines and their roles..... | 16 |
| 5.3 | Memory virtualization.....            | 17 |
| 5.4 | Storage virtualization .....          | 18 |
| 5.5 | Live migration .....                  | 19 |
| 6   | CLUSTERING .....                      | 22 |
| 6.1 | Operating principles.....             | 22 |
| 6.2 | Virtual clusters .....                | 23 |
| 6.3 | Network requirements.....             | 24 |
| 6.4 | Quorum.....                           | 24 |
| 6.5 | Fencing.....                          | 25 |
| 6.6 | Reliability of HA cluster.....        | 27 |
| 7   | BACKUP.....                           | 28 |
| 8   | PROXMOX VE.....                       | 29 |
| 8.1 | Principle of operation .....          | 29 |
| 8.2 | Features of Proxmox VE.....           | 30 |

|       |  |    |
|-------|--|----|
| 8.3   | Advantages and disadvantages.....  | 32 |
| 8.4   | System requirements.....   | 32 |
| 8.5   | Comparison with other platforms .....  | 33 |
| 9     | IMPLEMENTING A HA CLUSTER .....  | 35 |
| 9.1   | Technical specifications.....  | 35 |
| 9.2   | Installing the hypervisor (Proxmox 4.4.) on both physical servers .....                    | 36 |
| 9.2.1 | Installing configuring Debian Jessie OS and installing the Proxmox 4.4<br>hypervisor. .... | 37 |
| 9.2.2 | Installing Proxmox 4.4. hypervisor with Debian .....                                       | 39 |
| 9.3   | Connecting through the web interface .....   | 40 |
| 9.4   | Combining pve01 and pve02 nodes into the cluster .....                                     | 41 |
| 9.5   | Adding the shared storage.....   | 44 |
| 9.6   | Creating VMs .....   | 46 |
| 9.7   | Configuring live migration .....   | 50 |
| 9.8   | Configuring backup.....  | 52 |
| 10    | CONCLUSIONS .....  | 55 |
|       | REFERENCES .....   | 57 |
|       | APPENDICES   |    |

Appendix 1. Feature comparison table of different server virtualization platforms

## 1 INTRODUCTION

Most companies use server/client environment in their operation and work. In this technology computers are separated into two categories: clients and servers. Clients are the users of the company usually connected through a network. Servers provide data and resources to the clients. Usually, for greater performance and reliability server nodes connected together and form one cluster system. Cluster provides a faster I/O, data access and a failover. To minimize a downtime of system applications and to make system more reliable, high-availability can be implemented. In the high-availability cluster technology, in case of failure of one node occurs, the responsibilities go to another server node. With this technology high-available software applications can be used as well as downtime-sensitive applications.

For increasing the performance of the system, virtualization technology can be used with high-availability cluster technology. The virtualization intends the creation of virtual machines and virtual distribution of physical resources. With virtualization, migration of virtual machines running on one node to another node can be done. But before the migration, operation of the VM should be stopped what leads to downtime of application. Also, if virtual distribution of server resources are implemented, live migration can take place and there is no need to stop virtual machine's operation. With live migration the downtime of application decreases and this make the system highly available and sustain to failures. Also, storage sharing technology should be implemented, so that the images of virtual machines are kept on shared storage, but their operations distributed and run on different nodes. In addition, shared storage provides easy backup and restore system, as in case of node failure, all the data is kept on shared environment that stays available and can be used by other nodes.

Cluster technology saves money as the physical resources can be virtually distributed with virtualization technology and usage of them can be optimized and would not need additional resources.

This thesis work was written for the “OOO Mizar” company, placed in Russia, Petrozavodsk. The practical part was done with cooperation of company’s IT employees and implemented on firm’s hardware for its needs. The aim of the thesis work, is to combine two physical servers into one high-availability cluster with use of virtual machines. The main objective of the work is to do the system high-available, reliable and sustainable to failures. Second objective is to achieve the optimization of physical server resources. Methods that are used to achieve these objectives are: combining two server nodes into one cluster system, attaching it to the shared storage and creating virtual machines, which in future would act as separate servers. Future methods are: implementing the migration of virtual machines, also, making it possible to do the live migration of running virtual machines and implementing an easy backup system. To implement live migration in the cluster, easy backup/restore system and to achieve resource optimization, virtualization technology and shared storage is going to be used. With high-available cluster system, downtime of application, in case of failure, is decreased and sensitive applications can stay online maximized time and users and clients of the company do not need to wait for application availability.

This thesis work provides information how to implement a high-availability cluster technology. Also, it describes server networks, virtualization technology, storage sharing technology, backup technology and clustering.

## **2 STRUCTURE OF NETWORK**

Two or more computers that are connected to each other, exchange data and share resources form a network. Network types are often classified by scale. The most common ones are the following:

**LAN** (Local Area Network) is a high-speed network that operates in a small geographic area within a single building or campus. It interconnects from two to hundreds of nodes by cabling or wireless technology. This type of network is usually used by small businesses or by departments of corporations located in the same building or in connected buildings. Tanenbaum A. S., Wetherall D. J. 2011, 23-27.)

**WAN** (Wide Area Network) spans on large geographical area. It usually connects multiple LANs. Usually in a WAN hosts and subnets are owned and controlled by different organizations. Also, routers usually connect different kinds of networking technologies of composite networks. Entire LANs could be connected to the subnet, and this is how larger networks are built from smaller ones. (Tanenbaum A. S., Wetherall D. J. 2011, 23-27.)

**VPN** (Virtual Private Network) allows information to be securely sent across a public or unsecure network, such as the Internet. Common uses of a VPN are to connect branch offices or remote users to a main office. (Tanenbaum A. S., Wetherall D. J. 2011, 23-27.)

Also, because of variety of functions, multiple network architectures such as peer-to-peer, server/client and mainframe/terminal have been developed.

In a **peer-to-peer** architecture, all hosts in the network can request and provide data and services. Figure 1 shows the example of peer-to-peer network architecture. In other words, two computers which share the files would be considered as a peer-to-peer network. (Tanenbaum A. S., Wetherall D. J. 2011, 7-8.)

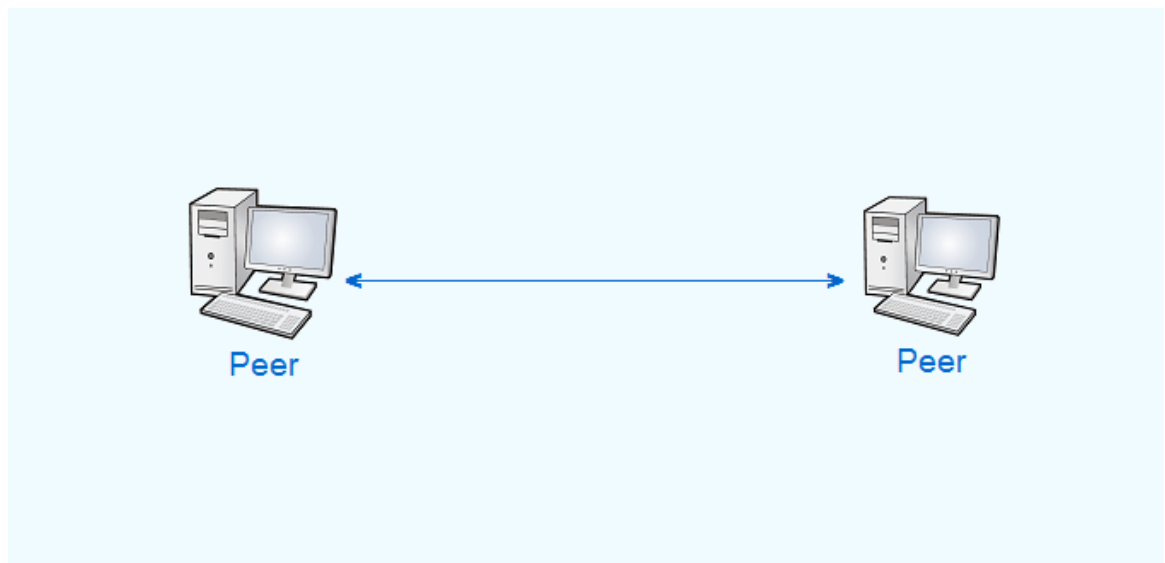


Figure 1. Peer-to-peer network architecture

The network is very simple to configure, but the architecture can present several challenges. For example it is difficult to manage data and backups in this network, because data is stored on multiple devices. In addition, many peer-to-peer systems, for example a Bit Torrent, do not have a central database of data. Instead, each user maintains his own database locally. Also, security in the peer-to-peer architecture is problematic, as users' accounts and permissions must be configured individually on each computer. Now, peer-to-peer communication is mostly used to share music and videos. And one more example of peer-to-peer communication is instant messaging. (Balchunas A. 2014, 4.)

In a **mainframe/terminal** architecture, the mainframe device stores all data and services for the network and provides centralized management and security of data. And there are dumb terminals connected to the mainframe. The mainframe performs all processing functions for the terminals. Thus, dumb terminals do not perform processing, they only serve input and output into the mainframe. Figure 2 shows the example of mainframe/terminal network architecture. (Balchunas A. 2014, 5.)

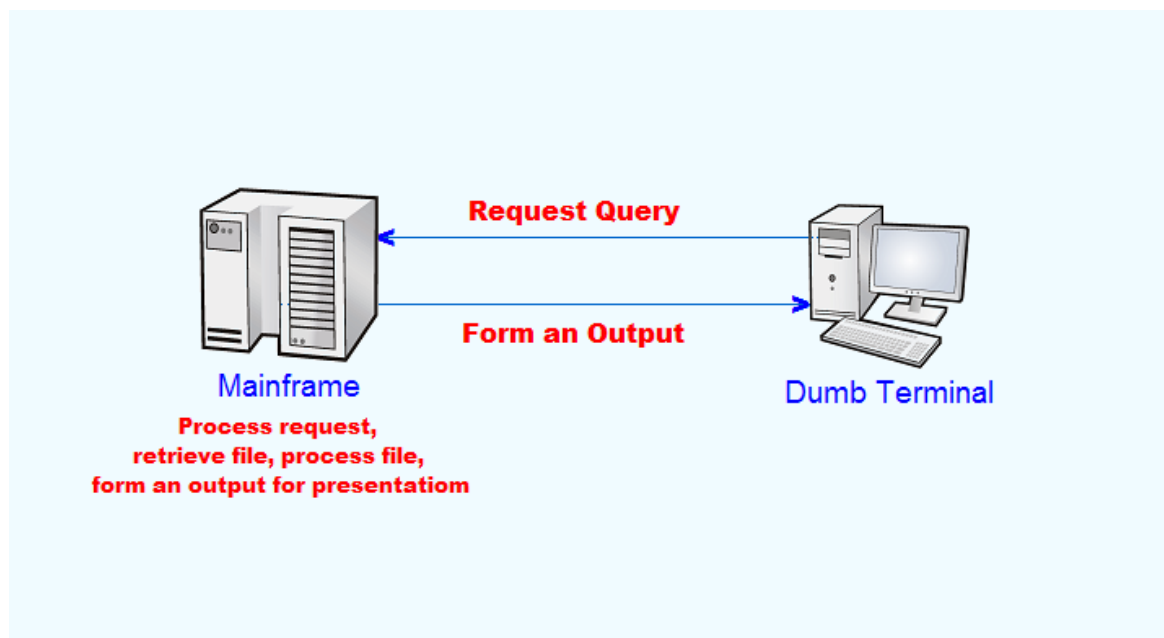


Figure 2. Mainframe/terminal network architecture

Also, there is a similar to mainframe/terminal the thin-client architecture, which nowadays has gained popularity. A thin-client architecture can be implemented



as a hardware device, or software running on top of another operating system. Like dumb terminals, thin-clients require a centralized system to perform processing functions. (Balchunas A. 2014, 5.)

And one more architecture type is **server/client**. As the name says it is mostly used in server networks and clusters. It is widely used and forms the basis of much network usage. In this type of network the server hosts resources for the rest of the clients that are using it. And the clients connect to the server and request the data and processes. Figure 3 shows the example of server/client network architecture. (Balchunas A. 2014, 6-7.)

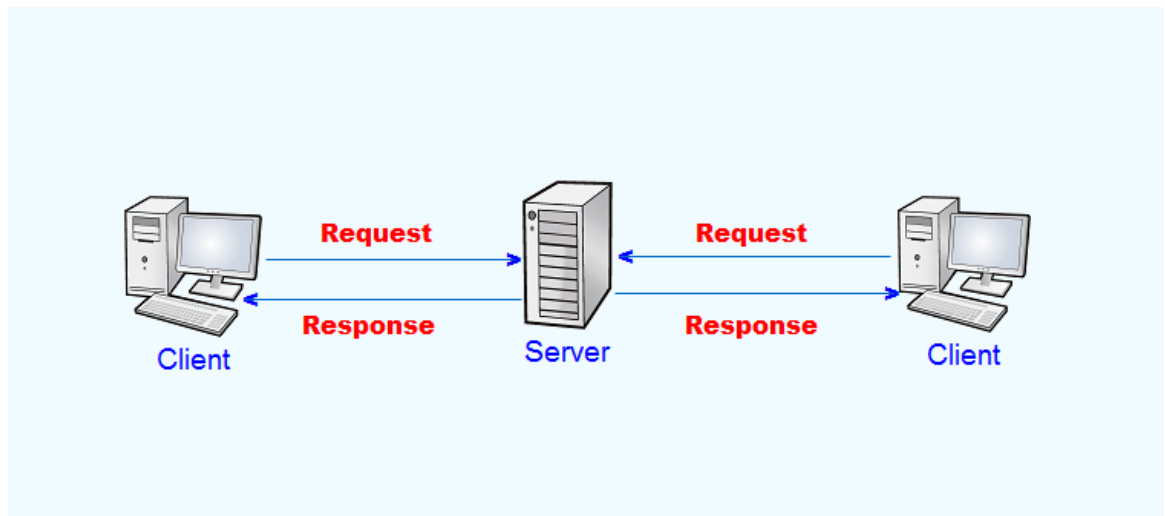


Figure 3. Server/client network architecture

In the server network users and devices must be registered and authenticated with the server to attempt the recourses. (Balchunas A. 2014, 6-7.)

Server/client architecture consists of server computer, clients, router, and switch and transmission media.

**Server** is the computer that is designed to process requests and deliver data to clients. It provides resource access to all the users on the network. There are many different types of servers depending on their roles in the network. (Balchunas A. 2014, 6-7.)

**Client** is the computer that is connected to the network and uses shared resources. Clients request and receive services from the server. (Balchunas A. 2014, 6-7.)

**Router** connects to and allows communication between two distinct networks or shares a single internet connection to multiple computers. The router determines the best path through the network. (Balchunas A. 2014, 41-45.)

**Switch** is the specialized device that filters and forwards packets between LAN segments. Switches operate at the data link layer (layer 2) and sometimes the network layer (layer 3) of the OSI Reference Model. (Balchunas A. 2014, 38-39.)

**Transmission media** connects devices in a network. It can be a twisted pair wire, coaxial cable, optical fiber cable or even wireless. (Tanenbaum A. S., Wetherall D.J. 2011, 95.)

### 3 SERVERS AND THEIR TYPES

In the server/client network architecture there can be different types of servers that perform different tasks for the network and its workstations. One physical server can house many different virtual servers. To network clients, each server appears as a completely separate device. Each server has a platform that it works on. A platform is the underlying hardware or software for a system and it is like the engine that drives the server. (Scottish Qualification Authority. 2010.)

The most common types of servers are the following ones:

1. File server stores data files.
2. Communication server handles many common communications functions for the network, such as e-mail, fax, remote access, firewalls or Internet services.
3. Application server shares network-enabled versions of common application software and eliminates the need for software to be installed on each work-station.

4. Database server manages common databases for the network, handling all data storage, database management and requests for data.
  5. Domain server authenticates and authorizes computers and users to access resources within the logical domain.
  6. Proxy server sits between a client program and an external server to filter requests, improve performance, and share connections.
  7. Mail server moves and stores mail over corporate networks and across the Internet.
  8. Web server provides static content to a web browser by loading a file from a disk and serving it across the network to a user's web browser.
  9. Log server saves and contains automatically produced and time-stamped documentation of events happened in a system.
- (Scottish Qualification Authority. 2010.)

## **4 SHARED STORAGE**

This thesis involves cluster technology. Clustering is discussed later in the chapter 6. Shared storage must be used to form a high-available and failover cluster.

Shared storage is a medium intended for file storage. It can be accessed by multiple computers simultaneously with no need to duplicate data to their local storage. But still local storage can be used as the system disk. (Rouse M. 2013.)

### **4.1 RAIDS**

Redundant Array of Independent Disks (RAID) is method to protect the system against disk failures. This technology does not provide the sharing of storage resources but it used for redundancy in SAN and NAS technologies, which are mentioned later in this chapter. Also, it can be implemented on the local disk. RAID technology combines multiple physical disks into one logical element. It provides redundancy, improved performance and fault tolerance. Technologies that used with RAID are disk striping, disk mirroring and disk striping with parity. (SNIA. 2016.)

RAID technology is divided into multiple levels that have different technologies of work. Some levels can be combined together for the improvement of RAID.

**RAID 0** consists of striping. It is a disk array with enhanced performance with interchange and without fault tolerance. RAID 0 is not a real array because there is no redundancy in it and that is why it is not common in server environments. Striping distributes the contents of the files equally among all the disks in the set. It makes the parallel read or write operations to multiple drives and almost inevitably leads to increased productivity. In server environments, it used together with other RAID technologies. (SNIA Technical Position. 2009, 13-31.)

**RAID 1** consists of data mirroring. Data is written identically to two disks, resulting in a "mirrored set" of drives. With this method, any drive in the set can service a read request thereby improving performance. However, actual read throughput of most RAID 1 implementations is slower than the fastest drive. And write throughput is always slower because every drive must be updated, and the slowest drive limits the write performance. If the damage is happened the array continues to operate as long as at least one drive is operating. (SNIA Technical Position. 2009, 13-31.)

**RAID 5** consists of block level striping with distributed parity. Data is distributed between all the disks in the array. This type of RAID requires at least three disks because if one disk in the pair is fails, the third disk in the array takes its place. This kind of implementation is susceptible to system failures. But while rebuilding the array it is needed to read all the data from all the disks, and there is a possibility to lose the whole array if a second drive fails. (SNIA Technical Position. 2009, 13-31.)

**RAID 6** consists of block level striping with double distributed parity. RAID 6 requires a minimum of four disks. It is almost like RAID 5, but double parity provides fault tolerance up to two failed drives. This type of array is mostly used

for a high-availability systems and large drive capacities. (SNIA Technical Position. 2009, 13-31.)

**RAID 01** combines RAID 0 and 1 together, what means it creates two stripes and mirrors them. If a single drive failure occurs, then it means that one of the stripes has failed and RAID 0 is running with no redundancy. (SNIA Technical Position. 2009, 84-92.)

**RAID 10** also combines RAID 0 and 1 together, but instead it creates a striped set from a series of mirrored drives. The array can sustain multiple drive losses so long as no mirror loses all its drives. (SNIA Technical Position. 2009, 84-92.)

**RAID 50** combines the straight block level striping with the distributed parity. Minimal RAID 50 configuration requires six drives. In RAID 5 the reliability of the system depends on quick replacement of the failed data drive, so it is common to include hot spares that can immediately start rebuilding the array upon failure. This level is recommended for applications that require high fault tolerance, capacity and random access performance. (SNIA Technical Position. 2009, 84-92.)

## 4.2 SAN

Storage Area Network (SAN) is the storage sharing technique which enables sharing the storage resources across multiple servers in the network. It provides block-based storage and uses the client's file system. SAN is represented by specialized high speed, high availability and shared storage devices using fiber channel technology to connect servers to storage disks. All servers in SAN can be managed centrally. There are different types of SANs based on the protocols they support. The types are:

**FC SAN** (Fibre Channel SAN) uses fibre channel protocol for the communication between servers and data storage devices. The FC interconnectivity supports point-to-point (P2P), fibre channel arbitrated loop (FC-AL) and fibre channel switched fabric (FC-SW). In point-to-point type, only two devices can

communicate at any point of time and it is not scalable. In FC-AL type, devices can be attached on the shared loop by using star or ring topology. Only one device can perform I/O at a time and can support only 126 devices at a time. In FC-SW all nodes on a logical space created on switches with unique domain identifier can communicate with each other. (Mushtaq N. U. 2016.)

**IP SAN** (IP based SAN) uses existing IP networks for transmitting the data. IP based protocols iSCSI and FCIP are used. iSCSI connects host and storage by encapsulating the SCSI commands and data into an IP packet and transports them using TCP/IP. (Mushtaq N. U. 2016.)

**FCoE SAN** (Fibre Channel over Ethernet SAN) combines LAN and WAN traffic over a single physical infrastructure. FCoE protocol is used for the communication. (Mushtaq N. U. 2016.)

Each SAN consists of nodes, storage devices and the fabric to connect them. Node is any device connected to the SAN (server, tape drive etc.). Also, all storage devices have a World-Wide Name (WWN) that uniquely identifies them. And a fabric contains all hardware that connects servers and workstations to storage devices by using fiber channel switching technology. (Mushtaq N. U. 2016.)

SAN has no restrictions on the amount of data that can be accessed by an individual server, as there are servers with direct attached disks. The storage can be accessed by multiple servers simultaneously with fast processing. And one more advantage is that storage resources can be centrally managed. (Mushtaq N. U. 2016.)

### **4.3 NAS**

Another storage sharing technique is Network Attached Storage (NAS). It is storage server directly connected to the network. NAS provides only file-based data storage services. It has its own filesystem that is provided to the client and appears as file server. Control and configuration of NAS is provided over the

network with use of browser. Usually It is not designed to be a general-purpose server thus more storage can be added to the network without any impact on the server operations. (Mushtaq N. U. 2016.)

The popular protocols used by NAS are NFS (Network File System) on Unix Systems, SMB (Server Message Block)/CIFS with windows operating system etc.

In NAS, adding more storage has no impact on operations done using the storage. Also, NAS can be easily managed through the Internet, which accelerates and facilitates configuration and management. One more benefit is that client can map network drives to shares on the NAS server. (Mushtaq N. U. 2016.)

#### **4.4 NAS-SAN hybrid**

The hybrid technology of SAN and NAS can be used. The difference between NAS and SAN is that NAS storage appears to the client as a file server whereas SAN storage appears to the client as a disk storage, thus can be formatted with a filesystem and mounted. So, NAS-SAN hybrid offers both, file level protocols (NAS) and block level protocols (SAN) from the same system. A shared disk filesystem can be run on top of SAN to provide filesystem services. This means that SAN which has "NAS head" can be bought and the same SAN can be used for both. (Desmond M. 2003.)

### **5 VIRTUALIZATION**

One more technology used in the cluster environment is virtualization.

Virtualization means creation of virtual version of operating system, server, storage device or network resources, which are abstracted away from the true underlying hardware or software. It can be implemented on different layers of network. The example of storage virtualization is a partition of the hard disk. And, example of OS virtualization is when a hardware runs multiple operating system images at the same time. Actually, key use of virtualization technology is server virtualization, which uses a software layer called a hypervisor to imitate the

underlying hardware. But the concept of virtualization has spread away to applications, networks, data and desktops. (Rose M. 2016.)

### **5.1 Virtualization in server network**

Virtualization in server environment is done by dividing a physical machine into smaller virtual servers to help maximize machine's resources. Server resources, such as number and identity of individual physical servers, processors, and operating systems are hidden from users. Special software application is used to divide one physical server into multiple isolated virtual environments. Server virtualization includes storage virtualization, network virtualization, and workload management. Server virtualization can be used for more efficient use of server resources, improved server availability, testing and development, to assist in disaster recovery and to centralize server administration. (Rose M. 2009.)

### **5.2 Virtual machines and their roles**

Virtual machine (VM) is a software environment for imitating an OS. It imitates hardware resources, such as CPU, memory, hard disk, network, etc. Imitation is done under specialized software that is called a hypervisor. The hypervisor is also called a virtual machine monitor (VMM). Virtual machine's work is based on the host/guest model. It requires a real computing resources from the host and hypervisor is used to coordinate instructions to the CPU. With VMs one physical device can be split into multiple different operating systems isolated from each other for different purposes. This technique commonly used on server nodes, where one node runs few virtual machines with different roles and applications. Also, it allows the administrator to create guests that use different operating systems for their needs without any knowledge of that, because the real physical resources are hidden. VMs can be easily configured to share resources. (Rose M. 2016.)

Virtual machines do not need specialized, hypervisor specific hardware. However, it is required more bandwidth, storage and processing capacity to host multiple running virtual machines on one physical device. Nevertheless, hardware



is used efficiently and VMs decrease the amount of hardware that is used and reduce the cost.

The advantage of virtual environment is that backup, deployment and basic system administration management are simplified. Also, VMs have a disaster recovery. They can be easily moved, copied and reassigned between host servers through the network. In addition, virtual hardware does not fail. (Rose M. 2016.)

Also, use of virtual machines create important considerations. There are some risks to consolidation, including overtasking of resources and potentially experiencing outages on multiple VMs due to one physical hardware outage. While cost, savings increase, as more virtual machines share the same hardware platform, it does add risk. It is possible to place hundreds of virtual machines on the same hardware, but if the hardware platform fails, it can take out dozens or hundreds of virtual machines. (Rose M. 2016.)

### **5.3 Memory virtualization**

Memory virtualization technique allows virtual machines running on one node to share a single memory pool to overcome the limitations of physical memory and increase their performance. When memory virtualization appears, volatile random access memory (RAM) resources disconnect from individual device in the system, and then integrate those resources into a unique virtualized memory pool that is shared across all the VMs in one node. The distributed memory pool can be utilized as a high-speed cache, a messaging layer, or a large, shared memory resource for CPU or GPU applications. With memory virtualization technique, applications can use a large amount of RAM to improve performance, system resource utilization and improve memory utilization efficiency. The memory pool may be accessed at the application level or OS level. At the application level, the pool is accessible through an API or as a networked filesystem and builds a high speed shared memory cache. At the OS level, a page cache can be utilized as a very large pool of memory resources, which is much faster than local or networked storage.

Advantages of memory virtualization are the following:

- Improvement of memory utilization via sharing of memory between insufficient resources.
- Increase of efficiency and decrease of run time for intensive data and I/O bound applications.
- Applications on multiple servers share data without replication thereby decreasing total memory needs.
- Reducing delays and providing quicker access.

#### **5.4 Storage virtualization**

Also, virtualization can be done in storage environment. The technology saves resources of the system. Storage virtualization means combining of physical storage from multiple network storage devices into a single storage device that is managed from a central console. This technology is complicated and can be applied on different levels of network environment such as hardware, software applications or hardware and software hybrid appliances. There are different types of storage virtualization based on the level they are applied: host-based, array-based, OS-level, file-system and Fibre Channel storage virtualization. (Posey B. 2016, TechTarget. 2008.)

- **DRBD**

DRBD is a software which represents a distributed replicated storage system for the Linux platform. It is the easiest way to implement the storage virtualization and it. The technology makes a full mirroring of the all operations of a block device using an IP network. DRBD takes the data, records it to a local disk and sends to another host. Then the data is also written on that another host. DRBD provides synchronization between a local and a remote block device. Read requests can be carried out locally, not causing network traffic. (Proxmox Server Solutions GmbH. 2017)

DRBD technology is practically used to build a failover cluster system, but it can also be used to create large software defined storage pools with a focus on cloud integration. Figure 4 illustrates the path of data in DRBD kernel Drive

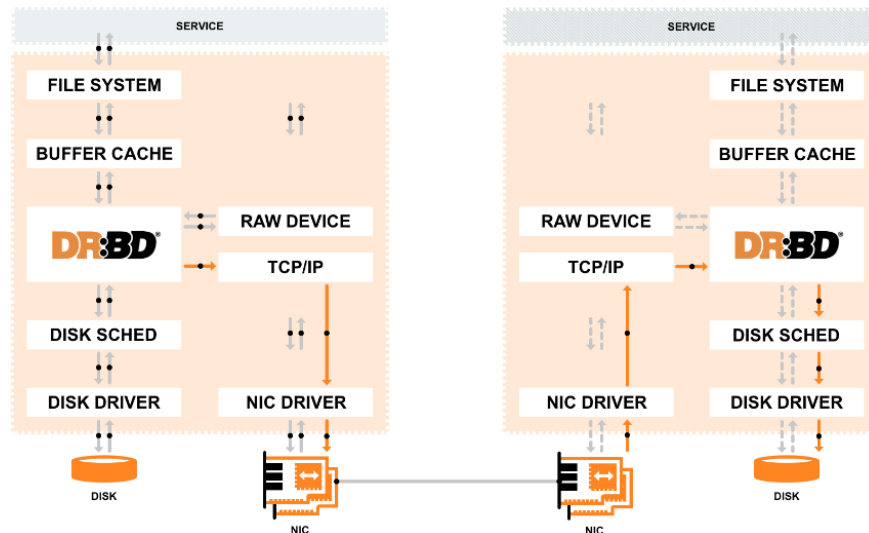


Figure 4. Path that data takes within the DRBD kernel drive

Each DRBD host (there can be a lot of DRBR hosts at the same time) can exist in one of two states: primary and secondary. Data from the primary server is replicated to the secondary server. If the primary server failures, the clustering software shifts the ownership to the secondary server, which uses the duplicate drives to run the applications and data. (Haas F., Reisner P., Ellenberg L. 2011.)

## 5.5 Live migration

In the server environment, if one node fails, the VMs that were running on it can be migrated or in other words transferred to another node and then launched there. This process is called a migration. Before the migration, the virtual machine's state should be saved and its process should be stopped. While the transferring process, virtual machine is offline and applications are not available. This time, when the machine is not available is called a downtime. To overcome it, there is a live migration which lets the virtual machine to be available with a minimized downtime. (Rose M. 2006.)

Live migration technique is used in high-availability cluster environment. Live migration is the movement of a virtual machine or application from one physical machine to another while it is running without disconnecting the client or application. Also, memory, storage, and network connectivity are transferred. While the process is running, users get no effect on their work and do not recognize it. This process allows to upgrade or maintenance the system without any downtime. Also, live migration can be used to make a dynamic infrastructure of the system. VMs can be moved from more loaded servers to the less loaded servers using a load balancing in order to optimize the utilization of available CPU resources. (Rose M. 2006.)

There are two techniques to move the virtual machine's memory state from the source to the destination hosts.

- **Pre-copy memory migration**

Stage 0: Pre-Migration - Begins with an active VM on physical host A. Target host may be preselected and resources required to receive migration will be guaranteed to speed up a process.

Stage 1: Reservation - Request to migrate an OS from host A to host B is issued. On this step, confirmation is made and the necessary resources are available on B and VM container of that size is reserved. The VM on host A continues to run to secure resources from failure.

Stage 2: Iterative Pre-Copy - During the first iteration, all pages are transferred from A to B. But the VM is running and some pages can be changed. So, next iterations occur to copy the original state of the VM. Next iterations copy only those pages which are differ from the previous transfer phase.

Stage 3: Stop-and-Copy - The running OS on host A is stopped and its network traffic is redirected to host B. Then, CPU state and all the remaining inconsistent memory pages are transferred. At the end of this step there is a copy of the VM

at both hosts A and B. The copy at host A is still considered to be primary and is resumed in case of failure.

Stage 4: Commitment - Host B indicates to host A that it has successfully received an OS image. Now host A may discard the original VM, and host B becomes the primary host.

Stage 5: Activation - The migrated VM on host B is activated. Post-migration code runs to reattach device drivers to the new machine and advertise moved IP addresses.

(Clark C., Fraser K., Hand S., Hansen J. G., Jul E., Limpach C., Pratt I., Warfield A. 2005.)

- **Post-copy memory migration**

Post-copy VM migration is initiated by stop for a time the VM at the source. While the source VM is stopped, the execution state of it is transferred to the target host. Then the VM is resumed at the target. Concurrently, the target VM is actively copying the remaining memory pages of the source host. At the target, if the VM tries to access a page which has not yet been copied, it generates a page fault. These faults are gathered at the target host and redirected to the source host. Each page is sent exactly once over the network in Post-copy migration technique. The VM's state is distributed over both source and destination hosts. If the destination fails during the migration, the VM cannot be recovered. (Hines M. R., Deshpande U., Gopalan K. No date.)

The time between stopping the VM on the source host and resuming it on destination is called downtime. There are some techniques to reduce live migration downtime. If downtime of a VM during a live migration is not noticeable to the end user, it is called a seamless live migration.

While using a Live Migration two hosts can be used only in one process of migration of only one VM. This means that in N-hosted cluster, there can be

simultaneously happen  $N/2$  migration processes. It is not possible to migrate few VMs from one host nor few VMs to one host at a time. (Hines M. R., Deshpande U., Gopalan K. No date.)

Common requirements for any form of live migration are the following:

- The system should support hardware virtualization.
- Processors from the same manufacturer should be used.
- Computers should belong to the same domains which trust each other.
- Virtual machines must be configured to use virtual hard disks or virtual Fibre Channel disks (no physical disks).

## 6 CLUSTERING

**Cluster** is grouping of two or multiple physical servers that are running and working together as one single system providing high availability of services for clients.

**High-availability cluster (HA Cluster)** is the type of cluster that supports a failover and in which the downtime is minimized. Servers in the HA clusters use high-availability software that provides redundancy.

Cluster technology uses virtualization technology and needs a shared storage to operate. All the nodes in the cluster must be connected to one shared storage. (Technopedia Inc. No date.)

### 6.1 Operating principles

Nodes in the cluster work together as one server to provide redundancy and load balancing. Each node in the cluster has the information whether the other nodes are online or available. Each server in a cluster runs the same critical applications. Thus, if one server fails the operation is resumed immediately on the other node. This is called a failover. Also, there is another process, called a failback, which

takes place when a failed server automatically goes online and performs its previous operations. Server clusters give users a constant access to important server based resources. (Technopedia Inc. No date.)

## 6.2 Virtual clusters

Clusters are made for high availability. And with virtualization technology they become even more effective and flexible. Virtual clusters are built with virtual machines installed on one or more physical servers. In this way, VMs are logically interconnected by a virtual network. Figure 5 shows the example of platform that has four virtual clusters made of three physical clusters divided differently. (Hwang K., Dongarra J., Fox G. 2011.)

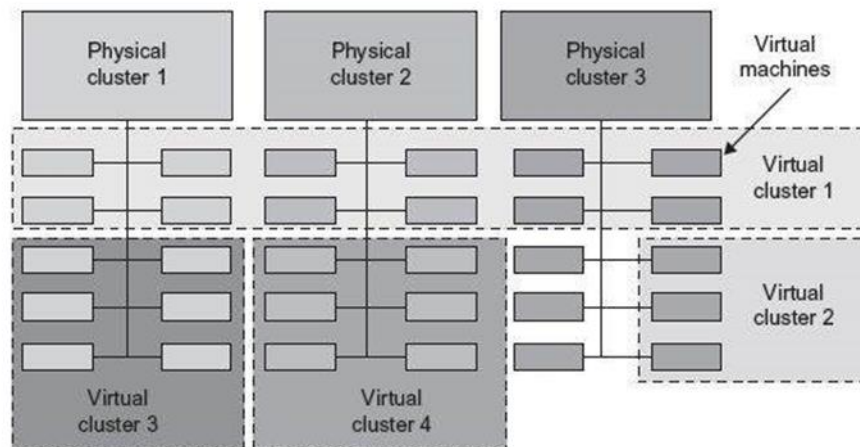


Figure 5. Example of virtual cluster

Clustering technology combined with virtualization provides the following features:

- Multiple VMs running different OSes can be deployed on the same physical node.
- Guest OS can differ from the host OS.
- VMs consolidate multiple functionalities on the same server. This greatly enhances server utilization and application flexibility.
- The number of nodes of a virtual cluster can grow dynamically.

- If any physical node fails, it might disable some of the VMs installed on the failing nodes. However, any VM failure will not pull down the host system. (Hwang K., Dongarra J., Fox G. 2011.)

Virtual clusters provide high performance. VMs can be installed on several physical hosts using a template which can contain the same configurations, such as name, disk image, network setting, allocated CPU and memory. This saves the time in administration of the cluster. Each user on the network can have his own profile that stores data block identification for corresponding VMs. When users modify the corresponding data, new data blocks are created. (Hwang K., Dongarra J., Fox G. 2011.)

### **6.3 Network requirements**

To implement the high-availability cluster environment it is required to have the same shared storage and data stores accessed by all hosts in the cluster, and it should be used by all virtual machines on the cluster. Also, all hosts should have the same virtual networking configurations, and when a new network component is added to one host, the same component should be added to all hosts in the cluster. Moreover, all hosts in the cluster should use DNS names to resolve the other hosts. In addition, there are some application requirements, because not every application can run in a high-availability cluster environment. Applications should have an easy way to start, stop, force stop and check the status. Applications should use a shared storage and store its state on non-volatile shared storage as much as possible. Also, the application should not corrupt data, if it crashes, and restart on another node at the last state by using the saved state from the shared storage. (Lowe S. 2010, Microsoft. 2010.)

### **6.4 Quorum**

To form a cluster, the quorum should be done on the majority of cluster members. If the majority of members vote, then it means that quorum is done successfully and cluster can function. If the running cluster loses too many votes this would mean that quorum is lost, and the cluster would suspend the



operation. In this case, connection manager continues to run to accumulate enough votes, so that the cluster can resume its operation. In the other words, quorum determines the number of failures that the cluster can sustain. Most recommended configuration of the quorum is the Node Majority and Node and Disk majority. It can sustain failures of the half of the nodes (rounding up). Quorum is necessary in the cluster, because if problems in communication between the nodes occur, the cluster splits into parts, where set of nodes cannot communicate with each other. In this situation, at least one of the sets of nodes should stop running as a cluster. To prevent issues that can be caused by the split, the cluster software requires that any set of cluster nodes should use a voting algorithm, which is based on the specific quorum configurations, to determine whether the set of nodes has a quorum.

Also, there is a cluster quorum disk that defines quorum algorithm and stores configuration database of the cluster. The quorum disk contains shared block device that allows simultaneous read/write access by all nodes in the cluster. (Fafrak S., Lola J., O'Brien D. 2002, Microsoft. No date.)

## **6.5 Fencing**

In the cluster environment, any node can crash or stop working correctly. If this occurs, fencing is used. Fencing is the process of isolating a node or protecting shared resources when a node is not working correctly.

If one node fails, it may have control over shared resources which need to be reclaimed and the rest of the system needs to be protected. Fencing may be done by different methods: disable the node, or disable shared storage access to the node. Figures 6 and 7 show that fencing agent sends the signal to power controller and that controller disables Node D.

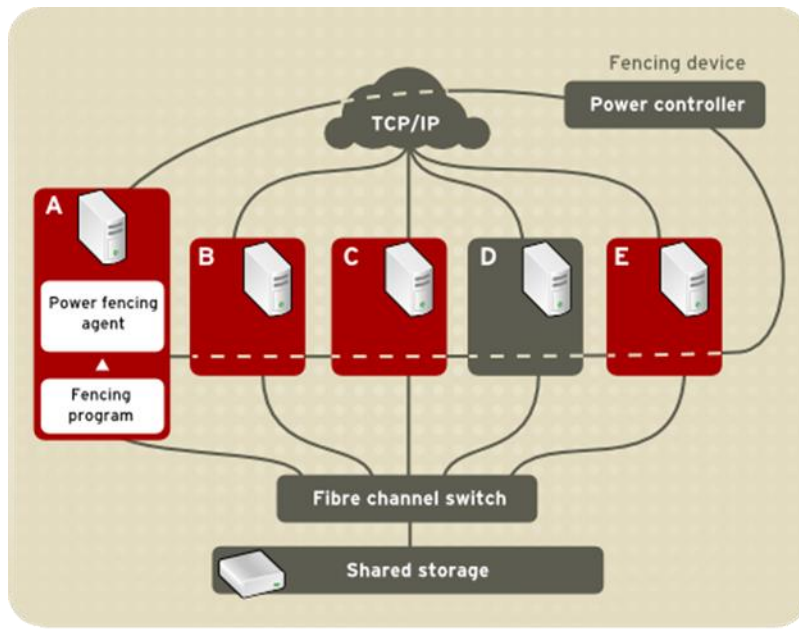


Figure 6. Powering off Node D with power controller.

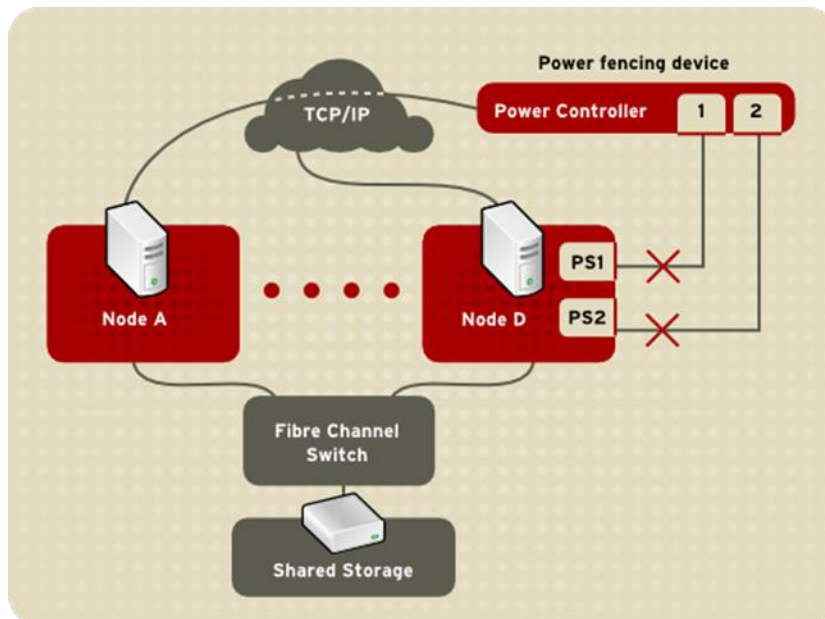


Figure 7. Powering off Node D with power controller.

Figure 8 shows how the node can be disabled from the shared storage. In this example, it is fibre channel shared storage and Node D is disabled from the access to the FC switch.

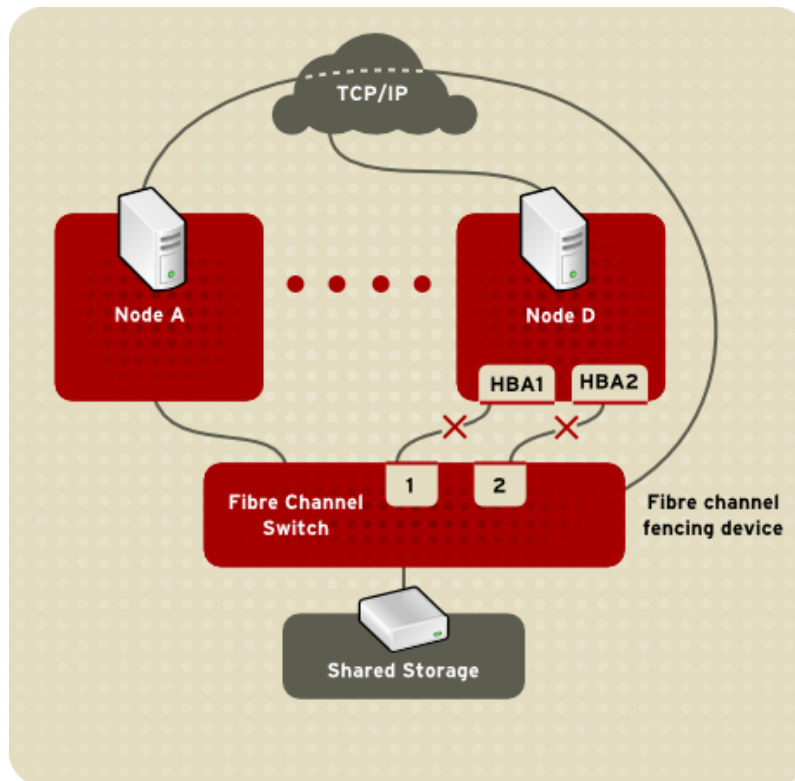


Figure 8. Disabling access to the shared storage

Isolating a node is done by blocking I/O from it. Fencing is typically done automatically by cluster infrastructure such as shared disk file systems to protect processes from other active nodes modifying the resources during node failures.

Also, there is a STONITH mechanism that fences failed node by resetting or powering it down. In the cluster, which has only two nodes, the reserve/release method may be used. If Node A is detected to be failed it will be disabled, if it tries to do I/O operations, and then Node B reserves and obtains all the resources. (RedHat Inc. 2007.)

## 6.6 Reliability of HA cluster

High-availability cluster environment requires the whole system to be stable and reliable. HA cluster uses all the available techniques to make the system and shared infrastructure to be as reliable as possible. This includes: disk mirroring (if internal disks fail, it does not lead to system crashes); redundant network connections (if some network component fails, it does not lead to network

outages) and redundant SAN/NAS connections (if some network component fails, it does not lead to loss of connectivity to the storage). These features help to minimize the chances that the unavailability between systems will occur.

## 7 BACKUP

Each server environment needs a backup process to be done regularly. Backup is the process of creating a data copy on a storage medium designed to restore data, if damage occurs. It restores the data in the original or new location.

Backup is a quick and inexpensive way to restore the information. Backup is used to protect an OS, system state, volumes, files, and application data. Backups can be saved to tape drives, DVDs and CDs, single or multiple HDD disks, USB compatible devices, remote shared folders, or in the special cloud systems. It can be scheduled to run automatically or manually. (Microsoft. No date.)

**Requirements** for the backup system: Storage should be reliable and fault tolerant, duplication of information and replacement of a lost copy should be made. It should support multiplatform systems; servers should be able to work in different operating environments and support clients on a variety of hardware and software platforms. It should have autonomy and easy operation. It should have a fast implementation, easy installation and configuration. (Microsoft. No date.)

**The key backup objectives** are **RPO**, Recovery Point Objective and **RTO**, Recovery Time Objective. RPO defines the rollback point - the point of time in the past to which the data will be restored, and the RTO defines the time it takes to restore the backup.

There are different types of backup exist. The types are the following:

- **Full backup.** It usually copies the entire system and all the files. It involves the creation of a complete copy of all data. Usually, it is performed when copying a large amount of data does not affect the work of the organization. Full backup is required when there is a need to prepare a copy of the system to quickly restore it from the scratch. (TGRMN Software. No date).

- **Differential backup.** In this type of backup each file that has been changed since the last full backup is recopied each time. Differential backup speeds up the recovery process of the system. All copies of files are made at certain points of time. (TGRMN Software. No date).
- **Incremental backup.** In these backups, only the files that have been changed since the last full or incremental backup are copied. Incremental backup takes less time, because fewer files are copied. However, the process of restoring the data takes more time, because all the data from the last full backup should be restored, as well as the data from all subsequent incremental backups. Unlike in differential backup, in incremental backup changed or new files do not replace old ones, but are added to the storage independently. (TGRMN Software. No date).
- **Real Time backup.** It allows creating copies of files, directories and volumes without interrupting the work and without restarting the computer (Posey B. 2010. WWW document.)

## 8 PROXMOX VE

In this thesis, to form a high-available cluster environment in the practical part, Proxmox VE server virtualization platform was used. Proxmox VE software is developed and maintained by Proxmox Server Solutions GmbH. Proxmox Virtual Environment is one of the products of Proxmox Server Solutions that provides an open source server virtualization software solution. Proxmox VE is based on the Linux kernel which is a stable and reliable platform for the servers. Also, Linux supports a wide range of hardware configurations – from cheap testing to high performance systems. (Proxmox Server Solutions GmbH. No date.)

### 8.1 Principle of operation

Proxmox VE combines two virtualization technologies under one platform – kernel-based virtual machine (KVM) virtualization and container-based virtualization with Linux Containers (LXC). It manages KVM virtual machines, Linux containers (LXC), storage, virtualized networks, and HA clusters. KVM is used for full virtualization for different types of operating systems, and lightweight

containers are used to run conflict free Linux applications. (Proxmox Server Solutions GmbH. 2016.)

## 8.2 Features of Proxmox VE

Proxmox VE has the following operational features:

- KVM and Containers

Proxmox VE is based on the Debian GNU/Linux Distribution. It uses two virtualization technologies: KVM and Linux Containers.

**Kernel-based Virtual Machine (KVM)** is a free, open source virtualization solution for Linux distributions x86 hardware. The user space component of KVM is included in mainline QEMU. KVM contains virtualization extensions such as Intel VT or AMD-V. It can run multiple virtual machines running unmodified Linux or Windows images. Also, each virtual machine has private virtualized hardware: network card, disk, graphics adapter, etc. (Proxmox Server Solutions GmbH. 2016.)

**QEMU** is an open source machine emulator and virtualizer. When it used as a machine emulator, it can run OSes and programs made for one machine on a different machine. When it used as a virtualizer, it executes the guest code directly on the host CPU. QEMU supports virtualization under the Xen hypervisor or using the KVM kernel module in Linux. (Proxmox Server Solutions GmbH. 2016.)

**Container** is the lightweight alternative to KVM with lower overhead. LXC is an OS level virtualization environment. It allows to run multiple isolated Linux systems on a single Linux control host. LXC works as a user space interface with Linux kernel features. System or application containers can be created and managed with API and simple tools. (Proxmox Server Solutions GmbH. 2016.)

- Management

In Proxmox VE nodes in the cluster can be managed centrally. Management tasks are done with the web-based management interface, which comes with the default installation and no installation of a separate management tool is needed. It allows to manage VMs and containers, storage or even the whole cluster from any node in the cluster. Furthermore, privileges and access for all objects in the cluster can be defined by using the role based permission management. Also, for advanced users Proxmox VE provides a CLI to manage all the components of the virtual environment. (Proxmox Server Solutions GmbH. 2016.)

- Backup and Restore

Backups in Proxmox VE are always full backups that contain the configuration and all the data. Backups can be easily done from the GUI or command line. Backup tool creates an archive of the data that includes the configuration files. Also, backups can be scheduled and executed automatically on specific time, for selected nodes and guest systems. (Proxmox Server Solutions GmbH. 2016.)

- High availability

The Proxmox VE provides a HA cluster environment. Resource manager monitors all virtual machines and containers on the whole cluster and automatically gets into action, if one of them fails. It does not require configuration and works out of the box. (Proxmox Server Solutions GmbH. 2016.)

- Firewall

Proxmox VE has a built-in firewall. Firewall's rules can be setup for all hosts inside a cluster, or defined for virtual machines and containers only. The firewall contains the following features: firewall macros, security groups, IP sets and aliases. It supports IPv4 and IPv6 and filters traffic for both protocols by default. (Proxmox Server Solutions GmbH. 2016.)

- Networking

Proxmox VE uses a bridged networking model. Bridges attached to physical network cards, assigned a TCP/IP configuration for connecting VMs to the outside of the network. For more flexibility, VLANs and network bonding are possible. (Proxmox Server Solutions GmbH. 2016.)

- Storage

In the Proxmox VE storage is flexible. Images of virtual machines can be stored on one or several local storages or on the shared storage, like NAS and SAN. All storage technologies that are available for Debian Linux, can be used. Running VMs that are stored on shared storage are capable for live migration without a downtime. (Proxmox Server Solutions GmbH. No date.)

### 8.3 Advantages and disadvantages

**Advantage** of Proxmox VE is that it is an open source software and has no vendor lock-in. It has web-based management interface and simple in deployment. It runs on top of a Debian that can be used for several other functions. Proxmox VE can use lots of files systems and support different guest operating systems. It has low possibility to lose a CPU/RAM power. Also, administration costs are low.

**Disadvantage** of Proxmox VE is that it runs only with Linux. It is not so easy to set it up, and creating of the cluster is complicated and takes some time. Also, Proxmox can be downloaded and installed for free, but to upgrade the system or to use additional packages, it is needed to buy a subscription.

### 8.4 System requirements

To use all the features of Proxmox and create a HA cluster, it is needed to meet the following hardware requirements:



- CPU: 64bit (Intel EMT64 or AMD64)
  - Intel VT/AMD-V capable CPU/Mainboard (for KVM Full Virtualization support)
  - 8 GB RAM or more
  - Hardware RAID with batteries protected write cache (BBU) or flash protection
  - Fast hard drives
  - Two or more Gbit NIC (for bonding), additional NIC's depending on the preferred storage technology and cluster setup
- (Proxmox Server Solutions GmbH. 2016.)

But, if only the testing of the system is needed, the hardware should meet the following minimum requirements:

- CPU: 64bit (Intel EMT64 or AMD64)
  - Intel VT/AMD-V capable CPU/Mainboard (for KVM Full Virtualization support)
  - Minimum 1 GB RAM
  - Hard drive
  - One NIC
- (Proxmox Server Solutions GmbH. No date.)

## 8.5 Comparison with other platforms

Proxmox VE is one of the server virtualization platforms. Also, there are other server virtualization software alternatives, such as VMware's vSphere, Microsoft Hyper-V, Citrix XenServer, Linux KVM, and others. The full feature comparison table of different platforms is shown in the Appendix 1. The main features shown in the table are: size of environment in which the platform can be used, the hypervisor type, virtualization type, architecture, storage type which the platform supports, platform type it work on, license type, OS type it supports for host and guests, and which management features it has. (Graphiq Inc. No date.)

Following features are the most significant technical specifications of different platforms: hypervisor type, virtualization type, architecture and compatible storage type

There are two **hypervisor type**: **Type 1** is Bare Metal and **Type 2** is Hosted. Type 1 hypervisor runs directly on top of hardware (VMware, ESXi, Citrix, XenServer, and Microsoft Hyper-V.). Type 2 hypervisor operates as an application on top of an existing operating system (KVM, Oracle VM VirtualBox and VMware Server). (Graphiq Inc. No date.)

Also, there are several **types of server virtualization**: Full Virtualization, Paravirtualization, and OS Virtualization, which can be either with Hardware Assistance or without Hardware Assistance.

**Full Virtualization** option allows operating systems on their own virtual machine to coexist on a single server (e.g. VMware ESX/ESXi). In **Para-virtualization** type, each guest OS is made to detect that it has undergone virtualization, so time spent by the OS performing functions that are more difficult to do when virtualized is reduced (e.g. VMware ESX/ESXi, Citrix XenServer, and Oracle VM Server). **Hardware-Assisted Virtualization** type utilizes the facilities of the host's hardware and processing power and makes it possible to utilize certain features of high performance CPUs and allow guest OS to work independently (e.g. Microsoft Hyper-V, VMware ESX/ESXi, and Citrix XenServer). **OS Virtualization** type uses a single OS on a host server. Within the system there exists certain spaces known as containers, each with their own set of resources (e.g. Parallels Virtuozzo and Solaris Containers). (Graphiq Inc. No date.)

**Compatible server architecture** is one of the most important elements to consider as the software must fit the architecture of the platform in order to work properly. (Graphiq Inc. No date.)

And **compatible storage types** are the types of storage that the virtualization software is compatible with. (Graphiq Inc. No date.)

Proxmox VE is the Bare Metal (Type 1) hypervisor, supports full virtualization and OS virtualization, compatible with x64 and x86 architecture and compatible with FC, iSCSI, NFS storage types. (Graphiq Inc. No date.)

More detailed comparisons and features can be found on <http://virtualization.softwareinsider.com/> website (Graphiq Inc).

## 9 IMPLEMENTING A HA CLUSTER

In the practical part of this thesis work I used hardware provided by the company. Debian Jessie as operational system and Proxmox VE as hypervisor were installed on physical servers. Then, two server nodes were combined into one HA cluster.

### 9.1 Technical specifications

The network has two servers, one shared storage and network router. Also, one more computer is used to configure server nodes through the web interface. The environment that I used is shown in Figure 9.

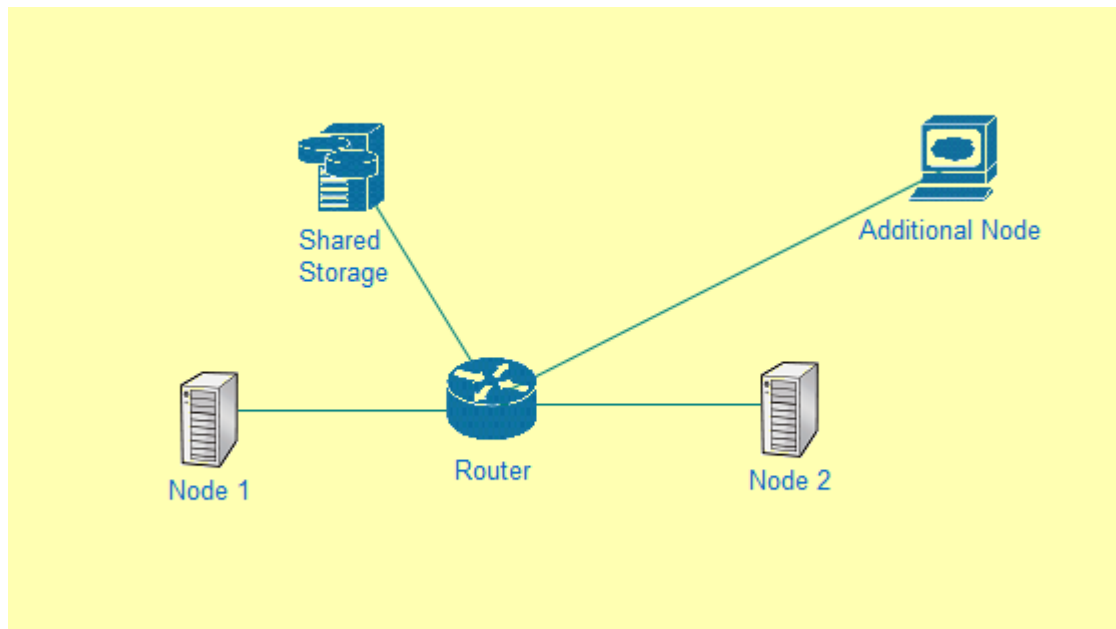


Figure 9. The environment that was used

The network has 192.168.18.0/24 IP address.

**First Server** has 192.168.18.251 255.255.255.0 IP address and is called a pve01. Also, RAID 1 is configured on it. It has the following characteristics:

- Intel Core i5-3337u, Quad-Core
- 2,7 GHz
- 4 Gb RAM
- 500 Gb HDD

**Second Server** has 192.168.18.100 255.255.255.0 IP address and is called a pve02. Also, RAID 1 is configured on it. It has the following characteristics:

- Intel Core i5-2400, Quad-Core
- 3,1 GHz
- 16 Gb RAM
- 120 Gb HDD

In the system, the **network attached shared storage** is used. It has 192.168.18.200 255.255.255.0 IP address. It contains images of VMs and backups.

**Network router** has 192.168.18.0/24 network IP addresses on different interfaces connected to nodes and shared storage.

And **addition computer** connected to the same network is used to connect through the web interface to configure and manage the nodes.

**Note:** For the future configuration of Live Migration the Intel Virtualization should be turned on in BIOS before the start of the nodes. Without it the Live Migration will not appear.

## 9.2 Installing the hypervisor (Proxmox 4.4.) on both physical servers

There are 2 ways to install Proxmox VE.

**The first way** is longer and more difficult one. But it gives the features to manually configure the system. Firstly, Debian Jessie is installed and configured. After that, Proxmox VE 4.4 is installed on it.

**The second way** is simpler and takes about 1 hour. But all the configurations are done automatically. In this way, Proxmox VE already goes on Debian and configurations cannot be done manually. They are already done by Proxmox manufacturer. The image can be downloaded from the official Proxmox website.

In this work, we did the installation in both ways. But, in practice it would be better to use first way, rather than second. It would be better, as the system and hardware resources can be configured and distributed manually with the needs of the system requirements. Also, the installation process can be controlled and we can install and configure the packages and features which we need.

Also, on the nodes hard disks have RAID 1 configured on them. This brings some problems in the installation. When we install Debian and Proxmox separately, we should firstly install Debian on RAID 1, and only then install Proxmox.

### **9.2.1 Installing configuring Debian Jessie OS and installing the Proxmox 4.4 hypervisor.**

First step is to install a standard Debian Jessie (amd64). It is preferred to install a standard package without any additional features, as the Proxmox VE brings its own packages. During the installation, we make a manual partition of hard disks in the section “Guided partition”. The disks should be clean without any partitions. We divide disks into three partitions: boot that has a 100MB, root that has 50GB and one more with the rest size of the disk. The boot partition should have the bootable flag on, formatted with Ext2 filesystem and be a primary partition. The root partition should have a bootable flag on and also be a primary partition. Then, we go in the “Configure software RAID” section to create two separate RAID arrays. After that we choose “Create MD device”. Then, we choose RAID 1

and 2 active devices, we choose two partitions with size of 100MB. And then, we do the same and choose two partitions with size of 50GB and finish. Next step, is to format filesystem on RAID arrays. On the boot partition, we choose a mount option as “/boot” and use Ext2 file system. On the root partition, we choose a mount option as “/” and use Ext4 file system. After that, we choose “finish partitioning and write changes to disk” and continue the installation. In the end, there will be a choice to which disk to install a loader. We choose any disk because later we install the loader on the second disk. It is done, in order the system can boot from each of the disks.

Now, we have Debian OS installed and need to make basic configurations. Also, we need to configure the network parameters. We assign IP addresses as said in the technical specifications.

Then, we install GRUB bootloader on both hard disks with the `dpkg-reconfigure grub-pc` command. In all questions, we leave default values and in the end, we choose both hard disks for the installation.

Now, we can go to the second step and start installing the Proxmox. Firstly, we need to edit “/etc/hosts” file with `mcedit /etc/hosts` command. This file should have only following lines and if not, error during Proxmox installation can occur:

```
127.0.0.1 localhost.localdomain localhost

192.168.18.251 proxmox.local proxmox pvelocalhost
(192.168.18.100 for second node)
```

(Proxmox is the name of the server we chose during the installation. And local means the domain chosen during the installation. 192.168.18.251 and 192.168.18.100 are IP addresses of the server nodes)

Then, we add the Proxmox VE repository with following commands:

```
mcedit /etc/apt/sources.list deb
```

```
http://download.proxmox.com/debian jessie pve-no-  
subscription
```

After that, we add the Proxmox VE repository key with command

```
wget -O- "http://download.proxmox.com/debian/key.asc" | apt-  
key add -
```

Further, we update repository and system and reboot it:

```
apt-get update && apt-get dist-upgrade  
  
reboot
```

Then, we install Proxmox VE system with following command:

```
apt-get install proxmox-ve ntp ssh postfix ksm-control-  
daemon open-iscsi systemd-sysv
```

After the installation, we reboot the system. The new Proxmox VE kernel should be automatically selected in the GRUB menu.

Now, the Proxmox VE is installed and we can connect to it through the web interface and continue with configurations.

**Remember that we do these installations twice on both nodes.**

### 9.2.2 Installing Proxmox 4.4. hypervisor with Debian

This step takes less time to install and configuration is done automatically by installation GUI. Firstly, the image of Proxmox should be downloaded from the

official website. Then, it should be burned to bootable media. After that, we boot the image from the media and start the installation.

During the installation, the name of the host, IP address and other information about node is entered manually, but the partition of hard disks and volumes are created automatically. This installation saves the time, but resources are not allocated as it is needed. At the end, the ready Proxmox VE starts and there is no work with Debian OS. Proxmox VE has no GUI, so for the future convenient work with nodes the graphical web interface is used. Also, the manual configuration can be done from CLI.

With this type of installation, we also install the system twice on both nodes.

### 9.3 Connecting through the web interface

The next step is to connect to both nodes through the web interface. In the browser address line we write the IP address of nodes and we use port 8006 and https. In this case https://192.168.18.251:8006 and https://192.168.18.100:8006 were used. To login into the web interface the username and password should be entered.

Figures 10 and 11 show that there is only one node shown in the datacenter, it means that nodes pve01 and pve02 do not see each other now. And also they have only local storages. Further, we will add them to the datacenter (cluster) and add shared storage to them.

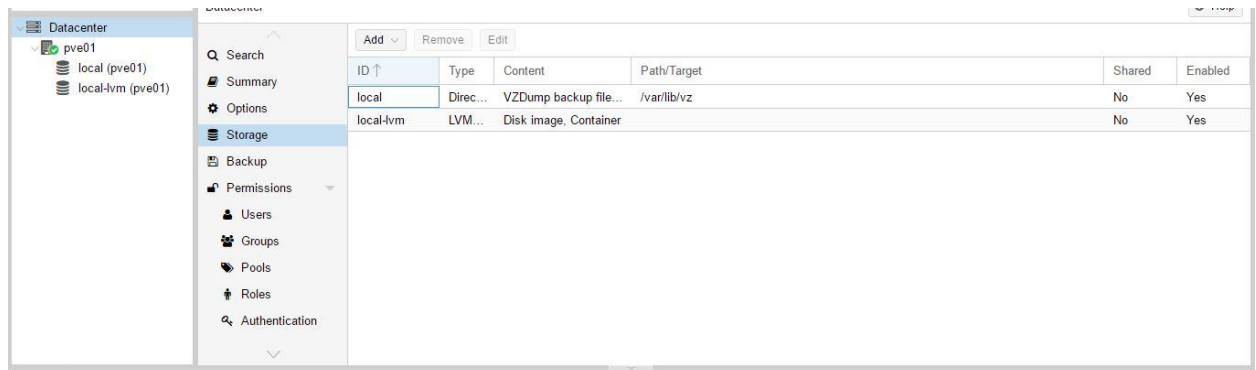


Figure 10. Pve01 node



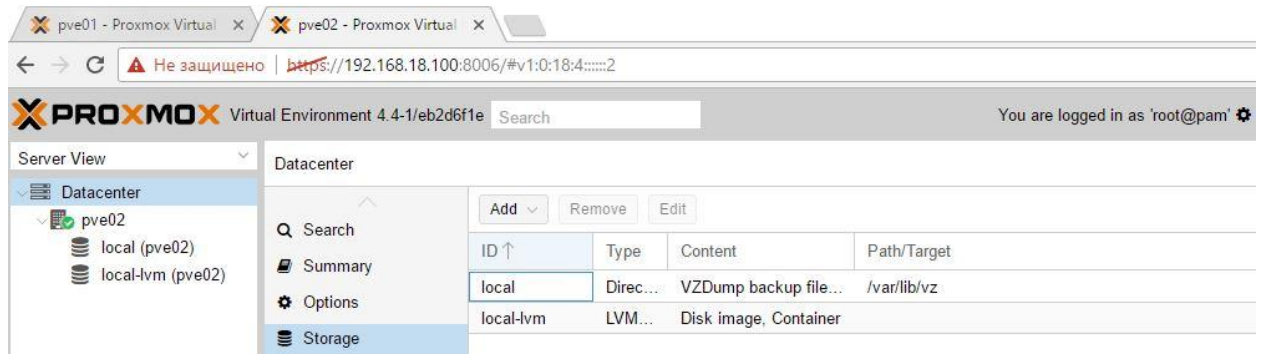


Figure 11. Pve02 node

## 9.4 Combining pve01 and pve02 nodes into the cluster

The next step is to combine nodes into the cluster. Combining is done through the web-interface and shell console with commands. Figures from 12 to 18 show the process of creating the cluster: commands and their output.

The first step is to create a cluster on the first node (pve01) and to add the second node (pve02) to the created cluster. For this purpose, the shell console is used and clustering is done with commands. Also, both nodes should be clear, without any VMs and storage added. Node pve01 is a domain node in this cluster, as we started the configuration from it.

`pvecm create <name>` command is used to create the cluster on the pve01 node.

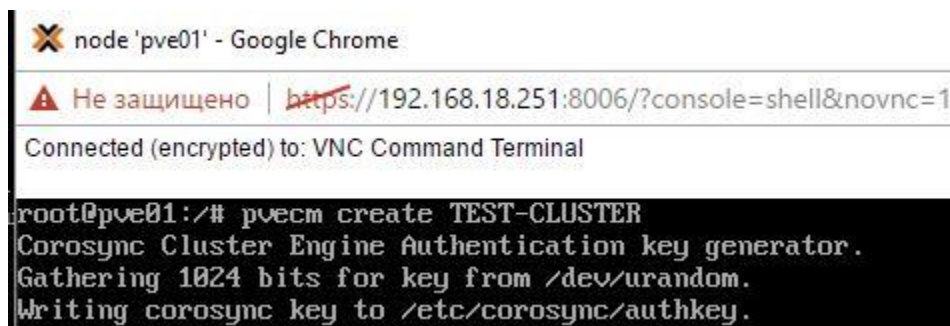


Figure 12. Creating the cluster on the first node

Then, we add the pve02 node and use the `pvecm addnode pve02` command on the pve01 node, so the first node knows about the second node.

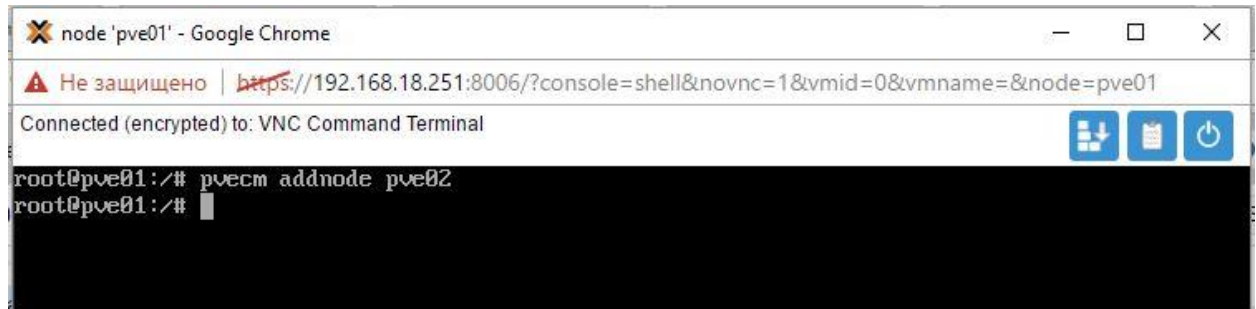


Figure 13. Adding the second node to the first node

After creating the cluster, we check its state with the `pvecm status` command. This is done to see if the cluster has been created.

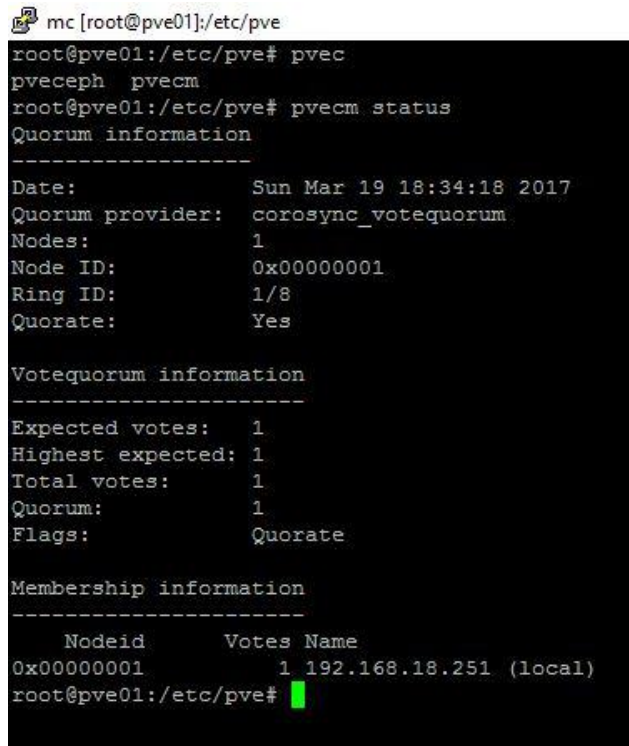


Figure 14. Cluster on the pve01 node is created

The next step is to add the pve02 node to the cluster. The IP address of the pve01 is used for this purpose. Also, password of the pve01 should be entered for the authentication. The `pvecm add 192.168.18.251` (instead of this the name of the node can be used) command is entered from the pve02. As a result, the pve02 node is successfully added to the cluster.

```

root@pve02:~# pvecm add 192.168.18.251
root@192.168.18.251's password:
node pve02 already defined
copy corosync auth key
stopping pve-cluster service
backup old database
waiting for quorum...OK
generating node certificates
merge known_hosts file
restart services
successfully added node 'pve02' to cluster.

```

Figure 15. Adding the second node to the cluster

After adding the second node to the cluster, the `pvecm status` command should show that there are two nodes in the cluster. But sometimes the quorum does not go automatically. Then, a forced quorum should be done. So, if there is only one node shown in the cluster, we make the forced quorum on both nodes, and then check the status again. Forced quorum is done with the `pvecm expected 1` command.

```

root@pve01:~# pvecm expected 1
root@pve01:~# pvecm status
Quorum information
-----
Date:                Sun Mar 19 19:06:34 2017
Quorum provider:     corosync_votequorum
Nodes:               1
Node ID:              0x00000001
Ring ID:              1/16
Quorate:              Yes

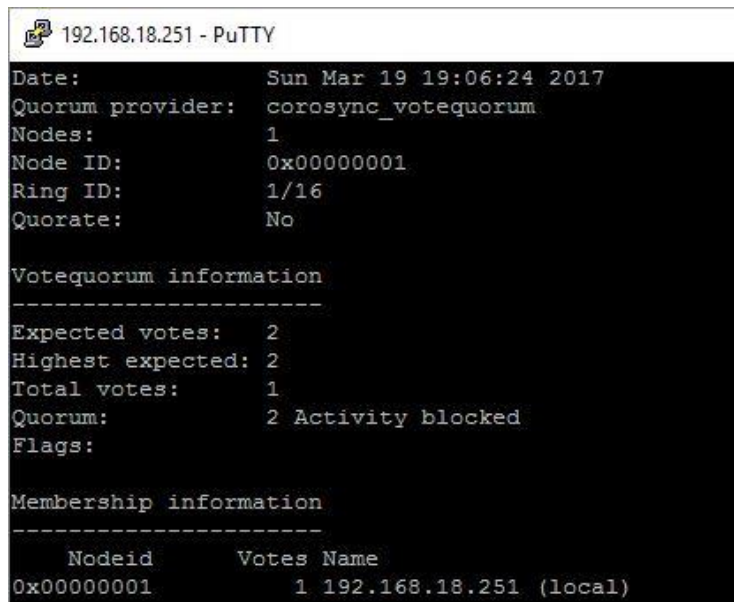
Votequorum information
-----
Expected votes:       1
Highest expected:     1
Total votes:          1
Quorum:               1
Flags:                Quorate

Membership information
-----
    Nodeid    Votes Name
0x00000001    1 192.168.18.251 (local)

```

Figure 16. Checking the cluster status

So, now after entering the command on both nodes and checking the status, we see that there are two nodes in the cluster, because both nodes participate in the quorum.



```

192.168.18.251 - PuTTY
Date: Sun Mar 19 19:06:24 2017
Quorum provider: corosync_votequorum
Nodes: 1
Node ID: 0x00000001
Ring ID: 1/16
Quorate: No

Votequorum information
-----
Expected votes: 2
Highest expected: 2
Total votes: 1
Quorum: 2 Activity blocked
Flags:

Membership information
-----
Nodeid Votes Name
0x00000001 1 192.168.18.251 (local)

```

Figure 17. Nodes combined into the cluster.

Then, we connect from the pve02 to the pve01 through SSH and the authentication is done. For this step to be successful, the keys on both nodes must match.

Figure 18 shows that now nodes are combined into the cluster and they see each other. The web-interface shows them in one datacenter.

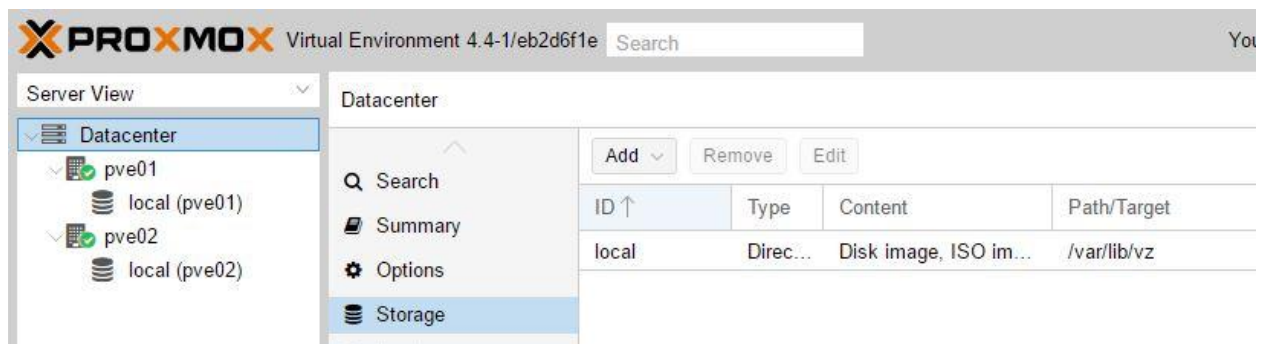


Figure 18. Cluster with two nodes included

For now, nodes have only local storages attached and cannot share data with each other.

## 9.5 Adding the shared storage

In this system, we used a network attached storage (NAS) with NFS protocol. Shared storage is used for making the failover cluster and it stores the images of VMs and backups. With the shared storage, nodes can share the data and do not need to copy it to their local storages. NAS should be configured before attaching it to the cluster. In this thesis work, I did not configure the shared storage, it was already preconfigured by IT-employees of the company. But the basic configurations are: combining disks into RAID arrays, creating of volumes based on the needs of the system, configuring network access and quorum configurations. Also, additional features can be configured such as automatic RAID rebuilding or configuring disks as hot spare. (Posey B. 2015. WWW document.)

There is no need to add the shared storage manually to each node. It is done from the datacenter and is added to all the nodes in the cluster automatically. There are two storage types that can be configured in Proxmox VE: data storage and backup storage which keep the data and backups respectively. To add the storage, we click “add storage icon” in the web-interface. Then, we enter the name of the storage, IP address of the storage, directory of exporting the data and choose a content that will be stored on this storage type.

Then, we add the Data shared storage as shown in Figure 19. The directory of export is /backup/pve-data. And we write the IP address of storage 192.168.18.200.

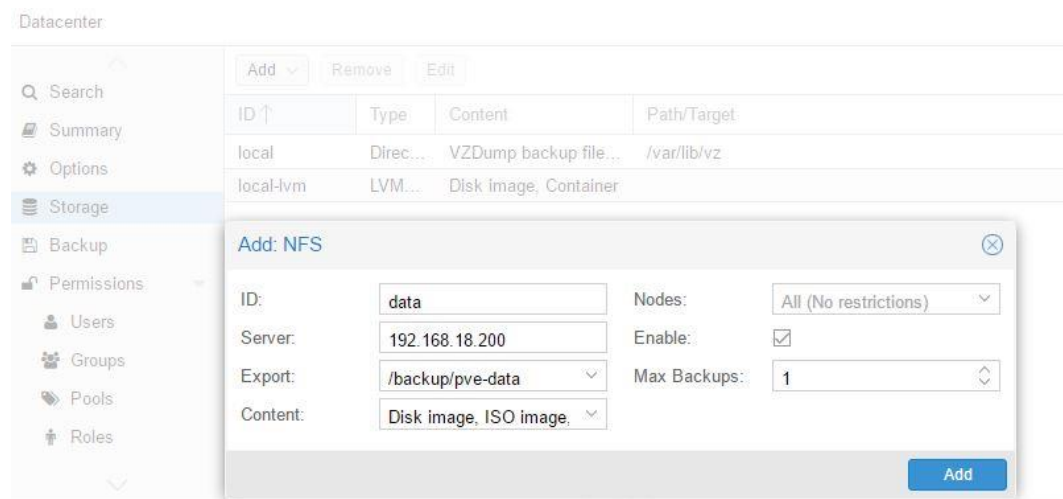


Figure 19. Adding Data shared storage

Then, we add the Backup shared storage (Figure 20). The directory of export is /backup/pve-backup, and we write the IP address of the storage 192.168.18.200.

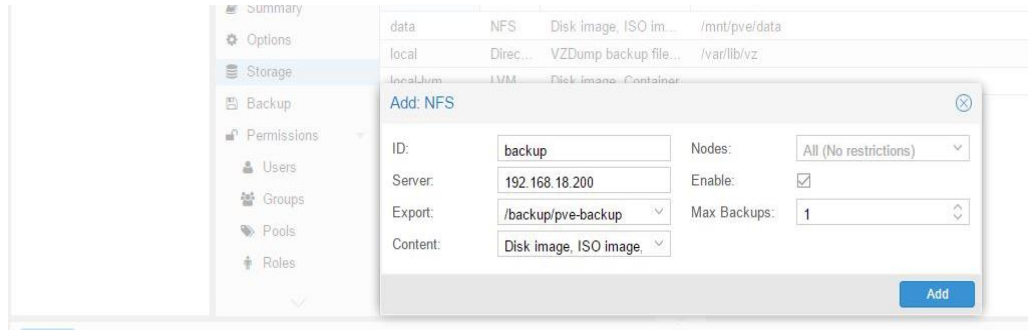


Figure 20. Adding Backup shared storage

These processes add the shared storage to the cluster. And, now it is seen that Data and Backup storages are on both nodes. Also, nodes can share only NFS resources and cannot share local resources. It is seen that both nodes have the same shared storage types and that they are shared (Figure 21).

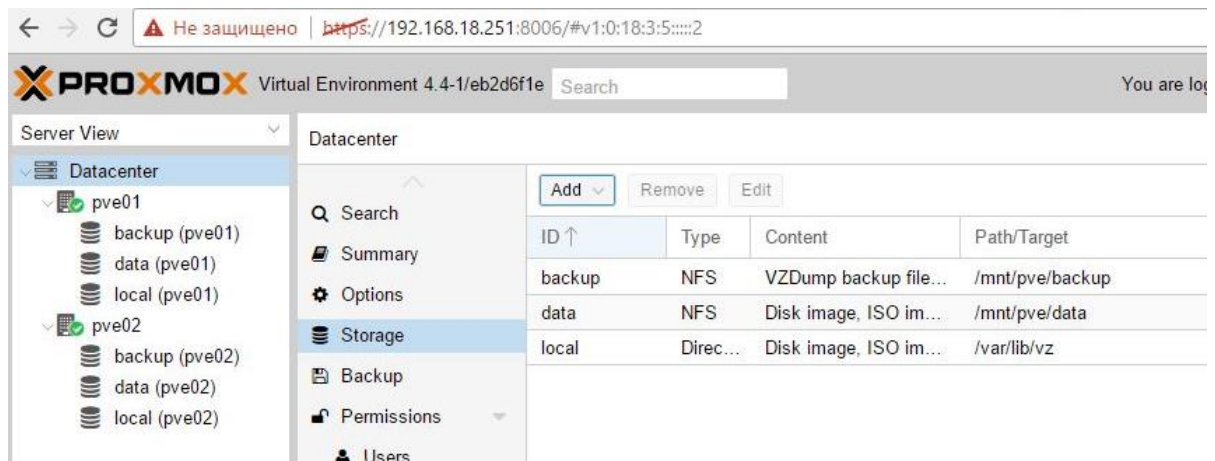


Figure 21. Shared storage added to the cluster

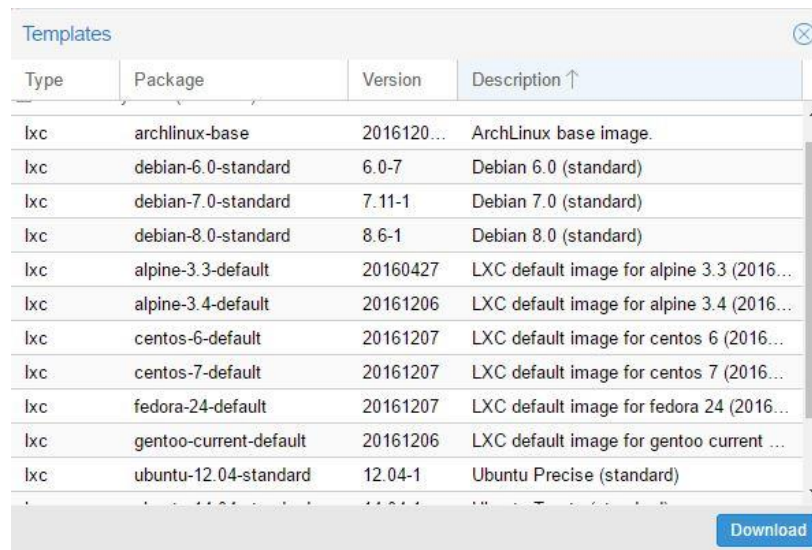
## 9.6 Creating VMs

After adding the shared storage, the next step is to create virtual machines. VMs are created and run on each node separately, but their images are stored on the NAS shared storage so, in case of failure they can be transferred to the other reliable node. Also, their backup copies are stored on NAS.



To create the VM, we make the following steps: firstly, we choose the node and click the “create VM” button. After that a menu with different features appears. We choose the OS that will be installed on the VM.

In addition, Proxmox VE has two ways to install operational system on the virtual machine: the first way is to have a ready .iso image on the local storage or CD/DVD drive. The second way is to download the template of VM or container from the list of templates in the Proxmox VE menu (Figure 22).



| Type | Package                | Version    | Description ↑                             |
|------|------------------------|------------|---|
| lxc  | archlinux-base         | 2016120... | ArchLinux base image.                     |
| lxc  | debian-6.0-standard    | 6.0-7      | Debian 6.0 (standard)                     |
| lxc  | debian-7.0-standard    | 7.11-1     | Debian 7.0 (standard)                     |
| lxc  | debian-8.0-standard    | 8.6-1      | Debian 8.0 (standard)                     |
| lxc  | alpine-3.3-default     | 20160427   | LXC default image for alpine 3.3 (2016... |
| lxc  | alpine-3.4-default     | 20161206   | LXC default image for alpine 3.4 (2016... |
| lxc  | centos-6-default       | 20161207   | LXC default image for centos 6 (2016...   |
| lxc  | centos-7-default       | 20161207   | LXC default image for centos 7 (2016...   |
| lxc  | fedora-24-default      | 20161207   | LXC default image for fedora 24 (2016...  |
| lxc  | gentoo-current-default | 20161206   | LXC default image for gentoo current ...  |
| lxc  | ubuntu-12.04-standard  | 12.04-1    | Ubuntu Precise (standard)                 |

[Download](#)

Figure 22. OS template menu

Then, after we have an .iso image, we choose it in the menu bar and continue with the following configurations. We choose the hard disk, CPU, memory and network configurations and confirm them. After that, the VM is created. Further, we run the VM and install the OS with the configurations needed. Now, the VM is created and running.

Firstly, we write a name of VM, ID, and choose the node on which the VM will be created. The whole process of creation is shown in Figures from 23 to 31.

Figure 23 shows the general configuration features of the VM.



Create: Virtual Machine

General OS CD/DVD Hard Disk

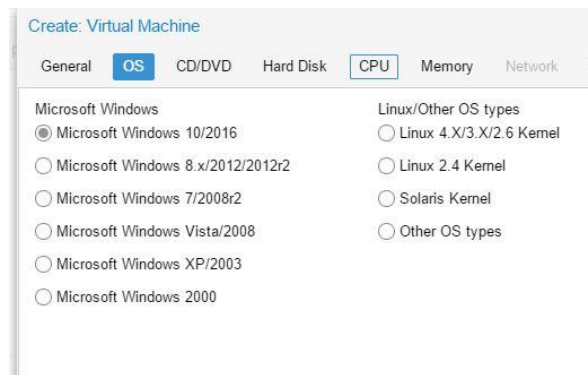
Node: pve01

VM ID: 100

Name: WinServ16

Figure 23. General features configuration

Then, we choose the OS type that will be installed on the VM. Figures 24 and 25 demonstrate examples of Microsoft Windows 10/2016 OS family, as we used Windows Server 2016, and Linux 4 X/3 X/2 6 kernel, as we used Linux Ubuntu Server 14.04.4.



Create: Virtual Machine

General OS CD/DVD Hard Disk CPU Memory Network

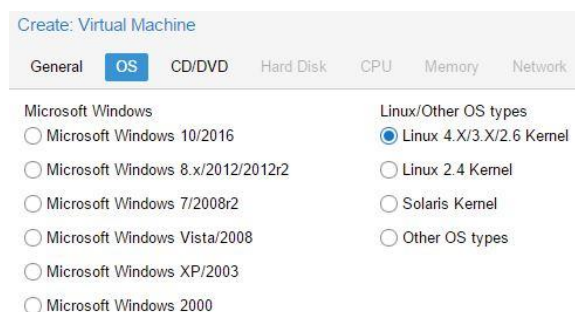
Microsoft Windows

- ☒ Microsoft Windows 10/2016
- ☐ Microsoft Windows 8.x/2012/2012r2
- ☐ Microsoft Windows 7/2008r2
- ☐ Microsoft Windows Vista/2008
- ☐ Microsoft Windows XP/2003
- ☐ Microsoft Windows 2000

Linux/Other OS types

- ☐ Linux 4.X/3.X/2.6 Kernel
- ☐ Linux 2.4 Kernel
- ☐ Solaris Kernel
- ☐ Other OS types

Figure 24. Choosing Windows 10/2016



Create: Virtual Machine

General OS CD/DVD Hard Disk CPU Memory Network

Microsoft Windows

- ☐ Microsoft Windows 10/2016
- ☐ Microsoft Windows 8.x/2012/2012r2
- ☐ Microsoft Windows 7/2008r2
- ☐ Microsoft Windows Vista/2008
- ☐ Microsoft Windows XP/2003
- ☐ Microsoft Windows 2000

Linux/Other OS types

- ☒ Linux 4.X/3.X/2.6 Kernel
- ☐ Linux 2.4 Kernel
- ☐ Solaris Kernel
- ☐ Other OS types

Figure 25. Choosing Linux 4 X/3 X/2 6 kernel

Now, we choose the place where OS is stored as shown in Figure 26.



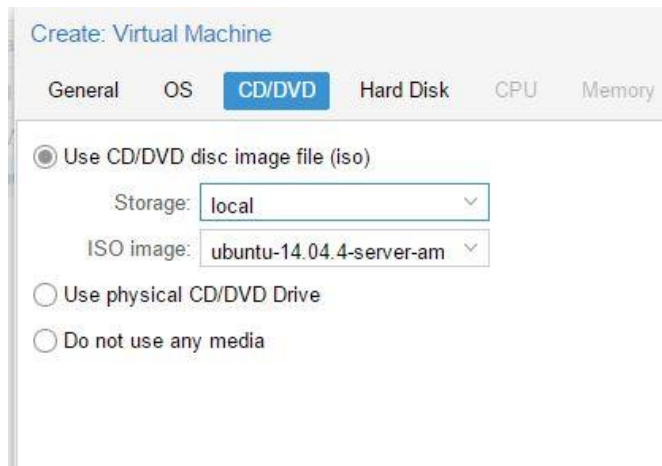


Figure 26. Source of OS image

Figure 27 demonstrates configurations for the hard disk.

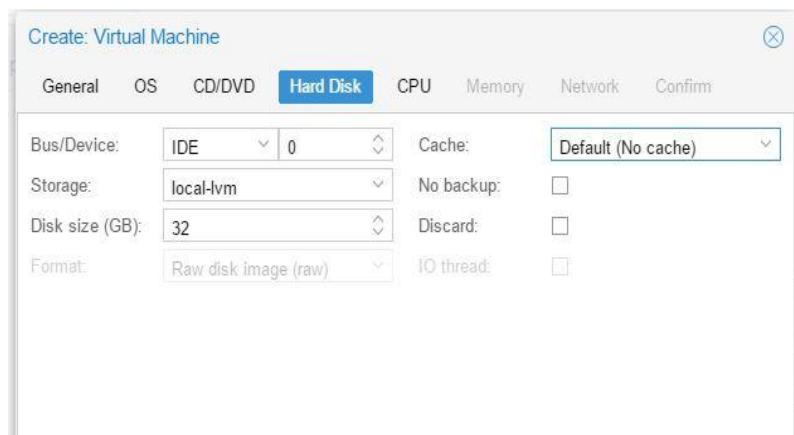


Figure 27. Hard disk configuration features

After that, CPU configurations should be chosen. Figure 28 shows the example of CPU configuration features.

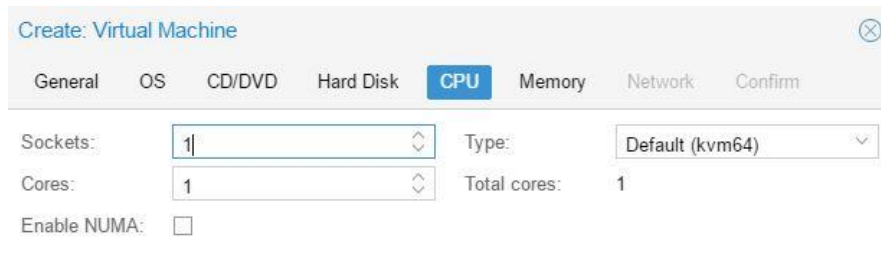


Figure 28. CPU configuration features

Figure 29 demonstrates memory configurations features.

Create: Virtual Machine

General OS CD/DVD Hard Disk CPU **Memory** Network

☒ Use fixed size memory

Memory (MB): 3072

Ballooning: ☒

☐ Automatically allocate memory within this range

Maximum memory (MB): 1024

Minimum memory (MB): 512

Shares: Default (1000)

Figure 29. Memory configuration features

And in the end, network configurations are shown in Figure 30.

Create: Virtual Machine

General OS CD/DVD Hard Disk CPU Memory **Network** Confirm

☒ Bridged mode

Model: Intel E1000

VLAN Tag: no VLAN

MAC address: auto

Bridge: vmbr0

Rate limit (MB/s): unlimited

Firewall: ☐

Multiqueues:

☐ NAT mode

Disconnect: ☐

☐ No network device

Figure 30. Network configuration features

Figure 31 demonstrates the whole list of configurations that were made and their confirmation.

Create: Virtual Machine

General OS CD/DVD Hard Disk CPU Memory Network **Confirm**

Settings

| Key ↑    | Value  |
|----------|--|
| ide0     | local-vm:32  |
| ide2     | local-iso/SW_DVD9_Win_Svr_STD_Core_and_DataCtr_Core_2016_... |
| memory   | 3072   |
| name     | WinServ16  |
| net0     | e1000,bridge=vmbr0   |
| nodename | pre01  |
| numa     | 0  |
| ostype   | win10  |
| sockets  | 1  |
| vmid     | 100  |

Back Finish

Figure 31. Confirmation of configuration

## 9.7 Configuring live migration

Intel Virtualization must be ON in BIOS; otherwise live migration will not work.

Firstly, we ping to VM to check, if it works and is available. Also, this is done during the whole process to make sure that VM is always reachable. In the other words this means that live migration really works and migration does not interrupt the work of VM.

Then, we choose the VM that will be migrated and click on it. After that, we push the “migrate” button and choose in the icon the target node where the VM will be placed after migration. Also, there is an “online” button which means that it is live migration. Therefore, regular migration can be done, but the VM’s work must be stopped. Figures from 32 to 34 show the process of live migration.

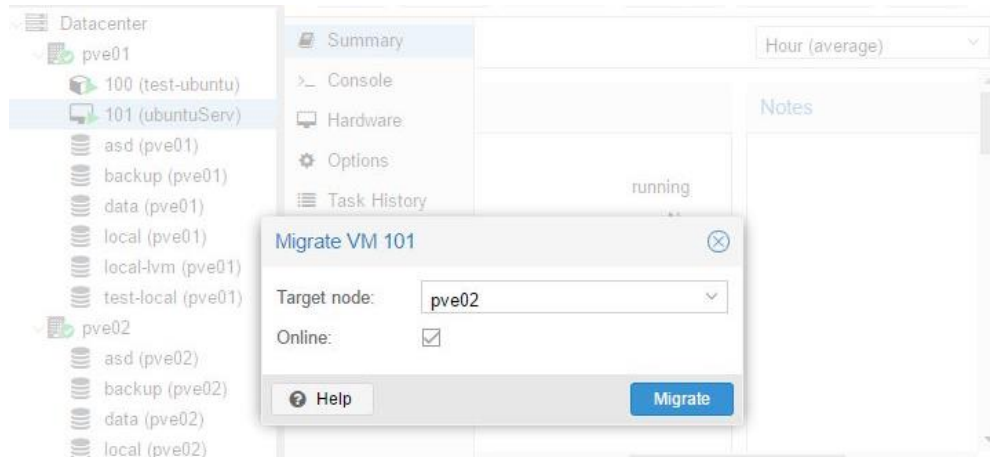
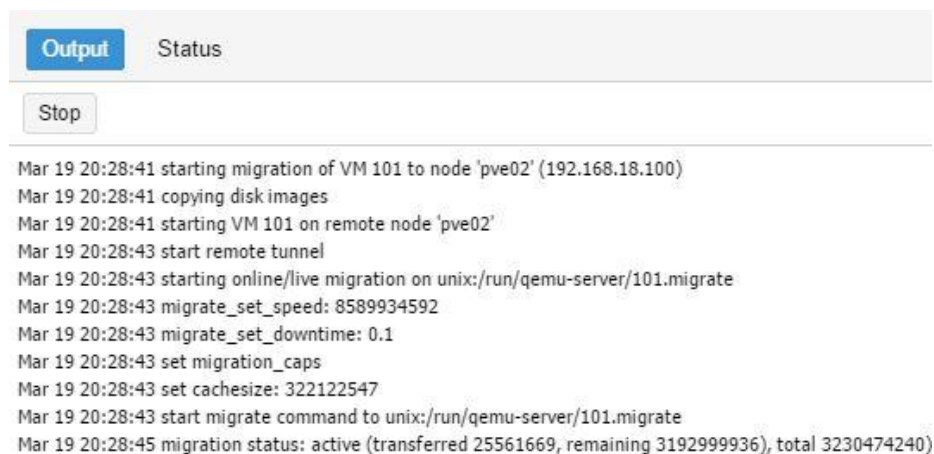


Figure 32. Choosing the target node



```

Mar 19 20:29:31 migration status: active (transferred 558198611, remaining 2629632), total 3230474240
Mar 19 20:29:31 migration xbzrlc cachesize: 268435456 transferred 0 pages 0 cachemiss 6844 overflow 0
Mar 19 20:29:31 migration speed: 64.00 MB/s - downtime 43 ms
Mar 19 20:29:31 migration status: completed
Mar 19 20:29:34 migration finished successfully (duration 00:00:53)
TASK OK

```

Figure 33. Live migration in process

And after the migration, the VM starts automatically on the target node (Figure 34).

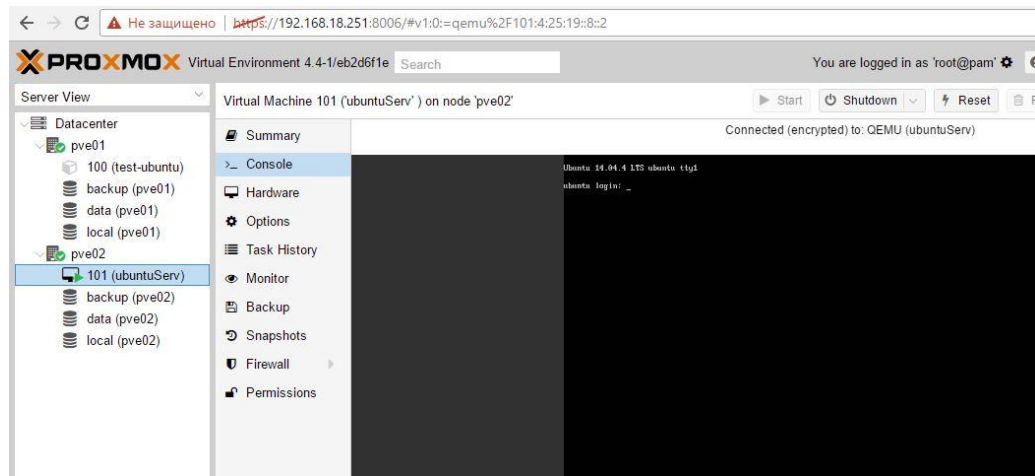


Figure 34. VM automatically starting on the other node

## 9.8 Configuring backup

In Proxmox VE backup can be scheduled with different features. We can specify the node which will be backed up, the day and time, etc. After the configuration, the backup will be done automatically with these configurations. In figure 35 shown scheduled backup configuration features.

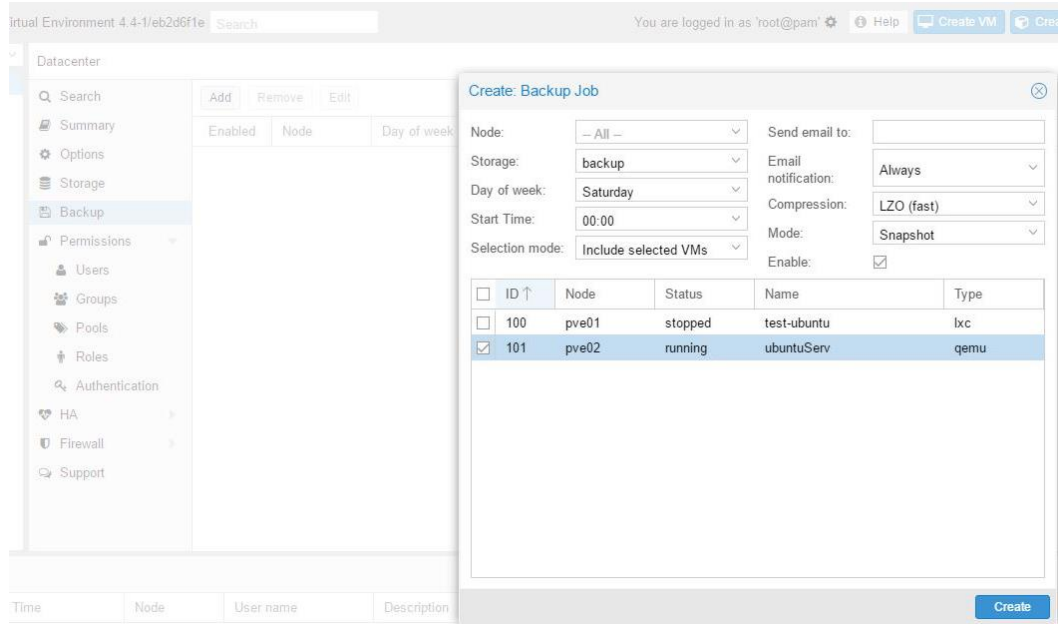


Figure 35. Scheduled backup configuration features

Also, backup can be forced, and there is the button “backup now” for carrying this out. It will immediately create backup of the chosen VM (Figure 37). There is an option where the backup will be stored, with options for mode and type of compression. Figure 36 illustrates forced backup configuration features.

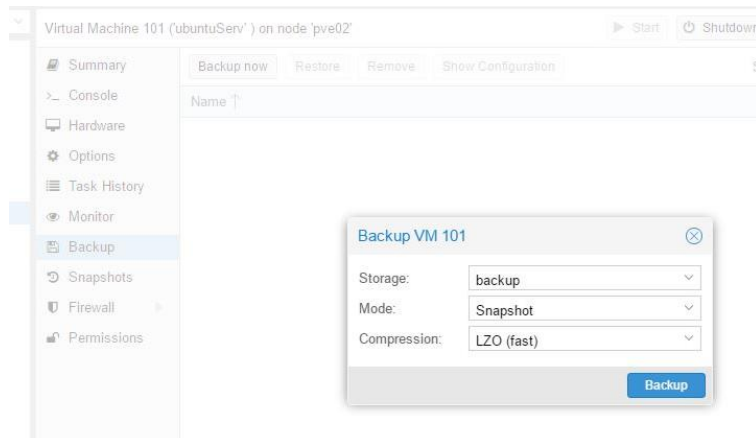


Figure 36. Forced backup configuration features

There are three modes of backup: stop, suspend and snapshot.

**Stop mode** provides the highest consistency of the backup and the highest downtime. It stops the VM and after backup is done, the VM is resumed. (Ahmed W. 2015.)

**Suspend mode** suspends the VM before calling the snapshot mode. It is recommended to use a snapshot mode instead of this one, because suspend mode does not necessarily improve the data consistency and provides higher downtime than the snapshot mode. (Ahmed W. 2015.)

**Snapshot mode** has small inconstancy risk, but it provides the lowest downtime. This mode provides live backup and data is copied while the VM is running. (Ahmed W. 2015.)

Also, there are compression options for backup. There are LZO and GZIP compression types. Also, none option can be used with no compression.

**LZO compression** balances between speed and compression ratio and it is a fast backup method.

**GZIP** compression does the higher possible compression but consumes more CPU and memory resources of the node. (Ahmed W. 2015.)

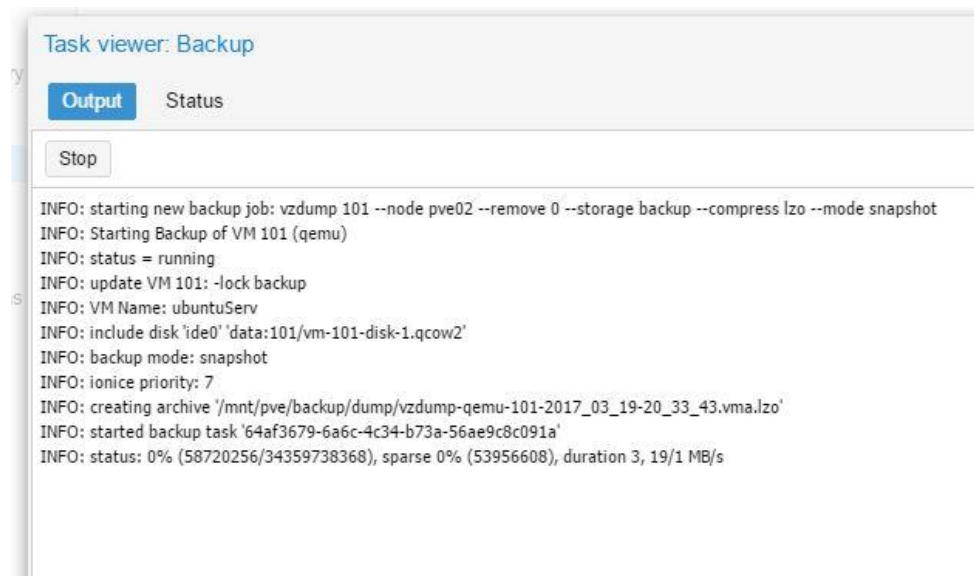


Figure 37. Backup in process

## 10 CONCLUSIONS

The goal of this thesis work was to create a high-availability cluster environment of two physical servers. The objectives were achieved. We created a highly available and reliable cluster system with supported live migration and easy backup system. Also, optimization of physical server resources was achieved.

To achieve the goal, we used two physical server nodes and connected them together into a cluster using the Proxmox VE server virtualization platform and attached NAS shared storage. We also created several virtual machines. Later, operational systems were installed on them and they acted as separate virtual servers. We implemented the migration of virtual machines and also made it possible to do live migration of the running virtual machines. And in the end, we configured backup/restore system.

Installation and configuration of Proxmox VE was complicated. Even if we choose the Proxmox which already comes with Debian, it is not easy to manage it. And of course, for the appropriate work of the system we should choose manual installation of Debian and only then install Proxmox. This method takes time to configure the system, but provides better resource allocations and manual configuration of different features. Most complicated part of this installation method is to properly install Debian and make basic configurations for Proxmox VE installation.

The next complicated part of the cluster configuration is to group nodes into the cluster and it takes some time. It should be done manually from the shell CLI with use of commands. And during the configuration process lots of errors may occur. So, it is needed to be accurate and patient with the configurations.

As the result, we have high-availability environment which provides optimization of resources, easy migration and copying of VMs, live migration inside the cluster and easy backup/restore system.

With high-availability cluster system, in case of failure, downtime of applications is decreased and sensitive applications can stay online maximized time. Clients and users of the company can get support 24/7 and there is no need to wait for application availability.



## REFERENCES

24HourData. 2012. What Are Nested RAID Levels and How Do They Work. WWW document. Updated 04 February 2012. Available at:

<http://www.24hourdata.com/blog/what-are-nested-raid-levels-and-how-do-they-work> [Accessed 25 December 2016]

Ahmed W. 2015. Proxmox Cookbook. 1st edition. Birmingham: Packet Publishing Ltd.

Balchunas A. 2014. Cisco CCNA study guide. Version 2.71.

Clark C., Fraser K., Hand S., Hansen J. G., Jul E., Limpach C., Pratt I., Warfield A. 2005. Pdf document. Live Migration of Virtual Machines. Available at:

<http://www.cl.cam.ac.uk/research/srg/netos/papers/2005-migration-nsdi-pre.pdf> [Accessed 16 February 2017]

Desmond M. 2003. WWW document. SAN/NAS convergence: proceed with caution. Updated September 2003. Available at:

<http://searchstorage.techtarget.com/magazineContent/SAN-NAS-convergence-proceed-with-caution> [Accessed 20 January 2017]

Fafarak S., Lola J., O'Brien D. 2002. WWW document. TruCluster Server Handbook. Chapter 17.3 Quorum and Voting. Updated July 2002. Available at:

<http://flylib.com/books/en/2.387.1.147/1/> [Accessed 8 April 2017]

Graphiq Inc. No date. WWW document. Compare Virtualization Software & Hypervisors. Available at: <http://virtualization.softwareinsider.com/> [Accessed 25 March 2017]

Haas F., Reisner P., Ellenberg L. 2011. WWW document. DRBD Users Guide 8.0-8.3. Updated 15 July 2011. Available at: <http://drbd.linbit.org/en/doc/users-guide-83/users-guide> [Accessed 03 February 2017]

Hines M. R., Deshpande U., Gopalan K. No date. Pdf document. Post-Copy Live Migration of Virtual Machines. Available at:

[https://kartikgopalan.github.io/publications/hines09postcopy\\_osr.pdf](https://kartikgopalan.github.io/publications/hines09postcopy_osr.pdf) [Accessed 17 February 2017]

Hwang K., Dongarra J., Fox G. 2011. WWW document. Distributed and cloud computing: keep virtual clusters manageable. Updated November 2011.

Available at: <http://searchtelecom.techtarget.com/feature/Distributed-and-Cloud-Computing-Keep-virtual-clusters-manageable> [Accessed 3 December 2016]

Hwang K., Dongarra J., Fox G. 2011. WWW document. Virtualization: Physical vs. Virtual Clusters. Updated 2011. Available at: <https://technet.microsoft.com/en-us/library/hh965746.aspx> [Accessed 3 December 2016]

Low S. 2010. WWW document. HA cluster configuration: Requirements and steps. Updated July 2010. Available at: <http://searchitchannel.techtarget.com/feature/HA-cluster-configuration-Requirements-and-steps> [Accessed 1 December 2016]

Microsoft. 2010. WWW document. Understanding Requirements for Failover Clusters. Updated July 8 2010. Available at: [https://technet.microsoft.com/en-us/library/cc771404\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc771404(v=ws.11).aspx) [Accessed 29 November 2016]

Microsoft. No date. WWW document. Understanding Backup and Recovery Basics for a Failover Cluster. Available at: [https://technet.microsoft.com/en-us/library/cc771973\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc771973(v=ws.11).aspx) [Accessed 16 December 2016]

Microsoft. No date. WWW document. Understanding Quorum Configurations in a Failover Cluster. Available at: [https://technet.microsoft.com/en-us/library/cc731739\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc731739(v=ws.11).aspx) [Accessed 8 April 2017]

Mushtaq N. U. 2016. WWW document. Network Attached Storage. Updated 23 October 2016. Available at: <http://cctvinstitute.co.uk/network-attached-storage/> [Accessed 20 January 2017]

Mushtaq N. U. 2016. WWW document. Storage Area Network. Updated 24 October 2016. Available at: <http://cctvinstitute.co.uk/storage-area-network/> [Accessed 20 January 2017]

Ortiz J. 2011. What Is RAID. WWW document. Updated 4 November 2011. Available at: [http://www.storage-switzerland.com/Articles/Entries/2011/11/4\\_What\\_Is\\_RAID\\_Part\\_1.html](http://www.storage-switzerland.com/Articles/Entries/2011/11/4_What_Is_RAID_Part_1.html) [Accessed 25 December 2016]

Ortiz J. 2012. What Are Nested RAID Levels. WWW document. Updated January 13 2012. Available at: [http://www.storage-switzerland.com/Articles/Entries/2012/1/13\\_What\\_Are\\_Nested\\_RAID\\_Levels\\_-\\_What\\_Is\\_RAID\\_Part\\_3.html](http://www.storage-switzerland.com/Articles/Entries/2012/1/13_What_Are_Nested_RAID_Levels_-_What_Is_RAID_Part_3.html) [Accessed 25 December 2016]

Posey B. 2010. WWW document. Data backup types explained: Full, incremental, differential and incremental-forever backup. Updated July 2010. Available at: <http://searchdatabackup.techtarget.com/tip/Data-backup-types-explained-Full-incremental-differential-and-incremental-forever-backup> [Accessed 17 January 2017]

Posey B. 2015. WWW document. How to configure a rack-mounted NAS in 10 easy steps. Updated May 2015. Available at: <http://searchstorage.techtarget.com/tip/How-to-configure-a-rack-mounted-NAS-in-10-easy-steps> [Accessed 13 April 2017]

Posey B. 2016. WWW document. Five types of storage virtualization: pros and cons. Updated January 2016. Available at: <http://searchstorage.techtarget.com/tip/Five-types-of-storage-virtualization-Pros-and-cons> [Accessed 18 April 2017]

Proxmox Server Solutions GmbH. 2016. WWW document. Backup and Restore. Updated 8 November 2016. Available at: [http://pve.proxmox.com/wiki/Backup\\_and\\_Restore](http://pve.proxmox.com/wiki/Backup_and_Restore) [Accessed 16 February]

Proxmox Server Solutions GmbH. 2016. WWW document. Install Proxmox VE on Debian Jessie. Updated 20 December 2016. Available at: [http://pve.proxmox.com/wiki/Install\\_Proxmox\\_VE\\_on\\_Debian\\_Jessie](http://pve.proxmox.com/wiki/Install_Proxmox_VE_on_Debian_Jessie) [Accessed 10 February]

Proxmox Server Solutions GmbH. 2016. WWW document. Installation. Updated 13 December 2016. Available at: <http://pve.proxmox.com/wiki/Installation> [Accessed 10 February 2017]

Proxmox Server Solutions GmbH. 2016. WWW document. Main Page. Updated 1 November 2016. Available at: [http://pve.proxmox.com/wiki/Main\\_Page](http://pve.proxmox.com/wiki/Main_Page) [Accessed 20 November 2016]

Proxmox Server Solutions GmbH. 2017. WWW document. DRBD. Updated 5 January 2017. Available at: <https://pve.proxmox.com/wiki/DRBD> [Accessed 03 February 2017]

Proxmox Server Solutions GmbH. No date. WWW document. About Proxmox. Available at: <https://www.proxmox.com/en/about> [Accessed 20 November 2016]

Proxmox Server Solutions GmbH. No date. WWW document. Features Proxmox VE. Available at: <https://www.proxmox.com/en/proxmox-ve/features> [Accessed 13 January 2017]

Proxmox Server Solutions GmbH. No date. WWW document. Proxmox Virtual Environment. Available at: <https://www.proxmox.com/en/proxmox-ve> [Accessed 12 December 2016]

Proxmox Server Solutions GmbH. No date. WWW document. Proxmox. Available at: <https://www.proxmox.com/en/> [Accessed 25 November 2016]

Proxmox Server Solutions GmbH. No date. WWW document. System Requirements. Available at: <https://www.proxmox.com/en/proxmox-ve/requirements> [Accessed 13 January 2017]

RedHat Inc. 2007. WWW document. Red Hat Cluster Suite for Red Hat Enterprise Chapter Linux 5. 1.3.3. Fencing. Updated 2007. Available at: [https://www.centos.org/docs/5/html/Cluster\\_Suite\\_Overview/s2-fencing-overview-CSO.html/](https://www.centos.org/docs/5/html/Cluster_Suite_Overview/s2-fencing-overview-CSO.html/) [Accessed 20 January 2017]

Rose M. 2006. WWW document. Live migration. Updated October 2006. Available at: <http://searchservirtualization.techtarget.com/definition/live-migration> [Accessed 15 February 2017]

Rose M. 2009. WWW document. Server virtualization. Updated June 2009. Available at: <http://searchservvirtualization.techtarget.com/definition/server-virtualization> [Accessed 22 January 2017]

Rose M. 2016. WWW document. Virtual machine (VM). Updated July 2016. Available at: <http://searchservvirtualization.techtarget.com/definition/virtual-machine> [Accessed 22 January 2017]

Rose M. 2016. WWW document. Virtualization. Updated October 2016. Available at: <http://searchservvirtualization.techtarget.com/definition/virtualization> [Accessed 22 January 2017]

Rouse M. 2013. Shared storage. WWW document. Updated April 2013. Available at: <http://whatis.techtarget.com/definition/shared-storage>. [Accessed 18 April 2017]

Scottish Qualification Authority. 2010. LO4: Implement Basic Networks and Security. WWW document. Available at <https://www.sqa.org.uk/e-learning/HardOSEss04CD/index.htm> [Accessed 15 January 2017]

SNIA Technical Position. 2009. Common RAID Disk Data Format Specification. Version 2.0, Revision 19.

SNIA. 2016. SNIA Dictionary. WWW document. Updated April 2016. Available at: <https://www.snia.org/education/dictionary/r#raid> [Accessed 25 December 2016]

Tanenbaum A. S., Wetherall D. J. 2011. Computer networks. 5<sup>th</sup> edition. Boston: Pearson Education Inc.

Technopedia Inc. No date. WWW document. Computer cluster. Available at: <https://www.techopedia.com/definition/6581/computer-cluster> [Accessed 1 December 2016]

TechTarget. 2008. WWW document. A guide to storage virtualization. Updated June 2008. Available at: <http://searchstorage.techtarget.com/feature/A-guide-to-storage-virtualization> [Accessed 18 April 2017]

TGRMN Software. No date. WWW document. Real-Time Backup. Available at: <http://www.tgrmn.com/web/kb/item108.htm> [Accessed 19 January 2017]

**Feature comparison table of different server virtualization platforms**

| <b>Name of the platform</b>      | <b>Intended user</b>                                       | <b>Hypervisor type</b> | <b>Virtualization type</b>                              | <b>Architecture</b> | <b>Supported storage</b>                                       | <b>Platform</b>         | <b>License</b>     |
|----------------------------------|--|------------------------|---|---------------------|--|-------------------------|--------------------|
| Citrix XEN Server                | Personal<br><br>Enterprise<br><br>Small-medium<br>Business | Bare Metal (Type 1)    | Hardware Assisted<br><br>OS<br><br>Paravirtualization   | x86<br>x64          | DAS<br>FC<br>iSCSI<br>NAS<br>NFS<br>SAS<br>SATA<br>SCSI<br>USB | Windows<br>Linux        | Proprietary        |
| Microsoft Hyper-V Server 2008 R2 | Small-medium<br>Business                                   | Bare Metal (Type 1)    | Full<br><br>Hardware Assisted<br><br>OS                 | x86<br>x64          | DAS<br>FC<br>iSCSI<br>SAS<br>SATA                              | Windows<br>Online       | Proprietary        |
| VMware vSphere ESXi Hypervisor   | Small-medium<br>Business                                   | Bare Metal (Type 1)    | Full<br><br>Hardware Assisted<br><br>Paravirtualization | x86<br>x64          | DAS<br>FC<br>FCoE<br>iSCSI<br>NAS<br>SSD for Swap<br>USB       | Windows<br>Linux        | Proprietary        |
| Oracle Virtual Box               | Enterprise<br><br>Small-Medium<br>Business                 | Hosted (Type 2)        | Hardware Assisted<br><br>Paravirtualization             | x86<br>x64          | iSCSI<br>PATA<br>SAS<br>SATA<br>SCSI<br>USB                    | Windows<br>Mac<br>Linux | Open Source (Free) |

|                           |   |                     |   |                     |  |                  |                    |
|---------------------------|---|---------------------|---|---------------------|--|------------------|--------------------|
| IBM Power VM              | Enterprise<br><br>Small-Medium Business | Bare Metal (Type 1) | Full  | Power               | FC<br>Firewire<br>iSCSI<br>NFS<br>PATA<br>SATA<br>SCSI<br>SSD for Swap | Linux            | Proprietary        |
| iCore Virtual accounts    | Small-medium Business                   | Hosted (Type 2)     | OS  | x86                 | FC<br>Firewire<br>NAS<br>PATA<br>SATA<br>SCSI<br>USB                   | Windows          | Proprietary        |
| Open VZ Linux Containers  | Personal<br><br>Small-Medium Business   | Hosted (Type 2)     | OS  | x86<br>x64          | DAS  | Linux            | Open Source (Free) |
| Wind River Linux          | Small-Medium Business                   | Bare Metal (Type 1) | Full<br><br>Hardware Assisted<br><br>Paravirtualization | x86<br>x64<br>Power | USB  | Linux            | Proprietary        |
| Oracle Solaris Containers | Small-Medium Business                   | Hosted (Type 2)     | OS  | x86<br>SPARC        | FC<br>iSCSI<br>NAS   | Linux            | Proprietary        |
| Odin Virtuozzo            | Small-Medium Business                   | Hosted (Type 2)     | Hardware Assisted<br><br>OS                             | x86<br>x64          | iSCSI<br>NFS   | Windows<br>Linux | Proprietary        |

|   |   |                     |   |                              |   |                  |                    |
|---|---|---------------------|---|------------------------------|---|------------------|--------------------|
| KVM                                     | Personal<br><br>Small-Medium Business   | Hosted (Type 2)     | Full<br><br>Hardware Assisted<br><br>Paravirtualization           | x86<br>x64<br>Power          | USB   | Linux            | Open Source (Free) |
| Microsoft Windows Virtual PC            | Small-Medium Business                   | Bare Metal (Type 1) | Full<br><br>Hardware Assisted                                     | x86<br>x64                   | -   | Windows          | Proprietary        |
| Proxmox VE                              | Personal<br><br>Small-Medium Business   | Bare Metal (Type 1) | Full<br><br>OS  | x86<br>x64                   | FC<br>iSCSI<br>NFS                          | Windows<br>Linux | Open Source (Free) |
| Parallels Server Bare Metal             | Enterprise<br><br>Small-Medium Business | Bare Metal (Type 1) | Full<br><br>Hardware Assisted<br><br>OS                           | x64<br>Power                 | eSATA<br>NFS<br>SATA<br>SSD for Swap<br>USB | Linux<br>Online  | Proprietary        |
| LynxSecure Separation Kernel Hypervisor | Enterprise<br><br>Small-Medium Business | Bare Metal (Type 1) | Full<br><br>Hardware Assisted<br><br>OS<br><br>Paravirtualization | x86<br>x64<br>Power<br>SPARC | NFS<br>USB                                  | Windows<br>Linux | Proprietary        |
| Nuxis                                   | Enterprise<br><br>Small-Medium Business | Bare Metal (Type 1) | Full<br><br>Hardware Assisted<br><br>Paravirtualization           | x86<br>x64                   | FC<br>iSCSI<br>NFS<br>SATA<br>SCSI          | Linux            | Open Source (Free) |

| <b>Name of the platform</b>      | <b>Management features</b>   | <b>Supported host operating systems</b>   | <b>Supported guest operating systems</b>  |
|----------------------------------|--|---|---|
| Citrix XEN Server                | Change Reports<br>Dynamic Resource<br>High Availability<br>Live Migration<br>Multiple Host Resource Pools<br>Performance Metrics<br>Performance Reports<br>Power Management<br>Real Time Alerts<br>Storage Migration<br>VM Migration | Mandrake Linux<br>Red Hat Enterprise Linux<br>SUSE Linux Enterprise Server<br>Turbolinux Enterprise Server<br>Windows 2000 Server<br>Windows NT Server<br>Windows NT Terminal Server Edition<br>Windows Server 2003 | -   |
| Microsoft Hyper-V Server 2008 R2 | Capacity<br>Planning/Management<br>Change Reports<br>Configuration Snapshots<br>Dynamic Resource<br>High Availability<br>Live Migration<br>Performance Reports<br>Shared Resource Pools<br>Storage Migration<br>VM Migration         | Windows Server 2008 R2  | CentOS<br>Fedora<br>Free BSD<br>Mandrake Linux<br>Novell<br>Red Hat Linux<br>Small Business Server 2003<br>Solaris x86 Platform Edition<br>SUSE Linux<br>Turbolinux<br>Windows              |
| VMware vSphere ESXi Hypervisor   | Configuration Mapping<br>Dynamic Resource<br>Failover<br>Live Migration<br>Thin Provisioning<br>Virtual Firewall   | Free BSD<br>Windows Server 2008 R2  | Free BSD<br>Mac OS X<br>Mandrake Linux<br>MS DOS<br>Novell Linux Desktop<br>Red Hat Linux<br>Solaris x86 Platform Edition<br>Sun Java Desktop System<br>SUSE Linux<br>Turbolinux<br>Windows |



|                                |   |   |  |
|--------------------------------|---|---|--|
| Oracle<br>Virtual Box          | Asset Management<br>Automated Workflows<br>Configuration Mapping<br>Failover<br>High Availability<br>Live Migration<br>Multiple Host Resource Pools<br>P2V Conversion<br>Shared Resource Pools<br>VM Cloning  | Fedora<br>Mac OS X<br>Oracle Solaris<br>Red Hat Linux<br>Solaris x86 Platform Edition<br>SUSE Linux Enterprise Server<br>Ubuntu<br>Windows Server 2008 R2 | CentOS<br>Fedora<br>Free BSD<br>Mandrake Linux<br>MS DOS<br>Novell Linux Desktop<br>Oracle Solaris<br>Red Hat Linux<br>SUSE Linux Enterprise Server<br>Ubuntu<br>Windows |
| IBM Power<br>VM                | Asset Management<br>Automated Workflows<br>Dynamic Resource<br>High Availability<br>Live Migration<br>Multiple Host Resource Pools<br>Power Management<br>Real Time Alerts<br>Shared Resource Pools<br>Storage Migration<br>Thin Provisioning<br>VM Migration | -   | AIX<br>CentOS<br>Fedora<br>Mandrake Linux<br>Novell<br>Red Hat Linux<br>SUSE Linux<br>Ubuntu   |
| iCore Virtual<br>accounts      | Maintenance Mode<br>Shared Resource Pools   | Windows XP Professional   | Windows XP Home Edition<br>Windows XP Professional   |
| Open VZ<br>Linux<br>Containers | Automated Workflows<br>Dynamic Resource<br>Failover<br>High Availability<br>Live Migration<br>Performance Reports<br>Shared Resource Pools<br>Thin Provisioning<br>Virtual Firewall<br>VM Backup/Restore<br>VM Cloning and migration                          | CentOS<br>Fedora<br>Free BSD<br>Mandrake Linux<br>Novell Linux Desktop<br>Red Hat Linux<br>SUSE Linux Enterprise Server<br>Ubuntu                         | CentOS<br>Fedora<br>Free BSD<br>Mandrake Linux<br>Novell Linux Desktop<br>Red Hat Linux<br>SUSE Linux<br>Ubuntu  |

|                                    |   |  |   |
|------------------------------------|---|--|---|
| Wind River<br>Linux                | Power Management<br>VM Migration  | Fedora<br>openSUSE<br>Red Hat Enterprise Linux<br>SUSE Linux<br>Ubuntu   | Red Hat Linux<br>SUSE Linux<br>Windows  |
| Oracle<br>Solaris<br>Containers    | Configuration Snapshots<br>Live Migration<br>P2V Conversion   | Red Hat Enterprise Linux   | Novell Linux Desktop<br>Red Hat Linux   |
| Odin<br>Virtuozzo                  | Dynamic Resource<br>High Availability<br>Live Migration<br>Storage Migration  | CentOS<br>Fedora<br>Free BSD<br>Mandrake Linux<br>Red Hat Linux<br>SUSE Linux Enterprise Server<br>Ubuntu<br>Windows Server 2003<br>Windows Server 2008 R2 | CentOS<br>Fedora<br>Free BSD<br>openSUSE<br>Red Hat Linux<br>SUSE Linux<br>Ubuntu<br>Windows  |
| KVM                                | Asset Management<br>Configuration Snapshots<br>Live Migration<br>Performance Metrics<br>Storage Migration<br>VM Migration | CentOS<br>Fedora<br>Free BSD<br>Gentoo<br>Mandrake Linux<br>OpenBSD<br>openSUSE<br>Red Hat Linux<br>Slackware<br>Ubuntu                                    | CentOS<br>Fedora<br>Free BSD<br>MS DOS<br>Novell Linux Desktop<br>OpenBSD<br>Oracle Solaris<br>Red Hat Linux<br>Solaris x86 Platform Edition<br>SUSE Linux<br>Ubuntu<br>Windows |
| Microsoft<br>Windows<br>Virtual PC | Change Reports  | Windows 7  | Windows   |

|  |  |   |  |
|--|--|---|--|
| Proxmox VE                                       | Capacity<br>Planning/Management<br>High Availability<br>Live Migration<br>Storage Migration<br>Virtual Firewall<br>VM Backup/Restore<br>VM Cloning   | Debian Sarge  | CentOS<br>Debian Sarge<br>Fedora<br>Free BSD<br>Gentoo<br>Mandrake Linux<br>MS DOS<br>Red Hat Linux<br>Small Business Server 2003<br>SUSE Linux<br>Turbolinux<br>Ubuntu<br>Windows       |
| Parallels<br>Server Bare<br>Metal                | Automated Workflows<br>Live Migration<br>Storage Migration<br>VM Migration   | -   | CentOS<br>Debian Sarge<br>Fedora<br>Free BSD<br>openSUSE<br>Oracle Solaris<br>Red Hat Linux<br>Solaris x86 Platform Edition<br>SUSE Linux<br>Turbolinux Workstation<br>Ubuntu<br>Windows |
| LynxSecure<br>Separation<br>Kernel<br>Hypervisor | Failover<br>High Availability<br>Performance Metrics<br>VM Backup/Restore  | CentOS<br>Red Hat Enterprise Linux<br>Red Hat Linux | CentOS<br>Mac OS X<br>Red Hat Linux  |
| Nuxis  | Failover<br>High Availability<br>Live Migration<br>Maintenance Mode<br>Multiple Host Resource Pools<br>Performance Metrics<br>Thin Provisioning<br>VM Backup/Restore<br>VM Cloning and Migration | -   | -  |