

Samu Leppänen

CYBERGAME VIRTUALIZATION AND STORAGE CLUSTER DESIGN

Bachelor's Thesis
Information Technology

2017



Kaakkois-Suomen
ammattikorkeakoulu

Tekijä/Tekijät	Tutkinto	Aika
Samu Leppänen	insinööri (AMK)	Toukokuu 2017
Opinnäytetyön nimi Cybergame virtualization and storage cluster design		56 sivua
Toimeksiantaja Kymenlaakson ammattikorkeakoulu		
Ohjaaja Lehtori Vesa Kankare		
Tiivistelmä <p>Tämän opinnäytetyön ensisijaisena tavoitteena oli suunnitella ja toteuttaa virtualisointiympäristö ”Kyberturvallisuusosaamisen ja liiketoiminnan kehittäminen” -projektille käyttäen VMware vSphere -tuotteita. Ympäristöllä tuli olla korkea saatavuusaste ja viansietokyky. Tietoturvan lisäämiseksi tuli olla mahdollisuus rajata toisen kerroksen verkkoliikennöintiä. Toteutettua ympäristöä tulotaisiin käyttämään osana kyberturvallisuuspeliprojektin infrastruktuuria.</p> <p>Peliprojekti nojaa virtualisointiin simuloidakseen realistisia kyberturvallisuuden harjoituskeinoja. Tämän vuoksi syntyi tarve virtualisointialustalle, joka tyydyttäisi peliprojektin tarpeet. Tässä opinnäytetyössä keskitytään useiden VMware-ohjelmistojen toteutuksiin virtuaalialustan toteuttamisessa.</p> <p>Tämän työn toteutus rakennettiin kokonaan uudelleen edellisen virtualisointiympäristön päälle. Virtuaalikone-monitorit, keskitetty hallinta, virtualisoitu verkko ja virtualisoitu muisti-alue toteutettiin ympäristössä haluttujen tavoitteiden saavuttamiseksi. Ohjelmistototeutuksen jälkeen aiheutettiin useita simuloituja vikatilanteita korkean saatavuuden testaamiseksi.</p> <p>Työtä varten annettuihin tavoitteisiin päästiin onnistuneesti. Virtualisointiympäristö on tuotantokäytössä kaikki halutut ominaisuudet käyttöönotettuina. Verrattuna edelliseen ympäristöön uusi toteutus tarjoaa enemmän hyödyllisiä ominaisuuksia kuten automatisoidut vikasietoisuuskäytännöt ja korkea saatavuusaste. Toteutuksen aikana kohdattiin joitain ei-kriittisiä ongelmia. Suurin osa niistä ratkaistiin ja jäljelle jääneistä ongelmista voidaan jalostaa tulevaisuuden projekti-ideoita opiskelijoille.</p>		
Asiasanat datakeskus, kyberturvallisuus, virtualisointi, VMware		

Author (authors)	Degree	Time
Samu Leppänen	Bachelor of Engineering	May 2017
Thesis Title		56 pages
Cybergame virtualization and storage cluster design		
Commissioned by		
Kymenlaakso University of Applied Sciences		
Supervisor		
Vesa Kankare, Senior Lecturer		
Abstract		
<p>The main goal of this bachelor's thesis was to design and implement a virtualization environment for the "Cybersecurity Expertise and Business Development" project using VMware vSphere products. The environment had to be fault tolerant and have high availability. For added security, layer 2 network traffic restrictions needed to be implemented. The implemented environment would be used as a part of the infrastructure of the Cybergame project.</p> <p>Cybergame relies on virtualization for realistic simulation of cybersecurity training scenarios. As such, there was a need for a virtualization platform which would meet the needs of the game. The thesis focuses on multiple VMware software implementations for implementing the platform.</p> <p>The implementation of this thesis work was carried out by completely rebuilding over the previous virtualization environment. Hypervisors, centralized management, virtualized networking and virtualized storage area were implemented to achieve the desired goals. After implementing the software, multiple failure situations were simulated to test high availability.</p> <p>The given goals for the thesis work were achieved successfully. The virtualization environment was implemented with all desired features and is in production use. Comparing with the previous environment, the new implementation offers more beneficial features such as automated failovers and high availability. During the implementation, some non-critical problems were encountered. Most of them were resolved and the remaining problems could be carried out as future project ideas for students.</p>		
Keywords		
cybersecurity, data center, virtualization, VMware		

TABLE OF CONTENTS

1	INTRODUCTION	7
2	CYBERGAME	8
2.1	Network	8
2.1.1	Design	9
2.1.2	Rules	10
2.1.3	Addressing & VLANs	11
3	VIRTUALIZATION	13
3.1	Virtual machines	13
3.2	Hypervisors	13
4	VMWARE	14
4.1	vSphere	14
4.1.1	ESXi	15
4.1.2	vCenter	16
4.1.3	Distributed Switch	16
4.1.4	Cluster features	18
4.2	vSAN	19
5	PRIVATE VLANS	20
6	IMPLEMENTATION	21
6.1	Hardware	21
6.2	Hypervisor	22
6.2.1	Installation	22
6.2.2	Post-installation tasks	23
6.3	vCenter	24
6.3.1	Deployment	25
6.3.2	Updating the vCenter appliance	26

6.3.3	Authentication from AD	28
6.3.4	Cluster configuration	29
6.3.5	Licensing	30
6.3.6	Updating the hosts	31
6.4	Networking	34
6.4.1	Implementing vDS	34
6.4.2	Implementing vDS port groups	35
6.4.3	Private VLANs	38
6.5	vSphere Virtual SAN	40
6.5.1	Prerequisites	40
6.5.2	Configuration	41
6.5.3	Testing & monitoring	44
6.6	Additional cluster configuration	46
6.7	Failover test	48
7	CONCLUSIONS	50
	REFERENCES	52
	LIST OF FIGURES	

ABBREVIATIONS

AD	Active Directory
CPU	Central Processing Unit
DMZ	Demilitarized Zone
DRS	Distributed Resource Scheduler
EVC	Enhanced vMotion Compatibility
HA	High Availability
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IOPS	Input / Output Operations Per Second
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
NIC	Network Interface Card
NTP	Network Time Protocol
PVLAN	Private Virtual Local Area Network
RODC	Read-Only Domain Controller
SSD	Solid-State Drive
SQL	Structured Query Language
TCP	Transmission Control Protocol
vDS	vSphere Distributed Switch
VLAN	Virtual Local Area Network
VM	Virtual Machine
vSAN	Virtual Storage Area Network
VUM	vSphere Update Manager

1 INTRODUCTION

There is an ongoing game application project at South-Eastern Finland University of Applied Sciences referred to as Cybergame. In short, the purpose of the game is to educate players on cybersecurity through simulated training scenarios. The scenarios aim to imitate the realistic behavior of the computers and network devices in the scenario. To achieve this, the game relies on virtualization to create exercises which could be useful in real life situations. (Kaakkois-Suomen Ammatikorkeakoulu Oy s.a.; Työ- ja elinkeinoministeriö s.a.)

For the Cybergame project to operate as envisioned there was a need for a virtualization platform which would meet the needs of the project. Some groundwork, such as design and acquisition of hardware aspects for the platform was done previously in a bachelor's thesis work by Suvi Lumivirta in the spring of 2015 (Lumivirta 2015). However, the previous platform was deemed obsolete and needed to be redesigned and reimplemented to match the current needs of the project.

The goal of this bachelor's thesis was to design and implement a production ready virtualization environment with the VMware vSphere products ESXi, vCenter and vSAN. The environment had to be very tolerant to faults such as device- or network failures and have high availability. For added security, there was also a need to implement network traffic restrictions for layer 2 connectivity. The implemented environment would be used as a part of the infrastructure of the Cybergame project.

The implementation of this thesis was carried out by first purging the old environment and starting anew. Hypervisors, centralized management, virtualized networking and virtualized storage pools were implemented to achieve the desired goals. After implementation, simulated cases of failures were conducted to test the implementation.

2 CYBERGAME

South-Eastern Finland University of Applied Sciences, Lappeenranta University of Technology and Cursor Oy have, at the time of writing, an ongoing European Union backed project called “Cybersecurity Expertise and Business Development”. The goal of the project is to create a platform which would utilize the available knowledge of data centers, gamification and cybersecurity for research and innovation. The platform would create new cybersecurity-related information and technologies, such as educational applications. In addition, the platform could potentially create job opportunities. (Kaakkois-Suomen Ammattikorkeakoulu Oy s.a.; Työ- ja elinkeinoministeriö s.a.)

The main project has a parallel project called Cybergame, which is the project this bachelor’s thesis is associated with. Cybergame is a game application concept, which aims to attain information and develop strategies on gamification and cybersecurity-related issues. Other topics like monetization and revenue models are also studied. (Kaakkois-Suomen Ammattikorkeakoulu Oy s.a.; Työ- ja elinkeinoministeriö s.a.)

The game prototype itself aims to educate the players in different areas of cybersecurity through scenarios. The scenarios are cybersecurity oriented exercises, which are supposed to imitate real life IT environments. In the scenarios, the player’s goal is to complete tasks according to instructions and progress in the game by doing as such. The scenarios, their required devices and networks are contained in virtualized environments. The scenarios are virtualized in the university’s data center using the virtualization environment implemented in this bachelor’s thesis.

2.1 Network

As the Cybergame network is a cybersecurity oriented environment, multiple network aspects need to be taken into consideration to design a network as securely as possible. For example, the network needs to have layered protection and be properly separated from the IT department’s own network despite running in the

same data center. It is likely that the network will be breached by attackers at some level in the future, so the design takes that into consideration as well.

2.1.1 Design

The Cybergame network is designed to implement multiple security zones. Splitting parts of the network into zones allows for precise control of traffic and potential isolation of attackers. The zones are separated by functions and breach severity levels so traffic between zones can be policed as desired. Zone separation is done by Cisco ASA -series firewalls in the Cybergame network. The severity levels are specified so the least critical zones are closest to the untrusted network (the Internet). Refer to figure 1 for a visualization of the security zones.

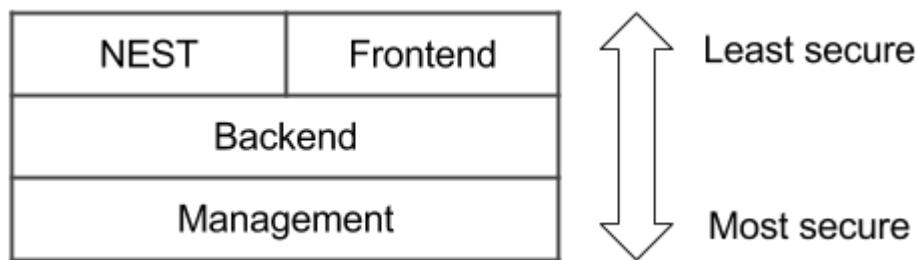


Figure 1. A visualization of the security levels of different security zones

Cybergame network mostly consists of four functional zones: NEST, Frontend, Backend and Management. All the zones have their own subnets and VLANs, which are listed in chapter 2.1.3. NEST subnet contains the virtualized environments of the game, i.e. virtual machines with the game scenario and its related virtual components. Frontend has the game website information and related functions such as the player's user interface. Backend stores the game logic and databases, which contain data such as the player's progression and scoring. Management subnet has all the management-related connections, such as monitoring functions and virtualization hypervisor management.

The physical network of Cybergame is designed in a way that allows for most topology changes to be configured without making any physical changes. This is achieved by utilizing virtualization in networking and software. For example, all the uplinks connected to the virtualization hypervisor servers utilize trunking links,

which allow all required VLANs to pass through the links to the VMs. The routing and traffic policing inside the network is handled by the firewalls. Routing to the Internet is handled by the application delivery controllers as required. An untrusted and strictly policed connection is also available from the management network to the IT department's network for management purposes. Refer to figure 2 for the physical topology.

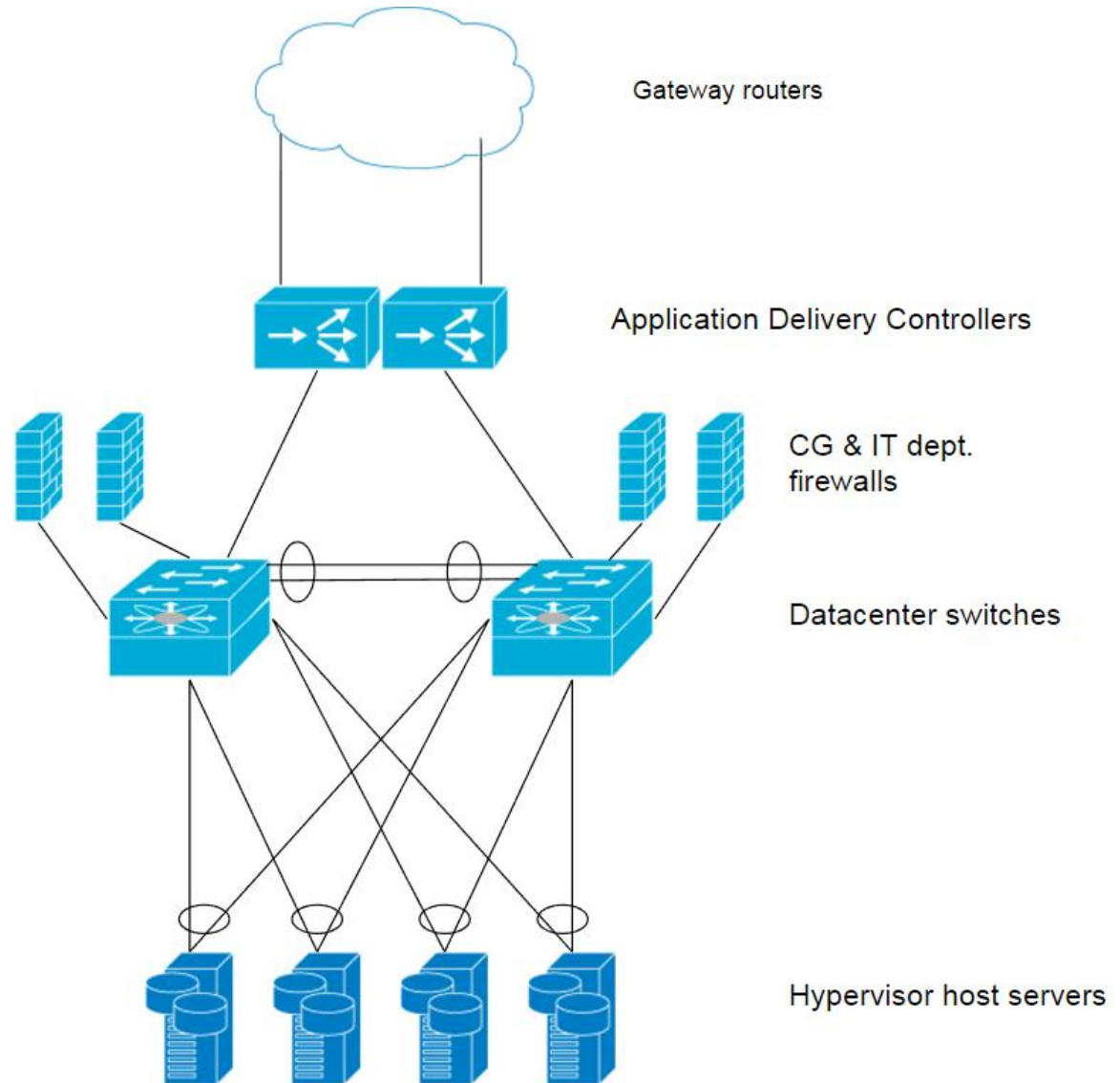


Figure 2. The physical topology

2.1.2 Rules

There is a plethora of different rules in place in the Cybergame network. The rules exist to create a network which is as secure as possible and to keep away

unwanted clutter or notable differences in device configurations. The basic traffic rule on the network is to restrict all traffic that is not specifically needed. Any needed traffic, e.g. for applications, needs to be separately allowed in the firewalls' policy configuration.

Especially any connectivity to the IT department's network is policed strictly. Users in the management subnet are authenticated with read-only domain controllers residing in the IT department's DMZ subnet. Using RODCs allow for secure authentication from the IT department's active directory authentication servers without allowing other traffic. The RODCs can also be used as nameservers in the management network if needed.

All devices in the network use synchronized date and time. The time zone of all devices is set to UTC to prevent differentiating logging times and daylight saving time -related problems. Synchronization is done using the NTP protocol, using Funet's NTP servers as source. The network also has a syslog logging server for monitoring events or alarms from multiple devices. Having the correct time on devices is critical especially because of logging events to syslog in the correct manner. It is important to have no discrepancy of timestamps in the logs to match occurred events to timestamps. Reliable timestamps can help with forensics by confirming the trail of events in case of breach or similar incidents.

To avoid unnecessary clutter, the rules contain a clear naming and numbering scheme for parts of the network. For naming devices, all devices are to have "CG- " prefix. In addition, the name should not include the manufacturer or model of the device. Numbering is related to network addressing and VLANs, listed in more detail in chapter 2.1.3.

2.1.3 Addressing & VLANs

The numbering scheme in the Cybergame network aims to be logically simple to avoid clutter. All the subnets in the network utilize the private address spaces of 172.16.*.* or 172.18.*.*. Each network function is assigned a subnet, a primary VLAN entry and an isolated PVLAN entry (if required). All primary VLAN entries

are prefixed with “21” and all the isolated PVLAN entries are prefixed with “22”. The subnet for vSAN functionality is unroutable and is only used for communication between the hypervisor hosts for internal traffic. Refer to figure 3 for all relevant functions and their network details.

Function	Primary VLAN	Isolated PVLAN	Subnet
NEST	2160	2260	172.16.0.0/16
Frontend	2180	2280	172.18.0.0/24
Backend	2181	2281	172.18.1.0/24
Management	2182	2282	172.18.2.0/24
vSAN	2183	-	172.18.3.0/24

Figure 3. Details of the network functions VLANs and subnets

Prior to implementation, the components related to virtualization were given IP addresses from the management subnet, apart from the update manager and vSAN addresses. The update manager resides in the IT department’s DMZ network, so the address given is not from the Cybergame network. The hypervisor hosts (CG-ESXn) are managed through the management network addresses, but vSAN requires another address for vSAN networking. The additional addresses for vSAN are the same apart from the subnet as seen in figure 4. vSAN addresses reside in the vSAN subnet (see figure 3), i.e. the address scheme is 172.18.3.1n where n is the host number. Refer to figure 4 for the addressing.

Component	IP address
vCenter	172.18.2.10
CG-ESX1	172.18.2.11
CG-ESX2	172.18.2.12
CG-ESX3	172.18.2.13
CG-ESX4	172.18.2.14
Update Manager	(Redacted, DMZ subnet)

Figure 4. IP addressing of the virtualization-related components

3 VIRTUALIZATION

In general, virtualization in IT refers to the conversion of a physical component to its logical equivalent. Virtual components offer a plethora of benefits over physical objects, such as performance and availability increases, reduced operation costs or simpler management of components. Virtualization is available as an option for many components like whole computers or core networking. (Portnoy 2012, 2, 10.)

3.1 Virtual machines

Virtualized computer hardware is called, for short, a virtual machine (VM). VMs are virtual computers, which use a physical host computer's, e.g. a server's resources for computing. A VM utilizes the physical resources via a hypervisor (see chapter 3.2). VMs, like real computers, have a processor, memory, networking and other core components required for operation. Hypervisor oversees sharing the resources if there are multiple VMs under a certain hypervisor. Virtual machines can be utilized, for example, in virtualizing many different servers on a single host. (Portnoy 2012, 9, 15-16, 19, 35-36.)

3.2 Hypervisors

A hypervisor is a software layer which allows for some physical hardware to act as a host for virtual components. A hypervisor oversees sharing the physical resources available and acts as a management interface for the virtual components associated to it. Hypervisors are generally split into two groups: types 1 and 2. Type 1 hypervisors, or "bare-metal" hypervisors operate directly on a server hardware on operating system level. Type 2 hypervisors require an operating system to run atop on. (Portnoy 2012, 19, 21-23.) Refer to figure 5 for a simplified comparison of the hypervisor types.

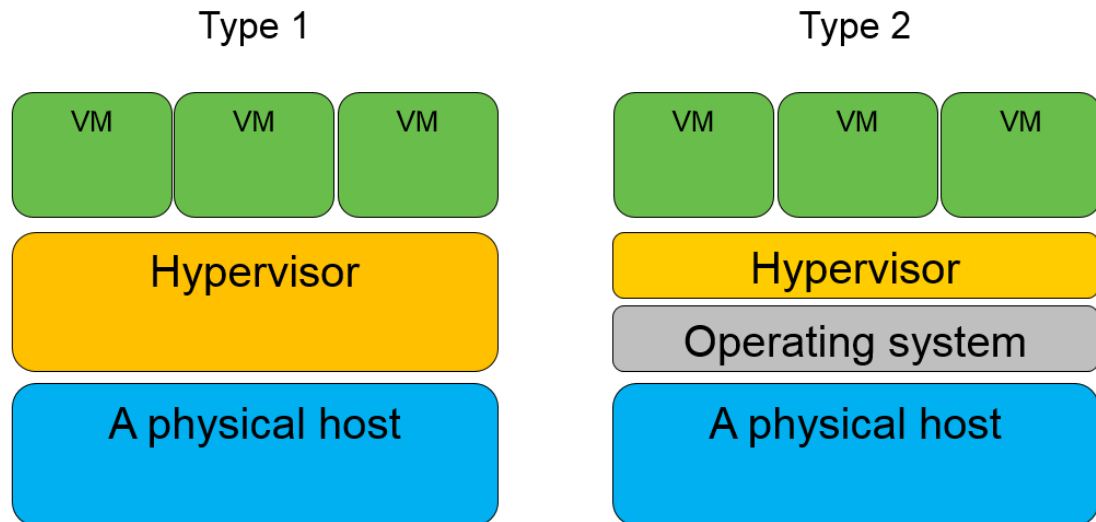


Figure 5. A simplified comparison between the two hypervisor types (Portnoy 2012, 22-23).

As type 1 hypervisors operate on operating system level, no other operating systems are present on the server. This allows the hypervisor to efficiently utilize the hardware of the host as there are no interruptions between the hardware and the hypervisor. Type 1 hypervisors are generally more efficient than type 2 hypervisors because there are no additional software components in type 1. (Portnoy 2012, 21-24.) This bachelor's thesis focuses on an environment with only type 1 hypervisors.

4 VMWARE

VMware, Inc. is a US-based company that focuses on virtualization and software-defined infrastructure products (VMware 2017a; VMware 2017b). The company's first product, a type 2 hypervisor was released in 1999. Being the first commercially available product of its kind, the company got an advantage on the market. (Portnoy 2012, 27-28.) According to Gartner analysts Bittman et al. (2016) VMware has maintained its position as the market share leader of x86 server virtualization infrastructure.

4.1 vSphere

VMware vSphere is a suite of software used for data center virtualization infrastructure. It consists of a plethora of different components and features, such as

a hypervisor, management software and storage solutions. vSphere aims to implement a complete platform for virtualizing most core components of a data center to streamline running the infrastructure and keeping the costs by doing so. (VMware 2015a.)

4.1.1 ESXi

ESXi is the hypervisor software of the vSphere platform. It is a type 1 hypervisor (see chapter 3.2) so it does not need an operating system to run. It is the current market share leader utilized for virtualizing x86 server computers and their hardware (Bittman et al. 2016). ESXi can support powerful VMs with such capabilities as 128 virtual processors and 4 TB of memory. The hypervisor is available as a free and limited version or as a part of the commercial vSphere suite. (VMware 2017c.)

VMkernel is the core of ESXi. It handles all the core functions of the hypervisor, such as sharing the resources for the VMs and management connections for host administration. In addition, it runs all the software components of ESXi, like a NTP client or a syslog client. VMkernel also supports more advanced functions like migrating VMs from one host to another live without interruptions to availability (see chapter 4.1.4). Some of the features require purchasing a license to operate. (Portnoy 2012, 29; VMware 2015a; VMware 2017c.)

ESXi can be configured in multiple ways. There is an application programming interface available for scripting support for languages like PowerShell. VMware also provides a PowerShell-based client called vSphere PowerCLI for configuration scripting. Some configuration can be done locally on the host utilizing the local interface referred to as direct console user interface (DCUI). If available, most day-to-day configuration operations are done with vCenter Server via the vSphere Client. (VMware 2017c.)

4.1.2 vCenter

vCenter Server is a centralized management system for the vSphere environment. Most aspects, like the hypervisors hosts or the VMs of the environment can be configured using vCenter. It aims to streamline and simplify the process of administering the platform. vCenter also assists in some functions of the environment like high availability failovers. (VMware 2015b.)

The environment can be configured using a vSphere Client, available with the vCenter server installation. VMware offers multiple clients, of which the HTML5-based web client referred to as vSphere Client is the preferred one. There is also a flash-based client available, called vSphere Web Client, which was the preferred client before vSphere 6.5 release. A C#-based legacy client also exists, but is no longer available as of release 6.5. (Prabhudev 2016; VMware 2015b.) As the implementation of this bachelor's thesis was based on vSphere 6.0, the flash client was used.

vCenter Server can be installed as a software installation on a server or as an appliance. The vCenter Server Appliance is a virtual machine, which runs SUSE Enterprise Linux as a server. The VM is preinstalled with vCenter Server and its required components. In addition, the VM comes preconfigured with an installation wizard utility, which can be utilized for a straightforward initial configuration and installation of vCenter Server. (VMware 2017d.) Deploying vCenter as an appliance over a software installation is ideal in some situations, such as when simplicity is desired or the environment is of small to medium size (Fenech 2016).

4.1.3 Distributed Switch

VMware vSphere Distributed Switch (vDS) is a component of the vSphere suite available with the Enterprise Plus -license. vDS is a virtual network switch capable of functionality such as layer 2 switching, VLANs, private VLANs, teaming of NICs and failover policing. In addition, the VDS acts as a centralized network

management interface accessible on vCenter Server via vSphere Client. Configuring the virtual network through vDS unifies the configuration for all ESXi hosts associated with the vDS. (VMware 2017e; VMware 2017f.)

Teaming physical NICs in vDS allows for redundant failover setups. The NICs can be grouped into uplink groups to team together multiple physical links from multiple ESXi hosts. The uplink groups can then be associated with vDS port groups which manage the network policies as desired. (VMware 2017e.)

A distributed port group acts as a network intermediary between the VMs and the uplinks in the environment. The port groups can direct the traffic to correct uplink groups and VLANs (if applicable). In addition, network policies such as failover behavior, load balancing and traffic shaping methods can be implemented specific to port groups. For example, a port group with failover policy enabled could be associated to an uplink group with teamed NICs to provide redundant network connectivity to VMs of the port group. An example of a vDS setup like this can be seen in figure 6. (VMware 2017e.)

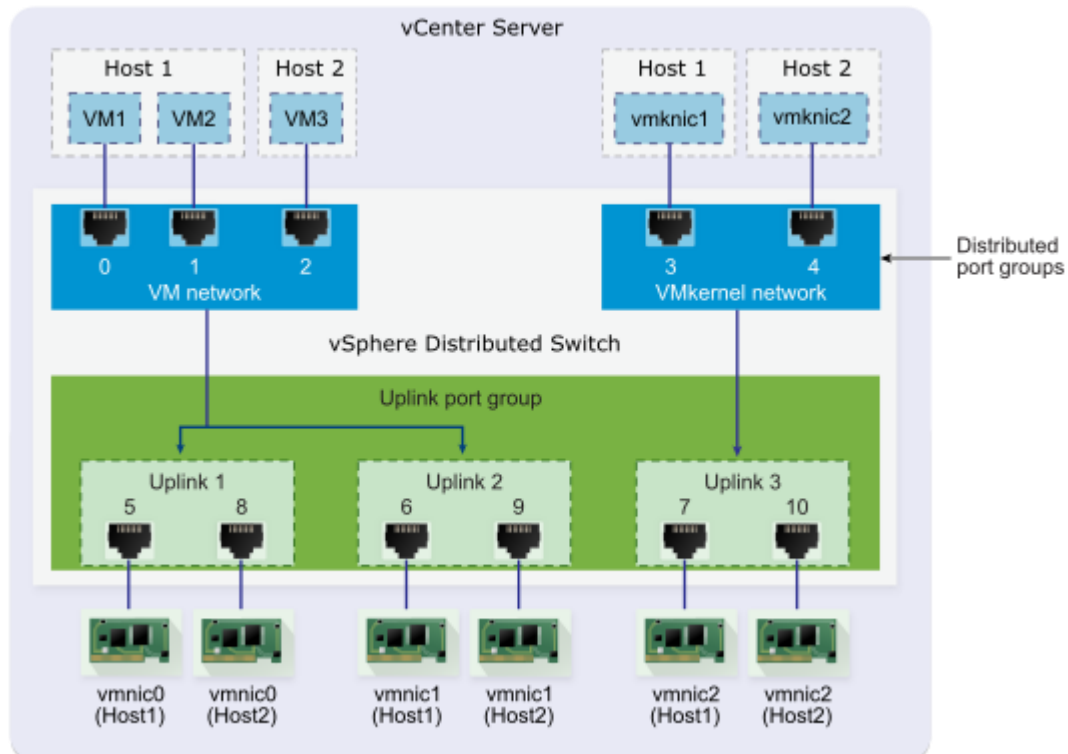


Figure 6. An example vDS setup with teamed NICs (VMware 2017e).

4.1.4 Cluster features

This chapter describes some of the features of the virtualization clusters available in the vSphere environment. The described features are relevant to the implementation phase of this bachelor's thesis.

vSphere vMotion is a technology that provides functionality for migrating VMs across ESXi hosts, storage nodes and vCenter Servers. Migration of VMs can be done live with the VMs powered on without causing interruptions to service. The migration is done by capturing the full state of a running VM and transferring the data to another host utilizing a fast network connection. If the VM is stored in a storage shared between the hosts, no storage data needs to be moved and the migration process will take minimal time to complete. (VMware 2007; VMware 2015a.)

vSphere High Availability (HA) is a feature which implements some automation for cases of failure in the vSphere environment. HA is orchestrated and configured by a vCenter Server installation. In a HA enabled cluster, an ESXi host is elected as a master, which then exchanges information about possible failures with the vCenter. HA can monitor failure conditions like host-, VM- or network partition failures and act accordingly. In cases of failure, HA can automatically handle restoration of failed services like VMs. (VMware 2015a; VMware 2017g.)

vSphere Distributed Resource Scheduler (DRS) is a feature available for clusters in the vSphere environment which allows for monitoring and automation of load balancing. When the feature is enabled in a cluster, DRS monitors the associated ESXi hosts for the status of available resources such as CPU utilization and active memory. Configurable by policies, DRS can automatically apply recommendations of migrating VMs from one host to another to achieve optimal workload balance. (VMware 2015a; VMware 2017h.)

Enhanced vMotion Capability (EVC) is a configuration option for clusters in the vSphere environment which, when enabled, prevents ESXi hosts with incompatible CPUs from entering a cluster. EVC is configured a predefined baseline set of features which a CPU needs to support, or it will be denied entry to the cluster. The purpose of enabling EVC is to prevent vMotion migrations from failures caused by CPU incompatibility in the cluster. (VMware 2017i.)

4.2 vSAN

VMware Virtual SAN (vSAN) is a storage technology for ESXi hosts which utilizes the locally connected storage devices of the hosts to create a virtualized storage area. vSAN is enabled per cluster, so all the hosts associated to a cluster contribute to the combined vSAN datastore. The datastore also acts as shared storage for all the hosts in the cluster, enabling the use of vSphere features such as vMotion, HA and DRS (see chapter 4.1.4) without the need for external storage. vSAN also implements fault tolerance by distributing the data across the storage pool. Refer to figure 7 for a visualization of vSAN operation. (VMware 2017j; VMware 2017l.)

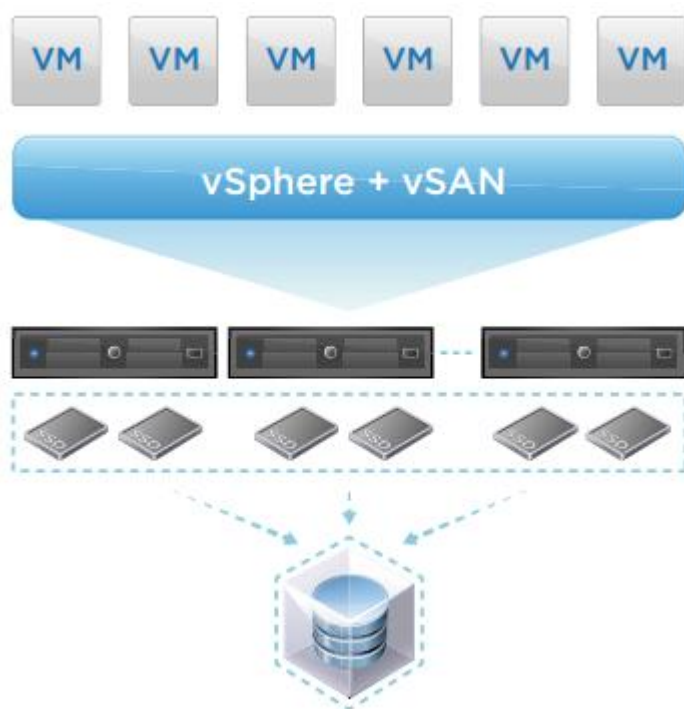


Figure 7. A visualization of vSAN operation (VMware 2017l).

One of the key features of vSAN is the capability to add read & write buffers which utilize flash disk drives. The clusters can be built in a way that uses traditional hard disk drives for capacity storage and flash disks on top of them for caching I/O requests. The cache layer can speed up disk operation dramatically in such disk setups. Disk setups can also be flash only, if high operation speeds are desired. (VMware 2017k; VMware 2017l.)

vSAN integrates itself into vSphere environments well, as the software is a part of the ESXi VMkernel. All management and monitoring operations can be done centralized through a vCenter Server installation. vSAN aims to drive down costs by removing the need for external storage units and integrating the management into vCenter. In addition, vSAN can scale automatically by adding more storage devices to hosts or adding hosts to the cluster. (VMware 2017l.)

5 PRIVATE VLANS

Sharing a regular VLAN subdomain can raise concerns on security. For example, an unauthorized user could listen to all layer 2 broadcasts if the user resides in the same VLAN. Private VLANs aim to mitigate these kinds of issues by implementing traffic controls inside the VLAN. (RFC 5517: 2010, 3.)

Private VLANs (PVLAN) partially function in a comparable manner to regular layer 2 VLANs, but PVLANS have some additional security measures implemented. The purpose of PVLANS is to add traffic restrictions to a VLAN broadcast domain. PVLAN separates regular VLANs into multiple “port” types, which indicate the type of traffic restrictions enforced on the ports. (RFC 5517: 2010, 4.)

There are three distinct types of PVLAN ports: promiscuous, isolated and community. Promiscuous ports can send and receive traffic to all other port types. Community ports can, in addition to promiscuous ports, communicate with other community type ports within the same PVLAN domain. Isolated ports are not able to communicate with any other port regardless of type apart from promiscuous ports. Refer to figure 8 for a summary of possible communication types. (RFC 5517: 2010, 4-6.)

	Isolated	Promiscuous	Community1	Community2
Isolated	deny	permit	deny	deny
Promiscuous	permit	permit	permit	permit
Community1	deny	permit	permit	deny
Community2	deny	permit	deny	permit

Figure 8. A summary of possible communication privileges between port types (RFC 5517: 2010, 6).

When implementing PVLANS, a unique VLAN identifier needs to be paired with PVLAN ports for the implementation to function correctly. The unique VLAN ID is referred to as a primary VLAN and it functions as a promiscuous PVLAN port. PVLAN ports paired with the primary VLAN are referred to as secondary VLANs. Secondary VLANs can be either isolated or community type PVLAN ports. One or more secondary VLANs can be paired to the primary VLAN. (RFC 5517: 2010, 7.)

6 IMPLEMENTATION

This part of the bachelor's thesis covers the implementation of the virtualization environment used in the Cybergame project, focusing on software. All implementation phases are described, starting from clearing the previous installation leading up to the fully implemented system. In the end, multiple failure tests were conducted and documented.

6.1 Hardware

At the start of the implementation phase, all the hardware was already physically in place. The server hardware consisted of four Dell PowerEdge R720xd servers with 2x10 gigabit ports and 2x1 gigabit ports. For storage, the servers had 2x200GB SSD flash disks and 8x1TB hard disks combined to 2x4TB with RAID0.

In addition, there already was an obsolete implementation of the virtualization environment installed on the server hardware. After examining the implementation, it was deemed unusable to build upon in its current state. Because of this, all the server hardware was purged completely and the installation was started anew

from scratch. Purging the old environment was done by manually wiping all storage partitions to ensure there would not be any obsolete data left over from the old implementation.

As the hardware had already been installed beforehand, the servers were already connected to the Dell N4032F-series data center switches. The 10Gb uplinks were connected to different switches physically for redundancy. There was no dedicated management uplink available, so one was decided to be installed. A 1 Gb network link was connected from the Cybergame management switch to all the ESXi hosts' port 3 (identified by nic2 in vCenter). The 10 Gb uplinks were configured as trunk interfaces while the management ports were restricted to the management VLAN only. The physical topology of the environment can be seen in chapter 2.1.1.

6.2 Hypervisor

After fully purging the old environment and confirming the physical connectivity was in order, ESXi hypervisor software was to be installed on the server hardware. As there was no remote access to the servers available at this point, ESXi was installed locally. The installation media was acquired on a DVD disc.

6.2.1 Installation

The DVD media contained a bootable installation wizard utility. After starting the installation process, the installer asked for all required variables, such as root user login information and the storage where the software was to be installed. ESXi was specified to be installed on the first flash disk of each host. The first disk was selected to allow as much as possible storage space to be assigned for vSAN storage later. After the installation wizard was complete, the servers were rebooted to confirm successful installation and to configure the initial network settings for remote administration.

6.2.2 Post-installation tasks

The initial configuration was done in the ESXi direct console, i.e. locally on the server hardware using the root user configured in chapter 6.2.1. For remote administration, a working network connection was needed. To achieve this, static IP- and gateway addresses were configured for the hosts according to the Cybergame network design as seen in chapter 2.1.3. In addition, to ensure that management traffic would go through the correct interface, nic2 (port 3) was selected as the default management interface. Other adapters were left deselected. To check if the settings were configured correctly and there indeed was network connectivity, a ping request (ICMP) was sent to the gateway address.

As the connection succeeded, the hosts could now be remotely configured. Most of the configuration in the environment is done in vCenter, which was not yet deployed at this point. Before deploying a vCenter installation, a local datastore needed to be added on at least one of the hosts to allow vCenter deployment. The datastore was necessary to temporarily store the vCenter appliance before vSAN was deployed later.

As all the server hardware had the same disk configuration, the datastores were configured on all of them in the same manner. The datastores were assigned to use all available space left after the ESXi installation on the first flash disk. The first flash disk was chosen since the second flash disk was to be completely claimed by vSAN later. This also allowed making use of all the leftover disk space on the first flash disk and have a host specific local datastore as a backup in case one is ever needed. For identification, all the datastores were named CG-ESXn-SSD where n indicates the number of the specific host (1-4).

The hosts were configured using vSphere Client, as vCenter was not yet implemented. First, a configuration connection was made using each hosts' IP address and the root user account configured in chapter 6.2.1. An automatically partitioned datastore was then created using a wizard function found in *Inventory > Datastores > Add datastore*. Refer to figure 9 for a summary of the new datastore.

Disk layout:

Device	Drive Type	Capacity	LUN
Local DELL Disk (naa.6b083fe0cfb49d001ca59192055666a...	SSD	185,75 GB	0

Location
/vmfs/devices/disks/naa.6b083fe0cfb49d001ca59192055666a1

Partition Format
GPT

Primary Partitions

	Capacity
Legacy MBR (Local DELL Disk (naa.6b083fe0cfb49d001ca59..	4,00 MB
Legacy MBR (Local DELL Disk (naa.6b083fe0cfb49d001ca59..	250,00 MB
Legacy MBR (Local DELL Disk (naa.6b083fe0cfb49d001ca59..	250,00 MB
VMware Diagnostic (Local DELL Disk (naa.6b083fe0cfb49d001c..	110,00 MB
Legacy MBR (Local DELL Disk (naa.6b083fe0cfb49d001ca5919...	286,00 MB
VMware Diagnostic (Local DELL Disk (naa.6b083fe0cfb49d001c..	2,50 GB
Legacy MBR (Local DELL Disk (naa.6b083fe0cfb49d001ca5919...	4,00 GB
VMFS (Local DELL Disk (naa.6b083fe0cfb49d001ca5919205566...	178,37 GB

File system:

Properties

Datastore name: CG-ESX1-SSD

Formatting

File system: vmfs-5
Block size: 1 MB
Maximum file size: 2,00 TB

< Back Finish Cancel

Figure 9. A summary of the new datastore

After creating the datastores, NTP service was enabled on all the hosts to ensure the times and dates are consistent on the environment. This was done early on to prevent potential problems, like authentication failures, during the implementation. NTP was enabled by navigating to *Host configuration > Time configuration > Properties > Options > NTP settings*. As per Cybergame network rules the hosts were configured to use all available Funet NTP servers. When the changes were saved, the NTP service was automatically started.

6.3 vCenter

As the ESXi hosts were running correctly, vCenter Server Appliance was to be deployed next. The vCenter was chosen to be deployed as an appliance instead

of a server software installation. At this point, the environment had no server installations available, nor was it too big sized for an appliance deployment. In addition, deploying vCenter as an appliance is a streamlined and uncomplicated way of deploying a vCenter installation.

6.3.1 Deployment

The deployment media was in a disk image file, which was mounted on a computer. The media contained a setup wizard, which allowed for simplified initial configuration with a web browser. For the wizard to function, a client integration browser plugin included with the installation image was installed to the local computer.

The wizard first asked for an ESXi host to which it would deploy the VM. As the VM could later be moved anywhere within the cluster, it would not matter on which host the VM would be deployed at first. The first host was selected as a target. Additional details, such as the name of the VM and credentials for vCenter specific root user were asked. The *Embedded Platform Services Controller* -option was selected to ensure that vCenter would use its own database implementation. The installer then asked to create a new single sign-on domain for the embedded controller. The domain would be used as a secondary authentication method in vCenter in case the IT department's active directory server cannot be reached. The domain was also used for authentication before implementing authentication from the AD server. For creating the domain, administrator credentials, domain- and site name were specified.

Next, the installer asked for the size and destination datastore for the deployment. Deployment size was set to small, as it was the most ideal option for this environment (up to 100 hosts & 1000 VMs). The host's local SSD datastore (see chapter 6.2.2) was selected, as there were no alternative datastores available at this point. Finally, network details such as virtual network adapter, IP address, NTP server addresses were specified. The addressing for vCenter was set according to Cybergame's design as seen in chapter 2.1.3. The virtual network adapter automatically assigned itself with the only available default network.

Funet NTP servers. Summary and details of the deployment can be seen in figure 10.

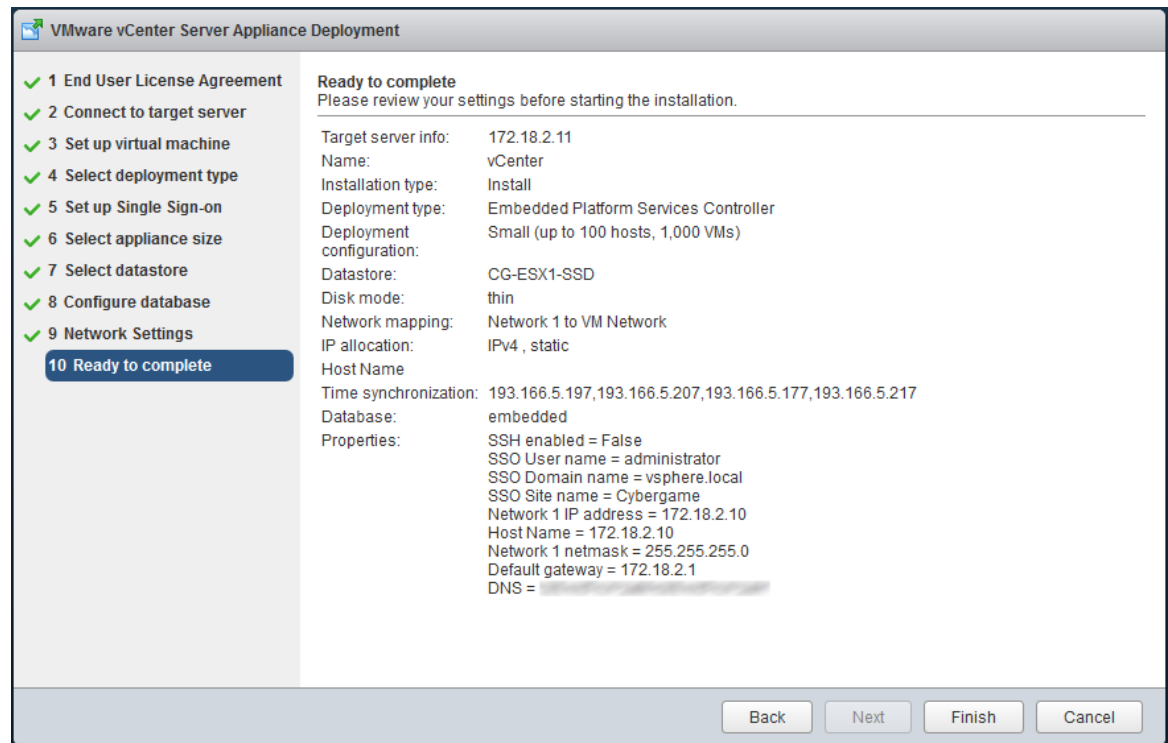


Figure 10. A summary of the vCenter deployment

After clicking finish, the deployment wizard started the deployment process without problems. When the process was finished, the freshly deployed appliance VM was started and verified working by logging in to vSphere Web Client with the root account. The web client was accessible with a web browser by navigating to the vCenter's IP address using HTTPS.

6.3.2 Updating the vCenter appliance

To ensure stability, the availability of the latest features and to prevent possible software bugs, the vCenter server was to be patched with the latest updates before proceeding further. The update process could be initiated through the appliance management interface, which could be accessed by browsing to the vCenter's IP address through port 5480 using the root account.

In this environment, per Cybergame network rules, the vCenter Appliance VM did not have access to the Internet. Because of this, the newest available update was acquired as a disk image file. The image file was copied to the local SSD datastore of the first host with vSphere Client. After the file was copied to the datastore successfully, the disk image was mounted to the vCenter Appliance VM through VM specific settings.

When the disk image was mounted to the VM and *Check CDROM* -button was clicked in the management interface, the update was detected from the disk image. *Install all updates* -button was clicked and the automated update process was started. Finally, after the updates were successfully installed, the appliance VM was restarted. Refer to figure 11 for an example update scenario.

Update	
Current version details	
Vendor	VMware, Inc.
Appliance name	VMware vCenter Server Appliance
Update version	6.0.0.10000 Build Number 3018521
Description	vCenter Server with an embedded Platform Services Controller
Available updates	
Update status	Update source: ISO. Updates are available.
Reboot required	Yes
Update last checked at	5/17/2016, 8:37:56 AM
Update version	6.0.0.20000 Build Number 3634791
▼ More details	
Appliance name	VMware vCenter Server Appliance
Description	VC-6.0.0U2-Appliance-FP
Release date	March 15, 2016
Severity	critical
Category	bugfix
Summary	VMware vCenter Server Appliance 6.0 Update 2
Knowledge base	http://kb.vmware.com/kb/2138600

Figure 11. Management interface detected the available update from the disk image

6.3.3 Authentication from AD

As all users of this environment already have an account and possible user groups specified in the IT department's AD, it was decided that authentication for vCenter administration was to utilize the IT department's AD servers. In addition to vCenter authentication, the AD also provides centralized user management for other services as well. The implementation eliminated the need for multiple user management systems.

To enable authentication via AD in vCenter, a new single sign-on identity source needed to be added to vCenter. Related settings were configured with vSphere Web Client by navigating to *Administration > Single Sign-On > Configuration > Identity sources > Add*. As explained in chapter 2.1.2, the AD servers could only be accessed via the read-only domain controllers from inside the Cybergame network. As the RODCs act like LDAP-type servers, *AD as an LDAP server* -option was selected for the new identity source type. Other required information specified included base domain name for users and groups, domain name, primary and secondary LDAP URL and privileged AD credentials. Refer to figure 12 for detailed identity source configuration.

Identity source type:

☐ Active Directory (Integrated Windows Authentication)
☒ Active Directory as an LDAP Server
☐ Open LDAP
☐ Local OS

Identity source settings

Name:

Base DN for users:

Domain name: ⓘ

Domain alias:

Base DN for groups:

Primary server URL: ⓘ

Secondary server URL:

Username: ⓘ

Password:

Figure 12. Summary of the new identity source configuration

After creating the identity source, the necessary administration rights were linked with correct user groups retrieved from the IT department's AD servers. This was done by assigning the "SysOps" and "Staff" -user groups from AD to all vCenter's administration-related user groups. The group settings were accessed by navigating to *Administration > Single Sign-On > Users and Groups*. The groups were added by selecting *Add member* and then specifying the correct groups from the list. Finally, the default single sign-on domain option was changed to the newly created identity source by clicking *Set as Default Domain* in the *Identity Sources* menu. The implementation was tested by logging in with multiple accounts and was confirmed working.

6.3.4 Cluster configuration

At this point, the ESXi hosts were to be added to vCenter's inventory. After adding the hosts, most of the configuration could be done using vSphere Web Client instead of using the command line or the vSphere client. Before adding the hosts,

it was necessary to add a datacenter object. The object was created by navigating to *Hosts & inventory* and selecting *Create datacenter*. The hosts were then added in the same view, choosing *Add host* instead. Adding the hosts required using the root credentials created in chapter 6.2.1 in addition to specifying the hosts' IP addresses.

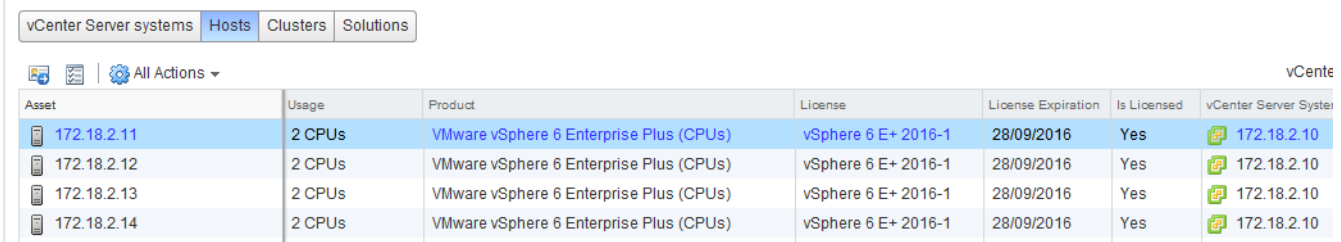
After the hosts were successfully added, a cluster object was created for them by right-clicking the datacenter object created earlier and selecting *New cluster*. At this point, none of the additional cluster features, such as high availability, was enabled to mitigate potential problems during the implementation. The hosts were then added to the new cluster by dragging and dropping them inside the cluster using the inventory view.

6.3.5 Licensing

Before moving forward, all the ESXi hosts and the vCenter server were to be licensed. License management could be accessed by navigating to *Administration > Licensing*. Before assigning licenses to hosts and the vCenter, the necessary licenses were created under *Licenses*-tab. The license keys were entered and assigned identifying names. After the licenses were added, they were assigned to the vCenter server and the four hosts via the *Assets*-tab by clicking *Assign license*. Refer to figure 13 for example on adding license keys and figure 14 for a view of licensed assets.

New Licenses			
✓ 1 Enter license keys			
✓ 2 Edit license names			
3 Ready to complete			
Edit license names Each license key is placed in a separate license. Review the licenses and name them as appropriate.			
License name:	vCenter 6 STD 2016-1		
License key:		Expires:	/2016
Product:	VMware vCenter Server 6 Standard (Instances)		Capacity: 1 Instances
License name:	vSphere 6 E+ 2016-1		
License key:		Expires:	/2016
Product:	VMware vSphere 6 Enterprise Plus (CPUs)		Capacity: 8 CPUs

Figure 13. Adding new licenses



The screenshot shows the vCenter interface with the 'Hosts' tab selected. Below the navigation bar, there is a table titled 'vCenter Server systems' showing the following data:

Asset	Usage	Product	License	License Expiration	Is Licensed	vCenter Server System
172.18.2.11	2 CPUs	VMware vSphere 6 Enterprise Plus (CPUs)	vSphere 6 E+ 2016-1	28/09/2016	Yes	172.18.2.10
172.18.2.12	2 CPUs	VMware vSphere 6 Enterprise Plus (CPUs)	vSphere 6 E+ 2016-1	28/09/2016	Yes	172.18.2.10
172.18.2.13	2 CPUs	VMware vSphere 6 Enterprise Plus (CPUs)	vSphere 6 E+ 2016-1	28/09/2016	Yes	172.18.2.10
172.18.2.14	2 CPUs	VMware vSphere 6 Enterprise Plus (CPUs)	vSphere 6 E+ 2016-1	28/09/2016	Yes	172.18.2.10

Figure 14. Licensed assets

6.3.6 Updating the hosts

To keep the hosts up to date, it was decided to implement vSphere Update Manager in the environment. VUM needed to be installed on a Windows-based server, so a VM with Windows Server 2012 was created. The VM was configured with 2 virtual hard disks with 40 GB of disk space for both disks. First disk would contain the Windows Server installation and related files, while the second disk would have all data related to VUM and Microsoft SQL Server required by VUM. The second disk was thin provisioned, so actual storage would not be allocated before it would really be used. The second disk could also be resized later if ever needed.

Windows Server 2012 was first installed and configured with remote administration access. VUM installation disk image file was mounted to the VM's optical drive. The media contained an installation wizard utility, which made the installation and initial configuration straightforward. At the beginning of running the utility, *Microsoft SQL Server 2012* was selected to be installed for simple database compatibility, requiring no additional configuration. The second virtual hard disk was selected as the target for all VUM-related data. The wizard also asked to specify information such as the VUM VM's IP address and the ports used for VUM functionality, vCenter's IP address and the vCenter's administrative account.

At first, the installer could not properly connect to the vCenter server because of network traffic restrictions in the Cybergame network. For vCenter, TCP ports 9084, 8084 and 9087 and for hosts TCP 9084 were opened and the problem was

resolved (VMware 2017n). During the installation phase, the wizard utility automatically downloaded the newest update files and registered the update manager plugin to vSphere Web Client.

With the installation of VUM, some default baselines were included for simple update manager configurations. For updating hosts, critical and non-critical host upgrade baselines were to be attached to the cluster object created in chapter 6.3.4. The baselines were attached by navigating to the cluster's *Manage*-tab > *Update manager* -subtab and selecting *Attach baselines*. After attaching the baselines, *Scan for updates* -button was clicked in the same tab. Refer to figure 15 for an example of attaching baselines to hosts.

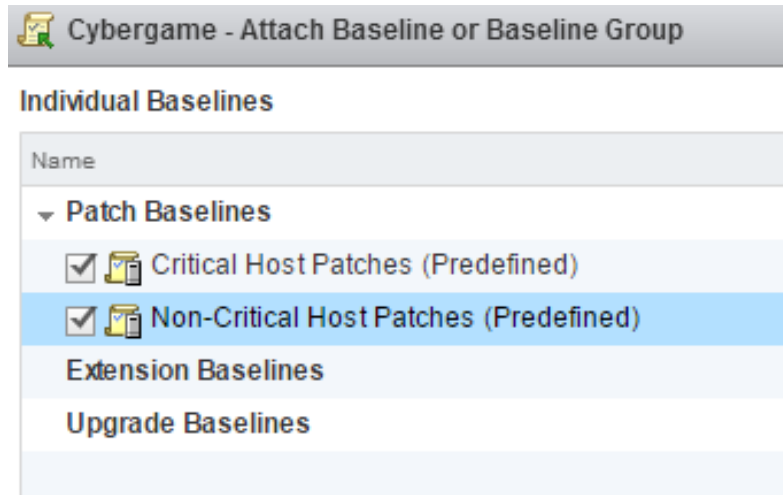


Figure 15. Attaching baselines to clustered hosts

After the newest updates were queried from the VUM server, the update process was to be tested by updating a single host. All the hosts were not updated in parallel at this point of time, because there was no high availability support or vSAN yet implemented to the environment. This prevented moving the powered-on VMs, like the vCenter appliance, between hosts in the cluster. The vCenter appliance must stay powered on and available during the updates, so updating all hosts at once could not be tested at this phase of implementation.

Still in the same tab, *Remediate* -button was clicked to run the remediation wizard. Both attached baselines, host 4 and all offered patches were selected. Update was scheduled to run immediately and other remediation options were left untouched. After clicking *Finish*, the selected host was updated with the latest patches and rebooted as expected. This procedure could be used at a later point to update other hosts in parallel by selecting all the hosts at once and configuring other possible remediation options. Refer to figure 16 for a summary of remediation events.

Cybergame - Remediate

Ready to complete
Review your settings selections before finishing the wizard.

Baselines

Remediation type	Host remediation
► Patch baselines	2

Target Objects

► Hosts	1
---------	---

Patches and Extensions

► Patches	303
► Extensions	21

Scheduling

Remediation time	Immediately
------------------	-------------

Maintenance mode options

Do Not Change VM Power State
Retry entering maintenance mode every 5 minutes, 3 retries

Cluster remediation options

Disable Distributed Power Management

Back Next Finish Cancel

Figure 16. Summary of remediation wizard utility

Before proceeding with the implementation, an automatic VUM schedule was enabled in the vSphere Web Client. The scheduling settings could be accessed by navigating to *Update Manager* > The VUM server object > *Manage*-tab > *Settings*-subtab and selecting either *Download Schedule* or *Notification Check Schedule*. By clicking *Edit*, a schedule was set in both download- and notification checks so the newest data would be retrieved from the Internet daily and the hosts would query the VUM server hourly. The schedule was confirmed to be firing as intended by looking at the vCenter's task monitor later.

6.4 Networking

As all the hosts share the same physical NICs and high availability was one of the goals in the environment, it made sense to utilize the vSphere Distributed Switch architecture. The vDS would be used for VM traffic, vSAN traffic, handling network failovers and traffic restriction through PVLANS. All networking apart from host management traffic would utilize the vDS.

Host management connections utilize host-specific VMware Standard Switches, which are set up by default at the time of installing ESXi to the hosts. Before implementing any networking with vDS, all traffic through the management VMkernel virtual adapter (vmk0 by default) was restricted to management traffic only. This was done in vSphere Web Client by navigating to a host > *Manage*-tab > *Networking*-subtab > *VMkernel adapters* > vmk0 > *Edit* and unchecking all checkboxes except *Management traffic*.

6.4.1 Implementing vDS

The implementation of vDS began with creating a vDS in the inventory. The vDS was created by navigating to the datacenter object, right-clicking it and selecting *New distributed switch*. The vDS was created with version 6.0.0 as there are only ESXi 6 hosts in the environment. The vDS was also given a name (Cybergame) and 2 uplinks, as there are 2 NICs for vDS enabled networking available per host. Other settings were left untouched.

After the vDS was created, the uplink group and the uplinks were named for identification purposes. The uplinks were renamed by right-clicking the vDS object > *Edit settings* > *Edit uplink names*. After renaming the uplinks, the uplink group was renamed by right-clicking the uplink group object in the inventory and selecting *Rename*. The uplink group was renamed “10G Uplinks” and the uplinks were named “Primary (1)” and “Secondary (2)”, where physical nic0 is 1 and nic1 is 2.

Next, the hosts’ physical adapters were to be assigned to the uplink group in the vDS. The process began with navigating to the vDS object, right-clicking it and

selecting *Add and Manage Hosts*. First, *Add hosts* was chosen and all four hosts were selected to be added. As there was no need to reassign VMkernel adapters or VM networks, only *Manage physical adapters* was chosen at this point. In the next screen the hosts' physical adapters could be assigned to desired uplinks in the vDS. On all hosts, both NICs were assigned to the uplinks by selecting a NIC, clicking *Assign uplink* and selecting the uplink. After all the NICs were assigned, the task was finished by proceeding to the end of the management task and clicking *Finish*. Refer to figure 17 for an example of assigned NICs.

Add and Manage Hosts

✓ 1 Select task

✓ 2 Select hosts

✓ 3 Select network adapter tasks

4 Manage physical network adapters

5 Manage VMkernel network adapters

6 Analyze impact

7 Ready to complete

Manage physical network adapters

Add or remove physical network adapters to this distributed switch.

Assign uplink

Unassign adapter

Reset changes

View settings

Host/Physical Network Adapters	1 ▲	In Use by Switch	Uplink	Uplink Port Group
▼ 172.18.2.11				
▼ On this switch				
vmnic0		Cybergame	Primary (1)	10G Uplinks
vmnic1		Cybergame	Secondary (2)	10G Uplinks
▼ On other switches/unclaimed				
vmnic2		vSwitch0	--	--
vmnic3		--	--	--
▼ 172.18.2.12				
▼ On this switch				
vmnic0		Cybergame	Primary (1)	10G Uplinks
vmnic1		Cybergame	Secondary (2)	10G Uplinks
▼ On other switches/unclaimed				
vmnic2		vSwitch0	--	--
vmnic3		--	--	--

Figure 17. An example of host NICs assigned to uplinks

6.4.2 Implementing vDS port groups

Before any VM networking could be done via vDS, port groups needed to be created. For each VLAN used, a port group was to be configured. The process of creating a port group was the same for all VLANs apart from different traffic policies in the “vSAN”-port group.

The task for creating port groups was started by navigating to the vDS object, right-clicking it and selecting *Distributed port group > New*. First, the port group was given a name according to which VLAN was being created. Next, the VLAN type was set as *VLAN* and the VLAN number was specified. *Customize default*

policies configuration -checkbox was selected to enable uplink customization while creating the port group. Port settings were left untouched, as by default the amount of virtual ports will scale up automatically if needed. Refer to figure 18 for these settings.

Figure 18. General port group settings

In the policy configuration phase, all sections apart from the *Teaming and Failover* -section were left untouched. In the *Teaming and Failover* -section, uplink failover behavior could be customized specific to the port group in question. Failover behavior was configured in port group options mainly for two reasons. First, as explicit uplink selection per VLAN was desired, configuring the uplink priority per port group is required. Second, configuration through vDS-related options streamlines configuration operations, as it can in some cases eliminate the need to configure other devices like switches if changes are needed.

All VLANs apart from “vSAN” were configured to explicitly use the “Primary” (nic0) uplink in normal conditions. In case of link failure, “Secondary” (nic1) uplink would activate automatically and return to normal when the link is restored. This behavior was achieved by selecting *Use explicit failover order* as the load balancing mode, and assigning uplinks to active and standby groups as described. Other settings were left untouched. For the “vSAN” port group, the uplink order was reversed to separate VM traffic from vSAN traffic. This policy was created to

potentially increase operation speed for critical services. After configuring the fail-over-related changes, the task was proceeded to the last page and saved by clicking *Finish*. Refer to figure 19 for an example policy setup and figure 20 for an example of port group topology.

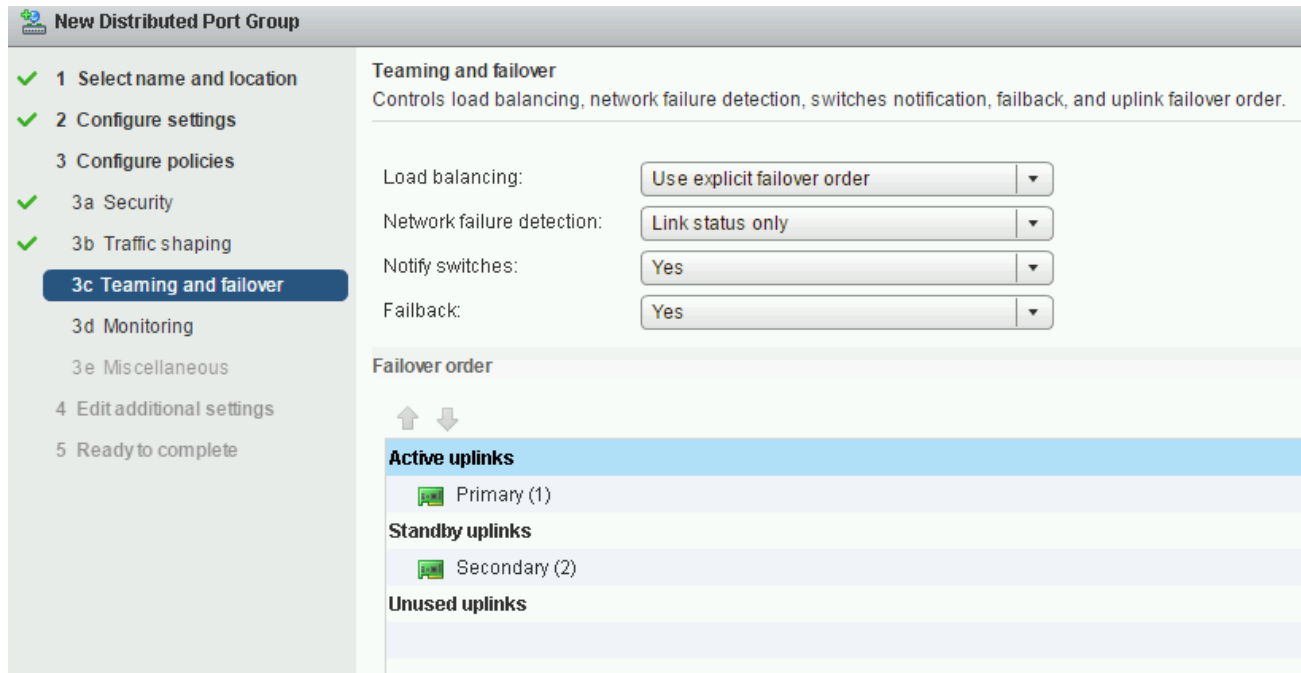


Figure 19. Port group policy setup for most VLANs

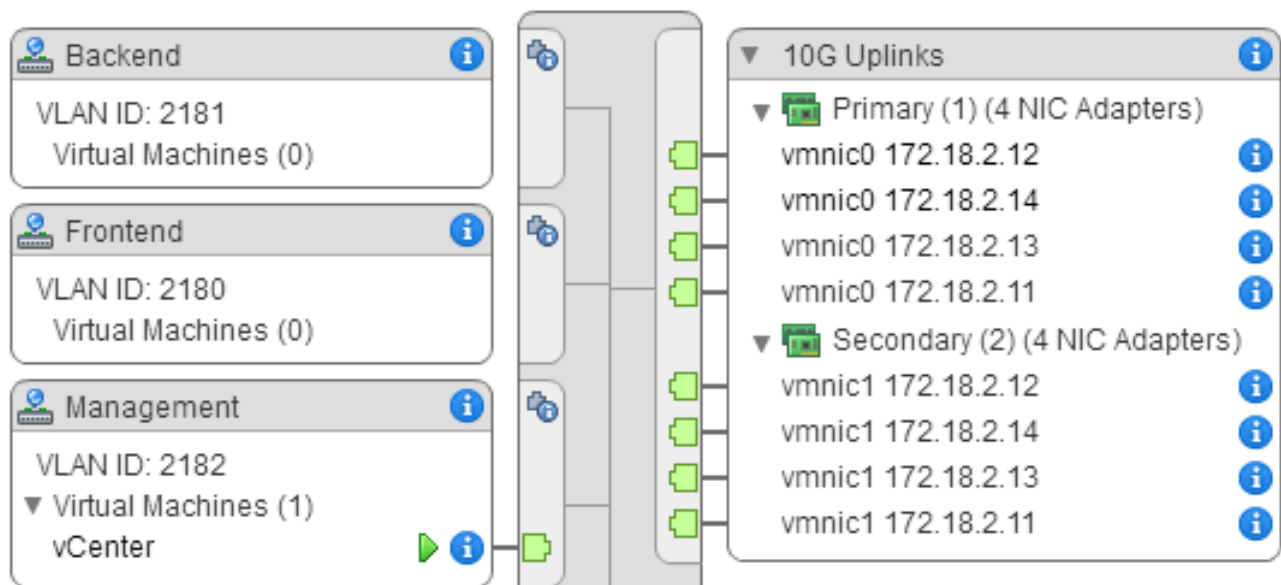


Figure 20. An example of the topology view of some port groups with uplinks visible

The functionality of the port groups was tested on a basic level by using a test VM with network connectivity. For every port group, an IP address from the VLAN

in question was assigned for the VM and the virtual network adapter was connected to the specific port group. Then, a ping request (ICMP) was sent to the VLAN's gateway from the VM. As the request was successful, the port group was deemed working.

6.4.3 Private VLANs

During the first attempt of implementing Private VLANs, it was quickly discovered that the VLANs could not be mapped to PVLANS. This was because all the VLANs were already assigned to existing port groups previously, which prevented any PVLAN mappings. Because of this, the port groups created previously had to be removed altogether. The PVLANS could then be configured correctly.

Before creating any PVLAN mappings in the vDS, the PVLANS needed to be configured into the Cisco ASA firewalls and the data center switches in the Cybergame environment. Compatibility for the PVLANS in the firewalls was achieved by adding a secondary VLAN entry into the already existing VLAN interfaces in the firewall configuration. The secondary VLAN ID used for each VLAN was the corresponding isolated PVLAN ID. The configuration line added to all VLAN interfaces was *vlan 21XX secondary 22XX*, where 21XX was the primary VLAN, and 22XX the isolated PVLAN.

For PVLAN compatibility in the data center switches, the existing VLAN entries needed to be linked with new PVLAN entries. For existing VLAN entries, the following lines were added to the configuration:

```
private-vlan primary
private-vlan association 22XX
```

New entries to match the link were created with the following configuration:

```
vlan 22XX
name "(VLAN name)-Isolated"
private-vlan isolated
```

The configuration for PVLANS could be found by navigating to the vDS object > *Manage*-tab > *Private VLAN* > *Edit*. For each VLAN, a primary entry was created with its VLAN number. Creating a primary entry automatically created a promiscuous entry with the same VLAN number as the primary entry. In addition, an isolated PVLAN entry was manually created under the primary entry according to the VLAN design seen in chapter 2.1.3. Refer to figure 21 for the PVLAN setup.

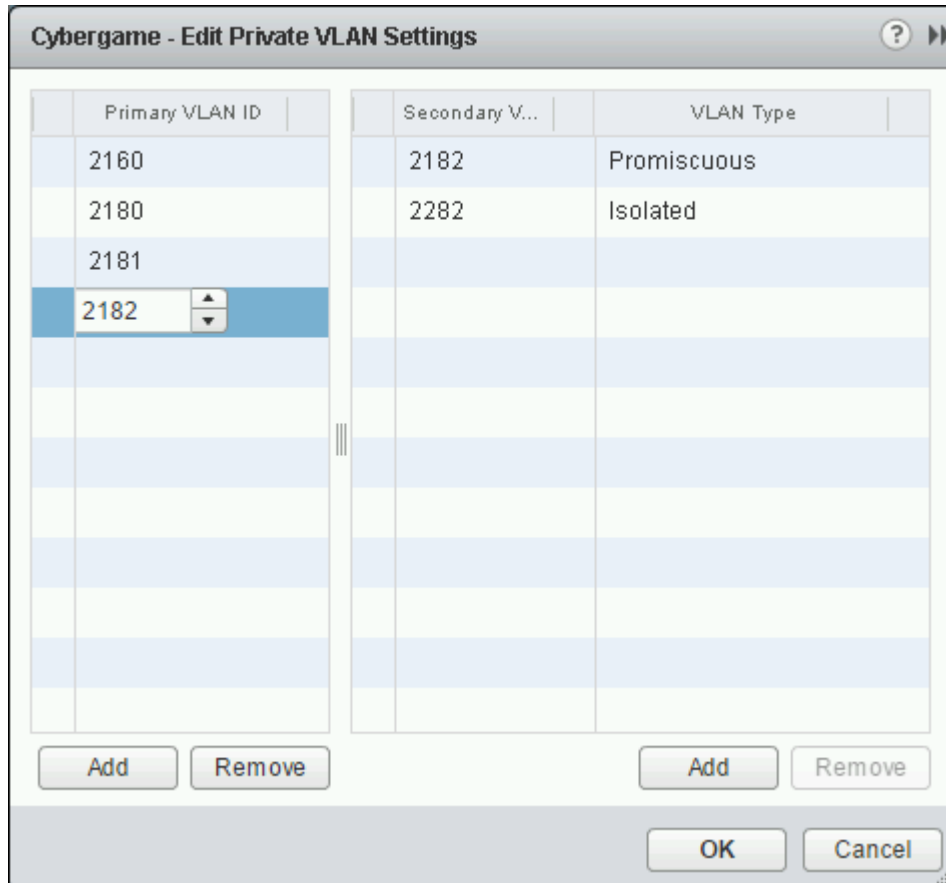


Figure 21. PVLAN configuration window

After the PVLANS were mapped properly, the vDS port groups were to be created once more. A port group can only have one type of VLAN access active, so instead of one port group per VLAN, two port groups per VLAN were created. The port groups were created in the same manner as seen in chapter 6.4.2, apart from the VLAN type selection. *Private VLAN* was selected as the type instead, and the correct PVLAN mapping was selected as the ID. The new port groups were named (VLAN name)-Promiscuous and (VLAN name)-Isolated according to which PVLAN was used.

Basic testing of the implementation was done with two VM test hosts by sending a ping request (ICMP) first to the VLANs gateway address and then to the other host in the same VLAN. If the hosts were in promiscuous PVLAN, all requests should have succeeded. If the hosts were in isolated PVLAN, the gateway request should have succeeded and the request to the other hosts should have failed. The tests succeeded without problems.

6.5 vSphere Virtual SAN

Network connectivity was confirmed working, so vSAN was to be implemented next. At the start of implementing vSAN, an evaluation license was used for a short while. The license was used because the actual license key was not yet acquired at the time of the implementation. When the license key was later acquired, it was added to the system in the same manner as described in chapter 6.3.5. Switching from evaluation license to the actual license did not cause any interruptions to the system.

6.5.1 Prerequisites

Before enabling any vSAN related functions, the flash disks of all hosts were to be marked as flash disks to enable using the disks as cache in the vSAN disk groups. By default, all the disks were marked as hard disks. Changing the type of flash disks to flash was done in vSphere Web Client by navigating to a host > *Manage*-tab > *Storage*-subtab > *Storage devices*, then selecting all the flash disks and clicking the *F*-button above the list of disks. This procedure was repeated for all hosts until all the flash disks were of correct type. Refer to figure 22 for an example of the storage devices after the changes.

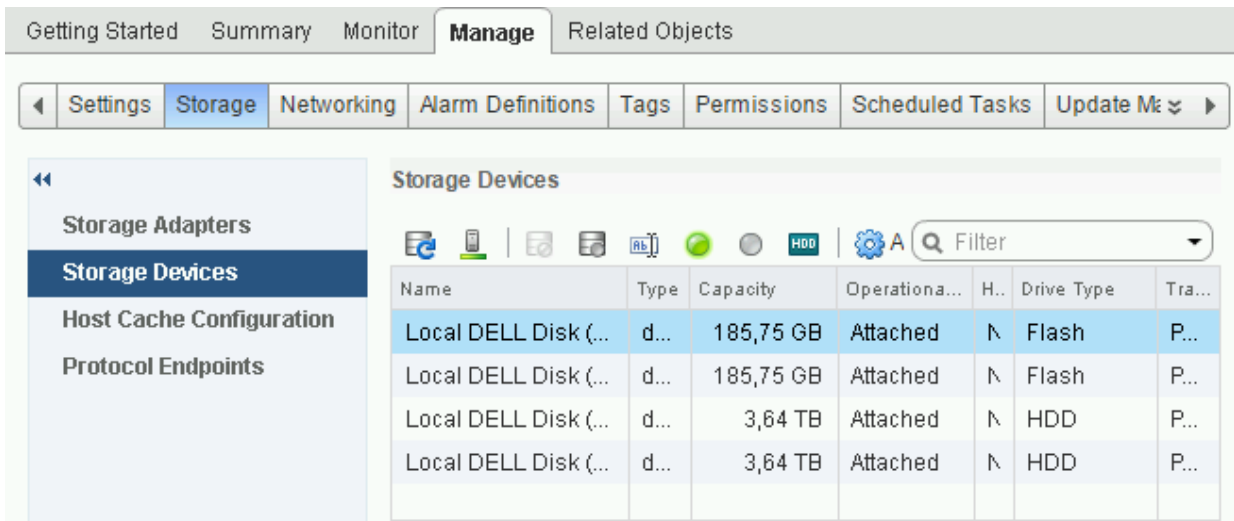


Figure 22. Example of storage devices after marking flash disks. *F*-button is shown as *HDD*-button.

To work, vSAN needed another VMkernel adapter in addition to the default management adapter. In addition to vSAN traffic, the same adapter would be handling vMotion and high availability as well. The adapter was created by navigating to a host > *Manage*-tab > *Networking*-subtab > *VMkernel adapters* > *Add host networking*. For connection type, *VMkernel network adapter* was selected. For target device, the “vSAN” -port group created in chapter 6.4.2 was selected. By using the “vSAN” -port group as the target, the secondary uplink would automatically be utilized for traffic flowing through this VMkernel adapter. In the *Port properties* -screen vSAN, vMotion, provisioning and fault tolerance were selected as enabled services. With this policy, all the storage related traffic would use this VMkernel adapter. Finally, the VMkernel adapter was given an IP address according to the network design seen in chapter 2.1.3. The newly created adapter was automatically given the device ID “vmk1”.

6.5.2 Configuration

With these changes, vSAN could now be enabled in the cluster. The vSAN configuration wizard utility was initiated by navigating to the cluster object > *Manage*-tab > *Virtual SAN – General* > *Configure*. Disk claiming was set to manual to ensure adding the disks in a correct manner. *Deduplication and Compression* was disabled, as it is only supported in flash-only disk groups. *Create fault domains*

was also selected to allow for fault domain configuration. In the *Network validation* -window, all hosts were confirmed to have a correctly configured “vmk1”-adapter. Next, the disks were to be claimed for use by vSAN. The wizard listed all available disks and they were manually marked for use in either cache- or capacity tier. The flash disks were marked as cache and hard disks as capacity. The disks were selected in the list view, then marked by clicking the *Flash*-button for cache tier or the *Disk*-button for capacity tier. Refer to figure 23 for the list of disks after claiming.

Claim disks

Select disks to contribute to the Virtual SAN datastore.

Select which disks should be claimed for cache and which for capacity in the VSAN cluster. The disks below are grouped by model and recommended selection has been made based on the available devices in your environment.

The number of capacity disks must be greater than or equal to the number of cache disks claimed per host.

Group by: Disk model/size Q Filter

Disk Model/Serial Number	Claim For	Drive Type	Total Capacity	Disk Distribution/Host
▶ DELL PERC H710P, 185.75 GB disks	Cache tier	Flash	743.00 GB	1 disk on 4 hosts
▶ DELL PERC H710P, 3.64 TB disks	Capacity tier	HDD	29.09 TB	2 disks on 4 hosts

Total cache: 743.00 GB **Total capacity:** 29.09 TB

Configuration validation:

✓
Configuration correct.

Figure 23. Disks claimed for vSAN

The fault domains for vSAN were created so a single host was in its own fault domain. The fault domains were created by clicking the *Add*-button and named “FD-ESXn” where n is the identification number of the host (1-4). After creating the fault domains, the hosts were assigned into their respective fault domains. The result can be seen in figure 24. The configuration wizard was then completed by clicking *Finish* in the summary window. Refer to figure 25 for the resulting vSAN disk groups.

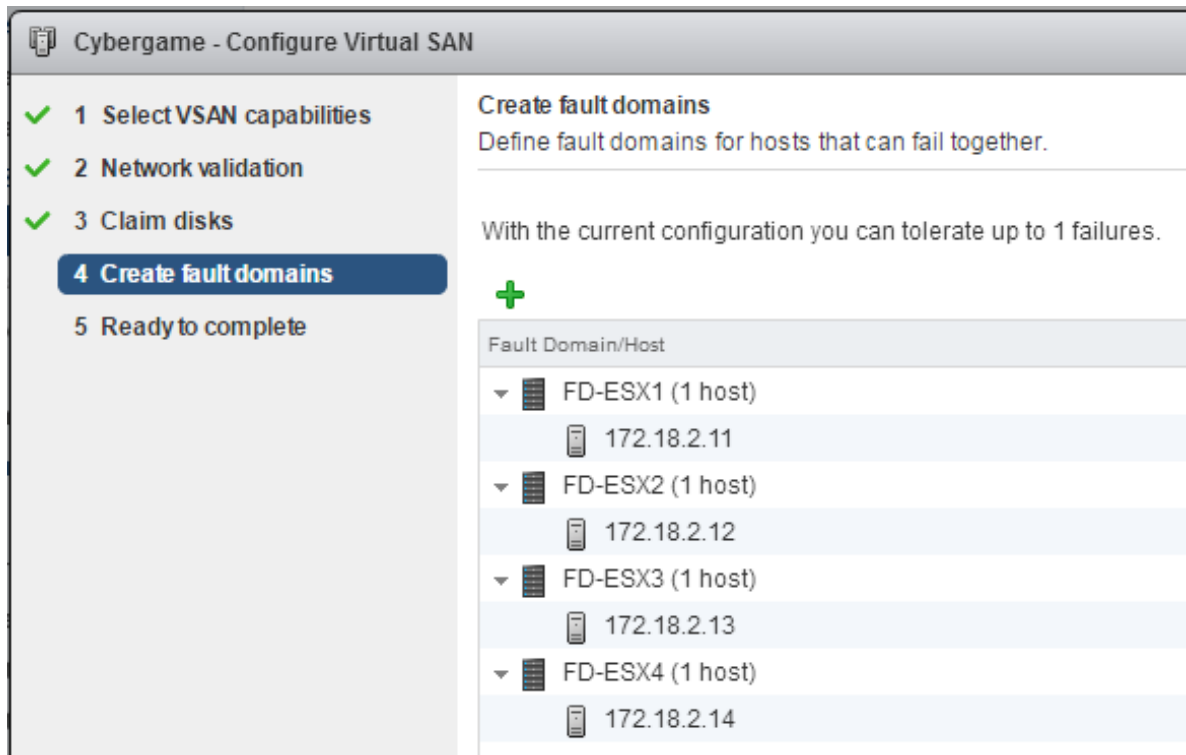


Figure 24. Assigned fault domains

Disk Group	Disks in Use	State	Virtual SAN ...	Type	Fault Domain
172.18.2.11	3 of 3	Connected	Healthy		FD-ESX1
Disk group (02000000006b083fe0cfb49d001ca591ad06f3...	3	Mounted	Healthy	Hybrid	
172.18.2.12	3 of 3	Connected	Healthy		FD-ESX2
Disk group (02000000006b083fe0cfb9ad001ca830c20702...	3	Mounted	Healthy	Hybrid	
172.18.2.13	3 of 3	Connected	Healthy		FD-ESX3
Disk group (02000000006b083fe0cfb272001ca83467047...	3	Mounted	Healthy	Hybrid	
172.18.2.14	3 of 3	Connected	Healthy		FD-ESX4
Disk group (02000000006b083fe0cfb27f001cafc78c05044...	3	Mounted	Healthy	Hybrid	

Disk group (02000000006b083fe0cfb49d001ca591ad06f37cf3504552432048): Disks					
Name	Drive Type	Disk Tier	Capacity	Virtual SAN Health Status	State
Local DELL Disk (naa.6b083fe0cfb49d001ca591ad06f37cf3)	Flash	Cache	185.75 GB	Healthy	Mounted
Local DELL Disk (naa.6b083fe0cfb49d001ca591bf0808ef25)	HDD	Capacity	3.64 TB	Healthy	Mounted
Local DELL Disk (naa.6b083fe0cfb49d001ca591d6096bcc66)	HDD	Capacity	3.64 TB	Healthy	Mounted

Figure 25. vSAN disk groups

Later, it was discovered that according to VMware's best practices documentation fault domains with only single hosts are not required. Instead, single hosts without fault domains are handled in the same manner as with a setup of one

host per fault domain. (VMware 2017m.) The existing setup was left as is, because both solutions have the same result and therefore should not cause problems either way.

After the configuration wizard utility was finished, the vSAN was ready for use. A new datastore with the name “vsanDatastore” was created automatically. A default storage policy was also automatically enabled in the datastore. The default storage policy was deemed suitable for the environment, so no policy adjustments were made. With the policy, one host failure is tolerated (one VM replica is created) and the disk space reserved by VMs is claimed only after it is really used. Refer to figure 26 for details of the datastore and the policy.



Rule-Set 1: VSAN		Storage Consumption Model				
Number of failures to tolerate		1	A virtual disk with size 100 GB would consume:			
Number of disk stripes per object		1				
Force provisioning		No	Storage space 200,00 GB			
Object space reservation (%)		0	Initially reserved storage space 0,00 B			
Flash read cache reservation (%)		0,0000	Reserved flash space 0,00 B			
Storage Compatibility		Total Capacity		Virtual SAN Capacity		
Compatible		28,80 TB		28,80 TB		
Name		Datacenter		Type	Free Space	Capacity
 vsanDatastore		 Cybergame		vsan	27,33 TB	28,80 TB

Figure 26. Summary of the storage policy and the vSAN datastore

6.5.3 Testing & monitoring

There are multiple methods of monitoring vSAN performance and health. For background monitoring, the vSAN performance monitor was enabled by navigating to the cluster object > *Manage*-tab > *Settings*-subtab > *Virtual SAN – Health and Performance* and clicking *Turn on* in the performance service settings. Some performance statistics were now being collected in the background and could be observed by navigating to the cluster’s *Monitor*-tab > *Performance*-

subtab and selecting either of the vSAN related monitor views. Refer to figure 27 for an example of statistic graphs.

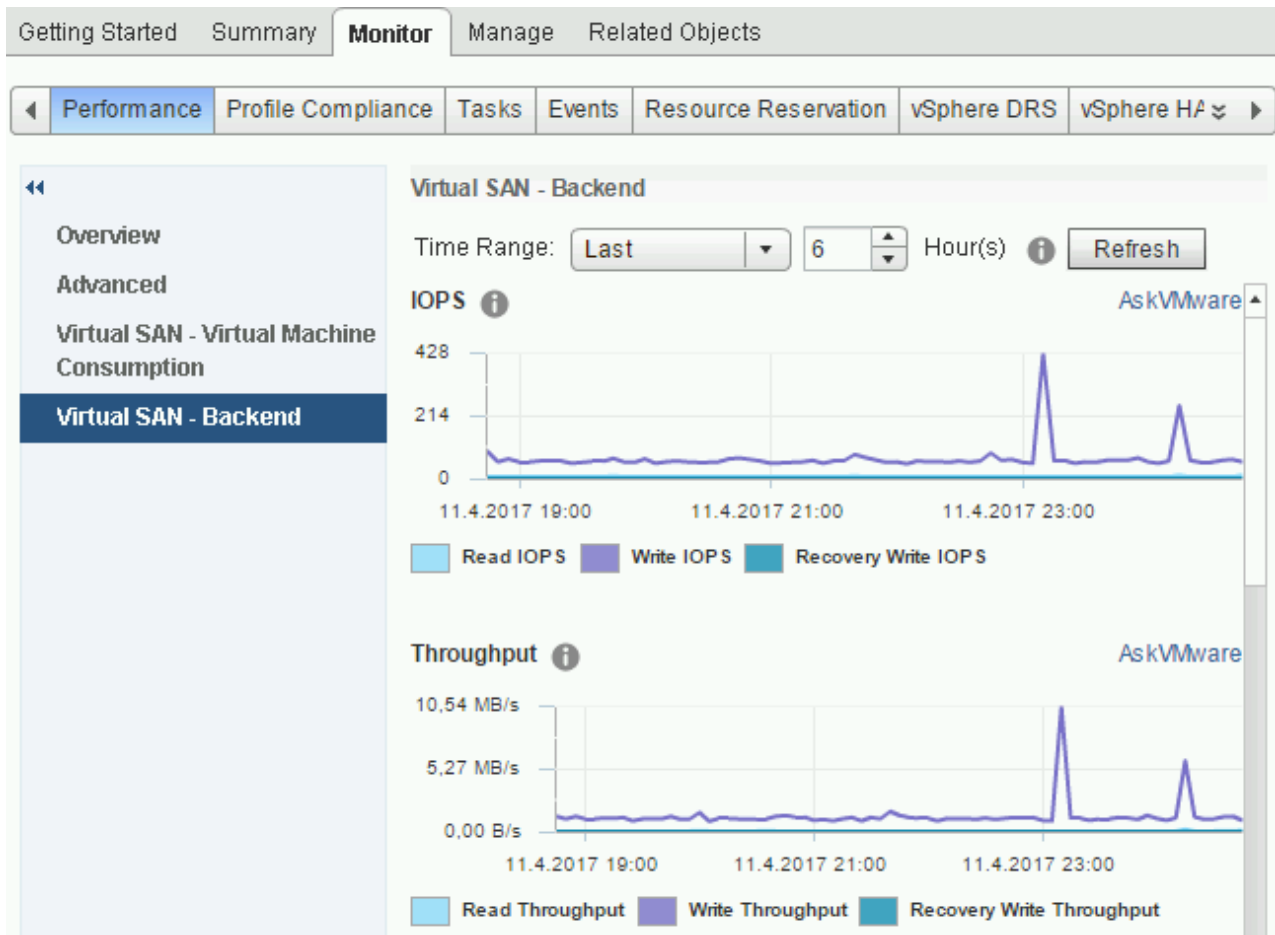


Figure 27. An example of graphs available in vSAN performance monitoring

Still inside the cluster's *Monitor*-tab, the *Virtual SAN*-subtab offered more health and monitoring related tools. By selecting *Health*, an overview all health-related tests could be observed. No warnings or failures were found, apart from an out-dated hardware compatibility list warning.

By selecting *Proactive tests*, a VM creation test, multicast performance test or storage performance tests could be done. The VM creation test was used for testing the actual vSAN storage functionality itself, which would indicate whether the system could be used with actual VMs. When the creation test was ran, a test VM was automatically created and then deleted on all hosts. As there were no problems, the test passed and the vSAN datastore was deemed working.

There were multiple test scenarios available for storage performance testing. At this phase, a low workload stress test was carried out to get a general figure of performance. The test was ran for 15 minutes on all hosts. Refer to figure 28 for the output.

Host	IOPS	Through-put (MB/s)	Avg latency (ms)	Max latency (ms)
ESX1	4253	16.61	0.23	19.43
ESX2	4497	17.57	0.22	10.88
ESX3	3036	11.86	0.33	18.36
ESX4	4211	16.45	0.23	17.91

Figure 28. Output for a low stress test in the vSAN

The multicast performance test was the only test causing problems. The test was done multiple times, but it never passed properly. The test results were reporting lower received bandwidth than expected, but despite this, no actual problems could be observed in the system.

6.6 Additional cluster configuration

As the vSAN datastore was working as intended, additional cluster features were to be implemented. The features could have been enabled earlier on as well, but were postponed as it could have caused unnecessary complexities before the vSAN datastore was usable. For example, high availability could not have migrated the VMs without a datastore shared with all the hosts. High availability, distributed resource scheduler and enhanced vMotion compatibility were to be enabled in the cluster.

First, vSphere HA was enabled in the cluster's settings by navigating to *vSphere HA > Edit* and checking the *Turn on vSphere HA* checkbox. In addition, some adjustments were made to the default HA configuration. *Host monitoring* and *Protect against Storage Connectivity Loss* were enabled to ensure HA in case of host-, network- or vSAN failure. Host isolation response was set to *Shut down and restart VMs* to prevent more than one instance of a VM from running at the

same time in case of failure. VM monitoring sensitivity was set to *high* to ensure fast response to failure. In *Admission control*, failover capacity was set to one host to ensure there are enough free resources to failover all VMs in the cluster. Other settings were left untouched. The HA functionality was tested later in a failover test documented in chapter 6.7.

Implementing vSphere DRS did not require any additional configuration apart from enabling the feature itself. DRS was enabled in the cluster's settings by navigating to *vSphere DRS > Edit >* and checking the *Turn on vSphere DRS* checkbox. By default, *DRS Automation* was set to *Fully automated*, which was the desired operational mode. With this configuration, DRS would automatically do load balancing if there are good improvements available by doing so. Monitoring DRS actions can be done by navigating to the cluster's *Monitor-tab > vSphere DRS* and selecting any of the sidebar items. Refer to figure 29 for examples of CPU and memory balance.

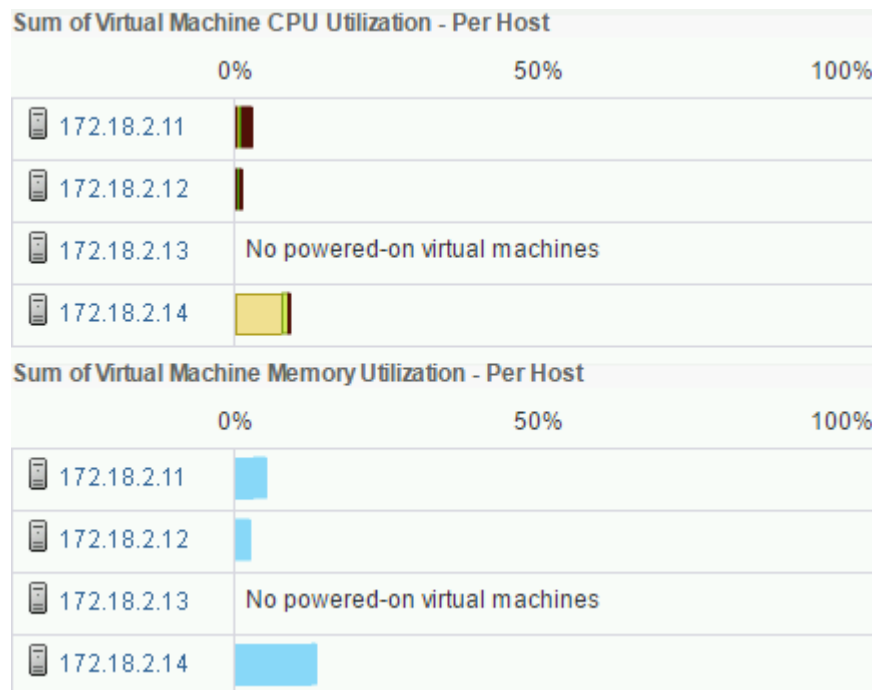


Figure 29. Examples of CPU and memory utilization automatically balanced by DRS.

Finally, EVC was enabled in the cluster's settings by navigating to *VMware EVC > Edit* and selecting *Enable EVC for Intel hosts*. For EVC mode, *Intel Ivy Bridge*

generation was selected. With EVC enabled, compatibility between hosts is enforced when using vMotion migration. With the current hosts having an identical physical configuration, EVC was enabled just in case for future changes. To confirm vMotion was working properly with EVC enabled, a powered-on test VM was migrated from a host to another. While migrating, the VM was sending a ping request to its gateway address every second. The migration took approximately 28 seconds, the VM lost no ping packets in the process and was migrated successfully to another host.

6.7 Failover test

When the environment had the functionality needed for production use (vSAN, redundant vDS networking), the ability to recover from failures was tested. Hypothetically, the environment should continue operating normally even if one host goes down. The testing was carried out in four different scenarios:

1. Primary uplink (nic0) fails
2. Secondary uplink (nic1) fails
3. Both uplinks fail simultaneously
4. Both uplinks and management link fail (simulated host failure)

The link failures were simulated by shutting down the corresponding port in the switch they were connected to, causing instantaneous connectivity loss. A VM in production was also used for measuring availability loss to a service in case of failover. The VM in question was a syslog server with a web interface. During the tests, continuous ping requests (ICMP) were being sent to the VM's IP address in one second intervals. The web interface was accessed manually multiple times before, during and after the failover.

When the primary uplink failed during the first test, the traffic was instantaneously routed through the secondary uplink instead. There was no packet loss observed and the VMs web interface could be reached normally. An alarm event for uplink redundancy loss was raised by vCenter and the failed link was visible in the topology monitor. When the connectivity was restored to its original state, brief downtime was observed. One ping request was lost, and the web interface was

unavailable for approximately 6 seconds, so the effect on availability was negligible. Refer to figure 30 for warnings by vCenter.

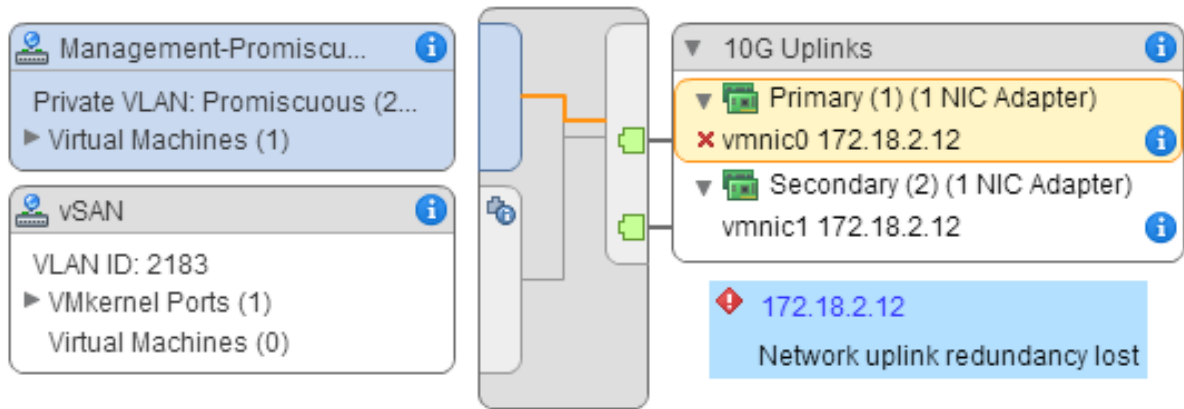


Figure 30. Failed link visible on vDS topology with alert for redundancy loss

The second test produced no observable downtime whatsoever. No packet loss was observed and the VM's web interface remained available. The same alarm as with the first test for uplink redundancy loss was raised by vCenter. Restoring connectivity had no observable effects either.

The last two tests had the most impact on the system. In the third test, for the VM to become usable again, the failover took approximately a minute and 48 seconds. When returning to the original state after regaining connectivity, the VM was down for approximately one minute. For more detailed log of events during the failover, refer to figure 31. Other effects, such as alarms can be seen in figures 32 and 33. The fourth test produced the same results as the third, apart from failover taking approximately 7 seconds longer.

Time	Event
0:00	Both uplinks fail
0:08	HA availability state changed to unreachable
0:09	Host 4 notifies some nodes in vSAN are not reachable anymore
0:13	HA detects a possible host failure, HA availability state changed to host failed
0:15	Hosts 1 and 3 notifies some nodes in vSAN are not reachable anymore

1:13	HA initiates a failover action on the VM (shutdown and restart VMs)
1:24	HA failover migration completes
1:33	The VM is restarted by HA failover action
1:48	The VM responds to ping requests and the web interface is reachable (failover completed)

Figure 31. Log of events during failover test 3, starting from simulated link failure on host 2

Issue	Type	Trigger Time	Status
Host cannot communicate with all other nodes in the Virtual SAN enabled cluster	Configuration Issue	6/2/2016 1:09:51 PM	
vSphere HA host status	Triggered Alarm	6/2/2016 1:09:40 PM	Alert
vSphere HA detected a possible host failure of this host	Configuration Issue	6/2/2016 1:09:40 PM	
Network connectivity lost	Triggered Alarm	6/2/2016 1:09:31 PM	Alert
Network uplink redundancy lost	Triggered Alarm	6/2/2016 1:09:31 PM	Alert

Name	Operational State	VM Storage Policy	Compliance Status
CG-SYSLOG	Healthy		
VM Home	Healthy	Virtual SAN Default Storage Policy	Noncompliant
Hard disk 1	Healthy	Virtual SAN Default Storage Policy	Noncompliant

Figure 32. Alarms as seen on the host with failed uplinks

Capacity Overview	
0 TB	7.20 TB
Used - Physically written	54.27 GB
Used - VM overreserved	33.00 GB
Used - Total	87.27 GB
Virtual SAN system overhead	36.70 GB
Free	7.08 TB

Figure 33. vSAN capacity reduced during connectivity loss (normally ~28 TB)

7 CONCLUSIONS

Overall, the thesis work provided new information and improved upon old knowledge regarding the covered topics. As vSphere products were a core part of the implementation, they became quite familiar in the process. Software-defined everything is a popular subject already, so any knowledge of software-de-

finned data centers will most likely prove valuable in the future. In addition, the importance of good documentation became clear when producing the report of this thesis. All implemented features were tested at least to some extent, but some of the tests could have been covered more thoroughly.

Implementing the virtualization platform has answered all desired needs regarding virtualization with the Cybergame project. Comparing the state of the new and the obsolete installation, the new is more feature-rich than the previous one. The new environment improves on the previous by adding features such as better fail-over capabilities and PVLANS. Implementation of features like these could benefit any virtualized data center installation.

The goals of the thesis were successfully met in the given timeframe. The virtualization environment with all the desired features was implemented and is already in production use as a part of the Cybergame infrastructure. Problems encountered during the implementation were minor and most of them were resolved. Some non-critical issues remain in the system, but they should not affect day-to-day operation.

As there are still minor issues present, the issues could be carried over to future project ideas. For example, the VUM service on the Windows Server VM stops responding to vCenter randomly from time to time, for which a fix could not be found. The issue could be worked around by restarting the service manually. An improved way of updating the vCenter appliance inside the Cybergame network could be researched. vSAN performance could potentially be increased by tuning the storage policy, research if it is possible in practice. Finally, more thorough benchmarking or testing of some aspects of the environment such as vSAN performance.

REFERENCES

- Bittman, T. J., Dawson, P. & Warrilow, M. 2016. Magic Quadrant for x86 Server Virtualization Infrastructure. Available at: <https://www.gartner.com/doc/reprints?id=1-3E2WESI&ct=160804&st=sb> [Accessed: 5 May 2017].
- Fenech, J. 2016. vCenter Server for Windows and vCSA compared. Available at: <http://www.altaro.com/vmware/vcenter-server-windows-vcsa-compared/> [Accessed: 6 May 2017].
- Foschiano, M. & HomChaudhuri, S. 2010. RFC 5517: Cisco Systems' Private VLANs: Scalable Security in a Multi-Client Environment. Available at: <https://tools.ietf.org/pdf/rfc5517.pdf> [Accessed: 7 May 2017].
- Kaakkois-Suomen Ammattikorkeakoulu Oy s.a. Kyberturvallisuus ja liiketoiminta-osaamisen kehittäminen. Available at: <https://www.xamk.fi/tutkimus-ja-kehitys/kyberturvallisuuden-kehittaminen-keskiossa/> [Accessed: 12 May 2017].
- Lumivirta, S. 2015. Kyberturvallisuuslaboratorion aktiivilaitteet. Bachelor's thesis. Available at: http://www.theseus.fi/bitstream/handle/10024/90819/Lumivirta_Suvi.pdf?sequence=1 [Accessed: 7 May 2017].
- Portnoy, M. 2012. Virtualization Essentials. Indianapolis: John Wiley & Sons, Inc.
- Prabhudev, A. 2016. What's New in vSphere 6.5: vCenter management clients. Available at: <https://blogs.vmware.com/vsphere/2016/12/new-vcenter-management-clients-vsphere-6-5.html> [Accessed: 6 May 2017].
- Työ- ja elinkeinoministeriö s.a. RR-tietopalvelu -hankekuvaus A70554, Kyberturvallisuusosaamisen ja liiketoiminnan kehittäminen. Available at: <https://www.eura2014.fi/rrtiepa/projekti.php?projekтикoodi=A70554> [Accessed: 3 May 2017].
- VMware, Inc. 2007. vMotion Datasheet. Available at: https://www.vmware.com/pdf/vmotion_datasheet.pdf [Accessed: 6 May 2017].
- VMware, Inc. 2015a. vSphere 6.0 Datasheet. Available at: <http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vsphere/vmw-vsphr-datasheet-6-0.pdf> [Accessed: 5 May 2017].

VMware, Inc. 2015b. vCenter Server Datasheet. Available at:

<http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vCenter/vmware-vcenter-server-datasheet.pdf> [Accessed: 5 May 2017].

VMware, Inc. 2017a. About Us. Available at: <http://www.vmware.com/company.html> [Accessed: 4 May 2017].

VMware, Inc. 2017b. Products. Available at: <http://www.vmware.com/products.html> [Accessed: 4 May 2017].

VMware, Inc. 2017c. ESXi. Available at: <https://www.vmware.com/products/esxi-and-esx.html> [Accessed: 5 May 2017].

VMware, Inc. 2017d. vCenter Server Appliance Overview. Available at: <https://pubs.vmware.com/vsphere-60/topic/com.vmware.vsphere.vcsa.doc/GUID-223C2821-BD98-4C7A-936B-7DBE96291BA4.html> [Accessed: 6 May 2017].

VMware, Inc. 2017e. vSphere Distributed Switch Architecture. Available at: <https://pubs.vmware.com/vsphere-60/topic/com.vmware.vsphere.networking.doc/GUID-B15C6A13-797E-4BCB-B9D9-5CBC5A60C3A6.html> [Accessed: 6 May 2017].

VMware, Inc. 2017f. Overview of vNetwork Distributed Switch concepts. Available at: <https://kb.vmware.com/kb/1010555> [Accessed: 6 May 2017].

VMware, Inc. 2017g. How vSphere HA Works. Available at: <https://pubs.vmware.com/vsphere-60/topic/com.vmware.vsphere.avail.doc/GUID-33A65FF7-DA22-4DC5-8B18-5A7F97CCA536.html> [Accessed: 6 May 2017].

VMware, Inc. 2017h. Distributed Resource Scheduler, Distributed Power Management. Available at: <http://www.vmware.com/products/vsphere/drs-dpm.html> [Accessed: 6 May 2017].

VMware, Inc. 2017i. About Enhanced vMotion Compatibility. Available at: <https://pubs.vmware.com/vsphere-60/topic/com.vmware.vsphere.vcenter-host.doc/GUID-9F444D9B-44A0-4967-8C07-693C6B40278A.html> [Accessed: 6 May 2017].

VMware, Inc. 2017j. Introduction to Virtual SAN. Available at:

<https://pubs.vmware.com/vsphere-60/topic/com.vmware.vsphere.virtualsan.doc/GUID-18F531E9-FF08-49F5-9879-8E46583D4C70.html> [Accessed: 7 May 2017].

VMware, Inc. 2017k. Virtual SAN Concepts. Available at:

<https://pubs.vmware.com/vsphere-60/topic/com.vmware.vsphere.virtualsan.doc/GUID-ACC10393-47F6-4C5A-85FC-88051C1806A0.html> [Accessed: 7 May 2017].

VMware, Inc. 2017l. vSAN 6.6 Datasheet. Available at:

<http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vsan/vmware-virtual-san-datasheet.pdf> [Accessed: 7 May 2017].

VMware, Inc. 2017m. Managing Fault Domains in Virtual SAN Clusters. Available

at: <https://pubs.vmware.com/vsphere-60/topic/com.vmware.vsphere.virtualsan.doc/GUID-8491C4B0-6F94-4023-8C7A-FD7B40D0368D.html> [Accessed: 21 April 2017].

VMware, Inc. 2017n. Network ports required to access vCenter Server, ESXi, and ESX hosts. Available at: <https://kb.vmware.com/kb/1012382> [Accessed: 8 May 2017].

LIST OF FIGURES

Figure 1. A visualization of the security levels of different security zones

Figure 2. The physical topology

Figure 3. Details of the network functions VLANs and subnets

Figure 4. IP addressing of the virtualization-related components

Figure 5. A simplified comparison between the 2 hypervisor types. Portnoy, M. 2012. Virtualization Essentials. Indianapolis: John Wiley & Sons, Inc.

Figure 6. An example vDS setup with teamed NICs. VMware, Inc. 2017. vSphere Distributed Switch Architecture. Available at: <https://pubs.vmware.com/vsphere-60/topic/com.vmware.vsphere.networking.doc/GUID-B15C6A13-797E-4BCB-B9D9-5CBC5A60C3A6.html> [Accessed: 6 May 2017].

Figure 7. A visualization of vSAN operation. VMware, Inc. 2017. vSAN 6.6 Datasheet. Available at: <http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vsan/vmware-virtual-san-datasheet.pdf> [Accessed: 7 May 2017].

Figure 8. A summary of possible communication privileges between port types. Foschiano, M. & HomChaudhuri, S. 2010. RFC 5517: Cisco Systems' Private VLANs: Scalable Security in a Multi-Client Environment. Available at: <https://tools.ietf.org/pdf/rfc5517.pdf> [Accessed: 7 May 2017].

Figure 9. A summary of the new datastore

Figure 10. A summary of the vCenter deployment

Figure 11. Management interface detected the available update from the disk image

Figure 12. Summary of the new identity source configuration

Figure 13. Adding new licenses

Figure 14. Licensed assets

Figure 15. Attaching baselines to clustered hosts

Figure 16. Summary of remediation wizard utility

Figure 17. An example of host NICs assigned to uplinks

Figure 18. General port group settings

Figure 19. Port group policy setup for most VLANs

Figure 20. An example of the topology view of some port groups with uplinks visible

Figure 21. PVLAN configuration window

Figure 22. Example of storage devices after marking flash disks. F-button is shown as HDD-button.

Figure 23. Disks claimed for vSAN

Figure 24. Assigned fault domains

Figure 25. vSAN disk groups

Figure 26. Summary of the storage policy and the vSAN datastore

Figure 27. An example of graphs available in vSAN performance monitoring

Figure 28. Output for a low stress test in the vSAN

Figure 29. Examples of CPU and memory utilization automatically balanced by DRS.

Figure 30. Failed link visible on vDS topology with alert for redundancy loss

Figure 31. Log of events during failover test 3, starting from simulated link failure on host 2

Figure 32. Alarms as seen on the host with failed uplinks

Figure 33. vSAN capacity reduced during connectivity loss (normally ~28 TB)