samk

Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

TUUKKA MANSIKKA

Suunnitelma virtuaalikoneiden migraatiosta ESXi:stä Proxmox VE:hen

TIETOJENKÄSITTELYN KOULUTUSOHJELMA
2020

Tekijä(t)
Mansikka, Tuukka

Julkaisun laji
Opinnäytetyö, AMK

Päivämäärä
12 2020

Sivumäärä
21

Julkaisun kieli
Suomi

Julkaisun nimi
Suunnitelma virtuaalikoneiden migraatiosta ESXi:stä Proxmox VE:hen

Tutkinto-ohjelma
Tietojenkäsittely

Tiivistelmä
Tämän opinnäytetyön aiheena on virtuaalikoneiden migraation testaus ja suunnittelu ESXi:stä Proxmox VE:hen taloushallintopalveluihin erikoistuvassa yrityksessä. Yrityksessä pohdittiin virtualisointiympäristön vaihtoa. Yritys halusi selvittää etukäteen, miten virtuaalikoneiden siirto käytännössä tapahtuisi ja miten sen voisi toteuttaa.

Tässä opinnäytetyössä käsitellään virtualisoinnin periaatteita sekä esitellään Proxmox VE yleisellä tasolla. Opinnäytetyössä tarkasteltiin virtuaalikoneiden migraatiota teoriassa, tehtiin käytännön testi testiympäristössä sekä selostettiin suunnitelma migraatiolle. Testimigraatio tehtiin opinnäytetyön tekijän koneella virtualisoimalla ESXi ja Proxmox VE. ESXi:hin luotiin kaksi virtuaalikonetta, josta ne siirrettiin Proxmox VE:hen.

Asiasanat
migraatio, esxi, proxmox, virtuaalikone, virtualisointi, vmware

Author(s) Mansikka, Tuukka	Type of Publication Bachelor's thesis	Date 12 2020
	Number of pages 21	Language of publication: Finnish

Title of publication
Plan for migrating virtual machine from ESXi to Proxmox VE

Degree program
Degree Programme in Business Information Systems

Abstract

The purpose of this thesis was to study and plan the migration of virtual machines from ESXi to Proxmox VE in a company specializing at financial management. The company was planning to change the virtualization environment. They wanted to investigate how to migrate virtual machine from one environment to another and how to implement the migration.

This thesis addresses the basics of virtualization and introduces Proxmox VE in general. In this thesis the migration of virtual machines is examined, migration is demonstrated in test environment and the migration plan is explained. Test migration was conducted on authors PC by virtualizing ESXi and Proxmox VE. Two virtual machines were created in ESXi from where they were migrated to Proxmox VE.

Key words
migration, esxi, proxmox, virtual machine, virtualisation, vmware

SISÄLLYS

1 JOHDANTO 5
2 VIRTUAALIKONE 5
2.1 Mikä on virtuaalikone? 5
2.2 Hypervisor 6
2.3 Virtuaalikone vs konttiteknologia 7
3 PROXMOX VE 8
3.1 Yleistä 8
3.2 Proxmox VE:n asennus 9
3.3 VMware ESXi vs Proxmox VE 14
4 VIRTUAALIKONEEN MIGRAATIO ESXI:STÄ PROXMOX VE:HEN 15
4.1 Taustatietoa 15
4.2 Virtuaalikoneen migraatio 16
4.3 Suunnitelma migraatiolle yrityksessä 18
5 LOPUKSI 19
5.1 Ongelmat ja haasteet 19
5.2 Omaa pohdintaa 21

1 JOHDANTO

Virtuaalikoneen käsite on ollut minulle tuttu jo useamman vuoden. Opiskellessani tietojenkäsittelyä sain tutustua virtuaalikoneisiin tarkemmin. Silloin kiinnostukseni virtualisointiin heräsi. Minua alkoi kiehtoa ajatus, että voin asentaa minkä tahansa käyttöjärjestelmän käyttäen vain yhtä fyysistä laitteistoa. Sain aiheen opinnäytetyölleni työpaikaltani, jossa pohdittiin mahdollisuutta ESXi:n korvaamista Proxmox virtualisointiympäristöllä.

Tämän opinnäytetyön toisessa luvussa käsitellään virtuaalikonetta. Aluksi tarkastellaan, mikä on virtuaalikone ja hypervisor sekä tehdään lyhyt vertailu virtualisoinnin ja konttiteknologian eroista. Kolmannessa luvussa tutustumme Proxmox VE:hen yleisesti, sen asentamiseen ja vertaillaan sitä kilpailevaan tuotteeseen ESXi:hin. Neljäs luku käsittelee tämän opinnäytetyön pääaihetta: virtuaalikoneiden siirtämistä ESXi:stä Proxmox VE:hen. Selostan tapauksen taustatietoja ja käyn läpi virtuaalikoneen migraatiota. Viidennessä luvussa kerron työssä ilmenneitä ongelmia ja haasteita sekä esitän omaa pohdintaani tehdystä työstä

2 VIRTUAALIKONE

2.1 Mikä on virtuaalikone?

Virtuaalikone on ohjelmallisesti toteutettu näköiskone, joka ei ole suoraan yhteydessä varsinaiseen laitteistoon. Virtuaalikoneiden luomiseen ja hallintaan käytetään hypervisoria. Virtuaalikoneet jakavat isäntäjärjestelmän resursseja eli suorittimen, muistin, tallennustilan sekä muut isäntäjärjestelmään kytketyt laitteet. Tämä mahdollistaa sen, että yhdellä fyysisellä laitteistolla voidaan luoda ja käyttää useampia virtuaalikonetta.

Virtuaalikoneista on useita hyötyjä. Niillä voidaan luoda omia ympäristöjä tiettyä tarkoitusta varten. Virtuaalikoneet toimivat kuten tavalliset koneet ja niille voidaan käytännössä asentaa mikä tahansa käyttöjärjestelmä. Lisäksi virtuaalikoneet pystytään eristämään muista ympäristöistä, mikä kasvattaa tietoturvaa. Esimerkiksi yhdellä virtuaalikoneella voi olla asennettuna Windows-käyttöjärjestelmä työtehtäviä varten, kun taas toiselle virtuaalikoneelle on asennettuna Linux-palvelin palvelintoimintoja varten. Kolmannella virtuaalikoneella on asennettuna Windows-palvelin varmuuskopiointia varten. Nämä kaikki virtuaalikoneet voidaan luoda keskitetysti yhdelle fyysiselle koneelle ilman, että jokaiselle ympäristölle tarvitsi hankkia oma laitteensa.

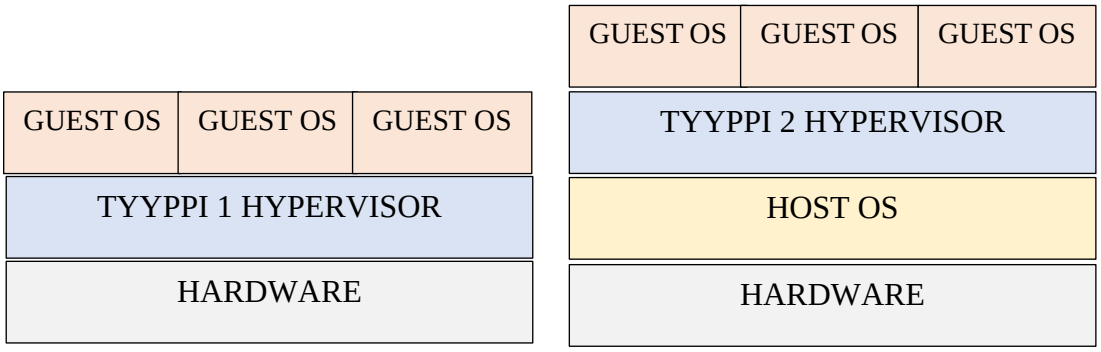
Virtuaalikoneilla voidaan kasvattaa myös toimintavarmuutta sekä vikasietoisuutta. Mikäli virtuaalikone kaatuu tai menettää toimintakykynsä, se on helppo palauttaa snapshotista takaisin toimivaan tilaan. Laitteiston hajotessa virtuaalikone voidaan siirtää uudelle laitteistolle ja palauttaa takaisin toimintaansa.

2.2 Hypervisor

Hypervisorilla tarkoitetaan ohjelmaa, jolla hallitaan virtuaalikoneita. Hypervisor vastaa resurssien jakamisen guest-koneille toimien välikätenä laitteiston tai käyttöjärjestelmän ja virtuaalikoneiden välillä. Hypervisorit voidaan jakaa kahteen tyyppiin:

- *Tyyppi 1 (type 1)*: tyypin 1 hypervisoreita kutsutaan myös *bare metal hypervisoriksi* ja *native hypervisoriksi*. Tyypin 1 hypervisor voidaan asentaa suoraan laitteelle. Tämä takaa paremman toimivuuden ja tietoturvan, koska hypervisor ei altistu toisen käyttöjärjestelmän haavoittuvuuksiin. Tyypin 1 hypervisoreita käytetään enemmän yritysmaailmassa. Niiden hyödyt ovat korkeampi suorituskyky tyypin 2 hypervisoreihin verrattuna, vähäisempi resurssien käyttö ja parempi hallittavuus. (VMWaren www-sivut 2020, Taylor 2020). Esimerkkejä tyypin 1:n hypervisoreista ovat VMwaren ESXi, KVM ja Proxmox VE.
- *Tyyppi 2 (type 2)*: kutsutaan myös *hosted hypervisor*. Tyypin 2 hypervisor on sovelluksena toimiva hypervisor, joka vaatii asennuksen käyttöjärjestelmän päälle. Tyypin 2 hypervisorilla ei ole suoraan pääsyä laiteresursseihin, mikä kasvattaa virtuaalikoneen latenssia. Tyypin 2 hyperivisoreita käytetään usein

järjestelmien nopeaan testaamiseen. Tyyppi 2 hypervisoreiden hyödyt ovat helppo pystyttäminen ja hallinta eivätkä ne vaadi yhtä aktiivista järjestelmänvalvontaa kuin tyypin 1 hypervisorit. (VMWaren www-sivut 2020, Taylor 2020). Esimerkkejä tyypin 2 hypervisoreista ovat Virtualbox, VirtualPC ja VMware Workstation.

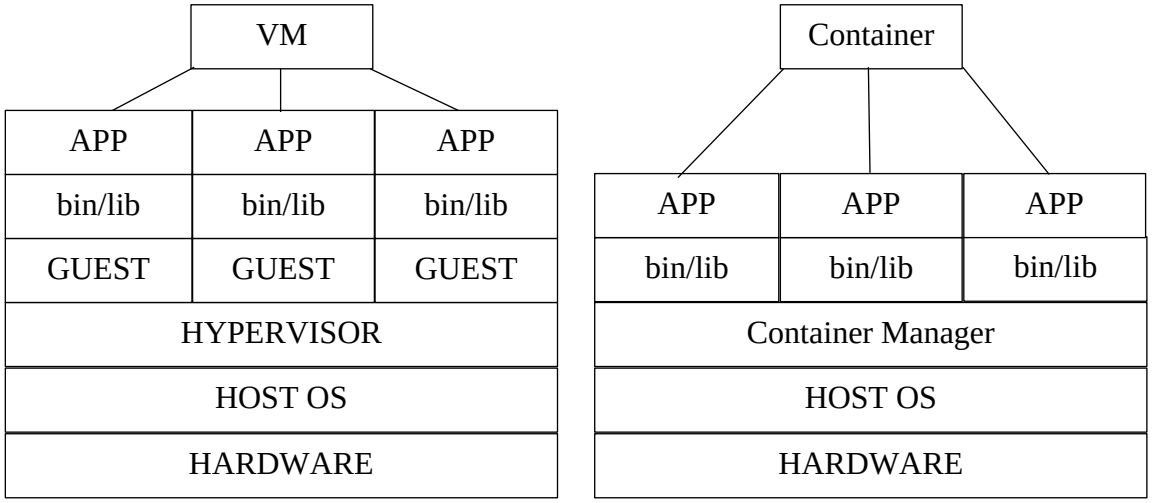


Kuva 1. Tyypin 1 ja 2 hypervisoreiden rakennehierarkia (Taylor 2020)

2.3 Virtuaalikone vs konttiteknologia

Puhuttaessa virtualisoinnista on hyvä mainita siihen pohjautuva konttiteknologia. Koneiden virtualisointi on hyvä tapa hyödyntää laitteiston resursseja sekä eristää ympäristöt keskenään. Virtuaalikoneet saattavat kuitenkin varata enemmän resursseja, kuin mitä tarvitsevat tehtäväänsä varten. Lisäksi kokonaisen virtuaalikoneen luominen yksittäistä tehtävää varten saattaa hidastaa ohjelmien toimintaa. Käyttöjärjestelmät saattavat asentaa ajureita, kirjastoja sekä muita tietoja, jotka eivät ole käytettävän ohjelman kannalta olennaisia. Ne voivat olla jopa haitallisia ohjelman toiminnalle. (Bauer 2018).

Konttiteknologian avulla pyritään eliminoimaan edellä mainitut ongelmat. Kontteja voidaan suorittaa yhdeltä koneelta. Tällä tavalla jokaista ohjelmaa varten ei tarvitse luoda omaa virtuaalikonetta. Kontit hyödyntävät isäntäjärjestelmän kerneliä sekä peruskirjastoa ohjelmien suorittamiseen. Ohjelmaan vaaditut kirjastot ja tiedostot pakataan ”konttiin”, josta sitten konttien orkestroija lukee tarvittavat tiedot. Näin vältytään lataamasta turhia kirjastoja ja ohjelmia yksinkertaisten tehtävien suorittamiseen. Virtuaalikoneiden koot ovat usein suurempia kuin 1Gt, kun taas konttien koot voivat olla alle 100Mt. Tällä tavalla säästytään turhalta resurssien haaskaamiselta. (Bauer 2018).



Kuva 2. Virtuaalikoneen ja konttien rakennehierarkian vertailua (Bauer 2018)

3 PROXMOX VE

3.1 Yleistä

Proxmox VE (virtualization environment) on avoimen lähdekoodin virtualisointiympäristö. Proxmox sai alkunsa vuonna 2005, kun Martin ja Dietmar Mauer perustivat Proxmox Server Solutions GmbH. Heidän ensimmäinen tuotteensa oli Proxmox Mail Gateway, joka tarjosi suojaa roskaposti- ja tietojenkalasteluviesteiltä sekä haittaohjelmilta. Vuonna 2008 Proxmox Server Solutions GmbH julkaisi stabiilin version Proxmox Virtualization Environmentista. Yritys on tämän jälkeen julkaissut säännöllisesti uusia ominaisuuspäivityksiä molempiin tuotteisiinsa. (Proxmox 2020).

Proxmox VE on Unix-pohjainen järjestelmä, joka yhdistää konttiteknologian sekä KVM hypervisorin. Se hyödyntää Linux-kerneliä ja on avoimen lähdekoodinsa vuoksi ilmainen ohjelmisto. Proxmox tarjoaa myös kuukausittaista maksua vastaan teknistä tukipalvelua. Nämä maksulliset palvelut eivät ole välttämättömiä eivätkä ne rajoita Proxmox VE:n käyttöä. Proxmox VE voidaan asentaa suoraan laitteiston päälle. Asennuksen jälkeen Proxmox VE:tä voidaan käyttää selainpohjaisen GUI:n kautta. (Proxmox 2020).

3.2 Proxmox VE:n asennus

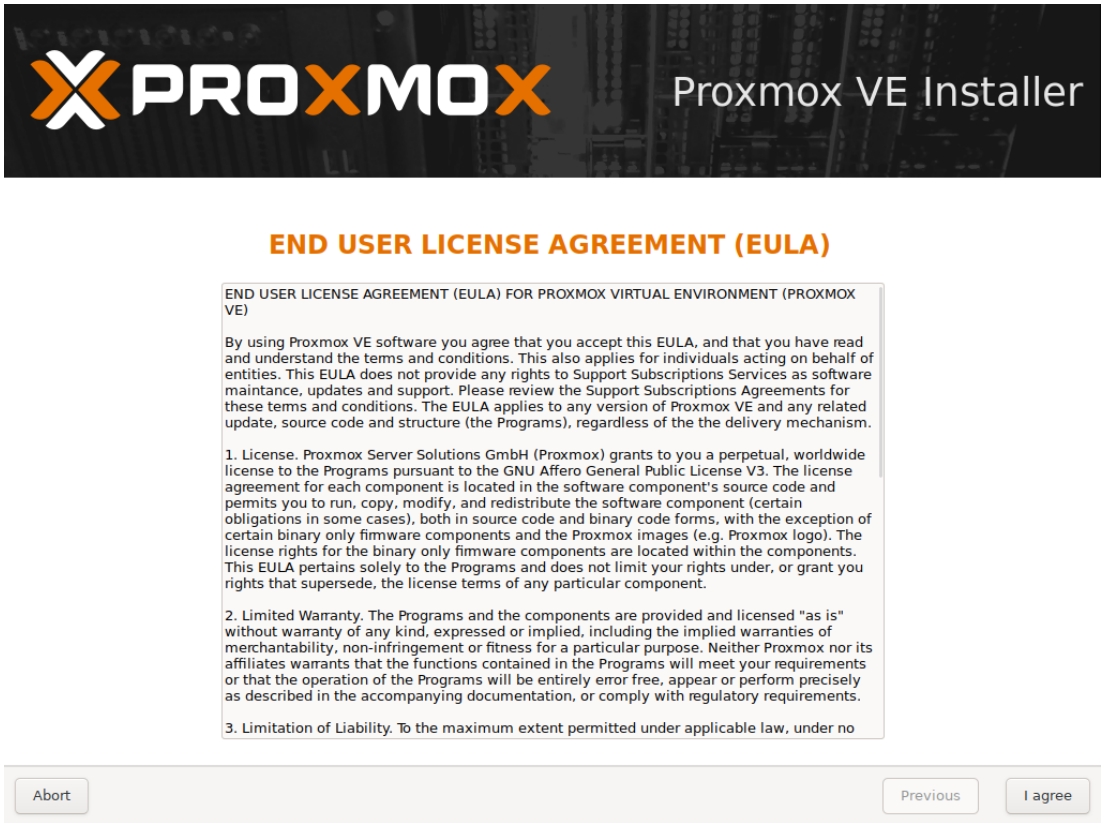
Proxmox VE:n asennus on yksinkertainen. Asennusta varten tarvitaan Proxmox VE:n ISO-tiedosto ja CD-levy tai USB-tikku sekä laite, jolle Proxmox VE asennetaan. Vaihtoehtoisesti Proxmox VE voidaan asentaa sisäistetysti toisen käyttöjärjestelmän päälle. Asennus aloitetaan polttamalla levykuva bootattavalle CD-levylle tai USB-muistitikulle. Tämän jälkeen levy tai muistitikku liitetään koneeseen ja kone käynnistetään. Tämä aloittaa Proxmox VE:n asennusprosessin.

Kuva 3. Proxmox VE:n aloitusnäkymä.

Proxmox VE:n asennuksen aloitusnäkymästä (kuva 3) valitaan ”Install Proxmox VE”. Asennusohjelma purkaa tarvittavat asennustiedostot asennusta varten. Seuraavaksi hyväksytään loppukäyttäjän lisenssisopimus eli EULA (end user license agreement) (kuva 4). Painamalla ”I agree” asennusohjelma etenee seuraavaan vaiheeseen. Seuraavaksi valitaan asema, jolle Proxmox VE halutaan asentaa (kuva 5). Asennusohjelma luo osiot automaattisesti. Tämän jälkeen syötetään maa, valitaan aikavyöhyke ja näppäimistöasettelu (kuva 6). Kun alueasetukset on annettu, määritetään root-käyttäjälle salasana (kuva 7). Salasanan määrittämisessä on hyvä luoda vahva salasana, koska

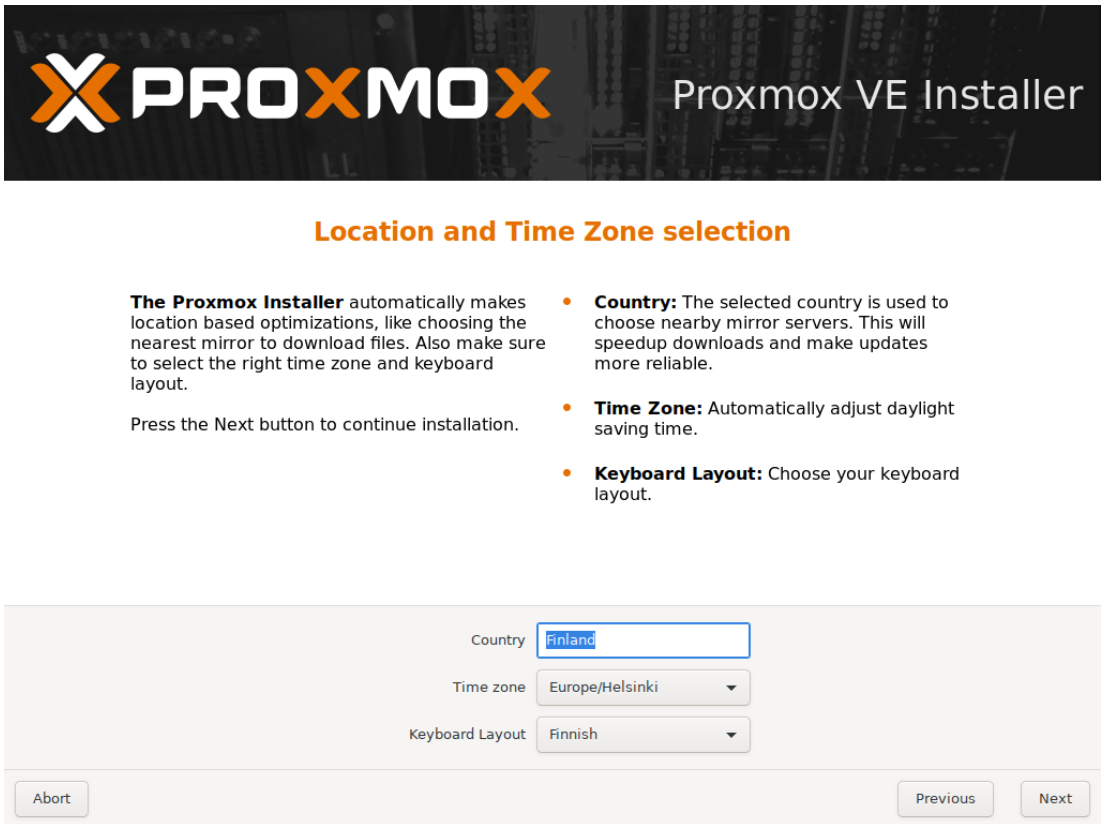
root-käyttäjällä voi tehdä muutoksia koko järjestelmään. Viimeisenä määritetään Proxmox VE:n verkkoasetukset (kuva 8). ”Management Interface” kohtaan valitaan laite, jota Proxmox käyttää verkkoyhteydessään. ”Hostname” kohdassa voi määrittää Proxmox VE:n isäntänimen. ”IP Address” kohtaan voi määrittää Proxmox VE:n IP-osoitteen, josta voi yhdistää GUI:hin. Muut verkkoasetukset voi jättää oletukseksi.

Lopuksi asennusohjelma näyttää järjestelmäkokoonpanon tiedot (kuva 9). Tästä pystyy tarkistamaan määritykset ja tiedot ennen varsinaista asennusta. Painamalla ”Install” asennus käynnistyy. Asennuksen päätyttyä asennusohjelma näyttää osoitteen, johon yhdistämällä pääsee graafiseen käyttöliittymään (kuva 10). Asennuksen viimeistely vaatii uudelleen käynnistyksen. Uudelleen käynnistyksen jälkeen selaimella voi ottaa yhteyden Proxmox VE:n GUI:in (kuva 11). Tästä käyttöliittymästä voi muuttaa hypervisorin asetuksia sekä virtualisoida koneita. Proxmox on valmis käyttöön.

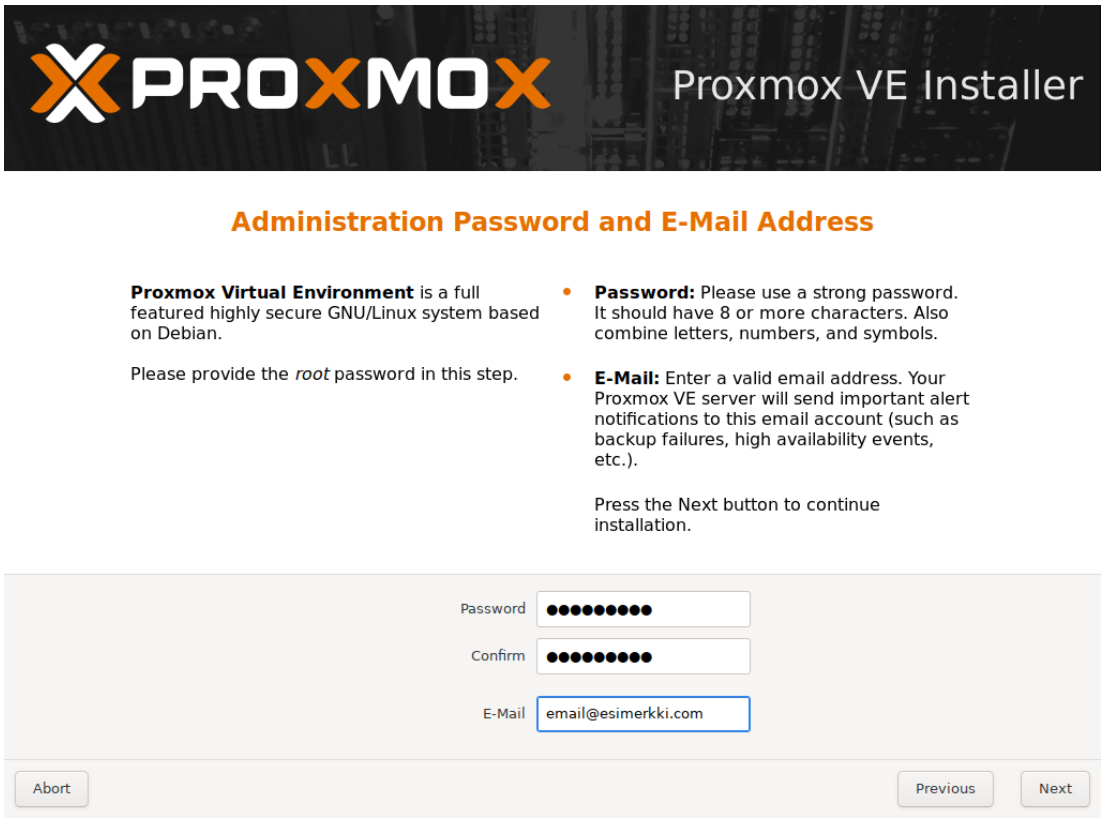


Kuva 4. Proxmox VE:n EULA

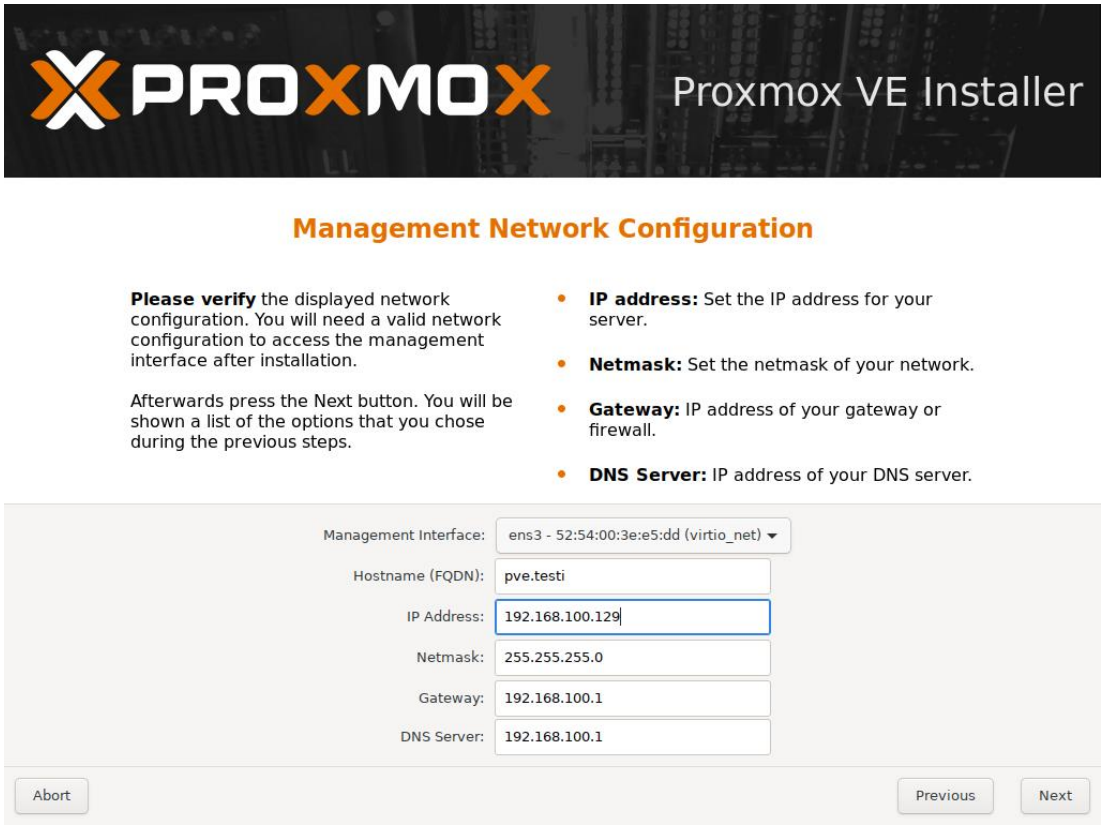
Kuva 5. Asennusaseman valinta.



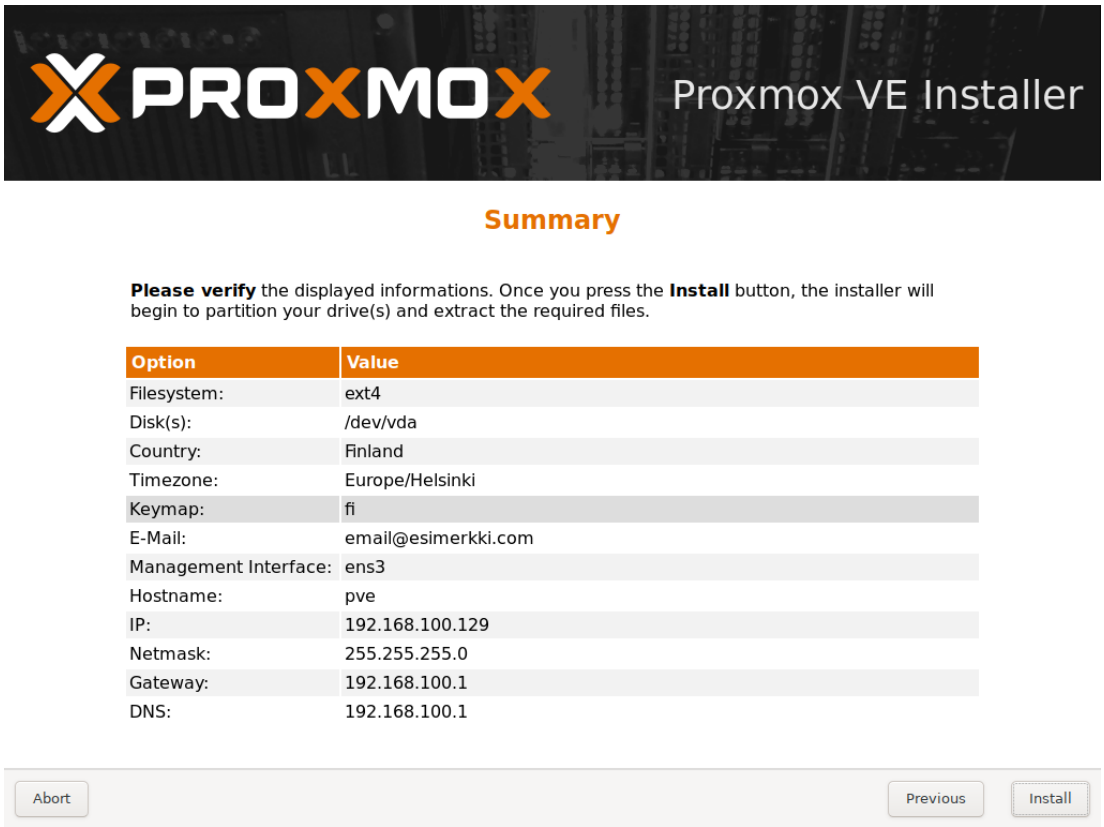
Kuva 6. Maan, aikavyöhykkeen ja näppäimistöasettelun valinta.



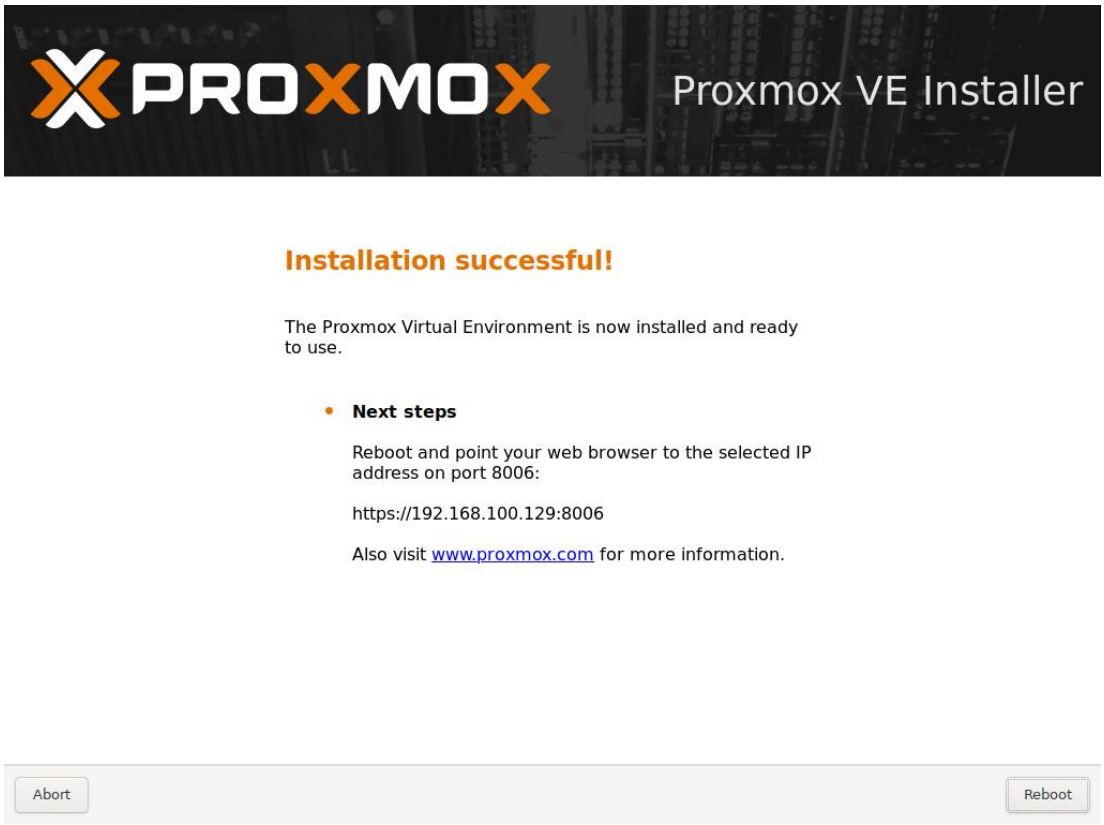
Kuva 7. root -käyttäjän salasanan asettaminen ja sähköpostin syöttäminen.



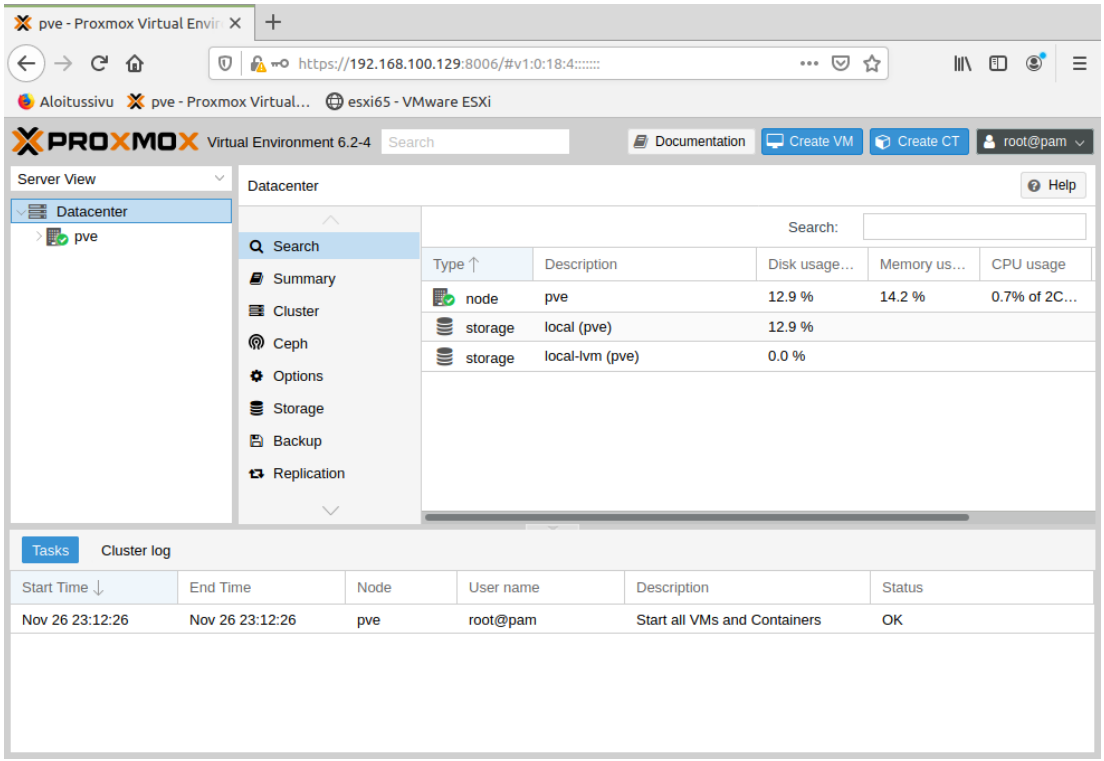
Kuva 8. Verkkoasetusten määritys.



Kuva 9. Kokoonpanon tarkistus.



Kuva 10. Asennus on valmis.



Kuva 11. Proxmox VE graafinen käyttöliittymä Firefox-selaimessa.

3.3 VMware ESXi vs Proxmox VE

Yksi suosituimmista virtualisointityökaluista on VMWaren ESXi. VMWare ESXi on Proxmox VE:n tapaan tyypin 1 hypervisor, jolla luodaan ja hallitaan virtuaalikoneita. Vaikka molemmat ovat kehitetty samaan tarkoitukseen, on niiden välillä merkittäviä eroja.

VMWare ESXi on suljetun lähdekoodin järjestelmä ja se käyttää omaa VMkerneliä. Proxmox VE on avoimen lähdekoodin järjestelmä ja pohjautuu Debian järjestelmään hyödyntäen muokattua Ubuntu LTS kerneliä. Proxmox VE tarjoaa kaikki ominaisuudet ilmaiseksi. ESXi:stä on ladattavissa ilmainen versio, mutta valtaosa ominaisuuksista on saatavilla vain maksullisissa versioissa. (Millman 2020).

Molempia hypervisoreita voidaan nähdä yrityksillä käytössä, mutta Proxmox VE on sopivampi pienemmille yrityksille ja ohjelmistokehitykselle. VMWaren ESXi sopii paremmin isoille yrityksille, jotka muun muassa tarjoavat IaaS-palveluita (Infastructure as a service). (Millman 2020).

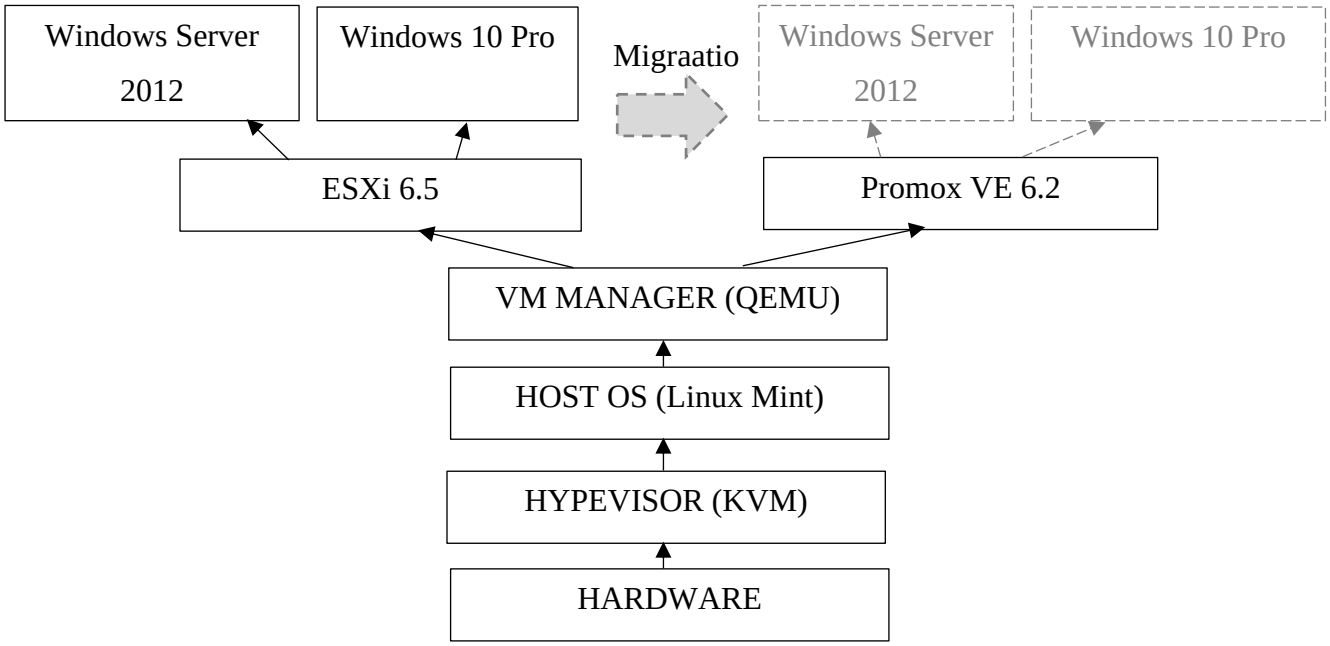
4 VIRTUAALIKONEEN MIGRAATIO ESXI:STÄ PROXMOX VE:HEN

4.1 Taustatietoa

Tässä opinnäytetyössä tarkastelen teoriassa sekä käytännössä virtuaalikoneiden migraatiota kahden eri virtualisointiympäristön välillä. Taloushallintopalveluihin pääsääntöisesti erikoistuvassa yrityksessä on pohdittu VMWaren ESXi:n korvaamista Proxmox VE:llä. Syitä tälle vaihdolle ovat muun muassa Proxmoxin ominaisuuksien määrä verrattuna sillä hetkellä yrityksessä käytössä olevaan ESXi:n ilmaisversioon. Proxmox nähtiin kustannustehokkaampana sekä luotettavampana ratkaisuna verrattuna ESXi:hin. Yritys halusi selvittää, miten virtuaalikoneet pystyttäisiin siirtämään ESXi:stä Proxmoxiin. Tässä työssä tarkastellaan käytännön tapaa siirtää virtuaalikoneet yhdestä ympäristöstä toiseen hyödyntäen virtualisointia.

Koska en omista palvelinlaitteistoa, toteutan työn seuraavalla parhaalla mahdollisella tavalla: Virtualisoin ESXi- ja Proxmox-palvelimet omalla koneellani hyödyntäen KVM-teknologiaa. Tätä varten asensin koneelleni Windows 10 käyttöjärjestelmän ohelle Linux Mint käyttöjärjestelmän. Yksi mahdollisuus olisi ollut käyttää myös Windows 10:n sisäänrakennettua Hyper-V hypervisoria, mutta Proxmox VE pystyy hyödyntämään KVM:ää myös sisäistetyssä virtualisoinnissa. Linux Mint käyttöjärjestelmässä on valmiiksi asennettuna KVM, joten siksi valitsin Linux Mintin.

Sekä ESXi että Proxmox VE asennetaan käyttäen samanlaista virtualisoitua laitteistoa. Molemmat palvelimet saavat käyttöönsä neljä ydintä, 8Gb muistia sekä 60Gb dynaamista tallennustilaa. Verkko silloitetaan koneeni Ethernet-portista, jolloin ESXi ja Proxmox VE on mahdollista saada samaan aliverkkoon. ESXi:hin luodaan kaksi virtuaalikonetta. Toiseen virtuaalikoneeseen asennetaan Windows 10 Pro työasema ja toiselle Windows Server 2012 R2 palvelin. Molemmat käyttöjärjestelmät ovat valikoitu case-yrityksen käyttämien järjestelmien mukaisesti. Windows 10 työasemalle asennetaan Windows 10 mukana tulevien oletusohjelmien lisäksi Google Chrome-selain ja Notepad++-tekstinmuokkain varmistamaan, että sama virtuaalikone siirtyy ympäristöstä toiseen.



Kuva 12. Testiympäristön rakennehierarkia

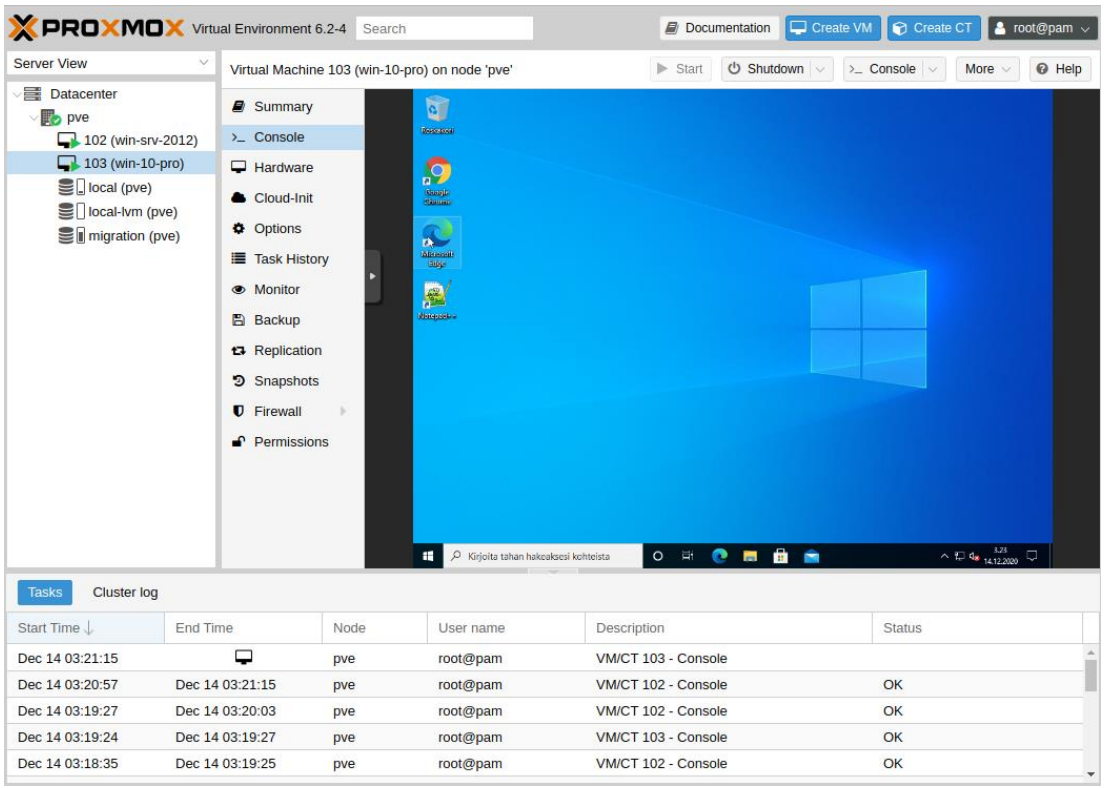
4.2 Virtuaalikoneen migraatio

Virtuaalikoneen migraatioprosessi aloitetaan virtuaalikoneiden viemisellä ESXi-palvelimelta. Aluksi suoritan muutaman toimenpiteen virtuaalikoneille viennin onnistumiseksi. Ennen vientiä poistamme koneilta VMWare Tools -ohjelman. VMWare Tools on VMWaren kehittämä ohjelma, joka sisältää ajureita ja työkaluja helpottamaan virtuaalikoneiden käyttöä ja toimintaa. VMWare Toolsia ei voi käyttää Proxmox VE:ssä vaan se tulee poistaa molemmilta koneilta. Kun ohjelman poistoasennus on onnistunut, molemmat virtuaalikoneet sammutetaan.

ESXi:n selainkäyttöliittymästä viedään virtuaalikoneiden VMDK-tiedostot. VMDK-tiedosto sisältää virtuaalikoneen ”kovalevyn”. Ladattuani sekä Windows 10 työaseman ja Windows Server 2012 tiedostot koneeni tallennustilaan, loin Filezilla-tiedostojensiirto-ohjelmalla FTP-yhteyden Proxmox VE-palvelimelle. Loin uuden tallennusaseman Proxmox VE:hen tiedostojen siirtoa varten, mikä toimii tallennustilana

migraatiota varten VMDK-tiedostoille. Näin löydän helpommin Filezillalla kohdesijainnin, jonne haluan siirtää VMDK-tiedostot.
Seuraavaksi tulee luoda kaksi virtuaalikonetta Proxmox VE:hen. Kummallekin koneelle määritetään väliaikaisesti virtuaalinen tallennusasema, jotka poistetaan myöhemmin. Proxmox VE ei pysty suoraan lukemaan VMDK-tiedostoa vaan tiedosto pitää muuttaa luettavaan muotoon. Tätä varten käytetään Proxmoxin komentorivillä komentoa *qm importdisk* VMDK-tiedoston muuntamiseksi QCOW2-tiedostoksi. Tämä prosessi voi viedä aikaa, vaikka muunnettavat tiedostot olisivatkin pieniä (alle 10Gt). Onnistuneiden muunnosten jälkeen QCOW2-tiedostot ilmestyvät määrättyyn datastorageen. Omassa tapauksessani halusin QCOW2-tiedostojen tallennuskohteeksi *migration* datastoragen. Seuraavaksi määritettiin virtuaalikoneiden asemiksi muunnetut virtuaalikovalevyt. Kovalevyjen emuloinniksi suositeltiin SCSI-emulointia, mutta casetapauksessa SCSI-emulointi johti käynnistyessään jatkuviin *sinisiin kuolemanruutuihin* (blue screen of death, BSOD). Virtuaalikoneet kaatuivat joka kerta, kun koneet käynnistettiin. Ongelma korjattiin muuttamalla emulointi SATA-emuloinniksi. Se aiheutti uuden ongelman käynnistysjärjestyksessä. Koska SCSI-asemaa ei enää ollut, kone ei kyennyt käynnistymään vanhentuneen käynnistysjärjestyksen vuoksi. Päivittämällä käynnistysjärjestyksessä SCSI-aseman tilalle SATA-aseman ongelma korjaantui.

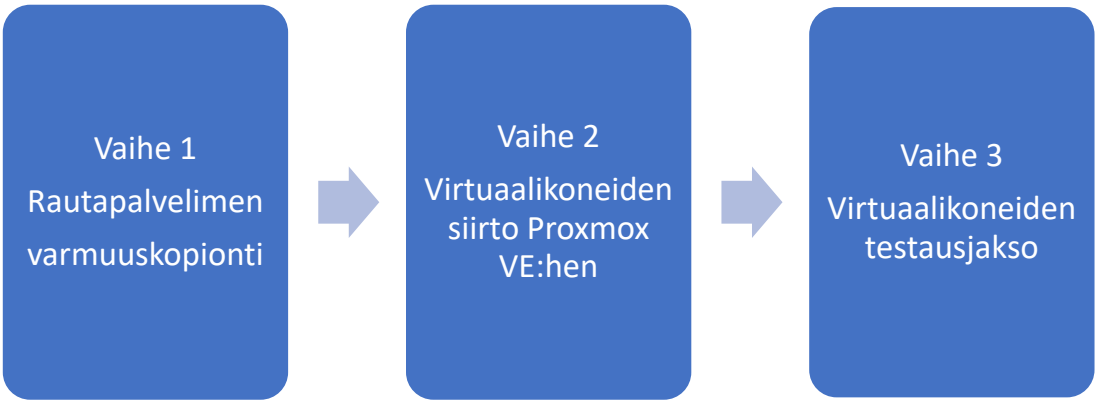
Kun molemmat virtuaalikoneet ovat käynnistyneet työpöydälle asti, virtuaalikoneiden toiminnallisuuden voidaan tarkastaa. Molemmat koneet toimivat samalla tavalla kuin ESXi-ympäristössä ja ehkä jopa hieman paremminkin. Windows 10 työasemalle asennetut ohjelmat Chrome ja Notepad++ toimivat edelleen (kuva 13). Koneet yhdistävät verkkoon ja ovat toimintakelpoisia. Migraatio on onnistunut.



Kuva 13. Windows 10 Pron käynnistettynä Proxmox VE:ssä migraation jälkeen.

4.3 Suunnitelma migraatiolle yrityksessä

Yrityksessä on käytössä kolme fyysistä palvelinta. Jokaisella palvelimella on noin 8-10 virtuaalikonetta. Yhdellä palvelimella sijaitsevien virtuaalikoneiden koko on yhteensä noin 1Tt. Tällaisen määrän migraatio kestää pitkän aikaa. Siksi migraatio toteutetaan kolmessa vaiheessa. Vaihe koostuu kolmesta osasta: palvelimen varmuuskopiointi, migraatio ja testausjakso. Ensiksi koko palvelimesta tehdään varmuuskopio. Jos migraatiossa jokin epäonnistuu, palvelin voidaan vielä palauttaa vanhaan tilaansa varmuuskopioista. Varmuuskopioinnin jälkeen palvelimelta viedään virtuaalikoneet ESXi:stä Proxmox VE:hen. Virtuaalikoneet tuodaan suoraan Proxmox VE-palvelimelle. Viimeisenä vaiheena on testijakso. Testijakso kestää noin kaksi viikkoa ja sen aikana virtuaalikoneiden toimivuutta testataan ja seurataan. Mikäli testijakson aikana ei ilmene ongelmia, migraation suoritetaan seuraavalle palvelimelle. Tämä toistetaan, kunnes migraatio on suoritettu loppuun.



Kuva 14. Virtuaalikoneiden migraatioprosessi

Migraatio tulisi suorittaa silloin, kun työntekijät eivät käytä virtuaalikoneita. Viikonloppuillat ovat siksi parhaimmat ajankohdat. Migraatio suoritetaan ensin vähemmän kriittisille virtuaalikoneille, seuraavaksi kriittisemmille virtuaalikoneille ja lopuksi tärkeimmille virtuaalikoneille. Koko migraatiolle menee aikaa noin 6-8 viikkoa. Todellinen aika voi olla pitempi, jos migraatiossa tapahtuu virheitä tai sopivia ajankohtia migraatiolle ei tule.

5 LOPUKSI

5.1 Ongelmat ja haasteet

Tämä työ osoittautui huomattavasti haastavammaksi, mitä olin aluksi ajatellut. Yksi suurimmista haasteista oli uusien järjestelmien käyttäminen. Käytin työssäni Linux-pohjaista käyttöjärjestelmää virtualisoimaan ESXi:n ja Proxmox VE:n. Vaikka olen aiemmin käyttänyt Debian-pohjaista käyttöjärjestelmää, Linux-järjestelmässä oli paljon opeteltavaa. Linuxin terminaali on hyvin keskeinen osa järjestelmää ja eri komentojen muistaminen ja osaaminen on merkittävää. Koska käytän itse pääosin Windows-pohjaisia järjestelmiä, terminaalin käyttäminen oli melko vierasta minulle. Yksi vaikeimmin opittava asia oli erityisesti tiedostopolkujen oikeinkirjoitus terminaalissa. Windows-järjestelmä antaa hyvin paljon anteeksi polkujen oikeinkirjoituksessa.

Esimerkiksi Windowsin komentorivi tulkitsee polut C:/users/ ja C:/USERS samana polkuna. Linuxissa isoilla ja pienillä kirjaimilla on merkitystä. Terminaali tulkitsee polut /home/user ja /home/USER kahtena eri polkuna. Lisäksi Linuxin terminaali on hieman rajatumpi erikoismerkkien suhteen verrattuna Windowsin komentokehotteeseen. Esimerkiksi alaviivamerkkiä ei voi käyttää tiedostonimissä vaan tavuviivan käyttäminen on suositeltavampaa. Tämä aiheutti monesti ylimääräistä työtä nimetessäni tottumuksestani tiedostoja haluttuun muotoon, jota terminaali ei pystynyt tunnistamaan.

Vaikka migraation selittäminen ja tekeminen sai työn näyttämään yksinkertaiselta, oli se kaikkea muuta kuin helppoa. KVM:n käyttöönotto ja QEMU:n asentaminen oli vaivalloista Linux-ympäristön ollessa vielä vieras. Ensimmäiset virtuaalikoneet eivät käynnistyneet tai asentuneet monista yrityksistä huolimatta. KVM:n käyttöäkin olin jo lopettamassa ja siirtymässä minulle tutumpaan hypervisoriin, Oracle VirtualBoxiin. VirtualBoxilla sain virtualisoitua ESXi:n ja Proxmox VE:n. Yrittäessäni luoda ESXi:hin virtuaalikonetta huomasin, ettei ESXi antanut asentaa 64-bittisiä käyttöjärjestelmiä. Tämä johtui vanhentuneesta VirtualBoxista. Edellä mainittu ongelma olisi ehkä saanut korjattua asentamalla uusimman version, mutta Linux-paketeista haettuna sai vain vanhentuneen version. Oracle-sivuilta ladattuna asennus ei onnistunut poistetun vanhentuneen version haamutiedostojen estäessä asennuksen. Luovuin lopulta VirtualBoxista ja palasin takaisin KVM:n puoleen.

Sain virtualisoitua ESXi:n ja Proxmoxin KVM:n avulla. Seuraavana ongelmana oli viedä virtuaalikoneen VMDK-tiedosto. Latasin tiedoston ESXi:n selainkäyttöliittymän kautta käyttämällä Firefox-selainta. Windows 10-työaseman VMDK-tiedostoa viedessä lataus aina epäonnistui viime hetkellä. Ongelma todennäköisesti liittyi Firefoxiin, joten siirryin käyttämään Chrome-selainta. ESXi 6.5 versiossa oli bugi, joka ajaa ESXi:n virhetilaan selainkäyttöliittymään kirjautuessa, joka taas esti selainkäyttöliittymän käytön. Internetin foorumeita selatessa monet vastaukset osoittivat ongelman olevan Flash playerissä. Ratkaisuksi osoittautui kuitenkin VMWaren oma päivitetty vSphere Embedded Host. Päivityksen jälkeen selainkäyttöliittymä alkoi toimia Chrome selaimella ja sain VMDK-tiedostot ladattua onnistuneesti.

Tähän lopuksi haluaisin huomioida, että luomani testiympäristö käytti kooltansa pieniä virtuaalikoneita. Vajaan 10Gt kokoisen virtuaalikoneen muuntaminen Proxmox VE:lle luettavaan muotoon kesti arviolta 15min. Yrityksessä virtuaalikoneiden koko voi kasvaa monen sadan gigatavun kokoisiksi. Muunnosprosessi on tärkeä, koska Proxmox VE ei osaa suoraan lukea VMDK-tiedostoa. Proxmox VE tukee muun muassa Linuxista tuttua QCOW2-tiedostoa. Mikäli yrityksessä on käytössä esimerkiksi tietokantoja sisältävä palvelin, voi muuntoprosessiin mennä paljon aikaa. Tämä tulisi huomioida migraation aikataulutuksessa.

5.2 Omaa pohdintaa

Ennen tätä työtä en ollut kuullut Proxmox VE:stä. Tutustuessani Proxmox VE:hen totesin Proxmox VE:n olevan paljon parempi ratkaisu verrattuna ESXi:n ilmaisversioon. Migraation jälkeen huomasin virtuaalikoneiden toimivan Proxmoxissa paremmin kuin ESXi:ssä. Syytä en tiedä, mutta Proxmox VE saattaa toimia virtualisoituna paremmin kuin ESXi. Lisäksi KVM:n käyttäminen virtualisoinnissa saattoi parantaa Proxmox VE:n suorituskykyä.

Tämä opinnäytetyö oli iso oppimisprosessi. Vaikka työ koostui enimmäkseen kokeilusta, yrittämisestä, onnistumisesta ja epäonnistumisesta, opein paljon Linux-ympäristössä työskentelystä sekä virtualisoinnista. Työn loppupuolella sain inspiraatiota pystyttää oman Proxmox-ympäristön palvelimien testaamiseen ja opetteluun. Työ ei ollut minulle helppo, sillä jouduin työskentelemään paljon oman osaamisalueeni ulkopuolella. Oma tietämykseni järjestelmistä koostui enimmäkseen Windows pohjaisista järjestelmistä. Vaikka Linux-järjestelmä on avoimempi ja muokattavissa käyttäjän tarpeiden mukaan, koen Windowsin käyttäjäystävällisyyden merkittävänä asiana. En ehkä itse jatka Linuxin käyttöä työasemana, mutta palvelinjärjestelmissä saatan tutustua Linuxiin syvemmin.

LÄHTEET

Bauer, R. 2018. ’What’s the Diff: VM’s vs Containers’. Backblaze. 28.6.2018. Viitattu. 23.11.2020. https://www.backblaze.com/blog/vm-vs-containers/

Fedoseenko, V. 2019. ’A brief history of virtualization, or why do we divide something at all’. Ispsystem. 25.11.2019. Viitattu 14.12.2020. https://www.ispsystem.com/news/brief-history-of-virtualization

Millman, R. 2020. ’VMware vSphere vs Proxmox: Which is best for your business?’. ITPro 20.10.2020. Viitattu 30.11.2020. https://www.itpro.co.uk/virtualisation/34516/vmware-vsphere-vs-proxmox-which-is-best-for-your-business

Proxmox www-sivut 2020. Viitattu 25.11.2020. https://proxmox.com/

Rouse, M. 2012. ’open virtualization format (OVF)’. TechTarget 5.2012. Viitattu 7.12.2020. https://searchvmware.techtarget.com/definition/open-virtualization-format-OVF

Taylor, C. 2020. ’What is a Hypervisor Server?’. ServerWatch 17.11.2020. Viitattu 23.11.2020. https://www.serverwatch.com/virtaulization/hypervisor-server/

VMWare www-sivut 2020. Viitattu 22.11.2020. https://www.wmvare.com/topics/glossary/content/hypervisor