

Opinnäytetyö (AMK)

Tietojenkäsittely

Tietoliikenne

2013

Niko Rantanen

# VIRTUALISOINTI KVM- HYPERVISORILLA



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Tietoliikenne

Helmikuu 2013 | 68 sivua

Esko Vainikka

Niko Rantanen

## VIRTUALISOINTI KVM-HYPERVISORILLA

Opinnäytetyön tavoitteena oli perehtyä virtualisointiin teknisenä ratkaisuna teoriassa ja käytännön osuudessa asentaa Linux-käyttöjärjestelmä virtualisointialustaksi käyttäen avoimen lähdekoodin KVM-hypervisoria. Lisäksi tarkoitus oli tutustua kyseisen virtualisointiratkaisun eri ominaisuuksiin ja testata virtualisointia käytännön ratkaisuin.

Tutkimusmenetelminä on käytetty käytäntöön valmistavaa teoriaa ja sen pohjustamaa eksperimentaalista projektia, jossa työn otsikon mukaista virtualisointiratkaisua toteutetaan asennuksesta käyttöönottoon aina palveluiden ajamiseen alustalla.

Ubuntu- ja Fedora-käyttöjärjestelmiä alustoina käyttäen KVM-hypervisorilla virtualisoitiin onnistuneesti erilaisia palveluja, kuten Windows XP ja thin client -palvelin. Lisäksi testattiin onnistuneesti KVM:n eri ominaisuuksia, kuten live migrationia ja jaetun tallennuksen käyttöä.

KVM-hypervisorilla virtualisointi sopii niin aloittelijoille kuin edistyneemmillekin käyttäjille ja tarjoaa edellytykset myös raskaaseen kaupalliseen käyttöön. KVM:n asentaminen ja peruskäyttö edellyttävät jonkin verran perustietämystä Linux-käyttöjärjestelmässä toimimisesta.

### ASIASANAT:

Virtualisointi, KVM, hypervisor, virtuaalikone

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Data communications

February 2013 | 68 pages

Esko Vainikka

Niko Rantanen

## VIRTUALIZATION WITH KVM HYPERVISOR

The objective of this thesis was to familiarize oneself with virtualization as a technical solution both in theory and practice, using the open-source KVM hypervisor running on Linux operating system. Another purpose was to study the aforementioned solution's various features and test virtualization in the form of practical actions.

The thesis is divided into three parts: two consist of theory and the final one applies the learned theory into practice. The first part aims at answering the question: What is virtualization? The second part deals more closely with implementing virtualization by investigating matters and issues that should be considered before moving on to virtualization. The final part executes an experimental project that builds a virtualization platform on Linux utilizing the KVM hypervisor.

The implementation of virtualization and its features were examined for the thesis in practice with the KVM hypervisor, as well as the creation, configuration and running of virtual service offerings.

The installation and basic use of KVM requires a basic knowledge of working under Linux. In general, virtualization with KVM is suitable both for beginners and advanced users. The platform is also able to handle demanding enterprise-level virtualization loads.

### KEYWORDS:

Virtualization, KVM, hypervisor, virtual machine

# SISÄLTÖ

<b>KÄYTETYT LYHENTEET JA SANASTO</b>	<b>7</b>
<b>1 JOHDANTO</b>	<b>8</b>
<b>2 VIRTUALISOINTI</b>	<b>9</b>
2.1 Virtualisointi ja pilvilaskenta	9
2.2 Hypervisorit	9
2.2.1 Bare-metal-hypervisorit	10
2.2.2 Hosted-hypervisorit	11
2.2.3 Suosituimmat hypervisorit	12
2.3 Virtuaalikoneet	16
2.4 Kerrokset ja tyypit	17
2.5 Käyttäjärjestelmän virtualisointi	20
2.6 Hyödyt	21
2.7 Haitat	22
2.8 Kulut	23
<b>3 VIRTUALISOINNIN TOTEUTTAMINEN</b>	<b>25</b>
3.1 Laitteistovaatimukset	25
3.2 P2V-konversio tai uusi virtuaalikone	25
3.3 Suunnittelu	26
3.3.1 Verkon valmistelu ja tietoturva	27
3.3.2 Tallennuksen valmistelu	30
3.4 Virtuaalikoneiden luominen	32
3.5 Käyttäjärjestelmien asentaminen virtuaalikoneeseen	35
<b>4 KÄYTÄNNÖN TOTEUTUS KVM-HYPERVISORILLA</b>	<b>36</b>
4.1 KVM:n ominaisuudet	36
4.2 Valmistelu	38
4.2.1 Laiteresurssit	38
4.2.2 Kiintolevyn osiointi ja Linuxin asentaminen	40
4.2.3 Virtualisoitavat kohteet	41
4.2.4 Hallintatyökalut	41
4.3 KVM-ympäristön luominen	42

4.3.1 KVM:n asentaminen	42
4.3.2 Verkon asetukset	43
4.3.3 Virtuaalikoneen luominen Virt-manager-työkalulla	45
4.3.4 Thin client -palvelimen asennus ja käyttöönotto	46
4.4 KVM-ympäristössä toimiminen	49
4.4.1 KVM:n hallitseminen oVirt-hallintatyökalulla	49
4.4.2 Virtuaalikoneiden hallitseminen Windowsista	54
4.4.3 Jaettu tallennus ja live migration	56
<b>5 YHTEENVETO JA JOHTOPÄÄTÖKSET</b>	<b>64</b>
<b>LÄHTEET</b>	<b>66</b>

## KUVAT

Kuva 1. Siltasovittimen (br0) lisäys /etc/network/interfaces-verkkoasetustiedostoon.	44
Kuva 2. Create a new virtual machine -painike.	45
Kuva 3. Thin client -palvelimen (LTSP server) asennus KVM-virtuaalikoneeksi.	47
Kuva 4. /etc/ltsp/dhcpd.conf-tiedoston lähiverkkokohtaiset rivit ja arvot lihavoituna punaisella.	47
Kuva 5. Päätelaitte virtualisoitiin VirtualBoxilla.	48
Kuva 6. Päätelaitte käynnistymässä palvelimelta 192.168.100.21.	48
Kuva 7. Päätelaitteelle verkosta käynnistynyt Ubuntu-käyttäjärjestelmä.	49
Kuva 8. Vastaukset annettuna oVirtin engine-setupin kysymyksiin.	51
Kuva 9. Virheeseen keskeytynyt oVirt-engineen asennus.	51
Kuva 10. oVirtin käyttöliittymä.	52
Kuva 11. Uuden virtuaalikoneen luominen oVirtilla.	53
Kuva 12. Virtual Machines -lista ja Console-painike, josta instansseja pääsee katselemaan.	54
Kuva 13. Virt-manager etänä Windowsista käsin SSH- ja X window -yhteydellä.	55
Kuva 14. Virt-viewer-etähallinta.	56
Kuva 15. Virtualisoidut levyt tiedostopalvelimella Volumes-välilehdellä.	58
Kuva 16. Verkkojaon luominen tiedostopalvelimella.	59
Kuva 17. Isäntäpalvelimen yhdistäminen NFS-verkkojakoon.	59
Kuva 18. Palomuurin konfigurointi live migrationia varten.	60
Kuva 19. Virt-manager toiseen isäntäkoneeseen yhdistämisen jälkeen.	61
Kuva 20. Virt-managerin Live migration -asetukset.	62
Kuva 21. Slitaz1-virtuaalikoneen live migration toiselle isäntäpalvelimelle.	62
Kuva 22. Slitaz1-virtuaalikoneen live migration suoritettu.	63

## KUVIOT

Kuvio 1. 1-tyyppin hypervisor (NI Developer Zone 2009).	11
Kuvio 2. 2-tyyppin hypervisor (NI Developer Zone 2009).	12
Kuvio 3. Xen-hypervisor (Xen 2012).	13
Kuvio 4. Hyper-V isäntäosioilla ja lapsiosioilla kuvattuna (Portnoy 2012, 31).	15
Kuvio 5. VMware-isännän verkkotyöskentely (Portnoy 2012, 173.)	29
Kuvio 6. Virtuaalitalennuksen toiminta (Portnoy 2012, 153).	31
Kuvio 7. Tilakaappausten muodostama levyketju (Portnoy 2012, 47).	34

## KÄYTETYT LYHENTEET JA SANASTO

Hypervisor	Sovellusalausta laitteiston päällä, jolla virtualisoituja instansseja voidaan ajaa (Portnoy 2012, 2).
Isäntä	Virtuaalikoneita ajava laite tai käyttöjärjestelmä.
KVM	Kernel-based Virtual Machine. Linuxin kerneliin integroitu hypervisor (Virtuutopia 2012).
Live migration	Live migration tarkoittaa virtualisoinnin ominaisuutta, jossa ajossa oleva virtuaalikone siirretään lennosta toiseen fyysiseen isäntäkoneeseen. Prosessi kopioi koko virtuaalikoneen muisti-intanssin yhdeltä fyysiseltä palvelimelta toiselle niin nopeasti, että uusi virtuaalikone voi jatkaa operoimista ilman keskeytystä sekä vahingoittamatta tiedon eheyttä tai käyttökokemusta. (Ruest & Ruest 2009, 68; Portnoy 2012, 177.)
Pilvilaskenta	Internetissä sijaitsevan etäpalvelinverkoston käyttö tiedon tallennukseen, hallitsemiseen ja laskentaan paikallisen palvelimen sijaan (Merriam-Webster 2013).
Resurssipooli	Laitteistoresurssien kokoelma.
Vieras	Isännällä ajettava virtualisoitu laite tai käyttöjärjestelmä.
Virtuaalikone	Virtualisoitu tietokone, jolla on samat ominaisuudet kuin fyysisellä verrokillansa (Portnoy 2012, 35).
Virtualisointi	Tietojenkäsittelyssä tekniikka, jolla luodaan virtuaalinen versio komponentista, laitteesta tai resurssista, kuten palvelimesta, tallennuslaitteesta, tietoverkosta tai käyttöjärjestelmästä (Webopedia 2013).

# 1 JOHDANTO

Virtualisointi on yksi kuumimmista tietotekniikan trendeistä tänä päivänä. Lähes kaikki IT-maailman palvelut pyritään virtualisoimaan tavalla tai toisella erilaisten hyötyjen siivittäminä ja tästä johtuen virtualisoinnin mahdollistavat tekniikat kuten hypervisorit kehittyvät jatkuvasti eri ratkaisujen kilpaillessa keskenään. Nopeasti kehittyvä KVM-hypervisor tarjoaa avoimen lähdekoodin virtualisointiratkaisun, jonka ominaisuudet ovat markkinoiden kärkeä palvelinvirtualisointikäytössä.

Opinnäytetyön tavoitteena on tutkia virtualisointia avoimen lähdekoodin ratkaisuja käyttäen. Teoriaosuudessa selvitetään alan kirjallisuutta ja verkkolähteitä käyttäen virtualisoinnin perusteita ja tekniikoita, ominaisuuksia, käyttömahdollisuuksia, hyötyjä ja haittoja sekä toteuttamista teoriassa. Empiirisessä osiossa teoriaa sovelletaan käytäntöön asentamalla Linux-käyttöjärjestelmä KVM-hypervisorilla, jonka avulla testataan virtualisointia käytännössä luomalla virtuaalikoneita ja testaamalla ratkaisun tarjoamia palveluita lähiverkosta.

Pääasiallisena lähdeaineistona käytettiin englanninkielistä materiaalia suomenkielisen materiaalin vähäisen saatavuuden vuoksi. Kirjallisuudesta valittiin tarkan harkinnan tuloksena kaksi kirjaa, joista uudemman kivenkova ajankohtaisuus täydentää toisen, hieman vanhemman kirjan massiivista tietomäärää. Kirjallisia lähteitä tukevat lukuisat ajankohtaiset verkkojulkaisut, jotka toimivat myös itse KVM:n asentamisen, ominaisuuksien ja käytön lähteinä. Lähteiden valinnassa on keskitytty etenkin ajankohtaisuuteen ja luotettavuuteen.

Tutkimusmenetelminä on käytetty teoreettista pohjustusta ja sen pohjustamaa eksperimentaalista projektia, jossa työn otsikon mukaista virtualisointiratkaisua toteutetaan asennuksesta käyttöönottoon aina palveluiden ajamiseen alustalla.



## 2 VIRTUALISOINTI

### 2.1 Virtualisointi ja pilvilaskenta

Virtualisoinnista riippuvainen pilvilaskenta (Cloud computing) on yksi tämän päivän tunnetuimmista IT-trendeistä. 1990- ja 2000-luvun taitteen Internet-buumin tavoin monet yritykset ovat ottamassa pilvilaskennan mahdollisuuden tarjontaansa, ja suuret yritykset tarjoavatkin jo monenlaisia pilvipalveluita. Esimerkiksi Applen iCloud-palvelun käyttäjät voivat tallentaa omia kuviaan, musiikkiaan ja muuta digitaalisessa muodossa olevaa omaisuuttaan, jolloin ne ovat käytettävissä mistä tahansa. Applen lisäksi mm. Microsoft, Amazon ja Google tarjoavat samantyyppisiä pilvipalveluita. (Portnoy 2012, 14.)

Pilvilaskenta tarjoaa yksinkertaisen tavan käyttää ja hyödyntää resursseja. Virtualisointi toimii pilvilaskennan moottorina tarjoamalla skaalautuvat, helposti saatavat ja käytettävät resurssit. Ennen virtualisaatiota järjestelmien ylläpitäjät kuluttivat 70 prosenttia ajastaan rutiinitehtäviin ja ongelmiin, mikä jätti hyvin vähän aikaa innovatiivisille ratkaisuille ja kasvulle. Virtualisointi ja sen siivittäjä pilvilaskenta tarjoavat automaattioratkaisuja, jotka vähentävät ylläpitokustannuksia ja antavat mahdollisuuden dynaamiselle palvelutuotannolle. Pilvilaskenta luo virtuaalisen konesalin, joka tarjoaa resursseja sitä mukaa kuin niitä tarvitaan. (Portnoy 2012, 14.)

### 2.2 Hypervisorit

Hypervisor on laitteiston päällä sijaitseva sovellusalue, jolla virtuaalikoneita voidaan ajaa. Sitä kutsutaan myös Virtual Machine Monitoriksi (VMM). Hypervisor antaa virtuaalikoneelle saman laitteistoalustan, jolla itse hypervisor operoi. Sillä on käytössään täydet järjestelmäresurssit, ja virtuaalikoneen suorituskyky ei saa erota merkittävästi tai ollenkaan fyysisestä verrokista. (Portnoy 2012, 2-3.)

Virtuaalikoneen käyttöjärjestelmä eli vieras käyttöjärjestelmä (guest operating system) huijataan hypervisorin toimesta uskomaan virtuaaliseen todellisuuteen, jossa se pystyy oikeasti näkemään ja kommunikoidaan isäntäkoneen fyysisten laitteiden kanssa. Todellisuudessa jokainen vieras käyttöjärjestelmä saa käyttöönsä vain murto-osan fyysisen isäntäkoneen resursseista. Isäntäkoneessa voi olla esim. 64 Gt muistia, mutta vieras uskoo, että resursseja on käytössä vain 4 Gt. D-levy voi olla 200 Gt, vaikka se todellisuudessa on vain pieni osa moninkertaisen tallennuskapasiteetin omaavasta massamuistijärjestelmästä. Prosessoreita voi olla kaksi ja verkkokortteja yksi, mutta isäntäkoneessa voi olla moninkertainen määrä kumpaakin. (Portnoy 2012, 24.)

Hypervisorin tehtävänä on myös tasapainottaa vieraiden työmäärät. Hypervisor toimii kaikkien pyyntöjen välikätenä vieraiden ja laitteistoresurssien välillä tarjoten asianmukaiset ja ajallisesti järkevät resurssit kaikille. Vertauskuvallisesti hypervisor on kuin liikennepoliisi, joka ohjaa liikennettä tasapuolisesti tarjoamalla kohtuulliset odotusajat jokaiselle suunnalle ja käsittelee jokaista tietä puolueettomasti. (Portnoy 2012, 25.)

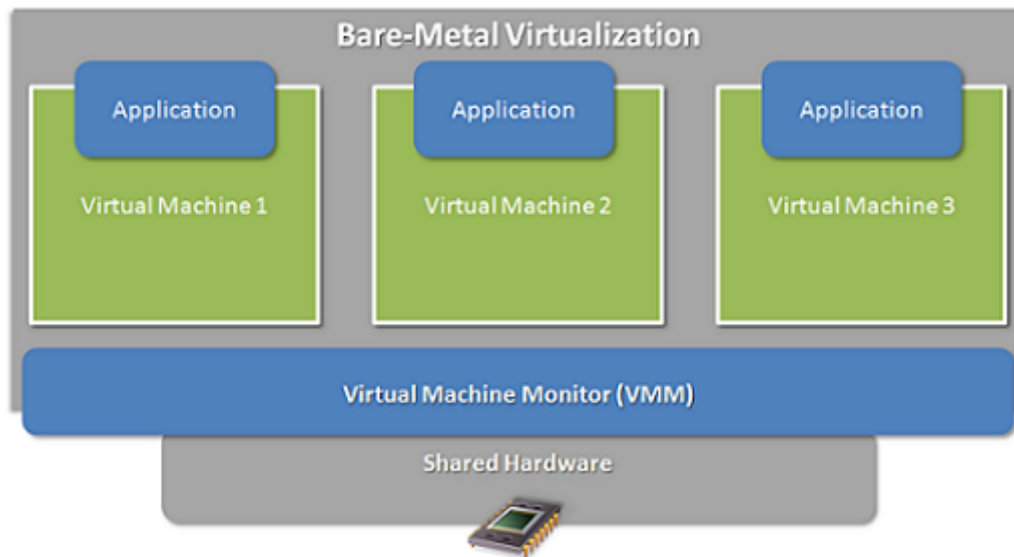
Hypervisor hoitaa kaikki pyynnöt samalla periaatteella. Kun vieras sovellus pyytää levyn lukua vieraalta käyttöjärjestelmältä, käyttöjärjestelmä lähettää pyynnön edelleen näkemälleen virtuaalilevylle. Tässä kohtaa hypervisor kaappaa pyynnön ja kääntää sen todellisuutta vastaavaksi sekä syöttää sen tallennusaliijärjestelmään. Pynnön tullessa takaisin hypervisor syöttää datan takaisin vieraalle, joka vastaanottaa paketin kuin se tulisi suoraan fyysiseltä laitteelta. (Portnoy 2012, 25-26.)

Hypervisorit voidaan jakaa kahteen luokkaan.

### 2.2.1 Bare-metal-hypervisorit

1-tyyppin hypervisorit ajetaan suoraan palvelinlaitteiston päällä (Kuvio 1) ilman palvelinkäyttöjärjestelmää, jolloin hypervisorin ja laitteiston välinen viestintä on

tehokkaampaa ja turvallisempaa. Vihamieliset vieraat ja ohjelmakoodit eivät pääse vaikuttamaan tämän tyyppin hypervisorin. (Portnoy 2012, 21-22.)

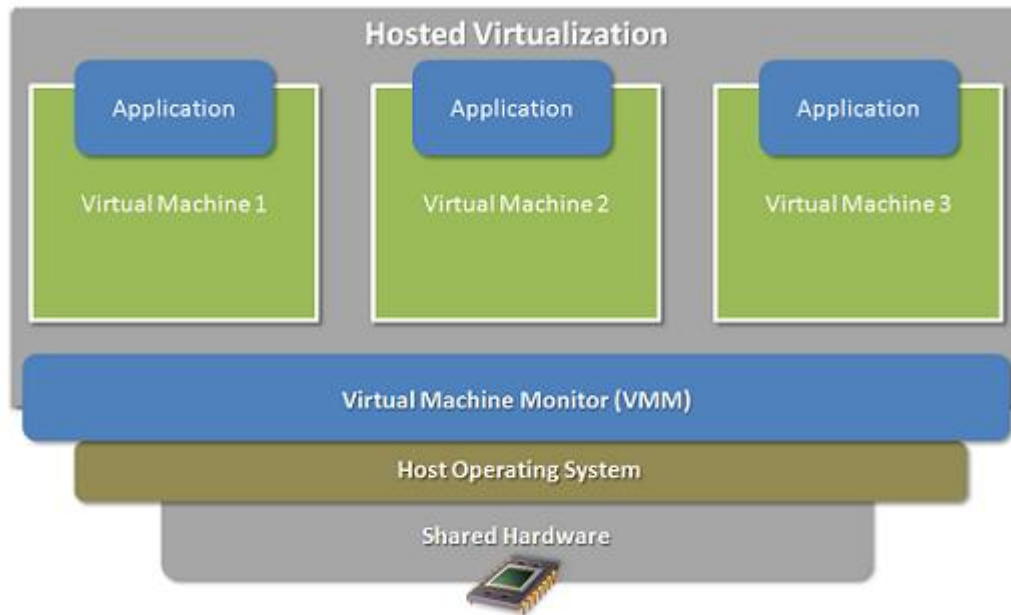


Kuvio 1. 1-tyypin hypervisor (NI Developer Zone 2009).

### 2.2.2 Hosted-hypervisorit

Tyyppin 2 hypervisor on perinteisessä käyttöjärjestelmässä ajettava sovellus, joka on helpompi ottaa käyttöön (Kuvio 2). Tyyppin 2 hypervisor joutuu käyttämään virtuaalikoneilta ja laitteistolta tulleet I/O-pyyntöt isäntäkäyttöjärjestelmän kautta, mikä hidastaa prosessointia. Lisäksi käyttöjärjestelmässä ajettava hypervisor on altis isäntäkäyttöjärjestelmässä tapahtuviin katkoihin esim. uudelleenkäynnistyksen vaativien päivitysten muodossa, joka pakottaa myös kaikki virtuaalikoneet käynnistymään uudelleen. (Portnoy 2012, 23.)

Hosted-hypervisorin hyötyjä ovat sen käytön helppous ja vaivattomuus; virtuaalikoneita voi käynnistää, kun niitä tarvitaan ja sammuttaa tai jäädyttää, kun niitä ei enää tarvita (Portnoy 2012, 54). Tämä mahdollistaa esim. tarvittavilla työkaluilla varustetun käyttöjärjestelmäkoonpanon hyödyntämisen nopeasti ja vaivattomasti (Pitkänen 2012).



Kuvio 2. 2-tyyppin hypervisor (NI Developer Zone 2009).

### 2.2.3 Suosituimmat hypervisorit

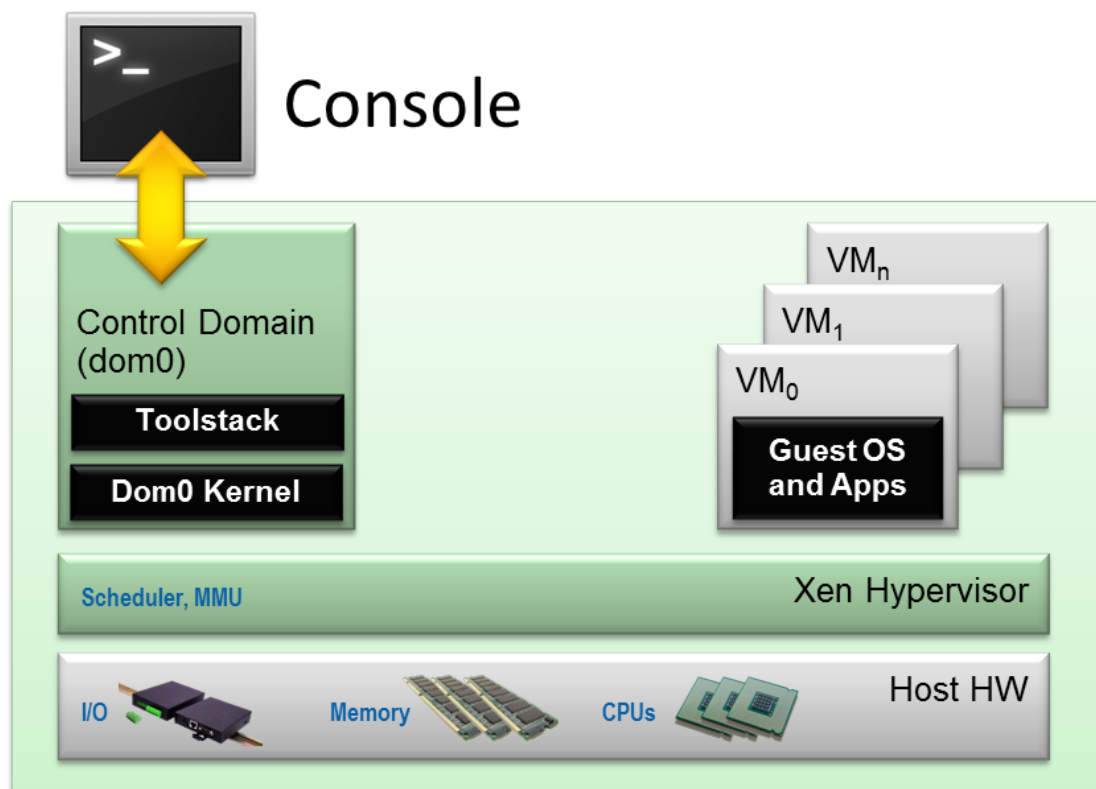
Suomessa palvelinvalmistajat tukevat aktiivisesti kolmea palvelinvirtualisointiratkaisua eli hypervisoria: VMware, Microsoft Hyper-V ja Linux-lähtöinen Xen. Xen on käytössä erityisesti Citrixin Xenserverin tuotteistuksessa. Kaikki nämä hypervisorit tukevat sekä Windowsin että Linuxin virtualisoinnista. (Hämäläinen 2009a.)

Tarkasteluun on otettu tunnetuimmat ilmaiset hypervisorit, ylivoimaisesti isoimman markkinaosuuden omaavan VMwaren sekä kaupalliset että ilmaiset versiot ja kaksi muuta kaupallisten palvelinratkaisujen käyttämää hypervisoria Microsoftilta ja Citrixiltä. Kaikki nämä hypervisorit tukevat laitteistovirtualisoinnin edellyttämiä Intel VT -ja AMD-V-teknologioita.

### Citrix Xen

Citrixin Xen on ainoa markkinoilla oleva avoimen lähdekoodin 1-tyyppin hypervisor. Xen käyttää Linux-pohjaista Domain 0 -domainia, joka sisältää

laitteistoajurit ja työkalut virtuaalikoneiden luomisen, tuhoamisen ja konfiguraatioiden hallintaan. Toisin sanoen Domain 0 on virtuaalikone, joka pääsee laitteistoon käsiksi erikoisoikeuksilla, käsittelee kaikki järjestelmän I/O-funktiot ja vuorovaikuttaa muiden virtuaalikoneiden kanssa (Kuvio 3). Lisäksi Domain 0 sisältää hypervisorin hallintatyökalut ja saattaa hallinnan käyttöliittymän näkyviin. Xen voi suorittaa täysin virtualisoituja tai paravirtualisoituja vieraita. (Xen 2012.)



Kuvio 3. Xen-hypervisor (Xen 2012).

Xen-hypervisorin käyttämä live migration -teknologia on nimeltään XenMotion.

Citrixin Xenserver-tuotteistuksesta on neljä versiota:

- Free-versio sisältää Xen-hypervisorin perusominaisuudet.
- Advanced Edition -versio sisältää korkean saatavuuden ja edistyskellisen hallinnan työkaluja.

- Enterprise Edition -versio sallii palvelimille dynaamiset resurssit resurssipoolista.
- Platinum Edition -versio sisältää sekä virtuaalisten että fyysisten palvelimien dynaamisen provisioinnin. (Citrix 2012; Hämäläinen 2009a.)

Vaikka Xen on Linux-pohjainen, sen hallintakäyttöliittymä toimii myös Windows-käyttöjärjestelmässä (Hämäläinen 2009a).

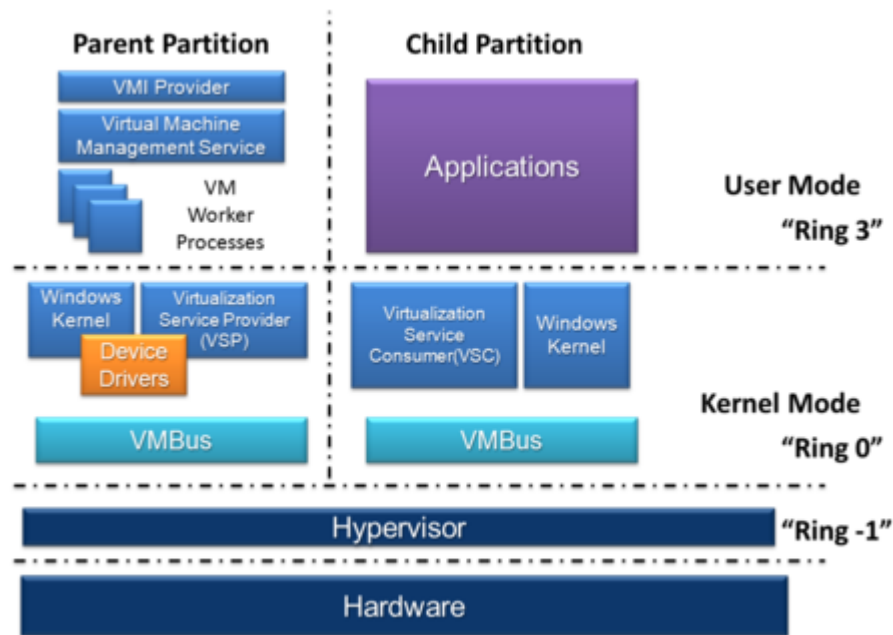
### **Kernel-based Virtual Machine (KVM)**

Kernel-based Virtual Machine on Linuxin kerneliin integroitu tyyppi 2 hypervisor, ja se on täysin avoimen lähdekoodin ohjelmisto. Kernel-tason virtualisoinnissa isäntäkäyttöjärjestelmä toimii erityisesti muokatulla ytimellä eli kernelillä, joka sisältää laajennuksia, jotka on suunniteltu hallinnoimaan useita virtuaalikoneita (Virtuatopia 2012). KVM on täyden virtualisoinnin tarjoava ratkaisu, joka koostuu ladattavista kernel-moduuleista `kvm.ko` ja `kvm-intel.ko` tai `kvm-amd.ko` (riippuen prosessorista). KVM:llä voi ajaa muokkaamattomia Linux- ja Windows-levykuvia. (KVM 2012a, KVM 2012b.)

Tarkemmat kuvaukset KVM:n toiminnasta ja ominaisuuksista löytyvät opinnäytetyön luvusta neljä, joka pohjautuu KVM-hypervisorin.

### **Microsoft Hyper-V**

Microsoftin hypervisor Hyper-V tuli virtualisointimarkkinoille kolmesta suuresta viimeisenä (Hämäläinen 2009a). Hyper-V on 1-tyypin hypervisor, jonka ajamia virtuaalikoneita kutsutaan osioiksi vieraiden sijaan. Xenin Domain 0:n tapaan Hyper-V:llä on erityinen isäntäosio Windows Server 2008/2012 -käyttöjärjestelmällä varustettuna, jolla on suora pääsy laitteistoresursseihin. Isäntäosio luo ja hallitsee lapsiosioita sekä hoitaa järjestelmänhallintafunktiot ja laiteajurit. (Portnoy 2012, 31.)



Kuvio 4. Hyper-V isäntäosiolla ja lapsiossiolla kuvattuna (Portnoy 2012, 31).

Hyper-V integroituu Windows-palvelinten työkaluihin Server Manageriin ja Mmc-hallintakonsoliin (Hämäläinen 2009a).

Koska Hyper-V pohjautuu Xenin referenssiarkkitehtuuriin ja käyttää samaa vhd-formaattia, Hyper-V ja Citrixin Xenserver ovat suurilta osin yhteensopivia ja hallittavissa samoilla hallintatyökaluilla (Hämäläinen 2009a).

### Oracle VM VirtualBox

Oraclen VirtualBox on ilmainen avoimeen lähdekoodiin perustuva 2-tyyppin hypervisor. Se on helppokäyttöisyyden lisäksi myös ominaisuuksiltaan hyvin monipuolinen ja sopii näin ollen myös edistyneemmille käyttäjille. Yksityiskohtaista dokumentaatiota työkalun käyttöön on jo reilusti jokaiseen tarkoitukseen. VirtualBox on saatavilla Windowsille, Linuxille ja Mac OS X:lle. (Pitkänen 2012.)

## VMware

VMware mahdollisti ensimmäisenä hypervisorin ajamisen suoraan laitteiston päältä 1-tyyppinä. Ilmainen VMware ESXi on pieni 32 megatavun hypervisor, joka ladataan sisäiseen USB-porttiin palvelimen laitteistossa. Tämä tekee uusien isäntäpalvelimien lisäämisestä helppoa. (Ruest & Ruest 2009, 114.)

ESXi on kehitetty aikaisemmasta ESX-hypervisorista, jossa oli hypervisorin lisäksi Linux-pohjainen service console -hallintapaneeli. Hallintapaneeli on riisuttu uudemmasta ESXi-hypervisorista ison kokonsa ja tietoturvaohjelmien vuoksi. Vuoden 2011 viidennestä julkaisusta lähtien näistä kahdesta vain ESXi-arkkitehtuuri on saatavilla. (Portnoy 2012, 28.)

VMware Server (ent. GSX) on palvelinkäyttöön tarkoitettu 2-tyypin hypervisor, joka on ladattavissa ilmaiseksi (Portnoy 2012, 27).

Muita 2-tyypin hypervisoreita VMwarelta ovat työpöytäkäyttöön tarkoitetut VMware Workstation (Windows-versio), VMware Fusion (Mac-versio) ja VMware Player (ilmainen, riisuttu versio Workstationista ja Fusionista).

### 2.3 Virtuaalikoneet

Virtuaalikoneella on pääpiirteittäin samat ominaisuudet kuin fyysisellä koneella tai palvelimella. Se kannattelee käyttöjärjestelmää sille määriteltyjen resurssien avulla, joita virtuaalikone voi käyttää samoin kuten fyysinen verrokkinsa. Yhden käyttöjärjestelmän sijaan yhdellä palvelimella voidaan ajaa useita virtuaalikoneita, jotka kaikki pitävät sisällään oman erillisen käyttöjärjestelmänsä. (Portnoy 2012, 35-36.)

Virtuaalikone koostuu konfiguraatitiedostosta ja virtuaalisista levytiedostoista (virtual disk files). Konfiguraatitiedosto kertoo levytiedostojen sijainnin ja määrittelee käytettävissä olevat resurssit eli virtuaalisen laitteiston: prosessorit, muistit, tallennusvälineet, verkkoyhteydet, DVD-asemat. Levytiedostot sisältävät samat tiedot kuin yksittäisellä fyysisellä kiintolevyllä. Jokainen virtuaalinen



kiintolevy on siis tiedosto, joka käyttäytyy kuin tyypillinen lohkoerustainen kiintolevy. Virtuaalikoneen käyttöjärjestelmä sijaitsee levytiedostossa, ja virtuaalikoneella voi olla useita levytiedostoja fyysisen koneen tapaan. Levytiedostot voi määrittellä kasvamaan käytön mukaan, jolloin ne ovat vain niin isoja, kuin niillä on sisältöä. (Ruest & Ruest 2009, 30-31; Portnoy 2012, 35-36.)

Virtuaalikoneet mahdollistavat fyysisen palvelimen täyden hyödyntämisen, sillä palvelimella voi olla niin monta virtuaalikonetta kuin laitteistoresurssit antavat myöden (Ruest & Ruest 2009, 30.).

## 2.4 Kerrokset ja tyypit

Virtualisointi on laajentunut useille konesalin kerroksille. Tämän seurauksena virtualisoinnista on olemassa eri tyyppisiä, joilla on merkittäviä eroja. Täyden hyödyn virtualisoinnista saavassa, dynaamisessa konesalissa on käytössä ainakin seitsemän virtualisointikerrosta. (Ruest & Ruest 2009, 25.)

### **Palvelinvirtualisointi**

Palvelinvirtualisointi eli Server Virtualization (SerV) jakaa fyysisen käyttöjärjestelmäinstanssin virtuaaliseksi instanssiksi tai virtuaalikoneeksi. Fyysisestä palvelimesta tulee näin ollen isäntä kaikille virtuaalisille käyttöjärjestelmille tai virtuaalikoneille, joita siinä ajetaan. Todellisen palvelinvirtualisoinnin tarjoavilla hypervisoreilla voidaan virtualisoida kaikki x86 - ja x64 -arkkitehtuurin käyttöjärjestelmät, kuten Windows, Linux ja UNIX. (Ruest & Ruest 2009, 25.) Palvelimen virtualisointi voidaan toteuttaa kahdella tavalla:

- **Sovellusvirtualisointi**, jolloin virtuaalinen käyttöjärjestelmä ajetaan isäntäkäyttöjärjestelmän virtuaalisella sovellusalustalla.
- **Laitteistovirtualisointi**, jolloin virtuaalinen käyttöjärjestelmä ajetaan suoraan laitteiston päällä olevalla virtuaalisella sovellusalustalla ilman jo olemassa olevaa käyttöjärjestelmää. (Ruest & Ruest 2009, 25.)

## **Tallennusvirtualisointi**

Tallennusvirtualisointi eli Storage Virtualization (StoreV) yhdistää usean fyysisen tallennuslaitteen kapasiteetin yhdeksi loogiseksi tallennusvälineeksi (Ruest & Ruest 2009, 27).

## **Verkon virtualisointi**

Verkon virtualisointi eli Network Virtualization (NetV) jakaa verkkoresurssin kaistanleveyden itsenäisiksi kanaviksi, jotka voidaan antaa tiettyjen resurssien käyttöön. Yksinkertaisin esimerkki tästä on virtuaalinen lähiverkko (virtual local area network, VLAN), joka erottelee fyysisen lähiverkon loogisiksi osiksi. (Ruest & Ruest 2009, 27.)

## **Hallinnan virtualisointi**

Hallinnan virtualisointi eli Management Virtualization (ManageV) käsittää konesalin hallintaan käytetyt - sekä fyysiset että virtuaaliset - teknologiat (Ruest & Ruest 2009, 28).

## **Työpöytävirtualisointi**

Työpöytävirtualisointi eli Desktop Virtualization (DeskV) on työpöytänäkymien tarjoamista virtuaalikoneiden kautta. Henkilökohtaiset työpöydät on näin ollen keskitetty yhteen sijaintiin, johon käyttäjät ottavat yhteyden omista laitteistaan (Ruest & Ruest 2009, 28). Jokainen käyttäjä saa käyttöönsä oman virtuaalikoneen, jossa on henkilökohtainen työpöytä ja ohjelmistot määrittämineen. Virtuaalisen työpöydän visuaalinen ilme ja toiminnallisuus ovat täsmälleen samat kuin fyysisen työpöytäkoneen vastaavat. (Knorr 2010.)

Virtuaalisia työpöytiä ajetaan konesalin palvelimilla, joiden laitteistot ovat paljon tehokkaampia ja vakaampia kuin perinteisissä PC-tietokoneissa. PC:iden sijaan

virtuaalisia työpöytiä voi käyttää ns. thin client -pääteistä, joiden käyttöikä on 7-10 vuotta ja jotka käyttävät vain 5-10 prosenttia PC:n sähkönkulutuksesta. Thin clientin rikkoutuminen ei vaikuta virtuaaliseen työpöytään mitenkään, vaan tiedot ovat turvassa palvelimella, jolla myös varmuuskopioiminen hoituu käyttäjistä riippumattomasti. Lisäksi tietoturvaohjelmat voidaan keskittää yhdelle isäntäpalvelimen virtuaalikoneelle, jolloin se suojaa kaikkia palvelimella olevia virtuaalisia työpöytiä. (Portnoy 2012, 16.)

Kaksi suosittua kaupallista työpöytävirtualisointiratkaisua ovat Citrix XenDesktop ja VMware View (Portnoy 2012, 16).

### **Esitystapavirtualisointi**

Esitystapavirtualisointi eli Presentation Virtualization (PresentV) tarjoaa vain esitystapakerroksen keskitetystä sijainnista käyttäjälle. Esitystapavirtualisoinnin protokollat ovat tärkeitä työpöytä- ja palvelinvirtualisoinnille, koska niitä tarvitaan virtuaalisten resurssien käyttöön ja hallintaan. (Ruest & Ruest 2009, 29.)

### **Sovellusvirtualisointi**

Sovellusvirtualisointi eli Application Virtualization (AppV) on kuten sovelluspohjainen palvelinvirtualisointi, mutta koko käyttöjärjestelmän moottorina toimimisen sijaan se erottaa sovelluksia käyttöjärjestelmästä. Tietty sovellus virtualisoidaan vain kerran, minkä jälkeen virtuaalimoottori jakaa sovelluksen käyttöliittymän palvelimelta verkon kautta käytettäväksi missä tahansa Windows-käyttöjärjestelmässä. Esimerkiksi jotkin työkalut voivat konvertoida kaikki MSI-paketoidut sovellukset sovellusvirtuaalimuotoon automaattisesti. (Ruest & Ruest 2009, 29; Hämäläinen 2009.)

Sovellusten virtualisointiin on olemassa kaksi syytä: käyttöönoton helppous ja sovellusten välinen vuorovaikutus. Uuden sovellusversion tullessa markkinoille virtuaalinen sovellus tulee päivittää vain kerran sen sijaan, että jokaikiselle PC:lle pitäisi suorittaa toimenpide erikseen. Toinen hyöty on ohjelmien

kapsuloiminen, jolloin toisiin sovelluksiin tehdyt muutokset eivät vaikuta kapsuloituun sovellukseen ja toisinpäin. Suosittuja sovellusvirtualisointiratkaisuja ovat Microsoft App-V, Citrix Application Streaming ja VMware ThinApp. (Portnoy 2012, 17.)

## 2.5 Käyttöjärjestelmän virtualisointi

Käyttöjärjestelmien virtualisointi voidaan jakaa kolmeen luokkaan: paravirtualisointi, täysi virtualisointi ja laitteistovirtualisoinnin avustama täysi virtualisointi.

### **Paravirtualisointi**

Paravirtualisoinnissa vieraan käyttöjärjestelmän kernel on mukautettu suoritettavaksi jollakin tietyllä hypervisorilla. Muokkauksen ansiosta kernel voi välittää kaikki tärkeät toimenpiteet nopeasti suoraan hypervisorille sen sijaan, että se yrittäisi itse niitä suorittaa prosessorin korkeimman etuoikeuden omaavalla suoritustasolla. Paravirtualisointi onnistuu lähinnä avoimen lähdekoodin vapaasti muokattavilla käyttöjärjestelmillä, kuten Linuxilla, ja niillä, joiden omistajat ovat suostuneet tekemään tarvittavat muunnokset tietyn hypervisorin tukemiseksi. (Virtuutopia 2012.)

### **Täysi virtualisointi**

Täysi virtualisointi puolestaan tukee muokkaamattomia käyttöjärjestelmiä, jolloin hypervisor käsittelee ja muokkaa kernelien pyytämät etuoikeudet ja suojatut CPU-toimenpiteet prosessoriemuloinnilla. Emulointi vaatii aikaa ja järjestelmäresursseja, joten se ei ole niin tehokasta kuin paravirtualisoinnin tarjoama toimintamalli. (Virtuutopia 2012.)

## Laitteistovirtualisointi

Uusien prosessorien Intel VT- ja AMD-V-tekniologiat lisäävät prosessoriin hypervisorin käyttöön yhden suoritustason, jolla on kaikkein korkeimmat etuoikeudet. Näin ollen muokkaamattomat käyttöjärjestelmät voivat käyttää edelleen vapaana olevaa oman kernelinsä määrittelemää CPU-suoritustasoa ilman emuloinnista johtuvaa resurssihukkaa. Laitteistovirtualisointi on kaikkein yleisin virtualisointimalli tänä päivänä. (Virtuatopia 2012.)

### 2.6 Hyödyt

Yksittäiset fyysiset palvelimet käyttävät kapasiteetistaan keskimäärin vain 5-10 prosenttia. Tällainen yksittäinen palvelin tarvitsee kuitenkin täyden määrän virtaa, täyden jäähdytyksen ja täyden laitepaikan konesalista. Virtualisoinnin avulla samainen yksittäinen palvelin voi olla isäntänä yli kymmenelle virtuaalikoneelle, palvelimelle tai työpöydälle. Nämä virtuaalikoneet eivät vie omaa sähköä tai laitepaikkaa eivätkä tuota turhaa lämpöä. Ne kaikki ovat kuitenkin ominaisuuksiltaan ja palveluiltaan samoja kuin fyysiset verrokkinsa. (Ruest & Ruest 2009, 34.)

Palvelinvirtualisoinnin hyödyt voidaan jakaa seuraavasti:

- **Käyttöönotto.** Täysin uuden virtuaalikoneen voi rakentaa ja kustomoida käyttövalmiiksi alle 20 minuutissa. Tämä on huomattavasti nopeampaa kuin fyysisen koneen käyttöönotto.
- **Liikkuvuus.** Virtuaalikoneen voi siirtää isännältä toiselle milloin tahansa ilman käytön katkeamista Live Migration -tekniikan avulla.
- **Käytön helppous.** Virtuaalikoneen rakennuspalikoita voi muokata haluamukseen. Halutusta virtuaalikoneesta voi luoda standardin kokoonpanon, ja uuden virtuaalikoneen voi luoda kloonamalla standardin virtuaalikoneen lähdetiedostot.

- **Testaus.** Uuden virtuaalikoneen voi luoda käyttövalmiiksi muutamassa minuutissa testaustarkoituksiin esim. ohjelmistokehittäjille. Testauksen valmistuttua virtuaalikoneen voi poistaa napin painalluksella.
- **Tietoturvallisuus.** Virtuaalikoneet voi eristää täysin milloin tahansa katkaisemalla niiden tietoliikenne.
- **Skaalautuvuus.** Virtuaalikoneita voi tehdä lisää helposti, ja laiteresurssien (RAM, suorittimet ym.) lisäys onnistuu virtuaalisesti.
- **Elpyvyys.** Virtuaalikoneiden tiedostot voidaan helposti palauttaa toimivaan isäntäkoneeseen. (Ruest & Ruest 2009, 37-38.)

Virtualisoinnin suosioon on johtanut etenkin sähkö- ja viilennyskulujen räjähdysmainen kasvu. Fyysisten palvelinten ongelma on yleisen mielipiteen mukaan 1) liiallinen tilankäyttö, 2) alikäyttö, 3) liiallinen lämmöntuotanto, 4) liiallinen sähkötarve, 5) tarve monimutkaisille liiketoiminnan jatkuvuussuunnitelmille, 6) jatkuva häiriöaika ja 7) laitehallinnan monimutkaisuus. (Ruest & Ruest 2009, 66.)

Palvelinten virtualisointi tarjoaa merkittävät säästöt kustannuksissa, koska laitteistojen käyttö tehostuu ja ylläpito sekä hallinta keskittyy. Työasemien virtualisointi saavuttaa samat hyödyt. Sovellusvirtualisointi helpottaa sovellusten hallintaa huomattavasti, koska vain palvelimella oleva ohjelmisto tarvitsee ylläpitoa kymmenien yksittäisten työasemien sijaan.

## 2.7 Haitat

Virtualisoinnin tarjoava eri palvelinten tai palvelujen tiivistäminen yhteen laitteeseen tarkoittaa myös huomattavasti suuremman vikaantumispisteen muodostumista, jolloin yhden laitteen hajoamisen vaikutukset voivat olla moninkertaiset vanhaan malliin verrattuna. Tämän välttämiseksi virtualisoidut palvelut pitäisi hajauttaa useammalle fyysiselle laitteelle niin, ettei esim. yhden fyysisen palvelimen vikaantuminen lamaannuta kaikkia virtuaalikoneita. (Dinesh 2011; Skyline 2011.)

Vaikutukset suorituskykyyn ovat minimaaliset, mutta kuitenkin olemassa. Virtualisoinnin aiheuttama ylimääräinen kommunikointikerros laitteiston ja ohjelmiston välillä voi vaikuttaa kriittisten työmäärien suorituskykyyn ja yleisesti työmäärien toimivuuden ennustettavuuteen. (CA Technologies 2012.)

Järkevästi hallittavan virtuaalisen ympäristön rakentaminen kriittisine palvelimineen ei ole niin helppoa kuin fyysisten palvelimien hallinta. Yksittäisten palvelinkokoonpanojen hallinta ja ominaisuudet ovat useimmilla IT-ammattilaisilla ”selkäytimessä”, mutta eri hypervisorien kanssa toimiminen hallitusti edellyttää erikoisosaamista juuri tietylle tuotteelle, oli se sitten KVM, VMware, Hyper-V tai Xen. (Dinesh 2011.)

Jotkin sovellukset eivät ole vielä optimoituja tai valmiita virtualisoinnille. Eräiden toimivien sovellusten tukipalvelut eivät kata toimimista virtuaalisessa ympäristössä lainkaan. Tästä syystä tulisi varmistua, että kaikki virtualisoitavat sovellukset tukevat niiden virtualisoimista myös valmistajan puolelta. Useimmilta käyttöjärjestelmiltä tuki löytyy, mutta vielä on paljon varsinkin vanhempia ohjelmia, joilta se puuttuu. (Dinesh 2011.)

Loppujen lopuksi virtualisoinnin heikkouksien ei kuitenkaan pitäisi estää siihen siirtymistä, sillä hyödyt painavat vaakakupissa huomattavasti enemmän (Dinesh 2011).

## 2.8 Kulut

Virtualisoinnin aikaansaamat rahalliset säästöt ovat merkittäviä, koska laitteistohankinnat vähenevät ja tarvitaan vähemmän fyysisiä palvelimia sekä työpöytiä. Säästöä tulee lisäksi sähkönkulutuksen vähentymisestä, lisenssikulujen laskusta, jäähdytystarpeen pienentymisestä ja tilan säästymisestä. (Ruest & Ruest 2009, 51-52.)

Itse hypervisorit ovat usein palvelinlaitteiden vakio-ohjelmia tai ilmaisia. Näin ollen varsinainen liiketoiminta koostuu hallintaohjelmistoista ja palveluista, joita tarvitaan esim. useiden isäntäkoneiden yhteistyössä ja harmoniassa.

## Lisenssimaksut

Microsoftin palvelinkäyttöjärjestelmien lisenssit sallivat useita virtuaalisia ajoja, ja työpöytäkäyttöjärjestelmien lisenssit sallivat neljän virtuaalisen Windows-työpöydän käytön yhtäaikaisesti yhdelle käyttäjälle kerrallaan. Windows-työpöydät voivat olla samalla lisenssillä eri versioita; esim. kaksi Windows 7:ää, yksi Vista ja yksi XP. (Ruest & Ruest 2009, 51.)

Microsoftin Virtualization Calculator -laskimella voi tarkastella omien laitteistojen virtualisoinnista koituvia hyötyjä Windows-palvelimien osalta. Microsoft tarjoaa myös return on investment (ROI) -laskimen, jolla voi laskea virtualisointiin siirtymisestä koituvia voittoja. (Ruest & Ruest 2009, 51-52.)

Lisäksi säästöjä tulee energian ja operoinnin kuluissa, koska sähköä kuluttavien laitteiden määrä vähenee ja hallinnointi voidaan hoitaa keskitetysti.



## 3 VIRTUALISOINNIN TOTEUTTAMINEN

### 3.1 Laitteistovaatimukset

Virtualisointiin käytettävien prosessorien tulee olla tarpeeksi nykyaikaisia, jotta niissä on laitteistovirtualisoinnin edellyttämät ominaisuudet. Intelin prosessoreissa käytettyjä virtualisointitekologioita ovat Virtual Technology (VT) ja Extended Memory 64 Technology (EM64T), kun taas AMD:llä käytetty teknologia on AMD-V (Ruest & Ruest 2009, 67).

#### **Live Migration**

Jos halutaan käyttää live migration -ominaisuutta eli ajossa olevan virtuaalikoneen siirtämistä lennosta toiseen isäntäkoneeseen, kaikkien isäntien kaikissa prosessoreissa on oltava sama stepping-arvo. Prosessi kopioi myös prosessorien käskykannat alkuperäisen isännän käyttömuistista toiseen koneeseen, joten kummankin osapuolen prosessorien tulee tukea samaa käskykanta. Tästä syystä prosessi ei toimi Intelin ja AMD:n prosessorien välillä, eikä useimmissa tapauksissa myöskään eri stepping-arvoja sisältävien prosessorien välillä. Stepping-arvon voi tarkistaa esim. CPU-Z-työkalun avulla. (Ruest & Ruest 2009, 68.)

### 3.2 P2V-konversio tai uusi virtuaalikone

Uusien virtuaalikoneiden luomiselle on olemassa kaksi tapaa: fyysisestä virtuaaliseksi -konversio (Physical to virtual, P2V) ja tyhjästä uuden luominen. Täysin uuden virtuaalikoneen luominen on yleensä suositumpi tapa, koska siten voidaan asentaa uusin käyttöjärjestelmäversio päivityksineen ja jättää taakse vanhan järjestelmän käytöstä aiheutuneet rapautumat. Lisäksi virtuaalisessa

ympäristössä ei tarvita kaikkia samoja ajureita ja prosesseja. (Portnoy 2012, 52.)

P2V-työkalut konvertoivat koko käyttöjärjestelmän sellaisenaan fyysiseltä palvelimelta virtuaalikoneeksi. Tämä voi olla riskialtista etenkin vanhempien palvelinten tapauksessa, joiden sovellukset ja ympäristöt eivät ole enää nykyaikaisia. Täydellisessä tapauksessa laite konvertoituu virtuaaliseksi alkuperäisessä tilassaan alustariippumattomaan ympäristöön, jossa sitä voidaan käyttää ja hallita koko jäljellä olevan sovelluslinkaarensa ajan ilman pelkoa laitteistorikosta. Toinen hyöty saavutetaan, kun sovellusta ei tarvitse siirtää uuteen käyttöjärjestelmään, jossa se ei välttämättä toimi kunnolla. (Portnoy 2012, 52.)

P2V:n heikkous voi olla myös sen vahvuus. P2V-prosessi kloonaa olemassa olevan koko fyysisen palvelimen virtuaalikoneeksi. Prosessi muuttaa esim. verkkokonfiguraatiota ja fyysiset ajurit virtuaalisiksi, mutta pohjimmiltaan se kopioi lähdepalvelimen virtuaalikoneeseen. (Portnoy 2012, 52-53.)

### 3.3 Suunnittelu

Suunnitteluvaiheessa ensimmäiseksi kannattaa aloittaa potentiaalisten pullonkaulojen etsiminen. Yksi tärkeimmistä pullonkauloista on muisti eli random access memory (RAM). Muistia pitää olla niin paljon kuin mahdollista, koska jokainen virtuaalikone tarvitsee käyttöönsä sille määrätyn määrän muistia. Muistiresurssien täydelliseen hyödyntämiseen tarvitaan 64-bittinen suoritinarkkitehtuuri. (Ruest & Ruest 2009, 46.)

Virtualisointi tarvitsee suuren määrän muistia ja useita prosessoriytimiä, mutta myös tallennuksen ja verkon suorituskyky on erittäin olennainen osa virtuaalista infrastruktuuria, mikä usein unohtuu aloittelijoilta. Useat virtuaalikoneet palvelimella moninkertaistavat myös tallennus- ja tiedonsiirtoväylien rasituksen, jolloin niiden kapasiteettia pitää lisätä. Yleensä paras vaihtoehto on suosia useita pieniä kiintolevyjä muutaman suuren sijaan, vaikka suuremmilla olisi parempi suorituskyky. Tallennusjärjestelmät kykenevät lukemaan ja

kirjoittamaan levyiltään rinnakkaisesti, joten suurempi määrä levyjä sallii enemmän samanaikaista rinnakkaista suorituskykyä, joka on aina nopeampaa kuin tiedon käsittely sarjassa. (Portnoy 2012, 163.)

Virtuaaliseen infrastruktuuriin siirtymisen tärkeänä osana on resurssien järkevä jakaminen ja hallinta virtuaalikoneiden kesken. Tämä tulee ottaa huomioon laitteistokokoonpanoa mietittäessä. Kaikkien isännän virtuaalikoneiden käyttämät resurssit pitää laskea yhteen ja varmistaa, että isännällä on yhteenlasketut resurssit ja vähän enemmän, jotta resursseja jää myös hypervisorille ja ajoittaisille suorituskyvyn kasvun tarpeille. (Portnoy 2012, 27.)

Virtualisoinnin dynaamisen luonteen vuoksi hyvänä käytäntönä on aloittaa pienillä resursseilla ensimmäisen virtuaalikoneen kohdalla, jonka jälkeen resursseja voi lisätä tarpeen mukaan. Paras käytäntö muutamaa poikkeusta lukuun ottamatta on luoda kaikki uudet virtuaalikoneet yhdellä virtuaalisella prosessorilla ja lisätä niitä lisää suorituskyvyn heiketessä. Muistin ja prosessorin käyttö ovat virtualisoinnissa erittäin hyvin optimoituja, joten useimmiten yksi prosessori on tarpeeksi. (Portnoy 2012, 64-65.)

Virtualisointi sulloo useita virtuaalikoneita samaan isäntäkoneeseen, joten kaistanleveyden rajat tulee ottaa tarkasti huomioon virtuaalista ympäristöä suunniteltaessa (Portnoy 2012, 189).

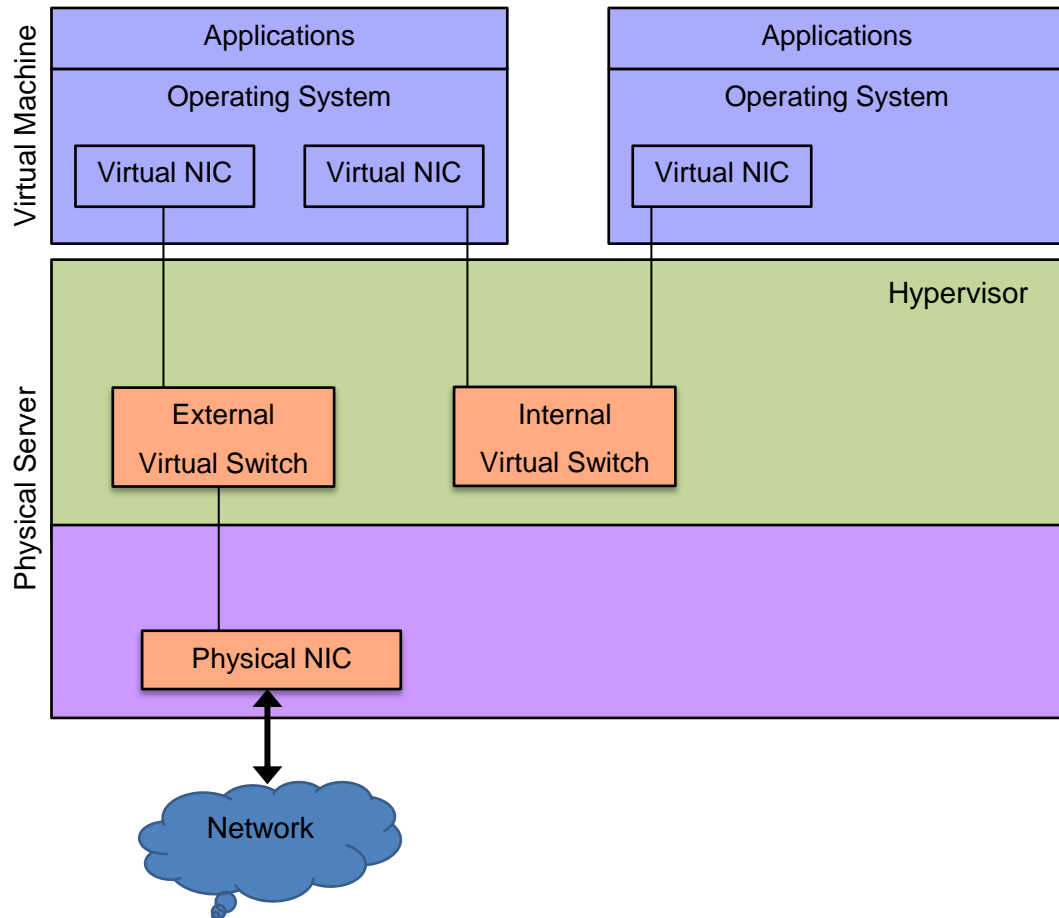
### 3.3.1 Verkon valmistelu ja tietoturva

Verkon nykyinen käyttöaste tulee olla tiedossa. Tiedonsiirtokapasiteetti määrittelee, onko esim. mahdollista hyödyntää live migration -ominaisuutta tai tiedon tallennusta Internet Small Computer Systems Interface (iSCSI) -jakopinnan yli. (Ruest & Ruest 2009, 70.) Verkkotallennus ja etenkin live migration saattavat vaatia erillisen tiedonsiirtoväylän toimiakseen kunnolla (Ruest & Ruest 2009, 74).

Jokaisella virtuaalikoneella tulee olla käytössään jonkinlainen verkkoliitäntä. Virtualisointi tukee virtuaalikoneen eristämistä, yhteyksien sallimista vain

isäntäpalvelimelle, tai yhteyksien täydellistä sallimista. Täydellinen salliminen edellyttää ylimääräisiä verkkokortteja isäntäkoneella, joten verkkokorttien riittävä määrä tulee varmistaa, jos virtuaalikoneita on kymmeniä. Lisäksi verkkoinfrastruktuurin kestävyys pitää varmistaa OSI-mallin verkko- ja siirtokerroksilla, joilla kytkimet ja reitittimet sijaitsevat. (Ruest & Ruest 2009, 73.)

Hypervisorit ylläpitävät virtuaalisia kytkimiä isäntäkoneiden sisällä sekä kokonaisia virtuaalisia verkkoja. Esimerkiksi VMware vSphere -hypervisorissa on kaksi virtuaalista kytkintä, joista toinen on yhdistetty fyysiseen verkkokorttiin ja ulkopuoliseen fyysiseen verkkoon. Toinen virtuaalikytkin ei ole yhteydessä mihinkään fyysiseen kommunikaatioporttiin. Kuviossa 5 on kaksi virtuaalikonetta, joista vain ensimmäinen on kytketty ulkopuoliseen verkkoon johtavaan kytkimeen. Virtuaalikoneiden välillä on sisäinen kytkin, jolloin toinen virtuaalikone voi kommunikoida vain ensimmäisen virtuaalikoneen kanssa. Tämä on virtuaaliympäristöjen yleinen tietoturvakäytäntö, jossa kuvion vasemmanpuoleinen virtuaalikone toimii palomuurina rajoittaen muiden virtuaalikoneiden haavoittuvuutta ja näkyvyyttä ulkopuoliseen verkkoon. (Portnoy 2012, 172-173.)



Kuvio 5. VMware-isännän verkkotyöskentely (Portnoy 2012, 173.)

Sisäisen virtuaalikytkimen liikenne tapahtuu täysin isäntäkoneen muistissa, joka on erittäin paljon nopeampaa kuin jos liikenteen pitäisi kulkea fyysisen kytkimen kautta. Siispä ne virtuaalikoneet, joiden sovellusten välillä tapahtuu paljon tiedonvaihtoa, kannattaa sijoittaa samaan isäntäkoneeseen. (Portnoy 2012, 174.)

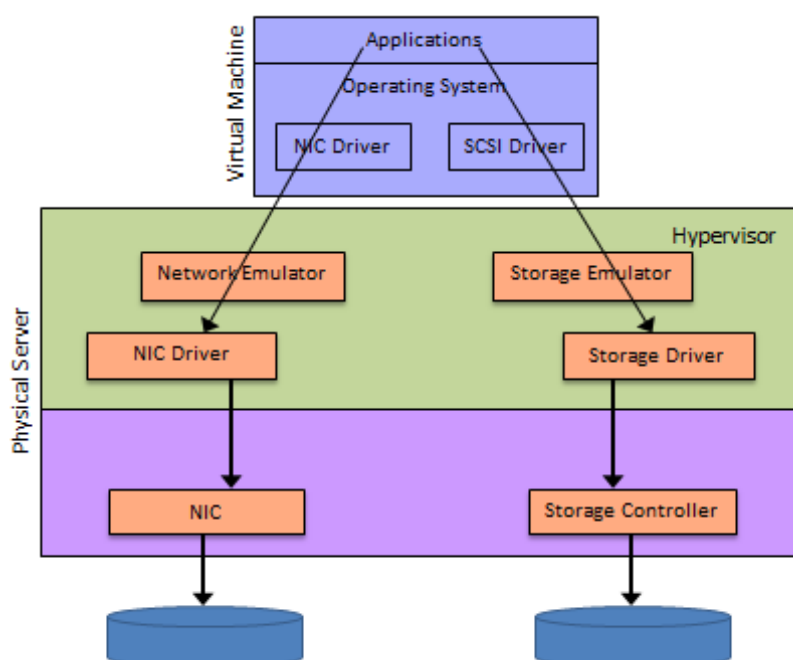
Virtuaalikoneiden verkkoliikennettä voi eristää toisiltaan segmentoimalla sisäistä liikennettä lisäämällä isäntäkoneeseen toinen fyysinen verkkokortti, jolloin hypervisor luo myös toisen ulkoisen virtuaalikytkimen. Virtuaalikoneita voi nimittää eri ulkoisille virtuaalikytkimille, mikä eristää ne muiden virtuaalikytkimien virtuaalikoneista. (Portnoy 2012, 174.)

Virtuaalikytkimet tarjoavat samat toimintaperiaatteet ja konfiguraatiomahdollisuudet esim. turvallisuus-, saatavuus- ja suorituskykyominaisuuksille kuin fyysiset verrokkinsa (Portnoy 2012, 177).

### 3.3.2 Tallennuksen valmistelu

Tallennussäilöjen sisällön tilan voi selvittää esim. Windows File System Resource Manager -työkalulla. Tallennustapoja on kolme: direct-attached (DAS), network-attached (NAS) ja storage area network (SAN). (Ruest & Ruest 2009, 70.) DAS on suoraan työasemaan tai palvelimeen kytketty tallennuslaite, NAS on tietoverkossa oleva tallennusjärjestelmä (esim. tiedostopalvelin), ja SAN on useiden NAS-tallennusjärjestelmien muodostama verkko. Toinen vaihe on selvittää, mitä tallennusmedioita voisi uudelleenkäyttää virtuaalikoneiden säilöntäpaikkana (Ruest & Ruest 2009, 70).

Kuvio 6 havainnollistaa tallennuspyynnön kulkua virtuaalikoneen sovellukselta tallennusohjaimelle (storage controller) tai verkkosovittimelle (NIC). Käyttöjärjestelmä vastaanottaa pyynnön ja lähettää sen eteenpäin tallennuksen sijainnista riippuen; DAS-laitteelle suuntautuvat pyynnöt ohjataan paikalliselle tallennusohjaimelle ja verkkoon suuntautuvat (NAS- ja SAN-järjestelmät) ohjataan verkkosovittimelle, joka ohjaa pyynnön edelleen verkossa sijaitsevaan järjestelmään. (Portnoy 2012, 152.)



Kuvio 6. Virtuaalitalennuksen toiminta (Portnoy 2012, 153).

Jaetut tallennusmenetelmät, kuten NAS ja SAN, varmistavat maksimaalisen saatavuuden ja toimintavarmuuden, sillä yhden isäntäkoneen hajoaminen ei vaikuta tiedontallennukseen. Uudet menetelmät mahdollistavat myös erillisten levyjen, kuten palvelimien sisäisten kiintolevyjen yhdistämisen jaetuksi resurssiksi, jota useat järjestelmät voivat käyttää. Näitä menetelmiä ovat esim. HP P4000 LeftHand SAN solutions ja VMware Virtual Storage Appliance, jotka luovat jaetun tallennusjärjestelmän jo olemassa olevista tallennusvälineistä. (Portnoy 2012, 156.)

Tallennus voidaan toteuttaa verkkotallennuksena TCP/IP-pohjaisten protokollien, kuten NFS:n tai iSCSI:n avulla. Verkkotallennus vaatii suuren kaistanleveyden ja joissain tapauksissa jopa erillisen tiedonsiirtoväylän. Erillinen tiedonsiirtoväylä edellyttää tallennukselle varatun fyysisen verkkokortin isäntäkoneessa, jolloin myös jokaisella virtuaalikoneella on erillinen virtuaaliverkkokortti tiedontallennuksen tarpeisiin. Virtuaalikoneille voi antaa erillisiä tallenneresursseja yhdistämällä erilliset fyysiset verkkokortit eri tallenneresursseihin. (Portnoy 2012, 176.)

Tallennuksen vikasietoisuuden ja sujuvuuden parantamiseksi on mahdollista käyttää useita eri tekniikoita, kuten tiedon peilausta tai lomitusta. Peilaus tarkoittaa levyn täydellistä peilausta toiselle levyille, jolloin toisen levyn rikkoutuminen ei vaikuta tiedon eheyteen. Kaksi peilattua levyä mahdollistavat myös molempien käytön samanaikaisesti, jolloin tiedonsiirto nopeutuu konfliktien puolittuessa. Lomitus tarkoittaa tiedostojärjestelmän sisällön ripottelemista usealle erilliselle fyysiselle levyille, jolloin levyt voivat käsitellä dataa yhteistyössä kasvattaen tallennuksen suorituskykyä merkittävästi. Peilauksen ja lomituksen voi myös yhdistää. (Portnoy 2012, 164.)

Tehokkaan tallennuksen varmistamiseksi käytetään myös tiedon kertatallennusta eli deduplikointia, jolloin sama tieto säilytetään vain yhtenä kopiona. Tämä säästää tilaa, sillä vaikka kymmenellä käyttäjällä olisi sama sähköpostin liitetiedosto tallennettuna kotihakemistoon, se vie vain yhden kopion verran tilaa massamuistista. Kertatallennus etsii identtisiä tietolohkoja massamuistista, ja löytäessään merkitsee alkuperäisen ja korvaa duplikaatit osoitetiedoilla alkuperäiseen tietolohkoon. Tietolohkot voivat olla pieniä tavun mittaisia säikeitä, isompia tietolohkareita tai kokonaisia tiedostoja. Deduplikointia käytetään varsinkin varmistusjärjestelmissä. (Hämäläinen 2009a; Portnoy 2012, 164.)

### 3.4 Virtuaalikoneiden luominen

Virtuaalikoneita voi luoda monella eri tavalla ja tämän otsikon alla selvitetään, miten virtuaalikone luodaan tyhjästä, kloonauksella ja templaatista. Lisäksi virtuaalikoneista voi luoda tilakaappauksia, joihin voi helposti palata ongelmien ilmentyessä esim. uusien päivitysten kohdalla.

#### **Uuden virtuaalikoneen luominen**

Uusien virtuaalikoneiden rakentamiseen ja konfigurointiin tarvitaan työalusta eli hypervisor. Tämän opinnäytetyön empiirisen osion alustana on käytetty 2-tyypin



KVM-hypervisorina, joka käyttää isäntäkäyttöjärjestelmänään Linuxia. Virtualisointialustan valinta riippuu esim. käytössä olevasta käyttöjärjestelmästä ja maksuvalmiudesta (Portnoy 2012, 55). Suosituimpia ilmaisia hypervisoreita esim. sovelluskehittäjien keskuudessa ovat mm. VMware Player, Virtual Box ja Microsoft Windows Virtual PC (Portnoy 2012, 54).

Hypervisor asennetaan työasemalle tai palvelimelle, jolla on tarvittava kapasiteetti virtuaalikoneiden luomiseksi. Uudelle virtuaalikoneelle määritellään käytettävät resurssit, kuten muisti, levytila ja verkkokortit. Paras käytäntö on luoda uusi hakemisto virtuaalikoneen nimellä, johon tallennetaan kaikki virtuaalikoneen tiedostot. (Ruest & Ruest 2009, 134.)

Virtuaalikone on vastine yksittäiselle fyysiselle palvelimelle, joten virtuaalikoneen luomisen jälkeen se on pelkkä säiliö, johon käyttöjärjestelmä ladataan erikseen (Portnoy 2012, 65).

### **Virtuaalikoneen kloonaus**

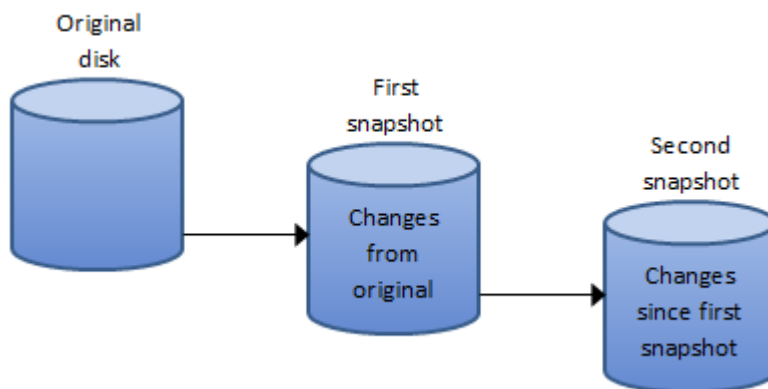
Uusi virtuaalikone voidaan kloonata jo olemassa olevasta virtuaalikoneesta. Kopion teon jälkeen uudelle virtuaalikoneelle tarvitsee muokata vain uniikit järjestelmätiedot, kuten järjestelmän nimi ja IP-osoite, minkä jälkeen se voidaan käynnistää ilman ongelmia. Hallintatyökalut helpottavat muutoksien tekemistä huomattavasti, jolloin koko prosessi voidaan suorittaa muutamalla hiiren painalluksella. (Portnoy 2012, 45.)

### **Templaattit**

Virtuaalikoneesta voi tehdä templaatin eli toimintakyvettömän virtuaalikoneen, joka toimii muottina uusille virtuaalikoneille. Templaatti luodaan rakentamalla puhdas ja toimiva virtuaalikone, joka konvertoidaan templaatiksi hypervisorin hallintatoiminnoilla tai muulla työkalulla. Templaatin avulla voidaan ottaa kloonauksen tapaan uusi virtuaalikone nopeasti käyttöön, jossa on jo valmiiksi esim. tarvittavat sovellukset ja asetukset. (Portnoy 2012, 46; 198.)

## Tilakaappaukset

Tilakaappaus (snapshot) tarkoittaa ominaisuutta, jossa virtuaalikoneen senhetkinen tila kaapataan tilannevedokseksi, johon voi helposti palata esim. muutoksien aiheuttaessa ongelmia virtuaalikoneeseen. Snapshot tallentaa virtuaalikoneen tilan, datan ja laitteistoasetukset. Tilakaappauksen jälkeen muutokset eivät tallennu enää alkuperäiseen levytiedostoon, vaan uudelle deltalevyille (ts. child disk), johon kertyy ensimmäisen snapshotin jälkeen tehdyt muutokset. Toisen snapshotin tapauksessa luodaan jälleen uusi deltalevy, johon kaikki sen jälkeiset muutokset tallentuvat. (Portnoy 2012, 47.) Kuvio 7 havainnollistaa alkuperäisen levyn ja deltalevyjen muodostamaa levyketjua.



Kuvio 7. Tilakaappausten muodostama levyketju (Portnoy 2012, 47).

Snapshotin muutosten yhteensovituksessa alkuperäisen virtuaalikoneen kanssa eli konsolidaatiossa deltalevyille kertyneet muutokset sulautetaan alkuperäiseen levytiedostoon. Virtuaalikoneen voi palauttaa tilakaappausta edeltäneeseen tilaan, jolloin kaikki siitä lähtien tehdyt muutokset raukeavat. Tilakaappaukset ovat käteviä etenkin testaus- ja kehitysympäristöissä, joissa voidaan testata esim. päivitysten toimivuutta tilakaappausten avustuksella. Tilakaappaukset eivät kuitenkaan korvaa kunnan varmuuskopioita. (Portnoy 2012, 48.)

### 3.5 Käyttöjärjestelmien asentaminen virtuaalikoneeseen

Iso määrä virtuaalikoneita ajaa käyttöjärjestelmänään Windowsia, ja tällä hetkellä myös Linux asennetaan yhä useammalle virtuaalikoneelle. Avoimen lähdekoodin Linux-jakeluversiot on otettu käyttöön monissa isoissakin konesaleissa lisenssimaksujen vähentämiseksi. Lisäksi on haluttu kasvattaa riippumattomuutta Microsoftiin. Tämän on mahdollistanut sovellusrajoitusten vähentyminen, sillä modernit sovellukset toimivat useilla käyttöjärjestelmillä. (Portnoy 2012, 97.)

Työpöytäkäyttöjärjestelmien kohdalla Windows omistaa markkinat täysin, mutta palvelinkäyttöjärjestelmien kohdalla tilanne on dynaamisempi (Portnoy 2012, 97). Windows on edelleen hallitseva käyttöjärjestelmä x86-arkkitehtuurin palvelimissa 48 prosentin markkinaosuudella liikevaihdosta laskettuna, mutta Linux on jatkanut osuutensa kasvattamista 22 prosenttiin. Unix-käyttöjärjestelmät edustavat 18 prosenttia palvelinliikevaihdosta. (IDC 2012.)

Windowsin, Linuxin tai minkä tahansa käyttöjärjestelmän asentamiseen tarvitaan asennusmedia (esim. .iso-tiedosto tai DVD-levy). Virtuaalisessa ympäristössä asennusmedia kannattaa muuntaa .iso-muotoiseksi tiedostoksi - ellei se sitä jo ole - jotta niiden käsitteleminen olisi helpompaa. .Iso-tiedostot käyttäytyvät kuin fyysiset CD- tai DVD-mediat virtuaalikoneissa. Käyttöjärjestelmän asennuksessa virtuaalikoneen CD/DVD-asema yhdistetään halutun käyttöjärjestelmän .iso-tiedostoon, minkä jälkeen asennus suoritetaan normaalien käytäntöjen mukaisesti. (Portnoy 2012, 72; Ruest & Ruest 2009, 134.)

## 4 KÄYTÄNNÖN TOTEUTUS KVM-HYPERVISORILLA

Opinnäytetyössä on toteutettu eksperimentaalinen virtualisointiprojekti, jonka alustana käytettiin Kernel-based Virtual Machine -hypervisorilla Ubuntu- ja Fedora-käyttöjärjestelmissä sekä käytetyimpiä hallintatyökaluja Virtual Machine Manageria (virt-manager) ja web-pohjaista oVirtia. Projektin tarkoituksena oli pystyttää kokeiluluontoinen virtuaalinen infrastruktuuri, jossa voi helposti luoda ja hallita virtuaalikoneita avoimen lähdekoodin työkaluilla sekä ajaa erimuotoisia palveluja, kuten virtuaalisia työpöytiä ja sovelluksia. Käytettävänä laitteistona toimi neliytimisellä Intelin prosessorilla varustettu PC, jolla voi laitteistoresurssien puolesta ajaa yhtäaikaisesti useita virtuaalikoneita.

Käytännön toteutuksen tukena käytettiin isäntäkoneen kanssa samassa lähiverkossa ollutta Windows 7 -kannettavaa, josta testattiin KVM:n etähallintaa ja virtualisoitujen palvelujen toimivuutta käyttäen mm. Google Chrome -selainta ja VirtualBox-hypervisorilla. Lisäksi live migration toteutettiin PC:n Windows 7 -käyttöjärjestelmällä virtualisoimalla kaksi KVM-isäntäpalvelinta ja NAS-palvelin VMware Workstation -hypervisorilla.

### 4.1 KVM:n ominaisuudet

KVM tukee useimpien tuotannossa käytettävien käyttöjärjestelmien virtualisointia. Toimivaksi testattuja ovat mm. Windowsin eri versiot (pl. 95 ja 98SE), Linuxin eri jakeluversiot, OpenBSD, FreeBSD ja useat muut UNIX-pohjaiset käyttöjärjestelmät. (KVM 2012c.)

KVM julkaistiin ensimmäisen kerran 4. helmikuuta 2007 Linux-kernel-versiossa 2.6.20. Näin ollen KVM-hypervisor on nuorempi kuin isot kilpailijansa VMware, VirtualBox, Xen ja Hyper-V. KVM on suuren osan olemassaoloajastaan yrittänyt ottaa kiinni vanhempia ja edistyneempiä kilpailijoitaan, mutta viime vuosien aikana se on itsenäistynyt ja tukee jo useita kehittyneitä ominaisuuksia. (Tozzi 2011.)

Näitä ominaisuuksia ovat muun muassa:

- tukee suorittimien virtuaalilaajennuksia (Intel VT ja AMD-V)
- tukee täyttä virtualisointia eli vieras käyttöjärjestelmä toimii muokkaamattomana
- käyttää Linuxin tietoturvamallia
- tukee kolmannen osapuolen hallintatyökaluja
- kehittyy Linuxin kanssa samaa tahtia eli tukee mm. kaikkia uusia ajureita ja tietojärjestelmiä
- tukee live migration -ominaisuutta
- tukee tilannekaappausten ottamista
- tukee USB-uudelleenohjausta, joka sallii virtuaalikoneelle pääsyn verkon yli USB-laitteeseen, joka on kytketty päätelaitteeseen. (KVM 2012d.)

KVM käyttää QEMU-emulaattoria, joka on geneerinen ja avoimen lähdekoodin kone-emulaattori ja virtualisoija. Emulaattorina toimiessaan QEMU voi ajaa tietyille laitteelle tehtyjä käyttöjärjestelmiä toisella laitteella, kuten esim. älypuhelimien käyttöjärjestelmän ajaminen omalla PC:llä. Dynaamisuuutensa ansiosta QEMU saavuttaa korkean suorituskyvyn. Virtualisoijana toimiessaan QEMU tarjoaa lähes natiivisuorituskyvyn, koska se suorittaa koodin suoraan isäntäkoneen suorittimella. Virtualisoija-tila vaatii x86-suoritinarkkitehtuurituen sekä isäntä- että vieraskoneessa. (OpenSiteSolutions 2008.)

### **Live migration**

KVM tukee live migration -teknologiaa myös AMD:n ja Intelin prosessorien välillä, jos suoritinarkkitehtuurit ovat yhteensopivia. 64-bittinen vieras voidaan siirtää vain 64-bittiseen isäntäkoneeseen. 32-bittisellä vieraalla ei ole rajoituksia, vaan se voidaan siirtää myös 32-bittisestä 64-bittiseen isäntään ja toisinpäin. Onnistuneen migraation jälkeen siirretty virtuaalikone jatkaa toimintaansa kohdeisäntäkoneessa (KVM 2012e.)

## Rajoitukset

KVM:llä on joitain rajoituksia, joita yritetään paikata tulevaisuudessa.

Eräitä merkittävimpiä rajoituksia ovat seuraavat:

- Vieraskäyttöjärjestelmistä Windows-versiot 95 ja 98 eivät asennu tai toimi (KVM 2012g).
- KVM ei ole yhteensopiva Xen-hypervisorin kanssa (IBM 2012b).
- Virtuaalisten prosessorien lisääminen lennosta virtuaalikoneelle ei onnistu kuin tietyillä laitteistokokoonpanoilla (KVM 2012g).

## 4.2 Valmistelu

### 4.2.1 Laiteresurssit

KVM:n täyden virtualisoinnin tukeen tarvitaan Intelin tai AMD:n prosessori tai prosessoreita, joissa on joko Intel VT- tai AMD-V-piirit. Kun Linux on asennettu, tuki voidaan tarkistaa selvittämällä, tuottaako komento `grep -E 'vmx|svm' /proc/cpuinfo` tulosteen.

vmx edustaa Intel VT -piiriä ja svm edustaa AMD-V:tä. Tulosteessa oleva KVM-flag auttaa oikean moduulin valitsemisessa, joten se kannattaa merkitä muistiin. (IBM 2012a.)

Lisäksi BIOS:sta pitää tarkastaa, onko virtualisointituki päällä. Esim. Intelin tapauksessa BIOS:n prosessoriasetuksissa pitäisi olla asetus "Intel Virtualization Technology" tilassa Enabled. (IBM 2012a.) Projektin PC:ssä olevassa Asuksen emolevyn BIOS:ssa asetus oli nimellä "Vanderpool Technology", joka asetettiin Enabled-tilaan.

## Projektissa käytetyn PC:n laitteistoresurssit

- Emolevy: Asus P5K
- Prosessori: Intel Dual Core 2 Quad Q6600 (Intel VT -teknologialla)
- Muisti: 4 GB DDR2 (2x2 GB Dual-Channel)
- Massamuisti: 500 GB EXT4-osio Samsung F4 EcoGreen 2 TB -kiintolevystä

Projektin PC:ssä oli käytössä 64-bittinen Windows 7 Professional -käyttöjärjestelmä, jonka rinnalle asennettiin 64-bittinen Ubuntu Server 12.04 LTS projektia varten tehdylle uudelle 500 GB:n kiintolevyosiolle. Myöhemmin osio puolitettiin ja toiselle puolikkaalle asennettiin Fedora.

## Live migration -resurssit

Projektin live migration -osiota varten tarvittiin kaksi KVM-hypervisoria ajavaa isäntäpalvelinta, joten puuttuvien laitteistoresurssien vuoksi ne jouduttiin virtualisoimaan, joka tehtiin monipuolisuuden maksimoimiseksi Windows 7 -käyttöjärjestelmässä käyttäen 2-tyypin VMware Workstation -hypervisoria. PC:n laitteistoresurssit olivat samat kiintolevyosiota lukuun ottamatta. Live migration -virtualisoinnin tallennustilana toimi saman kiintolevyn NTFS-osion vapaa tila, joka oli myös noin 500 GB.

Tässä ympäristössä hyödynnettiin ns. nested virtualization -menetelmää, jossa virtuaalikoneita ajetaan virtuaalikoneen sisällä tai useita hypervisoreita päällekkäin. Koska projektin PC:n prosessori ei sisältänyt nested virtualization -menetelmän edellyttämää toisen sukupolven virtualisointiteknologiaa (Intel EPT), live migration jouduttiin toteuttamaan ohjelmallisesti ilman KVM-laitteistotukea.

#### 4.2.2 Kiintolevyn osiointi ja Linuxin asentaminen

Windows 7 -käyttöjärjestelmää käyttäen Internetistä ladattiin avoimen lähdekoodin osiointisovellus GParted Live USB .iso-muodossa, joka asennettiin USB-muistitikulle käyttäen niin ikään avoimen lähdekoodin Tuxboot-sovellusta. Tuxboot osaa muuntaa USB-tikun niin, että sillä pystyy käynnistämään tietokoneen suoraan .iso-tiedoston sisältämään ohjelmaan, joka tässä tapauksessa oli GParted. GParted Live USB edellyttää noin 128 megatavua vapaata tilaa muistitikulla.

Ennen USB:ltä käynnistämistä BIOS-asetuksista tuli laittaa ensisijaiseksi käynnistyslaitteeksi USB. Kun tämä oli tehty, tietokone käynnistyi automaattisesti USB-asemassa olleeseen GParted-ohjelmaan. Ohjelmassa kohdekiintolevy oli nimellä `\dev\sdc1` ja se sisälsi vain yhden koko kapasiteetin kattavan osion, jonka koko muutettiin 1,34 teratavun kokoiseksi. Uusi ext4-osio luotiin vapautuneeseen 500 gigatavun tilaan.

Seuraava vaihe oli asentaa Ubuntu Server 12.04 uudelle ext4-osiolle niin ikään USB-tikulta.

Myös Ubuntu Server 12.04 -asennusmedia ladattiin netistä iso-muotoisena, jonka jälkeen se siirrettiin asennusta varten USB-tikulle [pendrivelinux.com](http://pendrivelinux.com)-sivuston ohjeiden mukaan käyttäen Universal USB Installer -ohjelmaa (Pendrivelinux 2012).

Kun USB-asennus lähti käyntiin, Ubuntu Server 12.04 asennettiin oletusasetuksilla ilman mitään ylimääräisiä paketteja, ml. virtualization host -paketti eli KVM, joka asennettiin myöhemmin.

Lisäksi luotiin 4 gigatavun Swap-osio 250 gigatavun root-osion rinnalle.

Koska Ubuntu Server-versiossa ei ole oletuksena käyttöliittymää vaan pelkkä konsoli, asennettiin päivitysten jälkeen GUI eli käyttöliittymä erikseen konsolilla komennoilla `sudo apt-get install ubuntu-desktop` ja `sudo reboot`.



### 4.2.3 Virtualisoitavat kohteet

Projektin tarkoituksena oli virtualisoida KVM-hypervisorilla käyttäen eri työpöytä- ja palvelinkäyttöjärjestelmiä sekä ohjelmia. Lisäksi tavoitteena oli luoda jaettu työpöytä käyttäen avoimen lähdekoodin Linux-ratkaisua.

KVM:n ominaisuuksista projektissa käytettiin virtuaalikoneen kloonausta ja kahden KVM-hypervisorin yhteistyötä kytkemällä ne yhteen virtuaalisen verkon yli käyttäen jaettua tallennusta. Yhteistyön tavoitteena oli virtuaalisesti toteuttaa ajossa olevan virtuaalikoneen live migration kahden virtuaalisen KVM-isäntäpalvelimen välillä.

Projektissa virtualisoitiin seuraavat palvelut:

- Windows XP Professional 32-bit
- Ensimmäisestä Windowsista tehty kloon
- Ulteo OVD Open Virtualization Desktop
- Linux Mint 14.1
- Ubuntu Desktop/Server 12.10
- Fedora 14
- Jaettu tallennus NFS-tekniikalla
- Live migration
- Slitaz
- Ubuntu LTSP Server 12.04 eli thin client -palvelin

### 4.2.4 Hallintatyökalut

KVM-hypervisorilla voi käyttää suoraan komentotulkista ja useilla eri graafisen käyttöliittymän sovelluksilla. Alunperin Xen-hypervisorille kehitetty graafinen Virtual Machine Manager (virt-manager) -sovellus on helppo vaihtoehto niille käyttäjille, jotka ovat aikaisemmin käyttäneet muita hypervisorinhallintaohjelmia (Hoffman 2012; Virt-manager 2012). Muut hallintatyökalut vaativat huomattavasti enemmän toimenpiteitä toimiakseen kunnolla, ja varsinkin ilman

suurempia hankaluuksia asennettavat web-käyttöliittymällä varustetut työkalut ovat edelleen harvassa. Tässä projektissa oli ensin tarkoitus käyttää pelkästään Ubuntu-käyttöjärjestelmää KVM:n alustana, mutta loppujen lopuksi Ubuntun rinnalle piti asentaa Fedora, koska web-työkaluista toimivimmat ja houkuttelevimmat toimivat ainoastaan sillä.

### 4.3 KVM-ympäristön luominen

KVM vaatii toimiakseen Linux-moduulin eli kernel-moduulin, joka koostuu kolmesta tiedostosta: `kvm.ko`, `kvm_intel.ko` (Intelin prosessoreille) ja `kvm_amd.ko` (AMD:n prosessoreille). Kernel-moduulit sisältyvät oletuksena Linuxiin kernel-versiosta 2.6.20 lähtien, joten niiden asentaminen erikseen on yleensä tarpeetonta. Useimmissa Linuxin jakeluversioissa moduulit ovat valmiina, ja niiden olemassaolosta voi varmistua seuraavalla komennolla:

```
modprobe -l | grep kvm
```

Tarvittaessa uusimmat KVM-kernel-moduulit ja käyttäjätilyökalut löytyvät sourceforge.net-sivustolta osoitteesta [http://sourceforge.net/project/showfiles.php?group\\_id=180599](http://sourceforge.net/project/showfiles.php?group_id=180599). Userspace-komponentteja ovat `qemu-kvm<versio>`, ja kernel-moduulit ovat `kvm-kmod<versio>`. Viimeisimmät versiot kummastakin voi ladata yhteisestä linkistä "Download qemu-kvm-<versio>.tar.gz". (KVM 2012b.)

#### 4.3.1 KVM:n asentaminen

Ubuntun versiosta 10.04 lähtien KVM-hypervisorin käyttöönottamiseksi tulee asentaa paketit:

- **qemu-kvm**, hypervisorin perusta
- **libvirt-bin**, eli libvirtd, jota tarvitaan qemu- ja kvm-instanssien hallintaan
- **ubuntu-vm-builder**, komentorivityökalu virtuaalikoneiden rakentamiseen
- **bridge-utils**, siltaus verkosta virtuaalikoneille. (Ubuntu 2012a.)

Lisäksi virtuaalikoneinstanssien näkemiseen ja graafisen hallintatyökalun asentamiseksi voidaan asentaa paketit:

- virt-viewer
- virt-manager. (Ubuntu 2012a.)

Projektin Ubuntu Server 12.04 -käyttöjärjestelmässä tarvittavien pakettien asennus suoritettiin komennolla `sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils virt-viewer virt-manager`.

Pakettien asennuksen jälkeen käyttäjä tulee lisätä ryhmään libvirtd, joka antaa oikeudet virtuaalikoneiden ajamiseen. Ubuntun versiosta 9.10 lähtien lisäys tapahtuu komennolla `sudo adduser käyttäjänimi libvirtd`.

Oikeuksien voimaantulo edellyttää uudelleen kirjautumisen. Uudelleen kirjautumisen jälkeen asennuksen onnistumisen voi testata komennolla `virsh -c qemu:///system list`, joka kaiken ollessa kohdallaan tulostaa tyhjän Id Name- ja State-taulukon. (Ubuntu 2012a.)

Fedora 17 -käyttöjärjestelmässä KVM:n asentaminen on yksinkertaisempaa ja tapahtuu komennolla `su -c "yum install @virtualization"`. Komento asentaa paketit qemu-kvm, python-virtinst, qemu, virt-manager ja virt-viewer sekä niiden riippuvuudet. Asennuksen jälkeen KVM käynnistetään komennolla `su -c "systemctl start libvirtd"`. (Fedora 2013.)

#### 4.3.2 Verkon asetukset

KVM-virtuaalikoneiden verkkoliikenne on yhtä kuin QEMU-emulaattorin verkkoliikenne, joten QEMU:n laajaa dokumentaatiota voi käyttää hyödyksi (KVM 2012f).

Debian-pohjaisten Linux-käyttöjärjestelmien (esim. Ubuntu) verkkoasetukset ovat määriteltyinä sijainnissa `/etc/network/interfaces`. Verkkoasetuksiin pitää koskea esim. siinä tapauksessa, että virtuaalikoneille halutaan NAT:n sijaan määrittää ulkoiset IP-osoitteet eli siltaavat yhteydet. (Dedoimedo 2011.)

## Julkinen siltaus

Julkisessa siltauksessa virtuaalikoneille annetaan IP-osoite, jolla ne tavoittaa lähiverkosta. Virtuaalikoneiden virtuaalisille verkkoliitännöille annetaan pääsy ulkoverkkoon fyysisen liitännän kautta, jolloin ne näkyvät normaaleina isäntinä muulle verkolle. Siltauksen määrittelemiseksi seuraavat komennot tulee olla asennettuina järjestelmään:

- `/sbin/ip`
- `/usr/sbin/brctl`
- `/usr/sbin/tunctl`. (KVM 2012f, Dedoimedo 2011.)

Ennen verkkoasetusten muokkaamista on hyvä sammuttaa verkko komennolla `sudo invoke-rc.d networking stop`. Etäyhteydellä tämä tietenkin on mahdotonta, jolloin verkko jätetään päälle ja käynnistetään uudelleen asetusten muokkauksen jälkeen. (Ubuntu 2012b.)

Julkinen siltaus otetaan käyttöön lisäämällä silta-määrittys verkkoasetustiedostoon, joka on Ubuntussa `/etc/network/interfaces`. Tiedoston avaaminen muokkausta varten tehtiin `gedit`-komennolla `sudo gedit /etc/network/interfaces`.

Tiedostoon lisättiin `br0`- eli siltasovitin kirjoittamalla lähiverkkokohtaiset siltausmäärittymät olemassa olevien `eth0`- ja `lo`-määritysten perään (Kuva 1).

```
auto br0
iface br0 inet static
    address 192.168.100.11
    network 192.168.100.0
    netmask 255.255.255.0
    broadcast 192.168.100.255
    gateway 192.168.100.1
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
    dns-nameservers 192.168.100.1
```

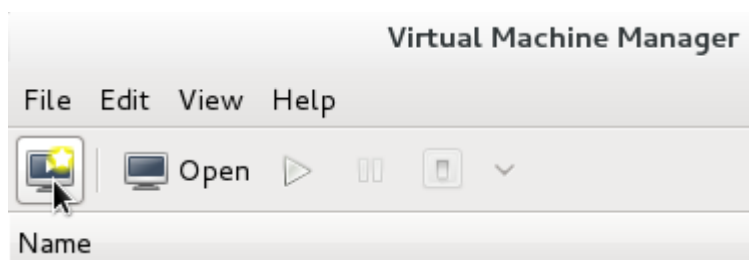
Kuva 1. Siltasovittimen (`br0`) lisäys `/etc/network/interfaces`-verkkoasetustiedostoon.

Tallennuksen jälkeen verkko käynnistettiin uudelleen komennolla `sudo /etc/init.d/networking restart`, jonka jälkeen uusi br0-sovitin näkyi `ifconfig`-komennon tulosteessa. Jotta uusi br0-sovitin saatiin näkyviin myös virt-manager-hallintatyökaluun, isäntäkäyttöjärjestelmä piti käynnistää uudelleen.

#### 4.3.3 Virtuaalikoneen luominen Virt-manager-työkalulla

Uuden virtuaalikoneen luominen Virt-managerilla on helppoa. Ohjatun virtuaalikoneen luomisen yhteydessä ohjelma ehdottaa käyttöjärjestelmän asentamiseksi asennusmediaa, joka voi sijaita joko määrättyssä verkkosijainnissa, CD-levyllä tai .iso-muotoisessa tiedostossa. Asennusmedian sijaan virtuaalikoneelle voi valita myös verkkokäynnistyksen. Virt-manager ei tällä hetkellä tunnista asennusmedian käyttöjärjestelmää automaattisesti, kuten esim. VMware Workstation, vaan se pitää valita käsin annetuista vaihtoehdoista.

Virt-manager käynnistetään terminaalista komennolla `virt-manager` tai ohjelmavalikosta, jossa se on nimellä Virtual Machine Manager. Uusi virtuaalikone luodaan käynnistämällä ohjattu virtuaalikoneen luominen `Create a new virtual machine` -painikkeesta (Kuva 2).



Kuva 2. Create a new virtual machine -painike.

Avautuvassa ohjatussa luomisikkunassa on viisi vaihetta:

1. Virtuaalikoneen nimi ja asennustapa, joista yleisin on paikallinen asennusmedia. Muita vaihtoehtoja ovat asennus verkosta, verkkokäynnistys ja olemassa olevan levykuvan tuonti.

2. Asennusmedian ja käyttöjärjestelmätyypin- ja version määrittely.
3. Muistin ja prosessoriytimien määrittely.
4. Tallennuksen määrittely. Tässä vaiheessa voi luoda uuden levykuvan virtuaalikoneelle tai käyttää jo olemassa olevaa levykuvaa. Uuden levykuvan voi asettaa käyttämään koko määritelty tila heti tai sitä mukaa, kun dataa kertyy (Allocate entire disk now -asetus). Tallennuslaitteen voi myös jättää pois kokonaan.
5. Viimeisessä vaiheessa näkee yhteenvedon valituista asetuksista sekä voi halutessaan määrittää lisäasetuksia. Lisäasetuksia ovat verkkolaitteen valinta, MAC-osoitteen muokkaus, virtualisointityyppi (KVM tai QEMU-emulaatio) sekä arkkitehtuuri. Ruksi kohdassa "Customize configuration before install" aukaisee vielä lisäikkunan tämän vaiheen jälkeen, jossa voi kustomoida virtuaalikoneen laitteistoa lisää.

Viimeisen vaiheen jälkeen virtuaalikone lähtee käyntiin määritetyllä asennusmedialla ja käyttöjärjestelmän voi asentaa normaalilla tavalla. Tässä vaiheessa virt-managerin pääikkunaan on ilmestynyt uusi virtuaalikone määritetyllä nimellä Running-tilassa. Virtuaalikoneinstanssi tulee automaattisesti näkyviin virt-viewerin kautta ja sen saa näkyviin manuaalisesti myöhemmin valitsemalla valikosta Open. Virtuaalikoneen laitteistoa pääsee muokkaamaan myöhemmin virtuaalikoneinstanssin työkalurivin Info/Details-painikkeella.

#### 4.3.4 Thin client -palvelimen asennus ja käyttöönotto

Tässä luvussa on dokumentoitu Thin Client -palvelimen asennus ja käyttöönotto KVM-ympäristössä. Fyysisen thin client -päätelaitteen puuttuessa kyseinen laite piti virtualisoida, johon käytettiin VirtualBox-hypervisorina.

Thin client -palvelimeksi valittiin avoimen lähdekoodin Ubuntu 12.04 Alternate, joka asennettiin suoraan virtualisoiduksi thin client -palvelimeksi (Kuva 4). Linuxin thin client -ratkaisu on koodinimeltään LTSP eli Linux Terminal Server Project.



Kuva 3. Thin client -palvelimen (LTSP server) asennus KVM-virtuaalikoneeksi.

Palvelin on oletuksena konfiguroitu kahdelle verkkokortille, joten asennuksen jälkeen palvelin piti konfiguroida yhdelle verkkokortille. Tämä tehtiin määrittämällä palvelimelle kiinteä IP-osoite 192.168.100.21 ja muokkaamalla asetustiedoston `/etc/ltsp/dhcpd.conf` arvot vastaamaan lähiverkon määrittämiä (Kuva 4).

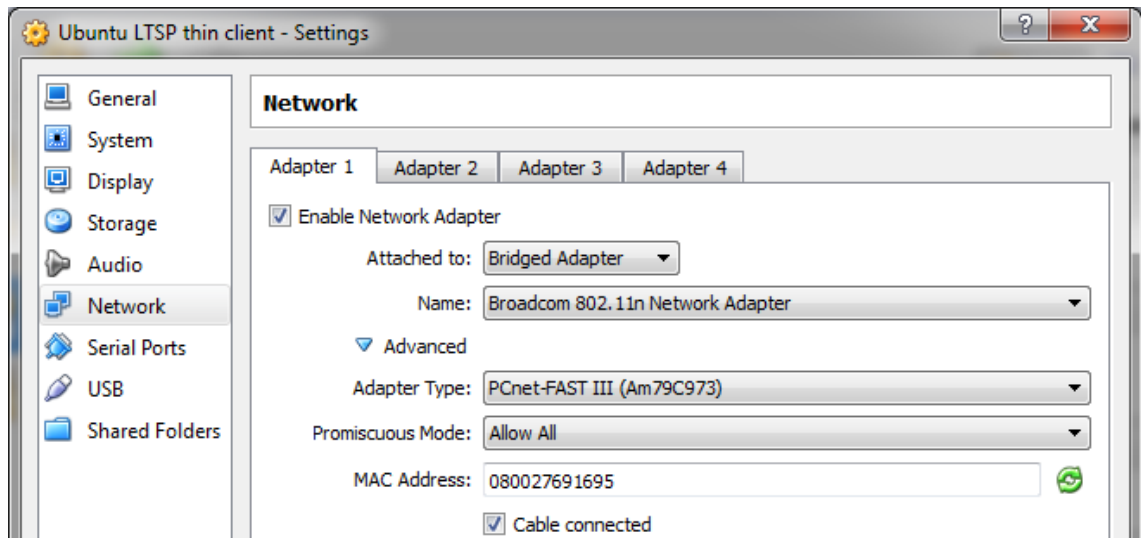
```

subnet 192.168.100.0 netmask 255.255.255.0 {
    range 192.168.100.200 192.168.0.253;
    option domain-name "example.com";
    option domain-name-servers 192.168.100.1;
    option broadcast-address 192.168.100.255;
    option routers 192.168.100.1;
}

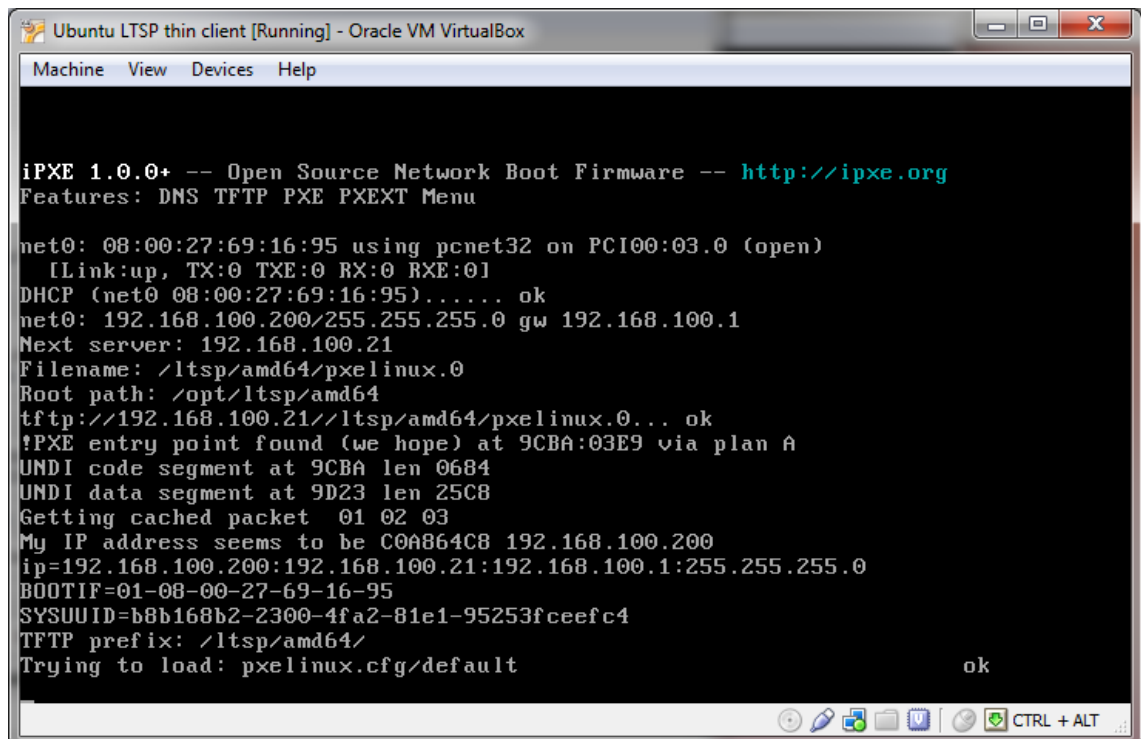
```

Kuva 4. `/etc/ltsp/dhcpd.conf`-tiedoston lähiverkkokohtaiset rivit ja arvot lihavoituna punaisella.

VirtualBoxilla luotiin tallennuslaitteeton virtuaalikone, jolle määriteltiin verkkokortiksi PCnet-FAST III, joka soveltuu verkkokäynnistykseen. Käynnistyslaitteeksi valittiin System-valikosta Network. Kuvassa 5 näkyy VirtualBoxilla virtualisoidun thin client -päätelaitteen verkkosovittimen asetukset. Näiden toimenpiteiden jälkeen thin client -käynnistys tehtiin onnistuneesti lähiverkon yli VirtualBoxilla virtualisoidulla päätelaitteella (Kuvat 6 ja 7).

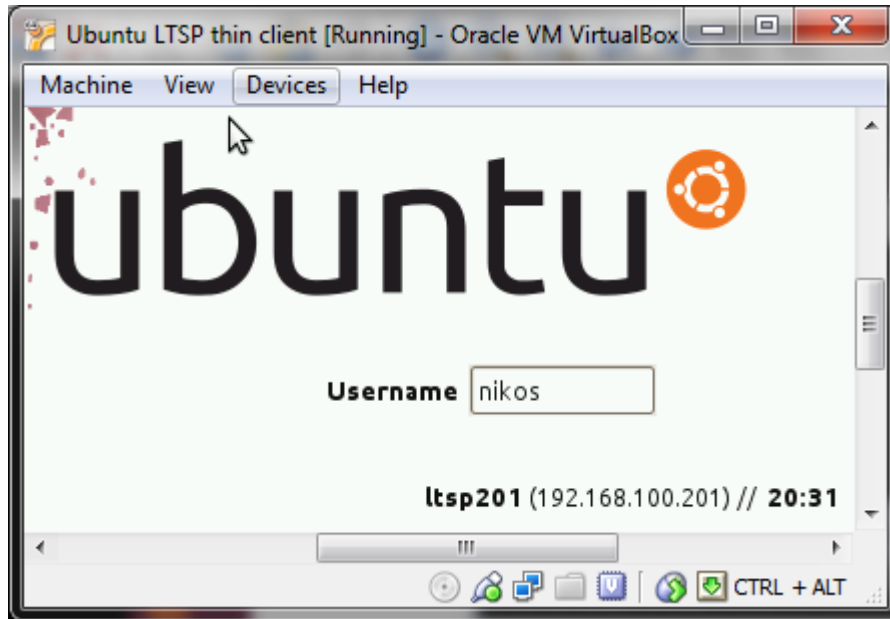


Kuva 5. Päätelaitte virtualisoitiin VirtualBoxilla.



Kuva 6. Päätelaitte käynnistymässä palvelimelta 192.168.100.21.





Kuva 7. Päätelaitteelle verkosta käynnistynyt Ubuntu-käyttöjärjestelmä.

Kuvassa 7 näkyy päätelaitteelle käynnistyneen henkilökohtaisen instanssin nimi ltsp201 ja IP-osoite 192.168.100.201.

#### 4.4 KVM-ympäristössä toimiminen

##### 4.4.1 KVM:n hallitseminen oVirt-hallintatyökalulla

oVirt-hallintatyökalun helppo asennus edellyttää Fedora 17 -käyttöjärjestelmää. Muilla käyttöjärjestelmillä asentaminen vaatii esim. rakentamista lähdekoodista, joten tässä käytännön projektissa päätettiin asentaa Fedora 17 oVirtin käyttöä varten Ubuntu-käyttöjärjestelmän rinnalle. Fedoraa varten Ubuntu sisältävä 500 gigatavun osio jaettiin puoliksi GParted-ohjelmalla, jonka jälkeen toiselle 250 gigatavun osiolle asennettiin Fedora 17.

oVirtin oletusasennuspaketti asentaa vain hallinta-alustan tietokoneelle olettaen, että varsinaiset KVM-isäntäpalvelimet ovat erillisiä hallintatietokoneesta, jotka linkitetään myöhemmin hallinta-alustaan. oVirt-hallintaympäristöön kuuluvat isäntäpalvelimet sisältävät VDSM-taustaprosessin, joiden avulla oVirt hallinnoi niitä (OVirt 2012). Yhden palvelimen

projektiympäristössä VDSM-taustaprosessi piti saada samalle palvelimelle kuin hallinta-alusta, johon on olemassa myös oma asennuspaketti. Kyseinen paketti on oVirt All-in-one, joka asentaa hallinta-alustan lisäksi myös VDSM-taustaprosessin samalle tietokoneelle.

oVirt All-in-one-paketin asennus ei ollut niin yksinkertaista kuin ohje antoi ymmärtää, vaan asennuksen aikana piti selvittää muutamia ongelmia. Kävi ilmi, että oVirt all-in-one -pika-asennusohje oli tehty Fedora 17:n minimiasennuksella, ja käytetyssä Desktop-versiossa on palveluita, jotka estivät asennuksen sujumisen ilman kriittisiä virheitä:

- **NetworkManager**, palvelu tulee ottaa pois päältä ja käytöstä komennolla `sudo systemctl stop NetworkManager.service && sudo systemctl disable NetworkManager.service`
- **sshd**, palvelu tulee ottaa käyttöön ja aktivoida komennolla `sudo systemctl start sshd && sudo systemctl enable sshd`. (Brooks 2012.)

Virheisiin päättyneet asennukset saatiin nollattua komennolla `engine-cleanup`. Kokonaisuudessaan asennuksen kulku oli seuraava:

1. Julkaisupaketin asennus: `yum install http://www.ovirt.org/releases/ovirt-release-fedora.noarch.rpm`
2. oVirt All-in-one -paketin asennus: `yum install ovirt-engine-setup-plugin-allinone`
3. Engine-setup, jonka kysymyksiin pohjautuen oVirtin moottori rakentuu (Kuva 8): `engine-setup`
4. Jos oVirtin moottorin rakennus (engine-setup) keskeytyy virheeseen (Kuva 9), tulee asennus nollata komennolla `engine-cleanup`, jonka jälkeen voi aloittaa uudelleen vaiheesta 3.
5. Asennus on valmis ja web-käyttöliittymä toimii.

```

root@virtualization-host-2:~
File Edit View Search Terminal Help
found under /usr/share/ovirt-engine/conf/iptables.example
Configure iptables ? ['yes'| 'no']: yes
Configure VDSM on this host? ['yes'| 'no'] [yes] :
Local storage domain path: /mnt/data
Confirm root password :

oVirt Engine will be installed using the following configuration:
=====
override-httpd-config:      yes
http-port:                  80
https-port:                 443
host-fqdn:                  virtualization-host-2.Elisa
auth-pass:                  *****
org-name:                   Testing
default-dc-type:           NFS
db-remote-install:         local
db-local-pass:             *****
nfs-mp:                     /mnt/iso
iso-domain-name:           iso
override-iptables:         yes
config-allinone:           yes
storage-path:              /mnt/data
superuser-pass:            *****
Proceed with the configuration listed above? (yes|no): █

```

Kuva 8. Vastaukset annettuna oVirtin engine-setupin kysymyksiin.

```

Handling HTTPD... [ DONE ]
AIO: Creating storage directory... [ DONE ]
AIO: Adding Local Datacenter and cluster... [ DONE ]
AIO: Adding Local host (This may take several minutes)... [ ERROR ]
Error: Could not install local host
Please check log file /var/log/ovirt-engine/engine-setup_2012_12_11_01_22_27.log
for more information

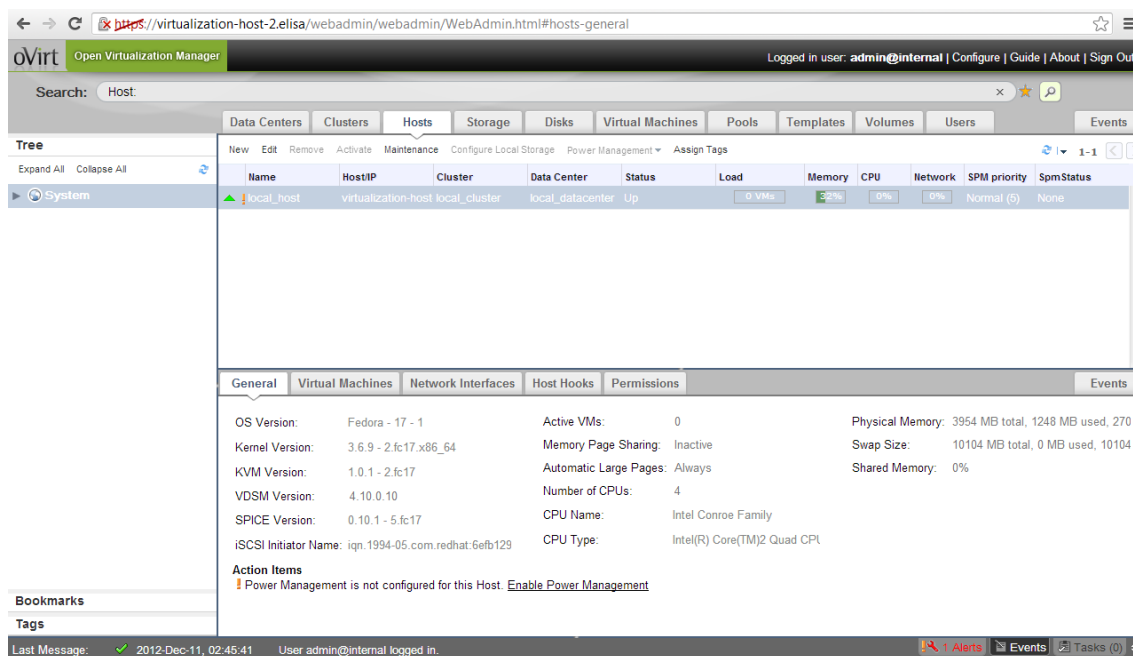
```

Kuva 9. Virheeseen keskeytynyt oVirt-enginen asennus.

Asennuksen jälkeen virtuaalikoneiden .iso-muotoiset asennusmediat voidaan siirtää engine-setupissa määriteltyyn nfs-mp-sijaintiin, jossa ne ovat oVirtin käytettävissä. Tässä tapauksessa nfs-mp:ksi määriteltiin sijainti /mnt/iso, jonne siirrettiin Ubuntu 12.04- ja Fedora 14 -asennusmediat. Asennusmediat eivät kuulu polun juureen, vaan sinne automaattisesti luotuun alihakemistoon, joka on muotoa /UUID-koodi/images/11111111-1111-1111-111111111111/. Siirron jälkeen kuluu vain muutamia sekunteja, kunnes asennusmediat ovat web-käyttöliittymän käytettävissä. (Brooks 2012.)

## Web-käyttöliittymä

oVirtin web-käyttöliittymään saa yhteyden isäntäpalvelimen IP-osoitteella tai asennuksen yhteydessä määritetyllä domain-nimellä. Projektissa yhteys otettiin verkon yli kannettavasta Google Chrome -selaimella IP-osoitteella 192.168.100.11 ja paikallisesti domain-nimellä localhost. Kumpikin kääntyi automaattisesti määriteltyksi domain-nimeksi virtualization-host-2.Elisa. Kirjautumisruudussa valitaan ”Administrator Portal” ja syötetään engine-setupin yhteydessä syötetyt tunnukset.



Kuva 10. oVirtin käyttöliittymä.

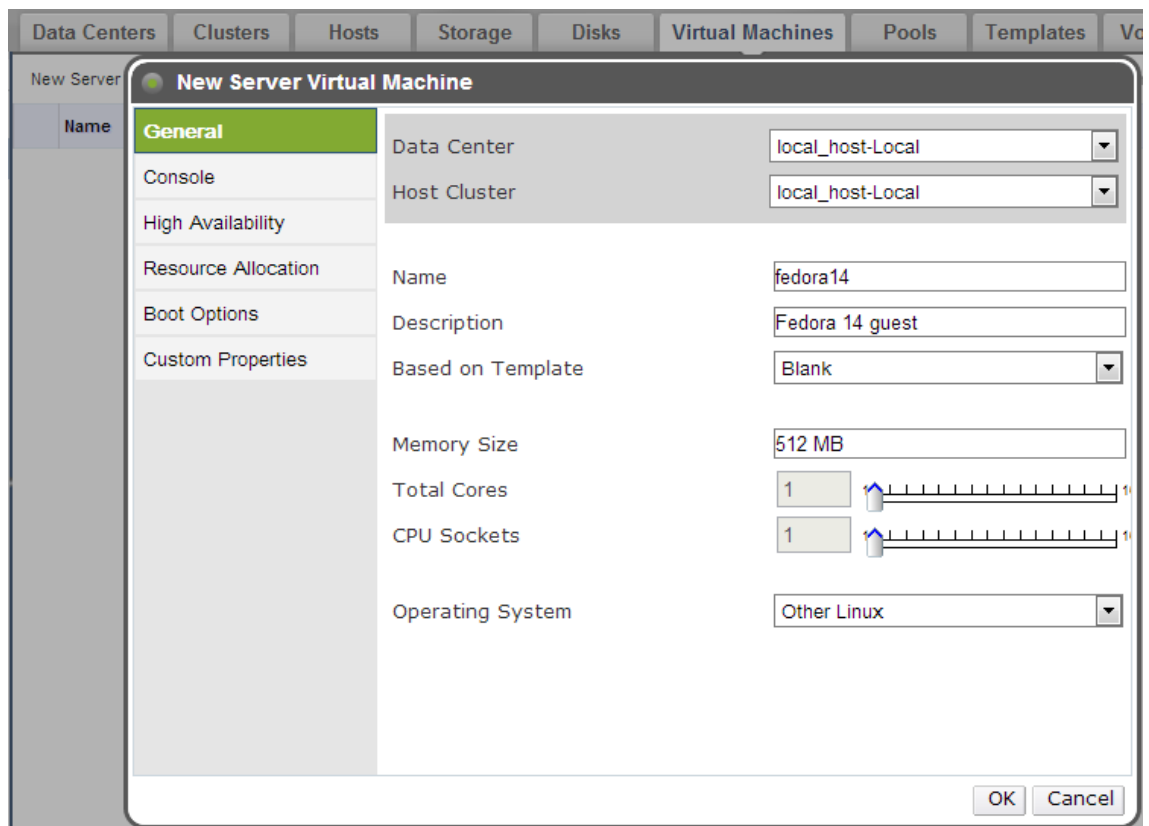
Ensimmäisen kirjautumisen jälkeen Hosts-välilehdellä näkyy isäntäpalvelin local\_host, joka saattaa siinä vaiheessa vielä asentua. Vihreä kolmio isännän vieressä tarkoittaa, että palvelin on käyttövalmis (Kuva 10). Sen jälkeen asetettiin isäntä ylläpitotilaan Maintenance-linkistä, jonka jälkeen voitiin konfiguroida tallennus:

1. Hosts-välilehden Configure Local Storage -linkistä asetettiin poluksi sama storage-path kuin engine-setupissa eli /mnt/storage

- Storage-välilehdeltä valittiin iso-domain, jonka jälkeen klikattiin alhaalle ilmestyvistä välilehdistä Data Center, josta kiinnitettiin iso-domain Attach-linkistä. Kun kiinnitys oli valmis, domain aktivoitiin Activate-linkistä.

Storage-välilehdellä iso-domain oli nyt iso ja tallennusdomain local\_host-Local.

Tallennuksen määrittelyn jälkeen palvelin toimii sekä oVirt-hallintapalvelimena että isäntänä virtuaalikoneille (Brooks 2012). Virtuaalikoneita luodaan ja hallitaan Virtual Machines -välilehdellä (Kuva 11).

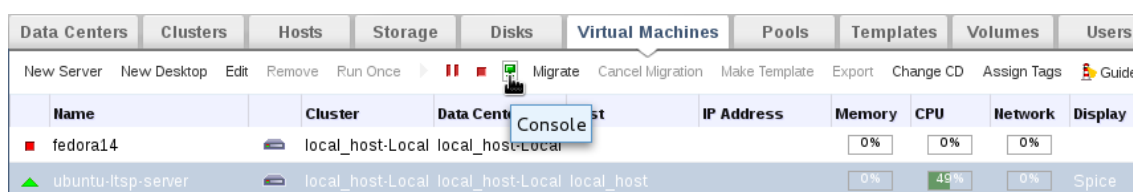


Kuva 11. Uuden virtuaalikoneen luominen oVirtilla.

Uusi virtuaalikone luodaan linkistä New Server tai New Desktop. Projektissa luotiin New Server Fedora 14 -käyttöjärjestelmällä (Kuva 11). Luomisikkunassa General-välilehti on ainoa, johon syötettiin tietoja. Nimen, muistin ja prosessorien määritysten jälkeen painetaan OK, jonka jälkeen avautuu ikkuna, jossa konfiguroidaan virtuaaliset verkkokortit ja levyt. Verkkokortti lisättiin oletusasetuksilla painamalla OK, ja kiintolevyksi valittiin Internal eli sisäinen levy

20 gigatavun kokoisena. Konfiguroinnin jälkeen ikkunasta poistuttiin Configure Later -painikkeella. Uusi virtuaalikone fedora14 ilmestyi Virtual Machines -listaan ja käynnistettiin Run Once -linkistä, josta avautui uusi ikkuna käynnistysvaihtoehtoineen. Kyseisestä ruudusta valittiin Fedora 14:n asennusmedia kohdasta "Attach CD". Tämän jälkeen OK:ta painamalla virtuaalikone käynnistyi.

Linuxilla käytettäessä virtuaalikoneinstansseja voi katsella spice- tai vnc-protokollilla Console-kuvalinkistä (Kuva 12). Projektissa käytettiin spice-xpilisäosaa, joka asennettiin isäntäpalvelimelle komennolla *yum install spice-xpi*.



Kuva 12. Virtual Machines -lista ja Console-painike, josta instansseja pääsee katselemaan.

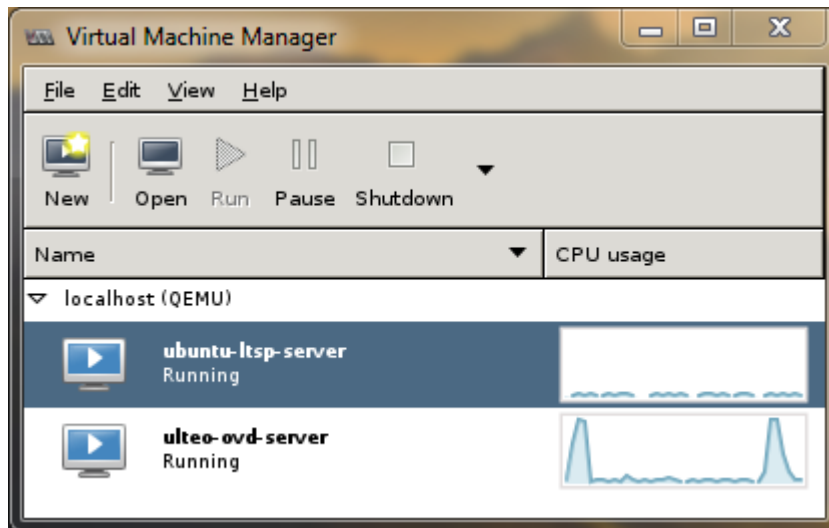
Valitettavasti oVirtin muistinkäyttö on valtava, joten useiden virtuaalikoneiden ajaminen samanaikaisesti neljän gigatavun muistilla oli mahdotonta. Pelkästään oVirtin ajaminen ilman virtuaalikoneita kuluttaa n. kolme gigatavua. Ubuntu LTSP -palvelimen asentaminen ja ajaminen kuluttivat jo kaikki muistiresurssit.

#### 4.4.2 Virtuaalikoneiden hallitseminen Windowsista

KVM:n virtuaalikoneita on mahdollista hallita Windowsin kautta web-pohjaisten hallintakäyttöliittymien avulla ja muodostamalla etäyhteyden virt-manager-työkaluun. oVirtin avulla virtuaalikoneita voi esim. lisätä, poistaa, sammuttaa ja käynnistää, mutta instanssien katseleminen eli etähallinta ei vielä onnistu Windowsista käsin. Virtuaalikoneiden etähallintaan Windowsista projekti löysi yhden helposti lähestyttävän ratkaisun eli etäyhteyden ottamisen virt-manager-työkaluun. Tätä varten Windowsiin tulee asentaa X Window Server -yhteysohjelma ja SSH client (Mattoon 2011). Projektissa käytettiin Xming X

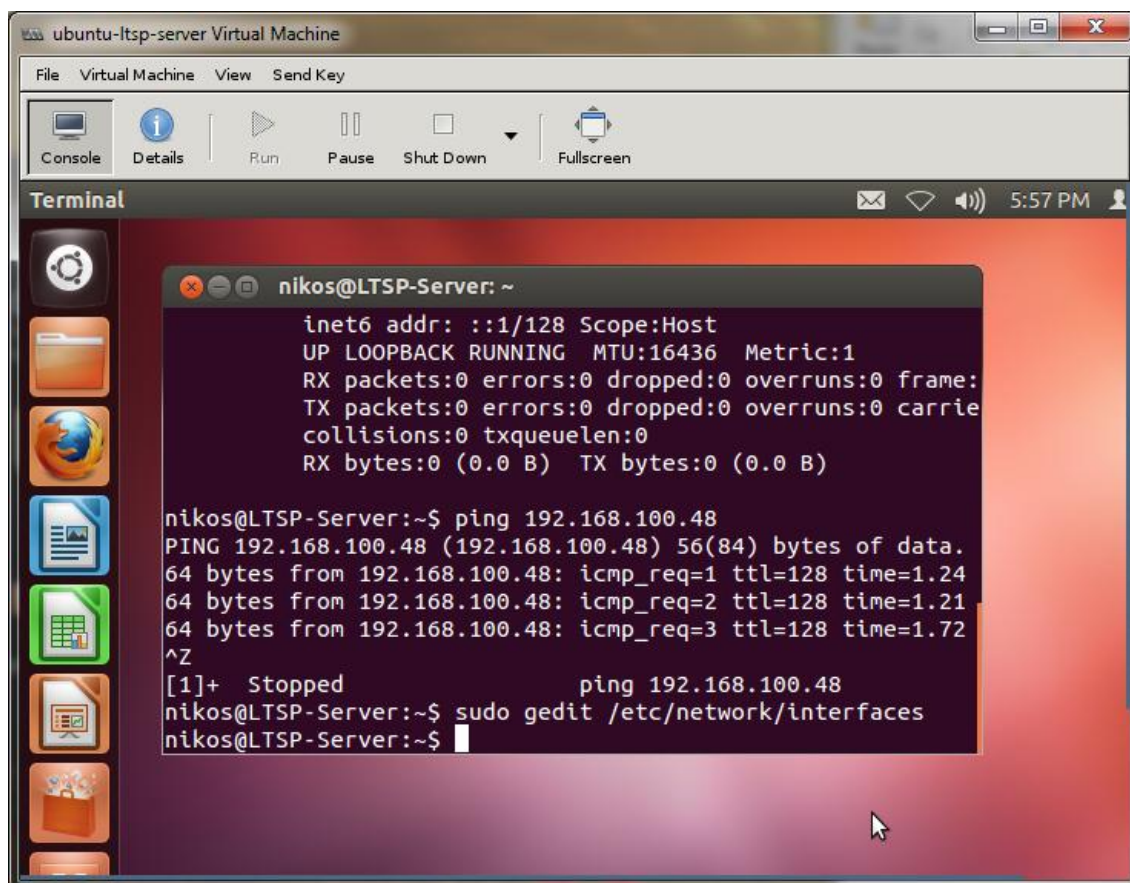
Server for Windows -ohjelmaa, joka sisältää molemmat vaaditut komponentit. KVM-isäntäpalvelimella riittää ssh-palvelimen lisäksi, että käytettävä ohjelma on asennettuna (esim. virt-manager ja virt-viewer) (Mattoon 2011).

Kun Xming on asennettu ja käynnissä, Putty-ohjelmassa asetetaan SSH-yhteyden asetuksista päälle asetus ”X11 Forwarding”, jonka jälkeen palvelimeen voi muodostaa SSH-yhteyden. Yhdistetyssä terminaali-ikkunassa haluttu ohjelma käynnistetään kuten paikallisestikin. Projektin tapauksessa virt-manager käynnistettiin komennolla *virt-manager*, jonka jälkeen virt-manager avautui Xming-ikkunassa, jossa sitä voi hallita normaalisti sekä etähallita virtuaalikoneinstansseja aivan kuten paikallisesti KVM-isäntäkoneellakin (Kuvat 13 ja 14).



Kuva 13. Virt-manager etänä Windowsista käsin SSH- ja X window -yhteydellä.

Kuvassa 14 näkyy virt-manager-ikkunasta käynnistetty virt-viewer-etähallinta ubuntu-itsp-server-virtuaalikoneelle Windowsissa.



Kuva 14. Virt-viewer-etähallinta.

#### 4.4.3 Jaettu tallennus ja live migration

Live migration -operaation toteuttaminen edellyttää vähintään kahta isäntäpalvelinta, jotka käyttävät jaettua tallennusta. Virtuaalinen jaettu tallennus toteutettiin virtualisoimalla Linux-pohjainen avoimen lähdekoodin Openfiler-käyttöjärjestelmä, joka osaa toimia isäntänä NAS- ja SAN-tallennusratkaisuille. Openfiler-käyttöjärjestelmään sisältyy helppokäyttöinen web-pohjainen hallintasovellus, jolla määritellään tallennuksien ominaisuudet ja käyttöoikeudet. Isäntäpalvelimet virtualisoitiin kahden Fedora 14 -virtuaalikoneen muodossa, joihin asennettiin KVM, virt-manager ja virt-viewer. Live migrationia varten asennettiin yksi virtuaalikone kummallekin isäntäpalvelimelle, joiden käyttöjärjestelminä toimi pieni Spitzaz.



Jaetun tallennuksen ja live migrationin toteuttamisen apuna on käytetty sivuston [www.everythingvm.com](http://www.everythingvm.com) ohjeita Openfilerin konfiguroimiseen sekä isäntäpalvelimien yhdistämiseen jaettuun tallennusjärjestelmään ja toisiinsa.

### **Openfilerin konfigurointi live migrationia varten**

Openfiler asennettiin VMware Workstation -hypervisorilla virtuaalikoneena, jolle annettiin 1024 megatavua muistia, kaksi suoritinydintä ja kaksi SCSI-kiintolevyä (15 gigatavua käyttöjärjestelmälle ja 80 gigatavua datalle).

OpenFiler-virtuaalikone konfiguroitiin NAS-palvelimeksi kahden KVM-isäntäpalvelimen käyttöön. Konfigurointi tehtiin mahdollisimman yksinkertaisesti ilman autentikointia sekä käyttäjätilien- ja ryhmien määrittämiä.

Pääsyn hallinta asetettiin System-välilehdeltä kohdasta Network Access Configuration kattamaan vain KVM-isäntäpalvelimien IP-osoitteet 192.168.100.21 ja 192.168.100.22.

### **Volumen ja verkkojaon luominen data-osiolle**

Uusi volume eli looginen levy luotiin Volumes-välilehdeltä:

1. Create new Physical Volume.
2. Valittiin /dev/sdb-osio eli datalle luotu 80 Gt:n osio (Kuva 15).
3. Volumen kooksi määriteltiin 80 Gt (Starting cylinder 1 ja Ending cylinder 10443), jonka jälkeen se luotiin Create-painikkeella.
4. Seuraavaksi luotiin volume-ryhmä Volume Groups -linkistä; nimeksi annettiin volgroup1 ja ruksitettiin ryhmälle lisättäväksi volumeksi kohdassa 3 luotu /dev/sdb1, jonka jälkeen ryhmä lisättiin Add volume group -painikkeella.
5. Ryhmän luonnin jälkeen luotiin looginen volume eli osio Add Volume -linkistä; nimeksi annettiin volume1 ja kooksi 40476 Mt eli reilu puolet

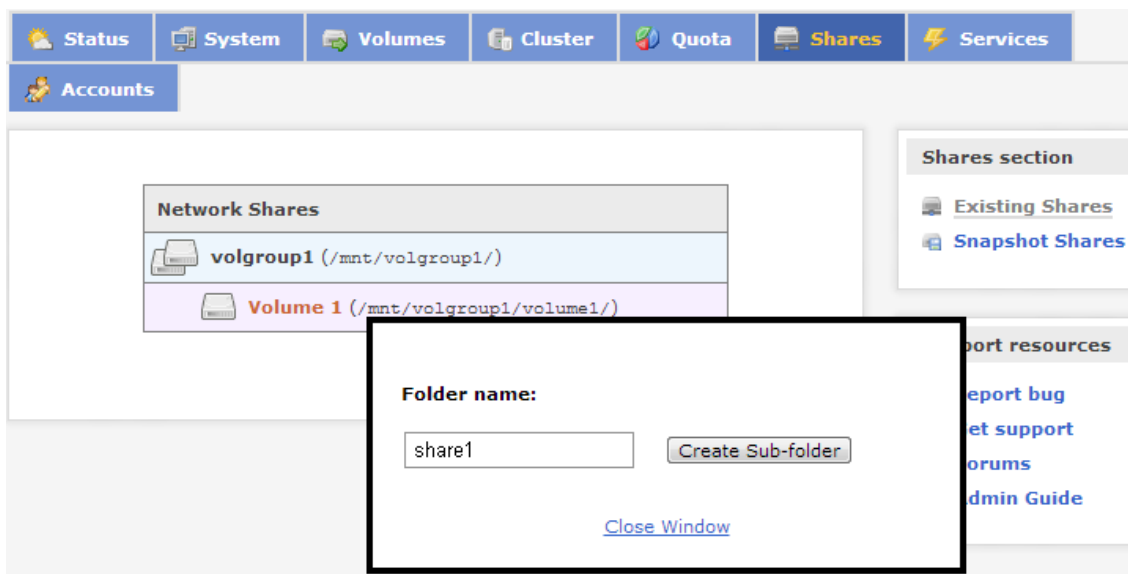
koko osiosta. Tiedostojärjestelmäksi valittiin Ext3. Luominen viimeisteltiin Create-painikkeella.



Kuva 15. Virtualisoidut levyt tiedostopalvelimella Volumes-välilehdellä.

Verkkojaon luominen:

1. Services-välilehdeltä aktivoitiin NFS-verkkojakopalvelu.
2. Share-välilehdeltä klikattiin volume1-osiota ja kirjoitettiin jaettavan hakemiston nimeksi share1, jonka jälkeen painettiin Create sub folder (Kuva 16).
3. Verkkojaon luominen viimeisteltiin painamalla share1-kansiolinkkiä ja avautuvasta valikosta "make share".
4. Share1-jaon sivulla Share Access Control Mode eli pääsyn hallinta asetettiin tlaan Public guest access ja painettiin Update.
5. Alempana Host access configuration -taulukossa annettiin NFS-protokollalle RW-oikeudet. RW-ruksin vieressä painettiin Edit ja avautuvasta valikosta UID/GID Mapping:ksi valittiin no\_root\_squash. Muutokset tallennettiin painamalla Update.



Kuva 16. Verkkojaon luominen tiedostopalvelimella.

## Isäntäpalvelimen yhdistäminen NFS-verkkojakoon

Fedora 14 -käyttöjärjestelmät yhdistettiin OpenFilerin NFS-verkkojakoon luomalla niille ensin hakemisto, johon jako linkitetään komennolla `mkdir /var/lib/libvirt/images/kvmshared`.

Tämän jälkeen varsinainen linkitys luotuun hakemistoon tehtiin komennolla `mount -t nfs 192.168.100.34:/mnt/volgroup1/volume1/share1 /var/lib/libvirt/images/kvmshared`. Jotta linkitys olisi pysyvä myös uudelleen käynnistyksen jälkeen, lisättiin seuraava rivi tiedostoon `/etc/fstab`:

```
192.168.100.34:/mnt/volgroup1/volume1/share1/
/var/lib/libvirt/images/kvmshared nfs
```

```
[nikos@Fedora14host ~]$ su -
Password:
[root@Fedora14host ~]# mkdir /var/lib/libvirt/images/kvmshared
[root@Fedora14host ~]# mount -t nfs 192.168.100.34:/mnt/volgroup1/volume1/share1
/ /var/lib/libvirt/images/kvmshared
[root@Fedora14host ~]# gedit /etc/fstab
```

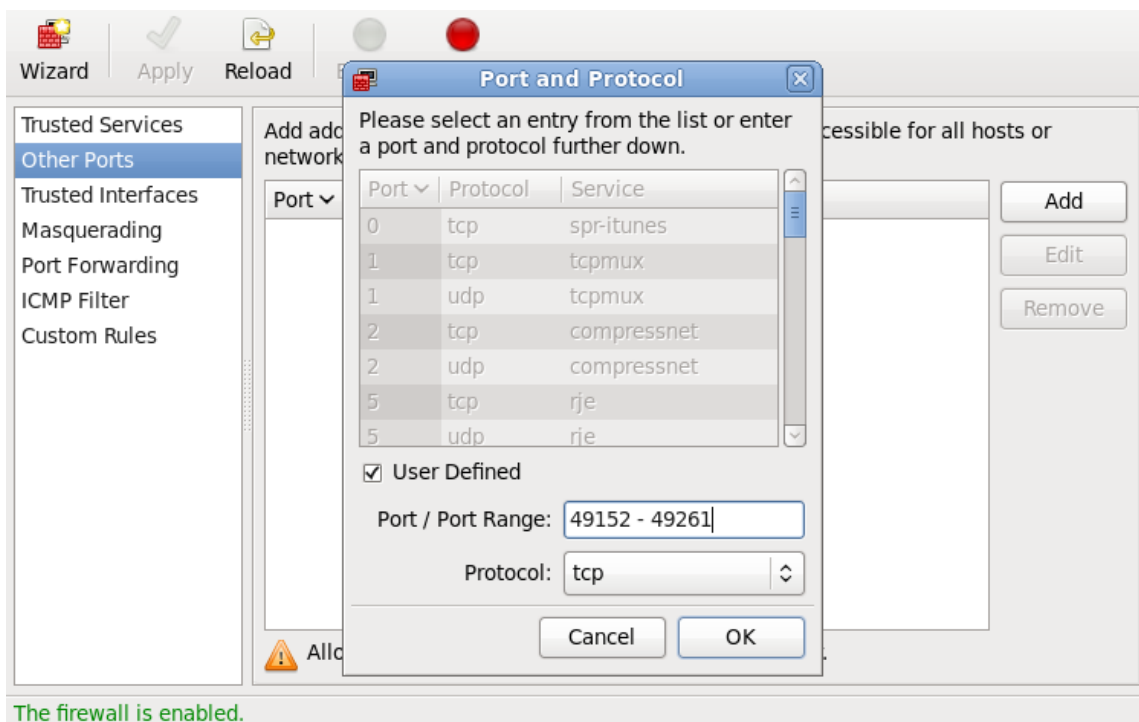
Kuva 17. Isäntäpalvelimen yhdistäminen NFS-verkkojakoon.

Lopuksi kvmshared-hakemiston käyttöoikeudet asetettiin kuntoon komennolla `chmod 755 -R var/lib/libvirt/images`, joka antaa images-hakemistolle ja sen alihakemistoille luku- ja suoritusoikeudet kaikille sekä lisäksi kirjoitusoikeudet omistajalle.

## Palomuurin asetukset

Etäyhteyden sallimiseksi isäntäpalvelimilta piti avata portti 22 SSH:ta varten ja portit 49152-49261 virtuaalikoneiden live migrationin sallimiseksi.

Palomuurin käyttöliittymä avattiin komennolla `/usr/bin/system-config-firewall`, jonka kautta portit avattiin polusta Other Ports - Add (Kuva 18).

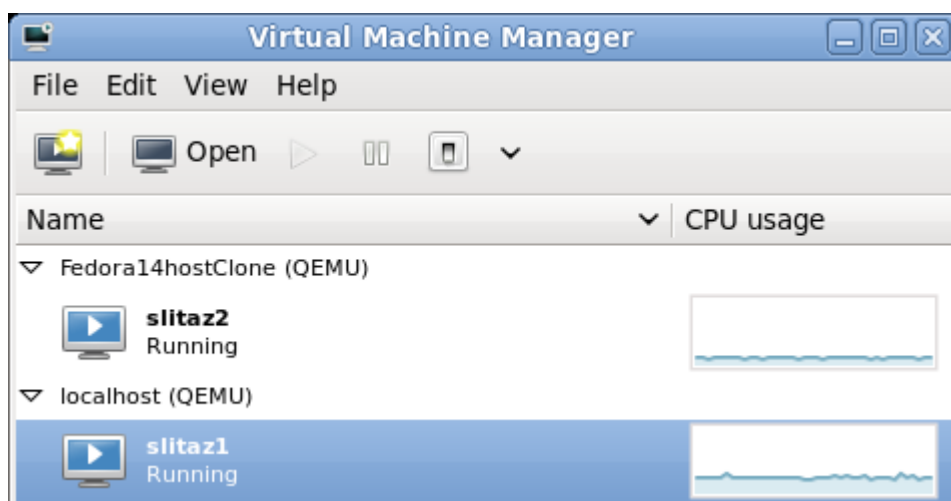


Kuva 18. Palomuurin konfigurointi live migrationia varten.

Yhteyteen tarvittava SSH-palvelin asennettiin kummallekin palvelimelle komennolla `yum install openssh-server` ja käynnistettiin komennolla `/sbin/service sshd start`.

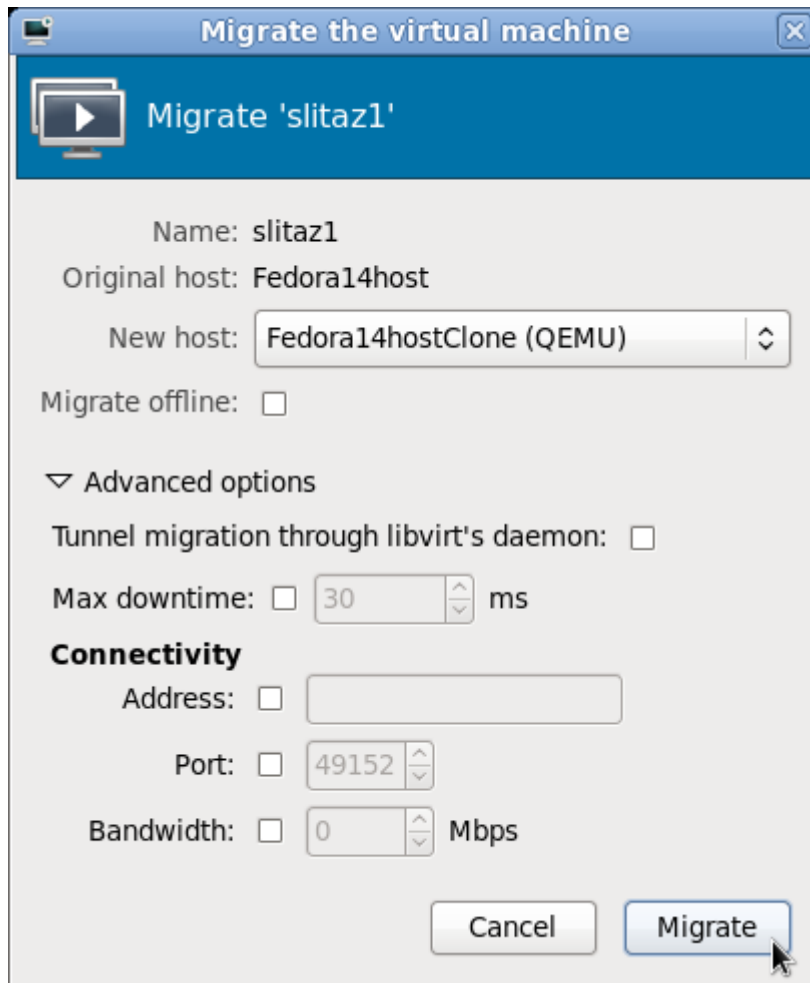
## Live migrationin suorittaminen virt-manager-työkalulla

Toisen isäntäpalvelimen virt-manager pitää ensin yhdistää toiselle palvelimelle pääikkunan valikosta File, josta valitaan Add connection. Avautuvassa Add Connection -valikossa valittiin hypervisoriksi QEMU/KVM ja hostname-kenttään kirjoitettiin toisen isäntäpalvelimen nimi Fedora14hostClone (kenttään voi kirjoittaa myös IP-osoitteen). SSH-yhteys kysyy salasanaa, johon syötettiin root-tunnuksen salasana. Yhteyden muodostamisen jälkeen Fedora14hostClone virtuaalikoneineen ilmestyi virt-managerin virtuaalikonelistaukseen (Kuva 19).

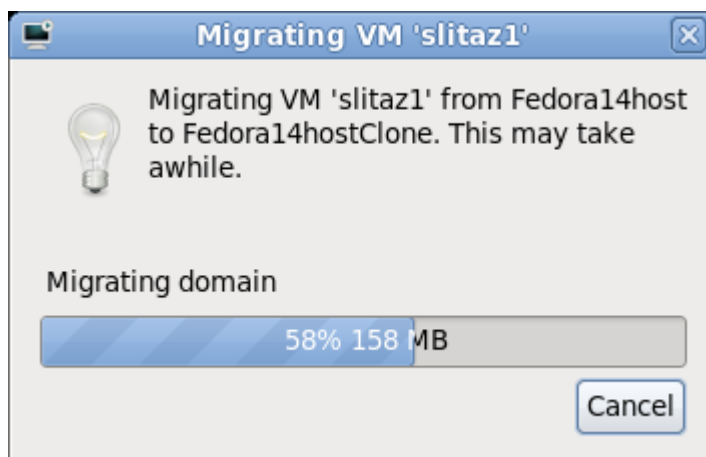


Kuva 19. Virt-manager toiseen isäntäkoneeseen yhdistämisen jälkeen.

Tämän jälkeen virtuaalikoneita voi siirtää isäntäpalvelimelta toiselle klikkaamalla virtuaalikonetta hiiren oikealla näppäimellä ja valitsemalla Migrate... sekä sammutettujen että päällä olevien virtuaalikoneiden tapauksessa. Migrate-valikosta voi valita uuden isännän ja mm. käytettävän kaistanleveyden ja portin (Kuva 20). Tässä tapauksessa live migration toteutettiin oletusasetuksilla.

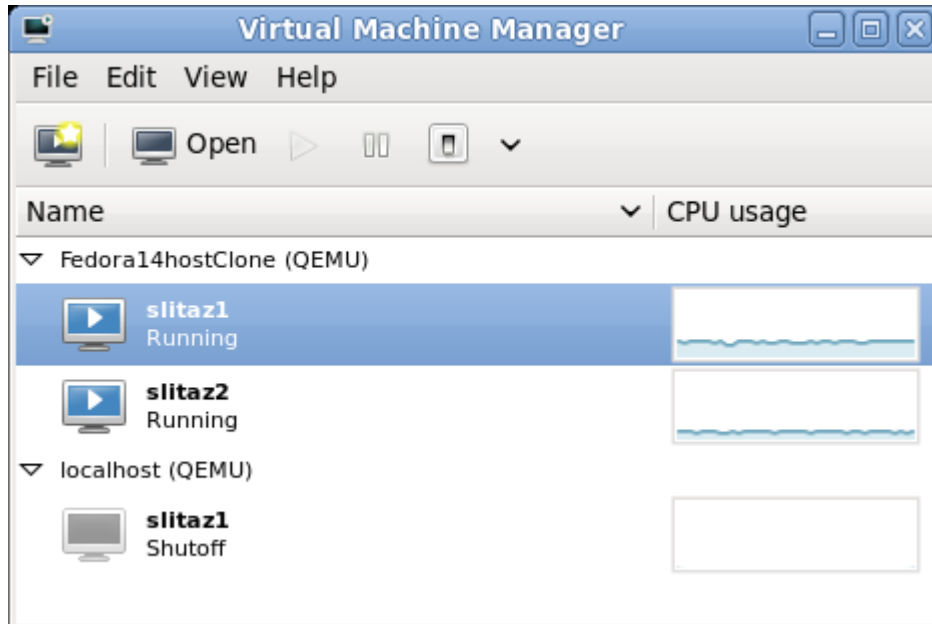


Kuva 20. Virt-managerin Live migration -asetukset.



Kuva 21. Slitaz1-virtuaalikoneen live migration toiselle isäntäpalvelimelle.

Live migrationin valmistuttua virt-managerissa näkyi, kuinka ajossa oleva slitaz1-virtuaalikone oli ilmestynyt Fedora14hostClone-isäntäpalvelimelle (Kuva 22). Vanha slitaz1 paikallisella isäntäpalvelimella ei ollut enää aktiivinen, joten sen voi poistaa.



Kuva 22. Slitaz1-virtuaalikoneen live migration suoritettu.

Siirron jälkeen nähtiin, että toisella isännällä oleva slitaz1-instanssi oli samassa tilassa kuin ennen siirtoa. Tilanne testattiin Sudokulla, jota pääsi jatkamaan samasta tilanteesta.

## 5 YHTEENVETO JA JOHTOPÄÄTÖKSET

Opinnäytetyössä käsiteltiin virtualisointia yleisesti ja tutkittiin lopuksi tietyn virtualisointialustan eli KVM-hypervisorin kanssa virtualisoinnin toteuttamista käytännössä. Aihetta lähestyttiin pääasiassa teknisestä näkökulmasta, joka antaa käytännönläheisen perustietämyksen virtualisoinnista yleisesti sekä spesifisesti KVM-hypervisorilla.

Kaksiosaisessa teoriaosuudessa tutustuttiin ensin virtualisointiin yleisesti käsittelemällä sen perusteita, teknillisiä ominaisuuksia, suosituimpia virtualisointialustoja sekä lopuksi virtualisointiympäristön toteuttamista teoriatasolla. Lisäksi on kartoitettu virtualisointiin siirtymisestä koituvia mahdollisia hyötyjä ja haittoja. Teoriaosuus pyrkii antamaan perustietämyksen virtualisoinnin eri osa-alueista sekä valmistamaan lukijaa ymmärtämään työn empiiristä osiota.

Opinnäytetyön empiirisessä osiossa tarkasteltiin KVM-hypervisoria eksperimentaalisen projektin avulla, jossa käsiteltiin sen ominaisuuksia, laitteistovaatimuksia, asennusta, käyttöönottoa sekä käyttöä virtualisoijana. Valmistavat toimenpiteet, asennus, käyttöönotto ja käyttö on pyritty kuvaamaan helposti ymmärrettävässä muodossa, jolloin lukija pystyy hahmottamaan, mitä on tehty, miten se on tehty sekä missä järjestyksessä.

Työn tekeminen oli erittäin opettava kokemus, koska virtualisointi ei ollut ennestään kovin tuttu aihe. Teoria antoi erinomaisen perustietämyksen ja auttoi ymmärtämään käytännön osa-alueet sekä toimenpiteet paremmin. Varsinkin eksperimentaalisen projektin myötä tuli paljon kokemusta erilaisista haasteista, joita virtualisointi KVM-hypervisorin kanssa saattaa tuoda tullessaan. KVM:n asentamisesta, käyttöönotosta sekä hallitsemisesta saatiin arvokasta kokemusta. Myös Linuxista tuli opittua paljon uusia asioita sen toimiessa KVM:n isäntäkäyttöjärjestelmänä. Lisäksi esim. Linuxin Thin Client -palvelinta virtualisoidessa tuli tutuksi asioita palvelimien asentamisesta sekä konfiguroinnista.



Työssä kului eniten aikaa käytännön toteutuksessa, jossa piti puutteellisen Linux-osaamisen vuoksi opetella paljon virtualisoinnin ulkopuolisia asioita. Lisäksi virtualisoinnin toteuttaminen ei sujunut ilman suunnittelematonta ongelmanratkaisua lähes missään työn vaiheessa. Toinen vaikeus työssä oli KVM-lähdemateriaalin löytäminen; virtualisoinnista yleisesti löytyi todella paljon tietoa, mutta KVM:stä selkeän ja yhtenäisen tietopaketin löytäminen tuntui olevan kiven alla.

Uuden aiheen mukanaan tuomiin yllättäviin vastoinkäymisiin oli vain sopeuduttava niiden ilmentyessä. Jälkikäteen ajateltuna dokumentaation ja käyttäjäkokemusten kattavampi etukäteistarkastelu olisi saattanut välttää esim. joitain ilmenneitä yhteensopivuusongelmia. Kokonaisuudessaan koetut haasteet antoivat kuitenkin hyvät valmiudet ratkaista ongelmia ja tulkita sekä hakea vaikeamminkin löydettävää tietoa virtualisoinnin parissa.

Linuxin kerneliin integroitu avoimen lähdekoodin KVM-hypervisor tarjoaa monipuolisen ja varsin kilpailukykyisen moottorin virtuaaliselle ympäristölle. Se sopii niin aloittelijoille kuin edistyneemmillekin käyttäjille ja tarjoaa edellytykset raskaaseen kaupalliseen käyttöön. KVM:n asentaminen ja sujuva peruskäyttö edellyttää jonkin verran perustietämystä Linuxissa toimimisesta. Vaativammassa käytössä Linuxista on hyvä olla edistynyt tietämys, koska muutoin esimerkiksi hallintatyökalujen asentaminen ja käyttöönotto voi olla työssä saavutettujen kokemusten perusteella hankalaa. Onneksi lähes kaikkiin ongelmatilanteisiin on olemassa jo valmis ratkaisu, kunhan sen vain löytää.

KVM:lle on tehty paljon dokumentaatiota, mutta varsinkin edistyneimmät niistä olettavat keskivertoa parempaa Linux-tietämystä. Lisäksi dokumentaatio on jonkin verran vielä keskeneräistä, koska KVM on melko tuore tekijä hypervisoreiden keskuudessa. Tuoreuden seurauksena myös sen ominaisuudet ovat vielä joiltain osin puutteellisia ja niitä lisätään jatkuvasti vastaamaan sekä ylittämään kilpailevat ratkaisut.

## LÄHTEET

Brooks, J. 2012. Up and Running with oVirt, 3.1 Edition. Jebpages 10.8.2012. Viitattu 17.12.2012 <http://blog.jebpages.com/archives/up-and-running-with-ovirt-3-1-edition/>.

CA Technologies 2012. Virtualization Best Practices. Viitattu 2.2.2013 [http://support.ca.com/phpdocs/0/common/impcd/r11/virtualization/virt\\_main.htm](http://support.ca.com/phpdocs/0/common/impcd/r11/virtualization/virt_main.htm).

Citrix 2012. XenServer features by edition. Viitattu 5.11.2012 <http://www.citrix.com/products/xenserver/features/editions.html>.

Dedoimedo 2011. How to setup bridged networking in KVM. Viitattu 19.11.2012 <http://www.dedoimedo.com/computers/kvm-bridged.html>.

Dinesh 2011. Disadvantages of Virtualization, What's Your Opinion? Viitattu 20.12.2012 <http://www.sysprobs.com/disadvantages-virtualization-opinion>.

Fedora 2013. Getting started with virtualization. Viitattu 5.2.2013 [https://fedoraproject.org/wiki/Getting\\_started\\_with\\_virtualization](https://fedoraproject.org/wiki/Getting_started_with_virtualization).

Hoffman, C. 2012. How to Install KVM and Create Virtual Machines on Ubuntu. How-To Geek 28.6.2012. Viitattu 11.11.2012 <http://www.howtogeek.com/117635/how-to-install-kvm-and-create-virtual-machines-on-ubuntu/>.

Hämäläinen, P. 2009. Kaikki virtualisoinnista, osa 4/4. Tietokone 9/2009. Viitattu 24.10.2012 [http://www.tietokone.fi/lehti/tietokone\\_5\\_2009/kaikki\\_virtualisoinnista\\_osa\\_4\\_4\\_7775](http://www.tietokone.fi/lehti/tietokone_5_2009/kaikki_virtualisoinnista_osa_4_4_7775).

IBM 2012a. Enabling KVM support on your hardware. Viitattu 14.11.2012 <http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/topic/liaai/kvminstall/liaaikvminstalle.html>.

IBM 2012b. Installing KVM on the host system. Viitattu 14.11.2012 <http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/topic/liaai/kvminstall/liaaikvminstallafter.html>.

IDC 2012. Worldwide Server Market Revenues 28.8.2012. Viitattu 8.11.2012 <http://www.idc.com/getdoc.jsp?containerId=prUS23665812>.

Knorr, E. 2010. What desktop virtualization really means. Infoworld 2010. Viitattu 25.10.2012 <http://www.infoworld.com/d/virtualization/what-desktop-virtualization-really-means-352>.

KVM 2012a. Choose the right KVM kernel version. Viitattu 6.11.2012 [http://www.linux-kvm.org/page/Choose\\_the\\_right\\_kvm\\_%26\\_kernel\\_version](http://www.linux-kvm.org/page/Choose_the_right_kvm_%26_kernel_version).

KVM 2012b. Downloads. Viitattu 6.11.2012 <http://www.linux-kvm.org/page/Downloads>.

KVM 2012c. Guest Support Status. Viitattu 14.11.2012 [http://www.linux-kvm.org/page/Guest\\_Support\\_Status](http://www.linux-kvm.org/page/Guest_Support_Status).

KVM 2012d. KVM-Features. Viitattu 14.11.2012 [http://www.linux-kvm.org/page/KVM\\_Features](http://www.linux-kvm.org/page/KVM_Features).

KVM 2012e. Migration. Viitattu 14.11.2012 <http://www.linux-kvm.org/page/Migration>.

KVM 2012f. Networking. Viitattu 14.11.2012 <http://www.linux-kvm.org/page/Networking>.

KVM 2012g. ToDo. Viitattu 5.12.2012 <http://www.linux-kvm.org/page/TODO>.

Mattoon, M. 2011. Linux-KVM: Managing KVM Guests Using Virt-Manager on Windows. It from all angles 21.3.2011. Viitattu 17.12.2012 <http://blog.allanglesit.com/2011/03/linux-kvm-managing-kvm-guests-using-virt-manager-on-windows>.

Merriam-Webster 2013. Cloud computing. Viitattu 5.2.2013 <http://www.merriam-webster.com/dictionary/cloud%20computing>.

NI Developer Zone 2009. Virtualization Basics. Viitattu 25.10.2012 <http://www.ni.com/white-paper/8708/en>.

OpenSiteSolutions 2008. Emulation / Virtualization. Viitattu 11.11.2012 <http://www.opensitesolutions.com/services/virtualization.php>.

Pendrivelinux 2012. Put Ubuntu on Flash Drive using Windows. Viitattu 11.11.2012 <http://www.pendrivelinux.com/put-ubuntu-10-04-on-flash-drive-using-windows/>.

Portnoy, M. 2012. Virtualization Essentials. Indianapolis: John Wiley & Sons.

Ruest, D. & Ruest, N. 2009. Virtualization: A Beginner's Guide. New York: McGraw-Hill. Pitkänen, J. 2012. Ilmaista virtualisointia moneen käyttöön. Tietokone 19.9.2012. Viitattu 2.11.2012 [http://www.tietokone.fi/softa/windows/oracle\\_virtualbox\\_4\\_2](http://www.tietokone.fi/softa/windows/oracle_virtualbox_4_2).

Skyline 2011. The Disadvantages of Virtualization. Viitattu 20.12.2012 <http://goarticles.com/article/The-Disadvantages-of-Virtualization/5534048/>.

Tamminen, T. 2011. VirtualBox 4.1. MBnet 29.7.2011. Viitattu 2.11.2012 [http://www.mbnet.fi/artikkeli/ajankohtaiset/viikon\\_ohjelmat/virtualbox\\_4\\_1?ref=lk\\_tk\\_2](http://www.mbnet.fi/artikkeli/ajankohtaiset/viikon_ohjelmat/virtualbox_4_1?ref=lk_tk_2).

Tozzi C. 2011. KVM Virtualization: Ready for the Desktop? The VAR Guy 3.8.2011. Viitattu 11.11.2012 <http://www.thevarguy.com/2011/08/03/kvm-virtualization-ready-for-the-desktop/>.

- Ubuntu 2012a. KVM/Installation. Viitattu 12.11.2012  
<https://help.ubuntu.com/community/KVM/Installation>.
- Ubuntu 2012b. KVM/Networking. Viitattu 12.11.2012  
<https://help.ubuntu.com/community/KVM/Networking>.
- Virt-manager 2012. Manage virtual machines. Viitattu 11.11.2012 <http://virt-manager.org/>.
- Virtuatopia 2012. An Overview of Virtualization Techniques. Virtuatopia. Viitattu 5.11.2012  
[http://www.virtuatopia.com/index.php/An\\_Overview\\_of\\_Virtualization\\_Techniques](http://www.virtuatopia.com/index.php/An_Overview_of_Virtualization_Techniques).
- Webopedia 2012. Virtualization. Viitattu 28.1.2013  
<http://www.webopedia.com/TERM/V/virtualization.html>.
- OVirt 2012. Wikipedia. Viitattu 11.12.2012 <http://en.wikipedia.org/wiki/OVirt>
- Xen 2012. Xen Overview. The Xen Wiki. Viitattu 4.11.2012  
[http://wiki.xen.org/wiki/Xen\\_Overview](http://wiki.xen.org/wiki/Xen_Overview).