Antti Lappalainen

Docker-säiliöt Openstack-pohjaisessa yksityispilvessä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

31.3.2015

Metropolia

Tekijä(t) Otsikko Sivumäärä Aika	Antti Lappalainen Docker-säiliöt Openstack-pohjaisessa yksityispilvessä 51 sivua + 1 liite 31.3.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikan koulutusohjelma
Suuntautumisvaihtoehto	Pilvipalvelut ja tietoverkot
Ohjaaja	Lehtori Harri Ahola

Insinöörityön toimeksiantajana oli Metropolia Ammattikorkeakoulu ja sen tarkoituksena oli suunnitella sekä toteuttaa vikasietoinen OpenStack pilvi-ympäristö sekä isännöidä palveluita hyödyntäen Docker-säiliöitä rakennetussa pilvi-infrastruktuurissa. Työn luonne on käytännönläheinen, mutta siinä käsitellään myös OpenStack-komponentteja sekä säiliöiden toimintaa teoriatasolla.

Työ alkaa teoreettisella osuudella, jonka jälkeen edetään pilvi-infrastruktuurin suunnitteluun, toteutukseen ja testaukseen. Lopuksi suoritetaan Docker-säiliöiden ylläpitoon liittyvät asennukset rakennetussa pilvessä sekä isännöidään haluttuja palveluita ja testataan niiden toimivuus. Työssä hyödynnetään lisenssinalaisia komponentteja Red Hat jakelijan tuoteperheestä, jonka vuoksi työn liitteissä ei tarjota konfiguraatiotiedostoja. Työhön liittyvät asennukset tehtiin sekä paikallisesti Metropolian tiloissa että etähallintaa hyväksikäyttäen.

Lopputuloksena työssä on laajennettava sekä vikasietoinen pilvi-infrastruktuuri, jota hyödynnetään Docker-säiliön isännöimiseen sekä palveluiden tarjoamiseen Metropolia Ammattikorkeakoulun laboratorioverkossa. Pilvi-infrastruktruurin sekä säiliön toimivuus testattiin käytännön tasolla ja se todettiin toimivaksi. Haasteita aiheuttivat lisenssin mukanaan tuoma aikarajoitteinen takaraja sekä monen virtualisoidun kerroksen hallinta sekä sen aiheuttama viive.

Työn tulokset osoittavat sen, että palveluita voidaan isännöidä suhteellisen helposti hyödyntäen avoimen lähdekoodin säiliö-teknologiaa sekä pilvi-infrastruktuuria.

Avainsanat	openstack docker virtualisointi pilvipalvelut

Author(s) Title Number of Pages Date	Antti Lappalainen Docker containers in Openstack cloud platform 51 pages + 1 appendix 31 March 2015
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Cloud services and networking
Instructor	Harri Ahola, Senior Lecturer

The goal of this final year project commissioned by Helsinki Metropolia University of Applied Sciences was to design and deploy a fault tolerant OpenStack cloud environment and utilize it in hosting services using Docker containers. The nature of this thesis is pragmatic but it also covers the OpenStack components and Docker containers from a theoretical viewpoint.

This thesis begins with the theoretical part and then advances to describing the design, deployment and testing of the cloud infrastructure. The final part will cover the installation, hosting and testing of the Docker container and hosted services. Configuration files will not be provided in the appendices section of this thesis as licensed products, features and components from the Red Hat cloud service family were utilized. Installations and configurations were conducted both on-site at the Metropolia Leppävaara campus and remotely.

The final outcome of this project is an expandable and fault tolerant cloud infrastructure which is used to host the Docker container and desired services in the Metropolia Network Laboratory. The operability of the OpenStack cloud and Docker container was tested and they proved functional. The main challenges were the time limit set by the licensed features and the delay produced by multiple layers of virtualization.

In conclusion, this thesis suggests that services can be hosted with relative ease utilizing open sourced OpenStack and Docker components.

Keywords	openstack, docker, virtualization, cloud service

Sisällys

Lyhenteet

AuFS	Advanced Multi-Layered Unification Filesystem. Kerrostava ja kirjaava tiedostojärjestelmä.
DHCP	Dynamic Host Configuration Protocol. IP-osoitteiden jakoon kehitetty verkkoprotokolla.
DNS	Domain Name System. Verkkotunnus järjestelmä.
IaaS	Infrastructure as a Service. Infrastruktuuri palveluna.
LXC	Linux Containers. Linux säiliöt.
NFS	Network File System. Verkkotiedostojärjestelmä.
NTP	Network Time Protocol. Verkkoaikaprotokolla.
PaaS	Platform as a Service. Sovellusalusta palveluna.
PXE	Preboot Execution Environment. Esikäynnistysympäristö.
SaaS	Software as a Service. Sovellus palveluna.
VPN	Virtual Private Network. Virtuaalinen erillisverkko.

1 Johdanto

Yritykset ovat pilvessä. RightScalen tutkimuksen mukaan 93 prosenttia yrityksistä on jo siirtynyt isännöimään palveluitaan pilvessä. (Weins. 2015.) Pilvipalvelut muotoutuvat moneen ja yritykset suuntaavat pilveen ajatuksenaan tehokkuuden parantaminen ja hallinnan helppous.

Tämän insinöörityön toimeksiajantaja on Metropolia Ammattikorkeakoulu, ja sen tarkoituksena on perehtyä skaalautuvan yksityisen pilven rakentamiseen sekä Docker -teknologiaa hyödyntävien säiliöiden ajamiseen rakennetussa pilvessä. Tavoitteena on rakentaa hallittava kokonaisuus hyödyntäen avoimen lähdekoodin Openstackin IaaS-ympäristöä sekä Red Hatin tuoteperheeseen kuuluvaa Red Hat Openstack Platformia. Rakennettu pilvi-infrastruktuuri integroidaan mahdollisesti myöhemmin Metropolia Ammattikorkeakoulun rakenteilla olevaan Internet Exchange -verkkoympäristöön.

Työ alkaa teoreettisella tiedolla pilven monipuolisuudesta sekä eri pilvimallien eroista ja ominaisuuksista. Lisäksi käsittellään pilvityyppejä ja kaikkien edellä mainittujen komponenttien esiintyvyyttä yrityksissä. Tämän jälkeen edetään tässä työssä käytettävään Openstack-ympäristöön ja sen osa-alueisiin. Sen rinnalla vilkaistaan Docker-säiliöiden rakennetta. Viimeiseksi käsittellään Openstackin käyttöönottoon liittyvät työvaiheet suunnittelusta asennukseen ja testaukseen sekä Docker-säiliöiden ajaminen käyttöönotetussa pilvessä.

Työssä esitelty infrastruktuuri on provisioitu käyttäen Red Hat Openstack Platform -tuotetta, joka on Red Hatin lisenssin alainen. Työn ympäristö on toteutettu Red Hatin tarjoamalla ilmaisella kokeilulisenssillä. Tästä johtuen työn lopussa ei tarjota konfiguraatiotiedostoja tai valmiita pohjia toisintoa varten. Työhön tarvitut lisenssit sisältävät Red Hat Enterprise Linux-, Red Hat Openstack Platform- sekä Red Hat Atomic Host -lisenssit.

2 Pilvipalvelu

2.1 Pilvimallit

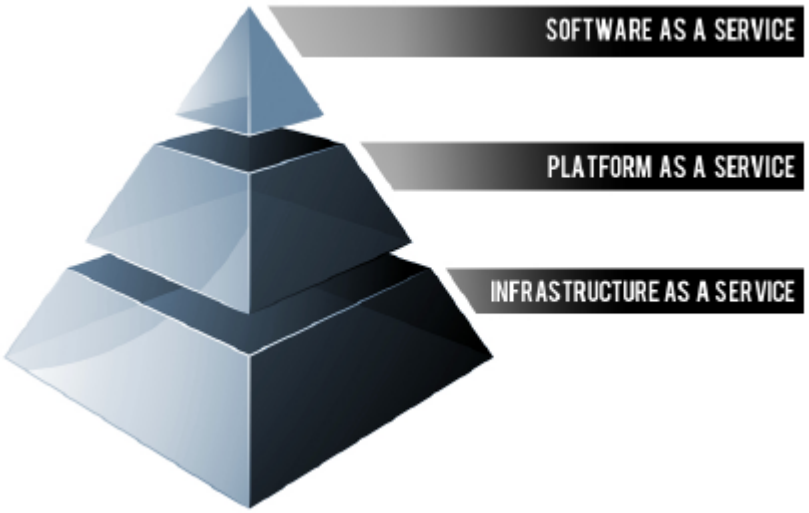
Pilvipalveluiden mallit jaetaan tyypillisesti kolmeen luokkaan, IaaS (Infrastructure as a Service), PaaS (Platform as a Service) ja SaaS (Software as s Service). Kuvitteellisen pyramidin pohjalla kuvassa 1 on ensimmäisenä IaaS, joka kuvataan itsepalveluna, jossa palveluntarjoaja ulkoistaa asiakkaalle tarjotun infrastuktuurituotteen hallinnan. Infrastruktuuri voi sisältää mm. näennäistettyjä verkkoja, virtuaalisia palvelimia, laskentaresursseja, kuormantasaajia ja palomuureja. Tyypillisesti mallissa laskutus on käyttöperusteinen, jolloin asiakas maksaa esimerkiksi tiedonsiirrosta tai laskentaresurssien käytöstä tunteina. (IaaS, PaaS, SaaS Explained and Compared. 2015.)

Asiakkaan näkökulmasta malli tarjoaa ratkaisun yrityksen skaalautuessa suuremmaksi tai pienemmäksi, sillä fyysisten laitteiden hankintaan tai poistoon ei tarvitse uhrata resursseja. Sen sijaan asiakas voi tarvittaessa hankkia lisää laskentatehoa suoraan palveluntarjoajalta ja näin välttää uusien laitteiden hankintaan liittyvät asennus- ja hankintakustannukset. IaaS mallin mukaisia palveluita tarjoaa mm. suuret toimijat kuten Google ja Amazon.

Seuraava mallitaso on PaaS, jossa asiakas hankkii sovellustason tai alustan palveluna. Mallissa infrastruktuuri on kokonaan tai osittain piilotettu asiakkaalta, jolloin skaalautuvuus tai sovellusten hajauttaminen monille palvelimille jää palveluntarjoajan vastuulle. Asiakas voi tällöin keskittyä esimerkiksi sovelluskehitykseen huolimatta pohjalla olevasta pilvi-infrastruktuurista, joka kuitenkin on läsnä myös PaaS mallissa.

PaaS-palveluntarjoajat tarjoavat usein valmiita kehitysalustoja tietyille ympäristöille, joita asiakas voi myös laajentaa tarjoituilla välineillä. Tämä mahdollistaa esimerkiksi sovelluksen nopean kehityksen, joskin alustan ominaisuudet voivat rajoittaa, minkä tyyppisiä sovelluksia sillä kannattaa rakentaa. Laskutus tapahtuu mallissa IaaS:n tapaan käytön mukaan. Palveluntarjoajina toimivat mm. Google. (IaaS, PaaS, SaaS Explained and Compared. 2015.).

Kuvitteellisen pyramidin ylimmällä tasolla kärkenä on SaaS. Mallissa asiakas saa käyttöönsä valmiin palvelun, jolloin palveluntarjoajan vastuu kasvaa entisestään. Esimerkkinä voidaan mainita Googlen Gmail, joka tarjoaa sähköpostin asiakkaalle. Keskeistä mallissa on se, että palveluita käytetään verkon kautta yleensä verkkoselaimen avulla, eikä asiakas ota kantaa palveluiden toimintaan tai kehitykseen. Laskutus hoituu mallissa myös käytön mukaan, esimerkiksi käyttäjien lukumäärän perusteella. (IaaS, PaaS, SaaS Explained and Compared. 2015.).



Kuva 1. Pilven palvelumallit. (Rackspace Support. 2015).

2.2 Pilvityypit

Pilvipalveluiden tyyppejä on neljä, yksityinen (private cloud), julkinen (public cloud), hybridi- (hybrid cloud) ja yhteisöllinen pilvi (community cloud). Pääosin erot tyyppien välillä muotoutuvat omistajuudesta, sijainnista ja ylläpidosta.

Yksityinen pilvi on on vain yhden organisaation käytettävissä. Palvelut yksityisessä pilvessä sijaitsevat usein yrityksen sisäisissä verkoissa eikä niihin pääse käsiksi yrityksen ulkopuolelta ilman esimerkiksi VPN-tunnelointia. Yksityisen pilven omistajuus voi olla joko organisaation tai toisen osapuolen hallinnassa. Tällaisen pilven ylläpito ja varsinkin rakentaminen ja suunnittelu voi vaatia hyvin paljon resursseja ja henkilökunnan koulutusta, mutta tällöin organisaatio on myös täysin päätäntävaltainen pilven ominaisuuksista. Lisäksi organisaatiolla voi olla huolia yksityisen tiedon päätymisestä julkiseen verkkoon, jolloin oman pilven rakentaminen tarjoaa herkän datan suojaisan tallennuksen yrityksen oman tietoturvan taakse. (Branca. 2014.)

Julkisen pilven omistaja on yleensä yritys tai organisaatio, jolla on palveluntarjoajan rooli. Se myös sijaitsee aina omistajan tiloissa. Palveluntarjoaja tarjoaa asiakkailleen edellä mainittuja pilvimalleja ja laskuttaa näitä käytön mukaan. Julkiseen pilveen sijoitettu tieto voi aiheuttaa ristiriitoja datan omistajuudesta ja mikäli yrityksen tieto on hyvin sensitiivistä julkinen pilvi voi kantaa mukanaan liian suuren riskin. Lisäksi muun muassa palveluntarjoajan palvelukatkot saattavat vaikeuttaa asiakkaan roolissa olevan yrityksen omien palveluiden saatavuutta. (Nippes. 2014.) Useat suuret yritykset, kuten Google, Amazon ja Microsoft, tarjoavat sekä yksityisille että yrityksille julkisen pilven palveluita.

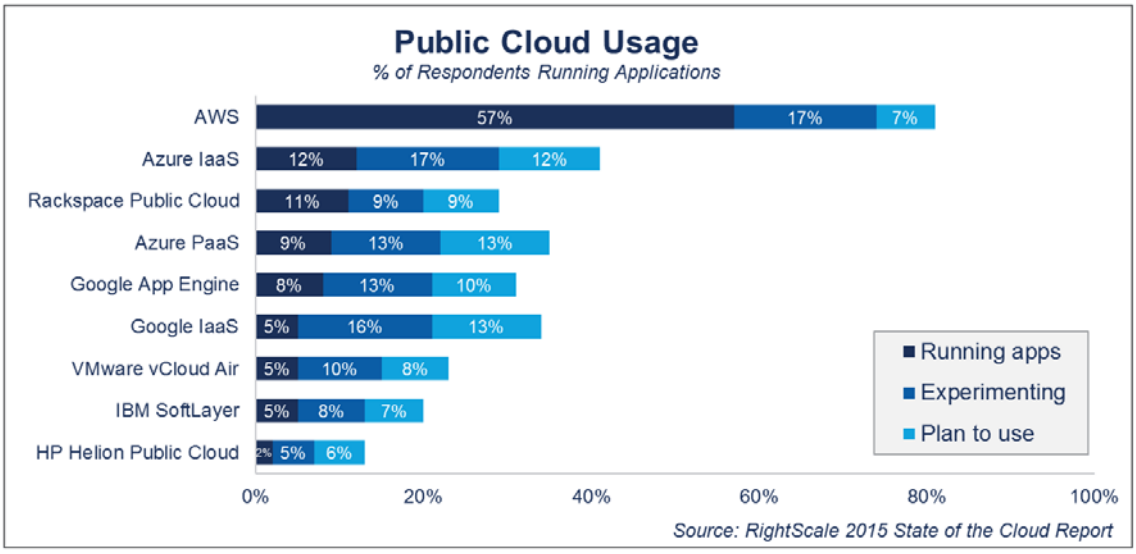
Yhteisöllinen pilvi voi olla yhden tai useamman organisaation hallinnassa tai omistuksessa. Sitä kuitenkin käyttää useampi kuin yksi organisaatio. Yhteisöllisyyden perustana voivat olla samanlaiset tavoitteet tai ajatusmallit esimerkiksi tietoturvasta. Lisäksi kuluja voidaan jakaa organisaatioiden välillä. Yhteisöllinen pilvi voi sijaita organisaatioiden tiloissa tai toisaalla. (Mell & Grance. 2011)

Hybridipilvessä kudotaan kaksi tai useampi pilvi-infrastruktuuri yhteen, jolloin voidaan taata mm. herkän tiedon saavutettavuus rajoitetusti ja silti voidaan hyödyntää julkisen pilven ominaisuuksia kuten helppoa skaalautuvuutta. Suuret toimijat ovat heränneet tähän ajatteluun ja tarjoavatkin jo nykyään työkaluja em. toimintojen suorittamiseen

omilla tuotteillaan. (Pervilä. 2014.) Pervilän haastatteleman Gartnerin varatoimitusjohtajan Lydia Leungin mukaan kahden tai usemman pilven välillä ”pomppiminen” voi kuitenkin aiheuttaa heiluriliikettä, joka ei ole tavoitteena hybridipilvessä. Hybridipilvi voi aiheuttaa suuriakin ylläpitokustannuksia, sillä usein infrastruktuurien yhteensovittaminen tuo mukanaan haasteita sekä tietoturvan, että teknisen toteutuksen saralta.

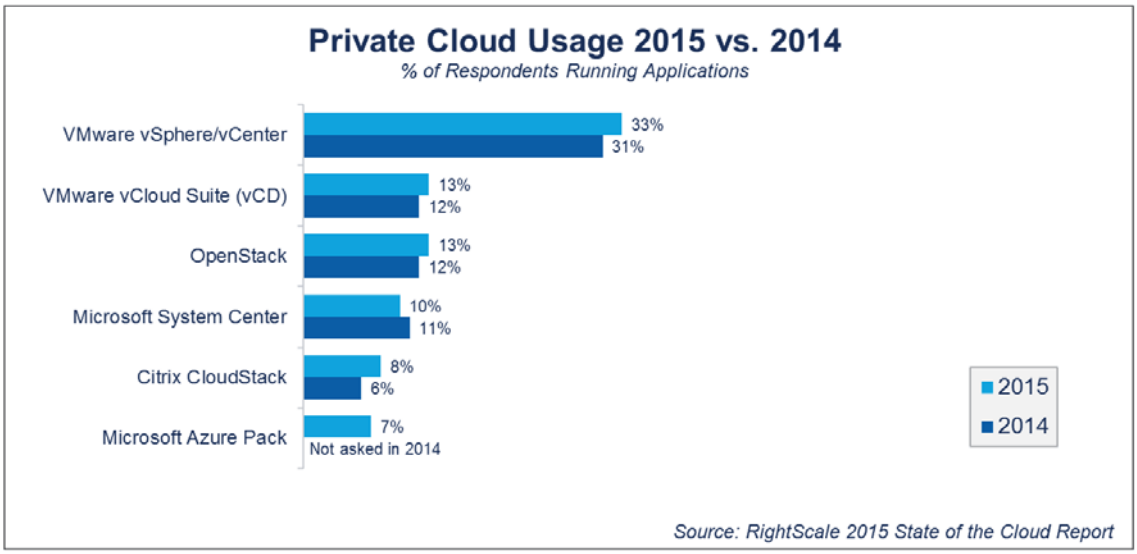
2.3 Pilvimarkkinat

Johdannossa mainittiin yritysten hyödyntävän jo vakaasti pilvipalveluita. Rightscalen tutkimuksen mukaan pilvessä palveluitaan isännöivien yritysten joukosta 88 prosenttia käyttää julkisia, 63 prosenttia yksityisiä ja 58 prosenttia molempia edellä mainittuja pilviä. (Weins. 2015.) Tutkimuksen mukaan julkisella puolella markkinoita johtaa Amazon suurella erolla seuraavaksi tulevaan Microsoftiin, kuten kuva 2 osoittaa.



Kuva 2. Julkisen pilven markkinat. (Weins. 2015).

Kuvassa 3 vertaillaan yksityisen pilven käyttöä. VMware on yhä vahvasti kiinni markkinajohtajan pokaalissa joskin Openstack on kasvattanut kannatustaan ja tulee laajentumaan merkittävästi. (Weins. 2015.)



Kuva 3. Yksityisen pilven markkinat. (Weins. 2015).

Tämän lisäksi tutkimuksesta selviää, että 13 prosenttia yrityksistä on siirtynyt käyttämään Docker teknologiaa hyödyntäviä säiliöitä. Sen lisäksi yli kolmasosa suunnitelee Docker-säiliöiden käyttöönottoa. (Weins. 2015.)

3 Openstack

Openstack on kokoelma ohjelmistoja, joiden avulla voidaan rakentaa sekä julkisia että yksityisiä pilvialustoja. Openstack määrittelee itsensä seuraavasti: ”Openstack on pilvikäyttöjärjestelmä joka kokoaa laskenta-, tallennus- ja verkkoresurssit hallittavaksi kokonaisuudeksi web-rajapinnan välityksellä”. Openstack toimii Apache 2.0 -lisenssin alaisuudessa, mikä antaa myös jakelijoille mahdollisuudet muokata lähdekoodia oman makunsa mukaan ja julkaista muokattua sisältöä ilman avointa muokatun koodin jakamista. (Apache License and Distribution FAQ. 2012.) Monet jakelijat hyödyntävät Openstackiä, kuitenkin suurimmat kehittäjät ovat tällä hetkellä HP, Red Hat sekä Mirantis lähes 50 prosentin yhteisosuudella. (Butler. 2014.)

3.1 Arkkitehtuuri

Openstack koostuu useista asennettavista ja konfiguroitavista komponenteista, mutta se voidaan kuitenkin jakaa karkeasti kolmeen osa-alueeseen.

- Compute - Laskenta
- Networking - Verkko
- Storage - Tallennus

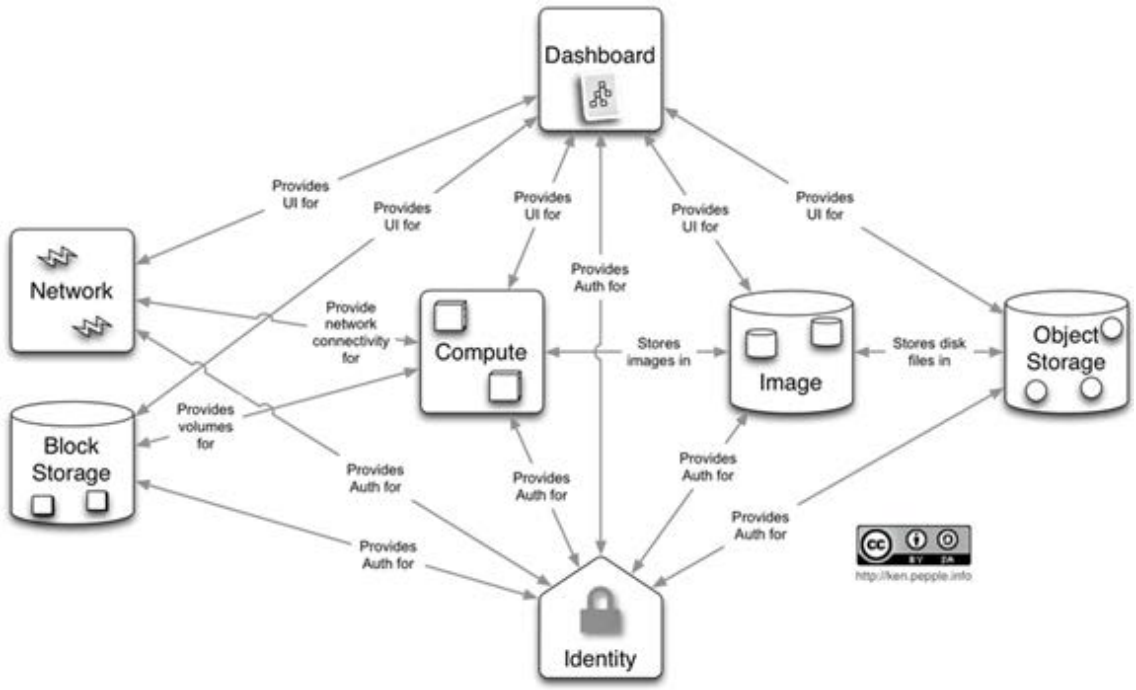
Osa-alueita hallitaan kojelaudaksi (dashboard) kutsutun web-hallinnan kautta, jonka avulla koottuja resursseja voidaan määrittää käytettäväksi esimerkiksi virtuaalipalvelimien ajamiseen.

Jakelijat, kuten Red Hat, ovat rakentaneet tämän päälle vielä yhden ylemmän hallintatason, jonka avulla ylläpitäjä voi suunnitella ja rakentaa useamman kuin yhden pilvi-infrastruktuurin, sekä ottaa käyttöön kyseiset ympäristöt. Hallintatason avulla voidaan kaikki käytettävissä olevat resurssit provisioida eli ottaa käyttöön jo olemassa olevaan infrastruktuuriin tai niiden avulla voidaan rakentaa uusi pilvi-infrastruktuuri.

Tässä työssä käytetään Red Hatin yrityskäyttöön tuotteistettua hallintatasoa, jonka avulla provisioidaan yksityinen pilvi-ympäristö olemassa olevaan verkkoinfrastruktuuriin Metropolia Ammattikorkeakoulussa. Tämän lisäksi rakennettu pilvi-infrastruktuuri voidaan myöhemmin liittää rakenteilla olevaan Internet Exchange -ympäristöön, jolloin Metropolia asettuu ikään kuin palveluntarjoajan rooliin. Tällöin Red Hatin tuotteen avulla voidaan suunnitella, rakentaa ja käyttöönottaa pilvialustoja asiakkaiden käyttöön. Tämän toteutuessa asiakas voi hallita pilvialustaansa Openstack kojelaudan avulla. Tämä mahdollistaa myös uusien laitteiden lisäämisen helposti olemassa oleviin pilvi-infrastruktuureihin.

3.2 Komponentit

Laitteistoriippumaton Openstack koostuu useista eri komponenteista, joilla kaikilla on selkeä rooli pilvi-infrastruktuurissa. Tämä mahdollistaa hyvin monipuolisen hajauttamisen, yksinkertaistaa vianetsintää sekä mahdollistaa palveluiden riippumattomuuden toisistaan. Toisin sanoen, kun kohdataan ongelmia, voidaan vika nopeasti rajata tiettyyn palveluun ja täten myös tiettyyn osaan palvelimia. Tämän lisäksi vian kohdistuessa vain yhteen palveluun se ei rajoita toisten palveluiden käytettävyyttä.



Kuva 4. Openstack-komponentit. (OpenStack Folsom Architecture. 2012).

Kuvassa 4 on Openstack-komponentit ja niiden liittyvyys toisiinsa käsitetasolla. Kuvan alin komponentti on ”Identity” eli henkilöllisyyden tunnistaminen. Siitä vastaa Keystone palvelu, joka mahdollistaa käyttäjien todennuksen ja palveluiden valtuutuksen.

Liikuttaessa kaaviossa ylöspäin kohdataan ”Block Storage”. Se mahdollistaa loogisen levyn liittämisen virtuaaliseen instanssiin. Tästä vastuussa olevaa palvelua kutsutaan Cinderiksi.

Verkkoinfrastruktuurin ”Network” tarjoavaa palvelua kutsutaan Neutroniksi. Se on vastuussa mm. pilven sisäisistä virtuaaliverkoista ja niiden liittämisestä muihin palveluihin.

”Compute” eli Openstackin käsitekielellä Nova tarjoaa pilvi-infrastruktuuriin laskentapalvelun. Sen vastuualueisiin kuuluu virtuaalikoneiden eli instanssien luominen ja ylläpito. Se ei kuitenkaan toimi hypervisorin eli alemman tason laitteistoläheisen virtualisointialustan roolissa, vaan on kykenevä käyttämään lukuisia hypervisoreita virtuaali-instanssien käynnistämiseen.

Levykuvien hallinnasta Openstackissa vastaa Glance. Sen avulla levykuvia voi tuoda ja ylläpitää pilvi-infrastruktuurissa tarvittaviin instansseihin. Glance on myös vastuussa instansseista luoduista tilan tallennoksista (snapshot). Näiden avulla voidaan instanssin tila ottaa talteen ja palata siihen tarvittaessa.

Dashboard eli Horizon tarjoaa infrastruktuuriin web-hallintapaneelin, josta käyttäjä voi hallita ja luoda ympäristöön esimerkiksi uusia virtuaalisia palvelimia tai verkkoja. Myös mm. levykuvien lataaminen sekä instanssien tilatallenteisiin liittyvät toimenpiteet tehdään Horizon-hallintapaneelin kautta.

Edellä mainitut komponentit ovat tässä työssä pääosassa. Komponentteja on lukuisia lisää ja ne tarjoavat monia eri palveluita, jotka tekevät Openstackin hyvin monipuoliseksi alustaksi. Kaikki palvelut on listattu liitteessä 1 englannin kielellä.

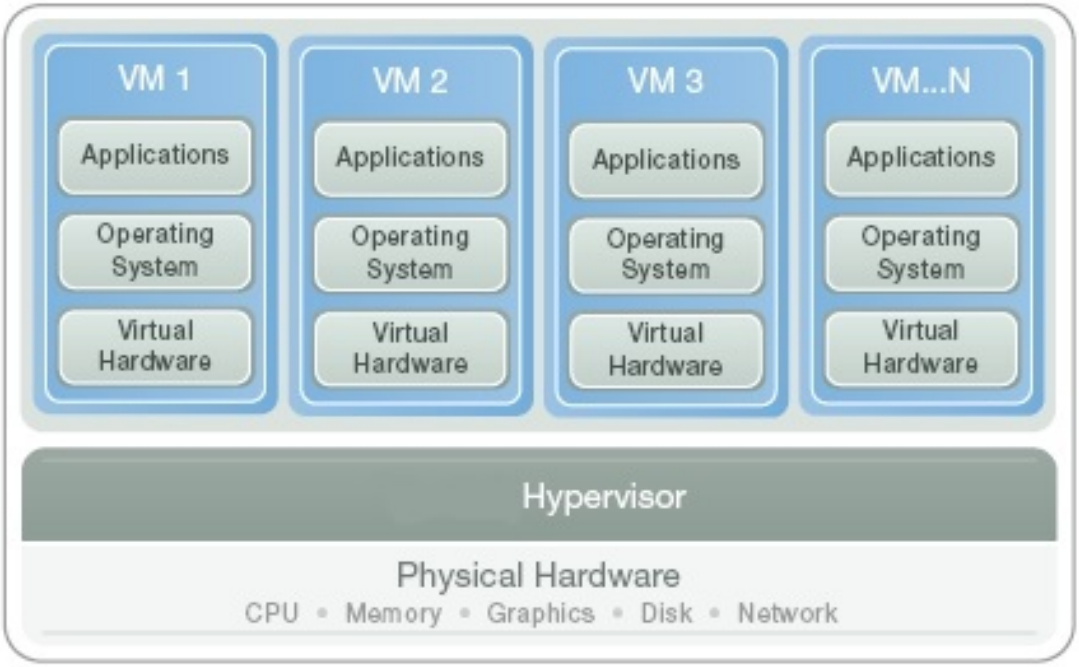
3.3 Red Hat Openstack Platform Installer System

Tässä työssä hyödynnetään Red Hat Openstack Platform Installer System -järjestelmää pilven rakentamiseen. Tällöin vältytään yksittäisten komponenttien asentamiselta, sillä yrityskäyttöön tuotteistettu sovelluspaketti tekee sen puolestamme. Se jättää aikaa keskittyä suunnittelemaan ja toteuttamaan ympäristön joka on vikasietoinen sekä takaa korkean kätettävyyden. Openstackin voi asentaa myös manuaalisesti komponentti kerrallaan, mutta se vaatii huomattavaa syventymistä komponenttien toimintaan sekä investointia scriptien muokkaamiseen, mikäli pilvestä halutaan tehdä ominaisuuksiltaan haluttu. Red Hat on kuitenkin tarjonnut mahdollisuuden kokeillä järjestelmäänsä aikarajoitteisella lisenssillä. Se takaa myös mahdollisuuden Metropolia Ammattikorkeakoululle asettua palveluntarjoajan rooliin, jolloin useamman pilven rakentaminen web-rajapinnan lävitse on tärkeä ominaisuus.

Tämän tyyppisen käyttöönoton kannalta on tärkeää hahmottaa kokonaisuus, sekä ymmärtää eri komponenttien roolit. Kuitenkin syventyminen yksittäisen komponentin toimintaan tai konfigurointiin ei tämän työn kannalta ole hyödyllistä, koska käytössä on tuote jonka osa-alueet on testattu markkinoiden suurimpien toimijoiden toimesta.

4 Säiliöt

Perinteisesti virtualisointi tarjotaan alustan eli hypervisorin avulla. Se mahdollistaa useiden yksittäisten eristettyjen käyttöjärjestelmien ajamisen vierekkäin yhden resurssienhallinnan päällä. Hypervisor asennetaan perinteisesti suoraan laitteeseen, palvelimeen ja se toimii ikään kuin alustana jonka päälle voidaan luoda virtuaalisia palvelimia hyödyntäen hypervisorin ominaisuuksia. Hypervisoreita on markkinoilla useita, sekä maksullisia että ilmaisia.



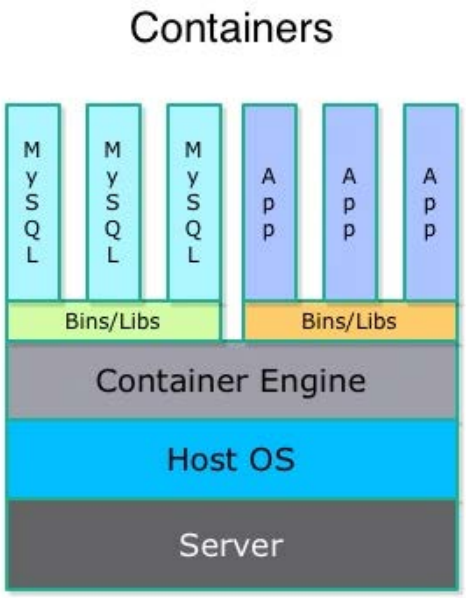
Kuva 5. Perinteinen virtualisointi hypervisorin avulla (Perlow. 2008).

Virtualisoinnin tarkoituksena on saada palvelinresurssit hyödynnettyä mahdollisimman tehokkaasti. Se voi olla kuitenkin kuluttaa huomattavan määrän resursseja, sillä jokainen virtualisoitu palvelin tarvitsee emuloidut virtuaaliset komponentit ja käyttöjärjestelmän, jonka päällä ajetaan tarjottuja palveluita.

Säiliöt (containers) tarjoavat ratkaisun liiallisten resurssien kulutukseen hyödyntämällä jaetun käyttöjärjestelmän mallia (Vaughan-Nichols. 2014). Yksittäiset säiliöt sisältävät kaikki tarvittavat riippuvuudet palvelun tai sovelluksen ajamiseksi. Tämän lisäksi yksittäinen käyttöjärjestelmä voi ajaa useita säiliötä samanaikaisesti ja täten tarjota huomattavan suuren hyödyntämisasteen.

James Turnbullin, Dockerin varatoimitusjohtajan mukaan jaetun käyttöjärjestelmän malli mahdollistaa sen, että 99,9 % turhasta virtuaalikoneiden emuloinnista voidaan jättää pois ja jäljelle jää hienosti paketoitu säiliö, joka sisältää halutun sovelluksen. Seuraussuhteena palvelinsovelluksia voidaan ajaa neljästä kuuteen kertaa suurempia määriä verrattuna perinteiseen virtualisointiin (Vaughan-Nichols. 2014).

Säiliöteknologioita on useita. Esimerkiksi Google on jo vuosia hyödyntänyt omaa lmctfy (Let Me Contain That For You) -teknologiaa. Käyttäjän näkökulmasta säiliöitä ei huomaa, mutta ”pellin alla” tapahtuu aina kun käyttäjä hyödyntää Googlen palveluita. Esimerkiksi käyttäessä Google Docsin tarjoamia toimisto-ohjelmia käyttäjälle osoitetaan uusi säiliö jossa palvelua ajetaan (Vaughan-Nichols. 2014). Kuvasta 6 havaitaan erot perinteiseen virtualisointiin, joka on esitetty kuvassa 5.



Kuva 6. Säiliöt kuvattuna. (Docker: Containers for the Masses. 2014).

4.1 Docker

Docker-teknologia valjastaa käyttöön Linuxin kernel- eli ydintason ominaisuudet. Se on rakennettu LXC (Linux Containers) -teknologiaa hyväksikäyttäen, joka hyödyntää ytimen nimiavaruutta säiliöiden eristämiseksi pohjalla olevasta käyttöjärjestelmästä (Merkel. 2014). Käyttäjä nimiavaruus (user namespace) erottaa säiliön ja käyttöjärjestelmän käyttäjätietokannan, jottei säiliön pääkäyttäjällä ole oikeuksia tehdä

muutoksia pohjalla olevaan käyttöjärjestelmään. Tämän lisäksi hyödynnetään prosessin nimiavaruutta (process namespace), joka on vastuussa säiliön sisällä ajettavien prosessien näyttämisestä ja hallinnoinnista (Merkel. 2014).

LXC hyödyntää myös kontrolliryhmiä (control groups, cgroups), jotka hallitsevat ja rajoittavat säiliön käyttämiä resursseja kuten muistia ja levyn käyttöä. Tällä varmistetaan, että jokainen säiliö saa tarvittavan osan käytössä olevissa resursseista.

Viimeinen Dockerin komponentti on AuFS (Advanced Multi-Layered Unification Filesystem). Se mahdollistaa yhden tai useamman tiedostojärjestelmän kerrostamisen siten, että mikäli prosessin on muutettava tiedostoa, AuFS tekee kopion muutettavasta tiedostosta ja täten ikään kuin kerrostaa tiedostojärjestelmää, jolloin tiedostoja useammasta eri tiedostojärjestelmästä voi sijaita saman katon alla. Se myös mahdollistaa säiliöiden versioimisen siten, että uusi versio on käytännössä pelkkä eroavaisuus (diff) edellisestä, jolloin levykuvatiedostojen koko voidaan pitää pienenä (Merkel. 2014).

Tässä työssä käytetään Red Hatin tarjoamaa Atomic Host -levykuvaa, joka on ikään kuin esimääritelty versio Red Hat Enterprise Linux -jakelusta, ja tarjoaa helpon tavan lähestyä docker-säiliöiden isännöimistä pilvessä. Docker-teknologia on valittu käytettäväksi tässä työssä sen monipuolisuuden sekä nousevan markkina-aseman takia. Sen lisäksi Atomic Host -levykuva on suunniteltu hyödyntämään docker-säiliöitä eikä sitä tarvitse esimääritellä ennen käyttöönottoa, sillä kaikki tarvittavat paketit ovat jo asennettu levykuvaan. Ainoastaan käyttäjiä koskevat määritykset on tehtävä ennen kuin säiliöiden hyödyntäminen voidaan aloittaa.

5 Pilvi-infrastruktuurin suunnitelu

5.1 Ympäristö

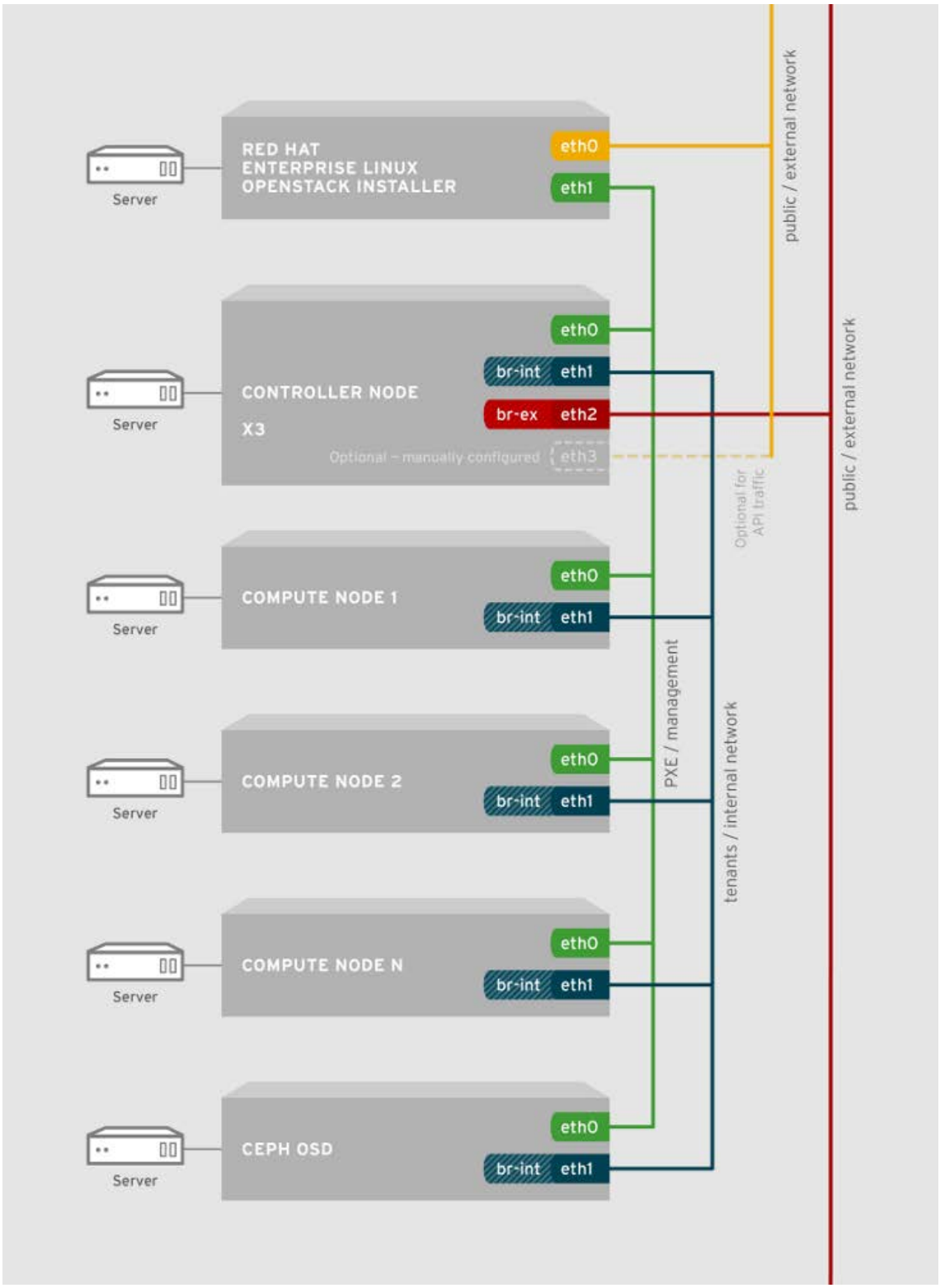
Ennen asennuksien aloittamista on tehtävä suunnitelma ympäristöstä ja listattava halutut ominaisuudet. Tässä työssä suunnitellaan vikaisietoinen sekä korkean saatavuuden takaava ympäristö, joka on laajennetavissa helposti mahdollisten lisäresurssien hankinnan jälkeen.

Käytössä olevat resurssit ovat tässä työssä rajalliset ja siksi ympäristö suunnitellaan VMwaren ESXi 5.5 -hypervisorin päälle virtualisoituna. Se tarjoaa mahdollisuuden liikkua taaksepäin vaiheittain asennuksien välissä (snapshot), mikäli asennuksissa on sattunut virheitä. Virtualisoitu ympäristö ei kuitenkaan ole suositeltu tapa, vaan Openstack suositellaan asennettavaksi suoraan laitteisto-tasolla oikeiden komponenttien päälle. Tällöin suorituskyky on parempi sekä vältyttään mahdollisilta sisäkkäisen virtualisoinnin ongelmilta. Suunnitteltu ympäristö vaatisi kuitenkin usean palvelimen käyttöönottoa, mikä ei resurssien puitteissa ollut mahdollista. Ympäristö on kuitenkin helposti laajennettavissa mahdollisilla tulevilla laitteilla.

Red Hat Openstack Platform tarjoaa yksinkertaisen keinon palvelinresurssien lisäämiseen ja poistamiseen rakennetusta pilvi-infrastuktuurista. Tämä mahdollistaa myös sen, että lisäresurssien tullessa tarjolle virtuaaliset resurssit voidaan korvata fyysisillä laitteilla ilman, että käyttöönotettuun ympäristöön aiheutuu merkittäviä palvelukatkoksia.

5.2 Suunnitelma

Suunnitelman perustana on siis kehittynyt Openstack-ympäristö, joka on laajennettavissa sekä vikasietoinen. Suunnitelman lähteenä käytetään Red Hatin tarjoamaa käyttöönotto-opasta kehittyneille ympäristöille. Se tarjoaa askel-askeleelta ohjeen ympäristön suunnitteluun, rakentamiseen ja käyttöönottoon ja on tarkoitettu yrityksille tueksi. Ympäristö koostuu kahdeksasta virtualisoidusta palvelimesta. Nämä palvelimet resursoidaan käytettäväksi siten että kuusi niistä on osa Openstack-ympäristöä. Loput kaksi resursoidaan käytettäväksi http-palvelimena, joka tarjoaa asennusmedian käyttöön http-yhteyden välityksellä sekä NFS-palvelimena, joka tarjoaa levypinnan Openstack-ympäristölle.



Kuva 7. Kehittynyt Openstack-ympäristö. (Red Hat. 2015).

Kuva 7 selventää palvelimia ja niihin liitetyiden verkkojen toimintaa. Tässä ympäristössä hyödynnetään yhtä Red Hat Openstack Platform Installer -järjestelmää.

Sen lisäksi otetaan käyttöön kolme Controller-palvelinta ja kaksi Compute-palvelinta. Kuvassa alimpana sijaitseva Ceph OSD tarjoaa levypintaa tallennusta varten eikä täten ole osa luotua pilviympäristöä, sillä jo aikaisemmin mainittua NFS-palvelinta hyödynnetään levypintojen jakamiseen.

Red Hat Opestack Platform hyödyntää pilvi-infrastuktuurin rakentamiseen ns. rooleja. Rooleja on tarjolla neljä: Controller, Compute, Ceph Storage ja Generic RHEL. Näistä rooleista kaksi on keskiössä tässä työssä.

Controller-roolin omaava palvelin tarjoaa käytännössä kaiken tarvittavan Openstack-ympäristön pyörittämiseen kuten tietokannat, web-käyttöliittymän instanssien hallintaan ja autentikaatiopalvelut (Red Hat. 2015). Tämän työn kannalta ei ole tärkeää tietää, mitkä yksittäiset paketit asennetaan Controller-palvelimeen, sillä Openstackiä ei konfiguroida manuaalisesti yksittäinen tarvittava palvelu kerrallaan, vaan hyödynnämme Red Hatin tuotteistettua ympäristöä käyttöönoton helpottamiseksi. Tämä mahdollistaa yksityisen pilven rakentamisen ja käyttöönoton lyhyessä ajassa ilman komentorivipohjaista hallintaa. Lisäksi se mahdollistaa esimerkiksi sen, että yrityksen henkilökunta voi rakentaa ja käyttöönottaa pilveä käyttäen web-käyttöliittymää ilman laajaa koulutusta, sillä yksityiskohtainen osaaminen ei ole välttämätöntä.

Compute-roolissa toimiva palvelin toimii pilvi-infrastuktuurissa virtualisointialustana (Red Hat. 2015). Se siis tarjoaa laskentatehoa, jota tarvitaan esimerkiksi palvelinresurssien eli instanssien isännöimiseen pilvessä. Laskentatehoa tarjoava palvelin ei kuitenkaan ole ns. kriittinen osa infrastruktuuria, sillä ympäristön hallinta on täysin Controller-palvelimien tarjoama. Kuitenkin mikäli asiakas tai muu taho haluaa suorittaa prosesseja hyödyntäen pilvi-infrastruktuuria on laskentateho välttämätöntä.

Verkkoympäristö koostuu Metropolia Ammattikorkeakoulun laboratoriverkosta sekä Openstackin sisäisistä verkoista siten, että kuvan 7 mukainen punainen viiva (public / external) kuvaa Metropolian laboratorioverkkoa. Openstackin vaatimat sisäiset verkot (kuvassa vihreä ja sininen viiva, PXE / management ja tenants / internal) toteutetaan VMwaren ESXi -hypervisorin tarjoamilla virtaalisilla kytkimillä. Virtuaalipalvelimet kytketään kuvan mukaisesti, jolloin muodostuu kolme eri verkkoa: julkinen, hallinta- sekä sisäinen verkko instansseja varten.

5.3 Resurssivaatimukset

Pilvi-infrastruktuurin ulkoisen hallintatason tarjoava Installer-järjestelmä, ympäristön sisäisen hallinnan tarjoavat Contoller-palvelimet sekä laskentatehoa infrastruktuuriin tuovat Compute-palvelimet omaavat tietyt resurssivaatimukset. Red Hatin mukaan Installer-järjestelmän on oltava käyttöjärjestelmältään Red Hat Enterprise Linux 7. Tämän lisäksi muistin määrä on suositeltu kahdeksaan gigatavuun. Controller- sekä Compute-palvelimet puolestaan tarvitsevat vähintään kaksi gigatavua keskusmuistia sekä 100 gigatavua paikallista levytilaa. Laskentatehoa tarjoavat palvelimet on kuitenkin järkevää varustaa resurssien suhteen niin, että myöhemmin ajettaville instansseille jää riittävästi kapasiteettia. Nyrkkisääntönä pidetään suhdetta 1:1, siten että mikäli halutaan ajaa esimerkiksi kaksiytimistä instanssia kahdeksalla gigatavulla muistia, niin vastaavat määrät olisi löydyttävä vähintään yhdestä Compute-palvelimesta (Red Hat. 2015).

Aikaisemman kuvan 7 mukaan kaikki palvelimet lukuun ottamatta Controlleria voidaan varustaa kahdella verkkokortilla. Controller tarvitsee kolme korttia. Tämän lisäksi palvelut, kuten PXE, DHCP ja DNS, on poistettava käytöstä hallintaan tarkoitetusta verkosta, sillä Red Hat Openstack Installer järjestelmä hyödyntää niitä ja täten vältetään mahdolliset konfliktit palveluiden välillä (Red Hat. 2015).

Hallintaverkkoa käytetään tarvittavien palvelimien provisiointiin. Installer-järjestelmä käyttää PXE-käynnistystä uusien palvelimien asennukseen ja suorittaa tarvittavat asennukset palvelimeen verkon läpi. Tässä verkossa on myös sijaittava käytettävä asennusmedia, joka on tässä tapauksessa RHEL 7 -käyttöjärjestelmä. Sen tarjoamiseen käytetään virtuaalikonetta, joka sisältää http-palvelimen, jonka juuressa sijaitsee Red Hat Enterprise Linux 7 -levykuva purettuna. Hallintaverkossa on myös sijaittava levypinnan Openstackille tarjoava NFS-palvelin, jonka roolin tässä työssä omaa Openfiler. Se on vapaata lähdekoodia hyödyntävä jakelu, jonka hallintapaneelista voidaan määritellä levypintojen jakamiseen käytettävä NFS-palvelu sekä myös esimerkiksi iSCSi-levyjä. Hallintaverkosta ei oletuksena ole pääsyä internetiin, mutta tässä työssä tullaan määrittämään yhteys ulos verkosta, koska Red Hat Openstack Platform on lisenssin alainen ja asennus vaatii palvelimien rekisteröinnin Red Hatille, joka tapahtuu julkisessa verkossa asennuksen käynnistyttyä.

Tenant-verkkoon sijoitetaan ajettavat instanssit. Tästä verkosta ei ole pääsyä julkiseen internetiin, mikäli sitä ei erikseen konfiguroida. Ulkoinen (public) verkko tarjoaa pääsyn palvelimille Metropolian sisäisestä laboratorioverkosta, joka on reititetty julkiseen verkkoon. Aikaisemmin mainittu hallintaverkko liitetään Installer-järjestelmässä tähän verkkoon, jotta palvelimien rekisteröinti voidaan taata.

6 Asennus

6.1 Installer-järjestelmän esimäärittelyt

Red Hat Openstack Installer -järjestelmä tarjoaa ylemmän hallintatason pilvi-infrastruktuurin rakentamiseen. Sen avulla voidaan rakentaa sekä uusia pilviympäristöjä että lisätä tai poistaa palvelinresursseja aikaisemmin luoduista ympäristöistä. Hallintatason käsittely tapahtuu verkkoselaimen avulla yksinkertaisella web-käyttöliittymällä.

Installer-järjestelmä asennetaan Red Hat Enterprise Linux 7 –käyttöjärjestelmään virallisista Red Hatin lähteistä. Ennen asennusta on tehtävä muutama esimäärittely, toimivuuden takaamiseksi. Komennot on suoritettava pääkäyttäjän oikeuksin:

- `systemctl stop NetworkManager.service`
- `systemctl disable NetworkManager.service`

Installer-järjestelmä käyttää network-palvelua networkmanagerin sijaan, joten jälkimmäinen on suljettava konfliktien välttämiseksi. Tämän lisäksi verkkokorttien asetustiedostoja on muokattava sijainnista /etc/sysconfig/network-scripts/ifcfg-X, jossa ”X” on verkkokortin liitännän nimi. Seuraavat parametrit on lisättävä edellä mainittuun sijaintiin.

- `NM_CONTROLLER=no`
- `ONBOOT=yes`

Tämän jälkeen käynnistetään network-palvelu ja poistetaan dnsmasq-palvelu seuraavilla komennoilla. Dnsmasq toimii hallintapalveluna DHCP- ja DNS-palveluille ja voi häiritä järjestelmän omaa DHCP-hallintaa.

- `systemctl start network.service`
- `systemctl enable network.service`
- `yum remove dnsmasq`

6.2 Rekisteröinti

Installer-järjetelmä on rekisteröitävä, sillä se on lisenssin alainen tuote. Rekisteröintiin käytetään Red Hat portaalin käyttäjätunnusta ja salasanaa, jotka kysytään rekisteröinnin aikana käyttäjältä. Rekisteröinti suoritetaan seuraavalla komennolla.

- `subscription-manager register`

Tuotteen rekisteröinnin jälkeen on löydettävä oikea pool-id-numero, joka yksilöi tarvittavat kanavat Installer-järjestelmän asentamiseksi. Komennot listaavat pitkän ID-numeron joka tarvitaan seuraavaa vaihetta varten.

- `subscription-manager list --available | grep -A8 "Red Hat Enterprise Linux Server"`
- `subscription-manager list --available | grep -A8 "Red Hat Enterprise Linux OpenStack Platform"`

Seuraavassa vaiheessa liitetään ID-numero ja tarvittavat kanavat käyttöön:

- `subscription-manager attach --pool=ID-numero`
- `subscription-manager repos --enable=rhel-7-server-rpms`
- `subscription-manager repos --enable=rhel-7-server-openstack-6.0-installer-rpms`
- `subscription-manager repos --enable=rhel-server-rhscl-7-rpms`

6.3 Asennus

Palomuurisäännöt ja SELinux-oikeudet lisätään automaattisesti asennuksen aikana. Mikäli lisättyjä sääntöjä halutaan tarkastella, ne löytyvät käyttöönotto-oppaasta (Red Hat. 2015). Asennus ei vaadi käyttäjältä suuria toimenpiteitä, joskin ympäristön

suunnitelma on tärkeä toteuttaa ennen asennuksen aloittamista seuraavalla komennolla.

- `yum install rhel-osp-installer`

6.4 Määritys

Paketin asennettua aloitetaan järjestelmän määritys komennolla:

- `rhel-osp-installer`

Tämän jälkeen käyttäjältä kysytään asennukseen tarvittavia tietoja, kuten hallintaverkon verkkokorttia, IP-osoitetta, verkkomaskia, DHCP-aluetta, DNS-palvelinta, toimialueen nimeä, NTP-palvelinta, ja aikavyöhykettä. Käyttäjä määrittelee tiedot oman suunnitelmansa mukaisesti. Tämän lisäksi käyttäjältä vaaditaan pilvi-infrastruktuuriin asennettavien palvelinresurssien root-käyttäjän salasana tai julkinen SSH-avain, asennusmedian sijainti http-palvelimella sekä Red Hat -portaalin käyttäjätunnus ja salasana asennettavien palvelimien rekisteröimiseksi.

Tässä työssä määrittellään Installer järjestelmän hallintaverkko osoiteavaruuteen 172.16.10.0/24, SSH-todennus salasanalla sekä asennusmedian sijainti ulkoiselta http-palvelimelta.

Tämän jälkeen hallintapaneeliin voi kirjautua Installer-järjestelmän IP-osoitteella tai host-nimellä. Kirjautumiseen tarvittavat salasana ja käyttäjätunnus ilmoitetaan komentorivillä asennuksen onnistuttua. Kuvassa 8 on esimerkki onnistuneesta asennuksesta.

```
Success!
* Foreman is running at https://www.example.com
    Initial credentials are admin / nziESrtcuNVFG79z
* Foreman Proxy is running at https://www.example.com:8443
* Puppetmaster is running at port 8140
The full log is at /var/log/rhel-osp-installer/rhel-osp-installer.log
```

Kuva 8. Esimerkki onnistuneesta asennuksesta. (Red Hat.2015).

Lisäksi on määritettävä jo aiemmin mainittu yhteys hallintaverkosta Metropolian laboratorioverkkoon, josta on reititetty yhteys julkiseen verkkoon palvelimien rekisteröinnin takaamiseksi. Ensimmäiseksi vaihdetaan /etc/sysctl.conf tiedoston net.ipv4.ip_forward-arvo ja otetaan se käyttöön seuraavin komennoin.

- `net.ipv4.ip_forward = 1`
- `sysctl –p`

Seuraavaksi otetaan käyttöön IP-naamioiminen (masquerading). If-nimi kenttään sijoitetaan verkkokortin nimi, johon verkkoliikenne uudelleenlähetetään. Parametrin XX.XX.XX.XX/XX tilalle sijoitetaan hallintaverkon verkko-osoite.

- `iptables -t nat -A POSTROUTING -o [if-nimi] -j MASQUERADE`
- `iptables -A FORWARD -s [XX.XX.XX.XX/XX] -j ACCEPT`
- `iptables -A FORWARD -d [XX.XX.XX.XX/XX] -j ACCEPT`
- `iptables -A FORWARD ! -s [XX.XX.XX.XX/XX] -j DROP`

Lopuksi tallennetaan muutokset ja käynnistetään verkkoyhteydet uudelleen muutoksien voimaatulemiseksi.

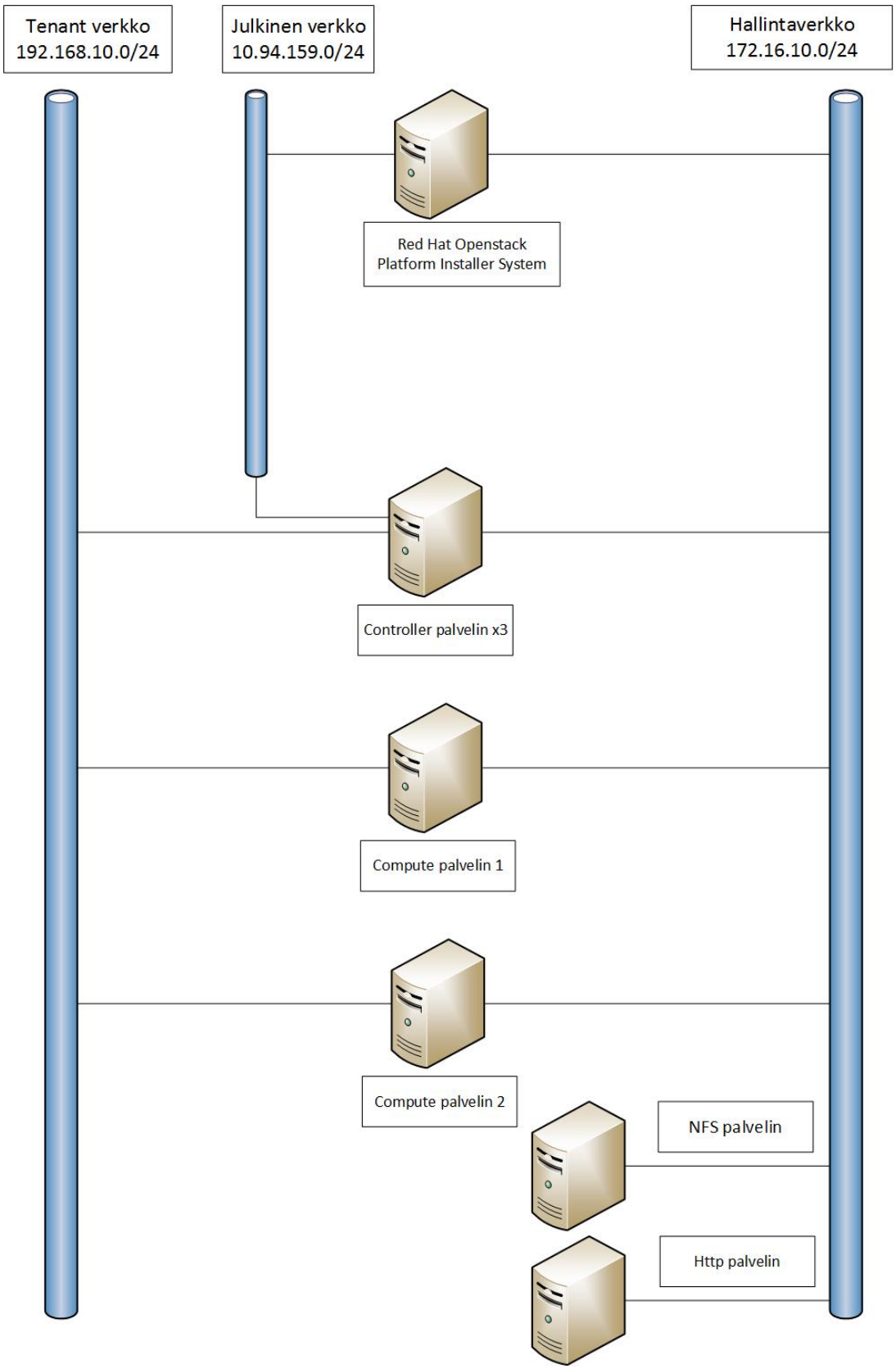
- `iptables-save > /etc/sysconfig/iptables`
- `systemctl restart network.service`

Nyt liikenne, joka saa alkunsa hallintaverkosta, käännetään verkkokortille, jolla on yhteys julkiseen verkkoon. Tämä mahdollistaa myös palvelinresurssien pääsyn ulkoisiin resursseihin.

7 Pilvi-infrastruktuurin käyttöönotto

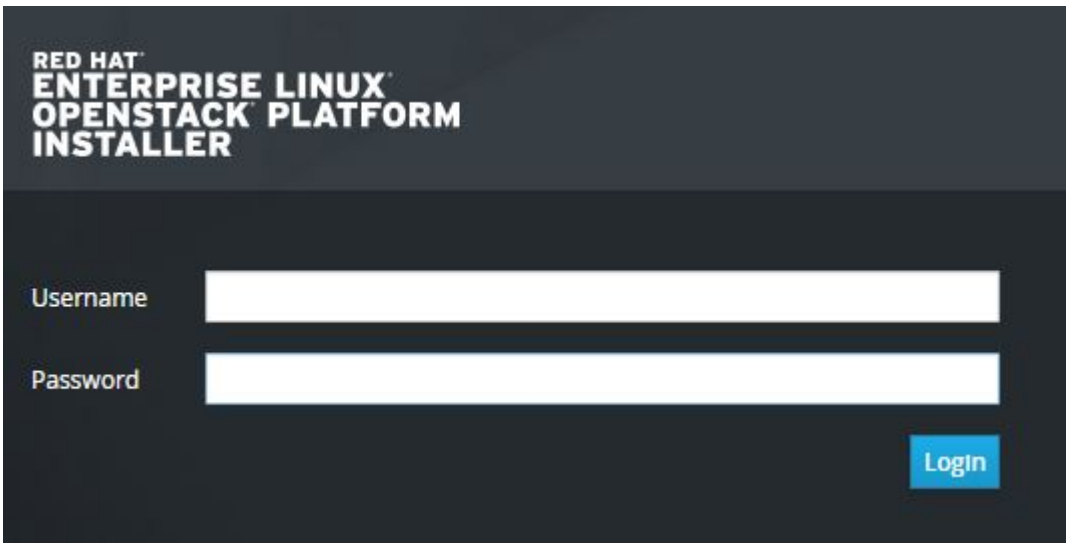
7.1 Verkkojen luonti

Ennen ympäristön käyttöönottoa on sille luotava virtuaaliset verkot, jotta Installer-järjestelmä osaa provisioida verkkokortit oikeisiin verkkoihin. Tässä työssä luodaan siis kolme verkkoa, joista yksi on olemassa oleva fyysinen verkko. Tässä vaiheessa verkot ikään kuin allokoidaan pilven käyttöön ja ne ovat täten käytettävissä käyttöönotetussa infrastruktuurissa, joskin tenant- sekä julkinen verkko on luotava myöhemmin myös käyttäen Horizon web-hallintaa, jotta niihin voidaan sijoittaa myös instansseja ja virtuaalisia reitittimiä. Verkot ovat esitettyinä kuvassa 9. Kuvasta voidaan myös todeta http-palvelin ja NFS-palvelin sekä käytettävien verkkojen osoitteet ja verkkomaskit.



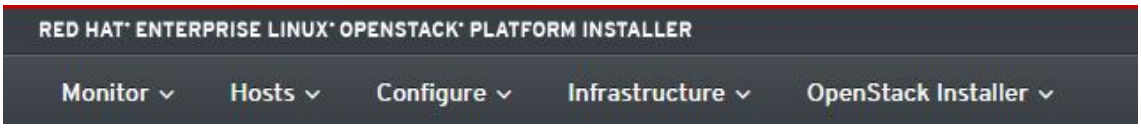
Kuva 9. Pilven topologia.

Verkkojen luominen toteutetaan web-hallinnan avulla. Luonti aloitetaan kirjautumalla web-selaimella Installer järjestelmään. On suositeltavaa vaihtaa admin-tason käyttäjän salasana ensimmäisellä kirjautumiskerralla. Kuvassa 10 on esimerkki kirjautumisikkunasta.



Kuva 10. Kuvakaappaus kirjautumisikkunasta web-hallintaan.

Hallintasivuston ylälaidassa on oheinen kuvan 11 valikko, jonka avulla tehdän kaikki tarvittavat toimenpiteet hallinnointia varten.

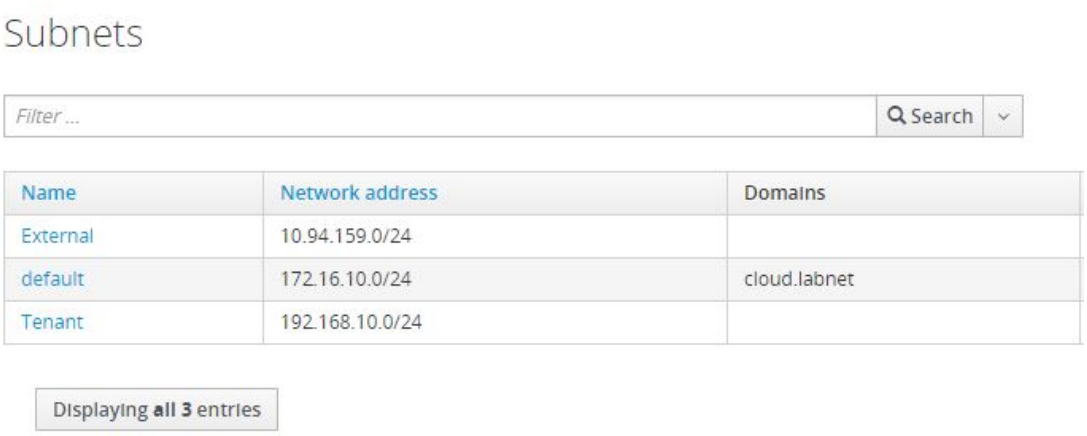


Kuva 11. Kuvakaappaus hallintavalikosta.

Siirtymällä välilehden Infrastructure Subnets osioon voidaan luoda uusia verkkoja. Verkot luodaan siten, että ainoastaan hallintaan tarkoitetulla verkolla on DHCP-palvelu aktiivisena. Muille verkoille voidaan käyttää joko sisäistä tietokantaa IP-osoitteiden allokoimiseen tai ulkoista DHCP-palvelinta. Myös manuaalinen asettaminen on mahdollista, joskin työlästä, mikäli palvelinresursseja on runsaasti. Kuvassa 12 on kuvakaappaus luoduista verkoista.

Tässä työssä hyödynnetään Installer-järjestelmän sisäistä tietokantaa IP-osoitteiden jakoon, jolloin verkkoa luodessa määritellään ainoastaan IP-osoitteiden aloitusosoite ja

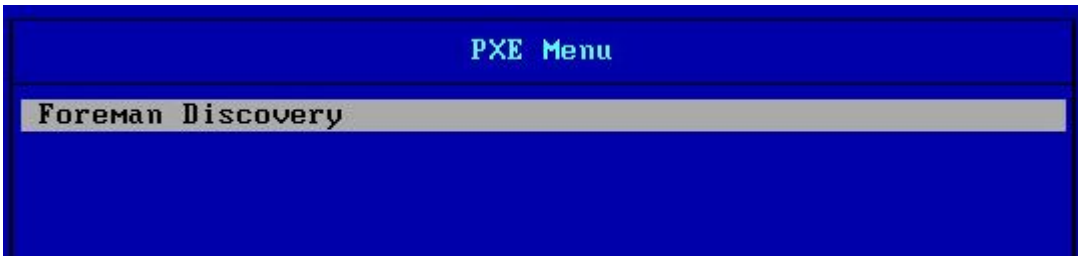
lopetusosoite. Tällöin Installer-järjestelmä määrää osoitteita palvelinresursseille järjestyksessä pienemmästä suurempaan.



Kuva 12. Kuvakaappaus luoduista verkoista.

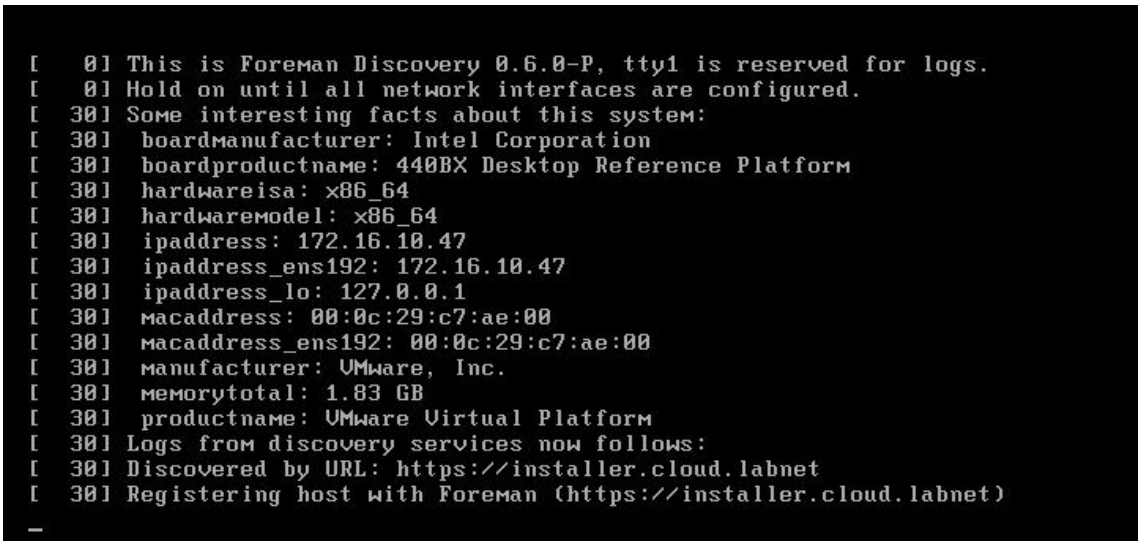
7.2 Palvelinresurssien paljastaminen

Palvelinresurssi paljastetaan Installer-järjestelmälle sijoittamalla ensimmäinen verkkokortti hallintaverkkoon ja käynnistämällä palvelin. Installer-järjestelmä hyödyntää PXE-käynnistystä ensimmäiseltä verkkokortilta ja käyttäjä valitsee käytettäväksi levykuvaksi ”Foreman Discovery” kuvan 13 mukaisesti.



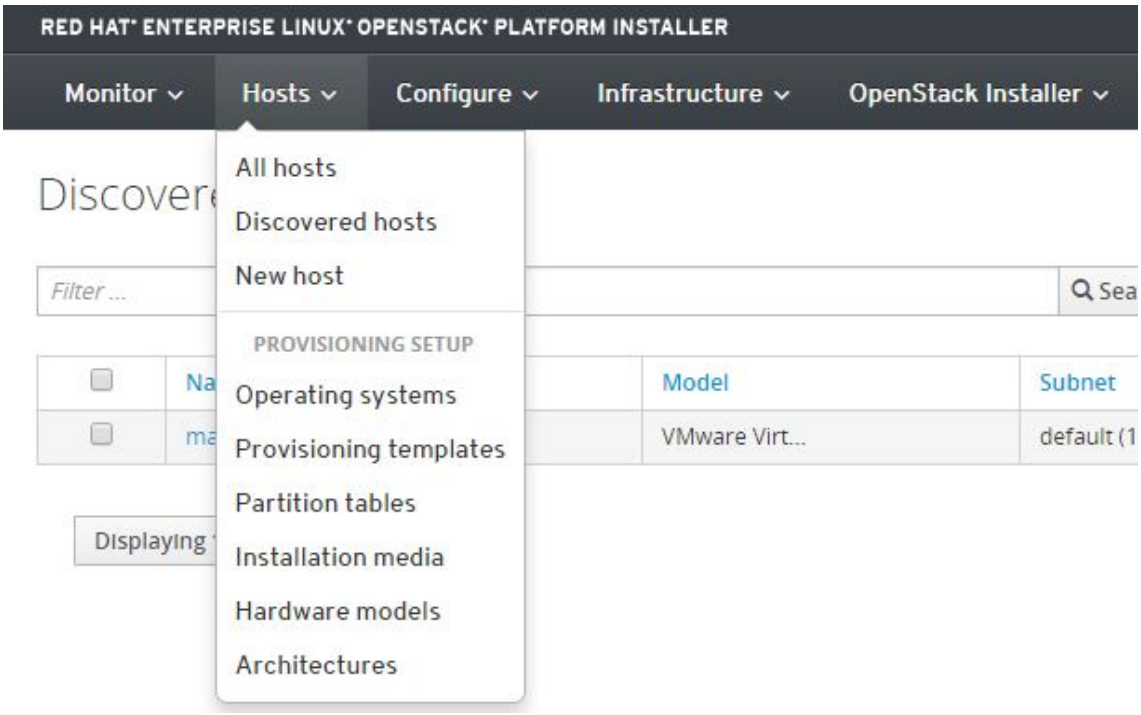
Kuva 13. PXE-käynnistysvalikko.

Tämän jälkeen palvelin lataa Installer-järjestelmästä ns. Foreman Discovery -levykuvan, jonka tarkoituksena on kartoittaa palvelimen käytössä olevat resurssit, kuten ytimien ja muistin määrä. Se myös rekisteröi palvelimen Installer-järjestelmän haltuun ja sijoittaa sen odotustilaan. Kuvassa 14 on kuvakaappaus onnistuneesta resurssin paljastamisesta.



Kuva 14. Onnistunut palvelinresurssin paljastaminen.

Tämän jälkeen palvelinresurssi on löydettävissä web-hallinnan Discovered Hosts –valikosta kuvan 15 mukaisesti.



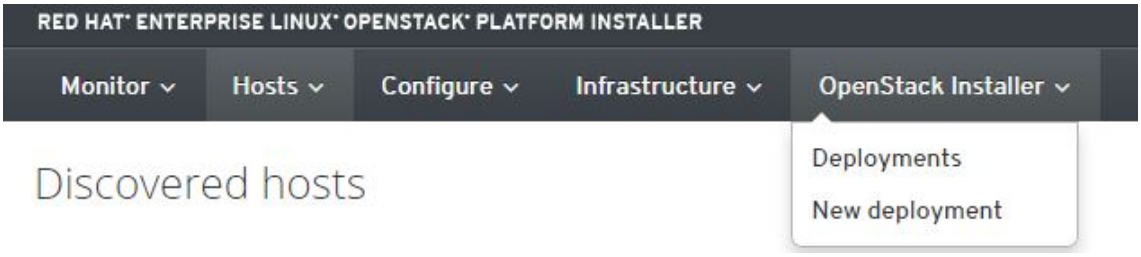
Kuva 15. Kuvakaappaus valikosta.

Kuva 16. Löydetyt palvelinresurssit

Kuvan 16 mukainen ns. löydetyt palvelinresurssit valikko pitää palvelimet varattuina käyttöönottoa varten. Palvelimet, jotka on sijoitettu tähän alueeseen, eivät ole vielä otettu käyttöön. Ne ikään kuin odottavat toimenpiteitä. On suositeltavaa paljastaa palvelimet pienissä erissä, sillä levykuvien lataaminen ja rekisteröinti Installer-järjestelmään voi syödä verkko- ja muistikapasiteettia huomattavasti. Lisäksi se helpottaa palvelimien tunnistamista, sillä oletuksena näkyviin tulee vain verkkokortin mac-osoite. Tässä työssä paljastettiin ensin Controller- ja sitten Compute-palvelimet tunnistamisen helpottamiseksi.

7.3 Pilven luominen

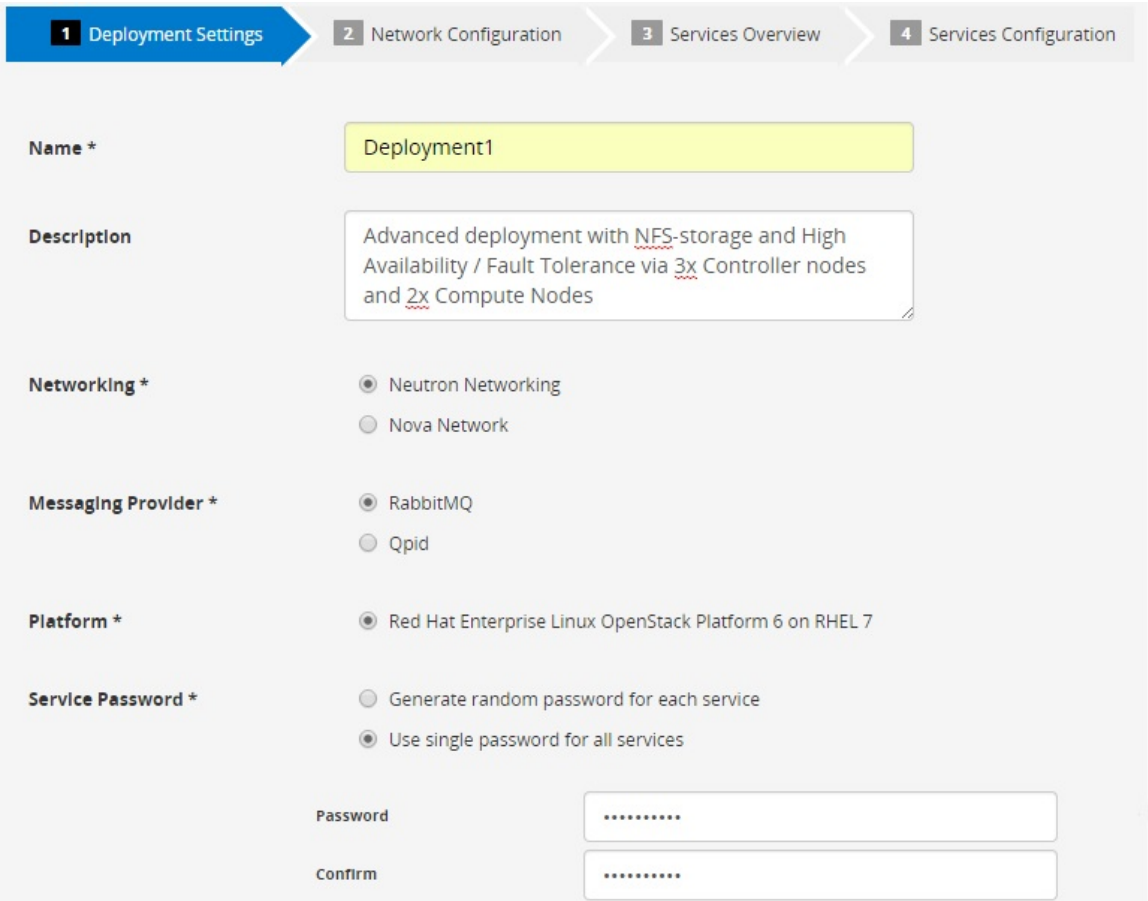
Kaikkien tarvittavien palvelinresurssien paljastamisen jälkeen pilven luominen tapahtuu siirtymällä välilehden Deployments, New Deployment osioon, kuten kuvassa 17 on esitetty.



Kuva 17. Uuden pilven luominen.

Ensimmäinen askel on nimen, kuvauksen ja salasanan luominen pilven tarvitsemille palveluille. Lisäksi valitaan verkkotyöskentelyyn sekä viestin välitykseen käytettävät Openstack komponentit. Tässä työssä käytetään verkkotyöskentelyyn Neutron -komponenttia sen tarjoaman monipuolisuuden vuoksi sekä viestien välitykseen

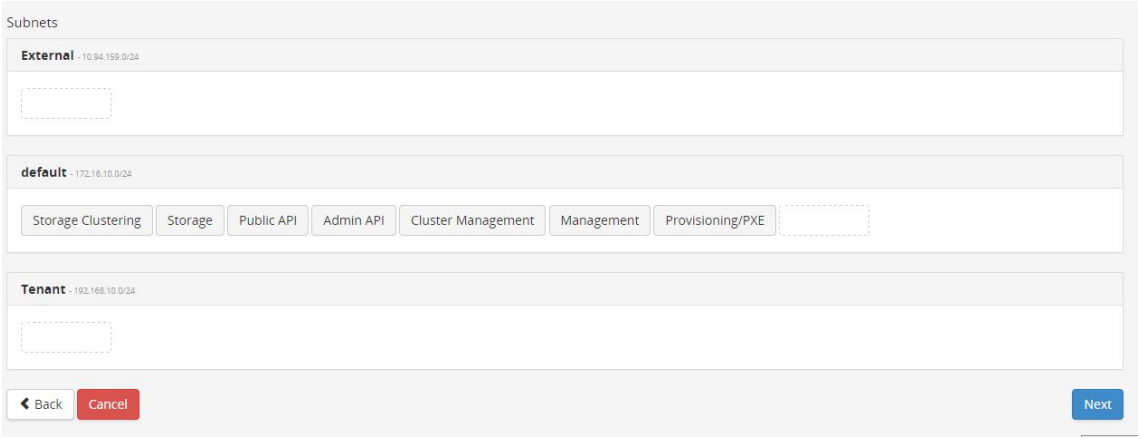
oletuskomponenttia RabbitMQ. Kuvassa 18 on kuvakaappaus ensimmäisestä vaiheesta.



Kuva 18. Käyttöönoton ensimmäinen vaihe.

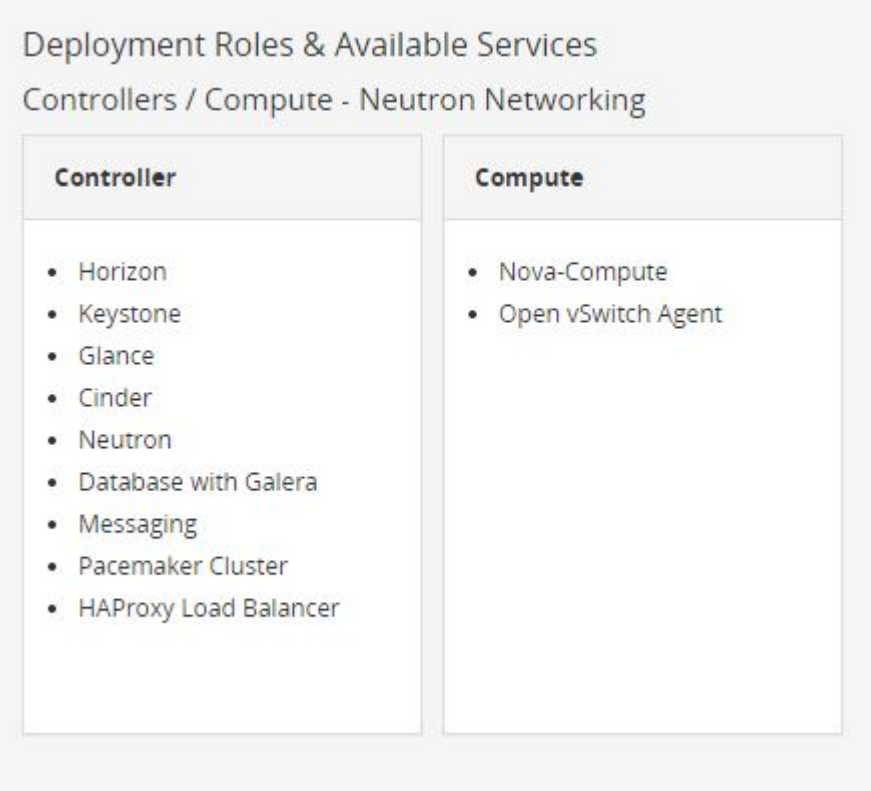
Seuraavassa vaiheessa määritellään rakennettavaan pilvi-infrastruktuuriin liitettävät verkot. Kuten aikaisemmin on mainittu tässä projektissa hyödynnetään kolmea verkkoa. Se on myös Red Hatin suosittelema konfiguraatiomalli. (Red Hat. 2015.) Verkkojen liittäminen tapahtuu sijoittamalla aikaisemmin luodut verkot niille osoitettuihin laatikoihin kuvien 19 ja 20 mukaisesti.

Kuva 19. Käytettävissä olevat verkot.



Kuva 20. Pilveen liitettävät määriteltävät verkot.

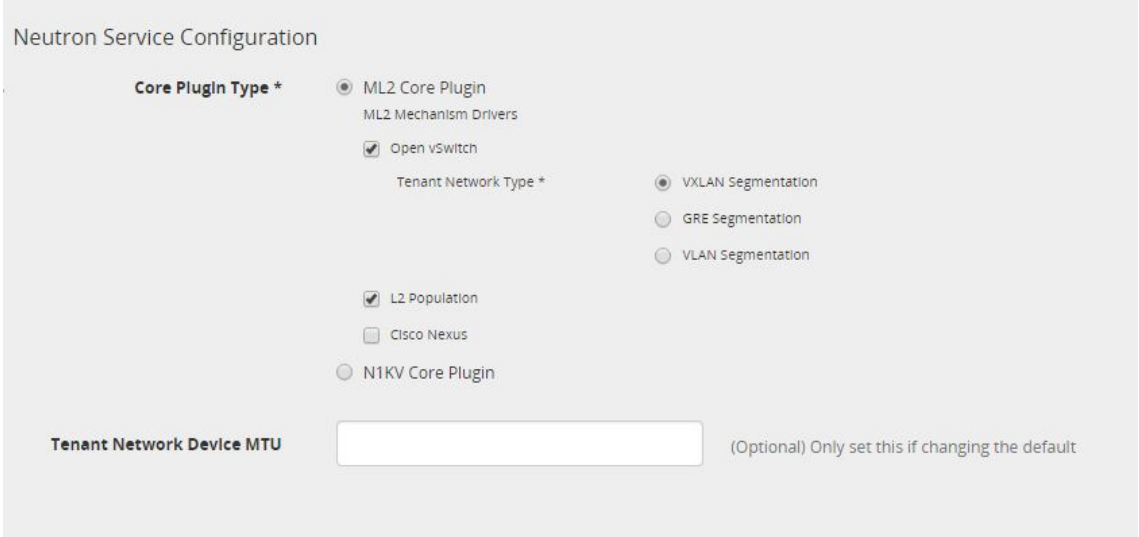
Seuraavassa vaiheessa voidaan tarkastella palvelimiin asennettavia Openstack-komponentteja.



Kuva 21. Asennettavat komponentit.

Tuotteistetun Openstackin hyödyt tulevat hyvin esille tarkastellessa kuvaa 21. Infrastruktuuria rakentaessa ei välttämättä tarvitse tietää komponenttien asennuskohteesta käytettäessä automaattista provisiointisovellusta.

Käyttöönoton luomisen viimeiset vaiheet ennen palvelinresurssien provisiointia ovat Neutron-, Glance- ja Cinder-komponenttien määritykset. Neutron-komponentti määritellään Red Hatin käyttöönotto oppaan mukaisesti kuvan 22 esittämällä tavalla.



Kuva 22. Neutron-komponentit määritykset.

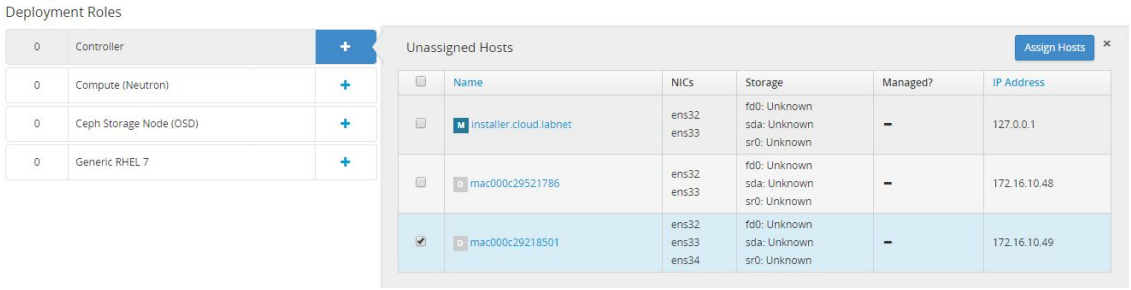
Glance- ja Cinder-komponentit määritellään käyttämään NFS-palvelinta levypintana, kuten kuvassa 23 ja 24 esitetään.

Kuva 23. Glance-komponentin määritykset.

Kuva 24. Cinder-komponentin määritykset.

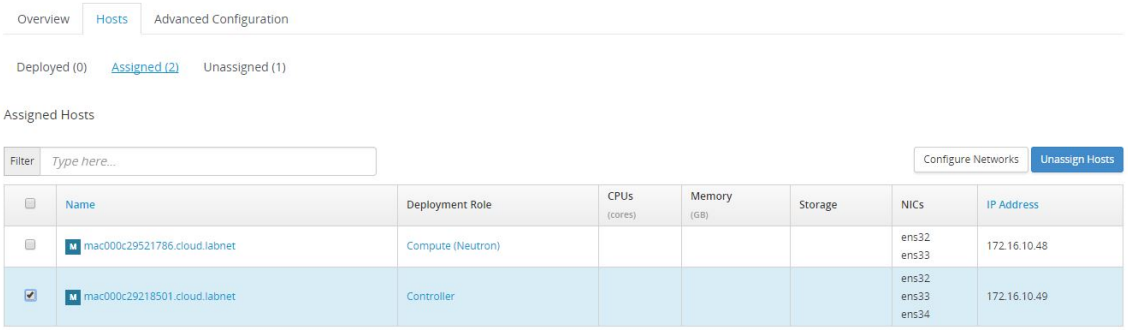
7.4 Palvelinresurssien provisiointi

Palvelinresurssien provisiointi ympäristöön on viimeinen vaihe ennen kuin infrastruktuuriin pääsee käsiksi. Ympäristön luonnin jälkeen tulee palvelimet määrittää oikeisiin rooleihinsa. Lisäys tapahtuu valitsemalla ”+”-ikonilla oikean roolin kohdalta halutut palvelinresurssit lisättäväksi kuvan 25 mukaisesti.



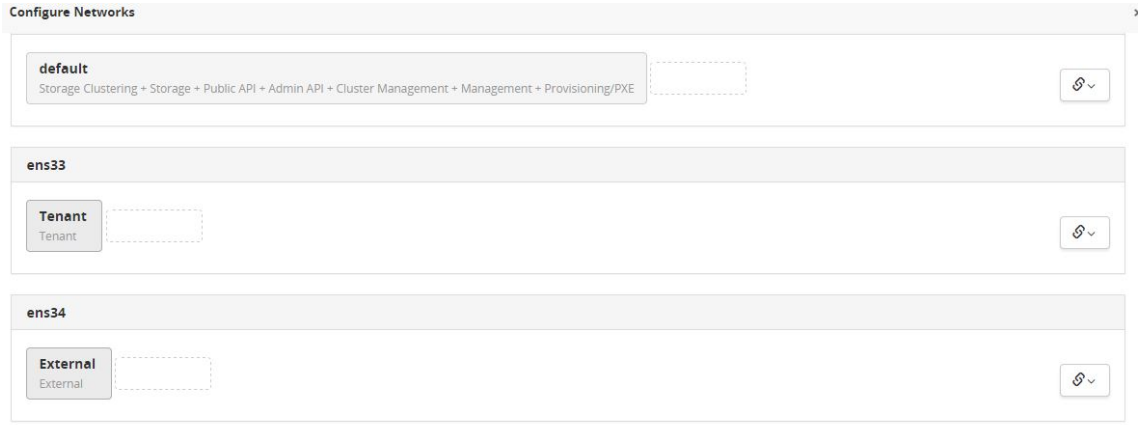
Kuva 25. Palvelinresurssien määritys rooleihin.

Tämän jälkeen on palvelimien verkkokortit osoitettava oikeisiin verkkoihin. Navigoidaan ”Hosts” -välilehden ”Assigned” -valikkoon, josta valitaan haluttu palvelin ja valitaan ”Configure Networks”, kuten kuvassa 26 on esitetty.



Kuva 26. Palvelinresurssien verkkojen määritys.

Tämän jälkeen siirretään pilveen määritetyt verkot oikeille verkkokorteille suunnitelman mukaisesti kuvan 27 tavoin.

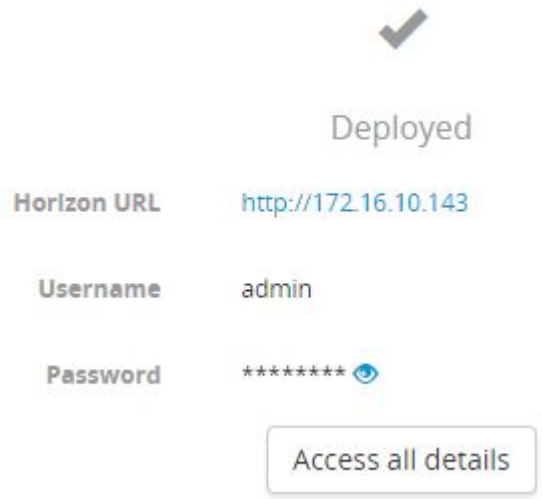


Kuva 27. Liitetyt verkot.

Viimeiseksi aloitetaan ympäristön palvelinresurssien provisiointi ”Deploy”-painikkeesta. Provisiointi vie aikaa ja tänä aikana prosentuaalinen edistyminen näytetään käyttäjälle. Palvelinresussit käynnistyvät uudelleen provisioinnin aikana.

Provisioinnin ollessa valmis oheinen kuvan 28 viesti näytetään käyttäjälle. Infrastruktuuri on nyt rakennettu sekä otettu käyttöön ja käyttäjä voi aloittaa pilven hyödyntämisen kirjautumalla pilveen viestissä mainittuun osoitteeseen, joka osoittaa Contoller-palvelimeen. Valmiiseen käyttöönotettuun ympäristöön voidaan tämän jälkeen lisätä esimerkiksi laskentatehoa tuottavia palvelinresursseja yksinkertaisesti lisäämällä palvelimet aikaisemmin mainitulla tavalla roolitusta hyväksikäyttäen.

Installer-järjestelmä ei tällöin muuta jo käyttöönotettuja palvelimia vaan ainoastaan lisää uudet lisätyt resurssit käyttöön.

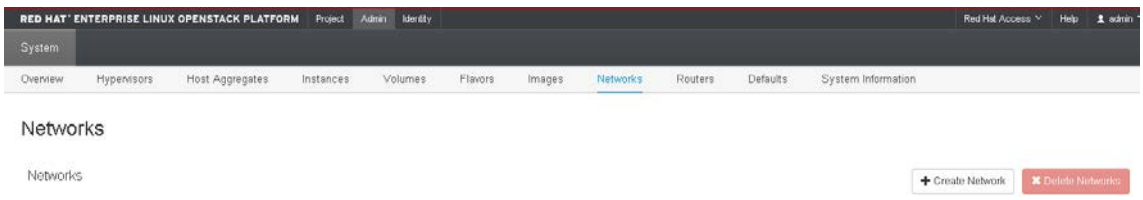


Kuva 28. Onnistunut pilven provisiointi.

8 Pilven käyttö

8.1 Verkkojen luonti

Infrastruktuuriin kirjaudutaan kuvassa 28 näkyvää IP-osoitetta hyödyntämällä. Sen jälkeen voidaan aloittaa luomalla julkinen verkko ulkopuolista yhteyttä varten sekä tenant-verkko instansseja varten. Vaikka verkot on luotu jo ylemmällä hallintatasolla ympäristön käyttöön, on ne ikään kuin otettava käyttöön myös infrastruktuurin sisäisen hallinnan osalta, jotta ne ovat käytettävissä.



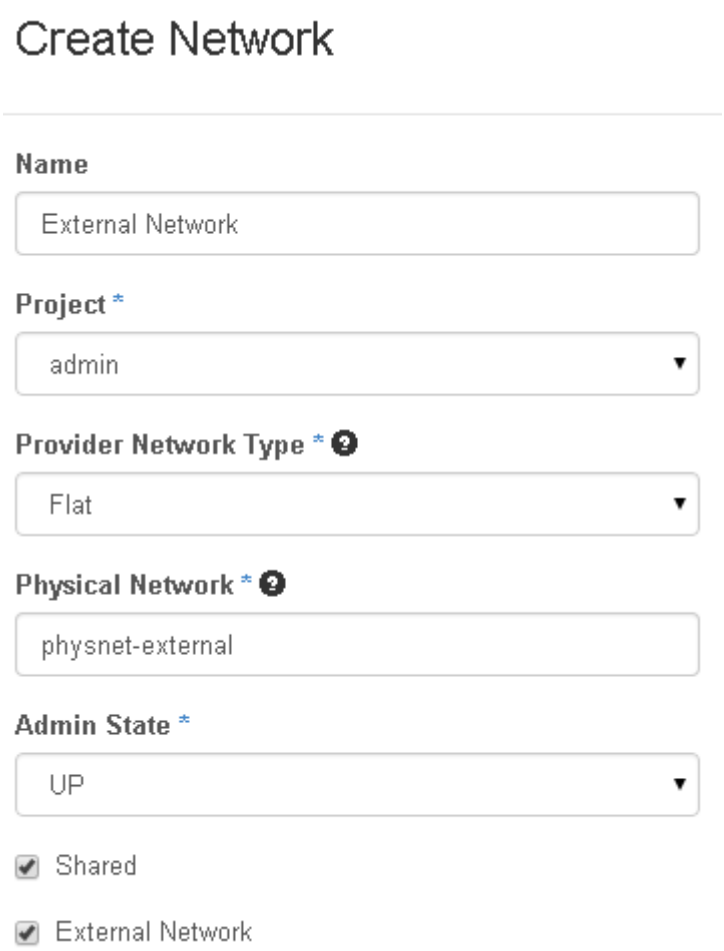
Kuva 29. Paneeli verkkojen luontiin.

8.1.1 Julkinen verkko

Verkot luodaan siirtmällä Admin-paneelin Network-osioon kuvan 29 mukaisesti. Ennen julkisen verkon luontia on kuitenkin kirjauduttava Controller-palvelimeen SSH-yhteyden avulla ja haettava ”Physical Network” -parametri julkista verkkoa varten. Seuraava komento suoritetaan Controller palvelimessa.

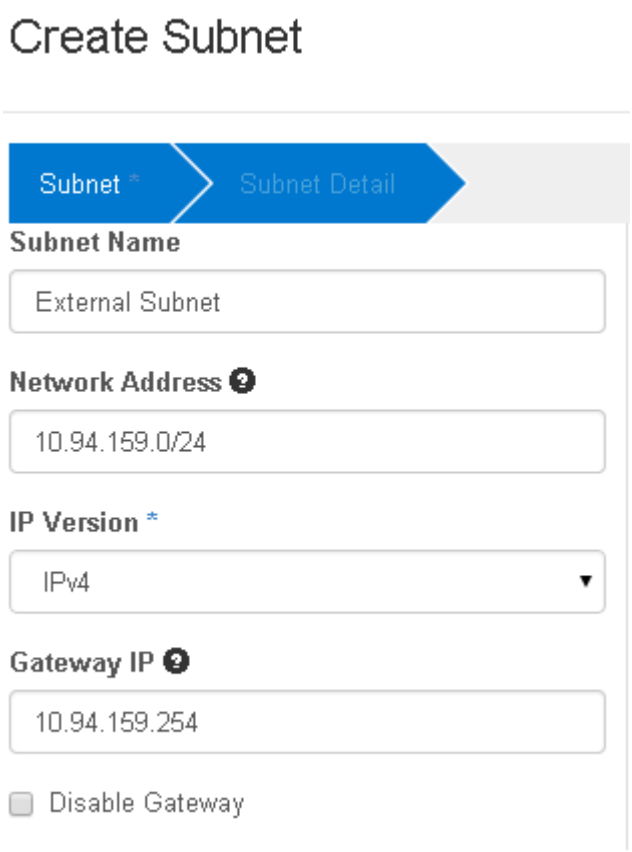
- `cat /etc/neutron/plugin.ini | grep "external"`

Komento palauttaa “network_vlan_ranges=physnet-external” -parametrin, jonka jälkimmäinen osa ”physnet-external” sijoitetaan ”Physical Network” -parametriksi.Tämän lisäksi tehdään verkosta julkinen ja jaettu sekä määritetään haluttu projekti, johon verkko kuuluu. Toimenpide on esitettynä kuvassa 30.



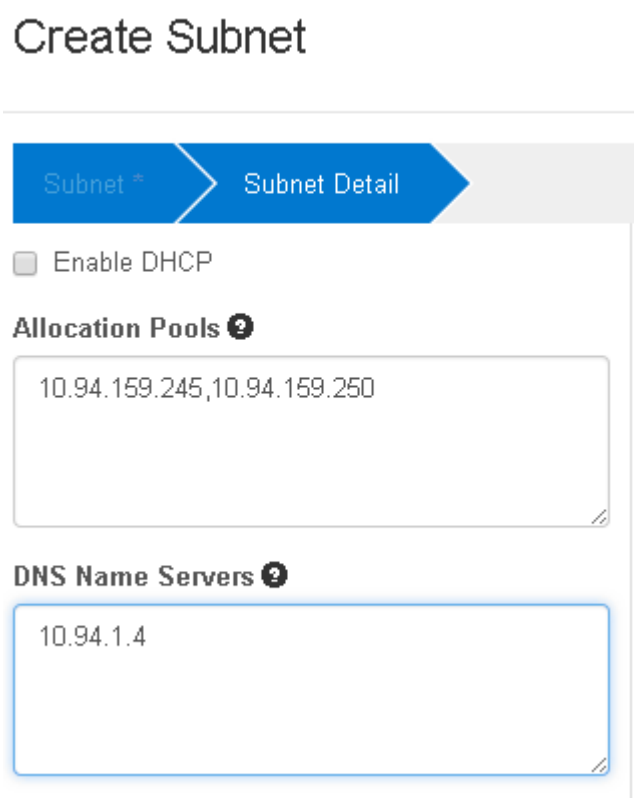
Kuva 30. Julkisen verkon luonti.

Tämän jälkeen on verkkoon luotava aliverkko. Napsauttamalla verkon nimeä päästään sen määrityksiin, josta löytyy ”Create Subnet” -painike.



Kuva 31. Aliverkon luonti.

Aliverkkoa luotaessa on määritettävä nimi sekä verkko-osoite ja yhdyskäytävä, joka on tässä tapauksessa fyysinen reitittävä laite Metropolian laboratorioverkossa. Verkot on mahdollista määrittää myös IPv6-osoitteita käyttäen. Aliverkon määritykset esitetään kuvissa 31 ja 32.

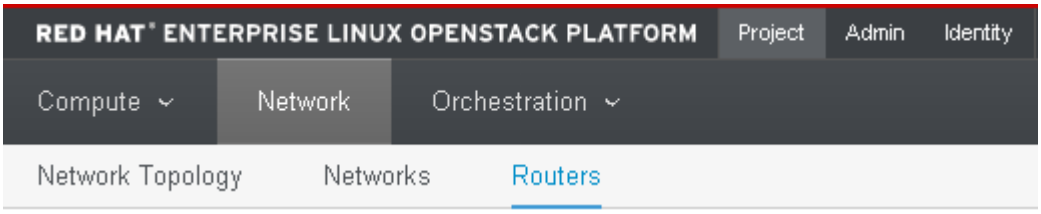


Kuva 32. Aliverkon tarkemmat määritykset.

Määrittäessä julkista verkkoa ylemmällä hallintatasolla Installer-järjestelmässä jätettiin DHCP käyttämättä kaikissa verkoissa lukuunottamatta hallintaverkkoa. Tätä suunnitelmaa noudatetaan myös nyt. Tämän lisäksi on määrittettävä allas, josta instansseille voidaan allokoida kelluvia IP-osoitteita (floating IP), jotka tarjoavat rajapinnan julkiseen verkkoon. Näitä osoitteita hyväksikäyttäen voidaan myöhemmin paljastaa haluttu instanssi ulospäin julkiseen verkkoon. DNS-palvelimen voi myös määrittää tässä vaiheessa.

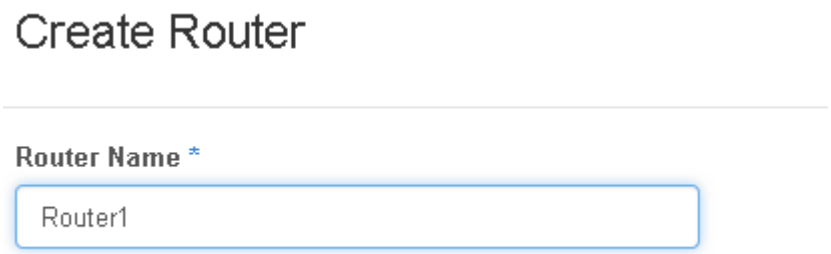
8.1.2 Reitityksen määritys

Seuraavana vaiheena luodaan virtuaalinen reititin julkisen ja tenant-verkon välille. Luonti onnistuu navigoimalla ”Project”-välilehden ”Network”-osioon, josta reitittimien luonti löytyy ”Routers”-kohdan alta kuvan 33 mukaisesti.

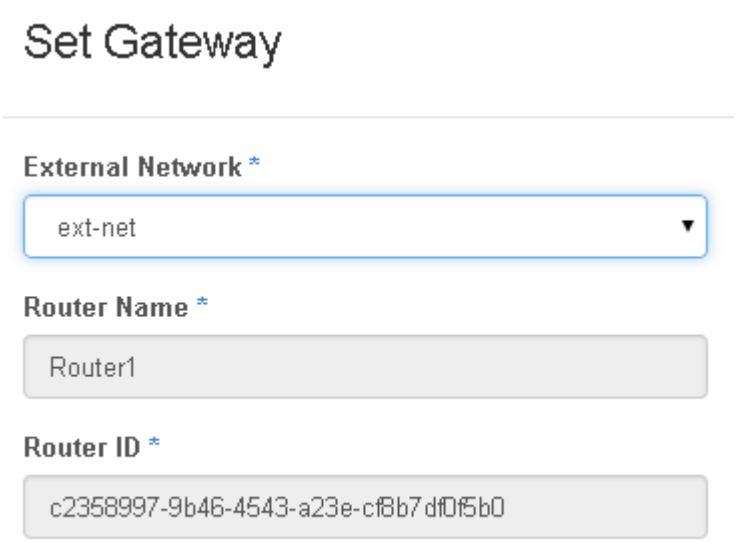


Kuva 33. Reitittimien luontipaneeli.

Luodaan reititin yksinkertaisesti nimeämällä se kuvan 34 mukaisesti, jonka jälkeen määritetään sille ”Gateway” kuvan 35 mukaisesti. Se on käytännössä rajapinta julkisen verkon ja reitittimen välillä. Rajapinta saa tällöin IP-osoitteen, joka on ensimmäinen osoite aikaisemmin määritetystä julkisen verkon osoitealtaasta.



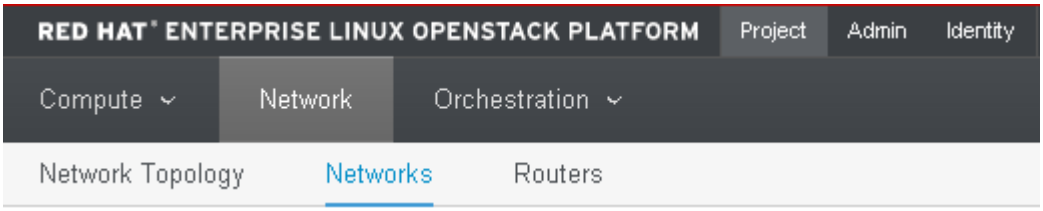
Kuva 34. Reitittimen nimeäminen.



Kuva 35. Rajapinnan luonti julkisen verkon ja reitittimen välille.

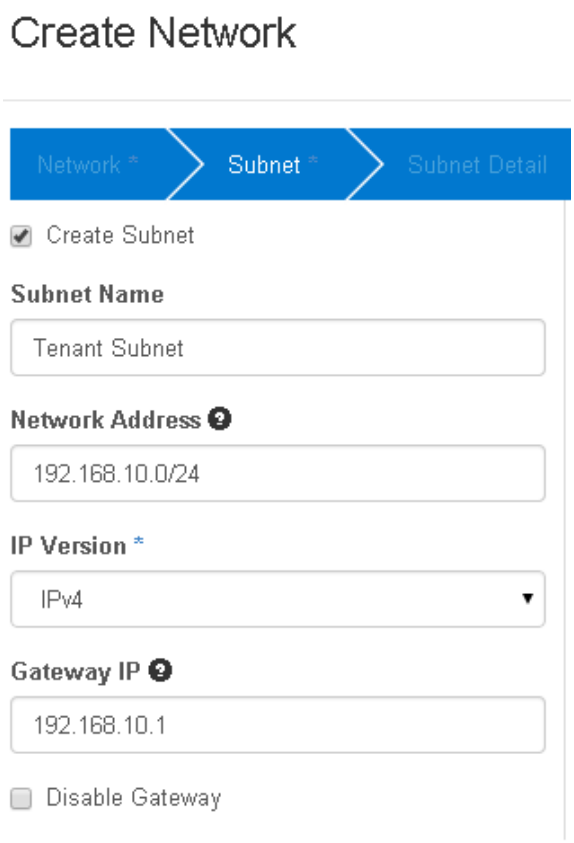
8.1.3 Tenant-verkko

Tämän jälkeen luodaan vielä tenant-verkko instanssien sijoittamiseen navigoimalla samassa paneelissa ”Networks”-osioon, kuten kuvassa 36 on esitetty.



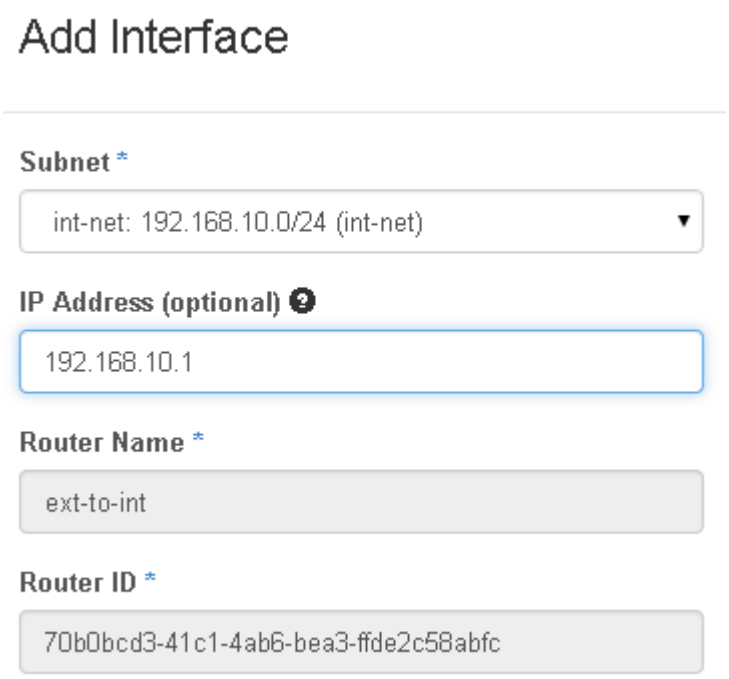
Kuva 36. Tenant-verkon luontipaneeli.

Verkolle määritetään jälleen nimi ja aliverkko. Tällä kertaa voidaan DHCP-palvelu jättää päälle, jotta myöhemmin käynnistettävät instanssit saavat osoitteensa suoraan ilman manuaalista määritystä. Yhdyskäytävän osoitteeksi voi määrittää haluamansa osoitteen. Tenant-verkon luonti on esitetty kuvassa 37.



Kuva 37. Tenant verkon aliverkko.

Tenant-verkon ja aikaisemmin luodun reitittimen välille on vielä luotava rajapinta, joka tehdään saman ”Project”-paneelin ”Routers”-osion alta napsauttamalla ”Add Interface” ja valitsemalla tenant-verkko. IP-osoitteeksi määritetään aikaisemmin tenant-verkkoa luodessa määritetty yhdyskäytävän osoite kuvan 38 mukaisesti.



Kuva 38. Rajapinta tenant verkkoon.

8.1.4 Testaus

Tarvittavat verkot on luotu ja rajapintojen toimivuus voidaan tarkastaa reitittimen yksityskohdista. Rajapinnat tulisi olla ”ACTIVE”-tilassa. Lisäksi julkisen verkon reunalla sijaitsevan reitittimen rajapinnan tulisi vastata ping-pyyntöön. Ping-komennon voi ajaa mistä tahansa laitteesta verkon sisällä, kuten esimerkiksi Installer-järjestelmästä tai Controller-palvelimesta. Kuvassa 39 on esitetty rajapintojen aktiivinen tila sekä kuvassa 40 rajapinnan vastaus ping-pyyntöön.

Interfaces

	Name	Fixed IPs	Status	Type
☐	(a88e5bf1)	192.168.10.1	ACTIVE	Internal Interface
☐	(fb4229d4)	10.94.159.245	ACTIVE	External Gateway

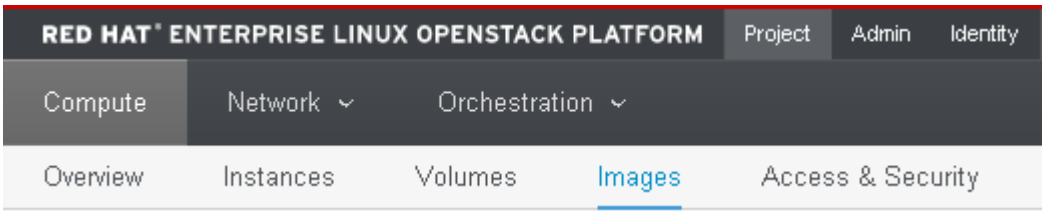
Kuva 39. Rajapinnat ”ACTIVE” tilassa.

```
[root@installer ~]# ping 10.94.159.245
PING 10.94.159.245 (10.94.159.245) 56(84) bytes of data.
64 bytes from 10.94.159.245: icmp_seq=1 ttl=64 time=0.485 ms
64 bytes from 10.94.159.245: icmp_seq=2 ttl=64 time=0.266 ms
64 bytes from 10.94.159.245: icmp_seq=3 ttl=64 time=0.348 ms
64 bytes from 10.94.159.245: icmp_seq=4 ttl=64 time=0.233 ms
```

Kuva 40. Virtuaalisen reitittimen julkisen verkon rajapinta vastaa ping-kutsuun.

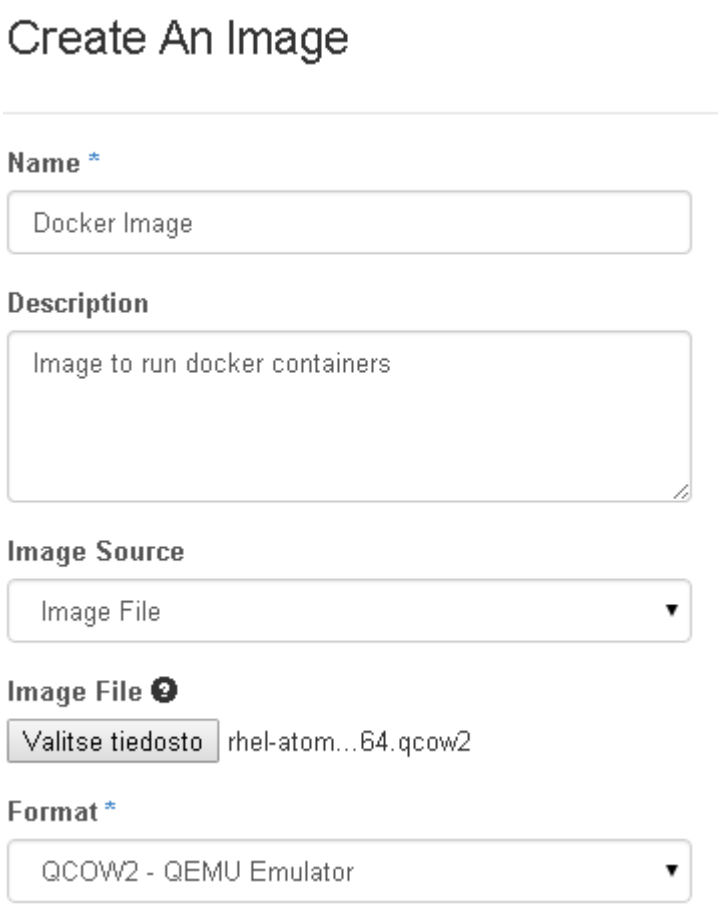
8.2 Levykuvan lataaminen pilveen

Tarvittava levykuva on ladattava ympäristön käyttöön, jotta Docker instassi voidaan luoda myöhemmin käyttäen tätä levykuvaa. Levykuvan voi ladata joko paikallisesta hakemistosta tai URL-osoitteesta. Siirtymällä ”Project”-välilehden ”Compute”-osioon löydetään ”Images”-kohta, josta levykuva voidaan luoda. Kuvassa 41 on esitetty levykuvien hallintapaneeli.



Kuva 41. Levykuvien hallintapaneeli.

Levykuvaa luodessa tarvittavia kenttiä ovat ainoastaan nimi, levykuvan lähde sekä levykuvan muoto, muut kentät voidaan jättää tyhjiksi. Tässä esimerkissä levykuvan muotona käytetään pilvi-levykuvaa QCOW2. Halutessa levykuvalle voidaan määrittää myös arkkitehtuuri ja pienimmät mahdolliset määrät käyttömuistia (RAM) sekä levytilaa. Kuvassa 42 luodaan levykuva Atomic Host -tiedostosta.



Kuva 42. Levykuvan luominen.

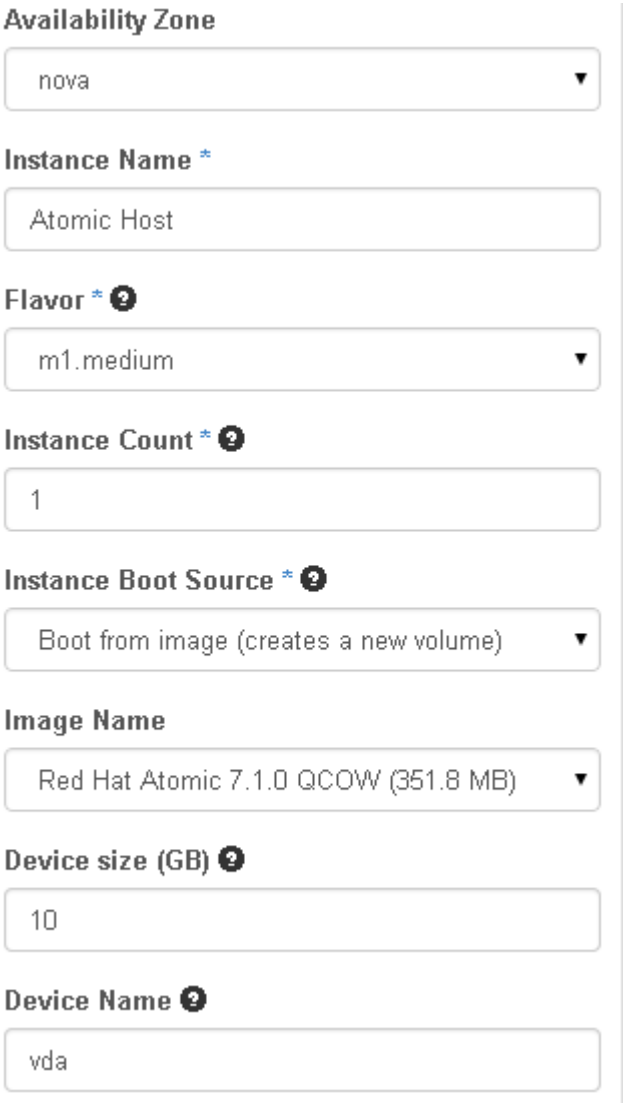
8.3 Secure Shell -avaimen luominen

Ympäristöön luotuun instanssiin pääsee kätevästi käsiksi konsoliyhteyden kautta web-hallinnasta. Jos pääsy halutaan määrittää kuitenkin myös SSH-yhteyttä hyväksikäyttäen, on luotava SSH-avaimet. Tämä voidaan luoda helposti ”Compute”-osion ”Access & Security” -paneelin ”Key Pairs” -osioista. Avaimelle määritetään nimi, jonka jälkeen se luodaan ja yksityinen avain latautuu käyttäjän työasemalle automaattisesti.

8.4 Instanssin luominen

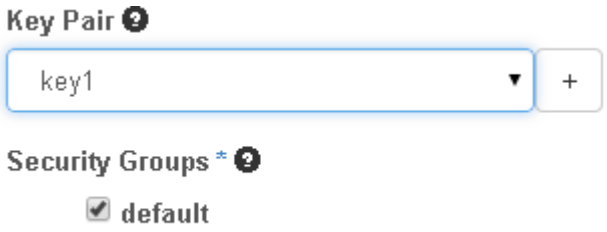
Instanssi eli virtuaalinen palvelin luodaan navigoimalla ”Instances”-osioon, joka löytyy saman välilehden alta kuin aikaisemmin mainitty levykuvien hallintapaneeli. Instanssin

luonnissa on otettava tietysti huomioon levykuvan tarvitsemat minimiresurssit. Red Hat Atomic Hostissa on samat minimivaatimukset kuin Enterprise-versiossa, joten Openstackin esiasennettu ”medium”-kokoluokka riittää täyttämään tarvittavat vaatimukset. Se tarjoaa kaksi ydintä varustettuna neljällä gigatavulla käyttömuistia. Tämän lisäksi instanssille tarjotaan kymmenen gigatavua levypintaa docker-säiliöiden ajamiseksi ja muutoksien tallentamiseksi kuvan 43 mukaisesti.



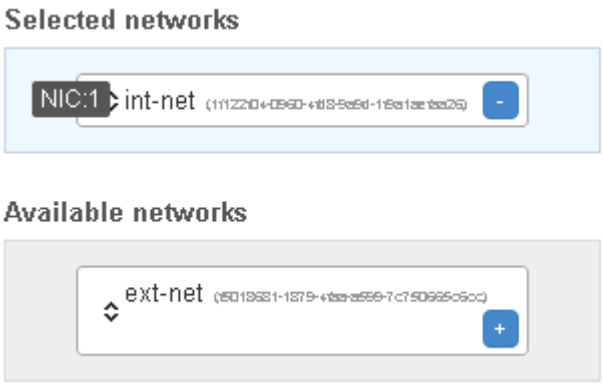
Kuva 43. Instanssin resurssit ja käynnistykseen tarvittava levykuva.

Tämän jälkeen määritetään instanssille SSH-avain, jotta siihen voidaan muodostaa Secure Shell -yhteys kuvan 44 mukaisesti.



Kuva 44. SSH-avaimen määritys.

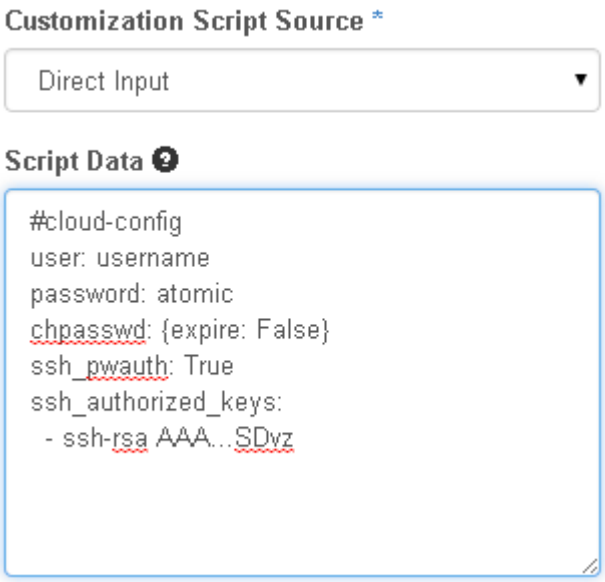
SSH-avaimen määrityksen jälkeen on instanssi sijoitettava virtuaaliseen tenant-verkkoon, kuten kuvassa 45 on tehty.



Kuva 45. Instanssin sijoittaminen tenant-verkkoon.

Viimeinen vaihe ennen instanssin käynnistämistä on räätälöinti, eli instassia muokataan käyttäjän toiveiden mukaisesti. Tässä tapauksessa määritetään ennalta vain pääkäyttäjä, joka voi kirjautua instanssiin pelkästään konsoliyhteyden läpi. Kirjautumisen jälkeen luomme lisää käyttäjiä mm. SSH-yhteyttä varten.

Räätälöintiskripti suoritetaan ensimmäisen käynnistyksen yhteydessä ja siksi se on aloitettava määrityksellä #cloud-config. Määrityksen tarkoitus on kertoa skriptin alkamisesta käyttöjärjestelmälle, jolloin se osaa tunnistaa määritystä seuraavat komennot räätälöinniksi. Aloitusmääritystä seuraa käyttäjänimi ja sen salasana sekä salasanan umpeutuminen. Seuraavat määritykset koskevat SSH-sisäänkirjautumista, sen sallimista ja julkista SSH-avainta. Tähän sijoitettava avain on oltava sama kuin aikaisemmin luotu ja instanssille määritetty avain. Räätälöintiskripti on kuvattuna kuvassa 46.



Kuva 46. Instanssin räätälöintiskripti.

Räätälöinnin jälkeen voidaan instanssi käynnistää. Ensimmäinen käynnistys vie huomattavan paljon aikaa ympäristön monen virtualisoidun kerroksen takia. Tämän lisäksi on odotettava, että käyttöjärjestelmä ehtii suorittaa räätälöintiskriptin ennen kuin instanssiin voi kirjautua.

Kirjautuminen instanssiin tapahtuu siis konsoliyhteyden avulla web-hallintapaneelista siirtymällä instanssin tietoihin napsauttamalla sen nimeä. Sen takaa löytyy oheinen kuvan 47 paneeli ja kuvan 48 konsoliyhteys. Konsoliyhteys hyodyntää VNC (Virtual Network Computing) -teknologiaa, mikä mahdollistaa käyttöjärjestelmän graafisen etäkäytön.

Kuva 47. Instanssin tiedot.

Siirtymällä ”Console”-välilehteen voidaan kirjautua instanssiin käyttämällä räätälöintiskriptissä luotuja käyttäjätunnuksia. SSH-yhteys ei vielä toimi, sillä

instanssille ei ole määritetty IP-osoitetta julkisesta verkosta, eikä pääkäyttäjän kirjautuminen SSH-yhteydellä ole oletuksena mahdollista.

Kuva 48. Konsoli-ikkuna instanssiin.

Seuraavaksi voidaan määrittää käyttäjä SSH-yhteyttä varten ajamalla seuraavat komennot.

- `useradd ”käyttäjä”`
- `passwd ”käyttäjä”`
- `useradd -g ”käyttäjä” wheel`

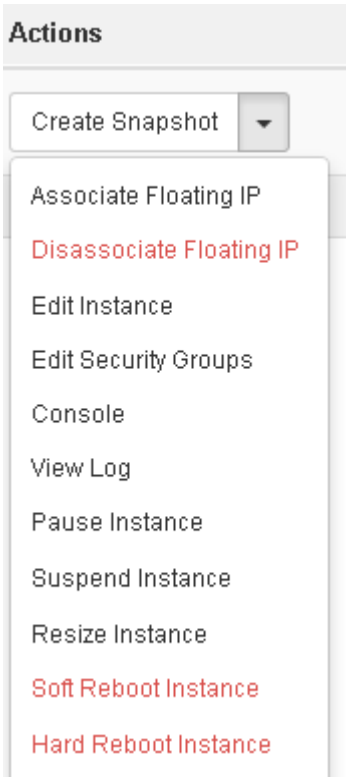
Käyttäjä on nyt luotu ja lisätty wheel-ryhmään. Tämän lisäksi on luotu käyttäjä lisättävä /etc/sudoers -tiedostoon seuraavalla komennolla.

- `echo ”käyttäjä ALL=(ALL) ALL” >> /etc/sudoers`

Tämän jälkeen luodulla käyttäjällä voidaan toteuttaa muutoksia pääkäyttäjän oikeuksin, jotka tarvitaan docker-säiliöiden ajamiseen, sekä instanssin rekisteröimiseen Red Hatin palvelimille.

8.5 Atomic Host -instanssin määritykset

Ennen docker-säiliöiden ajamista on instanssi rekisteröitävä Red Hatin palvelimille sen tuotteistuksen takia. Tähän tarvitaan kuitenkin rajapinta julkiseen verkkoon, joka voidaan tarjota instanssille määrittämällä sille ”Floating IP”. Kelluva IP-osoite tarjoaa instanssille rajapinnan aikasemmin luotuun julkiseen verkkoon, mikä mahdollistaa pääsyn julkiseen internetiin instanssin rekisteröimistä varten. Kelluva IP-osoite määritetään instanssille napsauttamalla instanssin oikealla puolella sijaitsevaa Actions-paneelista löytyvää lisämäärityksiä tarjoavaa nuolta kuvan 49 mukaisesti.



Kuva 49. Kelluvan IP-osoitteen määritys.

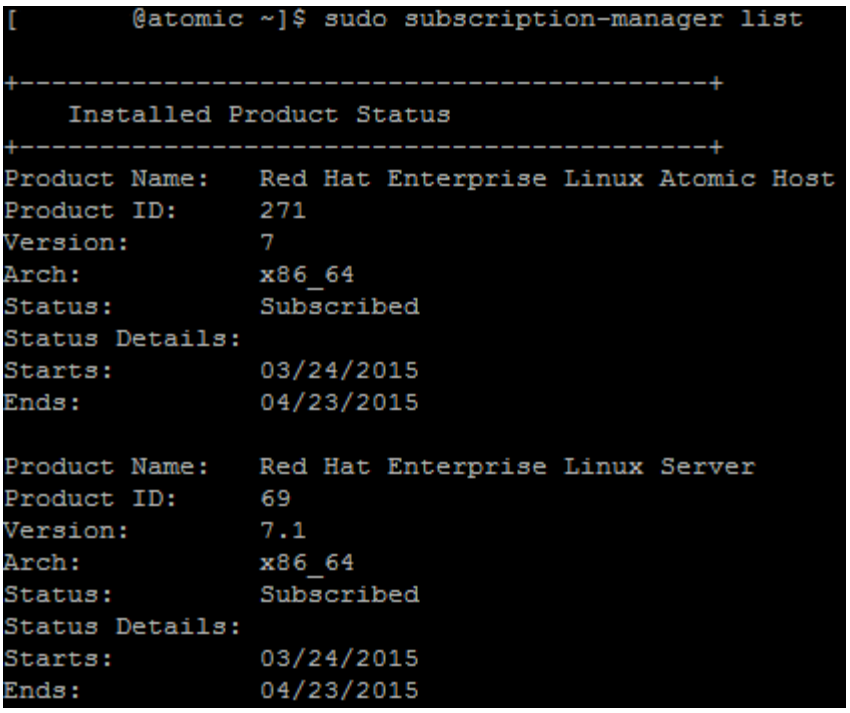
Määritettävä IP-osoite on järjestyksessä seuraava suurempi osoite aikaisemmin määritetystä altaasta. Altaan ensimmäinen osoite määrittyy automaattisesti virtuaalisen reitittimen julkisen verkon rajapinnalle. Kelluva osoite ei näy instanssille millään tavoin, vaan sen näkökulmasta sillä on vain yksi rajapinta tenant-verkkoon. Openstackin Neutron-komponentti on vastuussa liikenteen ohjaamisesta kelluvaan IP-osoitteeseen.

Osoitteen määrittämisen jälkeen voidaan instanssiin ottaa SSH-yhteys käyttämällä kelluvaa IP-osoitetta sekä aikaisemmin luotua käyttäjätunnusta sekä salasanaa tai SSH-avainta. Atomic Host instanssin rekisteröiminen tapahtuu yhdellä komennolla. Käyttäjänimellä tarkoitetaan Red Hat portaalin käyttäjänimeä, jolla on oltava aktiivinen lisenssi Red Hat Enterprise Linuxille.

- `sudo subscription-manager register –username=käyttäjänimi --auto-attach`

Docker-säiliöitä voi ajaa myös rekisteröimättömällä Atomic Host -instassilla, mutta tällöin menetetään mahdollisuus ajaa Red Hatin virallisia Red Hat Enterprise Linux -säiliöitä. Tämän lisäksi päivittäminen käyttäen automaattista yum-paketinhallintaa

virallisisista Red Hatin lähteistä on poissuljettu. Rekisteröinnin tilan voi tarkastaa olevan kuvan 50 mukaisesti.



Kuva 50. Rekisteröity Atomic Host instanssi.

8.6 Atomic Host -instanssin päivittäminen

Instassin voi päivittää käyttäen automaattista päivitystä. Se tapahtuu ajamalla komennot:

- `sudo atomic host upgrade`
- `sudo systemctl reboot`

Järjestelmä päivittyy ja käynnistää itsensä uudelleen. Mikäli halutaan vieriä taaksepäin asennuksien välillä päivittämättömään tilaan voidaan suorittaa seuraavat komennot:

- `sudo atomic host rollback`
- `sudo systemctl reboot`

Tällöin järjestelmä hylkää tehdyt muutokset ja palautuu aikaisempaan tilaan uudelleenkäynnistyksen jälkeen. Toiminto on kätevä esimerkiksi tilanteissa, joissa konfiguraatiomuutokset tai päivittäminen on aiheuttanut ongelmia. Tällöin voidaan helposti palata toimivaan tilaan ilman suuria toimenpiteitä (Red Hat. 2015).

9 Docker-säiliön ajaminen

9.1 Docker rekisterit

Atomic Host -järjestelmä on luotu ajamaan docker-säiliöitä, ja kaikki tarvittavat paketit on esiasennettu levykuvaan, joten asennuksia ei tarvitse tehdä ennen säiliöiden käyttöä.

Docker-säiliöiden levykuvia voidaan hakea joko etävarastoista tai niitä voidaan käyttää paikallisesta, yksityisestä varastosta. Tässä työssä käytämme Red Hatin virallista rekisteriä Red Hat Enterprise Linux -säiliön hakemiseen. Säiliön hakeminen tapahtuu ajamalla seuraava komento Atomic Host -instanssissa:

- `sudo docker pull registry.access.redhat.com/rhel`

Komennossa “pull” hakee säiliön levykuvan paikalliseen järjestelmään käytettäväksi. Tämän jälkeen osoitetaan rekisterin verkko-osoite, jota seuraa ”/rhel”-parametri, eli haettavan levykuvan nimi. Rhel on lyhenne sanoista Red Hat Enterprise Linux. Komentoa voitaisiin myös tarkentaa määrittämällä viimeiseen osaan esimerkiksi ”rhel:latest”, jolloin haettaisiin uusin levykuva halutusta rekisteristä (Red Hat. 2015).

9.2 Palvelun isännöinti Docker-säiliössä

Tässä työssä ajetaan Red Hat Enterprise Linux -säiliö hyödyntämällä docker-virtualisointia. Säiliön demonstroimiseksi käynnistetään yksinkertainen web-palvelin säiliön sisään ja määritetään se näkymään julkiseen verkkoon, jotta säiliön toimivuus voidaan todeta myös käytännön tasolla. Docker-säiliö ja sen sisällä ajettava web-palvelin käynnistetään ajamalla seuraavat komennot pääkäyttäjän oikeuksin Atomic Host -instanssissa.

- `mkdir -p /var/www/html`
- `restorecon -Rv /var/www`
- ```
  docker run -d -p 192.168.10.141:8000:8000 --name="python_web" \
        -v /usr/sbin:/usr/sbin -v /usr/bin:/usr/bin -v /usr/lib64:/usr/lib64 \
        -w /var/www/html -v /var/www/html:/var/www/html rhel \
        /bin/python -m SimpleHTTPServer 8000
  ```

Ensimmäinen komento tekee kansion haluttuun sijaintiin järjestelmässä, jonka jälkeen kansion turvallisuusprofiili muutetaan rekursiivisesti oletusarvoihin. Tämän jälkeen ajetaan docker-säiliö parametrillä ”-d”, jotta säiliö jää taustalle. Sen lisäksi määritetään portti ja rajapinta, johon docker-säiliö sidotaan Atomic Host -järjestelmässä. Tässä esimerkissä se on sidottu ainoaan verkkorajapintaan, josta on pääsy julkiseen verkkoon (Red Hat. 2015).

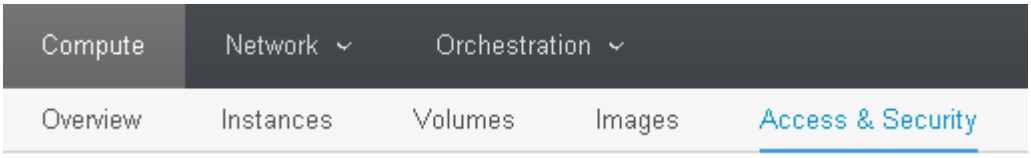
Säiliölle määritetään myös nimi sekä siihen liitetään ”-v”-parametrillä halutut kansiot Atomic Host -järjestelmästä sekä ”-w"-parametrilla käyttökansio. Tämän lisäksi ajetaan ”SimpleHTTPServer”-moduuli /bin/python-komennolla ja sidotaan se porttiin 8000 säiliössä (Red Hat. 2015).

Tämän jälkeen voidaan luoda index.html-tiedosto käyttökansion sijaintiin, jonne kirjoitetaan yksinkertainen html-koodi toimivuuden testaamiseksi kuvan 51 mukaisesti.

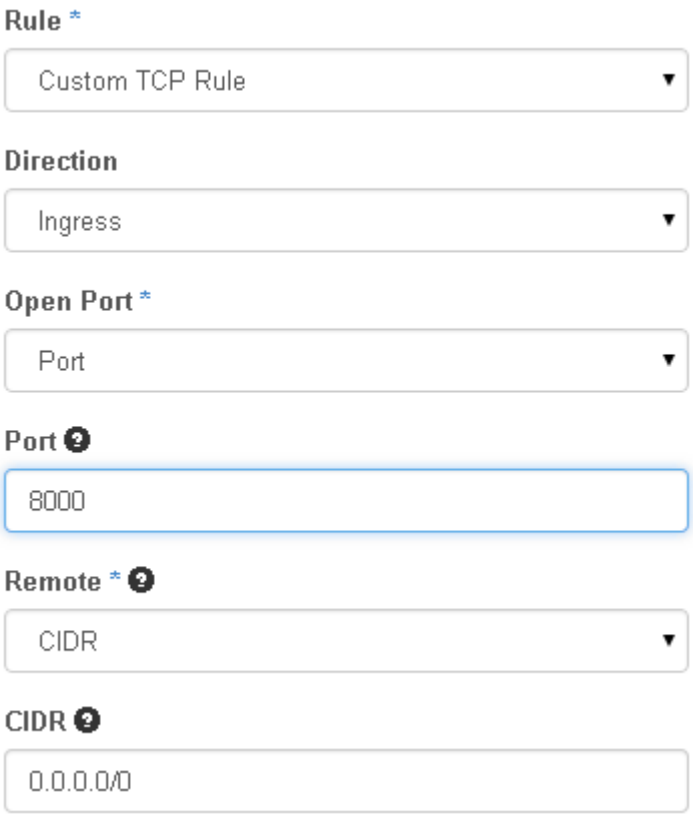
```
-bash-4.2# cat /var/www/html/index.html
<html>
<head>
<title> Web-palvelu docker-sailion sisalla toimii </title>
</head>
<body>
 Tama web-palvelu ajetaan docker-sailion sisalla </>
<br>
 This web-server runs inside a docker container 
</body>
</html>
```

Kuva 51. Esimerkkitiedosto web-sivua varten.
```

Oletuksena liikenne porttiin 8000 on kielletty, ja se on avattava Horizon web-hallinnasta ”Compute”-välilehden takaa löytyvästä ”Access & Security” -valikon ”Security Groups” –osiosta kuten kuvassa 53 on esitetty.

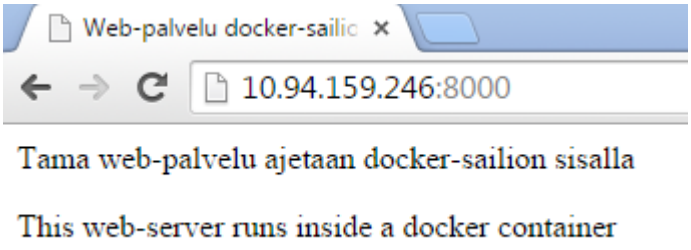


Kuva 52. Palomuurisääntöjen hallintapaneeli



Kuva 53. Portin 8000 avaaminen.

Portin avaamisen jälkeen voidaan navigoida selaimella kelluvaan IP-osoitteeseen ja määritettää http-liikenne käyttämään porttia 8000. Selaimen tulisi näyttää index.html-tiedostoon kirjoitettu teksti kuten kuvassa 54.



Kuva 54. Selainikkuna osoitettu kelluvaan IP-osoitteeseen ja porttiin 8000.

Säiliö voidaan myös pysäyttää, käynnistää uudelleen ja poistaa seuraavilla komennoilla.

- `docker stop python_web`
- `docker start python_web`
- `docker rm python_web`

Mikäli säiliötä ei poisteta, on sen uudelleenkäynnistäminen huomattavasti yksinkertaisempaa kuin uuden säiliön luominen alusta, sillä säiliöt jäävät Atomic Host -järjestelmään talteen myöhempää käyttöä varten, ja ne voidaan käynnistää helposti yhdellä komennolla ilman suurempaa määrittelyä.

9.3 Levykuvan luominen Docker-säiliöstä

Säiliöstä voidaan myös tehdä uusi levykuva, jolloin siihen voidaan valmiiksi esiasentaa esimerkiksi web-palvelu. Levykuvaa käyttäen voitaisiin säiliö esimerkiksi viedä toiseen järjestelmään helposti ajettavaksi kokonaisuudeksi, sillä se sisältää kaikki tarvittavat paketit palvelun ajamiseen. Levykuvan luominen tapahtuu helposti parilla komennolla Atomic Host -järjestelmässä.

- `docker run –i rhel /bin/bash –c “yum install –y httpd”;`
- `docker ps`
- `docker commit fd0saskw24 rhel_http`

Ensimmäinen komento luo uuden docker-säiliön Red Hat Enterprise Linux -levykuvasta ja asentaa siihen httpd-paketin ja kaikki tarvittavat riippuvuudet, jotka tarvitaan web-palvelun ajamiseen. Toinen komento listaa kaikki docker-säiliöt, jotka ovat läsnä järjestelmässä mukaan lukien seuraavassa komennossa tarvittavan säiliön id-numeron. Tätä numeroa hyväksikäyttäen luodaan levykuva tietystä säiliöstä ja sille annetaan nimi ”rhel_http” (Red Hat. 2015).

10 Johtopäätökset

Hallittavan Openstack-ympäristön luominen ja käyttöönotto helpottuu huomattavasti, mikäli käytetään tuotteistettua ympäristöä. Tämän lisäksi hallintatyökalut ovat helpompia ja ymmärrettävämpiä eikä komentorivi pohjaista käsittelyä juurikaan tarvita. Ainoana haittapuolena on jakelijoiden lisäämät koodit, jotka eivät ole julkisesti jaossa eivätkä täten myöskään toisten osapuolien hyödynnettävissä, jolloin menetetään osa avoimen lähdekoodin parhaista puolista.

Työssä onnistuttiin hyvin toteuttamaan haluttu Openstack-ympäristö sekä hyödyntämään sitä docker-säiliöiden ylläpitoon. Ainoastaan monet virtualisoidut kerrokset aiheuttivat järjestelmään hitautta, mikä oli odotettavissa. Ympäristö onnistuttiin kuitenkin rakentamaan niin, että se on laajennettavissa uusilla resursseilla ilman merkittäviä katkoksia, jolloin virtualisoituja kerroksia voidaan vähentää asteittain.

Ympäristön asennuksen aikana ei kohdattu merkittäviä ongelmia ja tuotteistetut sovelluspaketit toimivat, kuten niiden oletetaan. Suunnittelu ja asennukset tehtiin aikarajoitteisen lisenssin alaisuudessa, mikä jätti työn käytännön osuudelle 60 päivää aikaa. Tässä ajassa ehdittiin hyvin rakentaa ympäristö aikaisemmin tehdyn suunnitelman pohjalta sekä testata sen toimivuus.

Seuraava työvaihe ympäristön kannalta on siirtää se Metropolian laboratorioverkosta rakenteilla olevaan Internet Exchange -ympäristöön, jonka ominaisuudet vastaavat internet-palveluntarjoajan (ISP) piirteitä. Tämän vaiheen toteutuessa voidaan mahdollisille tuleville asiakkaille käyttöönottaa pilvipalveluita hyödyntäen tämän opinnäytetyön mallia. Tällöin tämän insinöörityön pilvimalli muuttuu hetkessä yksityisestä julkiseksi, jolloin lisensointi, tietoturva sekä säännöt muodostuvat tärkeäksi osaksi ympäristöä.

Lähteet

Apache License and Distribution FAQ. 2012. Verkkodokumentti. Apache. http://www.apache.org/foundation/license-faq.html. Luettu 24.2.2015.

Branca, Leone. 2014. Verkkodokumentti. IBM. Public, private and dynamic hybrid cloud: What's the difference?. http://www.smartercomputingblog.com/cloud-infrastructure/public-private-hybrid-cloud/. Luettu 24.2.2015.

Butler, Brandon. 2014. Verkkodokumentti. Networkworld. HP leapfrogs Red Hat to become top contributor to OpenStack. http://www.networkworld.com/article/2686462/public-cloud/hp-leapfrogs-red-hat-to-become-top-contributor-to-openstack.html. Luettu 24.2.2015.

Docker: Containers for the Masses. 2014. Verkkodokumentti. Patg.net. http://patg.net/containers,virtualization,docker/2014/06/05/docker-intro/. Luettu 16.4.2015.

IaaS, PaaS, SaaS (Explained and Compared). 2015. Verkkodokumentti. Apprenda Inc. http://apprenda.com/library/paas/iaas-paas-saas-explained-compared/. Luettu 24.2.2015.

Mell, Peter & Grance, Timothy. 2011. Verkkodokumentti. NIST. The NIST Definition of Cloud Computing. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. Luettu 24.2.2015.

Merkel, Dirk. 2014. Verkkodokumentti. Linuxjournal. Docker: Lightweight Linux Containers for Consistent Development and Deployment. http://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment. Luettu 17.3.2015.

Nippes, Daniel. 2014. Verkkodokumentti. Dataconomy. Data Ownership In The Cloud. http://dataconomy.com/data-ownership-in-the-cloud/. Luettu 24.2.2015.

OpenStack Folsom Architecture. 2012. Verkkodokumentti. . Ken Pepple. OpenStack Folsom Architecture. http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/. Luettu 09.04.2015.

Perlow, Jason. 2008. Verkkodokumentti. ZDnet. Parallels releases its Hypervisor -- on the Mac. http://www.zdnet.com/article/parallels-releases-its-hypervisor-on-the-mac/. Luettu 8.4.2015.

Pervilä, Markku. 2014. Verkkodokumentti. Tietoviikko. Hybridipilvi purjehtii navakassa myötätuulessa. http://www.tivi.fi/CIO/2014-07-11/Hybridipilvi-purjehtii-navakassa-my%C3%B6t%C3%A4tuulessa-3211446.html. Luettu 24.2.2015.

Rackspace Support. 2013. Verkkodokumentti. Rackspace. Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS. http://www.rackspace.com/knowledge_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas. Luettu 24.2.2015.

Red Hat. 2015. Verkkodokumentti. Red Hat. DEPLOYING OPENSTACK: ENTERPRISE ENVIRONMENTS. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Installer_and_Foreman_Guide/. Luettu 24.2.2015.

Red Hat. 2015. Verkkodokumentti. Red Hat. Getting Started with Red Hat Enterprise Linux Atomic Host. https://access.redhat.com/articles/rhel-atomic-getting-started#install. Luettu 31.3.2015.

Red Hat. 2015. Verkkodokumentti. Red Hat. Get Started with Docker Formatted Container Images on Red Hat Systems. https://access.redhat.com/articles/881893#getatomic. Luettu 31.3.2015.

Vaughan-Nichols, Steven J.. 2014. Verkkodokumentti. ZDnet. What is Docker and why is it so darn popular?. http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/. Luettu 17.3.2014.

Weins, Kim. 2015. Cloud Computing Trends. 2015. Verkkodokumentti. Rightscale. State of the Cloud Survey. http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2015-state-cloud-survey. Luettu 24.2.2015.

Zitzman, Sharone. 2014. Verkkodokumentti. Cloudify. OpenStack Wiki in Short – A Quick Guide to Open Cloud. http://getcloudify.org/2014/07/18/openstack-wiki-open-cloud.html. Luettu 9.3.2014.

Liite 1: Openstack komponentit (Sharone Zitzman. 2015)

Component Name	Description
OpenStack Compute (Nova)	OpenStack compute (codename: Nova) is the component which allows the user to create and manage virtual servers using the machine images. It is the brain of the Cloud. OpenStack compute provisions and manages large networks of virtual machines.
Block Storage (Cinder)	This component provides persistent block storage to running instances. The flexible architecture makes creating and managing block storage devices very easy.
Object Storage (Swift)	This component stores and retrieves unstructured data objects through the HTTP based APIs. Further, it is also fault tolerant due to its data replication and scale out architecture.
OpenStack Networking (Neutron)	It is a pluggable, scalable and API-driven system for managing networks. OpenStack networking is useful for VLAN management, management of IP addresses to different VMs and management of firewalls using these components.
Identity Service (Keystone)	This provides a central directory of users mapped to the OpenStack services. It is used to provide an authentication and authorization service for other OpenStack services.
OpenStack Image Service (Glance)	This provides the discovery, registration and delivery services for the disk and server images. It stores and retrieves the virtual machine disk image.
OpenStack Telemetry Service (Ceilometer)	It monitors the usage of the Cloud services and decides the billing accordingly. This component is also used to decide the scalability and obtain the statistics regarding the usage.
Dashboard (Horizon)	This component provides a web-based portal to interact with all the underlying OpenStack services, such as NOVA, Neutron, etc.

Orchestration Heat	This component manages multiple Cloud applications through an OpenStack-native REST API and a CloudFormation-compatible Query API.
Database as a Service (Trove)	Trove is Database as a Service for OpenStack. It's designed to run entirely on OpenStack, with the goal of allowing users to quickly and easily utilize the features of a relational database without the burden of handling complex administrative tasks. Cloud users and database administrators can provision and manage multiple database instances as needed. Initially, the service will focus on providing resource isolation at high performance while automating complex administrative tasks including deployment, configuration, patching, backups, restores, and monitoring.
Messaging as a Service (Marconi)	Marconi is a cloud messaging and notification service for developers building applications on top of OpenStack. The service features a web-friendly HTTP API, which developers can use to send messages between the various components of their SaaS and mobile applications, using a variety of communication patterns. Underlying the API is an efficient messaging engine designed with scalability and security in mind.