

OpenStack ympäristön luonti kotiloissa

Antti Salminen



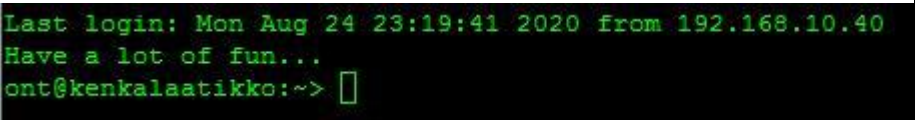
Tekijä(t) Salminen, Antti	
Koulutusohjelma Tietojenkäsittely	
Raportin/Opinnäytetyön nimi OpenStack ympäristön luonti kotiloissa.	Sivu- ja liitesivumäärä 79 + 0
<p>Tämän opinnäytetyön tarkoituksena on luoda OpenStack pilvilaskenta-alusta kotiloissa. Ensin käyn läpi keskeisimmät pilvipalvelukäsitteet. Esittelen myös suurimmat kaupalliset pilvipalvelujen tarjoajat ja vertailen niiden palvelukatalogeja jättäen kuitenkin huomioimatta asiakaskokemuksen, sillä työni ei keskity sen analysointiin. Sitten siirryn esittelemään OpenStackin.</p> <p>Työni teknisessä osiossa esittelen työssä käytetyn laitteiston ja luon OpenStackillä pilvilaskenta-alustan käyttäen minimaalista OpenStack asennusta. Minimaaliseen asennukseen kuuluu viisi OpenStackin keskeisintä komponenttia. Tämän lisäksi asennan vielä kuudennen komponentin, joka mahdollistaa käyttäjäystävällisen Hallintapaneelin käytön. Laitteistosta johtuen asennus ei mene aivan ohjeiden mukaisesti, joten esittelen ja ratkon myös vastaan tulleita ongelmia.</p> <p>Lopputuloksena on alusta, joka toimii, tavallaan. Ympäristö toimii, komponentit keskustelevat keskenään ja suorittavat toimintonsa. Seuraava askel olisi etsiä levykuva, joka toimii juuri näille laskentakoneille. Ongelma on suoritinarkkitehtuurissa. ARM-arkkitehtuuri ei kykene virtualisoimaan x86 arkkitehtuuria.</p> <p>Lopuksi tiivistän opinnäytetyöni yleisen hyödyn ohjeena ja pilvipalveluiden toimintaa avaavana raporttina. Lisäksi tiedostan kotilojen eroavaisuuden konesalioloihin.</p>	
Asiasanat Pilvipalvelu, pilvilaskenta, pilvi, OpenStack, TeinttseJaSäästin	

Sisällys

Sanasto	1
1 Johdanto	3
2 Mitä ovat pilvipalvelut?	3
2.1 AWS, Microsoft Azure ja Google Cloud.....	4
2.2 OpenStack.....	4
3 Laitteisto.....	5
4 Asennuksen valmistelu.....	6
4.1 Käyttöjärjestelmän asennus	6
4.2 Esivalmistelut, <i>kenkalaatiiko</i>	7
4.3 Esivalmistelut, <i>raspit</i>	9
4.4 Network Time Protocol (NTP) -asennus.....	11
4.5 OpenStackin pakettien haku ja asennus	14
4.6 SQL tietokanta	17
4.7 Viestijono	19
4.8 Memcached	20
4.9 Etcad	21
5 OpenStackin asennus	24
5.1 Identiteetti palvelu	24
5.1.1 Alkuvalmistelu	24
5.1.2 Keystone palvelu.....	25
5.1.3 Apache HTTP palvelin	26
5.1.4 Toimialueen, projektin, käyttäjien ja roolien luonti	27
5.1.5 Toiminnan varmistus	29
5.1.6 OpenStack ympäristömuuttaja skriptit	31
5.2 Levykuvapalvelu	32
5.2.1 Esivalmistelut	32
5.2.2 Komponenttien asennus ja määrittely.....	34
5.2.3 Toiminnan testaus.....	36
5.3 Resurssien seuranta palvelu	37
5.3.1 Esivalmistelut	37
5.3.2 Komponenttienasennus ja määrittely.....	39
5.3.3 Asennuksen toiminnan varmistus.....	40
5.4 Laskenta -palvelu.....	42
5.4.1 Esivalmistelut	42
5.4.2 Pakettien asennus ja määrittely kontrolleri koneelle	45
5.4.3 Pakettien asennus ja määrittely laskentakoneelle	50
5.4.4 Laskenta koneiden rekisteröinti kontrollerille	53

5.5	Verkko -palvelu	54
5.5.1	Esivalmistelut	54
5.5.2	Verkon asetusten määrittäminen	56
5.5.3	Virtuaaliverkkojen luonti	63
5.6	Hallintapaneeli	63
5.6.1	Pakettien asennus ja konfigurointi	64
6	Ensimmäisen istunnon käynnistys	68
6.1	Vian selvitys	71
6.2	Vian korjaus	71
6.2.1	Kernelin päivitys	71
6.2.2	Qemun päivitys	74
6.3	Nyt se sitten toimii?	75
7	Pohdinta	77
	Lähteet	78

Sanasto

Autentikaatio valtuus, <i>Authentication acces token</i>	Tietue, joka sisältää käyttäjän tunnistukseen tarvittavat tiedot.
<i>Flavor</i>	Tietue, joka sisältää virtualisoitavan koneen rautatason määrittäykset.
Git sekä GitHub	Git on suosittu ohjelmistokehitykseen suunniteltu versionhallinta ohjelmisto. GitHub on suosittu julkinen Online varasto Git:ille
HTTP -palvelu	Palvelu verkkosivujen näyttämiseen.
IP-osoite	Numeraalinen osoite, jonka verkkolaite tarvitsee voidakseen keskustella muiden verkon laitteiden kanssa. Tulee aina olla yksilöllinen verkon sisäisesti. Verrattavissa esimerkiksi puhelinnumeroon.
Isäntänimi <i>Hostname</i>	Koneen yksilöllinen nimi, jolla se voidaan tunnistaa toimialueen verkossa.
Kernel	Käyttöjärjestelmän keskeisin osa. Toimii tulkkina laitteiston rautatason sekä käyttöjärjestelmän välillä.
Komentorivi	Tämä: 
Levykuva - <i>Image</i>	Tiedosto, joka sisältää kaiken käyttöjärjestelmän käynnistykseen tarvittavan. Voi olla vain purettu, jolloin käyttöjärjestelmä asennetaan laitteeseen tai pikakuva järjestelmän tilasta, johon palataan levykuva käynnistettäessä.
Metapaketti	Ohjelmiston asennuspaketti, joka sisältää tiedot useamman ohjelmiston asennukseen. Esimerkiksi kuvitteellinen metapaketti <i>foometa</i> saattaisi sisältää asennusohjeet ohjelmistoihin <i>foo</i> , <i>bar</i> sekä <i>nano</i> , mutta käyttäjän tarvitsee asentaa vain <i>foometa</i> .
Ohjelmointirajapinnan loppupää - <i>API (Application programming interface) endpoint</i>	Ohjelmointirajapinta on määritelmä, jonka avulla ohjelmat vaihtavat tietoa keskenään. (Digiarjessa Blogi. 2016.) Loppupää on erityisesti määritelty TCP/IP-portti tähän tarkoitukseen.

<p>OSI-malli - <i>Open Systems Interconnection Reference Model</i></p>	<p>Seitsenkerroksinen tiedonsiirtoprotokollien kuvaus.</p> <table border="1" data-bbox="619 181 940 539"> <tr><td>7. Sovelluskerros</td></tr> <tr><td>6. Esitystapakerros</td></tr> <tr><td>5. Istuntokerros</td></tr> <tr><td>4. Kuljetuskerros</td></tr> <tr><td>3. Verkkokerros</td></tr> <tr><td>2. Siirtokerros</td></tr> <tr><td>1. Fyysinen kerros</td></tr> </table> <p>(Tampere University of Technology, Department of Computer Science. 2002.)</p>	7. Sovelluskerros	6. Esitystapakerros	5. Istuntokerros	4. Kuljetuskerros	3. Verkkokerros	2. Siirtokerros	1. Fyysinen kerros
7. Sovelluskerros								
6. Esitystapakerros								
5. Istuntokerros								
4. Kuljetuskerros								
3. Verkkokerros								
2. Siirtokerros								
1. Fyysinen kerros								
<p>Pakettivarasto; repo - <i>repository</i></p>	<p>Tiedostovarasto, jonne kaikki asennettavissa olevien ohjelmistojen paketit tallennetaan, yleensä käyttöjärjestelmän ylläpitäjien ja kehittäjien toimesta.</p>							
<p>SSH</p>	<p>Secure Shell on protokolla salatulle tietoliikenteelle. Yleisin käyttökohte on suojattu komentorivi etäyhteys. (SSH Communications. 2020.) Voidaan käyttää myös muun liikenteen (esimerkiksi HTTP) tunnelointiin, eli salaukseen.</p>							
<p>TCP/IP - <i>Transmission Control Protocol / Internet Protocol</i></p>	<p>Usean Internet-liikennöinnissä käytettävän protokollan yhdistelmä. Sisältää määrykset ja protokollat kaikkeen verkon yli tapahtuvaan liikenteeseen.</p>							
<p>Tervapallero</p>	<p>Yleisesti käytetty nimitys Linux-ympäristöissä suosituille pakatuille .tar-tiedostoille.</p>							
<p>Toimialue - <i>Domain</i></p>	<p>Hallittu kokonaisuus, joka koostuu erilaisista verkkoon liitetystä laitteista, käyttäjätunnuksista ja itse verkosta. (HowToGeek. 2017.) Esimerkkinä yrityksen sisäinen verkko, laitteinaan tietokoneet ja tulostimet.</p>							
<p>YaST (Yet another Setup Tool)</p>	<p>openSUSEn järjestelmänhallintatyökalu, jolla voidaan keskitetysti hoitaa lähes kaikki käyttöjärjestelmän hallintaan liittyvät toimet, kuten luoda käyttäjätunnuksia, hallinnoida oikeuksia sekä asentaa ja poistaa ohjelmistoja. (openSUSE YaST. 2020.) Verrattavissa Windows-käyttöjärjestelmien Ohjauspaneeliin.</p>							

1 Johdanto

Työn tavoitteena on luoda OpenStack laskentaympäristö kohtuullisella budjetilla ja helposti saatavilla olevilla laitteilla. Ennen asennuksen läpikäyntiä on kuitenkin hyvä varmistaa, että lukija ymmärtää mistä puhutaan. Aluksi käydään hieman läpi mitä tarkoittaa pilvi, pilvipalvelu, pilvilaskenta sekä käydään läpi yleisimmät termit ja käsitteet. Kun teoria on hallussa, käydään läpi pitkä ja aikaa vievä OpenStackin asennusprosessi. Lähteenä käytetään OpenStackin virallista asennusohjetta (Openstack.2019.). Kun asennus on suoritettu ja määrytykset määritetty, käynnistetään ensimmäinen instanssi. Ja kun se ei onnistu, käydään hieman läpi missä vika ja miten vika poistetaan. Lopuksi vielä pohditaan, oliko tässä projektissa mitään järkeä. Pilvipalvelualustan luominen kotiloissa ei palvele kaupallisia tarkoituksia, mutta tarjoaa pohjan sille, että tekijä saavuttaa perinpohjaisemman ymmärryksen, jota voi myöhemmin jatkohyödyntää myös kaupallisissa tarkoituksissa. Ideana on myös todistaa, että pilvipalveluympäristön luominen on mahdollista ilman mittavaa kone-saliympäristöä.

2 Mitä ovat pilvipalvelut?

Internet kuvataan hyvin usein pilvimäisenä pallerona erinäisissä tietoverkko kaavioissa. On siis luontevaa, että tässä mystisessä pallerossa tapahtuva laskentaa on alettu kutsua ”pilveksi” tai pilvilaskennaksi. Pilvipalveluilla tarkoitetaan internetissä tapahtuvaa laskentaa eli käytännössä verkon yli käytettäviä palveluita ja ohjelmistoja. Käyttäjälle pilvipalvelut tarjoavat moninaisia etuja. Pilvipalveluissa raskain laskenta siirretään omalta koneelta pilven tehtäväksi. Tämän ansiosta käyttäjän laitteiston ei tarvitse olla uusinta ja tehokkainta mahdollistaakseen raskaidenkin ohjelmistojen käytön. Toisaalta huomioitavaa tässä on, että käyttäjän ja pilvipalvelualustan välisen verkkoyhteyden tulee olla erittäin luotettava. Pilvipalvelussa myös kaikki data tallentuu pilveen, jolloin käyttäjä voi vaikka lennossa vaihtaa laitteistoa, ja päästä silti käsiksi tietoihinsa.

Pilvipalvelut voidaan jakaa kolmeen palvelutasoon.

- 1) **IaaS** (*Infrastructures as service*), jolla tarkoitetaan ympäristöä, jossa käyttäjälle tarjotaan palvelinsalin toiminnot ulkoistetulta alustalta.
- 2) **SaaS** (*Software as service*), joka on tilanne, jossa yksittäinen tai useampi ohjelmisto suorittaa laskentansa pilvessä.
- 3) **PaaS** (*Platform as service*), jolla tarkoitetaan ympäristöä, jossa käyttäjälle tarjotaan käyttöjärjestelmä infrastruktuurin päällä palveluna. Esimerkiksi ohjelmistokehityksessä tämä mahdollistaa kehittäjille samat resurssit saataville ja käytettäviksi etänä. (Airinen. 2010. s. 9-15)

2.1 AWS, Microsoft Azure ja Google Cloud

Kaupallisella puolella suurimmat pilvipalvelujen tarjoajat ovat AWS (Amazon Web Services), Microsoft Azure ja Google Cloud. AWS:ää tai Google Cloudia ei ole mahdollista asentaa omaan käyttöön, alusta pyörii aina omistajan eli Amazonin tai Googlen konesalissa (Google Cloud. 2020., AWS Documentation. 2020.). Näitä alustoja ei voi itse hallinnoida mitenkään, pääsy on ainoastaan virtuaali-infrastruktuuriin. Koska palvelimet saattavat olla missä vain, niissä noudatetaan sen maan lainsäädäntöä, jossa palvelimet sijaitsevat fyysisesti. Tämä saattaa aiheuttaa ristiriitoja esimerkiksi tietosuoja- ja tietoliikennelätkien kannalta kuluttajalle. Käyttäjällä ei myöskään ole varmuutta kenellä tosiasiasa on fyysinen pääsy konesaliin, jolloin tietoturvan menestyksekkäs ylläpito on täysin omistavan yrityksen vastuulla, eikä käyttäjä voi vaikuttaa siihen omilla toimillaan niiltä osin. Azuren alusta, toisin kuin Google Cloud tai AWS, voidaan asentaa käyttäjän omaan konesaliin. Tällöin käyttäjä ottaa kokonaisvaltaisen vastuun omasta tietoturvastaan. Azuren alusta on suljettua koodia, mutta hallintarajapinta avointa koodia (Microsoft Azure. 2020.). Kaikki tarjoavat samat pilvipalvelut, erot tulevat huomattaviksi lähinnä hinnassa ja skaalautuvuudessa. AWS:llä on kaupallisena etunaan laajin palvelukatalogi, joka sisältää eri kokoisia valmiiksi räätälöityjä ratkaisuja (AWS Documentation. 2020.).

2.2 OpenStack

OpenStack on pilvipalvelualusta, jonka eri komponenteista voidaan kasata tarpeisiin sopiva ratkaisu (OpenStack Software. 2020.). OpenStack skaalautuu helposti ja myös nodejen poisto ja lisäys tapahtuu helposti. Tässä opinnäytetyössä tehdään laskentaympäristö, johon kuuluu yksi hallintakone ja kolme laskentakonetta. Tämä muodostaa alustan mahdolliselle laaS-ympäristölle.

3 Laitteisto

Laitteistona työssä käytössä on yksi itserakennettu tietokone sekä kolme kappaletta Raspberry Pi 2 Model B- yhdenpiirin tietokonetta.

Itserakennetun koneen tekniset tiedot:

prosessori: Intel Core i3-2100T, 2,5GHz kaksisyttiminen

muisti (RAM): 8Gt, 1333MHz

tallennusmediat: 1 kappale 128Gt OCZ Vertex3 SSD

3 kappaletta 2Tt Wester Digital HDD Red, joista 2kpl RAID1 tilassa,
eli niin sanotussa peilaavassa RAIDissa

Raspberry Pi 2 Model B:n tekniset tiedot:

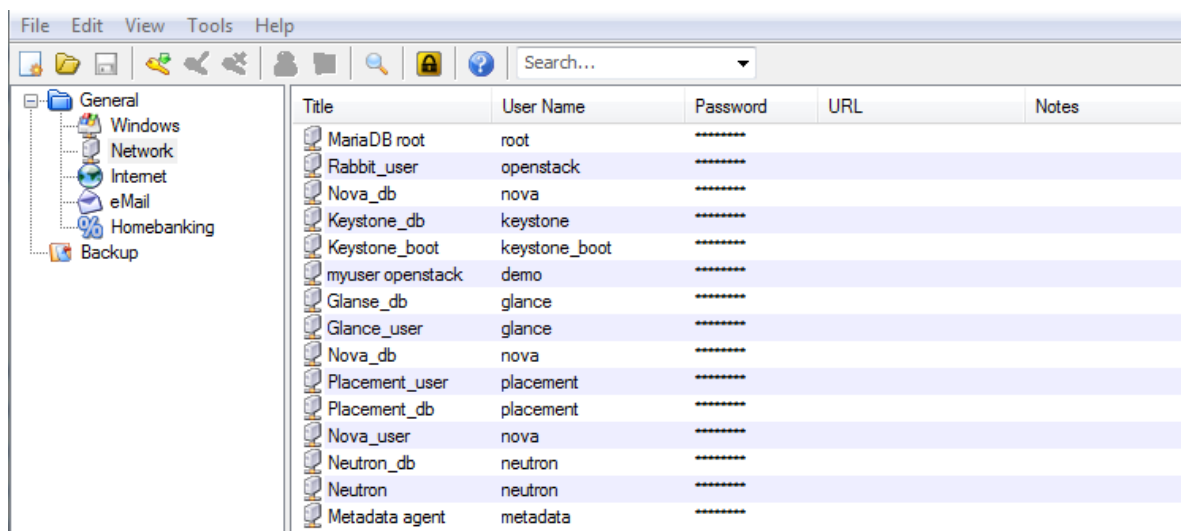
prosessori: ARM Cortex-A7, 900MHz neliydin

muisti (RAM): 1Gt

tallennusmedia: 32Gt mircoSD muistikortti

Itserakennettuun koneeseen viittaa tässä työssä jatkossa nimellä ”*kenkalaatikko*” ja Raspberry Pi:t ovat ”*mustikka*”, ”*vadelma*” sekä ”*herukka*”, kollektiivisesti ”*raspit*”. Lisäksi näihin koneisiin on liitettävissä näppäimistö, hiiri sekä televisio, joka toimittaa näytön virkaa sitä tarvittavissa vaiheissa. Suurin osa työstä kuitenkin tapahtuu SSH-yhteyden ylitse. Edellä mainittujen koneiden lisäksi käytössäni on viides kone, joka toimii Windows 7 käyttöjärjestelmällä ja jota käytän esivaiheiden levykuvien käsittelyyn.

Selkeyden ja muistin helpottamiseksi käytän myös työssä apunani salasana pankkia. Tähän tehtävään valitsin KeePass-nimisen ohjelmiston (*Kuva 1*).



Kuva 1 Työssä käytetyt salasanat KeePass-ohjelmassa

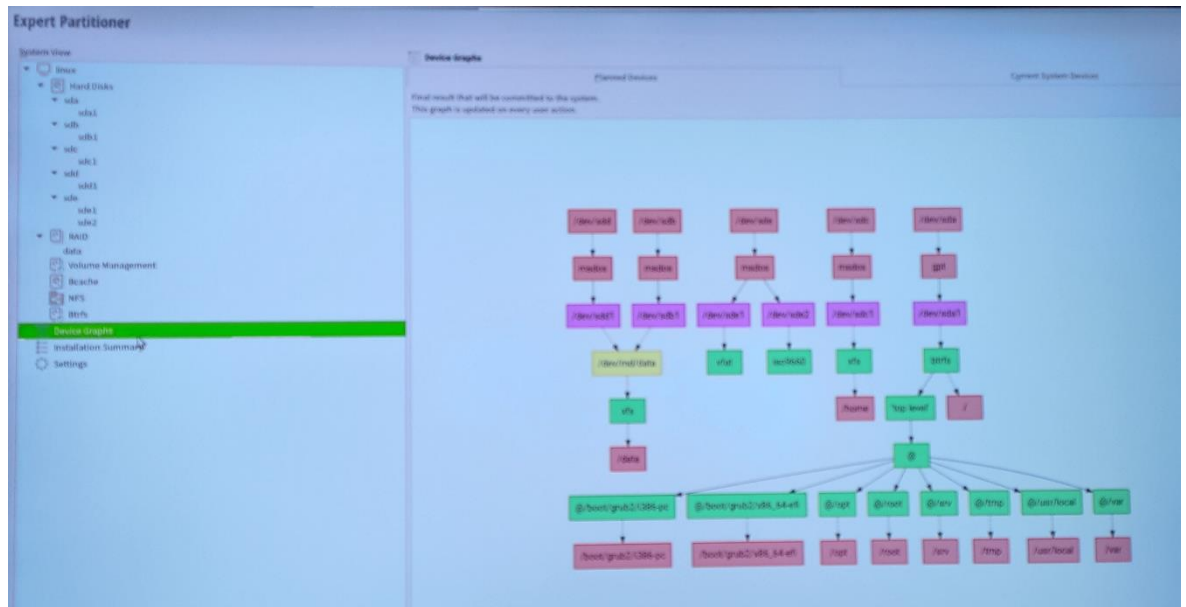
4 Asennuksen valmistelu

Tässä osiossa käydään läpi eri asennusvaiheet omiksi osioiksi lohkottuina.

Ensimmäisenä vaiheena on käyttöjärjestelmän asennus *kenkalaatikkoon*. Käyttöjärjestelmäksi on valittu openSUSE Leap versio 15.1, koska se on OpenStackin suositeltujen ja testattujen käyttöjärjestelmien listalla ja versio 15.1 koska se on työn teko hetkellä vakaa julkaisuversio. Lisäksi koen osaamiseni openSUSEsta olevan työn onnistumisen kannalta tarvittavalla tasolla. Raspeille valitsin Raspbian Buster käyttöjärjestelmän, koska se on raspeille suunniteltu sekä erittäin laajasti tuettu ja lisäksi uusin versio, tekohekellä.

4.1 Käyttöjärjestelmän asennus

OpenSUSEn asennus alkaa lataamalla uusin levykuva openSUSE -jakelun virallisilta sivuilta (<https://www.opensuse.org/>). Valitsin 64 -bittisen DVD-kuvan, joka sisältää kaikki tarpeelliset paketit itsenäiseen asennukseen. Seuraavaksi puretaan levykuva USB-tikulle samaisella sivustolla olevien ohjeiden mukaisesti. Suuren levykuva koon takia USB-tikun tulee olla yli 4Gt, käytössäni on 8Gt kokoinen tikku. Levykuvan tallennus USB-tikulle onnistuu *Etcher*-nimisellä ohjelmalla (<https://www.balena.io/etcher/>). Etcher luo levykuvasta käynnistävän USB-tikun, jolla käyttöjärjestelmän asennus voidaan aloittaa koneen käynnistyksen yhteydessä. Kun tikku on valmis, poistetaan se turvallisesti, liitetään *kenkalaatikkoon* ja käynnistetään *kenkalaatikko*. Valitaan BIOSin käynnistysvalikosta kyseinen tikku käynnistysmediaksi ja koneen ladattua tarvittu paketit valitaan "Asennus" kohta. Itse käyttöjärjestelmän asennus tapahtuu ohjeiden ja oletusasetusten mukaisesti, kunnes tullaan tallennusmedioiden osiointi kohtaan. Koska *kenkalaatikkossa* on 3 kappaletta 2:n teratavun kiintolevyä, tehdään kahdesta niistä RAID1 media. Näin ollen kaikki tähän kokonaisuuteen tallennettu data kopioidaan molemmille levyille ja se näkyy käyttöjärjestelmälle yhtenä 2 teran levynä. Lisäksi poistetaan swap-osio kokonaan (*Kuva 2*). Muuten asennus jatkuu loppuun oletusarvoilla.



Kuva 2 Osiointikartta

Raspbian asennus alkaa lataamalla Raspbian käyttöjärjestelmä Raspberry Pi säätöön sivuilta (<https://www.raspberrypi.org/downloads/raspbian-pi-os/>). Valitaan 32-bittinen versio, koska muuta ei ole saatavilla. Ladataan levykuva, joka sisältää käyttöjärjestelmän työpöytäominaisuudella. Kun levykuva on ladattu, voidaan se purkaa suoraan muistikortille käyttäen jälleen *Etcher*:ä. Kun levykuvan siirto on onnistunut, voidaan muistikortti liittää suoraan Raspberry-koneeseen ja käynnistää kone liittämällä virtapiuha. Käynnistyttyään on käyttöjärjestelmä käyttövalmis.

4.2 Esivalmistelut, *kenkalaatiiko*

Ennen kuin varsinaisen asennustyön voi aloittaa pitää vasta-asennetulle openSUSElle tehdä muutama muutos. Ensimmäiseksi vaihdetaan verkkoasetuksia hallitseva ohjelman Network Managerista YaSTin Wicked moduuliin. Koen YaSTin käytön helpommaksi ja muutosten teko on mahdollista komentoriviltä eikä vain graafisen käyttöliittymän kautta. Seuraavaksi asetetaan koneelle kiinteä IP editoimalla tiedostoa

`"/etc/sysconfig/network/ifcfg-eth0"`

```
GNU nano 2.9.6 /etc/sysconfig/network/ifcfg-eth0 Modified
BOOTPROTO='static'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR='192.168.10.110/24'
MTU=''
NAME=''
NETMASK=''
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='auto'
DHCLIENT_SET_DEFAULT_ROUTE='yes'

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell
```

Lisätään

myöhemmin asennettavien Rasprien nimet ja IP:t `/etc/hosts`-tiedostoon

```
GNU nano 2.9.6 /etc/hosts Modified
ff02::2      ipv6-allrouters
ff02::3      ipv6-allhosts
192.168.10.110 kenkalaatikko

192.168.10.120 mustikka
192.168.10.130 herukka
192.168.10.140 vadelma
```

Seuraavaksi sallitaan SSH-yhteydet sekä määriteetään SSH-palvelu käynnistymään aina kun kone käynnistyy.

```
ont@kenkalaatikko:~> sudo systemctl enable sshd.service
Created symlink /etc/systemd/system/multi-user.target.wants/sshd.service
→ /usr/lib/systemd/system/sshd.service.
ont@kenkalaatikko:~> █
```

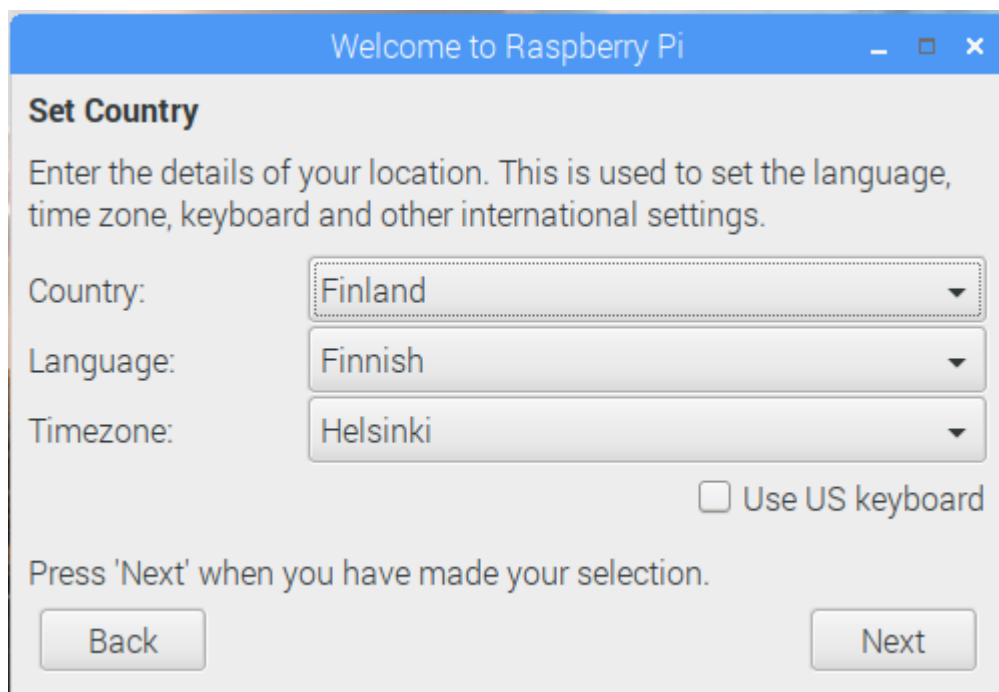
Nyt koneeseen on mahdollista ottaa terminaali etäyhteys ja asennustyö voi alkaa.

4.3 Esivalmistelut, *raspi*

Rasbianin käynnistyessä ensimmäistä kertaa, tervehtii se käyttäjää pienellä asetusikkunalla

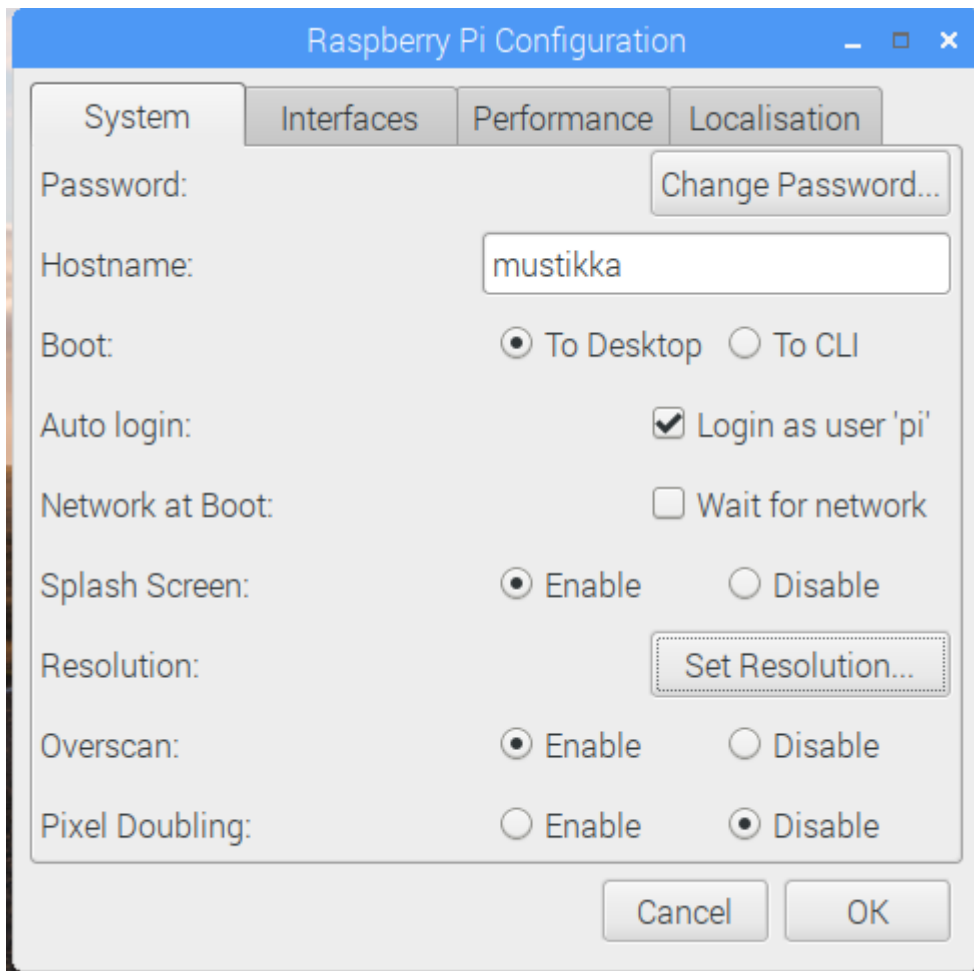


Valitaan "Next" ja siirrytään näkymään, jossa voidaan valita käytettävät lokalisointi asetukset, käyttökieli sekä aikavyöhyke



Seuraavassa näkymässä voidaan asettaa käytössä olevan *raspin* isäntänimi, sekä lisäksi päättää käynnistykö kone työpöytä näkymään vai pelkkään komentorivi tilaan. Valitaan

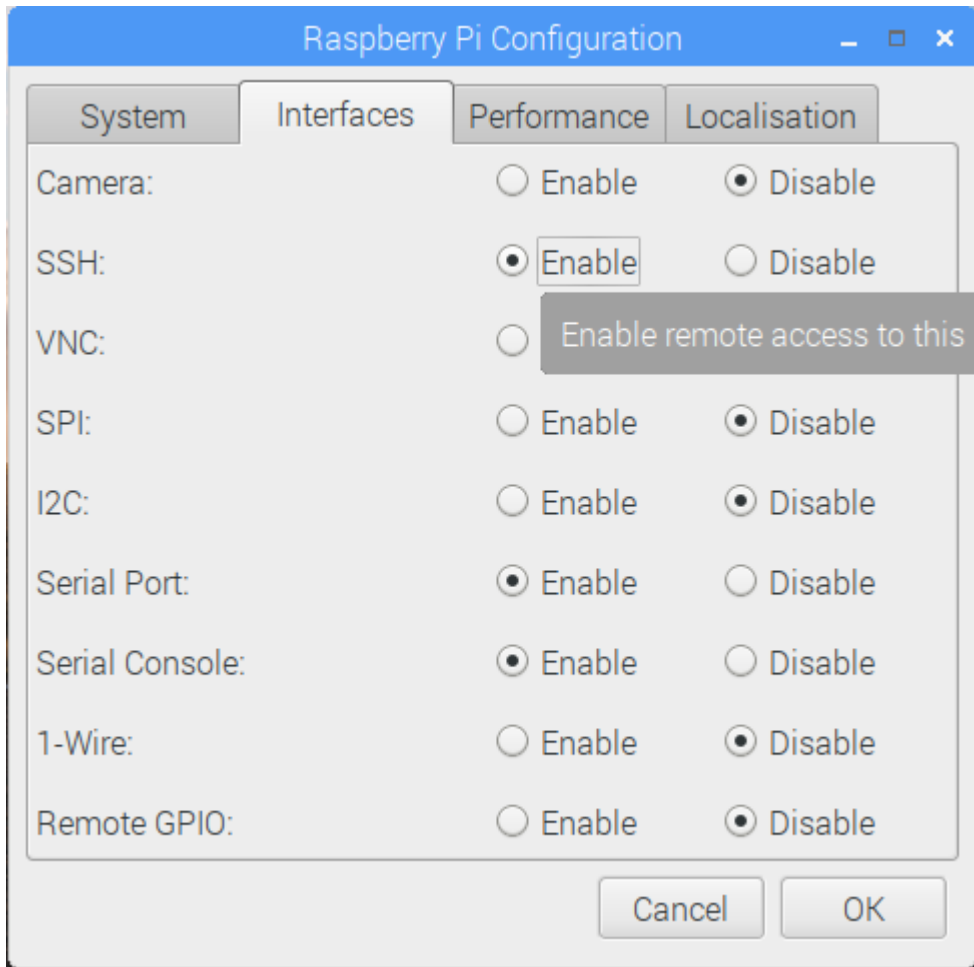
komentorivi muistin säätämiseksi.



Lisäksi tästä näkymästä päästään vaihtamaan salasana. Valitaan vahva.



Lopuksi vielä "Interface"-välilehdeltä sallitaan SSH



Tämän jälkeen määritetään kiinteä IP-osoite sekä lisätään muut koneet `/etc/hosts` tiedostoon

```
GNU nano 2.7.4 Tiedosto: /etc/hosts
127.0.0.1    localhost
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

127.0.1.1   mustikka
192.168.10.120 mustikka

192.168.10.110 kenkalaatikko
192.168.10.130 herukka
192.168.10.140 vadelma
```

Kun nämä vaiheet on suoritettu kaikille *raspeille*, voidaan siirtyä seuraavaan vaiheeseen.

4.4 Network Time Protocol (NTP) -asennus

Jotta eri palvelimilla ajettavat palvelut pysyvät samassa ajassa tulee niiden synkronoida kellonsa ajoittain. Tähän tarkoitukseen on NTP-protokolla ja palvelu nimeltään Chrony.

Chrony asennetaan kaikkiin koneisiin ja *kenkalaatikko*sta tehdään kontrolleri muille. Asennus hoituu komennolla

```
ont@kenkalaatikko:~> sudo zypper in chrony
Loading repository data...
Reading installed packages...
'chrony' is already installed.
No update candidate for 'chrony-3.2-1p151.8.6.x86_64'. The highest available version is already installed.
Resolving package dependencies...

Nothing to do.
ont@kenkalaatikko:~> █
```

Asennusohjelma kertoo, että Chronystä on jo asennettu uusin versio. Seuraavaksi muokataan Chronyn asetus tiedostoa *"/etc/chrony.conf"*

```
GNU nano 2.9.6 /etc/chrony.conf Modified
#log measurements statistics tracking

# Also include any directives found in configuration files in /etc/chrony.d
include /etc/chrony.d/*.conf

server kenkalaatikko iburst
allow 192.168.10.0/24█

^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit         ^R Read File   ^N Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line
```

Tiedostossa kohta "server" kertoo Chronylle kuunnella palvelinta, seuraavana palvelimen nimi ja viimeisenä asetus "iburst" joka, virallisen ohjeen mukaan, nopeuttaa synkronointi toimintoa. Koska *kenkalaatikko* kuuntelee itseään, on se Chronyn kontrolleri palvelin. Seuraava rivi kertoo, että sallitaan yhteydenotot samasta aliverkosta, jossa palvelin on. Lopuksi tallennetaan tiedosto ja käynnistetään Chrony palvelu.

```
ont@kenkalaatikko:~> sudo systemctl enable chronyd.service
ont@kenkalaatikko:~> sudo systemctl start chronyd.service
ont@kenkalaatikko:~> █
```

Nyt Chrony on toiminnassa ja odottaa muiden laitteiden liittymistä. Siirrytään siis *raspeihin*. Asennetaan Chrony komennolla

```
pi@mustikka:~ $ sudo apt install chrony
Luetaan pakettiluetteloita... Valmis
Muodostetaan riippuvuussuhteiden puu
Luetaan tilatiedot... Valmis
The following additional packages will be installed:
  libtomcrypt0 libtommath1
Seuraavat UUDET paketit asennetaan:
  chrony libtomcrypt0 libtommath1
0 päivitetty, 3 uutta asennusta, 0 poistettavaa ja 0 päivittömätöntä.
Noudettavaa arkistoa 541 kt.
Toiminnon jälkeen käytetään 10215 k t lisää levytilaa.
Haluatko jatkaa? [K/e]
```

Asennuksen jälkeen editoidaan tiedostoa `"/etc/chrony/chrony.conf"` ja lisätään tieto palvelimesta

```
GNU nano 2.7.4 Tiedosto: /etc/chrony/chrony.conf
# Welcome to the chrony configuration file. See chrony.conf(5) for more
# information about usable directives.
#pool 2.debian.pool.ntp.org iburst
server kenkalaatikko iburst
```

Käynnistetään palvelu uudestaan komennolla

```
pi@mustikka:~ $ sudo service chrony restart
pi@mustikka:~ $
```

Palataan hetkeksi *kenkalaatikkoon* ja ajetaan komennot

```
ont@kenkalaatikko:~> sudo chronyc sources
210 Number of sources = 1
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* kenkalaatikko            10 10 377 338m -34ns[+2441ns] +/- 4108ns
ont@kenkalaatikko:~> sudo chronyc tracking
Reference ID      : C0A80A6E (kenkalaatikko)
Stratum          : 11
Ref time (UTC)   : Fri Apr 03 16:27:52 2020
System time      : 0.000000001 seconds slow of NTP time
Last offset      : +0.000002475 seconds
RMS offset       : 0.000002475 seconds
Frequency        : 42.556 ppm slow
Residual freq    : -0.030 ppm
Skew             : 0.049 ppm
Root delay       : 0.000007834 seconds
Root dispersion  : 0.021938819 seconds
Update interval  : 0.0 seconds
Leap status      : Normal
ont@kenkalaatikko:~>
```

Ensimmäisestä tulosteesta voidaan todeta, että *kenkalaatikko* kuuntelee itseään Chronyn ajan säätäjänä. Jälkimmäinen komento kertoo tarkemmin, kuinka paljon edessä tai jäljessä *kenkalaatikko* on kellostaan. Siirrytään takaisin *raspeihin* ja ajetaan samat komennot

niissä

```
pi@mustikka:~ $ sudo chronyc sources
210 Number of sources = 1
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* kenkalaatikko           11  6   77   37   -19us[ -105us] +/-  22ms
pi@mustikka:~ $ sudo chronyc tracking
Reference ID      : C0A80A6E (kenkalaatikko)
Stratum          : 12
Ref time (UTC)   : Fri Apr 03 22:06:43 2020
System time      : 0.000022895 seconds slow of NTP time
Last offset      : -0.000086051 seconds
RMS offset       : 0.000086051 seconds
Frequency        : 2.063 ppm fast
Residual freq    : -0.015 ppm
Skew             : 3.054 ppm
Root delay       : 0.000549 seconds
Root dispersion  : 0.022101 seconds
Update interval  : 65.1 seconds
Leap status      : Normal
pi@mustikka:~ $ █
```

Tällä kertaa tulosteet kertovat kuinka paljon kyseinen *rarpi*, tässä tapauksessa *mustikka*, poikkeaa *kenkalaatikkon* kellosta. Näin ollen voidaan todeta Chrony palvelun olevan toiminnassa.

4.5 OpenStackin pakettien haku ja asennus

Seuraavat vaiheet suoritetaan *kenkalaatikkossa*. *Raspian* pariin palataan vaiheessa 5.4.3. Ensin lisätään OpenStackin pakettivarasto komennolla

```
ont@kenkalaatikko:~> sudo zypper addrepo -f obs://Cloud:OpenStack:Stein/openSUSE
_Leap_15.0 Stein
[sudo] password for root:
Adding repository 'Stein' .....[done]
Repository 'Stein' successfully added

URI          : https://download.opensuse.org/repositories/Cloud:/OpenStack:/Stein
/openSUSE_L
Enabled      : Yes

GPG Check    : Yes

Autorefresh  : Yes

Priority     : 99 (default priority)

Repository priorities are without effect. All enabled repositories share the same priority.
ont@kenkalaatikko:~> █
```

”*addrepo*” komennon perässä oleva -f lippu lisää pakettivaraston päivitettävien varastojen listaan, kun ajetaan komento ”*sudo zypper ref*”.

Seuraavaksi päivitetään pakettivarastot sekä asennetaan uusimmat paketit kaikkiin jo asennettuihin ohjelmistoihin komennolla

```
ont@kenkalaatikko:~> sudo zypper ref && sudo zypper dist-upgrade
Retrieving repository 'Stein' metadata -----[\]

New repository or package signing key received:

  Repository:      Stein

  Key Name:        Cloud:OpenStack OBS Project <Cloud:OpenStack@build.opensuse.
org>
  Key Fingerprint: 35B34E18 ABC1076D 66D5A86B 893A90DA D85F9316

  Key Created:     Wed 29 Aug 2018 21:29:12 EEST

  Key Expires:     Fri 06 Nov 2020 20:29:12 EET

  Rpm Name:        gpg-pubkey-d85f9316-5b86e5f8

Do you want to reject the key, trust temporarily, or trust always? [r/t/a/?] (r)
: a
```

Vastataan, että luotamme aina (a) Stein pakettivaraston allekirjoitus avaimeen.

```
Do you want to reject the key, trust temporarily, or trust always? [r/t/a/?] (r)
: a
Retrieving repository 'Stein' metadata .....[done]
Building repository 'Stein' cache .....[done]
Repository 'Non-OSS Repository' is up to date.
Repository 'Main Repository' is up to date.
Repository 'Main Update Repository' is up to date.
Repository 'Update Repository (Non-Oss)' is up to date.
All repositories have been refreshed.
Loading repository data...
Reading installed packages...
Warning: You are about to do a distribution upgrade with all enabled repositories. Make sure these repositories are compatible before you continue. See 'man zypper' for more information about this command.
Computing distribution upgrade...

The following NEW package is going to be installed:
  openSUSE-release-ftp

1 new package to install.
Overall download size: 7.4 KiB. Already cached: 0 B. After the operation,
additional 66.0 B will be used.
Continue? [y/n/v/...? shows all options] (y):
```

Asennus kertoo, että yksi uusi paketti on asennettavissa. Jatketaan painamalla rivivaihtoa tai antamalla ensin y-kirjain ja sitten rivivaihto.

```

Continue? [y/n/v/...? shows all options] (y):
Retrieving package openSUSE-release-ftp-15.1-lp151.298.1.x86_64
(1/1), 7.4 KiB (66 B unpacked)
Retrieving: openSUSE-release-ftp-15.1-lp151.298.1.x86_64.rpm .....[done]

Checking for file conflicts: .....[done]
(1/1) Installing: openSUSE-release-ftp-15.1-lp151.298.1.x86_64 .....[done]
ont@kenkalaatikko:~> █

```

Asennus on suoritettu ja kaikki päivitykset tehty, nyt voidaan asentaa OpenStackin asiakasohjelma komennolla

```

ont@kenkalaatikko:~> sudo zypper in python-openstackclient
[sudo] password for root:
Loading repository data...
Reading installed packages...
'python-openstackclient' not found in package names. Trying capabilities.
Resolving package dependencies...

```

Koska "openstackclient" on metapaketti, kertoo asennus seuraavaksi asentavansa 84 uutta pakettia sekä asennuksen tarvitsevan asentaa 5 suositeltua pakettia.

```

The following 84 NEW packages are going to be installed:
libpython2_7-1_0 python python2-appdirs python2-asn1crypto python2-Babel
python2-blinker python2-certifi python2-cffi python2-chardet
python2-cinderclient python2-cliff python2-cmd2 python2-cryptography
python2-cssselect python2-debtcollector python2-decorator
python2-dogpile.cache python2-ecdsa python2-futures python2-glanceclient
python2-heatclient python2-idna python2-iso8601 python2-jmespath
python2-jsonpatch python2-jsonpointer python2-jsonschema python2-keystoneauth1
python2-keystoneclient python2-lxml python2-monotonic python2-msgpack
python2-munch python2-ndg-httpsclient python2-netaddr python2-netifaces
python2-novaclient python2-oauthlib python2-openstackclient
python2-openstacksdk python2-osc-lib python2-oslo.config python2-oslo.i18n
python2-oslo.serialization python2-oslo.utils python2-os-service-types
python2-packaging python2-pbr python2-ply python2-PrettyTable python2-py
python2-pyasnl python2-pycparser python2-PyJWT python2-pykerberos
python2-pyOpenSSL python2-pyparsing python2-pyperclip python2-PySocks
python2-pytz python2-PyYAML python2-requests python2-requestsexceptions
python2-requests-kerberos python2-rfc3986 python2-setuptools
python2-simplejson python2-six python2-stevedore python2-subprocess32
python2-swiftclient python2-unicodectsv python2-urllib3 python2-warlock
python2-wcwidth python2-wrapt python-base python-contextlib2 python-enum34
python-funcsigs python-functools32 python-ipaddress python-xml xsel

The following 5 recommended packages were automatically selected:
python2-cryptography python2-idna python2-ndg-httpsclient python2-pyOpenSSL
python2-PySocks

84 new packages to install.
Overall download size: 24.1 MiB. Already cached: 0 B. After the operation,
additional 129.9 MiB will be used.
Continue? [y/n/v/...? shows all options] (y): █

```

Jatketaan painamalla rivinvaihtoa.

Koska myöhäisemmässä vaiheessa huomasin tehneeni virheen palasin tähän vaiheeseen ja lisäsin alkuperäisestä ohjeesta puuttuvan Train OpenStack pakettivaraston komennolla

```
ont@kenkalaatikko:~> sudo zypper addrepo -f obs://Cloud:OpenStack:Train/openSUSE_Leap_15.1 Train
Adding repository 'Train' ..... [done]
Repository 'Train' successfully added

URI      : https://download.opensuse.org/repositories/Cloud:/OpenStack:/Train/openSUSE_Leap_15.1
Enabled  : Yes
GPG Check : Yes
Autorefresh : Yes
Priority  : 99 (default priority)

Repository priorities are without effect. All enabled repositories share the same priority.
ont@kenkalaatikko:~> █
```

Tämän jälkeen vielä päivitin ja asensin uusimmat paketit komennolla

```
ont@kenkalaatikko:~> sudo zypper ref && sudo zypper dist-upgrade
Repository 'Stein' is up to date.
Retrieving repository 'Train' metadata ..... [done]
Building repository 'Train' cache ..... [done]
Repository 'Non-OSS Repository' is up to date.
Repository 'Main Repository' is up to date.
Retrieving repository 'Main Update Repository' metadata ..... [done]
Building repository 'Main Update Repository' cache ..... [done]
Repository 'Update Repository (Non-Oss)' is up to date.
All repositories have been refreshed.
Loading repository data...
Reading installed packages...
Warning: You are about to do a distribution upgrade with all enabled repositories. Make sure these repositories are compatible before
you continue. See 'man zypper' for more information about this command.
Computing distribution upgrade...

The following 35 NEW packages are going to be installed:
git-core libev4 libpcra2-8-0 libshaldetectcoll1 perl-Error python2-alabaster python2-backports python2-backports.ssl_match_hostname
python2-brotlipy python2-dnspython python2-docutils python2-dulwich python2-fastimport python2-gevent python2-geventhttpclient
python2-gpg python2-greenlet python2-imagesize python2-Jinja2 python2-MarkupSafe python2-olefile python2-Pillow python2-psutil
python2-pycryptodome python2-Pygments python2-reno python2-snowballstemmer python2-Sphinx python2-sphinxcontrib
python2-sphinxcontrib-websupport python2-sphinx_rtd_theme python2-SQLAlchemy python2-typing python2-Whoosh python2-zope.event

The following 33 packages are going to be upgraded:
libcares2 python2-cinderclient python2-cliff python2-cmd2 python2-debtcollector python2-decorator python2-glanceclient
python2-heatclient python2-keystoneauth1 python2-keystoneclient python2-monotonic python2-msgpack python2-novaclient
python2-openstackclient python2-openstacksdk python2-osc-lib python2-oslo.config python2-oslo.i18n python2-oslo.serialization
python2-oslo.utils python2-os-service-types python2-pbr python2-py python2-pycparser python2-PyMySQL python2-pyparsing
python2-PyYAML python2-requests python2-rfc3986 python2-six python2-stevedore python2-swiftclient python2-urllib3

33 packages to upgrade, 35 new.
Overall download size: 34.5 MiB. Already cached: 0 B. After the operation, additional 131.1 MiB will be used.
Continue? [y/n/v/...? shows all options] (y): █
```

Koska Train on uudempi versio kuin Stein, päivittää paketti hallinta automaattisesti kaikki OpenStackin paketit. Selkeyden vuoksi Stein paketti varaston voi poistaa komennolla

```
ont@kenkalaatikko:~> sudo zypper removerepo Stein
[sudo] password for root:
Removing repository 'Stein' ..... [done]
Repository 'Stein' has been removed.
ont@kenkalaatikko:~> █
```

4.6 SQL tietokanta

Koska suurin osa OpenStackin palveluista käyttävät SQL tietokantoja tiedon tallentamiseen asennetaan seuraavaksi SQL tietokanta palvelu kontrollerille.

Aloitetaan asentamalla tarvittavat paketit komennolla

```
ont@kenkalaatikko:~> sudo zypper in mariadb-client mariadb python-PyMySQL
[sudo] password for root:
Loading repository data...
Reading installed packages...
'python-PyMySQL' not found in package names. Trying capabilities.
'mariadb' is already installed.
No update candidate for 'mariadb-10.2.29-lp151.2.9.1.x86_64'. The highest available version is already installed.
'mariadb-client' is already installed.
No update candidate for 'mariadb-client-10.2.29-lp151.2.9.1.x86_64'. The highest available version is already installed.
Resolving package dependencies...

The following NEW package is going to be installed:
 python2-PyMySQL

1 new package to install.
Overall download size: 97.2 KiB. Already cached: 0 B. After the operation, additional 382.2 KiB will be used.
Continue? [y/n/v/...? shows all options] (y):
```

Asennusohjelma ilmoittaa, että ”mariadb-client” ja ”mariadb” paketit ovat jo asennettu sekä päivitettyinä uusimpiin versioihinsa. Asennusohjelma siis asentaa vain ”python-PyMySQL” paketin. Hyväksytään asennus painamalla rivinvaihtoa.

Seuraavaksi luodaan ja muokataan tiedostoa ”/etc/my.cnf.d/openstack.cnf”. Luodaan osio ”[mysqld]” sekä asetetaan palvelu kuuntelemaan kontrollerin IP-osoitetta. Lisäksi määritetään merkistöksi UTF-8.

```
GNU nano 2.9.6 /etc/my.cnf.d/openstack.cnf

[mysqld]
bind-address = 192.168.10.110

default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

Viimeistellään asennus asettamalla tietokantapalvelu käynnistymään järjestelmän uudelleenkäynnistyksen yhteydessä sekä käynnistetään palvelu heti. Tämä tapahtuu komendoilla

```
ont@kenkalaatikko:~> sudo systemctl enable mysql.service
Created symlink /etc/systemd/system/mysql.service â /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service â /usr/lib/systemd/system/mariadb.service.
ont@kenkalaatikko:~> sudo systemctl start mysql.service
```

Tämän jälkeen ajetaan ”mysql_secure_installation” skripti. Tässä vaiheessa valitaan tietokannan root-tunnukselle hyvä ja vahva salasana, sekä muun muassa poistetaan testikanta ja siihen pääsy ja estetään root-tunnusta kirjautumasta etänä.

```

ont@kenkalaatikko:~> sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] 

Remove anonymous users? [Y/n]
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n]
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n]
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n]
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
ont@kenkalaatikko:~> 

```

4.7 Viestijono

OpenStack käyttää viestijonoa operaatioiden koordinoimiseen ja tilannetietojen välittämi- seen eri palveluiden välillä. Tässä projektissa käytetään RabbitMQ nimistä palvelua.

Aloitetaan asentamalla ja määrittelemällä komponentit. Asennetaan paketti komennolla

```
ont@kenkalaatikko:~> sudo zypper in rabbitmq-server
[sudo] password for root:
Retrieving repository 'Main Update Repository' metadata .....[done]
Building repository 'Main Update Repository' cache .....[done]
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following 4 NEW packages are going to be installed:
  erlang erlang-epmd rabbitmq-server socat

4 new packages to install.
Overall download size: 36.2 MiB. Already cached: 0 B. After the operation,
additional 61.1 MiB will be used.
Continue? [y/n/v/...? shows all options] (y):
```

Kun palvelu on asennettu, asetetaan se käynnistymään järjestelmän käynnistyessä sekä käynnistetään se nyt, komennoin

```
ont@kenkalaatikko:~> sudo systemctl enable rabbitmq-server.service
Created symlink /etc/systemd/system/multi-user.target.wants/rabbitmq-server.service → /usr/lib/systemd/system/rabbitmq-server.service.
ont@kenkalaatikko:~> sudo systemctl start rabbitmq-server.service
```

Kun palvelu on päällä, lisätään "openstack"-niminen käyttäjä. Tämä tapahtuu komennolla

```
ont@kenkalaatikko:~> sudo rabbitmqctl add_user openstack [REDACTED]
Adding user "openstack" ...
ont@kenkalaatikko:~> █
```

Viimeisenä komennossa salasana, joka kuvassa tietoturvasyistä sotkettu. Jatkossa salasanaan viitataan "Rabbit user"-salasanana.

Annetaan vielä lopuksi juuri luodulle käyttäjätunnukselle määritys-, kirjoitus- ja lukuoikeudet viestijono palveluun komennolla

```
ont@kenkalaatikko:~> sudo rabbitmqctl set_permissions openstack ".*" ".*" ".*"
[sudo] password for root:
Setting permissions for user "openstack" in vhost "/" ...
ont@kenkalaatikko:~> █
```

4.8 Memcached

Identiteetti-palvelun autentikointimekaniikka palveluille käyttää Memcached:ia valtuuksien säilömiseen.

Aloitetaan asentamalla ja määrittelemällä komponentit. Asennetaan paketit komennolla

```
ont@kenkalaatikko:~> sudo zypper in memcached python-python-memcached
[sudo] password for root:
Loading repository data...
Reading installed packages...
'python-python-memcached' not found in package names. Trying capabilities.
Resolving package dependencies...

The following 2 NEW packages are going to be installed:
  memcached python2-python-memcached

2 new packages to install.
Overall download size: 149.6 KiB. Already cached: 0 B. After the operation, additional
371.4 KiB will be used.
Continue? [y/n/v/...? shows all options] (y): █
```

Seuraavaksi editoidaan tiedostoa `/etc/sysconfig/memcached`

```
GNU nano 2.9.6 /etc/sysconfig/memcached Modified
## Path:      Network/WWW/Memcached
## Description: start parameters for memcached.
## Type:      string
## Default:   "-l 127.0.0.1"
## Config:    memcached
#
# start parameters for memcached.
#
# see man 1 memcached for more
#
MEMCACHED_PARAMS="-l 192.186.10.110"
```

Vaihdetaan kohtaan `"MEMCACHED_PARAMS="-l 127.0.0.1"` kontrollerin IP-osoite, kuten kuvassa näkyy.

Tämän jälkeen asetetaan Memcached palvelu käynnistymään järjestelmän käynnistyksen yhteydessä ja käynnistetään palvelu myös nyt. Tämä tapahtuu komennoin

```
ont@kenkalaatikko:~> sudo systemctl enable memcached.service
Created symlink /etc/systemd/system/multi-user.target.wants/memcached.service â /usr/lib/systemd/system/memcached.service.
ont@kenkalaatikko:~> sudo systemctl start memcached.service
ont@kenkalaatikko:~> 
```

4.9 Etcid

OpenStack palvelut käyttävät Etcid:tä muun muassa määritysten säilömiseen sekä palveluiden tilan seuraamiseen. Tällä hetkellä Etcid3:sta ei ole OpenSuselle jakelukohtaista pakettia, joten asennus tehdään tervapallero asennuksena.

Aloitetaan asennus luomalla `etcd` -niminen käyttäjätunnus komentoilla

```
ont@kenkalaatikko:~> sudo groupadd --system etcd
[sudo] password for root:
ont@kenkalaatikko:~> sudo useradd --home-dir "/var/lib/etcd" --system --shell /bin/false -g etcd etcd
ont@kenkalaatikko:~> 
```

Ensimmäinen komento luo käyttäjäryhmän `"etcd"`. Jälkimmäinen komento luo käyttäjän ja lisää sen `"etcd"`-käyttäjäryhmään. Ensimmäinen kohta komennon jälkeen määrittää käyttäjän kotihakemistoksi `"/var/lib/etcd"`, seuraava kohta määrittää käyttäjän järjestelmäkäyttäjäksi. `--shell` kohdassa määritellään käyttäjälle oletus komentotulkiksi `/bin/false`, näin estetään käyttäjää kirjautumasta ja ajamasta komentoja. Seuraavaksi käyttäjä lisätään ryhmään `"etcd"`, joka luotiin edellisessä komennossa. Viimeinen kohta on käyttäjän käyttäjätunnus.

Asennuksen seuraava vaihe on luoda tarvittavat kansiot ja määrittää niiden oikeudet.

Tämä tapahtuu seuraavin komennoin

```
ont@kenkalaatikko:~> sudo mkdir -p /etc/etcd
ont@kenkalaatikko:~> sudo chown etcd:etcd /etc/etcd/
etcd/      ethers      ethertypes
ont@kenkalaatikko:~> sudo chown etcd:etcd /etc/etcd/
ont@kenkalaatikko:~> sudo mkdir -p /var/lib/etcd
ont@kenkalaatikko:~> sudo chown etcd:etcd /var/lib/etcd/
ont@kenkalaatikko:~> □
```

Ensin luodaan kansio `"/etc/etcd"`. Komennon `-p` lippu kertoo kansionluonti ohjelmalle luoda mahdolliset puuttuvat yläkansiot. Seuraavalla komennolla annetaan juuri luodun kansion omistusoikeus käyttäjälle `"etcd"` sekä ryhmälle `"etcd"`. Seuraavaksi luodaan kansio `"/var/lib/etcd"` ja toistetaan oikeus muutos myös sille.

Seuraavaksi tarkistetaan järjestelmän arkkitehtuuri komennolla

```
ont@kenkalaatikko:~> sudo uname -m
x86_64
```

Järjestelmä on x86-arkkitehtuuria sekä 64 -bittinen.

Ladataan ja asennetaan `etcd:n` tervapallero `x84_64` arkkitehtuurille komentoilla

```
ont@kenkalaatikko:~> ETCD_VER=v3.2.7
ont@kenkalaatikko:~> sudo rm -rf /tmp/etcd && mkdir -p /tmp/etcd
ont@kenkalaatikko:~> curl -L https://github.com/coreos/etcd/releases/download/${ETCD_VER}/etcd-${ETCD_VER}-linux-amd64.tar.gz -o /tmp/etcd-${ETCD_VER}-linux-amd64.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  153    100  153    0     0    455      0  --:--:--  --:--:--  --:--:--   455
100  620    100  620    0     0   1014      0  --:--:--  --:--:--  --:--:--   605k
100 9939k   100 9939k    0     0 3032k      0  0:00:03  0:00:03  --:--:--  5781k
ont@kenkalaatikko:~> sudo tar xzvf /tmp/etcd-${ETCD_VER}-linux-amd64.tar.gz -C /tmp/etcd --strip-components=1
etcd-v3.2.7-linux-amd64/Documentation/
etcd-v3.2.7-linux-amd64/Documentation/tuning.md
etcd-v3.2.7-linux-amd64/Documentation/rfc/
etcd-v3.2.7-linux-amd64/Documentation/rfc/v3api.md
```

Ensin määritellään muuttuja `ETCD_VER`, joka saa arvon `v3.2.7`. Seuraavaksi poistetaan kansio `"/tmp/etcd"` ja luodaan se uudestaan. Näin varmistetaan, että kansio on tyhjä. Ladataan `etcd:n` tervapallero omaan kansioonsa `/tmp/`:n alle. Viimeinen komento listalla purkaa ladatun tervapalleron kansioon `"/tmp/etcd/"`. Lopuksi vielä kopioidaan juuri puretuista paketeista sisältö käyttäjien ohjelmistokansioihin komennoin

```
ont@kenkalaatikko:~> sudo cp /tmp/etcd/etcd /usr/bin/etcd
ont@kenkalaatikko:~> sudo cp /tmp/etcd/etcdctl /usr/bin/etcdctl
ont@kenkalaatikko:~> □
```

Seuraavaksi luodaan ja editoidaan tiedostoa ”*etc/etcd/etcd.conf.yml*”. Tämä onnistuu helpoiten komennolla *sudo nano etc/etcd/etcd.conf.yml*. Editoidaan tiedosto seuraavaksi

```
GNU nano 2.9.6 /etc/etcd/etcd.conf.yml
[ ]ame: kenkalaatikko
data-dir: /var/lib/etcd
initial-cluster-state: 'new'
initial-cluster-token: 'etcd-cluster-01'
initial-cluster: kenkalaatikko=http://192.168.10.110:2380
initial-advertise-peer-urls: http://192.168.10.110:2380
advertise-client-urls: http://192.168.10.110:2379
listen-peer-urls: http://0.0.0.0:2380
listen-client-urls: http://192.168.10.110:2379
```

Eli vaihdetaan nimeksi palvelimen nimi *kenkalaatikko* ja jokaiseksi IP-osoitteeksi *kenkalaatikko*:n IP-osoite 192.168.10.110.

Seuraavaksi luodaan ja editoidaan tiedosto ”*/usr/lib/systemd/system/etcd.service*” näyttämään seuraavalta

```
GNU nano 2.9.6 /usr/lib/systemd/system/etcd.service
[Unit]
After=network.target
Description=etcd - highly-available key value store

[Service]
LimitNOFILE=65536
Restart=on-failure
Type=notify
ExecStart=/usr/bin/etcd --config-file /etc/etcd/etcd.conf.yml
User=etcd

[Install]
WantedBy=multi-user.target
[ ]
```

Uudelleen ladataan systemd:n palvelun tiedostot komennolla

```
ont@kenkalaatikko:~> sudo systemctl daemon-reload
ont@kenkalaatikko:~> [ ]
```

Lopuksi asetetaan etcd -palvelu käynnistymään järjestelmän käynnistyessä sekä käynnistetään palvelu nyt komennoin

```
ont@kenkalaatikko:~> sudo systemctl enable etcd
ont@kenkalaatikko:~> sudo systemctl start etcd
ont@kenkalaatikko:~> [ ]
```

5 OpenStackin asennus

OpenStack -järjestelmä koostuu useasta tärkeästä palvelusta, jotka kaikki asennetaan erikseen. Palvelut toimivat yhdessä, luoden perustan pilvipalvelulle, sisältäen komponentit kuten Laskenta-, Identiteetti-, Verkonhallinta-, Levykuva-, Tallennus-, Telemetria- sekä Tietokantapalvelut. Kaikki palvelut voidaan asentaa ja määrittellä itsenäisiksi tai osaksi keskustelevaa verkostoa.

5.1 Identiteetti palvelu

Tässä osiossa asennetaan ja määrittellään OpenStackin Identiteettipalvelun nimeltään Keystone.

5.1.1 Alkuvalmistelu

Asennetaan Python PyASN.1 sekä luodaan tietokanta, jota Identiteettipalvelu voi käyttää. PyASN.1:n asennus ja päivitys hoituu komennolla

```
ont@kenkalaatikko:~> sudo pip install pyasn1
Collecting pyasn1
  Downloading pyasn1-0.4.8-py2.py3-none-any.whl (77 kB)
    |aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa| 77 kB 1.3 MB/s
Installing collected packages: pyasn1
Successfully installed pyasn1-0.4.8
ont@kenkalaatikko:~> █
```

Pip on Pythonin oma sisäinen pakettihallintasovellus, jolla voidaan asentaa, poistaa ja päivittää Pythonin paketteja.

Kun PyAnsi.1 on asennettu, luodaan tietokanta. Käynnistetään tietokantahallintasovellus komennolla

```
ont@kenkalaatikko:~> mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.2.31-MariaDB SUSE package

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Salasana on se, joka määritettiin kohdassa 4.6 tietokannan root-käyttäjän salasanaksi.

Seuraavaksi luodaan kanta Keystonelle

```
MariaDB [(none)]> CREATE DATABASE keystone;
Query OK, 1 row affected (0.00 sec)
```

Annetaan tarvittavat oikeudet

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY '[REDACTED]';
-> ;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'% ' IDENTIFIED BY '[REDACTED]';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> []
```

Salasanaksi valitaan hyvä ja vahva salasana. Kuvassa salasanat on jälleen tietoturvasyistä sensuroitu sotkemalla. Salasanaan viitataan jatkossa nimellä "Keystone_db". Näiden komentojen jälkeen voidaan poistua tietokantahallintasovelluksesta "exit" komennolla.

```
MariaDB [(none)]> exit
Bye
```

5.1.2 Keystone palvelu

Seuraavaksi asennetaan ja määritellään Keystonen komponentit. Aloitetaan antamalla asennuskomento. (Koska myöhemmässä vaiheessa 5.1.4 tulee ongelmia alla olevalla komennolla, mainittakoon tässä, että asennettava paketti "apache2-mod_wsgi" tulee korvata paketilla "apache2-mod_wsgi-python3".)

```
ont@kenkalaatikko:~> sudo zypper in openstack-keystone apache2 apache2-mod_wsgi
[sudo] password for root:
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following 164 NEW packages are going to be installed:
```

Keystonen lisäksi asennetaan Apache2 HTTP palvelu, sekä siihen wsgi-lisäosa. Asennus kysyy jälleen, halutaanko varmasti asentaa kaikki 164 pakettia, johon vastataan kyllä. Asennuksen jälkeen editoidaan tiedostoa "/etc/keystone/keystone.conf". Kohdassa **[database]** määritellään tietokantayhteys

```
[database]

#
# From oslo.db
#

# If True, SQLite uses synchronous mode. (boolean value)
#sqlite_synchronous = true

# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend = sqlalchemy

# The SQLAlchemy connection string to use to connect to the database. (string
# value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql+pymysql://keystone:[REDACTED]@kenkalaatikko/keystone
[]
```

Salasana on "Keystone_db", joka määriteltiin edellisessä vaiheessa. Se on sensuroitu sotkemalla. Poistetaan tai kommentoidaan kaikki muu **connection** -osiossa. Seuraavaksi määritellään **[token]** osiossa tarjoajaksi Fernet

```
[token]
provider = fernet
[]
```

Tallennetaan muutokset ja poistutaan tekstieditorista.

Lisätään Identiteettipalvelun tietokantaan sen tarvitsemaa dataa komennolla

```
ont@kenkalaatikko:~> sudo su -s /bin/sh -c "keystone-manage db_sync" keystone
ont@kenkalaatikko:~> []
```

Komennon ajossa kestää hetki, riippuen verkon ja järjestelmän nopeudesta.

Ajetaan seuraavat komennot

```
ont@kenkalaatikko:~> sudo keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
ont@kenkalaatikko:~> sudo keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
ont@kenkalaatikko:~> []
```

Komennoilla kerrotaan käyttöjärjestelmän käyttäjätunnuksella ja -ryhmällä "keystone" olevan käyttöoikeudet Keystone -palvelun keskeisimpiin ominaisuuksiin.

Lopuksi vielä bootstrapataan Identiteetti palvelu komennolla

```
ont@kenkalaatikko:~> export OS_AUTH_URL=http://kenkalaatikko:5000/v3
ont@kenkalaatikko:~> sudo keystone-manage bootstrap --bootstrap-password [REDACTED] --bootstrap-admin-url http://kenkalaatikko:5000/v3/ --bootstrap-internal-url http://kenkalaatikko:5000/v3/ --bootstrap-public-url http://kenkalaatikko:5000/v3/ --bootstrap-region-id RegionOne
[sudo] password for root:
ont@kenkalaatikko:~> []
```

Valitaan bootstrap:ia varten vahva salasana, taas sotkettuna kuvassa. Tarvittaessa salasanaan viitataan nimellä "Keystone_boot".

5.1.3 Apache HTTP palvelin

Seuraavaksi määritellään aikaisemmin asennettu Apache -palvelu. Aloitetaan määrittelemällä Apache -palvelimen nimi. Editoidaan tiedostoa `/etc/sysconfig/apache2` ja asetetaan palvelimen nimeksi kontrollerin isäntänimi, eli `kenkalaatikko`.

```
GNU nano 2.9.6 /etc/sysconfig/apache2
APACHE_SERVERNAME="kenkalaatikko"
```

Luodaan ja editoidaan tiedostoa `"/etc/apache2/conf.d/wsgi-keystone.conf"`

```
GNU nano 2.9.6 /etc/apache2/conf.d/wsgi-keystone.conf Modified
Listen 5000

<VirtualHost *:5000>
    WSGIDaemonProcess keystone-public processes=5 threads=1 user=keystone group=keystone
    [display-name=${GROUP}]
    WSGIProcessGroup keystone-public
    WSGIScriptAlias / /usr/bin/keystone-wsgi-public
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On
    ErrorLogFormat "%{cu}t %M"
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/keystone_access.log combined

    <Directory /usr/bin>
        Require all granted
    </Directory>
</VirtualHost>
```

Tallennetaan tiedosto ja poistutaan editorista. Seuraavaksi muutetaan kansion `"/etc/keystone"` ja kaikkien sen alikansioiden omistajaksi käyttäjä ja ryhmä **keystone** komennolla

```
ont@kenkalaatikko:~> sudo chown -R keystone:keystone /etc/keystone
ont@kenkalaatikko:~> █
```

Lopuksi vielä asetetaan Apache käynnistymään aina järjestelmän käynnistyksen yhteydessä sekä käynnistään se nyt. Tämä tapahtuu komennoin

```
ont@kenkalaatikko:~> sudo systemctl enable apache2.service
Created symlink /etc/systemd/system/httpd.service â /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/apache.service â /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service â /usr/lib/systemd/system/apache2.service.
ont@kenkalaatikko:~> sudo systemctl start apache2.service
ont@kenkalaatikko:~> █
```

Lisäksi määritellään hallintatili muuttamalla ympäristömuuttujia. Näin voidaan antaa suoraan komentoja Keystone palvelulle. Salasana on `"Keystone_boot"`:ksi nimetty. Ympäristömuuttujat

```
ont@kenkalaatikko:~> export OS_USERNAME=admin
ont@kenkalaatikko:~> export OS_PASSWORD=
ont@kenkalaatikko:~> export OS_PROJECT_NAME=admin
ont@kenkalaatikko:~> export OS_USER_DOMAIN_NAME=Default
ont@kenkalaatikko:~> export OS_PROJECT_DOMAIN_NAME=Default
ont@kenkalaatikko:~> export OS_AUTH_URL=http://kenkalaatikko:5000/v3
ont@kenkalaatikko:~> export OS_IDENTITY_API_VERSION=3
ont@kenkalaatikko:~> █
```

5.1.4 Toimialueen, projektin, käyttäjien ja roolien luonti

Identiteettipalvelu tarjoaa autentikaatiopalvelun jokaiselle OpenStackin palvelulle. Autentikaatiopalvelu käyttää tähän tarkoitukseen toimialueita, projekteja, käyttäjätunnuksia sekä rooleja.

Luodaan alkuun "service" niminen projekti, johon aikanaan luodaan käyttäjätunnus jokaista ympäristössä olevaa palvelua varten. Tämä tapahtuu komennolla

```
ont@kenkalaatikko:~> openstack project create --domain default --description "Service Project" service
+-----+
| Field      | Value                               |
+-----+
| description | Service Project                     |
| domain_id  | default                             |
| enabled    | True                                |
| id         | d40bb9f1e6224274bd4681c701d3a650  |
| is_domain  | False                               |
| name       | service                             |
| options    | {}                                  |
| parent_id  | default                             |
| tags       | []                                  |
+-----+
ont@kenkalaatikko:~> []
```

Tulosteena komento antaa taulukon, jossa näky muun muassa yksilöivä id -numero.

Luodaan "myproject" niminen projekti ja "myuser" niminen käyttäjätunnus, joita käytetään, kun vaadittavat tehtävät tai muutokset eivät vaadi korotettuja oikeuksia. Projekti luodaan komennolla

```
ont@kenkalaatikko:~> openstack project create --domain default --description "Demo Project" myproject
+-----+
| Field      | Value                               |
+-----+
| description | Demo Project                       |
| domain_id  | default                             |
| enabled    | True                                |
| id         | 86723139bf114a40a10351a314dbb197  |
| is_domain  | False                               |
| name       | myproject                          |
| options    | {}                                  |
| parent_id  | default                             |
| tags       | []                                  |
+-----+
```

Komennon onnistuttua tulosteena taulukko, jossa projektin nimi, kuvaus ja personoiva id-numero.

Luodaan käyttäjätili "myuser" komennolla

```
ont@kenkalaatikko:~> openstack user create --domain default --password-prompt myuser
User Password:
Repeat User Password:
+-----+
| Field              | Value                               |
+-----+
| domain_id         | default                             |
| enabled           | True                                |
| id                | 5a4a1b298b8645d49d5b7d5a327b5be3  |
| name              | myuser                              |
| options           | {}                                  |
| password_expires_at | None                                |
+-----+
```

Asetetaan käyttäjätunnukselle vahva salasana sen luonnin yhteydessä. Salasanaan viitataan jatkossa nimellä "Keystone_myuser". Komennon onnistuneesta suorituksesta tulosteena taulukko, jossa käyttäjätunnuksen tiedoja.

Luodaan rooli "myrole" komennolla

```
ont@kenkalaatikko:~> openstack role create myrole
+-----+
| Field      | Value                                |
+-----+
| description | None                                  |
| domain_id  | None                                  |
| id         | b64268a555c440c6a7f5b97fc8e98f27 |
| name       | myrole                               |
| options    | {}                                    |
+-----+
ont@kenkalaatikko:~> █
```

Komennon onnistuessa tulosteena taulukko, jossa roolin nimi sekä yksilöivä numero. Lopuksi lisätään "myrole"-rooli "myproject"-projektiin sekä lisätään rooli myös "myuser" käyttäjälle. Nämä toimet suoritetaan komennolla

```
ont@kenkalaatikko:~> openstack role add --project myproject --user myuser myrole
ont@kenkalaatikko:~> █
```

Suorituessaan onnistuneesti komento ei anna mitään tulostetta.

5.1.5 Toiminnan varmistus

Ennen muiden palveluiden asentamista on erittäin tärkeää varmistaa, että Identiteetti palvelu on toiminnassa ja toimii oikein. Aloitetaan tyhjentämällä aikaisemmin asetetut ympäristö muuttujat **OS_AUTH_URL** sekä **OS_PASSWORD**.

```
ont@kenkalaatikko:~> unset OS_AUTH_URL OS_PASSWORD
ont@kenkalaatikko:~> █
```

Pyydetään **admin** -käyttäjänä autentikaatio valtuus komennolla

```
ont@kenkalaatikko:~> openstack --os-auth-url http://kenkalaatikko:5000/v3
in-name Default --os-user-domain-name Default --os-project-name admin --os-
oken issue
Password:
Password:
+-----+
| Field      | Value
+-----+
| expires    | 2020-03-09T14:22:22+0000
+-----+
| id         | gAAAAABeZkMOSMOdmJg7ovftI9d7M0QzvNuPduWmWgk4yxnsV6geeZyVLWK
fGLFGJtR7MYbx0c-eyh3srz4LmcoewayC4JoYE_8WaACLAa69WtGkJgcq_XLiN_-puV-g1RDNW
R6PURlibEnvAoEZg
+-----+
| project_id | 4a84a080e6b64d3195c74c0306d7769e
+-----+
| user_id    | c9bdbdd7650f4d2da228706b9d2cf42b
+-----+
ont@kenkalaatikko:~> []
```

Salasana on kohdan 5.1.2 lopussa määritetty salasana, nimeltään "Keystone_boot". Komennon onnistuessa tulosteena taulukko, jossa valtuuden vanhenemisaika, yksilöllinen numero, valtuutta koskevan projektin numero ja valtuuden omistavan käyttäjän numero. Johtuen käytetyn SSH-yhden ikkunan koosta taulukon rajat ovat hieman sekaisin, mutta tärkeät arvot ovat luettavissa.

Pyydetään myös edellisessä vaiheessa luodulle "myuser" -tunnukselle valtuus

```
ont@kenkalaatikko:~> openstack --os-auth-url http://kenkalaatikko:5000/v3 --os-project-domain-name Default --os-user-domain-name Default
--os-project-name myproject --os-username myuser token issue
Password:
Password:
+-----+
| Field      | Value
+-----+
| expires    | 2020-03-09T14:36:29+0000
+-----+
| id         | gAAAAABeZkZdbYZ_1GBMUNAg_gp1w3VY6suAMN-_34SzZbP_alKMcqte8hOXJiuiXy2vzizeIfJDDUzs6yuRbLIRBHDsTJBQjg_-nbT7apeBcN-XXoXG-7aRD
61bQbE11zqBOMSdTSI610VRby54-yKO7DB7etIhRe4kVgeqn3mE121Rdg3j1Q
+-----+
| project_id | 86723139bf114a40a10351a914dbb197
+-----+
| user_id    | 5a4a1b298b8645d49d5b7d5a327b5be3
+-----+
ont@kenkalaatikko:~> []
```

Tulosteena taas taulukko, jossa samat kohdat kuin "admin" -tunnuksella. Näin voidaan todeta, että Identiteettipalvelu toimii.

5.1.6 OpenStack ympäristömuuttaja skriptit

Edellisessä vaiheessa käytettiin sekä ympäristömuuttujia, että komentojen lisäasetuksia Identiteettipalvelun kanssa keskusteluun **openstack** asiakasohjelman kautta. Koska tämä tapa on hieman kankeaa ja hidasta, tukee OpenStack yksinkertaisia ympäristömuuttuja skriptejä. Nämä tunnetaan nimellä OpenRC -tiedostot. Näihin tiedostoihin voidaan tallentaa ympäristömuuttujat eri käyttäjätunnuksille ja projekteille.

Luodaan ympäristömuuttuja skriptit **admin** ja **myuser** -käyttäjille. Nämä skriptit voidaan tallentaa minne vain, mutta on suositeltavaa, että ne tallennetaan turvalliseen ja hallittuun sijaintiin, johon ei ole pääsyä kaikilla, sillä skriptit sisältävät muun muassa käyttäjien salasana selkokielisinä.

Luodaan ja editoidaan tiedostoa "admin-openrc"

```
GNU nano 2.9.6 admin-openrc
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=[REDACTED]
export OS_AUTH_URL=http://kenkalaatikko:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
[]
```

Salasana "Keystone_boot" salasana. Salasana jälleen sotkettu kuvassa.

Luodaan ja editoidaan tiedostoa "myuser-openrc"

```
GNU nano 2.9.6 myuser-openrc
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=myproject
export OS_USERNAME=myuser
export OS_PASSWORD=[REDACTED]
export OS_AUTH_URL=http://kenkalaatikko:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
[]
```

Salasana on "Keystone_myuser" salasana.

Nyt kun ympäristömuuttuja skriptit on luotu, testataan vielä niiden toiminta. Ladataan **admin** -tunnuksen ympäristömuuttujat ajamalla "admin-openrc" skripti

```
ont@kenkalaatikko:~> . admin-openrc
ont@kenkalaatikko:~> []
```

Seuraavaksi pyydetään jälleen autentikaatio valtuus

```
ont@kenkalaatikko:~> openstack token issue
+-----+
| Field      | Value
+-----+
| expires    | 2020-03-09T15:39:30+0000
| id         | gAAAAABeZ1UiOWzOKLkP_hJ17z2zH4t8t28dF182BzvB4QRVXl6SaluOnjdzPBHB_KnbuvYB0CBsA6oyzMxmGfp3B3WBBsh
| project_id | 4a84a080e6b64d3195c74c0306d7769e
| user_id    | c9bdbdd7650f4d2da228706b9d2cf42b
+-----+
ont@kenkalaatikko:~> █
```

Vertaamalla kohtia "project_id" ja "user_id" aikaisemmissa kohdissa luotujen projektien ja käyttäjien tunnistenumeroihin voidaan todeta, että tämä valtuus on annettu käyttäjälle "admin" projektiin "admin". Skripti siis toimii ja voimme siirtyä eteenpäin.

5.2 Levykuvapalvelu

Levykuvapalvelun ansiosta käyttäjät voivat etsiä, tallentaa ja noutaa levykuvia virtuaalikooneille. OpenStackin Levykuva -palvelu hallinnoi pyyntöjä koneiden ja palvelimien levykuihin sekä vastaa levykuvien luonnista ja varastoinnista. Levykuva -palvelun nimi OpenStack -ympäristössä on Glance.

5.2.1 Esivalmistelut

Ennen Levykuvapalvelun asennusta ja määrittelyä tulee sitä varten luoda oma tietokanta, palvelutunnukset ja ohjelmointirajapinnan loppupäät. Aloitetaan avaamalla yhteys tietokantapalvelimeen **root** -käyttäjänä

```
ont@kenkalaatikko:~> mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 20
Server version: 10.2.31-MariaDB SUSE package

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Luodaan tietokanta Glance:lle, nimeltään "glance"

```
MariaDB [(none)]> CREATE DATABASE glance;
Query OK, 1 row affected (0.00 sec)
```

Annetaan juuri luotuun "glance" -tietokantaan kaikki oikeudet "glance" -nimiselle käyttäjätunnukselle

```
;)MariaDB [(none)]>GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' IDENTIFIED BY 'XXXXXXXXXX';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' IDENTIFIED BY 'XXXXXXXXXX';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> []
```

Valitaan jälleen vahva salasana, kuvassa salasana sensuroitu. Salasanaan viitataan jatkossa nimellä "Glance_db". Tämän jälkeen poistutaan tietokantahallintasovelluksesta

```
MariaDB [(none)]> exit
Bye
ont@kenkalaatikko:~> []
```

Ajetaan kohdassa 5.1.6 luotu "admin-openrc" skripti

```
ont@kenkalaatikko:~> . admin-openrc
ont@kenkalaatikko:~> []
```

Näin varmistetaan, että käytössä ovat korotetut käyttöoikeudet. Nyt voidaan luoda levykuvapalvelun vaatimat käyttäjätunnukset OpenStack -ympäristöön. Aloitetaan luomalla käyttäjä "glance" komennolla

```
ont@kenkalaatikko:~> openstack user create --domain default --password-prompt glance
User Password:
Repeat User Password:
+-----+-----+
| Field | Value |
+-----+-----+
| domain_id | default |
| enabled | True |
| id | 9727676f92e84f78abd661086c5fe575 |
| name | glance |
| options | {} |
| password_expires_at | None |
+-----+-----+
ont@kenkalaatikko:~> []
```

Komento pyytää antamaan käyttäjälle salasanan, siihen viitataan jatkossa nimellä "Glance_user". Käyttäjätunnuksen luonnin onnistuttua tulosteena taulukko, jossa jälleen yksilöintinumero, nimi ja salasanan vanhenemispäivämäärä. Seuraavaksi lisätään käyttäjälle "glance" rooli "admin". Lisätään rooli "admin" myös projektiin "service".

```
ont@kenkalaatikko:~> openstack role add --project service --user glance admin
ont@kenkalaatikko:~> []
```

Seuraavaksi luodaan "glance" palvelu entiteetti OpenStack ympäristöön komennolla

```
ont@kenkalaatikko:~> openstack service create --name glance --description "OpenStack Image" image
+-----+-----+
| Field | Value |
+-----+-----+
| description | OpenStack Image |
| enabled | True |
| id | 961588953e1b480b9020ed51a79cf48c |
| name | glance |
| type | image |
+-----+-----+
ont@kenkalaatikko:~> []
```

Luodaan vielä loppupäät ohjelmointirajapinnalle. Julkisenverkon loppupää luodaan komennolla

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne image public http://kenkalaatikko:9292
+-----+
| Field      | Value                                     |
+-----+
| enabled    | True                                     |
| id         | 5a1f39f643b74b81836f659c203dc3b6       |
| interface  | public                                  |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | 961588953e1b480b9020ed51a79cf48c     |
| service_name | glance                                  |
| service_type | image                                   |
| url        | http://kenkalaatikko:9292             |
+-----+
ont@kenkalaatikko:~> □
```

Sisäisenverkon

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne image internal http://kenkalaatikko:9292
+-----+
| Field      | Value                                     |
+-----+
| enabled    | True                                     |
| id         | 752c37d8c975453d909938ffa3c2f74b     |
| interface  | internal                                 |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | 961588953e1b480b9020ed51a79cf48c     |
| service_name | glance                                  |
| service_type | image                                   |
| url        | http://kenkalaatikko:9292             |
+-----+
ont@kenkalaatikko:~> □
```

Ja lopuksi hallintaverkon

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne image admin http://kenkalaatikko:9292
+-----+
| Field      | Value                                     |
+-----+
| enabled    | True                                     |
| id         | c602c461fbe74b059211176c315a5211     |
| interface  | admin                                    |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | 961588953e1b480b9020ed51a79cf48c     |
| service_name | glance                                  |
| service_type | image                                   |
| url        | http://kenkalaatikko:9292             |
+-----+
ont@kenkalaatikko:~> □
```

5.2.2 Komponenttien asennus ja määrittäminen

Asennetaan Glance:n tarvitsemat paketit komennolla

```
ont@kenkalaatikko:~> sudo zypper in openstack-glance openstack-glance-api
[sudo] password for root:
Retrieving repository 'Train' metadata .....[done]
Building repository 'Train' cache .....[done]
Retrieving repository 'Main Update Repository' metadata .....[done]
Building repository 'Main Update Repository' cache .....[done]
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following 53 NEW packages are going to be installed:
```

Asennuksen onnistuttua editoidaan tiedostoa `"/etc/glance/glance-api.conf"`.

Lisätään kohtaan **[database]** "connection" -rivi

```
[database]
connection = mysql+pymysql://glance:[REDACTED]@kenkalaatikko/glance
```

Salasanana "Glance_db", joka määriteltiin "glance" -tunnukselle tietokannan luonnin yhteydessä.

Kohtaan **[keystone_authtoken]** lisätään seuraavat rivit

```
GNU nano 2.9.6 /etc/glance/glance-api.conf

[keystone_authtoken]

www_authenticate_uri = http://kenkalaatikko:5000
auth_url = http://kenkalaatikko:5000
memcached_servers = kenkalaatikko:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = glance
password = [REDACTED]
```

Salasanana "Glance_user", joka määriteltiin "glance" -tunnukselle OpenStack työkalulla.

Lisätään **[paste_deploy]** kohtaan seuraava rivi

```
GNU nano 2.9.6 /etc/glance/glance-api.conf

[paste_deploy]

flavor = keystone
```

Lopuksi vielä määritellään kohdassa **[glance_store]** levykuvien tallennusmuoto ja -sijainti, rivein

```
GNU nano 2.9.6 /etc/glance/glance-api.conf

[glance_store]

stores = file,http
default_store = file
filesystem_store_datadir = /var/lib/glance/images/[]
```

Tallennetaan tiedosto ja poistutaan tekstieditorista.

Nyt kun asetukset on määritelty, asetetaan vielä Levykuvapalvelu käynnistymään järjestelmän käynnistyessä sekä käynnistetään palvelu nyt komennoin

```
ont@kenkalaatikko:~> sudo systemctl enable openstack-glance-api.service
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-glance-api.service → /usr/lib/systemd/system/openstack-glance-api.service.
ont@kenkalaatikko:~> sudo systemctl start openstack-glance-api.service
ont@kenkalaatikko:~> []
```

5.2.3 Toiminnan testaus

Testataan Levykuvapalvelun toimintaa käyttäen CirrOS -nimistä, pientä, pilvipalveluille suunniteltua Linux jakelua. Aloitetaan lataamalla jälleen **admin** -tunnuksen ympäristömuuttujat. Vaikka ympäristömuuttujien ei pitäisi vaihtua kesken kaiken, on hyvä varmistaa, että oikeat muuttujat ovat käytössä. Ladataan muuttujat

```
ont@kenkalaatikko:~> . admin-openrc
ont@kenkalaatikko:~> █
```

Ladataan CirrOS:n levykuva viralliselta sivulta komennolla

```
ont@kenkalaatikko:~> wget http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img
```

Hetken päästä levykuva on latautunut

```
Length: 12716032 (12M) [application/octet-stream]
Saving to: âcirros-0.4.0-x86_64-disk.imgâ
cirros-0.4.0-x86_64-disk.img 100%[=====>] 12.13M 3.95MB/s in 3.1s
2020-03-10 16:06:58 (3.95 MB/s) - âcirros-0.4.0-x86_64-disk.imgâ saved [12716032/12716032]
ont@kenkalaatikko:~> █
```

Asetetaan levykuva Levykuvapalvelun käytettäväksi käyttäen OpenStackin tukemaan QCOW2 levyformaattia ja *bare* säiliöformaattia. Lisäksi asetetaan levykuva näkymään julkisesti, jotta kaikilla projekteilla on pääsy siihen. Tämä tapahtuu komennolla

```
ont@kenkalaatikko:~> glance image-create --name "cirros" --file cirros-0.4.0-x86_64-disk.img --disk-format qcow2
--container-format bare --visibility public
+-----+
| Property | Value |
+-----+
| checksum | 443b7623e27ecf03dc9e01ee93f67afe |
| container_format | bare |
| created_at | 2020-03-10T14:16:09Z |
| disk_format | qcow2 |
| id | 55e63437-2130-446a-a6c9-3cd23a4b9009 |
| min_disk | 0 |
| min_ram | 0 |
| name | cirros |
| os_hash_algo | sha512 |
| os_hash_value | 6513f21e44aa3da349f248188a44bc304a3653a04122d8fb4535423c8e1d14cd6a153f735bb0982e
| 2161b5b5186106570c17a9e58b64dd39390617cd5a350f78 |
| os_hidden | False |
| owner | 4a84a080e6b64d3195c74c0306d7769e |
| protected | False |
| size | 12716032 |
| status | active |
| tags | [] |
| updated_at | 2020-03-10T14:16:10Z |
| virtual_size | Not available |
| visibility | public |
+-----+
ont@kenkalaatikko:~> █
```

Onnistuneen suorituksen tulosteena jälleen taulukko, jossa levykuvan keskeisimmät tiedot. Varmistetaan vielä komennolla

```
ont@kenkalaatikko:~> glance image-list
+-----+
| ID | Name |
+-----+
| 55e63437-2130-446a-a6c9-3cd23a4b9009 | cirros |
+-----+
ont@kenkalaatikko:~> █
```

5.3 Resurssien seuranta palvelu

OpenStackissa käytetään Placement-nimistä palvelua resurssien seurantaan. Placement:in avulla muuta palvelu pitävät kirjaa kuinka paljon resursseja on käytössä. Esimerkiksi Lasketa -palvelu laskentakoneissa kertoo Placement:in kautta kontrollerille käytössä olevan kapasiteettinsa. Näin ollen kontrolleri voi päätellä mille laskentakoneelle seuraava instanssi voidaan antaa käsiteltäväksi.

5.3.1 Esivalmistelut

Aloitetaan luomalla tietokanta Placement -palvelua varten. Avataan siis tietokantahallinta-ohjelma komennolla

```
ont@kenkalaatikko:~> mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 20
Server version: 10.2.31-MariaDB SUSE package

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> 
```

Luodaan tietokanta komennolla

```
MariaDB [(none)]> CREATE DATABASE placement;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> 
```

Annetaan juuri luotuun "placement" -tietokantaan kaikki oikeudet "placement" -nimiselle käyttäjätunnukselle

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON placement.* TO 'placement'@'localhost' IDENTIFIED BY '':
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON placement.* TO 'placement'@'%' IDENTIFIED BY '':
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> 
```

Valitaan jälleen vahva salasana, kuvassa salasana sensuroitu. Salasanaan jatkossa viitataan nimellä "Placement_db". Tämän jälkeen poistutaan tietokantahallintasovelluksesta

```
MariaDB [(none)]> exit
Bye
ont@kenkalaatikko:~> 
```

Ladataan jälleen korotetut oikeudet OpenStack ympäristöön ajamalla ympäristömuuttaja skripti

```
ont@kenkalaatikko:~> . admin-openrc
ont@kenkalaatikko:~> 
```

Luodaan Placement -palvelun vaatima "placement" -käyttäjätunnus OpenStack -ympäristöön komennolla

```
ont@kenkalaatikko:~> openstack user create --domain default --password-prompt placement
User Password:
Repeat User Password:
+-----+-----+
| Field          | Value          |
+-----+-----+
| domain_id     | default        |
| enabled       | True           |
| id            | 438b0d950fe24dcb90a31e454b39efad |
| name          | placement      |
| options       | {}             |
| password_expires_at | None          |
+-----+-----+
ont@kenkalaatikko:~> █
```

Valitaan vahva salasana, johon myöhemmin viitataan nimellä "Placement_user". Lisätään "placement" -tunnus "service" projektiin sekä "admin" rooliin komennolla

```
ont@kenkalaatikko:~> openstack role add --project service --user placement admin
ont@kenkalaatikko:~> █
```

Luodaan Placement -palvelulle rajapinta palvelukatalogiin komennolla

```
ont@kenkalaatikko:~> openstack service create --name placement --description "Placement API" placement
+-----+-----+
| Field          | Value          |
+-----+-----+
| description    | Placement API  |
| enabled       | True           |
| id            | 9d3eba09b4364f54896532c821486eda |
| name          | placement      |
| type          | placement      |
+-----+-----+
ont@kenkalaatikko:~> █
```

Onnistuneen komennon tulosteena jälleen taulukko, jossa muun muassa yksilöintikoodi.

Luodaan loppupää ohjelmointirajapinnoille. Julkisen verkon loppupää

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne placement public http://kenkalaatikko:8778
+-----+-----+
| Field          | Value          |
+-----+-----+
| enabled       | True           |
| id            | bfdbf2be19e04692b9a57d7f7a70f7f5 |
| interface     | public         |
| region        | RegionOne      |
| region_id     | RegionOne      |
| service_id    | 9d3eba09b4364f54896532c821486eda |
| service_name  | placement      |
| service_type  | placement      |
| url           | http://kenkalaatikko:8778 |
+-----+-----+
ont@kenkalaatikko:~> █
```

Sisäisenverkon

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne placement internal http://kenkalaatikko:8778
+-----+-----+
| Field          | Value          |
+-----+-----+
| enabled       | True           |
| id            | 30ab92c0c79548498683f52b8570ee23 |
| interface     | internal       |
| region        | RegionOne      |
| region_id     | RegionOne      |
| service_id    | 9d3eba09b4364f54896532c821486eda |
| service_name  | placement      |
| service_type  | placement      |
| url           | http://kenkalaatikko:8778 |
+-----+-----+
ont@kenkalaatikko:~> █
```

Hallintaverkon

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne placement admin http://kenkalaatikko:8778
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | c715a3ca78d6473ea980539e4c6a7ece       |
| interface  | admin                                    |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | 9d3eba09b4364f54896532c821486eda       |
| service_name | placement                               |
| service_type | placement                               |
| url        | http://kenkalaatikko:8778              |
+-----+-----+
ont@kenkalaatikko:~> □
```

Jokaisessa komennossa onnistuessaan tulosteena taulukko, jossa rajapinnan loppupään tiedot.

5.3.2 Komponenttien asennus ja määrittely

Asennetaan Placement -palvelun tarvitsemat paketit komennolla

```
ont@kenkalaatikko:~> sudo zypper in openstack-placement
Retrieving repository 'Train' metadata .....[done]
Building repository 'Train' cache .....[done]
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following 6 NEW packages are going to be installed:
  openstack-placement python3-microversion_parse python3-openstack-placement python3-os-resource-classes
  python3-os-traits python3-wsgi_intercept

6 new packages to install.
Overall download size: 650.7 KiB. Already cached: 0 B. After the operation, additional 2.5 MiB will be used.
Continue? [y/n/v/...? shows all options] (y): □
```

Asennusohjelma kertoo, että asennettavana on 6 uutta pakettia. Vastataan kyllä asennuksen kysymykseen, jatketaan asennusta. Seuraavaksi editoidaan tiedosta `"/etc/placement/placement.conf"`. Lisätään kohtaan `[placement_database]` rivi `"connction"`

```
[placement_database]
connection = mysql+pymysql://placement:██████████@kenkalaatikko/placement
□
```

Salasanana "Placement_db", joka luotiin tietokantahallintaohjelmalla aikaisemmassa kohdassa. Kuvassa salasana sotkettu. Seuraavaksi lisätään kohtaan `[api]` rivi `"auth_strategy"`

```
GNU nano 2.9.6 /etc/placement/placement.conf

[api]
auth_strategy = keystone
```

Kohtaan **[keystone_authtoken]** lisätään seuraavat rivit

```
GNU nano 2.9.6 /etc/placement/placement.conf

[keystone_authtoken]

auth_url = http://kenkalaatikko:5000/v3
memcached_servers = kenkalaatikko:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = placement
password = ████████████████████
█
```

Salasanana "Placement_user"- niminen. Tallennetaan ja poistutaan tekstieditorista. Lisätään "Placement" -tietokantaan dataa komennolla

```
ont@kenkalaatikko:~> sudo su -s /bin/sh -c "placement-manage db sync" placement
[sudo] password for root:
ont@kenkalaatikko:~> █
```

Tässä kohtaa ohjeessa neuvotaan uudelleennimeämään tiedosto `"/etc/apache2/vhosts.d/openstack-placement-api.conf.sample"`. Jostain syystä tiedostoa ei löydy järjestelmästäni. Onneksi kuitenkin hätä ei ole tämännäköinen, sillä tiedoston voi ladata OpenStack'in Github jakelusta. Tehdään tämä ja nimetään tiedosto ohjeen mukaisesti, komennoin

```
ont@kenkalaatikko:~> wget https://raw.githubusercontent.com/openstack/rpm-packaging/master/openstack/openstack-pl
acement/openstack-placement-api.conf.sample
--2020-03-17 20:47:03-- https://raw.githubusercontent.com/openstack/rpm-packaging/master/openstack/openstack-pla
cement/openstack-placement-api.conf.sample
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.84.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.84.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 757 [text/plain]
Saving to: 'openstack-placement-api.conf.sample'

openstack-placement-api.conf 100%[=====] 757 --.-KB/s in 0s

2020-03-17 20:47:04 (16.3 MB/s) - 'openstack-placement-api.conf.sample' saved [757/757]

ont@kenkalaatikko:~> sudo mv ./openstack-placement-api.conf.sample /etc/apache2/vhosts.d/openstack-placement-api.
conf
ont@kenkalaatikko:~> █
```

Tässä kohtaa toki herää epäilykseni mitä muuta asennuksesta puuttuu. Tämän jälkeen vielä käynnistetään Apache -palvelu uudestaan komennolla

```
ont@kenkalaatikko:~> sudo systemctl reload apache2.service
ont@kenkalaatikko:~> █
```

5.3.3 Asennuksen toiminnan varmistus

Aloitetaan lataamalla taas **admin** ympäristömuuttujat

```
ont@kenkalaatikko:~> . admin-openrc
ont@kenkalaatikko:~> █
```


Nyt voidaan yrittää uudestaan listata resurssiluokat sekä -arvot komennoin

```
ont@kenkalaatikko:~> openstack --os-placement-api-version 1.2 resource class list --sort-column name
+-----+
| name |
+-----+
| DISK_GB |
| FPGA |
| IPV4_ADDRESS |
| MEMORY_MB |
| MEM_ENCRYPTION_CONTEXT |
| NET_BW_EGR_KILOBIT_PER_SEC |
| NET_BW_IGR_KILOBIT_PER_SEC |
| NUMA_CORE |
| NUMA_MEMORY_MB |
| NUMA_SOCKET |
| NUMA_THREAD |
| PCI_DEVICE |
| PCPU |
| PGPU |
| SRIOV_NET_VF |
| VCPU |
| VGPU |
| VGPU_DISPLAY_HEAD |
+-----+
ont@kenkalaatikko:~> 
```

```
ont@kenkalaatikko:~> openstack --os-placement-api-version 1.6 trait list --sort-column name
```

Jälkimmäisen komennon tulosteena erittäin pitkä lista, joka ei mahdu pituudeltaan SSH-ikkunaan.

5.4 Laskenta -palvelu

Laskenta-palvelu on OpenStackin loppukäyttäjälle näkyvin osuus. Se vastaa, muiden palveluiden tuella, käyttäjien istuntojen tehtävien suorittamisesta. Laskenta-palvelu jakautuu kahteen osaan: kontrollerin hoitamaan hallinnolliseen osaan sekä laskentakoneiden suorittamaan varsinaiseen laskentaosioon.

5.4.1 Esivalmistelut

Ennen Laskenta -palvelun asennusta, luodaan jälleen palvelun tarvitsema tietokanta, palvelu tunnukset sekä ohjelmointirajapinnan loppupää. Aloitetaan tietokantojen luonti avaamalla tietokantahallintaohjelmisto

```
ont@kenkalaatikko:~> mysql -u root -p
Enter password: 
```

Seuraavaksi luodaan palvelulle tietokannat **nova**, **nova_api** sekä **nova_cell0**

```
MariaDB [(none)]> CREATE DATABASE nova_api;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> CREATE DATABASE nova;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> CREATE DATABASE nova_cell0;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> 
```

Luodaan käyttäjä **nova** sekä annetaan sille juuri luotuihin tietokantoihin tarvittavat oikeudet komennoin

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'localhost' IDENTIFIED BY '
';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'%' IDENTIFIED BY '
';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' IDENTIFIED BY '
';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY '
';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_cell0.* TO 'nova'@'localhost' IDENTIFIED BY '
!!!';
Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_cell0.* TO 'nova'@'%' IDENTIFIED BY '
';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>
```

Salasanaksi vahva salasana, johon jatkossa viitataan nimellä "Nova_db". Salasanat jälleen sotkettuina. Kun oikeudet on annettu, voidaan poistua tietokantahallintaohjelmasta

```
MariaDB [(none)]> exit
Bye
```

Ladataan jälleen **admin** -käyttäjän ympäristömuuttujat

```
ont@kenkalaatikko:~> . admin-openrc
ont@kenkalaatikko:~>
```

Luodaan **nova** -käyttäjätunnus Openstack ympäristöön komennolla

```
ont@kenkalaatikko:~> openstack user create --domain default --password-prompt nova
User Password:
Repeat User Password:
+-----+
| Field          | Value                                     |
+-----+-----+
| domain_id     | default                                 |
| enabled       | True                                    |
| id            | 7d4ca9cdea6a4de6b8e314b3aceb8b6f      |
| name         | nova                                    |
| options       | {}                                       |
| password_expires_at | None                                   |
+-----+-----+
ont@kenkalaatikko:~>
```

Valitaan jälleen vahva salasana, joka jatkossa nimellä "Nova_user". Lisätään juuri luodulle **nova** -käyttäjälle **admin** -rooli, komennolla

```
ont@kenkalaatikko:~> openstack role add --project service --user nova admin
ont@kenkalaatikko:~>
```

Luodaan **nova** -palvelu entiteetti komennolla

```
ont@kenkalaatikko:~> openstack service create --name nova --description "OpenStack Compute" compute
+-----+
| Field      | Value                                     |
+-----+
| description | OpenStack Compute                       |
| enabled     | True                                     |
| id          | 1c5fd7394beb41e8be00f1e355620f1b      |
| name        | nova                                     |
| type        | compute                                  |
+-----+
ont@kenkalaatikko:~> █
```

Luodaan vielä lopuksi ohjelmointirajapinnoille loppupää. Aloitetaan julkisenverkon loppupäällä

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne compute public http://kenkalaatikko:8774/v2.1
+-----+
| Field      | Value                                     |
+-----+
| enabled     | True                                     |
| id          | 748126803fe04434b76e0e49b66f6667      |
| interface   | public                                  |
| region      | RegionOne                               |
| region_id   | RegionOne                               |
| service_id  | 1c5fd7394beb41e8be00f1e355620f1b      |
| service_name | nova                                    |
| service_type | compute                                  |
| url         | http://kenkalaatikko:8774/v2.1         |
+-----+
ont@kenkalaatikko:~> █
```

Sisäinen verkko

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne compute internal http://kenkalaatikko:8774/v2.1
+-----+
| Field      | Value                                     |
+-----+
| enabled     | True                                     |
| id          | 9ef09aadf1cc48d3b526729ac048c86b      |
| interface   | internal                                 |
| region      | RegionOne                               |
| region_id   | RegionOne                               |
| service_id  | 1c5fd7394beb41e8be00f1e355620f1b      |
| service_name | nova                                    |
| service_type | compute                                  |
| url         | http://kenkalaatikko:8774/v2.1         |
+-----+
ont@kenkalaatikko:~> █
```

Ja lopuksi vielä hallintaverkko

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne compute admin http://kenkalaatikko:8774/v2.1
+-----+
| Field      | Value                                     |
+-----+
| enabled     | True                                     |
| id          | 5620ad677efa4676a223e84ae10557b1      |
| interface   | admin                                    |
| region      | RegionOne                               |
| region_id   | RegionOne                               |
| service_id  | 1c5fd7394beb41e8be00f1e355620f1b      |
| service_name | nova                                    |
| service_type | compute                                  |
| url         | http://kenkalaatikko:8774/v2.1         |
+-----+
ont@kenkalaatikko:~> █
```

5.4.2 Pakettien asennus ja määrittys kontrolleri koneelle

Asennetaan paketit komennolla

```
ont@kenkalaatikko:~> sudo zypper in openstack-nova-api openstack-nova-scheduler openstack-nova-co
nductor openstack-nova-novncproxy iptables
[sudo] password for root:
Retrieving repository 'Main Update Repository' metadata .....[done]
Building repository 'Main Update Repository' cache .....[done]
Loading repository data...
Reading installed packages...
'iptables' is already installed.
No update candidate for 'iptables-1.6.2-1p151.2.3.x86_64'. The highest available version is alrea
dy installed.
Resolving package dependencies...

The following 29 NEW packages are going to be installed:
 libdpdk-18_11 libibverbs libibverbs1 libmlx4-1 libmlx5-1 libopenvswitch-2_11-0 libsodium23
 novnc openstack-nova openstack-nova-api openstack-nova-conductor openstack-nova-novncproxy
 openstack-nova-scheduler python3-gssapi python3-neutronclient python3-nova
 python3-os-client-config python3-oslo.versionedobjects python3-os-vif python3-ovs
 python3-ovsdbapp python3-paramiko python3-PyNaCl python3-pyroute2 python3-tooz
 python3-voluptuous python3-websocketify python-websocketify-common rdma-core

The following recommended package was automatically selected:
 python3-gssapi

29 new packages to install.
Overall download size: 16.3 MiB. Already cached: 0 B. After the operation, additional 91.1 MiB
will be used.
Continue? [y/n/v/...? shows all options] (y): █
```

Editoidaan tiedostoa `"/etc/nova/nova.conf"`. Kohdassa **[DEFAULT]** lisätään rivi `"enabled_apis"`

```
GNU nano 2.9.6 /etc/nova/nova.conf

# List of APIs to be enabled by default (list value)
enabled_apis = osapi_compute,metadata
```

Kohdassa **[api-database]** lisätään rivi `"connection"`

```
GNU nano 2.9.6 /etc/nova/nova.conf

[api_database]
#
# The *Nova API Database* is a separate database which is used for information
# which is used across *cells*. This database is mandatory since the Mitaka
# release (13.0.0).
#
# This group should **not** be configured for the ``nova-compute`` service.
#
# From nova.conf
#
# The SQLAlchemy connection string to use to connect to the database. Do not set
# this for the ``nova-compute`` service (string value)
connection = mysql+pymysql://nova:[REDACTED]@kenkalaatikko/nova_api
█
```

Salasana `"Nova_db"`, eli sama kuin aikaisemmin määriteltiin tietokantojen luonnin yhteydessä, kuvassa sensuroituna.

Kohdassa **[database]** lisätään myös rivi "connection"

```
GNU nano 2.9.6 /etc/nova/nova.conf

[database]
#
# From oslo.db
#
# If True, SQLite uses synchronous mode (boolean value)
#sqlite_synchronous = true
#
# The back end to use for the database (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend = sqlalchemy
#
# The SQLAlchemy connection string to use to connect to the database (string
# value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql+pymysql://nova:[REDACTED]@kenkalaatikko/nova
█
```

Salasana "Nova_db", sama kuin edellisessä kohdassa, jälleen sensuroitu sotkemalla. Lisätään vielä **[DEFAULT]** kohtaan rivi "transport_url"

```
GNU nano 2.9.6 /etc/nova/nova.conf

transport_url = rabbit://openstack:[REDACTED]@kenkalaatikko:5672/
█
```

Tässä salasanana "Rabbit_user" niminen salasana. Kuvassa salasana jälleen sotkettuna. Kohtaan **[api]** lisätään rivi "auth_strategy"

```
GNU nano 2.9.6 /etc/nova/nova.conf

[api]
#
# Options under this group are used to define Nova API.
#
# From nova.conf
#
#
# Determine the strategy to use for authentication.
# (string value)
# Possible values:
# keystone - Use keystone for authentication.
# noauth2 - Designed for testing only, as it does no actual credential checking.
# 'noauth2' provides administrative credentials only if 'admin' is specified as
# the username.
auth_strategy = keystone
█
```

Kohtaan **[keystone_authtoken]** lisätään seuraavat rivit ja poistetaan muut valinnat

```
GNU nano 2.9.6 /etc/nova/nova.conf

# The maximum number of retries that should be attempted for retrieable HTTP
# status codes (integer value)
#status_code_retries = <None>

# Delay (in seconds) between two retries for retrieable status codes. If not set,
# exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is
# used (floating point value)
#status_code_retry_delay = <None>

[keystone_authtoken]

www_authenticate_uri = http://kenkalaatikko:5000/
auth_url = http://kenkalaatikko:5000/
memcached_servers = kenkalaatikko:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = [REDACTED]
```

Salasana on "Nova_user", joka määriteltiin **nova** -käyttäjätunnuksen luonninyhteydessä edellisessä vaiheessa. Salasana jälleen sotkettuna kuvassa.

Palataan jälleen kohtaan **[DEFAULT]** ja lisätään rivit "my_ip", "use_neutron" sekä "fire-wall_driver"

```
GNU nano 2.9.6 /etc/nova/nova.conf

[DEFAULT]

enabled_apis = osapi_compute,metadata
transport_url = rabbit://openstack:[REDACTED]@kenkalaatikko:5672/
my_ip = 192.168.10.110
use_neutron = true
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

Kohdassa **[vnc]** lisätään rivit "enabled", "server_listen" sekä "server_proxyclient_address"

```
GNU nano 2.9.6 /etc/nova/nova.conf

[vnc]

enabled = true
server_listen = $my_ip
server_proxyclient_address = $my_ip
```


Luodaan **cell1** solu komennolla

```
ont@kenkalaatikko:~> sudo su -s /bin/sh -c "nova-manage cell_v2 create_cell --name=cell1 --verbose" nova
d5e9ba2b-82d0-4d27-8030-817cbdb1b85b
ont@kenkalaatikko:~> █
```

Komennon tulosteena solun yksilöintinumero. Täytetään seuraavaksi **nova** tietokanta komennolla

```
ont@kenkalaatikko:~> sudo su -s /bin/sh -c "nova-manage db sync" nova
/usr/lib/python3.6/site-packages/pymysql/cursors.py:170: Warning: (1831, 'Duplicate index `block_device_mapping_instance_uuid_virtual_name_device_name_idx`. This is deprecated and will be disallowed in a future release')
  result = self._query(query)
/usr/lib/python3.6/site-packages/pymysql/cursors.py:170: Warning: (1831, 'Duplicate index `unique_instances0uuid`. This is deprecated and will be disallowed in a future release')
  result = self._query(query)
ont@kenkalaatikko:~> █
```

Tulosteena muutama huomautusviesti, jotka ohjeen mukaan voidaan jättää huomioimatta. Varmistetaan vielä, että solut **cell0** ja **cell1** ovat rekisteröityneet oikein komennolla

```
ont@kenkalaatikko:~> sudo su -s /bin/sh -c "nova-manage cell_v2 list_cells" nova
-----+-----
| Name |          UUID |          Transport URL |          Database Connection | Disabled |
-----+-----
| cell0 | 00000000-0000-0000-0000-000000000000 |          none:/ | mysql+pymysql://nova:***@kenkalaatikko/nova_cell0 | False |
| cell1 | d5e9ba2b-82d0-4d27-8030-817cbdb1b85b | rabbit://openstack:***@kenkalaatikko:5672/ | mysql+pymysql://nova:***@kenkalaatikko/nova | False |
-----+-----
ont@kenkalaatikko:~> █
```

Tulosteena taulukko, jossa solujen yhteysosoitteet sekä tieto, että niitä ei ole poistettu käytöstä.

Lopuksi viimeistellään asennus lisäämällä "Nova API", "Nova scheduler", "Nova conductor" sekä "Nova novncproxy" järjestelmän kanssa käynnistyviin palveluihin, sekä käynnistetään palvelut heti. Tämä tapahtuu komennoin

```
ont@kenkalaatikko:~> sudo systemctl enable openstack-nova-api.service openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-novncproxy.service
[sudo] password for root:
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-nova-api.service â /usr/lib/systemd/system/openstack-nova-api.service.
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-nova-scheduler.service â /usr/lib/systemd/system/openstack-nova-scheduler.service.
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-nova-conductor.service â /usr/lib/systemd/system/openstack-nova-conductor.service.
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-nova-novncproxy.service â /usr/lib/systemd/system/openstack-nova-novncproxy.service.
ont@kenkalaatikko:~> █
```

```
ont@kenkalaatikko:~> sudo systemctl start openstack-nova-api.service openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-novncproxy.service
```

Komento ei anna onnistuessaan mitään tulostetta, joten voidaan siirtyä seuraavaan vaiheeseen.

5.4.3 Pakettien asennus ja määrittäminen laskentakoneelle

Seuraava vaihe tehdään kaikille kolmelle laskentakoneelle, *mustikka*, *vadelma* sekä *herukka*. Aloitetaan asentamalla tarvittavat paketit

```
pi@mustikka:~ $ sudo apt-get install nova-compute
Luetaan pakettiluetteloita... Valmis
Muodostetaan riippuvuussuhteiden puu
Luetaan tilatiedot... Valmis
The following additional packages will be installed:
```

Asennuksen jälkeen editoidaan tiedostoa `/etc/nova/nova.conf`. Kohtaan **[DEFAULT]** määritellään viestijonon tiedot, kunkin kone yksilöllinen IP-osoite, kerrotaan järjestelmälle, että käytetään Neutron-palvelua sekä määritellään palomuurin ajuri

```
GNU nano 3.2 /etc/nova/nova.conf
[DEFAULT]
#
# From nova.conf
#
transport_url = rabbit://openstack:Rabbit_user@192.168.10.130:5672@kenkalaatikko
my_ip = 192.168.10.130
use_neutron = true
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

Salasana "Rabbit_user", sotkettu tietoturva syistä. Kohdissa **[api]** sekä **[keystone_auth-token]** määritellään Identiteetti-palvelun tarvitsemat tiedot

```
GNU nano 3.2 /etc/nova/nova.conf
[api]
#
# Options under this group are used to define Nova API.
#
# From nova.conf
#
auth_strategy = keystone
```


Kohtaan **[oslo_concurrency]** määritellään lukkotiedoston sijainti

```
GNU nano 3.2 /etc/nova/nova.conf

[oslo_concurrency]

#
# From oslo.concurrency
#

lock_path = /var/lib/nova/tmp
```

Lopuksi vielä kohdassa **[placement]** määritellään Resurssihallinta-palvelun rajapinnan yhteydet

```
GNU nano 3.2 /etc/nova/nova.conf

[placement]

#
# From nova.conf
#

region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://kenkalaatikko:5000/v3
username = placement
password = ██████████
```

Salasana "Placement_user", jälleen sotkettu. Nyt tiedosto voidaan sulkea. Kun asetukset on määritetty, tarkastetaan, tukeeko käytössä oleva kone rautatason kiihdytystä virtuaali-koneille. Tämä tapahtuu komennolla

```
pi@mustikka:~ $ egrep -c '(vmx|svm)' /proc/cpuinfo
0
pi@mustikka:~ $ █
```

Koska vastaus on "0", eli ei tue, tulee koneen määrittää käyttämään **Qemu**:a **KVM**:n sijasta virtualisoinnin suorittamiseen. Editoidaan tiedostoa `"/etc/nova/nova-compute.conf"` ja

lisätään kohdan **[libvirt]** alle seuraava rivi

```
GNU nano 3.2 /etc/nova/nova.conf

[libvirt]
#
# Libvirt options allows cloud administrator to configure related
# libvirt hypervisor driver to be used within an OpenStack deployment.
#
# Almost all of the libvirt config options are influence by ``virt_type`` config
# which describes the virtualization type (or so called domain type) libvirt
# should use for specific features such as live migration, snapshot.

virt_type = qemu
```

Lopetetaan asennus käynnistämällä Laskenta-palvelun komponentit uudestaan komennolla

```
pi@herukka:~ $ sudo service nova-compute restart
```

5.4.4 Laskenta koneiden rekisteröinti kontrollerille

Kun kaikkien laskentakoneiden asennus on suoritettu loppuun, siirrytään takaisin kontrollerikoneelle *kenkalaatikko* ja ladataan käyttöön **admin**-tunnuksen ympäristömuuttujat

```
ont@kenkalaatikko:~> . admin-openrc
ont@kenkalaatikko:~> █
```

Tarkastetaan seuraavaksi ovatko laskentakoneet rekisteröityneet kontrollerin laskenta tietokantaan komennolla

```
ont@kenkalaatikko:~> openstack compute service list --service nova-compute
+-----+-----+-----+-----+-----+-----+-----+
| ID | Binary | Host | Zone | Status | State | Updated At |
+-----+-----+-----+-----+-----+-----+-----+
| 8 | nova-compute | mustikka | nova | enabled | up | 2020-04-19T11:13:56.000000 |
| 9 | nova-compute | herukka | nova | enabled | up | 2020-04-19T11:14:00.000000 |
+-----+-----+-----+-----+-----+-----+-----+
ont@kenkalaatikko:~> █
```

Tässä vaiheessa *mustikka* sekä *herukka* ovat rekisteröityneet. Rekisteröityminen voidaan myös pakottaa komennolla

```
ont@kenkalaatikko:~> sudo su -s /bin/sh -c "nova-manage cell_v2 discover_hosts --verbose" nova
[sudo] password for root:
Sorry, try again.
[sudo] password for root:
Found 2 cell mappings.
Skipping cell0 since it does not contain hosts.
Getting computes from cell 'cell1': d5e9ba2b-82d0-4d27-8030-817cddb1b85b
Checking host mapping for compute host 'herukka': 7c0fa9a2-98d8-4049-ab50-ec4779f0cd3e
Creating host mapping for compute host 'herukka': 7c0fa9a2-98d8-4049-ab50-ec4779f0cd3e
Found 1 unmapped computes in cell: d5e9ba2b-82d0-4d27-8030-817cddb1b85b
ont@kenkalaatikko:~> █
```

Ja näin kaikki koneet saadaan rekisteröityä

```
ont@kenkalaatikko:~> openstack compute service list --service nova-compute
+-----+-----+-----+-----+-----+-----+-----+
| ID | Binary          | Host      | Zone | Status | State | Updated At          |
+-----+-----+-----+-----+-----+-----+-----+
| 8  | nova-compute    | mustikka | nova | enabled | up    | 2020-04-20T10:36:16.000000 |
| 9  | nova-compute    | herukka  | nova | enabled | up    | 2020-04-20T10:36:08.000000 |
| 10 | nova-compute    | vadelma  | nova | enabled | up    | 2020-04-20T10:36:09.000000 |
+-----+-----+-----+-----+-----+-----+-----+
```

5.5 Verkko -palvelu

OpenStack Verkko -palvelu mahdollistaa muiden OpenStackin palveluiden käyttämien verkkolaitteiden lisäyksen ja keskitetyn hallinnoinnin. Liitännäisten avulla voidaan palvella erilaisia verkkolaitteita ja ohjelmistoja, tarjoten näin entistä enemmän joustavuutta OpenStack ympäristön arkkitehtuuriin ja toteutukseen. Verkko -palvelu keskustelee pääsääntöisesti Laskenta -palvelun kanssa tarjoten sen istunnoille verkon ja hallitut yhteydet. Verkko -palvelun nimi on Neutron.

5.5.1 Esivalmistelut

Esivalmisteluina jokaiselle koneelle tulee määrittää kiinteä IP-osoite sekä muut verkon vaatimat osoitteet, kuten oletusyhdyskäytävä ja nimipalvelimen osoite. Tämä vaihe on jo suoritettu aivan ensimmäisenä valmistautumistoimenpiteenä, joten sitä ei tarvitse enää tehdä uudestaan.

Esivalmistelut, joita kuitenkin tulee tehdä, on tietokannan sekä ohjelmointirajapintojen yhteyspisteiden luonti. Tämä vaihe tehdään kontrolleri koneella. Aloitetaan tietokannan luonti käynnistämällä tietokantahallintaohjelmisto komennolla

```
ont@kenkalaatikko:~> mysql -u root -p
Enter password: █
```

Kun ohjelmisto on käynnissä, luodaan tietokanta **neutron** komennolla

```
MariaDB [(none)]> CREATE DATABASE neutron;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> █
```

Annetaan tietokantaan oikeudet käyttäjälle **neutron** sekä asetetaan käyttäjälle **neutron** vahva salasana tietokantaa varten. Komennot

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' IDENTIFIED BY '██████████';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' IDENTIFIED BY '██████████';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> █
```

Salasanaan viitataan jatkossa nimellä "Neutron_db". Tämän jälkeen voidaan poistua tietokantahallinnasta.

```
MariaDB [(none)]> exit
Bye
```

Seuraavaksi luodaan **neutron** -tunnus OpenStack ympäristöön. Aloitetaan lataamalla **admin** ympäristömuuttujat

```
ont@kenkalaatikko:~> . admin-openrc
ont@kenkalaatikko:~> █
```

Luodaan tunnus

```
ont@kenkalaatikko:~> openstack user create --domain default --password-prompt neutron
User Password:
Repeat User Password:
+-----+-----+
| Field          | Value                               |
+-----+-----+
| domain_id     | default                             |
| enabled       | True                                 |
| id            | 533b30ca56254f579b8f8847b78df364   |
| name          | neutron                             |
| options       | {}                                   |
| password_expires_at | None                               |
+-----+-----+
ont@kenkalaatikko:~> █
```

Valitaan jälleen tunnukselle vahva salasana, johon jatkossa viitataan nimellä "Neutron_user" Onnistuneesti suoritettujen komentojen tulosteena jälleen taulukko, jossa muun muassa käyttäjän yksilöivä tunnusluku. Lisätään juuri luodulle tunnukselle **admin** rooli komennolla

```
ont@kenkalaatikko:~> openstack role add --project service --user neutron admin
ont@kenkalaatikko:~> █
```

Seuraavaksi luodaan **neutron** -palvelu entiteetti komennolla

```
ont@kenkalaatikko:~> openstack service create --name neutron --description "OpenStack Networking" network
+-----+-----+
| Field          | Value                               |
+-----+-----+
| description    | OpenStack Networking               |
| enabled       | True                                 |
| id            | 8b2549c35da64f1bb24b4365afbc913e   |
| name          | neutron                             |
| type          | network                             |
+-----+-----+
ont@kenkalaatikko:~> █
```

Lopuksi vielä luodaan ohjelmointirajapinnoille yhteyspisteet. Julkisenverkon yhteyspiste komennolla

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne network public http://kenkalaatikko:9696
+-----+-----+
| Field          | Value                               |
+-----+-----+
| enabled       | True                                 |
| id            | 4fa6d27d241e42fbb5b5dd6174cc9073   |
| interface     | public                             |
| region       | RegionOne                           |
| region_id    | RegionOne                           |
| service_id   | 8b2549c35da64f1bb24b4365afbc913e   |
| service_name | neutron                             |
| service_type | network                             |
| url          | http://kenkalaatikko:9696          |
+-----+-----+
ont@kenkalaatikko:~> █
```

Sisäisenverkon

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne network internal http://kenkalaatikko:9696
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | 1e548afb433a452c9e28cbc7a0cb23d2       |
| interface  | internal                                 |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | 8b2549c35da64f1bb24b4365afbc913e     |
| service_name | neutron                                 |
| service_type | network                                 |
| url        | http://kenkalaatikko:9696             |
+-----+-----+
ont@kenkalaatikko:~> █
```

Ja vielä hallintaverkon

```
ont@kenkalaatikko:~> openstack endpoint create --region RegionOne network admin http://kenkalaatikko:9696
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | 3e0bf0cfd8b4e719128b5ba24e234c8       |
| interface  | admin                                    |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | 8b2549c35da64f1bb24b4365afbc913e     |
| service_name | neutron                                 |
| service_type | network                                 |
| url        | http://kenkalaatikko:9696             |
+-----+-----+
ont@kenkalaatikko:~> █
```

Ja jokaisella komennolla suorituu onnistuneesti tulosteena taulukko, jossa verkon keskeisimmät tiedot.

5.5.2 Verkon asetusten määrittäminen

Neutronissa on valittavissa kaksi erilaista verkon tyyppiä. Nämä ovat 1) Tarjoajan verkko ja 2) Itsepalvelu verkko. Käytännössä suurin ero on reitityksessä; hoitaako Neutron OSI-mallin tason 3 operaatiot (vaihtoehto 2) vai hoitaako sen ulkopuolinen taho (vaihtoehto 1). Tässä asennuksessa käytän vaihtoehtoa 1, tällöin ulkoinen reititin hoitaa reitityksen.

Koska OpenStackin oletusympäristö vaatii palvelimissa olevan kaksi verkkokorttia, ja käytössäni olevissa laitteissa on vain yksi verkkokortti virallisen ohjeen lisäksi, käytän lähteenä OpenStackin DevStack-kokeiluympäristöä varten tehtyä ohjetta (DevStack. 2019.). Aloitetaan kontrollerikoneella. Asennetaan tarvittavat paketit komennolla

```
ont@kenkalaatikko:~> sudo zypper install --no-recommends openstack-neutron openstack-neutron-server openstack-neutron-linuxbridge-agent openstack-neutron-dhcp-agent openstack-neutron-metadata-agent bridge-utils
[sudo] password for root:
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following 24 NEW packages are going to be installed:
```

Lisälippu "--no-recommends" kertoo asennusohjelmalle, että asennetaan vain tarvittavat paketit jättäen suositeltavat asentamatta. Asennusohjelma kertoo, että asennettavana on

24 pakettia. Sallitaan asennus, jonka jälkeen siirrytään editoimaan tiedostoa `"/etc/neutron/neutron.conf"`. Kohtaan **[database]** lisätään määrittelyt tietokantayhteydelle

```
GNU nano 2.9.6 /etc/neutron/neutron.conf

[database]
connection = mysql+pymysql://neutron:[REDACTED]@kenkalaatikko/neutron
#
```

Salasana on "Neutron_db", joka määriteltiin tietokantahallintaohjelmistossa käyttäjälle **neutron**. Salasana sensuroitu sotkemalla kuvassa.

Seuraavaksi kohtaan **[DEFAULT]** lisätään tieto käytettävästä liitännäisestä, Modular Layer 2 (ML2), poistetaan käytöstä muut liitännäiset, lisätään viestintämääritykset RabbitMQ -viestijonoon sekä kerrotaan, että käytetään Keystonea tunnistautumiseen

```
GNU nano 2.9.6 /etc/neutron/neutron.conf

[DEFAULT]
core_plugin = ml2
service_plugins =
transport_url = rabbit://openstack:[REDACTED]@kenkalaatikko
auth_strategy = keystone
#
```

Salasanana "Rabbit_user"-salasana, jälleen sotkettuna kuvassa. Varmistetaan vielä, että kaikki muut valinnat ovat kommentoituna lisäämällä #-merkki rivin ensimmäiseksi, jos se puuttuu.

Kohtaan **[keystone_authtoken]** lisätään tieto keystone -palvelimesta ja sen käyttämistä porteista. Lisätään myös OpenStack ympäristön tiedot sekä OpenStack ympäristöön tehdyn käyttäjän **neutron** nimi ja salasana

```
GNU nano 2.9.6 /etc/neutron/neutron.conf

[keystone_authtoken]
www_authenticate_uri = http://kenkalaatikko:5000
auth_url = http://kenkalaatikko:5000
memcached_servers = kenkalaatikko:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = neutron
password = [REDACTED]
#
```

Salasana siis "Neutron_user", sensuroitu sotkemalla.

Kohtaan **[nova]** lisätään nova -palvelimen osoite ja portti, lisäksi tarvittavat OpenStack projektin tiedot sekä OpenStack ympäristöön tehdyn käyttäjän **nova** nimi ja salasana

```
GNU nano 2.9.6 /etc/neutron/neutron.conf
# Config Section from which to load plugin specific options (string value)
#auth_section = <None>

[nova]

auth_url = http://kenkalaatikko:5000
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = nova
password = ██████████
```

Salasana "Nova_user" ja jälleen sotkettu.

Määritellään vielä **[DEFAULT]** kohdassa, että Nova -palvelulle ilmoitetaan, jos verkon topologiassa tapahtuu muutoksia

```
GNU nano 2.9.6 /etc/neutron/neutron.conf

[DEFAULT]

notify_nova_on_port_status_changes = true
notify_nova_on_port_data_changes = true
```

Lisätään vielä kohtaan **[oslo_concurrency]** rivi

```
GNU nano 2.9.6 /etc/neutron/neutron.conf

[oslo_concurrency]

lock_path = /var/lib/neutron/tmp
```

Nyt tiedosto voidaan tallentaa ja sulkea.

Siirrytään editoimaan tiedostoa *"/etc/neutron/plugins/ml2/ml2_conf.ini"*. Kohdassa **[ml2]** määritellään käytettäväksi flat, vlan, gre sekä vxlan -verkkoja, määritellään liittyvien verkkojen tyyppiä flat sekä määritellään käytettäväksi mekaniikaksi OpenVSwitch

```
GNU nano 2.9.6 /etc/neutron/plugins/ml2/ml2_conf.ini

[ml2]

type_drivers = flat,vlan,gre,vxlan
tenant_network_types = flat
mechanism_drivers = openvswitch
```

Kohta **[ml2_type_flat]**

```
GNU nano 2.9.6 /etc/neutron/plugins/ml2/ml2_conf.ini

[ml2_type_flat]

flat_networks = *

#
```

Kohdassa **[securitygroup]** sallitaan "ipset" tehostamaan turvallisuuskäytäntöjä, asetetaan valvonta käyttäjäryhmät käyttöön sekä määritellään palomuuuri ajuri

```
GNU nano 2.9.6 /etc/neutron/plugins/ml2/ml2_conf.ini

[securitygroup]

enable_security_group = true
enable_ipset = true
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
[]
#
```

Tämän jälkeen tiedosto voidaan tallentaa ja sulkea.

Määritellään seuraavaksi Linuxin silta agentti. Editoidaan tiedostoa `"/etc/neutron/plugins/ml2/linuxbridge_agent.ini"`. Lisätään kohtaan **[ovs]** viittaus myöhemmässä vaiheessa luotavaan virtuaaliseen verkkosiltaan `br-eth0`

```
GNU nano 2.9.6 /etc/neutron/plugins/ml2/openvswitch_agent.ini

[ovs]

bridge_mappings = external:br-eth0
```

Määritellään vielä DHCP-agentti, joka vastaa virtuaalikoneiden verkoista. Avataan tiedosto `"/etc/neutron/dhcp_agent.ini"` ja editoidaan kohtaa **[DEFAULT]**

```
GNU nano 2.9.6 /etc/neutron/dhcp_agent.ini

[DEFAULT]

interface_driver = ovs
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
enable_isolated_metadata = true
```

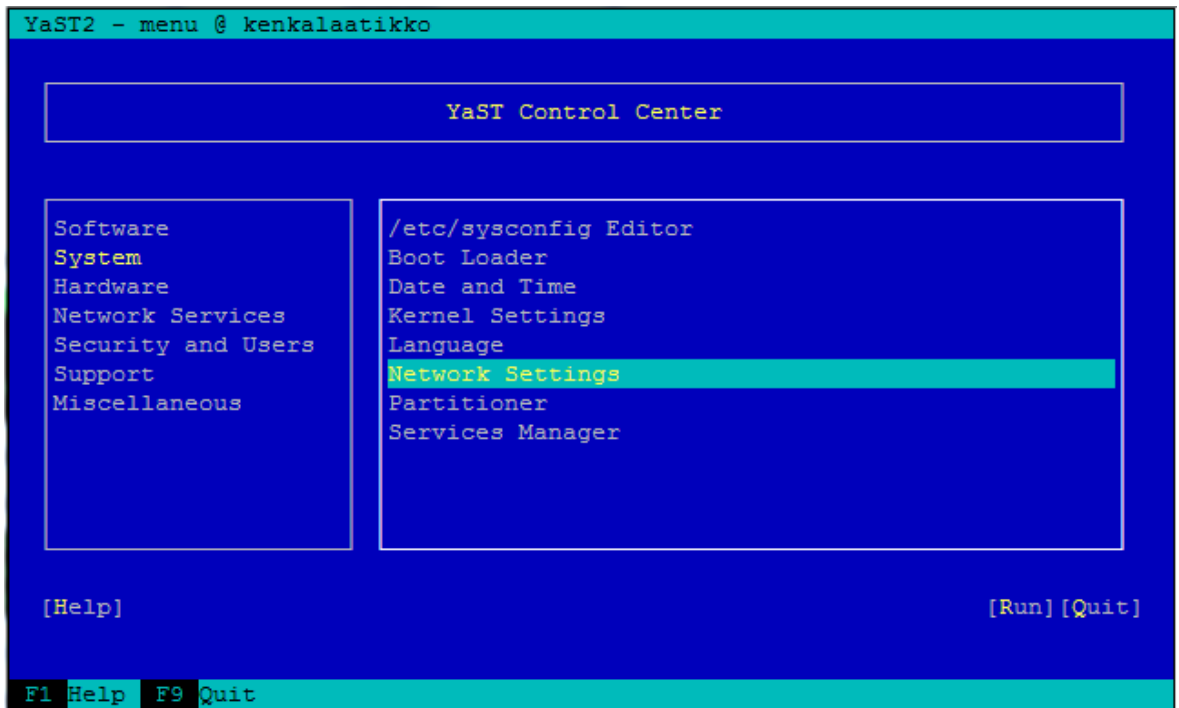
Luodaan lopuksi verkkosilta `br-eth0` ja ohjataan fyysisen verkkokortin `eth0` liikenne kulkemaan sen kautta

```
ont@kenkalaatikko:~> sudo ovs-vsctl add-br br-eth0
ont@kenkalaatikko:~> sudo ovs-vsctl add-port br-eth0 eth0
ont@kenkalaatikko:~> █
```

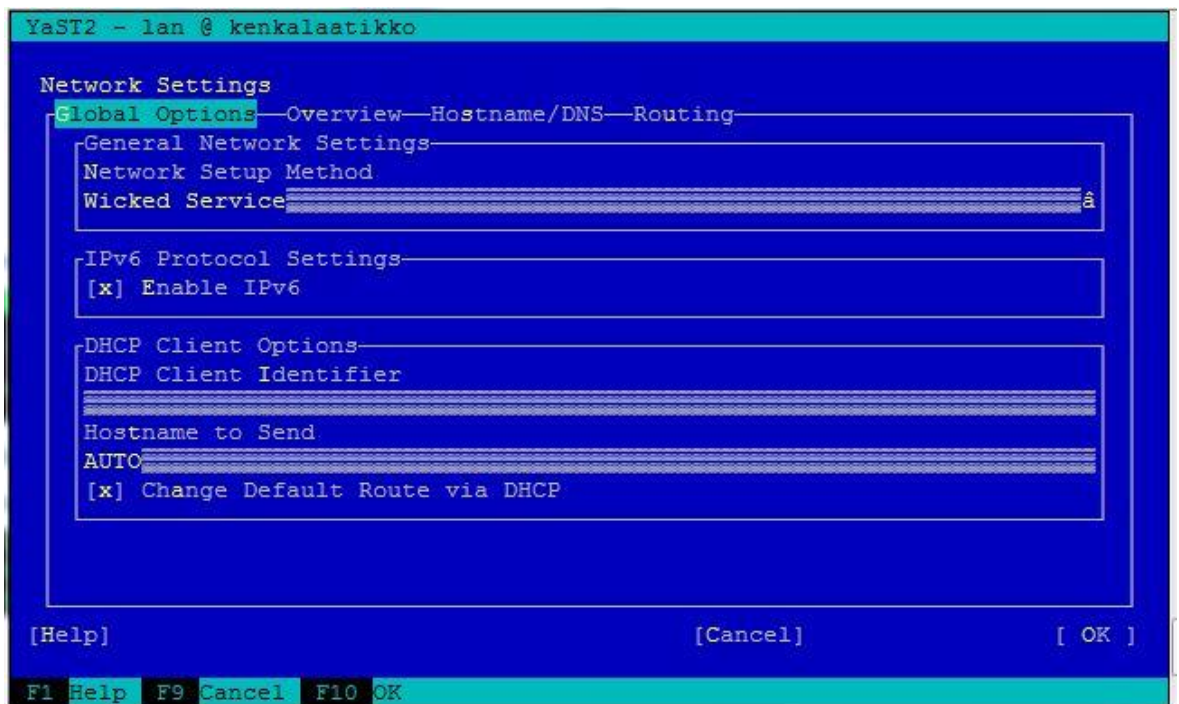
Jotta OpenVirtualSwitch toimisi kunnolla täytyy järjestelmäasetuksissa verkonhallinta palveluksi olla valittuna `Wicked`. Varmistetaan tämä. Avataan siis YaST komennolla

```
ont@kenkalaatikko:~> sudo yast
[sudo] password for root:
```

Ja tämän jälkeen siirrytään kohtaan *System > Network Settings*



Valitaan *Global Options* valikosta *Network Setup Method* kohtaan *Wicked Service*.



Tarkistetaan samalla *Overview* valikosta, että verkkosillan luonti on onnistunut

```
YaST2 - lan @ kenkalaatikko

Network Settings
Global Options Overview Hostname/DNS Routing

Name | IP Address | Device | Note
-----|-----|-----|-----
82579V Gigabit Network Connection | 192.168.10.115 | eth0 |
Ethernet Network Card | 192.168.10.110 | br-eth0 |

82579V Gigabit Network Connection
MAC : e0:69:95:eb:55:28
BusID : 0000:00:19.0

[Add] [Edit] [Delete]

[Help] [Cancel] [ OK ]

F1 Help F3 Add F4 Edit F5 Delete F9 Cancel F10 OK
```

Luonnin voidaan todeta onnistuneen, listalla näkyy fyysinen kortti laitteena *eth0* sekä luotu silta laitteena *br-eth0*. (Lisä huomiona, mainittakoon, että johtuen jostain *Wicked* sekä *OpenVirtualSwitch* palveluiden yhtäaikaisesta halusta käynnistyä ennen toista, täytyy virtuaalinen verkkokortti käynnistää manuaalisesti fyysisen koneen uudelleen käynnistytksen jälkeen. Tämä tapahtuu komennoin

```
ont@kenkalaatikko:~> sudo /usr/share/openvswitch/scripts/ovsctl start
redirecting to systemctl start .service
/etc/openvswitch/conf.db does not exist
Creating empty database /etc/openvswitch/conf.db
Starting ovsdb-server
system ID not configured, please use --system-id
Configuring Open vSwitch system IDs
Inserting openvswitch module
Starting ovs-vswitchd
Enabling remote OVSDB managers
warning
done
done
failed
done
done
done
done
```

```
ont@kenkalaatikko:~> sudo wicked ifup br-eth0
[sudo] password for root:
br-eth0 up
ont@kenkalaatikko:~> █
```

Tämä siis tarkoittaa, ettei palvelin ole tavoitettavissa etäyhteyksin, ennen kuin edellä mainittu operaatio on suoritettu.)

Seuraavaksi siirrytään laskentakoneisiin *vadelma*, *herukka* ja *mustikka*. Editoidaan jokaisessa koneessa tiedostoa `/etc/nova/nova.conf` kohdassa **[neutron]**

```
GNU nano 3.2 /etc/nova/nova.conf

[neutron]
#
# Configuration options for neutron (network connectivity as a service).

auth_url = http://kenkalaatikko:5000
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = ██████████
```

Salasanana "Neutron_user", jälleen sotkettuna- Kun tiedosto on tallennettu kaikkiin laskenta koneisiin, voidaan palata *kenkalaatikko*:on ja viimeistellä asennus ensin poistamalla käytöstä AppArmor ohjelman DNS maskeeraus ominaisuus. Tämä ominaisuus haittaa muun muassa DHCP agentin toimintaa. Ominaisuuden poisto tapahtuu komennolla

```
ont@kenkalaatikko:~> sudo ln -s /etc/apparmor.d/usr.sbin.dnsmasq /etc/apparmor.d/disable
ont@kenkalaatikko:~> █
```

Tämän jälkeen uudelleen käynnistetään Laskenta-palvelun rajapintapalvelu

```
ont@kenkalaatikko:~> sudo systemctl restart openstack-nova-api
ont@kenkalaatikko:~> █
```

Asetetaan Verkko palvelut käynnistymään koneen käynnistyksen yhteydessä

```
ont@kenkalaatikko:~> sudo systemctl enable openstack-neutron.service openstack-neutron-linuxbridge-agent.service openstack-neutron-openvswitch-agent.service openstack-neutron-dhcp-agent.service openstack-neutron-metadata-agent.service
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-neutron-server.service → /usr/lib/systemd/system/openstack-neutron-server.service.
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-neutron-linuxbridge-agent.service → /usr/lib/systemd/system/openstack-neutron-linuxbridge-agent.service.
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-neutron-openvswitch-agent.service → /usr/lib/systemd/system/openstack-neutron-openvswitch-agent.service.
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-neutron-dhcp-agent.service → /usr/lib/systemd/system/openstack-neutron-dhcp-agent.service.
Created symlink /etc/systemd/system/multi-user.target.wants/openstack-neutron-metadata-agent.service → /usr/lib/systemd/system/openstack-neutron-metadata-agent.service.
ont@kenkalaatikko:~> █
```

Ja näin Verkko palvelu on toiminnassa.

5.5.3 Virtuaaliverkkojen luonti

Luodaan vielä lopuksi virtuaaliverkko sekä sille aliverkotus säännöt. Verkonluonti hoituu komennolla

```
ont@kenkalaatikko:~> openstack network create --share --external \  
> --provider-physical-network provider \  
> --provider-network-type flat provider  
+-----+  
+-----+  
| Field | Value |  
+-----+  
| admin_state_up | UP |  
+-----+  
| availability_zone_hints | |  
+-----+
```

Ja aliverkotussäännöt saadaan komennolla

```
ont@kenkalaatikko:~> openstack subnet create --network provider --allocation-pool start=192.168.10.240  
,end=192.168.10.254 --dns-nameserver 8.8.4.4 --gateway 192.168.10.1 --subnet-range 192.168.10.0/24 pro  
vider  
^[[D^[[C+-----+  
+-----+  
| Field | Value |  
+-----+  
| allocation_pools | 192.168.10.240-192.168.10.254 |  
| cidr | 192.168.10.0/24 |  
| created_at | 2020-04-29T17:29:45Z |  
| description | |  
+-----+
```

5.6 Hallintapaneeli

Hallintapaneeli -palvelu mahdollistaa OpenStack palveluiden keskitetyn käytön selainpohjaisella, graafisella, käyttäjä ystävällisellä hallintapaneelilla. Palvelun nimi on Horizon.

5.6.1 Pakettien asennus ja konfigurointi

Asennetaan Hallintapaneeli -palvelun paketit komennolla

```
ont@kenkalaatikko:~> sudo zypper in openstack-dashboard
[sudo] password for root:
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following 53 NEW packages are going to be installed:
 libmaxminddb0 openstack-dashboard python3-argon2-cffi python3-Brotli python3-calmjs
 python3-calmjs.parse python3-calmjs.types python3-csscompressor python3-Django
 python3-django-appconf python3-django-babel python3-django-compressor python3-django-debreach
 python3-django-pyscss python3-geoip2 python3-horizon python3-maxminddb python3-openstack_auth
```

Asennuksen jälkeen kopioidaan Hallintapaneeli verkkosivun asetukset Apache palvelun asetus kansioon, komennolla

```
ont@kenkalaatikko:~> sudo cp /etc/apache2/conf.d/openstack-dashboard.conf.sample /etc/apache2/conf.d/o
penstack-dashboard.conf
ont@kenkalaatikko:~> █
```

Sallitaan Apachen URL-osoitteen uudelleenkirjoitus liitännäinen

```
ont@kenkalaatikko:~> sudo a2enmod rewrite
ont@kenkalaatikko:~> █
```

Tämän liitännäisen avulla verkkosivu -palvelu ohjaa käyttäjän osoitteesta `http://kenkalaatikko` suoraan Hallintapaneelin kirjautumissivulle `http://kenkalaatikko/auth/login/?next=`. Editoidaan tiedostoa `/srv/www/openstack-dashboard/openstack_dashboard/local/local_settings.py`. Aloitetaan kertomalla, mikä on OpenStack -palveluiden kontrolleri noodin isäntänimi

```
GNU nano 2.9.6 /srv/www/openstack-dashboard/openstack_dashboard/local/local_settings.py
OPENSTACK_HOST = "kenkalaatikko"
```

Seuraavaksi määritellään sallitut URL- osoitteet. Koska kyseessä on suljettu testi ympäristö, käytetään tässä kohtaa valintaa

```
GNU nano 2.9.6 /srv/www/openstack-dashboard/openstack_dashboard/local/local_settings.py
ALLOWED_HOSTS = ['*', ]
```

joka sallii kaikki osoitteet. Huomioitavaa on, ettei tämä ole kovin turvallista tietoturvan kannalta. Määritellään liitäntä alussa luotuun Memcached -palveluun

```
GNU nano 2.9.6 /srv/www/openstack-dashboard/openstack_dashboard/local/local_settings.py
# as shown below:
SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
```

```
GNU nano 2.9.6 /srv/www/openstack-dashboard/openstack_dashboard/local/local_settings.py
# https://docs.djangoproject.com/en/1.11/topics/http/sessions/.
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': 'kenkalaatikko:11211',
    }
}
```

Seuraavaksi määritellään rajapinta versiot, sekä hallintapaneelin kautta luotavien käyttäjien oletus toimintaympäristöksi "Default" ja oletus rooliksi "user"

```
GNU nano 2.9.6 /srv/www/openstack-dashboard/openstack_dashboard/local/local_settings.py
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
OPENSTACK_API_VERSIONS = {
    "identity": 3,
    "image": 2,
    "volume": 3,
}
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
```

Poistetaan käytöstä kaikki OSI-mallin tason 3 toiminnot

```
GNU nano 2.9.6 /srv/www/openstack-dashboard/openstack_dashboard/local/local_settings.py
# The OPENSTACK_NEUTRON_NETWORK settings can be used to enable optional
# services provided by neutron. Options currently available are load
# balancer service, security groups, quotas, VPN service.
OPENSTACK_NEUTRON_NETWORK = {
    'enable_quotas': False,
    'enable_router': False,
    'enable_distributed_router': False,
    'enable_ha_router': False,
    'enable_fip_topology_check': False,
    'enable_lb': False,
    'enable_firewall': False,
    'enable_vpn': False,
```

Ja vielä lopuksi määritellään aikavyöhyke

```
/srv/www/openstack-dashboard/openstack_dashboard/local/local_settings.py
# The timezone of the server. This should correspond with the timezone
# of your entire OpenStack installation, and hopefully be in UTC.
TIME_ZONE = "Europe/Helsinki"
```

Nyt tiedosto voidaan sulkea ja asennus päättää käynnistämällä Apache- ja Memcache-palvelut uudelleen uusin asetuksin

```
ont@kenkalaatikko:~> sudo systemctl restart apache.service memcached.service
[sudo] password for root:
ont@kenkalaatikko:~>
```

Nyt Hallintapaneeli on toiminnassa osoitteessa "http://kenkalaatikko" (tai "http://192.168.10.110"). Paitsi, että sivusto antaa HTTP virheen 404, joka tarkoittaa, ettei sivusto ole saavutettavissa tai ettei sitä ole olemassa. Virheen korjausta varten avataan

editoitavaksi tiedosto `/etc/apache2/conf.d/openstack-dashboard.conf`

```
GNU nano 2.9.6 /etc/apache2/conf.d/openstack-dashboard.conf
#
# OpenStack Horizon (Dashboard) Apache2 example configuration.
#
# Required Apache2 modules:
# - mod_rewrite (If you use the port 80 -> 443 rewrite rule below)
# - mod_ssl (If you enable the HTTPS vhost)
#

<VirtualHost *:80>
    ServerName openstack-dashboard.kenkalaatikko
    ServerAdmin webmaster@openstack-dashboard.example.com

    ErrorLog /var/log/apache2/openstack-dashboard-error_log
    TransferLog /var/log/apache2/openstack-dashboard-access_log

    WSGIScriptAlias / /var/lib/openstack-dashboard/openstack_dashboard/wsgi.py
    WSGIDaemonProcess horizon user=wwwrun group=www processes=3 threads=10
    WSGIApplicationGroup %{GLOBAL}
    WSGIProcessGroup horizon
    SetEnv APACHE_RUN_USER wwwrun
    SetEnv APACHE_RUN_GROUP www

    DocumentRoot /var/lib/openstack-dashboard/
    Alias /media /var/lib/openstack-dashboard/media
    Alias /static /var/lib/openstack-dashboard/static
```

Vaihdetaan kohtaan "ServerName" palvelun osoite, eli `openstack-dashboard.kenkalaatikko`. Lisäksi tiedostoa tarkastelemalla huomataan, että kaikki asetukset viittaavat kansioon `/var/lib/openstack-dashboard` vaikka ohjeessa mainitaan, että Ohjauspaneelin tiedostojen pitäisi olla kansiossa `/srv/www/openstack-dashboard`. Ongelmaan on kaksi ratkaisua; kopioidaan tiedostot kansioista `/srv/www/openstack-dashboard` kansioon `/var/lib/openstack-dashboard` tai muutetaan kansioviittaus asetustiedostossa. Valitsin kopiomisvaihtoehdon, olettaen, että kansioon `/var/lib/openstack-dahboard` viitataan myös muissa asetustiedostoissa, joihin en ole tekemässä muutoksia. Kopiointi tapahtuu komenolla

```
ont@kenkalaatikko:~> sudo cp -r /srv/www/openstack-dashboard/ /var/lib/
[sudo] password for root: █
```

Käynnistetään Apache- ja Memcached-palvelut jälleen uudelleen

```
ont@kenkalaatikko:~> sudo systemctl restart apache.service memcached.service
[sudo] password for root:
ont@kenkalaatikko:~> █
```

Nyt verkkosivu antaa http virheen 403, eli ei oikeutta tarkastella pyydettyä sisältöä. Tämä johtuu siitä, että kansion `var/lib/openstack-dashboard` omistaja on käyttäjä "root". Aikaisemmin editoidusta tiedostosta `/etc/apache2/conf.d/openstack-dashboard.conf` huomataan, että Apache-palvelu käsittelee tiedostoja käyttäjän "wwwrun" oikeuksin. Vaihdetaan siis kansion ja sen kaikkien alikansioden ja tiedostojen omistajaksi käyttäjä "wwwrun" ja

ryhmäksi "www", sekä varmuudenvuoksi annetaan kaikille lukuoikeudet kaikille komen-
noin

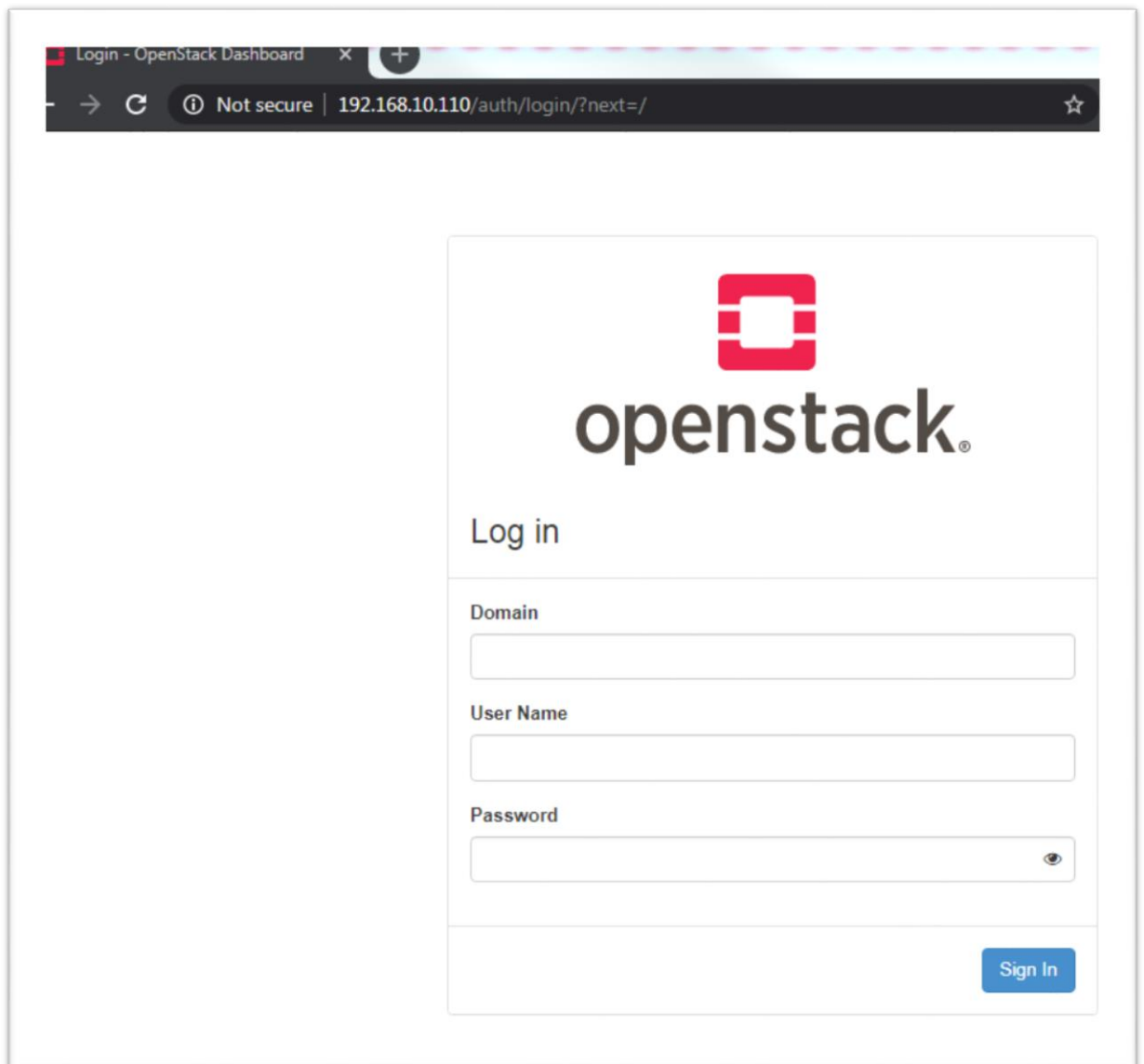
```
ont@kenkalaatikko:~> sudo chown -R wwwrun:www /var/lib/openstack-dashboard/  
[sudo] password for root: [ ]
```

```
ont@kenkalaatikko:~> sudo chmod -R a+r /var/lib/openstack-dashboard/  
[sudo] password for root: [ ]
```

Käynnistetään jälleen Apache- ja Memcached-palvelut uudelleen

```
ont@kenkalaatikko:~> sudo systemctl restart apache.service memcached.service  
[sudo] password for root:  
ont@kenkalaatikko:~> █
```

Ja sivusto toimii



6 Ensimmäisen istunnon käynnistys

Nyt kun järjestelmä on toiminnassa halutulla asteella, on aika käynnistää ensimmäinen istunto. Aliotetaan luomalla *flavor*, tiedosto, jolla määritellään virtuaalisen koneen perusarvot, kuten RAM-muisti ja prosessoriydinten määrä. Koska OpenStackin oletus *flavor* käyttää 512Mt muistia, luodaan **m1.nano** *flavor*, joka käyttää vain 64Mt muistia per istunto, sekä yhtä prosessoriydintä

```
ont@kenkalaatikko:~> openstack flavor create --id 0 --vcpus 1 --ram 64 --disk 1 m1.nano
+-----+-----+
| Field | Value |
+-----+-----+
| OS-FLV-DISABLED:disabled | False |
| OS-FLV-EXT-DATA:ephemeral | 0 |
| disk | 1 |
| id | 0 |
| name | m1.nano |
| os-flavor-access:is_public | True |
| properties | |
| ram | 64 |
| rxtx_factor | 1.0 |
| swap | |
| vcpus | 1 |
+-----+-----+
ont@kenkalaatikko:~> █
```

Koska useimmat pilvilevykuvat tukevat julkisen avaimen tunnistautumista luodaan seuraavaksi avain pari. Aloitetaan lataamalla "myuser" käyttäjän ympäristö muuttujat ja generoidaan avaimet

```
ont@kenkalaatikko:~> . myuser-openrc
ont@kenkalaatikko:~> ssh-keygen -q -N ""
Enter file in which to save the key (/home/ont/.ssh/id_rsa):
ont@kenkalaatikko:~> █
```

Lisätään avain OpenStack ympäristöön nimellä "mykey"

```
ont@kenkalaatikko:~> openstack keypair create --public-key ~/.ssh/id_rsa.pub mykey
+-----+-----+
| Field | Value |
+-----+-----+
| fingerprint | 57:98:c5:ac:7e:13:c5:94:9d:90:c7:f3:94:67:ed:3e |
| name | mykey |
| user_id | 5a4a1b298b8645d49d5b7d5a327b5be3 |
+-----+-----+
ont@kenkalaatikko:~> █
```

Varmistetaan, että avainpari tuli lisätyksi oikein

```
ont@kenkalaatikko:~> openstack keypair list
+-----+-----+
| Name | Fingerprint |
+-----+-----+
| mykey | 57:98:c5:ac:7e:13:c5:94:9d:90:c7:f3:94:67:ed:3e |
+-----+-----+
ont@kenkalaatikko:~> █
```

Sama avain, joten voidaan jatkaa.

Seuraavaksi luodaan turvallisuus säännöt, joilla sallitaan virtuaalikoneiden ping'aus sekä SSH-yhteyden muodostus koneisiin. Koska **default** määrittys koskee nimensä mukaisesti oletuksena kaikki istuntoja, lisätään säännöt siihen. Sallitaan ICMP-protokolla (ping) komennolla

```
ont@kenkalaatikko:~> openstack security group rule create --proto icmp default
+-----+
| Field          | Value                                     |
+-----+-----+
| created_at     | 2020-05-02T16:30:05Z                    |
| description    |                                           |
| direction     | ingress                                  |
| ether_type     | IPv4                                     |
| id             | 35c7e392-7510-486e-94a7-e60a12b9b8f7   |
+-----+-----+
```

Sekä SSH-yhteys komennolla

```
ont@kenkalaatikko:~> openstack security group rule create --proto tcp --dst-port 22 default
+-----+
| Field          | Value                                     |
+-----+-----+
| created_at     | 2020-05-02T16:32:34Z                    |
| description    |                                           |
| direction     | ingress                                  |
| ether_type     | IPv4                                     |
| id             | 6377dbff-2deb-4820-a450-f74ae826b061   |
+-----+-----+
```

Nyt kun kaikki tarvittavat määrittymiset on tehty, palautetaan muistiin istunnon tarvitsemat tiedot. Nämä ovat *flavor*, levykuva, verkko sekä turvallisuus säännöt. Aloitetaan lataamalla **myuser** käyttäjän ympäristö muuttujat

```
ont@kenkalaatikko:~> . myuser-openrc
ont@kenkalaatikko:~> █
```

Listataan käytettävissä olevat *flavor*:it

```
ont@kenkalaatikko:~> openstack flavor list
+-----+-----+-----+-----+-----+-----+-----+
| ID | Name      | RAM | Disk | Ephemeral | VCPUs | Is Public |
+-----+-----+-----+-----+-----+-----+-----+
| 0  | m1.nano  | 64  | 1    | 0          | 1     | True      |
+-----+-----+-----+-----+-----+-----+-----+
ont@kenkalaatikko:~> █
```

Listataan käytettävissä olevat levykuvat

```
ont@kenkalaatikko:~> openstack image list
+-----+-----+-----+
| ID | Name | Status |
+-----+-----+-----+
| 55e63437-2130-446a-a6c9-3cd23a4b9009 | cirros | active |
+-----+-----+-----+
ont@kenkalaatikko:~> █
```

Listataan käytettävissä olevat verkot

```
ont@kenkalaatikko:~> openstack network list
+-----+-----+-----+
| ID | Name | Subnets |
+-----+-----+-----+
| 5816b026-48c1-4df0-ad96-1225870eaef0 | provider | b8301b05-8951-4ce3-8489-a35d3dece46e |
+-----+-----+-----+
ont@kenkalaatikko:~> █
```

Ja vielä lopuksi lista turvallisuus säännöistä

```
ont@kenkalaatikko:~> openstack security group list
+-----+-----+-----+-----+-----+
| ID | Name | Description | Project | Tags |
+-----+-----+-----+-----+-----+
| 4682b0cc-d7ff-4ba3-94af-1754dedaac67 | default | Default security group | 86723139bf114a40a10351a314dbb197 | [] |
+-----+-----+-----+-----+-----+
ont@kenkalaatikko:~> █
```

Nyt voidaan luoda ensimmäinen istunto

```
ont@kenkalaatikko:~> openstack server create --flavor m1.nano --image cirros --security-group default
--key-name mykey provider-instance
+-----+-----+
| Field | Value |
+-----+-----+
| OS-DCF:diskConfig | MANUAL |
| OS-EXT-AZ:availability_zone | |
| OS-EXT-STS:power_state | NOSTATE |
| OS-EXT-STS:task_state | scheduling |
| OS-EXT-STS:vm_state | building |
| OS-SRV-USG:launched_at | None |
| OS-SRV-USG:terminated_at | None |
| accessIPv4 | |
| accessIPv6 | |
| addresses | |
+-----+-----+
```

Luonnissa sekä käynnistyksessä kestää hetki. Tilanne voidaan tarkastaa ajamalla komento

```
ont@kenkalaatikko:~> openstack server list
+-----+-----+-----+-----+-----+
| ID | Name | Status | Networks | Image | Flavor |
+-----+-----+-----+-----+-----+
| 45d65995-221a-4852-b935-8f94abb844bc | test | ERROR | | cirros | m1.nano |
| 8442dc10-4241-4a45-9674-f89094a2678f | provider-instance | ERROR | | cirros | m1.nano |
+-----+-----+-----+-----+-----+
ont@kenkalaatikko:~> █
```

Yllä juuri luotu istunto nimellä "provider-instance" sekä testimielessä luotu istunto "test". Listata huomataan myös, että molemmat istunnot ovat "ERROR" tilassa, jossain on siis vika. Aloitetaan selvitys.

6.1 Vian selvitys

Ensimmäiseksi täytynee sanoa, ettei tilanne yllätä Raspberry Pi 2:t eivät ole kovin tehokkaita ja osana projektia lähinnä mielenkiinnosta. Tilannetta selvittäessä lokeja lukemalla ja virhe koodeja googlaamalla sain selville muutaman asian:

1. Raspberry Pi 2:n prosessori ei ole suunniteltu tavalle, jolla KVM/Qemu käsittelee virtuaalikoneita (flexVDI. 2015).
2. Ominaisuus on olemassa, mutta se pitää erikseen kytkeä päälle Rasbianin kernelissä (flexVDI. 2015).
3. Rasbianin pakettivarastosa oleva paketti Qemu:lle on vanha, se pitää päivittää manuaalisesti (Rasberrypi forums. 2019).

6.2 Vian korjaus

Edessä oli kaksi vaihetta, joilla yrittää korjata tilanne, kernelin päivitys laskentakoneisiin ja Qemu ohjelmiston päivitys samoihin koneisiin. Päätin ottaa yhden koneista, *herukka*, testi koneeksi, johon yritän ensin tehdä muutokset ja niiden auttaessa tilanteeseen, kopioida muutokset muihin laskentakoneisiin.

6.2.1 Kernelin päivitys

Aloitetaan asentamalla joukko tarvittavia paketteja, joita käytetään, kun käännetään ohjelmistoja suoraan lähdekoodista

```
sudo apt-get install -y g++-arm-linux-gnueabi flex bison libglib2.0-dev:armhf
sudo apt-get install libpixman-1-dev
```

```
sudo apt install git bc bison flex libssl-dev make
```

Kuvakaappaukset poikkeuksellisesti otettu komentohistoriasta. Seuraavaksi ladataan vakaan kernelin tarvitsemat tiedostot Raspberry yhteisön Github-palvelusta komennolla

```
pi@mustikka:~ $ git clone --depth=1 https://github.com/raspberrypi/linux
Cloning into 'linux'...
remote: Enumerating objects: 66216, done.
remote: Counting objects: 100% (66216/66216), done.
remote: Compressing objects: 100% (61591/61591), done.
remote: Total 66216 (delta 5170), reused 17424 (delta 3699), pack-reused 0
Receiving objects: 100% (66216/66216), 177.97 MiB | 2.64 MiB/s, done.
Resolving deltas: 100% (5170/5170), done.
Checking out files: 100% (62373/62373), done.
pi@mustikka:~ $
```

Siirrytään lataus kansioon komennolla

```
pi@mustikka:~ $ cd linux/
pi@mustikka:~/linux $
```

Tarkistetaan, että kaikki tiedostot tuli ladattua

```
git checkout rpi-4.19.y
```

Kuvakaappaus jälleen historiasta otettuna. Luodaan tilapäinen työskentelykansio

”/tmp/rpi2”, jonne päivitetty kernel luodaan

```
mkdir /tmp/rpi2
```

Seuraavaksi luodaan asetustiedosto, jota lukemalla kääntäjä selvittää mitä luotavalta kerneliltä halutaan

```
make O=/tmp/rpi2 menuconfig
```

Komennon annettua aukeaa uusi ikkuna, jossa voidaan kytkeä päälle tai pois haluttuja kernel optioita. Asetetaan seuraavat ominaisuudet päälle

```
.config - Linux/arm 4.19.120 Kernel Configuration
> System Type
Linux/arm 4.19.120 Kernel Configuration
x Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
x Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
x features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] x
x excluded <M> module < > module capable
x
x (-) Sigma Designs Tango4 (SMP87xx)
x [ ] NVIDIA Tegra ----
x [ ] Socionext UniPhier SoCs
x [ ] ST-Ericsson U8500 Series ----
x [ ] ARM Ltd. Versatile Express family ----
x [ ] WonderMedia WM8850
x [ ] ZTE ZX family ----
x [ ] Xilinx Zynq ARM Cortex A9 Platform
x *** Processor Type ***
x *** Processor Features ***
x [*] Support for the Large Physical Address Extension
x [*] Support Thumb user binaries
x [ ] Enable ThumbEE CPU extension
x
x (+)
x
x <Select> < Exit > < Help > < Save > < Load >
```

```
.config - Linux/arm 4.19.120 Kernel Configuration
Linux/arm 4.19.120 Kernel Configuration
x Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
x Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
x features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] x
x excluded <M> module < > module capable
x
x *** Compiler: arm-linux-gnueabi-hf-gcc (Raspbian 8.3.0-6+rpi1) 8.3.0 ***
x
x General setup --->
x [*] Patch physical to virtual translations at runtime
x System Type --->
x Bus support --->
x Kernel Features --->
x Boot options --->
x CPU Power Management --->
x Floating point emulation --->
x Power management options --->
x Firmware Drivers --->
x [*] ARM Accelerated Cryptographic Algorithms --->
x [ ] Virtualization ----
x
x (+)
x
x <Select> < Exit > < Help > < Save > < Load >
```

```

.config - Linux/arm 4.19.120 Kernel Configuration
> General setup
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
excluded <M> module < > module capable
[*] Enable access to .config through /proc/config.gz
(17) Kernel log buffer size (16 => 64KB, 17 => 128KB)
(12) CPU kernel log buffer size contribution (13 => 8 KB, 17 => 128KB)
(13) Temporary per-CPU printk log buffer size (12 => 4KB, 13 => 8KB)
[*] Control Group support --->
[*] Namespaces support --->
[ ] Checkpoint/restore support
[*] Automatic process group scheduling
[ ] Enable deprecated sysfs features to support old userspace tools
[*] Kernel->user space relay support (formerly relayfs)
[*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
( ) Initramfs source file(s)
[*] Support initial ramdisk/ramfs compressed using gzip
<Select> < Exit > < Help > < Save > < Load >

```

```

.config - Linux/arm 4.19.120 Kernel Configuration
> Boot options
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
excluded <M> module < > module capable
[*] Flattened Device Tree support
[ ] Support for the traditional ATAGS boot data passing
(0x0) Compressed ROM boot loader base address
(0x0) Compressed ROM boot loader BSS address
[*] Use appended device tree blob to zImage (EXPERIMENTAL)
[*] Supplement the appended DTB with traditional ATAG information
Kernel command line type (Use bootloader kernel arguments if available)
(console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 root=/dev/mmcblk0p2 rootfstype=ext4)
[ ] Build kdump crash kernel (EXPERIMENTAL)
[*] Auto calculation of the decompressed kernel image address
[ ] UEFI runtime support
<Select> < Exit > < Help > < Save > < Load >

```

Viimeiseksi vielä Virtualisoinnin alta sallitaan KVM

```

.config - Linux/arm 4.19.120 Kernel Configuration
Linux/arm 4.19.120 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
excluded <M> module < > module capable
General setup --->
[*] Patch physical to virtual translations at runtime
System Type --->
Bus support --->
Kernel Features --->
Boot options --->
CPU Power Management --->
Floating point emulation --->
Power management options --->
Firmware Drivers --->
[*] ARM Accelerated Cryptographic Algorithms --->
[*] Virtualization --->
General architecture-dependent options --->
<Select> < Exit > < Help > < Save > < Load >

```

```
.config - Linux/arm 4.19.120 Kernel Configuration
> Virtualization
--- Virtualization
[*] Kernel-based Virtual Machine (KVM) support
    < > Host kernel accelerator for virtio net (NEW)
    [ ] Cross-endian support for vhost (NEW)
...
<Select> < Exit > < Help > < Save > < Load >
```

Tämän jälkeen tallennetaan asetukset ".config"-nimiseen tiedostoon ja poistutaan asetuk-
sien hallinnasta. Tämän jälkeen voidaan luoda uusi kernel komennolla

```
make O=/tmp/rpi2 all -j 4
```

Komennossa "O=" määrittää valmiin kernelin tallennuskansion, "all" kertoo, että tehdään
kaikki ja lopun lippu "-j" arvolla 4 kertoo kääntäjälle, että käytetään kaikkia neljää ydintä.
Kun kernel on valmis, kopioidaan vanha talteen varmuudenvuoksi

```
sudo mv /boot/kernel7.img /boot/kernel7.img,bak
```

(Ei huomioida kirjoitusvirhettä, sen pitäisi olla piste, ei pilkku.) Kääntäjä luo uuden kernelin
nimellä "zImage", nimetään se uudelleen "kernel7.img" ja siirretään /boot/-kansioon

```
mv /tmp/rpi2/arch/arm/boot/zImage /tmp/rpi2/kernel7.img
sudo cp /tmp/rpi2/kernel7.img /boot/
```

Lopuksi vielä editoidaan tiedostoa "/boot/cmdline.txt" ja lisätään sen loppuun sana
"isolcpus=3". Tämä kertoo kernelille, että neljäs ydin on eristetty ja näin ollen varattu virtu-
alisoinnille. Käynnistetään kone uudestaan ja ajetaan komennot

```
pi@herukka:~$ dmesg | grep HYP
[ 0.009606] CPU: All CPU(s) started in HYP mode.
pi@herukka:~$ dmesg | grep Virtu*
[ 0.000000] Virtual kernel memory layout:
[ 0.009611] CPU: Virtualization extensions available.
pi@herukka:~$
```

Ja näin voidaan todeta, että uusi kernel on käytössä ja virtualisointi on päällä.

6.2.2 Qemun päivitys

Kuten kernel tulee myös Qemu kääntää suoraan lähdekoodista. Aloitetaan lataamalla tar-
vittavat tiedostot Qemun omasta Git-varastosta

```
git clone git://git.qemu-project.org/qemu.git
```

Siirrytään latauskansioon ja ajetaan seuraavat komennot järjestyksessä

```
./configure
make
sudo make install
```

Tiedosto `./configure` lataa kääntäjän tarvitsemat asetukset, `make` käynnistää kääntäjän ja `sudo make install` asentaa käännettyt paketit järjestelmään.

6.3 Nyt se sitten toimii?

Ei. Jostain syystä koneiden virtualisointi ei edelläkään toimi OpenStack ympäristössäni. Laskenta-koneet kyllä näkyvät toimivina laskenta-alustoina Hallintapaneelissa

Hostname	Type	VCPUs (used)	VCPUs (total)	RAM (used)	RAM (total)	Local Storage (used)	Local Storage (total)	Instances
herukka	QEMU	0	4	512MB	926MB	0Bytes	29GB	0
mustikka	QEMU	0	4	512MB	926MB	0Bytes	28GB	0
vadelma	QEMU	0	4	512MB	926MB	0Bytes	28GB	0


openstack. Default • admin admin

Project > Admin > Overview > Compute > **Hypervisors**


Admin / Compute / All Hypervisors

All Hypervisors


Hypervisor Summary



VCPU Usage
Used 0 of 12



Memory Usage
Used 1.5GB of 2.7GB



Local Disk Usage
Used 0Bytes of 85GB

Hypervisor | Compute Host

Filter

Displaying 3 items

Host	Availability zone	Status	State	Time since update	Actions
mustikka	nova	Enabled	Down	0 minutes	Evacuate Host
herukka	nova	Enabled	Up	0 minutes	Disable Service
vadelma	nova	Enabled	Down	0 minutes	Evacuate Host

Displaying 3 items

Mutta kun istuntoa yrittää käynnistää komentokehotteen tai Hallintapaneelin kautta tulee virhe, ettei sopivaa virtualisointi alustakonetta ole käytössä. Kokeilun vuoksi yritin myös tehdä ARM-arkkitehtuuri *flavorin* sekä latasin useamman ARM-arkkitehtuurin levykuvan, kuten CirrOS-ARM sekä OpenSuse-JeOS-arm, joita yritin käynnistää eri vaihtoehdoin ja *flavorein*. Koen kuitenkin, että projektille asettamani lopputulos ”toimiva OpenStack ympäristö kotiloissa” on saavutettu, ainakin suurimmalta osin. Ympäristö on olemassa, laitteet keskustelevat keskenään sen kautta, jokainen laskentakone näkyy kontrollerille hyväksytyinä laskentakoneena ja koneita voi hallinnoida Hallintapaneelin kautta. Viimeinen vaihe olisi löytää toimiva levykuva-*flavor* yhdistelmä, mutta koen sen olevan projektin ulkopuolinen tehtävä.

7 Pohdinta

Työ sai ideansa ja alkunsa minulle usein työhaastattelutilanteessa esitetystä kysymyksestä osaamisesta pilvipalveluiden hallinnasta. Työn laitteistoksi valikoitui jo hallussani olevia laitteita. Olin jo alkuun hieman skeptinen onnistuuko ympäristön asennus kyseisille laitteille, lähinnä Raspberry Pi:n heikohkon laskentatehon takia. Päätin kuitenkin ottaa tämän haasteena, enkä niinkään esteenä. Itse asennusprosessi onnistui hyvin, muutamasta ohjeissa olleesta virheestä huolimaatta. Merkittävimmin näistä mieleen jäi koko uusimman OpenStackin jakelun pakettivaraston puuttuminen ohjeista. Aikaa toki asentamiseen kului. Koska jokaisessa asennusvaiheessa piti myös määrittellä erinäisiä asetuksia, jotta komponentit saatiin keskustelemaan keskenään, jäi minulle erittäin hyvä käsitys siitä, kuinka koko kokonaisuus lopulta toimii. Koen oppineeni valtavasti sekä yleisesti pilvipalveluiden toiminnasta, että OpenStackin sisäisestä toiminnasta ja logiikasta. Koin myös erittäin kiinnostavaksi sekä opettavaiseksi loppuvaiheen virheilmoitusten tulkinnan ja ratkaisuiden etsinnän. Toki harmittamaan jäi, ettei lopullinen järjestelmä toiminut aivan toiveideni mukaisesti ja itse instanssien kokeilu jäi tällä raudalla välistä. Tosin jotain tällaista epäilin jo alussa.

Toki on hyvä myös mainita, ettei pilvipalvelua kannata alkuunkaan tehdä tarjottavaksi loppukäyttäjille kotiloista. Ongelmaksi koituu sekä sähkö- että tietoliikenneverkkojen jatkuvan saatavuuden varmistaminen. Sähkökatkoja en muista kokeneeni, mutta työtä tehdessä Internet-yhteyteni katkesi useammankin kerran. Nämä ovat ongelmia, jotka ovat usein ensimmäisenä ratkaistu kaupallisissa konesaleissa, sähkön saanti varmistettu UPS-laitteiden ja Internetin katkeamattomuus useamman fyysisen reitin avulla jo ennen kuin yhtään laitetta on kannettu sisään. Opinnäytetyöni toimiikin lähinnä ohjeena, kuinka minimaalisen ympäristön voi tehdä ja että se on tehtävissä myös kuluttajatason laitteistolla. Tämä puolestaan auttaa lisäämään ymmärrystä pilvipalvelualustoiden yleisestä toimintalogiikasta. Tämä syvällisempi ymmärrys mahdollistaa kokonaisvaltaisemman pilvipalveluiden ja -alustojen hahmottamisen ja soveltamisen.

Lähteet

Digiarjessa Blogi. 2016. API:t ovat modernin integraatiostrategian ydin. Luettavissa: <https://blog.digia.com/rest-api> Luettu: 27.8.2020.

Tampere University of Technology, Department of Computer Science. 2002. OSI-malli. Luettavissa: <http://www.cs.tut.fi/etaopetus/titepk/luku19/OSI.html> Luettu: 27.8.2020.

SSH Communications. 2020. SSH (Secure Shell) Home Page Luettavissa: <https://www.ssh.com/ssh/> Luettu: 27.8.2020.

HowToGeek. 2017. What is a Windows Domain and How Does It Affect My PC? Luettavissa: <https://www.howtogeek.com/194069/what-is-a-windows-domain-and-how-does-it-affect-my-pc/> Luettu: 27.8.2020.

openSUSE YaST. 2020. YaST, powerful installation and configuration tool for Linux. Luettavissa: <https://yast.opensuse.org/> Luettu: 27.8.2020.

OpenStack. 2019. OpenStack Installation Guide. Luettavissa: <https://docs.openstack.org/install-guide/> Luettu: usein, alkaen 10.2.2020.

Airinen, Petri. 2010. PILVILASKENTA JA PILVIPALVELUT – Pilvialustojen vertailu. Luettavissa: <http://urn.fi/URN:NBN:fi:amk-201103213428> Luettu: 29.8.2020

Google Cloud. 2020. Cloud Computing Services | Google Cloud. Luettavissa: <https://cloud.google.com/> Luettu: 29.8.2020

AWS Documentation. 2020. AWS Documentation, Guides and API References. Luettavissa: <https://docs.aws.amazon.com/index.html> Luettu: 29.8.2020

Microsoft Azure. 2020. Get to Know Azure | Microsoft Azure. Luettavissa: <https://azure.microsoft.com/en-us/overview/> Luettu: 29.8.2020

OpenStack Software. 2020. Open Source Cloud Computing Platform Software – OpenStack. Luettavissa: <https://www.openstack.org/software/> Luettu: 29.8.2020

DevStack. 2019. Using DevStack with neutron Networking. Luettavissa: <https://docs.openstack.org/devstack/latest/guides/neutron.html> Luettu: 24.4.2020

flexVDI. 2015. Enabling KVM virtualization for Raspberry Pi 2. Luettavissa:
<https://blog.flexvdi.com/2015/03/17/enabling-kvm-virtualization-on-the-raspberry-pi-2/> Lu-
ettu: 30.4.2020

Rasberrypi forums. 2019. Setting up Qemu on pi 4? Luettavissa: [https://www.raspber-
rypi.org/forums/viewtopic.php?t=246340](https://www.raspberrypi.org/forums/viewtopic.php?t=246340) Luettu:30.4.2020