

Seppo Eiranto

OPENSTACK AUTOMAATIOKÄYTÖSSÄ

Tekniikan ja luonnontieteiden tiedekunta
Kandidaatintyö
Maaliskuu 2020

TIIVISTELMÄ

Seppo Eiranto: Openstack Automaatiokäytössä
Kandidaatintyö
Tampereen yliopisto
Tekniset tieteet, TkK
Helmikuu 2020

Tässä kandidaatintyössä tutkitaan Openstack pilvialustan soveltuvuutta automaatiokäyttöön tutustumalla jo tehtyihin tutkimuksiin ja Openstackin dokumentaatioon. Soveltuvuutta automaatiokäyttöön arvioidaan automaatiojärjestelmien kolmen perusvaatimuksen kautta: luotettavuus, turvallisuus ja reaaliaikaisuus. Työssä myös perehdytään dokumentaation ja kirjallisuuden kautta Openstackin Compute-osan toimintaan ja suoritetaan Openstackin Queens version testiasennus kahdelle virtuaaliserverille.

Tutkimuksen tuloksissa päästiin tulokseen, että Openstack soveltuu yksityisesti ylläpidettyihin ei kovaa reaaliaikaisuutta vaativiin automaatiojärjestelmiin. Työssä suoritettu testiasennus oli onnistunut ja alustalla kyettiin ajamaan virtuaalista Linux-konetta.

Avainsanat: Openstack, automaatio, reaaliaikaisuus, compute

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ALKUSANAT

Kirjoitin kandidaatintyöni lyhyissä pätkissä yli puolen vuoden aikana muiden opintojeni ohessa aina kun siltä tuntui. Ohjaajaltani Mikko Salmenperältä sain hyviä ohjeita aloitukseen ja aina kirjoituksen aikana niitä pyytäessäni.

Tampereella, 18.2.2020

Seppo Eiranto

SISÄLLYSLUETTELO

1.	JOHDANTO	1
2.	PILVIPALVELUT	2
	2.1 Pilvipalvelut yleisesti	2
	2.2 Pilvipalvelumallit	3
	2.2.1 Mallien jaottelu palveluiden perusteella	3
	2.2.2 Mallien jaottelu palveluiden avoimuuden perusteella	4
3.	OPENSTACK-ASENNUS	5
	3.1 Openstack	5
	3.2 Asennus	5
	3.2.1 Ympäristö	6
	3.2.2 Keystone	6
	3.2.3 Glance	7
	3.2.4 Nova	8
	3.2.5 Neutron	10
	3.2.6 Horizon	10
	3.3 Testaus	11
4.	OPENSTACK COMPUTE	12
	4.1 Nova	12
	4.2 Zun	14
	4.3 Qinling	15
5.	AUTOMAATION VAATIMUKSET	17
6.	OPENSTACK AUTOMAATIOSSA	18
	6.1 Reaaliaikaisuus	18
	6.2 Luotettavuus	19
	6.3 Turvallisuus	20
7.	YHTEENVETO	22
	LÄHTEET	23

LYHENTEET JA MERKINNÄT

AWS	Amazon Web Services
SaaS	engl. Software as a Service, sovellus palveluna
PaaS	engl. Platform as a Service, alusta palveluna
IaaS	engl. Infrastructure as a Service, infrastruktuuri palveluna
HaaS	engl. Hardware as a Service, laitteisto palveluna
API	engl. Application Program Interface, ohjelmoitava sovellusrajapinta
HTTP	engl. Hypertext Transfer Protocol, hypertekstin siirtoprotokolla
DoS	engl. Denial of Service, palvelunesto

1. JOHDANTO

Pilvipalvelut jatkavat suosionsa kasvattamista ja monet yritykset siirtävät toimintojaan tapahtumaan pilvessä niiden skaalautuvuuden ja kustannustehokkuuden takia. Pilvipalvelut kuten AWS tarjoavat esimerkiksi palveluita, jotka skaalaavat automaattisesti asiakkaan tarpeiden mukaan käytettävissä olevaa laskentatehoa. Tämän kaltainen palvelu säästää yrityksiltä rahaa merkittävästi, kun ei tarvitse sijoittaa ylimääräiseen kapasiteettiin, jota tarvitaan vain harvoin ruuhkaisina päivinä.

Openstack on avoimen lähdekoodin pilvikäyttöjärjestelmä, jolla voidaan hallita suuria määriä laskenta, tallennus ja tietoverkko resursseja. Openstack koostuu erilaisista ohjelmisto moduuleista, joita voidaan yhdistellä käyttäjän tarpeiden mukaisesti.

Tämän kandidaatintyön tarkoituksena on tutkia Openstack ohjelmiston laskentatyökaluja ja Openstackin soveltuvuutta automaatiokäyttöön tarkastelemalla sen reaaliaikaisuutta, luotettavuutta ja turvallisuutta. Luvussa 2 käydään lyhyesti läpi mitä pilvipalvelut ovat yleisesti. Kolmannessa luvussa esitellään Openstack ja asennetaan Openstack ohjelmisto kotitietokoneelle virtuaalikoneiden avulla. Luvussa 4 syvennyttään tarkastelemaan Openstackin Compute-osaa, joka koostuu kolmesta osasta. Viidennessä luvussa määritellään automaationvaatimukset. Luvussa 6 tutkitaan Openstackin soveltuvuutta automaation käyttötarkoituksiin automaation keskeisimpien vaatimusten näkökulmista. Tutkimuskeinona käytetään kirjallisuuskatsausta.

2. PILVIPALVELUT

Termille pilvipalvelu ei ole yhtenäistä määritelmää, vaan lähes jokaisella palvelutarjoajalla on omanlaisensa määritelmä. Virallisimman määritelmän pilvipalveluille on julkaissut NIST (National Institute of Standards and Technology). Pilvilaskenta on malli, joka maailmanlaajuisesti mahdollistaa pääsyn helposti konfiguroitaviin yhteisiin laskentaresursseihin, jotka voidaan nopeasti ottaa käyttöön (Mell & Grance 2011). Yleisesti pilvipalvelulla tarkoitetaan, että käyttäjälle tarjotaan verkon yli sovelluksia, tietoliikenneyhteyksiä, laskenta- tai tallennuskapasiteettia, palvelua tai yhdistelmää näistä (Jadeja & Modi 2012).

2.1 Pilvipalvelut yleisesti

NIST luettelee määritelmälleen viisi tärkeintä piirrettä:

1. 'On-demand'¹-itsepalvelu
Kuluttaja voi yksipuolisella päätöksellä allokoida resursseja tarpeidensa mukaisesti ilman ihmiskontaktia.
2. Erilaisten päätelaitteiden toimivuus
Palvelua voi käyttää erilaisten standardi päätelaitteiden kautta, kuten tietokoneen, älypuhelimien ja tablettitietokoneen.
3. Resurssien yhteiskäyttö
Resurssit kerätään yhteen 'multi-tenant'-mallia varten, jossa resursseja jaetaan dynaamisesti kuluttajille.
4. Nopea joustavuus
Resurssien dynaaminen jako on nopeaa ja kuluttajan näkökulmasta vaikuttaa loputtomalta.
5. Monitoroitu palvelu
Palvelun käyttöä automaattisesti mitataan ja laskutetaan jonkin sopivan resurssin suhteen, kuten varastointitilan, laskentatehon tai kaistanleveyden (Mell & Grance 2011).

Näistä viidestä piirteestä, jotka NIST on määritellyt, voidaan päätellä useita pilvipalvelun tuomia etuja kuluttajalle. Organisaatiot voivat suunnitella laskenta-, varastointi, ja yhteyskapasiteettinsa normaalikulutuksensa pohjalta ja kulutuspiikkien aikana siirtää kulutusta pilveen, jossa voidaan lähes rajattomasti lisätä resursseja. Piikkikapasiteetin lisääminen

¹ Kääntäminen suomeksi tekisi piirteen nimestä harhaanjohtavan. 'On-demand' tässä kontekstissa tarkoittaa tarvittavien resurssien jakoa vasta tarpeen tullessa.

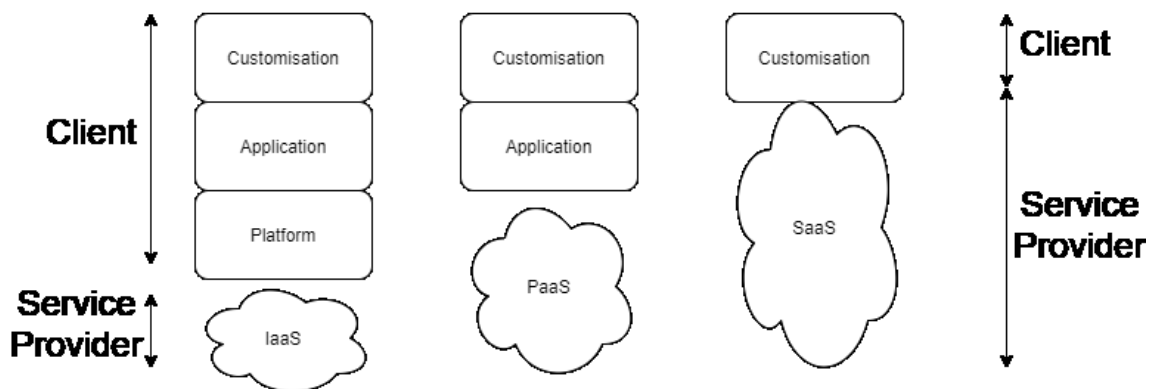
pilvellä voi pienentää IT-kuluja merkittävästi. Startupit hyötyvät myös pilvipalvelutarjoajien lisääntymisestä, mikäli startupilla ei ole pääomaa tai osaamista tehdä omaa IT-infrastruktuuria, jolloin pilvipalvelun käyttö hyödyttää niitä suuresti.

2.2 Pilvipalvelumallit

Pilvipalvelumalleja on paljon erilaisia mutta suurin osa niistä voidaan jakaa kategorioihin niiden tarjoamien palveluiden ja palveluiden avoimuuden perusteella.

2.2.1 Mallien jaottelu palveluiden perusteella

Pilvipalvelut voidaan jakaa neljään luokkaan pilven käyttäjän saaman palvelun perusteella: Software as a Service (SaaS), Platform as a Service (PaaS) ja Infrastructure as a Service (IaaS). Luokat rakentuvat toistensa päälle SaaS täydentäen PaaS:ia ja PaaS täydentäen IaaS:ia. Kuva 1 nähdään vasemmalta oikealle, kuinka neliöissä olevien asiakkaiden hallinnassa ja vastuulla olevien asioiden määrä pienenee.



Kuva 1. Palveluiden eri tasot

Sovellus palveluna (SaaS) on pilvipalveluiden muoto, jota useimmat käyttävät tietämättä käyttävänsä pilvipalvelua. Paras esimerkki tästä on Facebook. Sovellus tarjotaan kuluttajalle joko selaimen tai ohjelmiston kautta. Kuluttaja ei SaaS-ratkaisussa kontrolloi sovelluksen mahdollistavaa infrastruktuuria muuten kuin mahdollisten rajoitettujen käyttäjäkohtaisten asetusten kautta (Mell & Grance 2011).

Alusta palveluna (PaaS) tarjoaa kuluttajalle alustan, jossa voi ajaa itsetehtyjä tai hankittuja ohjelmistoja (Mell & Grance 2011). Käyttäjä ei kuitenkaan voi hallita käyttäjärjestelmää tai muuta pilvi-infrastruktuuriin kuuluvaa toiminnallisuutta.

Infrastruktuuri palveluna (IaaS) tarjoaa kuluttajalle resursseja, ja kuluttaja voi asentaa ohjelmiaan tai käyttöjärjestelmiä infrastruktuurin päälle ja jopa muokata palomuurin asetuksia. IaaS-ratkaisussa kuluttajan hallinnan ulottumattomiin jää vain itse pilvipalvelun infrastruktuuri.

2.2.2 Mallien jaottelu palveluiden avoimuuden perusteella

Palvelut jaetaan neljään erilaiseen infrastruktuuriin: yksityinen, yhteisö, julkinen ja hybridi pilvi, joka on yhdistelmä kahta tai useampaa näistä. Yksityinen pilvi on konsernitason intranettiin verrattavissa oleva vain konsernin sisäiseen käyttöön oleva pilvipalvelu (Birje et al. 2017). Yksityiset pilvet helpottavat tietoturvallisuuden ylläpitoa tehden niistä turvallisimman avoimuusratkaisun ja ohjelmistojen käyttöönottoa keskittämällä sen suurilta osin vain pilvipalveluun (Aryotejo et al. 2018).

Julkiset pilvipalvelut ovat yleisesti käytettävissä verkkoselaimen kautta, ja niiden käyttäjät maksavat niistä käyttönsä mukaan (Birje et al. 2017). Julkiset pilvipalvelut voivat vähentää IT-kustannuksia merkittävästi satunnaisesti käytettävien resurssien kohdalla ja helpottaa sovelluskehitystä mahdollistamalla nopeasti skaalautuvan testiympäristön ohjelmistolle. Turvallisuus on julkisissa palveluissa kuitenkin huonoin eri avoimuusmalleista (Aryotejo et al. 2018), ja niihin voi kohdistua enemmän hyökkäyksiä kuin muihin malleihin.

Kun yksityinen pilvipalvelu laajennetaan tai tehdään sopimuksin kahden tai useamman organisaation käytettävissä olevaksi yhteiseksi pilveksi, sitä kutsutaan yhteisöpilveksi. Yhteisöpilvipalvelut voivat olla yhteisön sisäisesti tai kolmannen osapuolen tuottamia (Birje et al. 2017). Yhteisöpilvipalvelussa voidaan päästä halvempaan toteutukseen kuin yksityisessä, mutta silti saada lähes yksityisen veroinen tietoturvan hallinta (Aryotejo et al. 2018).

Hybridimallissa yksityinen pilvi on linkitetty yhteen tai useampaan julkiseen pilvipalveluun (Birje et al. 2017). Näin saadaan tietoturva pidettyä lähes yhtä hyvänä kuin yksityisessä tai yhteisömallissa samalla mahdollistaen satunnaisen suuren resurssitarpeen tyydyttämisen.

3. OPENSTACK-ASENNUS

3.1 Openstack

Openstackin perusta syntyi Nasalle alihankintaa tekevän Anso Labsin ja Rackspacen välille syntyneestä yhteistyöstä vuoden 2010 ensimmäisinä kuukausina, ja syyskuussa 2012 Openstack Foundation perustettiin hallitsemaan Openstackin kehitykseen ja markkinointiin käytettäviä resursseja (Openstack Foundation 2019c).

Openstack on IaaS-mallin avoimen lähdekoodin pilvialusta, jolla hallitaan konesaleissa suuria pooleja laskenta-, varastointi- ja verkkoresursseja loputtomalla skaalautuvuudella. Resursseja hallitaan ja jaetaan API-yhteyksien kautta hyödyntäen yleisesti käytössä olevia autentikointimenetelmiä (Openstack Foundation 2019r).

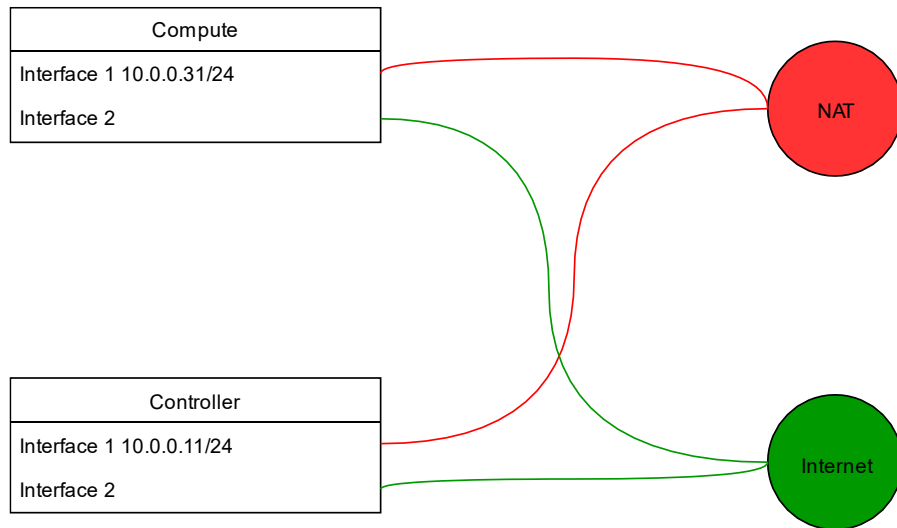
Openstack koostuu useista moduuleista, joista valitaan käyttötarkoitukseen parhaiten soveltuva kokonaisuus. Moduulit on jaettu useisiin kategorioihin niiden käyttötarkoituksen perusteella. Avoin lähdekoodi ja API rajapinnat moduulien välillä tekevät Openstackistä helposti kustomoitavan ja muihin palveluihin integroitavan (Openstack Foundation 2019r). Esimerkkinä integroitavuudesta Openstack tukee ja kannustaa hyödyntämään lisäabstrahointi kerrosta loppukäyttäjän ja itsensä välissä, jotta voidaan tarjota PaaS ja serveritön laskenta-alusta palveluita. Openstackillä on myös esimerkki kokonaisuuksia erilaisiin käyttötarkoituksiin esiteltyinä sivuillaan. Openstack Foundation julkaisee uuden version Openstackistä puolen vuoden sykleissä, mutta julkaisut ovat taaksepäin yhteensopivia, joka mahdollistaa päivittämisen instanssi kerrallaan eikä palvelua tarvitse sulkea (Openstack Foundation 2019q).

3.2 Asennus

Tässä kandidaatintyössä asennetaan Openstackin Queens versio kannettavalla tietokoneella luotaviin virtuaalikoneisiin, joihin asennetaan Linux Ubuntu Server 18.04.2 käyttöjärjestelmät. Asennuksessa seurataan Openstackin sivuilta löytyviä ohjeita (Openstack Foundation 2019n). Virtuaalikoneet luodaan Oracle VM VirtualBoxilla. Toinen virtuaaliservereistä toimii controller noodina, jolle asennetaan Keystone identiteettipalvelu, Neutron tietoverkkopalvelu, Placement resurssienseuranta palvelu, Glance levykuvapalvelu, Horizon käyttöliittymä ja Nova laskentapalvelun API, scheduler ja conductor osat. Compute noodiin asennetaan vain Novan compute osa.

3.2.1 Ympäristö

Ennen Openstackin osien asennusta controll ja compute noodeihin täytyy asettaa virtuaalinen verkko, jossa niillä on kummallakin staattinen ip, jota noodit käyttävät toistensa kanssa kommunikointiin. Toisesta verkkorajapinnasta noodit ovat yhteydessä internetiin mahdollistaen ohjelmistojen lataukset ja päivittämiset. Kaaviossa 1 on verkkokonfiguraatio kuvattuna.



Kuva 2 Virtuaalikoneiden verkkokonfiguraatio

Verkon lisäksi noodeihin tulee asentaa Chrony, joka toteuttaa ajan synkronoinnin noodeissa NTP:llä (Network Time Protocol). Chronyn konfigurointi tehdään niin, että muut noodit synkronoivat aikaansa controller noodin kanssa, joka saa aikansa ulkopuoliselta serveriltä. Controller noodiin asennetaan lisäksi SQL-tietokanta, viestienvälittäjä ja tokenien tallennus ohjelmisto, jota Keystone käyttää.

3.2.2 Keystone

Ensimmäinen asennettava palvelu on Keystone identiteetti palvelu, koska muut palvelut käyttävät sitä myös varmistamaan, että käyttäjä on valtuutettu. Asennus aloitetaan luomalla uusi tietokanta Keystoneille ja antamalla kaikki käyttöoikeudet "keystone" käyttäjälle.

Tämän jälkeen varsinainen asennus tapahtuu yksinkertaisella

```
apt install keystone apache2 libapache-mod-wsgi
```

käskyllä. Keystoneen kanssa asentuu Apache niminen http-serveri hoitamaan palvelupyynnöitä. Asennuspaketti hoitaa Apache serverin ja mod-wsgi moduulin konfiguroinnin.

Keystone käyttää todentamiseen yhdistelmää domainista, projekteista, käyttäjistä ja rooleista. Asennettaessa Keystone automaattisesti luo "default" domainin, jota käytetään tässä asennuksessa. Default domainiin luodaan "service" niminen projekti, johon tehdään omat käyttäjät jokaiselle lisättävälle Openstack palvelulle.

Openstack käyttäjien kirjaaminen Openstackin käskyillä on työlästä ja yksittäisten operaatioiden tekeminen tietyllä käyttäjällä venyttää käskyt todella pitkiksi, minkä takia Openstack tukee skriptejä, joita kutsutaan OpenRC tiedostoiksi. Näissä tiedostoissa asetetaan projektin domain, käyttäjän domain, projektin nimi, käyttäjänimi, salasana, Keystone URL ja Keystone API versio.

1.

```
openstack --os-auth-url http://controller:5000/v3 \
--os-project-domain-name Default --os-user-domain-name Default \
--os-project-name demo --os-username demo token issue
```

Kuva 3 Komento, jolla pyydetään identifiointi tokeni Keystoneelta

Skriptin ajaminen esimerkiksi demokäyttäjälle lyhentää Kuva 3 komennon muotoon
openstack token issue.

Asennusprosessin helpottamiseksi luodaan OpenRC tiedosto "admin-openrc", joka ajamalla saadaan ylläpitäjän käyttöoikeudet asetettua.

3.2.3 Glance

Glance on Openstackin levykuvapalvelu. Palvelu mahdollistaa käyttäjälle levykuvien löytämisen, rekisteröinnin ja noutamisen REST API:n kautta. Levykuvat voidaan tallentaa erilaisin tavoin, kuten yksinkertaiseen tiedostojärjestelmään controller noodiin tai Openstackin olio tallennus palvelu Swiftiin (Openstack Foundation 2019i).

Asennus aloitetaan samalla tavalla kuin Keystone, luomalla Glanceelle oma tietokanta ja asettamalla kaikki käyttöoikeudet "glance" käyttäjälle. Tietokannan valmistuttua ladataan ylläpitäjän käyttöoikeudet ja luodaan "default" domainiin käyttäjä glance, joka liitetään Keystoneen asennuksessa tehtyyn "service" projektiin. Projektille ja äskettäin luodulle käyttäjälle annetaan ylläpitäjän rooli, jonka jälkeen luodaan Glance palvelukokonaisuus käskyillä

```
openstack service create --name glance --description "OpenStack Image" image,
sekä loppupisteet, joiden kautta muut palvelut ottavat yhteyden Glanceen. Levykuvapalvelu asennetaan komennolla
```

```
apt install glance.
```

Seuraavaksi levykuvapalvelun konfigurointitiedostossa määritetään yhteys tietokantaan käyttämään aiemmin luotua "glance" käyttäjää ja asetetaan Keystoneen osoitteet ja glance salasana. Komennolla

```
su -s /bin/sh -c "glance-manage db_sync" glance
```

täytetään tietokanta ja lopuksi Glance käynnistetään uudelleen komennolla

```
service glance-registry restart
```

ja

```
service glance-api restart.
```

Kun Glance on käynnistynyt, lisätään sinne levykuva, josta voidaan myöhemmin käynnistää virtuaalikone. Ensin levykuva täytyy ladata. Tässä asennuksessa ladataan Cirros nimisen käyttöjärjestelmän levykuva käskyllä

```
wget http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img.
```

Kun levykuva on ladattu, se asetetaan Glance palveluun Openstackin käytettäviin käskyllä

```
openstack image create "cirros" \
  --file cirros-0.4.0-x86_64-disk.img \
  --disk-format qcow2 --container-format bare \
  --public
```

Levykuvan asetus voidaan varmistaa käskyllä

```
openstack image list.
```

Mikäli levykuva on käytössä, tulostuu lista, jossa on "cirros" niminen levykuva ja sen status on aktiivinen.

3.2.4 Nova

Nova on Openstackin Compute moduuli, joka esitellään kappaleessa 4.1 tarkemmin. Queens versiossa Nova tarvitsee toimiakseen vähintään Keystone, Glancen ja verkkopalvelu Neutronin.

Novan asennus pohjatyö aloitetaan controller noodista luomalla uudet tietokannat Novalle, Nova-apille ja nova_cell0:lle. Tietokantaan asetetaan käyttäjä "nova", jolle annetaan kaikki oikeudet tietokannan muokkaamiseen. Admin oikeudet otetaan käyttöön käskyllä

```
. admin-openrc
```

, jonka jälkeen käyttäjä luodaan käskyllä

```
openstack user create --domain default --password-prompt nova
```

ja tälle käyttäjälle lisätään admin rooli openstack projektiin (Openstack Foundation 2019l). Nova palvelukokonaisuus luodaan käskyllä

```
openstack service create --name nova --description "OpenStack Compute" compute.
```

Novalle luodaan myös loppupisteet palveluiden keskistä tiedonsiirtoa varten. Controller noodille asentuu Novan osana Placement palvelu, joka tarkkailee resurssien kokonaisuutta ja kulutusta (Openstack Foundation 2019o). Palvelulle luodaan placement niminen käyttäjä käskyllä

```
openstack user create --domain default --password-prompt placement
```

ja lisätään sille admin rooli. Seuraavaksi lisätään Placement palvelu openstackin palveluihin käskyllä

```
openstack service create --name placement --description "Placement API" placement
```

ja luodaan pääteipisteet.

Novan asennus tapahtuu käskyllä

```
apt install nova-api nova-conductor nova-consoleauth nova-novncproxy nova-scheduler nova-placement-api,
```

joka asentaa kaikki controller noodiin kuuluvat Novan osat (Openstack Foundation 2019l). Asennuksen jälkeen Novan konfigurointi tiedostoissa määritetään se käyttämään tietokantaansa, oikeaa viestienvälityspalvelua ja muita asennettuja osia. Määritysten jälkeen täytetään nova-apin tietokanta käskyllä

```
su -s /bin/sh -c "nova-manage api_db sync" nova
```

(Openstack Foundation 2019l). Rekisteröidään aiemmin luotu nova_cell0 tietokanta ja luodaan uusi cell1. Lopuksi, jotta muutokset tulevat käyttöön, käynnistetään kaikki controller noodin Nova palvelut uudestaan.

Compute noodin Nova asennus aloitetaan asentamalla nova-compute käskyllä

```
apt install nova-compute.
```

Asennuksen jälkeen konfiguroidaan muut asennetut palvelut kuten muissakin osissa. Lisäksi on määriteltävä, tukeeko compute noodi laitteistokiihdytystä virtuaalikoneille. Mikäli käsky

```
egrep -c '(vmx|svm)' /proc/cpuinfo
```

palauttaa nollaa suuremman luvun, kone tukee laitteistokiihdytystä ja virtualisointityyppiä ei tarvitse muuttaa (Openstack Foundation 2019l). Jos compute noodi ei tue laitteistokiihdytystä, virtualisointitekniikka täytyy muuttaa Qemu:ksi. Tässä asennuksessa näin tehtiin. Viimeinen asia on nova-computen uudelleen käynnistys.

Novan asennuksen viimeistelyksi compute noodi täytyy lisätä aiemmin luotuun cell1:een controller noodissa käskyllä (Openstack Foundation 2019l)

```
su -s /bin/sh -c "nova-manage cell_v2 discover_hosts --verbose" nova.
```

3.2.5 Neutron

Neutron moduuli mahdollistaa virtuaalisten verkkojen tekemisen ja jakaa IP-osoitteita luoduille virtuaalikoneille. Neutron asennetaan controller noodiin, mutta compute noodiin asennetaan Neutron agentti, joka luo turvallisuus ryhmiä ja hoitaa yhteyksiä virtuaalikoneille (Openstack Foundation 2019j).

Asennus aloitetaan controller noodista ja luodaan tietokanta, käyttäjä, annetaan admin oikeudet ja luodaan palvelukokonaisuus Openstackiin, kuten muidenkin moduulien kohdalla. Tämän jälkeen asennuksessa täytyy tehdä päätös, käytetäänkö olemassa olevia verkkoja, jolloin Openstack ei luo verkkoja, vai käytetäänkö virtuaalisia verkkoja (Openstack Foundation 2019k). Tässä asennuksessa käytetään olemassa olevia verkkoja, jolloin Neutronin tehtävänä on vain jakaa IP osoitteita virtuaalikoneille. Controllerille asennetaan tarvittavat Neutronin osat "apt install" käskyllä ja asetetaan konfigurointi vastaamaan päätöstä käyttää olemassa olevia verkkoja. Controllerin Neutron asennus viimeistellään täyttämällä tietokanta ja käynnistämällä nova-api ja Neutron uudelleen.

Compute noodin Neutron agentti asennetaan "apt install" käskyllä. Asennuksen jälkeen konfigurointi tulee asettaa vastaamaan päätöstä käyttää olemassa olevia verkkoja. Viimeistely tapahtuu käynnistämällä nova-compute ja Neutron agentti uudelleen.

3.2.6 Horizon

Horizon on Openstackin kojelauta, joka tuo selainpohjaisen graafisen käyttöliittymän Openstackiin. Horizon asennetaan controller noodiin käskyllä " apt install openstack-dashboard", jonka jälkeen asetetaan se käyttämään Keystonea ja Neutronia. Neutronin asetuksissa tulee huomioida valittu Neutronin asennus ja määrittää se oikein (Openstack Foundation 2019m).

Asennus viimeistellään lataamalla uudelleen web-serverin asetukset käskyllä "service apache 2 reload". Asennuksen onnistuminen varmistetaan menemällä selaimella osoitteeseen <http://10.0.0.31/horizon>, tämä avaa verkkokäyttöliittymän, mikäli asennus onnistui.

3.3 Testaus

Openstackin toimivuutta voidaan testata luomalla Cirros-käyttöjärjestelmällinen virtuaalikone Openstackin selainkäyttöliittymän kautta. Mikäli kaikki Openstackin osat toimivat voidaan virtuaalikoneelle allokoida resurssit ja käynnistää se.

Toimivuuden varmistuksessa avataan selain ja siirrytään controller koneen IP-osoitteessa olevaan selainkäyttöliittymään. Mikäli etusivu avautuu Horizon on asennettu oikein. Seuraavaksi syötetään aiemmin luodut käyttäjä tunnukset ja varmistetaan Keystone toiminnasta. Kun käyttöliittymä on auennut, valitaan Admin ja Flavors. Tällä välilehdellä "Create Flavor" napin ja määrittelemme tulevan virtuaalikoneen resurssit.

Resurssien määrittelyn jälkeen siirrytään Compute ja sen alta Instances välilehteen. Instances näkymästä nähdään kaikki käynnissä olevat virtuaalikoneet ja se on toistaiseksi tyhjä. Aloitetaan uuden virtuaalikoneen luominen napista "Launch Instance". Ponnahdus ikkunassa välilehdellä source valitaan levykuva, josta virtuaalikone käynnistetään. Asennuksessa Glanceen lisätty Cirros levykuvan pitäisi näkyä saatavilla olevien levykuvien listauksessa. Valitaan Cirros levykuvaksi ja edetään "flavor" kohtaan, jossa valitaan aiemmin luotu flavor. Viimeisenä valitaan verkko, johon virtuaalikone liitetään ja painetaan napista "Launch Instance".

Virtuaalikone allokoidaan ja hetken päästä sen "power state" on "running". Klikataan instanssin nimeä ja valitaan console välilehti. Välilehdellä saadaan yhteys virtuaalikoneeseen. Mikäli virtuaalikone käynnistyy, on Nova ja Neutron asennukset myös onnistuneet.

Tässä työssä suoritettu Openstack asennus on tämän testauksen perusteella onnistunut.

4. OPENSTACK COMPUTE

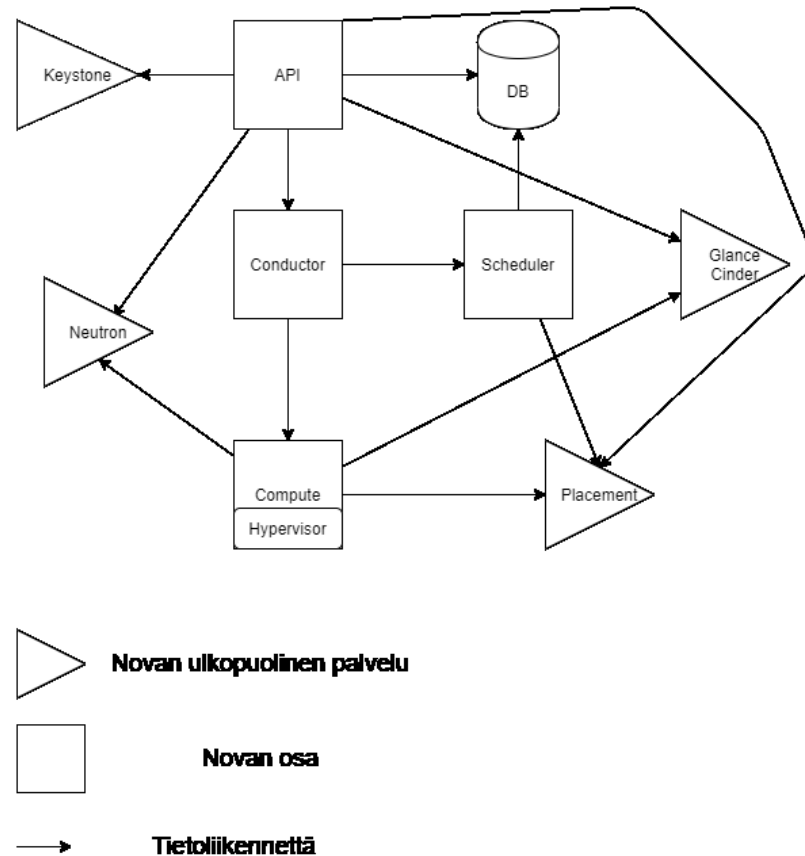
Openstack Compute on Openstackin laskennan suorittavat osat. Compute koostuu kolmesta moduulista: Nova, Zun ja Qinling.

4.1 Nova

Nova on Openstackin moduuli, joka mahdollistaa laskentainstanssien eli virtuaaliserve-
reiden luomisen ja jakamisen kuluttajille ja on vastuussa resurssien tarjoamisesta niille
(Openstack Foundation 2019d). Nova tukee virtuaalikoneita, bare-metal servereitä ja
jossain määrin käyttöjärjestelmä tason säiliöintiä. Kun jokin edellä mainituista palveluista
käynnistetään Nova varaa sille resurssit ja säättää niitä palvelun käytön aikana tarpeiden
mukaan. Palvelun tullessa tarpeettomaksi Nova vapauttaa resurssit muiden palveluiden
käytettäväksi. Novalla luotuja virtuaalikoneita käytetään SSH-yhteyden tai VNC:n graa-
fisen käyttöliittymän kautta.

Nova yhdistää Openstackin käytössä olevat serverit yhdeksi resurssipooliksi käyttäen
käyttäjän valitsemaa hypervisoria, joka virtualisoi resurssit. Hypervisoreita tuetaan useita
erilaisia, mutta tuetuissa ominaisuuksissa on vaihtelevuutta. Parhaiten tuettuja hypervi-
soreja ovat KVM ja QEMU (Openstack Foundation 2019b).

Novan kaikki loppukäyttäjä toiminnot ovat käytettävissä REST API:n kautta ja sitä voi-
daan hyödyntää automatisointiin tai uudenlaisen toimintalogiikan luomiseen (Openstack
Foundation 2019e). Hallinta on myös mahdollista Openstackin graafisen käyttöliittymän
Horizonin, komentorivikäyttöliittymä Openstack Clientillä ja Nova Clientillä.



Kuva 4 Novan esimerkki arkkitehtuuri

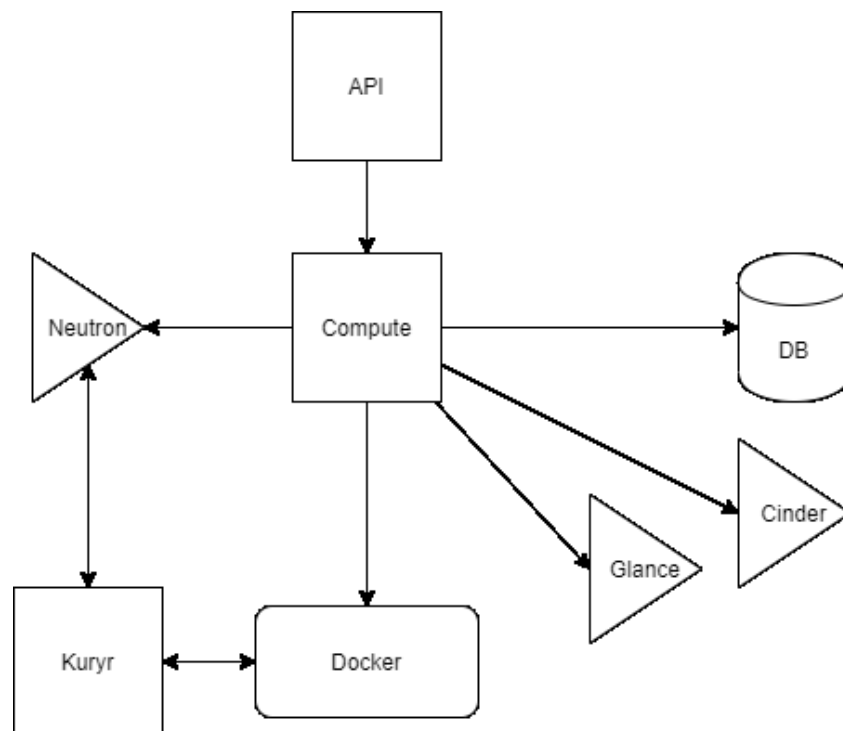
Kuva 4 on Novan esimerkki arkkitehtuuri yhdistettynä autentikointimoduuli Keystoneen, verkkomoduuli Neutroniin, resurssienvälontamoduuli Placementtiin, levykuvamoduuli Glanceen, tallennuslohkomoduulin Cinder ja tietokantaan (Openstack Foundation 2019e). Edellä mainitut moduulit ovat välttämättömiä Novan perustoiminnalle uusimassa Openstackin julkaisussa eli Steinissa. Nova tarvitsee toimintaansa vähintään kaksi noodia, controller ja compute, toimiakseen. Compute on ainoa osa Novasta, joka asennetaan compute noodiin. Loput osista asennetaan controller noodiin. Nämä voidaan asentaa samaan laitteeseen, mutta pilvilaskennan hyödyt tulevat ilmi vasta kun useampia laitteita käytetään palveluiden ajamiseen.

Novan API lohko ottaa vastaan HTTP pyyntöjä, muuntaa käskyjä ja kommunikoi muiden komponenttien kanssa. Conductor lohko toimii välityspalvelimena Compute lohkon ja tietokantalohkon välissä parantaen turvallisuutta. Compute lohkokosta aiotaan myös tulevaisuudessa siirtää toiminnallisuutta Conductoriin (Openstack Foundation 2019a). Tällä py-

ritään parantamaan virheen käsittelyä ja monen laskentanoodin kattavia toimintoja. Compute lohko kommunikoi hypervisorin kanssa ja luo virtuaalikoneita levykuvilla Glancesta. Scheduler lohko valitsee Compute noodin, joka luo virtuaalikoneen.

4.2 Zun

Zun on Openstackin sovellustason säiliöiden ajo moduuli, jolla voidaan ajaa säiliöitä huolehtimatta servereistä tai klustereista (Openstack Foundation 2019t). Zunin, kuten Novan, ominaisuudet ovat käytettävissä REST API:n, Horizonin, Openstack Clientin ja Zun Clientin kautta. Noodeja Zun tarvitsee myös vähintään kaksi; controller ja compute. Toimiakseen Zun tarvitsee vähintään kaksi muuta moduulia: Keystone ja Neutronin, mutta riippuen konfiguraatiosta Zun voi kommunikoida jopa kuuden muun moduulin kanssa (Openstack Foundation 2019t).



Kuva 5 Zun arkkitehtuuri

Compute noodi, joita voi olla monta, ajaa kuvan Kuva 5 zun-computea ja oletusarvonaan Zun käyttää säiliöiden ajoon Dockeria, mutta on myös mahdollista konfiguroida muita säiliöohjelmistoja tai useampia eri säiliöohjelmistoja käytettäväksi (Openstack Foundation 2019t). Noodissa ajetaan myös verkkopalvelua, joka yhdistää säiliöt virtuaaliverkkoihin ja tekee palomuurit instansseille.

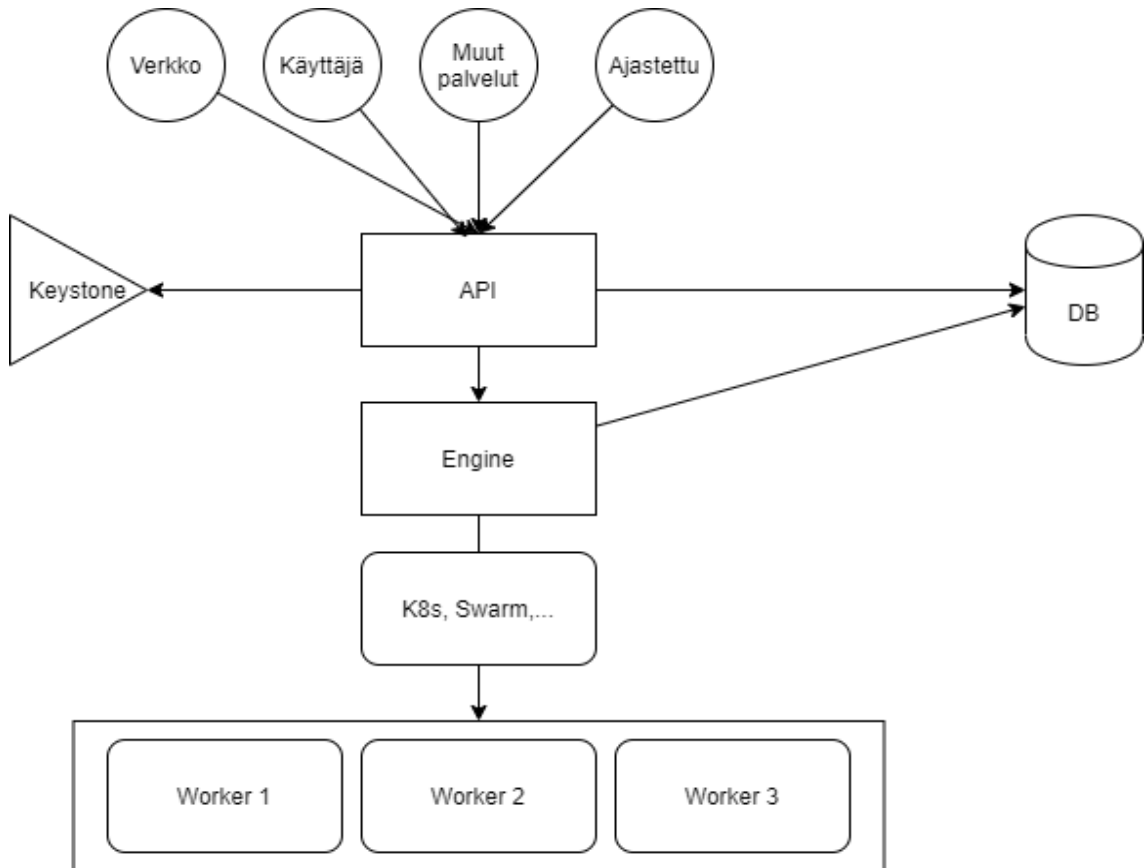
Controller noodi ajaa kuvassa Kuva 5 näkyvää zun-apia ja tietokantaa ja aiemmin mainittuja Zunin toiminnalle pakollisia moduuleja (Openstack Foundation 2019t).

Sovellusten ajaminen säiliöissä säästää resursseja huomattavasti verrattuna virtuaalikoneisiin, koska säiliöt eivät tarvitse täyttää käyttöjärjestelmää toimiakseen vaan säiliöinti-prosessissa pakataan kaikki ohjelman riippuvuudet mutta ei ylimääräistä. Tarvittaessa säiliöille voidaan määrittää vain niille tarkoitetut resurssit, jolloin Zun etsii noodin, jossa on tarvittavat resurssit ja käynnistää säiliön tuolla noodilla (Singh 2017). Säiliöinti myös mahdollistaa ohjelman identtisen suorituksen alustasta huolimatta koska säiliö käyttää järjestelmän resursseja suoraan, eikä käyttöjärjestelmän kautta. Zunilla luodut säiliöt voivat muodostaa yhteyksiä Novalla luotuihin virtuaalikoneisiin Kuryr kirjaston avulla, joka käyttää Neutronia luomaan verkkoresurssit (Singh 2017).

4.3 Qinling

Qinling on Openstackin moduuli, jolla voidaan ajaa yksittäisiä funktioita huolehtimatta resurssien allokoinnista, Qinling automaattisesti skaalaa resursseja ylös ja alaspäin suoritettavan funktion tarpeiden mukaisesti.

Funktioiden suorittaminen voidaan asettaa ajastetuksi, jolloin Openstack huolehtii kaikesta, tai millä tahansa web tai mobiilisovelluksella. Suoritus voidaan myös laukaista muista Openstackin palveluista. Suoritus kutsussa voidaan määrittää toteutetaanko suoritus synkronisesti, jolloin kutsuva prosessi pysähtyy kunnes Qinlingissä suoritettava funktio on suoritettu loppuun vai asynkronisesti mahdollistaen funktion suoritusta kutsuvan prosessin eteneminen (Openstack Foundation 2019p).



Kuva 6 Qinling arkkitehtuuri

Kuvassa 6 nähdään funktio kutsujen suorituksen eteneminen Qinlingissä, kutsu lähetetään qinling-api:lle, joka tarkistaa kutsujan identiteetin Keystone palvelulla ja tekee tietokantaan merkinnän uudesta suorituksesta. Mikäli kutsu on määritetty asynkroniseksi qinling-api vastaa tietokanta merkinnän jälkeen kutsujalle, että funktio on suorituksessa. Seuraavaksi suoritus siirretään qinling-enginelle, joka on vastuussa kommunikaatiosta säiliöintiohjelmiston kanssa ja resurssien ylläpidosta. Säiliöintiohjelmisto siirtää työn, jollekin worker kokonaisuudelle, jonka toinen osa lataa funktiopaketin ja toinen suorittaa funktion. Qinling-engine päivittää tietokantaan suorituksen statuksen (Lingxian 2018).

5. AUTOMAATION VAATIMUKSET

Automaatio asettaa käyttämilleen järjestelmille erityisvaatimuksia koska automaatiojärjestelmän pettäminen voi tarkoittaa lämpökattilan sulamista sähkölaitoksessa tai ihmisvahinkoja autotehtaalla. Automaation erityisvaatimukset ICT mielessä ovat reaaliaikaisuus, luotettavuus ja turvallisuus.

Reaaliaikaisuus tarkoittaa, että ohjelmisto käsittelee sisään tulevaa dataa niin että tulokset ovat saatavilla ennalta määrättyssä ajassa tai aikaikkunassa. Reaaliaikaisuus voidaan jakaa neljään erilaiseen osaan: kova, isokrooninen, pehmeä ja rajaton reaaliaikaisuus. Kovan reaaliaikaisuuden järjestelmissä tulokset ovat hyödyllisiä aikarajaan asti, mutta sen jälkeen täysin hyödyttömiä. Isokroonisen reaaliaikaisuuden järjestelmissä tulokset ovat hyödyllisiä vain tietyn aikaikkunan sisällä, joka ei välttämättä ala välittömästi ohjelman suorituksesta. Pehmeän reaaliaikaisuuden järjestelmissä tulosten hyödyllisyys laskee aikarajan jälkeen ajan funktiona.

Erilaiset automaatiosovellukset vaativat eritasoisia reaaliaika pystyvyyttä. Säättöpiiri vaatii kovaa reaaliaikaisuutta, jotta säädettävä prosessi pysyy vakaana, mutta tuotannonohjaus järjestelmälle riittää pehmeä reaaliaikaisuus.

Automaatio järjestelmille luotettavuus on olennaista, koska järjestelmien pettäessä aiheutuneet vahingot voivat olla mittavat. Luotettavuus on kyky tuottaa palvelua, johon voidaan oikeutetusti luottaa. Sille voidaan määrittää viisi attribuuttia:

- saatavuus: oikean palvelun valmius
- luotettavuus: oikean palvelun jatkuvuus
- turvallisuus: katastrofisten seurausten puute
- eheys: väärin systeemi muutosten puute
- ylläpidettävyys: helposti muutettavissa ja korjattavissa (Avi zienis et al. 2004)

Turvallisuus voidaan jakaa kahteen luokkaan; tietoturvallisuuteen, ja henkilö- ja ympäristöturvallisuuteen. Tietoturvallisuuteen sisältyy datan ja yhteyksien suojaaminen, jotta järjestelmään ei pystytä pääsemään käsiksi ulkopuolella. Henkilö- ja ympäristöturvallisuus on lain määrittelemä vaatimus järjestelmän turvallisuudesta sen kanssa työskenteleville ihmisille.

6. OPENSTACK AUTOMAATIOSSA

Tämän luvun tarkoitus on tarkastella Openstackin sopivuutta erilaisiin automaatiosovel-
luksiin kolmesta eri vaatimuksen näkökulmasta. Kappaleessa 6.1 tarkastellaan
Openstackin tarjoamia vaihtoehtoja eritason reaaliaikaisuuksiin. Kappale 6.2 käsittelee
Openstackin luotettavuutta ja kappaleessa 6.3 tutkitaan Openstackin turvallisuutta.

6.1 Reaaliaikaisuus

Reaaliaikaisuuden vaatimusten täyttymiseksi Openstackillä luotujen virtuaalikoneiden
laskentateho pitäisi pystyä takaamaan. Openstack tarjoaa resurssien allokointiin virtuaa-
likoneille ylläpitäjän luomia esiasetettuja luokkia ("flavor"), joilla pystytään määrittämään
koneelle tulevan virtuaalisten prosessorien, keskusmuistin ja tallennustilan määrän.
Tämä ei kuitenkaan mahdollista virtuaaliprosessorien tehon varmistamista, sillä virtuaa-
liprosessorit eivät saa yhtä tiettyä todellista vaan käyttävät satunnaisesti isäntäkoneen
prosessoreita (Openstack Foundation 2019h).

Myös samaa todellista prosessoria voi käyttää useampi virtuaaliprosessori, jolloin rea-
liaikaisuutta vaativa virtuaalikone voidaan keskeyttää, jotta toisen virtuaaliprosessin suo-
ritusta voidaan jatkaa. Tätä prosessorien ja virtuaaliprosessorien määrien suhdetta voi-
daan säätää ja sen oletusarvo on 16:1 (Openstack Foundation 2018b), eli jokaista fyys-
istä prosessoria kohti Openstack näkee 16 virtuaalista.

Prossessorien "ylibuukkauksella" ja satunnaisten prosessorien käytöllä saavutetaan hyvä
resurssien käyttösuhde, mutta sillä on negatiivinen vaikutus virtuaalikoneiden nopeu-
teen. Virtuaalisten ja fyysisten prosessoreiden suhteen vaihtaminen yhdeksi ei korjaa
tilannetta. Prosessoreilla voidaan silti ajaa Openstackkiä ajavan käyttöjärjestelmän teh-
täviä ja tämä voi aiheuttaa piikkejä latenssiin. Myös käytetty virtualisoija itsessään voi
aiheuttaa piikkejä.

Ratkaisuksi näihin ongelmiin Openstackiin on sisällytetty Pike julkaisusta lähtien mah-
dollisuus luoda virtuaalikoneita, joille varataan fyysisiä prosessoreita mahdollistamaan
laskentatehon vakaus (Openstack Foundation 2019h). Näillä varatuilla prosessoreilla ei
ajeta mitään muuta kuin virtuaalikoneen laskentaa, edes Openstackin alla toimivan ser-
verin ohjelmiston, kuten Ubuntu serverin. Näin voidaan saavuttaa pehmeää reaaliaikai-
suutta toteuttavia järjestelmiä.

Virtualisoinnin aiheuttamien latenssi piikkien poistamiseksi ja deterministisyyden saavut-
tamiseksi tarvitaan kuitenkin myös reaaliaikaa varten tehty virtualisoija. Reaaliaikaisuutta

tukevia virtualisoijia on vähän ja niiden toimivuutta kovaan reaaliaikaisuuteen Openstackin kanssa ei ole tutkittu riittävästi.

Toimivan kovaa reaaliaikaisuutta tukevan virtualisoijan, joka toimii Openstackin kanssa on kuitenkin tehnyt Xi *et al.* (2015). Virtualisoijan käyttö kuitenkin vaati myös muokkauksia Openstackiin, jotta virtuaalikoneiden sijoittaminen ja resurssien valitseminen tukisivat reaaliaikaisia virtuaalikoneita ja, jotta tiedot pystyttäisiin välittämään virtualisoijalle. Testeissä pystyttiin ajamaan 11 virtuaalista kovaa reaaliaikaa vaativaa konetta samalla kun loput vapaana olevat resurssit hyödynnettiin täysin ei-reaaliaikaisilla virtuaalikoneilla (Xi *et al.* 2015). Kehitettyä ratkaisua ei kuitenkaan ole yhdistetty viralliseen Openstackiin, joten se ei saa tukea, kehitystä, tietoturvapäivityksiä eikä sen vakautta ole kattavasti testattu.

Kaupallinen Openstackin tarjoaja Red Hat esittää konfigurointi ohjeita tuotteelleen (Red Hat 2019), joilla heidän tarjoama Openstackin versio pystyy reaaliaikaisuuteen. Red Hat ei kuitenkaan esitä testituloksia tai arvoja, eikä spesifioi puhuuko kovasta vai pehmeästä reaaliaikaisuudesta.

6.2 Luotettavuus

Luotettavuuden testaus monimutkaisissa pilvialustoissa, kuten Openstack, ei ole yksiselitteistä eikä yksinkertaista. Luotettavuus limittyy myös turvallisuuteen, jota käsitellään kappaleessa 6.3 ja ei näin käsitellä tässä.

Openstack voidaan konfiguroida korkean saatavuuden pilvialustaksi, käyttäen joko Openstack Foundationin julkaisemia ohjeita (Openstack Foundation 2018a) tai hyödyntämällä kaupallisten järjestelmätoimittajien, kuten Red Hat tai Mirantis, tarjoamia ratkaisuja. Järjestelmätoimittajien kautta hankittaessa voidaan vaadittava saatavuus määrittää palvelutasosopimuksessa.

Du *et al.* ehdottaa standardoitua tapaa mitata Openstackin saatavuutta injektoimalla viikoja järjestelmään ja mitata järjestelmän kykyä tarjota palveluita viasta huolimatta tai aikaa, joka palvelun palaamiseen kuluu (Du *et al.* 2017). Tätä mittaustapaa ei kuitenkaan ole vielä käyttöön ja ainoat mittaukset on julkistanut Openstack Foundation. Nämä mittaukset on tehty Openstack Mitakalle, joka julkaistiin 2016 ja ovat osa Openstackin käyttäjille suunnattua suoritustehon varmistamisen testaus opasta (Openstack Foundation 2017). Saatuja tuloksia ei kuitenkaan ole tulkittu mitenkään ja niiden tulkitseminen jätetään tämän kandidaatintyön ulkopuolelle.

Pilvipalveluiden ylläpito verrattuna perinteisiin työkoneisiin on erinomainen. Koska virtuaalikoneet eivät ole sidottuja tiettyihin resursseihin, voidaan päivitettävä tai viallinen prosessori, kovalevy tai muu osa poistaa käytöstä ja Openstack siirtää suorituksen muille resursseille, jotka ovat saatavilla. Pilvipalvelut ovat luonteeltaan redundanttisia ja Openstack tukee laajennuksia resursseihin ja palveluihin käynnissä olevaan pilveen (Openstack Foundation 2018c).

6.3 Turvallisuus

Openstackin, kuten muidenkin pilvialustojen, turvallisuus nojaa ohjelmistonsa lisäksi asennuksen tekijään, ylläpitäjiin ja käyttäjiin. Asennukset täytyy suorittaa tietoturvasuhteellisuuden ja Openstackin alla toimivien servereiden käyttöjärjestelmien tulee olla ajan tasalla. Mikäli serverit, joilla Openstackkiä ajetaan eivät ole turvallisia, ei voi myöskään Openstack olla. Pilvialusta vaatii ylläpitoa, mahdollisia päivityksiä ja valvontaa, että käyttöoikeuksia on vain oikeilla tahoilla.

Paikallinen asennus, jossa serverit sijaitsevat tuotantolaitoksessa, mahdollistaa paikallisen verkon käytön automaatio-sovelluksiin. Automaatio-sovellusten siirtyessä paikalliseen verkkoon pienennetään hyökkäyspinta-alaa merkittävästi järjestelmistä, mutta vaihtokaupassa annetaan internetin yli toiminnanohjaaminen ja datan siirto. Tuotantotiloissa olevat serverit olisivat myös haavoittumattomia Denial of Service (DoS) hyökkäyksille ja internet-yhteyden katkeamisille. Järjestelmiin pääsy vaatii fyysisen läsnäolon. Yksityinen pilvi myös poistaa pilven käyttäjien välisten hyökkäysten mahdollisuuden.

Elia et al. tekemässä haavoittuvuusanalyysissä todettiin että 2/3 Openstackin haavoittuvuuksista viiden vuoden ajalta ovat olleet pilven sisäisiä haavoittuvuuksia, joissa toiset pilven käyttäjät hyökkäävät muita käyttäjiä vastaan (Elia et al. 2017). Analyysissä myös todettiin, että enemmän kuin 20% haavoittuvuuksista on ollut huomaamatta yli vuoden. Elia et al. myös pääättelee tästä, että haavoittuvuuksia, joita ei ole vielä huomattu on todennäköisesti huomattava määrä.

Useat yritykset valitsevat palveluntarjoajia, kuten Mirantis tai Red Hat. Palveluntarjoajat käyttävät pohjana Openstackkiä ja voivat hoitaa asennukset ja päivitykset asiakkaan puolesta. Palveluntarjoajien etuna on löydettyjen tietoturvaongelmien pikainen paikkaus palveluntarjoajan toimesta ja tarjotut arkkitehtuurit ovat testattuja (Anon n.d.) (Anon n.d.). Monet palveluntarjoajista myös lisäävät omia ohjelmistojaan helpottamaan pilvenvalvontaa.

Yksi tunnetuista tietoturva ongelmista Openstackissä on että Openstack mahdollistaa tallennustilan suojaamisen kryptaamalla sen, mutta kun tämä tallennustila kiinnitetään virtuaalikoneeseen kryptaus puretaan (Anisetti et al. 2015). Tämä mahdollistaa salatun tiedon lukemisen, mikäli virtuaalikoneeseen päästään murtautumaan, myös ilman sa-lausavainta.

Openstack on konfiguroitavissa täyttämään useita tietoturva standardeja. Esimerkkeinä näistä standardeista Openstack Foundation esittää ISO 2700 1/2, jota suositellaan läh-tökohdaksi tietoturvan sertifiomisessa, PCI-DSS, joka standardoi maksukorttidatan kä-sittelyn, ja Amerikan hallituksen jakaman FedRAMP sertifikaatin (Openstack Foundation 2019g). Tästä voidaan päätellä, että oikein konfiguroituna Openstack on puutteistaan huolimatta verrattain turvallinen alusta.

7. YHTEENVETO

Tämän työn tarkoitus oli tutkia Openstackin soveltuvuutta automaatio-sovelluksiin, selvittää Openstackin laskentaosien toimintaperiaatteita ja tehdä koeasennus virtuaaliserveille. Soveltuvuutta tutkittiin automaation vaatimusten kautta ja laskentaosien tutkimiseen käytettiin Openstackin omaa dokumentaatiota, sekä Lingxianin ja Singhin raportteja.

Openstackin soveltumisen arvioimisessa käytettiin dokumentoituja ominaisuuksia ja tehtyjä tutkimuksia. Openstackin ominaisuuksia käsitteleviä artikkeleita ei kuitenkaan ole kovinkaan montaa tehty, joten tehdyt päätelmät pohjautuvat monissa kohdissa dokumentaatioissa ilmoitettuihin ominaisuuksiin.

Turvallisuutensa puolesta Openstack ei Elia et al. tekemän analyysin perusteella sovellu julkisissa- tai hybridipilvissä automaatio-sovellusten ajamiseen, koska Openstack on ollut aina haavoittuvainen käyttäjien välisille hyökkäyksille. Yksityisenä pilvenä se voi kuitenkin tarjota riittävän turvallisuuden tuotannonohjausjärjestelmiä tai tuotannon data-analyysiä varten. Tämän puolesta puhuu myös Openstackin sertifioitavuus (Openstack Foundation 2019g) ja se, että esimerkiksi japanilainen teräsvalmistaja JFE Steel (Bain 2016) ja Volkswagen Group käyttää järjestelmässään (Urban et al. n.d.) Openstackia.

Luotettavuus on Openstackin huonoiten dokumentoitu ja tutkittu ominaisuus, joten luotettavuutta voitiin tarkastella vain saatavuuden ja ylläpidettävyyden pohjalta. Saatavuutta ei ole tutkittu riippumattomasti, mutta Openstack Foundation tarjoaa tuloksia esimerkki kokoonpanosta (Openstack Foundation 2016). Ylläpidettävyyttä tarkasteltiin pilvialustojen yleisen toimintaperiaatteen pohjalta ja Openstackin oman laajennettavuuden kautta. Ylläpito Openstackissa on helpotettua yhtenäisen hallintonäkymän ansiosta ja koska mahdollisen laitteiston hajoamisen tapauksessa virtuaalikoneet voidaan vain siirtää muulle laitteistolle. Pilvialusta mahdollistaa myös helpon laajentamisen, resurssi tarpeen kasvaessa voidaan uusia resursseja yksinkertaisesti liittää järjestelmään.

Reaaliaikaisuuden puolesta automaatio-sovelluksilla on eritasoisia vaatimuksia, tuotannonohjausjärjestelmä ei tarvitse kovaa reaaliaikaisuutta mutta prosessin säätöjärjestelmä vaatii. Openstack ei tue ilman merkittäviä muutoksia kovaa reaaliaikaisuutta, (Xi et al. 2015) mutta saadaan tukemaan pehmeää reaaliaikaisuutta.

LÄHTEET

Anisetti, Marco; Ardagna, Claudio A; Damiani, Ernesto; Gaudenzi, Filippo & Veca, Roberto 2015 'Toward Security and Performance Certification of Open Stack' in 2015 *Proceedings - 2015 IEEE 8th International Conference on Cloud Computing, CLOUD 2015* Institute of Electrical and Electronics Engineers Inc.: 564–571.

Anon 'OpenStack | Mirantis' Available at: <https://www.mirantis.com/software/mcp/openstack/> Accessed 4.2.2020a.

Anon 'Red Hat OpenStack Platform' Available at: <https://www.redhat.com/en/technologies/linux-platforms/openstack-platform> Accessed 4.2.2020b.

Aryotejo, Guruh; Kristiyanto, Daniel Y & Mufadhol 2018 'Hybrid cloud: Bridging of private and public cloud computing' in 2018 *Journal of Physics: Conference Series*.

Avi zienis, Algirdas; Laprie, Jean-Claude; Randell, Brian; Landwehr, Carl & Member, Senior 2004 'Basic Concepts and Taxonomy of Dependable and Secure Computing' *IEEE Transactions on Dependable and Secure Computing* 1/1:11–33.

Bain, Nic 2016 'How JFE Steel Corporation forged an integrated data center with help from IBM Japan and OpenStack - Superuser' Available at: <https://superuser.openstack.org/articles/how-jfe-steel-corporation-forged-an-integrated-data-center-with-help-from-ibm-japan-and-openstack/> Accessed 5.2.2020.

Birje, Mahantesh N; Challagidad, Praveen S; Goudar, RH & Tapale, Manisha T 2017 'Cloud computing review: concepts, technology, challenges and security' *International Journal of Cloud Computing* 6/1:32.

Du, Qingfeng et al. 2017 'High availability verification framework for OpenStack based on fault injection' in 2017 *Proceedings of 2016 11th International Conference on Reliability, Maintainability and Safety: Integrating Big Data, Improving Reliability and Serving Personalization, ICRMS 2016* Institute of Electrical and Electronics Engineers Inc.

Elia, Ivano Alessandro; Antunes, Nuno; Laranjeiro, Nuno & Vieira, Marco 2017 'An Analysis of OpenStack Vulnerabilities' in 2017 *Proceedings - 2017 13th European Dependable Computing Conference, EDCC 2017* Institute of Electrical and Electronics Engineers Inc.: 129–134.

Jadeja, Yashpalsinh & Modi, Kirit 2012 'Cloud computing - Concepts, architecture and challenges' *2012 International Conference on Computing, Electronics and Electrical Technologies, ICCEET 2012* /March 2012:877–880.

Lingxian, Kong 2018 'How to implement FaaS in OpenStack for public cloud | OpenStack Summit Videos' Available at: <https://www.openstack.org/videos/summits/vancouver-2018/how-to-implement-faas-in-openstack-for-public-cloud> Accessed 31.10.2019.

Mell, Peter & Grance, Timothy 2011 'The NIST Definition of Cloud Computing (Draft) Recommendations of the National Institute of Standards and Technology' *NIST Special Publication*.

Openstack Foundation 2016 'OpenStack Docs: 1. Introduction' Available at: <https://docs.openstack.org/performance-docs/latest/introduction.html> Accessed 5.2.2020.

'OpenStack Docs 2017: Performance Documentation' Available at: <https://docs.openstack.org/performance-docs/latest/index.html> Accessed 26.1.2020.

'OpenStack Docs 2018a: OpenStack High Availability Guide' Available at: <https://docs.openstack.org/ha-guide/> Accessed 26.1.2020.

'OpenStack Docs 2018b: Overcommitting CPU and RAM' Available at: <https://docs.openstack.org/arch-design/design-compute/design-compute-overcommit.html> Accessed 25.1.2020.

'OpenStack Docs 2018c: Working with Hardware' Available at: <https://docs.openstack.org/operations-guide/ops-maintenance-hardware.html> Accessed 27.1.2020.

'OpenStack Docs 2019a: Conductor' :7–9 Available at: <https://docs.openstack.org/mitaka/config-reference/compute/conductor.html> Accessed 12.8.2019.

'Feature Support Matrix 2019b' :1–17 Available at: <https://docs.openstack.org/nova/latest/user/support-matrix.html> Accessed 12.8.2019.

'Introduction: A Bit of OpenStack History' 2019c :1–4 Available at: <https://docs.openstack.org/project-team-guide/introduction.html> Accessed 10.8.2019.

'Nova Compute Docs' 2019d :1–4 Available at: <https://docs.openstack.org/nova/latest/index.html> Accessed 12.8.2019.

'Nova Compute Docs' 2019e Available at: <https://docs.openstack.org/nova/latest/index.html> Accessed 12.8.2019.

'Nova Example Architecture' 2019f Available at:
<https://docs.openstack.org/nova/latest/user/architecture.html> Accessed 12.8.2019.

'OpenStack Docs 2019g: Certification and compliance statements' Available at:
<https://docs.openstack.org/security-guide/compliance/certification-and-compliance-statements.html> Accessed 21.1.2020.

'OpenStack Docs 2019h: CPU topologies' Available at:
<https://docs.openstack.org/nova/train/admin/cpu-topologies.html> Accessed 5.11.2019.

'OpenStack Docs 2019i: Image service overview' Available at:
<https://docs.openstack.org/glance/queens/install/get-started.html> Accessed 19.11.2019.

'OpenStack Docs 2019j: Install and configure compute node' Available at:
<https://docs.openstack.org/neutron/queens/install/compute-install-ubuntu.html>
Accessed 7.1.2020.

'OpenStack Docs 2019k: Install and configure controller node' Available at:
<https://docs.openstack.org/neutron/queens/install/controller-install-ubuntu.html>
Accessed 7.1.2020.

'OpenStack Docs 2019l: Install and configure controller node for Nova Ubuntu' Available at:
<https://docs.openstack.org/nova/queens/install/controller-install-ubuntu.html>
Accessed 5.1.2020.

'OpenStack Docs 2019m: Install and configure for Ubuntu' Available at:
<https://docs.openstack.org/horizon/queens/install/install-ubuntu.html> Accessed
7.1.2020.

'OpenStack Docs 2019n: OpenStack Installation Guide' Available at:
<https://docs.openstack.org/install-guide/index.html> Accessed 14.11.2019.

'OpenStack Docs 2019o: Placement API' Available at:
<https://docs.openstack.org/nova/queens/user/placement.html> Accessed 5.1.2020.

'Qinling Overview' 2019p :1–4 Available at: <https://docs.openstack.org/qinling/stein/>
Accessed 20.8.2019.

'Upgrading Openstack' 2019q :1–6 Available at:
<https://docs.openstack.org/cinder/latest/contributor/rolling.upgrades.html> Accessed
10.8.2019.

'Vision for OpenStack Clouds Purpose' 2019r :1–7 Available at:
<https://governance.openstack.org/tc/reference/technical-vision.html> Accessed
10.8.2019.

'What is openstack?' 2019s :1–5 Available at: <https://www.openstack.org/software/>
Accessed 14.8.2019.

'Zun overview' 2019t :8–10 Available at:
<https://docs.openstack.org/zun/https://docs.openstack.org/zun/stein/install/overview.html>
Accessed 12.8.2018.

Red Hat 2019 'Chapter 25. Configuring Real-Time Compute Red Hat OpenStack Platform 13 | Red Hat Customer Portal' Available at:
https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html/advanced_overcloud_customization/realtime-compute
Accessed 25.1.2020.

Singh, Pradeep 2017 *Containers in OpenStack* 1st editio ed Packt Publishing

Urban, Holger; Schulz, Tilman & Group, Volkswagen 'OpenStack and Cars-the OpenStack Journey of the Volkswagen Group'.

Xi, Sisu et al. 2015 'RT-Open Stack: CPU Resource Management for Real-Time Cloud Computing' in 2015 *Proceedings - 2015 IEEE 8th International Conference on Cloud Computing, CLOUD 2015* IEEE: 179–186