

Joel Holmi

Pilvipalvelut ja yksityinen OpenStack-ympäristö opiskelukäytössä

Tradenomi
Tietojenkäsittely
Syksy 2019



**KAMK • University
of Applied Sciences**

Tiivistelmä

Tekijä(t): Joel Holmi

Työn nimi: Pilvipalvelut ja yksityinen OpenStack-ympäristö opiskelukäytössä

Tutkintonimike: Tradenomi, tietojenkäsittely, Datacenter-Ratkaisut

Asiasanat: OpenStack, Pilvipalvelut, Google Cloud, VMware, Virtualisointi, AWS, Packstack

Tämän opinnäytetyön tavoitteena oli pystyttää OpenStack-ympäristö Kajaanin Ammattikorkeakoulun DC-laboratorioon. DC-laboratorio on datacenter-opiskelijoille tarkoitettu luokkahuone, jossa sijaitsee konesali. Konesalissa on viisi palvelintelinettä, joissa on verkkolaitteita, tallennustilaa sekä useita laskentapalvelimia. OpenStack toimii yksityisenä pilviympäristönä, joka on tarkoitettu opiskelijoiden ja opettajien käyttöön. Pilvipalvelut ovat jatkuvassa kasvussa ja opinnoissa ei tällä hetkellä ole montaa kurssia, jotka käsittelevät pilvipalvelujen toimintaa. OpenStack toimisi siis hyvänä alustana teknologialle, joka tulee hyvin todennäköisesti monille opiskelijoille työelämässä vastaan.

Työssä on käyty läpi teoriaa pilvipalveluiden toiminnasta ja niiden tilanteesta nykymaailmassa. Pilvipalvelut ovatkin olleet olemassa jo pitkään, mutta läheskään kaikki eivät ole vielä siirtyneet niiden käyttöön. Kasvu pilveen siirtymisessä jatkuu ja kovan kasvun lisäksi siirtymisen suunta näyttää olevan hybridiratkaisuihin. Pilvipalveluiden tilannetta katsottaessa todettiin AWS:n olevan suosituin alusta. OpenStackilla todettiin olevan erittäin pieni osa markkinoista. OpenStack ratkaisuun päädyttiin kuitenkin, sillä käytössä oli jo valmiiksi laitteistoa. OpenStackin asennus ja ylläpito toimivat hyvänä opettelumahdollisuutena.

Työ kertoo OpenStackin ja tarkemmin sen komponenttien toiminnasta. OpenStackin komponenteista Nova on tärkeimmässä roolissa, sillä se hallitsee resurssien virtualisointia. Työssä myös verrattiin eri pilvipalveluita ja vertailu keskittyi palveluiden IaaS-tarjontaan, joka on parhaiten verrattavissa OpenStackiin. IaaS-tarjonnassa on huomattu hyvin paljon samankaltaisuuksia keskenään hinnoittelujen ja resurssien valinnan puolesta. OpenStackia verrattaessa VMwareen todettiin ettei OpenStack missään tapauksessa voisi korvata VMwarea, mutta toisi hyvän vaihtoehdon opiskeluun VMwaren lisäksi.

Käytännön työtä varten tehtiin ensin yksinkertaisempi OpenStack asennus. Testaamisen pohjalta tehtiin alustava arkkitehtuurisuunnitelma, jonka perusteella ympäristö oli tarkoitus toteuttaa. Tuloksena oli alkuperäisestä suunnitelmasta poikkeava ympäristö, jossa hyödynnettiin nykyistä VMware-ympäristöä ja erillisiä fyysisiä palvelimia. Asennus toteutettiin hyödyntämällä Packstack-asennuspakettia, joka automatisoi OpenStackin asennuksen. Tämän lisäksi työssä syntyi kaksi erillistä dokumentaatiota toteutetuista ympäristöistä.

Abstract

Author(s): Joel Holmi

Title of the Publication: Cloud services and a private OpenStack-environment in educational use

Degree Title: Bachelor of Business, Information Technology, Data Center

Keywords: OpenStack, Cloud services, Google Cloud, VMware, Virtualization, AWS, Packstack

The goal of this Bachelor's thesis was to set up an OpenStack environment in a server room located in Kajaani University of Applied Sciences. The server room is located inside a classroom and there are five server racks inside the room. The server racks have computing servers, storage capacity and network hardware inside them. OpenStack works as a private cloud environment that would make use of these resources. Cloud services are constantly growing and there are not many subjects in the KAMK study programme focused on cloud services. OpenStack, in this case, would serve as a learning platform for a technology, that many students will stumble upon in their future jobs.

This thesis goes through the theory background of cloud services and their current situation. Cloud services have been around for a long time already, but we are not even close to where everyone would be using the cloud. The growth of cloud services is fast and steady, while the direction is looking like the go-to solution is hybrid. When examining the situation of cloud services, we found out that AWS is the most popular platform and has been so for a long time. Meanwhile OpenStack was found out to have only a fraction of that popularity in market shares. However, OpenStack was chosen as the solution for various reasons, with one of them being that there was already physical hardware that could be made use of.

OpenStack and its components have been documented and it has been explained how these components work together. The most important piece of OpenStack is Nova, since that is the component in charge of handling virtualization of computing resources. This thesis also compares cloud services offered by different cloud providers and the comparison is focused on IaaS-offerings which are best comparison for this use case of OpenStack. The IaaS comparisons found more similarities than differences when it comes down to resources and pricings. In comparison of OpenStack and VMware, a conclusion was made that it would not be wise to use OpenStack as a replacement for VMware, but it would bring something different to use in studying besides VMware.

In the practical part of this thesis, a simple installation of OpenStack was done for testing purposes. Based on the testing, an architecture plan was made that would server as a base for the installation. The result was a working environment, but it differed from the original plan due to issues that came up during the installations. The final environment made use of the current VMware-environment and used physical servers. The installation was made using an installation package called Packstack, that automates the installation based on given configurations. In addition to the final environment, other results include documentation of the installation and an instructional documentation on how to install OpenStack on a single server.

Sisällys

1	Johdanto	2
2	OpenStack.....	3
2.1	Miksi juuri OpenStack.....	3
2.2	Pilvipalvelut.....	4
2.2.1	Pilvipalveluiden tilannekatsaus	6
2.2.2	Yksityinen pilvi ja OpenStackin paikka	6
2.2.3	Pilvipalveluihin siirtymisen vaikutus	7
2.3	Virtualisointi ja sen hyödyt	7
2.4	OpenStackin komponentit	8
2.4.1	Keystone (Identity Service).....	9
2.4.2	Nova (Compute Service).....	10
2.4.3	Glance (Image Storage Service)	11
2.4.4	Neutron (OpenStack Networking Service)	12
2.4.5	Horizon (Dashboard Service)	12
2.4.6	Cinder (Block Storage Service).....	13
2.5	OpenStack opiskeluympäristönä.....	14
2.6	Vertailu	15
2.6.1	IaaS-Vertailu	16
2.6.2	Käyttöoikeudet ja hallintatyökalut	19
2.6.3	Vertailu nykyiseen VMware-ympäristöön.....	22
3	Ympäristön asennus	23
3.1	Arkkitehtuurisuunnitelma.....	23
3.2	Packstack	24
3.3	Asennusvaihe	25
3.4	Käytännön työn tulokset.....	26
4	Loppuajatukset	28
	Lähteet	30
	Liitteet	
	Liite 1: Packstack-asennus	

Termit ja lyhenteet

Black Book	Black Book on kirjoituspohja datacenter-opiskelijoille, jota voidaan hyödyntää dokumentointiin ja arkistointiin.
DC	Lyhenne sanasta datacenter.
IAM	Identity and access management viittaa saman nimiseen käyttäjähallintapalveluun sekä Googlen että Amazonin pilviratkaisuissa.
Instanssi	Termillä instanssi viitataan pilviympäristöissä oleviin virtuaalikoneisiin.
Komponentti	Komponentilla viitataan yksittäisiin OpenStackin osiin, eli projekteihin.
LDAP	Lightweight Directory Access Protocol on tarkoitettu hyödyntämään hakemistopalveluja kuten Active Directory.
Proof-of-concept	POC on toteutus jostain ideasta, jolla voidaan todentaa sen toiminta.
RAID	RAID on useista kiintolevyistä koostuva kokonaisuus. RAID voi olla virtuaalista tai laiteohjelmistolla toteutettu. Tarkoituksena on tuoda tietojen redundanssia. Voidaan hyödyntää myös saamaan lisää suorituskkyä.

1 Johdanto

Tänä päivänä virtualisointi- ja pilvipalveluratkaisuja on laidasta laitaan. Mutta mihin väliin ratkaisu kuten OpenStack sopii ja mitä käyttötarkoitusta varten se on tehty? Tämä työ käy läpi, miksi päädyttiin käyttämään juuri OpenStackia, varsinkin kun on niin paljon muitakin hyviä vaihtoehtoisia ratkaisuja.

Työn toimeksiantajana toimii Kajaanin Ammattikorkeakoulu. Tehtävänä oli pystyttää OpenStack-ympäristö koulun Datacenter-laboratorioon. DC-laboratorio on luokkahuone, joka on tarkoitettu DC-opiskelijoiden käyttöön. Luokkahuoneessa on tietokoneita opiskelijoiden käyttöön, ja luokassa sijaitsee konesali, jossa on viisi palvelinkaappa. Palvelinkaapeissa on verkkolaitteita, laskentapalvelimia ja tallennustilaa. Konesalissa on oma verkkonsa, jossa on useita aliverkkoja eri käyttötarkoituksiin. Esimerkiksi eri lukuvuosien opiskelijoille on omat aliverkkonsa. Näihin konesalin verkkoihin saa yhteyden myös luokkahuoneen verkosta, joka siis helpottaa konesalin käyttöä suoraan luokkahuoneesta. Osa palvelimista on tarkoitettu tiettyihin käyttötarkoituksiin kuten VMware-ympäristöön ja jaetun tallennustilan käyttöön. Osa palvelimista on opiskelijoiden käytössä, mutta vaatii varauslistan käyttämistä. Palvelimia voi hyödyntää erilaisiin projekteihin, kursseihin ja vaikka opinnäytetyön tekoon, kuten tässä tapauksessa. Käytössä oleva konesali onkin ainutlaatuinen mahdollisuus opiskelijoille päästä oppimaan konesalin käyttöä käytännössä.

Opinnäytetyössä on käyty läpi teoriataustaa OpenStackista sekä pilvipalveluista. Teoriatausta käsittelee pilvipalveluiden tilannetta nykypäivänä ja mihin suuntaan ne ovat menossa. Tarkoituksena oli vertailla OpenStackia muihin ratkaisuihin. Julkisten pilvien vertailu on tehty kolmeen suurimmista kuuluviin pilvipalveluihin, Microsoft Azureen, Google Cloud Platformiin sekä Amazon Web Serviceen. Vertailuosio käy läpi IaaS-ratkaisuja ja eroavaisuuksia niiden välillä. Virtualisointiratkaisuiden osalta OpenStackia on verrattu vain VMware-ympäristöön, koska tarkoituksena oli tuoda OpenStack vaihtoehdoksi koulussa olevan VMware-ympäristön rinnalle. Teorian ja vertailun lisäksi on käyty läpi erilaisia OpenStackin asennustapoja sekä ympäristöjä, jotka tehtiin käytännönsuudessa. Asennuksissa käytetyt OpenStackin komponentit on dokumentoitu perustasolla läpi ja näistä laaditun arkkitehtuurikuvan pohjalta demonstroidaan kyseisen asennuksen toimintaa.

2 OpenStack

OpenStack on pilvien rakentamiseen tarkoitettu joukko avoimen lähdekoodin projekteja. OpenStackin projektit ovat rakennuskomponentteja, joilla voidaan rakentaa tarpeiden mukainen ratkaisu. Osa komponenteista on tärkeitä OpenStackin perustoiminnalle ja osa on valinnaisia riippuen siitä, mitä ominaisuuksia ratkaisulta vaaditaan. Yksittäiset komponentit yleisesti hallitsevat yhtä sille määriteltyä osa-aluetta. Peruskomponentit hallitsevat mm. virtualisointia, verkotusta ja käyttöoikeuksia.

2.1 Miksi juuri OpenStack

OpenStack on hieman erikoinen ratkaisu kaikkien muiden pilvi- ja virtualisointiratkaisujen joukossa, joten halusin lähteä tutkimaan sitä. OpenStack ja kaikki sen komponentit ovat avointa lähdekoodia. OpenStack mahdollistaa oman yksityisen pilven pystyttämisen kaikkien näiden julkisen pilven palveluiden keskellä. Yksityisissäkin pilvissä OpenStackin kohdalla voi valita, haluaako sen pystyttää täysin itse vai haluaako palvelun ostaa vaikka Red Hatilta, joka tarjoaa OpenStack-palveluita. OpenStack voisi mahdollisesti tulla hyödyksi nykyisen opiskeluympäristön rinnalle. Opinnäytetyöstä tulee monta lopputulosta; käytännön työ, ohjeistus, dokumentaatio sekä itse opinnäytetyön kirjoitus. Aiheesta sai siis erittäin laajan kokonaisuuden sekä taatun lopputuloksen.

OpenStackin kustannukset ovat hyvin riippuvaiset valitusta ratkaisumallista. Yleisesti OpenStack on tarkoitettu oman yksityisen pilven rakentamiseen, jolloin kustannukset tulevat olemaan täysin erilaiset kuin julkisessa pilvipalvelussa. Julkisessa pilvessä yleensä maksu hoituu vain käytön mukaan. Jos aletaan pystyttämään omaa ympäristöä, tällöin kustannuksissa tulee ottaa huomioon useita eri asioita. Kustannuksista esimerkiksi täytyy olla henkilöstö, joka ylläpitää omaa rautaa, asentaa OpenStack-ympäristön ja tietysti kustannukset oman raudan hankinnoista.

OpenStack löytyy vielä joidenkin palveluntarjoajienkin käytöstä julkisen pilven palveluna. Nykyiset pilvipalveluiden tarjoajat pystyvät tarjoamaan suhteessa halvalla resursseja käyttöön johtuen suurissa määrissä ostetuista resursseista. Mutta se, kuinka kustannuksissa pysytään budjetissa on loppujen lopuksi käyttäjästä kiinni, sillä pilvessä laskut voivat äkkiä lähteä nousemaan korkeaksi, jos ei ole tarkkana mistä maksaa. Tässä tapauksessa on kuitenkin käytössä jo valmiiksi laitteistoa, johon OpenStack voidaan hyvinkin helposti pystyttää ilman ylimääräisiä

kustannuksia. Näin voitaisiin hyödyntää OpenStackia koulun konesalissa, saataisiin resursseja hyödylliseen käyttöön ja se olisi uusi mielenkiintoinen ympäristö käytettäväksi. Tällä hetkellä opiskelijakäyttöön ei ole hyvin soveltuvaa pilviratkaisua, mutta yleisesti saatavilla on kuitenkin sekä opiskelija- että ilmaiskäyttäjaversioita mm. Amazon Web Servicen (AWS) käyttöön, jonka avulla voi päästä pintaa raapaisemaan. OpenStack voisi myös toimia hyvänä opetus- ja projektialustana siitä kiinnostuneille. Se tuo vaihtoehtoisen ympäristön jo saatavilla olevan opetusympäristön lisäksi. OpenStack tuo myös mahdollisuuden antaa muille opiskelijoille jatkoprojekteja, joissa he voivat lisätä uusia komponentteja OpenStackiin lisäten sen toiminnallisuutta ja hyötyä opiskeluun.

2.2 Pilvipalvelut

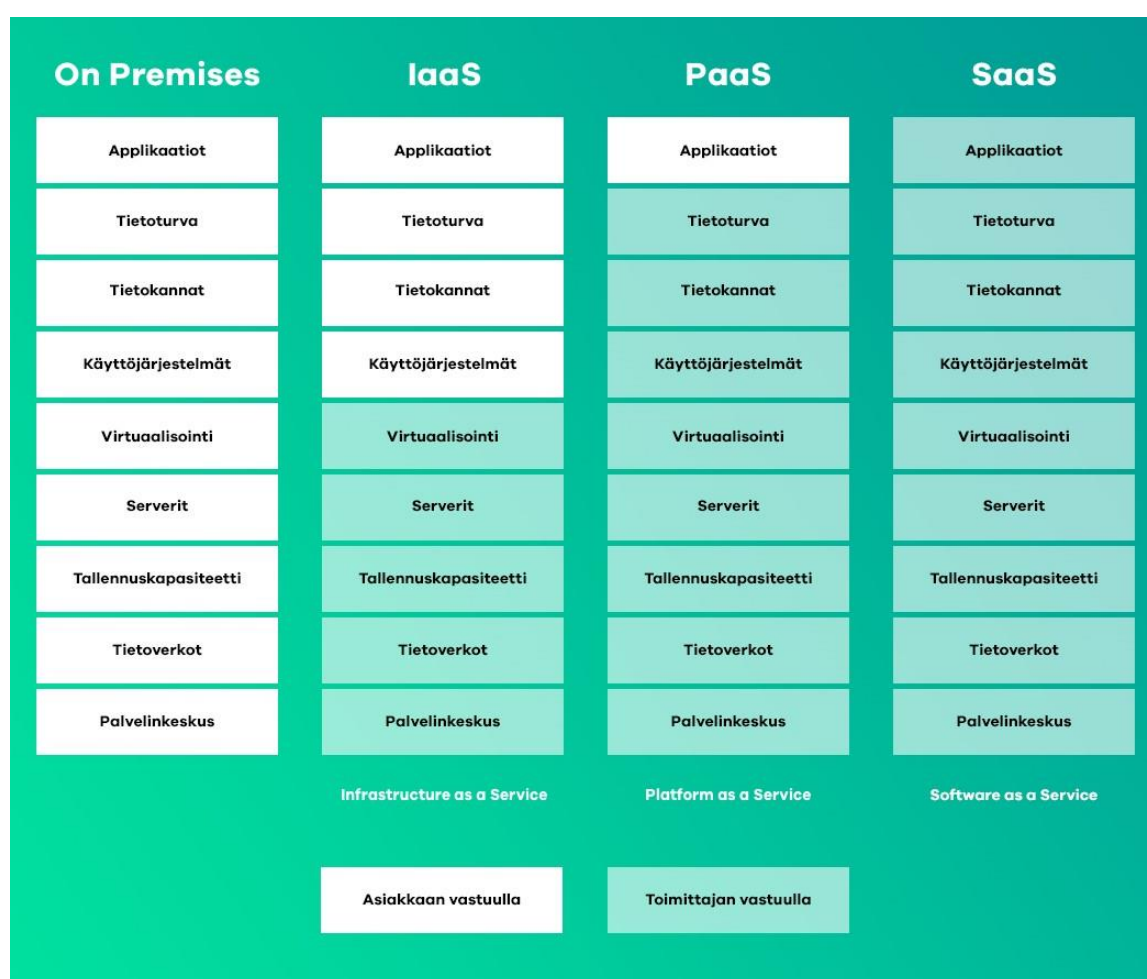
Mitä pilvipalvelut siis ovat? NIST eli US National Institute of Standards and Technology on määritellyt asian hyvin. Pilvipalveluissa täytyy päästä käyttämään resursseja itsenäisesti palveluntarjoajan tarjolla olevista resursseista. Pilvessä täytyy olla laaja verkon saatavuus ympäri maailmaa. Tulee olla suuret määrät resursseja, joihin asiakkaat pääsevät käsiksi, niin ettei asiakkaan tarvitse olla huolissaan rajojen vastaan tulemisesta. Pilven täytyy olla joustava, jolloin voidaan helposti ja nopeasti skaalata palveluita. Ja viimeiseksi pilvessä tulee olla hinnoittelussa vaihtoehtona ”Maksa vain mitä käytät”. Määritelmät nähdään vielä kuvassa 1.

What is cloud computing?



Kuva 1. NIST:n määritelmä pilvipalveluista [1]

Pilvipalveluita on monta eri mallia, ja yleisimmin tunnetut nimitykset ovat Infrastructure as a service (IaaS), Platform as a service (PaaS) ja Software as a Service (SaaS), kuten nähdään kuvasta 2. Suurin ero näissä malleissa on asiakkaalle annetun hallinnan määrä. IaaS tarjoaa palvelimet, mutta jättää asiakkaalle varaa valita, millä alustalla ja mihin tarkoitukseen palvelimet tulevat. IaaS antaa eniten vapautta verrattuna muihin ratkaisuihin. Tyypillisesti se antaa pääsyn myös verkon ja tallennustilan hallintaan. PaaS on taas hieman riisutumpi, mitä tulee hallintaan ja tässä mallissa asiakas vastaa vain ohjelmistoista, joita käyttää. Siinä ei tarvitse huolehtia palvelimista, vaan saadaan suoraan käyttövalmis paketti. SaaS-malli taas on kaikista riisutuim ja käyttötarkoitukseltaankin hyvin erilainen. Hyvänä SaaS-esimerkkinä on vaikka sähköposti. Sitä käyttäessä ei tarvitse huolehtia mistään muusta kuin itse ohjelman käytöstä. [2.]



Kuva 2. Pilven monet kasvot – IAAS, PAAS ja SAAS [3]

Näistä pilvipalvelun malleista OpenStack kuuluu On Premises malliin. On Premises tarkoittaa omassa hallinnassa olevaa ympäristöä, jossa kaikki on toteutettu itse laitteista ohjelmiin asti. Tämän lisäksi OpenStackia voidaan tarjota IaaS-palveluna.

2.2.1 Pilvipalveluiden tilannekatsaus

Pilvipalvelut ovat olleet olemassa jo vuosikausia, ja uusia tekijöitä ilmestyy vieläkin markkinoille. Amazon Web Services (AWS) on vanhin tekijä markkinoilla, ja sen virallinen julkaisu oli jo vuonna 2006. Tämän jälkeen muitakin jättiläisyhtiä on lähtenyt kilpailuun mukaan, kuten Microsoft, Google, IBM ja Alibaba. Edelleen suunta näyttää, että ollaan siirtymässä pilveen enemmän ja enemmän. On tehty jopa ennustuksia, että pilvipalveluiden liikevaihto tulisi olemaan lähes kaksinkertainen vuonna 2021 verrattuna vuoteen 2017 ja suurinta kasvua ovat nähneet SaaS-palvelut vuonna 2018 [4]. Amazon dominoi edelleen IaaS-markkinoita. Lähes 50 % kaikista IaaS-palveluiden markkinaosuudesta on Amazonilla. Ei ehkä niin yllättävä, mutta myös Alibaballa on suuri osa markkinoista ja on nähnyt suurimman kasvun vuonna 2018. Google on pysytellyt paljon jäljessä Amazonia, Microsoftia ja Alibabaa, mutta siitä huolimatta Googlen palvelu on saanut myös suurta kasvua vuonna 2018. Enlyftin keräämän datan mukaan OpenStackin osuus käytetyistä pilvipalveluista olisi vain 0,16 % [5]. [6.]

Syitä tähän pilvipalveluiden kasvamiseen on monia, ja se näkyy selvästi palveluiden suosion kasvussa. Kaikesta kehityksestä halutaan nykyään nopeampaa ja automaattisempaa. Tämäkin nähdään hyvin SaaS-palvelujen suuressa kasvussa. Näiden lisäksi palveluntarjoajat kehittävät jatkuvasti uusia työkaluja, jotka ovat nykyisten IT-trendien mukaisia, kuten tekoälyn hyödyntämistä. Tämä trendien seuraaminen varmasti houkuttelee lisää ihmisiä siirtymään pilveen tai ottamaan tiettyjä pilvipalveluita käyttöön. [6.]

2.2.2 Yksityinen pilvi ja OpenStackin paikka

Mihin väliin siis mahtuu avoimen lähdekoodin palvelu kuten OpenStack? Onko yksityisille pilville enää tilaa, kun kasvulukemien perusteella julkinen pilvi on räjähdysmäisessä kasvussa? Julkisen pilven kovasta kasvusta huolimatta monilla on vielä käytössään perinteisempiä ratkaisuja. Lisäksi julkiseen pilveen siirtymisen ei tarvitse olla kaiken infrastruktuurin siirtämistä yhdestä ratkaisusta toiseen. Yksityinen pilvi on alkanut näyttää enemmän merkkejä siirtymisestä hybridiratkaisuihin. Yritykset siirtävät lähinnä vain joitain palveluita pilveen, ja jotkut ottavat käyttöön hybridipilven, jossa yksityinen ja julkinen pilvi yhdistetään. Hybridiratkaisulla yritykset voivat yhdistellä parhaimmat puolet eri ratkaisuista. OpenStack on löytänyt edelleen suuriakin käyttäjiä kuten Walmart, eikä ole katoamassa mihinkään lähiaikoina [7]. Tämän hetkisen tilanteen perusteella OpenStackilla menee edelleen vahvasti, ainakin mitä tulee yksityisen pilven ratkaisuihin. On vain

odotettava, miten OpenStackin tulee käymään jatkossa. Tullaanko näkemään uusia mullistavia edistyksiä vaikka Red Hatilta OpenStackiin? [8.]

2.2.3 Pilvipalveluihin siirtymisen vaikutus

Miksi palveluita siirretään pilveen ja miten se vaikuttaa yritysten toimintaan? Selvimät erot nähdään pintapuolin palaamalla takaisin kuvaan 2. Pilviratkaisua valittaessa täytyy miettiä kuinka paljon hallintaa halutaan ja paljonko hallintaa on oikeasti tarpeen. Otetaan esimerkkitilanne, jossa yritys on kehittänyt uuden tuotteen ja haluaa nyt viedä sen pilveen. Täytyy vastata useampiin kysymyksiin, kuten kuinka paljon hallintaa tämä tuote vaatii, mitä ominaisuuksia tämä palvelu vaatii toiminnaltaan, vaatiiko palvelu vikasetoisuutta vai voiko palvelun viedä alas ilman huomautusta. Jos halutaan täyttää hallintaa, on yleensä vastaus IaaS. IaaS mahdollistaa yksityiskohtaisen konfiguraation ja tuo näin mahdollisuuden optimoida tämä uusi tuote toimimaan mahdollisimman hyvin. IaaS-käyttö myös tarkoittaa lisävastuuta ja se täytyy tiedostaa ratkaisua valittaessa. Jos tuottelle ei tarvikaan niin tarkkaa säätöä, voidaankin jättää IaaS pois. Nyt aika, joka olisi mennyt IaaS-hallintaan ja konfigurointiin, vapautuukin muita asioita varten. Ei tarvitsekaan enää huolehtia kaikesta infrastruktuurin pystytyksestä ja ylläpidosta, vaan kaikki on toteutettu valmiina SaaS- ja PaaS-palveluina.

2.3 Virtualisointi ja sen hyödyt

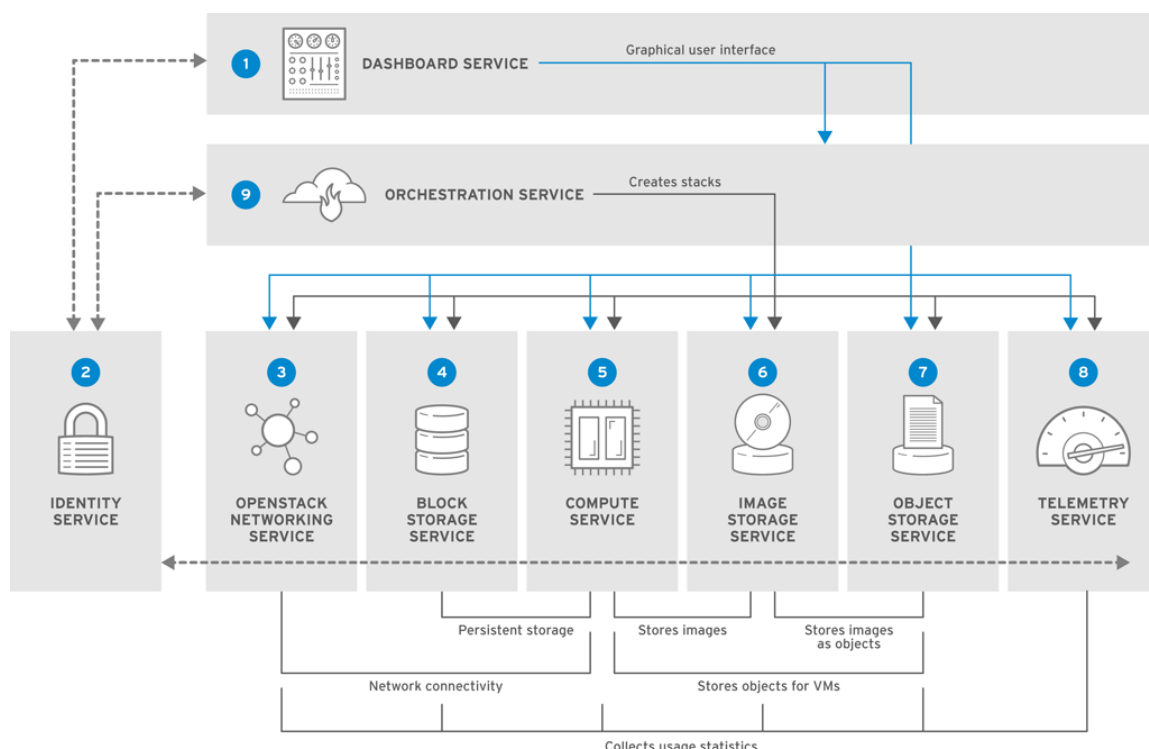
Virtualisointi on tärkeä osa pilvipalveluita ja niiden toimintaa. Virtualisointi yleisesti tarkoittaa fyysisten laitteiden abstraktiota, mutta on olemassa useampia eri virtualisointitapoja. OpenStackissa voidaan esimerkiksi Neutron-komponentin avulla luoda virtuaalisia verkkolaitteita, tarvitsematta fyysisiä verkkolaitteita. Pilvipalveluiden IaaS-ratkaisuissa fyysiset resurssit on virtualisoitu ja näitä resursseja tarjotaan asiakkaille. Tämän ansiosta asiakkaan ei tarvitse välittää, mitä fyysisiä laitteita palvelun takana on.

Myös OpenStack ja nykyinen VMware-ympäristö hyödyntävät virtualisointia. Näistä virtualisoiduista resursseista opiskelijat voivat ottaa resursseja käyttöön luodessaan virtuaalikoneita. Tämä virtualisointi on toteutettu VMwaren ESXi:llä, joka on asennettuna useammalle palvelimelle, joista resurssit saadaan käyttöön. OpenStackin osalta virtualisointi on toteutettu Nova-komponentilla.

Virtualisoinnin hyötynä on parempi resurssien käyttö. Fyysinen palvelin vaatisi erikseen käyttöjärjestelmän asennusta, ja kaikki asennus tapahtuisi samalle koneelle. Virtualisoinnissa kaikki nämä resurssit ovat käytettävissä paljon hajautetummin. Voidaan ottaa samoista resursseista osa käyttöön ja saadaan useampia pienempiä eristettyjä ympäristöjä. Saadaan siis enemmän eristystä, parempi resurssien hyödyntäminen sekä paljon joustavampi käyttö resursseille. Virtualisoinnilla voidaan myös saada muita etuja, kuten virtualisoitu verkkoratkaisu. Ei siis tarvita hankkia uutta laitetta, vaan voidaan luoda virtuaalinen verkko, joka hoitaa saman asian. Verkkojen virtualisointi helpottaa siis huomattavasti verkon hallintaa ja muokkausta. Pilvipalvelut hyödyntävät virtualisointia useammilla eri tavoilla. IaaS-palveluiden lisäksi tarjolla on myös virtualisoitua tallennustilaa ja verkkoratkaisuja.

2.4 OpenStackin komponentit

OpenStack koostuu useammista eri komponenteista. Niistä käydään läpi pääasiallisesti vain tässä työssä käytettäviä komponentteja. Osa komponenteista on erittäin tärkeitä OpenStackin perustoimintojen kannalta, kun taas osa komponenteista tarvitaan vain, jos halutaan käyttää kyseisen komponentin toimintoja. Lopulliseen ympäristöön kuuluvat komponentit ovat Keystone, Nova, Neutron, Horizon, Glance ja Cinder. Yhden palvelimen versioon asennetaan kaikki samat komponentit paitsi Cinder. OpenStackin voi rakentaa monella eri tavalla riippuen siitä, mitä kaikkia komponentteja haluaa asentaa ja kuinka monta palvelinta on käytössä. OpenStack tuo siis erittäin paljon joustavuutta sallien erittäin monta erilaista ratkaisumallia. Kuvassa 3 nähdään vielä Red Hatin laatima kaavio OpenStackin pääkomponenteista ja niiden suhteista toisiinsa. Kuvassa olevat Orchestration(Heat), object storage(swift) ja telemetry service(ceilometer) on jätetty tässä asennuksessa pois.



RHELOSP_347192_1015

Kuva 3. OpenStack-komponenttien arkkitehtuuri [9]

2.4.1 Keystone (Identity Service)

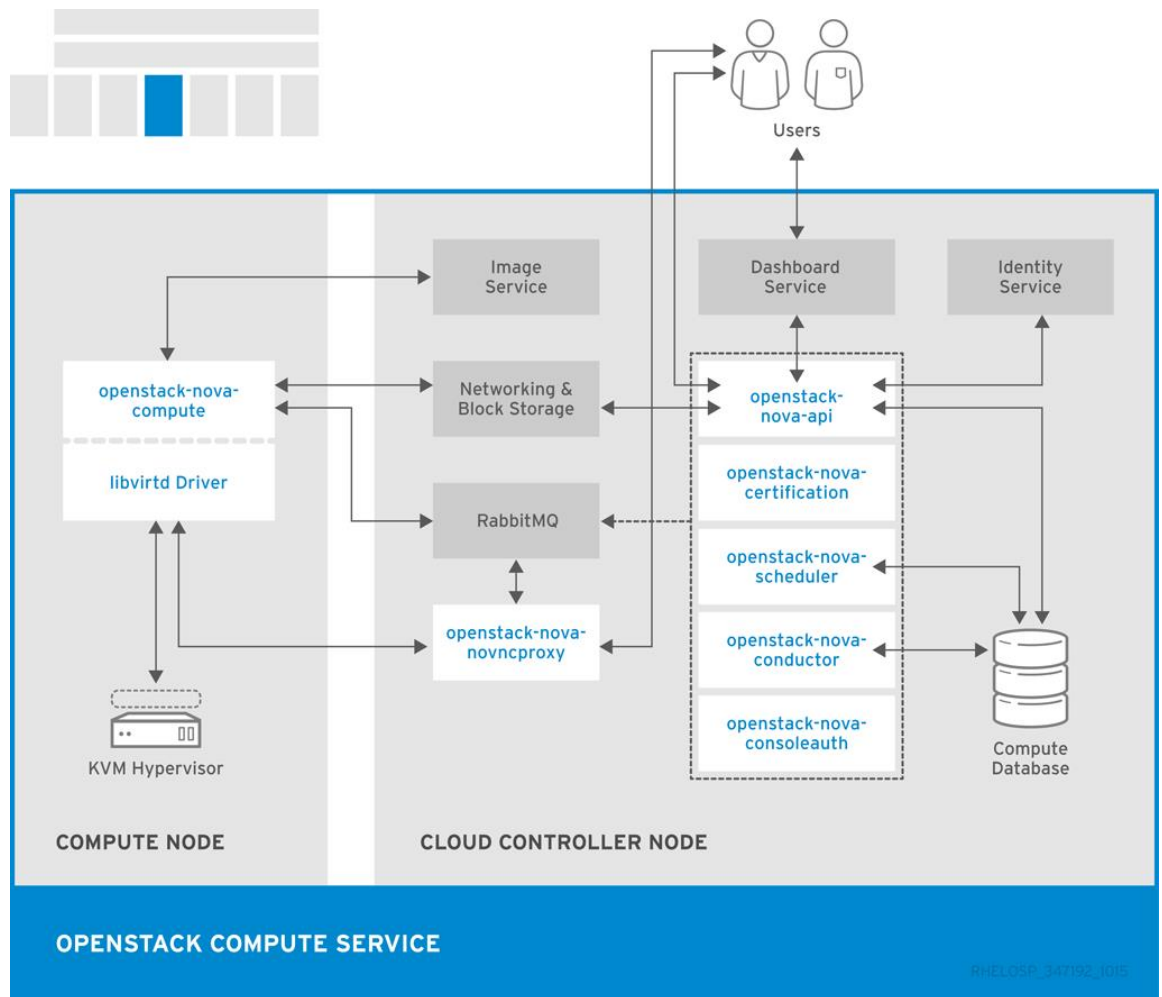
Keystone hallitsee OpenStackin autentikointia, käyttäjienhallintaa ja oikeuksia. Keystone-komponentti asennetaan projektin jokaisessa ympäristössä. Keystone-pääkäyttäjän salasanan voi asettaa manuaalisesti OpenStackin asennuksen yhteydessä, tai voidaan luoda automaattisesti generoitu salasana. Keystone OpenStackissa vastaa AWS:n ja GCP:n IAM-hallintapalveluita.

Keystonen hallintaa voi suorittaa helposti asennuksen jälkeen Horizonin kautta. Uusien käyttäjien luonti sekä käyttäjien oikeuksien hallinta voidaan suorittaa tätä kautta. Käyttäjänhallinnan lisäksi päästään hallitsemaan projekteja, ryhmiä ja rooleja. Yksi projekti pitää allaan oman ympäristönsä, johon voidaan luoda oma verkkonsa ja projektille voidaan määrittää henkilöt tai ryhmä, joilla on oikeudet projektiin.

2.4.2 Nova (Compute Service)

Nova on OpenStackin sydän. Se hallitsee OpenStackin virtuaalikoneita ja luodakseen niitä tarvitsee imaget Glancelta. Nova on OpenStackin IaaS-palvelu, eli se tarjoaa resursseja virtuaalikoneiden käyttöön. Nova voidaan asentaa uusille palvelimille, joita voidaan tarvittaessa lisätä ympäristöön, mahdollistaen ympäristön ylöspäin skaalautuvuuden. Nova vastaa OpenStackissa samaa kuin AWS:n EC2 ja Google Cloudin Compute Engine.

Nova itsessään ei ole hypervisor. Sen sijaan Nova tukee useita hypervisoreita abstraktin kerroksen kautta, kuten vSphere ESXi, Kernel-based Virtual Machine (KVM), Xen, Quick Emulator (QEMU) ja Hyper-V [10, s. 3]. Tässä työssä VMWareen asennetuissa yhden palvelimen ratkaisuihin on käytössä QEMU ja fyysisille palvelimille asennetussa ympäristössä on käytössä KVM. QEMU on emulaattori ja virtualisoija, joka voi myös hyödyntää eri hypervisoreita [11]. KVM on Linuxille tehty virtualisaattoriratkaisu, joka koostuu ladattavasta kernel-moduulista, joka luo pohjan virtualisoinnille [12]. Nova siis toimii vain välikätenä, joka liittää virtualisoinnin muihin OpenStackin komponentteihin. Novan yhteydet muiden komponenttien kanssa nähtävissä kuvassa 4.



Kuva 4. Novan toimintasuhteet muiden komponenttien kanssa [9]

2.4.3 Glance (Image Storage Service)

Glance on OpenStackin komponentti Imageja varten. Glancella hallitaan Imagejen tallennusta sekä käyttöä virtuaalikoneissa. OpenStackia varten olevat imaget ovat yleisesti nopeita käynnistää, ei tarvitse käydä asennusprosessia läpi ja koneeseen päästään kirjautumaan suoraan Secure Shell (ssh)-avainparia käyttäen. Glance tukee myös snapshotteja, eli laitteen sen hetkisen tilan tallennusta. Snapshotit toimivat palautuspisteenä virtuaalikoneille. Glance tukee useita eri image-muotoja, ja niiden käyttö riippuu käytössä olevasta hypervisorista. [13.]

2.4.4 Neutron (OpenStack Networking Service)

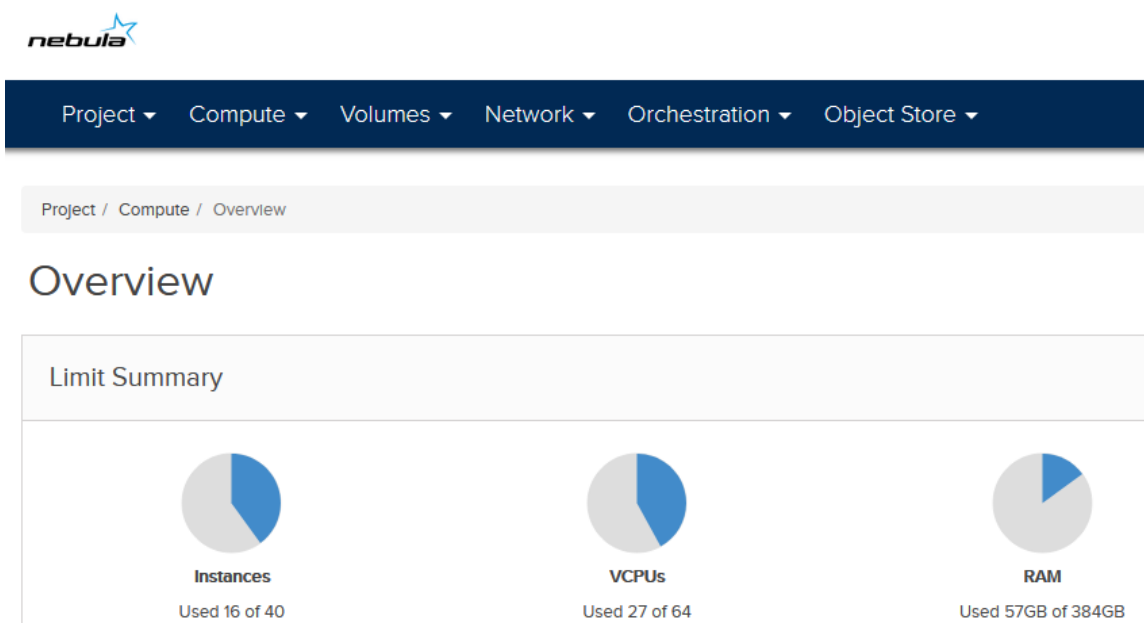
Neutron hallitsee kaiken verkotuksen OpenStackissa. Neutron on komponenttina erittäin monipuolinen. Sillä voidaan esimerkiksi luoda monia virtuaalisia laitteita ja antaa mahdollisuudet hyvin erilaisiin verkkoratkaisuihin. Tässä projektissa Neutronilla kuitenkin luodaan vain yksinkertainen verkko, joka sopii hyvin koulun konesaliin. OpenStackin Neutron vastaa AWS:n ja Google Clouding Networking ominaisuuksia.

Neutron koostuu useammasta eri osasta, jotka ovat neutron server, plugin agent, DHCP agent, L3 agent ja network provider services. Nämä Neutronin eri osat hallitsevat eri verkotuksen alueita OpenStackissa, mutta kaikki nämä osat eivät toimi suoraan palvelimelta, jossa Neutron on asennettu. Neutron server toimisi tyypillisesti verkkopalvelimella, mutta tässä tapauksessa se asennetaan jokaiselle palvelimelle. Neutron server valvoo verkolle asetettua mallia ja IP-osoitteiden jakoa porteille. Neutron server myös suorittaa verkko-API:tä ja sen osia. Plugin-agent on myös jokaisella laskentapalvelimella, jotta ne voivat keskustella muiden laitteiden kanssa. Plugin-agent hallitsee, mitkä agentit ovat käynnissä. DHCP agent nimensä mukaisesti on vastuussa DHCP:n ylläpitämisestä eli IP-osoitteiden jakamisesta virtuaalikoneille. [14.]

Tässä projektissa on käytetty tyyppiä "Flat Network", joka on yksinkertaisin malli ja helpottaa ympäristön käyttöönottoa DC-laboratorion konesaliin. Flat Network tuo virtuaalikoneet saman verkon alle, jossa kaikki OpenStackin isäntäkoneet ovat. Tämä mahdollistaa helpon käytettävyyden ja yhdistettävyyden virtuaalikoneiden, muiden palvelimien ja luokan välillä.

2.4.5 Horizon (Dashboard Service)

Horizon-komponentti on käyttöliittymä, joka on tarkoitettu OpenStackin ja sen muiden komponenttien hallintaan, joka nähdään kuvasta 5. OpenStackin hallinnointia voidaan hoitaa myös komentoriviltä, mutta käyttöliittymä on paljon käyttäjäystävällisempi. Komentorivin hallinta mahdollistaa esimerkiksi työtä nopeuttavien skriptien luomisen. OpenStackille löytyy myös erilaisia automatisaatio-ohjelmistoja, joilla ylläpitoa ja toistettavuutta voidaan parantaa. Horizoniin kirjaudutaan Keystoneen luomilla tunnuksilla, joita voidaan myös hallinnoida Horizonin kautta kirjautumisen jälkeen. Horizonissa voidaan suorittaa kaikki muiden OpenStackin komponenttien perustoiminnot.



Kuva 5. Kuvakaappaus Horizonin yleisnäkymästä Telia Inmics-Nebulan Incloud 9 palvelussa. Joel H. 29.9.2019

Horizon siis keskittää kaiken OpenStackin käyttöön liittyvän toiminnan. Tässä projektissa myös kaikki OpenStackin käyttö tullaan suorittamaan Horizonin kautta. Vain osa konfiguraatioista, joita ei voida suorittaa Horizonin kautta, suoritetaan komentoriviltä. Pääasiallisesti Horizonin kautta voidaan konfiguroida verkot (Neutron), hallita käyttäjiä (Keystone), hallita imageja (Glance) sekä asentaa ja käyttää virtuaalikoneita (Nova). Horizonin konsoli ei ole kuitenkaan kovin käytännöllinen virtuaalikoneiden käyttämiseen, vaan tähän on parempi käyttää muuta ohjelmaa, kuten Puttyä ssh-yhteyden luomiseen. Horizonin kautta voidaankin myös tuoda ssh-avaimia sekä asettaa ssh-avain virtuaalikoneen asennuksen yhteydessä käyttöön. Tämä tuo virtuaalikoneiden asennukseen nopeutta ja hyödyntää ssh-avainten käyttöä suoraan.

2.4.6 Cinder (Block Storage Service)

Cinder on yksi tallennustilan käyttöön tarkoitetuista OpenStackin komponenteista. Tässä työssä Cinderille luotiin konesaliin oma Network File System (NFS)-jako, jonka Cinder ottaa käyttöön ja jakaa eteenpäin OpenStack-ympäristössä. Cinder-volumit toimivat käytännössä hieman kuin ulkoinen kiintolevy, ja ne voidaan lisätä erikseen instansseille. Kaikki voluumille tallennettu data säilyy siellä, vaikka instanssi poistetaan, ellei toisin ole määritelty. Voluumeja voidaan myös poistaa käytöstä ja vaihtaa instanssilta toiselle. Voluumeja voidaan myös käyttää uusien instanssien luomiseen imagejen sijaan. Tämän opinnäytetyön asennuksessa Cinderillä asetettu

tallennustila on huomattavasti nopeampaa kuin itse laskentapalvelimien oma tallennustila. On siis suositeltavaa ottaa aina uuden instanssin luodessa uusi voluumi käyttöön ja käyttää tätä voluumia eri asennuksiin, kuten vaikka johonkin web- tai tietokantapalvelimeen. OpenStackin Cinder vastaa samaa kuin AWS:n Elastic Block Store tai Googlen Cloud Storage palvelu. Voluumeja voi myös hyödyntää monella muulla eri tavalla, esimerkiksi varmuuskopioiden tallennuspaikkana.

2.5 OpenStack opiskeluympäristönä

Koska OpenStack on ilmainen avoimen lähdekoodin joukko ohjelmistoja, tekee se tuotteesta varsinkin tässä tilanteessa houkuttelevan vaihtoehdon. Tilanne on ideaalinen OpenStack-ympäristön pystyttämiseen. Koulun kannalta saataisiin konesalin resursseja hyötykäyttöön alustalle, jota voidaan hyödyntää opinnoissa. Tämän työn palvelimet on dokumentoitu riittävän hyvin, jotta niitä voi jatkossa ylläpitää ja uudelleenpystyttää. Projektin ohella kirjoitin myös yhden palvelimen OpenStack-asennukseen suoran ohjeistuksen, joka on saatavilla DC-laboratorion levyjaosta, josta löytyvät opiskelijoiden kirjoittamat Black Book dokumentaatiot, niille nimetystä kansioista.

VMwareen verrattuna OpenStackilla saadaan useita virtuaalikoneita nopeasti ja vaivattomasti käyttöön. Käytössä on imageja, joissa asennusprosessi on kokonaan ohitettu. Täytyy vain asettaa kirjautumisavain virtuaalikoneelle ja näin virtuaalikoneeseen saa nopeasti yhteyden. IP-osoitteiden jako tulee olemaan hyödyntäen Dynamic Host Configuration Protocol (DHCP), joka myös nopeuttaa ja helpottaa toimintaa. DHCP:n käytön voi kuitenkin myös haluttaessa ohittaa ja luoda valmiita ip-konfiguraatioita, joita voi asettaa laitteille erikseen.

Varsinaista pilvialustaa ei vielä opiskelukäyttöön ole. Tällä hetkellä koululta tarjotaan opiskelijoille tarkoitettuja käyttäjiä ainakin AWS-käyttöön, mutta käyttöoikeudet näillä käyttäjillä ovat hyvin rajoitetut. Tämän projektin tarkoituksena on rakentaa opiskelukäyttöön yksityinen pilvi, jossa käyttöoikeudet eivät ole niin rajatut. Suurin osa palveluntarjoajista tarjoaa kuitenkin ilmaista käyttöaikaa, jolla voi päästä opettelemaan kyseistä ympäristöä (AWS, Azure, Google Cloud).

VMware ESXi hypervisor ja vSphere ovat tällä hetkellä virtualisointiratkaisuna ja ympäristönä opetukseen. Nämä teknologiat ovat hyvä ratkaisu, kun on pienempi määrä isoja palvelimia. Tässä tapauksessa voidaan skaalata ylöspäin lisäämällä uusia fyysisiä palvelimia, joissa on ESXi-

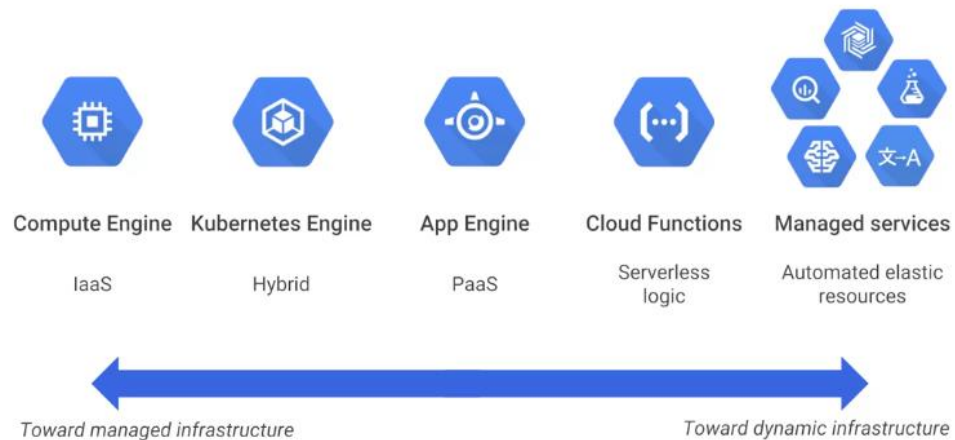
hypervisor. Yksi ongelma, joka tässä tulee vastaan, on jaettu tallennustila, joka tekee ulospäin skaalautumisesta vaikeampaa. [10, s. 2.]

OpenStack eroaa Vmwaresta, sillä se on pohjiltaan pilvialusta. Mikä siis on OpenStackin ja VMwaren ero? OpenStack on suunniteltu sitä varten, että sitä voidaan käyttää useampien eri sovellusten kanssa, jotka on suunniteltu skaalautumaan suoraan. OpenStackin komponentit hyödyntävät erilaisia Application Programming Interfaceja (API) tuomaan joustavuutta alustaan. Pilvialustalla resursseja voidaan saada enemmän lisäämällä sovelluksia ja tasoittamalla virtuaalikoneiden työkuormaa uudestaan. Tällä menetelmällä OpenStack pyrkii luomaan nopeaa skaalautuvuutta. Pilvialustat poistavat tarpeen jakaa kaiken arkkitehtuuripohjaisen päätöksenteon sovellusten välillä. [10, s. 2.]

2.6 Vertailu

OpenStackia on vaikea sijoittaa vertailussa yhteen tiettyyn paikkaan. OpenStackia voidaan verrata vaikka muihin julkisiin pilvipalveluihin, mutta sitä voidaan käyttää myös oman yksityisen pilven rakennukseen ja näin ottaa omia laitteistoja käyttöön. Vertauskohteina OpenStackille toimivat AWS, Azure sekä Google Cloud. Vertailu pyritään pitämään palveluiden tarjoamassa IaaS-tarjonnassa ja siihen liittyvissä palveluissa. Julkiset pilvet tarjoavat myös useita eri SaaS- ja PaaS-palveluita jotka eivät ole suoraan verrattavissa OpenStackiin tai muuten menevät opinnäytetyön rajojen ulkopuolelle. Myös OpenStackista löytyy useita SaaS- ja PaaS-ratkaisuja, mutta niitä ei ole tässä työssä asennettu lopulliseen ympäristöön. Esimerkkinä palveluiden laajuudesta nähdään Google Cloudin palveluita kuvassa 6, josta myös nähdään, kuinka pieni osa IaaS-palvelut lopulta ovat.

GCP computing architectures meet you where you are



Kuva 6. Google Cloud Platform arkkitehtuurit [15]

OpenStackilla voidaan valjastaa konesalin resursseja julkisen pilven käyttöön ja myydä virtuaalikoneita sekä muita palveluja, kuten esim. kuormantasausta. Suomessa vastaavaa palvelua tarjoaa ainakin Telia Inmics-Nebula palvelullaan nimellä INcloud 9. Telia Inmics-Nebula toimii Suomessa AWS:n jälleenmyyjänä. [16.][17.]

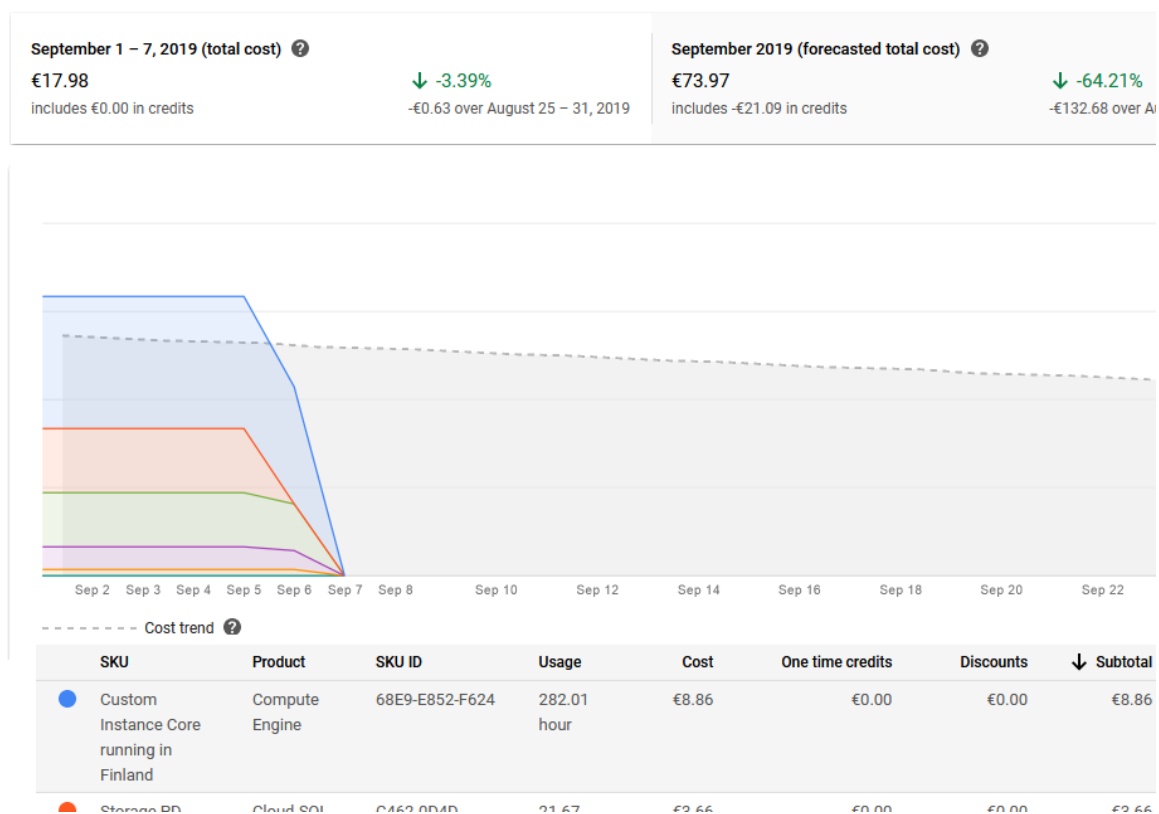
OpenStackille käytetään julkisen pilven esimerkkinä Telia Inmics-Nebulan tarjoamaa INcloud9-ympäristöä. Omalle laitteistolle pystytettyä OpenStack-ympäristöä ei voida suoraan vertailla muihin pilvipalveluihin, mutta toiminnallisuuksiltaan ne ovat hyvin samankaltaisia. Käydään läpi lyhyesti pääominaisuudet, jotka löytyvät kaikista eri palveluista ja löytyykö jostain palvelusta ominaisuuksia, jotka erottavat sen muista.

2.6.1 IaaS-vertailu

IaaS-vertailussa virtuaalikoneiden luomiseen löytyy Googelta Compute Engine, Amazonilta EC2, Microsoftilta Azure ja Suomessa toimiva Telia Inmics-Nebula, joka tarjoaa OpenStackiin pohjautuvaa palvelua nimeltä INcloud9. Jokainen palvelu tarjoaa ilmaista kokeilu-aikaa, joten ominaisuuksia ja niiden toimivuutta pääsee kokeilemaan ja vertailemaan myös käytännössä ennen päätöksen tekoa.

Sekä Azuresta että Google Cloudista löytyy laskin, jolla voi laskea virtuaalikoneista tulevat kustannukset. Laskimien lisäksi löytyy laskunseuranta, joista yhtenä esimerkkinä nähdään

Googlen laskutus kuvassa 7. Laskunseurantaa ei kuitenkaan löydy Telia Inmics-Nebulan tarjoamasta OpenStackista, mutta resurssien käyttöaikoihin löytyy laskin, jolla voi laskea tietyn aikavälin resurssien käytön. Azuressa ja Google Cloudissa on myös vuoden sekä kolmen vuoden pitkäaikaissopimuksen mahdollisuus, jolloin kuukausimaksut ovat alhaisemmat pitkällä aikavälillä. Amazon EC2 tarjoaa hinnoittelussa viittä eri vaihtoehtoa, joten varmasti löytyy sopimus joka tilanteeseen. Nämä ovat esimerkiksi "on-demand" eli kuinka paljon ja kuinka kauan virtuaalikoneita käytetään. Amazon mainostaa "Spot Pricing"-hinnoittelua halvempaan kuin on-demand, ottaen hyötyä käyttämättömistä EC2-resursseista. Spot Pricing-virtuaalikoneita voi käyttää vain rajoitetun ajan, eli ne on enemmän tarkoitettu nopeaan testikäyttöön. Google tarjoaa samanlaisia instansseja, jotka voivat olla korkeintaan päivän käynnissä ja voivat poistua käytöstä ennen aikojaan. Etuna näistä instansseista on se, että ne maksavat vain murto-osan normaalien instanssien hinnasta. Telia Inmics-Nebula tarjoaa OpenStack-ratkaisussaan myös "On-demand-" tai "Prepaid-"maksuvaihtoehtoa. Prepaid-maksuvaihtoehdossa maksetaan staattisista resursseista etukäteen ja mahdollisuudet pitkäaikaissopimukseen löytyvät kolmen kuukauden ja neljän vuoden väliltä. Hinnoittelussa kaikki palveluntarjoajat ovat siis hyvin samankaltaisia, Amazonin tarjotessa eniten maksuvaihtoehtoja ja Azuren tarjotessa halvinta vaihtoehtoa Windows Servereille. [18.][19.][20.][21.]



Kuva 7. Kuvakaappaus Googlen laskunseurannasta. Joel H. 7.9.2019





Verrattaessa virtuaalikoneiden kustomointia ensimmäinen huomio on Telia Inmics-Nebulan tarjoamat vaihtoehdot. Tarjonnasta ei löydy erikseen vaihtoehtoa saada paljon prosessoreita ja vähän RAM:ia, ja jopa heidän tarjoama ”Rakenna oma”-vaihtoehto tarjoaa vähintään 32 Gt RAM:ia, jos halutaan 16 prosessoriydintä [22]. Sama puute toistuu myös Azuren ja Amazonin valmiiden pakettien tarjonnassa. Google Cloudista löytyy tarkka ja kustomoitavin valikoima virtuaalikoneen resurssimäärille. Mutta se mitä Google saavuttaa resurssien kustomoitavuudessa, häviää se resurssien monipuolisuudessa. Googlen Compute Engine tukee Linuxia ja Windowsia, mutta AWS:llä ja Azurella on vielä sitäkin laajempi tuki eri alustoille. AWS sekä Azure tarjoavat esimerkiksi näytönohjain-laskentatehoa virtuaalikoneille. Kaikki palveluntarjoajat, paitsi Telia Inmics-Nebula tarjoavat eri maissa useita konesaleja sekä saatavuusalueita, jotka mahdollistavat korkean saatavuuden. Telia Inmics-Nebulalla on vain kaksi eri konesalia käytössä, joista molemmat näistä sijaitsevat Suomessa. [18.][19.][21.][22.][23.]

2.6.2 Käyttöoikeudet ja hallintatyökalut

Jokaisen palvelun käytettävyys virtuaalikoneiden luontiin on hyvin samankaltaista. Kaikki tarjoavat valmiita asennuskuvia ja ssh-yhteyden laitteille. Virtualikoneen luonti on yleensä vain muutaman klikkauksen takana ja päästään valitsemaan halutut resurssit käyttöön. Hallintatyökaluna jokaisesta palvelusta löytyy web-käyttöliittymät, komentorivityökalut ja useampi Application Programming Interface (API). Näiden lisäksi Google Cloudista ja Azuresta löytyy cloud shell ominaisuus. Se on web-käyttöliittymän kautta käytettävä komentorivityökalu, joka ei vaadi erillistä asentamista.

Eroavaisuudet alkavat tulla palveluissa paremmin esille, kun aletaan pystyttämään isompaa infrastruktuuria. Googlen palvelussa käyttäjienhallinta on omaa luokkaansa, siinä mitä tulee käyttöoikeuksien tarkkuuteen. Esimerkki tästä, kuvassa 8 nähdään vasta korkeimman tason annettavia oikeuksia. Näistä oikeuksista voidaan tarkentaa oikeudet yksittäisiin oikeuksiin, kuten vaikka oikeus ainoastaan nähdä, mitä instansseja yhden projektin alla on. Käyttäjänhallinnan periytyminen on kuvattu Googlen hallinnan osalta kuvassa 9. Googlessa käyttöoikeuksia voidaan määritellä organisaation, kansioden, käyttäjien, ryhmien, roolien ja palvelukäyttäjien avulla. AWS käyttäjienhallinnassa on käytössä ryhmiä, rooleja ja käyttäjiä. Azure hyödyntää hallintaryhmiä, joiden avulla voidaan hallinnoida pääsyä ja käyttöoikeuksia eri resursseihin. Hallintaryhmien lisäksi Azure hyödyntää myös rooleja ja ryhmiä oikeuksien hallintaan. OpenStackissa käyttäjienhallinta-komponentti on nimeltään Keystone. Keystoneen käyttäjienhallinnassa käytetään käyttäjiä, ryhmiä, projekteja sekä domaineja, joista domain on korkein asetetus jonka alle muut säännöt rakennetaan. [24.][25.][26.][27.]

Edit permissions

Member	Project
<div>Role</div> <div>Cloud Build Service Accou... ▼</div> <div>Can perform builds</div>	
<div>Role</div> <div>Cloud SQL Client ▼</div> <div>Connectivity access to Cloud SQL instances.</div>	
<div>Role</div> <div>Kubernetes Engine Develo... ▼</div> <div>Full access to Kubernetes API objects inside Kubernetes Clusters.</div>	
<div>Role</div> <div>Source Repository Writer ▼</div> <div>Read / Write access to repositories</div>	
+ ADD ANOTHER ROLE	

SAVE

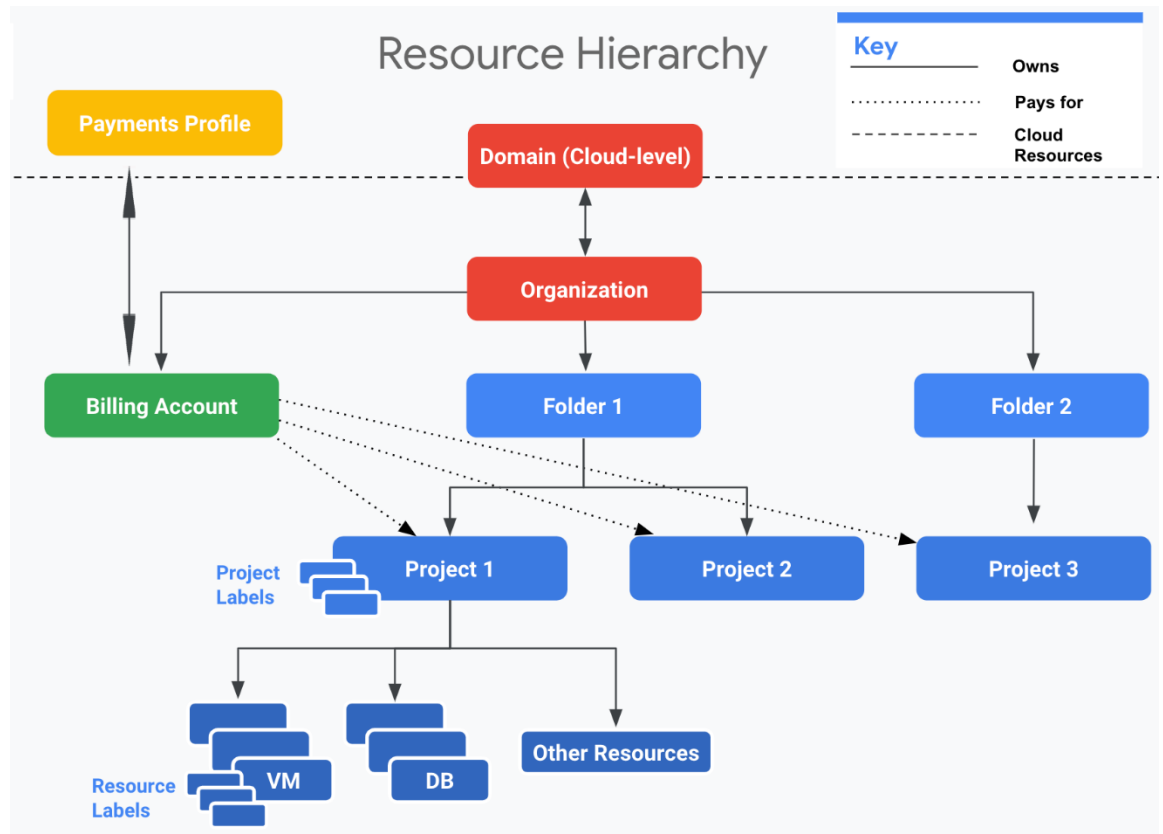
CANCEL

Kuva 8. Kuvakaappaus GCP-oikeuksienhallinnasta, IAM & Admin. Joel H. 7.9.2019.

Näistä vertauksista voidaan huomata jokaisessa ympäristössä toistuvat ominaisuudet, jotka ovat käyttäjät, ryhmät ja roolit. Ne toimivat samoilla periaatteilla jokaisen palveluntarjoajan ratkaisussa. Käyttäjä tarvitsee oikeuksia, ja oikeuksia ei välttämättä kannata asettaa yksi kerrallaan. Oikeuksia kannattaa asettaa roolien muodossa, joille on asetettu oikeuksia. Rooleja on valmiita, ja ne määrittelevät yleensä useampia eri käyttöoikeuksia, mutta lisäksi voidaan luoda omia rooleja, jotka täydentävät valmiita rooleja. Näiden ominaisuuksien on tarkoitus tehdä käyttäjänhallinnasta tehokkaampaa.

Google Cloud Platformin (GCP:n) hallintaa varten luodaan projekteja. Yhden projektin alla on omat resurssit käytössään, mutta useampi projekti voi olla sidottuna saman laskutuksen alle, joka voidaan nähdä kuvassa 9. GCP:ssä voidaan luoda organisaatiorakenteita, joissa on kansioita, ja

kansioissa on niihin luodut projektit. Projektien oikeudet ja toimintatavat ovat siis hyvin samankaltaiset OpenStackin kanssa. Tällä rakenteella oikeuksia voidaan jakaa ylhäältä alaspäin periytyvästi. GCP:ssä vain projektit ovat vaadittu rakenne, mutta ylemmän tason rakenteet voivat helpottaa asetusten organisointia isommassa skaalassa. [28.]



Kuva 9. Google Cloud Platform - resurssihierarkia [29]

Perustasolla löydetään hyvin paljon samankaltaisuuksia eri palveluiden välillä. Kun lähdetään tutkimaan palveluita pidemmälle, jää Telia Inmics-Nebulan OpenStack ratkaisu kuitenkin hyvin nopeasti jälkeen muiden palveluiden tarjonnasta. Virtuaalikoneiden seuranta, logitus ja hintojen seuranta ovat toista luokkaa julkisen pilven palveluissa. Sen lisäksi IaaS-palveluiden ulkopuolella OpenStackista ei ole vastusta näille julkisen pilven jäteille. Loppukysymys onkin, tarvitseeko näitä lisäominaisuuksia, joita ei ole välttämättä suoraan saatavilla OpenStackiin, vai onko OpenStackin oikea hyöty juuri oman yksityisen pilven pystyttämisessä?

2.6.3 Vertailu nykyiseen VMware-ympäristöön

Tämän opinnäytetyön kirjoitushetkellä koulussa on käytössä VMware-ympäristö, eikä olisi järkevää korvata sitä kokonaan OpenStackilla. OpenStack, kuten moni muukin vaihtoehtoinen ratkaisu, voi tuoda kuitenkin uudenlaisen opiskeluympäristön. OpenStackille on olemassa esimerkiksi vaihtoehto nimeltään ”VMware Integrated OpenStack”. Tätä voitaisiin vaikka hyödyntää suoraan olemassa olevan VMware-ympäristön päälle. Tämän tarkoituksena on siis saada OpenStackin ominaisuuksia käyttöön VMware-ympäristössä. Tässä projektissa se ei ole kuitenkaan haluttu lopputulos, vaan sen sijaan halutaan itsenäinen OpenStack-ympäristö. Tällä ympäristöllä otetaan käyttämättömiä resursseja hyötykäyttöön. OpenStack-ratkaisuun päädyttiin, koska käytössä oli jo valmiiksi fyysisiä resursseja, joita ei ollut hyödynnetty vielä muuhun käyttöön.

Oman käyttökokemuksen perusteella OpenStackista on ollut hyötyä esimerkiksi, jos aloittaa uutta projektia, jossa tarkoituksena on rakentaa klusteri tai muusta syystä tarvitaan suuri määrä virtuaalikoneita. OpenStackilla on huomattavasti nopeampi saada useampi virtuaalikone käyttövalmiiksi kuin käytössä olevalla VMwarella. Tämä opiskelun kannalta kuitenkin jättää olennaisia askelia pois, jotka opiskelijan on hyvä tietää, kuten vaikka käyttöjärjestelmän asennus. Mutta nämä eroavaisuudet ovat syy siihen, miksi useampi opiskeluympäristö olisi hyvä olla. Esimerkkinä molempia ympäristöjä voidaan hyödyntää eri käyttötarkoituksiin. Yhdessä ympäristössä voidaan opetella perusasiat paremmin, ja toisessa ympäristössä voidaan kiihdyttää vaikka projektityöskentelyä, tai opetella pilvipalvelun käyttöä, joka vastaa hyvin paljolti toimivuudeltaan isoja julkisten pilvien palveluita.

3 Ympäristön asennus

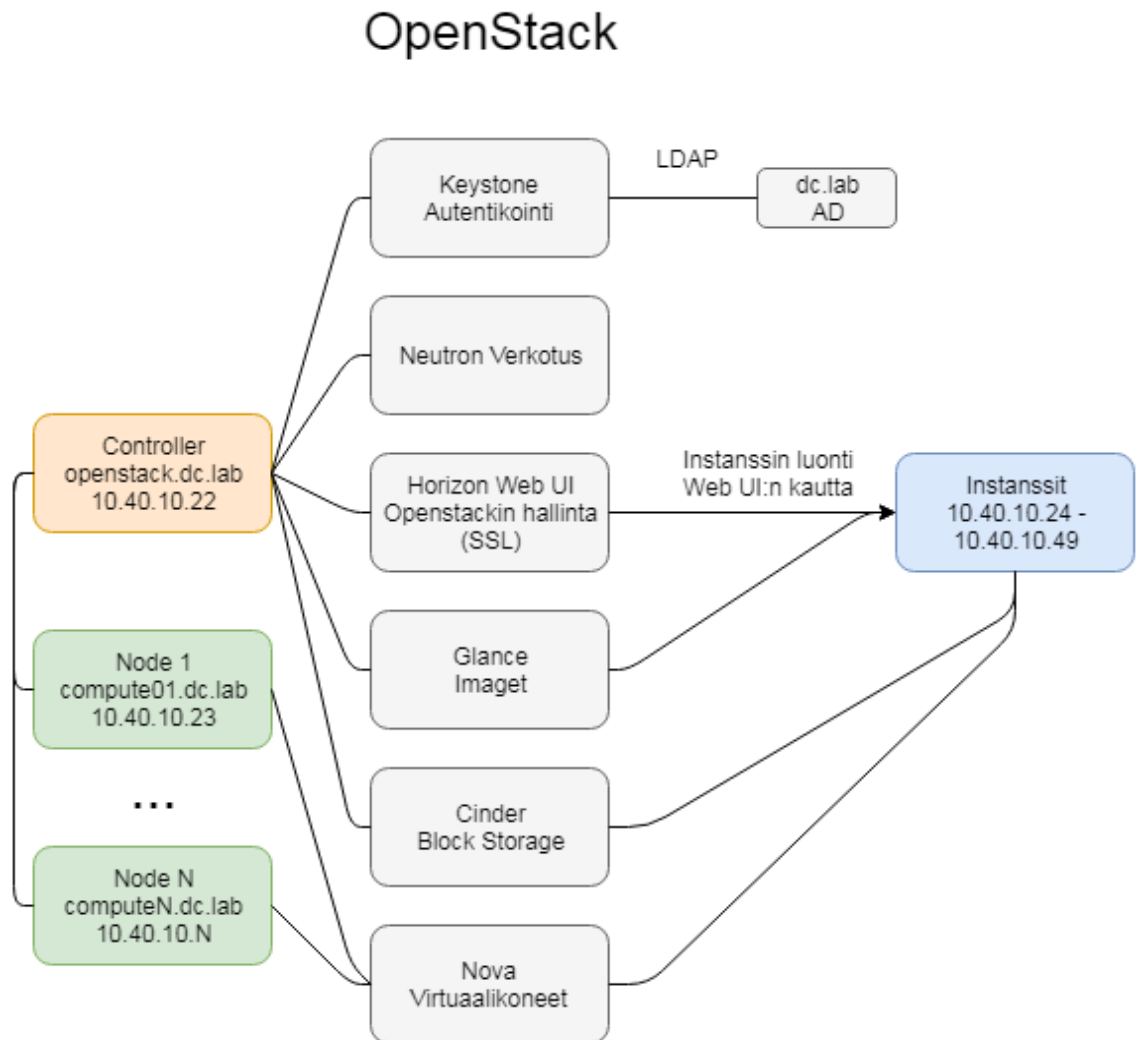
Käytännön työn tavoitteena oli saada fyysisten palvelimien resurssit virtualisoituna käyttöön OpenStackin kautta. OpenStackin asennus manuaalisesti on erittäin pitkä ja monimutkainen prosessi, mutta tämän korjaavat valmiit asennuspaketit. Asennuspaketeilla OpenStackin asennus on pitkälle automatisoitu. Asentajan ei tarvitse huolehtia muusta kuin alustavasta ympäristön konfiguroinnista, viimeistelystä ja ylläpidosta. Asennuspaketteja OpenStackille löytyy useampia, ja tässä opinnäytetyössä on valittu käytettäväksi asennuspaketti nimeltä "Packstack". Asennuspaketit helpottavat OpenStackin asennusta ja tekevät asennuksesta lähes vaivattoman. Packstackille annetaan asennuskonfiguraatio, jonka perusteella Packstack suorittaa asennuksen. Asennukset on tehty soveltaen Packstackin dokumentaatiota ja Keith Tenzerin kirjoittamaa OpenStack-asennusohjetta [30][31]. Näistä materiaaleista on muokattu tähän projektiin sopivat asennustavat, ottaen huomioon käytössä olevan ympäristön.

3.1 Arkkitehtuurisuunnitelma

Projektin testausvaiheessa laadittiin alustava arkkitehtuurisuunnitelma, joka on nähtävissä kuvassa 10. Suunnitelmaan kirjattiin kaikki asennettavat komponentit, alustavat IP-osoitteet ja komponenttien toimintasuhteet. Ei ollut varmuutta Lightweight Directory Access Protocol (LDAP) ja Secure Sockets Layer (SSL) asennuksista lopulliseen ympäristöön, mutta ne jätettiin kuitenkin suunnitelmaan. LDAP mahdollistaisi kirjautumisen Active Directory (AD)-tunnuksilla OpenStackiin ja helpottaisi käyttäjienhallintaa. LDAP sallisi nykyisten opiskelijatunnusten käytön suoraan OpenStackissa. Nykyiseen VMware-ympäristöön voidaan myös kirjautua AD-tunnuksilla. Palvelinten oma tallennustila on varsin hidasta, joten tarkoituksena on ollut saada ulkoista tallennustilaa käyttöön Cinderin avulla. Suunnitelman IP-osoitteet eivät ole lopullisia, vaan kehityskäyttöön tarkoitettuja, ja saattavat olla täysin erit lopullisessa ympäristössä. Laskentapalvelimia voidaan lisätä tarpeen mukaan, mutta tässä tapauksessa sovittiin kahden palvelimen riittävän ainakin aluksi tähän tarkoitukseen.

Arkkitehtuurisuunnitelman kuvasta 10 nähdään, kuinka asennus oli tarkoitus toteuttaa. Hallintapalvelimelle asennettiin Keystone, Neutron, Horizon, Glance ja Cinder. Laskentapalvelimille

asennettiin Nova ja Neutron. Hallintapalvelin hallinnoi muita palvelimia, ja kaikki liikenne OpenStackiin kulkee hallintapalvelimen kautta. Laskentapalvelimien tarkoituksena on jakaa resurssit käyttöön virtuaalikoneille.



Kuva 10. OpenStack-ympäristön alkuperäinen arkkitehtuurisuunnitelma

3.2 Packstack

Packstack on asennuspaketti, joka on tarkoitettu automatisoimaan OpenStackin asennusta. Packstack konfiguroidaan kevyeksi, ja jätetään asentamatta mitään ylimääräisiä komponentteja, joita ei tulla tarvitsemaan. On tärkeää myös konfiguroida Packstack oikein ennen asennusta, jotta vältetään ongelmia myöhemmissä vaiheissa. Packstack on tarkoitettu proof-of-concept-ympäristöjen asentamiseen, mutta voidaan myös, tässä tapauksessa, soveltaa isommankin

ympäristön luontiin. Packstack hyödyntää ohjelmistoa nimeltään Puppet, joka on tarkoitettu automatisaatioon. Packstackissa ei myöskään ole mukana mitään ylimääräistä, ja sillä saadaan kaikki tarvittavat OpenStackin komponentit asennettua tämän työn puitteissa.

3.3 Asennusvaihe

Asennus on suoritettu CentOS 7 käyttöjärjestelmällä ja OpenStackin versiolla 14 (Rocky). Asennuksesta löytyy tarkempi dokumentaatio lopullisesta OpenStack-ympäristöstä liitteenä. Asennusta lähdettiin tekemään arkkitehtuurisuunnitelman pohjalta. Esivaiheina varattiin tarvittavat IP-osoitteet sekä tarvittavat laitteet käyttöön. Luotiin myös alkuladattava CentOS 7 USB-tikku. Näillä esivaiheilla päästiin aloittamaan asennusta. Fyysisillä palvelimilla on käytössään erittäin hidasta tallennustilaa, joten hallintapalvelimella käytössä olevista kahdesta levystä luotiin RAID 0, joka nopeuttaa levyjen toimintaa. RAID 0 ei tuo minkäänlaista vikasietoisuutta ympäristöön, mutta tuo sen verran nopeutta tallennustilaan, jotta ympäristöä on siedettävä käyttää. RAID:n luonti tapahtui palvelimien BIOSin kautta, johon pääsee käsiksi palvelimen käynnistuksen yhteydessä. Näillä palvelimilla RAID:n luominen vaati myös laitteistokohtaisen ohjelmiston asennuksen. Tiedot aiheesta löytyvät paremmin laitekohtaisista dokumentaatioista.

Seuraavaksi asennettiin käyttöjärjestelmä. Käyttöjärjestelmän asennuksessa käytettiin alkuladattavaa USB-tikkua laskentapalvelimille. USB-tikku laitettiin fyysiseen palvelimeen kiinni, ja palvelin konfiguroitiin käynnistymään USB-tikulta. Käynnistuksen yhteydessä palvelimen näytöllä näkyy ohje, joka kertoo, mitä nappia täytyy painaa, päästäkseen valitsemaan käynnistysvaihtoehdon. Käyttöjärjestelmää asennettaessa annettiin järjestelmälle valmiiksi valittu IP-osoite ja hostname, kuten kuvassa 10. Näiden lisäksi palvelimille asetettiin näppäimistön kieleksi suomi, ja annettiin root-käyttäjälle salasana. Ei luotu kuitenkaan uutta käyttäjää, koska asennus tultiin suorittamaan root-käyttäjänä. Käyttöjärjestelmällä piti tehdä vielä muutamia muokkauksia ennen itse OpenStack-asennuksen aloittamista. Käyttöjärjestelmässä asetettiin palvelimien nimet ohjaamaan oikeisiin IP-osoitteisiin, otettiin firewall- ja NetworkManager-palvelut pois käytöstä. Sama asennus toistettiin jokaiselle palvelimelle.

OpenStack asennuksessa käyttöjärjestelmälle ladattiin Packstack-ohjelma. Packstack-asennusta varten luotiin asennuskonfiguraatio, jolla määriteltiin haluttu ympäristö. Packstack-konfiguraatiossa otettiin pois käytöstä kaikki turha, mitä ei haluta asennukseen, asetettiin kaikki IP-asetukset oikein ja varmistettiin, että halutut komponentit tulivat asennetuksi. Ennen kuin

asennusta suoritettiin, tuli Cinder-komponentille luoda levyjako. Packstack hoitaa itse OpenStack-asennuksen, sille annettujen konfiguraatioiden pohjalta.

Packstack asennuksen jälkeen kaikki ei kuitenkaan ole vielä valmista. Tässä vaiheessa täytyi vielä konfiguroida nettiasetukset oikein. Nettiasetukset tulee konfiguroida joka palvelimelle. Lisäksi Cinderin ja Novan asetukset tulee päivittää. Lopuksi päästiin luomaan verkko. Verkon voi luoda joko komentoriviltä tai jo suoraan OpenStackin käyttöliittymän kautta. Verkoksi luotiin flat-tyyppinen verkko, joka yhdistettiin aiemmin konfiguroituun siltaan, joka yhdistää virtuaalikoneet samaan verkkoon fyysisten palvelimien kanssa.

3.4 Käytännön työn tulokset

Aina kaikki ei mene tarkalleen suunnitelmien mukaan. Tässä tapauksessa haluttuun lopputulokseen ei päästy täydellisesti. OpenStackin komponentti Cinder koitui isoimmaksi haasteeksi projektissa. Ennen varsinaisen arkkitehtuurisuunnitelman tekoa oli olemassa yhden palvelimen toimiva ympäristö, jossa myös Cinder toimi. Varsinaista useamman palvelimen ympäristöä toteutettaessa ongelmia tuli, kun yritettiin saada tallennustilaa toimimaan Cinderin kanssa. Päädyttiin jättämään alkuperäinen tallennustila pois. Sen sijaan luotiin NFS-jako konesaliin, josta Cinder sai tallennustilaa käyttöönsä. Toiseksi kompastuskiveksi osoittautui useamman palvelimen verkon konfigurointi. Ongelma saatiin ratkaistua asentamalla Neutron myös hallintapalvelimelle ja konfiguroimalla Neutron oikein keskustelemaan laskentapalvelimien kanssa.

Loppujen lopuksi virtuaalikoneiden luonti saatiin toimimaan lähes mutkitta. Cinder saatiin myös toimimaan ja virtuaalikoneita pystyi käynnistämään suoraan Cinderin tallennustilan päälle. Ihan ongelmitta ei kuitenkaan selvitty, ja ssh-avaimien käytön kanssa ilmaantui ongelmia. Avaimet onnistui lisätä OpenStack-instansseille, mutta yhdistäessä annetulla ssh-avaimella instanssi hylkää avaimen. Ilman ssh-avainta yhdistäminen onnistui, mutta tällöin tarvitsisi imagen jossa käyttäjä ja salasana on ennestään tiedetty.

Lopputuloksena oli siis OpenStack-ympäristö, joka toimi muuten kaikin puolin paitsi ssh-avainten kanssa esiintyneen uuden ongelman kannalta. Ympäristöä ei sellaisenaan ole erittäin hyödyllistä ottaa käyttöön ja ympäristö poistui DC-laboratorion konesalin uudistusten yhteydessä. Ympäristöstä on kuitenkin vielä olemassa dokumentaatiot, joiden pohjalta sen voi pystyttää uudeksi ja käytetty Packstack-ohjelma tekee OpenStackin uudelleen asennuksen helpoksi.

Lopullisessa OpenStack-ympäristössä oli kolme palvelinta; hallintapalvelin ja kaksi laskentapalvelinta. Hallintapalvelinta varten hyödynnettiin jo valmista VMware-ympäristöä, eli hallintapalvelin on siis virtuaalikone eri ympäristössä. Hallintapalvelimella on IP-osoite samassa aliverkossa kuin laskentapalvelimilla. Kahden fyysisen palvelimen resurssit virtualisointiin hyödyntämällä Nova-komponenttia, joka käyttää virtualisointiratkaisuna KVM:ää. Neutronin osalta virtualisointia ei hyödynnetty, vaan yhdistettiin suoraan jo olemassa olevaan aliverkkoon. Ympäristö vaatii ylläpitoa, kuten käyttöjärjestelmien asennuskuvien päivittämistä. Ympäristö on luotu on-premises tavalla, mutta toimii käytännössä IaaS-palveluna muiden käyttöön. Opiskelijat voivat siis luoda virtuaalikoneita ympäristössä tarjolla olevista resursseista.

4 Loppuajatukset

Alun perin halusin aloittaa opinnäytetyön teon ajoissa. Sain koululta projektiksi OpenStack-ympäristön asennuksen, ja tästä projektista sain itselleni opinnäytetyöaiheen. Loppujen lopuksi opinnäytetyön kirjoittaminen venyi, sillä oli vaikea löytää aikaa työn kirjoittamiseen. Aihe osoitautui myös erittäin mielenkiintoiseksi aiheeksi, joka tuli myös työelämässä vastaan. Sain työssäni uutta tietoa OpenStackista, sillä käytössäni oli OpenStack-ympäristö, jota hallinnoin. OpenStackin lisäksi pääsin työssäni tutustumaan myös Googlen pilvipalveluun, ja tätä kautta sain lisää hyviä ideoita opinnäytetyön kirjoittamiseen.

Käytännön työn tavoitteena oli saada toimiva ja käytännöllinen opiskeluympäristö pystyyn. Matkan varrella tuli useita haasteita vastaan. Ympäristöä jouduttiin asentamaan uusiksi useita kertoja. Lopulliseksi ongelmaksi osoittautui konesalissa oleva tallennustila, joka haluttiin saada käyttöön. Tästä huolimatta ongelma saatiin ratkaistua käyttämällä eri tallennustilaa Cinderille, ja ympäristö oli lopulta pystyssä. OpenStackin hallintapalvelin asennettiin lopulta VMware-ympäristöön virtuaalikoneelle. Monista komplikaatioista johtuen ympäristö ei ole laajemmassa käytössä. Lopputuloksena oli kuitenkin kirjoittamani dokumentaatio yhden palvelimen OpenStack-asennuksesta. Lisäksi työstä kertyi laaja dokumentaatio lopulliseen asentamaani ympäristöön.

Vaikka haluttuun lopputulokseen ei päästy niin kuin olisin toivonut, niin opin kuitenkin erittäin paljon tärkeitä asioita tulevaa varten. Opin käyttämään Linuxia ja ssh-yhteyksiä luontevammin, opin vianetsintää ja logien hyödyntämistä paremmin. Opin paljon myös verkkojen ja eri tallennustilan ratkaisujen toiminnasta. Opin myös yhdistelemään fyysisiä laitteita ja virtuaalikoneita rakentaen yhteisen ympäristön, joka toimi samassa aliverkossa. Opin pilvipalveluiden toiminnasta, käyttäjienhallinnasta, verkotuksesta ja monista pilviteknologioista. Koko opinnäytetyöprosessi on toiminut hyvin opettavaisena ja paljon tiedonhakua vaativana projektina.

OpenStackille voi olla vaikea löytää paikkaa, kun saatavilla on niin paljon muitakin hyviä vaihtoehtoja. Tässä tapauksessa osoitettiin yksi tapaus, jossa OpenStackista saatiin oikeasti hyötyä irti. Saatavilla oli jo valmiiksi laitteistoa, eikä asennuksesta koituisi mitään ylimääräisiä kustannuksia. OpenStackin ainoina etuina näen yksityisen pilven, jossa halutaan laitteiston ja datan hallinta pitää omissa käsissä. Oman hallinnan lisäksi kustannukset voidaan pidemmällä aikavälillä saada mataliksi, kun käytetään omaa laitteistoa. OpenStackia on edelleen hyvin vaikea mennä suoraan verrannollistamaan mihinkään tiettyyn ratkaisuun. Jokaisen on tehtävä oma

päätös tarpeidensa ja käyttötarkoituksensa mukaan. Ei ole olemassa vain yhtä ja oikeaa pilviratkaisua, ja tämäkin opinnäytetyö raapaisee vain pilvipalveluiden pintaa IaaS-palveluiden osalta.

Lähteet

- 1 What is Cloud computing. Coursera 2019. Saatavilla: <https://www.coursera.org/learn/gcp-fundamentals/lecture/tnlSM/what-is-cloud-computing> Viitattu: 20.10.2019
- 2 Types of Cloud Computing, Amazon Web Services. 2018. Saatavilla: <https://aws.amazon.com/types-of-cloud-computing/> Viitattu: 24.10.2018
- 3 Pilven monet kasvot – IAAS, PAAS ja SAAS. Telia Inmics-Nebula 27.4.2018. Saatavilla: <https://www.nebula.fi/blogi/pilven-monet-kasvot-iaas-paas-ja-saas> Viitattu: 24.10.2018
- 4 Gartner forecasts worldwide public cloud revenue to grow 17.3 Percent in, Gartner. 2019. Saatavilla: <https://www.gartner.com/en/newsroom/press-releases/2018-09-12-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2019> Viitattu: 29.9.2019
- 5 Companies using OpenStack, Enlyft. 2019. Saatavilla: <https://enlyft.com/tech/products/open-stack> Viitattu: 7.10.2019
- 6 Worldwide IaaS public cloud services market grew by 31.3% in 2018, Gartner. Saatavilla: <https://www.information-age.com/iaas-public-cloud-services-market-gartner-123484377/> Viitattu: 29.9.2019
- 7 What's the reality of OpenStack and public cloud? Red Hat Blog. 2018. Saatavilla: <https://www.redhat.com/en/blog/whats-reality-openstack-and-public-cloud> Viitattu: 29.9.2019
- 8 OpenStack can't carry the private cloud without AWS, Infoworld, Matt A. 2015. Saatavilla: <https://www.infoworld.com/article/2919272/openstack-cant-carry-the-private-cloud-without-aws.html> Viitattu: 29.9.2019
- 9 OpenStack architecture guide. Red Hat OpenStack Platform 2019. Saatavilla: https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/11/html/architecture_guide/components Viitattu: 13.9.2019
- 10 Adding Speed and Agility to Virtualized Infrastructure with OpenStack, OpenStack Foundation. 2015. Saatavilla: <https://www.openstack.org/assets/pdf-downloads/virtualization-Integration-whitepaper-2015.pdf> Viitattu: 15.10.2018
- 11 QEMU. 2019. Saatavilla: https://wiki.qemu.org/Main_Page Viitattu: 13.9.2019

- 12 KVM, Red Hat. 2019. Saatavilla: https://www.linux-kvm.org/page/Main_Page Viitattu: 13.9.2019
- 13 Architecture Guide - Components, Red Hat. 2019. Saatavilla: https://access.redhat.com/documentation/en-us/red_hat_openshift_platform/11/html/architecture_guide/components#comp-image Viitattu: 8.9.2019
- 14 Networking architecture, OpenStack Project. 2018. Saatavilla: <https://docs.openstack.org/security-guide/networking/architecture.html> Viitattu: 26.10.2018
- 15 Google Cloud Platform computing architectures. Coursera 2019. Saatavilla: <https://www.coursera.org/lecture/gcp-fundamentals/gcp-computing-architectures-qWhgM> Viitattu: 8.9.2019
- 16 Amazon Web Services, Telia Inmics-Nebula. 2019. Saatavilla: <https://www.inmicsnebula.fi/fi/palvelut/konesali-ja-pilvipalvelut/amazon-web-services> Viitattu: 17.4.2019
- 17 Telia Inmics-Nebula, INcloud 9. 2019. Saatavilla: <https://www.inmicsnebula.fi/fi/palvelut/konesali-ja-pilvipalvelut/incloud-9> Viitattu: 17.4.2019
- 18 Cloud9 hinnasto, Telia Inmics-Nebula. 2019. Saatavilla: <https://cloud9.nebula.fi/prices.html> Viitattu: 17.4.2019
- 19 Linux Virtual Machines Pricing, Microsoft. 2019. Saatavilla: <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/> Viitattu: 17.4.2019
- 20 Amazon EC2 Pricing, Amazon Web Services Inc. 2019. Saatavilla: <https://aws.amazon.com/ec2/pricing/> Viitattu: 17.4.2019
- 21 Google Compute Engine Pricing, Google. 2019. Saatavilla: <https://cloud.google.com/compute/pricing> Viitattu: 17.4.2019
- 22 Palvelimien Hintalaskuri, Telia Inmics-Nebula,. 2019. Saatavilla: https://my.nebula.fi/#/shopping-cart/instances/new?_k=nm08z1 Viitattu: 17.4.2019
- 23 AWS vs. Azure vs. Google: Cloud Comparison, Datamation. 2019. Saatavilla: <https://www.datamation.com/cloud-computing/aws-vs-azure-vs-google-cloud-comparison.html> Viitattu: 13.9.2019

- 24 Identity & Access Management, Amazon Web Services. 2019. Saatavilla: <https://aws.amazon.com/iam/> Viitattu: 13.9.2019
- 25 Cloud IAM documentation, Google. 2019. Saatavilla: <https://cloud.google.com/iam/docs/overview> Viitattu: 13.9.2019
- 26 Management groups, Microsoft. 2019. Saatavilla: <https://docs.microsoft.com/en-us/azure/governance/management-groups/index> Viitattu 13.9.2019
- 27 Keystone Architecture, OpenStack Project. 2019. Saatavilla: <https://docs.openstack.org/keystone/latest/getting-started/architecture.html> Viitattu 13.9.2019
- 28 Cloud platform resource hierarchy, Google. 12.3.2019. Saatavilla: <https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy> Viitattu: 8.9.2019
- 29 Google Cloud Platform resource hierarchy. Google 2019. Saatavilla: <https://cloud.google.com/billing/docs/images/resource-hierarchy-overview.png> Viitattu: 8.9.2019
- 30 Packstack: create a proof of concept cloud, RDO. 2018. Saatavilla: <https://www.rdoproject.org/install/packstack/> Viitattu: 15.10.2018
- 31 OpenStack 13 (Pike) Lab Installation and Configuration Guide with Hetzner Root Servers, Keith T. 2018. Saatavilla: <https://keithtenzer.com/2018/07/17/openstack-13-queens-lab-installation-and-configuration-guide-for-hetzner-root-servers/> Viitattu: 15.10.2018

Liitteet

Liite 1:

1 Packstack-asennus

Asennus tehdään root-käyttäjänä, joten CentOSin asennusvaiheessa ei tarvitse luoda uutta käyttäjää ja salasana annetaan vain root-käyttäjälle. CentOSia asentaessa konfiguroidaan koneen asetukset, annetaan laitteelle oikea IP-osoite ja hostname. Controller on asennettu VMWaren päälle ja siihen luodaan nfs-jako Cinderiä varten. Aluksi on hyvä varmistaa myös palvelinten paikallinen nimiresoluutio.

```
# vi /etc/hosts
10.20.59.11 openstack.dc.lab openstack
10.20.59.13 compute00.dc.lab compute00
```

IP-osoitteita on käytettävissä 10.20.59.0 verkosta. Ennen asennuksen aloittamista tehdään muutamia muutoksia järjestelmään. Otetaan firewalld ja networkmanager pois käytöstä, sen sijaan käytetään yksinkertaisempaa networkia. Seuraavat vaiheet tehdään jokaiselle palvelimelle.

```
# systemctl disable --now firewalld

# systemctl disable --now postfix

# systemctl disable --now NetworkManager

# systemctl enable --now network
```

Vaihdetaan SELinux pois päältä tai permissive-tilaan.

Aukaistaan konfiguraatio alla olevasta sijainnista ja muokataan SELinuxin tila (kuva 1).
/etc/sysconfig/selinux

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Kuva 1. SELinux-asetukset

Lisätään seuraavat rivit tiedostoon /etc/environment (kuva 2).

```
LANG=en_US.utf-8
LC_ALL=en_US.utf-8
```

Kuva 2. /etc/environment-tiedoston sisältö muokkauksen jälkeen

Seuraavaksi ladataan openstack, packstack sekä python-pip

```
# yum install -y centos-release-openstack-rocky

# yum update -y

# yum install -y openstack-packstack

# yum makecache

# yum install epel-release

# yum install python-pip
```

Asennus ja answer.txt-konfiguraatio suoritetaan vain controllerille. Packstack asennuksen voi myös ajaa suoraan, mutta tässä tapauksessa luodaan answer.txt tiedosto jolla konfiguroidaan asennusvaihe. Tarkoituksena on konfiguroida asennusvaihe oikein. Konfiguroidaan myös yksinkertainen verkko, josta myöhemmin lisää.

Answer tiedoston luominen tapahtuu seuraavalla komennolla (kuva 3).

```
# packstack --gen-answer-file=answer.txt
```

```
[root@localhost ~]# packstack --gen-answer-file=answer.txt
Packstack changed given value to required value /root/.ssh/id_rsa.pub
[root@localhost ~]# ls
anaconda-ks.cfg  answer.txt
[root@localhost ~]#
```

Kuva 3. Answer-tiedoston generoiminen

Cinderiä varten on luotu uusi NFS-jako konesalin VNXe:ssä. Jotta jako saadaan itse koneella käyttöön, täytyy siihen asentaa nfs työkalut.

```
# yum install nfs-utils
```

Muokataan answer.txt-tiedoston asetuksia seuraavasti.

```
CONFIG_SERVICE_WORKERS=1
Change Swift to n
Change ceilometer to n
Change Aodh to n
CONFIG_NOVA_LIBVIRT_VIRT_TYPE=kvm
CONFIG_NEUTRON_OVS_BRIDGE_IFACES=br-ex:enp4s0f0
CONFIG_PROVISION_DEMO=n
CONFIG_CONTROLLER_HOST=10.20.59.11
CONFIG_COMPUTE_HOST=10.20.59.13
CONFIG_NETWORK_HOSTS=10.20.59.11,10.20.59.13
CONFIG_KEYSTONE_ADMIN_PW=*Passwordhere*
CONFIG_CINDER_BACKEND=nfs
CONFIG_CINDER_VOLUMES_CREATE=n
CONFIG_CINDER_VOLUME_NAME=cinder-volume
CONFIG_CINDER_NFS_MOUNTS=10.50.20.30:/cinder-volume
CONFIG_NEUTRON_OVS_BRIDGE_IFACES=br-ex:ens160
```

Pakotetaan KVM:n käyttö. Varmistetaan Cinder-asetukset ja liitetään fyysinen verkko Neutronin ulkoiseen verkkoon. Näiden asetusten jälkeen voidaan aloittaa asennusprosessi. Asennus voi kestää toistakymmentä minuuttia.

```
# packstack --answer-file=answer.txt
```

Asennuksen jälkeen varmistetaan vielä dhcp:n toiminta (kaikilla palvelimilla). Muokataan tiedostoa `/etc/neutron/dhcp_agent.ini` ja uudelleen käynnistetään palvelu.

```
enable_isolated_metadata=True

# systemctl restart neutron-dhcp-agent
```

Kopioidaan controllerin nettikonfiguraatiot `enp4s0f0` ja `br-ex` löytyen `/etc/sysconfig/network-scripts-` tiedostosta myös compute nodelle vaihtaen laitekohtaiset asetukset. Asennuksen jälkeen on vielä konfiguroitu compute-palvelimen fyysinen verkko neutronin verkkoon.

```
# vi /etc/neutron/plugins/ml2/openvswitch_agent.ini

bridge_mappings=extnet:br-ex
```

Käydään vielä läpi ja varmistetaan Cinder-konfiguraatio controllerilla

```
#vi /etc/cinder/cinder.conf
enabled_backends=nfs
volume_clear=none

/[nfs]
volume_backend_name=nfs
volume_driver=cinder.volume.drivers.nfs.NfsDriver
nfs_shares_config=/etc/cinder/nfs_shares.conf
```


2 Web-käyttöliittymä

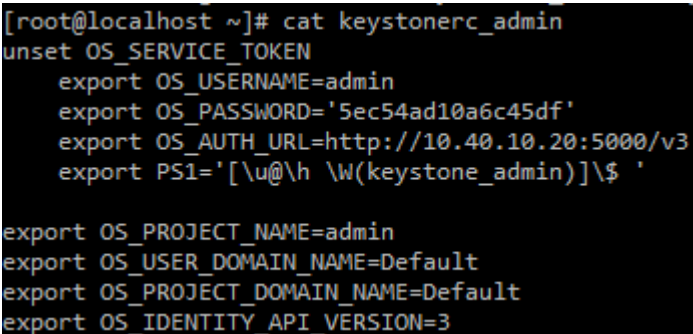
Openstack-asennuksessa asennettiin komponentti nimeltään Horizon. Horizon on Openstackin web-käyttöliittymä. Käymme läpi mm. yksinkertaisen verkon pystyttämisen Horizonin kautta. Käydään läpi instanssin luominen imagesta.

2.1 Käyttöönotto

Web-käyttöliittymä löytyy controllerin ip-osoitteesta.

Käyttäjätunnus: admin

Salasana: keystone_admin tiedostossa, tiedosto löytyy /root kansiota, esimerkki generoidusta salasanasta (kuva 4). Salasanan on voinut asettaa myös packstack-asennuksen yhteydessä.



```
[root@localhost ~]# cat keystone_admin
unset OS_SERVICE_TOKEN
export OS_USERNAME=admin
export OS_PASSWORD='5ec54ad10a6c45df'
export OS_AUTH_URL=http://10.40.10.20:5000/v3
export PS1='[\u@\h \W(keystone_admin)]\$ '

export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_IDENTITY_API_VERSION=3
```

Kuva 4. keystone_admin-tiedoston sisältö

2.2 Verkko

Luodaan uusi yksinkertainen flat network Openstackiin (kuvat 5, 6, 7).

Create Network



Network *

Subnet

Subnet Details

Name

external

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Project *

admin

Provider Network Type * ?

Flat

Physical Network * ?

extnet

☒ Enable Admin State☒ Shared☒ External Network☒ Create Subnet

Availability Zone Hints ?

nova

Cancel

« Back

Next »

Kuva 5. Network-asetukset

Create Network



Network *

Subnet

Subnet Details

Subnet Name

external

Network Address ?

10.40.10.0/24

IP Version

IPv4

Gateway IP ?

10.40.10.1

☐ Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Cancel

« Back

Next »

Kuva 6. Network Subnet-asetukset

Create Network



Network *

Subnet

Subnet Details

☒ Enable DHCP

Specify additional attributes for the subnet.

Allocation Pools ?

10.40.10.21,10.40.10.29

DNS Name Servers ?

8.8.8.8

Host Routes ?

Cancel

« Back

Create

Kuva 7. Subnet Details asetukset

Seuraavaksi luodaan uusi Ingress-sääntö ipv4:lle. Varmista, että oletus security groupista on poistettu ipv6-säännöt sekä ipv4-sääntö. Luo uusi sääntö + Add Rule painikkeesta ja valitse Rule kohtaan "Other Protocol", muita asetuksia ei tarvitse vaihtaa ja sääntö on valmis.

2.3 Koneiden luominen

Jotta voidaan luoda virtuaalikoneita OpenStackiin, tarvitaan ensiksi image. Openstackin käyttöön täytyy olla tietyntyyppiset imaget. Useamman linux distribuution sivuilta löytyy viralliset imaget tähän tarkoitukseen. Openstackin dokumentaatiosta löytyy myös lisää imageista. Ennen

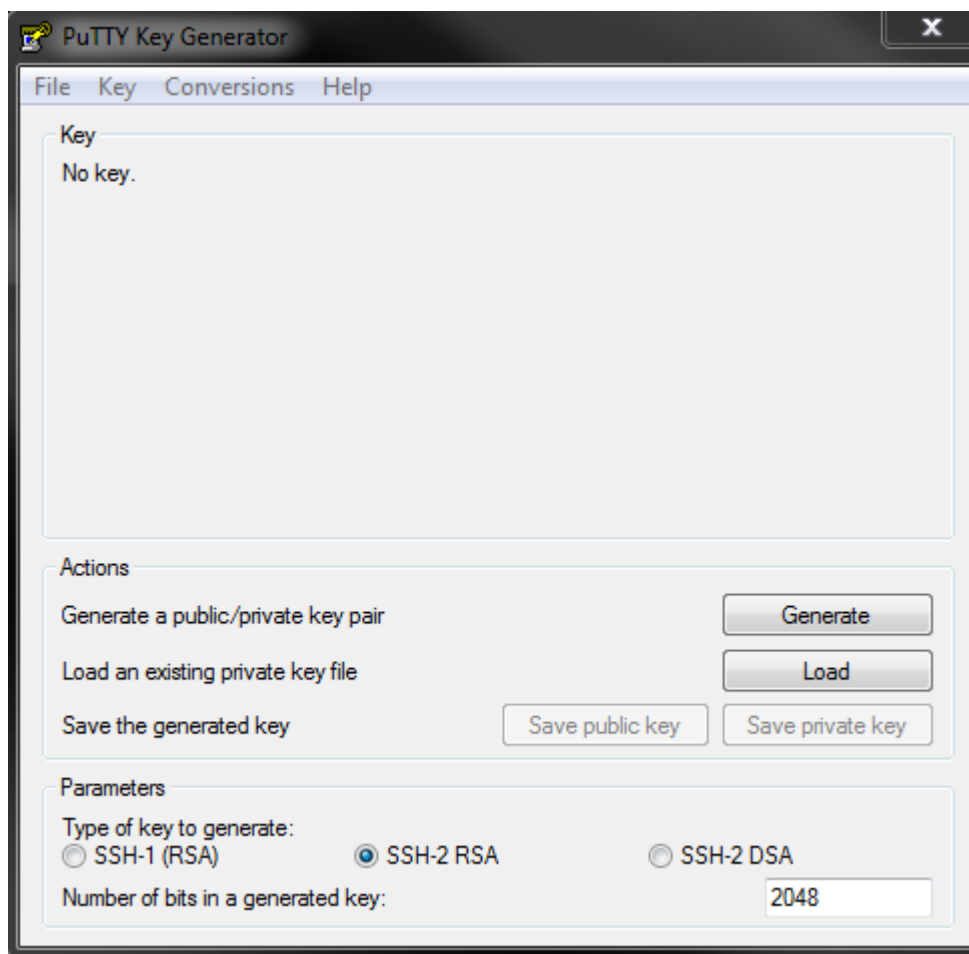
virtuaalikoneen luontia on hyvä tehdä ssh-avainpari. Ssh-avainparin voi tehdä haluamallaan tavalla, tässä ohjeessa käydään läpi myös avaimen generointi putty keygenillä.

2.3.1 Image

Navigoi paikkaan Project -> Compute -> Images -> Create Image. Tässä kohtaa annetaan Imagelle nimi ja kuvaus. Browse kohdasta valitset lataamasi Imagen. Format-kohtaan käytetään QCOW2 ja seuraavaksi voidaan määritellä imagelle minimi-vaatimukset ja image on valmis. Imagen lataamisen yhteydessä on hyvä ottaa muistiin Imagen oletuskäyttäjätunnus, joka yleisesti lukee lataus-sivulla.

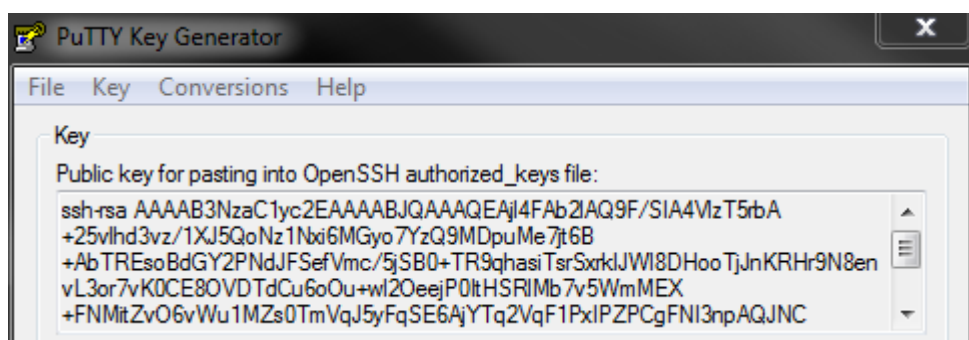
2.3.2 SSH-avain

Openstackista löytyy Key Pairs -välilehti, jossa voidaan luoda tai tuoda avain. Yksinkertaisimmillaan ssh-avaimen voi luoda suoraan OpenStackista tai linux ja mac koneella luoda avaimen komennolla "ssh-keygen" ja avaimen löytää komennolla "cat ~/.ssh/id_rsa.pub". Tässä tapauksessa luodaan avain Putty Keygenillä ja tuodaan se sieltä Openstackiin. Key Generaattorissa tarvitsee vain painaa "Generate" painiketta, jonka jälkeen putty pyytää liikuttamaan hiirtä, kunnes avain on generoitu (kuva 8).



Kuva 8. Putty Key generator

Kun avain on generoitu, kopioi avain tekstikentästä. Tallenna myös Private Key, tätä käytetään virtuaalikoneille kirjautumiseen (kuva 9).



Kuva 9. Generoitu avain

Kopioitu avain käydään viemässä Openstackiin. Project -> Compute -> Key Pairs -> Import Public Key (kuva 10).

Import Public Key

Key Pair Name *

PuttyKey



Load Public Key from a file

Choose file

No file chosen

Public Key * (Modified)

Content size: 397 bytes of 16.00 KB

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAIJDY+App9bQYRetSOx1OOpJ
osB17bsv1FPvi/pv5iNOKBaqgkEiAdWj94ciO7ZY9kV840x1Pf8FySK5lxP/
SXNIM85vSxJzgdqFvVRLp+4nmiTDwZwG9vQ1bpBiCkIC8xkU8Fe6BwQ
xxTiIBII82TOEU6mehc8YtYivS0V9z/RGXMIY4ew3ymSlipWPZiXVRD2t
Df0vOc5+OfBqICWcxEFrtorHFJmgeLpslbOwqpPDUmtT6btO22b+i6VwX
ndApiMhw45Sw+0xzPFIQrLQFY4suXJUkEk+FWEuH1tUjJnpQ6xXwtdl+i
XQYL1etfOWIbrKbGucIfK3J9MXlqwgQ== rsa-key-20180517
```

Kuva 10. Avaimen tuominen Openstackiin

2.3.3 Virtuaalikoneen luominen

Uusi virtuaalikone voidaan luoda välilehdeltä "Instances". Project -> Compute -> Instances -> Launch Instance. Annetaan ensin Instanssille nimi ja kuvaus. Seuraavaksi valitaan aiemmin haettu Image. Flavour-välilehdellä voidaan luoda uusia pohjia koneiden teknisille tiedoille. Network-välilehdet voidaan jättää välistä. Key Pair -välilehti on tärkeä, valitaan juuri tuomamme avain ja tämän jälkeen voidaan painaa "Launch Instance".

2.3.4 Virtuaalikoneeseen yhdistäminen

Koneen IP osoite löytyy suoraan Instances-välilehdeltä koneen kohdalta (kuva 11).

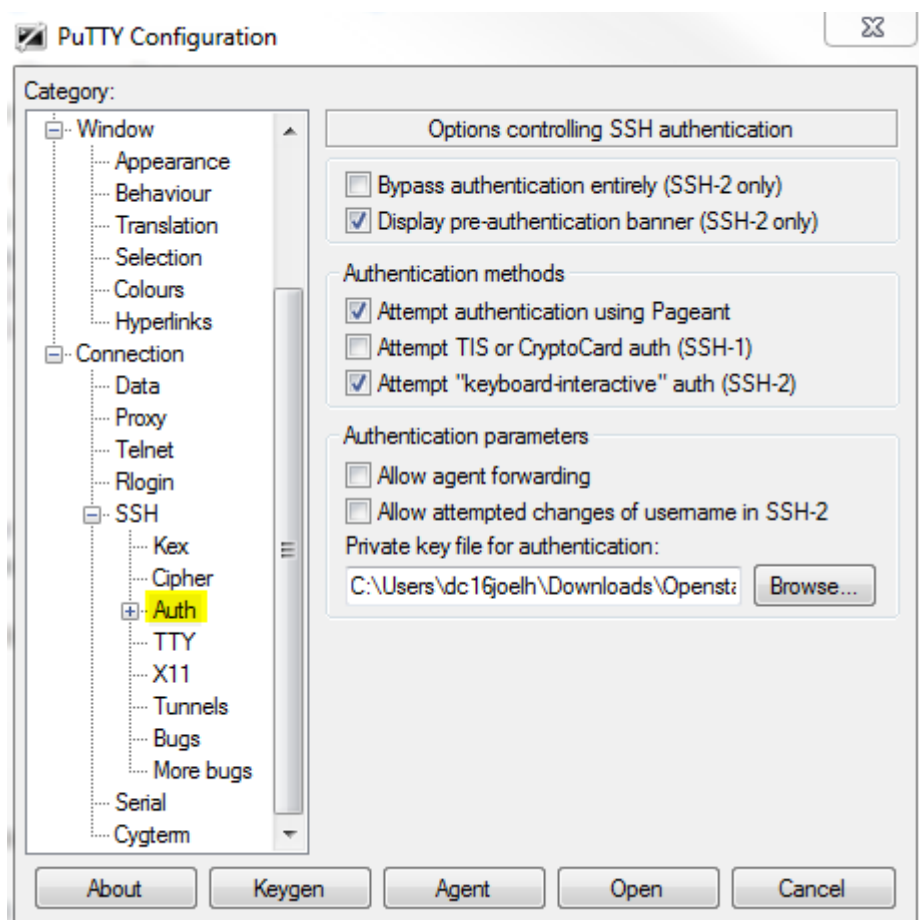
Specify the destination you want to connect to

Host Name (or IP address)	Port
ubuntu@10.40.10.23	22

Connection type:

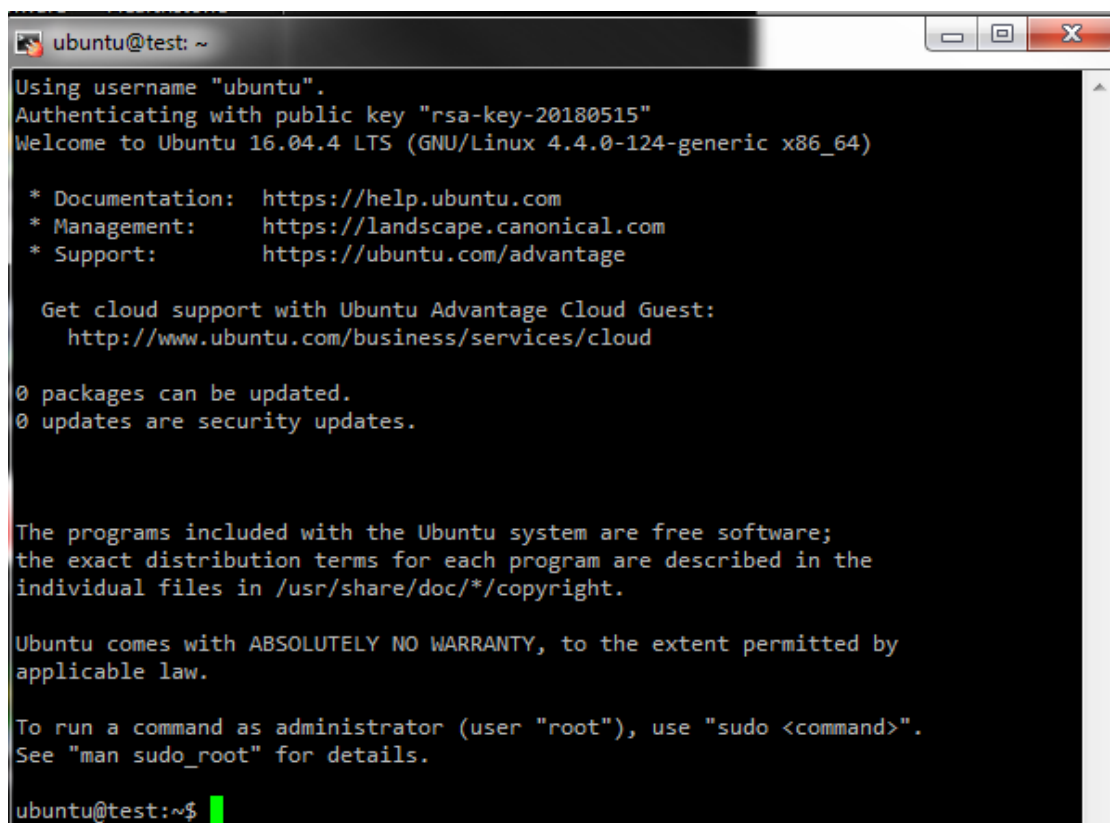
Kuva 11. Putty yhdistäminen, muut asetukset jätetään oletusarvoiksi.

Seuraavaksi käydään valitsemassa tallennettu Private key. Näillä tiedoilla yhdistäminen koneeseen pitäisi onnistua suoraan (kuva 12).



Kuva 12 SSH avain Puttyssä

Yhdistäessä tulee Putty Security Alert, johon voidaan vastata "Yes", jos halutaan muistaa tämä vastaus yhteydelle, tai "no", jos halutaan yhdistää vain tämän kerran lisäämättä avainta.

A terminal window titled 'ubuntu@test: ~' with standard window controls. The terminal displays the Ubuntu login process, including authentication with a public key and a welcome message for Ubuntu 16.04.4 LTS. It lists links for documentation, management, and support, followed by cloud support information. It also shows package update status and legal disclaimers. The prompt 'ubuntu@test:~\$' is visible at the bottom with a green cursor.

```
ubuntu@test: ~
Using username "ubuntu".
Authenticating with public key "rsa-key-20180515"
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-124-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@test:~$
```

Kuva 13. Valmis yhteys