FREENEST-PILVIPALVELUN TARJOAMINEN TURVALLISESTI VPN-TEKNIIKAN AVULLA

Marko Kaunismäki

Opinnäytetyö
Marraskuu 2012

Tietoverkkotekniikka
Tekniikan ja liikenteen ala

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

KUVAILULEHTI / DESCRIPTION

Tekijä(t) KAUNISMÄKI, Marko	Julkaisun laji Opinnäytetyö	Päivämäärä 14.11.2012
	Sivumäärä 71	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty (X)

Työn nimi
FREENEST-PILVIPALVELUN TARJOAMINEN TURVALLISESTI VPN-TEKNIIKAN AVULLA

Koulutusohjelma

Tietoverkkotekniikka

Työn ohjaaja(t)
LEINO, Janne

Toimeksiantaja(t)
RINTAMÄKI, Marko

Tiivistelmä

FreeNest on avoimen lähdekoodin projektityökalu, joka sisältää kaiken tarpeellisen projektityöskentelyyn kuten foorumin, chatin ja muita erilaisia työkaluja. FreeNest tullaan tarjoamaan asiakkaille pilvipalveluna OpenStackin avulla. OpenStack on avoimen lähdekoodin IaaS-mallin pilvialusta, jolla voidaan rakentaa yksityisiä ja julkisia pilviä. OpenStackin kehittäminen aloitettiin vuonna 2010 NASAn ja Rackspacen toimesta.

Opinnäytetyössä käsitellään aihetta tietoturvallisen yhteyden tarjoamisesta etätyöntekijän ja pilvipalvelun välillä. Ongelmana oli, että miten voitaisiin FreeNest-pilvipalveluun yhteyden ottaville käyttäjille tarjota mahdollisimman tietoturvallinen yhteys julkisen verkon yli. Tarkastelun kohteeksi valittiin VPN-tekniikka, joka on yleisesti käytetty ja tunnettu tekniikka kahden pisteen välisessä tietoturvallisessa liikennöinnissä.

Opinnäytetyössä kartoitettiin yleisesti pilvipalveluiden erilaiset käyttöönottomallit sekä huomioitavat asiat tietoturvan toteuttamisesta pilvipalveluissa. Suurin ongelma pilvipalveluissa tietoturvan kannalta on virtualisointi, joka monimutkaistaa verkon rakennetta. Onneksi OpenStackista oli kuitenkin olemassa kattavat dokumentaatiot OpenStackin toiminnasta ja konfiguroinnista, joiden pohjalta ratkaisua ongelmaan oli hyvä lähteä työstämään.

Opinnäytetyössä esiteltiin muutama mahdollinen ratkaisu. Lopullinen toteutettava ratkaisu löytyi kuitenkin OpenStackissa valmiina olevasta palvelusta nimeltään Cloudpipe. Cloudpipe on virtuaalikoneinstanssi, jonka pohjalla on Linux-käyttöjärjestelmä ja OpenVPN. Jokainen asiakas saa tämän instanssin omiin yksityisiin verkkoihinsa OpenStackin sisällä. Ratkaisu jätettiin toteutettavaksi HeartCloud-tiimille, joka kehittää ja ylläpitää OpenStack-pilvialustaa SkyNest-projektissa.

Avainsanat (asiasanat)

Pilvipalvelut, turvallisuus, VPN, OpenStack, Cloudpipe, FreeNest

Muut tiedot

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

KUVAILULEHTI / DESCRIPTION

Author(s)	Type of publication	Date
KAUNISMÄKI, Marko	Bachelor´s Thesis	14.11.2012
	Pages 71	Language Finnish
		Permission for web publication (X)

Title
OFFERING SECURE CONNECTIONS WITH VPN TO FREENEST CLOUD

Degree Programme
Data Network Technology

Tutor(s)
LEINO, Janne

Assigned by
RINTAMÄKI, Marko

Abstract

FreeNest is an open source project management tool that includes all necessary tools for well-functioning project work. FreeNest is to be offered as a cloud service with OpenStack. OpenStack is an Infrastructure as a Service (IaaS) cloud computing project started by Rackspace Cloud and NASA in 2010 for building private and public cloud services.

The focus of this thesis is on offering a secure connection between two points with VPN and in particular on creating a remote access point for remote users coming from different places at different time. VPN was selected since it is the most commonly used technique for offering secure connections.

This thesis discusses basic security considerations on cloud services. The major problem in cloud services emerges from the concept of virtualization which changes the whole concept of basic networking. Fortunately, OpenStack had excellent documentation on how the networking is carried out in OpenStack environment and that helped to find the final solution.

Several different solutions for the thesis's problem were considered. The final solution was found directly from OpenStack. It offers a technique called Cloudpipe. Cloudpipe is basically a pre-installed instance with Linux OS and OpenVPN. This instance is installed on every customer's own private network. This solution was passed on to HeartCloud-team that will conduct the final testing of this the proposed solution with the given configuration parameters.

Keywords
Cloudpipe, VPN, Cloud, Security, FreeNest, OpenStack

Miscellaneous

SISÄLTÖ

KUVIOT

TAULUKOT

LYHENTEET JA TERMIT

Amazon EC2	Elastic Cloud Compute, Amazonin tarjoama IaaS-tason palvelu.
API	Application Programming Interface.
ASP	Application Service Provider.
OpenStack Compute	OpenStackin ohjaava osa, joka pitää koko pakettia kasassa.
DDoS	Distributed Denial of Service, lukuisa määrä päätelaitteita ottaa yhteyden web-sivuun tuoden sen alas.
DHCP	Dynamic host control protocol, verkkoprotokolla jolla voidaan jakaa verkkokohtaisia asetuksia, esimerkiksi IP-osoitteita, verkkoon liittyville uusille laitteille.
ELPA	Elektroninen ICT-palvelutoiminta Jyväskylän ammattikorkeakoulun LabraNet-ympäristössä.
ENISA	European Network and Information Security Agency.
ETF	Engineering Task Force, UK-lähtöinen organisaatio, joka arvioi saatavilla olevia Grid-ratkaisuja.
FreeNest	Avoimen lähdekoodin projektialusta, jonka tarkoituksena on tarjota monipuoliset työkalut projektityöskentelyyn.
IaaS	Infrastructure as a Service, asiakkaalle tarjotaan virtuaalinen palvelinsali.
ICMP	Internet Control Message Protocol, koneelta toiselle pingatessa lähetetään ICMP-viestejä. Pingattava kone vastaa, mikäli se on saatavilla.
Instanssi	OpenStackissa ajettava virtuaalikone, joka sijoitetaan projektin/tenantin sisään.

NAT	Network Address Translation, osoitteenmuunnostekniikka esimerkiksi yksityisestä IP-osoitteesta julkiseksi IP-osoitteeksi.
NIST	National Institute of Standards and Technology.
Nova	Osa Cloudstacking Compute-osiota. Nova huolehtii OpenStackin verkkoliikennöinnistä.
OpenStack	Avoimen lähdekoodin pilvialusta.
OSI-malli	Open Systems Interconnection Reference Model, kuvaa tiedonsiirtoprotokollien yhdistelmän seitsemässä kerroksessa.
PaaS	Platform as a Service, asiakkaalle tarjotaan virtuaalinen alusta (käyttöjärjestelmä).
Projekti/Tenantti	Asiakas Openstack-ympäristössä, voi sisältää monia instansseja.
RFC-1918	Määritelmä yksityisten IP-osoitteiden avaruudelle
SaaS	Software as a Service, asiakkaalle tarjotaan sovellus palveluna.
SkyNEST	SkyNEST on JAMKin sisällä Teknologiayksikössä ICT-tulosalueella tapahtuvan Tekes-rahoitteisen ja ICT-SHOKiin kuuluvan, yhteensä 4-vuotisen Cloud Software -ohjelman osa.
VLAN	Virtual Local Area Network, virtuaalinen yksityinen aliverkko.
VPN	Virtual Private Network, tietoturvallinen liikennöintikanava kahden pisteen välillä.
VM	Virtuaalikone, fyysisen laitteiston päällä pyörivä virtuaalinen laitteisto.

VMM	Virtual Machine Manager eli hypervisor, ohjaa virtuaalikoneiden luontia, muokkausta, käynnistystä ja sammuttamista.
VXLAN	Virtual Extensible LAN, ehdotuksen alla oleva protokolla. Tarkoituksena kasvattaa VLAN-bittien määrää.

1 TYÖN LÄHTÖKOHDAT

1.1 Toimeksiantaja

Tämä työ on osa SkyNEST-projektia, jota viedään eteenpäin Jyväskylän Ammattikorkeakoulun toimesta. Työn tilaajana ja määrittelijänä toimi projektiasiantuntija Marko "NarsuMan" Rintamäki. Projekti on osa Cloud Software Finland-ohjelmaa, ja JAMK on luonnollisesti yksi tämän ohjelman jäsenistä. CSW-ohjelmaan kuuluu myös noin 20 muuta organisaatiota. Cloud Software Finlandin tavoitteena on merkittävästi kehittää suomalaisten sovelluksien asemaa kansainvälisillä markkinoilla. Ohjelma keskittyy nimenomaisesti pilvipalveluihin. Organisaation tavoitteena on tuottaa sovelluksia, jotka ovat käyttäjäystävällisiä sekä ympäristöystävällisiä. Ohjelma on alkanut vuonna 2010 ja sen on tarkoitus saada päätökseen vuonna 2014. CSW toimii SkyNEST-projektin pääasiallisena rahoittajana. Projektin tavoitteena on rakentaa referenssi tuotekehityspilvestä, jossa FreeNest-ympäristö toimii tutkimus/kehityskohteena palvelusta, jota voitaisiin tulevaisuudessa myydä tai käyttää uusien palvelujen pilvimuotoisessa kehittämisessä. FreeNest on kokonaisuus erilaisia tarpeellisia työkaluja, jotka ovat jo käytössä tai tulevat käyttöön jokapäiväisessä projektityöskentelyssä. Mukana on muun muassa erilaisia keskustelufoorumeja, IRC-client/server sekä tiedostojen jakopalvelu. Kyseessä on kaiken tarpeellisen kattava palvelu, jonne jokaisella tiettyyn projektiin osallistuvalla henkilöllä olisi pääsy. FreeNest on saatavilla virtuaalilevykuvana ja sen asennus onnistuu esimerkiksi testausta varten Virtualbox-ohjelmalla tai kuten tulevaisuudessa olisi tarkoitus, ajetaan ohjelma toimintaan suoraan pilveen virtuaalikoneena.

Pilviteknologiaksi on valittu OpenStack-niminen avoimen lähdekoodin pilvialusta. OpenStackin suurimpia etuja ovat muun muassa sellaiset asiat, kuten että sen kehittäminen on jatkuvaa ja sen käyttö on ilmaista. OpenStackin oikeanlaiseen valjastamiseen on viralliset dokumentaatiot olemassa, mutta lopullinen käyttöönotto ja testaaminen jäävät luonnollisesti itse käyttäjän harteille. Yleinen tietotaito OpenStackin suhteen ei yleisesti ole kovin suurta, joten opinnäyteöiden avulla tätä palvelua lähdetään pala palalta rakentamaan ja toteuttamaan. OpenStack on kuitenkin jo käytössä muutamilla kaupallisia palveluita tarjoavilla pilvipalveluyrityksillä kuten Rackspacella.

Rackspace on omalta osaltaan myös mukana OpenStackin kehittämisessä tarjoten ammattimaisen näkökulman palvelun kehittämiseen liittyen.

OpenStack pilvialustaa käyttöönottaessa tulee vastaan paljonkin erilaisia haasteita erilaisilla osa-alueilla, joista yksi on palvelun tietoturvallisuus. Mietittävää aiheuttavat monenlaiset seikat. Millä tavoin asiakkaalle pystytään varmasti lupaamaan tietoturvallista palvelua, ja mitä se asiakkaalle tai palveluntarjoajalle tulee kustantamaan? Nämä kaksi kysymystä ovat ehkä ne tärkeimmät kysymykset, kun mietitään tietoturvan osuutta pilvipalveluihin liittyen. Pilvipalveluiden suurimpia ongelmia ovat muun muassa seuraavanlaiset riskit:

1. Tiedonhallinnan menettäminen
2. Keskitetyn identiteettihallinnan puuttuminen
3. Käyttäjien tunnistaminen ja käyttöoikeuksien myöntäminen
4. Sertifioinnin mahdollinen menettäminen
5. Virtualisointiin liittyvät ongelmat

Kyseinen linjaus perustuu ENISAn selvitykseen ja Cloud Security Alliancen keskustelupalstan ketjuun pilvilaskennan suurimmista tietoturvaongelmista (Pirinen 2010).

Projektista aiheutuvat kustannukset oli myös tarkoitus pitää mahdollisen alhaisina, mikä toi mukanaan omat haasteensa. Laitehankintoja olisi tehtävä mahdollisimman vähän ja toimeen olisi tultava, mikäli mahdollista, jo annetuilla resursseilla. Ohjelmallisen puolen kannalta kaupallisten ratkaisujen käyttöä tuli välttää ja tukeutumisen oli hyvä kohdistua avoimen lähdekoodin ohjelmiin ja ratkaisuihin.

1.2 FreeNest Project Platform

FreeNest on avoimen lähdekoodin projektialusta, jonka avulla esimerkiksi ohjelmakehitysprojekti voidaan aloittaa mahdollisimman vaivattomasti ja nopeasti. Avoimella lähdekoodilla saavutetaan se etu, että alusta on täysin käyttäjänsä muokattavissa, mikäli jokin ominaisuuksista ei vastaa käyttäjän tarpeita, ja lisäksi se on ilmainen. FreeNest koostuu monesta eri avoimen lähdekoodin ohjelmapalasista ja FreeNest integroi nämä palaset toimimaan yhdellä alustalla, ja pystyy näin tarjoamaan mahdollisimman monipuolisen ja helpon järjestelmän käyttöönotettavaksi. Tämä on huomattava etu verrattuna perinteisiin ratkaisuihin. FreeNest-alustan tarkoituksena

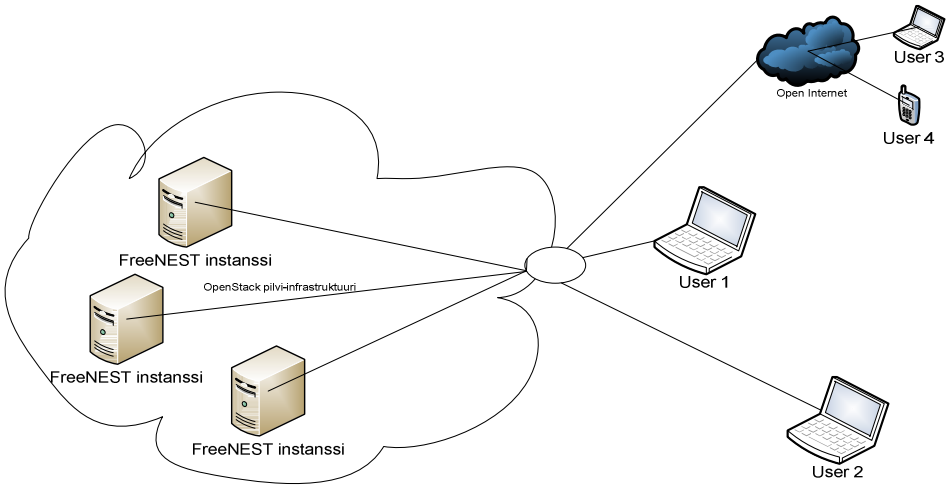
on siis luoda kanava, jota eri osapuolet pystyvät käyttämään yhteisenä kommunikaatio- ja yhteistyö-kanavana. FreeNest olisi tarkoitus tarjoilla asiakkaille mahdollisimman helppokäyttöisenä ja tietoturvallisena. Palveluntarjoajan harteille jää tässä tapauksessa huolehtiminen tietoturvallisuudesta ja asiakkaalle palvelun käyttö tulisi olla mahdollisimman läpinäkyvää, ja asiakkaalle koituisi mahdollisimman vähän vaivaa palvelun käytöstä.

1.3 Opinnäytetyön ongelma

FreeNest-palvelu olisi tarkoitus jalkauttaa toimintaan ELPA-verkon omaan pilvipalvelu infrastruktuuriin. Infrastruktuurin alustana käytetään OpenStack-nimellä kulkevaa avoimen lähdekoodin pilvilaskenta-alustaa. Palvelu olisi tarkoitus ottaa käyttöön erilaisiin projektitoimintoihin ja mahdollisuuksien mukaan sitä pitäisi pystyä tarjoamaan luonteiltaan erilaisille projekteille. Ongelmaksi kuitenkin muodostuu se, että miten tarjotut virtuaalikoneet voidaan erotella toisistaan sillä tavoin, että vain tietyllä käyttäjällä olisi oikeus käyttää vain kyseiselle käyttäjälle määriteltyä virtuaalikonetta. Tiettyyn projektiin sidotulla käyttäjällä ei saisi olla mahdollisuutta päästä toisiin toisten asiakkaiden virtuaalikoneisiin käsiksi ilman asianomaisia tunnuksia ja lupaa. Palvelu tulee toimintaan OpenStack-pilvialustalla, jossa saman fyysisen laitteiston alla saatetaan ajaa moniakin virtuaalikoneita eli instansseja, joten mahdolliset väärinkäytöt pitäisi pystyä estämään.

OpenStack koostuu kolmesta tärkeästä komponentista: Swift, Nova ja Imaging Service. Swift toimii OpenStackin tallennustilana, jonne tapahtuu suurin osa pilven sisällä olevista tietojen tallennuksista. Novan osassa tapahtuvat OpenStackin laskennalliset toimenpiteet, ja sitä voidaankin pitää koko paketin liimana, joka yhdistää nämä kolme komponenttia yhdeksi toimivaksi kokonaisuudeksi. Se ohjaa ja sen kautta ohjautetaan erilaisia toiminteita. Novan toimintoihin kuuluvat niin tietoverkon kuin erilaisten API:ien (Application programming interface) eli ohjelmointirajapintojen ohjaaminen. Imaging Servicen kautta päästään pilven sisällä noutamaan olemassa olevia virtuaalilevykuvia. Lisäksi se tarkistaa tietyin väliajoin niiden olemassaoloa koko pilviinfrastruktuurissa. Jokaisen osion sisällä on olemassa vielä lisää erilaisia toimintoja, joten koko paletti voi muuttua äkkiä hyvinkin laajaksi. (Pepple 2011.)

Yhtenä huolenaiheena on myös mahdollisen arkaluontoisen datan siirtäminen asiakkaan koneelta virtuaalikoneinstanssiin. Ongelman piirissä ovat erityisesti julkisen verkon yli tulevat asiakkaat eli palvelua tulisi olla mahdollisuus myös käyttää tietoturvallisesti myös yksityisen verkon ulkopuolelta. Asiakkaan pääsy palveluun ei saa olla riippuvainen asiakkaan asemapaikasta, ja tällä tavoin käyttäjillä olisi mahdollisuus etätyöskentelyyn. Ongelmaa ei varmasti tule tuottamaan virtuaalikoneinstanssien erottelu toisistaan, vaan se millä tavoin saataisiin rakennettua toimiva ratkaisu OpenStackiin, jonka kautta eri organisaatioiden käyttäjät pääsisivät helposti sisään palveluun ja käyttäjälle määriteltyyn FreeNest-palveluun. Kuviossa 1 nähdään ongelma esitettynä.



Kuvio 1. Ongelma kuviona

Käyttäjät ”User 1” ja ”User 2” haluaisivat saada yhteyden ylimpään virtuaalikoneinstanssiin, jossa heidän projektinsa odottelee lisätietoja ja päivityksiä. Käyttäjä ”User 3” taas haluaisi ottaa yhteyden keskimmäiseen virtuaalikoneinstanssi ja käyttäjä ”User 4” haluaisi ottaa yhteyden alimpaan virtuaalikoneinstanssiin. FreeNest-palvelut ovat käynnissä OpenStack-pilven sisällä.

Ongelmana on, että millä tavoin voidaan hallita eri käyttäjien oikeuksia päästä haluamaansa virtuaalikoneeseen ja millä tavoin liikennöinti julkisen verkon yli olisi mahdollisimman turvallista. Yksi tekniikka on ylitse muiden kun puhutaan julkisen verkon yli tapahtuvasta turvallisesta liikennöinnistä: VPN. Opinnäytetyössä lähdettiin käsittelemään ongelman ratkaisua siitä näkökulmasta, että ongelma olisi ratkaistavissa jollain tapaa VPN-tekniikan avulla.

2 PILVIPALVELUT

2.1 Pilvipalvelut käsitteenä

Pilvipalvelut ja pilvilaskenta ovat nykyisinä käsitteinään vielä melko tuoreita, joskin yleisesti median toimesta huomioitu. Asian suhteen käsitteet ovat kuitenkin vielä osittain hukassa. Ongelma käsitteiden vaikeudesta Suomessa johtuu osittain, kuten yleensä tietotekniikan termistön osalta, suomennuksen vaikeudesta. Alkuperäiskielen eli englannin suhteen ei termistön osalta samankaltaista ongelmaa ole. Tässä opinnäytetyössä käytettiin mahdollisuuksien mukaan sekä suomenkielistä että englanninkielistä vastinetta kyseisille termeille.

Paikallaan olisi erotella pari tärkeää käsitettä liittyen pilvipalveluihin ja huomata kuinka ne eroavat, mutta samalla kuitenkin sitoutuvat toisiinsa. Pilvilaskennalla eli englanniksi ”cloud computing” tarkoitetaan kehitysmallia ja toimitusmallia, jonka avulla voidaan tarjota julkisen verkon eli Internetin yli reaaliaikaista tuotteiden, palveluiden ja ratkaisujen toimitusta. Pilvipalvelut eli englanniksi ”cloud services” ovat taas sellaisia palveluita, joita voidaan tarjota käyttäjille Internetin yli pilvilaskennan avulla. (Gens 2008.)

2.2 Pilvipalvelut yleisesti

Pilvipalvelun sana pilvi on termi, joka kuvastaa verkkoa, jonka komponentteja ei tiedetä tarkasti. Termi pilvi on peräisin 1980-luvulta, jolloin puhelinoperaattorit omaksuivat tavan piirtää vastuualueen rajapinta asiakkaan puhelinoperaattorin välillä untuvaisella pilven kaltaisella objektilla. Tämä tapa on sittemmin siirtynyt tietoliikenneyhteyksiä esittäviin kaavioihin, joilla kuvataan monimutkaisia verkkokaavioita yksinkertaisessa muodossa. Käyttäjällä ei välttämättä ole tarkkaa tietoa esimerkiksi siitä, missä pilvipalvelun tarjoama komponentti sijaitsee, mitä komponentit ovat tai mitä kautta liikenne tarkkaan ottaen kulkee. Hyvänä esimerkkinä tämänkaltaisesta tavasta esittää asia, on Internet. Se kuvataan usein pilven kaltaisella objektilla, koska tarkkaa tietoa sen komponenteista ja sisällöstä ei ole. (Laaksonen 2011.)

Pilvipalvelut ovat tänä päivänä alati esillä oleva puheenaihe ajan tasalla olevissa IT-alan julkaisuissa sekä yrityksissä, jotka haluavat löytää itselleen kustannustehok-

kaampia tapoja toimia. Lähes viikoittain ilmestyy tuoreimpia uutisia pilvipalveluiden tämänhetkisestä kehityksestä. Usein kirjoituksen ja pohdinnan kohteena ovat sellaiset asiat kuten että kannattaako pilveen siirtyä, mitä siinä säästetään ja millaisia mahdollisia riskejä sen käyttöönottoon liittyy? Yleisimmät uutiset ja blogi-kirjoitukset koskevat hyvinkin usein pilvipalveluiden käyttöönottoa sellaisella tasolla, jossa asiakkaalle tarjotaan ohjelmat valmiina pakettina, ja jotka vaativat käyttöönoton suhteen vähemmän ponnisteluja asiakkaan osalta. Tämänkaltaiset palvelut ovat asiakkaille ja käyttäjille valmiita avaimet käteen-paketteja eli asiakkaan harteille jää ainoastaan palvelun käyttäminen ilman suurempia ponnisteluja palvelun käyttöönottoon liittyen. On olemassa kuitenkin myös sellaisiakin pilvipalveluiden tasoja, jossa asiakas pääsee itse luomaan pilven alusta asti. Nämä erilaiset tasot ovat tarkemmin esiteltynä kappaleessa 2.4.

Pilvipalveluiden nopean kehityksen ovat mahdollistaneet muutamat tärkeät edistysaskeleet tietotekniikan saralla. Tietokonelaitteistojen osalta suurimmat muutokset ovat olleet virtualisointi, laskentatehon kasvu sekä moniytimiset prosessorit. Nämä kaikki kolme tekijää mahdollistavat resurssien tehokkaan käytön vähäisillä fyysisillä hankinnoilla. Internet-teknologioiden osalta tärkeimpiä muutoksia ovat olleet web-palvelut, palvelukeskittyneet arkkitehtuurit sekä Web 2.0. (Buyya, Broberg & Goscinski 2011.)

2.3 Virtualisointi

Termi virtualisointi on noussut hyvin tärkeään rooliin nykypäivän pilvipalvelutoteutuksissa. Juuri se on mahdollistanut resurssien kustannustehokkaan hyödyntämisen ja käyttökustannuksien laskun. Virtualisoinnilla tarkoitetaan teknologiaa, jonka avulla voidaan saman fyysisen palvelimen resursseja hyödyntää virtuaalikerroksen avulla luoden sen alle monia virtuaalipalvelimia. Tavallisimmin yksi fyysinen palvelin hyödyntää noin kymmentä prosenttia sen kaikista käytettävissä olevista resursseista, mutta virtualisoinnin avulla voidaan ottaa kaikki teho irti saatavilla olevista fyysisistä resursseista. Jaettavia resursseja ovat muun muassa laskentateho, keskusmuisti ja kovalevytila. Virtuaalipalvelimet näkyvät käyttäjälle kuin mikä tahansa normaali palvelinkone. Jokaiselle virtuaalipalvelimelle jaetaan tietty määrä käytettäviä resursseja,

joita ne pystyvät hyödyntämään omiin suorituksiinsa. Nämä virtuaalipalvelimet voivat sisältää minkä tahansa käyttäjän valitseman käyttöjärjestelmän. (Järvinen 2006.)

2.4 Pilvipalvelun määritelmä

Pilvipalvelun määritelmä on ollut häilyvä jo sen ensimmäisestä käytöstä lähtien. Sen jatkuvan suosion ja kasvun vuoksi palvelulle on kuitenkin yleisen selvyyden vuoksi annettu mahdollisimman tarkat määritelmät. NIST (National Institute of Standards and Technology) on syyskuussa 2011 julkaistussa raportissaan (Mell & Grance 2011.) määritellyt ne ominaisuudet, mitä palvelun tulee sisältää, että sitä voidaan kutsua määritysten mukaiseksi pilvipalveluksi.

Pilvipalvelu on malli, jolla määritellään tietoverkossa käytettävien ja määriteltävissä olevien tietojenkäsittelypalveluiden kuten esimerkiksi tietoverkkojen, palvelimien, ohjelmien, tilojen sekä palveluiden rypästä. Pilvipalvelu mahdollistaa käytettävän sovelluksen käytön käyttäjän näkökulmasta katsoen helpoksi, joka paikassa saatavilla olevaksi, ja heti valmiina käyttöön. Näiden palveluiden tulisi olla käyttöönotettavissa nopeasti ja mahdollisimman vähäisellä hallinnalla tai palveluntarjoajan väliintulolla. Pilvipalvelut koostuvat viidestä tärkeästä ominaisuudesta, kolmesta palvelumallista ja neljästä käyttöönottomallista. (Mell & Grance 2011.)

Pilvipalveluiden määritelmän viisi tärkeää ominaisuutta ovat seuraavat.

1. Itsepalvelu: Käyttäjän on mahdollista omin avuin ottaa käyttöönsä lisää palveluita ja resursseja kuten palvelinaikaa tai kapasiteettia verkkotallennukseen, ilman tarvetta asioida erikseen palveluntarjoajan kanssa.
2. Monipuolinen pääsy verkkoon: Käyttäjän on mahdollista käyttää palvelua mistä tahansa, ja laajalla valikoimalla erilaisia päätelaitteita kuten matkapuhelinta, tablettia, kannettavaa tietokonetta tai pöytätietokonetta käyttäen. Lisäksi pääsy täytyisi olla käyttöjärjestelmäriippumaton.
3. Resurssien jako: Palveluntarjoajan tietojenkäsittelyresurssit jaetaan dynaamisesti palvelun käyttäjien tarpeen mukaan.
4. Nopea elastisuus: Palvelun suorituskapasiteettia voidaan elastisesti ottaa käyttöön tai vapauttaa, joissain tapauksissa automaattisesti, käyttäjän tar-

peen mukaan. Käyttäjän näkökulmasta katsoen suorituskapasiteettia ja resursseja on saatavilla äärettömältä vaikuttava määrä.

5. Palvelun mittaus: Pilvipalvelujärjestelmät ohjaavat ja optimoivat automaattisesti resursseja tarpeen mukaan, tiettyjä resursseja ja niiden arvoja lukien. Resurssien käyttöä voidaan monitoroida, ohjata ja raportoida. Saatujen mittausten perusteella ja havaintojen avulla voidaan tehdä muutoksia, joilla voidaan taata palvelun läpinäkyvyys sekä palveluntarjoajalle, että käyttäjille.

Käytännön määritelmänä pilvipalvelu voi tarkoittaa esimerkiksi suurta joukkoa palvelimia, jotka tarjoavat asiakkaille dynaamisesti skaalautuvaa laskentatehoa, tietoliikenneyhteyksiä tai tallennuskapasiteettia. Pilvipalveluilla pystytään tarjoamaan kustannustehokkaampaa toimintaa, sillä yhtenä osana pilvipalveluista saatavia etuja ovat muun muassa alhaiset käyttöönottokustannukset sekä resurssien skaalautuvuus. Pilvipalveluissa hyödynnetään pääsääntöisesti virtualisoinnin tuomia mahdollisuuksia. (Mts.)

2.5 Pilvipalvelujen eri kerrokset

Pilvipalvelut koostuvat kerroksittain kolmesta erilaisesta palvelumallista. Kutakin palvelumallia voidaan tarjota suoraan asiakkaille. Mitä alemmas palvelumallissa mennään, sitä enemmän vastuuta järjestelmän toimivuudesta on asiakkaalla itsellään. Nämä kolme palvelukerrosta ovat SaaS (Software as a Service), PaaS (Platform as a Service) ja IaaS (Infrastructure as a Service).

2.5.1 SaaS

SaaS eli vapaasti suomennettuna ohjelmistoja palveluina, on yksi kolmesta pilvipalvelujen palvelumallista. SaaS-sovellusta käytetään tyypillisesti web-selaimen avulla tietyllä kuukausihinnoittelulla. SaaS-palvelun ollessa selainpohjainen, on palvelua mahdollista käyttää lähestulkoon jokaisella laitteella josta vain löytyy yhteys Internetiin sekä www-selain. Tämä onkin yksi SaaS-palvelumallin valttikortteja eli se on käyttäjälleen varsin helposti käyttöönotettavissa. Käyttäjä voi ottaa palveluun yhteyden missä tahansa ja milloin tahansa. (Järvi, Karttunen, Mäkelä & Ipatti 2011.)

ASP-malli ja On-premise ovat SaaS-mallia edeltäviä palvelumalleja. SaaS muistuttaa käyttäjän näkökulmasta toimintamalliltaan ASP (Application Service Provider) mallia, joka on yksi osa palvelumallin kehittymistä kohti SaaS-mallia. SaaS-mallin ja ASP-mallin eroavaisuus on se, että ASP-mallissa on palveluntarjoajan toimesta osoitettu jokaiselle asiakkaalle oma dedikoitu palvelimensa, josta asiakkaat käyttävät tilattua palvelua. ASP-mallia edeltävässä On-premise-mallissa asiakas ostaa itselleen palveluntarjoajalta suoraan lisenssejä sovelluksen käyttöä varten ja sovelluksen varsinainen ylläpito jää asiakkaan harteille. ASP-malli tarjoaa asiakkaille näin helpompaa käyttöönottoa ja ylläpitoa, koska asiakkaalle ei enää ole tarvetta omille palvelimille ja fyysisille laitteille. ASP-mallissa asiakkaan on kuitenkin myös mahdollista ostaa itselleen lisenssi sovelluksen käyttöön. (Mts.)

ASP-mallista poiketen, SaaS-mallissa palveluntarjoaja tyypillisesti ylläpitää vain yhtä sovellusta, joka palvelee kaikkia kyseisen palvelun asiakkaita. Sovellus on rakennettu siten, että asiakkaat eivät ole tietoisia siitä että he käyttävät palvelua samanaikaisesti myös muiden palveluntarjoajan asiakkaiden kanssa. Tämä vähentää huomattavasti ylläpitotehtäviä molempien osapuolten osalta. Asiakas tiedostaa tämän muutoksen lähinnä halvemmissa palveluhinnoissa. Asiakkaan sovelluksessa käsittelemät tiedot tallentuvat joko suoraan asiakkaalle itselleen tai palveluntarjoaja tarjoaa jokaiselle asiakkaalle oman erotellun tallennustilan. (Mts.)

Tarjolla olevista SaaS-palveluista hyvänä esimerkkinä voidaan pitää Googlen tarjoamaa ”Google Docs”-palvelua. Se tarjoaa kuluttajalle toimivan tekstinkäsittelyohjelman, jonka tiedostot tallentuvat Googlen palvelimille, tai kuluttaja voi tarvittaessa tallentaa tiedostot myös omaan haluamaansa massamuistiin. Palvelua käytetään selainpohjaisesti millä tahansa käyttäjän valitsemalla selaimella. Asiakkaan tallentaessa tiedot Googlen palvelimelle, on asiakkaalla mahdollisuus päästä käsiksi tallentamiinsa dokumentteihin paikasta ja ajasta riippumatta.

2.5.2 PaaS

PaaS-palvelumallissa (Platform as a Service) asiakkaalle tarjotaan suoraan valmis alusta. Palveluntarjoaja tarjoaa asiakkaalle virtuaalisen laitteiston sekä asiakkaan haluaman käyttöjärjestelmän. Palvelualustan ulkoistaminen tuo mukanaan etuja, jotka näkyvät parhaiten esimerkiksi ohjelmistokehityksen ja liiketoiminnan näkökul-

masta. Käyttäjällä on vapaat kädet muokkailla palveluntarjoajalta saamaansa alustaa omia testailujaan varten. Tällä tavoin mahdollistetaan ohjelmistokehitys antamalla käyttäjälle vapaat kädet ladata omia sovelluksiaan osaksi kokonaisuutta. Käyttäjän vastuulla ei ole alla oleva pilvi-infrastruktuuri, jonka osa-alueisiin kuuluu verkko, palvelimet, käyttöjärjestelmät ja tallennustila. Käyttäjällä on kuitenkin hallinta käyttöönotettaviin ohjelmiin sekä mahdollisten konfiguraatioiden hienosäätöön, jotka kohdistuvat alla olevaan alustaan. (Grance & Mell 2011.)

PaaS-palvelumallin tärkein sovelluksille tarjoama helpotus ja etu on infrastruktuurin totaalinen piilottaminen. Sovelluksen halkomisesta palvelimille ei tarvitse PaaSin varassa kehittäessä huolehtia lainkaan, vaan palveluntarjoajan alusta skaalautuu automaattisesti asiakkaan tarpeiden mukaan. Asiakkaan lisätessä alustaan sovelluksia, kasvaa sen koko dynaamisesti asiakkaan tarpeen mukaan. IaaS-palvelumallin varaankin voi toki rakentaa automaattisesti skaalautuvan alustan, mutta PaaSien kanssa työskennellessä tämä työ on jo tehty, joka helpottaa työtä ja vapauttaa käytössä olevia resursseja muuhun käyttöön. (Järvi, Karttunen, Mäkelä & Ipatti 2011.)

PaaS-palveluita tarjoavia alustoja ovat muun muassa Windows Azure ja Google Apps Engine. PaaS voi rakentua myös kolmannen osapuolen IaaS-järjestelmän varaan. Esimerkki tämänkaltaisesta toteutuksesta on Ruby-sovelluksille tarkoitettu suosittu Heroku-alusta, jota ajetaan Amazon EC2-palvelun päällä. Tällöin PaaS-palveluntarjoajan ei tarvitse itse lähteä rakentamaan pilvi-infrastruktuuria IaaS-tasolta asti.

2.5.3 IaaS

IaaS-palvelumalli (Infrastructure as a Service) on pilvipalvelumallien alin kerros. Tässä mallissa asiakkaalle tarjotaan jo melko vapaat kädet luoda haluamansa ympäristö. Kyseessä ei ole enää yhtä helposti käyttöönotettava malli asiakkaan kannalta kuin edelliset mallit. Tässä mallissa asiakkaalle tarjotaan käytännössä virtuaalinen konesali. Puhutaan niin sanotusta virtuaalisesta konesalista pilvessä, josta löytyy muun muassa virtuaalikoneita ja virtuaalitallennuskapasiteettia eli kaikkea mitä normaalistakin konesalista löytyisi. Kokonaisuuteen sisältyvät yleensä myös verkkoyhteydet, tallennustila, palvelimet ja niiden ylläpito. Asiakkaan harteille jää kaikki muu kuten käyttöjärjestelmän käyttöönotto sekä asiakkaan käyttötarkoitukseen tarvittavat konfiguraa-

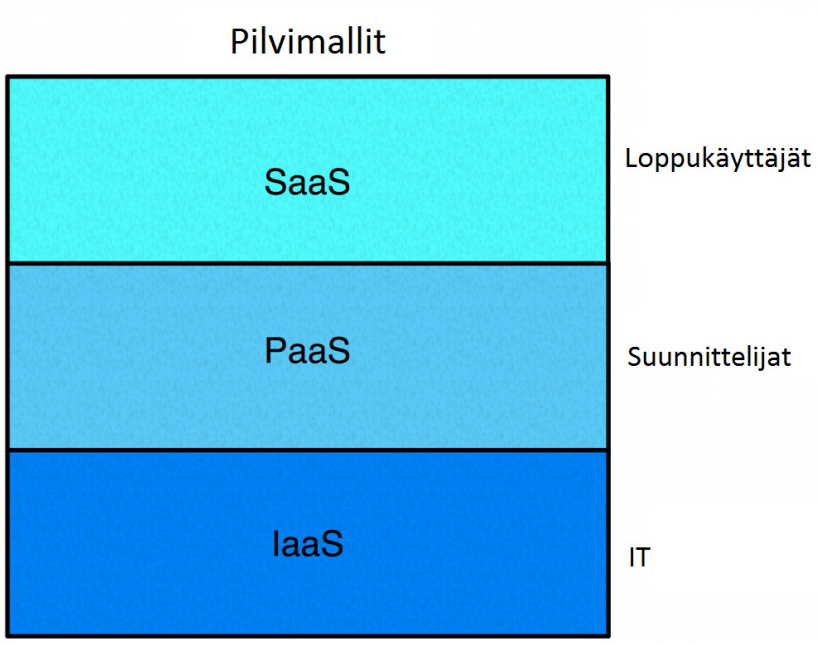
tiot. Lisäksi asiakkaalle voidaan tarjota esimerkiksi julkisia IP-osoitteita ja mahdollisuus pilven sisällä tapahtuvaan liikennöintiin virtuaalikoneiden välillä. Periaatteessa asiakkaalla ei ole minkäänlaisia rajoituksia järjestelmän rakentamisen suhteen. (Buyya, Broberg & Goscinski 2011.)

Virtualisoidut resurssit käyttäytyvät kuten fyysiset vastineensakin, mutta niiden hallinta on palveluntarjoajan näkökulmasta huomattavasti helpompaa ja nopeammin tapahtuvaa sekä kustannustehokkaampaa. Palvelimen lisääminen järjestelmään pitäisi onnistua yleensä paria nappia painalla ja muutokset tulevat voimaan minuuttien, ellei jopa sekuntien, kuluttua. Se, mikä IaaS-mallin todella erottaa muista ulkoistusvaihtoehdoista, on sen nopea skaalautuvuus, eikä se juurikaan rajoita tai ohjaa palveluidensa käyttöä. (Järvi, Karttunen, Mäkelä & Ipatti 2011.)

Amazon E2C on hyvä esimerkki saatavilla olevasta IaaS-tason palvelusta. Palvelussa käyttäjä voi ostaa Amazonilta virtuaalisia tietokoneita ja niihin sisältyvää laskentatehoa ja muistia. Palvelun kustannukset määräytyvät käyttäjälle esimerkiksi palvelimen käynnissäoloajan perusteella.

2.5.4 Yhteenveto kerroksista

Kuviosta 2 nähdään pilvipalvelun kerrokset karkeasti mallinnettuna niitä pääsääntöisesti käyttävien asiakkaiden mukaan.



Kuvio 2. Pilvimallit

Koko pilvipalvelumallien tarjoama lähtee liikkeelle IaaS-mallista, joka on kaikkien pilvipalveluiden perusta. Tärkeimpiä käsitteitä IaaS-tasolla syvemmälle mentäessä ovat virtualisointi, itse pilvi-infrastruktuuri sekä fyysinen laitteisto. Tätä kerrosta hallinnoi yleensä osaava IT-osasto tai vastaava. Tämän pohjalta voidaan lähteä tarjoamaan pilvipalvelumallin mukaisesti eritasoisia palveluita.

PaaS-tasolla toimivat erilaiset suunnittelijat ja järjestelmäsuunnittelijat, jotka tarvitsevat vain jonkinlaisen alustan kehittämiensä järjestelmien ja ohjelmistojen testaamiseen. PaaS-tason käyttäjät säästävät aikaa ja resursseja ottamalla suoraan käyttöön valmiin alustan. PaaS-tasolla asiakkaalle tarjoillaan suoraan valmista virtuaalikonetta käyttöjärjestelmän kera.

SaaS-tason käyttäjiä ovat niin sanotusti tavalliset tallaajat eli loppukäyttäjät, jotka käyttävät kehittäjien tarjoamia palveluita. Asiakkaille tarjoillaan valmista palvelua, jonka käyttöönoton pitäisi asiakkaan näkökulmasta olla helppoa.

2.6 Pilvipalveluiden käyttöönottomallit

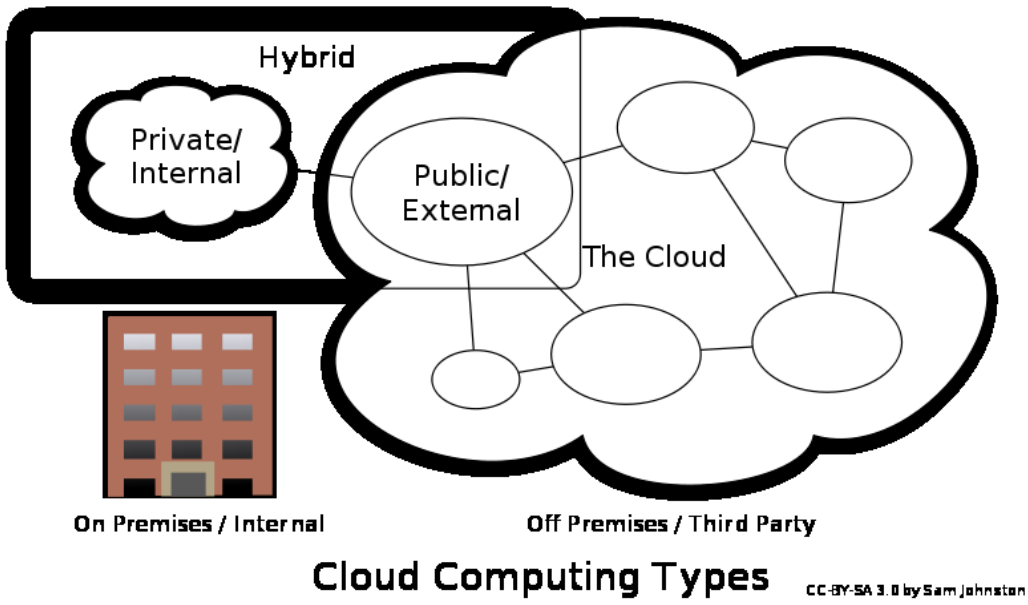
Pilvipalveluihin on myös määritelty erilaiset käyttöönottomallit niiden sijainnin ja verkon rakenteen mukaan. Näitä tyyppejä on pääsääntöisesti kolme kappaletta: yksityinen pilvi, julkinen pilvi ja hybridi-pilvi. Neljänneksi käyttöönottomalliksi voidaan laskea mukaan myös yhteisö-pilvi. Eri pilvipalvelutyypit määritellään NIST-organisaation (Grance & Mell 2011.) mukaisesti seuraavalla tavalla:

1. Yksityinen pilvi: Yksityisen pilven pilvi-infrastruktuuri annetaan ainoastaan yhden organisaation käyttöön, joka kuitenkin koostuu monista käyttäjistä (eri yksiköt). Sen omistaminen, hallinnointi ja käyttö tapahtuvat joko itse organisaation kautta, kolmannen osapuolen toimesta tai näiden yhdistelmästä. Yksityinen pilvi voi sijaita joko organisaatiossa tai sen ulkopuolella.
2. Julkinen pilvi: Julkisella pilvellä tarkoitetaan sellaista pilveä, joka avataan yleiseen käyttöön. Sen voi omistaa ja hallita yritys, akateeminen tai hallinnollinen organisaatio tai yhdistelmä näistä. Se sijaitsee ulkoisen pilvipalveluntarjoajan tiloissa.
3. Hybridi-pilvi: Hybridi pilvellä tarkoitetaan sellaista pilveä, joka on yhdistelmä kahta edellistä tai useampaa erilaista pilvityyppiä. Ne säilyttävät ominaisuu-

tensa, mutta ovat kuitenkin jollain tekniikalla yhdistetty toisiinsa. Tällä tavoin voidaan tietyt pilven osat pitää salaisina ja tietyt julkaista verkkoon.

4. Yhteisö-pilvi: Pilvi-infrastruktuuri jaetaan tietystä yhteisöstä tulevien muutaman organisaation kesken. Yhteisöllä on samat tavoitteet koskien esimerkiksi turvallisuutta. Pilven hallinta tapahtuu joko sisäisesti tai ulkoistetusti kolmannelle osapuolelle.

Kuviosta 3 nähdään pilvityyppien rakenteet. Yksityinen pilvi toimii sisäisesti asiakkaan omassa käytössä eikä sillä ole suoranaisia yhteyksiä julkiseen verkkoon. Asiakas perustaa itse palvelun, ja palveluun otetaan yhteys sisäisestä verkosta. Yksityinen pilvi voi myös toimi kuitenkin kolmannen osapuolen tiloissa. Erona kuitenkin on, että yleensä tällöin asiakas ei joudu jakamaan palveluntarjoajan tarjoamia resursseja muiden asiakkaiden kanssa, vaan yksityisyys on nimenomaisesti omistettu tälle tietylle asiakkaalle. (Cloudtweaks 2012.)



Kuvio 3. Pilvityypit (Johnston 2009)

Julkinen pilvi toimii yleensä kolmannen osapuolen tiloissa ja yhteys siihen on otettava Internetin yli. Asiakas voi kuitenkin toimia omissa tiloissaan eli se on asiakkaan sijainnista riippumaton (Cloudtweaks 2012.).

Hybridi-pilvessä taasen yhdistetään yksityinen ja julkinen pilvi. Organisaatio voi oman yksityisen pilvensä lisäksi valjastaa ulkopuolelta käyttöönsä lisää resursseja. Organisaation tulee kuitenkin pitää huoli siitä, että tietoturva on kunnossa, koska esimer-

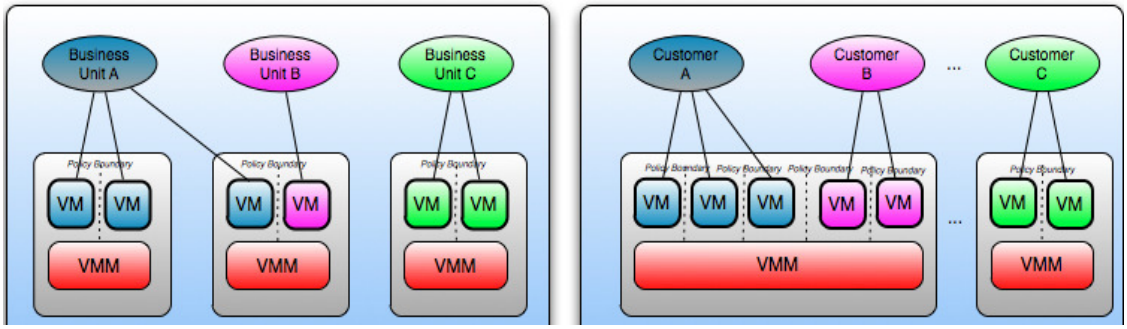
kiksi yhdistelmällä yksityinen pilvi ja julkinen pilvi, liikenne tulee väistämättä kulkemaan julkisen verkon yli. Organisaatio voi käyttää julkisen pilven tarjoamia resursseja vaikkapa jonkin vähemmän tärkeän datan säilyttämiseen, näin esimerkiksi. Asiakkaalla saattaa olla jo valmiina jotain osia omista palveluistaan käytössä esimerkiksi Amazonin EC2 palvelussa. Asiakas kuitenkin haluaa lähteä rakentamaan omaa pilveä ehkä arkaluontoisten tietojen takia tai asiakas haluaa, että hänellä on kaikki langat omissa käsissään. Hän tarvitsee kuitenkin myös Amazonin pilvessä ajettavat palvelut itselleen ja haluaa ne yhdistettäväksi omaan pilveensä. (Mts.)

2.7 Multi-tenanttisuus

Multi-tenanttisuudella eli vapaasti suomennettuna monivuokralaisuudella tarkoitetaan sellaista ominaisuutta pilvipalveluissa, jolla pystytään täyttämään tarve segmentoinnille, erottelulle, johdolle, palvelutasoille ja eri kuluttajatyyppien laskutusmalleille. Palveluntarjoajan on pystyttävä jollain tapaa erottamaan asiakkaat toisistaan, ja asiakkaiden tulisi olla toistensa kesken näkymättömiä. Asiakas A ei saa esimerkiksi tietää, että samalla fyysisellä laitteella toimivat myös Asiakkaan B palvelut. Pilvipalveluiden asiakkaat hyödyntävät julkisen pilveen palveluita toisten asiakkaiden kanssa, jopa oman organisaationsa eri osien kanssa ja jakavat saman pohjainfrastruktuurin. Tämä tarkoittaa sitä, että pilvipalvelun asiakkaat ja käyttäjät jakavat keskenään pilvipalveluntarjoajan tarjoamia resursseja. Resurssit ovat asiakkaiden itsensä päätettävissä ja ne jakautuvat pilvipalvelutarjoajan määräämällä tavalla. Etuina tämän kaltaisessa toiminnassa on, että palveluntarjoajat pystyvät rakentamaan verkko- ja datainfrastruktuureja, jotka hyödyntävät laskentatehoja tehokkaasti, ovat korkeasti skaalautuvia ja helposti kasvatettavia palvellakseen niitä monia samanaikaisia käyttäjiä, jotka jakavat keskenään palveluntarjoajan tarjoamia resursseja. (Cloud Security Alliance 2011.)

Kuviossa 4 nähdään käytännön kautta millaisia ratkaisuja multi-tenanttisuus mahdollistaa. Vasemmalla puolella kuviota 4 on ratkaisu yksityisessä pilvessä. Tässä tapauksessa eri yksiköt ovat merkattu eri värein ja eri kirjaimin. Kuviossa on näkyvillä kolme erillistä alustaa, joissa jokaiseen on asetettu toimintaan kaksi virtuaalikonetta (VM). Lisäksi virtuaalikoneita hallinnoi hypervisor, joka tässä tapauksessa esiintyy nimellä Virtual Machine Manager (VMM). Kuten kuviosta nähdään, on ”Business Unit A” vir-

tuaalikoneet jaoteltu kahdellekin eri fyysiselle laitteelle, kuitenkin siten, että se ei pysty näkemään tai olemaan yhteyksissä ”Business Unit B” virtuaalikoneisiin. (Mts.)



Kuvio 4. Multi-tenanttisuus (Cloud Security Alliance 2011.)

Oikealla puolella kuviota 4 nähdään esimerkki julkisen pilven toteutuksesta, jossa infrastruktuuri on palveluntarjoajan puolesta. Tässäkin kuviosta nähdään, että vaikka ”Customer A” ja ”Customer B” jakavat saman fyysisen laitteen, pystytään ne silti erottelemaan toisistaan (Mts.).

2.8 Pilvipalveluiden haasteet tietoturvan kannalta

2.8.1 Pilvipalveluiden tietoturvasta yleisesti

Pilvipalveluista on paljon etuja sen käyttöönottajalle, mutta kuten kaikissa teknologisissa ratkaisuissa myös pilvipalveluissa on omat varjopuolensa. Tietoturvan kannalta pilvipalveluissa on omat haasteensa. Haasteita lisää hankalempi ympäristö toimia, sillä pilvipalvelut koostuvat monista erilaisista uudenlaisista tavoista yhdistää tekniikoita ja kerroksia. Varsinkin virtualisointi aiheuttaa verkotuksen ja tietoturvan kannalta päänharmia, sillä yleensä virtuaalikoneiden liikennöinti tapahtuu yhden fyysisen rajapinnan kautta. Toisaalta pilvipalveluiden pohja perustuu myös yleisesti tiedossa oleviin tekniikoihin joten erilaisten tekniikoiden hyödyntäminen tulisi olla mahdollista.

Pilvipalveluiden käyttöönoton myötä on tullut esiin myös uudenlaisia tietoturvaongelmia, koska pilvipalvelut ovat palveluina vielä suhteellisen nuori ja näin altis kaikenlaisille hyökkäyksille, tietovuodoille ja muille tietoturvan kannalta riskeiksi luokitelluille asioille. Täytyy muistaa, että tietoturvassa tietoturvan rikkojat ovat aina askeleen edellä tietoturvaa ja sen ylläpitäminen on jatkuvaa oppimista. Siksi yritysten ja muiden organisaatioiden tuleekin huolella miettiä ensin miksi he haluavat siirtyä pilveen,

ja ovatko he valmiina ottamaan datan kannalta riskejä. Etukäteen olisi käytävä läpi mahdolliset riskit ja miettiä onko niihin olemassa ratkaisuja.

Organisaatioilla tulisi olla tarkasti määritelty metodiikka ennen pilvipalvelun käyttöönottamista. Organisaation tulisi lähestyä pilveen siirtymistä varovaisesti, harkiten ja ottaen huomioon yrityksen tietoturvatarpeet sekä kuinka arkaluontoista dataa pilveen olisi tarkoitus siirtää organisaation palomuurin ulkopuolelle. Pilvipalveluissa organisaation on luotettava palveluntarjoajan pilven turvallisuuteen, sillä yleensä pilveen siirtyessä organisaatiolla ei ole enää niin suurta fyysistä kontrollia omaan dataansa. Jokaisen organisaation tulisi tarkasti käydä läpi palveluntarjoajan tietoturvamääritelmät ja varmistaa, että nämä määritelmät sopivat myös organisaation tarkoituksiin. (Mathew 2012.)

2.8.2 Tietoturva käyttäjän näkökulmasta

Pilvipalveluissa on aina kaksi osapuolta: palvelin ja käyttäjä. Ensimmäinen askel kohti turvallisempaa käyttöympäristöä on huolehtia tietoturvan toteutumisesta käyttäjästä alkaen. Tämä on aina suuri haaste, sillä palvelun käyttäjät eivät aina ole kovin tietoisia mahdollista tietoturvariskeistä, ja mitä heidän omat tekemisensä saattavat yritykselle tai organisaatioille aiheuttaa. Organisaatiolla tuleekin olla hyvät määritelmät siihen, miten käyttäjät voivat ylläpitää tietoturvaa. VPN-tekniikan käyttöä tulisi aina suosia palvelimen/palvelun ja käyttäjän välisessä liikennöinnissä. (Mathew 2012.)

Pilvipalveluntarjoajat mahdollistavat asiakkaillensa palvelun monitoroinnin ja hallinnoimisen API-sovelluksia avulla. Niiden avulla voidaan seurata ja ohjata pilvipalveluita. Palveluntarjoajan on tärkeää suojata nämä tärkeät APIt mahdollisten vahinko- ja väärinkäytösyritysten vuoksi. Myös erilaiset selainten lisäosat ja sovellukset aiheuttavat vakavan tietoturvauhan. Pääasiassa tulisi keskittyä käyttäjien ja henkilökunnan kouluttamiseen ja opastaa heitä tietoturvallisen käyttäytymisen ylläpitämiseen. (Mts.)

2.8.3 Verkon tietoturva

Liikennöinti ostettaviin pilvipalveluihin tapahtuu useimmiten julkisen verkon yli. Julkisen verkon kautta tuleva pilvipalvelu on ehkä kustannustehokkaampi kuin oman palvelimen valjastaminen pilvipalveluiden käyttöön, mutta siinä piilee omat vaaran-

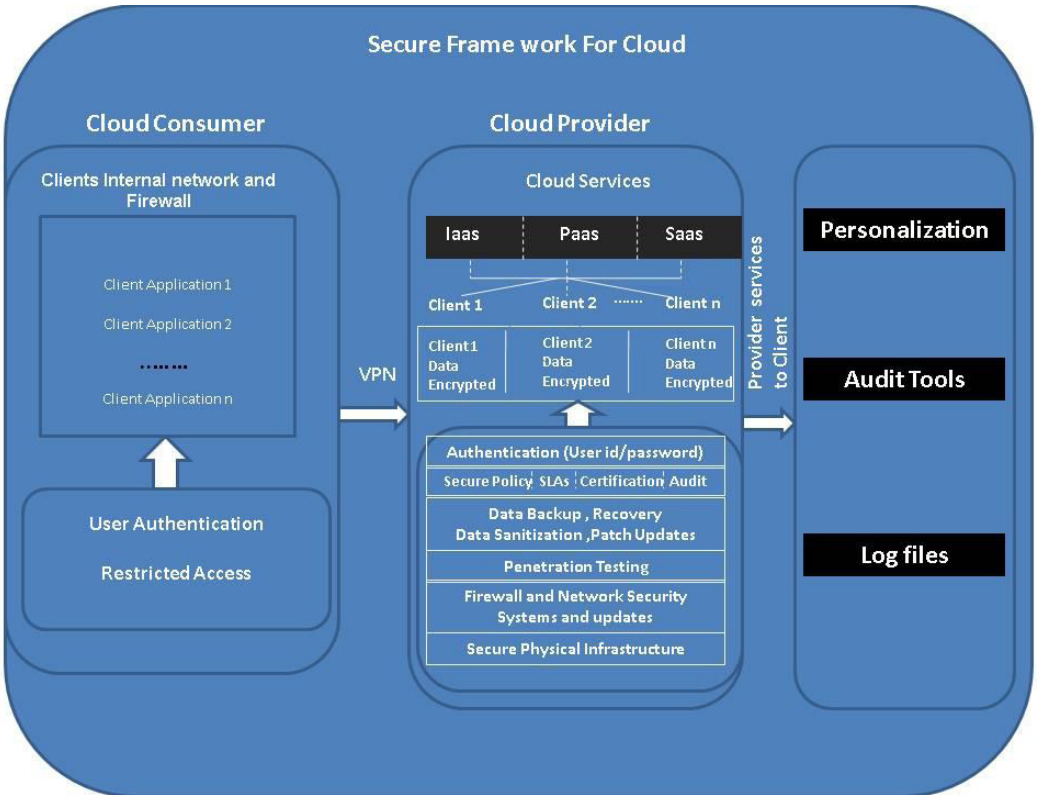
sa, jotka organisaation tulisi ottaa huomioon heidän päättäessä siirtyä pilvipalveluiden käyttäjiksi. Julkisen verkon liikennöinti on kaikille halukkaille ja osaaville nähtävissä, jos sitä ei ole suojattu oikein, huolimatta siitä, että asiakkaan oman sisäverkon tietoturva, ja esimerkiksi palomuuri olisikin kunnossa. (Mathew 2012)

Ohjelmat joita käytetään intranetin sisällä, ovat vaarassa altistua erilaisille verkossa piileville uhille ja Internetin haavoittuvuuksille kuten DDOS-hyökkäykset (Distributed Denial Of Service), kalastukselle, haittaohjelmille sekä erilaisille troijalaisille hevosille. Hyökkääjän onnistuttua saamaan oikeudet käyttäjän tunnuksiin, he voivat salakuunnella kaikkea liikennettä, manipuloida dataa, palauttaa vääränlaista tietoa ja ohjata käyttäjät väärennetyille sivuille. Käyttäjän tunnukset ovat tämän vuoksi jatkuvasti vaarassa. (Mts.)

Pilvipalvelun käyttäjän on huolehdittava palomuurinsa oikeanlaisesta käyttöönotosta. Palomuurin tulisi sijaita verkon kaikissa ulkoisissa rajapinnoissa. Olisi hyvä pitää listaa käytettävistä porteista ja palveluista palomuurissa. Palomuurin säännöt ja käyttöoikeudet tulisi pitää ajan tasalla tietyin väliajoin eli olisi hyvä käydä läpi listatut säännöt ja avoimet ja suljetut portit. Palomuuri tulisi asettaa myös seuraamaan sen läpi menevää liikennöintiä. Tällöin voidaan myöhemmin tarkastella loki-tiedoista tapahtunutta liikennöintiä, mikäli käyttäjän tai yrityksen palvelu on joutunut tietoturvahyökkäyksen kohteeksi. (Mts.)

2.8.4 Yleiset käytänteet

Professori Asha Mathew (Mathew 2012) on esittänyt pilvipalvelun tietoturvaa käsittelevässä raportissaan mainion kuvion siitä, millainen on tietoturvallinen pilvipalvelu ja mitä osa-alueita se pitää sisällään. Kuvio 5 on käytännössä melko summittainen, mutta siitä saa kuitenkin hyvän lähtökohdan tietoturvallisuuden parantamiseen ja sen kehittämiseen.



Kuvio 5. Pilvipalvelun tietoturva (Mathew 2012)

Kuviossa 5 esitetään, että käyttäjien tulisi ottaa käyttöön VPN-tekniikka ottaessaan yhteyttä palveluntarjoajan verkkoon ja tarjottaviin palveluihin. Pilvipalveluntarjoajalla on lukuisa määrä käyttäjiä ja se voi tarjota palveluitaan millä tahansa kolmesta palvelumallista. Tässä kehysrakenteessa palveluntarjoajan velvollisuus on hoitaa käyttäjien oikeuksia ja tarkistaa käyttäjien aitous. Sen jälkeen kun palveluntarjoaja on varma käyttäjän oikeuksista ja asianmukaisesta pääsystä, palveluntarjoaja salaa ja tallentaa käyttäjän tarjoaman datan. (Mathew 2012.)

Kuvion 5 keskimmäisessä kohdassa on esitetty asioita, jotka pilvipalveluntarjoajan tulisi omana osuutenaan koko asiakas-palveluntarjoaja-mallista hoitaa tietoturvallisen palvelun takaamiseksi. Palveluntarjoajalla tulisi olla virallinen tietoturvasuunni-

telma. Palveluntarjoajan työntekijöiden taustat täytyy selvittää, työntekijöiden kulkuoikeudet täytyy olla asianmukaiset, ja heillä täytyy olla tietämys tekniikoista joita he tulevat hallitsemaan. Salasanat täytyisi olla pakollisia vaihtaa tietyin väliajoin ja oletus-salasanoja ei saisi käyttää. Datan varmuuskopioinneista tulisi pitää huolta tietyin väliajoin ja varasuunnitelmia täytyy olla mahdollisten odottamattomien onnettomuuksien varalta. Asiakkaan data tulisi poistaa kun asiakkaan käyttämä palvelu on poistunut pilvestä. Järjestelmien päivitykset tulee pitää ajan tasalla ja järjestelmistä on pidettävä logi-tietoja ylhäällä, joilla voidaan seurata käyttäjiä, kun he ovat käyttäneet palvelua, koska he ovat käyttäneet, kuinka kauan käyttivät ja mitä muutoksia tehtiin. (Mts.)

2.8.5 Tietoturva eri tasoilla

SaaS-tasolla liikuttaessa vastuu tietoturvasta jää suurimmaksi osaksi palveluntarjoajan hoidettavaksi. Kuitenkaan ei pitäisi unohtaa myös käyttäjän vastuuta. Palveluntarjoajan on tarjottava asiakkaalle mahdollisimman tietoturvallista käyttöä tarjoamalleen palvelulle. Asiakkaalle jää pieni, mutta sitäkin tärkeämpi osa tietoturvallisuuden huolehtimisesta lähinnä normaalin tietoturvan tasolla, mitä asiakas omalla koneellaan hoitaisi, kuten esimerkiksi selaimessa asianmukaiset päivitykset, virus-ohjelmat voimassa ja mahdolliset palomuuriasetukset ovat asiakkaan osalta kunnossa. Palveluntarjoaja kuitenkin huolehtii siitä, että asiakkaan tiedot pysyvät palveluntarjoajan päässä turvassa eikä niihin pääse ulkopuoliset käsiksi missään muodossa palveluntarjoajan virheen vuoksi. (O'Neill 2011.)

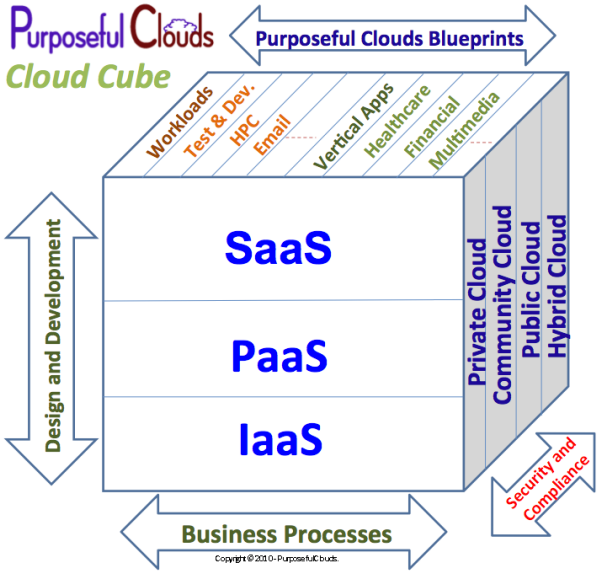
IaaS-tasolla vastuu tietoturvasta on enemmän jaettu asiakkaan ja palveluntarjoajan kesken. Palveluntarjoaja tarjoaa tietoturvaa virtuaalisille palvelin-instansseille ja verkkoyhteyksille matalalla tasolla. Asiakkaan luotua palvelin-instanssin, palveluntarjoajalla saattaa olla tarjottavana asiakkaalleen jotain työkaluja, mutta niin sanotun raudan kovettaminen tai suojaaminen kuuluu asiakkaalle. Tietoturvan tason tarpeet vaihtelevat sen mukaan mihin kyseistä instanssia aiotaan käyttää ja miten. (Mts.)

PaaS-tasolla raja tietoturvan hoitamisen suhteen ei ehkä olekaan niin selvä mitä SaaS- ja IaaS-tasolla. Palveluntarjoaja kyllä pystyy tarjoamaan joitain työkaluja tietoturvan säilyttämiseen, mutta tietoturvasta suurin osa jää asiakkaan vastuulle, joka luo käytettävän alustan ja lähtee sitä muokkaamaan. Asiakas on tässä tapauksessa

kahden selvän ratkaisun välimaastossa ja joutuu miettimään enemmän, mitä tietoturvaratkaisuja tässä mallissa eniten tarvitaan. Suositelluimmat tavat huolehtia tietoturvasta PaaS-tasolla ovat tietoturvallisen yhteyden rakentaminen asiakkaan ja palveluntarjoajan välille sekä auditointi eli tarkkaillaan molemmissa päissä tapahtuvaa käyttäytymistä. (Mts.)

2.9 Yhteenvetoa pilvipalveluista

Pilvipalvelut tulevat olemaan tulevaisuuden ratkaisumalli numero yksi. Niiden edut näyttävät paperilla hyviltä, mutta matka täysin toimivaan ja tietoturvalliseen käyttöön on kuitenkin pitkä. Näkökulmia on monia ja tämä on vain yksi niistä. Kuviossa 6 nähdään havainnollinen kuva pilvipalveluiden kerroksisuudesta.



Kuvio 6. The Purposeful Clouds Cloud Cube (Purposeful Clouds 2010)

Katsojaan kohti oleva sivu esittää kolme pilvipalvelu-infrastruktuuria. Mitä ylemmäs kiivetään, sitä monimutkaisemmaksi menee tarjottavien palveluiden vaativuus. SaaS on tässä tapauksessa korkein, koska palveluntarjoajan näkökulmasta katsoen heillä on eniten ylläpidettävää kyseisessä mallissa, mutta asiakkaalle yksinkertaisempaa, kun taas alemmilla tasoilla käyttäjän vastuu kasvaa. (Purposeful Clouds 2010.)

Neliön oikea sivu esittää erilaisia palvelumalleja. Yksityinen-malli ja hybridi-malli ovat eri ääripäissä. Yksityisessä mallissa käyttäjällä on enemmän valtaa ja mahdollisuuksia hallita järjestelmää, ja mukana on mahdollisimman vähän jaettuja resursseja, kun taas hybridi-mallissa käyttäjä jakaa resursseja muiden pilvipalveluntarjoajan asiak-

kaiden kanssa. Ylin sivu neliössä esittää pilvipalveluiden erilaisia mahdollisuuksia ja käyttötarkoituksia. (Mts.)

3 VPN

3.1 VPN yleisesti

VPN eli Virtual Private Network on tekniikka, jolla voidaan turvallisesti yhdistää kaksi tai useampi yksityinen verkko keskenään julkisen verkon yli, muodostaen näennäisesti yksityisen verkon. VPN-tekniikalla voidaan myös tarjota yhteys monelle yksittäiselle käyttäjälle julkisen verkon yli samaan palvelimeen, ja saman pisteen kautta. Turvallisuudella tarkoitetaan tässä tapauksessa sitä, että tieto pysyy muuttumattomana, ja luottamuksellisuuden säilyttäen. Lisäksi käyttäjät täytyy tunnistaa ja käyttöoikeuksia hallita. VPN:ssä luodaan turvallinen yhteys kahden verkon välillä tunneloimalla liikenne. Tunnelointi itsessään ei takaa turvallista yhteyttä vaan siirrettävän tiedon suojaamiseen on käytettävä jotain salausta, jolla voidaan estää se, että kolmas osapuoli ei näe siirrettävään datan sisältöä. Tiedon salaaminen tapahtuu molempien osapuolten toimesta. Salaaminen voidaan jakaa kahteen pääryhmään: Symmetrinen salaus ja epäsymmetrinen salaus. (VPN - Virtual Private Network and OpenVPN 2010.)

Symmetrisessä salauksessa tarvitaan yksi symmetrinen avain, jota molemmat osapuolet tulevat käyttämään liikenteen salaamisessa. Symmetrinen avain on ensin luotava ja jaettava molemmille osapuolille. Molemmat osapuolet käyttävät samaa salausavainta viestien salaamiseen ja salaamisen purkamiseen. Symmetrinen salaus on melko helppo konfiguroida, mutta sillä on kuitenkin heikkoutensakin. Palvelin voi ottaa vastaan vain yhden päätelaitteen kerrallaan. Avain on tallennettava tekstitiedostoon, joka saattaa altistaa avaimen paljastumiseen, ellei tiedostoa ole sen siirtovaiheessa asianmukaisesti suojattu. (Mts.)

Epäsymmetrisessä salauksessa molemmilla osapuolilla on oma julkinen avain ja oma yksityinen avain. Käytössä on kahden osapuolen välillä yhteensä neljä avainta. Lähettäjä salaa viestin vastaanottajan julkisella avaimella ja vastaanottaja purkaa salauksen omalla yksityisellä avaimellaan, ja koska vain vastaanottajalla on pääsy omaan

yksityiseen avaimeensa, täten vain vastaanottaja pystyy viestin purkamaan. Epäsymmetrinen salaus on hivenen hankalampi konfiguroitava, ja lisäksi sertifikaattien käyttö on pakollisia. Epäsymmetrisen salauksen avulla palvelin pystyy ottamaan vastaan useamman kuin yhden päätelaitteen. (Mts.)

Käyttäjien tunnistamiseen voidaan käyttää julkisen avaimen infrastruktuuria eli PKI:ta (Public Key Infrastructure). Se on menetelmä, jolla julkinen avain liitetään luotettavasti ja yksikäsitteisesti tiettyyn käyttäjään. PKI perustuu sertifikaatteihin eli varmenteisiin, joita varmenneviranomainen eli CA (Certificiation Authority) allekirjoittaa digitaalisesti. Yksi PKI-mekanismiin soveltuvista serfikaattistandardista on X.509-standardi. (Kaario 2002.)

Ehkäpä tärkeimpiä VPN-tekniikan avulla toteutettavia ratkaisuja verkon topologian kannalta ovat Site-to-Site sekä Remote Access. Remote Accessilla toteutetaan etäkäyttäjien pääsy asianomaiseen sisäverkkoon. Site-to-site:ssa luodaan nimensä mukaisesti yksi yhteys kahden pisteen välillä VPN-tunneloinnilla, jolloin kaksi sisäverkkoa voidaan yhdistää näennäisesti samaan sisäverkkoon julkisen verkon yli. Remote Accessissa avataan suuri yhtäaikainen määrä VPN-tunneleita ja ne voidaan toteuttaa VPN-palvelimella, joka käyttäjätunnuksia ja salasanoja vastaan antaa yhteyksiä niitä tarvitseville päätelaitteille ja käyttäjille, tai vaihtoehtoisesti voidaan käyttää myös muita luotettavia tunnistautumistapoja.

3.2 VPN-tekniikan toiminta välillä päätelaite-palvelin

Oletetaan että etätyöskentelijä jonka julkinen IP-osoite on 1.2.3.4, haluaa ottaa yhteyden palvelimeen. Tällä palvelimella on sisäinen IP-osoite 192.168.1.10 ja tämä palvelin ei ole yhteyksissä julkiseen verkkoon. Ennen kuin etätyöskentelijä saa yhteyden palvelimeen, täytyy tämän kulkea VPN-palvelimen ja mahdollisesti palomuurin läpi, jonka julkinen ip-osoite on 5.6.7.8 ja jonka sisäinen IP-osoite on 192.168.1.1. Kaikki koneiden välillä tapahtuva liikennöinti on pysyttävä salassa. VPN-palvelin luovutta etätyöskentelijälle IP-osoitteen VPN-serverin yksityisestä aliverkosta, esimerkiksi IP-osoitteen 192.168.1.30. VPN:n toiminta askel askeleelta:

1. Etätyöskentelijä ottaa yhteyden VPN-palvelimeen ulkoisen verkko-rajapinnan kautta

2. VPN-palvelin jakaa etätyöskentelijälle IP-osoitteen, esimerkiksi 192.168.1.30, VPN-palvelimen aliverkosta ja luo virtuaalisen verkkorajapinnan jonka läpi salatut paketit kulkevat VPN-tunnelin toiseen päähän.
3. Etätyöskentelijän halutessa ottaa yhteys palvelimeen, se valmistelee paketin meneväksi palvelimen sisäiseen IP-osoitteeseen, salaa sen ja paketoi mukaan ulkoiseen VPN-pakettiin. Tämä paketti lähetetään VPN-palvelimen julkiseen IP-osoitteeseen. Sisin paketti on salattu niin, että vaikka joku ulkopuolinen saisi paketin käsiinsä, ei hän pääsisi näkemään paketin sisältöä tarkemmin. Hän näkee vain, että etätyöskentelijä keskustelee jonkin palvelimen kanssa. Sisin salattu paketti sisältää lähde-osoitteen, joka on etätyöskentelijälle määritetty IP-osoite ja vastaanottajan osoitteena on virtuaalikoneen osoite. Uloin paketti sisältää etätyöskentelijän julkisen IP-osoitteen lähdeosoitteena ja vastaanottajan osoitteena on VPN-palvelimen julkinen IP-osoite.
4. Paketin saapuessa VPN-palvelimelle julkisesta verkosta, palvelin purkaa kapsuloinnin ja salauksen ja lähettää paketin eteenpäin vastaanottajalle
5. Jonkin ajan kuluttua VPN-palvelin saa vastaus-paketin virtuaalikoneelta, jonka on tarkoitus mennä etätyöskentelijälle. VPN-palvelin tarkistaa reittitaulustaan, että paketin on mentävä VPN-tunnelin kautta etätyöskentelijälle.
6. VPN-palvelin salaa vastauspaketin, kapsuloi sen VPN-pakettiin ja lähettää julkisen verkon yli.
7. Etätyöskentelijä vastaanottaa paketin. Etätyöskentelijän VPN client-sovellus purkaa kapsuloinnin ja salauksen.
8. Loppujen lopuksi on kuin molemmat osapuolet olisivat samassa lähiverkossa vaikka liikenne kulkeekin julkisen verkon yli.

Näin kaksi laitetta pystyy VPN-tekniikan avulla luomaan tunnelin julkisen verkon yli. Tunnelin molemmissa päissä olevat laitteet toimivat saman virtuaalisen sisäverkon sisällä. Ongelmia saattaa syntyä jos etätyöskentelijän kone on NATin takana ja etätyöskentelijän yksityinen verkko toimii samassa IP-osoiteavaruudessa kuin VPN-verkko. Tämänkaltaisen ongelman pitäisi olla ratkaistavissa NAT-muunnoksen avulla, jossa palvelimen päästä asiakkaalle ajetaan VPN-tunnelissa jokin toinen IP-osoite, joka ei ole päällekkäinen asiakkaan oman sisäverkon kanssa. Helpoin tapa ratkaista

ongelma on kuitenkin valita sellainen IP-aliverkko, joka ei osu päällekkäin asiakkaan aliverkon kanssa. (VPN - Virtual Private Network and OpenVPN 2010.)

Ongelmia saattaa esiintyä myös siinä, että asiakkaan ottaessa yhteyden VPN-tunneliin, ei asiakas päätelaitteellaan pääse enää julkiseen verkkoon ja normaalisti Internetiin. Tämä voidaan korjata split tunneling-tekniikalla. Split tunneling-tekniikan avulla voidaan ohjata liikenne kulkemaan siten, että vain VPN-palvelimeen kohdistuva liikenne ohjataan asiakkaan virtuaalisen rajapinnan kautta, joka luotiin VPN-yhteyttä varten. Kaikki muu liikenne reititetään kuin VPN-tunnelia ei olisikaan olemassa. (Roger's Information Security Blog - VPN Split Tunneling 2010.)

3.3 OpenVPN

OpenVPN on vuonna 2001 James Yonanin kehittämä avoimen lähdekoodin sovellus VPN-yhteyksiä varten. Se on kilpailijoihinsa nähden monipuolinen ratkaisu erilaisten VPN-yhteyksien luontiin. Se pystyy tarjoamaan käyttäjälleen VPN-ratkaisun joko OSI-mallin kerroksissa 2 tai 3 eli OpenVPN-tunnelit pystyvät kuljettamaan ethernet-kehyksiä, IPX-paketteja ja NETBIOS-paketteja, jotka ovat usein ongelma muille VPN-ratkaisuille. OpenVPN-yhteyksiä voidaan tunneloida melkein minkä tahansa palomuurin läpi. Lisäksi se voi käyttää joko TCP- tai UDP-protokollaa. OpenVPN mahdollistaa sen, että palomuuri tarvitsee vain yhden avonaisen portin sen käyttöä varten, sillä se tukee monia yhteyksiä saman TCP- tai UDP-portin kautta, samalla mahdollistaen erilaiset konfiguraatiot jokaiselle erilliselle yhteydelle. OpenVPN-tunneleiden kanssa voidaan käyttää erilaisia palomuurisääntöjä, uudellenohjausmekanismeja sekä NATia. OpenVPN on täysin vapaa NAT-tekniikan tuomista ongelmista. Kaikin puolin se on lähes ylivoimainen verraten muihin tarjottaviin VPN-tekniikoihin hinta-laatusuhteessaan. (Feilner 2006.)

OpenVPN-sovellus voi toimia sekä palvelimena, että päätelaitteena. Salausavainten vaihtoon OpenVPN käyttää SSL/TLS-protokollaa. Salausavaimet luodaan palvelimella ja ne voidaan siirtää tietoturvallista kuljetustapaa käyttäen päätelaitteen käytettäväksi. SSL VPN ero perinteisiin VPN-yhteyksiin on se, että se toimii OSI-mallin kerroksissa 4-7 kun taas esimerkiksi suosittu IPSec toimii OSI-mallin kerroksessa 3. IPSec kantaa itse huolen tiedonsiirron luotettavuudesta, kun taas SSL VPN-yhteydessä se

jää OSI-mallin kerroksessa 4 olevan tekniikan huolehdittavaksi. Käytännössä päätelaitteen ottaessa yhteyden palvelimen ennalta määrättyyn porttiin, luodaan palvelimen ja päätelaitteen välille SSL-tunneli. Päätelaite asettaa itselleen suljetun verkon IP-osoitteen, ja näiden kahden välille luodaan virtuaaliverkko. (Mts.)

4 OPENSTACK

4.1 OpenStack yleisesti

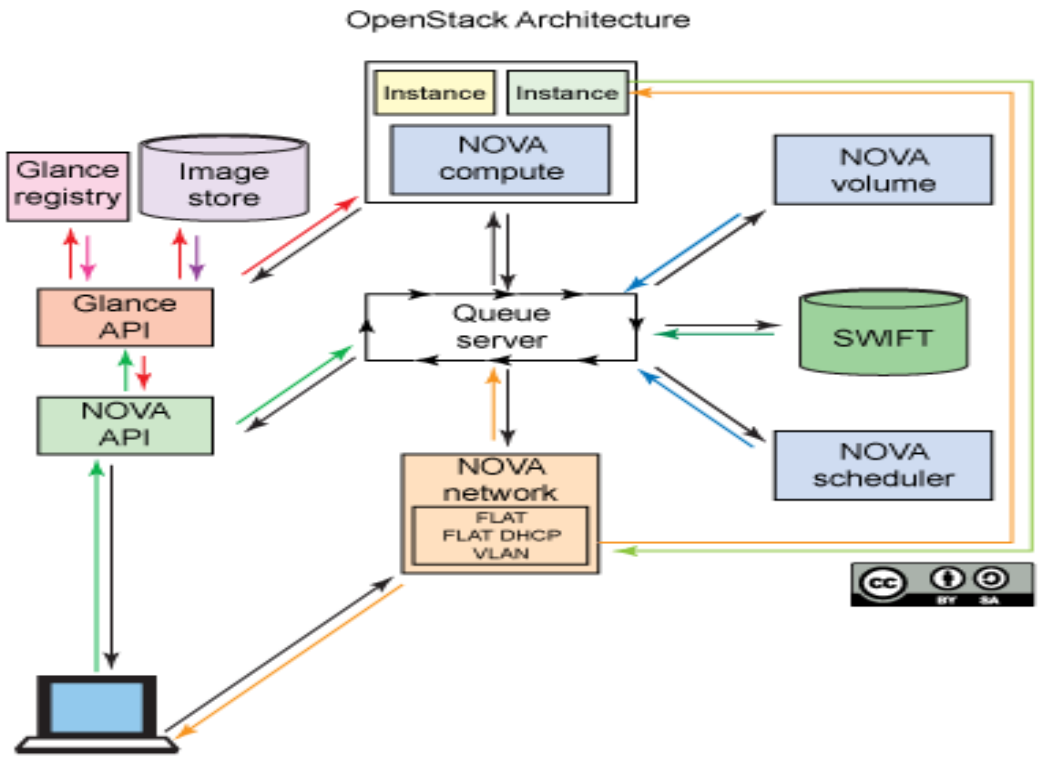
OpenStack on avoimen lähdekoodin pilvi-infrastruktuuri toteutus IaaS-tasolla. Se esiteltiin ensimmäisen kerran Heinäkuussa 2010 ja on siitä lähtien kasvattanut suosiotaan lähes räjähdysmäisesti. OpenStackin suosion kasvu on sen laadukkaiden yhteistyökumppaneiden ansiota. Muuan muassa NASA, Rackspace, Cisco ja Ubuntu ovat mukana OpenStackin kehittämisessä. Openstackin tarkoituksena oli luoda pilvipalvelualusta sekä julkisille että yksityisille pilvipalveluntarjoajille riippumatta palveluntarjoajan koosta. Openstackin suurin etu verraten esimerkiksi vertaiseensa palveluun Eucalyptukseen, on että OpenStack on varustettu valmiimmaksi kohti julkisia pilviverkkoja, sekä se tukee isojen verkkojen käyttöönottoa pienten yksityisverkkojen lisäksi. OpenStackin ollessa avoimen lähdekoodin toteutus, on sen kehitys nopeaa ja uusia ominaisuuksia julkaistaan melko nopeaan tahtiin. Se ei siis ole vielä täydellinen kokonaisuus vaan vaatii jatkuvaa kehittämistä sen käyttäjiltä saamien palautteiden mukaan. OpenStackin www-sivuilla on mahdollista ilmoittaa käyttöönotossa ja käytössä esiintyneistä virheistä.

Tällä hetkellä uusin OpenStack-versio on julkaistu 27.9.2012. Se on kuudes versio ja sille on ristitty nimeksi ”Folsom”. OpenStackin ollessa jatkuvassa kehityksessä ja vasta oikeastaan isojen yritysten käytössä kaupalliseen tarkoitukseen, ei sen käyttöönotosta kokonaisuudessaan löydy vielä moniakaan dokumentaatioita. OpenStackin omat dokumentaatiot pyritään pitämään ajan tasalla ja niitä korjaillaan jatkuvasti käyttäjiltä tulleiden ilmoitusten mukaan. (What is OpenStack? 2011.)

Openstack koostuu kolmesta eri pääosasta: Swift, Nova ja Glance. Swift on vikasietoinen ja skaalautuva tallennuskapasiteettipalvelu ”OS Object Storage”. Nova on koko pilvi-infrastruktuuria palveleva niin sanottu pilviohjain ”OS Compute”. Glance

puolestaan tarjoaa luotujen virtuaali-instanssien etsimistä ja niiden palauttamista tarpeen mukaan. (Mts.)

Kuviosta 7 nähdään OpenStackin arkkitehtuuri. Tärkeimmät osat ovat kuviossa olevat Nova compute ja network. Compute sisältää asiakkaat ja heidän virtuaalikoneensa eli instanssit. Nova network määrittää minkälaista verkotusta OpenStackin virtuaalisessa ympäristössä käytetään.



Kuvio 7. OpenStackin arkkitehtuuri (Markey 2012.)

4.2 OpenStackin ongelmakohdat

ETF (Engineering Task Force) on tehnyt OpenStackista arvioinnin. Arvioinnissa tutkitaan OpenStackin käyttöönottoa sekä yleistä dokumentointia ja luetellaan sen eri ominaisuuksia. Dokumentissa on perehdytty OpenStackin puutteisiin. Seuraavassa lyhyt luettelointi niistä puutteista, mitä ETF on havainnut OpenStackia koskien:

1. OpenStackin dokumentointi ei ole parasta mahdollista ja asiaa hankaloittaa online-dokumentoinnin kaksi eri sijaintia. Kaiken kaikkiaan dokumentointi on kuitenkin kohtuullisen selvää, mutta jotensakin kokonaisuutena hieman hajanainen, mikä johtuu OpenStackin yhteisöllisestä tavasta jakaa tietoa.

2. OpenStackin asennus on dokumentoitu hyvin, mutta itse palvelun konfiguroinnista löytyy vähemmän tietoa.
3. Varsinaisen käyttäjätuen puuttuminen, mikä on kuitenkin yleistä avoimen lähdekoodin ohjelmistoille. Suurin osa kehityksestä tapahtuu käyttäjien ehdotusten ja kokemusten perusteella, joten dokumentointi ja tuki tulee suoraan kehittäjiltä ja käyttäjiltä.

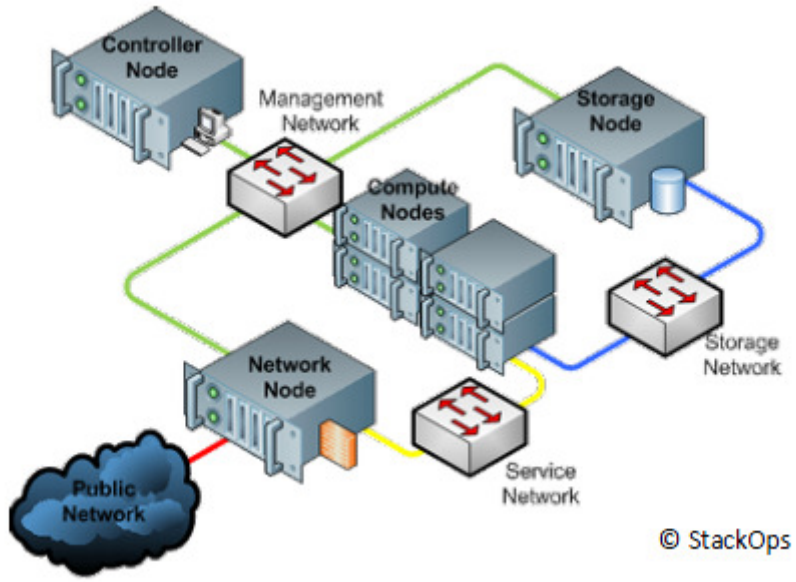
Nämä kolme puutetta tuovat hyvin esiin sen, miksi OpenStack vaatii tutkimista esimerkiksi opinnäytetöiden muodossa. (Thorn 2011.)

4.3 OpenStack fyysinen verkko

Virtualisoinnin vuoksi virtuaalikoneet ovat samalla fyysisellä palvelimella ja käyttävät esimerkiksi virtuaalikoneesta ulospäin kohdistuvaan liikennöintiin yhtä fyysistä rajapintaa. Itse virtuaalikoneet pitäisi kuitenkin olla jo virtualisointivaiheessa toisistaan eroteltuina. Virtualisoinnin ansiosta käyttäjien käyttöjärjestelmät tai tallennustila ei pitäisi mennä sekaisin. Ongelmaksi kuitenkin muodostuu itse verkon osuus. Virtuaalikoneet kuuluvat useimmiten samaan yksityiseen aliverkkoon. Tämä aiheuttaa sen ongelman, että koska koneet ovat samassa verkossa näkyvät ne myös toisillensa. OpenStackilla on kuitenkin ratkaisunsa tähän ongelmaan kolmen erilaiset verkotustavan muodossa, jotka ovat Flat, Flat DHCP ja VLAN DHCP. OpenStackin ympäristössä on myös olemassa käsitteet julkinen ja yksityinen verkko. Julkinen verkko on OpenStackin julkiseen verkon liikennöintiä varten ja yksityinen verkko OpenStackin sisäistä liikennöintiä varten. Julkinen verkko voi olla esimerkiksi suoraan operaattorilta saatuja julkisia IP-osoitteita tai yksityisen verkon osoitteita. Jos OpenStackin julkisen verkon osuudessa käytetään yksityisiä osoitteita, tarvitaan yksityiseen verkkoon pääsyksi esimerkiksi NAT-muunnos verkossa olevalta reitittimeltä. Tämä pätee varsinkin pienissä verkoissa, jossa voi esimeriksi olla vain yksi julkinen IP-osoite käytössä. (OpenStack Nova documentation 2012.)

OpenStackin oikeanlaiseen toimintaan vaaditaan, että käytettäviltä fyysisiltä palvelimilta löytyisi kaksi verkkokorttia. Toinen OpenStackin julkista verkkoa varten ja toinen OpenStackin yksityistä verkkoa varten. Julkisen verkon kautta OpenStack ja sen sisällä olevat virtuaalikoneet saisivat yhteyden Internetiin. Yksityisen verkon kautta

OpenStack kuljettaisi omaa dataansa eli esimerkiksi erilaiset käskyt ja virtuaalikoneiden välisen liikennöinnin. OpenStack-infrastruktuuri voi olla jaettuna useammalle fyysiselle laitteelle. Kuviossa 8 on esiteltynä esimerkki OpenStackin fyysisestä arkkitehtuurista.



Kuvio 8. OpenStack fyysinen arkkitehtuuri (StackOps 2011.)

Yhteys julkiseen verkkoon on toteutettu yhden fyysisen laitteen kautta, kuviossa Network Node. Laitteet kommunikoivat keskenään omassa yksityisessä verkossaan. Projektit ja niiden sisällä olevat instanssit sijaitsevat Compute Nodeissa. (Mts.)

4.4 Nova

Openstack Nova hallitsee pääsyä pilveen käyttäjien ja projektien kautta. Projektit perustetaan pääorganisaatiorakennelman pilven sisälle. Projektit ovat resurssisäiliöitä jotka sisältävät käyttäjät, virtuaalikonelevykuvat, instanssit, avaimet, loogiset levyt sekä VLANit. Jokaisella projektilla on oma ennalta määritelty osuutensa eri resursseista kuten loogisten levyjen määrä ja koko, instanssien määrä, käytetyt ytimet sekä käytettävien julkisten IP-osoitteiden määrä. OpenStackissa asiakkaalle tarjottua ympäristöä kutsutaan nimellä projekti, myös tenant on käytössä oleva sana. Asiakkaan projekti koostuu yhdestä tai monesta instanssista. Instanssit ovat virtuaalikoneita, joiden sisällä ajetaan käyttöjärjestelmiä. Instanssilla on tietyt annetut resurssit luo-

dun instanssi-tyypin mukaan. Vaihtelevia resursseja ovat muun muassa laskentateho, kapasiteetti ja muisti. (OpenStack Nova documentation 2012.)

Openstack on suunniteltu käytettäväksi monien käyttäjien kesken ja eri käyttäjien pääsy on hallinnoitu roolijakoisen ohjaimen avulla (role based access control, RBAC) ja roolit on jaettu eri käyttäjien kesken siten, että korkein rooli on koko pilven ylläpitäjä jolla on juurioikeudet koko pilvijärjestelmään. Seuraavaksi ylin rooli on nimeltään IT Security, joka sallii käyttäjän asettamaan instansseja karanteeniin. Seuraava rooli on projektin pääkäyttäjä, jonka oikeudet saavat yleensä projektin omistaja. Projektin pääkäyttäjä pystyy luomaan projektin sisällä instansseja, käyttäjiä, levykuvia, virtuaalikoneinstansseja sekä huolehtimaan virtuaalikoneinstanssien ylläpidosta. (Mts.)

4.5 Novan verkkoratkaisut

Openstack tarjoaa kahdenlaisia IP-osoitteita. Kiinteät IP-osoitteet, jotka määritellään virtuaalikoneelle sillä hetkellä kun se luodaan ja sama IP-osoite pysyy myös kyseisellä virtuaalikoneella, kunnes se tuhotaan. Toisena tapana jakaa IP-osoitteita on niin sanotut kelluvat osoitteet, jotka liitetään ja otetaan pois virtuaalikoneilta dynaamisesti. Kelluvat IP-osoitteet on tarkemmin esiteltynä kappaleessa 4.7. Jokaisen fyysisen palvelimen eli noden verkkokontrolleri luo virtuaaliverkot ja käyttää siltaamista virtuaaliverkkojen yhdistämisessä toisiinsa ja julkiseen verkkoon.

Tarjolla on kolme eri tapaa verkottaa OpenStackin sisällä olevat virtuaalikoneet ja projektit: Flat, Flat DHCP sekä VLAN DHCP. Flat modessa sekä Flat DHCP modessa ei tapahdu virtuaalikoneiden erottelua verkossa, koska näissä tavoissa jaetut IP-osoitteet kuuluvat samaan IP-osoiteavaruuteen. (OpenStack Nova documentation – Flat Network mode 2012.)

4.5.1 Flat Mode

Flat mode:ssa verkon ylläpitäjä määrittelee IP-avaruuden, josta kiinteitä IP-osoitteita jaetaan instansseille niiden käynnistyessä. IP-osoitteiden jako tapahtuu käsin. Tätä varten on konfiguroitava kaksi siltaa, yksi verkko-ohjaimeen ja yksi compute-ohjaimeen, johon kaikki virtuaalikoneet ovat yhteyksissä. (OpenStack Nova documentation – Flat Network mode 2012.)

4.5.2 Flat DHCP Mode

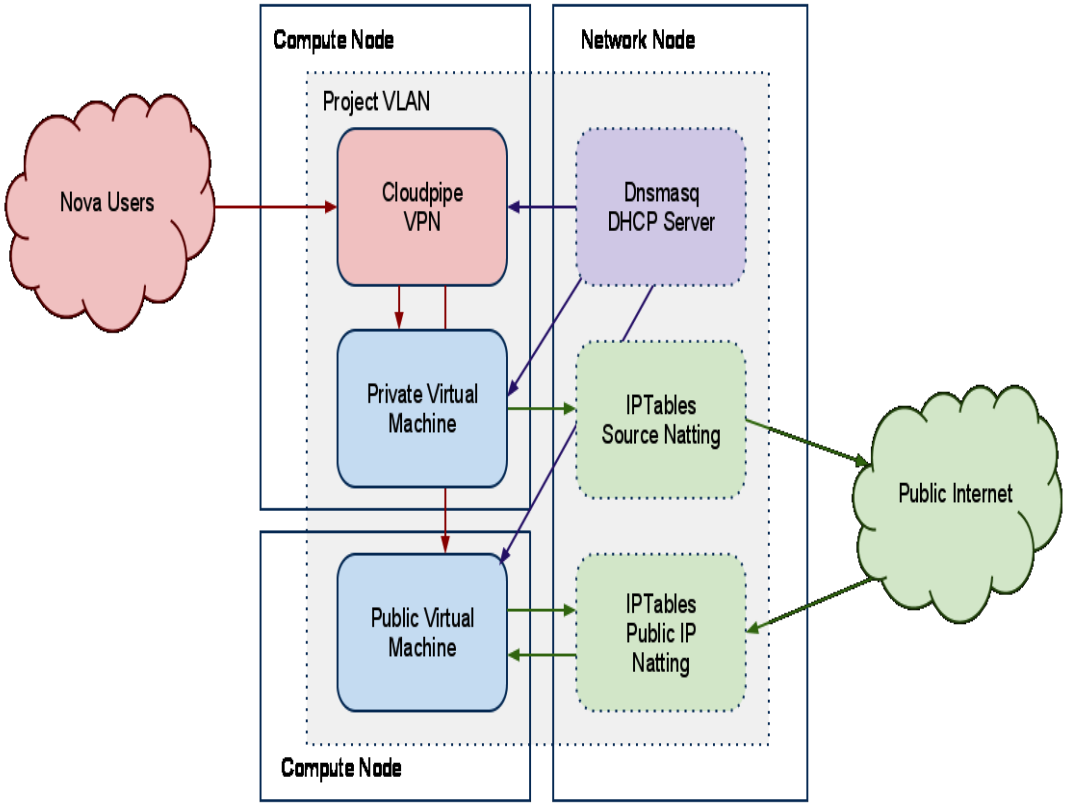
Flat DHCP mode:ssa määritellään myös tietty IP-avaruus, mutta IP-osoitteiden jako tapahtuu DHCP-serverin avulla. Dnsmasq-käsky ajetaan verkkokontrolleriin kuuntelemaan dhcpdiscover-pyyntöjä. (OpenStack Nova documentation – Flat Network mode 2012.)

4.5.3 VLAN DHCP Mode

VLAN DHCP on oletustapa tarjota verkkopalvelut OpenStackissa. Tässä tavassa tarjotaan jokaiselle projektille yksityinen verkkoalue, josta jaetaan IP-osoitteita instansseille, ja johon voidaan olla yhteyksissä projektille omistetun VPN-yhteyden kautta. VLAN DHCP:n tarkoituksena on tuoda OpenStackiin mukaan helppo sisäänrakennettu tapa erotella vuokralaisten eli projektien verkot toisistaan. Muissa tavoissa erottelut täytyisi tehdä esimerkiksi palomuurisääntöjen avulla ja verkon muita osia hallinnoimalla, esimerkiksi erillisiä VLAN-verkkoja määrittelemällä ja ynnä muiden tekniikoiden avulla. VLAN DHCP kuitenkin vaatii myös periaatteessa laitehankintoja, sillä verkosta on löydyttävä kytkin, joka tukee 802.1q-leimausta, jos kyseessä on monen fyysisen laitteen OpenStack-ympäristö. Tämän ei kuitenkaan pitäisi olla ongelma, sillä 802.1q on laajasti käytössä oleva standardi. VLAN-leimausta tukemattomia laitteita ei enää nykyaikaisissa verkoissa pitäisi olla. Jopa tavalliset kuluttajille suunnatut laitteet tukevat 802.1q-leimoja. Virtuaalikoneet voivat jakautua fyysisesti eri laitteille niiden ollessa kuitenkin samaa asiakasta ja samaa VLAN-verkkoa. (OpenStack Nova documentation – VLAN Network mode 2012.)

Jokaiselle projektille luodaan oma VLAN, verkkosilta sekä aliverkko. Aliverkot ja käytettävät VLAN-leimat määrittelee verkon ylläpitäjä ja ne jaetaan dynaamisesti projekteille sitä mukaa kun niitä luodaan. Jokaisella verkolla on dnsmasq-palvelu, jonka avulla IP-osoitteiden jako instansseille tapahtuu. Kaikki instanssit, jotka kuuluvat samaan projektiin sillataan samaan VLANiin projektin muiden virtuaalikoneinstanssien kanssa. Verkkoliikenteen ollessa auki saman VLANin sisällä olevien virtuaalikoneinstanssien välillä, Nova voi valvoa verkkoliikenteen erottelua eri projektien välillä asettamalla yhden VLANin per projekti. (Mts.)

Projekti saa tietyn alueen yksityisiä IP-osoitteita, joihin saadaan yhteys vain VLAN:in sisältä, ja jotta käyttäjä voisi saada yhteyden projektin sisällä oleviin instansseihin, tarvitaan tätä varten erityinen VPN-yhteys nimeltään Cloudpipe, jos halutaan että yksityistä verkkoa ei paljasteta julkiseen verkkoon. Cloudpipe on tarkemmin esiteltynä kappaleessa 4.6. Instanssiin voidaan päästä käsiksi myös kelluvien IP-osoitteiden avulla. Tällöin instanssi kuitenkin tulee näkymään julkisessa verkossa, sillä kelluvat IP-osoitteet ovat julkisia IP-osoitteita. Kelluvat IP-osoitteet on esiteltynä tarkemmin kappaleessa 4.7. Kuviosta 9 nähdään miten kommunikointi VLAN-verkon ja julkisen verkon välillä tapahtuu.

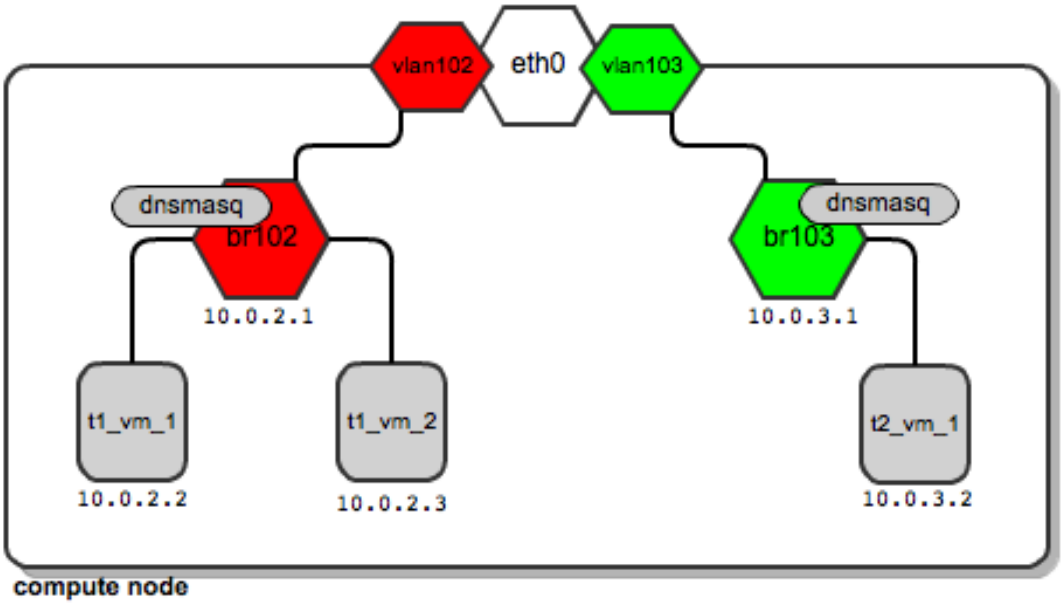


Kuvio 9. VLAN DHCP MODE (OpenStack Nova documentation – VLAN Network mode 2012)

VLAN on merkattu sinisillä katkoviivoilla ja kaksi pilveä esittävät julkista verkkoa. Käyttäjä ottaa yhteyden Cloudpipe VPN-palveluun, josta käyttäjälle jaetaan IP-osoite DHCP-palvelimen toimesta. Virtuaalisen privaattikoneen ja virtuaalisen julkisen koneen välillä tapahtuu NAT-muunnos, jonka avulla saadaan yhteys julkiseen verkkoon tai saadaan yhteys julkisesta verkosta.

Kuviosta 10 nähdään vielä tarkemmin miten VLAN DHCP toimii. Esimerkissä saman compute-noden sisään luodaan kaksi projektia ja näihin projekteihin luodaan in-

stanssit. Compute-node on periaatteessa yksi fyysinen palvelin, jonka sisällä on tässä tapauksessa kaksi asiakasta eli projektia. Toisella asiakkaalla on käynnissä kaksi virtuaalikonetta ja toisella yksi. Virtuaalikoneet ovat instansseja.



Kuvio 10. VLAN DHCPin toiminta (Siwczak 2012.)

Kyseisessä kuviossa kaksi projektia ovat saaneet kaksi erilaista VLAN-verkkoa itselleen. Tenant 1 on saanut VLAN-verkon tagilla 102 ja Tenant 2 VLAN-verkon tagilla 103. VLAN-verkot on luotu OpenStackin toimesta kyseisen palvelimen fyysiseen rajapintaan eth0. Eth0 rajapinnalla on yhteys OpenStack-verkon yksityiseen verkkoon. Lisäksi jokaisella projektilla on oma siltansa. DHCP-palvelimena toimiva dnsmasq jakaa IP-osoitteita instansseille sitä mukaa kun niitä luodaan. (Siwczak 2012.)

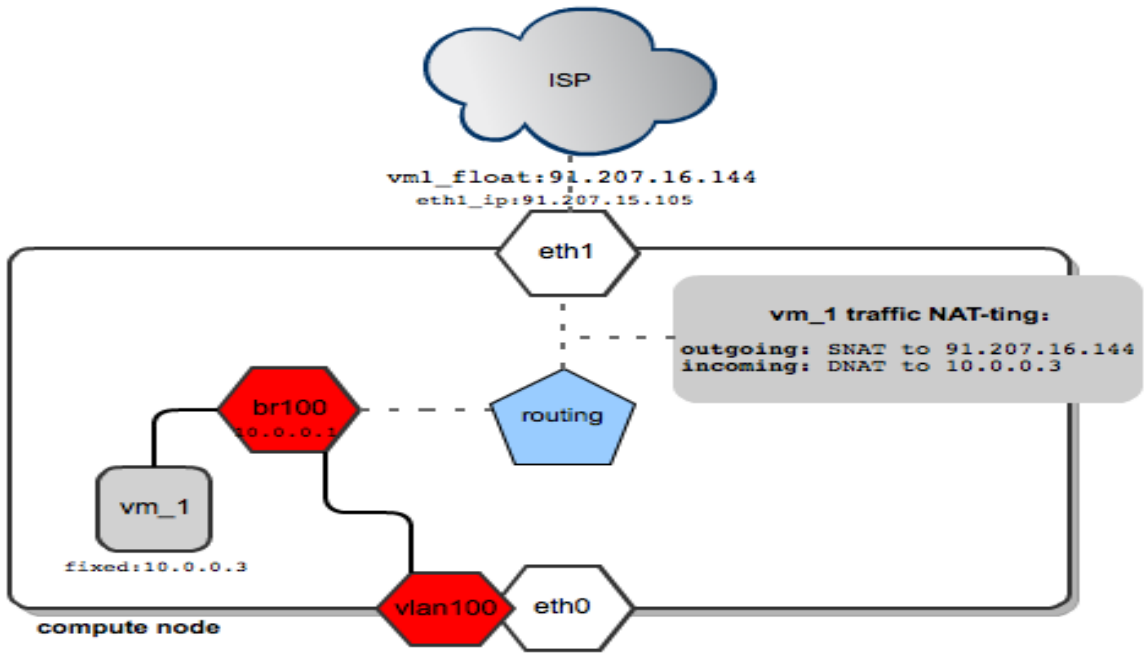
VLAN DHCP mode on ainoa järkeen otettava vaihtoehto mietittäessä minkä verkotustyypin ottaisi OpenStackiin käyttöön, kun kyseessä on hieman suurempi kuin pienen yhteisön kanssa jaettava pilvi-infrastruktuuri. Muut verkotustavat lähinnä soveltuvat OpenStackin testaamiseen. OpenStackin tavoitteet tällä VLAN DHCP-verkkotyypillä ovat sellaiset, että jokaisella projektilla olisi suojattu verkko-osionsa ja seuraavien määrittelyjen olisi tarkoitus käydä toteen, että tähän tavoitteeseen päästäisiin:

1. RFC-1918 mukainen IP-osoiteavaruus.
2. Julkinen IP-osoite NAT-muunnoksen avulla.
3. Oletuksena ei sisäänpäin tulevaa Internet liikennettä ilman julkista NATia.

4. Rajoitettu ulospäin suuntautuva Internet-liikenne (projektin ylläpitäjän määriteltävissä).
5. Rajoitettu liikenne projektien välillä (projektien ylläpitäjien määriteltävissä).
6. Kaikki yhteydet instansseihin ja pilvi API-rajapintoihin VPN:n avulla. VPN osoitetaan tiettyyn projektiin.

4.6 Kelluvat IP-osoitteet

Kelluvat IP-osoitteet ovat sellaisia osoitteita, joita järjestelmänvalvoja on määrittänyt käytettäväksi, ja joita voidaan myöhemmin dynaamisesti jakaa asiakkaiden eli projektien kesken. Kelluvat IP-osoitteet ovat pääasiallisesti julkisia IP-osoitteita joita käytetään, kun instanssi halutaan tuoda esiin julkiseen verkkoon. VLAN DHCP:ssa projektin instansseihin ei ole mahdollista ottaa yhteyttä muuta kuin kelluvien IP-osoitteiden tai VPN-yhteyden avulla. Käyttäjä voi projektin sisällä antaa kelluvan IP-osoitteen haluamalleen instanssille. Koko OpenStack-verkon ylläpitäjä määrittelee tietyn alueen IP-osoitteita, joita käyttäjät voivat ottaa käyttöönsä virtuaalikoneisiinsa, joko uudelleenkäynnistyksen yhteydessä tai dynaamisesti. Kelluvat IP-osoite osoitetaan tiettyyn instanssiin NAT-muunnoksen avulla projektin sisällä olevan instanssin yksityiseen IP-osoitteeseen. Kelluvat IP-osoitteet toimivat kuvion 11 mukaisella tavalla.



Kuvio 11. Kelluvien IP-osoitteiden toiminta (Siwczak 2012.)

Niiden käyttöönotto ei suoraan vaikuta instanssin toimintaan vaan kaikki konfiguraatiot mitä tarvitaan, tapahtuvat compute-nodessa itsessään. Nova-network luo NAT-muunnoksen kelluvan osoitteen ja instanssiin kiinteän yksityisen osoitteen väliin. Kuviossa 11 palvelin on varustettu kahdella verkkokortilla: eth0 ja eht1. Eth1-rajapinnasta on suora yhteys julkiseen verkkoon ja eth0-rajapinta on OpenStackin sisäistä liikennöintiä varten. Eth1-rajapinnassa sijaitsee compute-noden oletusyhdyskäytävä. NAT tekee muunnoksen sisäverkon osoitteesta julkisen verkon osoitteeksi. (Siwczak 2012.)

4.7 Cloudpipe

Cloudpipe on OpenStackissa tarjolla oleva keino, jolla loppukäyttäjät voidaan yhdistää heidän tarvitsemaansa projektiin turvallisesti VPN-tekniikan avulla, kun käytössä on VLAN DHCP-verkotustapa. Cloudpipen avulla mahdollistetaan loppukäyttäjiä ottamaan VPN-tekniikalla yhteys projektinsa yksityiseen verkkoon ja sen sisällä oleviin instansseihin. Pääsy tähän VPN-tunneliin on mahdollista julkisen kelluvan IP-osoitteen ja määritellyn julkisen portin avulla. Tämä antaa käyttäjille mahdollisuuden päästä projektiensa virtuaalikoneisiin ilman, että virtuaalikoneinstanssit näkyvät avoimesti julkisen verkon puolelle. OpenStack painottaa, että käyttäjillä on täysin vapaat kädet määritellä itse omat tietoturvaratkaisunsa käytettäväksi OpenStackin kanssa. Tämänkaltaisissa pilvi-infrastruktuureissa tarvitaan kuitenkin keskitettyä hallintaa verkosta ja sen liikennöinnistä, joten OpenStackin kehittäjät päättivät sisällyttää Cloudpipe-toiminteen OpenStackiin. (OpenStack Nova documentation – Cloudpipe 2012.)

Cloudpipe on kaikessa yksinkertaisuudessaan OpenVPN:llä ja Linux-käyttöjärjestelmällä varustettu virtuaalikoneinstanssi. Cloudpipe tarvitsee vain yksinkertaisen skriptin saadakseen käyttäjädataa metadata-palvelimelta, b64-koodatakseen saadut tiedot zip-tiedostoksi ja käynnistääkseen autorun.sh skriptin zip-tiedoston sisältä. Autorun-skriptin tarkoituksena on konfiguroida ja käynnistää OpenVPN käyttäen Novalta saatua dataa. (Mts.)

Cloudpipen käyttöönotto on esitetty yksityiskohtaisemmin liitteessä 3. Pääpirteittäin Cloudpipen käyttöönotto aloitetaan luomalla instanssi ja asentamalla Ubuntu, jonka

pohjalla on OpenVPN-sovellus, instanssiin. Kun tarvittavat konfiguraatiot on tehty, ilmoitetaan tästä luodusta instanssista Glanceen, josta tenantit/projektit voivat sen myöhemmin ottaa mukaan verkkoihinsa. Asetetut konfiguraatiot ovat siis kaikkiin projekteihin sopivia, ja Cloudpipen ajamien skriptien avulla ne tulevat vastaamaan kyseisen projektin verkkoa. Lisäksi on ilmoitettava Novan konfigurointi-tiedostoon tämän kyseisen luodun Cloudpipe-instanssin olemassaolo. Tämä jälkeen kyseinen instanssi käynnistetään ja Cloudpipelle asetetaan yksi kyseiselle asiakkaalle määritetyistä kelluvista IP-osoitteista. Cloudpipeen pääsy onnistuu esimerkiksi OpenVPN-sovelluksella. Käyttäjälle on jaettava projektiin luodut avaimet sekä IP-osoite, johon käyttäjä ottaa yhteyden. VPN-yhteyden ottavat käyttäjät saavat IP-osoitteen projektille määrätystä aliverkosta ja ovat näin täten osaa samaa yksityistä verkkoa. (Mts.)

Cloudpipea käynnistettäessä käyttäjälle, Nova etenee seuraavan prosessin mukaisesti:

1. Luodaan VPN avainpari tälle nimenomaiselle projektille ja avaimet tallennetaan avain-hakemistoon.
2. Luodaan uusi security-ryhmä ja portti 1194 ja ICMP-paketit on sallittu tälle ryhmälle. Novan pitäisi tehdä tämä automaattisesti.
3. Muut valtuutukset kuten sertifikaatti ja privaattiavain luodaan tälle VPN-instanssille ja tallennetaan hakemistoon CA/projects/${project id}
4. Kaikki tieto pakataan zip-tiedostoon ja koodataan base-64:sella
5. Käynnistetään m1.tiny instanssi eli pienin mahdollinen virtuaalikoneinstanssi

VLAN DHCP verkkototeutuksessa, jokaisen privaattiverkon IP-osoiteavaruuden toinen osoite varataan Cloudpipea varten. Tällä tavoin taataan yhdenmukainen IP-osoitteistus luotaville Cloudpipe-instansseille, jolloin nova-network pystyy luomaan ulkoverkosta eli Internetin yli tulevan liikenteen uudelleenohjaussääntöjä NAT-tekniikan avustamana. Jokaisen projektin verkolle annetaan tietty korkealukuinen portti, joka toimii yhdessä projektille määritellyn julkisen IP-osoitteen kanssa osoitteena VPN-yhteyksiä varten. Julkiset IP-osoitteet ovat tässä tapauksessa OpenStackiin käytettäväksi määriteltyjä kelluvia IP-osoitteita. Tähän osoitteeseen tuleva liikenne ohjataan automaattisesti OpenVPN:n käyttämään porttiin 1194. Porttien arvoja ja IP-osoitteita voidaan vaihtaa, mikäli niiden käytössä ilmenee syystä tai toi-

sesta ongelmia. Tämä onnistuu komennolla ”nova-manage vpn change [new_ip] [new_port]”, jossa new_ip kohtaan ilmoitetaan uusi käytettävä IP-osoite ja new_port kohtaan uusi käytettävä portti.(Mts.)

Cloudpipen turvalliseen toimintaan vaaditaan myös lipun use_project_ca käyttöä. Sen avulla jokainen projekti saa itselleen oman CA:n (Certificate authority). Tätä CA:ta käytetään sertifikaatin allekirjoittamiseen VPN:ää varten ja se jaetaan myös käyttäjille, jotta nämä voivat ottaa sen käyttöön halutessaan ottaa VPN-yhteyden projektiinsa. Käytöstä poistettuja sertifikaatteja hallitaan nova-managen avulla, joka luo uuden listan käytöstä poistetuista sertifikaateista eli CRL-listan. Niin kauan kun Cloudpipella on päivitetty CRL-lista, se estää listalla olevien käyttäjien pääsyn VPN:ään. Tällä hetkellä Cloudpipen käyttäjädata ei päivity automaattisesti kun sertifikaatteja otetaan pois käytöstä, joten sen vuoksi on tärkeää uudelleen käynnistää Cloudpipe kun käyttäjien tunnuksia on otettu pois käytöstä tai ne ovat vanhentuneet. (Mts.)

Ottaessa Cloudpipea käyttöön tulisi huomioida seuraava asia, että käyttöönotto sujuisi mahdollisimman mutkattomasti. Ylläpitäjän täytyy osata määritellä sopivan kokoinen IP-avaruus VPN:n kautta yhteyden ottavia VPN-clienteja varten. Ettei IP-osoitteita jaettaisi samasta avaruudesta myös instansseille, voidaan tietty arvo määritellä rivin --cnt_vpn_clients avulla. Ne IP-osoitteet jotka on määritelty päätelaitteiden käyttöön, ei jaeta enää instansseille. (Mts.)

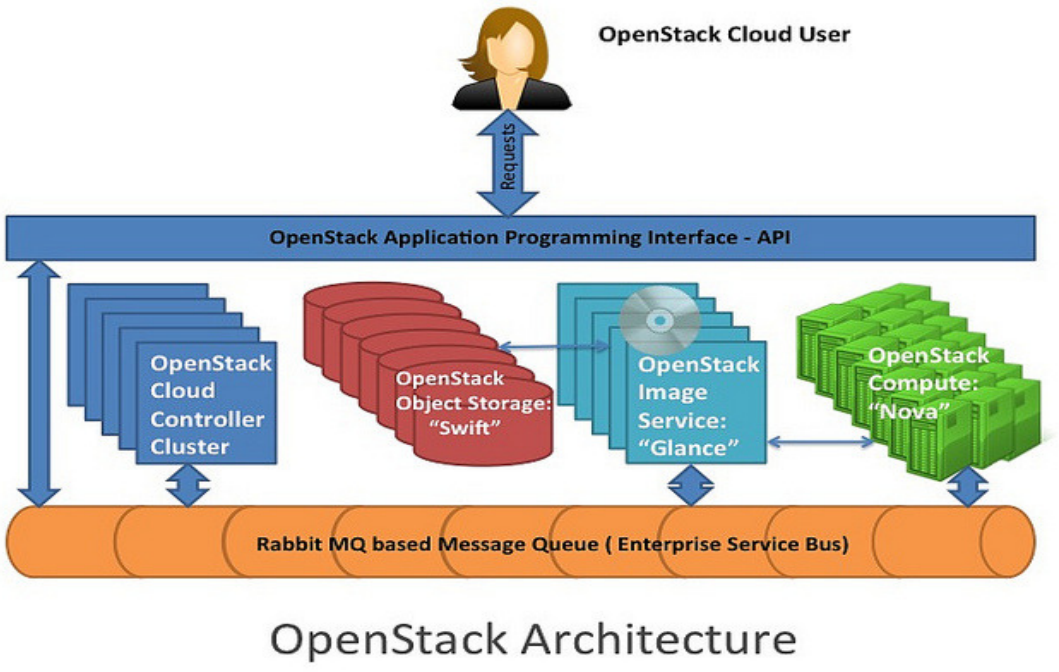
Cloudpipe on melko turvallinen tapa tarjota asiakkaalle yhteys omaan projektiinsa ja sen instansseihin. Cloudpipe ei kuitenkaan ole toimiva ratkaisu siinä kohdin, jos halutaan että jokaiselle etäkäyttäjällä olisi omat tunnuksensa vaan Cloudpipe on projektikohtainen ja samaan projektiin yhteyden ottavat etäkäyttäjät ovat saman sertifikaatin alaisuudessa. Avaimien jakaminen on helpompaa, sillä samaan projektiin kuuluvat käyttäjät saavat samat avaimet käyttöönsä. Huonona puolena tässä on se, että yksittäisten käyttäjien tunnuksia ei pystytä seuraamaan eikä täten myöskään kuolettamaan poistettujen käyttäjien oikeuksia päästä projektiin käsiksi. Cloudpipe voisi olla ratkaisu erilaisiin niin sanottuihin pop up-projekteihin. Toisaalta voisi olla aiheellista tutkia asiaa eteenpäin, että olisiko mahdollista tarjota erilaisia sertifikaatteja eri käyttäjille, jolloin käyttäjien hallinta olisi helpompaa.

4.8 OpenStack – Quantum

OpenStackissa on kirjoitushetkellä menossa vaihe, jossa pyritään tuomaan OpenStackin kolmen pääkomponentin lisäksi mukaan vielä yksi tärkeä komponentti, Quantum, joka tulisi hoitamaan OpenStackin verkkoliikenteen kulun ja hallinnan. Tällä hetkellä tämän puolen on OpenStackissa hoitanut Nova, mutta Nova on hoitanut myös OpenStackin laskennallisen osion eikä verkkoliikenteen hallinta Novalla ole kovin edistynyttä verraten pilvipalvelun asiakkaiden tarpeisiin. Quantum on kirjoitushetkellä mahdollista ottaa käyttöön plugin-ratkaisuna. Tämän kappaleen tarkoituksena on esitellä Quantumin tuomat mahdollisuudet tulevaisuuden kannalta. Toteutuksen suhteen Quantumilla ei kuitenkaan ollut merkitystä.

Quantum on virtuaalinen tietoliikennöintipalvelu (virtual network service), jonka tarkoituksena on tarjota tehokas API, joka määrittelee tietoliikennöinnin ja yhteydellisyyden muiden OpenStack palvelussa olevien laitteiden kanssa. (Quantum Admin Guide. 2012.)

Kuviossa 12 nähdään OpenStackin arkkitehtuuri ennen Quantumia karkeasti esitettynä. OpenStack Cloud Controller Cluster huolehtii koko OpenStack järjestelmää ja osien väliseen keskusteluun OpenStack käyttää Rabbit MQ:ta.



Kuvio 12. OpenStack arkkitehtuuri ennen Quantumia (OpenStack , Quantum and Open vSwitch – Part I 2011)

OpenStackin verkkoliikennöinti on sinänsä toiminut mainiosti kolmen olemassa olevan verkkoliikennöinti tavan avulla, ja ne ovat varteenotettavia lähestymistapoja verkkoliikennöintiin pilvessä. Verkkoa ei kuitenkaan palvella kuin ensimmäisen luokan kansalaista kuten computea ja storagea ja se syö verkkoarkkitehtuurin muokattavuutta. OpenStackin suhteen sillä on Novan sisäisellä verkotuksella on omat rajoitteensa:

1. Rajoitetut tietoverkkomahdollisuudet
2. Ei tarkasti määriteltyä tietoverkkorajapintaa
3. Yksinkertainen verkko-malli

Rajoitetut verkotusmahdollisuudet viittaavat OpenStackin kolmeen eri verkotustapaan, joista kaksi ovat hyvin samanlaisia sillä erotuksella että toisessa on käytössä DHCP. Nämä kolme tapaa ovat Flat mode, Flat DHCP mode ja VLAN DHCP mode. VLAN DHCP on OpenStackin kehittynein ratkaisu verkotukseen, mutta sillä on kuitenkin omat rajoituksensa, sillä ongelmia tulee vastaan VLAN-tekniikan kanssa. Näitä ovat muun muassa klassiset VLAN-tekniikan ongelmat eli IP-osoitteiden ja MAC-osoitteiden päällekkäisyyden tuen uupuminen, L2-tason kasvattaminen aliverkkojen yli, VLANin skaalausrajoitukset, MAC-taulujen skaalausrajoitukset ja niin edelleen. Lisäksi VLAN DHCP tukee vain yhden virtuaalisen rajapinnan käyttöä per virtuaalikone. (Mts.)

Quantumin avulla on mahdollista myös ottaa käyttöön erilaisia plugineita, joista tärkeimmät ovat tällä hetkellä Open vSwitch, joka on virtuaalinen kytkin. Sen avulla voidaan tehdä esimerkiksi L2-in-L3 tunnelointia eli VXLANeja. Tämän tekniikan avulla voidaan kuljettaa samaan pilveen kuuluvien virtuaalikoneiden välillä liikennettä vaikka virtuaalikoneet sijaitsisivatkin fyysisesti eri puolella maailmaa. Virtuaalikoneet kuuluvat tässä tapauksessa saman asiakkaan verkkoon. Tämä kyseinen Open vSwitch löytyy Quantumista sisäänrakennettuna. Lisäksi Quantumissa on Ciscon plugin valmiina, jota tarvitaan kun Quantum otetaan käyttöön ympäristössä, jossa on käytössä Ciscon UCS ja Nexus tekniikkaa/laitteistoa. (OpenStack , Quantum and Open vSwitch – Part I 2011.)

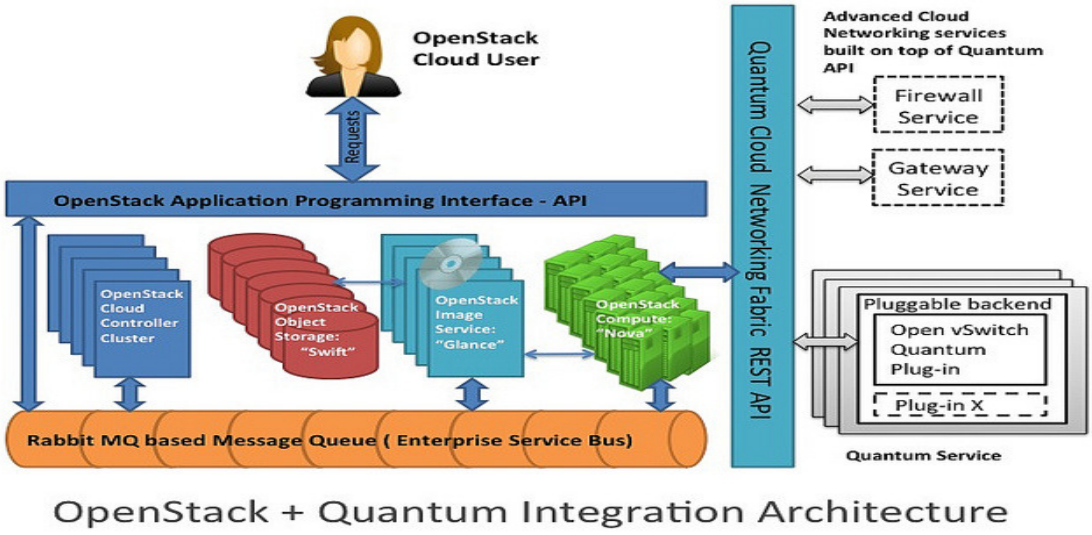
4.9 Quantum tekniset parannukset

Quantum tuo mukanaan monenlaisia parannuksia ja lisäominaisuuksia OpenStackin tavanomaiseen verkotustekniikkaan. Yksi tärkeimpiä on sen antamat mahdollisuuden vuokralaisille omien verkkotopologioiden luontiin ja mahdollisuuteen konfiguroida kehittyneitä tietoliikennesuodatussääntöjä pilven sisällä. Tämä mahdollistaa esimerkiksi monikerroksisten web-ohjelma topologian luonnin. Quantum mahdollistaa myös erilaisten pluginien käytön (avoimen ja suljetun lähdekoodin) joiden avulla voidaan toteuttaa esimerkiksi L2-in-L3 tunnelointia jolla ohitetaan VLANin rajoitukset. Tällä tavoin vapaana olevien VLAN-tagien määrä ei tule olemaan este. Lisäksi voidaan tarjota esimerkiksi end-to-end QoS:ia eli taataan palvelun laatu topologian päästä päähän. Tämän kaiken lisäksi Quantumin avulla voi kuka tahansa lähteä rakentamaan verkkopalveluita, jotka ovat yhteyksissä OpenStackin vuokralaisten verkkoihin. Tämmöisiä palveluja ovat muun muassa VPN-aaS, firewall-aaS, IDS-aaS ja niin edelleen. (OpenStack , Quantum and Open vSwitch – Part I 2011.)

Quantumin tarkoituksena olisi tuoda muun muassa seuraavia paranneltuja ominaisuuksia verrattuna OpenStackin Novaan:

1. Tarjoaa joustavan API:n sekä palveluntarjoajalle että vuokralaisille ja tarjoaa mahdollisuuden hallita OpenStackin verkkotopologioita.
2. Esittää loogisen API:n ja siihen kuuluvan plugin arkkitehtuurin. Tällä tavoin voidaan sallia virtuaalinen tietoliikennöinti virtuaalisissa kytkimissä, fyysisissä kytkimissä tai molemmissa. Tuki myös muihin pilvi-infrastruktuureihin olisi olemassa.
3. Tarjoaa API:n, joka on laajennettavissa compute APIiin, sallien plugineissa kehittyneempien tietoliikennetekniikoiden käyttöönoton kuten QoS, ACL ja niin edelleen.
4. Tarjoaa alustan kehittyneiden tietoliikenneratkaisujen integroimiseen:
    a. Olemassaolevat palomuuri-palvelut
    b. Kuormanjako-palvelut
    c. MPLS-infrastruktuuri

Kuviosta 13 nähdään miten Quantumin käyttöönottoa eroaa normaalista OpenStackin arkkitehtuurista.



Kuvio 13. OpenStack – Quantum (OpenStack , Quantum and Open vSwitch – Part I 2011)

Quantumin päärajapinta on ohjelmoitu RESTful API. Sen yläpuolella olevat abstraktiot, joita se ohjaa, ovat erittäin yksinkertaisia. Quantumin tarkoituksena on korvata OpenStackissa olevat verkotustavat yhdellä ohjelmallisella API:lla. Ideana on, että käyttäjät voisivat tämän API:n kautta pyytää haluttua verkkotopologiaa. Tällä tavoin pilven käyttäjät voisivat itse määrittää verkkotopologiansa, riippumatta infrastruktuurin asetettua verkotustapaa. (Mts.)

5 KAUPALLISET RATKAISUT

5.1 Edut ja haitat

Tarjolla näytti olevan myös valmiita kaupallisia ratkaisuja ongelmaan. Kaupallisissa ratkaisuissa haittapuolena on tietenkin niiden maksullisuus ja aina ei välttämättä ole tiedossa varsin syvällistä dokumentointia ratkaisujen toteutuksesta. Dokumentointi oli lähinnä keskittynyt ohjeisiin sekä käyttökohteiden esittelyyn. Kaupalliset tuotteet kuitenkin ovat useimmiten helppoja käyttöönotettavia ja niiden käyttöönotosta löytyi jokaiselta tämänkaltaisen palveluntarjoajalta varsin kattavat ohjeet. Asiakkaalta kuitenkin vaaditaan pilvipalvelujen osalta tietämystä jonkin verran eli varsinaisesti tietoverkoista tietämättömät eivät pääse tuotteita niin helposti käyttämään. Tämän

kappaleen tarkoituksena olisi esitellä muutama kaupallinen tuote yksityisen verkon ulkopuolelta tulevien käyttäjien turvalliseen liittymiseen pilveen niihin kuitenkaan syvemmin perehtymättä. Kaupallisissa ratkaisuissa on myös etuna niiden tarjoama tuki jatkossa.

5.2 OpenVPN Access Server

OpenVPN tarjoaa OpenVpn Access Server nimistä sovellusta. OpenVPN Access Server on joukko työkaluja, joka helpottaa VPN etäkäyttösovelluksen nopeaa käyttöönottoa. Access Server tarjoaa käyttäjälle valmiin paketin OpenVPN-palvelimen käyttöönottoon. Käyttäjän ei itse tarvitse huolehtia varsinaisesta konfiguroinnista vaan Access Server tekee ne käyttäjän puolesta. OpenVPN Access Server-sovelluksen hallinta tapahtuu selaimen kautta avattavasta hallintaohjelmasta. Loppujen lopuksi Access Server tuo ainoastaan helpomman keinon valjastaa OpenVPN palvelinkäyttöön. Käyttäjä pystyisi kyllä itsekin taitojensa mukaan konfiguroimaan tavallisen OpenVPN-sovelluksen palvelimeksi, mutta se vaatisi käyttäjältä huomattavan määrän aikaa tutustua OpenVPN-sovelluksen tarjoamiin mahdollisuuksiin.

OpenVPN Access Server ei kuitenkaan ole täysin ilmainen ratkaisu. Sovellus tarjoaa 2 käyttäjää ilman lisämaksua testejä varten, mutta jos ratkaisu halutaan ottaa laajempaan käyttöön, on ostettava lisenssiavaimia. Yksi lisenssiavain yhdelle käyttäjälle vuodessa maksaa viisi dollaria. Esimerkiksi jos ostaa 10 lisenssiavainta, mahdollistaa se 10 yhtäaikaisen käyttäjän liikennöinnin Access Serverin kautta eli maksu riippuu siitä kuinka monen käyttäjän halutaan käyttävän Access Serveriä yhtäaikaisesti. Palvelun ostajalla voi olla vaikkapa 1000 käyttäjää, mutta tarve vain kymmenelle yhtäaikaiselle käyttäjälle. (OpenVPN Pricing Guide 2012.)

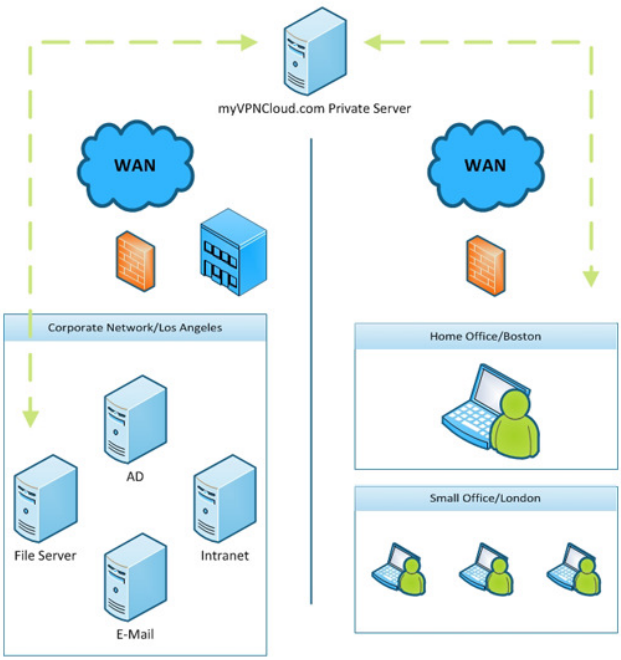
OpenVPN:n yksi monista eduista on sen helppo hallinta. Access Server tuo pääkäyttäjälle vain tärkeimmät ja hyödyllisimmät työkalut helpon web-pohjaisen rajapinnan avulla, jonka avulla palvelin saadaan pystyyn tehokkaasti ja yksinkertaisesti kuitenkaan turvallisuudesta tinkimättä. OpenVPN Access Server tarjoaa tuen ulkoisille tunnistus-tietokannoille. Käyttäjän ei tarvitse välttämättä lähteä luomaan jokaiselle VPN-käyttäjälle omia tunnuksia vaan se osaa hyödyntää mahdollisia jo olemassa olevia ratkaisuja, jotka voidaan yhdistää toimimaan Access Serverin kanssa. Yhtenä etuna

on myös helppokäyttöinen web-pohjainen ratkaisu käyttäjille. Käytännössä käyttäjän tulee vain avata web-selain, syöttää tunnukset ja VPN-palvelin on valmis konfiguroitavaksi. (OpenVPN Access Server System Administrator Guide. 2010.)

Tämän opinnäytetyön ongelman ratkaisemiseen OpenVPN saattaisi olla varsin varteenotettava vaihtoehto. Se olisi helppo käyttöönotettava ja sen konfiguroiminen tulisi tapahtumaan graafisen käyttöliittymän, mikä varmasti helpottaa asetusten muokkaamista myöhemmässä vaiheessa. Palvelin täytyisi sijoittaa ennen pilvi-infrastruktuuria ajavaa palvelinkonetta jolloin se toimisi julkisen verkon ja yksityisen virtuaalikoneiden välissä ja toimisi näin VPN-liikenteen kääntäjänä. OpenVPN Access Server ohjaisi asianmukaiset käyttäjät heille tarkoitettuihin yksityisiin verkkoihin ja estäisi asiattomien pääsyn yksityseen verkkoon. OpenVPN Access Serverin käyttöönottotapoja on monia ja vaihtoehtoja erilaisiin verkkoihin löytyy riittävästi.

5.3 MyVPNCloud ja muut

MyVPNCloud tarjoillaan asiakkaalle VPN-aaS palvelumallin mukaisesti. VPN-palvelin ei itsessään sijaitse asiakkaan verkossa. Kuviosta 14 nähdään tarkemmin MyVPN-Cloud-palvelun toiminta ja miten se asiakkaille tarjotaan.



Kuvio 14. MyVPNCloud (MyVPNCloud 2012)

Palvelu perustuu OpenVPN SSL-tekniikkaan eli MyVPNCloud on valjastanut käyttöönsä tämän ilmaiseksi tarjottavan tuotteen ja tehnyt siitä oman ratkaisunsa, jota tarjoavat asiakkailleen yrityksen määrittelemään hintaan. Palvelun hinta muodostuu palvelimen ja yhdistettyjen pisteiden välillä kulkevan datan määrän mukaan. Lisäksi palvelulla on jokin kiinteä kuukausimaksu jonka tarkempi summa selviää ottamalla yhteyttä kyseessä olevaan yritykseen. (MyVPNCloud 2012)

Tarjolla on myös monia muita samankaltaisia palveluita kuin MyVPNCloud. Ratkaisut keskittyvät monesti siihen, että asiakkaan päähän ei tehdä mitään asennuksia vaan VPN:ää tarjotaan ikään kuin palveluna, VPN-aaS (VPN as a Service), jossa itse VPN-palvelin sijaitsee useimmiten palveluntarjoajan omassa pilvessä. Muista palveluista mainittakoon Clearpath ja Aerohive Networks niihin kuitenkaan enempää puuttumatta.

6 OPINNÄYTETYÖN RATKAISU

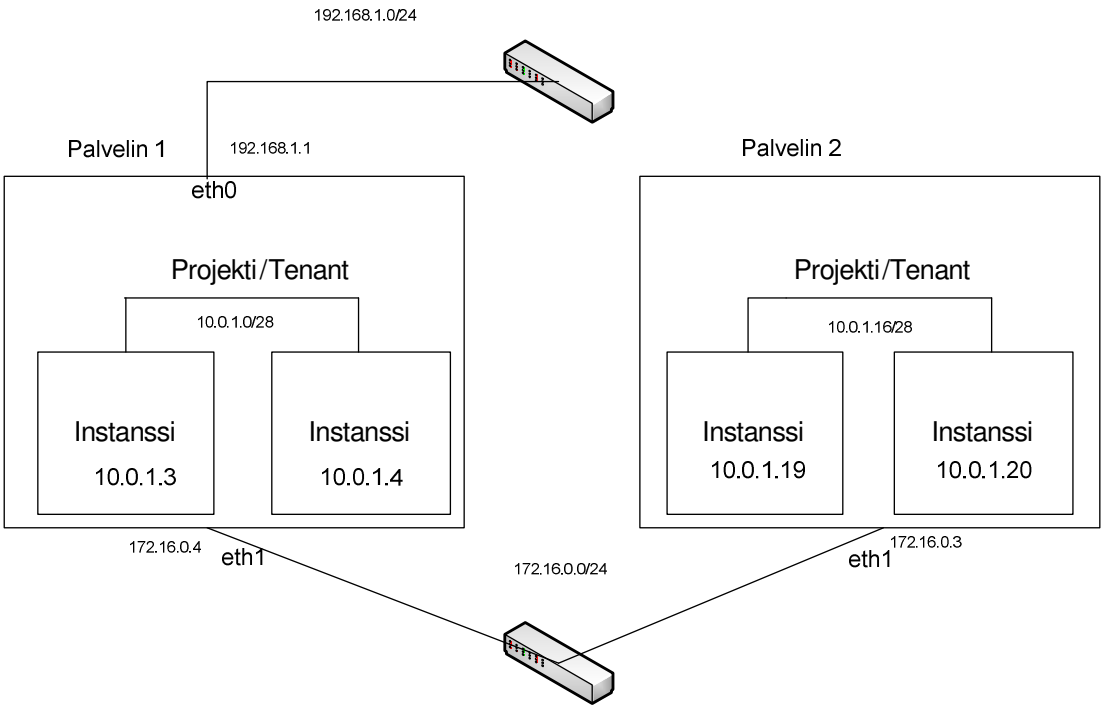
6.1 OpenStackin verkon toiminnan kertaus

OpenStack voidaan rakentaa yhden palvelimen varaan. Tämä kuitenkaan ei ole suotavaa, koska yhden palvelimen varaan ei kannata laskea koko OpenStackin toimintaa. Yhden palvelimen toteutukset ovatkin usein ainoastaan OpenStackin testaamiseen soveltuvia. Suositeltavaa olisi, että koko OpenStack-ympäristö koostuisi useammasta kuin kahdesta palvelinkoneesta. OpenStackissa on kaksi käsitettä fyysisille verkoille, julkinen ja yksityinen. Julkisen verkon kautta OpenStackissa ajettava virtuaalikoneet saavat yhteyden Internetiin tai Internetistä voidaan ottaa yhteys virtuaalikoneisiin. Yksityinen verkko on tarkoitettu OpenStackin sisäistä liikennöintiä varten. Yksityisen verkon avulla voidaan yhdistää eri fyysiset laitteet osaksi kokonaisuutta.

OpenStackin tarjotessa kolmea erilaista tapaa virtuaalikoneiden verkotukseen, paras näistä tavoista on VLAN DHCP. Tässä verkotuksessa luodaan käytettävät VLAN-verkot konfiguroimalla ne liitteen 1 mukaisella tavalla. OpenStackin pääkäyttäjä määrittelee nämä verkot ja niissä käytettävät IP-osoiteavaruudet. Määriteltävät IP-osoiteavaruudet ovat yksityisiä osoitealueita. Ainoat rajoittavat tekijät ovat käytettävät IP-osoitteet sekä käytettävät VLAN-leimat. VLAN-leimoja kun on rajallinen määrä

eikä IP-osoitteitakaan loputtomasti ole. OpenStackissa projektit, joita voidaan kutsua myös nimellä tenantti, saavat pääkäyttäjän määrittelemät VLAN-verkot sitä mukaa kun uusia projekteja luodaan. Projektit ovat asiakkaita ja nämä projektit sisältävät ajettavat virtuaalikoneet eli instanssit. Tähän projektin sisäiseen verkkoon voidaan liikennöidä vain OpenStackin yksityisen verkon kautta ja vain samaan VLANiin kuuluvat instanssit voivat liikennöidä toistensa kanssa. Projektin sisällä oleva instanssit voivat keskustella toistensa kanssa, koska ne kuuluvat samaan aliverkkoon. VLAN-verkkojen avulla mahdollistetaan eri asiakkaiden erottelu toisistaan. FreeNest-instansseja saisi olla vain yksi per projekti, jos halutaan opinnäytetyössä halutun erottelun tapahtuvan. Yksi projekti voi sisältää enemmän kuin yhden instanssin, mutta instanssit tulisivat olemaan kytköksissä toisiinsa ja kuulumaan samaan aliverkkoon.

Kuviosta 15 nähdään miten projektit ja instanssit toimivat. Projektin sisällä olevat instanssit ovat kytköksissä toisiinsa ja ovat samassa aliverkossa. Kuviossa olevat verkot ovat ainoastaan esimerkkiä varten ja ovat aina verkon käyttäjän määriteltävissä. Kuviossa ylin kytkin on yhteys julkiseen verkkoon. Alin kytkin toimii OpenStackin sisäisessä liikennöinnissä.



Kuvio 15. OpenStack verkko

Vaikka koneet ovatkin fyysisesti eri laitteissa, ovat ne OpenStackin näkökulmasta samaa virtuaalista kokonaisuutta. Palvelin 2 voi liikennöidä julkiseen verkkoon Palvelimen 1 kautta, liikenteen kulkiessa OpenStackin yksityisverkkoa pitkin.

OpenStackista löytyy myös kelluvat IP-osoitteet. Nämä IP-osoitteet ovat tarkoitettu julkisen verkon toimintaa varten. Optimitilanteessa nämä kelluvat IP-osoitteet olisivat oikeita julkisia IP-osoitteita, vaikkapa operaattorin kautta saatuna, jotka voidaan osoittaa tiettyihin virtuaalikoneinstansseihin. Kelluvilla IP-osoitteilla saadaan osoitettua esimerkiksi tietty instanssi näkymään julkiseen verkkoon. OpenStack pystyy tekemään tämän NAT-muunnoksen avulla. Kelluvia IP-osoitteita tarvitaan, että voidaan ottaa käyttöön OpenStackista löytyvä Cloudpipe. Useimmiten ei kuitenkaan ole mahdollisuutta saada operaattorilta kiinteitä julkisia IP-osoitteita käyttöön, varsinkin jos kyseessä on pieni kotikutoinen OpenStack-pilvi, jossa on käytettävissä ainoastaan yksi julkinen IP-osoite. Tässä tapauksessa joudutaan tekemään myös NAT-muunnos verkossa löytyvässä reitittimessä osoittamaan kelluvaan IP-osoitteeseen. Kelluvien IP-osoitteiden konfigurointi tapahtuu liitteen 2 mukaisesti. Jäljelle jää enää haluttujen projektien ja instanssien luonti, joihin ei ole tässä työssä otettu kantaa.

6.2 OpenVPN Access Server

Yhtenä ratkaisuna olisi perustaa verkon topologiassa OpenVPN Access Server ennen virtuaalikoneita, mutta kuitenkin niin että kyseiseen palvelimeen voidaan ottaa yhteys julkisen verkon puolelta. OpenVPN Access Server työkalujen avulla voitaisiin ohjata liikenne haluttuun virtuaalikoneinstanssiin sen tiedon varjolla, että jokaiselle virtuaalikoneinstanssille voidaan määrittää niin sanottu julkinen osoite kelluvien IP-osoitteiden avulla. Julkinen osoite olisi tässä tapauksessa näennäisesti julkinen. OpenVPN Access Serveriä hallitaan graafisesti selaimen kautta ja sen konfigurointi pitäisi olla helppoa.

6.3 OpenVPN palvelin

OpenVPN palvelin toimii samalla periaatteella kuin OpenVPN Access Server, mutta se on käyttäjän kannalta hieman haastavampi konfiguroitava. Access Serverin tehdessä säännöt automaattisesti graafisen käyttöliittymän avustuksella, vaatii OpenVPN palvelimen perustaminen käyttäjältään enemmän. Käyttäjä joutuu komentorivien avulla

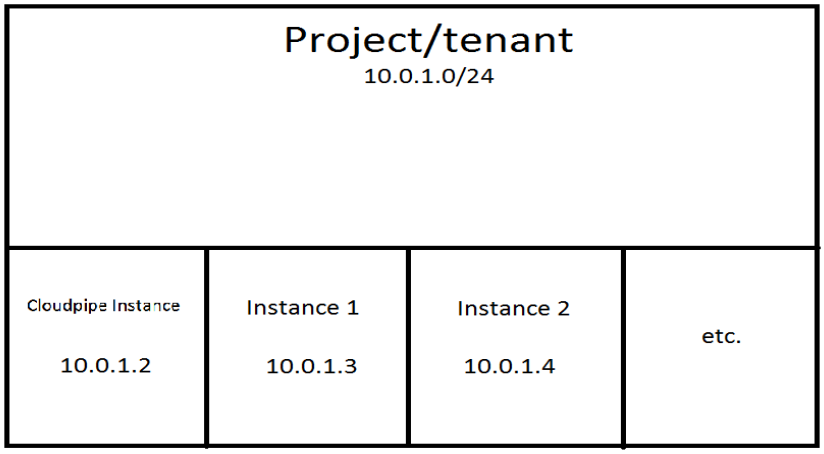
luomaan erilaisia palomuurisääntöjä sekä muokkaamaan OpenVPN:n konfigurointitiedostoja halutunlaiseksi. Ei kuitenkaan ole järkevää lähteä niin sanotusti keksimään pyörää uudelleen, sillä OpenStackissa tarjottava palvelu Cloudpipe perustuu jo valmiiksi OpenVPN:ään ja se pystyy itse muodostamaan tarvittavat palomuurisäännöt sekä osoitemuunnokset määriteltyjen verkkojen mukaan.

6.4 Cloudpipe

Cloudpipe oli näistä kaikista ratkaisuista kaikkein kustannustehokkain. Se ei sisältänyt minkäänlaisia lisähankintoja vaan kaikki tarvittavat ominaisuudet löytyivät valmiina OpenStack-ympäristöstä kattavan ohjeistuksen saattamana. Sen avulla pystyttiin tuomaan käyttäjät VPN-tekniikan avulla projektiensa/tenanttiensa sisäiseen verkkoon VLAN DHCP-verkotuksessa (tunnetaan uusimmissa dokumenteissa myös nimellä VLANManager). Hyvinä puolina Cloudpipella olivat suhteellisen helppo käyttöönotto kattavan ohjeistuksen ansiosta sekä OpenStackin automaatio Cloudpipen konfigurointien suhteen. Huonona puolena kuitenkin voitiin pitää sitä, että yksittäisiä käyttäjiä ei Cloudpipen avulla pystytty erottelemaan vaan luotavat avaimet ja sertifikaatit olivat projekti/tenantti-kohtaisia. Tämä myös siksi, että OpenStackin dokumentaation mukaan palvelu olisi näin turvallisempi. Yhtenä miinuspuolena voidaan pitää sitä, että Cloudpipen olleessa yksi instansseista, tulee se syömään resursseja jonkin verran. Menetyksen ei pitäisi kuitenkaan olla suuri, sillä palvelin toimii kyllä pienilläkin resursseilla.

Cloudpipen konfigurointi käydään yksityiskohtaisemmin läpi liitteessä 3. Cloudpipen konfigurointi aloitetaan luomalla yksi virtuaalikoneinstanssi, jonka pohjalla on jokin linux-käyttöjärjestelmä ja siihen asennetaan OpenVPN-sovellus. OpenVPN conf-tiedosto määritellään liitteen 3 ohjeistuksen mukaisesti. Sen jälkeen kun virtuaalikoneinstanssi on saatu luotua, tehdään siitä kopio Glanceen, joka on kaikkien projektien/tenanttien haettavissa. Aina kun luodaan uusi projekti/tenant, voidaan tämä VPN-instanssi hakea Glancesta tätä projektia/tenantia varten. Cloudpipe-instanssi sijoitetaan projektin/tenantin sisään. Tämän jälkeen määritellään kyseiselle Cloudpipe-instanssille kelluva IP-osoite. OpenStack tekee NAT-muunnoksen kelluvan julkisen IP-osoitteen ja Cloudpipen ykstyisen IP-osoitteen välillä. Cloudpipen pitäisi nyt olla toimintakunnossa ja valmiina käytettäväksi. Seuraavaksi jaetaan käyttäjille avaimet ja

sertifikaatti. Asiakas ottaa yhteyden Cloudpipe-instanssille määriteltyyn julkiseen IP-osoitteesee ja porttiin. OpenStack tekee NAT-käännöksen julkisesta IP-osoitteesta instanssin yksityiseen IP-osoitteeseen. Kuviossa 16 esitetään, miten Cloudpipe sijoittuu yksittäisen projektin sisään. Esimerkissä ei ole otettu huomioon VPN-asiakkaille jaettavia IP-osoitteita.



Kuvio 16. Cloudpipe

6.5 Vaihtoehtojen vertailua

Taulukosta 1 nähdään vertailua eri ratkaisujen välillä. Huomioon on otettu muun muassa sellaisia seikkojen kuten ohjeistus, käyttöönotto, lähdekoodin luokka, muokattavuus, käyttäjältä vaatimat taidot sekä kustannukset. Ohjeistuksessa on otettu huomioon ratkaisun tarjoamat ohjeistukset kuten erilaiset dokumentaatiot ja sovelluksen käyttöohjeet. Käyttöönotolla tarkoitetaan, että miten nopeasti ratkaisu saadaan otettua käyttöön ja millaisia ponnisteluja se käyttäjältään vaatii. Avoin/suljettu lähdekoodi viittaa ohjelman saatavuuteen. Muokattavuudella tarkoitetaan sitä, että voiko ratkaisua muokata helposti omanlaisekseen. Käyttäjän taidot kohdassa haetaan ratkaisun käyttäjäystävällisyyttä eli onnistuuko käyttöönotto perustason käyttäjältä vai vaaditaanko ratkaisun käyttöönottoa varten enemmän tietämystä tietoverkkotekniikoista ja sovelluksista. Kustannukset kohdan pitäisi olla melko itsestäänselvyys eli mitä ratkaisu tulee käyttäjälleen maksamaan.

Taulukko 1 Ratkaisujen vertailua

	Cloudpipe	OpenVPN AS	OpenVPN Palvelin	MyVPNCloud/muut kaupalliset ratkaisut
Ohjeistus	Kattavat	Kattavat	Kattavat, mutta vaatii käyttäjältä tietämystä eri osa-alueista	Heikot ohjeistukset, ratkaisu palveluntarjo-ajan puolesta
Käyttöönotto	Keskivaativa	Vaativahko	Vaativa	Helppo
Avoin / Suljettu lähdekoodi	Avoin	Avoin, mutta lisenssit ostettava	Avoin	Suljettu, mutta perustuu avoimeen
Muokattavuus	Muokattavissa, vaatii tietämystä OpenStackista ja OpenVPN:stä	Muokattavissa graafisella käyttöliittymällä	Täysin muokattavissa	Muokattavissa palveluntarjoajan kautta
Käyttäjän taidot	Vaatii käyttäjältä tietämystä eri tekniikoista, mutta ohjeistuksen avulla helppo	Graafinen käyttöliittymä helpottaa konfigurointia	Vaatii käyttäjältä enemmän kuin pe-rustason tietämystä	Käyttäjältä ei vaadita erityistä osaamista.
Kustannukset	Ilmainen	Kaksi ilmaista lisenssiä, loput 5 dollari kappale	Ilmainen	Neuvoteltavissa palveluntarjoajan kanssa

6.6 Toteutus

Parhaimmat tavat ongelman ratkaisemiseksi olivat OpenVPN-palvelimen perustaminen tai OpenStackista löytyvän Cloudpipen hyödyntäminen, joka myös perustuu OpenVPNään, ja se löytyy valmiiksi sisäänrakennettuna OpenStackin jo valmiiksi laajasta toimintojen kirjosta. Toteutettavaksi ratkaisuksi valittiin Cloudpipe sen laajan ohjeistuksen ja sisäänrakentuneisuutensa vuoksi. Cloudpipen käyttöönottoa koskien löytyi hyvinkin kattavat ja valmiit dokumentaatiot OpenStackin omasta ohjeistuksesta.

Työn toteutus jätettiin OpenStack-pilven käynnistämistä varta vasten perustetulle HeartCloud-tiimille. Cloudpipen konfiguraatiot perustuivat täysin OpenStackin viralliseen dokumentaatioon. Toteutettava OpenStack-pilvi tulee olemaan yksityinen pilvi, koska palvelu sijaitsee omassa yksityisessä verkossaan eikä esimerkiksi ulkoisen palveluntarjoajan pilvessä. Tiimillä oli käytössään niin sanottu Junkcloud eli kustannus-

tehokkaasti vanhoista laitteista koottu ympäristö. Tietokoneet olivat vanhoja, mutta kuitenkin laskentatehoiltaan ja muisteiltaan sopivia testiympäristön rakentamiseen. Tiimillä oli Junkcloudissa käytössä noin kymmenen keskusyksikköä, joista jokainen pyöritti erikseen jotain OpenStackiin kuuluvaa osiota.

HeartCloud-tiimille ohjeistettiin Cloudpipen konfiguraatio. Tällä hetkellä tiimillä oli OpenStack-pilvessä käytössään Flat DHCP-tyypin verkotus. Tämä verkotus ei kuitenkaan tue Cloudpipen käyttöönottoa, koska sen käyttöön vaaditaan että projektilla on oma aliverkkonsa määrätyllä VLAN-leimalla, joten olemassa olevaan OpenStack-ympäristöön ei ole mahdollista toteuttaa VPN:ää Cloudpipen avulla. Muuntaminen VLAN DHCP-verkotukseen olisi vaatinut huomattavan suuren muutoksen olemassa olevaan pilveen ja pahimmassa tapauksessa edessä olisi ollut jopa viikkojen asetusten uudelleenmäärittäminen. Jonkinlaista ratkaisun testaamista kuitenkin yritettiin Virtualboxin avulla siten, että OpenStackia olisi ajettu Virtualboxin sisällä. Ongelmaksi kuitenkin muodostui se, että yhteyttä Internetiin päin ei Virtualboxin kautta saatu tänä aikana toimimaan, mikä olisi tarvittu OpenVPN-paketin asentamiseen.

Testiympäristö tulisi lähteä rakentamaan täysin uudelle fyysiselle laitteistolle. Näiden kahden päivän aikana tätä ei kuitenkaan ehditty tekemään puutteellisten resurssien vuoksi. HeartCloud-tiimi oli kuitenkin sitä mieltä, että annetulla ohjeistuksella Cloudpipen luominen pitäisi onnistua. Paikanpäällä OpenStackin ympäristö yllätti toden teolla monipuolisuudellaan ja monimuotoisuudellaan.

7 POHDINTA

Opinnäytetyön aloitus oli haasteellista, koska suurin osa ajasta kului täysin uusien asioiden oppimiseen. Mitä pilvipalvelut ylipäätänsä ovat ja mitä etuja niiden käyttöönotosta on mahdollista saada? Monia uusia asioita selvisi koskien pilvipalveluita, ja niiden pohjalta oli hyvä lähteä miettimään asiaa eteenpäin. Tärkeää oli tutustua erityisesti pilvipalveluissa huomioitaviin tietoturvariskeihin ja niiden ennaltaehkäisyyn. Tärkeimmässä roolissa tietoturvan kannalta ovat käyttäjät, sillä vaikka järjestelmä olisi kuinka tietoturvallinen, saattaa vaaran aiheuttaa vähäisinkin väärinkäytös käyttäjän toimesta.

Tutkittava aihepiiri oli hyvin laaja ja moniulotteinen. Kuitenkin jo opinnäytetyön määrittelyvaiheessa päädyttiin ratkaisemaan ongelma VPN-tekniikan avulla, sillä se on luotetuin ja käytetyin tekniikka julkisen verkon yli tapahtuvan liikennöinnin salaamiseen. Ratkaistavaksi jäi, että millä tavoin VPN-tekniikka voitaisiin valjastaa käyttöön opinnäytetyön ongelman ratkaisemiseksi. Tarpeen oli siis tutustua myös itse VPN-tekniikkaan sekä tarkemmin OpenVPN-nimiseen sovellukseen, joka käyttää liikenteen salaukseen ja kuljettamiseen hieman erilaisempia tekniikoita kuin esimerkiksi muut perinteisesti käytetyt ratkaisut. OpenVPN valittiin myös siksi lähemmän tarkastelun kohteeksi, koska se oli avoimen lähdekoodin sovellus ja soveltui näin opinnäytetyössä rajattuun ehtoon käytettävien resurssien pitämisestä mahdollisimman pieninä.

Itse ongelman ratkaisun tutkiminen aloitettiin tutustumalla OpenStack-pilvialustaan ja sen tuomiin mahdollisuuksiin sillä oli tarve tietää miten se pääperiaatteiltaan toimii. OpenStack-palvelua rakentamaan lähdettäessä on mietittävä kokonaisuus läpi. Rakennettu pilvi-infrastruktuuri toimii IaaS-tasolla ja asiakkaille tarjottava palvelu SaaS-tasolla ja mikäli kyseessä on vielä vaativampi asiakas, toimii hänelle tarjottava palvelu PaaS-tasolla. Ratkaisun tulisi olla myös käyttöönoton helpottamisen vuoksi lähes läpinäkyvä asiakkaalle kuten kaikissa hyvissä palveluissa, asiakas itsessään haluaa selvitä mahdollisimman vähällä työllä, kun puheeksi tulee jonkin palvelun käyttöönotto. Kaikkea ei kuitenkaan asiakkaalle voi aina tarjota, joten kompromisseja täytyy aina tehdä sekä palveluntarjoajan että asiakkaan suunnalta.

Opinnäytetyössä kuitenkin keskityttiin tutkimaan tarkemmin OpenStackin verkkoa, koska oli tärkeää tietää miten liikennöinti OpenStackissa toimii. Selvisikin, että OpenStackissa oli mahdollisuus kolmeen erilaiseen tapaan luoda verkko virtuaaliseen ympäristöön: Flat, Flat DHCP sekä VLAN DHCP. Näistä käytetyin ja suositelluin tapa hoitaa OpenStackin verkotus on VLAN DHCP, joten se valittiin tarkemman tutkimuksen kohteeksi. VLAN DHCP:ssä jokainen projekti saa itselleen tietyn IP-osoitealueen sekä VLANin. Näihin projekteihin oli kuitenkin mahdollista päästä käsiksi vain Cloudpipen avulla. Projektin sisällä oleva instanssi oli mahdollista asettaa näkyväksi kelluvan IP-osoitteen avulla.

Mietinnän alla olivat sekä kaupalliset ratkaisut että avoimen lähdekoodin ratkaisut. Resurssien käyttö oli kuitenkin pidettävä mahdollisimman alhaisena, joten syvem-

pään analyysiin pääsi OpenVPN-sovellus. Vaihtoehtoja kuitenkin rajautui selvästi kaksi: lähdetään rakentamaan OpenVPN-palvelin alusta asti tai hyödynnetään OpenStackistä löytyvää Cloudpipea, jolla käyttäjät voitaisiin yhdistää omiin projekteihinsa VPN-tekniikan avulla. OpenVPN-palvelimen käyttöönoton kohdalla olisi mietittävänä ollut sen sijoittaminen, konfigurointi sekä virtualisoinnin tuomat haasteet. Cloudpipesta kuitenkin löytyi valmiina ratkaisu näihin kaikkiin kysymyksiin ja sitä päädyttiin testaamaan. Cloudpipe ei välttämättä tarjoa kaikkea sitä mitä työssä haettiin, mutta antoi kuitenkin sille hyvän alun ja osviittaa siihen, miten ratkaisua tarpeiden mukaan voitaisiin lähteä jalostamaan eteenpäin.

Pilvipalvelut tulevat olemaan tulevaisuuden ratkaisu numero yksi. Mietittäväksi jäävät kuitenkin pilvipalveluiden erilaiset käyttöönottomallit. Sellaisen pystyy perustamaan itse alusta asti tai sellaisen voi ostaa valmiina pakettina. Onko pilvipalveluita käyttöönottava valmis rakentamaan sellaisen IaaS-tasolta asti vai turvautuuko hän valmiiksi rakennettuun SaaS-tason palveluun? Todennäköisesti asiakas haluaa itselleen SaaS-tason palvelun. Tässä onkin tämän hetken rahakkain sijoitus eli pilvipalvelun perustaminen, jota asiakkaille tarjottaisiin. Käyttökustannukset ovat verrattain pienet ja saatava hyöty suuri. Pilvipalvelun perustajana tulee kuitenkin miettiä kaikki mahdolliset asiat läpi aina verkotuksesta tietoturvallisuuteen. Tietoturvan osuuden tärkeyttä ei pidä unohtaa, vaikka virtualisointi ja monimutkainen ympäristö tuovatkin omat haasteensa. Ei riitä, että tarjoillaan tietoturvallisia yhteyksiä vaan myös itse pilvipalvelun sisäisestä turvallisuudesta on huolehdittava, aivan kuten tavallisessakin palvelinympäristössä. Tärkeänä osana ovat myös käytössä olevat resurssit ja varsinkin minkälaisilla resursseilla toiminta voidaan aloittaa. Osana pilvipalveluiden suurta menestystä on myös vähäisempien resurssien tarve sekä ympäristöystävällisyys. Mitä vähemmän fyysisiä laitteita tarvitaan, sitä vähemmän ne myös kuluttavat energiaa. Mikä ennen pystyttiin tekemään kymmenellä fyysisellä palvelimella, voidaan nyt hoitaa yhdellä fyysisessä palvelimella virtualisoinnin avulla tehokkuudesta kuitenkaan tinkimättä.

FreeNestin tapauksessa uskoisin hankkeesta tulevan menestys. Projektityöskentelyssä on aina tarvetta uudenlaisille tavoille hoitaa asiat. SkyNEST-projektissa on mukana monia oman alansa ammattilaisia aina tietoliikennetekniikasta ohjelmointiin. Ei riitä kuitenkaan, että on olemassa pelkkä tietotaito tekniikasta vaan mukana täytyy olla

myös muita erilaisten alojen osaajia kuten markkinointi. Hyvän tuotteen kehittäminen on jo paljon, mutta pelkästään hyvä tuote ei riitä vaan sitä täytyy osata myös markkinoida oikealla tavalla. Juuri nyt FreeNest onkin mielestäni saanut itsensä esitellyksi sopivaan saumaan. Vilinä pilvipalveluiden ympärillä on nyt suurta, ja se tulee tulevaisuudessa kasvamaan yhä enemmän. FreeNest tarvitsee vielä vain kunnollisen markkinoinnin itselleen. Tämä opinnäytetyö oli vain pieni pintaraapaisu siitä kaikesta, mitä SkyNEST-projekti pitää sisällään.

LÄHTEET

Buyya, R., Broberg, J. & Goscinski, A. 2011. Cloud Computing: Principles and Paradigms. John Wiley & Sons, INC.

Cloud Security Alliance. 2011. Cloud Computing Architectural Framework. Viitattu 25.04.2012. https://wiki.cloudsecurityalliance.org/guidance/index.php/Cloud_Computing_Architectural_Framework

Cloudtweaks. 2012. The 4 Primary Cloud Deployment Models. Viitattu 27.10.2012. http://www.cloudtweaks.com/2012/07/the-4-primary-cloud-deployment-models/.

Feilner, M. 2006. OpenVPN - Building and Integrating Virtual Private Networks. Packt Publishing Ltd.

Gens, F. 2008. Defining “Cloud Services” and “Cloud Computing”. Viitattu 26.04.2012. http://blogs.idc.com/ie/?p=190.

Grance, T., Mell, P. 2011. The NIST (National Institute of Standards and Technology) Definition of Cloud Computing. Viitattu 17.04.2012. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Johnston, S. 2009. Cloud Computing Types: Public Cloud, Hybrid Cloud, Private Cloud. Viitattu 17.04.2012. http://samj.net/2009/03/cloud-computing-types-public-cloud.html.

Järvi, A., Karttunen, J., Mäkelä, T. & Ipatti, J. 2011. SaaS-käsikirja. Turku. Painosalama Oy.

Järvinen. J. 2006. Virtualisointi hajauttaa kuorman. Viitattu 16.09.2012. http://www.tietokone.fi/lehti/tietokone_1_2006/virtualisointi_hajauttaa_kuorman_2018

Kaario, K. 2002. TCP/IP-verkot. Docendo.

Laaksonen, A. 2011 Pilveen.net: Perusteet – Piirretäänpä tähän yksi pilvipalvelu. Viitattu 27.04.2012. http://www.pilveen.net/search/label/Perusteet.

Markey, S. 2012. Deploy an OpenStack private cloud to a Hadoop MapReduce environment. http://www.ibm.com/developerworks/cloud/library/cl-openstack-deployhadoop/. Viitattu 10.11.2012

Mathew, A. 2012. SECURITY AND PRIVACY ISSUES OF CLOUD COMPUTING; SOLUTIONS AND SECURE FRAMEWORK. International Journal of Multidisciplinary Research. Viitattu 24.10.2012. http://zenithresearch.org.in/images/stories/pdf/2012/April/ZIJMR/17_ZIJMR_APRIL12_VOL2_ISSUE4.pdf

MyVPNCloud. 2012. Business Users. Viitattu 31.10.2012. http://myvpncloud.com/business.html

OpenStack , Quantum and Open vSwitch – Part I. 2011. Viitattu 03.05.2012 http://openvswitch.org/openstack/2011/07/25/openstack-quantum-and-open-vswitch-part-1/

OpenStack Nova documentation – VLAN Network mode. 2012. Viitattu 17.04.2012 http://nova.openstack.org/runnova/network.vlan.html

OpenStack Nova documentation – Cloudpipe. 2012. Viitattu 24.10.2012. http://docs.openstack.org/trunk/openstack-compute/admin/content/cloudpipe-per-project-vpns.html

OpenStack Nova documentation – Flat Network mode. 2012. Viitattu 17.04.2012. http://docs.openstack.org/developer/nova/runnova/network.flat.html.

OpenVPN Pricing Guide. 2012. Viitattu 17.10.2012. https://openvpn.net/index.php/access-server/pricing.html.

OpenVPN Access Server System Administrator Guide. 2010. Viitattu 17.10.2012. http://openvpn.net/images/pdf/OpenVPN_Access_Server_Sysadmin_Guide_Rev.pdf.

O'Neill, Mark. 2011. SaaS, PaaS, and IaaS: A security checklist for cloud models. Viitattu 22.10.2012. http://www.csoonline.com/article/660065/saas-paas-and-iaas-a-security-checklist-for-cloud-models.

Pirinen, A. 2010. Pilvipalvelut: Kauas tiedot karkaavat?. Tietosuoja 3.

Pepple, K. 2011. OpenStack Nova Architecture. Viitattu 25.04.2012 http://ken.pepple.info/openstack/2011/04/22/openstack-nova-architecture/

Quantum Admin Guide. 2012. Viitattu 04.05.2012. http://docs.openstack.org/incubation/openstack-network/admin/quantum-admin-guide-trunk.pdf

Roger's Information Security Blog - VPN Split Tunneling. 2010. Viitattu 12.11.2012. http://www.infosecblog.org/2010/05/vpn-split-tunneling/

Siwczak, P. 2012. Configuring Floating IP addresses for Networking in OpenStack Public and Private Clouds. Viitattu 23.10.2012 http://www.mirantis.com/blog/configuring-floating-ip-addresses-networking-openstack-public-private-clouds/

StackOps. 2011. Install and Configure a Multinode Architecture. Viitattu 12.11.2012. http://docs.stackops.org/display/doc03/Install+and+Configure+a+Multinode+Architecture.

The Purposeful Clouds. 2012. Cloud Cube. Viitattu 18.04.2012. http://www.purposefulclouds.com/home/Cloud-Resources/cube.

Thorn, S. 2011. An evaluation of OpenStack for the ETF. Viitattu 20.4.2012. http://www.escience-etf.ac.uk/documents/OpenStack%20evaluation-FINAL.pdf/at_download/file

VPN - Virtual Private Network and OpenVPN. 2010. Viitattu 19.10.2012. http://linuxconfig.org/VPN_-_Virtual_Private_Network_and_OpenVPN.

What is OpenStack?. 2011. Viitattu 17.04.2012. http://docs.openstack.org/bexar/openstack-compute/admin/content/ch01s01.html

LIITTEET

Liite 1. OpenStack VLAN DHCP konfigurointi

OpenStackin konfiguroinnit aloitetaan yleensä määrittelemällä fyysisten rajapintojen IP-osoitteistukset. Tämä tehdään halutun suunnitelman mukaisesti, käytettävien resurssien mukaisesti sekä miten verkko halutaan rakentaa. Oletetaan, että käyttäjä on nämä haluamansa määreiden mukaan tehnyt ja aloitetaan konfiguroimaan itse VLAN DHCP-verkotusta. Määritellään ensin käytettävät VLAN-tagit sekä niille määriteltävät IP-osoiteavaruudet. Esimerkissä määritellään kaksi VLAN-verkkoa, mutta niitä voidaan luoda niin monta kuin on tarpeen:

nova-manage network create --label vlan1 --fixed_range_v4 10.0.1.0/24 --num_networks 1 --network_size 256 --vlan 1

nova-manage network create --label vlan2 --fixed_range_v4 10.0.2.0/24 --num_networks 1 --network_size 256 --vlan 2

Verkkojen pitäisi nyt määräytyä taulukon yksi mukaisesti. Taulukossa VLAN-sarake tarkoittaa määräytyvää VLAN-tagia, bridge määräytyvää siltaa, yhdyskäytävä verkolle määräytyvää yhdyskäytävän IP-osoitetta, VPN Cloudpipe-instanssia ja loput osoitteet jaetaan luotujen instanssien kesken.

VLAN:	Bridge:	Subnet:	Yhdyskäytävä:	VPN:	Jaettavat osoitteet:
1	br1	10.0.1.0/24	10.0.1.1/24	10.0.1.2	10.0.1.3 – 10.0.1.254
2	br2	10.0.2.0/24	10.0.2.1/24	10.0.2.2	10.0.2.3 – 10.0.2.254

Liite 2. Kelluvien IP-osoitteiden konfigurointi

Luodaan käytettävät kelluvat IP-osoitteet seuraavin komennoin:

root@kone1:~# nova-manage floating create --pool pool1 --ip_range "IP-avaruus"

root@kone1:~# nova-manage floating create --pool pool2 --ip_range "IP-avaruus"

Alue on täysin vapaasti valittavissa olemassa olevan suunnitelman ja resurssien mukaan.

Seuraavaksi käynnistetään instanssi ja jatkossa suoritettava komennot on tehtävä luodun instanssin sisällä.

Listan saatavilla olevista kelluvista IP-osoitteista saadaan näkyviin komennolla

nova floating-ip-pool-list

Napataan IP-osoite saatavilla olevasta osoiteavaruudesta komennolla:

nova floating-ip-create "Pool-name"

Tämän jälkeen IP-osoite voidaan kiinnittää tiettyyn instanssiin komennolla

nova add-floating-ip "Instanssin ID tähän" "napattu IP-osoite"

Tarkistetaan tilanne komennolla

nova floating-ip-list

Näkyviin pitäisi tulla lista, jossa näkyvät käytössä oleva julkinen IP-osoitte, instanssin id, instanssin privaattiosoite sekä mistä avaruudesta kelluva IP-osoite on saatu.

Liite 3. Cloudpipen konfigurointi

OpenStackiin sisältyy valmiina VPN-tekniikkaa hyödyntävä Cloudpipe-niminen työkalu. Sen konfigurointi on määritelty OpenStackin dokumentaatiossa seuraavalla tavalla:

Aloitetaan asentajalla tarvittava ohjelmat luotuun instanssiin. Kyseessä on puhdas Ubuntu.

 # apt-get update && apt-get upgrade && apt-get install openvpn bridge-utils unzip -y

Seuraavaksi lähdetään konffaamaan OpenVPN:n konfigurointitiedostoa halutunlaiseksi. Luotu konfigurointi-tiedosto tallennetaan kohteeseen /etc/openvpn/server.conf

port 1194
proto udp
dev tap0
up "/etc/openvpn/up.sh br0"
down "/etc/openvpn/down.sh br0"

persist-key
persist-tun

ca ca.crt
cert server.crt
key server.key # This file should be kept secret

dh dh1024.pem
ifconfig-pool-persist ipp.txt

server-bridge VPN_IP DHCP_SUBNET DHCP_LOWER DHCP_UPPER

client-to-client
keepalive 10 120
comp-lzo

max-clients 1 #lukua voi nostaa tarvittavan määrän mukaan

user nobody
group nogroup

persist-key

persist-tun

status openvpn-status.log

verb 3
mute 20

Luodaan skriptit joilla voidaan nostaa bridge-rajapinta ylös (up.sh) ja erillinen skripti, jolla lasketaan se alas (down.sh):

```
#!/bin/sh

BR=$1
DEV=$2
MTU=$3
/sbin/ifconfig $DEV mtu $MTU promisc up
/usr/sbin/brctl addif $BR $DEV
```

```
#!/bin/sh

BR=$1
DEV=$2

/usr/sbin/brctl delif $BR $DEV
/sbin/ifconfig $DEV down
```

Tehdään näistä skripteistä ajettavissa olevia sovelluksia komennolla:

 # chmod +x /etc/openvpn/{up.sh,down.sh}

Seuraavaksi editoidaan instanssin rajapinta-tiedostoa /etc/network/interfaces. Pää-rajapinta ajetaan alas ja käyttöön otetaan sillattu rajapinta:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet manual
  up ifconfig $IFACE 0.0.0.0 up
  down ifconfig $IFACE down
```

auto br0
iface br0 inet dhcp
bridge_ports eth0

Seuraavaksi muokataan rc.local-tiedostoa seuraavanlaiseksi.

#!/bin/sh -e

#

 # rc.local

#

 # This script is executed at the end of each multiuser runlevel.

 # Make sure that the script will "exit 0" on success or any other

 # value on error.

#

 # In order to enable or disable this script just change the execution

 # bits.

#

 # By default this script does nothing.

####### These lines go at the end of /etc/rc.local #######

. /lib/lsb/init-functions

echo Downloading payload from userdata

wget http://169.254.169.254/latest/user-data -O /tmp/payload.b64

echo Decrypting base64 payload

openssl enc -d -base64 -in /tmp/payload.b64 -out /tmp/payload.zip

```
mkdir -p /tmp/payload

echo Unzipping payload file

unzip -o /tmp/payload.zip -d /tmp/payload/


# if the autorun.sh script exists, run it

if [ -e /tmp/payload/autorun.sh ]; then

   echo Running autorun.sh

   cd /tmp/payload

   chmod 700 /etc/openvpn/server.key

   sh /tmp/payload/autorun.sh

   if [ ! -e /etc/openvpn/dh1024.pem ]; then

      openssl dhparam -out /etc/openvpn/dh1024.pem 1024

   fi

else

  echo rc.local : No autorun script to run

fi

exit 0
```

Autorun.sh skriptin avulla saadaan jäsennettyä instanssien verkko-asetukset reittien luomista varten. Instanssin pitäisi olla nyt valmis käytettäväksi Cloudpipea varten. Autorun-skriptin sisältö on seuraavanlainen:

```
#!/bin/bash

# vim: tabstop=4 shiftwidth=4 softtabstop=4
```

```
# Copyright 2010 United States Government as represented by the
# Administrator of the National Aeronautics and Space Administration.
# All Rights Reserved.
#
#    Licensed under the Apache License, Version 2.0 (the "License"); you may
#    not use this file except in compliance with the License. You may obtain
#    a copy of the License at
#
#         http://www.apache.org/licenses/LICENSE-2.0
#
#    Unless required by applicable law or agreed to in writing, software
#    distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
#    WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
#    License for the specific language governing permissions and limitations
#    under the License.


# This gets zipped and run on the cloudpipe-managed OpenVPN server


export LC_ALL=C
export VPN_IP=`ifconfig  | grep 'inet addr:'| grep -v '127.0.0.1' | cut -d: -f2 | awk '{print $1}'`
export BROADCAST=`ifconfig  | grep 'inet addr:'| grep -v '127.0.0.1' | cut -d: -f3 | awk '{print $1}'`
```

```
export DHCP_MASK=`ifconfig  | grep 'inet addr:'| grep -v '127.0.0.1' | cut -d: -f4 |
awk '{print $1}'`

export GATEWAY=`netstat -r | grep default | cut -d' ' -f10`


DHCP_LOWER=`echo $BROADCAST | awk -F. '{print $1"."$2"."$3"." $4 - 5 }'`

DHCP_UPPER=`echo $BROADCAST | awk -F. '{print $1"."$2"."$3"." $4 - 1 }'`


# generate a server DH

openssl dhparam -out /etc/openvpn/dh1024.pem 1024


cp crl.pem /etc/openvpn/

cp server.key /etc/openvpn/

cp ca.crt /etc/openvpn/

cp server.crt /etc/openvpn/

# Customize the server.conf.template

cd /etc/openvpn


sed -e s/VPN_IP/$VPN_IP/g server.conf.template > server.conf

sed -i -e s/DHCP_SUBNET/$DHCP_MASK/g server.conf

sed -i -e s/DHCP_LOWER/$DHCP_LOWER/g server.conf

sed -i -e s/DHCP_UPPER/$DHCP_UPPER/g server.conf

sed -i -e s/max-clients\ 1/max-clients\ 10/g server.conf
```

echo "push \"route 10.0.0.0 255.255.255.0 $GATEWAY\"" >> server.conf

echo "duplicate-cn" >> server.conf

echo "crl-verify /etc/openvpn/crl.pem" >> server.conf

/etc/init.d/openvpn start

Seuraavaksi Cloudpipe-instanssi täytyy ilmoittaa Glanceen, josta se voidaan myöhemmin noutaa projekteja varten.

Aloitetaan hakemalla käynnissä olevat instanssit ID komennolla:

$ nova list

Vastaan pitäisi tulla instanssi, jonka nimi on cloud-pipe. Luodaan tuolla instanssin ID:llä nyt levykuva:

$ nova image-create 739079a-b-0f8e-404a-ae6e-a91f4fe99c94

Tarkistetaan, että levykuva ilmestyi Glanceen komennolla:

$ nova image-list

Tehdään levykuvasta julkinen komennolla:

$ glance image-update 0bfc8fd3-1590-463b-b178-bce30be5ef7b is_public=true

Muokataan seuraavaksi /etc/nova.conf-tiedostoa. Sinne täytyisi lisätä seuraavat kohdat, että Nova osaa ottaa käyttöön luodun Cloudpipe-levykuvan:

cloud-pipe vpn client

--vpn_image_id=0bfc8fd3-1590-463b-b178-bce30be5ef7b

--use_project_ca-true

--cnt_vpn_clients=5

Käynnistetään kaikki palvelut uudelleen komennolla:

 # cd /etc/int.d && for i in $(ls nova-*); do service $i restart; done

Cloudpipe voidaan nyt ohjata käyttöön halutulle projektille/tenantille ID:n mukaan:

$ nova cloud-pipe create $tenant_id

Projektien/tenantien IP-osoitteet saadaan näkyviin komennolla:

$ keystone tenant-list

Toiminta voidaan tarkistaa komennolla:

$ nova cloudpipe-list

Listassa täytyisi näkyä projektin/tenantin ID, julkinen IP-osoite, julkinen portti ja privaatti IP-osoite.

Nova on luonut automaattisesti tarvittavat säännöt Cloudpipe-instanssia varten, joilla avataan VPN:ssä käytettävä portti sekä ICPM:tä eli pingausta varten:

ALLOW 1194:1194 from 0.0.0.0/0

ALLOW -1:-1 from 0.0.0.0/0