

Iiro Keskitalo

PILVIPALVELUT

Amazon Web Services

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Toukokuu 2014**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Kokkola-Pietarsaari	Aika Toukokuu 2014	Tekijä Iiro Keskitalo
Koulutusohjelma Tietotekniikan koulutusohjelma		
Työn nimi PILVIPALVELUT. Amazon Web Services		
Työn ohjaaja Sakari Männistö	Tekstin ohjaaja Helvi Pääkkönen	Sivumäärä 30 + 3
<p>Tämän opinnäytetyön tarkoituksena oli tutustua pilvipalveluihin yleisesti sekä käytännönläheisesti yhteen pilvipalvelutarjoajaan, joka tässä tapauksessa oli Amazon Web Services. Tarkoituksena oli luoda Amazoniin käyttäjätunnus sekä testata Elastic Compute Cloud (EC2) -tuotetta, joka mahdollistaa virtuaalipalvelimen luomisen. Virtuaalipalvelimelle oli tarkoitus asentaa LAMP-web-palvelin ja testata sen käyttöä.</p> <p>Oman oppimisen ja kokeilun perusteella oli tarkoituksena suunnitella ja ohjeistaa harjoitustehtäviä Centria-ammattikorkeakoulun opiskelijoille, jotta he pääsisivät tutustumaan käytännössä pilvipalveluihin. Suunnittelin ja ohjeistin kolme harjoitustehtävää, joihin sisältyvät tilin ja instanssin luonti sekä LAMP-web-palvelimen että WordPress-blogin asentaminen.</p>		

Asiasanat

Amazon Web Services, Elastic Compute Cloud, instanssi, LAMP web-palvelin, pilvilaskenta, pilvipalvelu

ABSTRACT

Department Kokkola-Pietarsaari	Date May 2014	Author Iiro Keskitalo
Degree programme Information Technology		
Name of thesis CLOUD SERVICES. Amazon Web Services		
Instructor Sakari Männistö	Language instructor Helvi Pääkkönen	Pages 30 + 3

The objective of this thesis was to review cloud services in general and explore in practice one of the cloud service providers, which in this case was Amazon Web Services. The purpose was to create a user account in Amazon and test Elastic Compute Cloud (EC2) product which enables to create a virtual server. The aim was to install a LAMP web server to the virtual server and test it in use.

Based on my own learning and testing, the aim was to design and instruct exercises to students of Centria University of Applied Sciences so that they would learn about of cloud services in practice. I designed and instructed three exercises that include creating an account, an instance and the installations of a LAMP web server and a WordPress blog.

Key words

Amazon Web Services, cloud computing, cloud service, Elastic Compute Cloud, instance, LAMP web server

KÄSITTEET JA LYHENTEET

AMI	Amazon Machine Image, malli, jonka pohjalta luodaan instanssi
Apache	avoimen lähdekoodin web-palvelin
AWS	Amazon Web Services, Amazon-yhtiön pilvipalvelut
DNS	Domain Name System, Internetin nimipalvelujärjestelmä, joka muuntaa verkkotunnuksia IP-osoitteiksi
EC2	Elastic Compute Cloud, Amazonin pilvipalvelu, johon voidaan luoda virtuaalipalvelin
IaaS	Infrastructure-as-a-Service, infrastruktuuripohjainen pilvipalvelumalli
IP	Internet Protocol, Internetin TCP/IP-mallin protokolla
LAMP	kokoelma avoimen lähdekoodin ohjelmia (Apache-palvelin, MySQL-tietokanta, PHP-tuki), jotka muodostavat www-palvelimen
MySQL	relaatiotietokantaohjelmisto
PaaS	Platform-as-a-Service, alustapohjainen pilvipalvelumalli
PHP	ohjelmointikieli, jota käytetään erityisesti web-palvelinympäristöissä
PuTTY	SSH-palvelimien käyttöön tarkoitettu asiakasohjelma/pääte-emulaattori
PuTTYgen	RSA- ja DSA-avainten luomiseen tarkoitettu apuohjelma
SaaS	Software-as-a-Service, sovelluspohjainen pilvipalvelumalli
SSH	Secure Shell, salattuun tietoliikenteeseen tarkoitettu protokolla
S3	Simple Storage Service, Amazonin pilvitallennustilaa varten tehty tuote
WinSCP	Windows Secure Copy, SFTP-, FTP- ja SCP-tiedostonsiirtomenetelmiä käyttävä ohjelma
WordPress	blogin ylläpitoon tarkoitettu avoimen lähdekoodin sisällönhallintaohjelma

**TIIVISTELMÄ
ABSTRACT
KÄSITTEET JA LYHENTEET
SISÄLLYS**

1 JOHDANTO	1
2 PILVIPALVELUT	3
3 PALVELUMALLIT	6
3.1 Infrastructure-as-a-Service	7
3.2 Platform-as-a-Service	8
3.3 Software-as-a-Service	8
4 PILVITYYPIT	10
4.1 Julkinen pilvi	10
4.2 Yksityinen pilvi	11
4.3 Yhteisöpilvi	12
4.4 Hybridipilvi	12
5 AMAZON WEB SERVICES	14
5.1 Tilin luonti	15
5.2 Elastic Compute Cloud	15
5.3 Simple Storage Service	15
5.4 Instanssin luominen ja yhdistäminen	16
5.5 LAMP-web-palvelimen asentaminen	18
5.6 WordPress-blogin isännöinti	20
6 HARJOITUSTEHTÄVIEN SUUNNITTELU JA OHJEISTUS	23
6.1 Harjoitustehtävä 1 – Tilin ja instanssin luominen	23
6.2 Harjoitustehtävä 2 – LAMP-web-palvelimen asennus ja testaus	23
6.3 Harjoitustehtävä 3 – WordPress-blogin asennus ja testaus	24
7 YHTEENVETO	26
LÄHTEET	29
LIITTEET	
LIITE 1. Harjoitustehtävä 1 – Tilin ja instanssin luominen	
LIITE 2. Harjoitustehtävä 2 – LAMP-web-palvelimen asennus ja testaus	
LIITE 3. Harjoitustehtävä 3 – WordPress-blogin asennus ja testaus	

1 JOHDANTO

Viime vuosien aikana informaatioteknologiassa on yhä enemmän noussut esiin käsite pilvipalvelu. Pilvipalvelulla tarkoitetaan palvelua, jota käyttäjä käyttää Internetin välityksellä riippumatta siitä ja usein myös täysin tietämättömänä siitä, missä palvelu fyysisesti sijaitsee. Periaatteessa pilvipalveluita on ollut olemassa jo hyvin pitkään, ja itsekkin olen käyttänyt pilvipalveluiksi luokiteltavia palveluja ajattelematta, että ne ovat pilvipalveluja. Näitä ovat esimerkiksi Facebook, Hotmail ja Dropbox.

Käsite pilvipalvelu kuitenkin tarkoittaa enemmän sitä, että on olemassa jokin pilvipalveluntarjoaja, joka antaa asiakkailleen virtuaalista laskentakapasiteettia, jonka avulla nämä voivat itse konfiguroida palvelimen, tietokannat ja niin edelleen. Asiakkaille voidaan tarjota myös valmiiksi määriteltyä laskentakapasiteettia, jossa asiakkaat voivat ajaa omia tekemiään sovelluksia. Loppukäyttäjille voidaan tarjota Internetin välityksellä ajettavia ohjelmia, ilman että ne ensin asennettaisiin omille tietokoneille.

Syy valita opinnäytetyön aiheeksi pilvipalvelut oli niin sanotusti oma ”tietämättömyys” aiheesta syvemmin sekä aiheeseen noussut mielenkiinto, koska ne ovat niin paljon yleistyneet. Toivottavasti tässä työssä olevien harjoitustehtävien myötä saadaan myös opiskelijoille kokemusta pilvipalveluista.

Tämän opinnäytetyön tarkoituksena oli tutustua yleisesti pilvipalveluihin ja niiden erilaisiin toimintamalleihin ja käyttötarkoituksiin sekä kirjoittaa tietoperusta oleellisimmista asioista pilvilaskentaan liittyen. Tämän jälkeen oli tarkoituksena tutustua yhteen pilvipalvelutarjoajaan. Tässä tapauksessa valitsin Amazon Web Servicesin, joka tarjoaa infrastruktuuripohjaisen pilvipalvelumallin. Eli Amazoniin voidaan luoda instanssi, johon asennetaan web-palvelin ja jonne voidaan sitten upottaa jokin sovellus tai nettisivut.

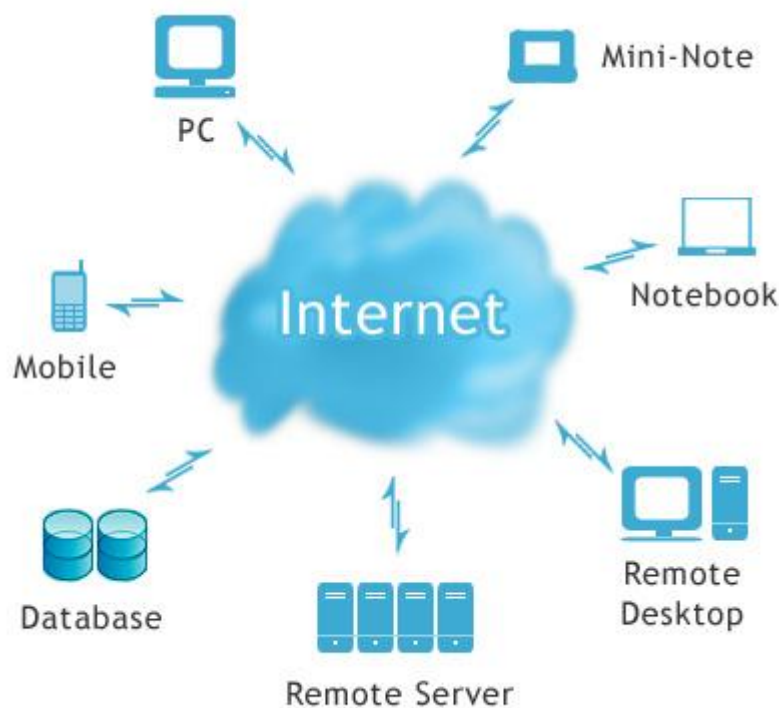
Oman oppimisen ja testaamisen perusteella oli tämän jälkeen tavoitteena suunnitella ja ohjeistaa muutama harjoitustehtävä Centria-ammattikorkeakoulun tietotek-

niikan opiskelijoille. Harjoitustehtävien tarkoituksena on tutustuttaa opiskelijat Amazonin pilvipalvelutarjontaan, niin että he saavat luoda instanssin ja asentaa web-palvelimen. Tarkoitus on myös saada opiskelijat näkemään, millaista on heittäytyä pilvipalvelutarjoajan tarjoamiin mahdollisuuksiin ja saada pohtimaan valintaa oman fyysisen laskentakapasiteetin tai pilvipalvelun tarjoaman laskentakapasiteetin väliltä, mikäli esimerkiksi aikoo omaa yritystä lähteä perustamaan joskus.

2 PILVIPALVELUT

Pilvilaskennalla (cloud computing) tarkoitetaan tiedon tallentamista tai ohjelmien ajamista Internetissä sen sijaan, että tiedot tai ohjelmat olisivat tallennettuna omalla tietokoneella. Internetiä on yleisesti kuvattu pilvellä, joten tästä syystä sana ”pilvi” on periytynyt myös pilvipalveluihin. (Griffith 2013.)

Pilvilaskenta (KUVIO 1.) kokonaisuudessaan tarkoittaa jotain ratkaisua, joka tarjoaa informaatioteknologiaa palveluna. Pilvilaskenta on Internet-pohjaisten järjestelmien laskentaratkaisu, jossa jaetaan resurssit tarjolle aivan kuten sähkö jaetaan sähköverkossa. (Walker 2010.)



KUVIO 1. Pilvilaskenta (Carmen 2012.)

Pilvilaskennalla voidaan esimerkiksi tarkoittaa joukkoa laitteistoja, verkkoja ja tallennustilaa, joita sitten tarjotaan pilvipalveluna. Pilvilaskennalla on tiettyjä ominaispiirteitä: jaettu infrastruktuuri, jossa fyysiset laitteet ja verkot voidaan Internetin välityksellä jakaa, elastisuus eli kyky säännöstellä laskentakapasiteettia perustuen

sen hetkiseen kysyntään, pääsy Internetiin kaikilla tavallisilla laitteilla, kuten tietokoneilla ja mobiililaitteilla, jotka käyttävät ohjelmointirajapintoja (Application programming interface, API) sekä käytön mittaaminen, jossa hallitaan ja optimoidaan raportointia ja laskutusinformaatiota. Laskutus perustuu siihen, kuinka paljon palvelua on käytetty laskutusjakson aikana. (Dialogic 2010.)

Pilvipalveluiden joustavuus perustuu resurssien jakamiseen kysynnän mukaan. Joustavuuden ansioista järjestelmien kumulatiivisten resurssien käyttäminen helpottuu, koska ei tarvitse asettaa tiettyä laitteistoa hoitamaan jotain tiettyä tehtävää. Ennen pilvilaskentaa nettisivujen ja palvelinpohjaisten sovellusten ylläpitoon käytettiin tietyn tietokoneen laskentakapasiteettia. Pilvilaskenta mahdollistaa sen, että resurssit voidaan käyttää yhdeltä virtuaalitetokoneelta. Tämä sulautettu konfiguraatio mahdollistaa sellaisen ympäristön, jossa sovellukset voivat käynnistyä itsenäisesti ilman mitään varta vasten niille tarkoitettua konfiguraatiota. (Walker 2010.)

Yrityksissä kohdataan päivittäin ongelmallisia rajoituksia (KUVIO 2.), jotka aiheuttavat vaivaa. Näihin kuuluvat

- palvelimien alhainen hyödyntäminen
- teho-, tila- ja kustannusrajoitukset
- uusien palveluiden käyttöönottoajat
- korkeat yleiskustannukset palveluiden ja käyttäjien säännöstelyssä
- informaatioteknologian epäselvä arvo. (Bair & Rhodon 2010.)

Problems	Solutions
Low utilization of servers	→ Virtualization, multi-tenancy
Power, space, cost constraints	→ Utility billing, resource elasticity
Delays in new services	→ On demand availability, self-service
Provisioning overhead	→ Self-service
Unclear value of IT	→ Utility billing
Internal infrastructure	→ Internet delivery

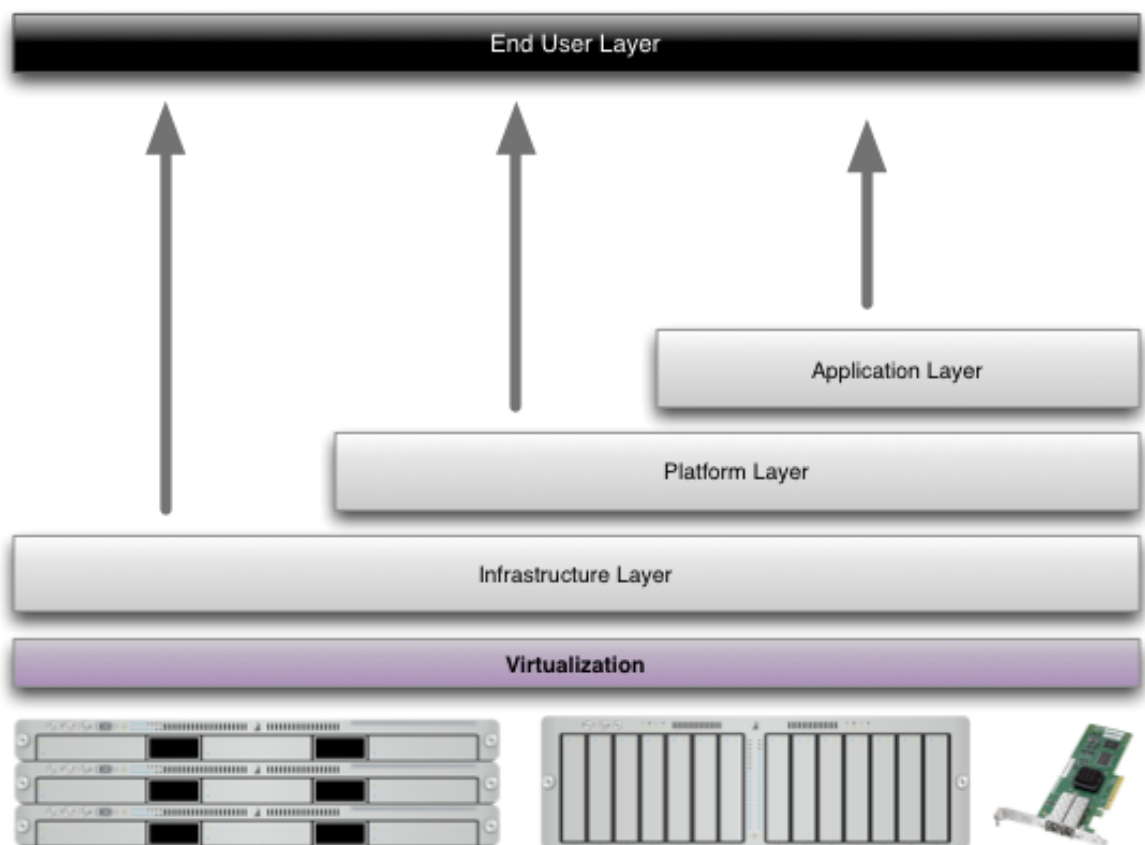
KUVIO 2. Yritysten ongelmat ja pilvilaskennan mahdollistamat ratkaisut (Bair & Rhodon 2010.)

NIST (National Institute of Standards and Technology) määrittää pilvilaskentaan viisi olennaista ominaisuutta, jotka ratkaisevat edellä mainittuja ongelmia. Ensimmäinen on On-demand Self-service eli palvelun tarpeiden tai vaatimusten itsepalvelu, jossa kuluttaja voi yksipuolisesti säännöstellä laskentakapasiteetin, esimerkiksi verkkovaraston koon, ilman että ihmisen täytyy olla vuorovaikutuksessa palveluntarjoajan kanssa. Toinen ominaisuus on Broad network access, jossa ominaisuudet ovat käytössä verkon yli ja joihin pääsee käsiksi standardimekanismeilla, kuten esimerkiksi työasemilla, kannettavilla tietokoneilla, matkapuhelimilla ja tableteilla. Kolmas ominaisuus on Resource pooling, jossa palveluntarjoajan resursseja yhdistetään palvelemaan useampaa kuluttajaa. Siinä käytetään monivuokralais-mallia, jossa sekä fyysisiä että virtuaalisia resursseja dynaamisesti määrätään ja uudelleen määrätään kuluttajakysynnän mukaan. Neljäs ominaisuus on Rapid elasticity eli nopea elastisuus. Siinä voidaan säännöstellä ja vapauttaa laskentakapasiteettia, jopa automaattisesti, skaalautumaan nopeasti ylöspäin tai alaspäin suhteessa kysyntään. Viimeinen ominaisuus on Measured service eli mitattu palvelu, jossa pilvijärjestelmät voivat automaattisesti hallita ja optimoida resurssien käyttöä käyttämällä mittausominaisuutta liittyen palvelun tyyppiin, esimerkiksi tallennustila, kaistanleveys tai aktiiviset käyttäjätunnukset. Mitattua palvelua käyttämällä saadaan avoimuutta palveluntarjoajan ja kuluttajan välille, kun resurssien käyttöä voidaan monitoroida, hallita ja raportoida. (Grance & Mell 2011.)

Päällimmäinen huolenaihe yrityksillä siirtyä käyttämään pilvipalveluja ovat turvallisuus ja yksityisyys. Yhtiöt, jotka pilvipalveluja tarjoavat, tietenkään ymmärtävät tämän huolenaiheen, koska ilman luotettavaa turvallisuuspolitiikkaa niiden liiketoiminta saattaisi aivan hyvin romahtaa. (Walker 2010.)

3 PALVELUMALLIT

Pilvipalveluiden toimitus koostuu kolmesta eri palvelumallista (KUVIO 3.), jotka ovat Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) ja Software-as-a-Service (SaaS). Jos pilvipalvelun käyttäjä käyttää esimerkiksi jotain palvelua infrastruktuurikerroksessa (infrastructure layer), hän voi ajaa omia sovelluksiaan pilven infrastruktuurin resursseilla, mutta on edelleen itse vastuussa sovellusten tuki-, ylläpito- ja turvallisuusasioista. Jos pilvipalvelun käyttäjä käyttää palveluita sovelluskerroksessa (application layer), tuen, ylläpidon ja turvallisuuden yleensä hoitaa pilvipalvelun tarjoaja. (C⁴S 2014a.)

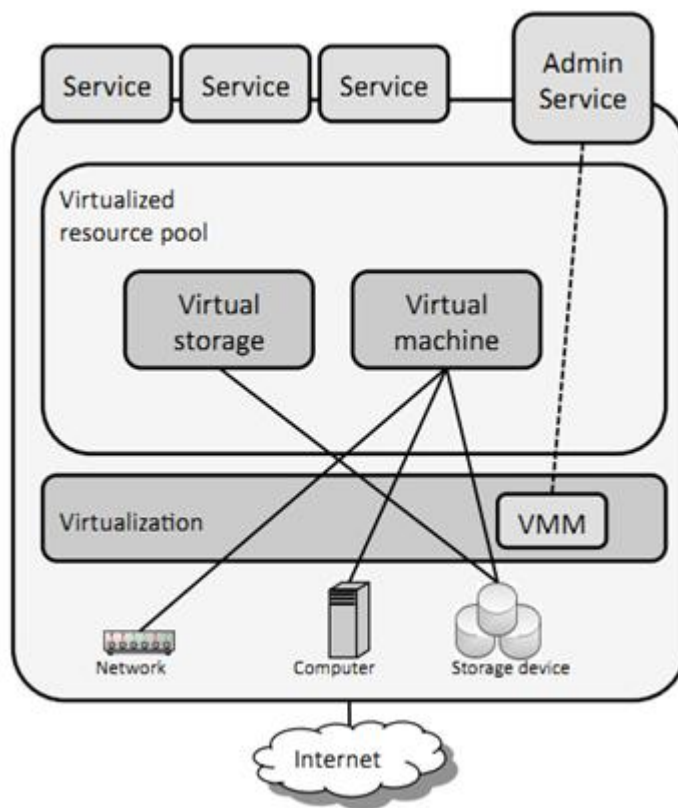


KUVIO 3. Palvelumallit ja loppukäyttäjän kerros (C⁴S 2014a.)

Kaikki pilvipalveluihin liittyvät palvelut sopivat johonkin edellä mainittuihin palvelumalleihin. Loppukäyttäjät käyttävät yleensä SaaSia, ohjelmistokehittäjätiimit PaaSia ja IT-osastot, joiden vastuulla on infrastruktuurin käyttö, IaaSia. (Czarnecki 2011.)

3.1 Infrastructure-as-a-Service

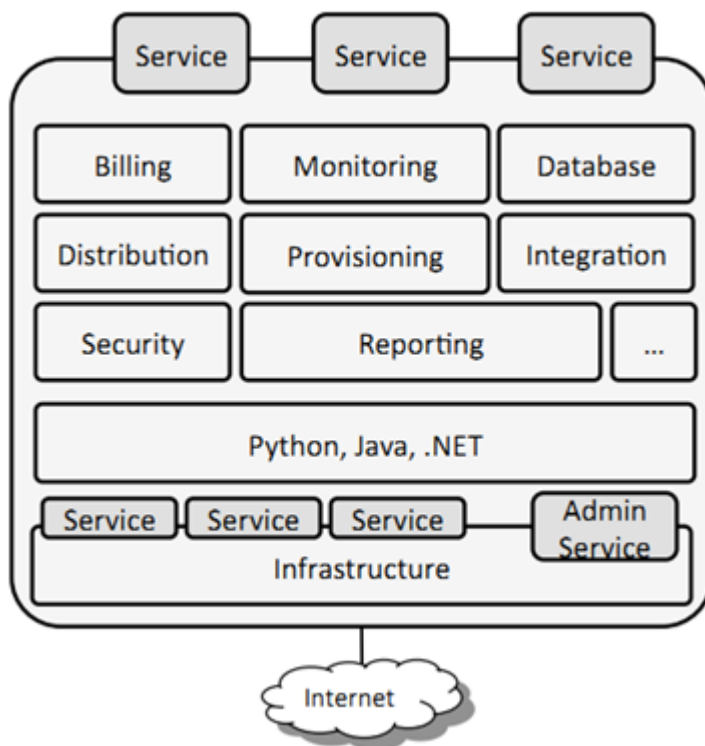
Palvelumalli Infrastructure-as-a-Service (IaaS) (KUVIO 4.) kattaa laajan valikoiman eri ominaisuuksia, joihin kuuluu muun muassa yksittäiset palvelimet, yksittäiset verkot, kiintolevyt, erilaiset tallennuslaitteet sekä sähköpostipalvelimet, nimipalvelimet ja viestintäjärjestelmät. Kaikki edellä mainitut voidaan sisällyttää vastaamaan kysyntää, ja niissä on yleensä mukana ohjelmistojen lisenssimaksut käyttöjärjestelmille, ja myös niihin liittyvät ohjelmistot on yleensä asennettuna palvelimille. Organisaatiot voivat rakentaa omaa kysyntäänsä varten täydellisen tietokoneinfrastruktuurin IaaS:n avulla. (Czarnecki 2011.)



KUVIO 4. Infrastructure-as-a-Service (IaaS) pino (C⁴S 2014a.)

3.2 Platform-as-a-Service

Palvelumallia Platform-as-a-Servicea (PaaS) (KUVIO 5.) käyttävät usein ohjelmistokehittäjäyhtiöt ajaakseen valmistamiaan ohjelmistotuotteita. Ohjelmistotuotteet tarvitsevat fyysisen palvelimen toimiakseen tietokantaohjelmiston kanssa, usein myös web-palvelimen. Nämä kaikki ovat alustoja, joita ohjelmisto vaatii toimiakseen. Tällaisen rakentaminen itse on aikaa vievää ja vaatii jatkuvaa seuranta ja päivittämistä. PaaS tarjoaa kaikki alustat, jotka mahdollistavat ohjelmistosovelluksen ajamisen, ilman että tarvitsee huolehtia alemman kerroksen komponenttien ylläpidosta itse. (Czarnecki 2011.)

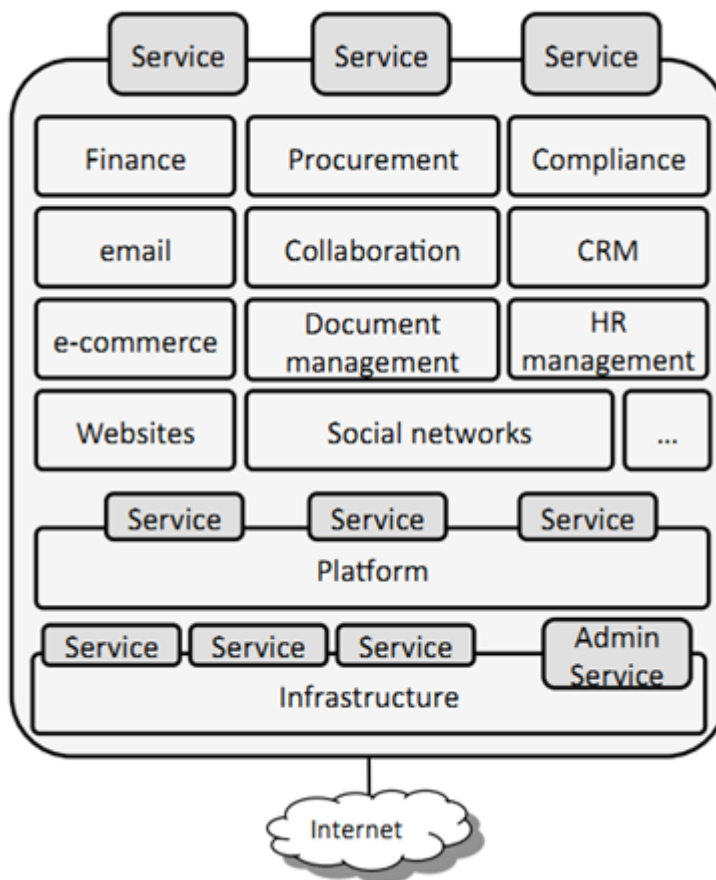


KUVIO 5. Platform-as-a-Service (PaaS) pino (C⁴S 2014a.)

3.3 Software-as-a-Service

Palvelumalli Software-as-a-Service (SaaS) (KUVIO 6.) on tyypillisesti loppukäyttäjäsovelluksia varten tarkoitettu kysynnän mukaan verkon yli ”maksa kun käytät” -periaatteella toimiva malli. Tämä malli ei vaadi asiakkaalta minkäänlaisia asenta-

misiä, ainoastaan nettiselaimen ja verkkoyhteyden. Yksi esimerkki SaaS:sta on Microsoft Office365. Jos käyttäjä haluaa esimerkiksi käyttää Wordia, hänen tulee ostaa se, asentaa se sekä ottaa itse varmuuskopiot asiakirjoistaan ja niin edelleen. Office365:n avulla Word ei vaadi asentamista, varmuuskopiot otetaan automaattisesti, ohjelma päivittyy automaattisesti ja siihen pääsee käsiksi missä vain. Kaikki tämä tulee kuukausimaksua vastaan, ja jos käyttäjä ei enää tarvitse Wordia, hän voi lopettaa kuukausimaksut ja asia on sillä selvä. (Czarnecki 2011.)



KUVIO 6. Software-as-a-Service (SaaS) pino (C⁴S 2014a.)

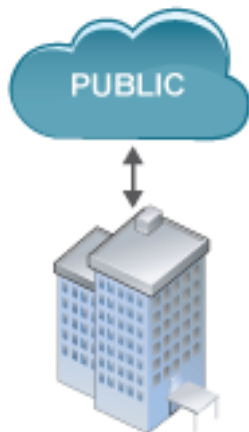
4 PILVITYYPIT

Pilvipalveluja voidaan luokitella eri tavoin erinäisten tekijöiden mukaan, kuten muun muassa sen, missä pilvipalvelua isännöidään, turvallisuusvaatimusten, halun jakaa pilvipalveluita, kyvyn hallita joitain tai kaikkia palveluita sekä personointimahdollisuuksien mukaan. On olemassa neljä eri pilvityyppiä, joiden avulla voidaan määrittää, kenellä on oikeus päästä pilvipalveluun käsiksi: julkinen pilvi (public cloud), yksityinen pilvi (private cloud), yhteisöpilvi (community cloud) sekä hybridipilvi (hybrid cloud). (Cloud.cio.gov 2014.)

4.1 Julkinen pilvi

Julkinen pilvi (public cloud) (KUVIO 7.) on julkisesti saatavilla oleva pilviympäristö, jonka omistaa kolmannen osapuolen pilvipalveluntarjoaja (Arciture Education 2014). Julkiselle pilvelle on ominaista, että pilvipalvelun tarjonta on julkisesti saatavilla, sekä se, että verkko, johon pilvipalvelu on yhteydessä, on myös julkinen. Pilvipalvelu ja pilviresurssit hankitaan erittäin suurista resurssipoolista, jotka on jaettu kaikille loppukäyttäjille. (C⁴S 2014b.)

Joitain parhaiten tunnettuja esimerkkejä julkisista pilvistä on Amazon Web Services (AWS), johon sisältyy Elastic Compute Cloud (EC2), ja Simple Storage Service (S3), joka muodostuu IaaS-pilvitarjonnasta. Google App Engine tarjoaa PaaS-asiakkailleen. Salesforce.com on taas tunnetuin SaaS-tarjonnasta asiakkuuden hallintapalveluillaan. (C⁴S 2014b.)



KUVIO 7. Julkinen pilvi (Cloud.cio.gov 2014.)

4.2 Yksityinen pilvi

Yksityisen pilven (private cloud) (KUVIO 8.) omistaa yksi organisaatio. Pilvipalvelun tarjoaja varaa tietyn palvelun vain yhdelle organisaatiolle eikä muille asiakkaille. Organisaatio määrittää, rakentaa ja kontrolloi resurssipooloja, jotka pilvipalvelun tarjoaja toimittaa standardisoituna palvelusetteinä. Yleinen syy organisaatioille hankkia yksityisiä pilviä on niiden kyky valvoa tietoturvasääntöjä ja kontrollointia. (Cloud.cio.gov 2014.)



KUVIO 8. Yksityinen pilvi (Cloud.cio.gov 2014.)

4.3 Yhteisöpilvi

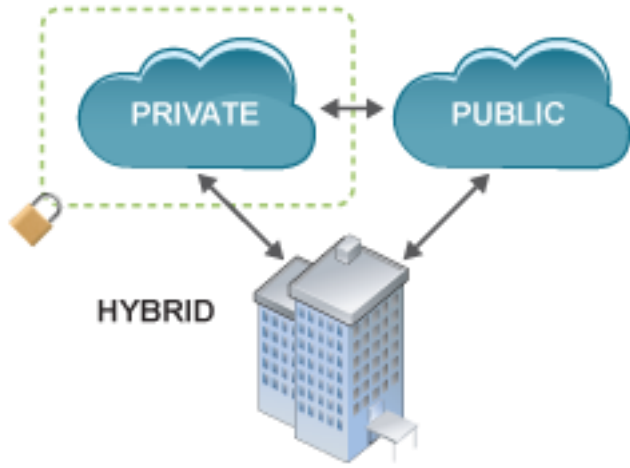
Yhteisöpilvi (community cloud) (KUVIO 9.) on liitetty useisiin organisaatioihin tai ohjelmiin, jotka jakavat tiettyjä tarpeita, kuten turvallisuutta tai samankaltaisia toiminnallisia näkökulmia. Kun organisaatioilla on yhteiset vaatimukset ja asiakkaat, yhteisöpilvi mahdollistaa varojen yhdistämisen sekä laskentaresurssien, tiedon ja kykyjen jakamisen. Kun eliminoidaan vastaavanlaisten järjestelmien päällekkäisyydet, organisaatiot voivat säästää rahaa ja pystyvät tehokkaammin kohdentamaan niukat resurssit. (Cloud.cio.gov 2014.)



KUVIO 9. Yhteisöpilvi (Cloud.cio.gov 2014.)

4.4 Hybridipilvi

Hybridipilveen (hybrid cloud) (KUVIO 10.) kuuluu kahden tai useamman pilven (julkinen, yksityinen tai yhteisö) sekoitus sekä sisäisesti että ulkoisesti isännöitäviä palveluita. Organisaatiot eivät todennäköisesti rajaisi itseään yhteen pilvityyppiin vaan sisällyttäisivät erilaisia ja päällekkäisiä pilvipalveluita saavuttaakseen omat vaatimuksensa. Hybridipilvityypit ovat monimutkaisia ja vaativat huolellista suunnittelua toteuttaa ja hallita, varsinkin kun kahden eri pilvityypin välinen kommunikointi on tarpeen. (Cloud.cio.gov 2014.)

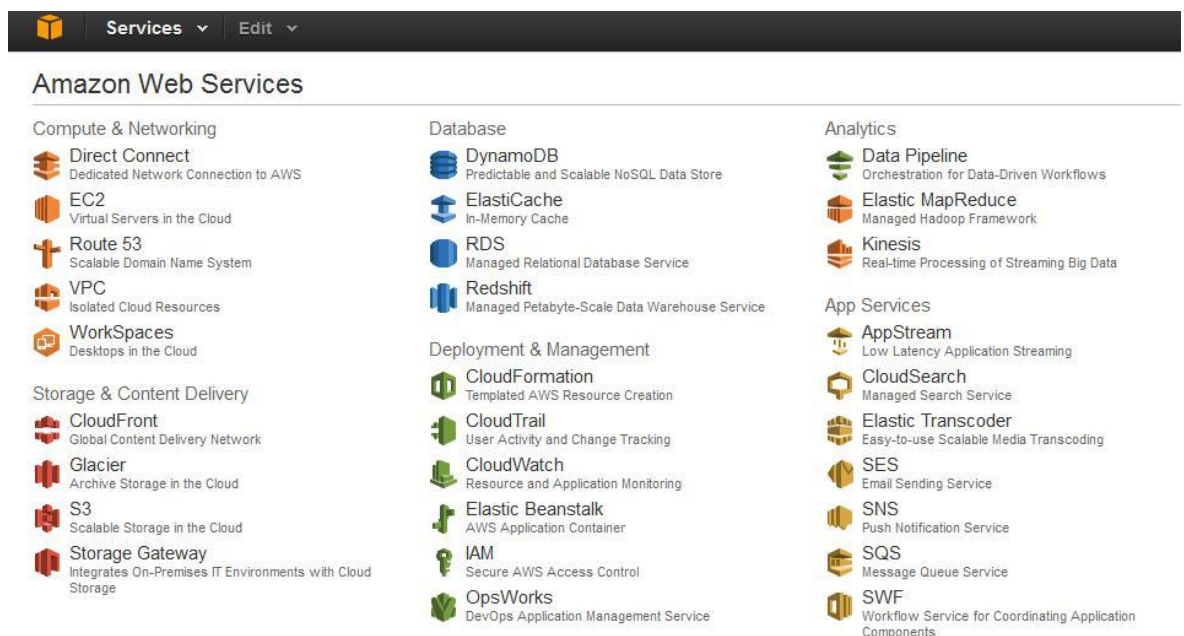


KUVIO 10. Hybridipilvi (Cloud.cio.gov 2014.)

5 AMAZON WEB SERVICES

Tässä luvussa tutustutaan Amazonin tarjoamiin pilvipalveluihin. Tarkoituksena on luoda sinne tili, ja päällimmäinen tavoite on konfiguroida oma laaS-tyyppinen virtuaalipalvelin. Virtuaalipalvelimen luomisen mahdollistaa Amazonin Elastic Compute Cloud (EC2) -tuote. Lisäksi tutustutaan Amazonin Simple Storage Service (S3)-tuotteeseen, joka mahdollistaa tiedostojen tallentamisen pilveen.

Amazonin pilvipalvelut jakautuvat kuuteen kategoriaan. (KUVIO 11.) Compute & Networking -kategoriaan kuuluu esimerkiksi edellä mainittu EC2 sekä Virtual Private Cloud (VPC), jonka avulla voidaan käynnistää Amazonin resursseja yksityisessä ja eristetyssä pilvessä. Tiedon varastoimista ja varmuuskopiointia varten olevat tuotteet löytyvät Storage & Content Delivery -kategoriasta. Deployment & Management -kategoriasta löytyy resurssien hallinnoimiseen ja monitorointiin soivia palveluita. App Services -kategoriasta löytyy muun muassa hakukone- ja sähköpostipalveluita. Database -kategoriasta löytyy tietokantojen hallintaan liittyviä palveluja ja Analytics -kategoriasta tiedonsiirtoon ja tiedonkäsittelyyn liittyviä palveluja.



KUVIO 11. AWS Management Console ja kaikki pilvipalvelut

5.1 Tilin luonti

Kun luodaan tiliä Amazoniin, sinne annettavat tiedot ovat nimi- ja osoitetiedot, puhelinnumero ja sähköposti sekä luottokorttitiedot. Puhelinnumeroa tarvitaan siksi, että Amazoniin rekisteröityessä tulee vaihe, jossa todennetaan henkilöllisyys. Tässä vaiheessa Amazonilta tulee puhelinsoitto, jossa rekisteröityjää vaaditaan näppäilemään puhelimeen tietokoneen näytölle ilmestynvä PIN-koodi. Puhelinsoiton jälkeen valitaan vielä tukisuunnitelma Amazoniin. Perustuki on ilmainen ja tätä opinnäytetyötä varten täysin riittävä. Kun tämä on suoritettu, tilin luonti Amazoniin on valmis.

5.2 Elastic Compute Cloud

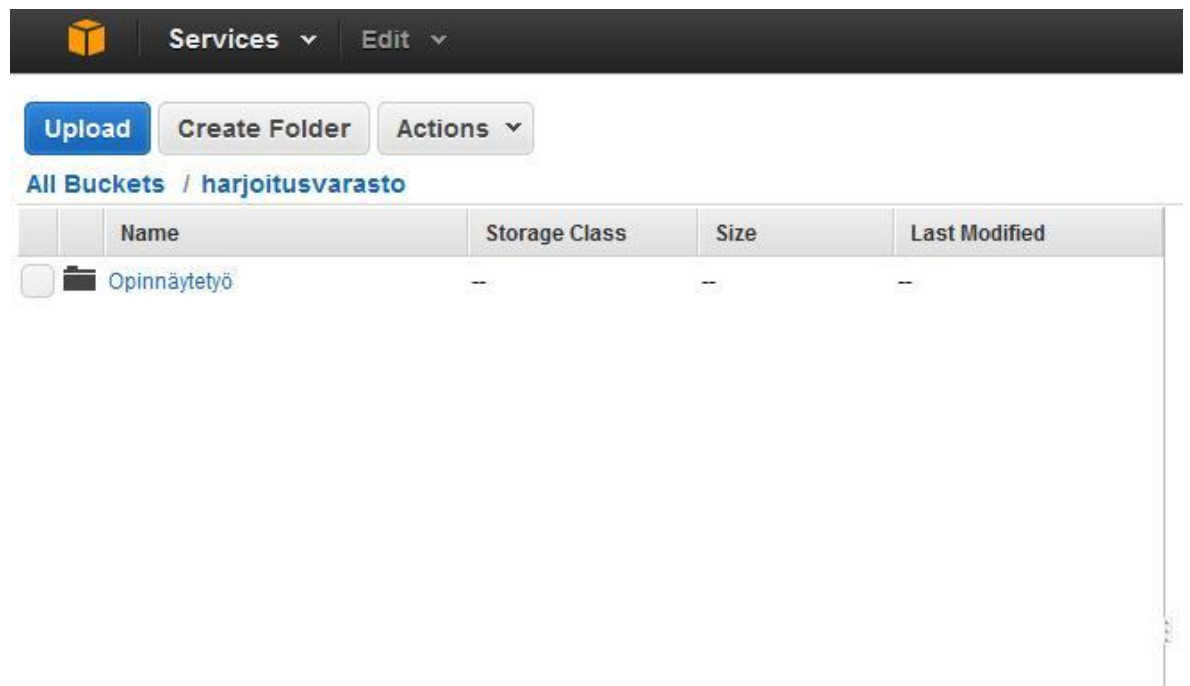
Amazonin Elastic Compute Cloud (EC2) on palvelu, joka mahdollistaa skaalautuvan laskentakapasiteetin pilvessä. Sen avulla saadaan hallinta laskentaresursseja varten. EC2 vähentää vaadittua aikaa, jota tarvitaan uuden palvelininstanssin hankkimiseen ja käynnistämiseen, sekä sen avulla voidaan skaalata tarvittavaa laskentakapasiteettia ylöspäin tai alaspäin sen mukaan, miten laskentavaatimukset muuttuvat. EC2:n käyttämisen maksut perustuvat kapasiteettiin, joka on käytössä. (AWS 2014a.)

5.3 Simple Storage Service

Amazonin Simple Storage Service (S3) on palvelu, joka antaa tallennustilan pilveen. S3:lla voidaan tallentaa ja hakea isojaakin määriä tietoa mistä ja milloin tahansa. Sinne voidaan tallentaa mitä vain, esimerkiksi web-sovelluksia tai media-tiedostoja. Palvelua voidaan käyttää joko nettiselaimella konsolissa tai ohjelmointirajapinnasta. (AWS 2014b.)

Päätin ottaa tämän palvelun yhdeksi testauksen kohteeksi, jotta tutustuminen Amazonin pilvipalveluihin olisi laajempi. Palvelua käyttöön ottaessa ensimmäiseksi

täytyy luoda "bucket" eli säilö, joka on Amazonin määrittelemä termi pilvitallennus-tilalle. Säilölle annetaan nimi ja valitaan, miltä alueelta Amazonin resurssit otetaan. Näin S3 on otettu käyttöön. (KUVIO 12.) Palvelun tiedostojärjestelmä on yksinkertaisen polkutyypinen. Tiedostojen lataaminen ja kansioiden lisääminen säilöön onnistuu helposti parilla painikkeen painalluksella käyttäessä nettiselainta. Halutessa voi asentaa beeta-version Java-pohjaisesta Enchanted Uploader -ohjelmasta, jonka avulla voidaan siirtää suoraan kokonaisia kansioita säilöön. Käytin S3:a tämän opinnäytetyön raportin varmuuskopioimiseen pilveen.



KUVIO 12. Simple Storage Service

5.4 Instanssin luominen ja yhdistäminen

Kun luodaan virtuaalipalvelin Amazoniin, ensimmäiseksi mennään omalla tilillä AWS Management Consoleen ja sieltä valitaan EC2. (KUVIO 13.) Siellä valitaan alue, josta resurssit pilvipalveluun otetaan. Valitsin alueeksi Irlannin. Amazonin Euroopan datakeskus sijaitsee Irlannissa.

Kun instanssia aletaan luoda, ensimmäiseksi valitaan Amazon Machine Image (AMI), joka on alusta, johon sisältyy käyttöjärjestelmä. Seuraavaksi valitaan instanssin koko. Tämän jälkeen voi luoda instanssille turvallisuusryhmän tai käynnistää instanssin suoraan. Oletusarvoinen turvallisuusryhmä luodaan automaattisesti joka tapauksessa (AWS 2014c). Valitsin AMI:ksi Amazon Linuxin. Koska valitsin AMI:n, joka sisältyy ilmaiskokeiluun, on ainoa mahdollinen koko instanssille mikro.

Seuraavaksi tulee luoda avainpari. AWS käyttää avainpareja salaamiseen, kun kirjaudutaan instanssiin. AWS säilyttää julkisen avaimen, ja käyttäjä itse säilyttää yksityisen avaimen, jonka hän tallentaa omalle tietokoneelleen. Kun avainparille on annettu nimi ja se on tallennettu omalle tietokoneelle, instanssin luominen on valmis. (AWS 2014c.)



KUVIO 13. EC2-hallintaruutu ja käytössä olevat resurssit

Koska kyseessä on Linux-pohjainen instanssi, silloin yhteys siihen saadaan PuTTY-ohjelmalla. Edellä tehty avainpari tallentuu .pem-muotoon, jota PuTTY ei tunnista, joten PuTTYgenillä joudutaan muuntamaan avainparin .pem-tiedosto .ppk:ksi. (KUVIO 14.) Tämän jälkeen yhteyden luominen instanssiin onnistuu. PuTTYlle annetaan käyttäjänimi sekä julkinen DNS-osoite ja portiksi laitetaan 22. Lisäksi SSH-tunnistusetuksiin haetaan aiemmin tehty avainparin .ppk-tiedoston sijainti. Mitään muita konfiguraatioasetuksia ei tarvitse tehdä ja näin instanssiin on saatu yhteys PuTTYlla. (AWS 2014d.)



KUVIO 14. Avainparin muuntaminen .ppk-muotoon PuTTYgenillä

5.5 LAMP-web-palvelimen asentaminen

Seuraavaksi asennetaan LAMP-web-palvelin, johon sisältyy Apache-www-palvelin, PHP-tuki sekä MySQL-tietokantaohjelmisto. Palvelinta voidaan käyttää staattisen nettisivun ylläpitoon tai sinne voi sijoittaa dynaaminen PHP-sovellus, joka lukee ja kirjoittaa tietoa tietokantaan. Ensimmäiseksi otetaan yhteys instanssiin PuTTYn avulla, ja sen jälkeen on suositeltavaa ajaa instanssiin viimeisimmät ohjelmistopäivitykset. Päivitysten jälkeen asennetaan Apache-www-palvelin, MySQL-tietokanta sekä PHP-tuki. (KUVIO 15.) Kun nämä on asennettu ja käynnistetty, voidaan toimivuutta testata laittamalla selaimeen instanssin joko julkinen DNS- tai julkinen IP-osoite. (AWS 2014e.)


```

ec2-user@ip-172-31-28-175:~
device-mapper-persistent-data.x86_64 0:0.2.8-2.3.amzn1
dracut.noarch 0:004-336.21.amzn1
libcurl.x86_64 0:7.33.0-1.41.amzn1
libjpeg-turbo.x86_64 0:1.2.1-3.3.amzn1
lvm2.x86_64 0:2.02.100-8.22.amzn1
lvm2-libs.x86_64 0:2.02.100-8.22.amzn1
perl.x86_64 4:5.10.1-136.21.amzn1
perl-Digest-SHA.x86_64 1:5.47-136.21.amzn1
perl-Module-Pluggable.x86_64 1:3.90-136.21.amzn1
perl-Pod-Escapes.x86_64 1:1.04-136.21.amzn1
perl-Pod-Simple.x86_64 1:3.13-136.21.amzn1
perl-libs.x86_64 4:5.10.1-136.21.amzn1
perl-version.x86_64 3:0.77-136.21.amzn1
python-backports-ssl_match_hostname.noarch 0:3.4.0.2-1.5.amzn1
python-bcdoc.noarch 0:0.12.0-1.0.amzn1
python-boto.noarch 0:2.19.0-1.0.amzn1
python-botocore.noarch 0:0.27.0-1.0.amzn1
python-jmespath.noarch 0:0.2.0-1.0.amzn1
sudo.x86_64 0:1.8.6p3-12.17.amzn1
tar.x86_64 2:1.26-27.21.amzn1

Complete!
[ec2-user@ip-172-31-28-175 ~]$ sudo yum groupinstall -y "Web server" "MySQL Data
base" "PHP Support"

```

KUVIO 15. LAMP-web-palvelimen asennuskomento PuTTYlla

Tämän jälkeen luodaan instanssiin www-ryhmä ja siihen ryhmään lisätään instanssin oletuskäyttäjälle omistajuus sekä kirjoitusluvat /var/www-hakemistoon, (KUVIO 16.) jotta tiedostojen ja sovellusten lisääminen palvelimelle olisi mahdollista. Tämän jälkeen palvelinta voidaan testata luomalla yksinkertainen PHP-tiedosto, joka näyttää PHP-informaationsivun. (AWS 2014e.)


```

ec2-user@ip-172-31-28-175:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Tue Dec 17 11:30:53 2013 from dsl-sjkbrasgw2-54f8ec-59.dhcp.inet.fi

  _ | _ | _ |
  _ | ( _ | /
  _ | \ _ | _ |
                Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2013.09-release-notes/
[ec2-user@ip-172-31-28-175 ~]$ groups
ec2-user wheel www
[ec2-user@ip-172-31-28-175 ~]$ sudo chown -R root:www /var/www
[ec2-user@ip-172-31-28-175 ~]$ sudo chmod 2775 /var/www
[ec2-user@ip-172-31-28-175 ~]$ find /var/www -type d -exec sudo chmod 2775 {} +
[ec2-user@ip-172-31-28-175 ~]$ find /var/www -type f -exec sudo chmod 0664 {} +
[ec2-user@ip-172-31-28-175 ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php

```

KUVIO 16. Omistajuus- sekä kirjoituslupakomennot ja phpinfo-testitiedoston luominen PuTTYlla

5.6 WordPress-blogin isännöinti

Koska blogin kirjoittaminen on itseäni aina kiinnostanut, päätin asentaa aiemmin asentamalleni LAMP-web-palvelimelle WordPress-blogin. Näin palvelimen testaaminen saadaan laajemmaksi ja sen asentaminen voidaan myös sisällyttää tuleviin laboratorioharjoituksiinkin. WordPress on PHP-kielellä kirjoitettu, ja se käyttää MySQL-tietokantaa tietojen tallentamiseen, joten näin saadaan hyödyksi kaikki asennetut paketit LAMP-web-palvelimesta.

WordPress-blogin asentaminen onnistuu PuTTYlla. Ensin wget-komennolla haetaan ja ladataan viimeisin WordPressin asennuspaketti ja puretaan se sen jälkeen wordpress-nimiseen kansioon. Tämän jälkeen käynnistetään jo aiemmin asennettu MySQL-palvelin ja kirjaudutaan aiemmin luodulla salasanalla sisään. Sitten luodaan tietokannalle käyttäjä ja keksitään salasana. Tämän jälkeen luodaan ja nimeetään tietokanta blogia varten sekä annetaan kaikki etuoikeudet äsken luodulle käyttäjälle juuri luodulle tietokannalle. Tämän jälkeen poistetaan vanhat etuoikeu-

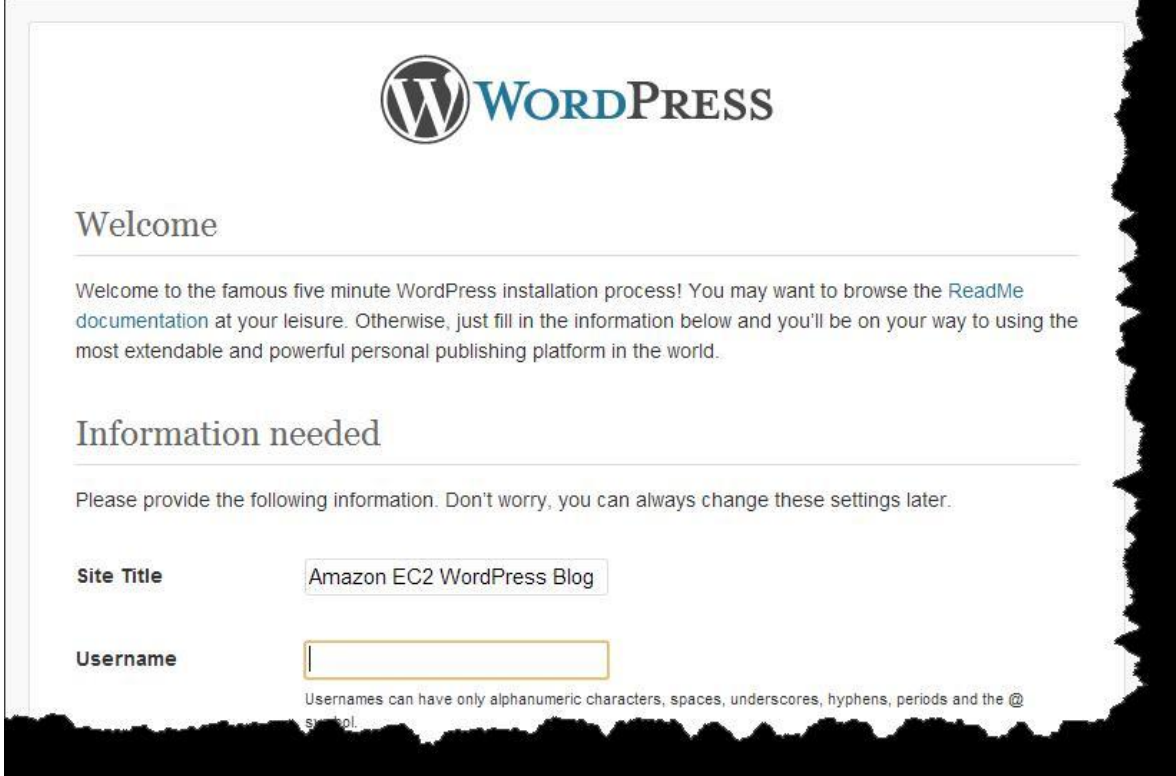
det, jotta tehdyt muutokset tulevat voimaan. Tämän jälkeen voidaan poistaa MySQL:stä. (AWS 2014f.)


Nyt luodaan WordPressin konfigurointitiedosto kopiaamalla wp-config-sample.php-tiedosto wp-config.php-tiedostoksi, jotta sample-tiedosto jää ikään kuin varmuuskopioksi. Sitten avataan wp-config.php-tiedosto jollakin tekstinmuokkausohjelmalla, esimerkiksi nanoilla, ja muutetaan siellä kohtiin DB_NAME, DB_USER ja DB_PASSWORD aiemmin luodut nimet tietokannalle ja käyttäjälle sekä käyttäjän salasana. (AWS 2014f.)

Sitten selataan tekstiä alaspäin, kunnes löydetään kohta "Authentication Unique Keys and Salts". Tähän kohtaan haetaan WordPressin sivuilta satunnaisesti luodut avainarvot ja kopioidaan ne oikeille kohdille. Nämä arvot haetaan siksi, että ne tarjoavat salauskerroksen selaimen evästeille, jotka WordPressin käyttäjät tallentavat paikallisille koneilleen. Kun arvot on kopioitu oikeille kohdilleen, voidaan tiedosto tallentaa ja samalla poistaa tekstinmuokkausohjelmasta. (AWS 2014f.)

Tämän jälkeen voidaan siirtää WordPressin asennus Apachen juureen tai alikansioon. Tämän jälkeen lisätään apache-käyttäjä LAMP-web-palvelinta asennettaessa luotuun www-ryhmään ja muutetaan omistajuus sekä luku- ja kirjoitusoikeudet samalla periaatteella kuin aiemmin ec2-user-käyttäjällekin LAMP-web-palvelinta asennettaessa. (AWS 2014f.)

Tämän jälkeen mennään selaimella instanssin julkiseen IP- tai DNS-osoitteeseen ja näkyviin tulee WordPressin asennusruutu. (KUVIO 17.) Ruudussa annetaan blogille nimi, luodaan käyttäjä ja salasana sekä annetaan sähköpostiosoite. Tämän jälkeen WordPress-blogi on valmis käytettäväksi, kun sinne kirjaututaan sisään. Kun WordPressiin on kirjattu sisään, voidaan sen asetuksia ja ulkoasua muokata sieltä käsin. (AWS 2014f.)





Welcome

Welcome to the famous five minute WordPress installation process! You may want to browse the [ReadMe documentation](#) at your leisure. Otherwise, just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol.

KUVIO 17. WordPress-blogin asennusruutu (AWS 2014f.)

6 HARJOITUSTEHTÄVIEN SUUNNITTELU JA OHJEISTUS

Tässä luvussa esitetään Centria-ammattikorkeakoulun opiskelijoille harjoitustehtäviä liittyen Amazonin Elastic Compute Cloud -tuotteeseen. Harjoitustehtäviin sisällytetään niin tilin ja instanssin luominen kuin LAMP-web-palvelimen asennuskin. Koska tilin ja instanssin luomiseen ei aikaa yhteensä paljoa kulu, sisällytin ne samaan harjoitustehtävään. Kolmantena harjoitustehtävänä on WordPress-blogin asentaminen, jotta saataisiin varmasti käytettyä hyödyksi myös MySQL-tietokanta sekä PHP-tuki. Tiliä luodessa voidaan käyttää joko koulun omaa luottokorttia tai opiskelijat voivat halutessaan käyttää omia luottokorttejaankin. Amazoniin rekisteröityessä alkaa kahdentoista kuukauden ilmainen kokeilu, joten pelkoa laskun keraantymisestä ei ole. Ja tilin voi tietenkin poistaa, ennen kuin kokeilujakso päättyy.

Harjoitustehtäviä varten tarvitaan kolme ohjelmaa. Ensimmäiseksi tarvitaan PuTTY, jonka avulla saadaan yhteys Linux-pohjaiseen instanssiin. Sitten tarvitaan PuTTYgen, jolla muutetaan avainparin tiedostomuoto PuTTYn ymmärtämään tiedostomuotoon. Kolmantena voidaan käyttää esimerkiksi WinSCP:tä, jolla voidaan kätevästi siirtää tiedostoja testausta varten palvelimelle, kun se on konfiguroitu valmiiksi.

6.1 Harjoitustehtävä 1 – Tilin ja instanssin luominen

Rekisteröityminen Amazoniin tapahtuu vaiheittain ja onnistuu kun seuraa ruudussa olevia ohjeita. Yksityiskohtainen ohjeistus tilin luomiseen löytyy liitteestä 1. Samoin instanssin luominen tapahtuu vaiheittain ja on selkeää kun lukee ruudulla näkyvät ohjeet. Myös instanssin luomiseen löytyy yksityiskohtainen ohjeistus liitteestä 1.

6.2 Harjoitustehtävä 2 – LAMP-web-palvelimen asennus ja testaus

Harjoitustehtävässä 1 luotu avainpari muunnetaan PuTTYgenillä .pem-tiedostomuodosta PuTTYn ymmärtämään .ppk-tiedostomuotoon. Instanssiin ote-

taan yhteys PuTTYlla ja ajetaan sinne viimeisimmät päivitykset. Tämän jälkeen asennetaan LAMP-web-palvelin, johon kuuluu Apache-www-palvelin sekä MySQL-tietokantaohjelmisto sekä PHP-tuki. Tämän jälkeen lisätään www-ryhmä ja siihen ryhmään lisätään instanssin oletuskäyttäjälle omistajuus sekä kirjoitusluvat web-palvelimen /var/www-hakemistoon. Tämän jälkeen palvelinta voi testata lisäämällä sinne WinSCP:llä haluamiaan tiedostoja tai vaikka nettisivut. Yksityiskohtainen ohjeistus PuTTYn kaikkiin tarvittaviin komentoihin ja tiedostojen lisäämiseen WinSCP:llä löytyy liitteestä 2.

6.3 Harjoitustehtävä 3 – WordPress-blogin asennus ja testaus

Nyt kun edellisissä harjoitustehtävässä on saatu kokonaisuudessaan käyttövalmiiksi LAMP-web-palvelin, voitaisiin ottaa hyödyksi LAMP:n MySQL-tietokanta sekä PHP-tuki WordPress-blogin asentamisen muodossa. Opiskelijat voisivat kirjoittaa blogiin raportin tilin ja instanssin luomisesta Amazon Web Servicesiin, LAMP-web-palvelimen asentamisesta ja testaamisesta sekä tästä WordPress-blogin asentamisesta.

WordPress-blogia asennettaessa PuTTYlla ensin ladataan sen asennuspaketti, joka puretaan latauksen jälkeen WordPress-kansioon. Tämän jälkeen käynnistetään MySQL-palvelin, jonne luodaan käyttäjä sekä tietokanta blogia varten, ja annetaan luodulle käyttäjälle kaikki etuoikeudet juuri luotuun tietokantaan. Tämän jälkeen poistutaan MySQL-asiakkaasta. Nyt kopioidaan wp-config-sample.php-tiedosto tiedostoksi wp-config.php, jotta alkuperäinen sample-tiedosto jää ikään kuin varmuuskopioksi.

Tämän jälkeen avataan wp-config.php-tiedosto jollakin tekstinmuokkausohjelmalla, esimerkiksi nanoilla. Tiedostoon muokataan kohtiin DB_NAME, DB_USER sekä DB_PASSWORD aiemmin luodut nimet käyttäjälle ja tietokannalle sekä salasana.

Tämän jälkeen tiedoston kohtaan "Authentication Unique Keys and Salts" haetaan WordPressin nettisivuilta satunnaisesti luodut avainarvot ja kopioidaan ne järjestyksessä oikeisiin kohtiin. Tämä tehdään siksi, että KEY- ja SALT-arvot antavat

salauskerroksen selaimen evästeille, jotka WordPressin käyttäjät tallentavat omille tietokoneilleen. Nyt tallennetaan tiedosto ja poistutaan tekstinmuokkausohjelmasta.

Tämän jälkeen siirretään tiedostot, jotka WordPress-kansioon luotiin, Apachen juureen tai vaihtoehtoisesti juureen voidaan tehdä alikansio, johon tiedostot siirretään. Tämän jälkeen muokataan omistajuus ja kirjoitusluvat Apache-palvelimelle eli käyttäjä apache lisätään aiemmassa laboratorioharjoituksessa luotuun www-ryhmään ja muokataan omistajuus sekä kirjoitusluvat samalla periaatteella. Tämän jälkeen uudelleenkäynnistetään Apache-palvelin, jotta muutokset tulevat voimaan.

Nyt voidaan testata WordPress-blogia menemällä selaimella instanssin julkiseen IP- tai DNS-osoitteeseen, ja ruutuun pitäisi ilmestyä WordPress-blogin asennusruutu. Asennusruutuun annetaan blogille nimi ja luodaan ylläpitokäyttäjää ja salasana sekä annetaan sähköpostiosoite. Nyt blogin asennus on valmis ja sitä voidaan testata kirjoittamalla raportti näistä harjoitustehtävistä ensimmäiseen blogikirjoitukseen. Yksityiskohtainen ohjeistus PuTTY-komentoineen sekä tiedostomuokkaukseen löytyy liitteestä 3.

7 YHTEENVETO

Tämän opinnäytetyön tarkoituksena oli tutustua pilvipalveluiden maailmaan. Teoriaosuuteen oli tarkoituksena etsiä ja kirjoittaa oleelliset tiedot ja käsitteet, jotka yhdessä määrittelevät sen, millaisia käsitteitä ovat pilvilaskenta sekä pilvipalvelu. Käytännön osuudessa oli tarkoituksena lähteä tutustumaan yhteen pilvipalveluntarjoajaan ja tutustua sen tarjontaan. Pilvipalveluntarjoajaksi valitsin Amazon Web Servicesin. Tämän jälkeen oli oman oppimisen ja kokeilemisen perusteella tavoitteena suunnitella ja ohjeistaa muutama harjoitustehtävä Centria-ammattikorkeakoulun tietotekniikan opiskelijoille, jotta hekin pääsisivät tutustumaan paremmin pilvipalveluihin.

Luodessani tiliä Amazon Web Servicesiin alkoi kahdentoista kuukauden ilmainen kokeilujakso, joten huolta rahan menosta ei ollut tätä opinnäytetyötä tehdessä. Tutustuttuani yleisesti Amazonin pilvipalvelutarjontaan oli sen jälkeen tarkoituksena luoda sinne virtuaalipalvelin ja testata sen käyttöä. Infrastructure-as-a-Service-toimintamallia edustavan virtuaalipalvelimen luominen onnistui Amazonin Elastic Compute Cloud (EC2) -tuotteella. Valitsin virtuaalipalvelinta luodessani Linux-pohjaisen instanssin ja asensin sinne LAMP-web-palvelimen kokonaisuudessaan, eli siihen kuuluivat Apache-palvelin, MySQL-tietokantaohjelmisto sekä PHP-tuki. SSH-yhteyden instanssiin sain PuTTY-ohjelmalla, jonka avulla LAMP-web-palvelimen asentaminen sitten onnistui. LAMP-web-palvelimen toimivuutta testasin upottamalla palvelimelle aikaisemmin tekemäni valmiit nettisivut. Myöhemmin vielä päätin asentaa sinne WordPress-blogin, jotta myös MySQL-tietokanta sekä PHP-tuki tulisivat hyötykäyttöön.

Kynnys lähteä tekemään virtuaalipalvelinta Amazoniin oli matala, sillä Amazon tarjosi mielestäni hyvän ja selkeän dokumentaation itse palveluun rekisteröitymiseen ja instanssin luomiseen eli EC2-tuotteen käyttämiseen sekä myös hyvän ohjeistuksen LAMP-web-palvelimen asentamiseen Linux-pohjaiseen instanssiin. Voisin väittää, että ilman näitä ohjeistuksia en olisi saanut mitään aikaan tai ainakin siinä olisi kestänyt liian pitkään, koska en ole aikaisemmin paljoa käyttänyt PuTTYä.

Tätä opinnäytetyötä tehdessäni päätin ottaa itselleni käyttöön vielä yhden Amazonin tarjoaman palvelun, joka oli Simple Storage Service (S3). S3:n antaa pilvitalennustilaa Internetiin, ja se oli mielestäni hyvin selkeä ja helppokäyttöinen tuote. Käytin S3:a ottaakseni varmuuskopiot Internetiin tästä tekemästäni opinnäytetyöraportista sekä ottamistani kuvista.

Oman oppimisen ja kokeilemisen perusteella oli sitten tarkoituksena suunnitella ja ohjeistaa harjoitustehtäviä. Niitä tuli yhteensä kolme kappaletta. Ensimmäisessä harjoitustehtävässä luodaan Amazoniin tili sekä instanssi EC2-tuotteella. Toisessa harjoitustehtävässä asennetaan luotuun instanssiin LAMP-web-palvelin ja testataan sen toimivuutta upottamalla sinne tiedostoja tai mitä opiskelijat itse ikinä haluavatkin. Kolmannessa harjoitustehtävässä asennetaan palvelimelle WordPress-blogi. Tässä harjoituksessa viimeistään tulee käyttöön myös MySQL-tietokanta sekä PHP-tuki, mikäli edellisessä harjoituksessa ei mitään näitä tarvittavia sovelluksia asennettu. Opinnäytetyön luvussa kuusi on lyhyet selitykset siitä, mitä kussakin harjoitustehtävässä tehdään, ja lopussa olevissa liitteissä on täydelliset ja yksityiskohtaiset ohjeistukset esimerkiksi PuTTY-komentoineen harjoitustehtävien suorittamista varten.

Omasta mielestäni harjoitustehtävien ohjeistaminen onnistui hyvin. Ohjeistuksista tuli mielestäni hyvin selkeät ja kohta kohdalta etenevät. Itse jos olisin vielä untuvikko, niin uskoisin vahvasti, että pystyisin näiden ohjeistusten avulla suoriutumaan harjoitustehtävistä hyvin.

Toivon, että näiden harjoitustehtävien tekeminen on mielekästä, ja vielä enemmän toivon, että näiden harjoitustehtävien pohjalta opiskelijoiden mielenkiinto lähteä tutustumaan pilvipalveluihin syvemmin nousisi. Nämä harjoitustehtävät varmaan antavat hyvän pohjakäsityksen siitä, millaisia pilvipalvelut ovat käyttää, ainakin Amazon Web Servicesin EC2-tuotteen osalta. Enkä usko, että muut samankaltaisia pilvipalveluja tarjoavat yritykset Amazonista konkreettisesti kovin paljoa poikkeaisivat.

Mikä ehkä on kaikista tärkeintä, oma mielenkiintoni pilvipalveluihin tämän opinnäytetyön tekemisen myötä kasvoi. Ennen tämän opinnäytetyön aloittamista pilvipalvelut olivat minulle kuitenkin aika tuntematon käsite. Tämän opinnäytetyön tekemisen myötä esimerkiksi tuon WordPress-blogin isännöinti voisi olla ihan arkipäivää itsellä jatkossakin, eikä se Amazonin pilvipalvelut ilmaisen kokeilujakson jälkeen kovin kallista olisi.

LÄHTEET

Arcitura Education. 2014. Public clouds. Www-dokumentti. Saatavissa: http://whatiscloud.com/cloud_deployment_models/public_clouds. Luettu 16.1.2014.

AWS. 2014a. Elastic Compute Cloud. Www-dokumentti. Saatavissa: <http://aws.amazon.com/ec2/>. Luettu 3.2.2014.

AWS. 2014b. Simple Storage Service. Www-dokumentti. Saatavissa: <http://aws.amazon.com/s3/>. Luettu 25.2.2014.

AWS. 2014c. Launch an Amazon EC2 Instance. Www-dokumentti. Saatavissa: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance_linux.html. Luettu 4.2.2014.

AWS. 2014d. Connect to Your Instance. Www-dokumentti. Saatavissa: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-connect-to-instance-linux.html>. Luettu 4.2.2014.

AWS. 2014e. Installing a LAMP Web Server. Www-dokumentti. Saatavissa: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html>. Luettu 5.2.2014.

AWS. 2014f. Hosting a WordPress Blog with Amazon EC2. Www-dokumentti. Saatavissa: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/hosting-wordpress.html>. Luettu 11.3.2014.

Bair, J. & Rhodon, J. 2010. Cloud Computing from the Ground Up. Pdf-dokumentti. Saatavissa: http://www.meredithsmith-cpa.com/images/Cloud_Computing_White_paper_123110.pdf. Luettu 27.3.2014.

C⁴S. 2014a. What are service models in cloud computing? Www-dokumentti. Saatavissa: <http://www.cloud-competence-center.com/understanding/cloud-computing-service-models/>. Luettu 15.1.2014.

C⁴S. 2014b. What are deployment models in cloud computing? Www-dokumentti. Saatavissa: <http://www.cloud-competence-center.com/understanding/cloud-computing-deployment-models/>. Luettu 16.1.2014.

Carmen, M. 2012. Cloud computing concerns and issues. Www-dokumentti. Saatavissa: <http://www.onbile.com/info/what-cloud-computing-means/>. Luettu 26.2.2014.

Cloud.cio.gov. 2014. Cloud computing deployment models. Www-dokumentti. Saatavissa: <http://cloud.cio.gov/topics/cloud-computing-deployment-models>. Luettu 16.1.2014.

Czarnecki, C. 2011. Cloud service models: comparing Saas, Paas and Iaas. Www-dokumentti. Saatavissa: <http://cloud-computing.learningtree.com/2011/11/09/cloud-service-models-comparing-saas-paas-and-iaas/>. Luettu 15.1.2014.

Dialogic. 2010. Introduction to Cloud Computing. Pdf-dokumentti. Saatavissa: <http://www.dialogic.com/~media/products/docs/whitepapers/12023-cloud-computing-wp.pdf>. Luettu 25.3.2014.

Grance, T & Mell, P. 2011. The NIST Definition of Cloud Computing. Pdf-dokumentti. Saatavissa: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>. Luettu 3.3.2014.

Griffith, E. 2013. What is cloud computing? Www-dokumentti. Saatavissa: <http://www.pcmag.com/article2/0,2817,2372163,00.asp>. Luettu 14.1.2014.

Walker, G. 2010. Cloud computing fundamentals. Www-dokumentti. Saatavissa: <http://www.ibm.com/developerworks/cloud/library/cl-cloudintro/index.html>. Luettu 26.2.2014.

Harjoitustehtävä 1 – Tilin ja instanssin luominen

Ensimmäiseksi luodaan tili Amazoniin. Amazon Web Services -sivustolla (aws.amazon.com) painetaan "sign up" -painiketta. Ensimmäisessä vaiheessa annetaan sähköpostiosoite, joka toimii käyttäjätunnuksena. Sitten annetaan sähköpostiosoite uudelleen sekä keksitään salasana. Seuraavaksi tulee antaa nimi- ja osoitetiedot, puhelinnumero sekä hyväksyä Amazonin säännöt ja ehdot. Rekisteröitymisen kolmannessa vaiheessa annetaan luottokorttitiedot. Neljännessä vaiheessa todennetaan henkilöllisyys puhelinsoitolla Amazoniin. Ruudussa näkyy aiemmin annettu puhelinnumero sekä "Call me now" -painike. Tätä klikkaamalla puhelin alkaa soida, ja samalla ruutuun ilmestyy PIN-koodi, joka näppäillään puhelimeen kun, robottiäni antaa siihen luvan. Viimeiseksi valitaan tukisuunnitelma Amazoniin. Tässä kannattaa valita "Basic" eli perus, koska se on ilmainen ja täysin riittävä. Nyt tili Amazoniin on luotu.

Nyt kun tili on luotu, päästään luomaan instanssi. Kun on kirjautettu sisään, valitaan "My Account / Console" -valinnasta "AWS Management Console" ja sieltä edelleen EC2. Tämän jälkeen voidaan haluttaessa vaihtaa aluetta, johon instanssi tulee. Oikeassa yläkulmassa on aluevalinta, jossa oletuksena lukee "Oregon". Sieltä voidaan valita Amazonin resurssit käytettäväksi esimerkiksi Euroopasta. Tämän jälkeen painetaan suurin piirtein ruudun keskellä olevaa "Launch Instance" -painiketta. Instanssin luomisen ensimmäisessä vaiheessa valitaan Amazon Machine Image (AMI). Oikealla voidaan tarkistaa "Free tier only" -laatikko, jolloin kaikki maksulliset AMI:t karsiutuvat valinnoista. Jäljelle jäävistä valitaan Amazon Linux AMI 64-bittisenä. Seuraavassa vaiheessa valitaan instanssin koko. Koska kyseessä on ilmainen kahdentoista kuukauden kokeilu, on ainoa mahdollinen instanssin koko mikro. Seuraavaksi tulee yhteenveto instanssista, ja kun kaikki on ok, painetaan "Launch Instance" -painiketta. Tämän jälkeen ruutuun ilmestyy avainparin tekoa varten ikkuna, jossa tehdään uusi avainpari, annetaan sille haluttu nimi ja tallennetaan se jonnekin omalle tietokoneelle. Nyt painetaan viimeisin kerran "Launch Instance" -painiketta, ja näin instanssin luonti on valmis.

Harjoitustehtävä 2 – LAMP-web-palvelimen asennus ja testaus

Ensimmäiseksi täytyy muuntaa avainparin .pem-tiedosto muotoon .ppk PuTTY-genillä, jotta yhteyden muodostaminen instanssiin PuTTYn avulla voisi onnistua. Käynnistä PuTTYgen ja laita alhaalla näkyvään avaintyyppiin SSH-2 RSA, ellei se ole jo oletuksena. Tämän jälkeen paina "Load" ja hae aiemmin tekemäsi avainpari polusta, jonne sen tallensit. Avaa se ja tallenna se privaattina. Paina "Yes", mikäli PuTTYgen varoittaa avainparin tallentamisesta ilman tunnuslausetta. Anna avainparille sama nimi kuin aiemmassa laboratorioharjoituksessa, PuTTYgen antaa sille automaattisesti .ppk-muodon.

Nyt otetaan instanssiin yhteys PuTTY:n avulla. Avaa PuTTY ja laita Host name-kenttään käyttäjätunnus, joka oletuksena Amazon Linux-instanssissa on ec2-user, sekä instanssin julkinen DNS-osoite "@"-merkillä erotettuna ec2-user:sta. DNS-osoitteen löydät Amazonin sivuilta AWS Management Consolesta instanssin tiedoista. Portti on 22 ja yhteyden tyyppi SSH. Sitten PuTTYn Category-paneelistä avataan Connection->SSH->Auth-polku. Paina "Browse" ja hae aiemmin tekemäsi avainpari .ppk-muodossa. Tämän jälkeen käynnistä sessio, ja mikäli PuTTY näyttää turvavaroituslaatikon kysymällä että luotatko tähän isäntään vastaa kyllä.

Seuraavaksi päästään asentamaan LAMP web-palvelinta PuTTYlla. Ensimmäiseksi kannattaa päivittää viimeisimmät ohjelmistopäivitykset instanssiin.

```
[ec2-user ~]$ sudo yum update -y
```

Tämän jälkeen voidaan asentaa web-palvelin, MySQL-tietokanta sekä PHP-tuki. Ne kaikki voidaan asentaa kerralla yhdellä komennolla.

```
[ec2-user ~]$ sudo yum groupinstall -y "Web Server" "MySQL Database" "PHP-Support"
```

Seuraavaksi asennetaan php-mysql-paketti.

```
[ec2-user ~]$ sudo yum install -y php-mysql
```

Sitten kun paketit on asennettu, voidaan käynnistää Apache-www-palvelin.

```
[ec2-user ~]$ sudo service httpd start
```

Seuraavaksi komennetaan Apache-www-palvelin käynnistymään joka kerta kun instanssikin käynnistyy.

```
[ec2-user ~]$ sudo chkconfig httpd on
```

Nyt LAMP web-palvelimen asennus on suoritettu. Sen toimivuutta voidaan testata menemällä selaimella instanssin julkiseen IP-osoitteeseen tai julkiseen DNS-osoitteeseen. Mikäli Amazon Linux AML:n testisivu aukeaa näkyviin, on asennus onnistunut.

Seuraavassa vaiheessa lisätään www-ryhmä instanssiin ja siihen ryhmään lisätään oletuskäyttäjä ec2-user ja hänelle omistajuus ja kirjoitusoikeudet /var/www-hakemistoon, jotta tiedostojen lisääminen, poistaminen ja muokkaaminen onnistuvat. Ensimmäiseksi lisätään www-ryhmä.

```
[ec2-user ~]$ sudo groupadd www
```

Seuraavaksi lisätään käyttäjä www-ryhmään.

```
[ec2-user ~]$ sudo usermod -a -G www ec2-user
```

Tämän jälkeen kirjaudutaan ulos "exit"-komennolla ja tullaan sitten takaisin, jotta juuri luotuun ryhmään päästään käsiksi. Tämän jälkeen "groups"-komennolla voidaan tarkistaa käyttäjän ec2-user jäsenyys www-ryhmässä. Sitten vaihdetaan www-ryhmän omistajuutta.

```
[ec2-user ~]$ sudo chown -R root:www /var/www
```

Tämän jälkeen muutetaan hakemiston /var/www ja sen alihakemistojen luvat, jotta www-ryhmän jäsenet saavat kirjoitusluvut niihin.

```
[ec2-user ~]$ sudo chmod 2775 /var/www
```

```
[ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} +
```

Sitten rekursiivisesti muutetaan /var/www-hakemiston ja sen alihakemistojen tiedostoluvat, jotta www-ryhmä voi niihin kirjoittaa.

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} +
```

Nyt on www-ryhmä luotu ja lisätty sinne käyttäjä ec2-user sekä annettu sille ryhmälle omistajuus ja kirjoitusoikeudet /var/www-hakemistoon. Nyt voidaan testata LAMP-web-palvelinta. Luodaan yksinkertainen php-tiedosto.

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Sitten avataan selaimella juuri tehty tiedosto. Se tapahtuu laittamalla julkinen IP-osoite tai julkinen DNS-osoite kautta tiedoston nimi. Esimerkiksi <http://julkinendns/phpinfo.php>.

Mikäli PHP-version informaatio sivu aukeaa näkyviin, on omistajuudet ja kirjoitusluvat tehty oikein. Turvallisuussyistä tämä tiedosto kannattaa palvelimelta poistaa.

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

Lopuksi kannattaa turvata MySQL-palvelin, vaikka sitä ei nyt tarvitsisikaan. Ensin käynnistetään se.

```
[ec2-user ~]$ sudo service mysqld start
```

Sitten ajetaan mysql_secure_installation.

```
[ec2-user ~]$ sudo mysql_secure_installation
```

Nyt PuTTY kysyy salasanaa. Koska root-tunnuksella ei oletuksena ole salasanaa, painetaan "enter". Sitten painetaan "Y" ja keksitään hyvä salasana, joka joudutaan

kirjoittamaan kahdesti. Tämän jälkeen painetaan taas "Y", jotta anonyymit käyttäjätunnukset poistuvat. Sitten painetaan taas "Y", jotta root-käyttäjän etäkirjautuminen poistuu, ja jälleen "Y", jotta testitietokanta poistuu, ja vielä kerran "Y", jotta etuoi-keutetut taulut uudelleenlatautuvat ja muutokset tallentuvat. Voit halutessasi tämän jälkeen pysäyttää MySQL-palvelimen, jos et tarvitse sitä nyt, tai voit komen-
taa sen käynnistymään aina instanssin käynnistyessäkin.

```
[ec2-user ~]$ sudo service mysqld stop
```

```
[ec2-user ~]$ sudo chkconfig mysqld on
```

Tämän harjoitustehtävän loppuksi voidaan vielä lisätä tiedostoja tai vaikka nettisivut palvelimelle WinSCP:n avulla. Käynnistä WinSCP ja laita Host name -kenttään instanssin julkinen DNS-osoite ja User name -kenttään instanssin oletuskäyttäjätunnus sekä hae Private key -kohtaan tietokoneeltasi avainpari PuTTYn ymmärtämässä .ppk-tiedostomuodossa. Nyt voit kirjautua sisään ja hakea palvelimelta hakemiston /var/www/html ja upottaa sinne mitä haluat.

Harjoitustehtävä 3 – WordPress-blogin asennus ja testaus

Tässä harjoitustehtävässä asennetaan LAMP web-palvelimelle WordPress-blogi, jotta LAMPiin asennetut MySQL-tietokanta ja PHP-tuki saadaan hyödyksi. Ensimmäiseksi avataan PuTTY ja kirjaudutaan instanssiin sisään samalla tavalla kuin aikaisemmissa harjoituksissa. Sitten ladataan WordPressin asennuspaketti wget-komennolla.

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
```

Tämän jälkeen puretaan WordPress-asennuspaketti.

```
[ec2-user ~]$ tar -xzf latest.tar.gz
```

```
[ec2-user ~]$ ls
```

Tämän jälkeen käynnistetään MySQL-palvelin, jotta tietokanta WordPressiä varten voidaan luoda.

```
[ec2-user ~]$ sudo service mysqld start
```

Sitten kirjaudutaan sisään aiemmassa harjoitustehtävässä tehdyllä salasanalla.

```
[ec2-user ~]$ mysql -u root -p
```

Enter password:

Nyt on kirjauduttu MySQL:ään. Nyt luodaan käyttäjänimi ja salasana MySQL-tietokantaan, jota WordPress tulee käyttämään.

```
mysql> CREATE USER 'käyttäjänimi'@'localhost' IDENTIFIED BY 'salasana';
```

Tämän jälkeen luodaan tietokanta WordPress-blogia varten.

```
mysql> CREATE DATABASE `tietokanta_nimi`;
```

Tämän jälkeen annetaan täydet etuoikeudet juuri luodulle käyttäjälle juuri luotuun tietokantaan.

```
mysql> GRANT ALL PRIVILEGES ON `tietokanta_nimi` .* TO "käyttäjänimi"@localhost";
```

Tämän jälkeen poistetaan vanhat MySQL-etuoikeudet, jotta tehdyt muutokset tulevat voimaan.

```
mysql> FLUSH PRIVILEGES;
```

Sitten voidaan poistua MySQL-asiakkaasta.

```
mysql> exit
```

WordPress-blogin asennuspaketti sisältää esimerkkikonfiguraatitiedoston, joka seuraavaksi kopioidaan ja muokataan. Tiedosto kopioidaan erinimiseksi, koska esimerkkitiedosto jätetään varmuuskopioksi.

```
[ec2-user ~]$ cd wordpress/
```

```
[ec2-user wordpress]$ cp wp-config-sample.php wp-config.php
```

Tämän jälkeen muokataan wp-config.php-tiedostoa tekstinmuokkausohjelmalla, esimerkiksi nanolla.

```
[ec2-user wordpress]$ nano wp-config.php
```

Tiedostoon muokataan aiemmin annetut nimet tietokannalle ja käyttäjälle sekä salasana eli seuraavat kohdat:

```
define ('DB_NAME', 'tietokanta_nimi');
```

```
define ('DB_USER', 'käyttäjänimi');
```

```
define ('DB_PASSWORD', 'salasana');
```

Tämän jälkeen selataan tekstiä alaspäin, kunnes löydetään kohta ”Authentication Unique Keys and Salts”. Tähän kohtaan haetaan WordPressin sivuilta avainarvot, jotka antavat salauskerroksen selaimen evästeille, jotka WordPress-blogin käyttäjät tallentavat. Avainarvot luodaan satunnaisesti, ja ne löytyvät osoitteesta:

```
https://api.wordpress.org/secret-key/1.1/salt/
```

Nämä kopioidaan oikeille kohdille tekstiin. Seuraavassa on esimerkki siitä, miten niiden tulee olla:

```
define('AUTH_KEY',          'VUS=W|FY2]1`Jb3OhDsnQqDyKpx0`DMHVi&%)Ks+(; -4olwWy&,%IE4+HWJaq48');
define('SECURE_AUTH_KEY',  '7B>`<g>]]1 Q.`Df+|C<F:o8][GPUF-:pb7;><<usNg0V5jK8aK5Fp-A;~^K*/u]');
define('LOGGED_IN_KEY',    '2R0|PZbH+Nn5cc@7ln]]4@lpE+#]Y&}}6h5,W6-9$[j|F]zm]+pg%?4!Do)hg0');
define('NONCE_KEY',        '>+}*n-i>a8fWuL^GTF*LN-HHoP_zlKcDCI!Snh{1@2`B!Ra6+2;7!tzJl-g$}}1b');
define('AUTH_SALT',        'q_6J+W}u*k|b%QR2T-nFU[`ltbZh^H+fYgPwWS*10T1[g*-->;Qsye?Vs4EvO3U');
define('SECURE_AUTH_SALT', 'SNc-`hspXg<(fGf3R?SO][X8{9[ )lx2lDd,?29 /8rO2yB/`Cg#VO}&Hk:VclW');
define('LOGGED_IN_SALT',   ' dtvN,e{_,uvj:Zy`$MKk:x~Uqk3&U:l.~gU>&L>!lgSz+mWlQjm9H|y0+-oo,>Q');
define('NONCE_SALT',       '&KOEztfS04e(F fMXfgA[nwwxWt+$4|Bh?Q l,/jGe6pP]kkvmQ>P<7ka7^;A:uP');
```

Tämän jälkeen tiedosto tallennetaan nanolla painamalla CTRL+O. Sitten nanosta voidaan poistua painamalla CTRL+X.

Sitten siirretään WordPressin asennustiedostot Apachen dokumenttijuureen. Voit halutessasi tehdä myös alikansion, johon tiedostot siirretään.

```
[ec2-user wordpress]$ mv * /var/www/html/
```

tai alikansioon, esimerkiksi:

```
[ec2-user wordpress]$ mkdir /var/www/html/blog
```

```
[ec2-user wordpress]$ mv * /var/www/html/blog
```

Tämän jälkeen muutetaan omistajuus ja kirjoitusluvat Apachen dokumenttijuureen, koska jotkin ominaisuudet voivat WordPressissä vaatia sellaiset oikeudet. Apache-

palvelimen käyttäjä on apache, joka nyt liitetään harjoitustehtävässä 2 tehtyyn www-ryhmään.

```
[ec2-user wordpress]$ sudo usermod -a -G www apache
```

Sitten muutetaan ryhmän omistajuus.

```
[ec2-user wordpress]$ sudo chgrp -R www /var/www
```

Sitten muutetaan polun hakemistojen ja alihakemistojen kirjoitusluvat.

```
[ec2-user wordpress]$ sudo chmod 2775 /var/www
```

```
[ec2-user wordpress]$ find /var/www -type d -exec sudo chmod 2775 {} +
```

Sitten rekursiivisesti muutetaan polun hakemistojen ja alihakemistojen kirjoitusluvat.

```
[ec2-user wordpress]$ find /var/www -type f -exec sudo chmod 0664 {} +
```

Tämän jälkeen Apache-palvelin käynnistetään uudelleen, jotta tehdyt muutokset tulevat voimaan.

```
[ec2-user wordpress]$ sudo service httpd restart
```

Nyt voidaan mennä selaimella instanssin julkiseen IP- tai DNS-osoitteeseen eli esimerkiksi <http://julkinendns>, tai jos teit alikansion, niin esimerkiksi <http://julkinendns/blog>. Ruutuun pitäisi ilmestyä WordPressin asennusruutu, jossa pyydetään antamaan blogille nimi ja keksimään käyttäjätunnus ja salasana sekä antamaan sähköpostiosoite. Kun nämä on annettu, on WordPress-blogin asennus valmis. Voit testata blogia kirjoittamalla sinne raportin kaikista tehdyistä harjoitustehtävistä.