

Johan Lindholm-Ventola

POWERSHELLIN HYÖDYNTÄMINEN OFFICE 365-  
YMPÄRISTÖSSÄ

Tietojenkäsittelyn koulutusohjelma  
2019

## POWERSHELLIN HYÖDYNTÄMINEN OFFICE 365-YMPÄRISTÖSSÄ

Lindholm-Ventola, Johan  
Satakunnan ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Elokuu 2019  
Ohjaaja: Grönholm, Jukka  
Sivumäärä: 33

Asiasanat: PowerShell, Office 365, ADFS, skriptaus

---

Tässä opinnäytetyössä tutkittiin PowerShell-ohjelman hyödyntämistä Office 365-ympäristössä. PowerShell osiossa käydään läpi historia, ominaisuudet, käyttölogiikka, komennot sekä skriptaaminen. Office 365-käyttöliittymän graafisella puolella ei pysty tekemään kaikkia toimintoja, joten tarvitaan PowerShell-työkalu mikä tekee nämä asiat. Office 365-palvelu sijaitsee pilvipalvelussa, joten ADFS-palvelu on välttämätön palvelu, kun halutaan SSO-kirjautuminen toimivaksi.

Teoriaosuudessa käydään läpi järjestelmän hallintaa graafiselta- ja komentopohjalta. Järjestelmänhallinnassa käydään historiaa läpi, mistä käyttöliittymät ovat saaneet alkunsa ja sekä niiden tulevaisuus.

Opinnäytetyöprojektin lopuksi on oma tuotokseni, jossa käsitellään PowerShell skriptiä, jonka tein poistamaan pois otettujen Office 365 käyttäjien lisenssit.

# UNILIZING POWERSHELL IN THE OFFICE 365 ENVIRONMENT

Lindholm-Ventola, Johan

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Business Information Technology

August 2019

Supervisor: Grönholm, Jukka

Number of pages: 33

Keywords: PowerShell, Office 365, ADFS, scripting

---

In this thesis I researched the use of PowerShell in the Office 365 environment. In the PowerShell portion we go through the history, features, logic of use, commands and scripting. You are not able to perform all the actions in the Office 365 graphical interface, so you need the PowerShell tool which makes these things happen. The Office 365 service is located at a cloud service, so an ADFS-service is necessary when you want a functioning SSO-login.

In the theory portion I go through system operating through a graphical interface and the command line. In system operating we go through the history; where were graphical- and command-based interfaces born, and also their future.

In the end of my thesis I have my own production, where I go through a PowerShell script which I made to remove disabled user licenses.

# SISÄLLYS

TERMIT JA LYHENTEET .....	5
1 JOHDANTO.....	7
2 JÄRJESTELMIEN HALLINTA.....	8
2.1 Komentorivi .....	8
2.1.1 Tulevaisuus.....	9
2.2 Graafinen käyttöliittymä .....	11
2.2.1 Tulevaisuus.....	13
3 WINDOWS POWERSHELL.....	14
3.1 Historia.....	14
3.2 Versiohistoria.....	15
3.3 Komennot ja cmdlet.....	16
3.4 Skriptaaminen .....	17
3.5 Tietoturva.....	18
4 PILVIPALVELUT .....	19
4.1 Pilvipalveluiden lajit .....	20
4.2 Pilvipalveluiden hallinta .....	22
4.3 Office 365 .....	22
4.3.1 Office 365-lisenssit yrityskäytössä.....	22
4.4 ADFS .....	23
4.4.1 Todentamisen hallinta .....	25
5 OFFICE 365 HALLINTA POWERSHELLILLÄ .....	25
5.1 Kirjautuminen Office 365-hallintaan.....	26
5.2 Käyttäjien ja ryhmien hallinta.....	27
6 OFFICE 365 LISENSSIEN POISTON AUTOMATISOINTI KESKI- SATAKUNNAN TERVEYDENHUOLLON KUNTAPIIRILLE.....	27
6.1 Ratkaisu.....	28
6.2 Skriptin ajoittaminen ja jatkokehitys .....	29
7 LOPUKSI .....	30
8 LÄHTEET .....	31

## TERMIT JA LYHENTEET

Kuori – Rajapinta käyttöjärjestelmään.

PowerShell – Microsoftin kehittämä komentorivipohjainen käyttöliittymä.

CLI – Tulee englannin kielestä command-line interface, mikä tarkoittaa komentoliittymää.

Käyttöliittymä – Ohjelmiston, laitteen tai muun tuotteen osa, jonka kautta käyttäjä ohjaa tuotetta.

VM – Tarkoitetaan tässä tutkimuksessa virtuaalikoneita

Active Directory – Tarkoitetaan Microsoftin toimialueen aktiivihakemistoa, minne kuuluvat mm. tietokoneet, käyttäjät, ryhmät ja postituslistat.

Komentotulkki – Komentopohjainen ohjelmisto, jolla ohjataan käyttöjärjestelmiä. Komennot ja ohjelmistot suoritetaan komentotulkissa kirjoittamalla ohjelmiston tai komennon nimen.

IWA – Microsoftin kehittämä todennuspalvelu, joka käyttää Kerberos tai NTLM -tekniikkaa.

Kerberos ja NTLM – Todennukseen käytettyjä protokollia.

MS-DOS-järjestelmä – Vuonna 1981 Microsoftin julkaisema käyttöjärjestelmä, joka perustuu x86-pohjaisiin tietokoneisiin.

Windows Subsystem for Linux – Mahdollistaa Linux-käyttöjärjestelmien käytön etänä Windows 10- ja Windows Server 2019 -käyttöjärjestelmissä.

PARC – Tunnetaan nykyään nimellä Palo Alto Research Center, joka on vuonna 1970 perustettu tutkimus- ja kehitystoimintayritys. Perustettiin alun perin Xerox-yhtiön tutkimusyksikkönä. Xerox on ollut kehittämässä lasertulostimia, lähiverkkoa, graafista käyttöliittymää ja monia muita tietotekniikan laitteita ja sovelluksia.

## 1 JOHDANTO

Tämä työ on tehty **Keski-Satakunnan terveydenhuollon kuntayhtymä (KSTHKY)** toimeksiannosta. KSTHKY on kunta alainen terveydenhuoltoapiiri, joka toimii Harjavan, Kokemäen, Luvian, Nakkilan sekä Eurajoen alueella. Henkilökuntaa on noin 350-400 henkilöä ja tietokoneita melkein saman verran.

Työn tavoitteena oli kerätä ja luoda ohjeistusta PowerShell-komentotulkin hyödyntämiseen Office 365-ympäristössä ja myös aiheeseen kuuluvaa asiaa. PowerShellin juuret saivat alkunsa jo 2000-luvun alussa kun Microsoft oli alkanut kehittää uudenlaista lähestymistapaa komentorivin hallintaan. Ensimmäinen versio julkaistiin vuonna 2006 ja nykyinen vakaa versio (6.2.1) on julkaistu maaliskuussa 2019.

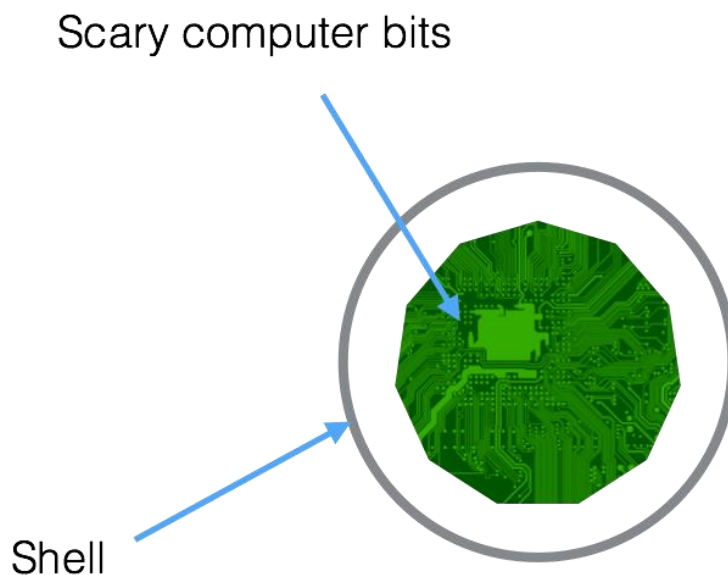
Microsoft Office 365 -palvelu koostuu useista palveluista ja tuotteista, joita hallitaan Office 365-ympäristön webbipohjaisella hallintapaneelilla, sekä PowerShell-komentotulkin avulla. Tutkimustyössäni keskityn PowerShell -komentotulkin käyttöön, skriptaamiseen ja etuihin.

Teoriaosuudessa kerrotaan järjestelmien hallinnasta, mistä graafiset- ja komentorivitulkki ovat saaneet alkunsa, ja mikä niiden tulevaisuus on tai mitkä niiden tulevaisuuden näkymät ovat. Luvussa kolme käsitellään PowerShell -komentotulkin historia, versiot, komentojen ja skriptien tekeminen sekä PowerShellin tietoturvaan liittyen neljän toteutustavan politiikka. Pilvipalveluista käydään läpi niiden lajit ja Office 365-palvelu sekä ADFS -palvelu yleisellä tasolla läpi. Luvussa viisi käsitellään Office 365-palvelun hyödyntämistä PowerShell -komentotulkin avulla.

Viimeisessä luvussa käsitellään PowerShell-skriptiä, jonka tein poistamaan käyttäjien turhat lisenssit Office 365 -palvelusta.

## 2 JÄRJESTELMIEN HALLINTA

Pidämme käyttöliittymiä itsestäänselvyytenä, mutta ne eivät ole aina olleet yhtä edistyneitä. 1960-luvulla varsinaista käyttöliittymää ei ollut. Varhaisiin tietokoneisiin pystyi tallentamaan tiedostoja, jotka pystyivät suorittamaan monia sovelluksia kerralla useille käyttäjille. Tietokoneet muuttuivat monimutkaisemmiksi ja oli välttämätöntä, että ohjelmiston ja laitteiston välillä olisi (kuori). Ohjelmiston ja laitteiston välinen kuori pystyy hallitsemaan vuorovaikutusta toistensa kanssa (Kuva 1). Käyttöjärjestelmän perustyö on aina ollut sama; mahdollistaa sovellusten ja niiden käyttäjien vuorovaikutuksen eri laitteistokomponenttien kanssa, Mukaan lukien kiintolevyn, verkon, näytönohjaimen, emolevyn sekä keskusmuistin. (Computer hope 2018.)



Kuva 1. Vihreäosuus kuvaa tietokoneen komponentteja ja harmaa ympyrä komentotulkkia (kuori). (Gitter 2016).

### 2.1 Komentorivi

Kaikki alkoi Batch-tietotekniikasta. Batch-tietokoneiden käyttöliittymä koostui rei'itetyn kortin tai vastaavan tietovälineen syöttämisestä ja vuorovaikutuksesta reaaliajassa ennenaikaisten tietokoneiden kanssa. Monimutkaisia käyttöliittymiä pidettiin tarpeettomina kustannuksina, koska alkeelliset käyttöliittymät oli suunniteltu hyödyntämään prosessoria mahdollisimman tehokkaasti. (Rounds 2016.)





Kuva 2. IBM 029 kortin rei'ittäjä (Rounds 2016.)

1960-70-luvulla tietokoneista tuli entistä älykkäämpiä. Tietokoneisiin pystyi tallentamaan levyille tiedostoja ja suorittamaan useita sovelluksia joissa oli useita käyttäjiä. Näitä asioita oli vaikea hallita ja tiedostoja pystyi rikkomaan herkästi. Syntyi idea piilottaa näkyvät bitit, mistä tulisi käyttökelpoinen komentotulkki käyttäjille. (Trevorah 2016.)

Komentoriviliittymät vähensivät huomattavasti viivettä päivistä tunteihin tai jopa tunteista sekunteihin, koska käyttöliittymässä oli joukko pyyntö- ja vastausoperaatioita. Pyyntö- ja vastausoperaatiot olivat tärkeitä, koska käyttäjä pystyi muuttamaan mieltään aikaisemmista tapahtumista reaaliaikaisen tiedon perusteella. Seuraava askel oli, että käyttöliittymät saivat videonäyttöpäätteet. Käyttäjä näki syöttämänsä komennot komentorivitulkissa, mistä hän pystyi myös muokkaamaan komentoja ja välttämään kirjoitusvirheet. (Rounds 2016.) Komentotulkeista on kehittynyt käyttöjärjestelmiä mm. Microsoft Windows, Apple Mac OS ja Linux. Ennen, kun nämä komentotulkit saivat graafiset käyttöliittymät, ne olivat pelkästään vain komentotulkkeja. (Trevorah 2016.)

### 2.1.1 Tulevaisuus

Vuosien aikana graafiset käyttöliittymät ovat kehittyneet huimasti, mutta iso osa järjestelmävalvojan töistä tehdään edelleen komentotulkillä. Kehittäjänä tai järjestelmävalvojana työskennellessä komentotulkki on keskeinen työnteon väline, koska graafisella käyttöliittymällä ei ole mahdollista tehdä kaikkea. Tunnetuin komentotulkki on

Microsoftin kehittämä *Command Prompt*. Command Prompt kuuluu MS-DOS-järjestelmän mukana toimitettuihin työkaluihin. Vuosien varrella Command Prompt on liitetty Windowsin PowerShell-konsoliin. Windows ja Windows Server-järjestelmän skriptauskieli on PowerShell, jossa on työkaluja Office 365 -palvelun ja Azure -etähallintaan. (Bisson 2018.)

Windows 10-käyttöjärjestelmään luotiin kolmas komentorivipohjainen ympäristö Windows Subsystem for Linux, joka luotiin Linux-käyttöjärjestelmän ansiosta. Suorittamalla natiiveja Linux-binäärejä Windows-järjestelmässä. Konsoli on tutun Bash -skriptausympäristön host, joka käsittelee Linux-versioita Suseesta, Canonicalista, Debianista ja turvallisuuteen keskittyneestä Kalista. Kaikki tämä johti huomattavaan uudistumiseen Command Prompt-komentokehötteen kohdalla, mikä on yksi vanhimmista Windows-sovelluksista, joka on kehitetty lähes 30 vuotta sitten Windows NT:n alkuperäisestä julkaisusta. (Bisson 2018.)

Nykyaikaiset komentotulkit ovat kehittyneet huomattavasti verrattuna alkuaikoihin. Komentorivit pystyvät näyttämään historian käytetyistä komennoista, jotta voimme nopeasti toistaa komentoja tai poimia ne käytettäväksi tuleviin komentosarjoihin. (Bisson 2018.)

```

rich@richturn02:~$ ll /
total 104
drwxr-xr-x 1 root root 4096 Apr 20 14:29 ./
drwxr-xr-x 1 root root 4096 Apr 20 14:29 ../
drwxr-xr-x 1 root root 4096 Apr 20 15:10 bin/
drwxr-xr-x 1 root root 4096 Sep 22 2017 boot/
drwxr-xr-x 1 root root 4096 Jun 5 17:21 dev/
drwxr-xr-x 1 root root 4096 May 29 15:47 etc/
drwxr-xr-x 1 root root 4096 Apr 18 17:07 home/
drwxr-xr-x 1 root root 104856 Dec 31 1969 init*
drwxr-xr-x 1 root root 4096 Apr 20 15:11 lib/
drwxr-xr-x 1 root root 4096 Apr 18 17:11 lib64/
drwxr-xr-x 1 root root 4096 Sep 22 2017 media/
drwxr-xr-x 1 root root 4096 Apr 20 12:51 mnt/
drwxr-xr-x 1 root root 4096 Sep 22 2017 opt/
dr-xr-xr-x 9 root root 0 Jun 5 17:21 proc/
drwx----- 1 root root 4096 Sep 22 2017 root/
drwxr-xr-x 1 root root - - - - -
drwxr-xr-x 1 root Windows PowerShell
drwxr-xr-x 1 root Copyright (C) Microsoft Corporation. All rights reserved.
dr-xr-xr-x 12 root
drwxr-xr-x 1 root loading personal and system profiles took 1367ms.
drwxr-xr-x 1 root C:\Users\richturn> ls /
drwxr-xr-x 1 root
rich@richturn02:~$

Directory: C:\
Mode                LastWriteTime         Length Name
----                -
d-----          3/8/2018   00:04         c22ce142af63e84db9c490
d-----          6/1/2018   14:54         dev
d-----          5/14/2018  14:58         Intel
d-----          5/16/2018  13:22         Microsoft
d-----          6/5/2018   05:34         Perflogs
d-r-----          6/5/2018  15:31         Program Files
d-r-----          6/5/2018  15:31         Program Files (x86)
d-----          6/1/2018   09:54         temp
d-----          5/22/2018  13:09         tools
d-r-----          6/5/2018  14:33         Users
d-----          6/5/2018  14:43         Windows
d-----          6/5/2018  14:42         Windows.old
d-----          6/5/2018  17:18         WU_logs

C:\Users\richturn> S

Microsoft Windows [Version 10.0.17134.1000]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\richturn\dir c:\
Volume in drive C has no label.
Volume Serial Number is ACEE-120A

Directory of C:\
03/08/2018  01:04 <DIR>          c22ce142af63e84db9c490
06/01/2018  14:54 <DIR>          dev
05/14/2018  14:58 <DIR>          Intel
05/16/2018  13:22 <DIR>          Microsoft
06/05/2018  05:34 <DIR>          Perflogs
06/05/2018  15:31 <DIR>          Program Files
06/05/2018  15:31 <DIR>          Program Files (x86)
06/01/2018  09:54 <DIR>          temp
05/22/2018  13:09 <DIR>          tools
06/05/2018  14:33 <DIR>          Users
06/05/2018  14:43 <DIR>          Windows
06/05/2018  14:42 <DIR>          Windows.old
06/05/2018  17:18 <DIR>          WU_logs
0 File(s)          0 bytes
13 Dir(s)          59,117,498,368 bytes free

C:\Users\richturn>

```

Kuva 3. Command Prompt, PowerShell ja Ubuntu Linux Windows Subsystem for Linux (Bisson 2018.)

Linux-käyttöjärjestelmien hallintaan on kehitetty paljon graafisia työkaluja, mutta hallinta tehdään pääasiassa komentokehoteelta. Graafisen käyttöliittymän ongelma on se, että monet toiminnot ovat monen klikkauksen päässä Toimintoa ei pysty suorittamaan graafiselta käyttöliittymältä. Graafinen käyttöliittymä on helpottanut tietokoneiden hallintaa intuitiivisuutensa takia. Käyttäjien on helpompi käyttää graafista käyttöliittymää kuin opetella komentokehoteen komennot ulkoa. (Levy 2018.)

## 2.2 Graafinen käyttöliittymä

Graafinen käyttöliittymä on käyttöliittymä, joka käyttää kuvakkeita tai muita visuaalisia ilmaisimia vuorovaikutuksessa elektronisten laitteiden kanssa eikä vain tekstipohjaisista komentokehoteista. Nykyään lähes kaikki digitaaliset rajapinnat ovat graafisia käyttöliittymiä. (Computer hope 2018.)

Graafinen käyttöliittymä huomattiin tarpeelliseksi kun tajuttiin, että komentojen suorittaminen tekstipohjaisesti on erittäin hidasta verrattuna kuvakkeiden klikkaamiseen. Komennot kirjoitettiin näppäimistöllä kehoitteeseen, jolloin saatiin vastaus tietokoneelta. Komentojen käyttö ja tarkkan oikeinkirjoituksen tarve loivat raskaan ja tehottoman käyttöliittymän. (Levy 2018.) Ensimmäinen henkilökohtainen tietokone, joka käytti modernista graafista käyttöliittymää oli Xerox Alto vuonna 1973. Xeroxin kehittämä käyttöliittymä ei ollut kaupallinen tuote ja pääasiassa sitä käytettiin yliopistojen tutkimuksissa. Vuonna 1981 Xerox julkaisi ensimmäisen käyttöjärjestelmän jota kutsuttiin täysin integroiduksi pöytäkoneeksi, mukaan lukien sovellukset ja graafinen käyttöliittymä. (Webdesignerdepot [www-sivusto](http://www-sivusto) 2009.)



Kuva 4. Kuvassa Xerox Alto 1973 (Schlosser 2016.)

Xerox kehitti ensimmäisen kaupallisesti saatavan graafisen käyttöliittymän, jota kutsuttiin nimellä ”PARC”. Graafista käyttöliittymää käytettiin Xerox 8010 -käyttäjärjestelmässä. Vuonna 1979 lopussa Applen insinöörin ryhmänjohtaja Steve Jobs näki graafisen käyttöliittymän potentiaalin vierailunsa aikana PARC-tutkimuskeskuksessa. Steve Jobs oli niin vaikuttunut, että hän lähti kehittämään kahteen uuteen Apple tietokoneeseen graafista käyttöliittymää. Käyttöliittymien nimiksi tulivat suunnitteluvaiheessa Lisa ja Macintosh. Jokaisella Applen tuotteella oli bittikartoitettu näyttö ja kämmenen kokoinen hiiri. Ohjelmistojen käyttöliittymä käytti päällekkäisiä ikkunoita näytön sijasta ja Xeroxin työpöydän teemaan sopivia kuvakkeita. Applen insinöörit toivat lisäksi oman innovaationsa, mukaan lukien ”valikkopalkki”, joka hiiren napsautuksella laski alasvetovalikoiman komentoja. Muut lisäykset sisälsivät vierityspalkit, kun ikkunat avattiin ja suljettiin. (Levy 2018.)

Lisa-käyttäjärjestelmä loi laajemmat periaatteet markkinapaikalle vuonna 1984, olemalla matalamman kustannustason Macintosh, joka toi miljoonia muunnoksia käyttäjärjestelmille. Monet kriitikot kuitenkin väittivät, että korkeampien kustannuksien ja hitaampien nopeuksien takia graafinen käyttöliittymä oli sopivampi peruskäyttäjille kuin ammattilaisille. Vuonna 1990 Microsoft julkaisi Windows 3.0 -käyttäjärjestelmän, jossa oli ensimmäinen hyväksyttävä graafinen käyttöjärjestelmä IBM-tietokoneille, jolloin graafisesta käyttöliittymästä muodostui vakio-ominaisuus henkilökohtaisiin tietokoneisiin. Tämä johti erilaisten graafisten rajapintojen kehittämiseen UNIX:lle ja muille käyttäjärjestelmille. Unix-järjestelmien graafisen käyttöliittymän

puute ja korkea hinta tarjosi markkinat Windows -käyttöjärjestelmille. Windows oli helppo asentaa olemassa oleville MS-DOS koneille. Vuonna 1995 Microsoft julkaisi ensimmäisen Windows -käyttöjärjestelmänsä nimellä Windows 95. Windows 95 oli entistä intuitiivisempi sekä visuaalisesti paremman näköinen käyttöjärjestelmä. (Levy 2018.)

### 2.2.1 Tulevaisuus

Computex 2019 tapahtumassa suuret teknologiayritykset kuten Microsoft, Apple ja Google esittelivät uusimpia tuotteitaan ja visioita tietotekniikan tulevaisuudesta. Microsoft esittelee myös tulevaisuuttaan blogipostauksissa, joissa korostetaan Windows 10 -käyttöjärjestelmän ja monien muiden Windows -käyttöjärjestelmien tulevaisuuden etenemistä. Microsoftilla on monia ominaisuuksia, joiden pitäisi olla keskeisiä, mitä kutsutaan ”nykyaikaseksi käyttöjärjestelmäksi”. Yksi hienoimmista modernin käyttöjärjestelmän ominaisuuksista on saumattomat päivitykset, jolloin päivitykset lataavat ja asentuvat automaattisesti taustalla. Tulevaisuudessa ne eivät vaadi käyttäjiltä keskeytyksiä, vaikka aiemmin Microsoftin päivitykset ovat yleensä edellyttäneet järjestelmän uudelleenkäynnistämisen ja mahdollisesti pitkän asennus prosessin.

Windows -maailmassa uutta käyttöjärjestelmää mainostetaan Windows Core OS tai WCOS-nimellä. Se on käyttöjärjestelmä, joka tullaan jakamaan kaikissa uusissa Windows-laitteissa. Windows Core OS on käyttöjärjestelmän moderni versio ja sen tarkoituksena on tehdä Windowsista yleinen alusta kaikille laitteille. Windows Core OS:n ensimmäinen versio on mobiililaitteille ja seuraavat versiot pöytäkoneille sekä Xbox-laitteille. (Rathnam 2019.)

Avoimen lähdekoodin maailman tulevaisuus näyttää positiiviselta, kun jopa Microsoft on tullut mukaan tukemaan avointa lähdekoodia. Avoimesta lähdekoodista on tullut ohjelmistokehityksen yleisin lähestymistapa. Avoimen lähdekoodin valtavan kehityksen keskiössä on kuitenkin yksi iso suuntaus jäänyt huomioimatta: Linux-pohjaisia käyttöjärjestelmiä ei käytetä enää pöytäkoneissa. (Valentin 2017.)

## 3 WINDOWS POWERSHELL

Windows PowerShell on .NET Frameworkin päälle rakennettu komentorivityökalu, jota voidaan laajentaa erilaisilla moduuleilla (Haaranen 2015.) PowerShell tuo ison avun järjestelmänvalvojille ja tehokäyttäjille automatisoimalla nopeammin käyttöjärjestelmiä ja prosesseja hallitsevia tehtäviä. PowerShell-komentojen kautta pystyy hallinnoimaan tietokoneita komentoriviltä. palveluntarjoajan avulla voit käyttää tietokantavarastoja, kuten rekisteri- ja sertifikaattivarastoja, yhtä helposti kuin tiedostojärjestelmässä. PowerShell oli aluksi tarkoitettu pelkästään Windows-pohjaisille järjestelmille, mutta elokuussa 2016 Microsoft avasi avoimen lähdekoodin muille alustoille, kuten esim. Mac OS ja Linux. (Thakoor 2018.)

### 3.1 Historia

Microsoftin tekninen asiantuntija Jeffrey Snover kehitti PowerShellin. Ennen kuin Snover liittyi Microsoftiin vuonna 1999, hän työskenteli Digital Equipment Corp -yhtiössä. Hän tuskaili työskennellessään Windows-järjestelmänhallintatehtävissä, koska Snover koki, että Windowsissa ei ollut resursseja ja joustavuutta automatisoida Unixissa käytettyjä samankaltaisia tehtäviä. Jeffrey Snover aloitti alun perin PowerShell projektin nimellä Monad. (Rouse 2017.)

Ennen PowerShell-työkalua Microsoftin ohjelmistopaketteja hallitsevat järjestelmä- ja verkonalvojat joutuivat käyttämään paljon erilaisia työkaluja, kieliä ja tekniikoita automaatio- ja integrointitehtävien suorittamiseksi. Joitakin tehtäviä varten järjestelmänvalvojat käyttivät Batch-tiedostoja, jotka voitiin suorittaa Command Prompt-komento-kehoteella. Muita tehtäviä varten käytettiin Visual Basic Scripting Editoria. Ylimääräisiä tehtäviä varten oli Windows Scripting Host ja monia muita ohjelmistoja. (Santos 2015.)

Järjestelmänvalvojien piti käyttää luovuutta, koska ei ollut olemassa yhtä kieltä ja työkalua, jota he voisivat käyttää eri Microsoftin ja muiden kuin Microsoft-tuotteiden tehtävien yhdistämiseen. Unix- ja Linux-järjestelmienvälvoijilla oli C-kuori ja luotettava

bash siihen. Microsoft ei onnistunut kehittämään tuohon aikaan tehokasta komentori-  
viä. (Santos 2015.) Vuonna 2005 Microsoft julkaisi kolme testiversiota Monadista.  
Samana vuonna huhtikuussa Monadin nimi muutettiin virallisesti Windows Po-  
werShelliksi ja näin siitä tuli tärkeä osa teknologiaa Microsoftin tarjonnalle. Ensim-  
mäinen versio PowerShellin (v1.0) julkaistiin syksyllä 2006 ja lopullinen julkaisu  
verkkoon tehtiin marraskuussa samana vuonna. Nykyinen vakaa versio v6.2.0 julkais-  
tiin huhtikuussa 2019. (Microsoft 2018a.)

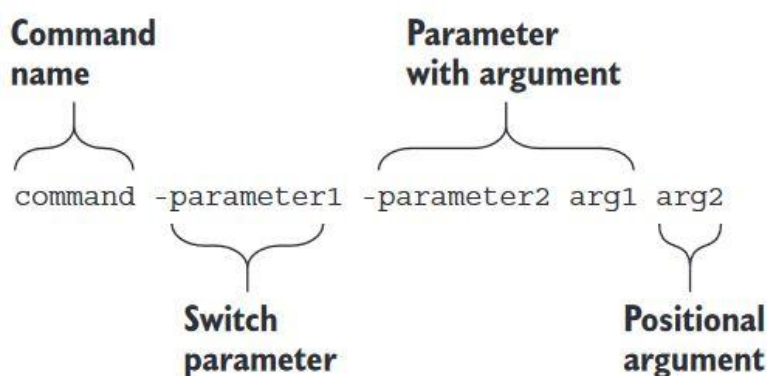
### 3.2 Versiohistoria

Taulukko 1. PowerShell versiohistoria ja julkaisupäivät (Pitroforte n.d.).

PowerShell ver- sio	Julkaisu- päivä	Oletus Windows versio	Saatavilla
PowerShell 1.0	Marraskuu 2006	Windows Server 2008(*)	Windows XP SP2 Windows XP SP3 Windows Server 2003 SP1 Windows Server 2003 SP2 Windows Server 2003 R2 Windows Vista Windows Vista SP2
PowerShell 2.0	Lokakuu 2009	Windows 7 Windows Server 2008 R2 (**)	Windows XP SP3 Windows Server 2003 SP2 Windows Vista SP1 Windows Vista SP2 Windows Server 2008 SP1 Windows Server 2008 SP2
PowerShell 3.0	Syyskuu 2012	Windows 8 Windows Server 2012	Windows 7 SP1 Windows Server 2008 SP2 Windows Server 2008 R2 SP1
PowerShell 4.0	Lokakuu 2013	Windows 8.1 Windows Server 2012 R2	Windows 7 SP1 Windows Server 2008 R2 SP1 Windows Server 2012
PowerShell 5.0	Helmikuu 2016	Windows 10	Windows 7 SP1 Windows 8.1 Windows Server 2012 Windows Server 2012 R2
PowerShell 6.0	Tammikuu 2017	Windows 10 Anniver- sary Update Windows Server 2016	Windows 7 SP1 Windows 8.1 Windows Server 2008 R2 SP1 Windows Server 2012 Windows Server 2012 R2
* Asennettava palvelinhallinnan kautta			
** Integroitu kaikkiin myöhempisiin Windows-versioihin			

### 3.3 Komennot ja cmdlet

Komennot ovat olennainen osa mitä tahansa komentotulkkia; niillä asiat saadaan toimimaan. Suuri osa PowerShellissä käytettävästä terminologiasta on tuttu Linux- tai Windows-käyttöliittymästä. Komento alkaa komennon nimellä, jota seuraa parametrit (Kuva 5). Parametrit voivat olla kytkin parametrejä (Switch parameter), jotka eivät ota vastaan mitään argumentteja tai (Parameter with argument), jotka ottavat argumentteja vastaan. PowerShellin sijaintiparametrit (Positional argument), joissa vastaava parametri johtuu argumentin asemasta komentorivillä. (Payette B, Siddaway R. 2018, 18.)



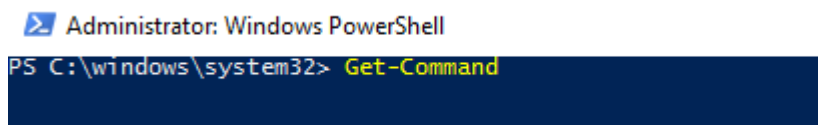
Kuva 5. PowerShell-komennon toimintapa. (Payette B, Siddaway R. 2018, 18.)

Kaikki komennot jaetaan komennon nimien mukaan, komennolle määritettyihin parametreihin ja näiden parametrien argumentteihin. Parametrejä voidaan ajatella vastaanottajan informaatioksi ja argumenttiä informaatioksi. (Payette B, Siddaway R. 2018, 18.) PowerShell käyttää verb-substantiiviparia cmdlet-nimien ja niiden kautta johdettujen Microsoft .NET Framework-luokkien nimissä. Esimerkiksi PowerShellin toimitamaa Get-Command -toimintoa käytetään haettaessa kaikkia PowerShelliin rekisteröityjä komentoja. Verbiosa tunnistaa cmdlet-toiminnon ja substantiiviosa tunnistaa kokonaisuuden, jolle toiminto suoritetaan. (Microsoft 2018b.)

Komennon ensimmäinen elementti on suoritettavan komennon nimi. PowerShell-komentotulkki tarkastelee nimeä ja määrittelee, mikä komento suoritetaan ja minkälaista komentoa suoritetaan. PowerShellissä on useita komentokategorioita; cmdlet, shell-funktiokomennot, komentosarjan komennot ja natiivit Windows-komennot. (Payette



B, Siddaway R. 2018, 18.) PowerShellistä löytyy todella runsaasti komentoja. *Get-Command* –komennon avulla voidaan listata käytettävät komennot.

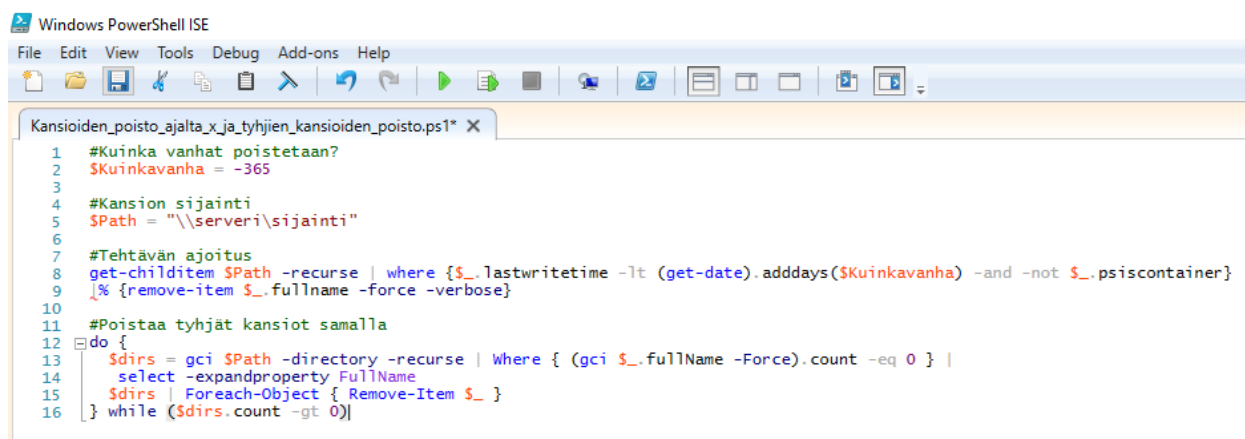


Kuva 6. Get-Command.

### 3.4 Skriptaaminen

Skriptien luominen ja suorittaminen onnistuu Windows PowerShellin ja Windows PowerShell ISE:n avulla. Tiedostoja voi muokata PowerShell ISE-ohjelmalla ja Skriptin voi avata komentotulkilla. PowerShell luo automaattisesti skriptitiedostopäätteellä (.ps1), Skriptin datatiedoston päätteellä (.psd1) ja Skriptin moduulitiedoston päätteellä (.psm1). Tiedostotyypit ovat syntaksiltaan värillisiä komentosarjan muokkausohjelmassa. Muut yleiset tiedostotyypit jotka voi avata komentosarjan ruudussa ovat kokoonpanotiedostot (ps1xml), XML-tiedostot sekä tekstitiedostot. (Aiello & Wheeler 2018.)

Skriptejä voidaan ajoittaa haluttuun aikaan *tehtävien ajoituksen* kautta. Skriptejä voidaan tehdä monenlaisia automatisointiin mm. varmuuskopiointeja ja tiedostojen poistamista tai synkronoimista. Alla olevassa kuvassa (Kuva 6.) on skripti, joka poistaa vanhat varmuuskopiot kun varmuuskopion ottamisesta on kulunut 365 päivää.



Kuva 7. Kansioden poistoskripti.

### 3.5 Tietoturva

Suorituspolitiikka on osa PowerShellin turvallisuusstrategiaa. Suorituskäytännöt määrittelevät sen, että voitko ladata määritystiedostoja, kuten PowerShell-profiilia tai suorittaa komentosarjoja, ja että onko skriptit allekirjoitettava digitaalisesti ennen niiden suorittamista. (Morgan 2018.)

Toteutuspolitiikkaa määrittelee *Set-ExecutionPolicy*-komento, jonka oletusalue on paikallinen tietokone, joka vaikuttaa kaikkiin tietokoneessa oleviin käyttäjiin. Toteutuspolitiikkaa vaihtaessa PowerShell pitää käynnistää käyttäen järjestelmänvalvojan tunnuksia. (Microsoft 2018c.)

- Restricted – rajoitettu
- AllSigned – kaikki allekirjoitettu
- RemoteSigned – etäohjattu
- Unrestricted and Bypass – rajoittamaton/ohitettu

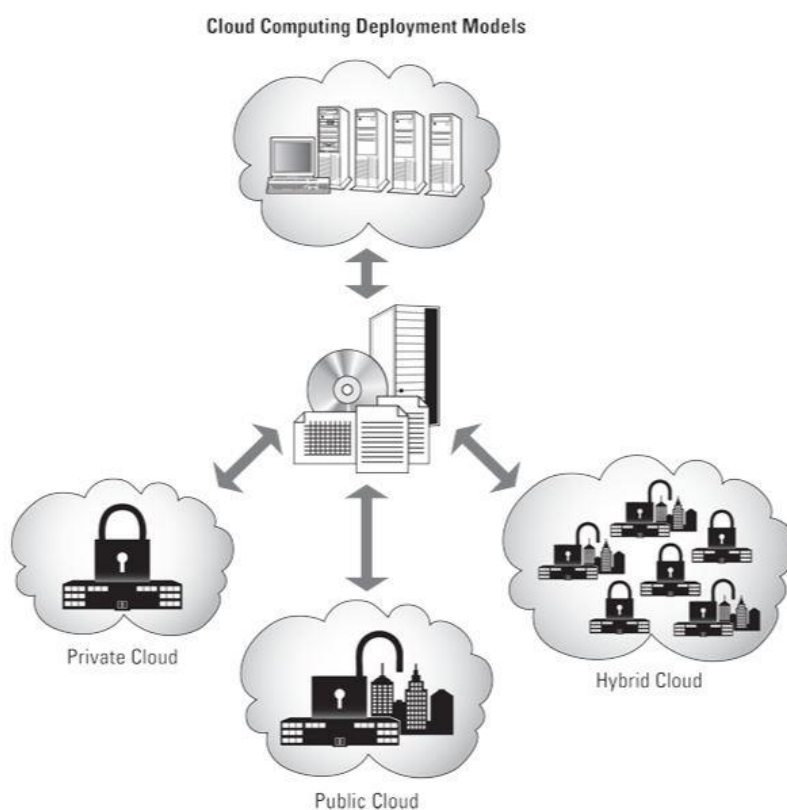
PowerShellillä on neljä tasoa toteutuspolitiikassa. Jos skripti halutaan suorittaa onnistuneesti tulee PowerShell-suorituskäytäntö asettaa sopivaan tasoon. (Morgan 2018.)

Windows Server 2012 R2:n ja Windows Server 2016:n oletusarvo on etäohjattu, Kun taas muut Windows-käyttöjärjestelmät ovat rajoitettuja etähallinnan osalta, jonka takia skriptit eivät toimi näissä järjestelmissä. Etähallinta estää muualta ladattujen ja kiellettyjen (allekirjoittamattomien) komentosarjojen käytön. Komentosarja hyväksytään luotettavaksi, kun se on kirjoitettu ja tallennettu verkkotunnukselle. Kaikki allekirjoitettumat kieltää kaikki komentosarjat, joilla ei ole luotettavaa allekirjoitusta. Tämä on mahdollisesti turvallisin taso, mutta luo eniten hallinnollisia yleiskustannuksia, koska jokainen skripti on allekirjoitettava aina, kun komentosarjaan tehdään muutos. Rajoittamattomat- ja ohitustilat eivät vaadi allekirjoituksia skripteissä, joita halutaan Suorittaa. Tämä on vaarallisin tila, koska kaikki skriptit suoritetaan ilman minkäänlaista tarkistusta. Suorituspolitiikan taso voidaan asettaa joko *Set-ExecutionPolicy*-komennolla tai käyttämällä ryhmäkäytäntöasetuksia, jotka vaikuttavat suureen määrään tietokoneita. (Morgan 2018.)

## 4 PILVIPALVELUT

Pilvi on metafora internetille. Yksinkertaisesti selitettynä tai yksinkertaisin sanoin pilvipalvelu tarkoittaa sitä, että Sovellukset, ohjelmistot, informaatio sekä tietotekniset tarpeet ovat käytettävissä sekä tallennettavissa ja tämä kaikki tapahtuu internetin välityksellä pilvessä. Facebook-tili, online-pelit tai tiedostonjakopaikka esim. Dropbox ovat kaikki toimivat pilvessä. Niitä käytetään informaation tallentamiseen, jota voidaan käyttää ja muuttaa internetin välityksellä.

Pilvipalveluista on kolme eri mallia; yksityinen pilvi, julkinen pilvi ja hybridipilvi (Kuva 7). Yksityinen pilvi on tyypillisesti omistettu yhdelle organisaatiolle yksistymällä oman pilven. Yksityinen pilvi ei jaa mitään tietoja. Yksityinen pilvi on yleensä organisaation välillä oleva jaettu kansio, minne vain pääsee organisaation sisäverkossa. Hybridipilvi on yhdistelmä julkista ja yksityistä pilveä. Esimerkiksi organisaatio voi käyttää sähköpostisovellustaan julkisessa pilvessä, mutta tallentaa asiakastiedot yksityisen pilven datakantaan.



Kuva 8. Yksityinen pilvi, julkinen pilvi ja hybridipilvi.

Yleinen uskomus on, että pilvipalvelu olisi uusi käsite, mutta näin ei ole. Termi ”Intergalactic Computer Network” kuultiin ensimmäisen kerran 1960-luvulla J.C.R Licklideriltä, yhdeltä tietotekniikan historian(?) kannalta tärkeimmältä ihmiseltä. Toiset ihmiset kreditoivat pilvipalvelujen syntymisen toiselle tietotekniikan tutkijalle: John McCarthyille. McCarthy ehdotti 1960-luvulla, että tietotekniikka toimitetaan yleishyödykkeeksi palvelukeskuksille, jotka tarjosivat palveluja yrityksille maksua vastaan.

Pilvipalvelut tarjoavat monia IT-palveluita, kuten sovellus- ja tietokantapalvelimen, sisäisen tallennuksen tai varmuuskopioinnin korvaamisen pilvivarastolla ja ohjelmistojen käyttämisen suoraan web-selaimesta ilman asennusta, esim. Microsoftin kehittämä Word Online. Näitä palveluita yleensä ylläpitää yrityksissä omat henkilöt. (Withee R, Withee K, Reed, J. 2016, 20-22 Microsoft Office 365 for dummies.)

#### 4.1 Pilvipalveluiden lajit



Kuva 9. Pyramidin huipulla SaaS(ohjelmisto palveluna), keskellä PaaS(Alusta palveluna) ja alimpana IaaS(infrastruktuuri palveluna).

#### **Ohjelmisto palveluna (SaaS)**

Tulee englannin kielen sanoista Software as a Service. Saas-palvelulla tarkoitetaan ohjelmistoa, joka sijaitsee pilvessä ja jota ylläpitää palveluntarjoaja. Saas-palvelut Tarjotaan yleensä verkkoselaimen kautta, sovelluksena tai näiden yhdistelmänä. Verkkoselain on suosituin näistä kolmesta vaihtoehdosta. Saas-palvelu on vuokrattava ohjelmisto, jolloin sen käyttäjä ei omista sitä. Saas-palvelussa hinnoittelu perustuu yleensä käytön määrän ja käyttöajan mukaan. Tavallisimpana hinnoitteluna voidaan pitää kuukausilaskutusta. (Pilvi www-sivusto n.d.)

### **Infrastrukturi palveluna (IaaS)**

IaaS-palveluntarjoaja tarjoaa infrastruktuurin palveluun asiakkaalleen tyypillisesti niin, että asiakkaan käyttöön tarjotaan web-pohjainen hallintaliittymä, jonka kautta hän voi itse perustaa tarvittaessa palvelimia sekä hallita palvelimen muistia, verkko-yhteyksiä ja palomureja. IaaS-palvelut ovat erityisesti ovat erityisesti hyviä organisaatioille, joissa on oma IT-osasto tai tarvittavaa osaamista muuta kautta. Palvelun käyttäjä huolehtii itse palvelimien tietoturvasta, palomureista, ohjelmistoista ja asennuksista. (Eronen 2016.)

### **Alusta palveluna (PaaS)**

Tulee englannin kielen sanoista Platform as Service. PaaS-palvelulla tarkoitetaan, että palveluntuottaja tarjoaa palveluita Sovelluslustoina, jotka ovat pakattu helposti käyttöön otettavaan muotoon. Pilvipalvelut tarjoavat suurta joustavuutta resurssien, päällekkäisyyksien ja skaalautuvuuden suhteen käyttäjien vaatimusten mukaan. Sovelluslustoja tarjotaan yleisesti ohjelmistokehittäjien käyttöön tai tarpeisiin. Palvelun käyttäjä voi tilata sopivan alustan ja halutessaan maksaa siitä joustavasti käyttöön perustuen sekä siirtää oman sovelluksensa palveluun. PaaS-alustoissa on tyypillisesti web-käyttöliittymän lisäksi muitakin tapoja muodostaa yhteyksiä palveluihin esim. FTP/SFTP, SSH, komentorivityökalut (CLI) sekä API-rajapinta, minkä kautta voidaan automatisoida toimia ja vaikkapa tuoda palvelut osaksi omaa sovellusta tai itsepalvelumallia. Yksinkertaisuudessaan PaaS-järjestelmä voi olla sellainen, että teet itse julkaisujärjestelmän (esim. WordPress tai Magento) avulla verkkosivut ja huolehdit niihin liittyvistä päivityksistä ja tietoturvasta. Fyysisistä palvelimista huolehtii isäntäpalvelu. (Eronen 2016.)

## 4.2 Pilvipalveluiden hallinta








Pilvipalveluiden hallintaan tarvitaan omat työkalunsa. Pilvipalveluiden tarjoamat julkiset työkalut ovat erikoistuneet mm. organisointiin, kustannuksien hallintaan, turvallisuuteen ja pilvipalveluiden ominaisuuksiin. Esimerkiksi Amazon Web Services (AWS) antaa käyttäjälle mahdollisuuden käyttää ja hallita pilvi-instansseja komentorivityökalulla, joka käyttää yksittäisiä komentoja ja komentosarjoja. Google Cloud Platform (GCP) tarjoaa seuranta -ja lokityökalun, Google tarjoaa Stackdriver joka tarjoaa suorituskykytietoja sovelluksista ja virtuaalikoneista, jotka toimivat GCP- ja AWS-käyttöliittymillä. Microsoftin Azure tarjoaa myös PowerShell -moduulin, jolla voidaan hallita Azuressa olevia virtuaalikoneita, ohjelmistoja tai muita palveluita. (Rouse 2017.)

## 4.3 Office 365

Microsoftin valmistama Office 365 on yksi maailman käytetyimmistä toimistosovelluksista. Office 365 on tilauspalvelu, jonka avulla käyttäjät voivat ladata ja asentaa Microsoft Office -sovelluksia käytettäväksi Windows ja OS X-käyttöjärjestelmissä. Microsoft Office 365:n ensimmäinen versio julkaistiin kesäkuussa 2011. Office 365 -ohjelmistopaketti oli alun perin tarkoitettu yrityskäyttäjille, mutta ajansaatossa se on laajentunut myös yksityiskäyttöön. Microsoft tarjoaa Office 365:lle ohjelmistoja palvelualustana, joka sisältää Office-sovellukset sekä SharePointin, Exchangen, Lyncin ja Officen web-sovellukset. Microsoft tarjoaa Office-sovellukset myös pilvessä ja Office 365 -sovellukset voidaan asentaa Windows, Android ja iOS -puhelimille

### 4.3.1 Office 365-lisenssit yrityskäytössä

Office 365-lisenssin valitseminen on ratkaiseva vaihe ennen siirtymistä pilveen. Pienillä, keskisuurilla ja isoilla yrityksillä on useiden vaihtoehtojen myötä omat tuotteensa. Office 365 -lisenssien vertailun avulla voidaan tehdä oikea lisenssivalinta käytettävissä olevista vaihtoehdoista. Lisenssit maksetaan käyttäjä per kuukausi hintaan, mistä tulee vuositulo.

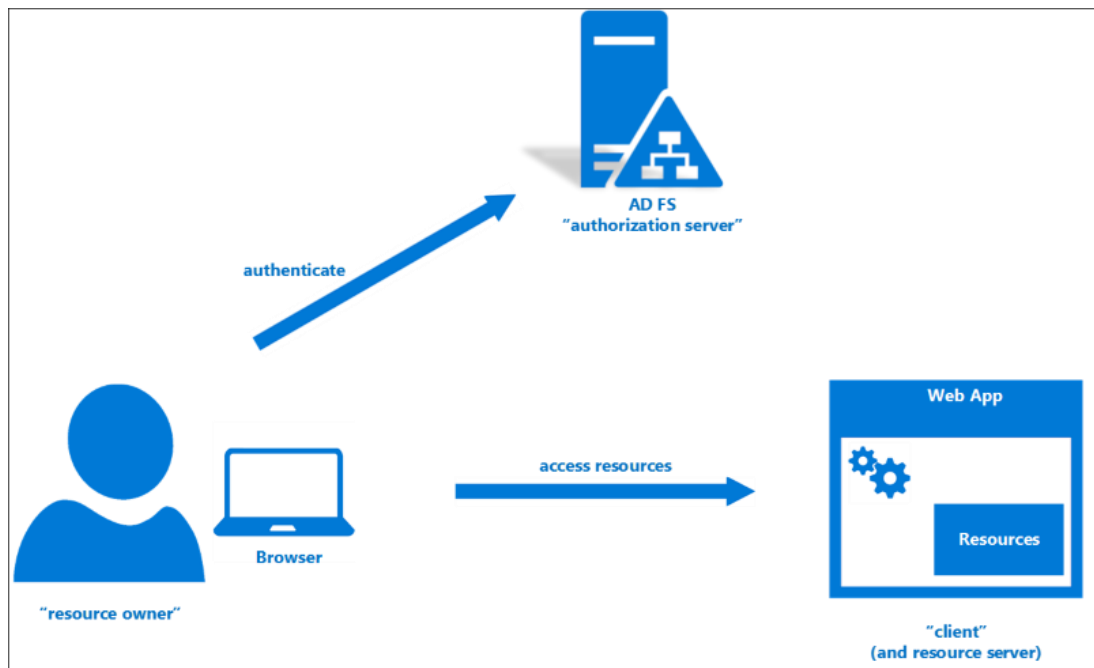
<p>12,80 € käyttäjä/kuukausi (vuositilaus)</p> <p>Office 365 ProPlus</p> <p><b>Osta nyt</b></p> <p>Lue lisää  ALV ei sisälly hintaan.</p> <p>Office-sovellukset, tiedostojen pilvitallennustila ja jakaminen. Yritystason sähköposti ei sisälly pakettiin.</p> <p><b>Sisältyvät Office-sovellukset</b></p> 	<p>6,70 € käyttäjä/kuukausi (vuositilaus)</p> <p>Office 365 E1</p> <p><b>Osta nyt</b></p> <p>Lue lisää  ALV ei sisälly hintaan.</p> <p>Yrityspalvelut: sähköposti, tiedostojen tallennustila ja jakaminen, Office Online, kokoukset ja pikaviestintä ja paljon muuta. Office-sovellukset eivät sisälly.</p> <p><b>Office-sovellukset</b></p> <p>(Eivät sisälly)</p>	<p>19,70 € käyttäjä/kuukausi (vuositilaus)</p> <p>Office 365 E3</p> <p><b>Osta nyt</b></p> <p>Lue lisää  ALV ei sisälly hintaan.</p> <p>Kaikki ProPlusin ja Office 365 E1:n ominaisuudet sekä tietoturvyökalut ja velvoitteiden noudattamisen työkalut, kuten säilyttämismääräykset ja tietojen menetyksen estäminen.</p> <p><b>Sisältyvät Office-sovellukset</b></p> 	<p>34,40 € käyttäjä/kuukausi (vuositilaus)</p> <p>Office 365 E5</p> <p><b>Ota yhteyttä myyntiin</b></p> <p>Lue lisää  ALV ei sisälly hintaan.</p> <p>Kaikki Office 365 E3:n ominaisuudet ja laajat tietoturva-, analytiikka- ja ääniominaisuudet.</p> <p><b>Sisältyvät Office-sovellukset</b></p> 
--	--	--	---

Kuva 10. Office 365 -palvelun yrityspaketteja.

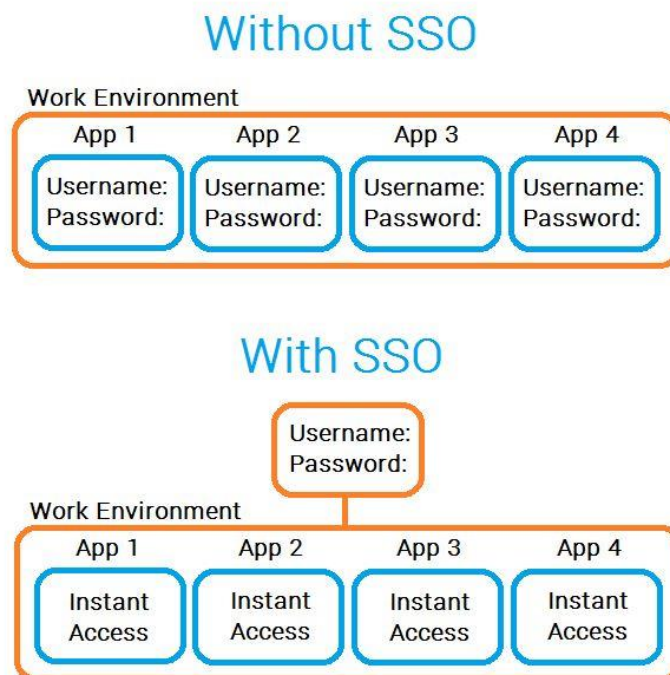
#### 4.4 ADFS

Tulee nimestä Active Directory Federation Services, joka on Microsoftin luoma kertakirjautuminen (SSO) -ratkaisu. Windows Server -käyttöjärjestelmissä se tarjoaa käyttäjille todennetun pääsyn sovelluksiin, jotka eivät pysty käyttämään integroitua Windows-todentamista (IWA) Active Directoryn kautta. ADFS antaa organisaatioille mahdollisuuden hallita käyttäjien sovelluskirjautumisia yhdestä paikkaa.

ADFS-todennusta hallinnoidaan proxy-palvelun kautta Active Directoryn ja hallitun sovelluksen välillä. ADFS käyttää Federated todennusta, joka on linkitetty ADFS:n ja valitun sovelluksen välille, mikä antaa käyttäjälle luotetun pääsyn valittuun sovellukseen (Kuva 10). Käyttäjät voivat kirjautua yhdistettyyn sovellukseen SSO:n kautta samoilla salasanoina mitkä heillä on (Kuva 11). (Bouchard 2018.)



Kuva 11. ADFS-palvelun todentaminen. (Microsoft 2017.)



Kuva 12. ADFS SSO -palvelun kertakirjautumisperiaate. (Bouchard 2018.)

ADFS:n tarve tuli ilmi, kun haluttiin kirjautumisten todentamiseen helpompi ratkaisu. Active Directory ja IWA ovat asettaneet rajoituksia nykyaikaisen todentami-



sen osalta ja ne eivät voi todentaa käyttäjiä, jotka käyttävät Active Directoryn integroitua sovelluksia ulkopuolisesti. ADFS ratkaisee ongelman käyttäjiltä, jotka tarvitsevat Active Directory integroitua sovelluksia, kun käyttäjät työskentelevät etänä tarjoamalla joustavan ratkaisun, jonka avulla he voivat todentaa käyttämällä tavallisia organisaation AD-valtuuksiaan Web-käyttöliittymän kautta. ADFS:n avulla käyttäjät pystytään todentamaan verkkotunnuksilla toisen osapuolen sovelluksessa (Shyamsundar 2018.)

#### 4.4.1 Todentamisen hallinta

Kuvittele, että olet lähdessä matkalle Espanjaan. Lentokentällä sinun tietosi tarkistetaan. Turvatarkastuksessa oletetaan tai odotetaan, että näytät passisi, koska lentokentän turvallisuushenkilö ei tunne sinua. He eivät voi luottaa sinuun, jos sinulla ei ole passia mukana tai passi ei ole enää voimassa. Luotettavuus tulee vasta, kun passi on mukana ja se on voimassa. Asiat toimivat samalla periaatteella ADFS-palvelussa. Lopukäyttäjää ei tunnisteta suoraan palveluntarjoajan kanssa, vaan sen kanssa keneltä on saanut todennuksen. Käyttäjän tunnistetiedot tallennetaan aina vain organisaatioon identiteetin tarjoajan kanssa, joka on yleensä Active Directory. Käyttäjä kirjautuu sovellukseen tai palveluun ja palveluntarjoaja luottaa henkilöllisyystoimittajan vahvistamiin käyttöoikeuksiin. Käyttäjä ei koskaan anna valtuuksia suoraan palveluntarjoajalle, vaan identiteetin tarjoajalle. (Bouchard 2018.)

## 5 OFFICE 365 HALLINTA POWERSHELLILLÄ

Office 365:n PowerShell-komentoja hyödyntämällä saa mahdollisesti ison hyödyn automaatioon ja komentosarjojen nopeuteen sekä lisäominaisuuksia palveluun. Office 365:n hallinnointi PowerShellillä voi paljastaa lisätietoja, joita ei näe Office 365:n järjestelmävalvojan hallintapaneelissa. Office 365:sta löytyy myös ominaisuuksia, joita voidaan määritellä vain Office 365 PowerShell järjestelmänvalvojan hallintapaneelissa. PowerShell on myös hyvä tietojen suodattamiseen ja erilaisten irtotoimintojen suorittamiseen. PowerShellin avulla pystyy myös hallitsemaan eri palvelintuotteita.

## 5.1 Kirjautuminen Office 365-hallintaan

Tässä työssä käytän yhteyden muodostamiseen Microsoftin Azure Active Directory -moduulia. Ennen moduulin asennusta pitää asentaa Azure Active Directoryyn Graph-moduuli. Graph-moduulin asennus onnistuu PowerShellin komentoriviltä järjestelmävalvojan tunnuksilla komennolla:

*Install-Module -Name AzureAD*

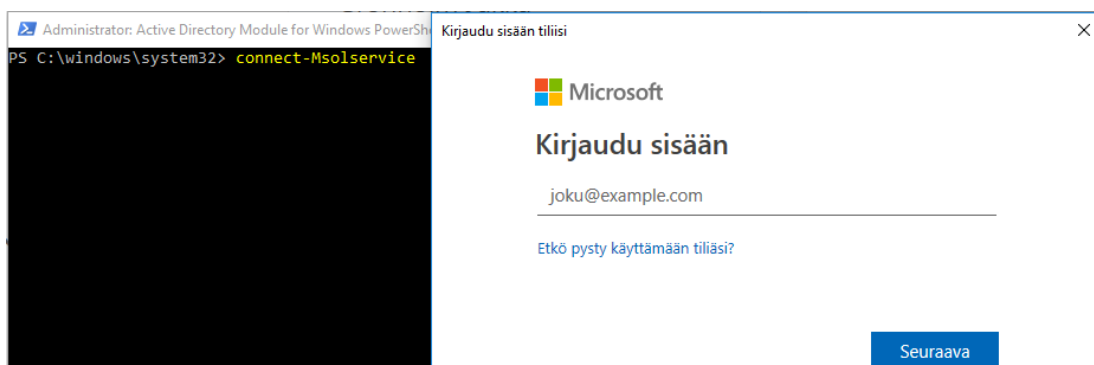
Kun Azure Active Directoryyn Graph-moduuli on asennettu, seuraavaksi asennetaan Active Directory Module for Windows PowerShell. Asennuksen jälkeen PowerShellin ja PowerShell ISE:n lisäksi ilmestyy Active Directory Module for Windows PowerShell -kuvake. Avataan yhteys Active Directory moduuliin järjestelmävalvojan tunnuksilla (Kuva 9).



Kuva 13. Active Directory Module for Windows PowerShell.

Seuraavaksi muodostetaan yhteys Office 365 –ympäristöön. Komennon jälkeen Active Directory -moduuli pyytää kirjautumistietoja.

*Connect-MsolService*



Kuva 14. Pyytää kirjautumistietoja Office 365-palveluun.

## 5.2 Käyttäjien ja ryhmien hallinta

Käyttäjien ja ryhmien hallinta onnistuu kätevästi PowerShellin avulla. PowerShell-moduulien avulla voidaan suorittaa kaikki samat toiminnot kuin Office 365:n graafisen käyttöliittymän puolella. Käyttäjien ja ryhmien massahallinta onnistuu parhaiten PowerShellin avulla, mutta niiden yksittäinen luominen ja hallinta on helpompaa Office 365:n graafisessa käyttöliittymässä. Käyttäjien massahallinta PowerShellillä onnistuu CSV-tiedostolla. CSV-tiedostoon laitetaan käyttäjän tiedot kuten etunimi, sukunimi, käyttäjänimi ja lisenssi. Käyttäjät voidaan listata ryhmien mukaan esimerkiksi komennolla `Get-MsolUser`

Käyttäjille voidaan asettaa salasanaikäytäntö, jossa määritellään milloin salasana menee vanhaksi, tai mahdollisesti että salasana ei vanhennu ollenkaan. On suositeltavaa, että salasana olisi vaihdettava 60 päivän välein, ja että se olisi vähintään 8 merkkiä sisältäen isoja kirjaimia, erikoismerkkejä sekä numeroita. Seuraavalla komennolla voidaan määrittää yksittäiselle henkilölle, että salasana ei vanhene koskaan: *Set-MsolUser -käyttäjänimi PasswordNeverExpires \$true*.

## 6 OFFICE 365 LISENSSIEN POISTON AUTOMATISOINTI KESKI-SATAKUNNAN TERVEYDENHUOLLON KUNTAPIIRILLE.

Ongelmana on Office 365:n ja Active Directoryn synkronointi. Tällä hetkellä Active Directoryn ja Office 365:n sähköpostisynkronointi tapahtuu Active Directorysta löytyvän sähköpostikaavakkeen avulla (Kuva 14). Kun sähköpostikaavakkeeseen laitetaan etunimi.sukunimi@ksthky.fi niin ADFS-palvelin synkronoi käyttäjän tiedot joka puolen tunnin välein Office 365-pilveen. Palvelussa käytetään SSO-kirjautumista, eli tietokoneelle ja sähköpostiin kirjautuessa käytetään samoja tunnuksia.

Johan Lindholm-Ventola Properties

Published Certificates | Member Of | Password Replication | Object  
 Security | Environment | Sessions  
 Remote control | Remote Desktop Services Profile | COM+  
 General | Address | Account | Profile | Telephones | Organization

Johan Lindholm-Ventola

First name:  Initials:

Last name:

Display name:

Description:

Office:

Telephone number:

E-mail:

Web page:

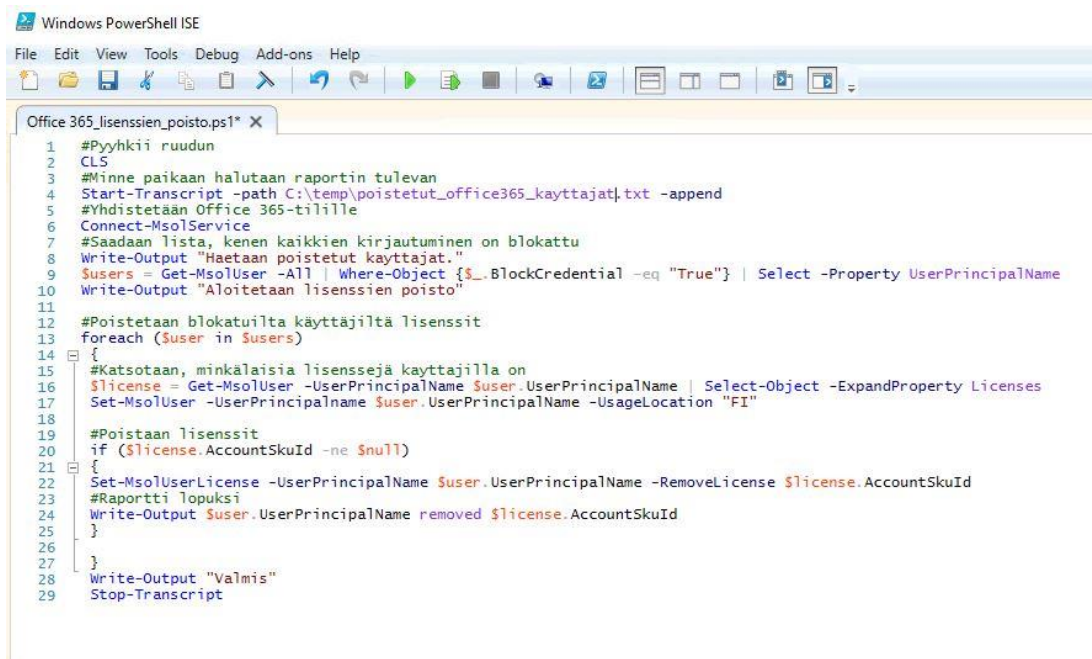
Kuva 15. Kuva Active Directoryn henkilön tiedoista. Keskellä kuvaa olevalla nuolella tarkoitetaan, mistä sähköposti synkronoiminen Office 365-pilveen tapahtuu.

Kun käyttäjän tili otetaan pois käytöstä Active Directoryn kautta niin se deaktivoituu myös Office 365 -ympäristössä, mutta ongelmaksi muodostuu Office 365 -lisenssi joka jää voimaan, vaikka käyttäjä on otettu pois käytöstä.

## 6.1 Ratkaisu

Ratkaisu on skripti, joka synkronoi Office 365 -pilvessä pois käytöstä olevat käyttäjät ja ottaa heiltä samalla tuotteiden lisenssit pois käytöstä. Skripti ajoitetaan suoritettavaksi joka kuukauden ensimmäinen päivä, jolloin se käy läpi Office 365 -palvelusta poistetut käyttäjät.

Tutkiessani internetistä löytyisikö apua skriptiin yllättäen huomasin, että monella muulla organisaatiolla on ollut sama ongelma. Skriptejä löytyi monenlaisia, mutta löysin hyvän pohjan mitä aloin itse muokkaamaan oman mallini mukaan. PowerShell-skripti yhdistää ensin Office 365 -palveluun, minkä jälkeen haetaan käytöstä poistetut käyttäjät. Tämän jälkeen skripti aloittaa lisenssien poiston poistetuilta käyttäjiltä ja lopuksi tulostaa raportin haluttuun paikkaan txt-tiedostona.



```

Office 365_lisenssien_poisto.ps1* X
1 #Pyyhkii ruudun
2 CLS
3 #Minne paikkaan halutaan raportin tulevan
4 Start-Transcript -path C:\temp\poistetut_office365_kayttajat.txt -append
5 #Yhdistetään Office 365-tilille
6 Connect-MsolService
7 #Saadaan lista, kenen kaikkien kirjautuminen on blokkattu
8 Write-Output "Haetaan poistetut kayttajat."
9 $users = Get-MsolUser -All | Where-Object {$_.BlockCredential -eq "True"} | Select -Property UserPrincipalName
10 Write-Output "Aloitetaan lisenssien poisto"
11
12 #Poistetaan blokatuilta käyttäjiltä lisenssit
13 foreach ($user in $users)
14 {
15     #Katsotaan, minkälaisia lisenssejä kayttajilla on
16     $license = Get-MsolUser -UserPrincipalName $user.UserPrincipalName | Select-Object -ExpandProperty Licenses
17     Set-MsolUser -UserPrincipalName $user.UserPrincipalName -UsageLocation "FI"
18
19     #Poistaa lisenssit
20     if ($license.AccountSkuId -ne $null)
21     {
22         Set-MsolUserLicense -UserPrincipalName $user.UserPrincipalName -RemoveLicense $license.AccountSkuId
23         #Raportti lopuksi
24         Write-Output $user.UserPrincipalName removed $license.AccountSkuId
25     }
26 }
27
28 Write-Output "Valmis"
29 Stop-Transcript
  
```

Kuva 16. Skripti Office 365 -palvelusta poistettujen käyttäjien lisenssien poistamiseen

## 6.2 Skriptin ajoittaminen ja jatkokehitys

Skripti on ajoitettu suoritettavaksi joka kuukauden ensimmäisenä päivänä. Skriptin ajoitus toimii ADFS-palvelimelta, mistä on suora pääsy Office 365-pilveen. Jatkokehityksenä mietin, että tekisin skriptiin toiminnon, joka listaa poistettujen käyttäjien lisenssien tiedot ja mahdollisesti näyttäisi, että montako eri lisenssiä on jäljellä.

## 7 LOPUKSI

Kiinnostukseni PowerShelliin alkoi ammattikouluaikoina. Kävimme siellä läpi perusteita, jonka jälkeen opiskelin aihetta lisää ammattikorkeakoulussa ja työpaikoillani. Monet samat komennot jotka toimivat cmd-komentotulkissa toimivat myös PowerShellillä, mikä helpotti käytännön opettelemista.

Idea opinnäytetyöhön sai alkunsa Office 365 -palvelun lisenssien automaattisesta poistosta, mistä sain hyvän alun tähän projektiin. Samalla huomasin, että Office 365 -palvelun graafinen käyttöliittymä on hyvin rajoittunut verrattuna PowerShelliin. PowerShell-skriptien teko ja historian tutkiminen oli parasta työssäni.

Opinnäytetyö antoi paljon intoa opetella PowerShellin käyttöä ja sen ominaisuuksia. Työpaikallani tarvitsen päivittäin PowerShell-ohjelmaa, koska jotkin toiminnot tarvitsevat järjestelmänvalvojan oikeuksia ja ovat integroitua paremmin Powershelliin. Luin myös paljon erilaisilta nettifoorumeilta PowerShellin käyttötavoista, tietoturvasta sekä hallinnasta. Oma osaaminen PowerShellin käytössä kasvoi huomattavasti tämän opinnäytetyön myötä. PowerShell-skriptien tekeminen ja ymmärtäminen tuntuu paljon loogisemmalta työn jälkeen.

## 8 LÄHTEET

Aiello J & Wheeler, S. Microsoft 2018. How to Write and Run Scripts in the Windows PowerShell ISE. Viitattu 12.5.2019.

<https://docs.microsoft.com/en-us/powershell/scripting/components/ise/how-to-write-and-run-scripts-in-the-windows-powershell-ise?view=powershell-6>

Bisson, S. 2018. Inside the updated windows console. Viitattu 19.4.2019.

<https://www.techrepublic.com/article/inside-the-updated-windows-console/>

Bouchard, N. 2018. Active Directory Federation Services: The Comprehensive Guide. Viitattu 12.6.2019.

<https://www.sherweb.com/blog/office-365/active-directory-federation-services/>

Bradbury, D. 2008. Power and control: a history of the operating system. Viitattu 14.4.2019.

<https://www.computerweekly.com/feature/Power-and-control-a-history-of-the-operating-system>

CLEVERISM www-sivusto n.d. PowerShell. Viitattu 20.6.2019.

<https://www.cleverism.com/skills-and-tools/powershell>

Computer hope 2018. GUI. Viitattu 16.4.2019.

<https://www.computerhope.com/jargon/g/gui.htm>

Gitter 2016. A Brief History of the Command Line. Viitattu 19.7.2019.

<https://www.freecodecamp.org/news/the-command-line-1fdb692b3bf/>

Haaranen, J. 2015. Mikä on PowerShell ja miksi IT-ammattilaisten pitäisi sitä osata?. Viitattu 18.5.2019.

<https://www.sulava.com/mika-on-powershell-ja-miksi-it-ammattilaisten-pitaisi-sita-osata/>

Microsoft 2018a. PowerShell. Viitattu 1.6.2019.

<https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-6>

Microsoft 2018b. Approved Verb for PowerShell Commands. Viitattu 20.6.2019.

<https://docs.microsoft.com/en-us/powershell/developer/cmdlet/approved-verbs-for-windows-powershell-commands>

The Scripting Guys. Microsoft 2015c. Table of Basic PowerShell Commands. Viitattu 20.2.2019.

<https://blogs.technet.microsoft.com/heyscriptingguy/2015/06/11/table-of-basic-powershell-commands/>

Microsoft 2018d. Set-ExecutionPolicy. Viitattu. 20.3.2019.

<https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.security/set-executionpolicy?view=powershell-6>

Levy, S. 2018. Graphical user interface. Viitattu 17.4.2019.

<https://www.britannica.com/technology/graphical-user-interface>

Payette B, Siddaway R. 2018. Windows PowerShell in action.

Pietroforte, M n.d. Differences between PowerShell versions. Viitattu 20.7.2019.

<https://4sysops.com/wiki/differences-between-powershell-versions/>

Rathnam, L. 2019. Windows Core OS: What it is - and is it the future of Windows? Viitattu 5.6.2019.

<http://techgenix.com/windows-core-os/>

Rounds, D. 2016. A short history of computer user interface design. Viitattu 19.4.2019.

<https://usabilla.com/blog/short-history-computer-user-interface-design/>

Rouse, M. 2017. Cloud management. Viitattu 10.6.2019.

<https://searchcloudcomputing.techtarget.com/definition/cloud-management>



Schlosser, K. 2016. 1973 Xerox Alto computer that inspired Paul Allen and others is restored and running at Seattle museum. Viitattu 20.6.2019.

<https://www.geekwire.com/2016/1973-xerox-computer-inspired-paul-allen-others-restored-running-seattle-museum/>

Shyamsundar, T. 2018. What is ADFS? Viitattu 13.6.2019.

<https://www.okta.com/blog/2018/06/what-is-adfs/>

Techterms www-sivusto n.d. GUI. Viitattu 18.4.2019.

<https://techterms.com/definition/gui>

Thakoor, R. 2018. PowerShell equivalents for some common Unix com. Viitattu 2.7.2019.

<https://www.supinfo.com/articles/single/6708-powershell-equivalents-for-some-common-unix-commands>

Trevorah, A. 2016 A Brief history of the command line. Viitattu 15.4.2019.

<https://medium.freecodecamp.org/the-command-line-1fdb692b3bf>

Webdesignerdepot www-sivusto 2009. Operating system interface between 1981-2009. Viitattu 17.4.2019.

<https://www.webdesignerdepot.com/2009/03/operating-system-interface-design-between-1981-2009/>

Withee R, Withee K, Reed, J. 2016, 20-22 Microsoft Office 365 for dummies.