

Jussi Lindfors

WINDOWS SERVER 2016 JA AKTIIVIHAKEMISTON KÄYTTÖ
POWERSHELLIN AVULLA

Tietojenkäsittelyn koulutusohjelma
2019

WINDOWS SERVER 2016 JA AKTIIVIHAKEMISTON KÄYTTÖ POWERSHELLIN AVULLA

Lindfors, Jussi
Satakunnan ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Helmikuu 2019
Sivumäärä: 41
Liitteitä: -

Asiasanat: PowerShell, Windows Server 2016, aktiivihakemisto

Tämän opinnäytetyön tavoitteena oli perehtyä Windows Server 2016 käyttöjärjestelmän hallintaan sekä aktiivihakemiston käyttöönottoon PowerShell -työkalun avulla. Työssä käytetyt esimerkkikoneet ovat virtuaalipalvelimia.

PowerShell on Microsoftin luoma komentorivityökalu, jonka avulla pystytään hallitsemaan Windows käyttöjärjestelmiä. Alun perin PowerShell on julkaistu vuonna 2006 ja sen pääarkkitehtina on toiminut Jeffrey P. Snover. Tutustumme työn aikana PowerShellin aikaisempiin versioihin, työkaluihin sekä itse PowerShellin käyttöön.

Windows Server 2016 on Microsoftin kehittämä palvelinkäyttöjärjestelmä. Tutustumme työn aikana sen rooleihin, ominaisuuksiin ja aktiivihakemistoon.

Työn käytännön osuudessa tulemme tutustumaan etäkäyttöön, asennamme Windows Server 2016 palvelimelle aktiivihakemistopalvelun sekä muokkaamme toisen palvelimen asetuksia niin että saamme liitettyä sen meidän toimialueeseemme. Aktiivihakemiston osalta tutustumme organisaatioyksiköiden, konetilien, käyttäjien sekä ryhmien luontiin. Kaikki muutokset tulemme tekemään PowerShellin avulla.

Opinnäytetyön tuloksista saatiin hyvä kuva siitä, miten PowerShellä pystytään käyttämään tehokkaasti eri kohteiden manipulointiin. Työn perusteella pystytään myös toteamaan, että PowerShell on tehokkaimmillaan, kun suoritetaan toistuvia tehtäviä sekä manipuloitaessa useampia objekteja samanaikaisesti.

USING WINDOWS SERVER 2016 AND ACTIVE DIRECTORY WITH POWERSHELL

Lindfors, Jussi

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Business Information Technology

February 2019

Number of pages: 41

Appendices: -

Keywords: PowerShell, Windows Server 2016, Active Directory,

The purpose of this thesis was to become familiar with the deployment and management of Active Directory Domain Services with PowerShell in a Windows Server 2016 environment. All used examples in this thesis are from virtual machines.

PowerShell is a command-line shell created by Microsoft. It is used to manage and manipulate Windows operating systems. It was first launched in 2006. Main architect of PowerShell is Jeffrey P. Snover. In this thesis we will go through earlier PowerShell versions, tools and look at the use of PowerShell itself.

Windows Server 2016 is a server-side operating system created by Microsoft. We will look at roles, features and Active Directory in Windows Server 2016.

In the practical part of this thesis we will look at remote controlling our Server, install Windows Server 2016 Active Directory Domain Services. We will also manage the settings of another server, so we can join it to our domain. In Active Directory we will familiarize ourselves with creating organizational units, computer accounts, user accounts and new groups. All tasks will be done using PowerShell.

The conclusion of this thesis gives us a good picture of how we can use PowerShell to manipulate different objects. Based on this thesis we can point out that PowerShell is at its strongest when used to automate frequent tasks and in manipulating multiple objects.

SISÄLLYS

1	JOHDANTO.....	6
2	POWERSHELL.....	7
2.1	Historia.....	7
2.2	Käyttöliittymä	13
2.3	ISE Skriptausalusta	13
3	WINDOWS SERVER 2016.....	15
3.1	Windows Server 2016.....	15
3.2	Roolit ja ominaisuudet	15
3.3	Aktiivihakemisto.....	16
3.4	Active Directory Users and Computers	17
4	POWERSHELLIN KÄYTTÖ	19
4.1	Komennon rakenne	19
4.2	Command-Let	19
4.3	Alias	19
4.4	Funktiot.....	20
4.5	Putkittaminen	21
4.6	Profiilit	21
4.7	Skriptaus ja skriptien ajo.....	22
4.8	.csv tiedostot	23
5	WINDOWS SERVERIN HALLINTA	25
5.1	Etäkäyttö	25
5.2	Roolien asentaminen.....	26
5.3	Uuden koneen konfigurointi ja liittäminen toimialueeseen	28
6	AKTIIVIHAKEMISTON HALLINTA	30
6.1	Organisaatioyksiköiden luonti	30
6.2	Konetilien luonti	31
6.3	Käyttäjätilien luonti	33
6.4	Ryhmän luonti.....	37
7	LOPUKSI.....	39
	LÄHTEET.....	40

TERMISTÖ JA LYHENTEET

.Net Framework	Ohjelmistokomponentti kirjasto, jota käytetään kehittäessä ohjelmia Windows laitteille.
AD DS	Active Directory Domain Services tai aktiivihakemisto on Microsoftin luoma toimialueen hallintaan käytettävä käyttäjätietokanta ja hakemistopalvelu.
Alias	PowerShellin käyttämä nimitys lyhennetyille komennoille.
CLR	Common language runtime on ajoympäristö joka mahdollistaa .Net ohjelmien ajamisen.
CMDLET	Command-Let, PowerShellin nimi peruskomennolle.
CSV	Comma-separated values, tekstitiedosto, jossa eritellään listan sisältö ylätunnisteen nimellä ja tunnisteen sisältö on eroteltu pilkku merkein.
DC	Domain Controller tai toimialueen ohjauskone.
DHCP	Dynamic Host Configuration Protocol mahdollistaa automaattisen ip-osoitteen jakamisen verkossa oleville laitteille.
DNS	Domain Name System tai nimipalvelin, joka muuntaa nimitunnuksen ip-osoitteeksi.
DSC	Desired State Configuration seuraa ja hallitsee laitteen konfiguraatiodataa.
ISE	Integrated Scripting Environment, PowerShell skriptaukseen käytettävä editori.
Nano Server	Minimalistinen Windows palvelin.
PowerShell	Microsoftin luoma komentorivitulkki.
SSH	Secure Shell, salattu tietoliikenne protokolla.

1 JOHDANTO

Opinnäytetyön aiheena on Windows PowerShell. Tällä Microsoftin luomalla komentorivityökalulla voidaan hallita Windows -käyttöjärjestelmiä ja sovelluksia sekä suorittaa toimintoja komentojen ja skriptien avulla.

PowerShell on kehitetty helpottamaan niin yksittäisten kuin myös toistuvien töiden suorittamista paikallisesti ja etänä Windows laitteissa. Se on helposti lähestyttävä ja yksinkertainen skriptauskieli, jonka avulla ylläpitäjät ja tehokäyttäjät pystyvät tehostamaan työntekoaan.

Teoriaosuudessa tutustutaan ensin PowerShellin käyttöliittymään, historiaan, perusteisiin ja sen yleisimpiin toimintoihin. Lisäksi tutustutaan Microsoftin Windows Server 2016 käyttöjärjestelmään ja aktiivihakemistoon.

Työn loppupuolella luodaan virtuaaliympäristöön toimialue, jota hallinnoidaan PowerShellin avulla. Tähän toimialueeseen luodaan kaksi palvelinta. Toinen palvelimista on ohjainpalvelin, johon luodaan aktiivihakemisto. Toinen palvelimista kantaa tunnusta SRV, jota käytetään havainnollistamaan toimialueeseen liittymistä sekä etäyhteyden luomista.

2 POWERSHELL

PowerShell on Windowsin komentorivityökalu, joka on eritoten suunniteltu ylläpitäjille ja tehokäyttäjille. Se sisältää interaktiivisen komentokehoteen ja skriptausalustan, joita voidaan käyttää joko yhdessä tai erikseen. Uusimassa versiossa PowerShell on myös järjestelmäriippumaton, joten on mahdollista suorittaa PowerShell komentoja käyttäen Linux tai Mac OS laitteita.

PowerShell on rakennettu .NET framework common language runtime (CLR) ja .NET frameworkin päälle. .NET framework on Microsoftin kehittämä ohjelmistokirjasto jota käytetään kun luodaan ohjelmia pääasiassa Windows laitteille. Tämä mahdollistaa sen, että voimme manipuloida näihin liittyviä objekteja PowerShellin avulla. (Microsoft docs 2017 a.) Tämä luku käsittelee PowerShellin historiaa, käyttöliittymää ja ISE (Integrated Scripting Environment) -skriptaustyökalua.

2.1 Historia

Vuonna 2002 Microsoft alkoi kehittää uutta tehokkaampaa tapaa käyttää komentoriviä. Tämän projektin nimi oli tuolloin Monad (Snover 2002, 1.) Samana vuonna projektin pääarkkitehti Jeffrey P. Snover julkaisi Monad Manifeston. Tässä manifestissa käytiin läpi ideoita ja komentorivin konseptia, joka johti lopulta sen kehittämiseen. Monad hyödynsi .NET frameworkia ja käytti sen pohjalta työkaluja, joilla saatiin hyötyjä kehittäjille, testaajille, tehokäyttäjille ja ylläpitäjille. (Snover 2002, 1.)

Snover esitti Monadin koostuvan viidestä eri kohdasta (Snover 2002, 1.):

- Monadin-automaatiomalli (MAM)
 - Automaatiomalli perustui .NET luokkiin, menetelmiin ja ominaisuuksiin, joilla luotiin Cmdlet:tejä. (Command-let)
- Monad-komentorivi (MSH)
 - .NET pohjainen skriptien suoritusympäristö.
- Monad-hallintamallit (MMM)

- Hallinnoituja koodiluokkia, joita voidaan käyttää tietyissä hallinnollisissa tilanteissa.
- Monad-etäskriptausta (MRS)
 - Työkalut, jotka mahdollistavat skriptien ajamisen etänä monille laitteille.
- Monad-hallintakonsoli (MMC)
 - .NET pohjainen malli jota käytettäisiin tuomaan graafisesta käyttöliittymästä tietoja komentoriviin käyttäjälle näkyviksi valmiiksi skripteiksi.

PowerShellin ensimmäinen versio julkaistiin marraskuussa 2006. Se sisälsi yli 130 komentorivin työkalua, joilla voitiin suorittaa yleisiä järjestelmän ylläpidollisia tehtäviä. Näihin tehtäviin kuului palveluiden- prosessien- tapahtumalokien ja sertifikaattien hallinta, rekisterin muokkaus sekä Windowsin hallinnan instrumentointi (WMI). (Microsoft support 2017 a.)

Komentorivin työkalut sisälsivät mahdollisuuden datan ja objektien järjestämiselle, suodattamiselle ja formatoinnille. Se sisälsi myös työkalut datan käsittelyyn eri säiliöissä ja formaateissa. Näihin lukeutui aktiivihakemiston palveluliittymät (ADSI), Windows -hallinnan instrumentointi (WMI), COM -objektit, ActiveX data -objektit (ADO), HTML ja XML. (Microsoft support 2017 a.)

PowerShell 1.0 ei kuulunut automaattisesti käyttöjärjestelmiin vaan se oli asennettava verkosta. Se vaati .NET framework 2.0:n (2.0.50727) toimiakseen ja sen sai asennettua seuraaviin käyttöjärjestelmiin:

- Windows XP SP2 & SP3
- Windows Server 2003 SP1 & SP2 & R2
- Windows Vista & Vista SP2

(Microsoft support 2017 a.)

PowerShellin versio 2.0 julkaistiin lokakuussa 2009 ja sen uusia ominaisuuksia olivat seuraavat:

- Etäkäyttö

- Mahdollistaa käskyjen ajamisen yhteen tai useampaan laitteeseen, jossa on asennettuna PowerShell -työkalu.
- ISE -skriptienhallintatyökalu
 - Pystytään ajamaan ja editoimaan skriptejä graafisen liittymän kautta. Tämän ominaisuuksiin kuului väripohjainen syntaksi, valikoiva suorittaminen, unicode tuki sekä asiayhteyteen liittyvät apupyynnöt.
- Moduulit
 - Moduuleilla voidaan organisoida ja osittaa valmiita koodeja paketeiksi, jonka joku muu voi ladata ja suorittaa. Tällä saadaan avattua esimerkiksi erilaisia toiminnallisuuksia jotka eivät ole oletuksena toimivia.
- Edistyneet funktiot
 - Käyttäjän voi itse luoda cmdlet komentoja.
- Taustatyöt
 - Voidaan ajaa taustalla käskyjä tai toimintoja ilman että tarvitsee olla tekemisissä konsolin kanssa.
- Yli 100 uutta cmdlet:tiä

PowerShell 2.0 tuli valmiiksi asennettuna Windows 7 ja Windows Server 2008 R2 -käyttöjärjestelmissä. Se vaati jonkin .NET framework -versioista 2.0, 3.0 tai 3.5 ja sen asennus onnistui seuraaviin käyttöjärjestelmiin:

- Windows XP SP2
- Windows Vista SP1 & SP2
- Windows 7 (oletuksena)
- Windows Server 2003 SP2 & Server 2008 SP1 & SP2 sekä 2008 R2 (oletuksena)
(Microsoft support 2017 b.)

PowerShell versio 3.0 julkaistiin syyskuussa 2012 ja se sisälsi paljon uusia ominaisuuksia ja niistä tärkeimmät olivat:

- PowerShell Workflow
 - Voidaan käyttää tilanteissa, joissa ajetaan esimerkiksi pitkiä ja itseään toistavia tehtäviä. Sen vahvuus normaaliin skriptiin verrattuna on se, että

samanaikaisesti voidaan suorittaa käsky usealla laitteella. Sisältää myös automaattisen virheenkorjauksen.

- Pääsy PowerShelliin verkon kautta
 - Ominaisuus, joka mahdollistaa käskyjen ja skriptien ajon web-pohjaisesta konsolista. Laitteella ei tarvitse olla asennettuna PowerShellä.
- Skriptausalustaan uusia ominaisuuksia kuten intellisense, komentoikkuna ja yhdistetty konsolipaneeli
- Tuki .NET framework 4.0
- Moduulien ohjeiden päivitys
- Parannellut verkko-ohjeet
- Uusia Cmdlet:tejä
(Siddaway 2012; Microsoft docs 2017 b.)

PowerShell 3.0 tuli valmiiksi asennettuna Windows 8 ja Windows Server 2012 -käyttöjärjestelmissä. Työkalu vaatii .NET framework 4.0 tai 4.5 version. Asennus onnistuu seuraaviin käyttöjärjestelmiin:

- Windows 7 SP1 & 8 (oletuksena)
- Windows Server 2008 SP2 & R2 SP1 sekä Windows 2012 (oletuksena)
(Microsoft docs 2017 b.)

PowerShellin versio 4.0 julkaistiin lokakuussa 2013 ja sen yhtenä uutena ominaisuutena esiteltiin Windows PowerShell Desired State Configuration (DSC). Kyseessä oli hallintajärjestelmä, jonka avulla pystytään käyttöönottamaan ja hallinnoimaan konfiguraatiodataa ohjelmien palveluissa ja ympäristöissä, joissa näitä käytetään. Muita pienempiä päivityksiä olivat muun muassa:

- Save-Help
 - Voidaan ladata päivitetty help-tiedosto ja viedä koneelle, jolla ei ole esimerkiksi verkkoyhteyttä.
- Päivityksiä skriptausalustaan kuten tuki Workflow'n testaukseen ja etänä tehdyn skriptin testaukseen
- Workflow sai myös uusia ominaisuuksia, kuten laitteen kaatumisen jälkeen automaattinen uudelleenyhdistys ja automaattinen sammutus, kun aktiiviset työt ovat loppuneet

(Microsoft docs 2017 c.)

PowerShell 4.0 tuli valmiiksi asennettuna Windows 8.1 ja Windows Server 2012 -käyttöjärjestelmissä. Työkalu vaati .NET framework 4.5 version ja asennus onnistuu seuraaviin käyttöjärjestelmiin:

- Windows 7 SP1 & Windows 8.1 (oletuksena)
- Windows Server 2008 R2 SP1 & 2012 sekä 2012 R2 (oletuksena)

(Microsoft docs 2017 c.)

PowerShellin versio 5.0 julkaistiin helmikuussa 2016 ja tämän version päivityksiin lukeutuivat muun muassa:

- Luokka -toiminto (Class)
- Moduuli, joka sisältää cmdlet:tejä, joiden avulla tiedostoja ja kansioita voidaan muuttaa, lukea ja päivittää ZIP -muotoon
- Tiedostojen ja kansioiden kopioiminen koneiden välillä, joissa on PowerShell -sessio auki. Tämä toimii myös Nano Servereillä
- Sisällön kryptauksen ja dekryptauksen mahdollistava cmdlet
- Pieniä päivityksiä skriptausalustaan

(Microsoft docs 2017 d.)

PowerShell 5.0 tuli valmiiksi asennettuna Windows 10 ja myöhemmin Windows Server 2016 -käyttöjärjestelmissä mukana. Se vaati .NET framework 4.5 version ja Windows Management Framework 5.0 version. Asennus onnistuu seuraaviin käyttöjärjestelmiin:

- Windows 7 SP1 & 8.1 & 10 (oletuksena)
- Windows Server 2008 R2 SP1 & 2012 & 2012 R2 SP1 (tuli myöhemmin oletuksena Server 2016 mukana)

(Microsoft docs 2017 d.)

Elokuussa 2016 Powershell 5.1 julkaistiin Windows 10 anniversary update paketin mukana. 5.1 toimii samoilla alustoilla kuin 5.0 ja tämä päivitys toi mukanaan verrattain vähän uutta.

PowerShell Core 6.0 julkaistiin 10. tammikuuta 2018 (Aiello 2018.) ja se on suurin muutos PowerShellin historiassa, kun Microsoft siirtyi käyttämään .NET frameworkin sijasta .NET Core kirjastoa. Tämä muutos on mahdollistanut sen, että PowerShellistä on saatu järjestelmäriippumaton ja sitä voidaan nyt käyttää niin Windows- Linux- ja Mac OS-laitteilla.

6.0 ei ole yhtä ominaisuusrikas kuin sen vanhemmat versiot. Aikaisempien versioiden käyttämä .NET framework oli niin kookas ja pitkälle kehitetty, että vastaavia toimintoja ei ole vielä ehditty muuttamaan toimivaksi uuteen .NET Core kirjastoon. Windows -laitteella voidaan käyttää rinnakkain 5.1- ja Core 6.0 versioita. Microsoft päivittää ja korjaa virheitä 3.0, 4.0 ja 5.x PowerShell -versioissa, mutta näillä näkymin 5.1 jää viimeiseksi PowerShell -versioksi ja kehitys painottuu Core 6.0 versioon. (Branscombe 2018.)

Versio 6.0 ei sisällä juurikaan uutta Windows -käyttäjän näkökulmasta. Uusia ominaisuuksia ovat muun muassa

- Docker-konttien tuki useilla Linux jakeluilla, Windows Server Corella ja Nano Corella
- PowerShell remoting protokollaa voidaan käyttää Secure Shell (SSH) protokollan kanssa
(Microsoft 2018 a.)

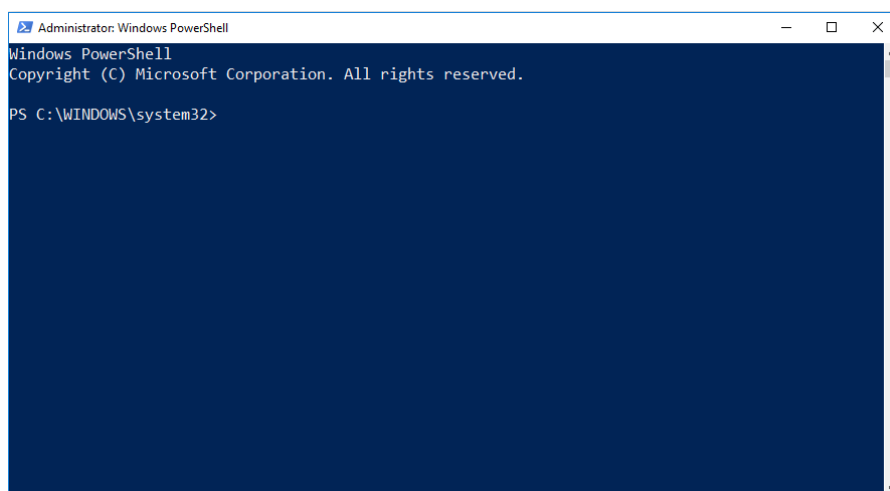
Koska uusi versio on järjestelmäriippumaton, sen saa ladattua vapaasti esimerkiksi Github-sivulta <https://github.com/PowerShell/PowerShell>. PowerShell Core 6.0 on nyt virallisesti tuettu seuraavilla käyttöjärjestelmillä:

- Windows 7, 8.1 ja 10
- Windows Server 2008 R2, 2012 R2 ja 2016
- Ubuntu 14.04, 16.04 ja 17.04
- Debian 8.7+ ja 9
- CentOS 7
- Red Hat Enterprise Linux 7
- OpenSuse 42.2
- Fedora 25 ja 26

- MacOS 10.12+
(Microsoft 2018 b.)

2.2 Käyttöliittymä

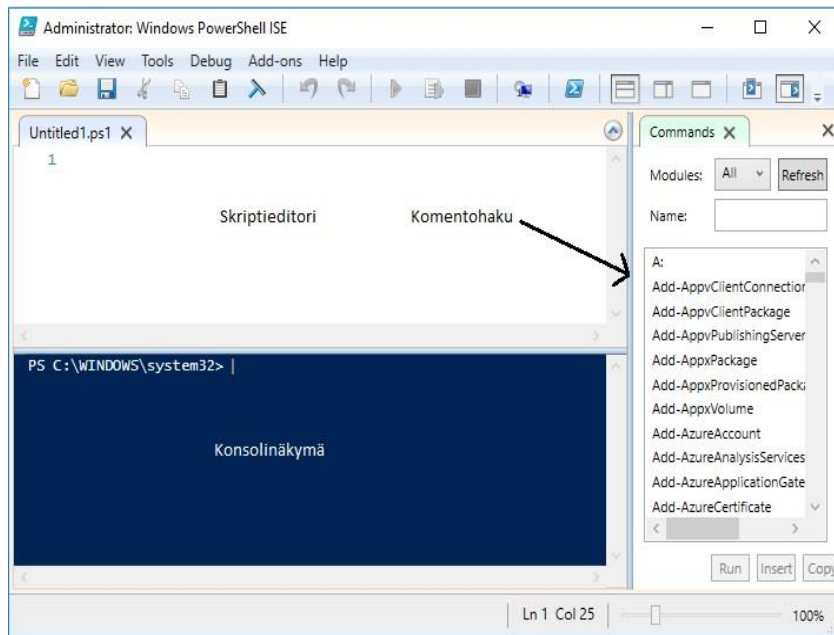
Käyttöliittymän perusnäkyvä vastaa ulkonäöltään Windowsin komentoriviä (kuva 1). Tässä konsolissa voidaan ajaa myös samoja komentoja kuin vanhassa komentorivissä esimerkiksi ”ping”, ”tracert”, ”netstat”, ”tasklist”, ”taskkill” ja niin edelleen. Näiden lisäksi on käytössä PowerShellin omat komennot ja mahdolliset käyttäjän kirjoittamat skriptit, joita voidaan ajaa myös tämän konsolin kautta.



Kuva 1. Käyttöliittymän perusnäkyvä.

2.3 ISE Skriptausalusta

ISE tai Integrated Scripting Environment sisältää konsolinäkymän lisäksi skriptieditorin ja komentohaun (kuva 2). ISE mahdollistaa komentojen suorittamisen sekä skriptien kirjoittamisen ja testauksen. Editorin hyötyihin normaalin käyttöliittymän sijaan kuuluu kopiointi, liittäminen, nopea editointi, automaattinen täyttö, syntaksin väritys ja välilehdet. (Pietroforte 2013.) Editoria ei voi asentaa Server Core -asennukseen. Näissä tilanteissa skriptin kirjoitus hoidetaan muulla laitteella ja suoritetaan sitten Server Core asennuksessa. (Jones & Hicks 2016, 14.)



Kuva 2. ISE skriptausalusta.

3 WINDOWS SERVER 2016

Windows Server on Microsoftin luoma tuoteperhe palvelinkäyttöön suunniteltuja käyttöjärjestelmiä. Se on suunniteltu jakamaan palveluita useiden käyttäjien kesken ja tarjoamaan laajat hallinnalliset ominaisuudet datan tallennukseen, käsittelyyn, ohjelmistoihin ja verkkoratkaisuihin. (Rouse 2017)

Ennen Windows Server -nimeä Microsoftin palvelinkäyttöjärjestelmät kulkivat nimellä Windows NT Server. Tähän tuoteperheeseen kuuluivat NT Server 3.5, NT Server 3.51 ja NT Server 4.0.

3.1 Windows Server 2016

Windows Server 2016 Käyttöjärjestelmä julkaistiin vuonna nimensä mukaisesti vuonna 2016. Se sisälsi parannuksia kohti pilvipalveluita, kuten tuki docker tekniikalle. Microsoft myös debytoi uutta Nano Serveriä. Se on minimalistinen versio palvelinkäyttöjärjestelmästä. Microsoftin mukaan Nano palvelin on kooltaan 93% pienempi kuin täysi palvelin asennus. (Rouse 2017)

Hyper-V virtualisointi sai myös lisäsuojasta Hyper-V Shielded VM ominaisuuden kautta. Ominaisuus hyödyntää kryptaamista suojatakseen datan vientiä ulos virtuaalikoneesta. Windows Server 2016 saa Standard ja Datacenter versioina. Nämä sisälsivät aikaisemmissa julkaisuissa samat ominaisuudet, mutta Server 2016 versiossa tämä on muuttunut. Nyt Standard versio ei sisällä edistyneempiä ominaisuuksia virtualisoinnissa, tiedon tallennuksessa ja verkostoissa. (Rouse 2017)

3.2 Roolit ja ominaisuudet

Palvelimelle asennettava rooli on joukko ohjelmia, jotka mahdollistavat erilaisia toiminnallisuksia. Esimerkiksi kun asennetaan DHCP rooli serverille, mahdollistaa se silloin serverin käytön DHCP palvelimena.

Ominaisuudet ovat ohjelmia, jotka eivät suoranaisesti ole roolien osia, mutta voivat tukea ja kasvattaa yhden tai useamman roolin toiminnallisuutta. Esimerkiksi voidaan asentaa Failover Clustering -ominaisuus, jonka avulla saadaan jaettua kriittisiä rooleja muiden ryhmässä olevien servereiden kanssa. Vikatilanteessa tämä ominaisuus mahdollistaa palvelun jatkumisen. (Microsoft e.)

3.3 Aktiivihakemisto

Aktiivihakemisto on Microsoftin tuote, joka sisältää palveluita, jotka voidaan asentaa Windows Serverille. Niillä voidaan hallita muun muassa lupapalveluita ja pääsyä verkkoresursseihin. (Rouse 2018) Seuraavaksi paneudutaan aktiivihakemiston olennaisimpaan osaan, joka on Active Directory Domain Services ja siihen viitataan myöhemmin aktiivihakemisto nimellä.

Aktiivihakemiston hallintaan ei ole olemassa yhtä ainoaa oikeaa työkalua, koska kokonaisuudessaan aktiivihakemisto on iso kokoelma erilaisia tekniikoita. Aktiivihakemisto onkin jaettu käytettäväksi eri hallintakonsoleissa. Yleisin hallintakonsoli, jota ylläpitäjät käyttävät, on Active Directory Users and Computers, jonka vuoksi se on valikoitunut tässä työssä lähemmän tarkastelun kohteeksi. Seuraavaksi on lyhyesti käyty läpi muutkin aktiivihakemiston osat. (Krause 2016, 111)

- Active Directory Domain and Trusts
 - Käytetään yleisesti isoissa ympäristöissä joissa on useita toimialueita saman verkon sisällä. Voidaan käyttää useita toimialueen nimiä helpottamaan resurssien ja palveluiden erottelua tai selventämään organisaatiorakennetta yhtiön sisällä. Pystytään myös muokkaamaan oikeuksia näiden toimialueiden välillä, jolla saadaan rajattua tai jaettua oikeuksia. (Krause 2016, 117)
- Active Directory Sites and Services
 - Käytetään enimmäkseen myös isoissa ympäristöissä. Esimerkiksi jos yritys on kasvanut niin, että eri toimipisteitä on ympäri maata tai maiden välisesti

saadaan tällä liitettyä eri toimialueita yhteen ja voidaan antaa esimerkiksi ohjeita mihin palvelimeen kirjautumiset ohjataan. (Krause 2016, 118-119)

3.4 Active Directory Users and Computers

Users and Computers -työkalua käytetään luomaan ja ylläpitämään kone- ja käyttäjätilejä, organisaatioyksiköitä sekä ryhmiä. Aktiivihakemisto rakentuu luoduista organisaatioyksiköistä, näistä voi puhua kansioina, vaikka ne eivät suoranaisesti kansioita olekaan. Näiden organisaatioyksiköiden avulla pystymme luomaan järjestelmällisen ja selvän hakemistorakenteen.

- Konetilit
 - Kun uusi laite liitetään toimialueeseen, luo aktiivihakemisto siitä automaattisesti konetilin. Uudet konetilit menevät oletuksena computers organisaatioyksikköön. Isoissa organisaatioissa, joissa tiedetään tulevien konetilien nimet, on hyödyllistä luoda nämä objektit aktiivihakemistoon manuaalisesti. Tämän avulla voidaan konetilit liittää valmiiksi haluttuun organisaatioyksikköön. Kun uuden laitteen nimi ja konetili täsmäävät ei siitä luoda enää automaattisesti uutta konetiliä. (Krause 2016, 114-115)

- Käyttäjätilit
 - Esimerkkinä voidaan käyttää tilannetta, jossa firmaan otetaan uusi työntekijä. Tälle työntekijälle tarvitsee luoda käyttäjätili yrityksen aktiivihakemistoon, jotta hän voi kirjautua sisään laitteeseen. Koska organisaatioyksiköt on lajiteltu esimerkiksi osaston mukaisesti, on meidän helppo luoda uusi tili oikeaan paikkaan ja kun tarve vaatii, pystytään isostakin hakemistosta löytämään henkilö helposti. (Krause 2016, 113)

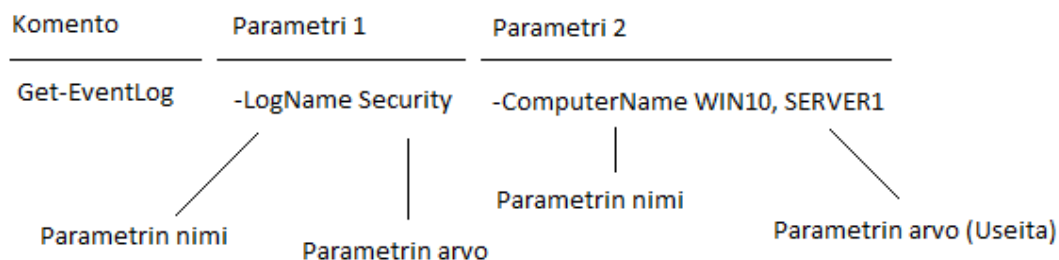
- Ryhmät
 - Ryhmiä käytetään lokeroimaan käyttäjätilejä tai konetilejä. Jos käytössä on esimerkiksi tiedostopalvelin ja sen sisällä kansioita tai kokonaisia kovalevyjä. Voidaan ryhmien avulla oikeuttaa pääsy tietyille

käyttöoikeusryhmille, jolloin vain niillä henkilöillä, jotka kuuluvat tähän tiettyyn ryhmään on pääsy tiedostojakeluun. Ryhmiä voidaan luoda käyttäjän, konetilin tai toisen ryhmän mukaan. (Krause 2016, 113-114) Tämä on hyödyllistä esimerkiksi isoissa organisaatioissa, joissa voi olla satoja käyttäjiä.

4 POWERSHELLIN KÄYTTÖ

4.1 Komennon rakenne

Peruskomento PowerShellissä rakentuu cmdlet:in ympärille. Kuvassa 3 on kuvattuna komento ja sen parametrit. Kun komennolle ”*Get-EventLog*” on annettu parametrejä, se suorittaa komennon näiden arvojen perusteella. Tässä tapauksessa se hakee kuvitteellisesta hakemistosta tapahtumat, joiden lokin-nimi on security ja se hakee ne ainoastaan koneista WIN10 ja SERVER1, kuten parametrissä 2 on ohjattu. (Jones, Hicks 2016, 39)



Kuva 3. PowerShell komennon rakenne. (Jones & Hicks 2016, 39)

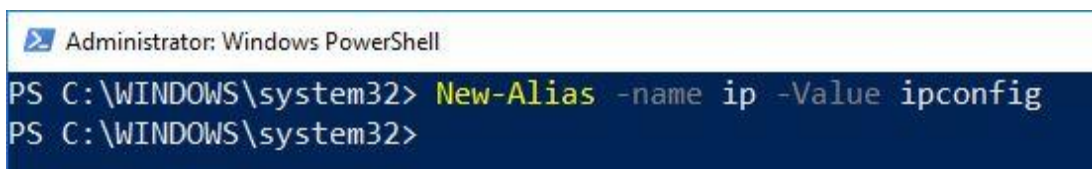
4.2 Command-Let

Command-let tai lyhennettynä cmdlet on PowerShellin oma natiivikomento. Se koostuu aina verbi-substantiivi -parista esimerkiksi ”*Get-Service*”, ”*Start-Process*” tai ”*Select-Object*”. Verbi määrittää toiminnan, joka suoritetaan ja substantiivi määrittää resurssin, josta suoritus tehdään. On myös mahdollista kehittää omia cmdlettejä jolloin käytetään siihen sopivaa verbiä, mutta kehittäjä itse voi päättää substantiivin. (Jones & Hicks 2017, 39-40.)

4.3 Alias

Vaikka cmdletit ja komennot voivat olla mukavan lyhyitä, löytyy seasta myös pitkiäkin komentoja. Näissä voidaan käyttää aliaksia eli lyhenteitä komennoista. Aliaksia on hyvä käyttää, jos huomaa vaikka useasti käyttävänsä jotain komentoa esimerkiksi, ”*Get-Service*”. Tällöin alias on ”*gsv*” ja se palauttaa samat asiat, mutta

on käyttäjälle tehokkaampaa käyttää. Listan aliaksista saa komennolla ”*Get-Alias*”. Aliaksia voi myös itse luoda, jos sellaista ei ole valmiina. (Jones, Hicks 2016, 40-41.). Kuvassa 4 luodaan uusi alias ”*ipconfig*” -käskylle. Luomisen jälkeen komento ”*ip*” palauttaa samat tiedot kuin *ipconfig*.



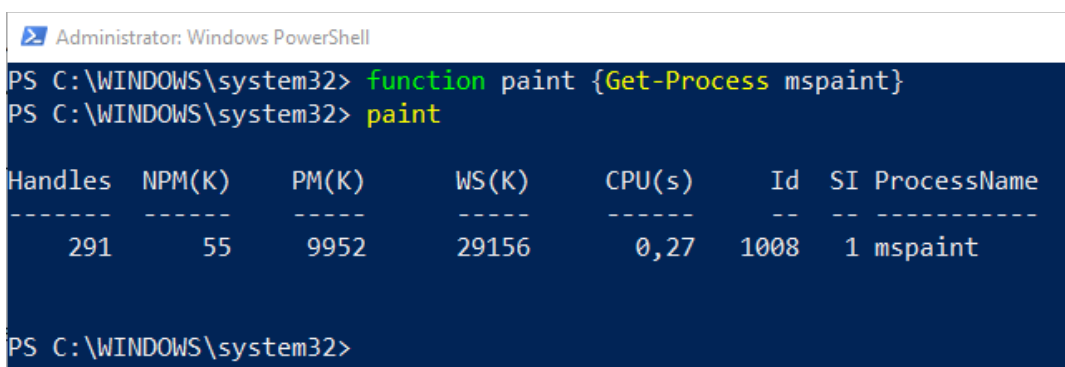
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> New-Alias -name ip -Value ipconfig
PS C:\WINDOWS\system32>
```

Kuva 4. Uuden aliaksen luominen.

4.4 Funktiot

Funktiot ovat luotuja käskyjä tai toimintoja, jota voidaan kutsua annetulla nimellä. Näitä funktioita käytetään yleensä toistuvissa tehtävissä. Sen sijaan että kirjoitettaisiin sama käsky aina uudelleen, saadaan funktiota käyttämällä skriptit pysymään paljon siistimpinä ja mahdollistetaan helppo skriptin uudelleenkäyttö. (Bertram 2015.)

Funktiot voivat olla äärimmäisen yksinkertaisia, esimerkkinä kuva 5, jossa funktiolle annetaan nimi käskyllä `function` (nimi) ja aaltosulkujen väliin annetaan suoritettava toiminto tai suoritettavat toiminnot. Tässä tapauksessa siis käytettävä komento on ”*Get-Process*” ja parametrin arvo ”*mspaint*”. Kun funktiota seuraavalla rivillä kutsutaan sille annetulla nimellä (*paint*) hakee se annetun käskyn mukaan prosessitiedot kohteesta ”*mspaint*”.



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> function paint {Get-Process mspaint}
PS C:\WINDOWS\system32> paint

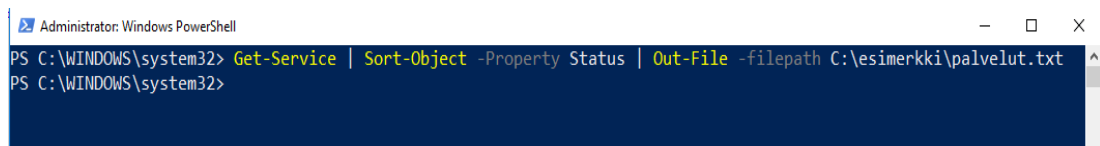
Handles      NPM(K)      PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
-----
291          55          9952       29156       0,27        1008  1 mspaint

PS C:\WINDOWS\system32>
```

Kuva 5. Funktion toiminta.

4.5 Putkittaminen

Komentojen putkittamisella saadaan yhdistettyä komentoja toisiinsa. Putkittaminen tapahtuu merkillä | eli putkitetun komennon rakenne on esimerkiksi Komento 1 | Komento 2 | Komento 3. Näitä käsitellään aina yksi kerrallaan, vasemmalta oikealle. (Microsoft docs 2018 c.)



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Get-Service | Sort-Object -Property Status | Out-File -filepath C:\esimerkki\palvelut.txt
PS C:\WINDOWS\system32>
```

Kuva 6. Esimerkki putkittamisesta.

Kuvassa 6 on putkitettu kolme komentoa. Ensinnäkin ”*Get-Service*” hakee palvelut ja se siirretään ”*Sort-Object*” komennolle. Tälle on annettu ohje järjestää palvelut niiden tilan mukaan. Viimeisenä on ”*Out-File*”, jolla tulokset saadaan tallennettua esimerkiksi tässä tapauksessa tekstitiedostoon. Tästä komentojen yhdistelmästä saadaan siis esimerkki-nimiseen kansioon palvelut.txt-tiedosto, joka sisältää tallennetut tiedot.

4.6 Profiilit

Kun lisätään uusia aliaksia, toimintoja ja skriptejä, ne pysyvät istunnossa vain siihen asti, kun kyseinen istunto on auki. Jos halutaan säilyttää näitä toiminnallisuuksia, voidaan tähän käyttää apuna profiileita. Kun profiiliin lisätään muutoksia, latautuvat ne aina uuden käynnistyksen jälkeen. Profiileja on neljä eri tyyppiä (kuva 7) ja istunnon profiilin saa selville kirjoittamalla ”*\$Profile*”. Jos ei tiedetä, onko profiili jo olemassa, voidaan se selvittää ”*Test-Path \$profile*” komennolla. Tämä palauttaa arvon true tai false. (Microsoft TechNet 2012.)

```

Administrator: Windows PowerShell
PS C:\Users\Administrator> $profile.AllUsersAllHosts
C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1
PS C:\Users\Administrator> $profile.AllUsersCurrentHost
C:\Windows\System32\WindowsPowerShell\v1.0\Microsoft.PowerShell_profile.ps1
PS C:\Users\Administrator> $profile.CurrentUserAllHosts
C:\Users\Administrator\Documents\WindowsPowerShell\profile.ps1
PS C:\Users\Administrator> $profile.CurrentUserCurrentHost
C:\Users\Administrator\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
PS C:\Users\Administrator>

```

Kuva 7. Profiilien eri tyypit.

4.7 Skriptaus ja skriptien ajo

Perusidea PowerShell -skripteissä on toistuvien töiden ajaminen ilman, että niitä tarvitsee kirjoittaa joka kerta uudelleen. Tätä voidaan verrata batch-tiedostoihin, joka itsessään on esimerkiksi notepad -ohjelmalla kirjoitettu yksinkertainen ohje, jonka mukaan tietokone suorittaa sille annettuja tehtäviä komentorivin kautta. Koska Windowsin komentorivissä on suhteellisen rajoittunut kieli, ei se anna mahdollisuuksia suorittaa kaikkein monimutkaisimpia töitä. (Jones & Hicks 2016, 263-264.)

Kirjoitetaan siis yksinkertainen skripti, joka tutkii viiden sekunnin välein, onko ohjelma -laskin auki. Jos laskin on auki, suljetaan se ja skripti jatkaa tätä tutkimista, kunnes skriptin ajo lopetetaan. (Kuva 8.)

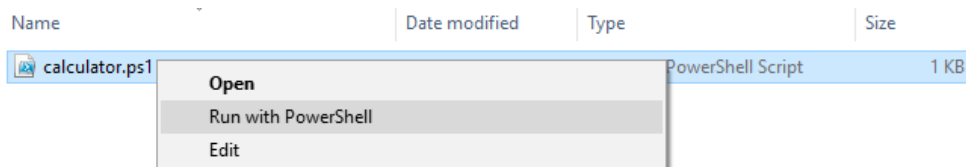
```

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1* X
1 do {
2     Start-Sleep -Seconds 5
3     Get-Process | Where-Object {$_.Name -eq "calculator"} | Stop-Process
4 } Until ($false)

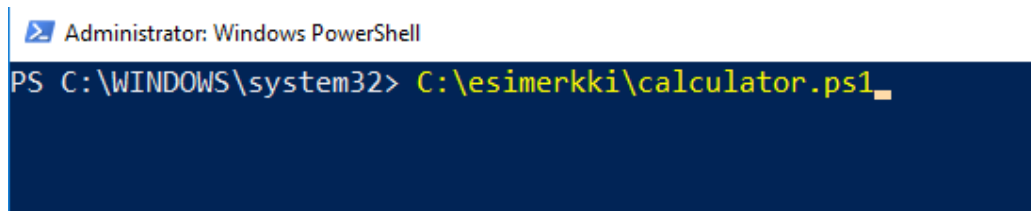
```

Kuva 8. PowerShell skripti.

Kun skripti on tallennettu, voidaan se suorittaa joko resurssienhallinnan kautta (kuva 9) tai suorittamalla se vaihtoehtoisesti PowerShellissä itsessään, hakemalla tiedosto nimellä ja suorittaa se. (kuva 10)



Kuva 9. Skriptin suoritus resurssienhallinnan kautta.



Kuva 10. Skriptin suoritus komentorivillä.

4.8 .csv tiedostot

Comma-separated values (CSV) -tiedosto on tavallinen tekstitiedosto, joka sisältää listan dataa. Näitä tiedostoja käytetään yleensä siirrettäessä dataa eri ohjelmien tai koneiden välillä. Esimerkiksi monet tietokannat tukevat .csv -tiedosto formaattia. Tiedoston rakenne on yksinkertainen. Tekstitiedoston ylätunnisteessa on eritelty listan sisältö ja data on tämän ylätunnisteen alla. (Hoffman 2018.) (kuva 11)



Kuva 11. .csv tiedoston rakenne.

Tätä voidaan hyödyntää PowerShellissä monin tavoin: saatamme esimerkiksi tarvita listan päätteen prosessitiedoista. Tämän saamme tehtyä komennolla ”*Get-Process / Select-Object -Property CPU, Id, ProcessName -Last 10 | Out-File C:\kansio\tiedostonnimi.csv*”. (Kuva 12)

```
PS C:\Windows\system32> Get-Process | Select-Object -Property CPU, Id, ProcessName -Last 10 | Out-File C:\prosessit\Prosessiliista.csv
PS C:\Windows\system32>
```

Kuva 12. CSV tiedoston luominen prosessitiedoista

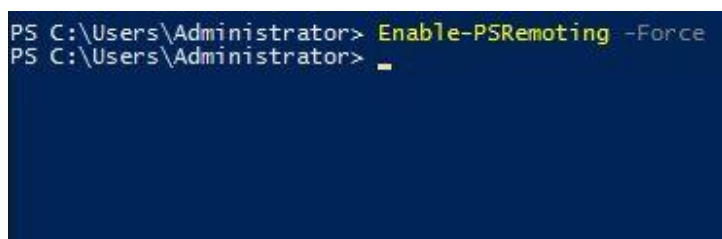
.csv -tiedostoja käytetään tässäkin työssä vielä myöhemmin, kun luodaan useampia organisaatioyksiköitä, koneita sekä käyttäjiä samanaikaisesti.

5 WINDOWS SERVERIN HALLINTA

Entäpä sitten PowerShellin käyttötarkoitus? Lähestytään asiaa esimerkin avulla: jos halutaan muuttaa palvelua tai ominaisuutta yhden kerran, onko silloin järkevää käyttää aikaa skriptin tai komennon kirjoittamiseen, kun saman tehtävän saa suoritettua yksinkertaisesti käyttöliittymässä. PowerShell on hyvä työkalu itseään toistavissa tehtävissä. Jos tiedetään, että tätä palvelua muutetaan usein tai asennetaan muille palvelimille, on silloin järkevää käyttää hetki aikaa skriptin tai komennon luomiseen, jotta työ helpottuisi jatkossa. Koska aihe on laaja, keskitytään tässä luvussa yleisimpiin tehtäviin, joita PowerShellin kanssa voidaan tehdä.

5.1 Etäkäyttö

PowerShellin etäkäyttö mahdollistaa komentojen ajamisen etänä ja se on verrattavissa esimerkiksi SSH (Secure Shell) käyttöön. Etäkäyttö on oletuksena estetty PowerShellissa, joten ensimmäiseksi on sallittava sen käyttö. Avataan seuraavaksi etäkäyttö SRV -palvelimelle komennolla ”*Enable-PSRemoting -Force*”. Tämä mahdollistaa yhteydenoton toiselta laitteelta.



```
PS C:\Users\Administrator> Enable-PSRemoting -Force
PS C:\Users\Administrator> _
```

Kuva 12. Etäkäytön salliminen

Seuraavaksi voidaan testata Domain Controller -palvelimelta, saammeko yhteyden SRV palvelimen PowerShell istuntoon. Käytämme komentoa ”*Enter-PSSession -ComputerName SRV -Credentials SAMK\Administrator*”. Kuvassa 13 näemme onnistuneen yhteyden muodostamisen. Istunto suljetaan komennolla ”*Exit-PSSession*”. (Kuva 13).

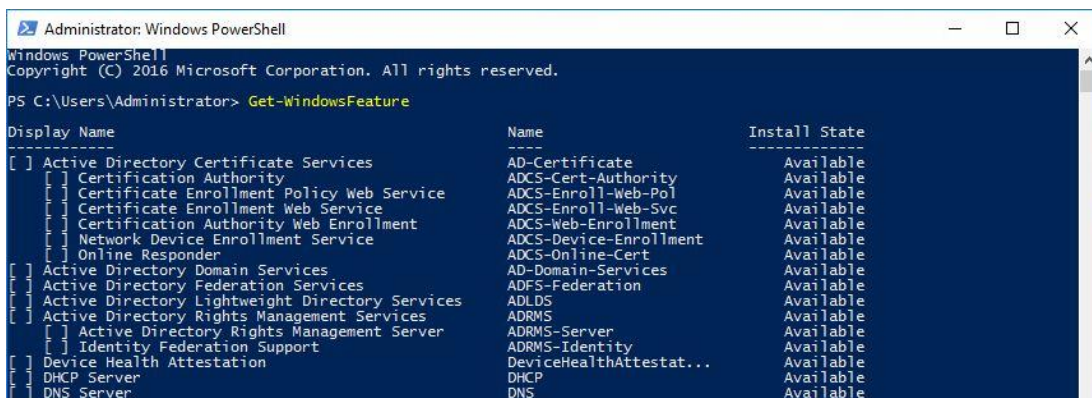
```
PS C:\Users\Administrator> Enter-PSSession -ComputerName SRV -Credential SAMK\Administrator
[SRV]: PS C:\Users\Administrator.SAMK\Documents> Exit-PSSession
PS C:\Users\Administrator>
```

Kuva 13. Yhteyden muodostaminen etänä ja sen sulkeminen

5.2 Roolien asentaminen

Useimmat ylläpitäjät ovat tottuneet asentamaan rooleja graafisen käyttöliittymän kautta. Rooleja ja ominaisuuksia voidaan asentaa myös PowerShellin kautta, asennetaan tässä kappaleessa esimerkkinä AD DS (Active Directory Domain Services), DHCP (Dynamic Host Configuration Protocol) ja DNS (Domain Name System). Nämä asennetaan Domain Controller -palvelimelle.

Avataan PowerShell ylläpitäjänä ja käytetään seuraavaa käskyä ”*Get-WindowsFeature*”. Tällä komennolla saadaan listattua asennettavissa olevat roolit ja ominaisuudet. Tästä nähdään myös jo asennetut roolit. (Krause 2016, 85)



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Get-WindowsFeature

Display Name                                     Name                                     Install State
-----
[ ] Active Directory Certificate Services         AD-Certificate                         Available
[ ] Certification Authority                     ADCS-Cert-Authority                   Available
[ ] Certificate Enrollment Policy Web Service   ADCS-Enroll-Web-Pol                   Available
[ ] Certificate Enrollment Web Service          ADCS-Enroll-Web-Svc                   Available
[ ] Certification Authority Web Enrollment      ADCS-Web-Enrollment                   Available
[ ] Network Device Enrollment Service           ADCS-Device-Enrollment                Available
[ ] Online Responder                            ADCS-Online-Cert                       Available
[ ] Active Directory Domain Services            AD-Domain-Services                    Available
[ ] Active Directory Federation Services        ADFS-Federation                        Available
[ ] Active Directory Lightweight Directory Services ADLDS                                  Available
[ ] Active Directory Rights Management Services ADRMS                                  Available
[ ] Active Directory Rights Management Server    ADRMS-Server                           Available
[ ] Identity Federation Support                 ADRMS-Identity                         Available
[ ] Device Health Attestation                   DeviceHealthAttestat...                Available
[ ] DHCP Server                                 DHCP                                    Available
[ ] DNS Server                                  DNS                                      Available
```

Kuva 13. Lista saatavilla olevista rooleista ja ominaisuuksista.

Koska asennus tehdään Domain Controller -palvelimelle, asennetaan kokeilumielessä myös DHCP- ja DNS -roolit. Kaikki kolme roolia saadaan asennettua samalla kertaa komennolla ”*Add-WindowsFeature -Name AD-Domain-Services, DHCP, DNS -IncludeManagementTools.*”

Asennus kestää hetken, jonka jälkeen PowerShell -istuntoon palautuu tieto siitä, että asentuivatko kaikki halutut roolit onnistuneesti. (kuva 14)

```
PS C:\Users\Administrator> Add-WindowsFeature -Name AD-Domain-Services, DHCP, DNS
Success Restart Needed Exit Code      Feature Result
-----
True      No           Success      {Active Directory Domain Services, DHCP Se...}

PS C:\Users\Administrator>
```

Kuva 14. Roolien asennus suoritettu

Normaalisti tämän jälkeen jatkettaisiin asennusta määrittämällä käyttöliittymän kautta tarvittavat asetukset, joita asennukselta halutaan. Nyt kuitenkin jatkamme asennusta PowerShellin avulla. Käytössä on kolme eri käskyä, joista valitaan sopivin:

- *"Install-ADDSDomain"*
Saadaan asennettua uusi aliosoite jo olemassa olevalle toimialueelle.
- *"Install-ADDSDomainController"*
Saadaan asennettua uusi Domain Controller jo olemassa olevaan toimialueeseen.
- *"Install-ADDSTForest"*
Saadaan asennettua kokonaan uusi Domain Controller uuteen toimialueeseen.
(Microsoft TechNet 2016)

Asennusta varten tarvitaan myös moduuli, joka saadaan komennolla *"Import-Module ADDSDeployment."*

Valitaan näistä vaihtoehdoista uuden toimialueen asennus. Syötetään PowerShelliin komento *Install-ADDSTForest -DomainName "samk.local"*. Tämä suorittaa asennuksen oletusarvoilla. *"DomainName"* on ainoa pakollinen parametri, joka tarvitsee antaa. (Kuva 15)

```
PS C:\Users\Administrator> Install-ADDSTForest -DomainName "samk.local"
SafeModeAdministratorPassword: *****
Confirm SafeModeAdministratorPassword: *****

The target server will be configured as a domain controller and restarted when this operation is complete.
Do you want to continue with this operation?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

Kuva 15. Uuden Domain Controllerin ja metsän asennus.

Kun komento on syötetty halutuilla parametreilla, pyytää PowerShell syöttämään ”*SafeModeAdministratorPassword*” salasanan. Tämä asettaa salasanan, jota voidaan käyttää, kun kone käynnistetään safe mode tilassa. (Microsoft TechNet 2016)

Jos halutaan nähdä kaikki asennetut roolit, voidaan käyttää samaa ”*Get-WindowsFeature*” cmdlettiä siten, että nähdään vain asennetut roolit. Kun putkitetaan ”*Get-WindowsFeature | Where Installed*” saadaan palautuksena lista asennetuista komponenteista. (Kuva 16) (Krause 2016, 86-87)

```
PS C:\Users\Administrator> Get-WindowsFeature | Where Installed
```

Display Name	Name	Install State
[X] Active Directory Domain Services	AD-Domain-Services	Installed
[X] DHCP Server	DHCP	Installed
[X] DNS Server	DNS	Installed
[X] File and Storage Services	FileAndStorage-Services	Installed
[X] File and iSCSI Services	File-Services	Installed
[X] File Server	FS-FileServer	Installed

Kuva 16. Lista asennetuista rooleista

5.3 Uuden koneen konfigurointi ja liittäminen toimialueeseen

Kun käytössä on uusi toimialue, voidaan laitteen asetuksia konfiguroida, jotta saadaan se liitettyä toimialueeseen. Seuraavaksi lähdetään muuttamaan koneen nimi joksikin, jonka voimme tunnistaa helpommin ja liitetään kone toimialueeseen. Käytössä on tällä hetkellä SRV -kone. Aloitetaan siis komennolla ”*Rename-Computer SRV*”. Tämän jälkeen, tulee pyyntö koneen uudelleen käynnistämisestä ja se tehdään käskyllä ”*Restart-Computer*”. (Kuva 17)

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Rename-Computer SRV
WARNING: The changes will take effect after you restart the computer WIN-EJV8KCUVQ1R.
PS C:\Users\Administrator>
```

Kuva 17. Uuden palvelimen nimen muutos.

Seuraavaksi kone liitetään toimialueeseen. Ensin tarvitsee muuttaa DNS palvelimen osoite. Tähän käytetään komentoa ”*Set-DnsClientServerAddress -InterfaceAlias Ethernet0 -ServerAddresses 10.223.55.20*”. Ethernet0 on tässä tapauksessa käytössä oleva verkkokortti. Kun DNS palvelimen osoite on muutettu, voidaan kone liittää toimialueeseen. Se onnistuu käyttämällä komentoa ”*Add-Computer -DomainName samk.local -Restart*”. Avautuva ikkuna pyytää vielä antamaan

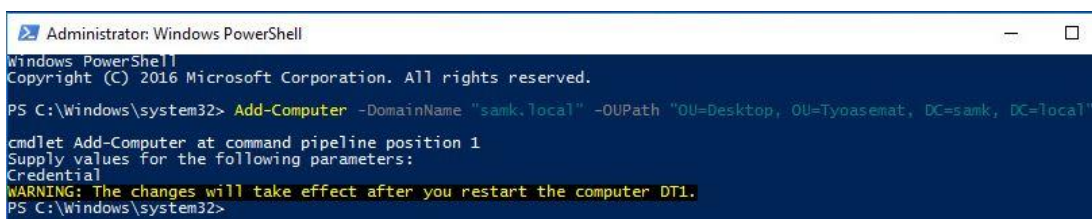
toimialuepalvelimen ylläpitäjän käyttäjätunnuksen ja salasanan (Kuva 18) jonka jälkeen kone käynnistyy uudelleen.



Kuva 18. Tunnistus liityttäessä uuteen toimialueeseen.

Nyt konetili on siirtynyt aktiivihakemistoon, mutta koska sille ei ollut sitä varten nimettyä konetiliä, aktiivihakemisto loi siitä objektin automaattisesti. Tämä objekti on mennyt organisaatioyksikköön nimeltä computers, joten siirrämme sen serverit yksikköön.

Jos halutaan liittää koneen toimialueelle niin, että se menee myös oikeaan organisaatioyksikköön, muutetaan käskyä lisäämällä siihen polku oikeaan organisaatioyksikköön. (Kuva 19)



Kuva 19. Koneen lisääminen tiettyyn organisaatioyksikköön.

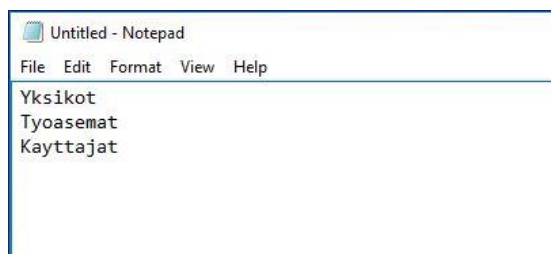
6 AKTIIVIAIKEMISTON HALLINTA

Tutustutaan seuraavaksi yleisimpien aktiivihakemistossa tehtävien töiden tekoon PowerShellin kautta.

6.1 Organisaatioyksiköiden luonti

Aktiivihakemistoon luodaan yhteensä viisi konetiliä, yksi palvelin ja neljä työpöytää. Näitä varten tarvitsee luoda organisaatioyksiköt. Domain Controller menee automaattisesti omaan Domain Controllers organisaatioyksikköön, joten sitä ei tarvitse muuttaa. Luodaan ensin vain yksi organisaatioyksikkö nimeltä serverit. PowerShellissä annetaan komento: `"New-ADOrganizationalUnit -Name "Serverit" -Path "DC=samk, DC=local" "`.

Voidaan myös luoda useampi yksikkö kerrallaan, mutta sitä ei voi tehdä yhden komennon sisällä. Sen suorittamiseksi täytyy siirtää tieto .csv tiedoston avulla, tämä voidaan tehdä esimerkiksi notepad -ohjelmalla. (kuva 20)



Kuva 20. Uudet organisaatioyksiköt.

Tämä tekstitiedosto tallennetaan esimerkiksi työpöydälle muotoon Yksikot.csv ja nämä tiedot voidaan ottaa käyttöön skriptillä:

```
"$ouPolku = "DC=samk, DC=local" $ouNimet = Import-CSV -Path
C:\Users\Administrator\Desktop\Yksikot.csv
foreach ($ou in $ouNimet) {
NewOrganizationalUnit -name $ou.Yksikot -Path $ouPolku
}"
```

```

PS C:\Users\Administrator> $ouPolku = "DC=samk, DC=local"
>> $ouNimet = Import-CSV -Path C:\Users\Administrator\Desktop\Yksikot.csv
>> foreach ($ou in $ouNimet) {
>> New-ADOrganizationalUnit -Name $ou.Yksikot -Path $ouPolku
>> }

```

Kuva 21. Useamman organisaatioyksikön luonti (Huom! Teksti on yhtämittainen)



Kuva 22. Uudet organisaatioyksiköt luotuna.

Aliyksiköiden luonti organisaatioyksiköiden sisään onnistuu myös käyttämällä .csv tiedostoa apuna, sekä miltei samaa skriptiä. Poluksi muutetaan haluttu organisaatioyksikkö, tässä tapauksessa ”Tyoasemat” sekä vaihdetaan .csv tiedostoa, joka sisältää määritetyt aliyksiköt. (Kuva 23)

```

PS C:\Users\Administrator> $ouPolku = "OU=Tyoasemat, DC=samk, DC=local"
>> $ouNimet = Import-CSV -Path C:\Users\Administrator\Desktop\Aliyksikko.csv
>> foreach ($ou in $ouNimet) {
>> New-ADOrganizationalUnit -Name $ou.Aliyksikko -Path $ouPolku
>> }
PS C:\Users\Administrator>

```

Kuva 23. Uusien aliyksiköiden luominen.

6.2 Konetilien luonti

Konetilejä voidaan luoda myös manuaalisesti aktiivihakemistoon. Tämä on hyödyllistä esimerkiksi siinä vaiheessa, kun liitettävänä on useita laitteita. Konetilejä

voidaan myös käyttää uudelleen. Esimerkiksi käyttäjä A saa uuden laitteen niin voidaan tämän käyttäjän vanha konetili siirtää yksikköön ”Vapaat”. Tätä esimerkkiä varten luodaan hakemistoon uusi organisaatioyksikkö nimeltä ”Vapaat”. Tähän vapaat yksikköön luodaan kolme kappaletta työasemien konetunnuksia. koska luodaan taas useampia objekteja kerralla, tarvitsee käyttää jälleen .csv tiedostoa.

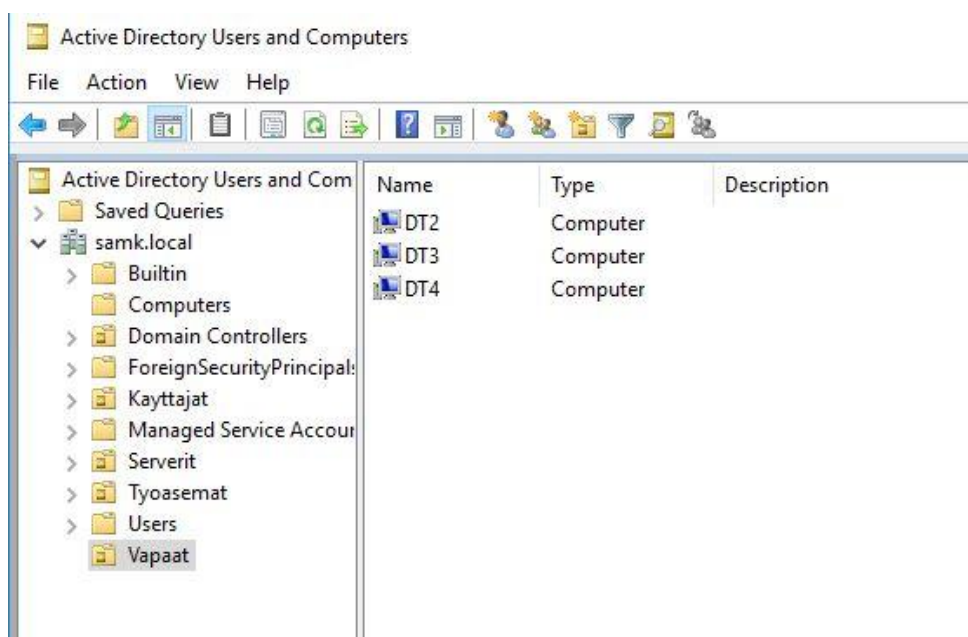
```
PS C:\Users\Administrator> Import-Csv .\Desktop\Koneet.csv
Koneet
-----
DT2
DT3
DT4
PS C:\Users\Administrator>
```

Kuva 24. Konetilit .csv tiedostossa

Konetilit saadaan luotua komennolla:

”Import-Csv-Path C:\Users\Administrator\Desktop\Koneet.csv | ForEach-Object {New-ADComputer -Name \$_.Koneet -Path ”OU=vapaat, DC=samk, DC=local” -Enabled \$True}”.

Komennon ajamisen jälkeen voidaan tarkistaa aktiivihakemistosta konetilit ”Vapaat” organisaatioyksiköstä. (Kuva 25)



Kuva 25. Uudet konetilit luotuna.

6.3 Käyttäjätilien luonti

Uusien käyttäjätilien luominen ja hallinnointi on yksi yleisimpiä ylläpitäjien töitä. Uudet käyttäjät tarvitsevat käyttöönsä tilin, jotta he pääsevät kirjautumaan toimialueeseen. Uusia käyttäjiä voi tulla myös useita kerrallaan esimerkiksi yrityskauppojen yhteydessä.

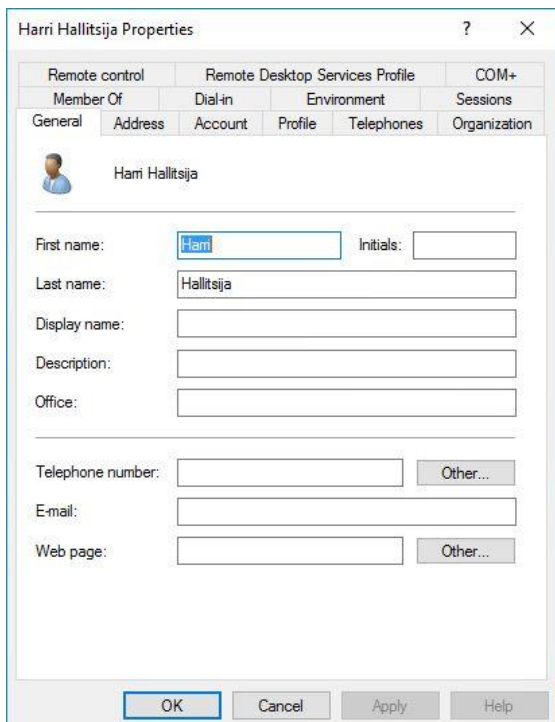
Yksittäisten tilien luonti onnistuu helposti käyttöliittymän kautta, mutta voidaan niitä luoda myös PowerShellillä niin yhden käyttäjän kohdalla, kuin myös useita tilejä yhdellä kerralla. Käyttäjätilien hallintaan on käytössä neljä eri käskyä, joista valitaan sopivin:

- *"New-ADUser"*
Luo uuden käyttäjän.
 - *"Get-ADUser"*
Palauttaa yhden tai useamman käyttäjän tiedot.
 - *"Set-ADUser"*
Mahdollistaa muutokset tilin ominaisuuksiin.
 - *"Remove-ADUser"*
Poistaa halutun tilin hakemistosta.
- (Tulloch 2015)

Tähän asti on käytetty oletuksena ollutta ylläpitäjän tiliä. Luodaan seuraavaksi yksittäinen tili, jota voidaan käyttää ylläpitäjän roolissa. Komento uuden käyttäjän luomiseen on:

```
"New-ADUser -Name "Harri Hallitsija" -GivenName "Harri" -Surname "Hallitsija"
-SamAccountName "Harhal" -UserPrincipalName "Harhal@samk.local" -
AccountPassword(Read-Host -AsSecureString "Anna Salasana") -Enabled $True"
```

Tämä käyttäjätili menee oletuksena users organisaatioyksikköön ja tämä tili kuuluu ainoastaan domain users ryhmään.



Kuva 26. Uusi käyttäjätili luotuna.

Nyt voidaan PowerShell istunnossamme tarkistella tämän käyttäjän perustiedot komennolla: ”*Get-ADUser (tilin nimi)*”.

```
PS C:\Users\Administrator> Get-ADUser Harhal

DistinguishedName : CN=Harri Hallitsija,CN=Users,DC=samk,DC=local
Enabled           : False
GivenName        : Harri
Name             : Harri Hallitsija
ObjectClass      : user
ObjectGUID       : 498b3c12-37ee-443d-b069-35499cff3fc9
SamAccountName   : Harhal
SID              : S-1-5-21-815748525-1409384458-622254336-1112
Surname          : Hallitsija
UserPrincipalName : Harhal@samk.local

PS C:\Users\Administrator>
```

Kuva 27. Käyttäjän perustiedot

Jotta uudella ylläpitäjän tilillä olisi myös oikeuksia tehdä muutoksia järjestelmätasolla, täytyy tämä käyttäjä siirtää ylläpidolle kuuluviin ryhmiin. Ensin tarkistetaan mihin ryhmiin oletuksen oleva ylläpitäjä tili kuuluu. Tämä onnistuu komennolla: ”*Get-ADPrincipalGroupMembership Administrator | Select name*”. Komento hakee tiedot tilin nimellä ja sitten putkituksen avulla valitsee vain ryhmien nimet. (Kuva 28)

```
PS C:\Users\Administrator> Get-ADPrincipalGroupMembership Administrator | select name
name
----
Domain Users
Administrators
Schema Admins
Enterprise Admins
Domain Admins
Group Policy Creator Owners

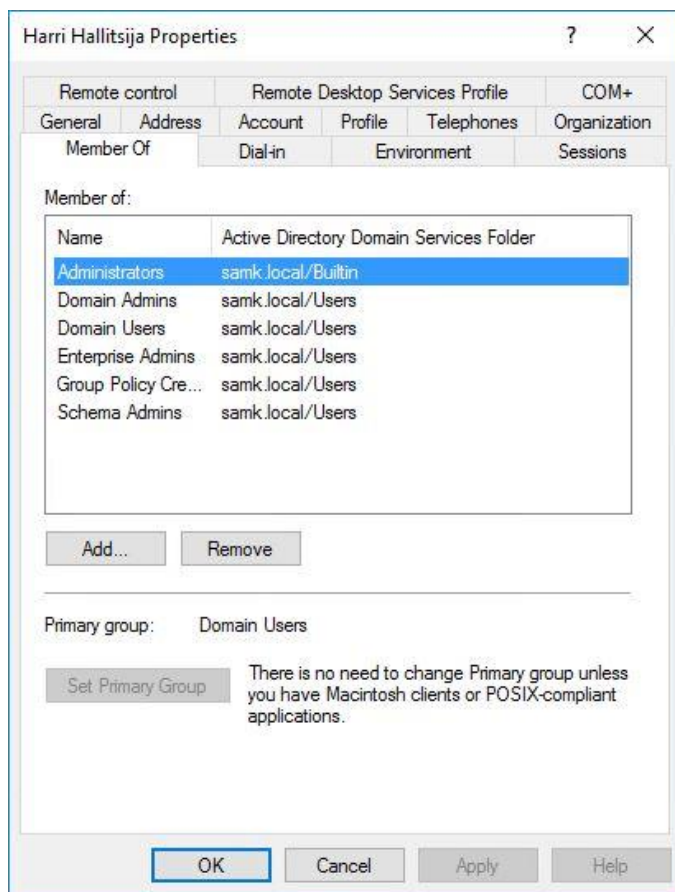
PS C:\Users\Administrator> _
```

Kuva 28. Ryhmät, joihin ylläpitäjä kuuluu.

Nyt kun tiedetään ryhmät, voidaan uusi ylläpitäjän tili liittää näihin ryhmiin, jotta saadaan lisää oikeuksia käyttäjälle. Ryhmien lisääminen käyttäjälle tapahtuu komennolla:

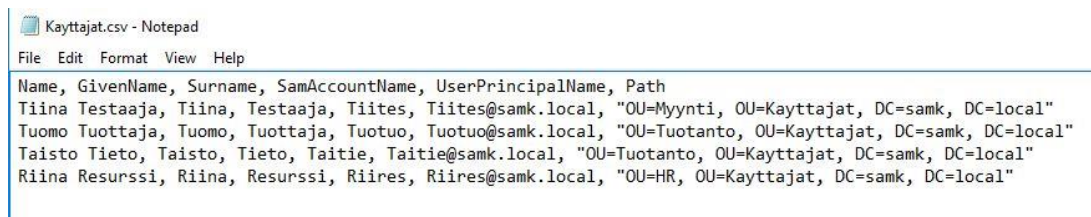
```
"Add-ADPrincipalGroupMembership -Identity Harhal -MemberOf "Administrators",
"Schema Admins", "Enterprise Admins", "Domain Admins", "Group Policy Creator
Owners""
```

Ryhmien tarkistaminen tapahtuu, joko aikaisemmalla *"Get-ADPrincipalGroupMemberShip"* komennolla tai käyttöliittymän kautta. (Kuva 29)



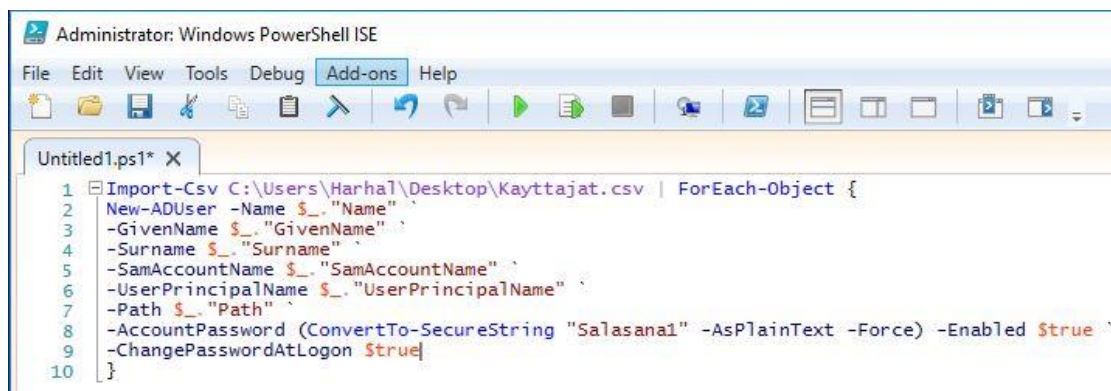
Kuva 29. Käyttäjä lisättyinä ryhmiin.

Useamman käyttäjätilin luomisessa tarvitsee jälleen käyttää .csv tiedostoa. Ensin syötetään kaikkien lisättävien käyttäjien tiedot käyttämällä notepad -ohjelmaa. (Kuva 30)



Kuva 30. Uudet käyttäjät listattuna .csv tiedostoon

Nyt näitä tietoja voidaan hyödyntää kirjoittamalla skripti, joka prosessoi tiedot ja luo neljä uutta käyttäjätiliä. Käyttäjätilit siirtyvät automaattisesti oikeisiin organisaatioyksiköihin koska niille on määritelty .csv tiedostossa polku, joten niitä ei tarvitse mennä jälkikäteen siirtelemään.



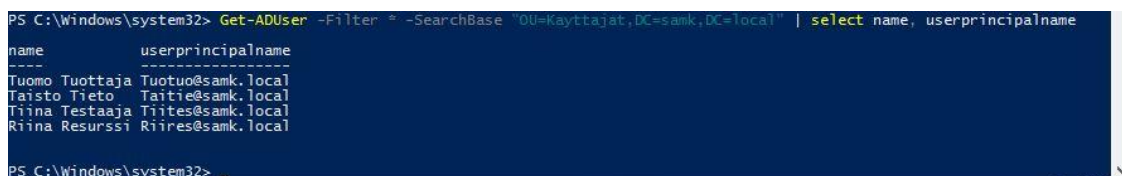
```

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1* X
1 Import-Csv C:\Users\Harha\Desktop\Kayttajat.csv | ForEach-Object {
2   New-ADUser -Name $_.Name `
3   -GivenName $_.GivenName `
4   -Surname $_.Surname `
5   -SamAccountName $_.SamAccountName `
6   -UserPrincipalName $_.UserPrincipalName `
7   -Path $_.Path `
8   -AccountPassword (ConvertTo-SecureString "Salasana1" -AsPlainText -Force) -Enabled $true `
9   -ChangePasswordAtLogon $true
10 }

```

Kuva 31. Usean käyttäjätilin luominen skriptin avulla.

Voimme vielä tarkistaa uudet tilimme komennolla: `"Get-ADUser -Filter * -SearchBase "OU=Kayttajat,DC=samk,DC=local" | select name, userprincipalname"`. (Kuva 32)



```

PS C:\Windows\system32> Get-ADUser -Filter * -SearchBase "OU=Kayttajat,DC=samk,DC=local" | select name, userprincipalname
name userprincipalname
----
Tuomo Tuottaja Tuotto@samk.local
Taimo Tieto Taitie@samk.local
Tiina Testaaja Tiites@samk.local
Riina Resurssi Riites@samk.local
PS C:\Windows\system32>

```

Kuva 32. Uudet käyttäjät listattuna.

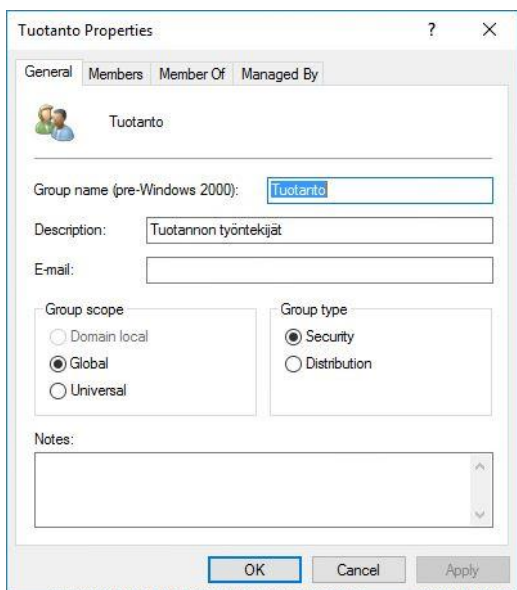
6.4 Ryhmän luonti

Ryhmien avulla voidaan erotella käyttäjien oikeuksia. Uusien ryhmien luomiseen ja hallitsemiseen on käytössä neljä komentoa:

- `"New-ADGroup"`
Tämän avulla luodaan uusi ryhmä.
- `"Get-ADGroup"`
Palauttaa yhden tai useamman ryhmän tiedot.
- `"Set-ADGroup"`
Voidaan modifioida ryhmän ominaisuuksia.
- `"Remove-ADGroup"`
Poistaa halutun ryhmän hakemistosta.
- `"Add-ADGroupMember"`
Lisää halutut käyttäjät ryhmään.

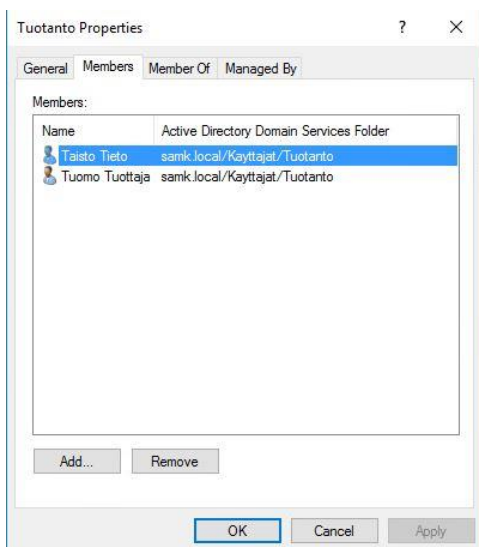
Luodaan ryhmä nimeltä ”Tuotanto”. Tämä onnistuu komennolla:

```
”New-ADGroup -Name ”Tuotanto” -SamAccountName Tuotanto -GroupCategory
Security -GroupScope Global -DisplayName ”Tuotanto” -Path
”CN=Users,DC=local,DC=samk” -Description ”Tuotannon työntekijät””.
```



Kuva 33. Uusi ryhmä luotuna.

Tähän ryhmään voidaan lisätä käyttäjiä komennolla: ”Add-ADGroupMember -Identity Tuotanto -Members Taitie, Tuotuo”.



Kuva 34. Käyttäjät lisättyinä uuteen ryhmään.

7 LOPUKSI

Tässä työssä tutustuttiin PowerShellin perusasioihin eli historiaan, versioihin sekä tutkittiin muutamia käytännön asioita ja miten niitä hyödynnetään ja käytetään. Tutustuttiin myös Windows Server 2016 käyttöjärjestelmään, rooleihin ja aktiivihakemistoon.

Powershellin käyttö nyt ja tulevaisuudessa on hyvä taito niin ylläpitäjille kuin myös tehokäyttäjille. Sitä ei tarvitse osata käyttää kaikessa vaan jo pienikin osaaminen riittää joidenkin tehtävien toistuvaan ajoon ja yksinkertaistamiseen. Koska PowerShell on nyt järjestelmäriippumaton, avaa se uusia ovia eri käyttäjille. Uskon, että PowerShellin käyttö tulee tulevaisuudessa kasvamaan. Välttämättä emme enää näe uutta versiota, joka on tarkoitettu vain Windows -laitteille vaan kehitys jatkuu juuri tästä järjestelmäriippumattomasta PowerShell Core 6.0:sta.

Työn rajaamiseen olisi tarvinnut käyttää enemmän aikaa, koska jokainen työn osa-alueista on niin suuri, että rajaaminen tuntui ajoittain vaikealta. Tämä on siitä huolimatta mielestäni onnistunut peruspaketti PowerShellin käytöstä aktiivihakemistossa ja palvelinympäristössä. Opin työtä tehdessä paljon lisää PowerShellin käytöstä. Tulen itse jatkossa käyttämään PowerShelliiä niin kokeilu- ja harjoittelumielessä kotona, kuin myös tulevissa työpaikoissa, jos sen käyttö vain suinkin on mahdollista.

LÄHTEET

Aiello, J. 2018. Powershell Core 6.0: Generally available (GA) and supported! Viitattu 21.3.2018

<https://blogs.msdn.microsoft.com/powershell/2018/01/10/powershell-core-6-0-generally-available-ga-and-supported/>

Bertram, A. 2015. Powershell functions introduction. Viitattu 21.12.2018.

<https://www.adamtheautomator.com/powershell-functions/>

Branscombe, M 2018. Microsoft's Powershell Core offers cross-platform automation. Viitattu 21.3.2018 <https://thenewstack.io/cross-platform-open-source-powershell-goes-net-core-runs/>

Hoffman, C. 2018. What is a CSV file, and how do I open it? Viitattu 12.1.2019

<https://www.howtogeek.com/348960/what-is-a-csv-file-and-how-do-i-open-it/>

Jones, D. & Hicks, J. 2017. Learn windows Powershell in a month of lunches. USA: Manning.

Krause, J. 2016. Mastering Windows Server 2016. UK: Packt Publishing ltd.

Microsoft 2009. Roles, Role Services, and Features. Viitattu 6.1.2019

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc754923\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc754923(v=ws.11))

Microsoft Docs 2017 a. Getting started with Windows Powershell. Viitattu

17.3.2018. <https://docs.microsoft.com/en-us/powershell/scripting/getting-started/getting-started-with-windows-powershell>

Microsoft Docs 2017 b. New features in Powershell 3.0. Viitattu 19.3.2018.

<https://docs.microsoft.com/en-us/powershell/scripting/whats-new/what-s-new-in-windows-powershell-50?view=powershell-6>

Microsoft Docs 2017 c. New features in Powershell 4.0. Viitattu 19.3.2018.

<https://docs.microsoft.com/en-us/powershell/scripting/whats-new/what-s-new-in-windows-powershell-50?view=powershell-6>

Microsoft Docs 2017 d. What's new in Powershell 5.0. Viitattu 19.3.2018.

<https://docs.microsoft.com/en-us/powershell/scripting/whats-new/what-s-new-in-windows-powershell-50?view=powershell-6>

Microsoft Docs 2018 a. What's new in PowerShell core 6.0. Viitattu 21.3.2018.

<https://docs.microsoft.com/en-us/powershell/scripting/whats-new/what-s-new-in-powershell-core-60?view=powershell-6>

Microsoft Docs 2018 b. Powershell Core support life cycle. Viitattu 21.3.2018

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

- Microsoft Docs 2018 c. About pipelines. Viitattu 25.3.2018.
https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_pipelines?view=powershell-6
- Microsoft Support 2017 a. Windows Powershell 1.0 localized installation package for windows server 2003 and for window xp. Viitattu 19.3.2018.
<https://support.microsoft.com/en-us/help/926140/windows-powershell-1-0-localized-installation-package-for-windows-serv>
- Microsoft Support 2017 b. Windows management framework (Windows Powershell 2.0, WinRM 2.0, and BITS 4.0). Viitattu 19.3.2018.
<https://support.microsoft.com/en-us/help/968929/windows-management-framework-windows-powershell-2-0-winrm-2-0-and-bits>
- Microsoft TechNet 2012. Windows PowerShell profiles. Viitattu 25.3.2018.
[https://technet.microsoft.com/en-us/library/bb613488\(v=vs.85\)](https://technet.microsoft.com/en-us/library/bb613488(v=vs.85))
- Microsoft TechNet 2016, Setting up Active Directory via PowerShell. Viitattu 7.1.2019. <https://blogs.technet.microsoft.com/uktechnet/2016/06/08/setting-up-active-directory-via-powershell/>
- Pietroforte, M. 2013. 10 Reasons for using Powershell ISE instead of the Powershell console. Viitattu 25.3.2018. <https://4sysops.com/archives/10-reasons-for-using-powershell-ise-instead-of-the-powershell-console/>
- Ring, J.E 2016. #PSTip new Powershell commands in Windows 10 Anniversary update. Viitattu 21.3.2018 <http://www.powershellmagazine.com/2016/08/02/pstip-new-powershell-commands-in-windows-10-anniversary-update/>
- Rouse, M. 2017. Microsoft Windows Server OS (operating system). Viitattu 22.1.2019. <https://searchwindowsserver.techtarget.com/definition/Microsoft-Windows-Server-OS-operating-system>
- Rouse, M. 2018. Active Directory. Viitattu 2.1.2019.
<https://searchwindowsserver.techtarget.com/definition/Active-Directory>
- Siddaway, R. 2012. Powershell workflow: The basics. Viitattu 20.3.2018
<https://blogs.technet.microsoft.com/heyscriptingguy/2012/12/26/powershell-workflows-the-basics/>
- Snover, Jeffrey. P 2002. Monad Manifesto. (Verkkajulkaisu) Viitattu 17.3.2018 saatavana: <https://blogs.msdn.microsoft.com/powershell/2007/03/18/monad-manifesto-the-origin-of-windows-powershell/>
- Tulloch, M. 2015. Creating Active Directory accounts using PowerShell. Viitattu 18.1.2019 <http://techgenix.com/creating-active-directory-accounts-using-powershell/>