

Alexi Frimodig

Verkonvalvonnan kehittäminen, Seinäjoen kaupunki/ Tietohallinto

Opinnäytetyö

Kevät 2017

SeAMK Tekniikka

Tietotekniikan tutkinto-ohjelma

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Seinäjoen ammattikorkeakoulu

Tutkinto-ohjelma: Tietotekniikan koulutusohjelma

Suuntautumisvaihtoehto: Tietoverkkotekniikka

Tekijä: Aleksi Frimodig

Työn nimi: Verkonvalvonnan kehittäminen, Seinäjoen kaupunki / Tietohallinto

Ohjaaja: Alpo Anttonen

Vuosi: 2017

Sivumäärä: 60

Opinnäytetyön tavoitteena oli parantaa Seinäjoen kaupungin Tietohallinnon verkonvalvontaa. Tietohallinto oli toteuttanut ja käyttöönottanut verkonvalvontajärjestelmän jo ennen työn aloittamista, mutta kaikkia sen ominaisuuksia ei oltu vielä hyödynnetty. Verkonvalvontaohjelmana toimi Zabbix, joka on yrityskäyttöön soveltuva, avoimeen lähdekoodiin perustuva ja ilmainen ohjelma.

Työ toteutettiin erillisessä demoympäristössä VMware-virtualisointiohjelman avulla. Demoympäristön fyysisiin laitteisiin kuului Windows 10 -käyttöjärjestelmällä toimiva tietokone ja Ciscon kytkin. Tietokoneeseen asennettiin virtualisointiympäristö, johon asennettiin Linuxin Ubuntu-palvelin. Ubuntu-palvelimelle asennettiin Zabbix-ohjelmisto. Myös Windows Server 2012 -virtuaalipalvelin asennettiin virtualisointiympäristöön, jotta testattavia laitteita saatiin lisää.

Työn lopussa demoympäristö ja verkonvalvonnasta tutkittavat ominaisuudet saatiin toimimaan hyvin ja tutkittavien ominaisuuksien käyttöönotosta tehtiin ohjeistus Tietohallinnolle. Tutkittavia ominaisuuksia olivat tietokantojen valvonta, automaattisesti generoidut raportit halutuista kohteista ylläpitäjän sähköpostiin, sekä mount point -kansioiden valvonta.

Avainsanat: verkonvalvontajärjestelmä, Zabbix, avoin lähdekoodi, demoympäristö, VMware-virtualisointiohjelma, Ubuntu, Windows

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Network Technology

Author: Aleksi Frimodig

Title of thesis: Enhancing the network monitoring for the City of Seinäjoki / information administration

Supervisor: Alpo Anttonen

Year: 2017

Number of pages: 60

The aim of this thesis was to enhance the network monitoring of the information administration of the City of Seinäjoki. The information administration had already deployed and implemented a network monitoring system before the thesis was started but all its features had not been exploited. As the network monitoring system was Zabbix which is a program that is based on open-source, is suited for enterprises and comes at no cost.

The thesis was implemented in a separated demo environment with the VMware virtualization program. The demo environment's physical machines consisted of a desktop that was running with a Windows 10 operating system and a Cisco Catalyst 3560-CG Series PoE network switch. The virtualization environment was installed to the desktop and in it there was installed a Linux Ubuntu server, which included the network monitoring program. A Windows Server 2012 virtual server was also installed to the virtualization environment so that there were more machines for testing.

At the end of the thesis the demo environment and the researched features of the monitoring system were functioning fine and an instruction manual concentrating on the researched features was written for the information administration. The features researched were database monitoring, monitoring of mount points, and automatically generated reports of the desired objects to the admin's email.

Keywords: network monitoring system, Zabbix, open-source, demo environment, VMware virtualization program, Ubuntu, Windows

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	1
Thesis abstract.....	2
SISÄLTÖ.....	3
Kuvaluettelo	5
Käytetyt termit ja lyhenteet	7
1 JOHDANTO	9
1.1 Työn tausta	9
1.2 Työn tavoite	9
1.3 Työn rakenne	9
2 VERKONVALVONTA.....	11
2.1 Verkonvalvonnan tarkoitus.....	11
2.2 SNMP ja sen toimintaperiaate.....	12
2.3 SNMPv1	13
2.4 SNMPv2.....	13
2.5 SNMPv3.....	14
2.6 ICMP	14
2.7 Trap-viestit	15
3 ZABBIX-VERKONVALVONTAJÄRJESTELMÄ JA SEN TOIMINNALLISUUS.....	16
3.1 Yleistä	16
3.2 Zabbix-palvelin	16
3.3 Visualisointi	17
3.3.1 Web-käyttöliittymä (Frontend).....	17
3.3.2 Dashboard	18
3.3.3 Kuvaajat.....	19
3.4 Datankeräys.....	19
3.4.1 Agentti.....	20
3.4.2 SNMP-tuki.....	20
3.4.3 Tietokantavalvonta.....	20
3.5 Auto-discovery	21

3.6	Template-mallit.....	21
3.7	Item-kohteet	21
3.8	Trigger-määritelmät.....	22
3.9	Tapahtumien ilmoitukset	22
4	YMPÄRISTÖT JA TEKNIIKAT	23
4.1	VMware Workstation Pro	23
4.2	Ubuntu	23
4.3	Perl 5.....	23
4.4	Cron	24
4.5	Mount point	24
4.6	ODBC.....	25
4.7	FreeTDS	25
5	TYÖN TOTEUTUS.....	26
5.1	Työn suunnittelu, aloitus ja eteneminen	26
5.2	Demoympäristön asentaminen	27
5.3	Automaattinen raportin lähetys sähköpostiin.....	42
5.4	Tietokantavalvonnan asennus.....	46
5.5	Mount point -kansioiden valvonta.....	50
6	YHTEENVETO JA POHDINTA	58
	LÄHTEET	59

Kuvaluettelo

Kuva 1. Trap-viestin toiminta.....	15
Kuva 2. Dashboard-näkymä.	18
Kuva 3. Kuvaaja tietokoneen suorittimen kuormituksesta.....	19
Kuva 4. Säännöt ajanmääritykseen Crontabissa.	24
Kuva 5. VMware-virtualisointiohjelman verkkoasetukset.	27
Kuva 6. Ohjelmakokoelmat.	28
Kuva 7. Zabbix-palvelimen verkkoasetukset.....	29
Kuva 8. Alkumääritysten tarkistus.....	32
Kuva 9. Tietokantayhteyden asetukset.	32
Kuva 10. Zabbix-palvelimen tiedot.....	33
Kuva 11. Yhteenveto asetuksista.....	33
Kuva 12. Asennus valmis.....	34
Kuva 13. Web-käyttöliittymän kirjautumisnäkyä	35
Kuva 14. Dashboard-osio	35
Kuva 15. Action-määritelmävalikko.....	37
Kuva 16. Conditions-määritelmät.....	38
Kuva 17. Operations-määritelmät.	38
Kuva 18. SNMP-konfigurointitiedosto.	39
Kuva 19. SNMP-laitteen lisääminen.....	40
Kuva 20.Template SNMP-laitteelle.	41

Kuva 21. Demoympäristön valvottavat laitteet Hosts-näkymässä.....	42
Kuva 22. Kuvaaja valvotun laitteen tilasta ylläpitäjän sähköpostissa.	46
Kuva 23. TCP-liikenteen avaaminen SQL Server Configuration Managerissa.	47
Kuva 24. Item-kohteen teko tietokantavalvontaa varten.	49
Kuva 25. Tietokantavalvontaa varten tehdyn item-kohteen palauttama arvo Latest data -välilehdellä.	50
Kuva 26. New Simple Volume Computer Management -osiossa.....	51
Kuva 27. Levytilan koon valitseminen.	52
Kuva 28. Mount point -kansion polku.	53
Kuva 29. Levyn asetukset.	54
Kuva 30. Test-kansion levykuvake.....	55
Kuva 31. Item-kohteen teko Mount point -kansion valvontaa varten.....	56
Kuva 32. Mount point -kansion item-kohteen palauttama arvo Latest data -välilehdessä.	57

Käytetyt termit ja lyhenteet

API	Application programming interface, on määritelmä, jonka mukaan eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja keskenään.
Ping	TCP/IP-protokollan työkalu, jolla voidaan kokeilla kohdelaitteen saatavuutta. Ping-kehote lähettää kohdelaitteelle ICMP echo request -paketin, johon kohdelaite vastaa omalla echo reply -paketilla.
Polling	Elektronisessa kommunikaatiossa polling/pollaus on jatkuvaa kyselyä toisen ohjelman tai laitteen tilasta.
Trap	Ilmoitus (Trap), jonka avulla valvonta-agentti voi kertoa valvonta-asemalle standardissa määritellyistä merkittävistä verkon tapahtumista.
UDP	User Datagram Protocol, protokolla, joka ei vaadi yhteyttä laitteiden välille, mutta mahdollistaa tiedostojen siirron.
MIB	Management Information Base, tietokanta, jota käytetään viestintäverkon kokonaisuuksiin.
IP	Internet Protocol, TCP/IP-mallin internetkerroksen protokolla, joka huolehtii IP-tietoliikennepakettien toimittamisesta perille pakettikytkentäisessä internetverkossa.
TCP	Transmission Control Protocol, tietoliikenneprotokolla, jolla luodaan yhteyksiä tietokoneiden välille, joilla on pääsy internetiin.
RFC	RFC:t (Request for Comments) ovat IETF-organisaation (Internet Engineerin Task Force) julkaisemia internetiä koskevia standardeja.
Host	Tietokone tai muu laite, mikä on yhdistetty tietoverkkoon.

Frontend	Käyttöliittymä käyttäjän ja backendin (palvelimen) välissä.
Script	Tietokoneohjelma, joka on kirjoitettu script-kielellä.
VIM	Tekstieditori Unix-käyttöjärjestelmille.
Proxy	Välityspalvelin eli välipalvelin varastoi ja suodattaa verkossa siirrettäviä tiedostoja.
UNIX	Unix on laitteistoriippumaton käyttöjärjestelmä.
Terminaali	Terminaali (tai komentorivi) on yksi Linux-järjestelmän tehokkaimmista työkaluista, josta voi suorittaa lähes kaikki mahdolliset tehtävät antamalla komentoja.
Schema	Teknologia, jolla voidaan kuvata XML-dokumenttien rakenne.
Partitio	Osio kiintolevyssä, mikä on eristetty muista segmenteistä.
MySQL	Suosittu relaatiotietokantaohjelmisto.
Apache	Avoimeen lähdekoodiin perustuva http-palvelinohjelma.

1 JOHDANTO

1.1 Työn tausta

Verkonvalvonnan avulla voidaan ennaltaehkäistä yrityksessä monia IT-infrastruktuurin ongelmia. Verkonvalvontaohjelma asennetaan fyysiselle tai virtuaaliselle palvelimelle ja se yleensä sisältää käyttöliittymän, jonka avulla valvontaohjelmaa voidaan käyttää myös muilla työasemilla selaimen välityksellä. Verkonvalvontaohjelmaan voidaan esimerkiksi määrittää hälytysrajat verkon tapahtumista. Kun verkonvalvontaohjelman hälytysrajat laukeavat, tulee viesti ylläpitäjän sähköpostiin automaattisesti ohjelman toimesta. Hälytyksiä voi tehdä esimerkiksi yrityksen palvelimen kiintolevyn levytilasta tai verkkolaitteiden vastaamisesta pingaukseen.

Seinäjoen kaupungin Tietohallinto oli jo käyttöönottanut Zabbix-verkonvalvontajärjestelmän, mutta lisää sen ominaisuuksia haluttiin hyödyntää. Työn tekeminen erillisessä demoympäristössä oli hyvä ratkaisu, sillä ilman oikeita laitteita ei ollut vaaraa tehdä vahinkoa järjestelmään ja opettelu oli sujuvaa.

1.2 Työn tavoite

Tavoitteena on saada kattavampi verkonvalvonta Tietohallinnon IT-infrastruktuurille. Työssä käyttöönotetaan Zabbix-verkonvalvontajärjestelmä erilliseen demoympäristöön ja tutkitaan ohjelman ominaisuuksia tietokantavalvonnan, automaattisen raportoinnin ja mount point -kansoiden valvonnan osalta. Ohjelman käyttöönotto opeteltiin Zabbix-järjestelmän nettisivuilta käyttöönottomanuaalista, sekä erilaisten foorumeiden ja Youtube-videoiden avulla.

1.3 Työn rakenne

Luvussa 2 kerrotaan verkonvalvonnasta yleisesti, sekä siinä käytettävistä protokollista.

Luku 3 käsittelee Zabbix-verkonvalvontajärjestelmää yleisesti ja sen toiminnallisuutta.

Luku 4 sisältää tietoa opinnäytetyön tekoon käytetyistä ympäristöistä ja tekniikoista.

Luvussa 5 kerrotaan työn suunnittelusta ja sen aloituksesta, sekä työn toteutuksesta.

Luku 6 sisältää pohdintaa ja yhteenvedon opinnäytetyöstä.

2 VERKONVALVONTA

2.1 Verkonvalvonnan tarkoitus

Verkonvalvonta auttaa verkko- ja järjestelmäylläpitäjiä tunnistamaan mahdolliset ongelmat, ennen kuin ne ilmenevät. Verkonvalvonta auttaa löytämään perimmäisen syyn ongelmiin, kun jokin menee vikaan verkossa. (SolarWinds [Viitattu 2.4.2017].)

Jotta verkonvalvonnasta olisi oikeaa hyötyä yritykselle, valvonnan toteutuksen tulisi mukailla peruseriaatteita. Valvonnan tulisi kattaa verkkoyhteydet, järjestelmät ja turvallisuus. Hyvä valvonta antaa myös tietoa verkosta ja sisältää raportoinnin, virheetunnistuksen, ratkaisun niihin ja verkon ylläpidon. (SolarWinds [Viitattu 2.4.2017].)

Verkonvalvonnan perusominaisuuksiin kuuluu datankeräys verkkolaitteista ja datan esittäminen ymmärrettävässä muodossa. Tämä voidaan käsittää raportoinniksi. Se auttaa verkon ylläpitäjää ymmärtämään verkon suorituskyvyn, sen nykyisen tilan ja mikä on tavanomaista toimintaa kyseiselle verkolle. Datan ja raportoinnin avulla ylläpitäjä voi tehdä perusteltuja päätöksiä kapasiteettisuunnitelmista, verkon kunnossapidosta, vianetsinnästä ja verkon turvallisuudesta. (SolarWinds [Viitattu 2.4.2017].)

Raportointi pelkästään ei auta ylläpitäjää pitämään korkeaa suorituskykyä verkossa. Toinen tärkeä vaatimus on kyky tunnistaa, mikä voi mennä pieleen verkossa. Raja-arvoihin perustuvat hälytykset ja trigger-määritelmät auttavat ylläpitäjää tunnistamaan mahdolliset verkon ongelmat, ennen kuin ne vievät verkon alas. (SolarWinds [Viitattu 2.4.2017].)

Verkonvalvonta voidaan toteuttaa agenteilla tai ilman niitä. Agentti on ohjelma, joka asennetaan valvottavaan laitteeseen, ja sillä on pääsy laitteen suorituskyvyn dataan. Verkonvalvontajärjestelmä tekee pyynnön datasta, jolloin se saa vastauksen, ja data lähetetään järjestelmälle laitteelta. Agentin avulla laitetta voidaan valvoa paremmin, sillä sille voidaan antaa yksityiskohtaisempia toimintoja ja pyyntöjä datasta. (SolarWinds [Viitattu 2.4.2017].)

Ilman agenttia pystytään myös valvomaan verkkoa. SNMP-protokolla on yleisin valvontatapa tällä saralla. Ilman agenttia toimivassa valvonnassa etuna on, että kohdelaitteelle ei tarvitse asentaa agenttiohjelmistoa. Tämän johdosta on alemmat huolto- ja käyttöönottokulut ja ilman agenttia toimivassa valvonnassa on lähes olematon vaikutus laitteen suoritukseen nähden, sillä sovellusta ei ole pyörimässä taustalla. Ilman agenteja laitteilta ei kuitenkaan saada yhtä kattavasti informaatiota, kuin agenteihin perustuvassa valvonnassa. (SolarWinds, [Viitattu 2.4.2017].)

2.2 SNMP ja sen toimintaperiaate

SNMP-protokolla määrittää, kuinka valvontainformaatio vaihtuu verkonvalvontaohjelmien ja valvonta-agenttien välillä. Verkonvalvontaohjelma pollaa jaksoittain SNMP-agenteja kyselemällä dataa, joka sijaitsee valvottavissa laitteissa. Jaksottaisessa SNMP-pollauksessa huonona puolena on, että siinä on viivästys tapahtuman ilmenemisen ja sen välillä, kun verkonvalvontajärjestelmä huomaa sen. Pollaustajuuden ja kaistanleveyden välillä käydään vaihtokauppa. Verkonvalvontaohjelma voi näyttää informaation verkonvalvonnan käyttöliittymässä. SNMP käyttää tähän UDP-protokollaa saadakseen ja lähettääkseen valvontainformaatiota, kuten MIB-suu-reita. (Froom, Sivasubramanian & Frahim 2010, 269.)

SNMP-valvonta-agentit, jotka sijaitsevat valvottavissa laitteissa, keräävät ja tallentavat informaatiota laitteesta ja sen operaatiosta, vastaavat valvontakyselyihin ja generoivat trap-viestejä informoidakseen ylläpitäjää määritellyistä tapahtumista. SNMP-trap-viestit lähetetään valvonta-agenttien toimesta verkonvalvontajärjestelmään, kun tietty tapahtuma ilmenee. Trap-ohjattu-ilmoitus voi saada aikaan huomattavat säästöt verkon ja agenttien resursseissa eliminoimalla joidenkin SNMP-kyselyjen tarpeellisuuden. (Froom, Sivasubramanian & Frahim 2010, 269.)

2.3 SNMPv1

SNMP-standardin alkuperäisversio SNMPv1 on määritelty RFC 1157 -standardissa. Verkonvalvonta käyttää seuraavia tavallisia SNMP-viestejä siirtääkseen dataa agenteilta, jotka sijaitsevat valvottavissa laitteissa:

- **Get Request -pyyntö:** Verkonvalvonta-aseman pyyntö, jota käytetään pyydettäessä tietyn MIB-suureen arvoa agentilta.
- **Get Next Request -pyyntö:** Käytetään alkuperäisen Get Request -pyynnön jälkeen, jotta verkonvalvontaohjelma saa takaisin seuraavan kohdetapauksen listasta.
- **Set Request -pyyntö:** Verkonvalvontaohjelma käyttää tätä komentoa MIB-suureen asettamiseen agentille.
- **Get Response -vastaus:** Käytetään agenttien toimesta, jotta ne saavat vastattua Get Request- tai Get Next Request -pyyntöihin verkonvalvonnalta.
- **Trap-viestit:** Valvonnassa olevan laitteen agentti huomauttaa verkonvalvontaa tilastaan ilman erillistä pyyntöä. (Froom, Sivasubramanian & Frahim 2010, 270.)

2.4 SNMPv2

SNMPv2 esiteltiin RFC 1441 -standardin kanssa, mutta IETF-järjestön alakomitean jäsenet eivät voineet hyväksyä SNMPv2:n turvallisuus- ja hallintaosioita. Tehtiin useita yrityksiä, jotta saavutettaisiin SNMPv2:n hyväksyntä kokeellisten muokattujen versioiden toimesta. Yhteisöön pohjautuva SNMPv2, määriteltynä RFC 1901 -standardissa, on yleisin toteutus. SNMPv2 esittelee kaksi uutta viestityyppiä:

- **Get Bulk Request -pyyntö:** Pyytää suurempia tietomääriä kerralla ja vähentää toistuvia pyyntöjä ja vastauksia, mikä parantaa suorituskykyä.

- **Inform Request -pyyntö:** Hälyttää SNMP-valvontaa tietyistä tiloista. Toisin kuin SNMP-Trap-viestit, jotka ovat vahvistamattomia, verkonvalvontajärjestelmä vahvistaa Inform Request -pyynnön lähettämällä Inform Response -vastauksen takaisin sitä pyytävälle laitteelle. (Froom, Sivasubramanian & Frahim 2010, 270., 271.)

2.5 SNMPv3

SNMPv3 on kuvailtu RFC 3410 -standardista 3415-standardiin. Se lisää tapoja varmistaa kriittisen datan turvallisen siirron valvottavien laitteiden välillä. SNMPv3 esittelee kolme turvallisuusluokkaa:

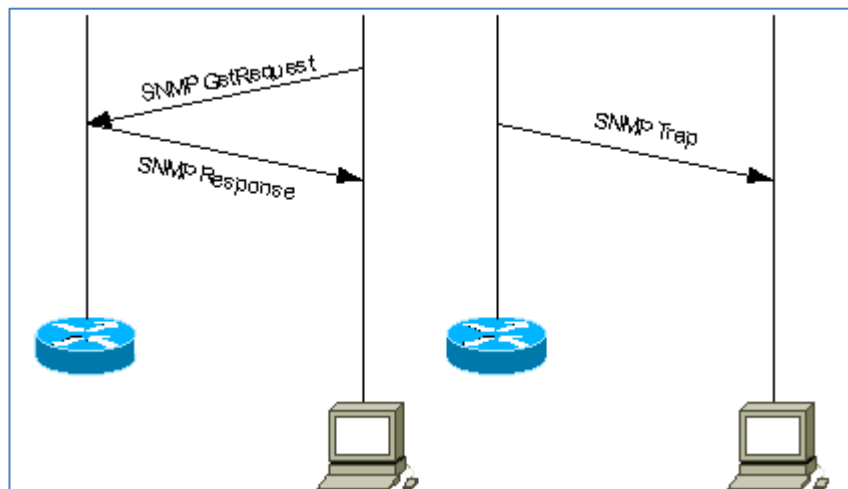
- **noAuthNoPriv:** Tunnistautumista ei tarvita ja yksilöllisyyttä ei tarvita
- **authNoPriv:** Tunnistautuminen pohjautuu tarkisteviestillä tulevaan varmistuskoodiin Message Digest 5:n (HMAC-MD5) kanssa tai tarkisteviestillä tulevaan varmistuskoodiin Secure Hash Algorithmin (HMAC-SHA) kanssa. Salausta ei vaadita.
- **authPriv:** Vaihtoehtoisesti todennukselle, Cipher Block Chaining-Data Encryption Standard (CBC-DES) salausta käytetään yksityisyysprotokollana. (Froom, Sivasubramanian & Frahim 2010, 271.)

2.6 ICMP

ICMP-protokollaa (Internet Control Message Protocol) käytetään verkon laitteiden valvontaviestien välityksessä. ICMP-protokollan avulla voidaan tutkia pakettien kuluaikoja verkossa, mutta yleisin käyttötarkoitus on pakettien takaisinkaiutus verkkoa testatessa. Ping-kehote on tarkoitettu tämän protokollan käyttöliittymäksi. ICMP-protokollan avulla siis saadaan tieto, onko jokin valvottava laite päällä verkossa. (Hakala & Vainio 2005, 297.)

2.7 Trap-viestit

Idea Trap-viestiin pohjautuvien hälytysten takana on, että jos ylläpitäjä on vastuussa monesta laitteesta, joista jokaisessa on paljon valvottavia kohteita, on ylläpitäjälle mahdotonta manuaalisesti kysellä kaikkien kohteiden tilaa jokaisesta laitteesta. Ratkaisuna jokaisen valvonnassa olevan laitteen agentti huomauttaa ylläpitäjää tilastaan ilman erillistä pyyntöä. Se tekee tämän lähettämällä viestin, jota kutsutaan tapahtuman trap-viestiksi. (Cisco 2006.)



Kuva 1. Trap-viestin toiminta.
(Cisco 2006)

3 ZABBIX-VERKONVALVONTAJÄRJESTELMÄ JA SEN TOIMINNALLISUUS

3.1 Yleistä

Zabbix on yritystason ohjelma, joka on suunniteltu valvomaan IT-infrastruktuurin laitteiden saatavuutta ja suorituskykyä. Ohjelma perustuu avoimeen lähdekoodiin ja on ilmainen. Zabbix-ohjelman loi Alexei Vladishev ja sitä tuetaan, sekä kehitetään jatkuvasti Zabbix SIA:n toimesta. (Zabbix [Viitattu 27.3.2017].)

Zabbix-ohjelma valvoo lukuisia parametreja verkossa, sekä palvelimien kuntoa ja ehjyyttä. Ohjelma käyttää joustavaa hälytysjärjestelmää, mikä antaa käyttäjien konfiguroida sähköpostipohjaisia hälytyksiä virtuaalisesti mistä tahansa tapahtumasta. Tämä mahdollistaa nopean reaktion palvelinongelmiin. Zabbix tarjoaa myös raportointi- ja datavisualisoinnin, jonka ominaisuudet pohjautuvat tallennettuun dataan. (Zabbix [Viitattu 27.3.2017].)

Zabbix tukee polling- ja trapping-kyselyitä. Kaikki ohjelman raportit ja statistiikat ja konfiguraatioparametrit ovat saatavissa web-pohjaisessa käyttöliittymässä. Web-pohjainen käyttöliittymä mahdollistaa verkon tilan tutkimisen kaikista sijainneista. Zabbix sopii niin isoihin kuin pieniinkin yrityksiin ja oikein konfiguroituna sillä on suuri merkitys yrityksen IT-infrastruktuurissa. (Zabbix [Viitattu 27.3.2017].)

3.2 Zabbix-palvelin

Zabbix-palvelin suorittaa pollauksen ja trap-viestit datalle, laskee trigger-määritelmät ja lähettää ilmoitukset käyttäjille. Se on keskeinen komponentti, jonne Zabbix-agentit ja proxyt raportoivat dataa järjestelmien kunnosta ja saatavuudesta. Palvelin pystyy itsenäisesti tarkistamaan verkossa olevia palveluita käyttäen yksinkertaisia palvelutarkistuksia. Zabbix-palvelimen käytettävyys voidaan jakaa kolmeen osaan:

Zabbix-palvelimeen, web-käyttöliittymään ja tietokantavarastoon. (Zabbix [Viitattu 25.2.2017].)

Kaikki Zabbix-konfigurointi-informaatio on tallennettu tietokantaan, jonka kanssa palvelin ja web-käyttöliittymä ovat vuorovaikutuksessa. Esimerkiksi kun tehdään uusi item-kohde web-käyttöliittymän puolella, se lisätään items-taulukkoon tietokannassa. Noin kerran minuutissa Zabbix-palvelin kysyy items-työkalulta listaa itemkohteista, jotka ovat aktiivisia, ja sitten tallentuvat välimuistiin Zabbix-palvelimessa. (Zabbix [Viitattu 25.2.2017].)

Turvallisuusvaatimuksista ja palvelimen kriittisten tehtävien luonteesta johtuen UNIX on ainoa käyttöjärjestelmä, joka voi jatkuvasti ylläpitää tarvittavan suorituskyvyn, häiriön sietokyvyn ja joustavuuden. Zabbix on testattu seuraavissa alustoissa: Linux, Solaris, AIX, HP-UX, Mac OS X, FreeBSD, OpenBSD, NetBSD, SCO Open Server ja Tru64/OSF1. (Zabbix [Viitattu 25.2.2017].)

3.3 Visualisointi

Kun on suuria määriä dataa, visualisointi helpottaa sen hahmottamista paremmin, kuin pelkkä lukuarvojen käyttö. Zabbix tarjoaa selkeän käyttöliittymän asioiden hallintaan, sekä kuvaajia halutusta datasta. (Zabbix [Viitattu 28.3.2017].)

3.3.1 Web-käyttöliittymä (Frontend)

Zabbix-ohjelman web-käyttöliittymä on sen virallinen käyttöliittymä, joka tarjoaa turvallisen pääsyn noviisikäyttäjille, sekä suuren mittakaavan konfigurointivaihtoehtoja suuriin asennuksiin. Web-käyttöliittymään pääsee antamalla selaimen URL-kenttään Zabbix-palvelimen IP-osoitteen ja perään /zabbix. (Zabbix [Viitattu 26.3.2017].)

3.3.2 Dashboard

Web-käyttöliittymän Monitoring->Dashboard-välilehtiosio on yhteenveto kaikista tärkeistä informaatioista. Tämä on aloitussivu, kun kirjaudutaan web-käyttöliittymään. (Zabbix [Viitattu 24.3.2017].)

The screenshot displays the Zabbix Dashboard interface. The top navigation bar includes 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. The main dashboard area is divided into several sections:

- Favourite maps:** Local network
- Favourite graphs:** New host: CPU load
- Favourite screens:** Zabbix server
- Last 20 issues:** A table listing recent issues with columns for Host, Issue, Last Change, Age, Info, Ack, and Actions.

HOST	ISSUE	LAST CHANGE	AGE	INFO	ACK	ACTIONS
New host	CPU load too high on New host for two minutes	2016-02-12 08:50:19	22s		No	1
New host	New host has just been restarted	2016-02-12 08:47:59	2m 42s		No	1
Zabbix server 1	Zabbix server 1 has just been restarted	2016-02-12 08:46:31	4m 10s		No	1
Zabbix server 1	Lack of free swap space on Zabbix server 1	2015-08-11 23:29:28	6m 4d 10h		Yes 4	
- Status of Zabbix:** A table showing system parameters.

PARAMETER	VALUE	DETAILS
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled/templates)	54	10 / 1 / 43
Number of items (enabled/disabled/not supported)	356	350 / 0 / 6
Number of triggers (enabled/disabled/problem/ok)	95	94 / 1 / 4 / 90
Number of users (online)	3	2
Required server performance, new values per second	4.79	
- System status:** A table showing the status of various host groups across different severity levels.

HOST GROUP	DISASTER	HIGH	AVERAGE	WARNING	INFORMATION	NOT CLASSIFIED
Clouds	0	0	0	0	0	0
Database servers	0	0	0	0	0	0
Discovered hosts	0	0	0	1	1	0
JB applications	0	0	0	0	0	0
Linux servers	0	1	0	0	1	0
Network devices	0	0	0	0	0	0
SNMP hosts	0	0	0	0	0	0
Virtual machines	0	0	0	0	0	0
Web servers	0	0	0	0	0	0
Windows servers	0	0	0	0	0	0
Zabbix servers	0	0	0	1	1	0
- Discovery status:** A table showing the status of discovery rules.

DISCOVERY RULE	UP	DOWN
Local network2	19	1
- Web monitoring:** A table showing the status of web monitoring for different host groups.

HOST GROUP	OK	FAILED	UNKNOWN
Discovered hosts	1	0	0
Zabbix servers	1	0	0

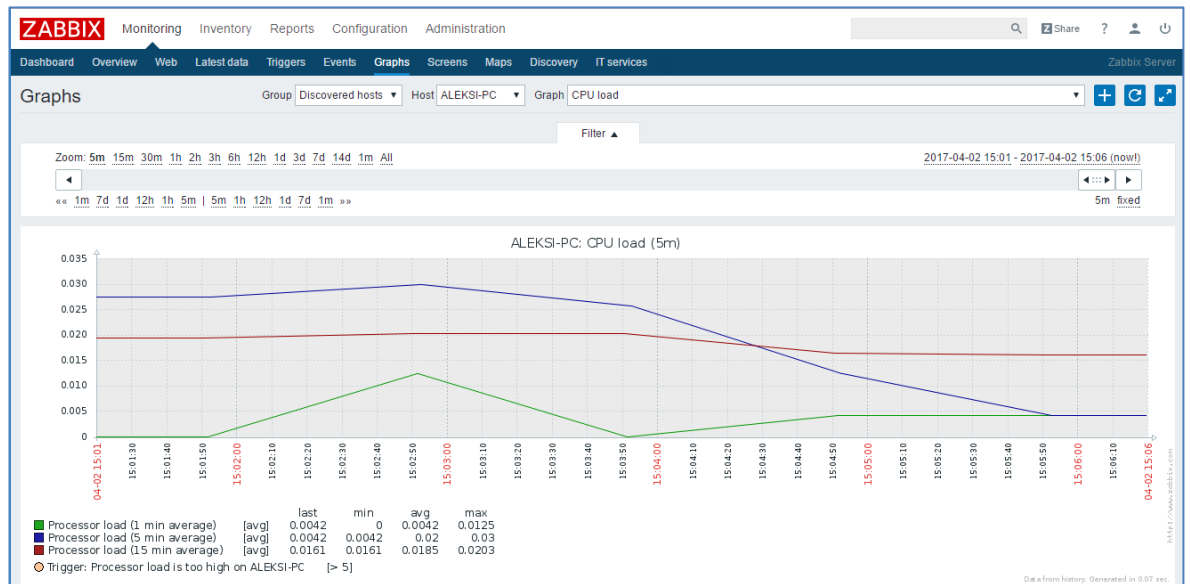
Kuva 2. Dashboard-näkymä. (Zabbix [Viitattu 24.3.2017])

Suosikkeihin (kuvassa 2 vasemmalla) voi luoda nopeita oikopolkuja tärkeimpiin kuvaajiin, monitoreihin, diaesityksiin ja karttoihin (Zabbix [Viitattu 24.3.2017]).

Tilawidgetit: Zabbix-ohjelman tila, Järjestelmän tila, Host-laitteiden tila, Viimeiset 20 ongelmaa, Web-valvonta ja Discovery-tila esittävät yhteenvedon kullekin kuuluvasta datasta (Zabbix [Viitattu 24.3.2017]).

3.3.3 Kuvaajat

Zabbix pystyy luomaan kuvaajia jo kerätystä ja tallennetusta datasta. Tavalliset kuvaajat numeerisista arvoista ovat käytettävissä ilman konfigurointia, sillä ne on luotu ajonaikaisesti. Kuvaajia pääsee tarkastelemaan Monitoring-välilehden Graphs-osi-ossa. Kaikkien kuvaajien näyttämälle datalle voidaan muokata ajanjaksoa, mistä ne esittävät tietoa. (Zabbix [Viitattu 23.3.2017].)



Kuva 3. Kuvaaja tietokoneen suorittimen kuormituksesta.

Mukautetut kuvaajat ovat tehokkaampia, kuin yksinkertaiset tavalliset kuvaajat. Mukautetuissa kuvaajissa pystytään vertailemaan useiden item-kohteiden dataa ja kuvaajan tyyliä, sekä ulkoasua voidaan muokata. Näitä voidaan luoda yhdelle tai useammalle host-laitteelle tai yhteen template-malliin. (Zabbix [Viitattu 23.3.2017].)

3.4 Datankeräys

Zabbix-järjestelmällä on paljon tekniikoita yhden tai useampien laitteiden valvontaan.

3.4.1 Agentti

Zabbix-agentti on sijoitettu valvottavaan kohteeseen, jotta voidaan aktiivisesti valvoa paikallisia resursseja ja palveluita. Agentti kerää operatiivista informaatiota paikallisesti ja raportoi datan Zabbix-palvelimelle pidempään käsittelyyn. Virheiden varalta Zabbix-palvelin voi aktiivisesti hälyttää virheen antaman koneen ylläpitäjää. (Zabbix [Viitattu 22.3.2017].)

Zabbix-agentit voivat suorittaa passiivisia ja aktiivisia tarkistuksia. Passiivisessa tarkistuksessa agentti vastaa datakyselyyn. Zabbix-palvelin kysyy dataa ja agentti palauttaa sille vastauksen. Aktiiviset tarkistukset vaativat mutkikkaampaa käsittelyä. Agentin pitää ensin vastaanottaa lista item-kohteista Zabbix-palvelimelta itsenäistä käsittelyä varten. Sitten se jaksottaisesti lähettää uudet arvot palvelimelle. (Zabbix [Viitattu 22.3.2017].)

Agentit ovat saatavilla seuraaville alustoille: Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Max OS X, Solaris 9, 10, 11, Windowsin kaikki pöytäkone- ja palvelinversiot 2000-luvun jälkeen (Zabbix [Viitattu 22.3.2017]).

3.4.2 SNMP-tuki

Zabbix-palvelin pystyy keräämään dataa laitteilta SNMP-agenttien v1-, v2- ja v3-versioilla. SNMP-valvontaa käytetään pääasiassa tulostimille, verkkokytkimille, reitittimille tai UPS-varavirtalähteille, sillä näille olisi vaikeaa asentaa agentti-ohjelmistoja. Zabbix-palvelimeen pitää asettaa oikeat asetukset SNMP-tuelle, ja SNMP-liikenne pitää sallia valvottavasta kohteesta, jotta valvonta alkaa toimia oikein. SNMP-kyselyt suoritetaan pelkästään UDP-protokollaa käyttäen. (Zabbix [Viitattu 21.3.2017].)

3.4.3 Tietokantavalvonta

ODBC-tekniikan ja sen ajurien avulla Zabbix-palvelin voi kerätä mitä tahansa dataa DRBMS (Relational Database Management System) -tietokannoista, kuten

MySQL-, PostgreSQL-, Oracle- ja Microsoft Sql Server -tietokannoista. Kun kysely on suoritettu, sen palautusarvo on tallennettu. Tämä mahdollistaa kuvaajien luonnin ja hälytykset sekä huomautukset virheistä suorituskäytössä ja vastaamattomuudessa pingaukseen. Tämä ominaisuus tekee valvonnasta tehokkaampaa, sillä se kerää informaatiota suoraan tietokannan datasta, välttäen vääriä kokonaisuuksia. (Zabbix [Viitattu 20.3.2017].)

3.5 Auto-discovery

Zabbix tarjoaa automaattisen verkonetsimistöiminnön, mikä on järkevä määrittää, jos on paljon lisättäviä kohteita valvonnan piiriin. Tämän avulla ohjelma löytää itse laitteet ja lisää ne valvonnan piiriin ilman käyttäjän toimenpiteitä. Discovery-sääntöihin voi määrittää IP-osoitealueen, josta ohjelma hakee laitteita. Laitteisiin pitää määrittellä oikeat asetukset ennalta, jotta ohjelma osaa ne tunnistaa ja lisätä ne valvonnan piiriin. (Zabbix [Viitattu 19.3.2017].)

3.6 Template-mallit

Template-mallien käyttö tekee Zabbix-ohjelman ylläpidosta helpompaa. Joukko kokonaisuuksia (itemeitä, trigger-määritelmiä, kuvaajia, sovelluksia, screenejä, discovery-sääntöjä ja web-skenaarioita) voidaan määrittää template-malliksi. Template-malli voidaan linkittää usealle host-laitteelle. Kaikki template-mallin sisältämät kokonaisuudet lisätään linkitetylelle host-laitteelle. Jos template-mallia muutetaan, myös siihen linkitettyjen host-laitteiden kokonaisuudet muuttuvat. (Zabbix [Viitattu 18.3.2017].)

3.7 Item-kohteet

Item-kohteet ovat datankeräyksen perusta Zabbix-ohjelmassa. Ilman item-kohteita ei ole dataa, koska vain item-kohteet määrittelevät yksittäiset mittarit tai mitä dataa kerätään host-laitteilta. Item-kohteeseen määritellään Item key -avaimen avulla, minkälaista dataa kerätään laitteelta. (Zabbix [Viitattu 17.3.2017].)

3.8 Trigger-määritelmät

Trigger-määritelmät ovat johdonmukaisia määritelmiä jotka arvioivat item-kohteiden keräämää dataa ja kuvailevat nykyistä järjestelmän tilaa. Kun item-kohteet keräävät dataa, ylläpitäjän olisi epäkäytännöllistä seurata niiden arvoja jatkuvasti, kunnes jokin näyttäisi hälyttävältä. Trigger-määritelmä mahdollistaa raja-arvon määrittelyn millä arvoilla data on kelvollinen. Kun datan arvo ylittää kelvolliseksi määritellyn raja-arvon, trigger-määritelmä laukaistaan. (Zabbix [Viitattu 16.3.2017].)

3.9 Tapahtumien ilmoitukset

Action-toimenpiteet koostuvat tiloista ja operaatioista. Kun jokin tila on saavutettu, operaatio suoritetaan. Kaksi yleisintä operaatiota ovat ilmoituksen lähettäminen viestillä ja etäkomennon suorittaminen. Action-toimenpiteitä on järkevää käyttää, sillä olisi työlästä pitää kirjaa koko ajan trigger-määritelmistä ja käydä katsomassa niiden tilaa säännöllisesti. Action-toimenpiteet osaavat siis itsenäisesti antaa tiedon esimerkiksi ylläpitäjän sähköpostiin, kun jokin trigger-määritelmä on lauennut. Zabbix voi antaa myös ilmoitukset seuraavissa muodoissa: SMS, Jabber, Ez Texting ja mukautetussa hälytysscriptissä. (Zabbix [Viitattu 15.3.2017].)

4 YMPÄRISTÖT JA TEKNIIKAT

4.1 VMware Workstation Pro

VMware Workstation Pro on teollisuusstandardi useiden virtuaalikäyttöjärjestelmien ajamiseen yhdellä tietokoneella. Käyttäjät voivat kopioida palvelimen, työaseman ja tabletin ympäristöt virtuaaliseen koneeseen, jotta ohjelmia voidaan ajaa samanaikaisesti käyttöjärjestelmien välillä ilman uudelleen käynnistämistä. Workstation tuottaa myös turvallisen ja eristetyn ympäristön uusien systeemien, testiohjelmistojen ja päivitysten arvioimiseen. (VMware [Viitattu 20.2.2017].)

4.2 Ubuntu

Ubuntu on suosittu Linux-levitysversio. Sen perusti eteläafrikkalainen Mark Shuttleworth vuonna 2004. Ubuntu perustuu vapaaseen Debian-jakeluun. Ubuntusta on erikoisversioita palvelimille, OpenStack-pilville ja mobiililaitteille. Kaikki versiot jakavat keskenään saman infrastruktuurin ja ohjelman. (Ubuntu [Viitattu 20.2.2017].)

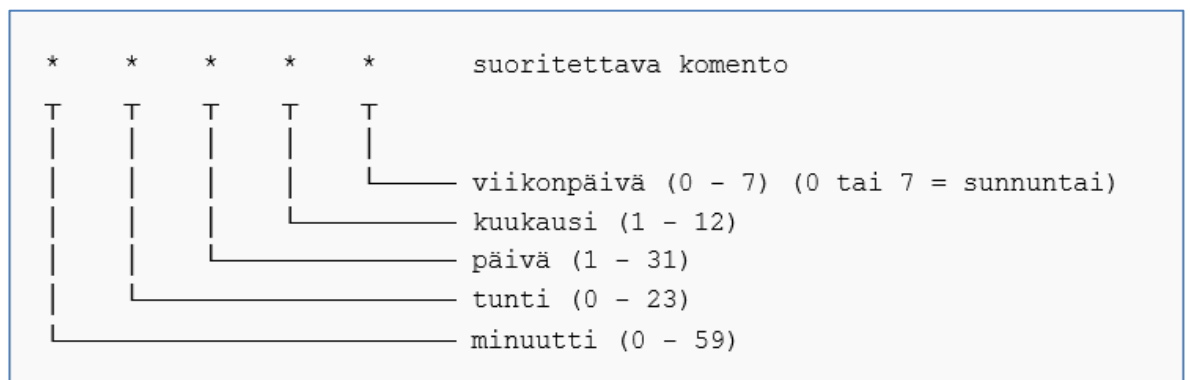
4.3 Perl 5

Perl (Practical Extraction and Report Language) on laajasti käytetty ohjelmointikieli, joka sopii pienistä projekteista suuriin sovelluskehityksiin. Sen yleisohjelmointipalvelut tukevat proseduurisia, toiminnallisia ja objekti kohdistuneita ohjelmointiajatusmalleja. (Perl [Viitattu 20.2.2017].)

Perl tukee Unicode-kieltä todella hyvin ja se voi käsitellä kaiken tyyppistä rakenteista tekstiä monilla laajennuskokoelmilla. CPAN (Comprehensive Perl Archive Network) -arkistoon kerätyt kirjastot tarjoavat valmiiksi tehtyjä ratkaisuja moneen ongelmaan. (Perl [Viitattu 20.2.2017].)

4.4 Cron

Cron-muoto on yksinkertainen, tehokas ja joustava tapa määrittää aika ja toistuvuus eri toimenpiteille Unix-pohjaisissa käyttöjärjestelmissä. Crontab on käyttäjien töiden ajastamiseen käytetty komento, joka käsittelee samannimistä asetustiedostoa. Crontabia pääsee säätämään komennolla `crontab -e`. Tällöin aukeaa editori, johon voidaan asettaa ajettava komento ja sen aikamääre. Crontabin editorissa pätee kuvan 4 mukaiset säännöt ajan määrittämiseen. (Linux.fi 2016.)



Kuva 4. Säännöt ajanmäärittämiseen Crontabissa. (Linux.fi 2016)

Esimerkiksi asettamalla editoriin `0 * * * * perl /usr/share/zabbix/scripts/graphreport.pl`, Cron ajaa kyseisen komennon automaattisesti jokaisen alkavan tunnin alussa.

4.5 Mount point

Mount point on paikallisesti saatavilla oleva linkki, johon otetaan yhteys ulkoisesta laitteesta. Windowsissa ja Mac OS X -käyttöjärjestelmissä mount point on esitetty levykuvana. (Indiana University 2013.)

Käytännössä siis tiedostokansiosta voidaan tehdä levykuvake, johon voidaan ottaa jälkeensä yhteys. Tämä on käytännöllistä, jos pitää tehdä niin monta levykuvaketta samaan järjestelmään, että levykuvakkeiden tunnuskirjaimet (C, D, E...) loppuvat kesken. Levykuvakkeita voidaan nimetä uniikeilla tiedostonimillä.

4.6 ODBC

Microsoft Open Database Connectivity (ODBC) -rajapinta on C-ohjelmointikielen rajapinta, joka mahdollistaa ohjelmien pääsyn eri tietokantojen valvontajärjestelmien dataan. ODBC on matalan tason korkeasuorituskykyinen rajapinta, joka on suunniteltu erityisesti relaatiotietokannoille. (Microsoft 2017.)

ODBC-rajapinta sallii maksimaalisen yhteen toimivuuden – ohjelma voi päästä käsiin dataan monimuotoisista tietokantojen hallintajärjestelmistä yhden rajapinnan kautta. Ohjelman käyttäjät voivat lisätä ohjelmistokomponentteja, joita kutsutaan ajureiksi, joiden liittymäkohta on ohjelman ja tietokannan hallintajärjestelmän välissä. (Microsoft 2017.)

4.7 FreeTDS

FreeTDS on kirjastokokoelma Unix- ja Linux-käyttöjärjestelmille. Se sallii ohjelmien keskustelun Microsoftin Sql Serverille ja Sybase-tietokantoihin. Teknisesti FreeTDS on TDS-protokollan avoimeen lähdekoodiin perustuva sovellus, joita tietokannat käyttävät niiden asiakasohjelmiin. Se tukee monia eri protokollan puolia ja kolmea API:a päästäkseen niihin. Lisäksi FreeTDS toimii muiden ohjelmien, kuten Perl-kielen ja PHP:n kanssa. (FreeTDS [Viitattu 20.2.2017].)

5 TYÖN TOTEUTUS

5.1 Työn suunnittelu, aloitus ja eteneminen

Aluksi Tietohallinnon kanssa sovittiin, mitä tutkitaan ja millä kokoonpanolla. Tutkitaviksi osa-alueiksi muodostui tietokantojen valvonta ja automaattisen tilanneraportin lähetyksen sähköpostitse. Loppuvaiheessa tutkittiin vielä, kuinka mount point -kansioita voidaan valvoa.

Työ tehtiin virtuaalisella palvelimella erillisessä demoympäristössä. Ohjelmistoversiot ja käyttöliittymät tarkistettiin samoiksi Tietohallinnon kanssa, jotta yhteensopivuusongelmia ei syntisi, kun tutkimus saataisiin valmiiksi. Demoympäristöön asennettiin lisäksi Cisco Catalyst 3560-CG Series PoE -kytkin, jotta oikeita fyysisiä laitteita saataisiin enemmän pöytäkoneen lisäksi. Demoympäristössä virtuaalipalvelimien tekoon käytettiin VMware Workstation 12 Pro -ohjelmistoa. Tällä tehtiin 64-bittinen Ubuntu 14.04 -palvelin, johon Zabbix-ohjelmisto asennettiin. Zabbix-palvelimen ohelle tehtiin myös yksi virtuaalinen Windows Server 12 -käyttöjärjestelmällä toimiva palvelin.

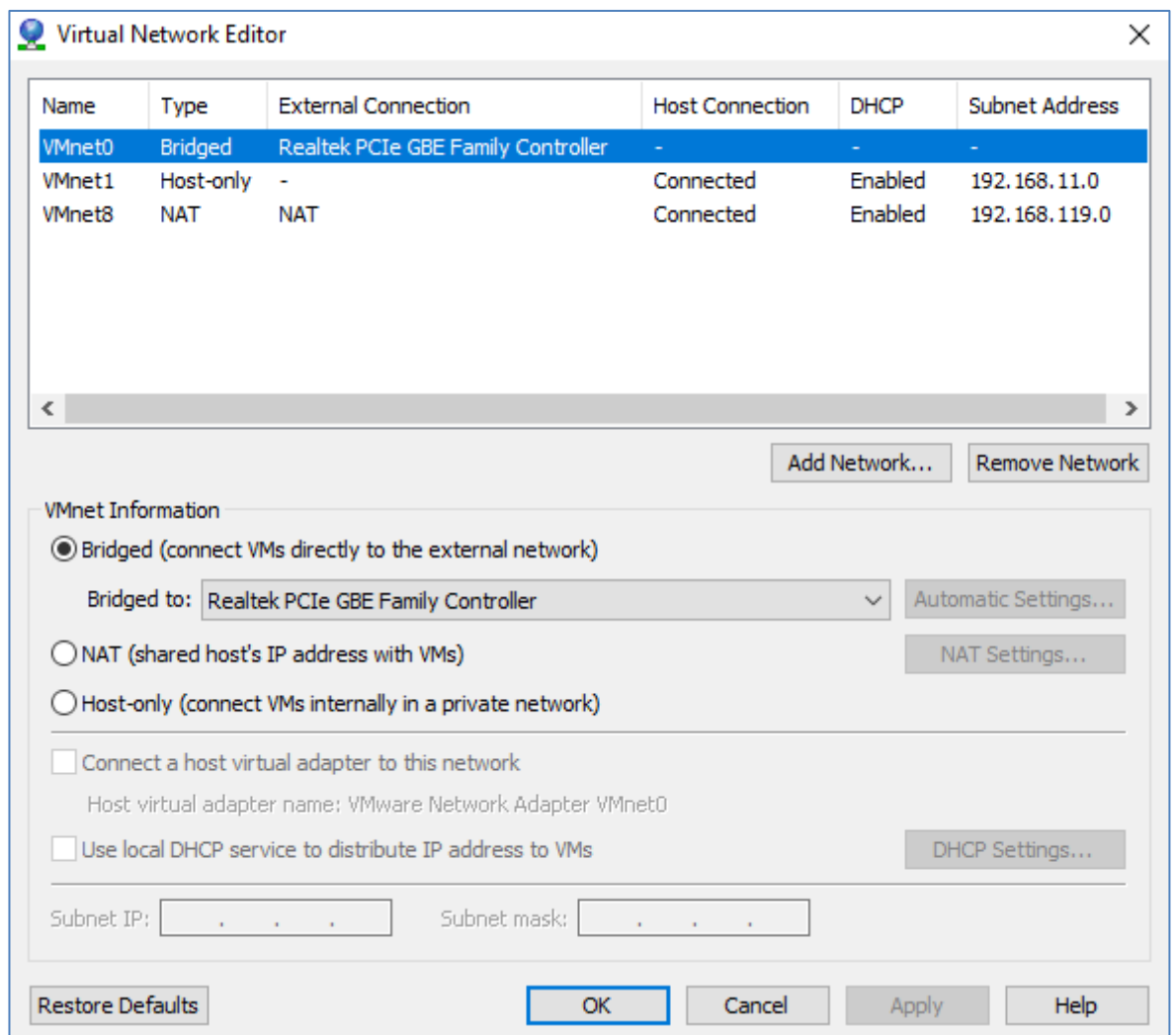
Työn alkuvaiheessa hankittiin paljon tietoa Zabbix-järjestelmää koskien internetistä ja muutenkin verkonvalvonnasta, sillä aihealue ei ollut tuttu entuudestaan. Hyväksi koettiin erilaiset Youtube-videot, joissa opetettiin Zabbix-järjestelmän käyttöä, sekä asennusta. Demoympäristö saatiin loppujen lopuksi hyvin toimimaan ja valvonta toimi halutuille laitteille, niin kuin pitikin.

Kun demoympäristö saatiin toimimaan, tutkittiin Tietohallinnon toivomia osa-alueita, joista olisi heille varsinaista hyötyä. Tietoa näihin löytyi erilaisilta foorumeilta ja Zabbix-ohjelman manuaalista. Työn edistyessä pidettiin yhteyttä toimeksiantajan kanssa ja keskusteltiin työstä. Ensimmäinen tutkittava osa-alue verkonvalvonnan ominaisuuksista oli tietokantojen valvonta. Tämä saatiin toteutettua ODBC-rajapinnan avulla. Valvonta kohdistui Windows Server 2012 -virtuaalipalvelimella sijaitsevaan Sql Server 2014 -tietokantaan. Toinen tutkittava aihe oli valvontaohjelman au-

tomaattinen raportointi ylläpitäjälle. Tämä onnistui Perl-scriptillä ja Cron-ajastuspalvelun avulla. Viimeisenä valvontaan lisättiin mount point -kansioiden valvonta, joka toteutettiin item-kohteen määrittämisen avulla web-käyttöliittymässä.

5.2 Demoympäristön asentaminen

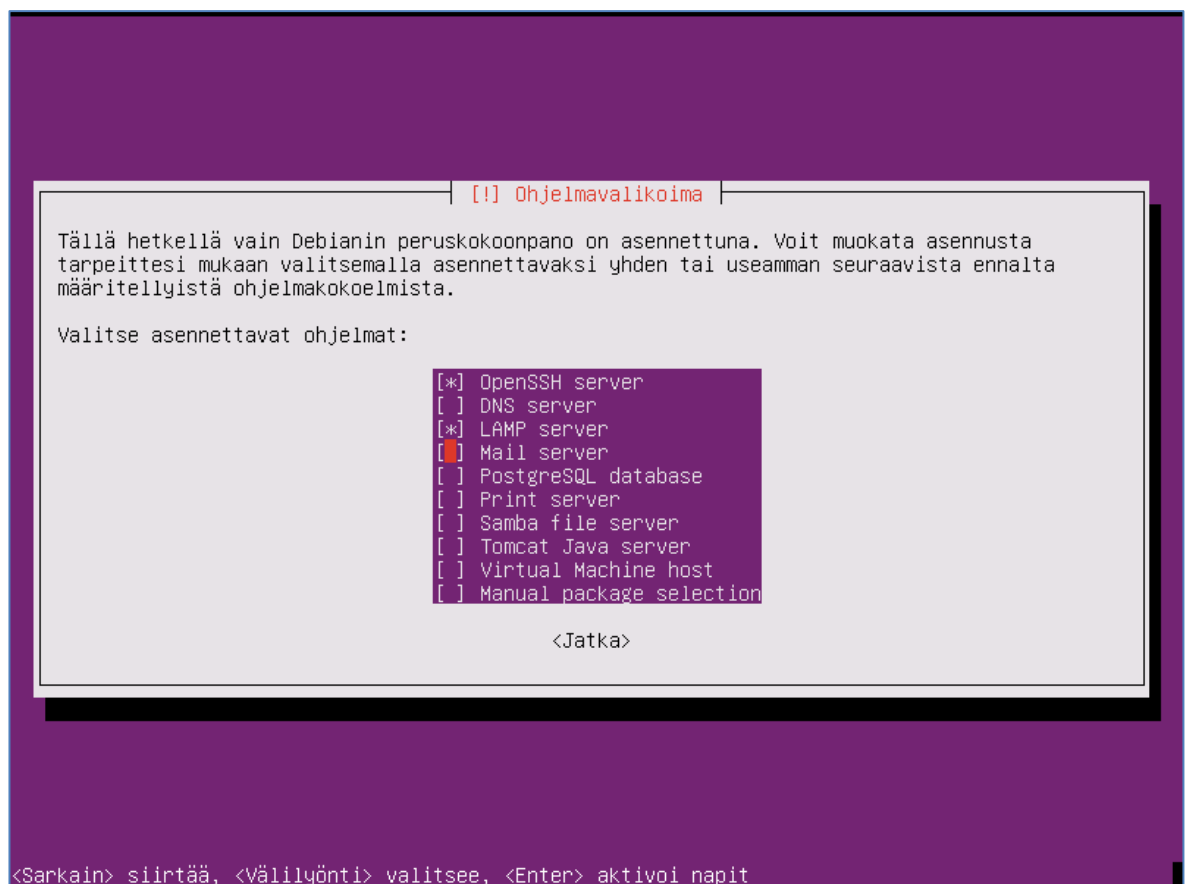
Työn ensimmäinen vaihe oli perustaa demoympäristö, jossa tutkimusta tehtäisiin. Keskeinen fyysinen laite oli tietokone, johon asennettiin VMware-virtualisointiohjelma ja liitettiin Ciscon Catalyst 3560-CG Series PoE -kytkin. VMwaren verkkoasetukset asetettiin käyttämään tietokoneen fyysistä verkkokorttia sillatulla yhteydellä.



Kuva 5. VMware-virtualisointiohjelman verkkoasetukset.

VMware-virtualisointiohjelmaan asennettiin Linux Ubuntu -palvelimen 14.04-versio. Ubuntu-palvelimeen asennettiin Zabbix-ohjelman 3.0-versio. Ohjelman asennuksessa ja käyttöönotossa piti määrittää oikeat verkkoasetukset ja konfiguraatio. Ubuntu sai ladattua ilmaiseksi sen nettisivulta. Asennus tapahtui ISO-levykuvan avulla, mitä VMware tukee.

Ubuntu-palvelimen asennusvaiheessa valittiin myös asennettavaksi LAMP server- ja OpenSSH server -paketit. LAMP-paketti sisältää Linux-käyttöjärjestelmän, Apachen, MySQL-tietokantarajapinnan ja PHP-, Perl- ja/tai Python-komentosarjakielen. OpenSSH-pakettia käytettiin etäyhteyden ottamiseen.



Kuva 6. Ohjelmakokoelmat.

Kun Ubuntu oli asentunut, ensimmäiseksi määriteltiin verkkoasetukset palvelimelle. Näihin päästiin antamalla terminaali-ikkunassa komento:

```
$ sudo vi /etc/network/interfaces
```


Seuraavana vuorossa oli MySQL-tietokantarajapinnan määrittäminen. Tämä asennettiin komennolla terminaali-ikkunassa:

```
$ apt-get install zabbix-server-mysql zabbix-frontend-php
```

Zabbix-tietokanta piti määrittää manuaalisesti MySQL-rajapinnalle. Ensiksi täytyi mennä MySQL-tietokantaan sisään terminaali-ikkunakomennolla:

```
$ mysql -u root -p <root-käyttäjän salasana>
```

MySQL-tietokannassa annettiin seuraavat komennot Zabbix-tietokannan luomiseen:

```
mysql> create database zabbix character set utf8 collate utf8_bin;

mysql> grant all privileges on zabbix.* to zabbix@localhost identified by
'<password>';

mysql> quit;
```

Tämän jälkeen tuotiin alustava schema ja data antamalla komento terminaali-ikkunaan:

```
$ zcat /usr/share/doc/zabbix-server-mysql/create.sql.gz | mysql -uzabbix -p
zabbix
```

Zabbix-palvelimen konfigurointiasetukseen piti muokata host, nimi, käyttäjä ja salasana. Konfigurointiasetukseen päästiin terminaali-ikkunassa komennolla:

```
$ vi /etc/zabbix/zabbix_server.conf
```

Avautuvaan tiedostoon kirjoitettiin seuraavat määritelmät:

```
DBHost=localhost

DBName=zabbix

DBUser=root

DBPassword=<salasana>
```

Konfiguroinnin jälkeen voitiin käynnistää palvelinprosessi komennolla terminaali-ikkunassa:

```
$ service zabbix-server start
```

Zabbix-palvelimelle tuli myös asettaa oikeat PHP-konfigurointiasetukset.

Näihin päästiin komennolla terminaali-ikkunassa:

```
$ sudo vi /etc/zabbix/apache.conf
```

Tähän konfigurointitiedostoon asetettiin seuraavat parametrit:

```
php_value max_execution_time 300  
php_value memory_limit 128M  
php_value post_max_size 16M  
php_value upload_max_filesize 2M  
php_value max_input_time 300  
php_value always_populate_raw_post_data -1  
php_value date.timezone Europe/Helsinki
```

Määritysten jälkeen tiedosto suljettiin ja terminaali-ikkunaan annettiin komento Apache2-palvelun uudelleenkäynnistämiseen:

```
$ service apache2 restart
```

Tämän vaiheen jälkeen oli kaikki alkuasetukset tehty palvelimelle ja päästiin Zabbix-ohjelman web-käyttöliittymään antamalla selaimen osoite 192.168.0.110/zabbix/setup.php.

Seuraavaksi määriteltiin vielä web-käyttöliittymän asennus selaimessa. Asennuksen alussa ohjelma tarkisti (kuva 8), että alkumäärittelyt ovat oikein.

ZABBIX

- Welcome
- Check of pre-requisites
- Configure DB connection
- Zabbix server details
- Pre-installation summary
- Install

Check of pre-requisites

	Current value	Required	
PHP version	5.5.9-1ubuntu4.20	5.4.0	OK
PHP option "memory_limit"	128M	128M	OK
PHP option "post_max_size"	16M	16M	OK
PHP option "upload_max_filesize"	2M	2M	OK
PHP option "max_execution_time"	300	300	OK
PHP option "max_input_time"	300	300	OK
PHP option "date.timezone"	Europe/Helsinki		OK
PHP databases support	MySQL		OK
PHP bcmath	on		OK
PHP mbstring	on		OK
PHP option "mbstring.func_overload"	off	off	OK

Back
Next step

Kuva 8. Alkumäärittysten tarkistus.

Seuraavaksi määriteltiin asetukset tietokantayhteyteen (kuva 9).

ZABBIX

- Welcome
- Check of pre-requisites
- Configure DB connection
- Zabbix server details
- Pre-installation summary
- Install

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type

Database host

Database port

0 - use default port

Database name

User

Password

Back
Next step

Kuva 9. Tietokantayhteyden asetukset.

Tämän jälkeen annettiin Zabbix-palvelimen tiedot (kuva 10).

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Zabbix server details

Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional).

Host

Port

Name

[Back](#) [Next step](#)

Kuva 10. Zabbix-palvelimen tiedot.

Lopuksi tuli yhteenveto asetuksista (kuva 11).

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Pre-installation summary

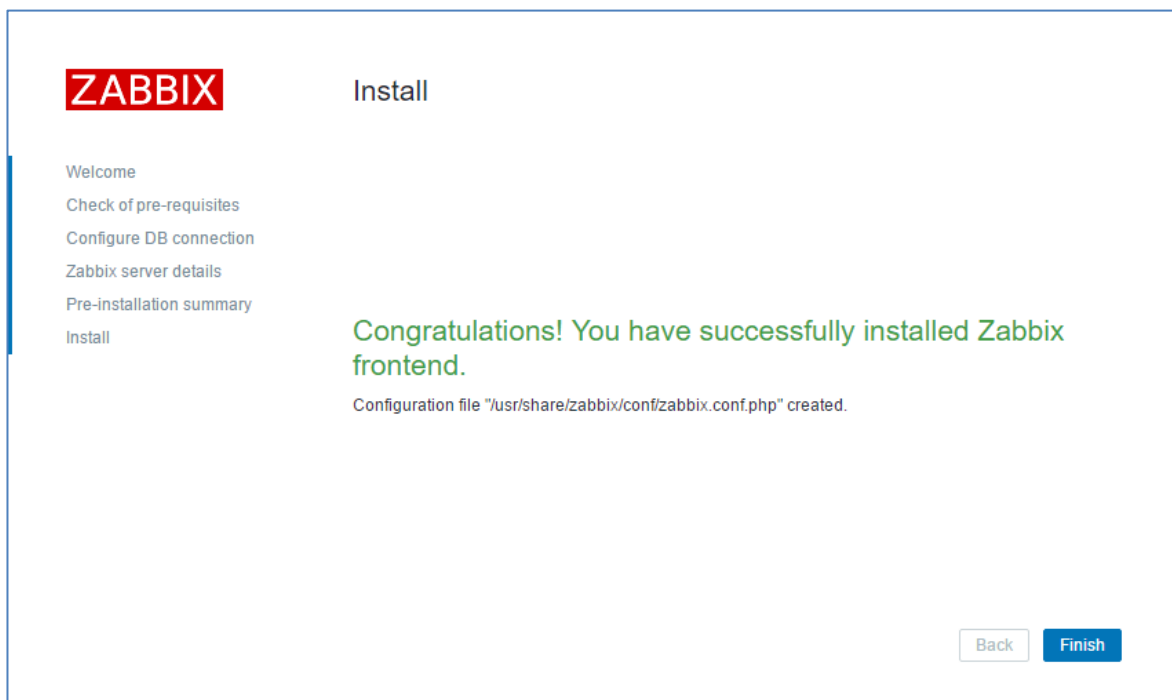
Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.

Database type	MySQL
Database server	localhost
Database port	default
Database name	zabbix
Database user	root
Database password	*****
Zabbix server	localhost
Zabbix server port	10051
Zabbix server name	Zabbix Server

[Back](#) [Next step](#)

Kuva 11. Yhteenveto asetuksista.

Viimeisessä ikkunassa web-käyttöliittymä kertoi asennuksen onnistuneen (kuva 12).



Kuva 12. Asennus valmis.

Sitten päästiin web-käyttöliittymän kirjautumisvalikkoon (kuva 13).

Kuva 13. Web-käyttöliittymän kirjautumisnäky

Kun web-käyttöliittymään kirjaudutaan sisään, ohjelma avaa Dashboard-osion aina ensin (kuva 14).

Kuva 14. Dashboard-osio

Tästä lähtien suurin osa määryksistä pystyttiin tekemään web-käyttöliittymässä sijaitsevien paneelien kautta. Vielä kuitenkin piti asentaa agentit palvelimelle ja valvottaville kohteille, jotta Zabbix tunnistaisi ne. Ensimmäisenä asennettiin agentti Zabbix-palvelimeen. Agentin asennus tapahtui terminaali-ikkunakomennolla:

```
$ apt-get install zabbix-agent
```

Asennuksen jälkeen määriteltiin konfiguraatio agentille avaamalla tiedosto komennolla terminaali-ikkunassa:

```
$ sudo vi /etc/zabbix/zabbix_agentf.conf
```

Tänne annettiin Server- ja ServerActive-kohtaan loopback-osoite 127.0.0.1. Sitten voitiin agenttipalvelu käynnistää terminaali-ikkunasta komennolla:

```
$ service zabbix-agent start
```

Windows-koneille asennus oli hieman erilainen. Agenttipaketin pystyi lataamaan internetistä osoitteesta http://www.zabbix.com/downloads/3.0.4/zabbix_agents_3.0.4.win.zip.

Tiedosto purettiin C-asemalle tehtyyn Zabbix-kansioon. Purun jälkeen zabbix_agentd-tiedostoon annettiin seuraavat määritelmät:

```
LogFile=c:\zabbix_agentd.log  
  
SourceIP=192.168.0.105  
  
Server=192.168.0.110  
  
ListenPort=10050  
  
ServerActive=192.168.0.110  
  
HostMetadataItem=system.uname
```

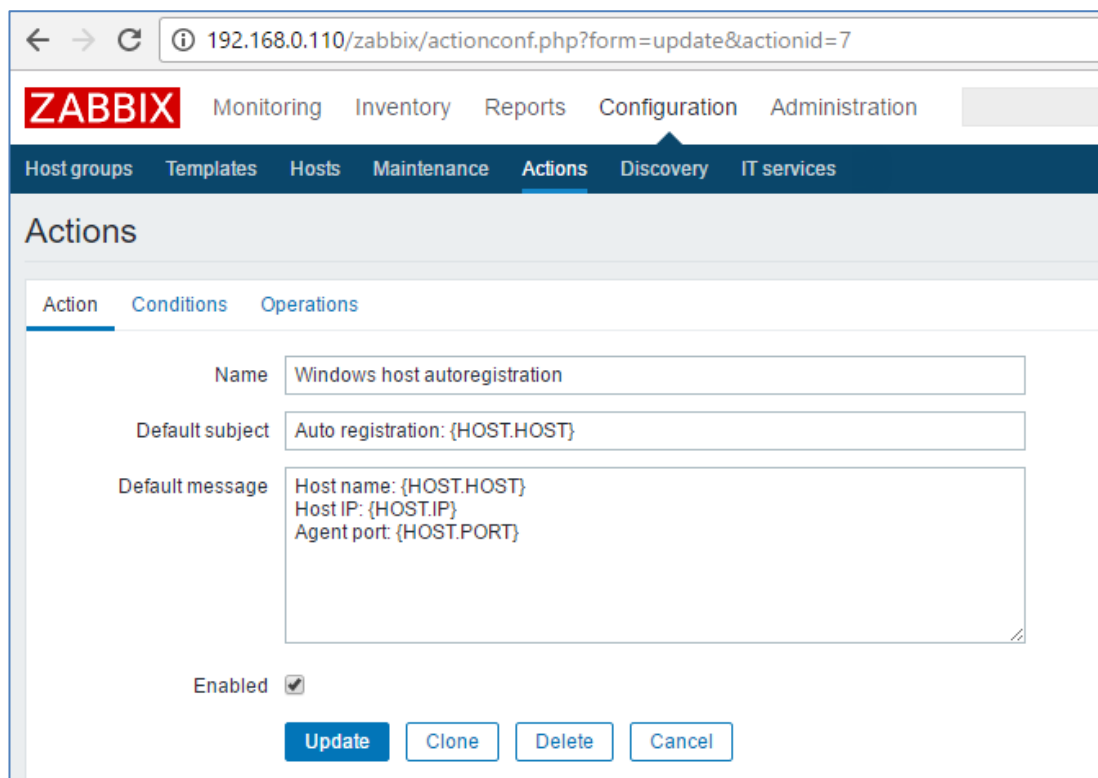
Määrittysten asettamisen jälkeen asennettiin konfiguraatiotiedosto ja käynnistettiin uusi agenttivalvelu Windowsin komentokehotteesta komennoilla:

```
zabbix_agentd.exe --config C:\Zabbix\zabbix_agentd.conf --install  
zabbix_agentd.exe --config C:\Zabbix\zabbix_agentd.conf -start
```

Tämä tehtiin fyysiselle pöytäkoneelle, jossa oli Windows 10 -käyttöjärjestelmä, sekä virtuaaliselle Windows Server 2012 -palvelimelle.

Kun agentit oli asennettu, pystyttiin määrittellä automaattinen laitteiden etsimistöiminto Zabbix-web-käyttöliittymässä. Web-käyttöliittymän välilehdiltä navigoitiin osioon Configuration->Actions ja valittiin Event source-kohtaan Auto registration ja painettiin Create Action -painiketta.

Avautuvaan määritelmävalikkoon laitettiin asetukset kuvien 15, 16 ja 17 mukaisesti.



← → ↻ 192.168.0.110/zabbix/actionconf.php?form=update&actionid=7

ZABBIX Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance **Actions** Discovery IT services

Actions

Action Conditions Operations

Name Windows host autoregistration

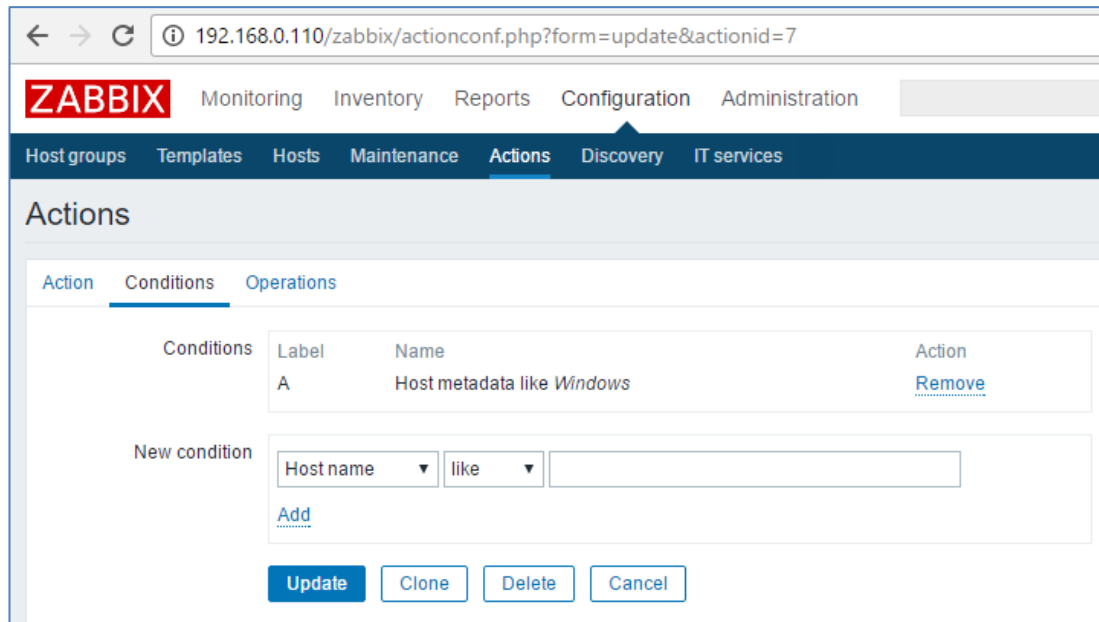
Default subject Auto registration: {HOST.HOST}

Default message Host name: {HOST.HOST}
Host IP: {HOST.IP}
Agent port: {HOST.PORT}

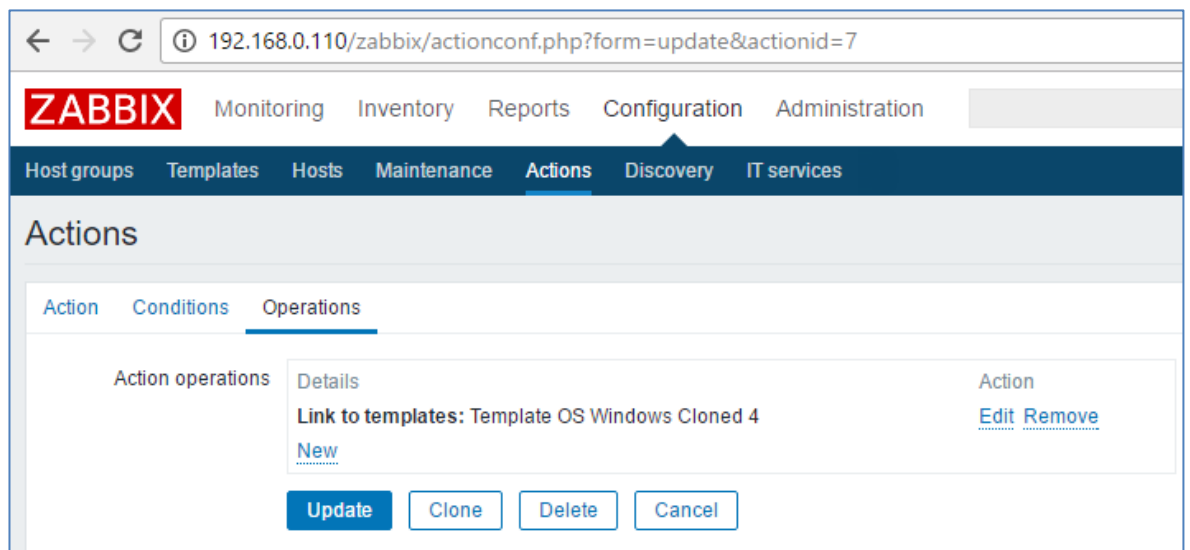
Enabled

Update Clone Delete Cancel

Kuva 15. Action-määritelmävalikko.



Kuva 16. Conditions-määritelmät.



Kuva 17. Operations-määritelmät.

Tällä tavalla tehty automaattinen etsimistoiminto osasi automaattisesti etsiä samassa verkossa Zabbix-palvelimen kanssa toimivia Windows-koneita. Näihin lisättiin automaattisesti web-käyttöliittymässä sijaitseva Windows-template, joka sisälsi

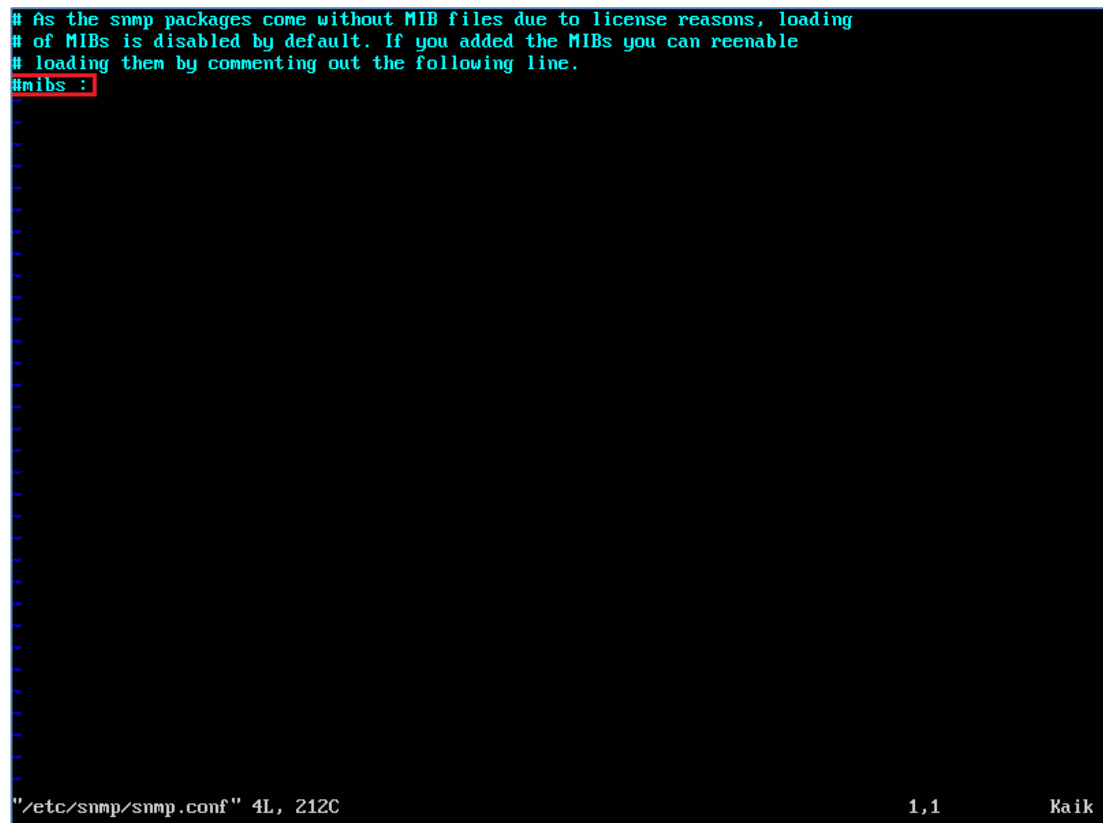
monia Windows-käyttöjärjestelmälle sopivia valvottavia kohteita. Windows-koneet tulivat näkyviin Hosts-välilehdelle, josta pystytään hallitsemaan valvottavia laitteita.

Ciscon Catalyst -kytkin lisättiin manuaalisesti ympäristöön. Ensin Ubuntu-palvelimelta piti asentaa SNMP-valvontaan tarvittavat paketit terminaaali-ikkunakomennolla:

```
$ apt-get install snmp snmp-mibs-downloader
```

Sitten SNMP-konfigurointitiedostosta kommentoitiin pois mibs: määritelmä.

```
$ sudo vi /etc/snmp/snmp.conf
```



```
# As the snmp packages come without MIB files due to license reasons, loading
# of MIBs is disabled by default. If you added the MIBs you can reenale
# loading them by commenting out the following line.
#mibs :

```

"/etc/snmp/snmp.conf" 4L, 212C 1,1 Raik

Kuva 18. SNMP-konfigurointitiedosto.

Tämän jälkeen annettiin komento terminaaali-ikkunaan tarpeellisten MIB-kirjastojen asennukseen:

```
$ download-mibsmibs
```


Lopuksi käynnistettiin Zabbix-palvelin uudelleen komennolla terminaali-ikkunassa:

```
$ sudo service zabbix-server restart
```

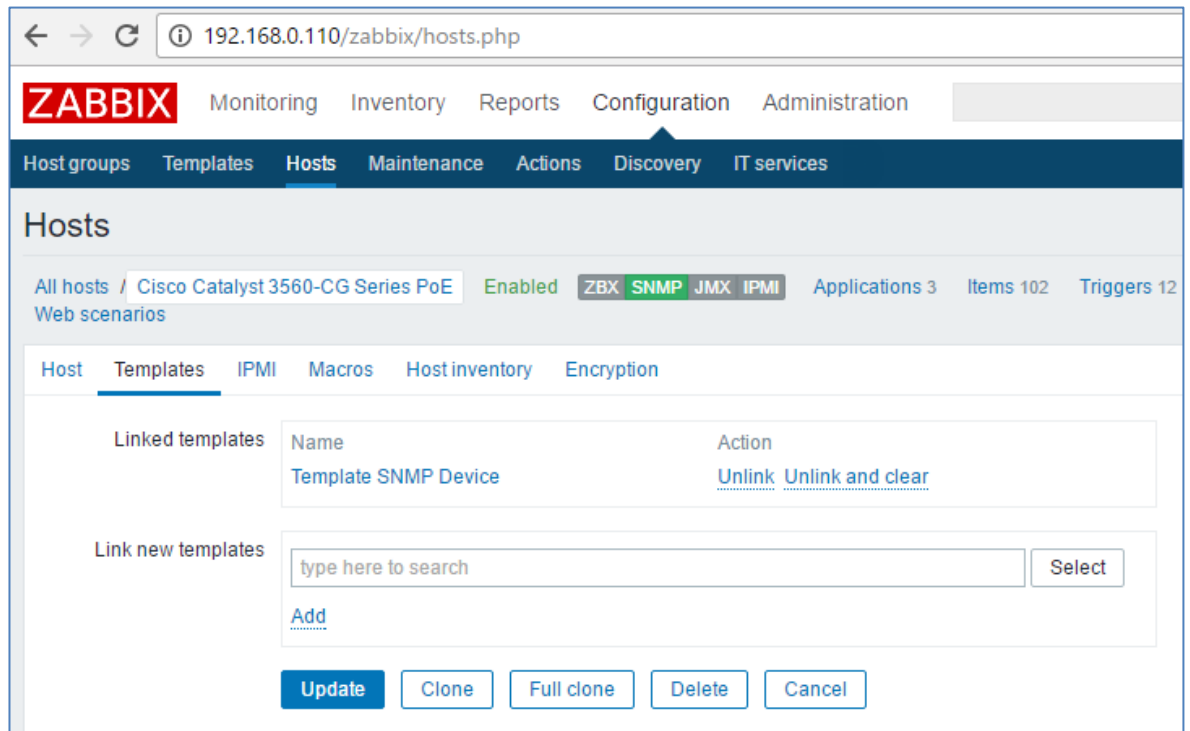
Sitten voitiin lisätä kytkin valvonnan piiriin web-käyttöliittymässä.

Valittiin käyttöliittymän välilehdeltä Configuration->Hosts ja painettiin Create host -painiketta. Tänne määriteltiin uusi Switches-ryhmä ja annettiin kytkimelle uniikki nimi. SNMP interfaces -kohtaan laitettiin kytkimen IP-osoite.

The screenshot shows the Zabbix web interface for configuring a host. The browser address bar shows the URL: 192.168.0.110/zabbix/hosts.php?form=update&hostid=10114&groupid=0. The Zabbix logo and navigation menu are visible at the top. The main content area is titled 'Hosts' and shows the configuration for a host named 'Cisco Catalyst 3560-CG Series PoE'. The 'SNMP interfaces' section is highlighted with a green box, showing the IP address '192.168.0.107' and port '161'. The 'Agent interfaces' section is also visible, showing the IP address '192.168.0.107' and port '161'. The 'Description' field is empty. The 'Monitored by proxy' dropdown is set to '(no proxy)'.

Kuva 19. SNMP-laitteen lisääminen.

Sitten annettiin kytkimelle vielä valmis SNMP-pohjainen template, joka hakee tietoa kytkimen tilasta.



Kuva 20. Template SNMP-laitteelle.

Lopuksi painettiin Update-painiketta ja kytkin tuli näkyviin pian toimivana Hosts-välilehdelle. Hosts-välilehdellä näkyvät myös muut valvottavaan piiriin löydetyt laitteet. Jos laite näkyy vihreänä Availability-kohdassa (kuva 21), on siihen saatu yhteys onnistuneesti.

The screenshot shows the Zabbix web interface at the URL 192.168.0.110/zabbix/hosts.php?ddreset=1. The page title is 'Hosts' and it includes a search bar and filter options. The main content is a table of hosts with the following data:

Name	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Templates	Status	Availability	Agent encryption	Info		
ALEKSI-PC	9	80	14	27	2	Web	192.168.0.105:10050	SqlServerTemplate, Template OS, Windows Cloned 4 (Template App Zabbix Agent)	Enabled	ZBX	SNMP	JMX	IPMI	NONE
Cisco Catalyst 3560-CG Series PoE	3	6			2	Web	192.168.0.107:161	Template SNMP Device (Template SNMP Generic, Template SNMP Interfaces), Template SNMP Processors	Enabled	ZBX	SNMP	JMX	IPMI	NONE
WINSERVER	9	67	12	23	2	Web	192.168.0.108:10050	Template OS, Windows Cloned 4 (Template App Zabbix Agent)	Enabled	ZBX	SNMP	JMX	IPMI	NONE
Zabbix server	11	75	46	13	2	Web	127.0.0.1:10050	Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	Enabled	ZBX	SNMP	JMX	IPMI	NONE

At the bottom of the table, there are buttons for '0 selected', 'Enable', 'Disable', 'Export', 'Mass update', and 'Delete'. The page indicates 'Displaying 4 of 4 found'.

Kuva 21. Demoympäristön valvottavat laitteet Hosts-näkymässä.

5.3 Automaattinen raportin lähetys sähköpostiin

Ensimmäisen tutkittavan osa-alueen periaatteena oli, että Zabbix-verkonvalvonta-ohjelmisto pystyisi automaattisesti lähettämään viikoittaisen tilanneraportin halutuista laitteistoista sähköpostitse verkonvalvonnan ylläpitäjälle.

Tilanneraportin kokoaminen ja lähetys onnistuttiin tekemään Perl-scriptin avulla ja tämän automaattinen toteutus Cron-ajastuspalvelun avulla.

Aluksi piti tehdä Zabbix-palvelimessa uusi Perl-scripti halutulla nimellä ja haluttuun polkuun. Tässä työssä tiedoston nimeksi valittiin graphreport.pl ja poluksi /usr/share/zabbix/scripts. Scriptiin löytyi hyvä pohja eräältä Zabbix-foorumilta ja pienellä muokkauksella se saatiin toimimaan halutulla tavalla. Valmis scripti poimi Zabbix-web-käyttöliittymän aloitussivun Favorites-osiossa olevat kuvaajat ja lähetti ne valvojalle määritettyyn sähköpostiin.

Seuraavana on kuvattu valmis Perl-scripti kokonaisuudessaan.

```

#!/usr/bin/perl
use DateTime;
use File::Path;
use MIME::Lite;
use DBI;

my $zabbix      = "zabbix.domain.com/zabbix"; #zabbix frontend-osoite
my $login       = 'zabbix'; # zabbix api käyttäjä
my $pass        = 'password'; # zabbix api salasana
my $cookie      = "/tmp/cookie.txt";

my $width       = 570;
my $height      = 234;

my $dsn         = 'DBI:mysql:zabbix';
my $db_user_name = 'zabbix'; #Zabbix-tietokannan käyttäjänimi
my $db_password = 'zabbix'; #Zabbix-tietokannan salasana
my %user_screen;
my %graphs;
my %screens;
my $period;
my $type;

my $path        = '/tmp/graphreport';
if(-e $path) { rmtree($path); }
mkdir($path);

my $dt = DateTime->now( time_zone => 'Europe/Helsinki' ); # create a
DateTime object set to the current date and time, with no particular
timezone

if ($ARGV[0] eq "-month") {
    $dt->subtract( months => 1 )->set day(1)->set hour(0)->set mi-
nute(0);
    $lt = $dt->clone->add( months => 1 )->subtract( days => 1 )-
>set_hour(0)->set_minute(0);
    $period = 2592000;
    $type = "Month of";
} else {
    $dt->subtract( days => 0 )->set_hour(2)->set_minute(0); #tässä
days kohtaan laitetaan monenko paivan ajanjaksolta kerataan data, tassa 0
tarkoittaa samaa paivaa kun scripti ajetaan
    $lt = $dt->clone->add( days => 0 )->set_hour(2)->set_minute(0);
    $period = 3600; #sekunteissa aikajakso
    $type = "Week of";
}

#if ($dt->day >= 25) {
#    $dt->set_month( $dt->month - 1 ); # set the month to it's current
value plus 1
#    $dt->set_day( 1 ); # set the day to the 1st
#}
my $stime       = $dt->strftime("%Y%m%d%H%M%S");
my $repstart    = $dt->strftime("%a, %b %d, %Y");
my $repend      = $lt->strftime("%a, %b %d, %Y");

my $query       = " SELECT p.value_id, IF(s.resourceid IS
NULL,p.value_id,s.resourceid) as graphid, p.idx, IF(s.resourceid IS NOT
NULL,ss.name,'Graphs') as title,

```

```

        u.name, u.surname, u.userid, m.sendto, p.source
FROM profiles p
        LEFT JOIN screens_items s ON ( p.value_id =
s.screenid )
        LEFT JOIN screens ss ON ( s.screenid = ss.screenid
)
        LEFT JOIN users u ON ( p.userid = u.userid )
        LEFT JOIN media m ON ( u.userid = m.userid )
        WHERE (p.idx like 'web.favorite.screenids' OR p.idx
like 'web.favorite.graphids')
        AND m.mediatypeid = 1;";

my $db          = DBI->connect($dsn, $db_user_name, $db_password);
my $zab         = $db->prepare($query);
$zab->execute();

my $cmdLogin    = `curl -k -s -c $cookie -d 'form=1&form_re-
fresh=1&name=$login&password=$pass&enter=Sign%20in' $zabbix/index.php`;

while (my ($value_id, $graphid, $idx, $title, $first, $last, $userid,
$email, $source) = $zab->fetchrow_array()) {
    my $output = "$path/$title.$graphid.png";
    if ($source eq "itemid") {
        $chart = "chart.php?itemids";
    } if ($source eq "graphid") {
        $chart = "chart2.php?graphid";
    }
    my $cmdImage = `curl -s -b $cookie -o "$output" '$zab-
bix/$chart=$graphid&period=$pe-
riod&stime=$stime&width=$width&height=$height``;

    $user_screen{$userid}{'email'} = $email;
    $user_screen{$userid}{'name'} = $first;
    push( @{$user_screen{$userid}{'graphs'} }, $title."":".$graphid
);
    push( @{$user_screen{$userid}{'titles'}{"$title"} }, $graphid );
}

$db->disconnect();

foreach $userid (keys(%user_screen)) {
    print "Sending: $user_screen{$userid}{'name'} =>
$user_screen{$userid}{'email'}\n";
    my $subject    = "$type $repstart: Zabbix report for
$user_screen{$userid}{'name'}";
    my $msg        = MIME::Lite->new(
        From      => 'zabbix@domain.com',
        To        => $user_screen{$userid}{'email'},
        Subject   => $subject,
        Type      => 'multipart/related' );
    my @usergraphs = @{$user_screen{$userid}{'graphs'} };
    my $phtml      = "zzzzzz";
    my $pattach    = "zzzzzz";

    my $str        = "<body style=\"font-family: Helvetica, sans-
serif\;\">\n";
    $str           .= "<p style=\"font-size: 18px\; font-weight:
bold\;\">$type $repstart through $repend</p>\n";
    foreach my $title (keys{%$user_screen{$userid}{'titles'}}) {

```

```

        #print "$title => @{ $user_screen{sele$userid}{'titles'} }{\$title} }\n";
        $str .= "<br>\n<h3>$title</h3>\n";
        my $i = 0;
        foreach my $g (@{ $user_screen{$userid}{'titles'}{\$title}
    )){
            $str .= "<br>\n<div align=\"center\"><img
src=\"cid:$g.$i.png\"></div>\n";
            $i++;
            #print "  Attaching: $g\n";
        }
        $str .= "<br><hr width=\"300\" align=\"center\"><br>\n";
    }
    $str      .= "<p>You can change the graphs sent to you by
changing your saved favorite graphs/items from the dashboard.</p>\n";
    $str      .= "</body>\n";
    $msg->attach( Type => 'text/html; charset= utf-8', #windows-1251',
        Data => $str );

    foreach my $title (keys%{$user_screen{$userid}{'titles'}}) {
        my $i = 0;
        foreach my $g (@{ $user_screen{$userid}{'titles'}{\$title}
    )){
            my $file = "$path/$title.$g.png";
            $msg->attach( Type      => 'image/png',
                Path      => $file,
                Id        => "$g.$i.png");
            #print "$g.$i.png  =>  $file\n";
            $i++;
        }
    }
    $msg->send();
}

exit ;

```

Perl-kieleen piti päivittää Datetime- ja Mime:Lite-paketit, jotta kaikki scriptin komennot toimivat. Datetime-moduuli sisältää aikaan liittyviä metodeja ja Mime:Lite-moduuli sähköpostiviestin luomiseen helpottavia metodeja. Tämän jälkeen scripti pystyttiin ajamaan komennolla terminaali-ikkunassa:

```
$ sudo /usr/bin/perl /usr/share/zabbix/scripts/graphreport.pl
```

Zabbix-palvelimen komentokehote antoi ilmoituksen onnistuneesta lähetyksestä ja grafiikat tulivat sähköpostiin odotetusti (kuva 22).



Kuva 22. Kuvaaja valvotun laitteen tilasta ylläpitäjän sähköpostissa.

Scriptin automaattiseen suoritukseen käytettiin Cron-ajastuspalvelua. Terminaali-ikkunaan kirjoitettiin komento `sudo crontab -e`, jotta päästiin asettamaan aikamäärittelmä, komento ja polku. Arvoiksi asetettiin:

```
0 * * * * perl /usr/share/zabbix/scripts/graphreport.pl
```

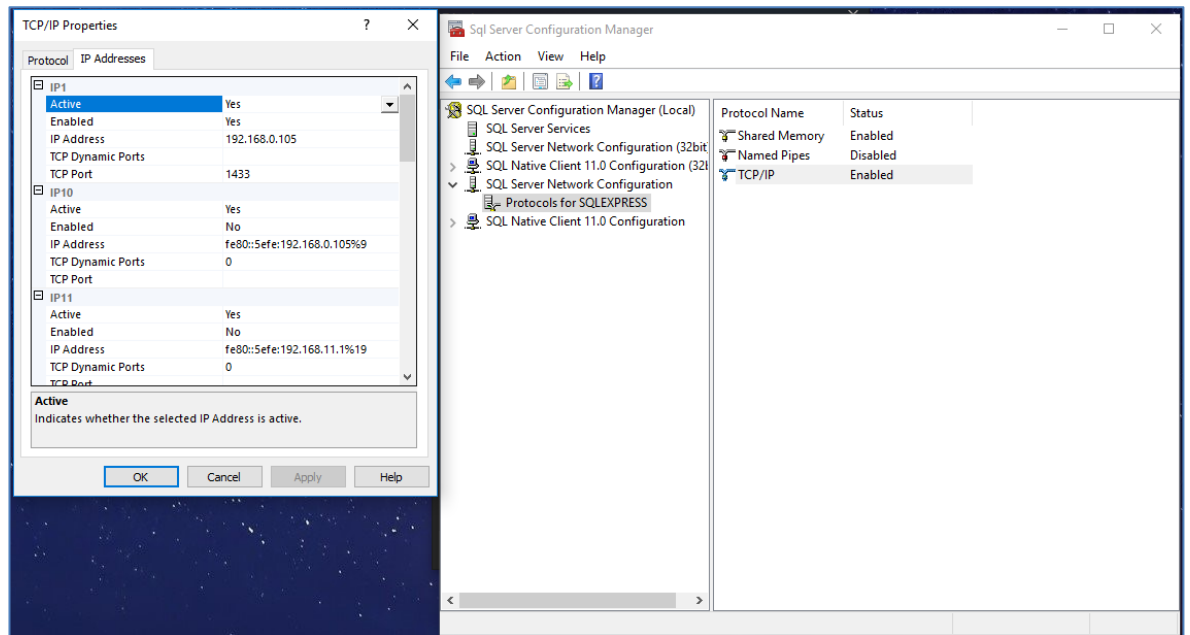
Tallennuksen jälkeen Cron-ajastuspalvelu alkoi ajaa kyseisen komennon automaattisesti jokaisen alkavan tunnin alussa.

5.4 Tietokantavalvonnan asennus

Zabbix pystyy valvomaan haluttua tietokantaa Microsoftin ODBC-rajapinnan avulla. Demoympäristössä testattiin tietokantavalvonta virtuaalisen Windows Server 2012 -palvelimen sisältämään Sql Server 2014 -tietokantaan. Ensimmäisenä vaiheena ODBC-rajapinta asennettiin Zabbix-palvelimelle. Lisäksi tarvittiin ajurit, joihin asetettiin valvottavan tietokannan tiedot. Lisäksi asennettiin FreeTDS-ajurit, sillä ne olivat ilmaiset ja todettiin toimiviksi.

Ensimmäiseksi avattiin IP/TCP-liikenne tietokannalle. Tämä tapahtui muuttamalla tietokannan asetuksia Sql Serverin Configuration Managerilla. Täältä avattiin SQL

Server Network Configuration -välilehden Protocols for SQLEXPRESS -osio, josta vaihdettiin TCP/IP-protokolla Enabled-muotoon. Protokollan Properties-ikkunasta valittiin IP1-osion Active-kohtaan Yes, Enabled-kohtaan Yes, IP Address -kohtaan palvelimen osoite, TCP Dynamic Ports -kohta tyhjäksi ja TCP Port -kohtaan 1433. 1433 on tietokantavalvontaan käytetty tavallinen porttimääritelmä.



Kuva 23. TCP-liikenteen avaaminen SQL Server Configuration Managerissa.

ODBC-rajapinnan ja ajurien asennus tapahtui Zabbix-palvelimen terminaali-ikkunaan kirjoittamalla

```
$ sudo apt-get install unixodbc unixodbc-dev freetds-dev freetds-bin tdsodbc
```

Tämän jälkeen piti avata tekstinkäsittelyohjelmalla Odbcinst.ini-tiedosto polusta /etc/odbcinst.ini ja asettaa tänne ajurin tiedot. Tiedostoon asetettiin seuraavat parametrit:

```
[FreeTDS]
```

```
Description = v0.91 with protocol v7.2 #Informaatiota ajurista
```

```
Driver = /usr/lib/x86_64-linux-gnu/odbc/libtdsodbc.so #Ajuritiedoston polku
```


Tämän jälkeen tiedosto tallennettiin. Seuraavaksi avattiin Odbc.ini-tiedosto polusta /etc/odbc.ini. Tähän taas asetettiin tietokannan tiedot, mihin haluttiin ottaa yhteys. Tässä tapauksessa tietokanta sijaitsee virtuaalisella Windows Server 2012 -palvelimella. Alla tiedoston parametrit:

```
[windowsdatabase]      #uniikki nimi tietokannalle

Driver = FreeTDS       #tähän ajurin nimi, mikä asetettiin odbcinst.ini -
tiedostoon

Server = 192.168.0.110 #tähän valvottavan tietokannan osoite

Port = 1433            #tämä on normiportti yhteyden ottamiseen tietokan-
taan

TDS_Version = 7.2     #ajurin versio

ClientCharset = UTF-8  #asetetaan oikea merkistö
```

Tiedosto tallennettiin ja yhteys tietokantaan voitiin kokeilla isql-komennon avulla. Isql-komentoon annettiin tietokannan nimi, joka asetettiin odbc.ini-tiedostoon ja tämän jälkeen tietokannan käyttäjänimi ja salasana. Zabbix-palvelimen terminaalikokona annettiin komento isql -v windowsdatabase käyttäjä salasana, jonka jälkeen yhteys avautui. -V tarkoittaa tässä komennossa englanninkielisen sanan verbose ja se antaa enemmän informaatiota virhetilanteesta, jos yhteys ei toimi.

Viimeisenä vaiheena tehtiin uusi Item-kohde web-käyttöliittymässä. Ensiksi piti tehdä uusi template-malli, joten web-käyttöliittymän välilehdeltä valittiin Configuration→Templates→Create template. Template-malliin annettiin haluttu nimi ja siirryttiin Items-kohtaan ja täältä Create item -kohtaan. Item-kohteelle asetettiin halutut arvot ja painettiin lopuksi Update. Kuvissa 24 ja 25 on esitetty tiedot Item-kohteesta ja sen palautusarvosta Latest data -välilehdellä.

192.168.0.110/zabbix/items.php?form=update&hostid=10107&itemid=23766

All templates / SqlServerTemplate Applications **Items 1** Triggers Graphs Screens Discovery rules Web scenarios

Name

Type

Key

User name

Password

SQL query

Type of information

Update interval (in sec)

Custom intervals

Type	Interval	Period	Action
<input checked="" type="checkbox"/> Flexible <input type="checkbox"/> Scheduling	<input type="text" value="50"/>	<input type="text" value="1-7,00:00-24:00"/>	<input type="button" value="Remove"/>

[Add](#)

History storage period (in days)

New application

Applications

Populates host inventory field

Description

Enabled

Kuva 24. Item-kohteen teko tietokantavaltontaa varten.

The screenshot shows the 'Latest data' page in Nagios XI. The navigation bar includes: Dashboard, Overview, Web, Latest data, Triggers, Events, Graphs, Screens, Maps, Discovery, IT services. The main content area has a 'Filter' section with the following fields:

- Host groups: type here to search [Select]
- Hosts: ALEKSI-PC [Select]
- Application: [Select]
- Name: [Text input]
- Show items without data:
- Show details:

Below the filter are 'Filter' and 'Reset' buttons. The main table has the following structure:

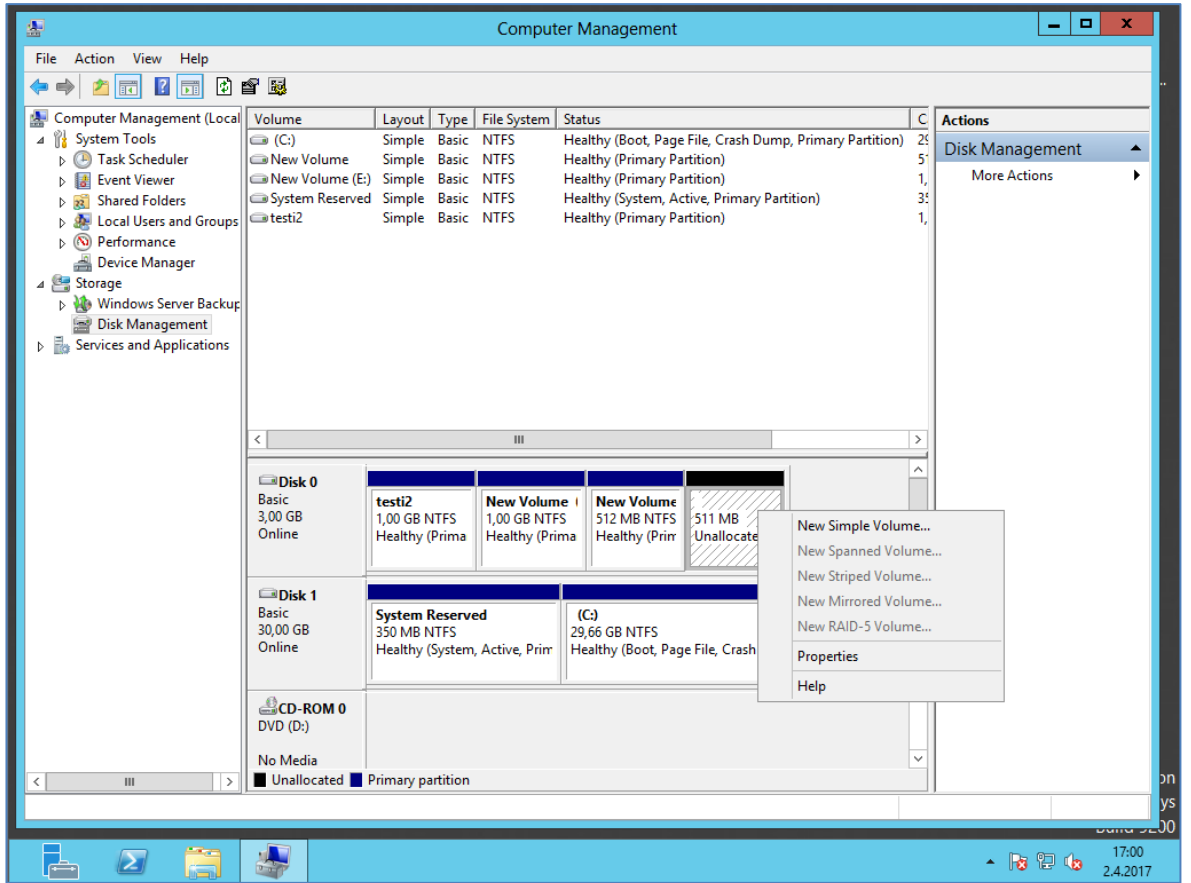
Name	Last check	Last value
CPU (3 items)		
Filesystems (24 items)		
General (2 items)		
Memory (5 items)		
Network interfaces (40 items)		
OS (2 items)		
Performance (7 items)		
Processes (1 item)		
Zablix agent (3 items)		
- other - (1 item)		

At the bottom of the table, there is a 'Select' button and the text '2017-02-13 12:20:07' and 'asa_123456'. Below the table are buttons for '0 selected', 'Display stacked graph', and 'Display graph'.

Kuva 25. Tietokantavalvontaa varten tehdyn item-kohteen palauttama arvo Latest data -välilehdellä.

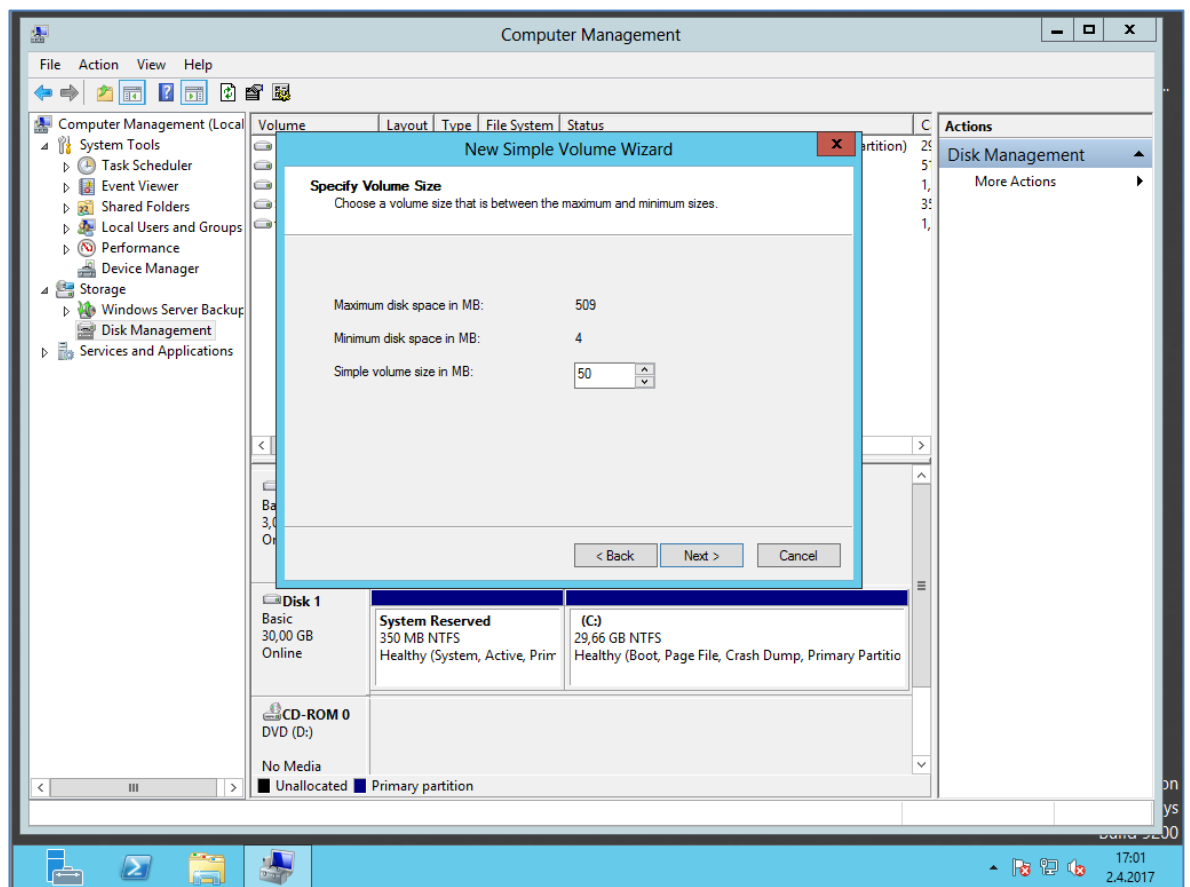
5.5 Mount point -kansioiden valvonta

Mount point -kansio tehtiin Windows Server 2012 -palvelimelle valvontaa varten. Ensimmäisenä vaiheena tehtiin uusi kansio palvelimen C-asemalle. Kansion nimeksi valittiin testi. Seuraavaksi mentiin palvelimen Computer Management -osiin. Täällä tehtiin vapaaseen levytilaan New Simple Volume.



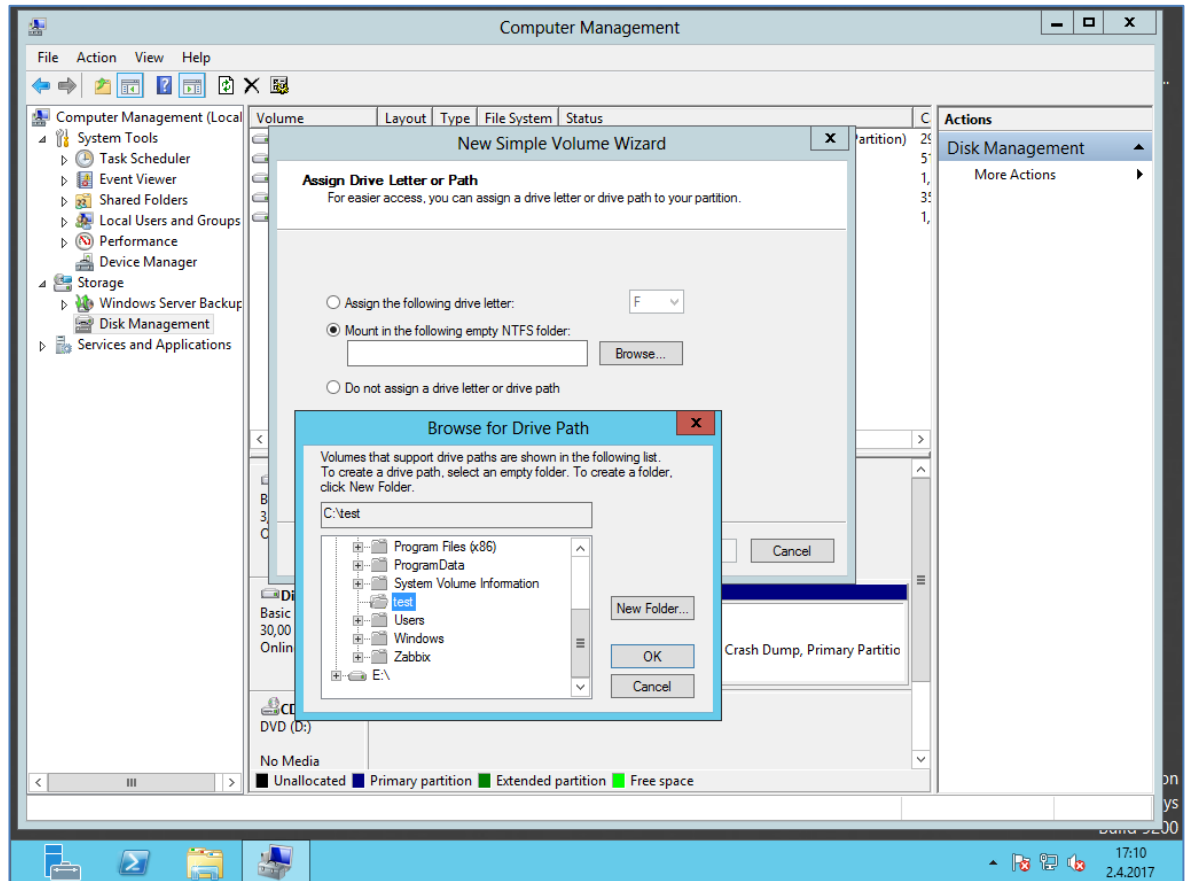
Kuva 26. New Simple Volume Computer Management -osiossa.

Tämän jälkeen asennusohjelma avautui ja alkuvalikosta painettiin Next. Tämän jälkeen valittiin levyn koko (kuva 27). Kooksi valittiin 50 megatavua.



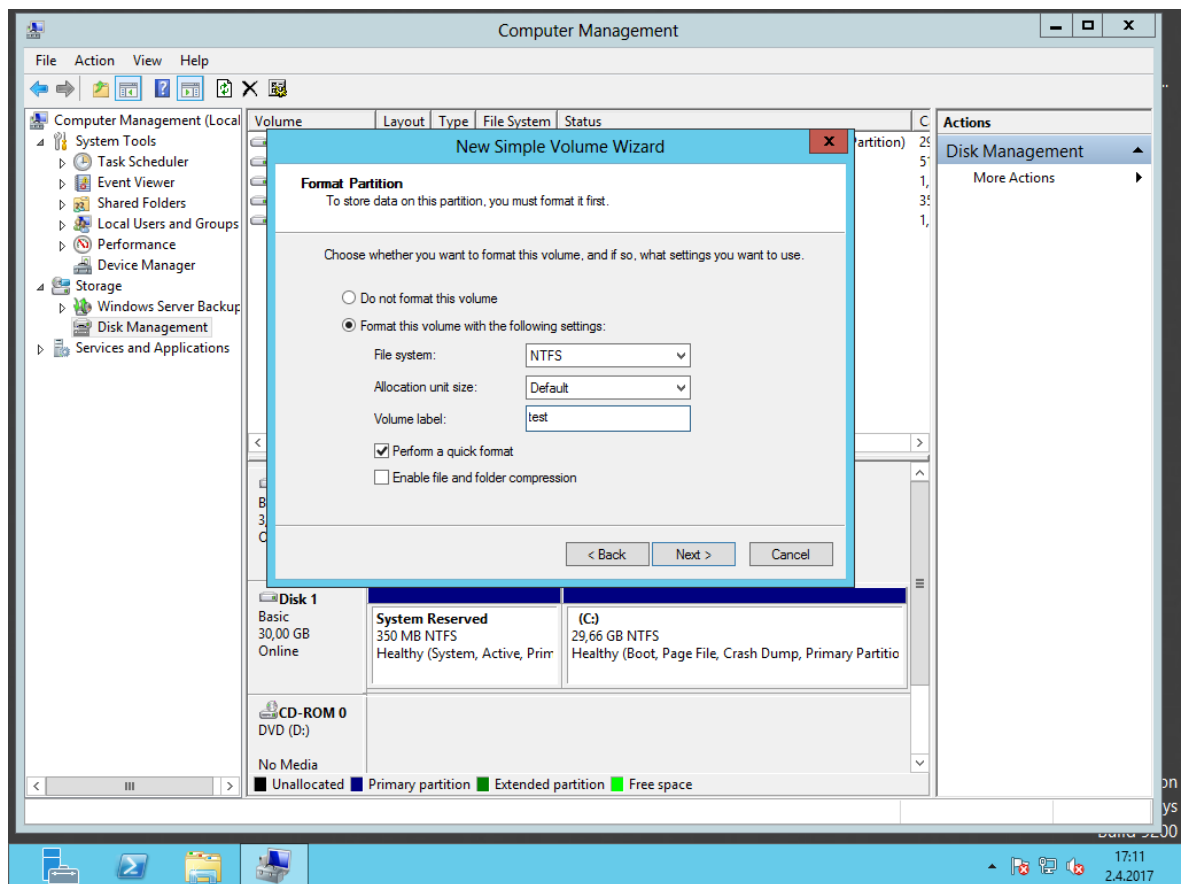
Kuva 27. Levytilan koon valitseminen.

Sitten painettiin Next ja päästiin valitsemaan, mistä kansioista haluttaisiin tehdä mount point. Poluksi valittiin aiemmin tehty uusi test-niminen kansio C-asemassa (kuva 28).



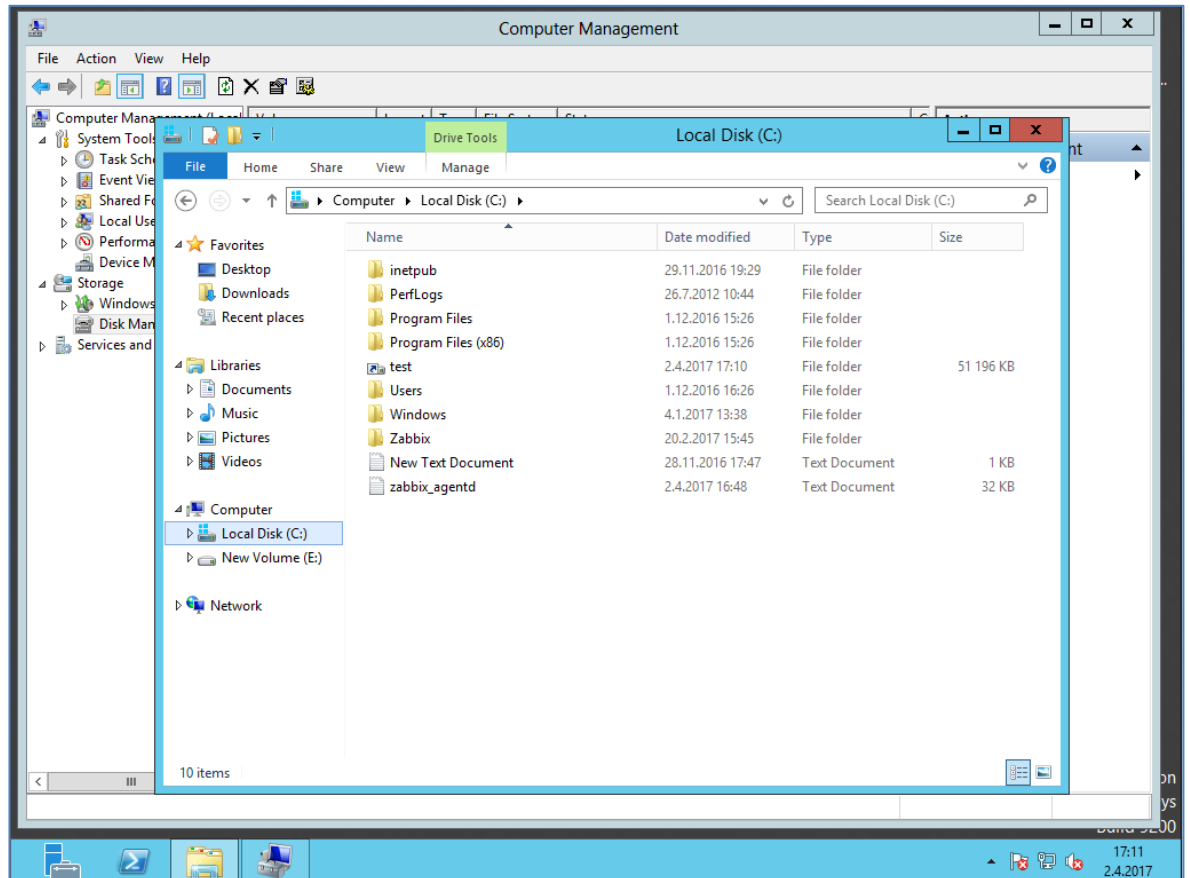
Kuva 28. Mount point -kansion polku.

Seuraavaksi painettiin Next ja valittiin tiedostojärjestelmän tyyppi, tilanvarausyksi-
kön koko ja sen nimi (kuva 29).



Kuva 29. Levyn asetukset.

Tämän jälkeen tuli yhteenvetoikkuna ja asennus oli valmis. Kun menttiin tarkastamaan tehtyä test-kansiota, oli kansion kuvakkeeksi muodostunut levykuvake (kuva 30).



Kuva 30. Test-kansion levykuvake.

Uusi Item-kohte tehtiin Zabbix-web-käyttöliittymässä Windows-palvelimelle mount point -kansion valvontaa varten. Yhteys muodostettiin kansioon antamalla palvelimen IP-osoite ja key-arvo, johon määriteltiin polku mount point -kansioon (kuva 31).

The screenshot shows the Zabbix web interface for configuring a new item. The browser address bar indicates the URL: 192.168.0.110/zabbix/items.php?form=update&hostid=10109&itemid=23860. The page title is 'Items'.

The configuration details are as follows:

- Name:** Mount point testi free space (percent)
- Type:** Zabbix agent
- Key:** vfs.fs.size[c:\test,pfree]
- Host interface:** 192.168.0.108 : 10050
- Type of information:** Numeric (float)
- Units:** %
- Use custom multiplier:** 1
- Update interval (in sec):** 60
- Custom intervals:**

Type	Interval	Period	Action
Flexible	Scheduling	50	1-7,00:00-24:00
- History storage period (in days):** 7
- Trend storage period (in days):** 365
- Store value:** As is
- Show value:** As is
- New application:** (empty field)
- Applications:**
 - None-
 - CPU
 - Filesystems**
 - General
 - Memory
 - Network interfaces
 - OS
 - Performance
 - Processes
 - Zabbix agent
- Populates host inventory field:** -None-
- Description:** (empty text area)

Kuva 31. Item-kohteen teko Mount point -kansion valvontaa varten.

Latest data -välilehdelle saatiin palautus item-kohteelta.

The screenshot shows a monitoring interface with a search bar at the top and a list of metrics. The 'Mount point free space (percent)' item is highlighted in blue, indicating it is the selected item for the latest data view.

Name	Last check	Last value	Change
CPU (3 items)			
Filesystems (14 items)			
Average disk read queue length	2017-02-20 17:27:32	0	
Average disk write queue length	2017-02-20 17:27:33	0	
File read bytes per second	2017-02-20 17:27:34	0 Bps	-1.34 KBps
File write bytes per second	2017-02-20 17:27:35	2.33 KBps	-1.63 KBps
Free disk space on C:	2017-02-20 17:27:31	3.98 GB	
Free disk space on C: (percentage)	2017-02-20 17:27:33	13.41 %	
Free disk space on E:	2017-02-20 17:27:32	982.11 MB	
Free disk space on E: (percentage)	2017-02-20 17:27:34	95.91 %	
Mount point free space (percent)	2017-02-20 17:27:33	25.63 %	
mount_point_free_space	2017-02-20 17:27:54	672.52 MB	
Total disk space on C:	2017-02-20 16:37:35	29.66 GB	
Total disk space on E:	2017-02-20 16:37:36	1024 MB	
Used disk space on C:	2017-02-20 17:27:37	25.68 GB	
Used disk space on E:	2017-02-20 17:27:38	41.88 MB	

Kuva 32. Mount point -kansion item-kohteen palauttama arvo Latest data -välilehdessä.

6 YHTEENVETO JA POHDINTA

Työssä perustettiin demoympäristö, jonka avulla tutkittiin Zabbix-verkonvalvontajärjestelmän ominaisuuksia. Tutkittavia ominaisuuksia olivat tietokantavalvonta, automaattisen raportin lähetys ylläpitäjän sähköpostiin ja mount point -kansioiden valvonta.

Työssä oli haastetta, sillä kokemusta verkonvalvonnan perustamisesta tai sen käytöstä ei ollut ennen työn aloittamista. Tämän johdosta työssä kului aikaa jonkin verran asioiden opetteluun. Zabbix-ohjelman sivuilta ja sen foorumeilta, sekä Zabbix-ohjelman käyttöä koskevien Youtube-videoiden avulla saatiin kuitenkin tarvittava tieto työn toteutukseen. Lopputuloksena demoympäristö saatiin perustettua ja tutkittavat ominaisuudet saatiin toimimaan hyvin demoympäristössä. Tietohallinnolle tehtiin ohje työssä tutkittujen ominaisuuksien lisäämisestä valvontaan.

Työ antoi paljon hyvää tietoa ja osaamista verkonvalvonnan osa-alueelta. Myös Ubuntu-palvelimen käyttö tuli paremmin tutuksi, sekä MySQL- ja Sql Server 2014 -tietokantojen käyttö.

LÄHTEET

- Cisco. 2006. Understanding Simple Network Management Protocol (SNMP) Traps. [Verkkosivu]. Cisco Systems Inc. [Viitattu 13.3.2017]. Saatavissa: <http://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-snmp/7244-snmp-trap.html>
- FreeTDS. Ei päiväystä. Home. [Verkkosivu]. FreeTDS. [Viitattu 20.2.2017]. Saatavissa: <http://www.freetds.org/>
- From, R., Sivasubramanian, B. & Frahim, E. 2010. Implementing Cisco IP Switched Networks (SWITCH) Foundation Learning Guide. 1. painos. Indianapolis: Cisco Systems Inc.
- Hakala, M. & Vainio, M. 2005. Tietoverkon rakentaminen. 1. painos. Jyväskylä: Docendo Finland Oy
- Indiana University. 2013. What is meant by mounting a drive?. [Verkkosivu]. Indiana University. [Viitattu 20.2.2017]. Saatavissa: <https://kb.iu.edu/d/anqk>
- Kaario, K. 2002. TCP/IP -verkot. 1. painos. Jyväskylä: Docendo Finland Oy
- Linux.fi. 2016. Komentojen ajastaminen. [Verkkosivu]. Linux.fi. [Viitattu 29.2.2017]. Saatavissa: https://linux.fi/wiki/Komentojen_ajastaminen
- Microsoft. 2017. Microsoft Open Database Connectivity (ODBC). [Verkkosivu]. Microsoft Corp. [Viitattu 4.3.2017]. Saatavissa: <https://docs.microsoft.com/en-us/sql/odbc/microsoft-open-database-connectivity-odbc>
- Perl. Ei päiväystä. Perl Programming Documentation. [Verkkosivu]. Perl. [Viitattu 20.2.2017]. Saatavissa: <http://perldoc.perl.org/perl.html#DESCRIPTION>
- SolarWinds. Ei päiväystä. Network monitoring Design Philosophy. [Verkkosivu]. SolarWinds Worldwide LLC. [Viitattu 18.3.2017]. Saatavana: <http://www.solarwinds.com/network-monitoring-design-philosophy>
- Ubuntu. Ei päiväystä. The Ubuntu Story. [Verkkosivu]. Canonical Ltd. [Viitattu 20.2.2017]. Saatavissa: <https://www.ubuntu.com/about/about-ubuntu>
- VMware. Ei päiväystä. Workstation for Windows. [Verkkosivu]. VMware Inc. [Viitattu 20.2.2017]. Saatavissa: <http://www.vmware.com/products/workstation.html>
- Zabbix. Ei päiväystä. Zabbix Documentation 3.0. [Verkkosivu]. Zabbix SIA. [Viitattu 15.3.2017]. Saatavissa: <https://www.zabbix.com/documentation/3.0/manual/config/notifications>

- Zabbix. Ei päiväystä. Zabbix Documentation 3.0. [Verkkosivu]. Zabbix SIA. [Viitattu 16.3.2017]. Saatavissa: <https://www.zabbix.com/documentation/3.0/manual/config/triggers>
- Zabbix. Ei päiväystä. Zabbix Documentation 3.0. [Verkkosivu]. Zabbix SIA. [Viitattu 17.3.2017]. Saatavissa: <https://www.zabbix.com/documentation/3.0/manual/config/items>
- Zabbix. Ei päiväystä. Templating & Import/Export. [Verkkosivu]. Zabbix LLC. [Viitattu 18.3.2017]. Saatavissa: http://www.zabbix.com/template_import_export
- Zabbix. Ei päiväystä. Zabbix Documentation 3.0. [Verkkosivu]. Zabbix SIA. [Viitattu 19.3.2017]. Saatavissa: https://www.zabbix.com/documentation/3.0/manual/discovery/network_discovery
- Zabbix. Ei päiväystä. Database Monitoring. [Verkkosivu]. Zabbix LLC. [Viitattu 20.3.2017]. Saatavissa: http://www.zabbix.com/database_monitoring
- Zabbix. Ei päiväystä. SNMP and IPMI Agents. [Verkkosivu]. Zabbix LLC. [Viitattu 21.3.2017]. Saatavissa: http://www.zabbix.com/snmp_ipmi_agent
- Zabbix. Ei päiväystä. Zabbix Agent. [Verkkosivu]. Zabbix LLC. [Viitattu 22.3.2017]. Saatavissa: http://www.zabbix.com/zabbix_agent
- Zabbix. Ei päiväystä. Graphs. [Verkkosivu]. Zabbix LLC. [Viitattu 23.3.2017]. Saatavissa: <http://www.zabbix.com/graphs>
- Zabbix. Ei päiväystä. Global Dashboard. [Verkkosivu]. Zabbix LLC. [Viitattu 24.3.2017]. Saatavissa: http://www.zabbix.com/global_dashboard
- Zabbix. Ei päiväystä. Zabbix Web Frontend. [Verkkosivu]. Zabbix LLC. [Viitattu 26.3.2017]. Saatavissa: http://www.zabbix.com/zabbix_web_frontend
- Zabbix. Ei päiväystä. Zabbix Documentation 3.0. [Verkkosivu]. Zabbix SIA. [Viitattu 25.2.2017]. Saatavissa: <https://www.zabbix.com/documentation/3.0/manual/concepts/server>
- Zabbix. Ei päiväystä. Zabbix Documentation 3.0. [Verkkosivu]. Zabbix SIA. [Viitattu 27.3.2017]. Saatavissa: <https://www.zabbix.com/documentation/3.0/manual/introduction/about>
- Zabbix. Ei päiväystä. Zabbix Documentation 3.0. [Verkkosivu]. Zabbix SIA. [Viitattu 28.3.2017]. Saatavissa: <https://www.zabbix.com/documentation/3.0/manual/config/visualisation/graphs>