Haaga-Helia
ammattikorkeakoulu Oy

# Nagios XI monitorointiratkaisu Haaga-Helia ammattikorkeakoululle

Matilda Sinervo, Joonas Valsta

Opinnäytetyö
Tietojenkäsittelyn
koulutusohjelma
2019

**Tiivistelmä**

Haaga-Helia
ammattikorkeakoulu Oy

**Tekijät**
Matilda Sinervo, Joonas Valsta

**Koulutusohjelma**
Tietojenkäsittelyn koulutusohjelma

**Opinnäytetyön nimi**
Nagios XI monitorointiratkaisu Haaga-Helia ammattikorkeakoululle

**Sivu- ja liitesivumäärä**
92 + 3

Tämä opinnäytetyö toteutettiin toimeksiantona Haaga-Helia ammattikorkeakoulu Oy:lle ja sen tavoitteena oli tutkia ja toteuttaa yrityksen palvelinympäristöön sopiva monitorointiratkaisu käytännössä soveltuvuusselvityksenä. Lisäksi opinnäytetyössä pohdittiin monitoroinnin liittämistä osaksi keskeisimpiä prosesseja. Työ ajoittui keväälle 2019.

Työ jakautui kahteen eri osaan, joista ensimmäinen keskittyi toimeksiantajan organisaatioon ja sen tarpeisiin sekä monitoroinnin periaatteisiin, toimintatapoihin ja yleisimpiin teknologioihin. Lisäksi ensimmäisessä osassa tutustuttiin eri monitorointiohjelmistoihin ja valittiin niistä vertailemalla Haaga-Helian ympäristöön sopiva ohjelmisto.

Työn toisessa osassa otettiin käyttöön valittu ohjelmisto Nagios XI ja se asennettiin Haaga-Helian verkossa toimivalle erilliselle palvelimelle Nagioksen määrittelemien parhaiden käytäntöjen mukaisesti. Monitorointia varten luotiin erilaisia testitapauksia. Työn vaiheet raportoitiin osaksi opinnäytetyötä. Työn tarkoituksena oli keskittyä luomaan yksinkertaisia testitapauksia tietyille ennalta valituille palvelimille ajatuksella, että syntyvän dokumentaation pohjalta olisi mahdollista jatkokehittää monitorointi kattamaan myös muita palveluita ja palvelimia.

Osana monitoroinnin konfiguraatiota otettiin käyttöön myös erilaisiin virhetilanteisiin liittyviä hälytyksiä, jotka toimivat automatisoidusti. Hälytykset konfiguroitiin toimimaan sähköpostitse, minkä lisäksi monitoroinnin tilaa on mahdollista tarkastella web-käyttöliittymän kautta.

Lopuksi työssä käytiin läpi työn tuloksia ja esitettiin kehitysehdotuksia monitoroinnin jatkokehitystä ajatellen sekä keinoja liittää monitorointi vahvemmin osaksi jo olemassa olevia prosesseja.

**Asiasanat**
Monitorointi, Nagios, verkonvalvonta, palvelinvalvonta, palveluvalvonta

## Sisällys

## Sanasto

**AD, Active Directory** Microsoftin Windows-toimialueen käyttäjätietokanta ja hakemistopalvelu, joka sisältää tietoa käyttäjistä, tietokoneista ja verkon resursseista.

**Agentti** Monitoroitavaan laitteeseen asennettava ohjelma, joka kerää tietoa laitteesta ja lähettää sitä eteenpäin.

**Aikatietokanta, Time Series Database (TSDB)** Aikaleimoihin perustuvan tiedon säilömiseen suunniteltu tietokanta.

**Apache HTTP Server** Avoimen lähdekoodin HTTP-palvelinohjelma.

**API, Application Programming Interface** Ohjelmointirajapinta on määritelmä, minkä avulla ohjelmat voivat keskustella keskenään. Mahdollistaa muun muassa ohjelmalliset kutsut jonkin toisen ohjelman kautta.

**Blackbox monitorointi** Monitorointia, jossa monitoroitavasta kohteesta ja sen arkkitehtuurista ei ole etukäteistietoa.

**CPU, Central Processing Unit** Tietokoneen osa, joka suorittaa tietokoneohjelman sisältämiä konekielisiä käskyjä. Tietokoneen toiminnallisuuden keskeisin komponentti.

**DHCP, Dynamic Host Configuration Protocol** Verkkoprotokolla, minkä tehtävänä on jakaa IP-osoitteita lähiverkkoon kytkeytyville laitteille.

**DNS, Domain Name System** Internetin nimipalvelujärjestelmä, minkä avulla muunnetaan verkkotunnuksia IP-osoitteiksi.

**Domain Controller** Palvelin, joka vastaa esimerkiksi tunnistautumispyynnöistä toimialueen sisällä, esimerkiksi tarkistamalla kirjautumistiedot ja käyttöoikeudet.

**ESX** VMwaren kehittämä käyttöjärjestelmä, minkä avulla ajetaan virtuaalikoneita.

**ESXi** ESX palvelimen ydin, joka käsittelee vain virtuaalikoneiden toimintaa sekä niiden resurssien hallintaa.

**FOSS, Free Open Source Software** Avoimen lähdekoodin ilmainen ohjelma.

**Host** Verkkoon liitetty päätelaite, joka keskustelee verkon muiden laitteiden kanssa. Esimerkiksi palvelin tai päätelaite.

**HTML, Hypertext Markup Language** Avoimen standardin kuvauskieli, jolla kuvataan hyperlinkkejä sisältävää tekstiä, eli hypertekstiä. HTML:llä voidaan merkitä tekstin rakenne.

**HTTP, Hypertext Transfer Protocol** Protokolla, jota selaimet ja verkkopalvelimet käyttävät tiedonsiirtoon.

**HTTPS, Hypertext Transfer Protocol Secure** HTTP-protokollan ja TLS/SSL-protokollan yhdistelmä, jota käytetään tiedon suojattuun siirtoon verkon ylitse.

**HP JetDirect** HP:n kehittämä teknologia, jonka avulla tulostimia voidaan yhdistää suoraan lähiverkkoon.

**ITIL, Information Technology Infrastructure Library** Prosessikehys, joka sisältää IT-palveluiden hallinnan ja johtamisen käytäntöjä, joiden tarkoituksena on mukauttaa IT-palvelut liiketoiminnan tarpeisiin.

**LDAP, Lightweight Directory Access Protocol** Hakemistopalveluiden käyttöön tarkoitettu verkkoprotokolla, jonka yleisin käyttötarkoitus on käyttäjätunnusten ja käyttöoikeuksien autentikaatio.

**Load balancer** Tasaa palveluun tulevan työmäärän kahdelle tai useammalle palvelimelle.

**MIB, Management Information Base** Tietokanta, joka sisältää tiedot laitteen eri muuttujista.

**MySQL** Avoimen lähdekoodin relaatiotietokantaohjelmisto.

**NCPA, Nagios-Cross-Platform Agent** Nagioksen agentti.

**NRPE, Nagios Remote Plugin Executor** Nagioksen agentti.

**NSClient++** Nagioksen agentti Windows-järjestelmille.

**OID, Object Identifier** Jokaiselle verkon objektille määritelty yksilöivä tunnus. Osa MIB-kuvausta.

**Perl** Skriptimäinen ohjelmointikieli.

**PING** TCP/IP -protokollan työkalu, jonka tarkoituksena on varmistaa tietyn IP-osoitteen olemassaolo ja saavutettavuus. Saavutettavuus varmistetaan lähettämällä kohteeseen *echo request* -paketti, johon kohdekoneen tulee vastata *echo reply* -paketilla.

**Plugin** Käännetty exe-tiedosto tai skripti, joka voidaan ajaa komentoriviltä monitoroitavan kohteen statuksen varmistamiseksi.

**Probe tai poller** Sovellus tai fyysinen komponentti, jonka tehtävänä on kysellä ja lähettää eteenpäin tietoa monitoroitavan kohteen tilasta.

**RAM, Random Access Memory** Hajasaantimuisti eli luku-/kirjoitusmuisti.

**RAID 1** Kiintolevyjen vikasietoisuuttava kasvattava tekniikka, jossa kiintolevyt peilataan tallentamaan dataa kahdelle tai useammalle erilliselle levylle, jolloin data säilyy, vaikka toinen levyistä hajoaa.

**Skripti** Jonkin tehtävän ajoon tai automatisointiin suunniteltu pieni ohjelma tai erä komentoja, joka ajetaan usein komentoriviltä. Skriptit pyritään pitämään yksinkertaisina, eikä niitä tarvitse kääntää.

**SSH, Secure Shell** Salattuun tietoliikenteeseen tarkoitettu protokolla.

**SNMP, Simple Network Management Protocol** TCP/IP -verkkojen hallinnassa käytettävä protokolla, jonka rooli monitoroinnissa on kysyä ja lähettää statustietoja monitoroitavista laitteista.

**SLA, Service Level Agreement** Palvelutasosopimus, jonka tarkoituksena on määritellä jonkin palvelun vaatimustasot yhdessä asiakkaan ja palvelun tarjoajan välillä.

**SDK, Software Development Kit** Usein kokoelma työkaluja, joita hyödynnetään sovellusten kehityksessä.

**SQL, Standard Query Language** IBM:n kehittämä standardoitu kyselykieli, jolla relaatiotietokantaan voidaan tehdä erilaisia kyselyitä, muutoksia tai lisäyksiä.

**SSL, Secure Sockets Layer** Salausprotokolla, jolla suojataan verkkosovellusten tietoliikenne verkkojen yli.

**TLS, Transport Layer Security** Salausprotokolla, jolla suojataan verkkosovellusten tietoliikenne verkkojen yli. Korvaa vanhemmat SSL-versiot.

**Webapp** Jonkin ohjelmiston tain palvelun verkkokäyttöliittymä.

**Whitebox monitorointi** Monitorointia, jossa monitoroinnin kohde ja sen arkkitehtuuri tunnetaan hyvin.

**WMI, Windows Management Instrumentation** Microsoftin kehittämä infrastruktuuri standardisoidun datan hallinnointiin ja seurantaan ilman tarvetta erilliselle agentille.

**VMware** Virtualisointiohjelmistoja kehittävä yritys.

# 1 Johdanto

Palvelujen monitorointi on nykyaikana yhä tärkeämpää, sillä monet yritykset tuottavat palvelujaan tuotteena asiakkailleen tai muille yrityksille. Työskentely ylipäätään on siirtynyt valtaosin digitaaliseksi. Liiketoiminnalle kriittisiä palveluja tulisi pystyä valvomaan automatisoidusti sekä luomaan hälytyksiä niiden käyttökatkoksista, jotta mahdolliset vikatilat pystytään ratkaisemaan mahdollisimman nopeasti.

Tämä opinnäytetyö perustuu Haaga-Helia ammattikorkeakoulun toimeksiantoon ja sen tarkoituksena on kehittää organisaation verkon, palvelimien ja palveluiden monitorointia.

Haaga-Helialla on käytössään oma palvelinkonesali, jossa valtaosa Haaga-Helian palvelimista sijaitsee. Näillä palvelimilla sijaitsevat myös useimmat Haaga-Helian tarjoamista palveluista. Palveluiden tarjoaminen itse oli yksi merkittävä syy sille, että monitoroinnin tasoa haluttiin korottaa.

Haaga-Helialla on useampi palvelin, laite tai palvelu, jotka kaipaavat jonkinlaista monitorointia. Palvelimia ja palveluita on kymmeniä, mutta niistä monitoroitiin työn aloitushetkellä vain muutamaa. Monitorointia toteutettiin hyvin perustasolla ja valvottavista kohteista saatettiin tarkkailla vain palvelinten päällä oloa. Yksittäisten verkkopalveluiden monitorointia oli aikaisemmin toteutettu GFI:n EventsManager -työkalulla, joka lataa verkkosivuja ja tarkastelee sivujen lähdekoodia tiettyjen avainsanojen perusteella; jos avainsanaa ei löydy, työkalu lähettää siitä vastaavalle henkilölle hälytyksen sähköpostitse. GFI:n työkalulla oli lisäksi toteutettu kytkinten ping-testejä sekä joidenkin palvelinten käyttöasteiden valvontaa.

Haaga-Helia oli jo aiemmin hankkinut lisenssin Nagios XI -monitorointityökaluun, jota ei kuitenkaan toistaiseksi ollut otettu kattavasti käyttöön yrityksen verkkoinfrastruktuurissa ja palvelinympäristössä. Nagiokseen liitettyjä palvelimia, joihin oli asennettu peruskäyttöä seuraava Nagioksen agentti, olivat DHCP-palvelimet sekä muutama VMwaren ESX -palvelin. Muita Nagioksen piiriin liitettyjä palvelimia seurattiin vain pingin avulla.

Toimeksiantaja määritteli muutamia kriteerejä monitorointipalvelulle, joista yhtenä kriteerinä oli se, että palveluiden monitorointi tapahtuisi keskitetysti, eikä niin, että päällekkäistä monitorointia tapahtuu samanaikaisesti monessa eri paikassa.

Haaga-Helian tarjoamien verkkopalveluiden monitorointi onkin Haaga-Helian kannalta organisaationa erittäin hyödyllistä, sillä opetus, eli Haaga-Helian suurimman sidosryhmän

käyttämä päätoiminen palvelu, lamaantuu herkästi, jos opetuksen kannalta kriittiset palvelut ovat alhaalla. Tämän vuoksi verkkopalvelujen monitorointi ja niiden vikatiloista hälyttäminen on erittäin tärkeää. Monitoroinnin avulla voitaisiin myös kerryttää niin sanotusti korkean saatavuusasteen statistiikkaa, jonka avulla voidaan vertailla vuosittaista palvelujen alhaalla oloaikaa, jota voitaisiin hyödyntää myös investoinnin sidosryhmien palavereissa niin sanottuna palveluindikaationa.

Opinnäytetyön tavoitteena oli vertailla erilaisia monitorointiratkaisuja ja -menetelmiä sekä lopulta toteuttaa varsinainen monitorointiratkaisu Haaga-Helian palvelinympäristöön. Monitorointiratkaisu päätettiin rajata kattamaan jokin tietty palvelin tai palvelu, joka dokumentoitiin kattavasti niin, että dokumentaation pohjalta on mahdollista jatkaa monitoroinnin jatkokehitystä ja ottaa monitorointi osaksi tietohallintoyksikön prosesseja. Päädyimme keskusteluissa toimeksiantajan kanssa rajaamaan monitoroinnin lopulta ensisijaisesti opetuskäytössä tärkeän Moodle-palvelun kehitysympäristön palvelimeen. Rajauksen perusteena oli ympäristön laajuus ja opinnäytetyön rajallinen resurssimäärä sekä palvelun tärkeys liiketoiminnan näkökulmasta. Lisäksi opinnäytetyötä tehtiin ilman varsinaista pääsyä monitoroitaville palvelimille, joten käytännön syistä oli järkevämpää tehdä yhteistyötä jonkin tietyn palvelimen ylläpitäjän kanssa sen sijaan, että projekti olisi työllistänyt useampia yksikön työntekijöitä.

Työn tekijöillä, Matilda Sinervolla ja Joonas Valstalla, oli aiemman Haaga-Helian tietohallintoyksikössä suoritetun työharjoittelun pohjalta jo olemassa olevaa tietoa yrityksestä ja sen toimintatavoista ja menetelmistä sekä käsitys yksikön haasteista ja tarpeista. Olimme aiemmin opintojemme yhteydessä toteuttaneet Haaga-Helian virtualisointipalvelimia monitoroivan harjoitustyön, joka omalta osaltaan loi kipinän myös varsinaisen palvelinmonitoroinnin toteutukselle samassa ympäristössä. Näiden kokemuksien pohjalta uskoimme, että meillä oli hyvä mahdollisuus lähteä kehittämään monitorointiratkaisua, joka sopii osaksi Haaga-Helian prosesseja.

Opinnäytetyö oli luonteeltaan työtoiminnan kehittämishanke, missä työn tekijät olivat perehtyneet monitorointiin ja siinä käytettäviin menetelmiin sekä työkaluihin niin käytännön työn kuin kirjallisuusselvitysten kautta.

Opinnäytetyö toteutettiin parityönä, jonka vuoksi suunnittelimme työtä tehdessämme työnjakoa niin, että molemmat osapuolet pyrkivät toteuttamaan joko määrällisesti tai haasteellisesti yhtä kattavan osion. Työnjaon raportointi kattaa siis konkreettisen dokumentaation lisäksi myös teknisen toteutuksen ja sen toteuttamiseksi tehdyn selvitystyön. Jotkin työn osiot olivat luonteeltaan sellaisia, että niiden osoittaminen vain toiselle henkilölle ei olisi

ollut järkevää. Tällaisia syitä olivat esimerkiksi osion haastavuus tai työn kulkua pohdiskelevat osiot, joissa oli oleellista saada kummankin osapuolen näkökulma esille. Lisäksi pyrimme jakamaan työtä niin, että kummallakin oli mahdollisuus päästä toteuttamaan omia oppimistavoitteitaan. Työnjako on esitetty taulukkona liitteessä 1 ja sen rakenne pohjautuu pitkälti opinnäytetyön sisällysluettelon rakenteeseen. Työn aikana toteutimme tehtävien jakoa ja niiden edistymisen seurantaa hyödyntäen Trello-projektinhallintatyökalua.

## 2 Haaga-Helia toimeksiantajana

Haaga-Helia on Suomen pääkaupunkiseudulla ja sen lähialueilla toimiva ammattikorkeakoulu, jolla on kampuksia Pasilassa, Malmilla, Haagassa, Porvoossa sekä Vierumäellä (Haaga-Helia ammattikorkeakoulu 2019, 39). Oppilaitoksesta valmistuu asiantuntijoita muun muassa liiketalouden, hyvinvoinnin ja digitaalisten palveluiden aloille sekä opetustyön ammattilaisia (Haaga-Helia ammattikorkeakoulu 2019, 9-14).

Haaga-Heliassa opiskelee vuosittain noin 9000 ammattikorkeakoulututkintoon tähtäävää opiskelijaa, jonka lisäksi vuonna 2018 oli noin 950 ylemmän ammattikorkeakoulututkinnon opiskelijaa ja 530 opettajakorkeakoulun opiskelijaa (Haaga-Helia ammattikorkeakoulu 2019, 21). Henkilöstön jäseniä Haaga-Heliassa oli vuonna 2018 yhteensä 648 (Haaga-Helia ammattikorkeakoulu 2019, 27).

Tietohallintopalveluiden yksikkö vastaa Haaga-Helian tietojärjestelmien kehittämishankkeista. Vuonna 2018 kehitystyötä tehtiin ahkerasti, kun Haaga-Helia uusi niin opetukseen kuin hallinnollisiin toimintoihinsa liittyviä palveluita kokonaisvaltaisesti. Uusina työkaluina otettiin käyttöön muun muassa uusi toiminnanohjausjärjestelmä sekä projektinhallintajärjestelmä. Lisäksi uusittiin myös talon sisäisen IT-asiakaspalvelun ylläpitämä helpdesk-portaali. (Haaga-Helia ammattikorkeakoulu 2019, 25.)

# 3 Monitoroinnin hyödyt Haaga-Heliassa

Projektia aloittaessamme oli oleellista miettiä kuinka Haaga-Helia voi konkreettisesti hyötyä monitoroinnin toteutuksesta. Monitoroinnin voi helposti ajatella koskettavan vain suoraan monitorointipalvelua ylläpitävää tahoa tai henkilöä, mutta todellisuudessa uskomme monitoroinnin hyödyttävän koko organisaatiota ja jopa sen ulkopuolisia tahoja.

Haaga-Heliassa monitorointia hyödyntävät konkreettisimmin IT-asiakaspalvelu, eli helpdesk, sekä palveluista vastaavat järjestelmien asiantuntijat, eli ylläpitäjät. Helpdesk vastaa tietohallintopalveluiden asiakaspalvelusta, vikatilojen ja muutoksien informoinnista sekä loppukäyttäjien tietoteknisten ongelmien ratkaisusta. Helpdeskin asiakkaat, eli käytännössä Haaga-Helian henkilöstö ja opiskelijat, ovat usein suoraan yhteydessä helpdeskiin tai seuraavat jotakin sen ylläpitämistä tiedotuskanavista. Tiedotusta voidaan toteuttaa myös talon ulkopuolelle yhteistyöorganisaatioille tai muuten ammattikorkeakoulun palveluita hyödyntäville tahoille.

Helpdeskin toiminta on sidoksissa tietohallintopalveluiden muuhun toimintaan ja erityisesti järjestelmien asiantuntijoiden toimintaan. Järjestelmien asiantuntijat vastaavat palveluiden ylläpidosta ja kehityksestä. Myös järjestelmien asiantuntijoilla on oma roolinsa vikatiloista tiedottamisessa ja erityisesti niihin reagoinnissa. Kommunikaatioyhteyden sujuvoittamiseksi ja nopeuttamiseksi monitorointiratkaisulla voidaan poistaa tai ainakin vähentää tiedotuksesta ylimääräisiä vaiheita tuottamalla monitorointia, joka on laajasti saatavilla sekä yksikön muulle henkilöstölle, kuten helpdeskille, tai halutessa jopa suoraan loppukäyttäjille.

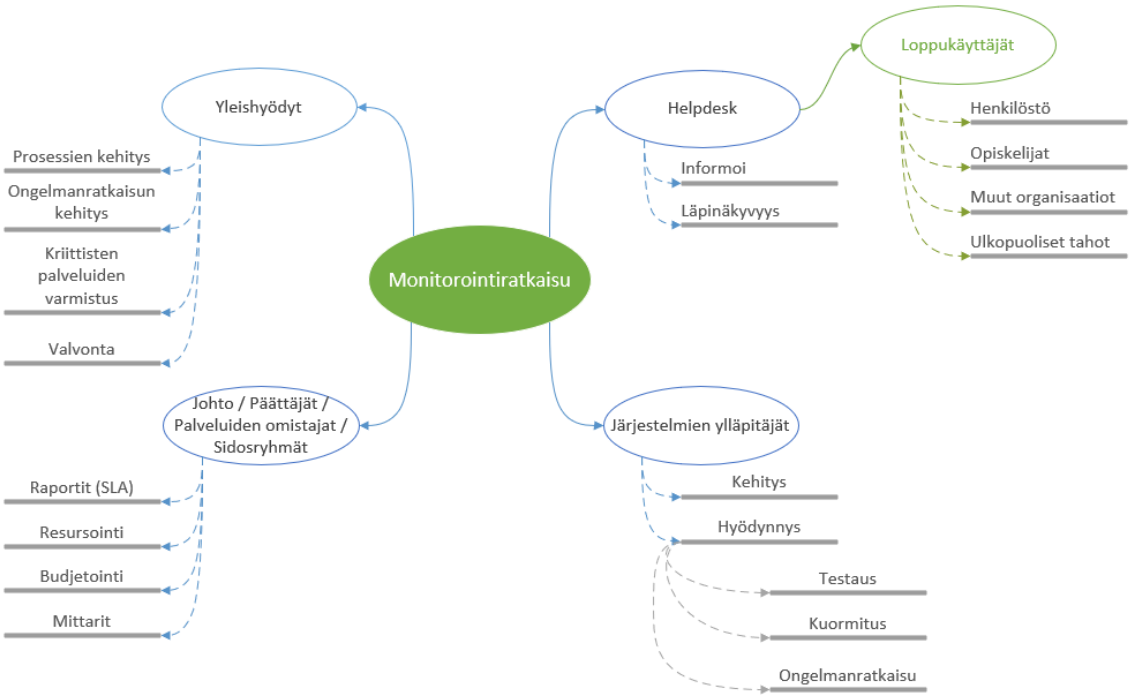
Siinä missä helpdesk ja järjestelmien asiantuntijat ovat yleensä ensimmäiset, joita monitoroinnin tuottama data koskettaa, on tieto oleellista myös hallinnollisella tasolla. Johtoryhmä suunnittelee ja neuvottelee tulevaisuuden ratkaisuista pitkälti aiempaan perustuvan raportoinnin perusteella. Monitoroinnin avulla palveluiden toiminnasta ja saatavuudesta saadaan konkreettista dataa, jonka pohjalta toimintaa on helpompi arvioida ja päätöksenteko perustuu varmistettuun tietoon.

On helppo ajatella, että monitoroinnilla tarkoitetaan ensisijaisesti vain suorituskyvyn mittaamista tai muuta komponenttien tarkkailuun kohdistettua toimintoa. Todellisuudessa monitorointi voidaan ottaa osaksi prosesseja laajemminkin. Monitorointidataan perustuvalla tiedolla voi olla organisaatiossa suuri merkitys prosessien ja ongelmanratkaisukyvyn kehityksessä sekä kriittisten prosessien valvonnassa ja varmistamisessa jo ennakoivasti.

## 3.1 Sidosryhmät

Sidosryhmäteoria yleistyi ensimmäisen kerran R.E. Freemanin vuonna 1984 julkaiseman kirjan *Strategic Management: A Stakeholder Approach* myötä (Mitchell, Agle & Wood 1997, 853). Sittemmin sidosryhmäteorian tarkoituksesta ja merkityksestä on käyty laajaa keskustelua ja on todettu, että eri tahot hyödyntävät sitä eri tavoin ja tarkoituksin (Donaldson & Preston 1995, 66). Jo pelkästä sidosryhmän määritelmästä on lukuisia eri variaatioita, joista yksi on Freemanin ajatus, jossa sidosryhmällä tarkoitetaan ryhmää tai henkilöä, joka joko vaikuttaa organisaation tavoitteiden onnistumiseen tai johon saavutetut tavoitteet suoraan vaikuttavat (Mitchell ym. 1997, 854).

Donaldson ja Preston esittelevät artikkelissaan kaksi mallia sidosryhmien kuvaamiseksi. Toisessa sijoittajat, työntekijät ja tavarantoimittajat ovat toimintaa tukevia sidosryhmiä, kun taas asiakkaat ovat toiminnasta hyötyvä sidosryhmä. Toinen malli taas korostaa sitä, että kaikki osapuolet hyötyvät toisistaan. (Donaldson & Preston 1995, 68.) Valitsimme oman pohdintamme perustaksi mallin, jossa kaikki osapuolet hyötyvät toisistaan, sillä ajattelemme sen olevan myös monitoroinnin tavoite. Sidosryhmämallinnukseen otettiin mukaan sekä yrityksen ulkoisia että sisäisiä sidosryhmiä. Kuvassa 1 on mallinnettu sidosryhmiä ja niille ajateltuja hyötyjä miellekarttaan.

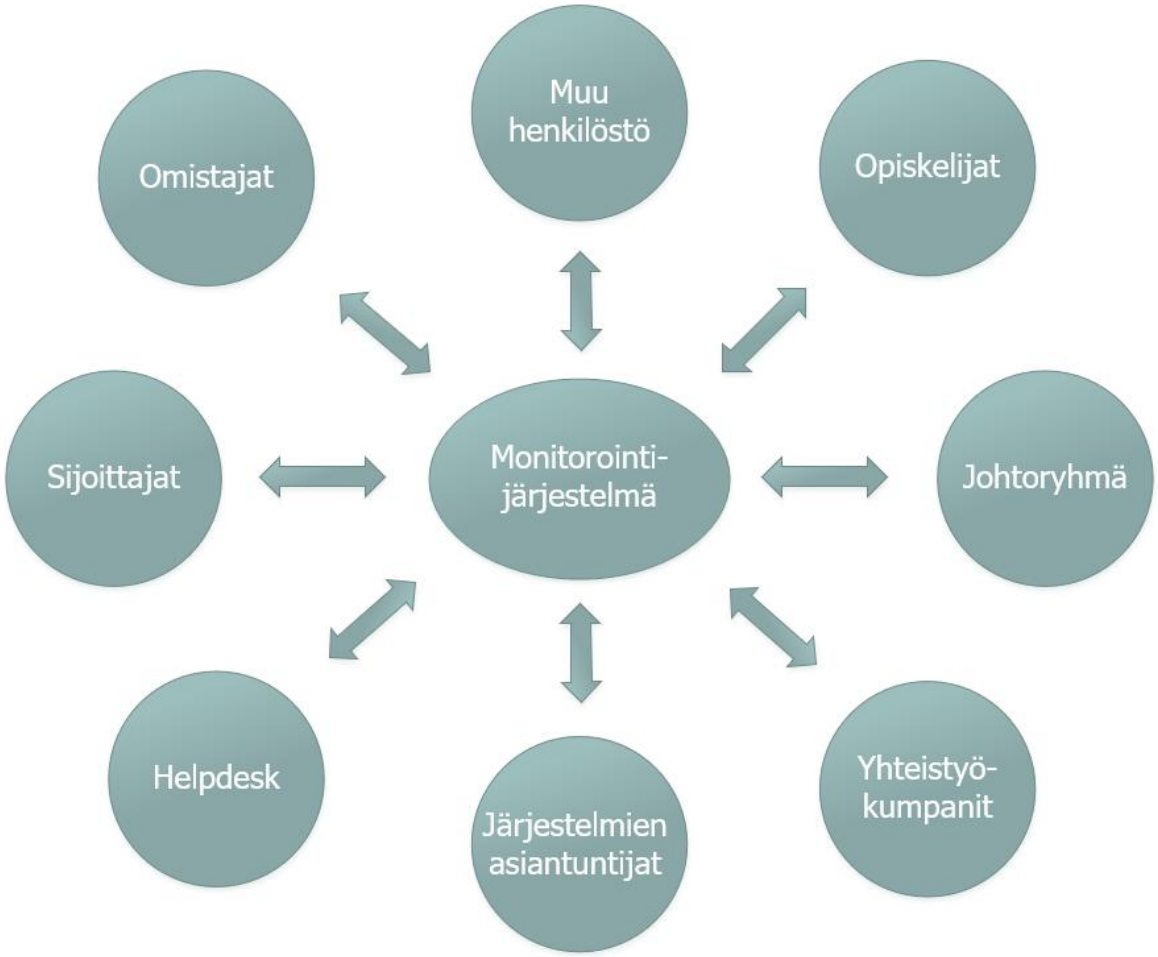



Kuva 1. Monitorointiin liittyvät ryhmät ja niiden roolit miellekarttana

Ajattelemme monitorointijärjestelmän sidosryhmiksi ensisijaisesti kuvassa 2 esitetyt järjestelmien asiantuntijat, helpdeskin, opiskelijat, muun henkilöstön, johtoryhmän, yhteistyökumppanit, omistajat sekä sijoittajat. Sidosryhmiä on pohdittu koko Haaga-Helian näkökulmasta ja mallinnusta olisi mahdollista viedä vielä yksityiskohtaisemmallekin tasolle, esimerkiksi liittäen mukaan erilaisia talon sisäisiä sidosryhmiä. Rajasimme sidosryhmämallinnuksen kuitenkin kattamaan laajempia kokonaisuuksia, sillä yksityiskohtaisempi määritys ei olisi ollut mahdollista ilman kattavampaa perehtymistä Haaga-Heliaan organisaationa ja sen prosessikäytäntöihin.

Helpdesk ja järjestelmien asiantuntijat hyödyntävät ja kehittävät monitorointia työssään konkreettisimmin tarkastelemalla eri palveluiden ja palvelinten tilaa. Toisaalta monitorointijärjestelmä edesauttaa myös asiakaspalvelun laatua ja kykyä reagoida erilaisiin ongelmatilanteisiin. Opiskelijat ja muu henkilöstö ovat monitorointijärjestelmän piiriin kuuluvien palveluiden loppukäyttäjiä, joille monitorointijärjestelmän ansiosta voidaan tarjota parempaa tietoa järjestelmän tilasta sekä varmistaa palveluiden saatavuus. Opiskelijat ja henkilöstö taas osaltaan kykenevät ylläpitämään ja kehittämään organisaation muita prosesseja silloin kun järjestelmät toimivat. He voivat lisäksi toimia monitorointidatan lähteenä ja tämän kautta edistää erilaisten monitoroitavien kohteiden kehitystä tai testitapausten luontia. Yhteistyökumppanien saama hyöty on hyvin vastaava kuin opiskelijoiden ja henkilöstön, mutta lisäksi monitorointia voidaan kehittää puolin ja toisin myös yhteistyöprojektien ja yhteistyöorganisaatioiden ideoinnin kautta, erityisesti mikäli yhteistyöorganisaation tarjoamat palvelut ovat vastaavia Haaga-Helian kanssa.

Omistajat ja johtoryhmä hyötyvät monitorointijärjestelmästä saamalla konkreettista tietoa yrityksen toiminnan kannalta kriittisten järjestelmien ja palveluiden tilasta ja palvelutasoista. Monitorointijärjestelmän kehitys taas vaatii resursseja, joita nämä sidosryhmät voivat päätöksenteossaan osoittaa eri projektien kehitykselle. Sijoittajat puolestaan saavat sijoitukselleen paremman vastineen, kun palveluiden saatavuus on korkea. Vastaavasti toimivan monitorointiratkaisun myötä saatavuustasoa voidaan nostaa, jolloin hyöty on molemminpuolinen.




Kuva 2. Monitorointijärjestelmään liittyvät sidosryhmät

### 3.2 Prosessikuvaus

Monitorointia voidaan hyödyntää monissa eri prosesseissa, mutta valitsimme opinnäytetyössä esitettäväksi esimerkkikuvauksen it-tukipalvelua tarjoavan yrityksen mahdollisesta tavoitetilasta, jossa monitorointi on liitetty osaksi prosessia. Haaga-Helian omat prosessit eivät ole esitetyn kuvauksen kanssa identtisiä, vaan monitorointia laajemmin käyttöönottaessa on syytä käydä prosesseja erikseen läpi, jotta monitorointi saadaan liitettyä järkevällä tavalla osaksi niitä. Prosessi on kuvattu uimaratakaaviona liitteessä 2. Uimaratakaavio on toteutettu hyödyntäen *Process Modeling Notations and Workflow Patterns* -tutkimusta, joka vertailee BPMN (Business Process Model and Notation) ja UML 2.0 (Unified Modeling Language*)* -mallinnustapojen hyödyntämistä osana prosessikuvausta (White 2008, 1-25).

Mikäli organisaatiossa hyödynnetään monitorointia, voi prosessi saada alkunsa joko monitorointijärjestelmän hälytyksestä tai loppukäyttäjän ilmoituksesta. Toimivan monitoroinnin yhteydessä tulisi monitorointijärjestelmän hälyttää ongelmasta riippumatta siitä, tuleeko tieto sen lisäksi myös loppukäyttäjän suunnalta.

Loppukäyttäjän huomatessa ongelman ottaa hän yleensä yhteyttä helpdeskiin, joka vastaanottaa tukipyynnön ja arvioi ongelman perusteella jatkosuunnitelman sen ratkaisemiseksi. Mikäli loppukäyttäjän esittämän ongelmatilanteen voi ratkaista heti, ilman järjestelmätason tuntemusta, lähettää helpdeskin työntekijä loppukäyttäjälle ehdotuksen ongelman ratkaisemiseksi. Mikäli ongelma ei ratkennut vielä ensimmäisen saadun ratkaisuehdotuksen pohjalta, voi loppukäyttäjä avata tukipyynnön uudelleen. Mikäli ongelma kuitenkin ratkeaa heti, voidaan tukipyyntö kuitata ratkaistuksi ja prosessi päättyy. Mikäli ongelma ei vielä ratkennut, arvioi helpdeskin työntekijä uudelleen, onko ongelma ratkaistavissa hänen toimestaan uuden ratkaisuehdotuksen myötä vai eskaloidaanko tukipyyntö seuraavalle tasolle. Mikäli tukipyyntö päätetään siirtää seuraavalle tasolle, siirtää helpdeskin työntekijä sen jollekin ongelman kohteena olevan järjestelmän asiantuntijoista. Asiantuntija vastaanottaa tukipyynnön ja arvioi, onko kyseessä yksittäinen juuri tätä loppukäyttäjää koskeva ongelma, vai laajempi häiriö, joka vaikuttaa myös muihin käyttäjiin. Mikäli kyseessä on yksittäinen ongelma, pyrkii asiantuntija ratkaisemaan sen itsenäisesti ja lähettämään tiedon ratkaisusta loppukäyttäjälle. Samoin kuin aiemmassa vaiheessa, prosessi päättyy, kun toimiva ratkaisu on löydetty.

Mikäli järjestelmän asiantuntija toteaa ongelman koko järjestelmää koskevaksi laajemmaksi häiriöksi, laatii hän tilanteesta tiedotteen it-palveluista vastaavan yksikön tiedoksi sekä informoi asiasta eteenpäin esimiehelleen. Esimies laatii ongelmasta tarvittaessa tiedotteen loppukäyttäjille ja pohtii sopivaa lähestymistapaa ongelman ratkaisemiseksi. Järjestelmän asiantuntijat pyrkivät tämän jälkeen ratkomaan ongelman oman osaamisensa ja saamiensa ehdotusten pohjalta. Ongelman ratkettua järjestelmän asiantuntija laatii tilanteesta uuden tiedotteen sekä yksikön, että esimiehen tiedoksi, ja ilmoittaa ongelman ratkeamisesta loppukäyttäjälle. Mikäli tilanteesta oli aiemmin laadittu tiedote, laatii esimies tiedotteeseen nyt päivityksen ongelman ratkeamisesta. Mikäli tieto ongelmasta oli havaittavissa myös monitorointijärjestelmän kautta, kuittaa järjestelmän asiantuntija hälytyksen ratkaistuksi monitorointijärjestelmästä.

Tapauksessa, jossa ensitieto häiriöstä saadaan monitorointijärjestelmän kautta, monitorointijärjestelmä generoi virhetilanteesta hälytyksen, joka lähetetään eteenpäin ennalta määriteltyjä kanavia, kuten esimerkiksi sähköpostia, hyödyntäen. Järjestelmän asiantuntija tai toiminnanohjausjärjestelmä vastaanottaa tämän jälkeen hälytyksen. Ensimmäinen jatkotoimi hälytykselle on tilanteen arviointi samoin kuin loppukäyttäjän ilmoittaman hälytyksen kohdalla, eli määritellään, onko kyseessä laajempi häiriö vai ei. Hälytysviestit kirjataan monitorointiratkaisuun tiedostetuiksi ongelmiksi, jolloin monitorointijärjestelmä ei lä-

hetä ongelmatilanteesta lisää hälytyksiä, jos niin on määritelty. Ongelman ratkettua monitorointijärjestelmä huomaa sen ratkenneen itsenäisesti omien tarkastuksiensa avulla, ja merkitsee monitoroitavan kohteen järjestelmästä ratkaistuksi. Monitorointijärjestelmä voidaan myös konfiguroida niin, että se lähettää ilmoituksen, kun monitoroitava kohde on taas raja-arvojen sisäpuolella. Näin voidaan varmistua siitä, että ongelma on ratkaistu myös monitorointijärjestelmän mielestä.

Erityisesti laajempien häiriötilanteiden jälkeen, tai tietyin aikavälein osana normaalia yksikön prosessia, monitorointijärjestelmästä voidaan hakea raportteja. Monitorointijärjestelmä voi lähettää raportit itse, tai vaihtoehtoisesti järjestelmän asiantuntija tai yksikön esimies voi itse generoida raportin määrittelemillään kriteereillä. Tämän jälkeen monitorointijärjestelmä luo raportin, jota voidaan hyödyntää esimerkiksi osana yksikön prosessien kehitystä ja ongelmatilanteiden läpikäyntiä. Raporttien myötä myös palvelun omistaja saa arvokasta tietoa yksikön prosessien ja toimintojen kulusta esimerkiksi suorituskykymittareiden muodossa.

## 3.3 Monitoroinnin tuottamat mittarit

Monitoroinnin avulla voidaan tuottaa dataa, jonka avulla voidaan määritellä ja korreloida eri tapahtumien vaikutusta organisaation verkkopalveluissa suorituskykymittarien muodossa. Suorituskykymittarilla tarkoitetaan mittaria, joka indikoi miten hyvin organisaation toiminta on sujunut verrattuna sille määriteltyyn toiminnan tasoon. (Julian 2017, luku 5.1.)

Monitorointi mittareineen ja mittaustuloksineen hyödyttää monia osapuolia, mutta erityisesti mittaustulokset auttavat tietohallintoyksikköä sen raportoidessa muun muassa palveluiden saatavuusasteesta palvelun omistajalle tai johtoryhmälle. Palveluiden saatavuuden paraneminen on myös viesti yrityksen omistajille ja rahoittajille, kun organisaation vuosikertomuksessa esitetään konkreettisia tietoja palvelinten ja palveluiden ylläpitobudjettina. Muita mittareita voivat olla esimerkiksi taulukossa 1 esittämämme käyttäjämäärät tai käyttäjien viettämä aika palveluissa ja niiden korrelaatio esimerkiksi palvelun palvelinpuolella, eli *back-endissä*, tapahtuneisiin ongelmiin.

Taulukko 1. Ehdotuksia Moodle-verkkopalvelun mittareiksi ja niiden korrelaatioiksi

| Liiketoiminnan mittari | Tekniset mittarit |
|---|---|
| Käyttäjät palvelussa | Tilannekohtainen käyttäjämäärä palvelussa |
| Käyttäjien kirjautuminen | Ongelmatilanteet kirjautumisessa, kirjautumisen latenssi |
| Käyttäjien ongelmatilanteet | Ongelmatilanteet tiedostojen, kommenttien, verkkotentien ja muiden palautuksien lisäämisessä |
| Käyttäjien tyytyväisyys | Sivuston latautumisnopeus ja latenssi |
| Palvelun saatavuusaste | Palvelun saatavuus ja palvelimen ylhäällä oloaika |
| Palvelun ongelmatilanteet | Tietokantavirheet, load balancer ongelmat tai muut backend-ongelmatilanteet |

Mikäli palvelua tarjoavan palvelimen ongelmat ja käyttäjämäärän pudotus tai käyttäjien kirjautumisvirheet kohtaavat, voidaan analysoida niiden merkitystä toisiinsa ja mahdollisesti perustella suurempaa resursointia palvelimen tai palvelun kehitykseen ja ylläpitoon. (Julian 2017, luku 5.2.)

# 4 Monitoroinnista yleisesti

Monitoroinnilla tarkoitetaan prosessia, jonka tarkoituksena on ylläpitää valvontaa järjestelmän tilan ja tietovirran muutoksien olemassaolosta ja laajuudesta. Seurannan tarkoituksena on tunnistaa vikatilat ja helpottaa ongelmanratkaisua. Tiedon keruuseen, sen käsittelyyn ja esitykseen käytettäviä ohjelmistokomponentteja kutsutaan monitorointijärjestelmäksi. Hälytykset ovat monitorointijärjestelmän ominaisuus, joiden avulla huomatuista ongelmista voidaan tiedottaa asianmukaisia tahoja. (Ligus 2012, luku 1.)

## 4.1 Monitoroitavat kohteet

Tietotekniikassa monitoroinnilla tarkoitetaan yleisesti palvelimien, laitteiden, komponenttien, palvelujen sekä verkon valvontaa.

Palvelimien tyypillisiä monitoroitavia aiheita ovat komponenttien, kuten prosessorin, muistien, kovalevyjen ja verkkosovittimien käyttöasteet. Myös palvelimella ajettavien prosessien ja käyttäjien luomaa kuormaa voidaan eritellä ja monitoroida. Palvelimia monitoroidaan yleensä niiden kapasiteettien suunnittelun kannalta, jotta tulevaisuudessa niiden resursseja voidaan optimoida. Kerättyä dataa voidaan lisäksi hyödyntää myös mahdollisten vikojen etsimiseen komponenttien osalta. (Julian 2017, luku 8.1.)

Palveluiden monitoroitavia aiheita ovat tyypillisesti käytettävyysaika, saatavuus ja vasteaika, jotka muodostavat yleiskuvan palvelun suorituskyvystä. Suorituskyvystä kerätään dataa, jota analysoimalla voidaan osoittaa, onko palvelun suorituskyky laskussa esimerkiksi jonkin versiopäivityksen jälkeen. Esimerkkinä voidaan pohtia käyttääkö uusi versio palvelusta mahdollisesti enemmän palvelimen resursseja, joka aiheuttaa pullonkaulan resurssien osalta. Vaikuttaako kyseinen versiopäivityksestä aiheutunut suorituskyvyn lasku esimerkiksi palvelun latensseihin, huonontaen käyttäjäkokemusta ja laskien käyttäjien tyytyväisyyttä. Nämä kaikki seikat ovat monitoroitavissa ja niitä tulisi hyödyntää palveluiden kehityksessä ja optimoinnissa. (Julian 2017, luku 2 & luku 6.)

Tyypillisesti monitoroitaviin laitteisiin eivät kuulu käyttäjien päätelaitteet, kuten tietokoneet, tabletit tai puhelimet, sillä niiden laajan volyymin tuoma taakka olisi hankala resursoida. Erilaisia yksittäisiä laitteita, kuten kriittisiä tulostimia, voidaan kuitenkin tarpeen vaatiessa lisätä monitoroitaviksi kohteiksi, jos ne täyttävät monitoroinnin vaatimat edellytykset. Esimerkiksi toimeksiantajan infrastruktuurista löytyy tällaisia tulostimia, jotka ovat opetuksen ja muun toiminnan kannalta tärkeitä.

## 4.2 Monitorointi ja ITIL

Monitorointia voidaan ajatella myös eräänlaisena tapana avustaa ongelmien käsittelyssä. Ongelmien tai tapahtumien, eli niin sanottujen incidenttien, hallinta on tärkeä osa nykypäivän liiketoimintaa. Tunnetuin näistä hallintatavoista IT-maailmassa on ITIL eli Information Technology Infrastructure Library, joka on prosessikehys, joka sisältää IT-palveluiden hallinnan ja johtamisen käytäntöjä. ITIL:n päätarkoitus on avustaa ja suunnata IT-palveluita toimimaan osana yrityksen liiketoimintaa. (Paul 2009, 5.)

ITIL tarjoaa prosessikehyksen tapahtumien hallinnalle, johon monitorointi voidaan liittää osaksi käytettävää tapahtumaketjua. Monitorointi tarjoaa mahdollisuuden vaikuttaa tapahtumien hallintaan eri tavoin sen eri vaiheiden mukaan, esimerkiksi ongelmien tunnistamisen, kirjausten tai analysoinnin myötä.

Monitorointia voidaan hyödyntää ongelmien tunnistamisessa jo ennen ongelman tapahtumahetkeä tai hälytysten muodossa ongelmien sattuessa. Kirjaamisella tarkoitetaan monitoroinnin tuottamien tietojen hyötyä silloin, kun tapahtumia tuodaan esiin. Tapahtui kirjaus sitten sähköpostin muodossa hälytyksenä tai tukipyynnön automaattisena luomisena toiminnanohjausjärjestelmään, monitorointi voi tarjota hyödyllistä lisätietoa ongelman syystä tai sijainnista. Analysoinnilla tarkoitetaan tapahtumien diagnosointia monitoroinnin avulla, kuten edellä mainittu monitoroinnin tarjoama näkökulma tapahtuneesta tapahtumahetkellä. (Julian 2017, luku 3.2.)

Kokonaisvaltainen monitorointi voi olla hyödyllistä myös erilaisten ongelmien juurisyitten löytämisessä ja ratkaisussa. Mikäli dataa on kerätty suurella otannalla, jolloin tietoja on saatavilla myös palveluun sidoksissa olevista muista komponenteista, voidaan tuota dataa hyödyntää osana ongelmanratkaisua. Voidaan esimerkiksi pohtia, onko ongelman syynä esimerkiksi se, että jonkin palvelun integraatioon liittyvän tärkeän komponentin suorituskyky on laskenut toiminnan kannalta kriittisellä hetkellä. (Julian 2017, luku 3.2.)

## 4.3 Monitoroinnin periaatteet

Julian (2017, luku 2.1) esittelee kirjassaan *Practical Monitoring* miten tietotekniikassa monitorointi voidaan jakaa viiteen eri komponenttiin tai vaiheeseen: datan keruu, datan säilöntä, datan visualisointi, analytiikka ja raportointi sekä hälyttäminen.

### 4.3.1 Datan keruu

Dataa voidaan kerätä yleisesti monella eri tavalla, mutta se miten kerättyä dataa siirretään monitorointiratkaisuun, voidaan lähtökohtaisesti tehdä kahdella eri tavalla. Siirto voidaan tehdä joko puskemalla (push) sitä monitorointipalveluun tai niin, että monitorointipalvelu niin sanotusti vetää dataa kohteestaan (pull). (Turnbull 2016, 24-26.)

Puskumallissa monitoroitava kohde lähettää kerättyä dataa itse monitorointiratkaisuun. Tätä voidaan tehdä joko tietyin väliajoin tai esimerkiksi silloin, kun jokin lokitapahtuma ilmenee. Puskumallin monitorointiratkaisu on helposti skaalautuva, sillä jokainen monitoroitava kohde lähettää datan itsenäisesti tiettyyn paikkaan, josta monitorointiratkaisu pystyy käsittelemään kerättyä dataa. Puskumallin monitorointiratkaisu on erityisen järkevä toteuttaa pilvipalveluissa, sillä tällöin jokaista kohdetta ei tarvitse tietää etukäteen, kun kohteet tekevät tiedonsiirron itse niin sanottuun keskusvarastoon. Puskumallissa käytetään agentteja, joilla tarkoitetaan monitorointiratkaisun jatketta, jonka tarkoituksena on keskustella monitorointiratkaisun palvelun kanssa. (Julian 2017, luku 2.1.)

Puskumallin vastakohta on vetomalli, jossa monitorointiratkaisu hakee itse dataa monitoroitavista kohteista pyytämällä jokaista kohdetta erikseen lähettämään tietoa itsestään. Vetomallissa käytetään erilaisia protokollia datan kyselyä varten, kuten SNMP- tai WMI-protokollia, joita käsitellään myöhemmin opinnäytetyön luvuissa 4.6 ja 4.7. Käytännössä verkkolaitteiden, kuten kytkinten tai reitittimien, monitoroinnissa ainoa vaihtoehto on usein vetomalli, sillä näiden laitteiden käyttöjärjestelmät eivät välttämättä tue muita menetelmiä. (Julian 2017, luku 9.1.) Vetomalli ei kuitenkaan skaalaudu yhtä hyvin kuin puskumallin monitorointi, sillä vetomallissa monitorointiratkaisun täytyy ennalta tietää kaikki monitoroitavat kohteet, niiden osoitteet, ajankohta datan keruulle sekä kirjautumistiedot. (Julian 2017, luku 2.1.)

### 4.3.2 Datan säilöntä

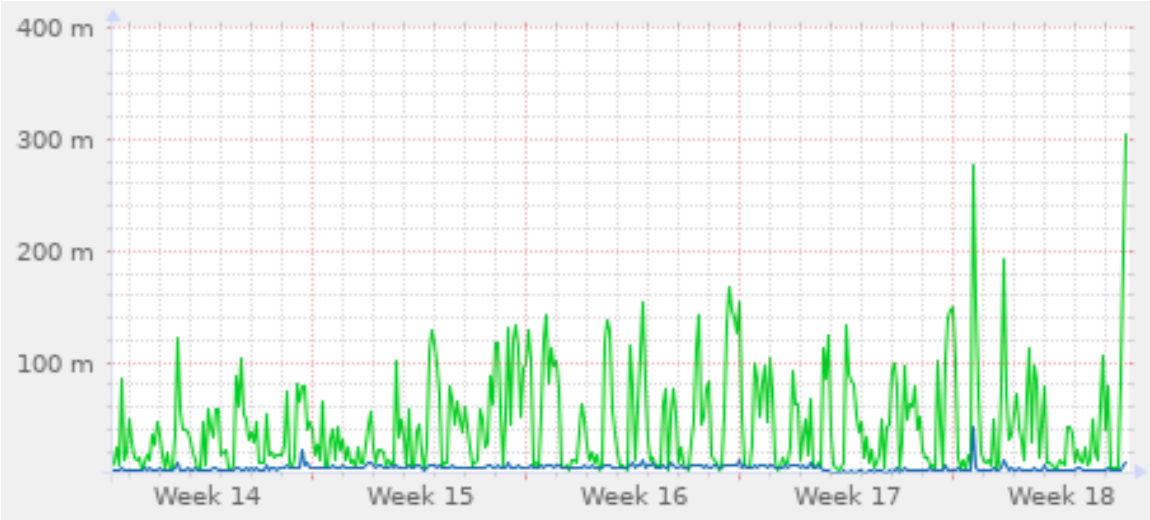
Datan keruun jälkeen data täytyy säilöä jonnekin monitorointiratkaisun käytettäväksi. Datan tyypin mukaan se voidaan säilöä eri tavalla ja eri paikkoihin. Esimerkiksi metrinen data säilytetään yleensä aikasarjatietokannassa (Time Series Database, TSDB). Aikasarjatietokanta on tähän tarkoitukseen suunniteltu tietokanta, johon tallennetaan dataa nimenomaan aikaleimojen perusteella. Lokidataa voidaan säilöä myös muihin tietokantoihin, tai esimerkiksi yksinkertaiseen flat-tiedostoon. (Julian 2017, luku 2.1.) Lokitiedostoja voidaan käsitellä myös erilaisilla elastisilla hakukoneilla, kuten *Elasticsearchilla*, jotka pystyvät hakemaan lokidataa ja järjestelemään sitä helposti luettavaan ja käsiteltävään muotoon, jota voidaan hyödyntää myös monitorointiratkaisuissa (Elasticsearch B.V. 2019).

Dataa haetaan ja säilötään yleensä hyvin lyhyinä ajanjaksoina, kuten 60 sekunnin välein, tai kriittisissä järjestelmissä jopa lyhyemmillä aikaväleillä. Palvelusta ja palvelimesta riippuen aikavälit ovat kuitenkin usein pidempiäkin. (Julian 2017, luku 2.1.) Aikaväli, jolta dataa säilötään, vaikuttaa suoraan siihen, kuinka paljon tilaa monitorointiratkaisu vaatii. Dataa ei siis kannata säilöä hyvin pitkiltä ajanjaksoilta, ainakaan täydessä resoluutiossa, sillä tietomäärä voi olla valtava arkkitehtuurista riippumatta. (McCabe 2011, 98.) Datan säilyttäminen pidemmiltä ajanjaksoilta ei myöskään ole aina niin tärkeää, etenkään jos jo kerättyä dataa ei hyödynnetä tai tulla hyödyntämään. Mikäli tavoitteena on analysoida dataa pidemmältä ajanjaksolta, on tilanne tällöin toinen, mutta siinäkin tapauksessa on syytä harkita, onko oleellista säilöä dataa minuutin tarkkuudella koko vuoden ajalta, jos data voidaan tiivistää kattamaan esimerkiksi vain päivät tai viikot. Esimerkiksi palvelimen prosessorin tarkalla käyttöasteella ei todennäköisesti ole merkitystä edes muutaman päivän jälkeen, sillä monitorointiratkaisun tarkoitus on lähtökohtaisesti nopeuttaa ongelmanratkaisua ja varoittaa ongelmista. (Julian 2017, luku 2.1.)

### 4.3.3 Datan visualisointi

Kerättyä dataa on hyvä pystyä kuvaamaan jollakin tavalla, jotta sitä voidaan analysoida myöhemmin. Datan visualisointi on hyödyllistä erityisesti silloin, kun halutaan saada nopea kuva jonkin palvelimen toiminnasta esimerkiksi graafina. Dataa voidaan visualisoida monella muullakin tavalla, mutta tärkeää visualisoinnissa on kuitenkin se, että datan täytyy olla ymmärrettävässä muodossa, jotta siitä on hyötyä ihmiselle. (Julian 2017, luku 2.1.) Monitorointia voidaan suorittaa myös ilman datan visualisointia, mutta visualisointi on kuitenkin suositeltavaa datan luettavuuden kannalta. Osa tässä dokumentaatiossa esitellyistä ratkaisuista ei sisällä ohjelmistoon sisäänrakennettua keinoa visualisoida kerättyä dataa, mutta tarkoitusta varten on kehitetty erilaisia kolmannen osapuolen ratkaisuja, kuten myöhemmin esitelty *Grafana*.

Yleisin keino visualisoida monitoroinnista kerättyä dataa on esittää se kuvan 3 mukaisessa viivakaaviossa. Viivakaaviosta saa kuvan tapahtumien määrästä ajallisesti joko yksinään tai yhdessä vertailun helpottamiseksi. (Julian 2017, luku 2.1.) Yksi nopeasti tulkittava keino on esittää eri palveluiden ja monitoroitavien kohteiden tietoja liikennevaloina. Liikennevalomallissa liikennevalojen väri muuttuu palvelimen tilan mukaan. Esimerkiksi ongelmatilanteessa palvelun tilaa kuvaava liikennevalo voi muuttua vihreästä punaiseksi.




Kuva 3. Visualisoinnissa käytetty esimerkkigraafi käyttäjien tekemistä DNS-kyselyistä

### 4.3.4 Analytiikka ja raportointi

Analytiikka ja raportointi ovat monitoroinnin kannalta hyödyllinen osa-alue, sillä niitä voidaan hyödyntää esimerkiksi palvelutasosopimuksen selvittämiseksi.

Palvelutasosopimuksessa (*service level agreement*) voidaan sopia esimerkiksi jostain tietystä luvatusta saatavuusasteesta. Saatavuusaste on yksi helposti kerättävä ja raportoitava tieto monitoroinnista. Ilman monitorointia on vaikea tarkkaan määritellä, miten kauan jokin palvelu on ollut ylhäällä joko prosentuaalisesti tai ajallisesti. Esimerkiksi 99% saatavuusaste tarkoittaa vuodessa jopa miltei neljän päivän alhaalla oloaikaa, eli päivittäin hieman yli 14 minuuttia. (Blomberg, Milne, Palislamovic & Sonderegger 2009, 18.)

Saatavuusasteiden pohjalta on helppo luoda ymmärrettävä kuva myös siitä, mitkä jonkin tietyn palvelun tietyt osat ovat olleet niin sanotusti heikoimpia lenkkejä palvelun toiminnallisuuden kannalta. Tietoa voidaan käyttää hyväksi muun muassa siihen, että resursseja voidaan jatkossa ohjata oikeiden palveluiden kehittämiseen.

### 4.3.5 Hälytykset

Hälytyksillä tarkoitetaan jonkin palvelun vikatilan tai muun ongelman ilmoittamista jollekin tietylle taholle. Hälytysten tehtävänä on nopeuttaa ja auttaa täsmentämään ongelmatilanteiden ratkomista. Tämä mahdollistaa myös sen, että palveluiden vikatiloista saadaan ilmoituksia myös silloin, kun palvelusta vastaava taho ei ole aktiivisesti valvomassa palvelua työpaikallaan. (Julian 2017, luku 2.1.)

Hälytykset voidaan pääsääntöisesti jakaa kahteen ryhmään: kriittisiin hälytyksiin ja tiedottaviin hälytyksiin. Kriittisiä hälytyksiä pitäisi tapahtua erittäin harvoin, sillä nämä hälytykset ovat usein sen kaltaisia, että ne vaativat välittömiä toimenpiteitä ja ratkaisun. Kriittinen hälytys voisi laueta esimerkiksi tilanteessa, jossa yrityksen kaikki verkkopalvelimet ovat alhaalla eikä yrityksen pääsivu enää vastaa. Tiedottavat hälytykset ovat luonteeltaan sellaisia, jotka eivät vaadi välittömiä toimenpiteitä, mutta voivat ajan myötä eskaloitua kriittisiksi ongelmiksi. Potentiaalinen tulevaisuuden kriittinen uhka saadaan siis tiedoksi jo ennen kuin se toteutuu. Tiedottavat hälytykset voivat olla esimerkiksi sellaisia, joissa järjestelmä ilmoittaa jonkin yön yli tehtävän varmuuskopioinnin epäonnistuneen. Molemmat ovat tärkeitä hälytyksiä, mutta tiedottavat hälytykset eivät välttämättä kuitenkaan ansaitse yhtä suurta prioriteettia. Yksi potentiaalinen toimenpide niiden kohdalla voisi olla esimerkiksi automaattinen tukipyynnön luonti toiminnanohjausjärjestelmään kyseisestä palvelusta tai toiminnallisuudesta vastaavalle taholle. (Julian 2017, luku 2.1.)

Toisen ajatuksen mukaan hälytykset voidaan jakaa neljään ryhmään: kriittisiin, kiireellisiin, huomiota vaativiin ja ehkäistäviin. Kriittiset hälytykset ovat yllättäviä tapahtumia, jotka estävät käyttäjien pääsyn palveluun tai merkittävästi haittaavat sen toimintaa. Hälytyksillä on negatiivinen vaikutus liiketoimintaan. Kiireellisillä hälytyksillä tarkoitetaan tilanteita, joissa on havaittu osittaisia saatavuusongelmia, jolloin osa käyttäjistä ei voi käyttää palvelua tai merkittävä osa palvelusta on pois käytöstä. Kiireelliset hälytykset vaativat nopeita toimia vaikutuksen minimoimiseksi. Huomiota vaativat hälytykset ovat sellaisia, joihin järjestelmän asiantuntijan tulee puuttua, jotta ne eivät kasva kriittisiksi ongelmiksi. Tällaiseksi hälytykseksi voitaisiin ajatella esimerkiksi kasvanut muistinkäyttö. Neljäntenä hälytyksenä on ehkäistävät hälytykset, joilla ei ole kriittistä vaikutusta järjestelmään, mutta jotka myös voivat lopulta kasvaa sellaisiksi, ellei ongelmaan puututa tarpeeksi ajoissa. (Ligus 2012, luku 3.) Tämän hälytysmallin eri hälytysten väliset erot ovat pienempiä, kuin aiemmassa kahden hälytystyypin mallissa. Monitoroinnin kehityksessä tulee siis määritellä tarkemmat rajat eri tyyppisille hälytyksille. Perusajatus hälytysten priorisoinnista on kuitenkin sama.

Hälytykset ovat vain yksi lopputulos monitoroinnille, mutta ei kuitenkaan tärkein. Monitoroinnissa hälyttämisen perimmäinen tarkoitus on ilmoittaa epäonnistumisista, joko ohjelman toiminnassa, komponenteissa, joilla se toimii tai integraatiossa toisten ohjelmien tai komponenttien kanssa. (Julian 2017, luku 2.1.) Monitoroinnin päätarkoituksena esittää kysymyksiä ja laajentaa asiantuntijoiden käsitystä palveluiden integraatiosta ja vaikutuksesta toisiinsa, kuten mistä jokin pullonkaula palvelussa johtuu (Josephsen, 2016).

## 4.4 Laatikkotestaukset monitoroinnissa

Ohjelmistojen testauksissa käytetään usein termejä *black-box testing*, eli mustan laatikon testaus, ja *white-box testing*, eli valkoisen laatikon testaus. Näitä termejä ja toimintamalleja voidaan hyödyntää myös monitoroinnissa, sillä monitorointi on eräänlaista jatkuvaa testausta, jolla testataan, että ohjelmisto tai palvelu toimii niin kuin sen pitää.

### 4.4.1 Black-box monitorointi

*Black-box monitoring* (mustan laatikon monitorointi) tarkoittaa monitorointia, joka keskittyy vain palvelimien tai applikaatioiden tilan tarkastamiseen. Mustan laatikon monitorointiratkaisuun lasketaan yleisesti kaikki monitorointi, joka tarkastelee esimerkiksi palvelimien eri tiloja ja käyttöasteita, kuten palvelimen prosessorin tai levyn käyttöastetta tai kytkimen pingausta. Mustan laatikon monitorointi ei auta käyttäjää ymmärtämään miksi palvelin tai applikaatio on joutunut kyseiseen vikatilaan. Mikäli palvelimen levyn käyttöaste on noussut esimerkiksi 90% ja monitorointiratkaisu lähettää tästä hälytyksen, saa hälytystä tarkasteleva henkilö tiedon ongelmasta, mutta ei tietoa syystä kyseisen ongelmatilanteen synnylle. Mikäli hälytyksen syytä ei tunneta, ei välttämättä pystytä arvioimaan kuinka kiireellisestä ongelmasta on kyse. Viasta johtuva levyn äkillinen täyttyminen saattaa estää kriittisten palveluiden käytön, mikäli levytila loppuu. Mustan laatikon monitorointi on siis hyvä tapa kerätä yleispätevää tilannetietoa palvelimista ja applikaatioista, mutta se ei välttämättä tarjoa paljoakaan tietoa ongelmatilanteiden ratkomiseen. (Curtis, Jones & Hettich 2016, 59; Turnbull 2016, 27.)

Mustan laatikon monitorointi voidaan ottaa käyttöön käytännössä kaikilla laitteilla niiden omien osien puitteissa, ilman tarvetta erityiselle lisäkonfiguraatiolle tai tiedoille järjestelmästä. Nimitys ”musta laatikko” viittaakin siihen, ettei järjestelmää tai sen komponentteja tarvitse tuntea etukäteen, vaan ne ovat niin sanotusti pimennossa.

### 4.4.2 White-box monitorointi

*White-box monitoring* (valkoisen laatikon tai lasilaatikon monitorointi) tarkoittaa monitorointia, jossa monitoroitava alusta tunnetaan hyvin ja sitä pystytään tarkastelemaan yksittäisten osien osalta, eikä niinkään kokonaisuutena. Tämä mahdollistaa monipuolisemman vianselvityksen esimerkiksi lokitiedostojen avulla. Lasilaatikon monitoroinnissa voidaan monitoroida esimerkiksi verkkopalvelun tuottamien HTTP-kyselyiden määrää, applikaation tuottamaa omaa statistiikkaa sekä minkälaisia tai tyyppisiä kyselyitä tietokantaan tehdään tietokantapalvelimella. Lasilaatikon monitoroinnissa voidaan myös tarkastella erilaisia ta-

pahtumaketjuja, esimerkiksi käyttäjien kirjautuessa verkkopalveluun, jotta saadaan selville, tapahtuuko kirjautuminen odotetulla tavalla vai onko tapahtumaketjussa jotain poikkeavaa, kuten yrittääkö käyttäjä kirjautua palveluun ilman salasanaa tai yrittääkö hän suorittaa tietokantakyselyitä lomakekenttien avulla kirjautuessaan. (Curtis ym. 2016, 59; Turnbull 2016, 27.) Lasilaatikon monitoroinnissa voidaan hyödyntää myös esimerkiksi ohjelmistojen rajapintoja monitoroinnin toteuttamisessa, jossa tehdään kyselyitä suoraan rajapintaan.

Lasilaatikon monitorointi vaatii laajaa suunnittelua ja tietoa monitoroitavasta järjestelmästä ja sen toiminnasta, sillä nimensä mukaisesti kaikki osat tulisi olla läpinäkyviä ja järjestelmän osat määriteltävissä.

## 4.5 Agentiton monitorointi

Agentittomalla monitoroinnilla tarkoitetaan sellaista monitorointiratkaisua, jossa monitoroitavalle palvelimelle ei asenneta erillistä monitorointiratkaisun kanssa yhdessä toimivaa ohjelmistoa, eli niin sanottua agenttia, joka tarkkailee palvelimen tilaa esimerkiksi prosessorin, muistin ja kovalevyjen käyttöasteen kannalta. Agentiton monitorointi tarkoittaa myös joissain tilanteissa heikompaa monitorointia, sillä kaikkea ei pystytä monitoroimaan ilman agenttia. Agentittomalla monitoroinnilla voidaan kuitenkin kerätä joitain yksinkertaisia, mutta tärkeitä tietoja, kuten ovatko palvelimet, palvelut tai laitteet ylhäällä ja vastaavatko ne yksinkertaiseen ping-kyselyyn. (Adato 2018, 19.)

Nykypäivänä palveluita voidaan monitoroida jo melko hyvin ilmankin agenttia, jos palvelu on rakennettu jonkin API:n (application programming interface), eli ohjelmistorajapinnan päälle. Tällöin kyselyitä voidaan tehdä suoraan API:a hyödyntäen, jolloin ei tarvita ylimääräistä agenttia. Tämä ratkaisu vaatii kuitenkin omanlaisensa työkalun tai skriptin, joka on räätälöity juuri tätä tarkoitusta varten. (IDERA, Inc. 2016, 4.)

Useimmat nykyaikaiset monitorointiratkaisut hyödyntävät agentittomassa monitoroinnissa esimerkiksi SNMP- tai WMI-protokollia, joiden avulla kyselyillä pystytään keräämään dataa esimerkiksi palvelimista tai verkkolaitteista ilman, että niille asennetaan omaa työkalua datan keruuta varten. (IDERA, Inc. 2016, 4.)

## 4.6 Simple Network Management Protocol

*Simple Network Management Protocol*, eli SNMP, on standardoitu tietoliikenneprotokolla, jota verkkolaitteet käyttävät muun muassa verkon elementtien monitorointiin. SNMP:n yk-

sinkertaisuuden vuoksi miltei kaikki verkkolaitteet tukevat sitä. Protokollaa käytetään verkossa olevien laitteiden tilojen ja tietojen keruuseen. (Zoho Corp. 2019.) SNMP:aa voidaan hyödyntää miltei kaikkien verkkolaitteiden kanssa, myös palvelimien ja työpöytäkäyttöliittymien, kuten Windowsin, Linuxin tai Mac OS:n kanssa.

SNMP ei kuitenkaan ole täydellinen ratkaisu verkkolaitteiden monitorointiin, sillä se vaatii ylimääräisiä lisäosia monitorointiratkaisun toteuttamiseksi (Julian 2017, luku 8.3). Lisäosia kutsutaan niin sanotuiksi SNMP managereiksi, tai pollereiksi, joiden tehtävä on toimia välikappaleena monitorointiratkaisun ja monitoroitavan laitteen välillä protokollaa käytettäessä. SNMP managerin tehtävä on siis keskustella verkkolaitteen kanssa käyttäen tätä protokollaa hyödyksi ja välittää kerätty data joko suoraan monitorointiratkaisun hyödyntämälle palvelimelle tai sen käyttämälle agentille.

Protokollaa ei voi kutsua tietoturvalliseksi, sillä se on alun perin kehitetty salaamattomana, jolloin myöhemmätkin versiot ovat lähtökohtaisesti kehitetty tälle salaamattomalle alustalle. Tämän takia SNMP:aa ei tulisi käyttää palvelimien kanssa, sillä se saattaa paljastaa liikaa tietoa palvelimen toiminnasta ulkopuolisille. (Julian 2017, luku 8.3.) SNMP versio 3 kuitenkin auttaa tietoturvaongelmien ratkaisussa lisäämällä protokollaan autentikoinnin, auktorisoinnin, kulunvalvonnan sekä yksityisyyden. Nykypäivänä ei siis tulisi käyttää mitään muuta kuin protokollan uusinta versiota, sillä aiemmissa versioissa tietoa siirretään salaamattomana, joka johtaa tietoturva-aukkojen mahdolliseen löytämiseen, jos jonkin laitteen toimintahierarkia paljastetaan. (SNMP Research International, Inc., 2019.)

SNMP:aa voidaan hyödyntää toimeksiantajan verkkoinfrastruktuurin valvonnassa esimerkiksi verkkolaitteiden, kuten tulostinten, sensoreiden, kytkimien ja esimerkiksi Citrixin pilvipalvelimien monitoroinnissa, tai ESXi hypervisorin monitoroinnissa.

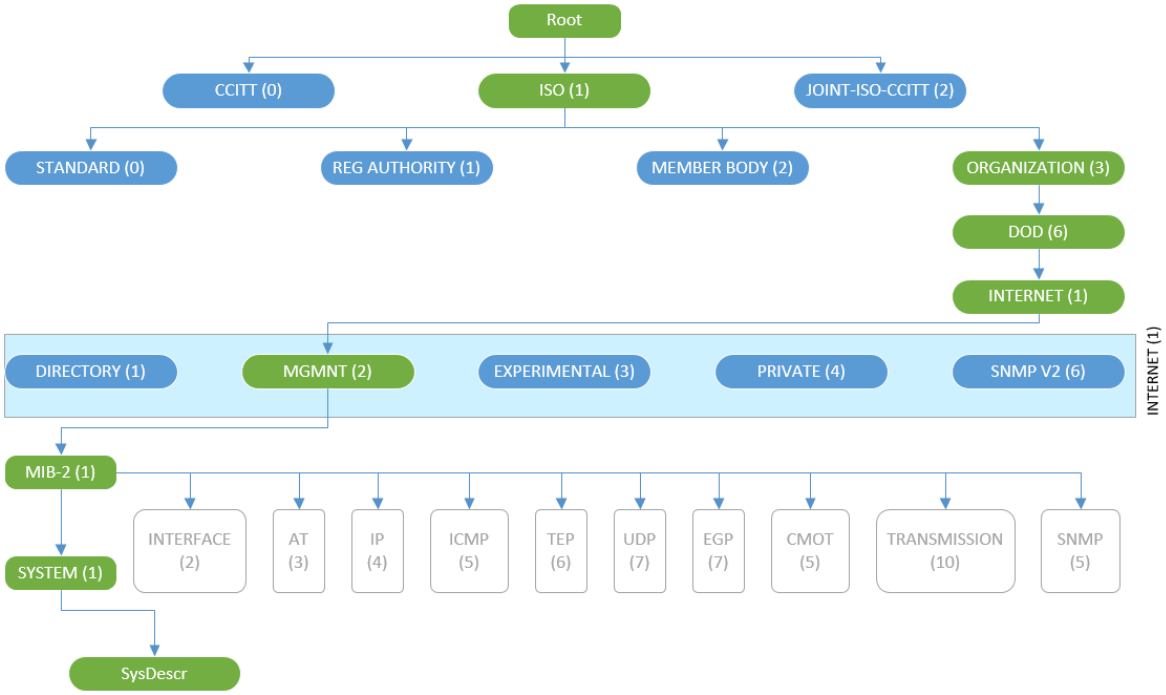
### 4.6.1 Management Information Base ja Object Identifier

*Management Information Base*, eli MIB, on tärkeä osa verkonvalvontamallia. MIB yhdistetään useimmiten SNMP kanssa, sillä SNMP managerit käyttävät laitteen kanssa keskustellessaan MIB:a hyödykseen, kun ne tekevät kyselyitä laitteesta ja sen tiedoista. (Cisco Systems, Inc. 2007.)

MIB on eräänlainen abstrakti tietokanta (Cisco Systems Inc. 2007), joka on kokoelma laitteen eri osista numeraalisesti identifioitavia avaimia, toisin sanoen *Object Identifiereitä* (OID). Nämä objektien indentifioivat avaimet muodostavat laitteen hierarkian. Tämä hie-

rarkia sisältää laitteen kaikki hallittavat ominaisuudet. MIB:a käsitellään usein puurakenteessa, jossa se organisoi OID:it hierarkkisesti sen yksilöllisten muuttujien tunnisteiden mukaisesti. Jokaisella puurakenteen haaralla on oma numero ja nimi, ja jokainen piste on nimetty kokonaisen polun, puun yläosasta alaspäin, puolelle, joka johtaa kyseiseen pisteeseen. (Zoho Corp. 2019.)

MIB:a hyödynnetään esimerkiksi silloin, kun halutaan kysyä verkon ylitse laitteen kuvausta sen järjestelmän komponenteista ja käyttöliittymästä. Tämä kysely voidaan toteuttaa kutsumalla tietoa OID polulla 1.3.6.1.2.1.1.1. (Orange SA 2015). Kyseinen OID-polku on tarkennettu ja selkeytetty kuvassa 4 vihreällä, selittäen mistä kyseiset numeraaliset arvot periytyvät. Kuvan 4 puurakenteessa polku alkaa kohdasta Root, minkä jälkeen kysely ohjataan OID:ien avulla oikeaan osoitteeseen.




Kuva 4. MIB puurakenne (mukaillen Zoho Corp. 2019)

### 4.6.2 SNMP versiot

SNMP:sta on kehitetty kolme versiota: SNMPv1, SNMPv2c ja SNMPv3. Näistä ainoa suositeltava versio on kolmas versio, sillä se on ainoa versio, joka tukee kommunikoinnin salaamisen ja autentikoinnin. Ensimmäisessä versiossa kaikki tieto kulkee salaamattomana verkon ylitse, eli esimerkiksi jonkinlaisen pakettien kuunteluohjelman avulla salasanat voidaan lukea suoraan lennosta viestien seasta. Ensimmäistä versiota voidaan käyttää miltei kaikkien verkkolaitteiden kanssa, mutta sitä ei kuitenkaan suositella käytettäväksi sen tietoturvan puuttuvuuden vuoksi. Toisessa versiossa salasanat pystytään salaamaan MD5:llä, mutta salaus pitää erikseen konfiguroida. Kolmannessa versiossa pystytään lisäksi autentikoimaan salasanat, ettei kuka tahansa, jolla on *community string*, eli niin sanottu viiteavain, pysty keskustelemaan laitteen kanssa SNMP:n avulla. (Mason & Newcomb 2001, 60-63.)

SNMPv3:ssa on taulukkoon 2 kootusti kolme eri tapaa autentikoida protokollan käyttöoikeus: noAuthNoPriv, authNoPriv ja authPriv. Näistä kolmesta autentikaatiomuodosta suositellaan käytettäväksi mahdollisuuksien mukaan vain authPriv-tasoa. (Cisco Systems, Inc. 2015, 2.)

Taulukko 2. SNMPv3 autentikaatio- ja salaustasot (Cisco Systems, Inc. 2015, 2.)

| Taso | Autentikaatio | Salaus |
|---|---|---|
| noAuthNoPriv | Username | Ei |
| authNoPriv | MD5 tai SHA | Ei |
| authPriv | MD5 tai SHA | DES |

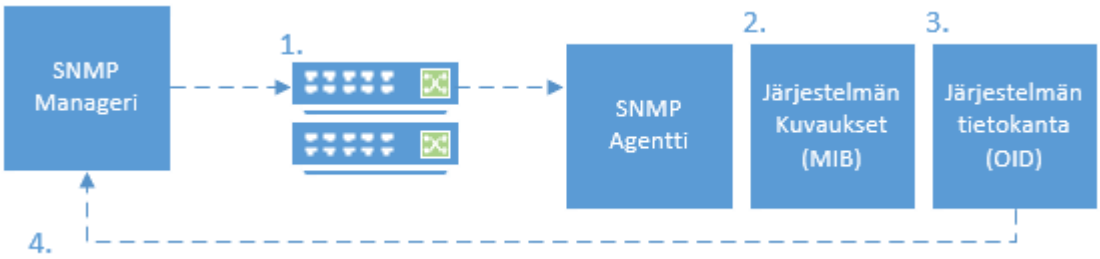
### 4.6.3 SNMP komennot

SNMP käyttää taulukkoon 3 koottuja erilaisia komentoja. Komentojen avulla SNMP kerää tietoa ja kommunikoi verkkolaitteiden kanssa. Näitä komentoja käytetään eri tarkoituksiin, kuten esimerkiksi SNMP Trap tarkoittaa eräänlaista ansaa, joka niin sanotusti laukeaa, kun jokin tapahtuma tapahtuu. Komento lähetetään SNMP managerille SNMP agentin avulla. (Zoho Corp. 2019.)

Taulukko 3. SNMP komennot ja niiden selitykset (Zoho Corp. 2019)

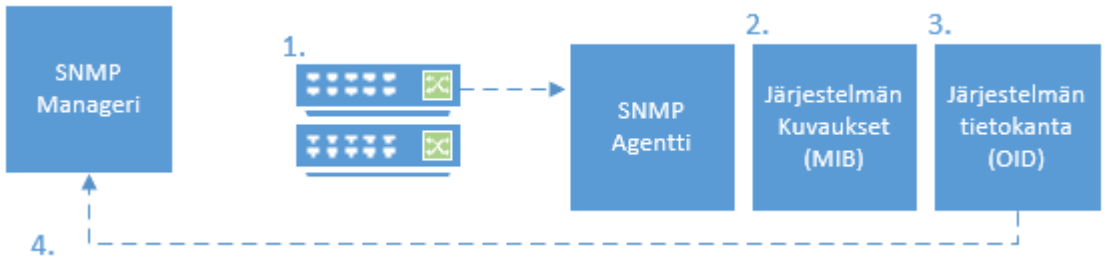
| Komento | Toiminto |
|---|---|
| GET | Komento, jonka SNMP manageri lähettää laitteelle. GET-komento palauttaa yhden tai useamman arvon laitteesta managerille. |
| GET NEXT | Samankaltainen kuin GET-komento, mutta GET NEXT palauttaa seuraavan OID:n arvon MIB hierarkiassa. |
| GET BULK | Käytetään haettaessa tietoa massana isosta MIB-taulukosta. |
| SET | Komento, jonka avulla SNMP manageri voi muokata tai määrittää jonkin arvon laitteelle. |
| TRAP | Komento, jonka agentti aloittaa ja lähettää SNMP managerille, kun jokin tapahtuma tapahtuu. Esimerkiksi kun laitteen lämpötila nousee kriittiseksi. |
| INFORM | Samankaltainen kuin TRAP-komento, mutta INFORM-komento vaatii konfirmaation SNMP managerilta, kun se on saanut SNMP-paketin perille. |
| RESPONSE | Lähettää kyselyitä tai muutoksia, joita SNMP manager toteuttaa |

GET, GET NEXT, GET BULK ja SET -komentoja suorittaessa SNMP managerin täytyy itse lähettää kyseiset komennot verkkolaitteelle. Automaattisessa monitoroinnissa komennoille on säädettävä jokin aikaväli, jotta laitteesta saadaan kerättyä tarpeeksi tarkkaa tilannedataa. Komentoja käsiteltäessä SNMP manageri lähettää ensin SNMP-komennon verkkolaitteelle, kuten esitetty kuvan 5 kohdassa 1. Laite käsittelee komennon itse ja siirtää sen omalle SNMP agentilleen, joka tarkistaa komennon sisällön omasta MIB:sta, kuvan 5 kohdissa 1 ja 2, ja yrittää lähettää itsestään keräämänsä tiedon takaisin SNMP managerille, kuten havainnollistettu kuvan 5 viimeisessä kohdassa 4. Tässä tapauksessa SNMP manageri keskustelee molempiin suuntiin verkkolaitteen kanssa. (Zoho Corp. 2019.)




Kuva 5. SNMP-komentojen tapahtumaketju (mukaillen Zoho Corp. 2019)

SNMP Trapin tapauksessa monitorointiratkaisun SNMP managerin ei tarvitse itse muodostaa yhteyttä valvottavan laitteen kanssa, vaan laite itse lähettää itsestään tietoa kyseiseen niin sanottuun ansaan, jota SNMP manageri kerää, kuten esitetty kuvassa 6. Muutoin SNMP:n tapahtumaketju on sama kuin aiemmin mainitussa kuvauksessa. Tämä edesauttaa verkkolaitteiden monitorointia paljon, sillä tietoja ei tarvitse valituin aika-ajoin kysellä laitteelta, vaan laite ilmoittaa tapahtumista itsenäisesti heti kun ne tapahtuvat, nopeuttaen monitoroinnin toimintaa huomattavasti. Tässä tapauksessa ainoastaan itse verkkolaite lähettää tietoa itsestään SNMP managerille, joten molemman suuntaista keskustelua ei tarvita. (Zoho Corp. 2019.)




Kuva 6. SNMP Trapin tapahtumaketju (mukaillen Zoho Corp. 2019)

### 4.7 Windows Management Instrumentation

*Windows Management Instrumentation*, eli WMI, on Microsoftin kehittämä infrastruktuuri ylläpitodatan ja operaatioiden hallitsemiseen Windows-pohjaisissa käyttöjärjestelmissä (Microsoft 2018). WMI:ia käytetään automatisoimaan hallinnollisia tehtäviä etäällä oleviin Windows-käyttöjärjestelmiin erilaisilla skripteillä tai ohjelmilla. Sen avulla voidaan myös tuottaa kyselyitä käyttöjärjestelmistä tai tuotteista sekä niiden osista. (Kennedy & Satran 2018.)
WMI mahdollistaa SNMP:n tavoin agentittoman keinon monitoroida Windows-käyttöjärjestelmiä, käyttäen WMI:in infrastruktuuria.

WMI:ia voidaan hyödyntää toimeksiantajan verkossa eri Windows-palvelimien osalta, joihin ei haluta asentaa erillistä agenttia valvontaa varten, jos tarkoituksena on valvoa vain

käyttöjärjestelmän tietoja, eikä yksittäisiä prosesseja. WMI:ia voidaan myös hyödyntää tulevaisuudessa työasemien valvonnassa, esimerkiksi vierailijoiden käytössä olevien tietokoneiden osalta.

### 4.8 Monitorointiratkaisu palveluna

Monitorointiratkaisuja tarjoavat nykypäivänä useat eri yritykset, jotka tuottavat monitorointiratkaisuja palveluna, jota kutsutaan termillä *SaaS*, eli *Software as a Service*. Monitorointiratkaisun hankinta palveluna on yrityksille usein huomattavasti halvempaa kuin monitorointijärjestelmän kehittäminen yrityksen sisällä, sillä yrityksen ei tarvitse itse hankkia aikaa, osaamista ja työntekijöitä monitorointiratkaisun kehitykseen ja ylläpidolle. Valmiiksi kehitettyjä ja jatkuvasti kehityksessä olevia ratkaisuja löytyy myös avoimen lähdekoodin ohjelmistoina, mutta niiden käyttöönotossa tulee ottaa huomioon henkilötyöpäivien hinta yritykselle. (Julian 2017, luku 2.3.)

Monitorointiratkaisujen kehittäminen vie usein useita henkilötyöpäivä, jotka ovat pois muusta työstä. Monitoroinnin hankinta palveluna vapauttaa työresursseja järjestelmien ylläpitäjiltä, sillä he voivat keskittyä järjestelmien kehittämiseen, sen sijaan, että he keskittyisivät niiden monitorointijärjestelmän kehittämiseen. Monitorointiratkaisun kehittäminen sisäisesti ei tuota yritykselle sen kehityksen aikana mitään tuloja, sillä kehitysvaiheessa siitä ei ole suoraa hyötyä sen käyttäjille. (Julian 2017, luku 2.3.)

Nykypäivänä monitorointia tarjotaan myös palveluna, jossa palvelua tarjoava yritys hoitaa itse monitorointiratkaisun käyttöönoton ja avustaa monitoroinnin kehityksessä, jolloin monitorointi räätälöidään asiakkaan palveluihin sopivaksi ja sitä kautta asiakkaalle hyödylliseksi. Tätä palvelumallia voidaan hyödyntää esimerkiksi verkkopalveluiden monitoroinnissa, jolloin asiakasyrityksen ei itse tarvitse käyttää resurssejaan esimerkiksi verkkosivun alla toimivan ohjelmakokonaisuuden valvontaan ja hälyttämiseen.

# 5 Monitorointi prosessien tukena

Julian (2017, luku 1) esittelee kirjassaan monitoroinnin kannalta huomioon otettavan termin, *antisuunnittelumallin*, jonka avulla hän luokittelee yleisiä monitoroinnin suunnittelumalleja, jotka kuitenkin konkretisoituessaan kostautuvat yritykselle. Monitorointia tulisi siis kehittää prosessina, eikä vain käsittää yhtenä tapana lähestyä ongelmanselvitystä.

Yksi näistä antisuunnittelumalleista on työkalukeskeinen työskentelytapa. Työkalukeskeinen työskentelytapa tarkoittaa sitä, että yritys huomaa nykyisessä toimintamallissaan ongelman, jolloin se huomaa myös tarpeen jollekin uudelle työkalulle, joka mahdollisesti auttaa heitä ratkaisemaan kyseisen ongelman, sen sijaan että olemassa olevan ongelman alkuperäistä prosessia tai työkalua parannettaisiin tai kehitettäisiin. Tässä ajattelutavassa on kuitenkin se ongelma, että kun uusia työkaluja hankitaan ratkaisemaan jotain tiettyä ongelmaa, voidaan myöhemmin huomata, ettei kyseinen uusi työkalu pystykään enää ratkaisemaan seuraavaa ongelmaa, jolloin hankitaan jälleen uusi korvaava työkalu tilalle. Ei ole kuitenkaan mahdollista löytää yhtä työkalua, joka ratkaisisi kaikki ongelmat ja pystyisi kaikkeen, eikä sellaista ole myöskään järkevää kehittää olemassa olevista työkaluista. (Julian 2017, luku 1.1.)

Julian (2017, luku 1.2) painottaa kirjassaan myös sitä, etteivät toisten yritysten itse kehittämät ja käyttämät työkalut välttämättä ratkaise samaa ongelmaa jossain toisessa yrityksessä. Samaan potentiaaliin pääseminen jossain toisessa yrityksessä voi olla haastavaa, sillä näiden työkalujen kehittämisen taustalla on yleensä ollut jokin tietty prosessi, jolla on olleet omat tietyt tarpeensa.

Hyödyllisen monitoroinnin toteuttaminen on monimutkaista, sillä monitoroinnin tarkoitus on auttaa järjestelmien ylläpitäjiä ratkaisemaan useita erilaisia ongelmia sekä varoittamaan mahdollisista tulevista ongelmista; sikäli kun niiden toteutumista on pystytty ennakoimaan monitorointiratkaisun kehittämisessä, eli monitoroinnin prosessissa. Hyvän monitoroinnin tulisikin siis olla prosessi, eikä vain työkalu, jota käytetään vain ongelmien valvomiseen ja hälyttämiseen.

Monitorointia tulisi kehittää prosessina kaikissa siitä hyötyvissä palveluissa, eikä vain palvelimien terveyden tarkastamisessa, esimerkiksi prosessorin tai muistin käyttöasteen osalta. On myös huomioitava, että monitoroitava palvelin tai palvelu saattaa toimia siitä huolimatta, että palvelimen prosessorin käyttöaste on korkea. Tällöin monitorointi ei suoranaisesti kerro onko palvelussa jotain vialla, sillä esimerkiksi tietokantapalvelimen pro-

sessorin käyttöaste voi olla jatkuvasti korkea, mutta silti hyväksyttävä jos tietokannan kyselyiden vasteajat ovat sallituissa rajoissa. Esimerkiksi verkkopalvelujen monitorointiin on järkevämpää tehdä täsmentäviä kyselyjä, joita tarkastelemalla voidaan selvittää, toimiiko palvelu halutulla tavalla. Monitorointia pitäisi siis kehittää ajan myötä sitä mukaa kun palvelua opitaan ymmärtämään paremmin, jolloin voidaan tehdä parempia ja täsmällisempiä tarkistuksia. (Julian 2017, luku 1.3.)

Monitoroinnin kehitys tulisi jakaa kaikkien järjestelmien asiantuntijoiden välille, eikä nimittää vain yhtä monitoroinnin vastuuhenkilöä. Yksittäinen vastuuhenkilö ei todennäköisesti pysty olemaan kaikkien palveluiden ja niiden tarvitsemien osien asiantuntija, eikä täten pysty itsenäisesti määrittelemään kaikkia hyödyllisiä monitoroitavia parametreja, jotka järjestelmän kehittäjä pystyisi määrittämään. (Julian 2017, luku 1.2.)

Monitorointia pitäisi myös harjoittaa tarpeeksi usein, eikä kerätä dataa vain tiettyinä harvoina väliaikoina. Liian harvasta monitorointivälistä voi koitua riski, etteivät jotkin kriittiset tapahtumat ikinä paljastu monitoroinnista. Tällainen tilanne voisi olla esimerkiksi se, että palvelun vasteaika kohoaa sallitun rajan ylitse 30 sekunnin välein, mutta monitorointiin kerätään dataa vain 5 minuutin välein. Yksittäisiä piikkejä ei välttämättä huomata lainkaan, ellei monitorointi sattumalta osu piikin kanssa samalle hetkelle. Monitoroinnin aikaväleistä käydään erimielistä keskustelua siitä, miten raskasta datan kerääminen lyhyillä aikaväleillä on ja kuinka paljon se kuormittaa palvelimia. Nykyaikaiset palvelimet ja verkot ovat kuitenkin niin tehokkaita, että niiden pitäisi soveltua käsittelemään monitoroinnin luomaa taakkaa entistä paremmin. Erilaisten verkkolaitteiden kohdalla monitoroinnin ajoitus vaatii kuitenkin huomiota, jotta verkkolaitteita ei kuormiteta kyselyillä liian useasti, sillä niiden hallinta ei ole yhtä nopeaa kuin palvelinten hallinta. Dataa ei myöskään välttämättä kannata kerryttää tai säästää kovin pitkiltä aikaväleiltä, ellei sille koeta erityistä tarvetta. (Julian 2017, luku 1.3.)

Monitorointia ei Julianin (2017, luku 1.4) mukaan myöskään kannata käyttää niin sanottuna kainalosauvana, jonka avulla pyritään ratkaisemaan ongelmia. Monitorointi on erittäin hyvä ja tärkeä työkalu, jonka avulla ongelmat voidaan huomata sekä niiden tapahtumahetkellä, että mahdollisesti jo ennakoivasti. Monitorointi ei siis ole tarkoitettu korjaamaan rikkinäistä järjestelmää tai palvelua, vaan rikkinäinen järjestelmä täytyy korjata, jotta monitoroinnista on hyötyä. Monitoroinnin tarkoitus ei myöskään ole jatkuvasti lähettää varoituksia ylläpidolle, vaan varoitusten rajat ovat hienosäädettävä jokaiselle palvelulle erikseen, jotta ei lähetetä niin sanotusti vääriä hälytyksiä.

Julian (2017, luku 1.5) puhuu kirjassaan myös siitä, kuinka monitoroitavien kohteiden konfiguroinnin tulisi olla automatisoitua, ilman että kaikkia monitoroitavia kohteita täytyy lisätä manuaalisesti erikseen. Julianin mukaan monitoroinnin konfiguraation ei kuulu olla erityisen aikaa vievä prosessi. Mikäli konfiguraatiossa kestää kauan, monitoroinnin kehittäminen muuttuu vuorostaan työlääksi, kun taas jos konfigurointi on nopeaa, sitä voidaan kehittää paljon mielekkäämmin kattamaan myös laajemmat tarpeet.

Monitoroinnin automaatio ei kuitenkaan ole suuressa asemassa Haaga-Helian kaltaisessa ympäristössä. Haaga-Helia ei tarvitse yhtä ketterää monitorointiratkaisua kuin jokin laajempi ympäristö saattaisi tarvita, sillä suurin osa Haaga-Helian palvelimista ja palveluista ovat staattisia ja niitä on verrattain pieni määrä. Lisäksi Haaga-Helialla ei ole virtualisoituja kontteja kehitysympäristöissä, joita pitäisi rakentaa ja pystyttää jatkuvasti uudelleen. Automatisoitu konfigurointi on kuitenkin yksi asia, joka olisi suositeltavaa ottaa käyttöön vähintään jonkinlaisen skriptin muodossa, joka tekee palvelimille tarvittavat konfiguraatiot niiden liittämiseksi osaksi monitorointia. Monet monitorointiratkaisut kuitenkin mahdollistavat agenttien asennuksen puoliautomatisoinnin yhdellä skriptillä, jonka käyttäjä ajaa palvelimellaan ja konfiguroi tietyt arvot vastaamaan monitorointiratkaisun push tai pull -mallia, jotta agentti osaa keskustella monitorointiratkaisun kanssa.

Julian mainitsee kirjassaan (2017, luku 2.4) tärkeänä asiana myös monitoroinnin jatkuvan kehityksen, ja painottaa erityisesti sitä, että monitorointia ei pitäisi suorittaa työkaluna, joka voidaan jättää huomioimatta, vaan prosessina, jota tulisi kehittää jatkuvasti. Vain pitkään kestäneessä prosessissa monitorointi alkaa tuottamaan haluttua tulosta. Hän nostaa esiin isojen yritysten monitoroinnin onnistuneisuuden, kuten Googlen, Facebookin, Twitterin ja Netflixin, joiden kaikkien monitorointi on erittäin onnistunutta ja kehittynyttä, mutta kyseisten yritysten monitorointia on kehitetty vuosia ja siihen on käytetty myös erittäin paljon resursseja. Monitorointia tulisikin lähestyä prosessina, eikä vain työkaluna.

# 6 Palvelujen monitorointi

Palvelujen monitorointi on Haaga-Helialle vähintäänkin yhtä tärkeää, ellei tärkeämpää kuin palvelimien ja verkkoinfrastruktuurin yleinen valvonta, sillä Haaga-Helia modernina koulutuspalvelua tarjoavana organisaationa hyödyntää monia eri palveluita, joita se itse ylläpitää – kuten esimerkiksi oppimisalusta Moodlea.

Palvelujen monitorointi tulisi pyrkiä aloittamaan käyttäjäkokemuksesta, eikä verkkoapplikaatiosta tai palvelimista. Toisin sanoen palveluiden monitorointi pitäisi aloittaa siitä kohdasta, mistä käyttäjien interaktio alkaa applikaatiossa, sillä käyttäjiä ei kiinnosta miten palvelu toimii, tai kuinka monta verkkopalvelinta pyörittää palvelua, vaan heitä kiinnostaa yksinkertaisesti toimiiko palvelu. (Julian 2017, luku 2.2.) Tätä varten monitoroinnissa voidaan hyödyntää yksinkertaisten verkkokyselyiden, kuten HTTP-kyselyiden, vastausten lisäksi myös ohjelmistorobotiikkaa, jonka avulla voidaan simuloida oikeiden käyttäjien toimintoja verkkosivulla. Moodlen tapauksessa tämä voitaisiin toteuttaa esimerkiksi niin, että käyttäjä hakee verkkosivua, jonka jälkeen hän yrittää kirjautua sivulle ja suorittaa sivulla joitain toimenpiteitä, joita oikea käyttäjä suorittaisi palvelun normaalissa käyttötapauksessa.

Palveluiden monitoroinnissa yksinkertaisin keino saada tietoa palvelusta on monitoroida palvelun vastauksia HTTP-kyselyihin. Mikäli palvelu vastaa kyselyyn esimerkiksi HTT-koodilla 2xx voidaan tehdä päätelmä, että HTTP-kysely toimi onnistuneesti. Jos kyselyyn kuitenkin saadaan vastaukseksi 5xx, saadaan tietää, että palvelimen puolella on jotain vialla. Sen jälkeen, kun käyttäjälähtöinen monitorointi on ratkaistu, tulisi siirtyä monitoroimaan esimerkiksi verkkopalvelimen eri komponentteja. Palveluiden monitoroinnissa tulisi kuitenkin aina pyrkiä parantamaan käyttäjäkokemusta. (Julian 2017, luku 2.2.)

# 7 Monitorointiratkaisuja

Tässä luvussa käymme läpi vertailuun valittuja monitorointityökaluja. Työkalut valittiin ennalta määritettyjen kriteereiden pohjalta sellaisiksi, että kukin vastaisi Haaga-Helian toiveita ainakin osittain. Valikoimme työkaluja myös sillä perusteella, että ne olivat alan ammattilaisten keskuudessa suosittujen keskustelupalstojen ja tiedonvälityskanavien perusteella suosituimpia ja eniten käytettyjä.

Luvussa käymme läpi ohjelmistojen toimintaperiaatetta yksinkertaistetusti. Monet ominaisuudet ovat eri ohjelmissa toteutettu hyvin samankaltaisesti, jonka vuoksi emme kata kuvauksissa kaikkia ominaisuuksia tai niiden toiminnallisuutta, vaan pyrimme keskittymään toiminnan kannalta oleellisimpiin komponentteihin. Opinnäytetyöprojektiin käytettävän ajan rajallisuuden vuoksi kerätty tieto on pääosin valmistajien itsensä ilmoittamaa tai luotettavista julkisista internet-lähteistä kerättyä. Eri yritysten avoimesti tarjoama tieto ohjelmistoista todettiin hyvin vaihtelevaksi, jonka vuoksi koostamamme lyhyet esittelytekstit eivät ole keskenään vertailukelpoisia. Vertailemme Haaga-Helian näkökulmasta oleellisimpia ominaisuuksia erillisessä taulukkovertailussa seuraavassa luvussa.

## 7.1 Nagios XI

Nagios XI on avoimen lähdekoodin ohjelmisto Nagios Coresta johdettu lisensoitu työkalu, jonka on kehittänyt Nagios Coren alkuperäinen kehittäjä Ethan Galstad sekä taustalla oleva yritys Nagios Enterprises. Nagios XI tuo ilmaisversioon uusia työkaluja kuten raportointiominaisuudet, käyttöliittymäpäivitykset sekä konfiguraatiomanagerin. Nagios on toteutettu ohjelmointikieli C:llä. (Nagios Enterprises 2019a.)

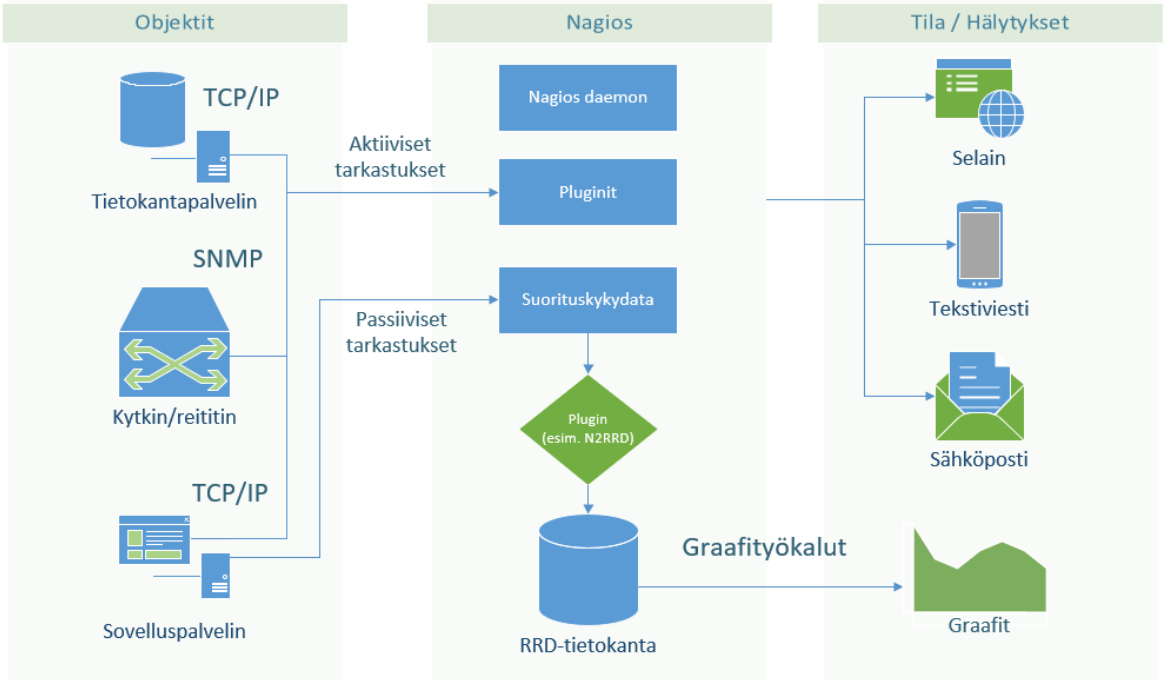
Nagios XI valittiin vertailuun ilmaisen Nagios Core -version sijasta, sillä suurin osa halutuista ominaisuuksista tai toiminnoista puuttuivat perusversiosta kokonaan. Lisäksi Haaga-Helia oli jo aiemmin hankkinut lisenssin XI -versioon.

Nagios vaatii toimiakseen plugineita, jotta palvelimilta kerätty data saadaan kerättyä ja datan perusteella voidaan luoda erilaisia hälytyksiä (Barth 2005, 17). Pluginit, yhdessä luodun konfiguraation ohella, määrittelevät sen, mitä dataa kerätään ja miten siihen reagoidaan. Ennalta määriteltyjen kriteereiden pohjalta tehtävää monitorointia, jossa Nagioksen hyödyntämä plugin kyselee palvelimilta tietoja tietyin väliajoin, kutsutaan aktiiviseksi monitoroinniksi (Nagios Enterprises 2019b). Plugin välittää keräämänsä tiedon takaisin Nagios daemonille. Passiiviseksi monitoroinniksi taas kutsutaan tilannetta, jossa erilliset applikaa-

tiot keräävät itse dataa ja säilövät sen myöhempää daemonin kyselyä varten (Nagios Enterprises 2019c).

Datan keräyksen jälkeen erillinen plugin lähettää tiedon tietokantaan. Data säilötään tietokantaan myöhempää käyttöä varten, mahdollistaen monitorointidatan käytön esimerkiksi graafisessa web-käyttöliittymässä tai hälytyksinä, jotka lähetetään käyttäjien sähköpostiin tai puhelimeen tekstiviestinä. (Barth 2009, 378-379.)

Nagioksen toimintaperiaatetta on tämän luvun kuvauksen ja lähdeaineiston pohjalta esitetty kuvassa 7. Kuvaan valitut palvelimet ja tietokanta eivät heijasta Haaga-Helian ympäristöä, vaan ne on valittu demonstraatiotarkoituksessa.




Kuva 7. Nagioksen toimintaperiaate yksinkertaistetusti kuvattuna

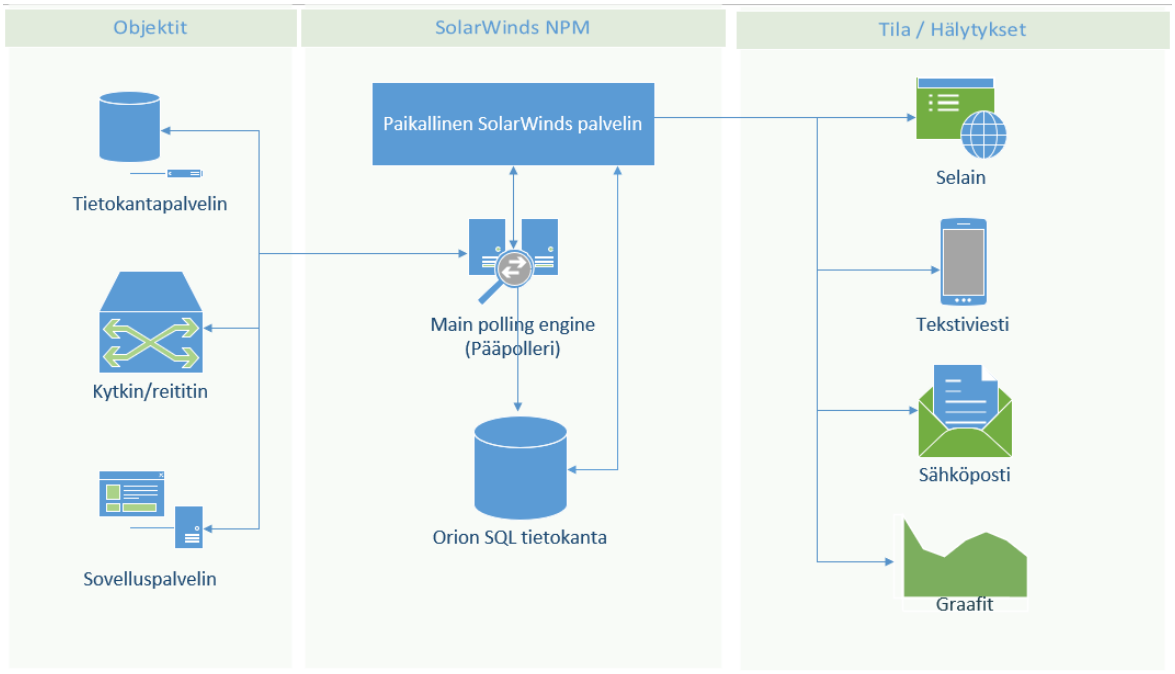
## 7.2 SolarWinds NPM

SolarWinds Inc. on kehittänyt useita verkkojen ja palvelinten hallintatyökaluja, mukaan lukien suljetun lähdekoodin monitorointityökalun *Network Performance Monitorin*. Monitorointityökalu on osa laajempaa työkalujen kokoelmaa, joita SolarWinds tarjoaa asiakkailleen. (Dissmeyer & Dissmeyer 2013, 333.) Monitorointityökalu on toteutettu hyödyntäen .NET Frameworkia (Dissmeyer & Dissmeyer 2013, 22).

SolarWindsin monitorointityökalu on mahdollista asentaa joko omalle dedikoidulle palvelimelleen tai osaksi jo olemassa olevaa SolarWinds Orion kokoonpanoa. SolarWindsin palvelin keskustelee erillisen *main polling enginen* kanssa, eli käytännössä kyseessä on erillinen laite, jonka pääasiallisena tehtävänä on huolehtia työpyyntöjen ajastuksesta, datan prosessoinnista ja kyselyistä monitoroitaville laitteille.

SolarWindsin lisensointi hinnoitellaan monitoroitavien elementtien kuten laitteiden, porttien, käyttöliittymien tai levymäärän perusteella. Laajemmissa ympäristöissä on mahdollista hankkia myös useampia pollereita toimimaan pääpollerin rinnalla. (SolarWinds, Inc. 2019, 10-13.) Pollerit lähettävät keräämänsä tiedon erilliseen SQL-tietokantaan, joka taas keskustelee varsinaisen SolarWinds-palvelimen kanssa (SolarWinds, Inc. 2019, 83). Palvelin hallinnoi monitorointidatan esitystä kuvaajina web-käyttöliittymässä sekä lähettää ennalta määritellyissä virhetilanteissa hälytykset joko selaimeen, tekstiviestitse tai sähköpostitse ja mahdollisesti suorittaa virhetilanteille erikseen määritellyt toiminnot, esimerkiksi palvelun uudelleenkäynnistyksen. Lisäksi kerätystä datasta voidaan luoda erilaisia raportteja. (SolarWinds, Inc. 2018, 33-42.)

SolarWindsin toimintaperiaatetta on pyritty kuvaamaan yksinkertaistettuna kuvauksena kuvassa 8, pohjautuen tässä luvussa esitettyihin tietoihin ja lähteisiin.




Kuva 8. SolarWinds NPM:n toimintaperiaate yksinkertaistetusti kuvattuna

## 7.3 PRTG

PRTG Network Monitor, aiemmin *Paessler Router Traffic Grapherina* tunnettu, monitorointityökalu on Paessler AG:n kehittämä lisensoitu monitorointijärjestelmä, joka on suunnattu Windows-käyttäjille. PRTG:n lisensointi perustuu käytössä olevien sensorien määrään, ja alle sadan sensorin järjestelmille on tarjolla ilmaisversio. PRTG tarjoaa lisäksi pilvipohjaisia monitorointiratkaisuja.

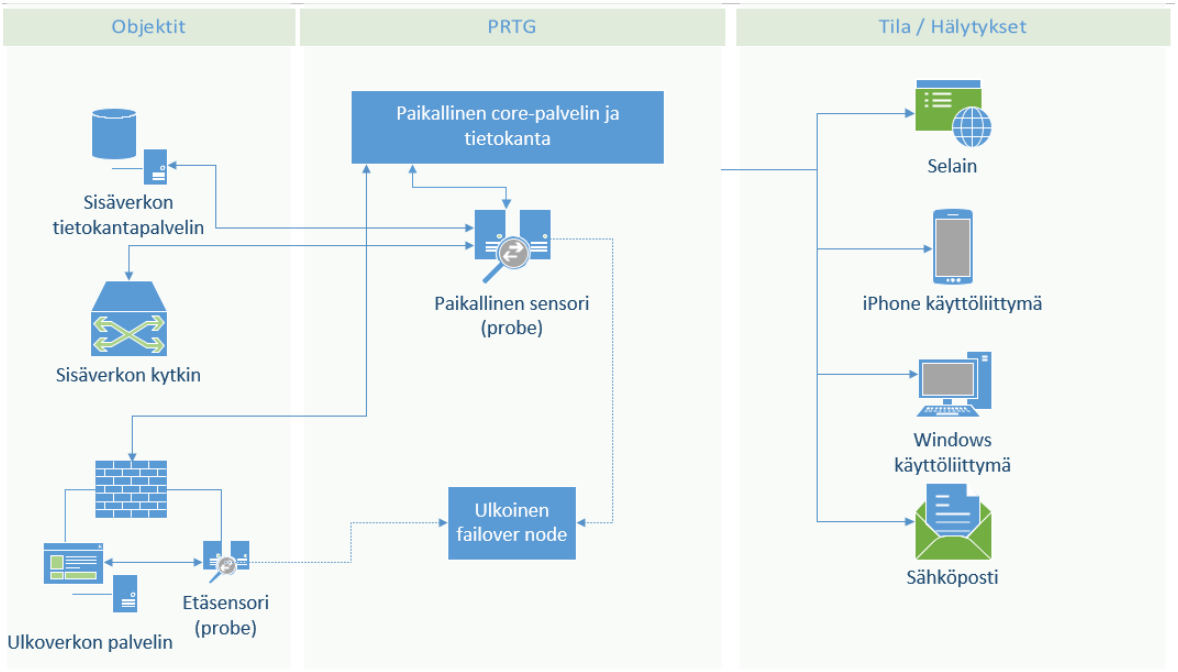
PRTG hyödyntää monitorointidatan tallennuksessa omaa tietokantaansa, jonka valintaa perustellaan datan määrän pienuudella. Tietojen tallentaminen omaan tietokantaan parantaa suorituskykyä, verkon kuormaa, konfiguraatiota ja ylläpitoa sekä helpottaa lisensointikuluissa, kun tietokanta tulee tuotteen mukana. (Paessler AG 2010.)

PRTG:n olennaisin osa on sen core-palvelin, joka huolehtii muun muassa tietojen talletuksesta paikalliseen tietokantaan, web-palvelimen ylläpidosta, raportoinnista ja hälytyksistä. Core-palvelimen kautta määritellään myös asetukset sensoreille, jotka huolehtivat monitorointidatan keruusta kohdekoneilta. (Paessler AG 2010.) Asetusten luontia varten PRTG tarjoaa erillisen konfigurointityökalun sekä sensoreille, että core-palvelimelle. Sensorit, eli PRTG:n termistön mukaan probet, voivat olla joko paikallisia tai ulkoverkossa toimivia etäsensoreita (remote probe). Lisäksi ympäristössä voidaan hyödyntää niin sanottuja cluster probeja, jotka huolehtivat useamman eri laitteen monitorointidatan keruusta itsenäisesti. (Paessler AG 2019a.)

Sensorit saavat konfiguraationsa aina core-palvelimelta ja raportoivat keräämänsä tiedon takaisin palvelimelle (Paessler AG 2019b). Mikäli kuitenkin halutaan varautua uhkaan, jossa probet eivät saa yhteyttä core-palvelimeen, esimerkiksi sähkökatkon tai komponenttivian vuoksi, voidaan monitorointiratkaisuun lisätä erillinen failover node, joka toimii varmuuskopiona paikallisen palvelimen rinnalla. Tällöin sensorit lähettävät tietonsa sekä paikalliselle palvelimelle, että failover nodelle, joka jakaa tietonsa core-palvelimen kanssa. Tällaisella ratkaisulla voidaan välttyä tietojen menetykseltä, kun palvelin on pidemmän aikaa pois käytöstä. (Paessler AG 2019c.)

Core-palvelin muodostaa keräämistään tiedoista monitorointigraafeja, joita käyttäjä voi tarkastella web-käyttöliittymän, erillisen Windows-ohjelman tai iPhone-käyttäjille suunnatun puhelinapplikaation kautta. Lisäksi hälytyksiä voidaan lähettää reaaliaikaisesti myös valittujen käyttäjien sähköpostiin. (Paessler AG 2019b.)

PRTG:n toimintaperiaatetta on yksinkertaistetusti kuvattu kuvassa 9. Kuva on muodostettu luvun sisällön ja siinä käytettyjen lähteiden pohjalta.

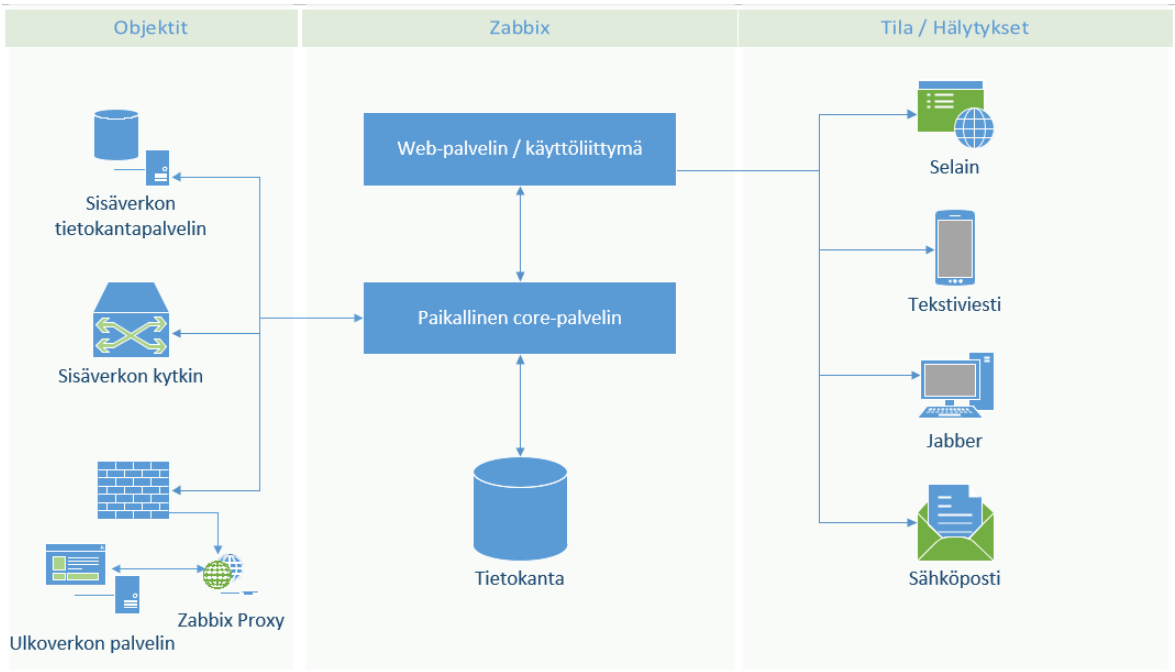



Kuva 9. PRTG:n toimintaperiaate yksinkertaistetusti kuvattuna

## 7.4 Zabbix

Zabbix on Zabbix LLC:n kehittämä avoimen lähdekoodin monitorointityökalu, joka on kehitetty ohjelmointikieli C:llä ja jonka selaimessa toimiva käyttöliittymä on toteutettu PHP:lla. Työkalu on suunnattu Windows, Linux ja Mac OS -käyttäjille. (Zabbix LLC 2017a.)

Zabbix on toimintaperiaatteeltaan hyvin pitkälti linjassa aiemmin esiteltyjen ratkaisujen kanssa, kuten esitetty kuvassa 10, ja myös sitä on mahdollista laajentaa ympäristöön sopivaksi ratkaisuksi. Zabbixia markkinoidaan muun muassa sillä, että se on keskitetty helppokäyttöinen kokonaisuus, jossa on jo valmiina natiivisti useimmissa UNIX-käyttöliittymissä toimivat agentit. Zabbix asennetaan Linux-palvelimelle ja ympäristön tai käytössä olevien komponenttien puitteissa myös tietokanta sekä web-palvelin voidaan asettaa toimimaan samalle palvelimelle. (Olups 2016, luku 1.) Hälytykset voidaan automatisoida toimimaan selaimessa, tekstiviestitse, sähköpostitse tai Jabber-pikaviestimessa (Zabbix LLC 2017b).




Kuva 10. Zabbixin toimintaperiaate yksinkertaistetusti kuvattuna (mukaillen Olups 2016)

## 7.5 Icinga 2

Icinga on avoimen lähdekoodin monitorointityökalu, joka on kehitetty itsenäisenä projektina Nagios Coren pohjalta. Icingan tavoitteena on ollut vastata yhteisön toiveisiin ja tarpeisiin, täydentäen Nagios Coren ominaisuuksia muun muassa lisäämällä työkaluun web-käyttöliittymä, kattavammat raportointiominaisuudet sekä laajempi tuki eri tietokannoille. Icingan taustalla ei ole yksittäistä yritystä, vaan alkuperäiset kehittäjät koostuivat Nagioksen yhteisön aktiivista jäsenistä, Nagioksen lisäosien kehittäjistä sekä Nagiosta edeltäneen Netwaysin jäsenistä. (Mobily 2012.)

Icingan ydin on kehitetty ohjelmointikieli C++:lla ja web-käyttöliittymä on toteutettu hyödyntäen PHP:tä. Nagioksen tapaan Icingan toiminnallisuuksia voidaan laajentaa yhteisön kehittämien lisäosien avulla. (Icinga 2019a.) Icinga lupaa, että Nagiokselle suunnitellut lisäosat toimivat myös Icingan kanssa (Icinga 2019b). Monitorointityökalu vaatii toimiakseen Linux-palvelimen, mutta varsinainen monitorointi voidaan toteuttaa myös Windows-ympäristössä (Icinga 2019c).

Icingan toimintaperiaate on Nagioksen kanssa hyvin samankaltainen, kuten esitetty luvussa 7.1. Icingan web-käyttöliittymä on kuitenkin toteutettu täysin omana ratkaisuna, jonka Icinga kuvailee olevan pikemminkin viitekehys (framework), johon voidaan lisätä erilaisia osia oman käyttötarpeen mukaan. Osia voisivat olla esimerkiksi UI, raportointi, lisäosat tai puhelinkäyttöliittymä. (Mobily 2012.) Web-käyttöliittymän yhteydessä voidaan

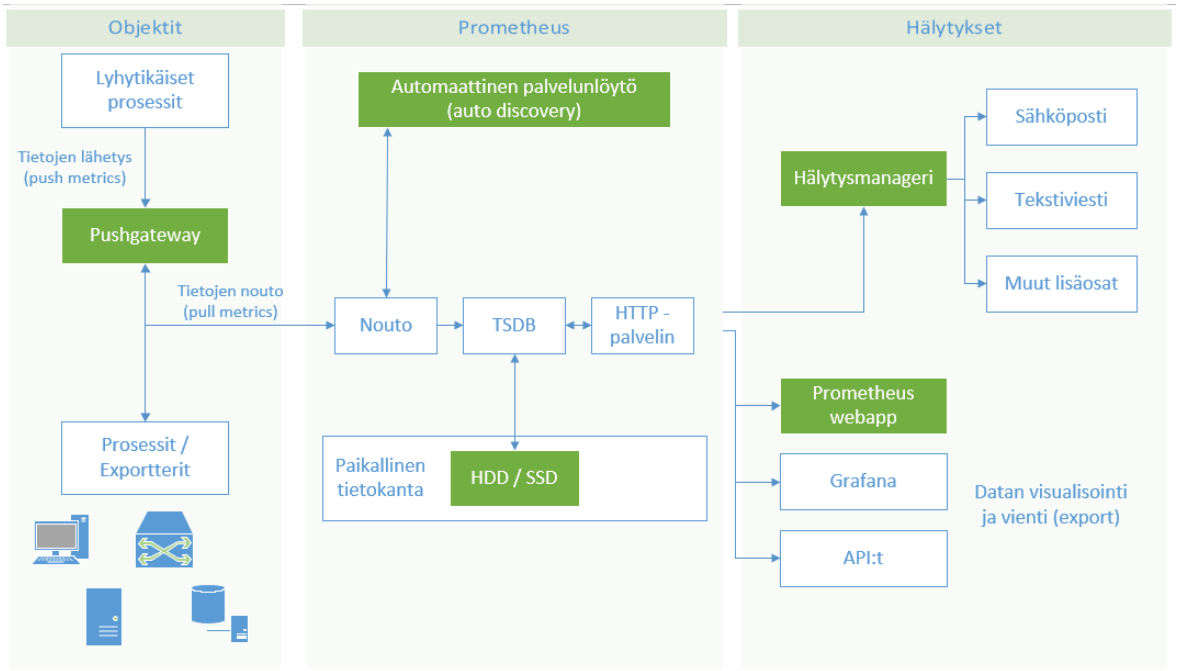
myös hyödyntää Icingan API:a, jonka avulla voidaan muun muassa muokata konfiguraatiotiedostoja ja luoda erilaisia komentoja (Icinga 2019d).

### 7.6 Prometheus

Prometheus on avoimen lähdekoodin monitorointijärjestelmä, jonka monet ominaisuudet ovat yhteisön kehittämiä. Prometheuksen kehitys alkoi alun perin *Soundcloud* audionjakelusivustolla, kun sivuston kehittäjät kokivat, etteivät jo olemassa olleet monitorointiratkaisut vastanneet heidän tarpeitaan. (Brazil 2018, luku 1.) Sittemmin Prometheuksesta on tullut Cloud Native Computing Foundationin projekti (Evans 2019). Työkalu on toteutettu ohjelmointikieli Go:lla. Prometheusta on mahdollista laajentaa hyvin monipuolisesti eri käyttötarpeisiin, mutta samalla laaja kustomointi voi vaatia käyttäjältä muita monitorointiratkaisuja laajempaa perehtymistä ohjelmiston käyttöön. Prometheus on suunniteltu asennettavaksi erilliselle palvelimelle, jolloin verkon tai muiden laitteistojen häiriöt eivät estä monitorointidatan tarkastelua. (Prometheus 2019a.)

Prometheus kerää erilaisia mittaustuloksia määritellyistä prosesseista, joko suoraan kyselemällä tai erillisen lyhytikäisille prosesseille suunnatun portin, *pushgatewayn,* kautta. Lyhytikäiset prosessit ovat sellaisia, jotka kestävät niin lyhyen aikaa, ettei tietoa prosessista ole välttämättä enää saatavilla datan keruuhetkellä. Nämä prosessit lähettävät omat tietonsa itse push gatewaylle, jolta Prometheus saa tiedon kysellessään tietoa prosesseista. (Prometheus 2019b.) Vastaavaa tiedon lähetys- ja keruumetodia on tässä dokumentissa kutsuttu myös termein passiivinen ja aktiivinen monitorointi.

Kerätty tieto tallennetaan lokaalisti aikasarjatietokantaan ja tietoa verrataan esimääriteltyihin sääntöihin, joiden pohjalta Prometheus päivittää tietokannassa jo olemassa olevaa dataa tai muodostaa muutosten pohjalta hälytyksen. Prometheuksessa itsessään ei ole datan visualisointia, vaan se voidaan toteuttaa erillisen laajennuksen avulla. Prometheuksen oma dokumentaatio suosittelee käytettäväksi *Grafana* nimistä visualisointiohjelmaa, mutta Prometheuksen kattavan laajennusmahdollisuuden takia visualisointi voidaan toteuttaa myös jollain muulla alustalla. (Prometheus 2019a.) Laajennettavuus on Prometheuksessa muutoinkin vahvasti esillä, sillä myös hälytykset voidaan toteuttaa hyvin monin eri tavoin erillisiä lisäosia hyödyntämällä. Prometheuksen kehittäjät tarjoavat käyttöön myös laajan määrän erilaisia export-työkaluja eri sovelluksille ja palvelimille, joissa Prometheuksen agentin asennus kohdekoneelle ei ole mahdollista tai järkevää (Prometheus 2019c). Prometheuksen toimintaperiaate on esitetty kuvassa 11.



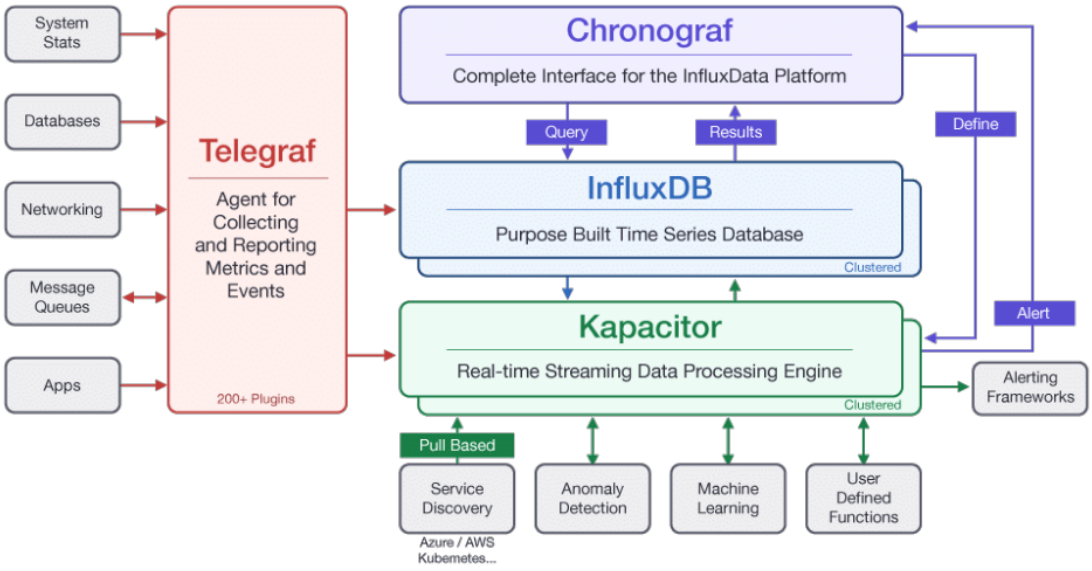

Kuva 11. Prometheuksen toimintaperiaate yksinkertaistetusti kuvattuna (mukaillen Prometheus 2019a)

## 7.7 TICK Stack

TICK Stack on yhteisnimitys InfluxData Incorporatedin tarjoamille avoimen lähdekoodin monitorointityökalujen kokoelmalle. TICK Stackiin kuuluvat oleellisena neljä eri komponenttia: Telegraf, Chronograf, InfluxDB ja Kapacitor. Yhdessä nämä neljä komponenttia muodostavat yhtenäisen monitorointiratkaisun, joka kerää dataa, vie sen tietokantaan ja siirtää datan monitorointityökalun käyttöliittymään, jossa tietoja voidaan tarkastella reaaliaikaisesti ja esimerkiksi luoda erilaisia hälytyksiä vikatilojen varalle. Prometheuksen tapaan myös TICK Stack on ohjelmoitu GO-ohjelmointikielellä. (InfluxData, Inc. 2019a.) Eri komponenttien keskusteluyhteyttä on esitetty kuvassa 12.

TICK Stackissa Telegraf on plugin-pohjainen agentti, joka kerää ja raportoi dataa. Data voidaan kerätä suoraan palvelimelta tai vaihtoehtoisesti Telegrafin voi liittää osaksi muita ohjemointirajapintoja. (InfluxData, Inc. 2019b.) InfluxDB on aikasarjatietokanta, joka on suunniteltu käsittelemään aikaleimattua dataa. Tietokannan tarkoituksena on kerätä dataa, jolla mitataan määreitä tai tapahtumia tietyllä ajanjaksolla. Tällaista dataa voisivat olla esimerkiksi suorituskyky, verkon kuormitus, sensoridata tai erilaiset tapahtumat, kuten klikkaukset. (InfluxData, Inc. 2019c.) Kapacitor puolestaan suorittaa datan käsittelyn ja mahdollistaa integraation muihin tietokantoihin, palveluihin ja sovelluksiin. Kapacitorin avulla voidaan luoda erilaisia kyselyitä, ajastettuja tapahtumia tai mitata tuloksia verraten niitä samankaltaisuuksiin. (InfluxData, Inc. 2019d.) Datan visualisoinnista vastaa Chronograf, joka saa tietonsa muilta komponenteilta ja koostaa niistä helposti tarkasteltavan

näkymän. Chronograf saa tietonsa ensisijaisesti InfluxDB:ltä, mutta Chronografissa voidaan määritellä ehdot erilaisille hälytyksille, joiden täyttymisehtoja Kapacitor käsittelee ja tiedottaa eteenpäin Chronografille. (InfluxData, Inc. 2019e.) Kapacitor, InfluxDB ja Chronograf keskustelevat pitkälti keskenään, ja hyödyntävät toisiltaan saamaansa tietoa. Telegraf taas ensisijaisesti kerää dataa ja jakaa sen InfluxDB:lle ja Kapacitorille.




Kuva 12. TICK Stackin toimintaperiaate yksinkertaistetusti kuvattuna (InfluxData, Inc. 2019f.)

# 8 Monitorointiratkaisujen vertailu

Valmistajien tarjotessa hyvin vaihtelevan tasoista dokumentaatiota palveluistaan, päätimme tehdä vertailutaulukon, johon valitsimme arviomme mukaan Haaga-Helian näkökulmasta oleellisimpia ominaisuuksia. Taulukon 4 tarkoituksena oli helpottaa varsinaiseen loppuratkaisuun käytettävän työkalun valintaa. Vertailutaulukon 4 kentät on selitetty taulukon alapuolella.

Taulukko 4. Monitorointiratkaisujen vertailu taulukkonäkymässä

| Ohjelmisto | Nagios XI | Solarwinds | PRTG | Zabbix | Icinga | Prometheus | TICK-Stack |
|---|---|---|---|---|---|---|---|
| WebApp | ● | ● | ● | ● | ● | ● | ● |
| Konfiguraatiovelho | ● | ● | ● | - | ● | - | - |
| Agentiton mahdollisuus | ● | ● | ● | ● | ● | ● | ● |
| Automaattinen palvelunlöytö | ● | ● | ● | ● | ●* | ● | ● |
| Datan visualisointi | ● | ● | ● | ● | ●** | ●** | ● |
| Trendien ennustus | ● | ● | ● | ● | ●** | ●** | ● |
| LDAP/AD integraatio | ● | ● | ● | ● | ● | - | - |
| Helppokäyttöisyys (1-5) | 4 | 2 | 3 | 3 | 3 | 1 | 2 |
| Laajennettavuus (1-5) | 3 | 3 | 2 | 3 | 3 | 5 | 5 |
| Hinta alkaen / FOSS | 3070 €/v | 2440€/v | 1488€ | ● | ● | ● | ● |

*Icingassa automaattinen palvelunlöytö voidaan toteuttaa pluginin, eli liitännäisen avulla.

**Icinga ja Prometheus eivät tuota trendigraafeja itse, mutta ohjelmasta voidaan hakea data ja esittää se graafisessa muodossa hyödyntäen kolmannen osapuolen työkalua, kuten Grafanaa.

WebAppilla tarkoitetaan sitä, onko monitorointityökalussa mukana selaimessa toimiva graafinen web-käyttöliittymä. Konfiguraatiovelho taas on käyttöliittymään sisäänrakennettu työkalu, jolla konfiguraatioita voidaan luoda ilman tarvetta muokata varsinaisia asetustiedostoja manuaalisesti. Agentiton mahdollisuus -kenttä määrittelee sen, voiko monitorointityökalua käyttää ilman palvelimelle asennettavia komponentteja, sillä joissain tapauksissa voi olla, ettei tarkasteltavalle palvelimelle ole mahdollista tai suositeltavaa asentaa mitään palvelimen ensisijaisen käyttötarkoituksen ulkopuolisia ohjelmia. Automaattisella palvelunlöydöllä vertailimme työkaluja, joiden avulla verkosta löytyviä palvelimia ja laitteita on mahdollista skannata ja lisätä osaksi monitorointia ilman, että kyseisten laitteiden IP- tai

MAC-osoitteet täytyy lisätä monitoroinnin alaisiksi manuaalisesti. Erityisesti Haaga-Helian laajuisessa ympäristössä ominaisuus vähentäisi ylläpidollista työtaakkaa huomattavasti.

Datan visualisoinnilla tarkoitetaan jotain visuaalista esitystapaa monitorointiratkaisun keräämälle datalle, visualisointi on nopea ja hyödyllinen keino saada tiivistetty katsaus jonkin monitoroitavan kohteen tilanteesta. Trendien tunnistuksella tarkoitetaan työkalua, joka osaa jo kerätyn datan perusteella arvioida tulevia trendejä, esimerkiksi kuormituksen tai levynkäytön osalta.

LDAP/AD -integraatiolla taas haetaan palvelua, jossa monitorointityökaluun voisi antaa käyttöoikeuksia suoraan AD-ryhmien (active directory) perusteella, jolloin paikallisia monitorointityökalun tunnuksia ei tarvittaisi.

Helppokäyttöisyyskentällä pyrimme arvioimaan sitä työmäärää, jota kunkin työkalun käyttöönotto ja ylläpito edellyttää. Helppokäyttöisyyden arviointi on haastavaa ja suhteellista siihen, millainen lähtötaso kullakin käyttäjällä on aiheesta jo ennestään. Helppokäyttöisyyttä on arvioitu asteikolla 1-5, jossa pienempi lukema on haastava ja suurempi lukema on helpompi. Lukema on tarkoitettu suuntaa antavaksi eikä se perustu luotettavaan tutkimustyöhön aiheesta.

Laajennettavuudella taas tarkoitetaan sitä, kuinka monipuoliset mahdollisuudet työkalu antaa esimerkiksi erilaisiin käyttötarpeisiin soveltuvien lisäosien asennukselle tai erilaisten käyttötarpeiden monitoroinnille, perinteisen suorituskyvyn mittauksen lisäksi. Laajennettavuutta on arvioitu asteikolla 1-5, jossa 1 on heikosti laajennettava ja 5 on erittäin laajennettava. Myös laajennettavuusarvio on suuntaa antava, eikä se perustu aiheesta tehtyyn luotettavaan tutkimustyöhön. Hinta-sarakkeeseen keräsimme tiedon siitä, onko palvelu maksullisen vai ilmainen avoimen lähdekoodin ohjelmisto (FOSS).

# 9 Valittu monitorointiratkaisu

Monitorointiratkaisuksi valittiin yhdessä toimeksiantajan kanssa Nagios XI. Valintaan vaikutti se, että toimeksiantajalla oli ohjelmistoon jo voimassa oleva lisenssi ja sille oli spesifioitu budjetista oma lohkonsa. Nagios XI täytti myös kaikki halutut kriteerit monitorointiratkaisulle. Lisäksi toimeksiantajalla oli aiemmin ollut käytössään Nagios XI:lle rakennettu testiympäristö, mutta sen ottaminen käyttöön oli ollut hyvin vähäistä, eikä se sellaisenaan soveltuisi lopullisen monitorointiratkaisun toteutukseen. Testiympäristö mahdollisti kuitenkin sen, että monitorointia päästiin kehittämään nopeammin, jolloin projektista saatiin myös jatkokehityksen kannalta kattavampi tulos.

## 9.1 Nagios XI:n laitevaatimukset

Nagios XI ei vaadi paljoa alustalta, jolle se asennetaan. Nagios XI voidaan toteuttaa järkevästi myös virtuaaliympäristössä, mutta Nagioksen tarjoaman dokumentaation mukaan yli 1000 laitteen tai 5000 palvelun monitoroinnissa on syytä resursoida erillinen fyysinen palvelin, jotta virtualisointialustan resurssien jako ei vaikuta Nagios XI:n toimintaan. (Nagios Enterprises 2018a.)

Lopullinen ratkaisu asennettiin Haaga-Heliassa sille varatulle fyysiselle palvelimelle, jotta se ei pyöri virtuaalisena ja jotta se voidaan halutessa eristää muusta verkkoinfrastruktuurista ja osoittaa omaan virtalähteeseensä korkean saatavuuden parantamiseksi. Korkean saatavuuden parantamiseksi suunnitteilla on hankkia lisäksi toinen palvelin toimeksiantajan infrastruktuurin ulkopuolelta, jolloin palvelimet keskustelisivat keskenään, eikä lokaalit ongelmatilanteet, kuten pitkäaikaiset sähkökatkokset, pääse vaikuttamaan monitoroinnin saatavuuteen.

Nagios XI:n laitevaatimukset muovautuvat sen käyttöasteen mukaisesti ja ne esitetään taulukossa 5. Mitä useampia palveluja monitoroidaan, sitä enemmän resursseja Nagios-palvelimelle olisi suositeltavaa allokoida, ettei resurssien riittämättömyys hidasta monitorointia tai luo yhteensopivuusongelmia suuren kuorman vuoksi, esimerkiksi muistin suhteen. Toimeksiantajan tarpeet eivät kuitenkaan ole niin suuret, että palvelimien tai palveluiden määrällä olisi suurta merkitystä laitevaatimuksissa.

Taulukko 5. Nagios XI laitevaatimukset (Nagios Enterprises 2018a)

| Monitored Nodes / Hosts | Monitored Services | Hard Drive Space | CPU Cores | RAM |
|---|---|---|---|---|
| 50 | 250 | 40 GB | 1-2 | 1-4 GB |
| 100 | 500 | 80 GB | 2-4 | 4-8 GB |
| > 500 | > 2500 | > 120 GB | > 4 | > 8GB |

## 9.2 Käytössä oleva laitteisto

Haaga-Helian puolesta monitorointiratkaisulle valittiin HP ProLiant DL360 Gen9 (843374-425) palvelin, joka kattaa Nagios XI:lle asetetut laitevaatimukset ja enemmänkin.

Palvelimen prosessoinnista vastaa kaksi Intel® Xeon® E5-2620 v4 prosessoria kellotaajuudella 2,10GHz. Palvelimessa on kaksi 16GiB DDR4 DIMM -muistikampaa, jotka toimivat 2400MHz taajuudella. Kovalevyinä palvelimessa on neljä 480Gb SSD-kovalevyä, jotka toimivat kahden levyn Raid 1 -pakassa peilattuina. Käytännössä peilaus tarkoittaa sitä, että data on monistettu useammalle eri levylle, jolloin yhden levyn hajoaminen ei johda tietojen menetykseen. Peilauksen jälkeen tallennustilaa jää käytettäväksi siis yhteensä 2x480Gb.

# 10 Nagios XI asennus ja konfigurointi

Nagios XI:n asennus on tehty yksinkertaiseksi Nagioksen tarjoamien skriptien avulla. Skriptit asentavat Nagios XI:n automaattisesti sekä kaikki sen vaatimat lisäosat. Asennuksessa skriptit myös konfiguroivat palvelimen muita osia sekä Nagioksen vaatimia lisäosia. Joitain asetuksia täytyy kuitenkin määrittää käsin. Konfiguraatioiden merkitys tulee myös ottaa huomioon, jotta monitorointiratkaisu toimii optimaalisesti sille määritellyn kapasiteetin ja verkon infrastruktuurin kannalta, näin menetellessä otetaan huomioon niin sanotut Nagios XI:n konfiguraation parhaat käytännöt. (Nagios Enterprises 2018b.)

## 10.1 Esivalmistelut

Esivalmisteluiltaan Nagios XI palvelin vaatii vain jonkin tyhjän käyttöjärjestelmän, jolle Nagios XI asennetaan. Tyhjä käyttöjärjestelmä on erittäin suositeltava sen takia, että Nagios XI:n laaja asennus tekee palvelimeen paljon eri asetuksia, jotka voivat vaikuttaa negatiivisesti muiden ohjelmien ja niiden käyttämien komponenttien toimintaan. (Nagios Enterprises 2018c.)

Kyseiselle Nagios XI:lle varatulle palvelimelle asennettiin Red Hat Linuxin toteutushetkellä uusin minimaalinen versio 7.6, jolle asennettiin myös kaikki päivitykset. Myöhemmin julkaistiin versio 8, mutta Nagios XI ei virallisesti tue sitä, joten emme päivittäneet uuteen versioon.

Nagios XI vaatii Red Hat Linuxilta *Optional Software Channel* -repositorion lisäyksen. Repositorio lisätään ensin asentamalla *yum-utils* -paketti, jonka jälkeen repositorio voidaan lisätä *yum-config-manager* -työkalulla. (Nagios Enterprises 2018c.)

```
yum install -y yum-utils
yum-config-manager --enable rhel-7-server-optional-rpms
```

Tämän jälkeen asennettiin paketit *wget* ja *firewalld*, jotka puuttuvat Red Hatin minimaalisesta asennuksesta. Wget:in avulla voidaan ladata tiedostoja internetin välityksellä, ja firewalld:llä voidaan tehdä yksinkertaistetusti palomuuriasetuksia, joita Nagios XI käyttää porttien avaamiseen ja sulkemiseen. (Nagios Enterprises 2018c.)

## 10.2 Nagios XI asennus

Nagios XI:n asennus on automatisoitu asennusskriptin avulla, joka ladataan Nagioksen palvelimelta. Asennus on yksinkertaista asennusskriptin avulla, sillä se tekee kaikki asennukset itsestään, jolloin käyttäjälle jää tehtäväksi vain asennuksen jälkeinen konfiguraatio.

```
cd /tmp
wget https://assets.nagios.com/downloads/nagiosxi/xi-latest.tar.gz
tar xzf xi-latest.tar.gz
cd nagiosxi
./fullinstall
```

Asennuksen jälkeen Nagios XI:n asennusskripti pyytää käyttäjää avaamaan Nagios XI:n verkkokäyttöliittymän osoitteessa https://<palvelimen_ip-osoite>/nagiosxi, jossa asennus viimeistellään. Asennuksen viimeistelyvaiheessa määritellään palvelimen verkkokäyttöliittymän osoite, aikavyöhyke sekä kieli, jonka jälkeen määritetään *nagiosadmin* -käyttäjätunnuksen salasana ja muut tiedot. Tämän toimenpiteen jälkeen Nagios XI on asennettu palvelimelle ja monitorointiratkaisun käyttöönotto voidaan aloittaa. (Nagios Enterprises 2018c.)

## 10.3 Nagios XI:n konfigurointi

Nagioksen konfigurointia helpotettiin kopioimalla konfiguraatiot vanhasta Nagios-virtuaalipalvelimesta, jolloin kaikki Nagiokseen tarvittavat lisenssit ja sertifikaatit voitiin helposti siirtää myös uudelle palvelimelle. Tämä nopeutti Nagioksen käyttöönottoa sen osalta, ettei esimerkiksi lisenssitietoja tarvinnut määritellä erikseen ja TLS-sertifikaatit saatiin kopioitua talteen muun muassa aktiivihakemiston integraatiota varten.

Nagioksen konfiguraatioiden migraatio tapahtui helpoiten ottamalla varmuuskopio aikaisemmalta palvelimelta, jolloin varmuuskopio voitiin suoraan ajaa uudelle palvelimelle varmuuskopion palauttamistyökalun avulla. Nagios kuitenkin edellyttää molempien palveluiden olevan samaa versiota. Aikaisemmalla palvelimella oli vanha versio Nagioksesta, jota ei ollut päivitetty uusimpaan versioon, joten se täytyi ensin päivittää. Päivityksen jälkeen Nagios pystyttiin varmuuskopioimaan siihen tarkoitetulla skriptillä. (Nagios Enterprises 2018d.)

```
sh /usr/local/nagiosxi/scripts/backup_xi.sh
```

Skripti varmuuskopioi Nagioksen ja muodosti siitä paketin hakemistoon */store/backups/nagiosxi/*, johon se loi tiedoston, tässä tapauksessa nimellä *1579858443.tar.gz*, joka siirrettiin uudelle palvelimelle, jossa voitiin ajaa varmuuskopion palauttamiseen tarkoitettu skripti.

*sh /usr/local/nagiosxi/scripts/restore_xi.sh /store/backups/nagiosxi/1579858443.tar.gz*

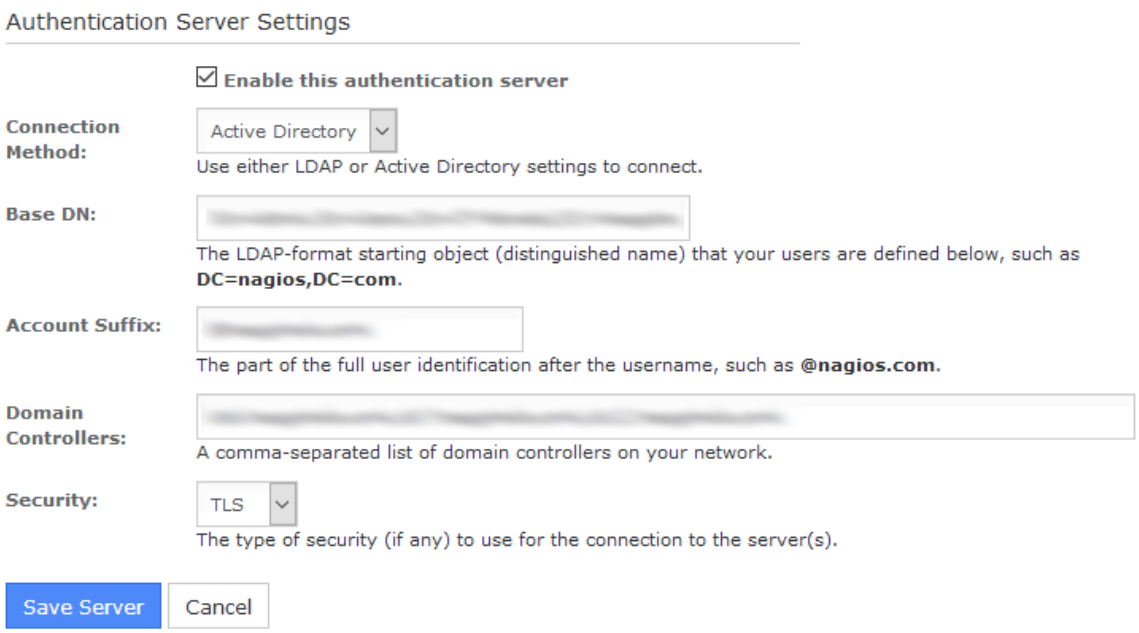
Varmuuskopion palauttamisen jälkeen palvelin käynnistettiin uudelleen mahdollisten ongelmien ratkaisua varten. Tämän jälkeen uuden Nagios-palvelimen konfiguraatiot käytiin läpi Nagioksen verkkokäyttöliittymää hyödyntäen ja ne testattiin toimiviksi. Uusi palvelin oli nyt lisensoitunut vanhan palvelimen lisenssiavaimella.

Uuden palvelimen konfiguraatiot käytiin lopuksi korjaamassa niiden tietojen osalta, jotka eivät vastanneet uuden palvelimen asetuksia, esimerkiksi verkkoasetuksien, kuten palvelimen nimen ja IP-osoitteen osalta.

### 10.4 Aktiivihakemiston integraatio

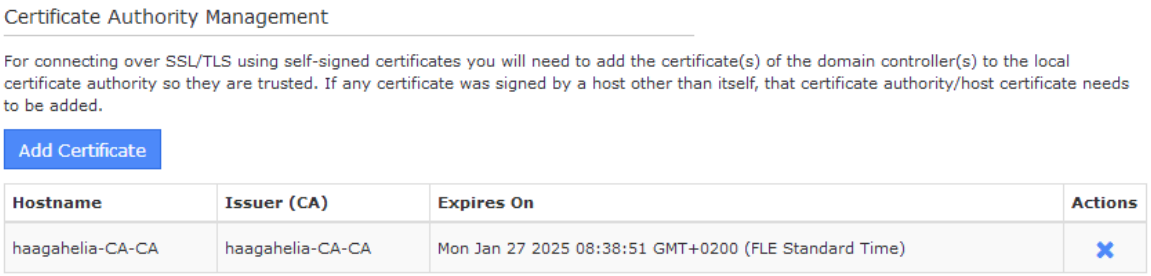
Haaga-Helian verkossa tietohallinnon käyttäjätunnukset on määritelty organisaation toimialueen aktiivihakemistossa. Samoja tunnuksia voidaan hyödyntää Nagios XI:n käytössä integroimalla nämä käyttäjätunnukset myös Nagiosiin ja jakamalla käyttöoikeuksia aktiivihakemiston organisaatioyksiköiden perusteella. Integraation avulla monitoroinnin käyttäjät ja kehittäjät voivat kirjautua Nagiokseen omilla organisaation admin-tunnuksillaan. Tämä parantaa käyttäjätunnusten hallittavuutta, sillä integraation avulla Nagiokseen ei tarvitse tehdä erillisiä paikallisia käyttäjätunnuksia. (Nagios Enterprises 2017.) Kuvissa 13-16 on esitetty yksittäisiä esimerkkejä siitä, miten aktiivihakemisto on integroitu toimeksiantajan verkossa.

Aktiivihakemiston integraatioon vaaditaan tieto siitä, missä osoitteessa organisaation *domain controllerit*¸ eli toimialueen ohjauspalvelimet sijaitsevat, tämä merkitään kohtaan *Domain Controllers*. Asetukset voidaan antaa suoraan Nagios XI:n web-käyttöliittymän kautta *Admin*-välilehden alta. Toimialueen ohjauspalvelinten lisäksi tulee määrittää organisaatioyksiköt, joissa halutut käyttäjätunnukset sijaitsevat, tämä merkitään kohtaan *Base DN*. Käyttäjätunnusten loppupääte organisaatiossa, eli *Account Suffix*, tulee myös määritellä. Esimerkkinä Nagioksen organisaatiossa tunnusten loppupääte olisi @nagios.com. (Nagios Enterprises 2017.)



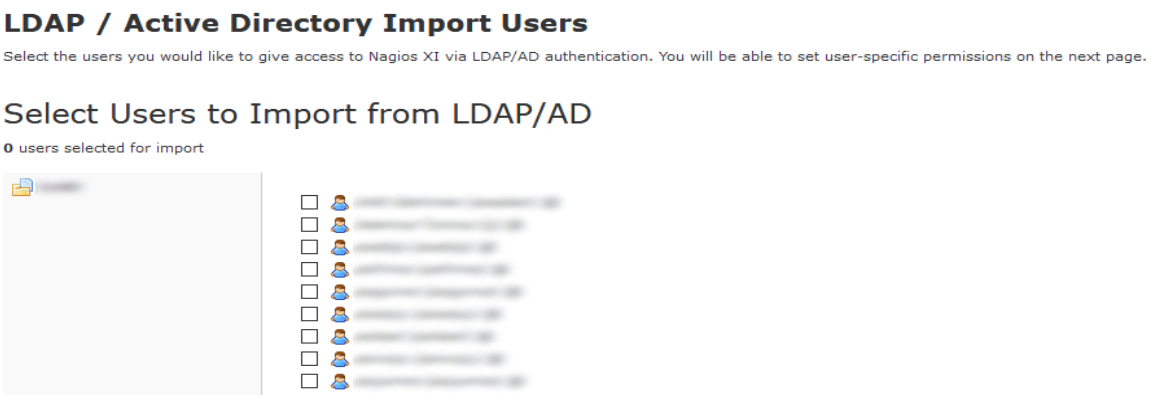

Kuva 13. Esimerkki aktiivihakemiston integraation konfiguroinnista

Integraatioon vaaditaan myös oma sertifikaatti, jonka avulla Nagios pystyy käyttämään aktiivihakemiston tietoja hyväkseen ja sitä tarvitaan myös käyttäjätunnusten autentikaatioon (Nagios Enterprises 2017). Kyseisen sertifikaatin olisi voinut kopioida esimerkiksi sertifikaattipalvelimelta, mutta työn yksinkertaistamiseksi ja nopeuttamiseksi sertifikaatti kopioitiin vanhalta palvelimelta ja lisättiin uudelle palvelimelle tätä kautta.



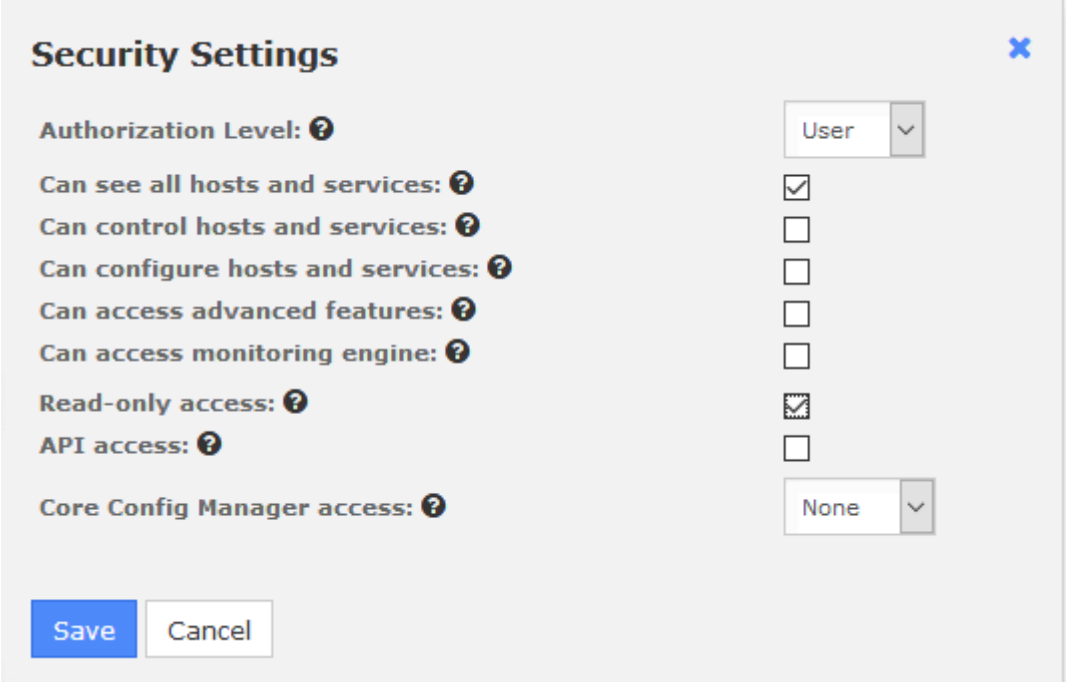

Kuva 14. Haaga-Helian sertifikaatti lisättynä Nagiokseen

Aktiivihakemiston integraation jälkeen lisäsimme organisaation tunnukset onnistuneesti Nagiokseen ja testasimme niiden toimivuuden. Käyttäjätunnukset voitiin lisätä joko yksitellen tai monta samanaikaisesti.




Kuva 15. Esimerkki aktiivihakemiston integraatiolla lisättävistä käyttäjätunnuksista

Käyttäjätunnusten lisäyksen yhteydessä tulee määritellä käyttäjätunnusten asetuksia, kuten sähköpostiosoite sekä aika-asetukset. Vaihtoehtoisesti voidaan asettaa oletusarvot, jotka määritellään erikseen Nagioksen konfiguraatiossa. Käyttäjätunnuksille tuli myös lisätä turva-asetukset, kuten minkälainen käyttäjätunnus on kyseessä, ja antaa haluttuja oikeuksia käyttäjälle. Kaikkien käyttäjien ei esimerkiksi tarvitse olla admin-tunnuksia, sillä kaikki Nagiosta hyödyntävät käyttäjät eivät tule kehittämään sitä.




Kuva 16. Esimerkki käyttäjätunnuksen suoja-asetuksista

# 11 Monitorointi toimeksiantajan palveluympäristössä

Haaga-Helian palvelin- ja palveluympäristön laajuuden takia päätimme yhdessä toimeksiantajan kanssa rajata monitorointiprojektimme kattamaan vain muutaman palvelimen ja jonkin liiketoiminnan kannalta oleellisen palvelun. Palvelimien monitorointi otettiin käyttöön vain muutaman Linux- ja Windows-palvelimen osalta, joista osa on virtuaalisia palvelimia ja osa niin sanottuja raudalla toimivia palvelimia.

Liiketoiminnan kannalta oleelliseksi palveluksi valikoitui Moodle-oppimisalusta, jota opettajat ja opiskelijat käyttävät kurssien materiaalipankkeina sekä tehtävien palautusalustana. Palvelun kriittisyyden lisäksi valintaa puolsi sen verrattain yksinkertainen palvelinympäristö, sillä Moodle ei vaadi toimiakseen useamman palvelimen kokoelmaa, vaan alusta pyörii yksittäisellä palvelimella, eikä täten edellytä laajaa tutustumista sovelluksen palvelinarkkitehtuuriin monitoroinnin toteuttamiseksi. Moodlen monitorointi toteutettiin kuitenkin vain sen testipalvelimen osalta, sillä tuotantopalvelimelle ei tule asentaa mitään, minkä pitkäaikaista vaikutusta muuhun kokonaisuuteen ei ole testattu etukäteen.

## 11.1 Monitorointi Nagios XI:ssä

Nagioksessa monitoroitavat kohteet voidaan jakaa pääosin kahteen eri osioon, objekteihin: *hosts*, eli monitoroitavat laitteet tai kohteet ja *services*, eli monitoroitavien laitteiden tai kohteiden sisäiset kohteet tai komponentit. Molemmille objekteille on myös omat ryhmät: *host groups*¸ johon voidaan asettaa nimensä mukaisesti yksi tai useampi *host*, ja *service groups*, johon voidaan asettaa yksi tai useampi *service*. Nämä ryhmät helpottavat Nagioksen objektien hallittavuutta keskitetysti. (Nagios Enterprises 2018h.)

Hosteilla tarkoitetaan usein fyysisiä päätelaitteita verkossa, kuten palvelimia, työasemia, reitittimiä, verkkokytkimiä ja tulostimia. Laitteiden ei kuitenkaan tarvitse olla fyysisiä, vaan ne voivat usein olla myös virtuaalisia, kuten virtuaalipalvelimia. Hostit voivat myös olla esimerkiksi verkkosivuja, joita monitoroidaan. Hosteille on määritetty jokin tietty osoite, kuten IP-osoite tai MAC-osoite. Hosteihin on liitetty yksi tai useampi service. Hosteilla voi olla isäntä tai lapsi -suhteita muihin hosteihin, mikä on verkkoinfrastruktuurissa tavallista. Verkkoinfrastruktuuri määrittelee missä järjestyksessä Nagioksen monitorointilogiikka toimii. (Nagios Enterprises 2018h.)

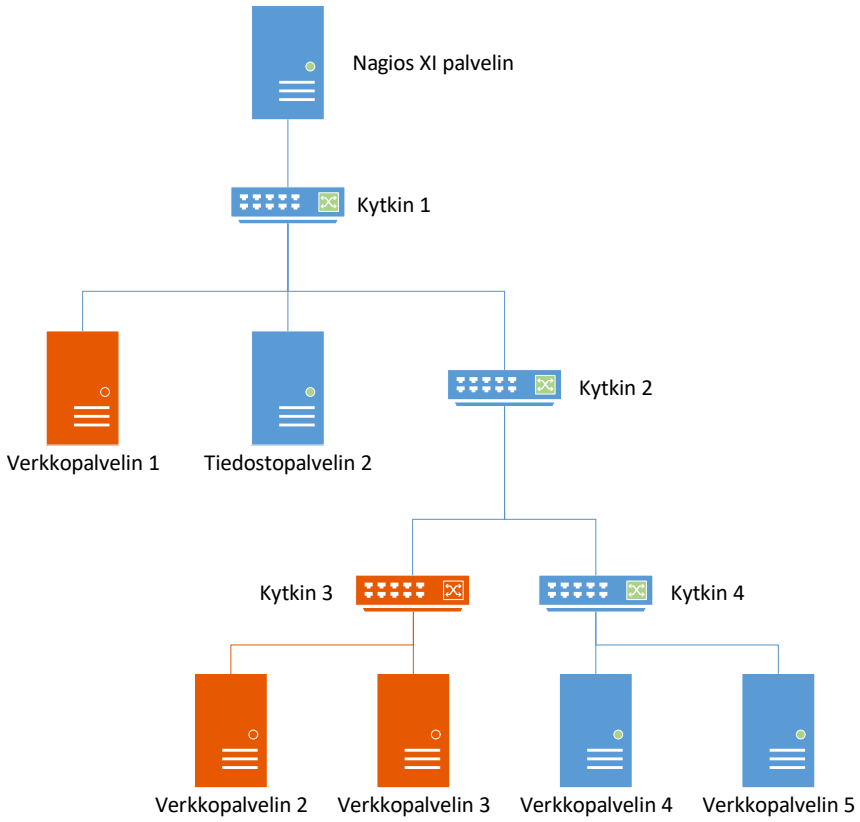
Servicellä tarkoitetaan jotain monitoroitavaa objektia ja ne liittyvät aina johonkin tiettyyn hostiin. Service tarkoittaa usein jonkin hostin attribuuttia, kuten prosessorin käyttöastetta, ylhäällä oloaikaa tai kovalevyn käyttöastetta. Servicet voivat olla myös monimutkaisempia

monitoroitavia kohteita, esimerkiksi hostilla toimivien palveluiden komponentteja, kuten HTTP, SSH ja tietokantaohjelmistoja tai DNS kyselyiden määriä. Servicet ovat siis käytännössä kaikkea sitä, mitä laitteesta tai palvelusta valvotaan. (Nagios Enterprises 2018h.)

## 11.2 Monitoroitavien kohteiden verkkohierarkia Nagios XI:ssä

Monitoroitavat kohteet on suositeltava jakaa niiden roolin kannalta loogisiin ryhmiin, kuten ”*kytkimet*”, *”Moodle”*, *”Linux-palvelimet”* tai *”Windows-palvelimet”*, jolloin tietyille ryhmille voidaan kohdistaa omia sääntöjä tai muutoksia keskitetysti, parantaen kohteiden hallittavuutta. Monitoroitavat kohteet voivat kuulua useampaan eri ryhmään.

Monitoroitaville kohteille tulisi myös verkkoinfrastruktuurin hierarkian kannalta määrittää isäntäkohde, joka käytännössä tarkoittaa kohdetta, jonka täytyy ensin toimia, ennen kuin kyseistä palvelinta tai palvelua on syytä monitoroida. Esimerkiksi verkkoliikenteen kannalta on tärkeää merkitä mikä verkkokytkin on runkoverkon päälaite, jonka kautta kaikki data liikkuu. Hierarkian avulla voidaan määritellä eräänlainen tapahtumaketju monitoroinnille, joka nopeuttaa ja helpottaa ongelmaselvitystä virheiden sattuessa. Hierarkiaa on kuvattu esimerkkiympäristössä kuvassa 17, jossa alemmalla tasolla olevat laitteet vaativat niiden yläpuolella olevien kytkinten toiminnan, jotta niitä voidaan monitoroida.




Kuva 17. Esimerkki verkkoinfrastruktuurin hierakiasta ja Nagioksen kohteiden saatavuuslogiikasta (mukaillen Nagios Enterprises 2018i)

Hierarkiasta voidaan käytännön esimerkkinä ajatella tilanne, jossa jokin kytkin hajoaa. Kytkimen takana olevat palvelimet tai muut laitteet vaativat niiden isäntälaitteena toimivan kytkimen toimivuuden, jotta Nagios voi luottaa niiden olevan osana toimivaa verkkoa. Kytkimen lakatessa toimimasta Nagios huomaa ongelman loogisen järjestyksen ja osaa päätellä, että kytkimen takana olevien palvelimien tila on *unreachable*, eli ne eivät ole saavutettavissa.

Nagios voidaan konfiguroida niin, että saavuttamattomissa olevista palvelimista ei lähetetä mitään vikailmoituksia, mikäli niiden isäntälaiteessa on havaittu vika. Mikäli isäntälaite ei ole saavutettavissa, Nagioksella ei ole keinoa päätellä ovatko hierarkkisesti sen alapuolella olevat laitteet todellisuudessa toiminnassa vai eivät. Kaikissa tapauksissa isäntälaitteen toiminnan lakkaaminen ei välttämättä estä hierarkkisesti sen alapuolella olevien laitteiden toimintaa. (Nagios Enterprises 2018i.)

### 11.3 Nagios XI agentit

Nagios käyttää lähtökohtaisesti agentteja monitoroinnin suorittamiseksi, joka mahdollistaa suoraviivaisen ja selkeän monitoroinnin. Nagios ei sisällä valmiiksi sisäänrakennettuja agentteja vaan turvautuu erillisiin plugineihin monitoroinnin mahdollistamiseksi (Nagios Enterprises 2018l). Työn kannalta oleellisia agentteja ovat NCPA (Nagios Cross-Platform Agent), NRPE (Nagios Remote Plugin Excecutor) ja NSClient++.

NCPA muodostuu kahdesta palvelusta, jotka yhdessä toimivat monitorointiagenttina. NCPA:n hyödyntämät palvelut ovat NCPA Listener ja NCPA Passive. NCPA Listenerin roolina on hallinnoida yhteyksiä web-käyttöliittymään, käsitellä ulkopuolelta tulevia API-pyyntöjä ja tarjota rajapinta graafeille sekä käytetyimmille prosesseille. Lisäksi se tarjoaa NCPA Passivelle sisäisen rajapinnan. NCPA Passive taas huolehtii passiivisten tarkastusten ajosta sekä toimittaa tulokset etukäteen määritettyyn kohteeseen. Jako kahteen palveluun mahdollistaa agentin passiiviset tarkistukset ilman, että agenttiin täytyy sallia yhteydet web-käyttöliittymään tai API:n. Mikäli käytössä ovat vain passiiviset tarkastukset, voidaan toinen osio agentista ottaa kokonaan pois käytöstä resurssinkäytön vähentämiseksi. NCPA on suunniteltu toimimaan Windows-, Linux- ja Unix-järjestelmissä. (Nagios Enterprises 2018m.)

NRPE on ulkoisten päätelaitteiden kanssa kommunikointiin suunniteltu agentti. Tällaisia ulkopuolisia päätelaitteita ovat esimerkiksi Linux- ja Windows-käyttöjärjestelmät. NRPE asennetaan kohdekoneelle, jota monitoroidaan, ja se odottaa käskyjä varsinaiselta Nagios

XI -palvelimelta. Käytännössä agentti mahdollistaa kommunikaation palvelimen ja monitoroitavan kohteen välillä Linux- ja Unix-koneilla, mutta se osaa keskustella myös Windows-koneiden kanssa NSClient++:aa hyödyntäen. (Nagios Enterprises 2018n.)

NSClient++ on vaihtoehtoinen agentti Windows-järjestelmien monitorointiin. Sen tehtävänä on sallia komentojen ajo monitoroitavalla koneella sekä monitoroitujen tietojen lähetys eteenpäin Nagios XI -palvelimelle. Lisäksi agentti suorittaa sille määrättyjä tehtäviä ja lähettää reaaliaikaista dataa määriteltyyn repositorioon. (Nsclient 2018.)

### 11.4 Palvelimien monitorointi

Nagioksen yksi tärkeimmistä ominaisuuksista on palvelimien monitorointi, sillä palvelimet toimivat alustana kaikille muille palveluille, toimivat palvelut sitten virtuaalipalvelimella, oikealla raudalla tai jollakin muulla tietokoneella, kuten Raspberry Pi:llä. Palvelimien monitorointi ei kuitenkaan ole monitoroinnin ainoa tarkoitus.

Yleisesti palvelimien monitoroinnilla käsitetään niiden käyttöasteiden monitorointia, mutta käyttöasteiden monitorointi ei välttämättä anna sellaista dataa, jota voitaisiin hyödyntää ongelmanratkaisussa ongelman rajaamiseksi. Käyttöasteiden monitorointia voidaan kuitenkin hyödyntää ongelmien selvityksen tukena, esimerkiksi selvittämällä millainen muistinkäyttö palvelimella oli, kun jokin ohjelma kaatui. Selittyykö ongelma muistin käytöllä, eli oliko muistin käyttö ainoa tekijä, joka johti ohjelman kaatumiseen. Tällöin voidaan epäillä, että muisti saattoi loppua palvelimesta kesken. Tämän takia käyttöasteiden monitorointi on erittäin hyödyllistä, mutta käyttöasteille ei normaaliolosuhteissa tulisi asettaa raja-arvoja, ellei juuri niitä ole tärkeää monitoroida. Mikäli palvelimen prosessorin käyttöaste on yli 90%, mutta palvelimella toimiva palvelu silti käyttäjän näkökulmasta toimii, voidaan ajatella, ettei mitään ongelmaa tällöin varsinaisesti ole, eikä siitä ei tarvitse hälyttää ketään.

Palvelimia tulisikin siis monitoroida niiden tarjoamien palveluiden kriittisten komponenttien osalta sekä palvelimien omien kriittisten komponenttien osalta. Esimerkiksi sertifikaattien erääntyminen tai palvelimen ajan täsmällisyys ovat sellaisia komponentteja, jotka voivat vaikuttaa verkkopalveluiden saatavuuteen. Muita tärkeitä monitoroitavia komponentteja ovat aikataulutetut tehtävät, joita suoritetaan automaattisesti palvelimella sekä avainprosessit, joita käyttöjärjestelmä ja palvelut tarvitsevat toimiakseen. Myös lokitapahtumia voidaan monitoroida esimerkiksi verkkopalvelimien HTTP-kyselyiden määrän määrittämiseksi. Komponenttien kriittisyyden vaihtelevuuden takia monitorointia pitäisi kehittää palvelimen, palvelun tai laitteen asiantuntijan tai pääkäyttäjän toimesta, jolloin pystytään

määrittelemään oikeat kriteerit esimerkiksi hälytyksien raja-arvoille. Tästä syystä monitorointia ei voida niin sanotusti suoraan pultata kiinni jo olemassa oleviin palveluihin, vaan se vaatii monitoroinnin huomioon ottamisen ja lisäyksen kohteen prosessissa.

Palvelimien monitorointi toteutetaan pääasiallisesti agenttien avulla. Agentit keskustelevat monitorointiratkaisun pääohjelmiston kanssa, jonka avulla voidaan myös lähettää ja vastaanottaa komentoja, joilla monitorointi suoritetaan. Esimerkiksi Nagioksen tapauksessa Nagios-palvelin voi lähettää agentille pyynnön suorittaa jokin komento, jonka jälkeen agentti palauttaa komennosta saadun arvon takaisin Nagios-palvelimelle. Monitorointia voidaan suoritta myös ilman agenttia, mutta agentitonkin monitorointi edellyttää jonkin muun ratkaisun, kuten WMI:n tai SSH:n, käyttöä. SNMP:n käyttö ei ole suositeltavaa palvelinten monitorointiin kuten luvussa 4.6 todettiin, mutta se on silti mahdollista.

## 11.5 Windows-palvelimien monitorointi

Windows-palvelimia on Haaga-Helian verkkoinfrastruktuurissa yli sata, joista jokaisella on jokin oma tehtävä. Suuri osa näistä palvelimista on virtualisoituja palvelimia, sillä näiden palvelinten tuottamat palvelut eivät vaadi suurta määrää dedikoituja resursseja. Virtualisointi on ollut järkevää resurssien ja virrankäytön optimoimiseksi.

### 11.5.1 Nagios XI agentin asennus Windows-palvelimelle

Windows-käyttöjärjestelmien monitorointi alkaa agentin asennuksesta kohdelaitteelle, oli kyseessä sitten palvelin tai työasema. Nagios tarjoaa oman agenttinsa Windows-tietokoneiden ja -palvelimien monitorointiin, mutta monitorointia voidaan suorittaa myös WMI:n avulla, jos palvelimelle ei pystytä sen rajoitteiden takia asentamaan Nagioksen agenttia. Nagios tarjoaa kaksi erilaista agenttia Windowsin monitorointiin, joista toinen on NSClient++, joka toimii vain Windows-käyttöjärjestelmien monitoroinnissa. Toinen agenteista on Nagioksen tarjoama NCPA-agentti. (Nagios Enterprises 2017b & Nagios Enterprises 2019d.)

Opinnäytetyössä käytettiin Nagioksen tarjoamaa NCPA-agenttia Windowsin monitorointiin. NCPA-agentin asennus on suoraviivaista Windowsissa, sillä kaikki asennuksen toimenpiteet voidaan tehdä graafisessa käyttöliittymässä. NCPA-agentti pyytää käyttäjältä ensimmäiseksi tietoja siitä, mitä porttia ja osoitteita halutaan kuunnella NCPA-agentin avulla. Asennuksen alussa määritellään myös *token*, eli merkkijono, joka toimii niin sanottuna salasanana tai avaimena, jota Nagios-palvelin käyttää keskustelussaan agentin

kanssa. Alussa määritellään myös salaus, jota halutaan käyttää. Agentti suosittelee oletuksena uusinta TLS-salausta (Transport Layer Security), mikä on dokumentaatiohetkellä tietoturvan kannalta suositeltavin ratkaisu. (Nagios Enterprises 2019d.)

Seuraavassa vaiheessa asennus pyytää tietoja passiivista monitorointia varten Nagios-palvelimen NRDP:n, eli Nagios Remote Data Processorin, osoitteesta, jota käytetään datan säilömisessä ja prosessoimisessa Nagios-palvelimella. Muita määriteltäviä tietoja on NRDP:n käyttämä token, joka saadaan selville esimerkiksi Nagios-palvelimen verkkokäyttöliittymän avulla. (Nagios Enterprises 2019d.)

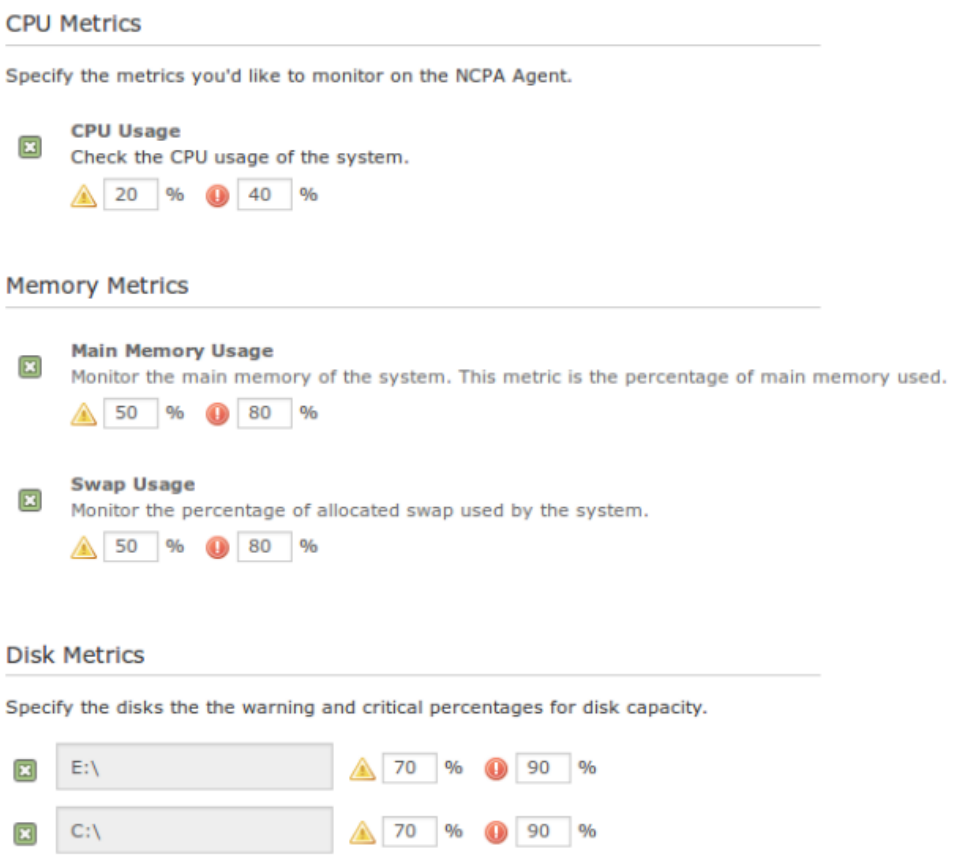
Viimeisenä vaiheena NCPA-agentin asennuksen valmistuttua on syytä tarkistaa, onko asennuksessa määritelty portti avattu Windowsin palomuurista, jotta keskustelu on mahdollista Nagios-palvelimen ja Windows-agentin välillä. (Nagios Enterprises 2019d.)

NCPA-agentti voidaan asentaa myös niin sanotussa hiljaisessa tilassa, jolloin käyttöjärjestelmän toiminta ei keskeydy esimerkiksi työasemia monitoroitaessa. Halutessaan Haaga-Helia voi hyödyntää NCPA-agentin käyttöönottoa organisaation kriittisissä työasemissa esimerkiksi System Center Configuration Managerin, eli SCCM:n, avulla, jota voidaan hyödyntää ohjelmistojen tai päivitysten asentamisessa henkilöstön työasemille.

### 11.5.2 Nagios XI agentin konfigurointi ja käyttöönotto Windows-palvelimella

Nagioksen Windows-agentin asennuksen jälkeen palvelin tai työasema voidaan liittää monitoroitavaksi kohteeksi Nagios-palvelimelle, esimerkiksi Nagioksen verkkokäyttöliittymän avulla. Monitoroinnin konfiguraatio aloitetaan määrittelemällä monitoroitava kohde, jolle agentti on tässä tapauksessa asennettu. Määritys tehdään hyödyntäen IP-osoitetta, porttia ja tokenia. (Nagios Enterprises 2017c.)

Tämän jälkeen Nagios tarkistaa keskusteluyhteyden agentin kanssa tarkistamalla tokenin. Seuraavassa vaiheessa määritetään ne servicet, eli komponentit tai prosessit, joita halutaan monitoroida kyseisen kohteen osalta, kuten kuvassa 18 esitellään. Kuvassa näkyvät keltaisella varoituskolmiolla merkityt raja-arvot esimerkiksi CPU:n, eli prosessorin, kohdalla tarkoittavat sitä raja-arvoa, jolloin monitorointi ilmoittaa monitoroitavan kohteen olevan varoitustilassa. Seuraava raja-arvo, joka on merkitty punaisella ympyrällä, tarkoittaa kriittisen tilan raja-arvoa. Nagioksen oletusarvot esimerkiksi prosessorin käyttöasteelle ovat alhaiset, joten ne on syytä muuttaa monitoroitavan kohteen mukaisiksi. (Nagios Enterprises 2017c.)




Kuva 18. NCPA Windows-agentin monitoroitavien kohteiden valinta ja raja-arvot

Monitoroitavan kohteen komponenttien määrityksessä on syytä ottaa huomioon myös käyttöjärjestelmässä toimivat kriittiset prosessit ja palvelut. Esimerkiksi jos Windows-palvelimella toimii jokin verkkopalvelin tai DHCP-palvelu, kuten esitetty kuvassa 19, olisi sitä syytä monitoroida, ja lähettää hälytys, jos prosessi kaatuu. Työasemien kohdalla monitoroitava prosessi voisi olla esimerkiksi Spooler, joka on tulostuksessa käytetty primääri komponentti, joka hallinnoi käyttöjärjestelmän tulostusjonoa. (Nagios Enterprises 2017c.)

Monitoroitavan kohteen konfiguraation jälkeen sen tilaa voidaan tarkastella esimerkiksi Nagioksen web-käyttöliittymästä.

| Host | Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|---|
| [redacted] | CPU Usage | Ok | 9d 23h 34m 14s | 1/5 | 2019-05-09 23:25:34 | OK (Sample Period 3597 sec) - Average CPU Utilisation 0.19% |
| | DHCP service | Ok | 15d 9h 54m 49s | 1/5 | 2019-05-09 23:58:37 | OK - Found 1 Services(s), 1 OK and 0 with problems (0 excluded). 'DHCP Server' (DHCPServer) is Running. |
| | Drive C: Disk Usage | Ok | 9d 23h 34m 27s | 1/5 | 2019-05-09 23:34:28 | OK - C: Total=99.51GB, Used=18.28GB (18.4%), Free=81.23GB (81.6%) |
| | Memory Usage | Ok | 15d 12h 43m 33s | 1/5 | 2019-05-09 23:48:17 | OK - Physical Memory: Total: 6GB - Used: 1.055GB (18%) - Free: 4.944GB (82%) |
| | Page File Usage | Ok | 9d 23h 34m 56s | 1/5 | 2019-05-09 23:35:54 | Overall Status - OK. Individual Page Files Detail: OK - C:\pagefile.sys Total: 1GB - Used: 284MB (28%) - Free: 740MB (72%), Peak Used: 313MB (31%) - Peak Free: 711MB (69%) |
| | Ping | Ok | 15d 9h 55m 2s | 1/5 | 2019-05-10 00:05:16 | PING OK - Packet loss = 0%, RTA = 0.57 ms |

Kuva 19. Esimerkki Windows DHCP-palvelimen monitoroinnista

## 11.6 Linux-palvelimien monitorointi

Haaga-Helian verkkoinfrastruktuurista löytyy Windows-palvelimien lisäksi kymmeniä Linux-palvelimia, joista suuri osa on virtualisoituja palvelimia ja osa, kuten Nagios XI:n palvelin, toimii muusta palvelinraudasta irrallisena omana yksikkönään.

Linux-palvelimia voidaan valvoa usealla eri tavalla, kuten NCPA- tai NRPE-agenttien avulla tai esimerkiksi SSH:n avulla. Opinnäytetyössä käytimme NRPE-agenttia monitoroinnin toteuttamiseen Linux-palvelimilla, jonka lisäksi osassa palvelimia hyödynnettiin SSH-yhteyttä käyttävää pluginia. Linux-palvelimen monitorointi aloitetaan Nagioksen NRPE-agentin asennuksesta, jonka jälkeen monitoroitava palvelin lisätään Nagioksen monitoroitaviin kohteisiin, jolloin sille määritellään monitoroitavat komponentit.

### 11.6.1 Nagios XI agentin asennus Linux-palvelimelle

Linux-palvelimien valvonta alkaa agentin asennuksesta. Agentin uusin versio on suositeltava ladata Nagioksen verkkosivuilta, jolloin se puretaan ja voidaan asentaa palvelimelle asennusskriptin avulla. (Nagios Enterprises 2017d.)

*cd /tmp || wget https://assets.nagios.com/downloads/nagiosxi/agents/linux-nrpe-agent.tar.gz*

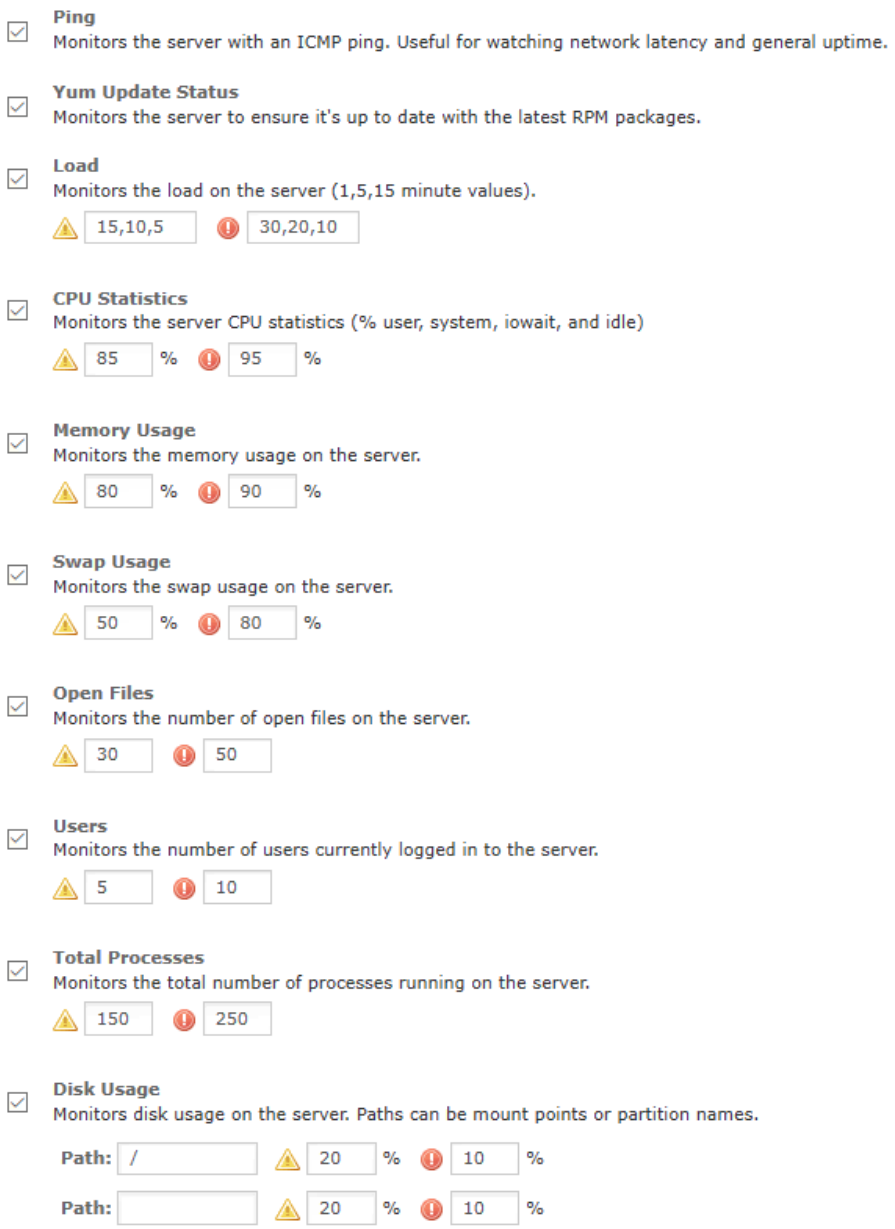
*tar xzf linux-nrpe-agent.tar.gz || cd linux-nrpe-agent*
*./fullinstall*

Agentin asennuksen yhteydessä asennusskripti pyytää Nagios-palvelimen IP-osoitteen, jonka avulla skripti osaa tehdä tarvittavat konfiguraatiot, jotta agentti pystyy keskustelemaan Nagios-palvelimen kanssa. Skripti tekee myös kaikki muut tarvittavat toimenpiteet palvelimen eri osiin, kuten palomuuriin. Lisäksi skripti asentaa kaikki agentin toiminnan kannalta tarvittavat paketit oikeista repositorioista. Agentin asennusskripti luo tarvittavat käyttäjätunnukset tietoturvan parantamiseksi, jotta agenttia ei tarvitse ajaa käyttöjärjestelmän root-käyttäjätunnuksella. (Nagios Enterprises 2017d.)

### 11.6.2 Nagios XI agentin konfigurointi ja käyttöönotto Linux-palvelimella

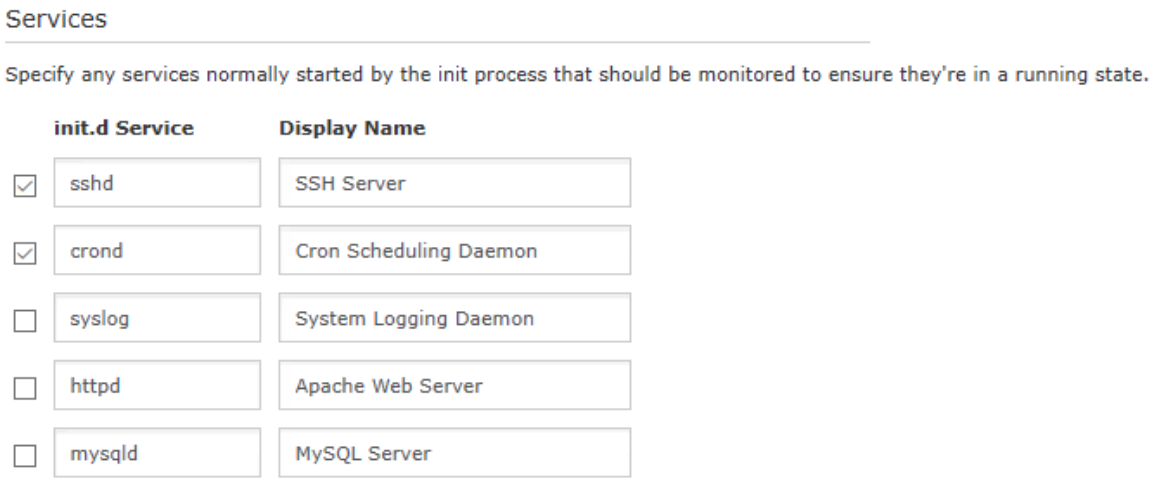
Seuraava askel Linuxin monitoroinnissa on määrittää monitoroitavat kohteet eli *servicet*. Nagioksen web-käyttöliittymässä voidaan määrittää monitoroitavat kohteet ja niiden raja-arvot helposti monitoroinnin käyttöönoton yhteydessä. Raja-arvoja ja monitoroitavia kohteita voidaan muuttaa, lisätä tai poistaa myös jälkikäteen laite- tai ryhmäkohtaisesti.

Oletusarvoisesti Nagios tarjoaa Linux-palvelimen monitorointiin muun muassa päivitysten seurantaa, komponenttien, kuten prosessorin, muistin ja kovalevyjen statistiikkaa ja käyttöasteita, kuten esitetty kuvassa 20. Lisäksi voidaan seurata, kuinka monta kirjautunutta käyttäjää palvelimella on tai kuinka monta aktiivista prosessia palvelimella toimii sekä valvoa avainprosessien ja -palveluiden monitorointia.




Kuva 20. Esimerkki Linux-palvelimelle tarjottavista oletusmonitoreista

Osassa palvelimista monitoroitiin myös Linuxin toiminnan ja hallittavuuden kannalta kriittisiä ja oleellisia käyttöjärjestelmä- ja ohjelmistoprosesseja, kuten SSH-palvelinta sshd ja ajastettujen toimintojen palvelua crond:tä, jotka on valittu monitoroitavaksi kuvassa 21.




Kuva 21. Esimerkki Nagios XI:n tarjoamasta palvelujen monitoroinnista Linux-palvelimella

Verkkopalvelimille voidaan määritellä monitoroitaviksi kohteiksi myös esimerkiksi verkkopalvelusta vastaava alusta, kuten Apache web-palvelin. Kuvassa 22 on esitetty esimerkki Linux-palvelimella monitoroitavista komponenteista ja prosesseista.

| Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|
| / Disk Usage | Ok | 25m 14s | 1/5 | 08/05/2019 22:11:16 | DISK OK - free space: / 43486 MB (84.97% inode=100%): |
| Apache HTTP Server | Ok | 29m 36s | 1/5 | 08/05/2019 22:08:02 | active |
| CPU Stats | Ok | 30m 9s | 1/5 | 08/05/2019 22:08:25 | CPU STATISTICS OK: user=0.53% system=0.06% iowait=0.00% idle=99.41% |
| Cron Scheduling Daemon | Ok | 29m 20s | 1/5 | 08/05/2019 22:11:22 | active |
| Current Load | Ok | 22m 48s | 1/4 | 08/05/2019 22:09:41 | OK - load average: 0.36, 0.26, 0.23 |
| Current Users | Ok | 29m 20s | 1/4 | 08/05/2019 22:11:14 | USERS OK - 2 users currently logged in |
| HTTP | Ok | 29m 20s | 1/4 | 08/05/2019 22:02:12 | HTTP OK: HTTP/1.1 200 OK - 3488 bytes in 0.001 second response time |
| Load | Ok | 25m 0s | 1/5 | 08/05/2019 22:11:27 | OK - load average: 0.39, 0.27, 0.24 |
| Memory Usage | Ok | 29m 20s | 1/5 | 08/05/2019 22:10:04 | OK - 30273 / 31877 MB (94%) Free Memory, Used: 1589 MB, Shared: 498 MB, Buffers + Cached: 5359 MB |
| Open Files | Ok | 30m 20s | 1/5 | 08/05/2019 22:09:54 | OK: 3328 open files (0% of max 3231832) |
| Ping | Ok | 1h 30m | 1/5 | 08/05/2019 22:10:42 | PING OK - Packet loss = 0%, RTA = 0.04 ms |
| Postfix Mail Transport Agent | Ok | 29m 20s | 1/5 | 08/05/2019 22:11:30 | active |
| Root Partition | Ok | 22m 48s | 1/4 | 08/05/2019 22:02:09 | DISK OK - free space: / 43514 MB (85.03% inode=100%): |
| Service Status - crond | Ok | 29m 20s | 1/4 | 08/05/2019 22:02:17 | ● crond.service - Command Scheduler |
| Service Status - httpd | Ok | 22m 48s | 1/4 | 08/05/2019 22:10:24 | ● httpd.service - The Apache HTTP Server |
| Service Status - mysqld | Ok | 29m 20s | 1/4 | 08/05/2019 22:10:47 | ● mariadb.service - MariaDB database server |
| Service Status - ndo2db | Ok | 22m 48s | 1/4 | 08/05/2019 22:11:25 | ● ndo2db.service - Nagios Data Out Daemon |
| SSH | Ok | 29m 20s | 1/4 | 08/05/2019 22:10:17 | SSH OK - OpenSSH_7.4 (protocol 2.0) |
| SSH Server | Ok | 29m 20s | 1/5 | 08/05/2019 22:09:08 | active |
| Swap Usage | Ok | 29m 20s | 1/5 | 08/05/2019 22:02:14 | SWAP OK - 100% free (16063 MB out of 16063 MB) |
| Total Processes | Ok | 22m 48s | 1/5 | 08/05/2019 22:09:04 | PROCS OK: 459 processes |
| Users | Ok | 24m 45s | 1/5 | 08/05/2019 22:06:45 | USERS OK - 2 users currently logged in |
| Yum Updates | Ok | 22m 16s | 1/5 | 08/05/2019 22:09:29 | YUM OK: O/S is up to date. |

Kuva 22. Esimerkki Linux-palvelimen monitoroinnista

Konfiguraatiossa voidaan määritellä myös erilliset arvot monitoroinnin tiheydelle, jos kyseessä on aktiivinen monitorointiratkaisu, jossa monitorointipalvelin itse kyselee tietoja kohdepalvelimelta määritellyllä tavalla.

Konfiguraation jälkeen Nagios lisää kyseessä olevan palvelimen monitoroitavaksi kohteeksi, jonka jälkeen sen keräämää statistiikkaa ja dataa voidaan tarkastella. Aluksi moni-

toroitavista kohteista ei ole mitään dataa, ennen kuin Nagios ottaa kohdepalvelimeen ensimmäisen kerran yhteyttä, jolloin agentti suorittaa määritellyt kyselyt ja palauttaa kerätyn datan Nagios XI -palvelimelle. Nagios-palvelin määrittelee, ovatko kerätyt arvot raja-arvojen sisällä. Mikäli raja-arvot ylittyvät, lähettää Nagios aiheesta varoituksen tai hälytyksen sille asetettujen ehtojen perusteella.

### 11.7 VMwaren virtualisointialustat

Linux- ja Windows-palvelimien monitoroinnin lisäksi Nagioksella voidaan monitoroida myös muita palvelinalustoja, kuten VMwaren tarjoamaa virtualisointialustaa ESX:ää. VMwaren virtualisointialustojen valvontaan Nagios XI kuitenkin tarvitsee lisäosia, jotta se pystyy valvomaan kyseisiä komponentteja.

Monitorointi vaatii VMware Perl SDK:n (Software development kit), joka vaatii Perl-moduuleja ja muuttujia toimiakseen. Perl-moduulit asennetaan CPAN, eli *Comprehensive Perl Archive Networkin* avulla. (Nagios Enterprises 2019e.)

```
yum install -y libxml2-devel xml2 libuuid-devel perl-XML-LibXML perl-Env
export PERL_MM_USE_DEFAULT=1
cpan -i App::cpanminus
cpanm --notest Module::Build Crypt::SSLeay
```

Nagios XI ei sisällä VMware Perl SDK:iä valmiina lisensoinnin takia, joten se täytyy itse ladata ja asentaa Nagios-palvelimelle. VMWare Perl SDK voidaan ladata VMwaren verkkosivuilta. (Nagios Enterprises 2019e.)

```
cd /tmp
tar xzf VMware-vSphere*SDK*.tar.gz
cd vmware-vsphere-cli-distrib/
./vmware-install.pl EULA_AGREED=yes
```

SDK:n latauksen jälkeen tar-paketti puretaan, jonka jälkeen SDK voidaan asentaa ja sen myötä mahdollistaa WMwaren virtualisointialustojen monitorointi. (Nagios Enterprises 2019e.)

Monitoroitavat kriteerit ja komponentit voidaan asettaa samalla tavalla Nagioksen web-käyttöliittymän avulla, kuin Windows- ja Linux-palvelimien monitoroinnissa. Konfiguraati-

ossa määritellään ensin VMwaren IP-osoite, käyttäjätunnus ja salasana, jolla kyselyjä voidaan tehdä. Seuraavaksi Nagios autentikoi annetut tunnustiedot VMware-palvelimella, jonka jälkeen voidaan siirtyä määrittelemään monitoroitavia kohteita. Nagios tarjoaa oletuksena yksinkertaisia monitoroitavia kohteita, kuten prosessorin käyttöaste, muistin käyttöaste, verkkolaitteen kaistanopeus ja käyttöaste, prosessien status sekä isäntäpalvelimen status, kuten esitetty kuvassa 23. (Nagios Enterprises 2019e.)


Kuva 23. Esimerkkikuva VMwaren isäntälaitteen monitoroitavista kohteista (Nagios Enterprises 2019e)

Monitoroitavien kohteiden määrittelyn jälkeen VMware-palvelin lisätään Nagioksen listaukseen monitoroitavista laitteista, jonka jälkeen sen tilaa voidaan tarkastella Nagioksen verkkokäyttöliittymän avulla, joka on nähtävissä kuvassa 24.

| Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|
| CPU Usage for VMHost | Ok | 2h 31m 39s | 1/5 | 2016-12-06 16:55:09 | ESX3 OK - cpu usage=3019.00 MHz (14.41%) |
| Datastore usage for VMHost | Ok | 2h 29m 39s | 1/5 | 2016-12-06 16:57:09 | ESX3 OK - storages : Q02SP02T01L01=190499.00 MB (20.27%), Q02SP01T01L01=502915.00 MB (60.06%), Q01SP04T01L01=2040170.00 MB (72.71%), Q01SP03T01L01=1053760.00 MB (37.56%), Q01SP02T01L01=1155703.00 MB (62.09%), Q01SP01T01L01=651663.00 MB (79.97%) |
| Input / Output for VMHost | Ok | 2h 31m 4s | 1/5 | 2016-12-06 16:55:51 | ESX3 OK - io commands aborted=0, io bus resets=0, io read latency=0 ms, write latency=1 ms, kernel latency=0 ms, device latency=1 ms, queue latency=0 ms |
| Memory for VMHost | Ok | 2h 30m 45s | 1/5 | 2016-12-06 16:56:12 | ESX3 OK - mem usage=81867.98 MB (62.52%), overhead=0.00 MB, swapped=0.00 MB |
| Networking for VMHost | Ok | 2h 29m 27s | 1/5 | 2016-12-06 16:57:18 | ESX3 OK - net receive=5.00 KB/s, send=781.00 KB/s, 2/7 NICs are disconncted |
| Services for VMHost | Ok | 2h 31m 26s | 1/5 | 2016-12-06 16:55:27 | ESX3 OK - services : DCUI (up), TSM (up), TSM-SSH (up), lbtd (up), lwsmd (down), ntpd (up), pcscd (down), sfcbd-watchdog (up), snmpd (down), vmsyslogd (up), vprobed (down), vpxa (up), xorg (down) |
| VM Status for VMHost | Ok | 2h 29m 31s | 1/5 | 2016-12-06 16:57:15 | ESX3 OK - 26/180 VMs up, overall status=green, connection state=connected, maintenance=no, 105 health issue(s), 2 config issue(s) |

Kuva 24. Esimerkkikuva VMware ESX klusterin monitoroinnista Nagioksen web-käyttöliittymässä (Nagios Enterprises 2019e)

VMwaren isäntälaitteen lisäksi on mahdollista monitoroida myös virtualisointialustalla toimivia virtuaalikoneita. Virtuaalikoneita voidaan halutessa valvoa perinteisellä Windows- tai

Linux-palvelimien monitoroinnilla, mutta myös WMwaren isäntäpalvelimen kautta on mahdollista valvoa sen päällä toimia virtuaalikoneita ja niiden resurssienkäyttöä koko kokonaisuudesta. Valvottavat virtuaalikoneet valitaan monitoroitavien kohteiden konfiguroinnin yhteydessä, kuten esitetty kuvassa 25. (Nagios Enterprises 2019e.)


Kuva 25. Esimerkki VMwaren klusterin virtuaalikoneiden valinnasta monitoroitaviksi kohteiksi (Nagios Enterprises 2019e)

Toimenpiteen jälkeen kyseiset virtuaalialustat on saatettu monitoroinnin piiriin, jolloin niitä voidaan tarkastella Nagioksen web-käyttöliittymän avulla sen jälkeen, kun Nagios on lähettänyt kaikki kyselyt virtualisointialustan valituille virtuaalikoneille. Kuvassa 26 esitetään esimerkki siitä, miten virtuaalikoneita voidaan valvoa virtualisointialustan kautta.

| Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|
| dc01 CPU Usage | Ok | 2h 48m 56s | 1/5 | 2016-12-06 17:22:48 | ESX3 OK - "dc01" cpu usage=95.00 MHz(2.72%) wait=19354.00 ms |
| dc01 Input / Output | Ok | 2h 47m 27s | 1/5 | 2016-12-06 17:22:50 | ESX3 OK - "dc01" io usage=0.04 MB, read=0.00 MB/s, write=0.04 MB/s |
| dc01 Memory | Ok | 2h 47m 16s | 1/5 | 2016-12-06 17:22:54 | ESX3 OK - "dc01" mem usage=3030.00 MB(3.99%), overhead=36.14 MB, active=122.88 MB, swapped=0.00 MB, swapin=0.00 MB, swapout=0.00 MB |
| dc01 Networking | Ok | 2h 48m 18s | 1/5 | 2016-12-06 17:22:57 | ESX3 OK - "dc01" net receive=0.00 KB/s, send=0.00 KB/s |
| dc01 VM Status | Ok | 2h 47m 30s | 1/5 | 2016-12-06 17:23:00 | ESX3 OK - "dc01" status=green, run state=UP, guest state=Running, max cpu=3491 MHz, max mem=3072 MB, console connections=0, tools status=Old, has no config issues |
| dc02 CPU Usage | Ok | 2h 49m 44s | 1/5 | 2016-12-06 17:23:03 | ESX3 OK - "dc02" cpu usage=92.00 MHz(2.63%) wait=19338.00 ms |
| dc02 Input / Output | Ok | 2h 48m 46s | 1/5 | 2016-12-06 17:23:06 | ESX3 OK - "dc02" io usage=0.07 MB, read=0.00 MB/s, write=0.07 MB/s |
| dc02 Memory | Ok | 2h 47m 19s | 1/5 | 2016-12-06 17:23:09 | ESX3 OK - "dc02" mem usage=3032.00 MB(6.99%), overhead=36.19 MB, active=215.04 MB, swapped=0.00 MB, swapin=0.00 MB, swapout=0.00 MB |
| dc02 Networking | Ok | 2h 47m 16s | 1/5 | 2016-12-06 17:23:12 | ESX3 OK - "dc02" net receive=0.00 KB/s, send=0.00 KB/s |
| dc02 VM Status | Ok | 2h 47m 16s | 1/5 | 2016-12-06 17:23:15 | ESX3 OK - "dc02" status=green, run state=UP, guest state=Running, max cpu=3491 MHz, max mem=3072 MB, console connections=0, tools status=Old, has no config issues |

Kuva 26. Esimerkki VMwaren klusterissa sijaitsevien virtuaalikoneiden valvonnasta Nagioksen web-käyttöliittymässä (Nagios Enterprises 2019d)

Virtuaalikoneiden osalta monitorointia voidaan parantaa monitoroimalla virtuaalikoneiden käyttöjärjestelmiä, jolloin saadaan yksityiskohtaisempaa dataa niillä ajettavista ohjelmistoista ja kokonaisuuksista. Kuvassa 26 esitellyistä virtuaalikoneista puuttuu kriittisten palveluiden monitorointi, joka on palvelinten kuvauksen (dc01 ja dc02, eli domain controllerit 1 ja 2) kannalta tärkeää.

## 11.8 Muut monitoroitavat kohteet

Nagios XI tarjoaa oletuksena työkaluja, joiden avulla voidaan monitoroida muitakin kohteita kuin vain palvelimia ja niillä toimivia palveluita. Monitorointia voidaan laajentaa erityisesti verkkokytkinten ja reitittimien monitorointiin, jolloin voidaan helposti määritellä tietoliikenteen sidossuhteita, esimerkiksi mitkä palvelimet ovat minkäkin kytkimen takana. Jos jokin kytkin ei vastaa, silloin ei tarvitse lähettää hälytyksiä kaikista sen takana olevista kohteista. Ensisijaisesti hälytyksiä tulisi aina lähettää hierarkiassa korkeimmalla olevasta kohteesta, jotta ongelmia voidaan ratkaista niiden muodostumisjärjestyksessä.

Verkkolaitteita voidaan monitoroida yksinkertaisella ping-kyselyllä, mutta monitorointi on lisäksi mahdollista myös laitteiden omien rajapintojen avulla, sikäli kun sellaisia on, tai esimerkiksi SNMP-kyselyiden avulla. Kuvassa 27 on esitetty yksinkertainen ping-kysely verkkokytkimelle ja kuvassa 28 kysely on toteutettu SNMP-kyselynä, jolloin monitoroinnista saatavaa tietoa on runsaammin.

| Host | Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|---|
| [redacted] | Ping | Ok | 1h 13m 51s | 1/5 | 10/05/2019 00:30:33 | OK - [redacted] rta 3.472ms, lost 0% |

Kuva 27. Esimerkki yksinkertaisesta verkkokytkimen monitoroinnista

| Host | Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|---|
| switch27.nagios.local | Bandwidth Spike | Ok | 13m 12s | 1/1 | 2019-05-11 08:21:34 | OK: 20 MB/s reported |
| | Ping | Critical | 78d 15h 48m 43s | 1/1 | 2019-05-11 08:20:56 | CRITICAL - 192.168.5.41: rta nan, lost 100% |
| | Port 1 Bandwidth | Ok | 420d 15h 39m 20s | 1/1 | 2019-05-11 08:21:55 | OK - Current BW in: 0Mbps Out: 0Mbps |
| | Port 1 Status | Ok | 1397d 9h 59m 22s | 1/1 | 2018-03-16 16:30:56 | OK: Interface Port: 1 Gigabit - Level (index 1) is up. |
| | Port 10 Bandwidth | Ok | 1397d 10h 44m 9s | 1/1 | 2019-05-11 08:22:13 | OK - Current BW in: 0Mbps Out: 0Mbps |
| | Port 10 Status | Critical | 1397d 11h 11m 19s | 1/1 | 2018-07-27 08:49:32 | CRITICAL: Interface Port: 10 Gigabit - Level (index 10) is down. |
| | Port 11 Bandwidth | Ok | 1397d 11h 7m 31s | 1/1 | 2019-05-11 08:21:13 | OK - Current BW in: 0Mbps Out: 0Mbps |
| | Port 11 Status | Critical | 420d 15h 48m 33s | 1/1 | 2018-03-16 16:39:56 | CRITICAL: Interface Port: 11 Gigabit - Level (index 11) is down. |
| | Port 12 Bandwidth | Ok | 1397d 11h 15m 21s | 1/1 | 2019-05-11 08:24:34 | OK - Current BW in: 0Mbps Out: 0Mbps |
| | Port 12 Status | Critical | 1397d 10h 12m 50s | 1/1 | 2018-03-16 16:42:45 | CRITICAL: Interface Port: 12 Gigabit - Level (index 12) is down. |
| | Port 13 Bandwidth | Ok | 1397d 10h 54m 19s | 1/1 | 2019-05-11 08:23:13 | OK - Current BW in: 0Mbps Out: 0Mbps |
| | Port 13 Status | Critical | 420d 15h 43m 22s | 1/1 | 2018-07-12 14:59:15 | CRITICAL: Interface Port: 13 Gigabit - Level (index 13) is down. |
| | Port 14 Bandwidth | Ok | 1397d 11h 14m 19s | 1/1 | 2019-05-11 08:24:13 | OK - Current BW in: 0Mbps Out: 0Mbps |
| | Port 14 Status | Critical | 1397d 11h 11m 19s | 1/1 | 2018-07-12 15:01:49 | CRITICAL: Interface Port: 14 Gigabit - Level (index 14) is down. |
| | Port 15 Bandwidth | Ok | 1397d 10h 53m 8s | 1/1 | 2019-05-11 08:22:55 | OK - Current BW in: 0Mbps Out: 0Mbps |
| | Port 15 Status | Critical | 302d 17h 20m 32s | 1/1 | 2018-07-12 15:04:14 | CRITICAL: Interface Port: 15 Gigabit - Level (index 15) is down. |
| | Port 16 Bandwidth | Ok | 1397d 11h 11m 19s | 1/1 | 2019-05-11 08:20:35 | OK - Current BW in: 0Mbps Out: 0Mbps |
| | Port 16 Status | Warning | 38d 21h 2m 33s | 1/1 | 2019-04-02 11:22:13 | WARNING: SNMP error: No response from remote host '192.168.5.41' |

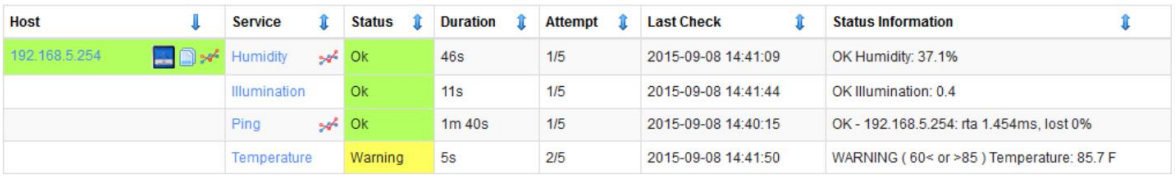
Kuva 28. Esimerkki SNMP-kyselyillä toteutetusta kytkimen monitoroinnista

Monitorointia voidaan laajentaa myös muihin laitteisiin, jos ne tukevat jotain keskustelutapaa, kuten SNMP:tä. Tulostinten kohdalla monitorointi voidaan toteuttaa hyödyntämällä jotain valmistajan omaa keskustelukanavaa tulostinten välillä. Nagios XI tarjoaa suoran tuen esimerkiksi HP:n JetDirect:n kanssa, jolla voidaan toteuttaa esimerkiksi tulostimen tilan tarkastuksia, kuten esitetty kuvassa 29. Canonin tulostimia voidaan monitoroida SNMP:n avulla (Canon, Inc 2015).

| Host | Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|---|
| 10.0.1.60 | Ping | Ok | 16h 41m 56s | 1/5 | 2017-08-01 20:42:40 | OK - 10.0.1.60: rta 15.976ms, lost 0% |
| | Printer Status | Warning | 7h 0m 42s | 1/5 | 2017-08-01 20:45:23 | Printer Offline ("Pause ") |

Kuva 29. Esimerkki verkkotulostimen monitoroinnista HP:n JetDirectin ja ping-tarkistuksen avulla

Nagios XI:n avulla voidaan toteuttaa myös muiden laitteiden, kuten IoT-laitteiden ja niiden sensorien monitorointia. Tällaisten sensoreiden monitorointiin ei usein kuitenkaan löydy mitään valmista pluginia Nagioksesta, sillä Nagios kattaa vain muutaman erilaisen sensorin monitoroinnin oletuksena. Sensoreiden monitorointiin voidaan hyödyntää itse tehtyjä ja räätälöityjä skriptejä, joita Nagios voi tarkkailla esimerkiksi SNMP:n, SSH:n tai esimerkiksi Raspberry Pi:lle asennettavan Linux-palvelimen agentin avulla. Kuvassa 30 on esitetty esimerkkinä tietoja, joita voidaan kerätä IoT-laitteesta.



| Host | Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|---|
| 192.168.5.254 | Humidity | Ok | 46s | 1/5 | 2015-09-08 14:41:09 | OK Humidity: 37.1% |
| | Illumination | Ok | 11s | 1/5 | 2015-09-08 14:41:44 | OK Illumination: 0.4 |
| | Ping | Ok | 1m 40s | 1/5 | 2015-09-08 14:40:15 | OK - 192.168.5.254: rta 1.454ms, lost 0% |
| | Temperature | Warning | 5s | 2/5 | 2015-09-08 14:41:50 | WARNING ( 60< or >85 ) Temperature: 85.7 F |

Kuva 30. Esimerkki Websensor EM08:n monitoroinnista (Nagios Enterprises 2018e)

# 12 Moodlen monitorointi

Kuten aiemmin todettiin, valitsimme ensisijaisesti monitoroitavaksi kohteeksi Moode-testipalvelimen. Moodlen monitoroinnilla on kaksi tarkoitusta, toinen niistä on Moodlen palvelimen ja palvelun käyttöasteiden monitorointi. Monitoroinnin kohteista voidaan saada dataa esimerkiksi siitä, mihin palvelimen resurssiin käyttäjämäärä vaikuttaa ja voiko sillä olla suhdetta palvelun vakauteen. Toinen monitoroinnin tarkoitus on kuormitustestaus, jolloin palvelua voidaan testata sen kuormankestävyyden kannalta. Näin voidaan kerätä tärkeää tietoa mahdollisten pullonkaulojen ja ongelmakohtien löytämiseksi.

## 12.1 Moodlen testipalvelimien komponenttien monitorointi

Palvelimen monitorointi tapahtuu samalla tapaa kuin minkä tahansa käyttöliittymän monitorointi Nagios XI:n avulla. Ensimmäiseksi asennetaan palvelua tuottavalle palvelimelle agentti, joka kerää palvelimen komponenteista dataa, kuten prosessorin käyttöasteesta ja muistin käytöstä. Moodlen testipalvelimelta kerättyä dataa on esitetty kuvassa 31.

| Host | Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|---|
| [redacted] | / Disk Usage | Ok | 18m 35s | 1/5 | 09/05/2019 23:38:10 | DISK OK - free space: / 9376 MB (24.43% inode=99%): |
| | CPU Stats | Ok | 18m 22s | 1/5 | 09/05/2019 23:38:23 | CPU STATISTICS OK: user=0.10% system=0.10% iowait=0.00% idle=99.80% |
| | Cron Scheduling Daemon | Ok | 18m 28s | 1/5 | 09/05/2019 23:38:17 | active |
| | Load | Ok | 17m 29s | 1/5 | 09/05/2019 23:39:15 | OK - load average: 0.00, 0.03, 0.08 |
| | Memory Usage | Ok | 17m 46s | 1/5 | 09/05/2019 23:38:57 | OK - 8364 / 15885 MB (52%) Free Memory, Used: 7521 MB, Shared: 158 MB, Buffers + Cached: 8036 MB |
| | Open Files | Ok | 14m 39s | 1/5 | 09/05/2019 23:37:04 | OK: 2720 open files (0% of max 1610050) |
| | Ping | Ok | 16m 40s | 1/5 | 09/05/2019 23:40:05 | [redacted] |
| | SSH Server | Ok | 15m 59s | 1/5 | 09/05/2019 23:40:45 | active |
| | Swap Usage | Ok | 17m 5s | 1/5 | 09/05/2019 23:39:40 | SWAP OK - 99% free (10101 MB out of 10254 MB) |
| | Total Processes | Ok | 17m 54s | 1/5 | 09/05/2019 23:38:49 | PROCS OK: 165 processes |
| | Users | Ok | 14m 47s | 1/5 | 09/05/2019 23:36:59 | USERS OK - 0 users currently logged in |
| | Yum Updates | Warning | 15m 1s | 3/5 | 09/05/2019 23:36:51 | YUM WARNING: O/S requires an update. |

Kuva 31. Esimerkki verkkopalvelimen monitoroinnista

Muita yksityiskohtaisempia tapoja kerätä palvelimesta dataa on monitoroimalla verkkopalvelua pyörittävää alustaa. Nagios XI tarjoaa valmiin liitännäisen Apache Tomcatin monitorointiin, jonka avulla voidaan seurata Apachen toimintaa, kuten esitetty kuvassa 32. Nagios tarjoaa myös keinon monitoroida tietokantoja, kuten esimerkiksi MySQL-tietokantaa, jonka avulla voidaan saada tietoa esimerkiksi epäonnistuneista tietokantakyselyistä.

| Host | Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|---|
| JMX | Heap Memory Usage | Ok | 17m 31s | 1/5 | 2017-07-27 10:37:47 | JMX OK - HeapMemoryUsage.used=192824184 |
| | Thread Count | Ok | 19s | 1/5 | 2017-07-27 10:39:59 | JMX OK - ThreadCount=75 |

Kuva 32. Esimerkki Apache Tomcatin JMX-monitoroinnista (Nagios Enterprises 2017e)

## 12.2 Moodlen verkkosivun monitorointi

Verkkosivuja voidaan monitoroida joko yksinkertaisilla menetelmillä tai skriptien avulla. Yksinkertaisia menetelmiä ovat esimerkiksi verkkosivujen latausnopeuden monitorointi, saatavuustiedot, sertifikaatin varmistus ja verkkosivun sisällön tarkistukset, kuten esitetty kuvassa 33. Yksinkertaiset menetelmät ovat helppo tapa antaa nopea kuva verkkosivun toimivuudesta. Ne eivät kuitenkaan suoranaisesti testaa verkkosivun toimivuutta käyttäjän näkökulmasta, sillä käyttäjät tekevät usein muutakin, kun vain avaavat verkkosivun. (Nagios Enterprises 2018f.)

| Host | Service | Status | Duration | Attempt | Last Check | Status Information |
|---|---|---|---|---|---|---|
| hhmoodle.haaga-helia.fi | DNS IP Match | Ok | 1h 13m 59s | 1/5 | 10/05/2019 00:35:44 | DNS OK: 0.008 seconds response time. hhmoodle.haaga-helia.fi returns 193.166.9.98 |
| | DNS Resolution | Ok | 1h 14m 24s | 1/5 | 10/05/2019 00:35:21 | DNS OK: 0.007 seconds response time. hhmoodle.haaga-helia.fi returns 193.166.9.98 |
| | HTTP | Ok | 1h 12m 23s | 1/5 | 10/05/2019 00:37:17 | HTTP OK: HTTP/1.1 200 OK - 27546 bytes in 0.215 second response time |
| | Ping | Ok | 1h 13m 56s | 1/5 | 10/05/2019 00:35:47 | OK - hhmoodle.haaga-helia.fi: rta 0.202ms, lost 0% |
| | SSL Certificate | Ok | 1h 13m 33s | 1/5 | 10/05/2019 00:36:07 | SSL OK - Certificate 'hhmoodle.haaga-helia.fi' will expire on 2021-01-05 14:00 +0200/EET. HTTP OK: HTTP/1.1 303 See Other - 981 bytes in 0.054 second response time |
| | Web Page Content | Ok | 1h 16m 36s | 1/5 | 10/05/2019 00:38:13 | HTTP OK: HTTP/1.1 200 OK - 27546 bytes in 0.291 second response time |

Kuva 33. Esimerkki verkkosivun monitoroinnista

Skriptien avulla monitoroidessa verkkopalvelussa suoritetaan synteettisiä testejä, joiden tarkoituksena on simuloida loppukäyttäjän verkkopalvelun käyttöä lähettämällä HTTP:n GET ja POST -kyselyitä. Kyselyiden avulla voidaan autentikoida esimerkiksi testikäyttäjän kirjautuminen verkkopalveluun tai tehdä tietokantakyselyitä hakemalla esimerkiksi Moodlen tapauksessa kursseja, kurssien sisältöä tai osallistua foorumikeskusteluihin. Tällaisilla testeillä voidaan testata Moodlen eri osien toimivuutta, joita ei muuten pystytä yksinkertaisilla testeillä toteuttamaan. Nagios tarjoaa synteettiseen testaukseen muun muassa pluginia, joka testaa verkkosivujen toimintaa WebInject HTTP-testaustyökalun avulla. (Nagios Enterprises 2018f.)

Synteettisten testausten kehitys ja käyttöönotto monitoroinnissa on suositeltava toteuttaa verkkopalvelun pääkäyttäjien ja asiantuntijoiden toimesta, sillä he pystyvät määrittämään tärkeimmät ja kriittisimmät ominaisuudet, joita synteettisillä testeillä halutaan testata.

# 13 Hälytykset

Nagioksessa voidaan määritellä tiedotteita, varoituksia ja hälytyksiä. Näitä viestejä voidaan lähettää oletusarvoisesti joko sähköpostitse tai tekstiviestien muodossa. Hälytyksiä voidaan halutessa lähettää myös muilla tavoin, kuten hyödyntäen puhelinsoittoja *text-to-speech* -muodossa, mutta nämä erikoisvalmisteiset tavat eivät ole valmiina Nagioksessa, vaan niitä voidaan määritellä hyödyntämällä Nagiokseen erikseen räätälöityjä komentoja, joita muut ohjelmat voivat hyödyntää.

Nagioksessa voidaan monitoroida kohteita joko aktiivisesti tai passiivisesti. Aktiivisessa monitoroinnissa Nagios oletusarvoisesti lähettää monitorointikyselyitä valvottavaan kohteeseen viiden minuutin välein. Tätä aikaväliä voidaan kuitenkin muuttaa, sillä kaikkia kohteita ei tarvitse monitoroida niin useasti, ja toisaalta osaa kohteista saatetaan haluta monitoroida tätäkin useammin. Passiivisessa monitoroinnissa Nagioksen agentit lähettävät tietoa itsestään Nagios-palvelimelle jonkin tapahtuman sattuessa, esimerkiksi SNMP:n avulla, mutta dataa voidaan valita lähetettäväksi Nagios-palvelimelle myös tietyin väliajoin.

Mikäli Nagios huomaa ongelman monitoroitavassa kohteessa, se lähettää varmistuskyselyitä kohteeseen oletusarvoisesti minuutin välein, tarkistaakseen onko ongelma vain hetkellinen. Monitorointiväliä nopeutetaan myös sen takia, että saadaan nopeasti tieto siitä, korjaantuuko ongelma mahdollisten automaattisten korjaustoimintojen myötä. Nagios lähettää viestin tapahtuneesta eteenpäin vasta tarkistuskyselyiden jälkeen.

Hälytyksiä voidaan lähettää Nagioksessa joko heti ongelman tapahtuessa, jonkin tietyn ajan jälkeen tai ei ollenkaan. Nagioksessa voidaan myös määritellä aikaväli, jonka jälkeen hälytys lähetetään uudelleen, jos ongelma ei ole korjaantunut tai muuttunut. Suuressa osaa virhetilanteista hälytyksiä ei ole aiheellista lähettää uudelleen ensimmäisen hälytyksen jälkeen, mutta kriittisissä tapauksissa tulisi aina määritellä hälytykselle uudelleenlähetyksen aikaväli, jotta ongelman hälytysviesti ei missään nimessä katoa muun informaatiovirran joukkoon tai jää muuten huomaamatta. Erittäin kriittisissä kohteissa tulisikin määritellä sähköpostin lisäksi myös esimerkiksi tekstiviestien lähetys.

Hälytykset määritellään Nagioksessa kahdella tavalla, Nagioksen objektien *hosts* ja *services* mukaan. Näille kahdelle objektille on omat hälytyksensä, jotka voidaan jakaa eri luokkiin niiden vakavuusasteen mukaisesti.

Monitoroitavat kohteet, eli *hostit*, voidaan jakaa kolmeen ryhmään kohteen tilan perusteella. *Up*, eli monitoroitava kohde on ylhäällä ja toimintakuntoisena. Toinen tila on *Unreachable,* joka tarkoittaa, että kyseisen kohteen *parentiin*, eli hierarkkisesti sen yllä olevaan kohteeseen, ei saada yhteyttä, jolloin myöskään kyseiseen hostiin ei saada yhteyttä sen riippuvuuden takia. Kolmas tila on *Down*, joka tarkoittaa, että kohde on niin sanotusti alhaalla, eli sen tiedetään olevan pois käytöstä, tai siihen ei saada yhteyttä. (Nagios Enterprises 2018j.)

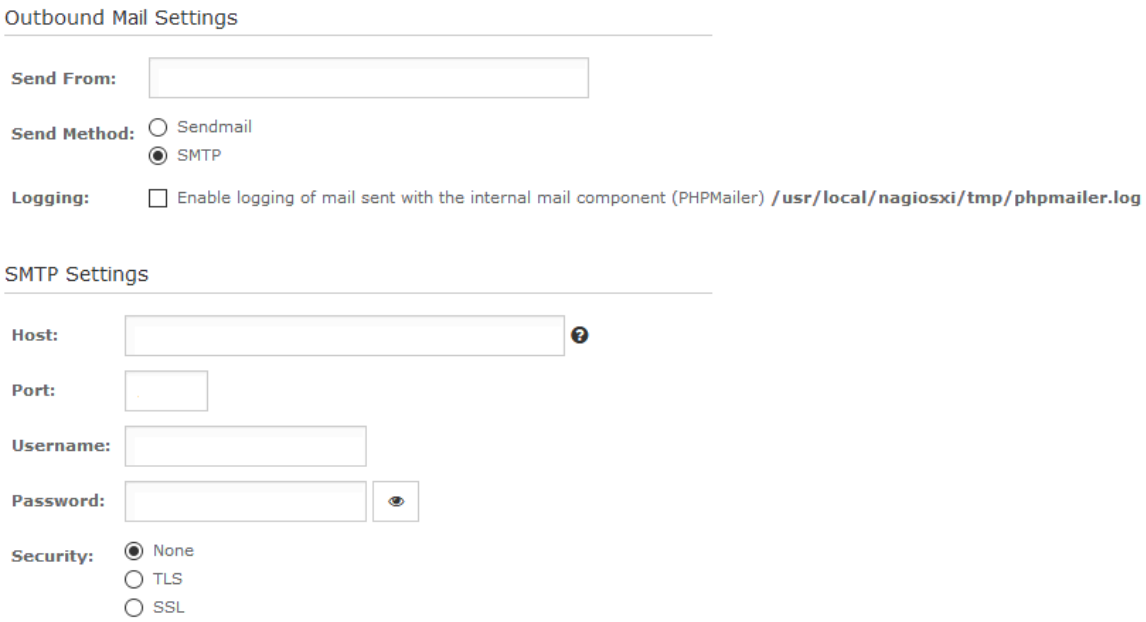
Nagioksen monitoroitavat palvelut, eli *servicet*, voidaan jakaa neljään ryhmään niiden tilan perusteella. *OK*, eli tilanne on normaali, jolloin palvelun tarkastus menee onnistuneesti läpi palauttaen tarkastettavan arvon oikein, jonka lisäksi arvo on sille määriteltyjen raja-arvojen sisällä. Toinen tilannetaso on *Warning*, joka tarkoittaa sitä, että palvelun tarkastus menee onnistuneesti läpi, mutta sen palauttama arvo ei ole raja-arvojen sisällä, mutta ei kuitenkaan vielä kriittisellä tasolla. Tätä tilannetasoa käytetään varoittamaan mahdollisista syntymässä olevista ongelmista. Kolmas taso palvelun tarkastukselle on *Unknown*, joka tarkoittaa sitä, että palvelun tarkastus ei toimi oikein. Esimerkiksi palvelun tarkastus on määritelty väärin, eikä sitä voida toteuttaa. Neljäs ja huolestuttavin tilanne monitoroitaville palveluille on *Critical*, joka tarkoittaa kriittistä tilannetta palvelun monitoroinnissa, eli käytännössä tarkastus on palauttanut jonkin hälyttävän raja-arvon, joka pitäisi ottaa heti huomioon, sillä ongelmalla voi olla katastrofaalinen vaikutus palvelun kohteeseen ja sen toimintaan. Esimerkkinä kriittisestä tilanteesta voi olla tilanne, jossa palvelimen muistinkäyttö on 100% tai jokin palvelun toimivuuden kannalta kriittinen prosessi on kaatunut. (Nagios Enterprises 2018k.)

### 13.1 Sähköposti

Kuten useimmilla monitorointiratkaisuilla, Nagios XI:llä voidaan lähettää hälytyksiä sähköpostin avulla. Toimeksiantajan toive oli lisätä monitorointiratkaisuun sähköpostin avulla lähetettävät hälytykset, jotka voidaan konfiguroida tilanne- tai ryhmäkohtaisesti monitoroitaviin kohteisiin. Käytännössä tämä tarkoittaa sitä, että jos jostain kohteesta saatavat monitoroinnin arvot vastaavat hälytykseen vaadittavia kriteereitä, hälytys voidaan lähettää sähköpostitse joko siitä vastaavalle taholle, jollekin sähköpostijakelulistalle tai vaihtoehtoisesti suoraan Haaga-Helian käytössä olevaan toiminnanohjausjärjestelmään. (Nagios Enterprises 2018g.)

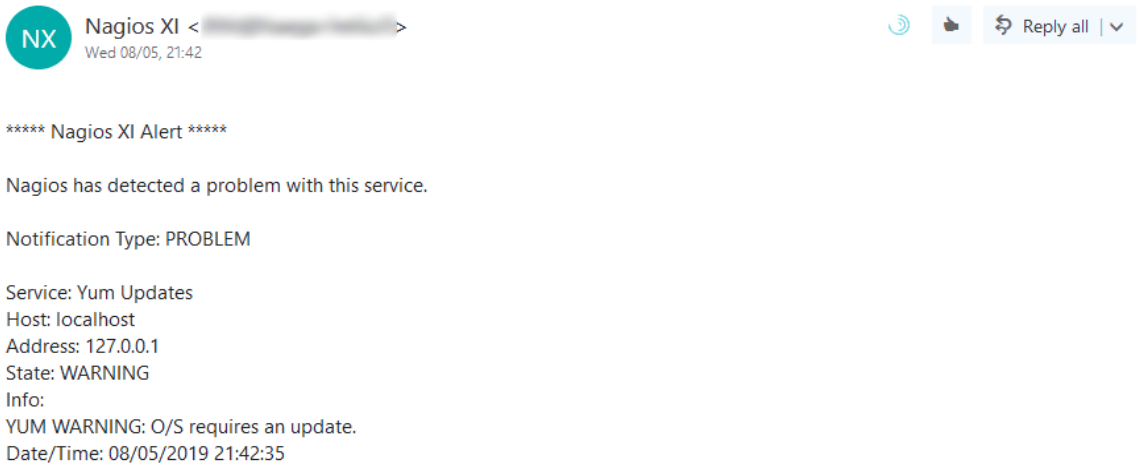
Sähköpostin ottaminen käyttöön Nagios XI:ssä on suoraviivainen prosessi Nagios XI:n web-käyttöliittymän avulla, jonka kautta voidaan konfiguroida sähköpostipalvelimelle oi-

keat arvot kuvan 34 mukaisella tavalla. On kuitenkin huomioitava, että sähköpostipalvelimelle on määritettävä oikeudet myös palvelimen osoitteesta tulevan postin osalta, jotta sähköpostipalvelin voi lähettää postin edelleen oikealle taholle. (Nagios Enterprises 2018g.)



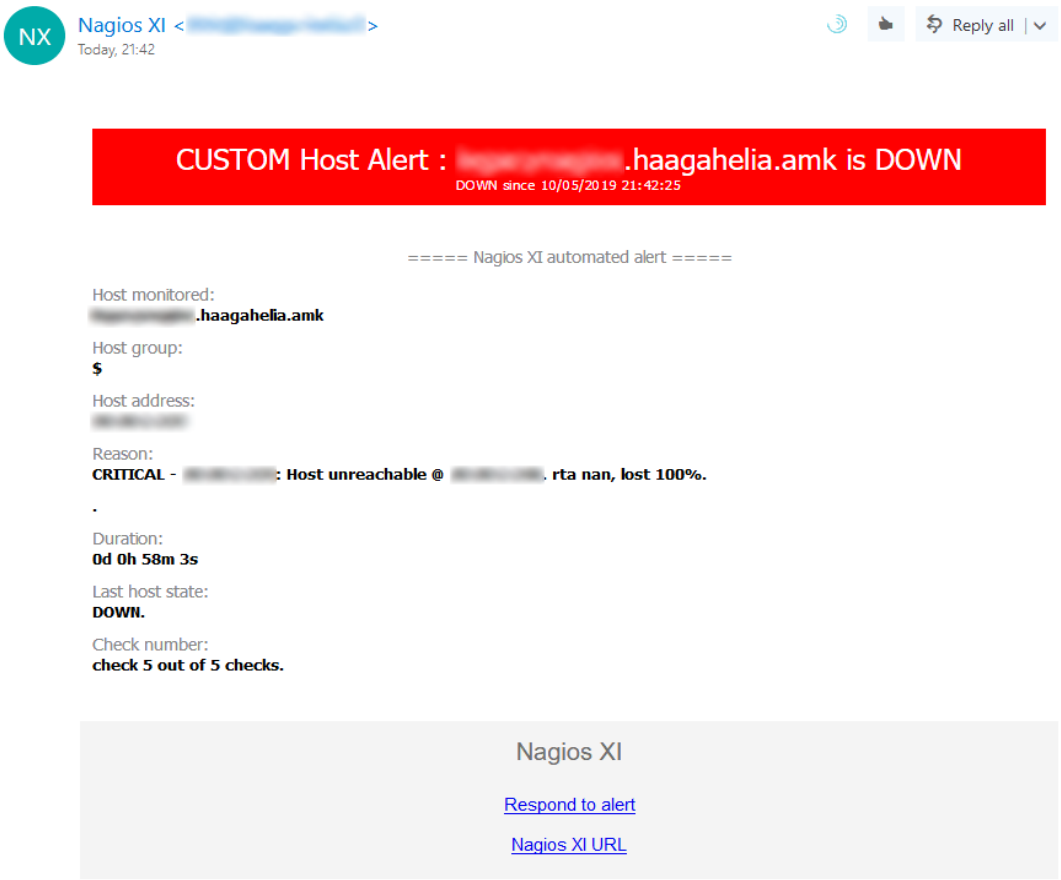

Kuva 34. Esimerkki lähtevän sähköpostin asetuksista Nagios XI:n web-käyttöliittymässä

Nagios XI:n tarjoamat oletusarvoiset viestit ovat tekstipohjaisia, kuten esitetty kuvassa 35, joka tekee niistä vaikealukuisia ja hitaasti ymmärrettäviä. Oletusarvoisesta viestistä ei myöskään saa irti kaikkea hyödynnettävää tietoa, kuten esimerkiksi monitoroitavan kohteen ryhmää tai kuinka pitkään ongelma on ollut olemassa.




Kuva 35. Esimerkki oletusarvoisesta viestistä Outlookissa

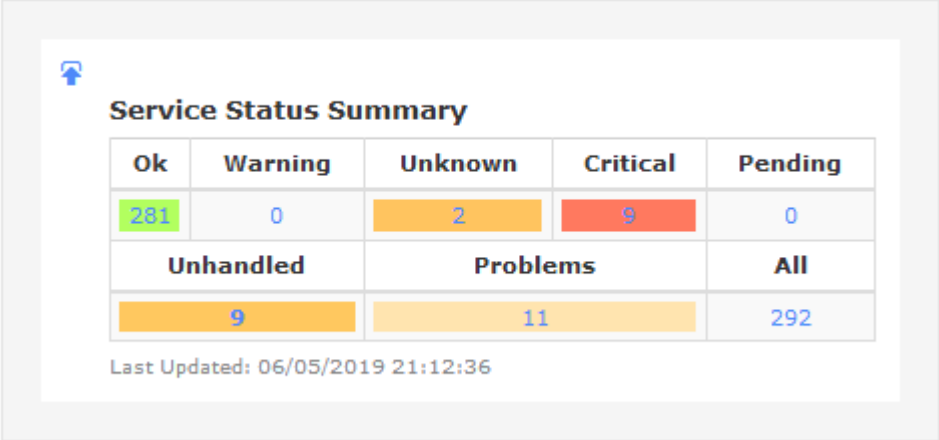
Nagioksen hälytykset tukevat HTML-koodikieltä, joten hälytyksiä voidaan muokata. Tätä ominaisuutta hyödyntäen muokkasimme hälytysviestit helpommin ja nopeammin ymmärrettäviksi. Nagios tukee erilaisia muuttujia, joiden avulla se täyttää hälytysviestin tietoja. Hyödynsimme näitä muuttujia viestin sisällä olevassa otsikossa määrittelemällä eri tilannetasoille värikoodit, jotka edesauttavat viestin nopeaa ymmärrettävyyttä. Esimerkki Nagioksen lähettämästä sähköpostihälytyksestä, joka hyödyntää luomaamme HTML-pohjaa, on esitetty kuvassa 36.




Kuva 36. Esimerkki uudistetusta HTML-viestistä Outlookissa

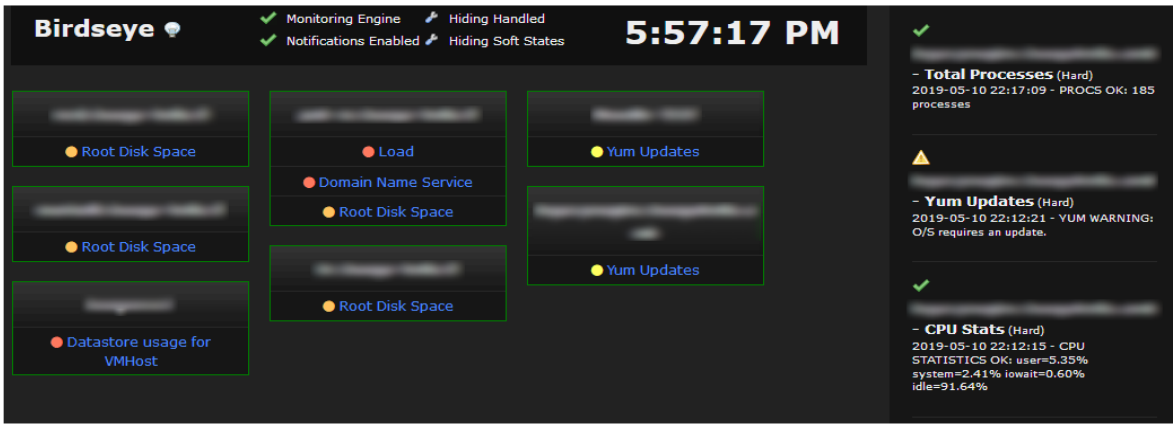
## 13.2 Web-käyttöliittymä hälytyksien tukena

Hälytyksiä ja varoituksia voidaan hyödyntää myös passiivisemmassa muodossa Nagioksen web-käyttöliittymässä, jossa palvelimien tai palveluiden tilasta voidaan ilmoittaa palvelun eri näkymissä, kuten etusivulla näkyvien reaaliaikaisten tilannetietojen yhteenvedon muodossa, joka on nähtävillä kuvassa 37.




Kuva 37. Esimerkki Nagios XI:n web-käyttöliittymän reaaliaikaisesta yhteenvedosta monitoroitavien service-kohteiden osalta

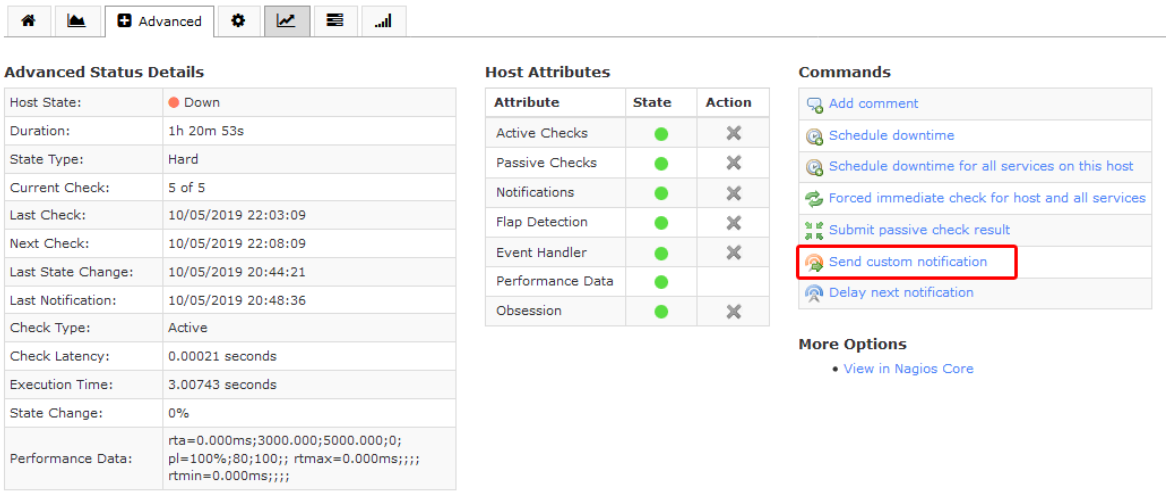
Nagios tarjoaa myös muita web-käyttöliittymän näkymiä, joita esitellään tarkemmin luvussa 14. Yhtä esitellyistä näkymistä, Birdseye-näkymää, voitaisiin hyödyntää helposti ja kätevästi esimerkiksi Haaga-Helian helpdeskin käytössä. Näkymä on esitetty kuvassa 38 ja sen avulla voidaan näyttää yhteenvetona tiedossa olevat ongelmat palvelinten ja palveluiden osalta. Birdseye-näkymä voitaisiin ottaa käyttöön erilliselle näytölle helpdeskin toimistossa, jolloin työntekijät saisivat nopean tavan tarkistaa, onko järjestelmissä havaittuja ongelmatilanteita. Ongelmatilanteista voidaan tämän myötä tiedottaa suoraan loppukäyttäjille ja tällöin loppukäyttäjien mahdollisiin ongelmiin voitaisiin tarjota heti ajankohtainen selitys.




Kuva 38. Esimerkki Birdseye-näkymästä

## 13.3 Hälytysten testaus

Hälytyksiä tulisi aina testata ennen niiden laajempaa käyttöönottoa, jotta voidaan välttyä mahdollisilta virheellisiltä hälytyksiltä väärille tahoille. Hälytyksiä pystytään testaamaan esimerkiksi web-käyttöliittymän avulla, lähettämällä jostakin monitoroitavasta kohteesta räätälöity tiedote, joka vastaa hälytystä ja käyttää samoja asetuksia kuin normaalit hälytykset. Kyseisiä testihälytyksiä tai tiedotteita voidaan lähettää valitsemalla ensin jokin monitoroitava kohde, jonka jälkeen valitaan ”*send custom notification”*, kuten osoitettu kuvassa 39. Valinnan jälkeen voidaan määritellä lisäarvoja tiedotteelle, kuten lähetetäänkö se pakotetusti riippumatta siitä, onko käyttäjä ottanut hälytykset pois käytöstä oman tunnuksensa asetuksista tai lisätäänkö viestiin erillinen kommentti, joka lisätään viestin sisältöön.

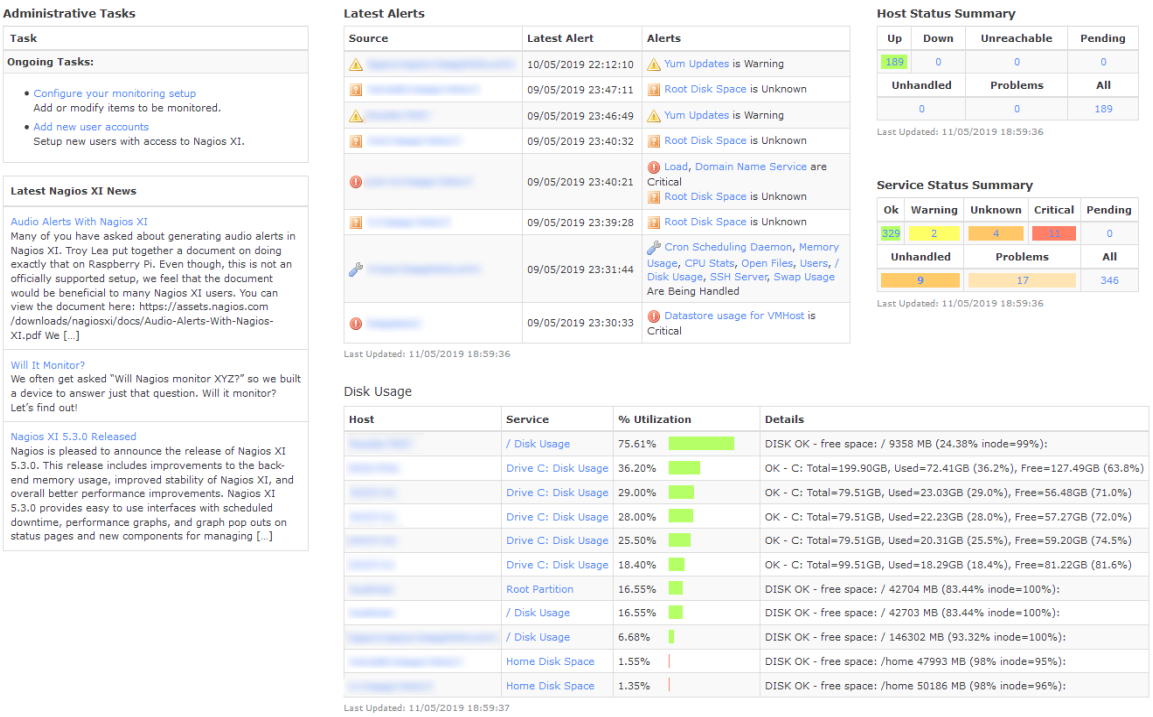



Kuva 39. Hälytysten testauksessa käytetty räätälöidyn tiedotteen lähetys

# 14 Nagios XI:n web-käyttöliittymän näkymät

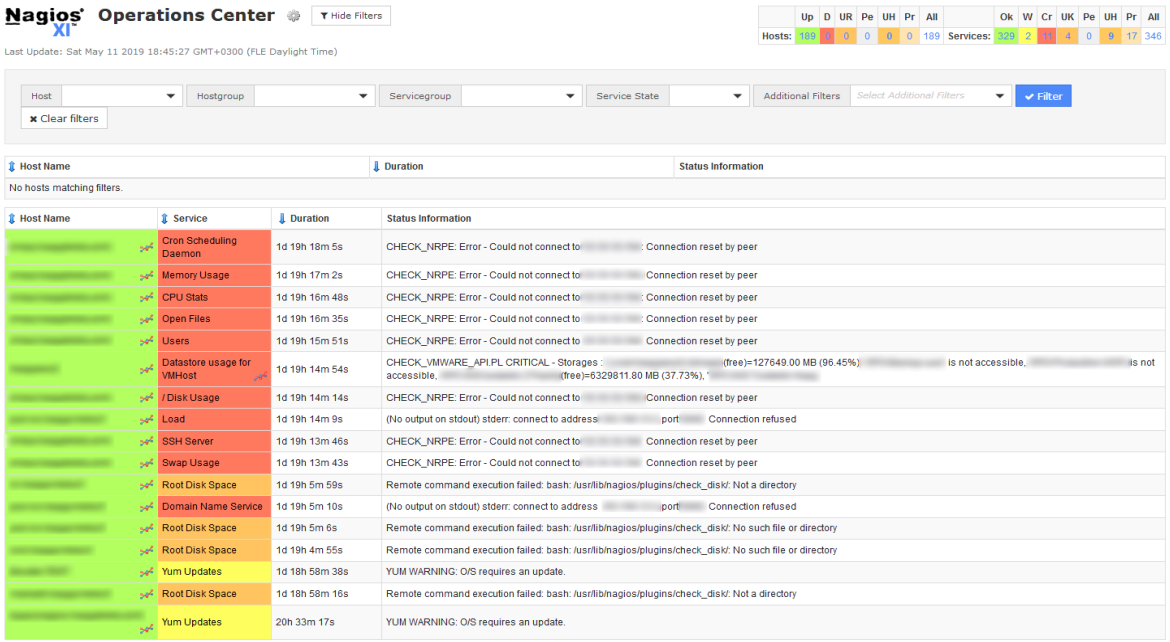
Tässä luvussa esittelemme muutamia Nagioksen web-käyttöliittymän näkymistä, joiden ajattelemme olevan Haaga-Helian kannalta hyödyllisiä. Näkymiä on mahdollista räätälöidä käyttäjäkohtaisesti, joten esimerkiksi helpdeskin tasolla voidaan valita näkyville erilaista tietoa kuin mistä esimerkiksi järjestelmän ylläpitäjä voisi olla kiinnostunut.

Kuvassa 40 on esimerkki käyttäjän henkilökohtaisesta näkymästä, joka on räätälöity näyttämään tietoja viimeisimmistä hälytyksistä, levynkäytöstä, yleiskatsauksen monitoroinnin piiriin kuuluvista päätelaitteista ja palveluista sekä tietoa niihin mahdollisesti liittyvistä ilmoituksista. Näkymän vasemmassa reunassa on nähtävillä Nagioksen konfigurointiin liittyviä tehtäviä. Tämä kyseinen näkymä on esillä Nagioksen etusivulla, eli palvelun käyttäjä näkee sen kirjautuessaan Nagioksen web-käyttöliittymään.




Kuva 40. Esimerkki yksittäisen käyttäjän räätälöidystä Nagios XI:n kotisivusta

Nagios Operations Center -näkymän kautta voidaan tarkastella NOC-tyylisessä (Network Operations Center) näkymässä tietoja viimeimmistä ratkaisemattomista ongelmista. Sivu päivittyy 30 sekunnin välein. Näkymä on suunniteltu erityisesti sellaiseksi, että sitä voidaan esittää esimerkiksi työhuoneessa sijaitsevalla infotaululla. Näkymä ei sisällä linkkejä, vaan se on tarkoitettu ensisijaisesti osoittamaan olemassa olevat häiriöt nopealla vilkaisulla. Esimerkki NOC-näkymästä esitetään kuvassa 41.

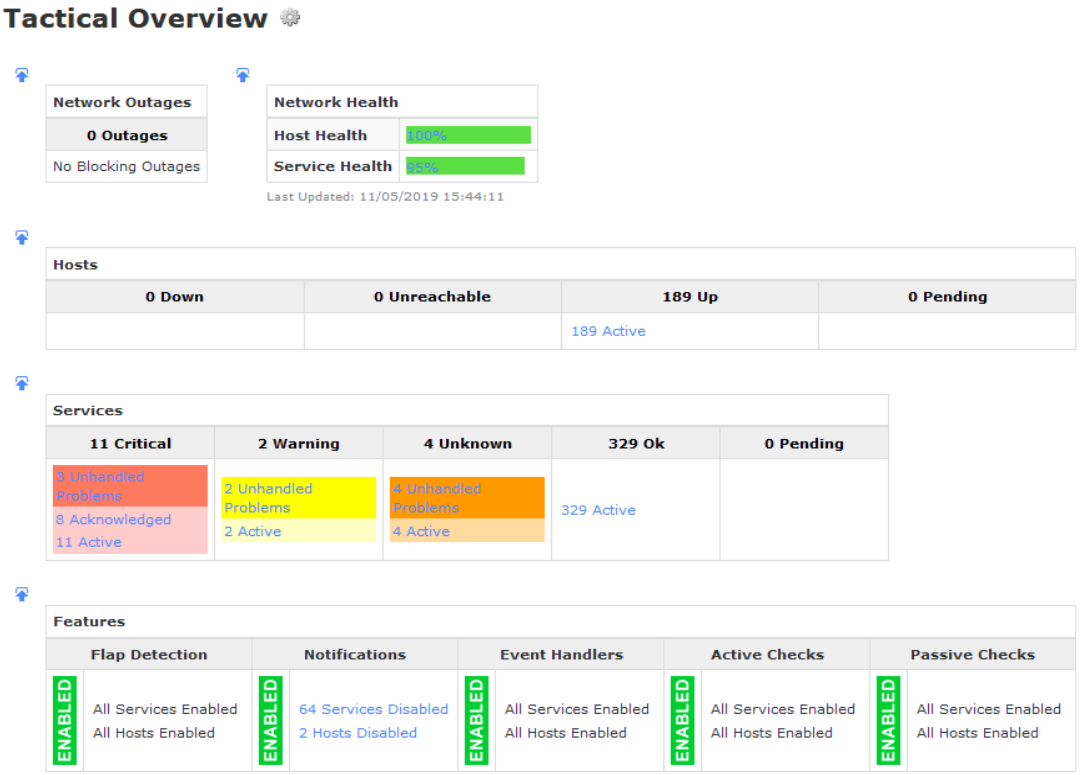



Kuva 41. Esimerkki Nagioksen Operations Center -näkymästä

Nagioksen niin sanotussa taktisessa näkymässä voidaan tarkastella tiivistetysti kaikkien monitoroitavien palveluiden tilaa. Taktinen näkymä on esitetty kuvassa 42. Näkymästä nähdään nopeasti, mikäli jossain palvelimessa tai palvelussa on häiriö. Häiriöilmoituksia klikkaamalla päästään erilliseen näkymään, jonka kautta voidaan tarkastella yksityiskohtaisempia tietoja kyseisistä häiriöilmoituksista.

Taktinen näkymä voi olla hyödyllinen esimerkiksi tietohallintoyksikön esimiestasolle, jolloin heille välittyy reaaliaikainen tilannekuva olemassa olevista ongelmatilanteista ilman, että joku erikseen informoi heitä aiheesta. Tilannekuvan avulla he pystyvät johtamaan tiimiään allokoimalla resursseja oikeiden ongelmien pariin. Taktinen näkymä antaa myös mahdollisuuden siihen, ettei esimiesten ja ongelmaa hoitavan tahon välillä tarvitse olla jatkuvaa väliaikaraportointia, joka mahdollisesti hidastaisi ongelman ratkaisua. Esimies näkee ongelman korjaantuneen taktisesta näkymästä suoraan. Ongelman syy ja selvitysketju voidaan nostaa myöhemmin esille esimerkiksi Nagioksesta saatavien raporttien avulla.

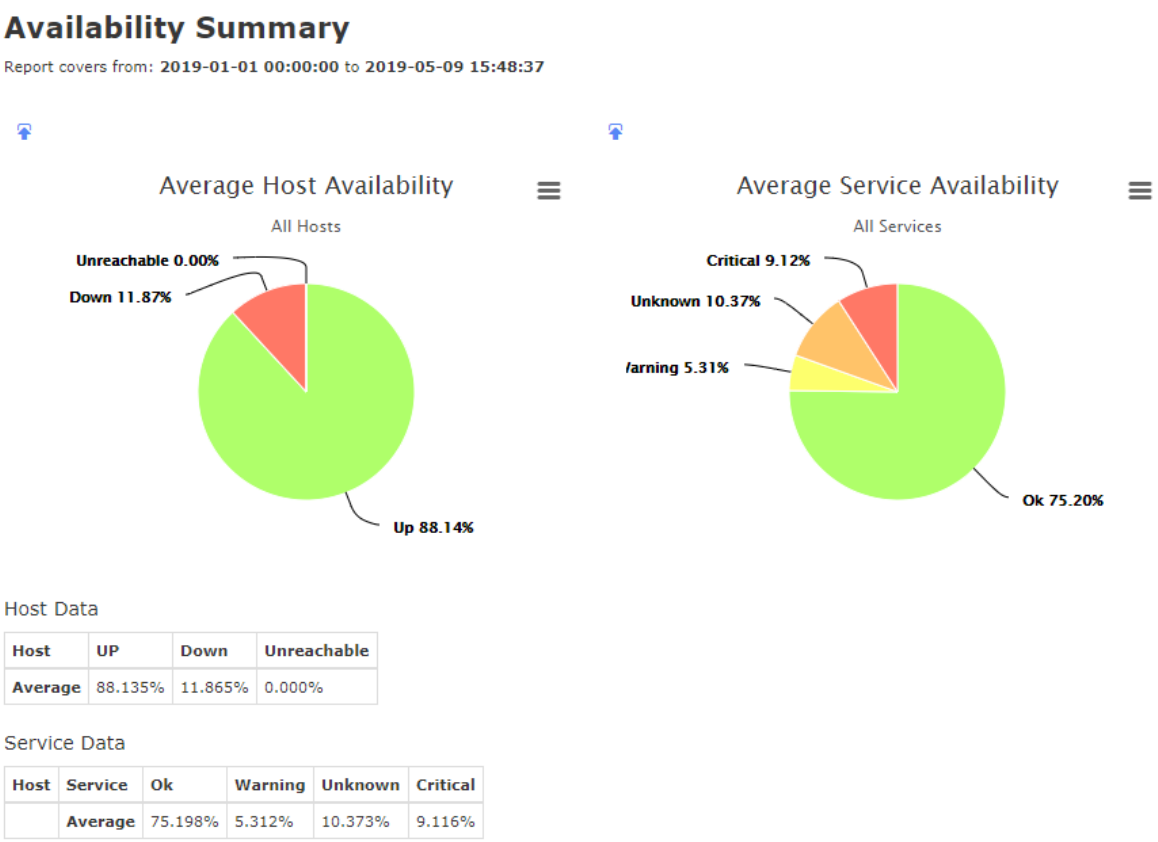
Palveluiden tilan lisäksi näkymästä voidaan tarkastella muun muassa monitoroinnin ominaisuuksia, ilmoituksia sekä aktiivisia ja passiivisia tarkistuksia. Näitä ominaisuuksia voidaan joko ottaa käyttöön tai poistaa käytöstä näkymän kautta klikkaamalla ”*enabled*” tai ”*disabled*” -linkkejä ominaisuuksien vieressä.




Kuva 42. Esimerkki Nagioksen Tactical Overview -näkymästä

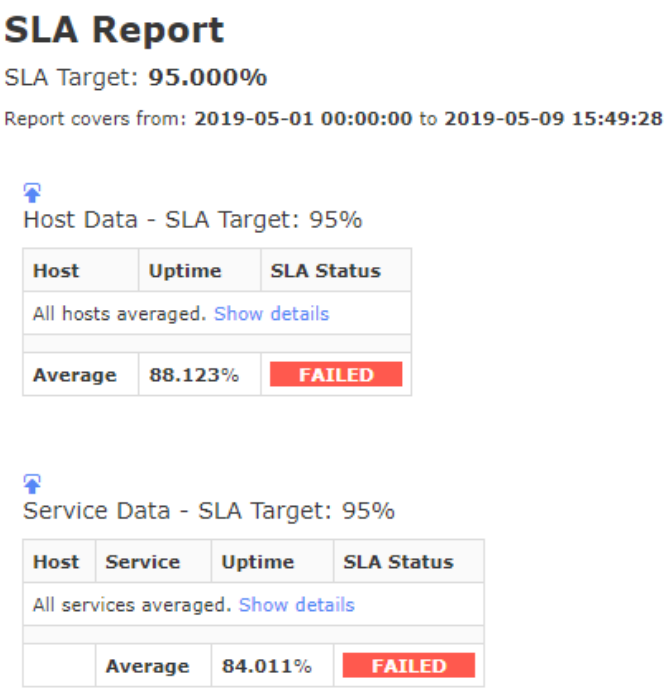
# 15 Nagios XI:n raportointimahdollisuudet

Nagios XI tarjoaa mahdollisuuden luoda raportteja monitoroitavista kohteista. Raporttien avulla voidaan esittää dataa ja mittareita monitoroitavista kohteista ja monitoroitavan verkkoinfrastruktuurin kunnosta yleisesti. Yksi raportoinnin esimerkkinäkymä on esitetty kuvassa 43. Raportteja voidaan hyödyntää suoraan erilaisissa mittareissa, joilla mitataan esimerkiksi palvelimien saatavuutta ja joita voidaan hyödyntää vertailuissa vuositasolla, kun halutaan tukea ja määritellä esimerkiksi palveluiden saatavuuden parantumista.




Kuva 43. Esimerkkiyhteenveto monitoroitavista laitteista (host) ja niiden monitoroitavista kohteista (service)

Nagioksen tuottamia raportteja voidaan ajastaa, jolloin haluttuja raportteja voidaan tuottaa automatisoidusti tiettyinä väliaikoina, kuten kerran kuukaudessa. Automatisoitujen raporttien avulla voidaan tarkastella esimerkiksi saatavuusastetta ja sitä, onko palvelukuvauksessa määritelty saatavuusaste toteutunut, kuten esitetty esimerkkikuvassa 44. Haaga-Helian tuottamat palvelut hyödyttävät ensisijaisesti organisaatiota itseään, jolloin niin sanottuja palvelusopimuksessa määriteltyjä saatavuusasteita ei todennäköisesti sanktioida samalla tapaa kuin joissain muille tahoille palveluita tuottavissa organisaatioissa saatettaisiin tehdä.



SLA Report

SLA Target: **95.000%**

Report covers from: **2019-05-01 00:00:00** to **2019-05-09 15:49:28**

Host Data - SLA Target: 95%

| Host | Uptime | SLA Status |
|---|---|---|
| All hosts averaged. Show details | | |
| **Average** | **88.123%** | **FAILED** |

Service Data - SLA Target: 95%

| Host | Service | Uptime | SLA Status |
|---|---|---|---|
| All services averaged. Show details | | | |
| | **Average** | **84.011%** | **FAILED** |

Kuva 44. Esimerkki yksinkertaisesta service level agreement -raportista

Kuvassa 45 on esitetty raportti viimeisimmistä hälytyksistä ja sen tietoja voidaan tarkastella monitoroinnin kehittämisen kannalta. Voidaan esimerkiksi pohtia, ovatko kaikki monitoroitavat kohteet ja hälytykset tarpeellisia tai tulisiko jotain jättää pois. Vaihtoehtoisesti voidaan harkita tulisiko jotain hälytystä tarkentaa tai kehittää, jolloin siitä voisi olla enemmän hyötyä. Toinen konkreettinen hyöty ilmenee, jos jokin palvelu tuottaa jatkuvia häiriöilmoituksia, vaikka hälytyksen raja-arvot ovat kunnossa. Tällöin voi olla syytä miettiä, mitä toimenpiteitä palvelun varmistamiseksi voidaan tehdä.

**Latest Alerts**

| Source | Latest Alert | Alerts |
|---|---|---|
| linux-snmp1.nagios.local | 2019-05-09 15:51:55 | SSH, MySQL, Apache, Apache 404 Errors are Critical<br>Bandwidth Spike, Linux Failed Logins are Warning |
| Network-Analyzer2.nagios.local | 2019-05-09 15:51:55 | Bandwidth Spike is Critical<br>Yum Updates, Total Processes are Warning |
| rhel1.nagios.local | 2019-05-09 15:51:34 | Test Process, Sendmail, Dovecot Mail Server are Critical<br>Total Processes, Youtube Usage are Warning<br>Yum Updates is Unknown |
| windowserver3.nagios.local | 2019-05-09 15:51:34 | Explorer is Critical<br>Facebook Usage, Port 80 Bandwidth, Youtube Usage are Warning |
| centos5.nagios.local | 2019-05-09 15:51:34 | Total Processes, Youtube Usage are Warning<br>Yum Updates is Unknown |
| centos2.nagios.local | 2019-05-09 15:51:23 | Yum Updates, Total Processes, Failed SSH Logins are Warning |
| Log-Server2.nagios.local | 2019-05-09 15:51:13 | Yum Updates, Total Processes, Youtube Usage are Warning |
| www.nagios.com | 2019-05-09 15:50:55 | DNS IP Match is Critical |
| europa.nagios.local | 2019-05-09 15:50:24 | Host Unreachable: CRITICAL - 192.168.4.54: rta nan, lost 100% |
| gateway.nagios.local | 2019-05-09 15:49:24 | ethernet0/0 Status is Critical<br>ethernet0/4 Status, ethernet0/5 Status, bgroup0 Status, bgroup1 Status, |

Kuva 45. Esimerkkiyhteenveto viimeisimmistä hälytyksistä

Ajastettuja raportteja voidaan määritellä myös automaattisesti lähetettäväksi sähköpostitse, esimerkiksi niistä huolehtiville tahoille tai niitä hyödyntäville johtoryhmän tahoille.

# 16 Tulokset

Tässä luvussa käymme läpi opinnäytetyöprojektin aikana läpikäytyjä työvaiheita sekä toimeksiantajan näkökulmasta konkreettisia tuloksia monitoroinnin osalta. Lisäksi perustelemme taustoja joidenkin valintojen takana.

## 16.1 Matka tulokseen

Opinnäytetyön aikana selvitimme monitoroinnissa yleisimmin käytettyjä ohjelmistoja ja valitsimme niiden pohjalta Haaga-Helian näkökulmasta kiinnostavimmat vaihtoehdot. Ohjelmistot valittiin sillä perusteella, että ne esiintyivät monitoroinnin ympärillä käytävissä keskusteluissa internetin asiantuntijapalstoilla tai aiheeseen liittyvien lehtien tai muiden julkaisujen yhteydessä.

Monitorointia lähestyttiin pitkälti siitä näkökulmasta, että tavoitteena on lopulta yhdistää monitorointi osaksi Haaga-Heliassa jo olemassa olevia prosesseja, erityisesti tietohallintoyksikön osalta, jolloin monitoroinnin hyödyt eivät jäisi vain yksittäisen ylläpitäjän tasolle, vaan valvonnasta hyötyisi mahdollisimman moni taho. Samalla varmistettiin myös toimeksiantajan toive siitä, että monitorointia toteutettaisiin yhdellä yksittäisellä alustalla. Tätä ajatusta konkretisoidaksemme pohdimme it-palveluita tuottavien yritysten mahdollisia prosesseja ja kuvasimme esimerkin tavoitetilasta, jossa monitorointi on onnistuneesti yhdistetty osaksi muuta toimintaa. Prosessikuvaus toteutettiin BPMN-mallia hyödyntäen, jonka lisäksi esittelimme Haaga-Heliassa monitorointiin liittyviä tahoja sidosryhmäteorian avulla.

Monitoroinnin kannalta oli oleellista käsitellä myös monitoroinnissa hyödynnettäviä protokollia, sillä protokollan valinta vaikuttaa siihen, minkälaista dataa järjestelmistä voidaan kerätä. Valitsimme työhömme esiteltäviksi protokolliksi SNMP- ja WMI-protokollat, sillä ne olivat vertailemissamme ohjelmistoissa laajimmin esillä ja lisäksi yleisesti hyvin tunnettuja.

Monitorointityökaluihin tutustumisen pohjalta valitsimme seitsemän eri ohjelmistoa, joiden uskoimme olevan potentiaalisia vaihtoehtoja lopullisen monitorointiratkaisun toteutukseen. Vertailusta jätettiin pois sellaiset ratkaisut, joita ei voitu käyttöönottaa paikallisesti Haaga-Helian palvelinympäristössä. Tutustuimme tarkemmin näihin ohjelmistoihin sekä niiden tarjoamiin ominaisuuksiin ja työkaluihin. Selvitystyö toteutettiin ensisijaisesti valmistajien julkaisemien dokumentaatioiden pohjalta, mutta hyödyntäen lisäksi myös aiheeseen liittyvää kirjallisuutta. Kirjallisuuslähteet keskittyivät yleensä tietyn ohjelmiston asennukseen ja konfiguraatioon, ja niistä saatu hyöty keskittyi projektin suunnitteluvaiheessa pääasiassa

ominaisuuksien kartoitukseen. Kirjallisuuslähteillä oli selkeä rooli myös monitoroinnin tarjoaman potentiaalin ymmärryksessä sekä hyvien monitorointimenetelmien esittelyssä.

Toimeksiantaja oli etukäteen esittänyt tiettyjä toiveita monitorointiratkaisun lopputulokselle, mutta varsinaisia tarkkoja teknisiä kriteereitä ei ollut etukäteen määritelty, vaan arvioimme ratkaisujen soveltuvuutta Haaga-Helian palvelinympäristöön itsenäisesti. Vertailua tehdessämme totesimme kuitenkin nopeasti, että valtaosa vertailuun valituista ohjelmista kattaa Haaga-Helian ympäristön asettamat kriteerit, joten lopullinen valinta tehtiin ensisijaisesti sillä perusteella, että ohjelmisto olisi helppo ottaa käyttöön ja sen jatkokehitys olisi realistista ilman, että työntekijöiden pitäisi käyttää ohjelmistoon perehtymiseen mittavaa määrää työtunteja.

### 16.2 Tulos ja tuloksen hyöty toimeksiantajalle

Toimeksiantajan kanssa käytiin keskustelua vertailun kohteena olleista ohjelmistoista ja työn lopulliseen tekniseen osioon päädyttiin valitsemaan Nagios XI -ohjelmisto. Nagios XI asennettiin erilliselle palvelimelle, joka toimii Haaga-Helian verkossa. Valintaa perusteltiin sillä, ettei monitorointia kannattaisi toteuttaa toisten palveluiden kanssa samalla palvelimella, sillä nämä eri palvelut saattaisivat häiritä toisiaan. Lisäksi erilliselle palvelimelle asennus mahdollistaa sen, ettei sen tarvitse olla riippuvainen Haaga-Helian verkon toimivuudesta.

Monitoroinnin saavutettavuuden ja hallittavuuden parantamiseksi lisäsimme Nagiokseen integraation Haaga-Helian aktiivihakemistoon. Aktiivihakemiston integraation myötä Nagiokseen voidaan jakaa käyttöoikeuksia toimialueen käyttäjäryhmien perusteella, joka puolestaan parantaa käyttöoikeuksien keskitettyä hallittavuutta. Integraation ansiosta kaikki ne käyttäjät, jotka hyödyntävät monitorointia työssään, voivat kirjautua Nagioksen web-käyttöliittymään käyttäen omia jo olemassa olevia organisaatiotunnuksiaan.

Projektin pääfokus oli Moodle-testipalvelimen monitoroinnissa, mutta konfiguroimme monitoroinnin samalla myös varsinaiselle Nagios-palvelimelle. Nagios-palvelimen monitorointi on perusteltua erityisesti laitteen komponenttien osalta, jotta monitoroinnin jatkuvuus voidaan turvata myös muiden, toiminnan kannalta kriittisten, palveluiden osalta. Katoimme dokumentaatiossa monitorointiagenttien asennuksen sekä Linux- että Windows-ympäristöihin, joiden lisäksi käsittelimme monitorointia myös VMwaren virtualisointialustojen, kytkinten ja IoT-laitteiden näkökulmasta, sillä ne olivat kaikki sellaisia, joita Haaga-Helia hyödyntää ympäristössään. Lisäksi nostimme Windows-työasemien ja tulostinten monitoroinnin konfiguroinnin mahdollisuuden esille dokumentaatiossa.

Moodle-testipalvelimen osalta luotiin konfiguraatio, joka monitoroi sekä palvelinta itseään, että siinä toimivaa palvelua. Palvelimen osalta monitorointiin komponentteja, kuten levytilaa, prosessorin- ja muistinkäyttöä, saavutettavuutta, prosessien ja käyttäjien määrää, kuormaa, SSH-palvelun tavoitettavuutta sekä palvelimen päivitysten tilaa. Varsinaisen Moodle-palvelun toimintaa varmistimme testaamalla Moodlen kotisivua. Kotisivun osalta luotiin testit, jotka varmistivat saavutettavuuden ping-testillä sekä sen, että sivun DNS ohjautuu oikein, eli toisin sanoen sivu latautuu oikealla verkko-osoitteella. Lisäksi testi varmistaa, että sivun SSL-sertifikaatti on voimassa ja että http-kyselyt menevät läpi. Testien myötä on mahdollista tarkastella myös sivun latausnopeutta sekä sitä, että sivun sisältö latautuu oikein. Sivun sisällön osalta luotiin testi, joka varmistaa, että sivulta löytyy merkkijono ”Haaga-Helia”. Apache web-palvelimen monitorointi jäi tämän projektin osalta Moodle-testipalvelimella toteuttamatta ja se olisikin seuraava looginen jatkumo Moodle-monitoroinnin kehittämiseksi, yhdessä synteettisten käyttäjätarinatestausten kanssa.

Nagios-palvelimen osalta toteutimme monitoroinnin konfiguraation Moodlea vastaavasti palvelimen komponenteille, joiden lisäksi monitoroimme joitain palvelimella ajettavia prosesseja. Monitoroitavia prosesseja olivat muun muassa Apachen web-palvelin sekä SSH-palvelin. Nagios hyödyntää Apachen web-palvelinta web-käyttöliittymän osalta.

Monitoroitavien kohteiden pohjalta konfiguroimme myös sähköpostihälytykset asettamalla sähköpostipalvelimen asetukset ja luomalla kustomoidun HTML-koodia hyödyntävän hälytyspohjan, jonka tarkoituksena oli saattaa monitorointihälytykset selkeästi ja nopeasti hälytyksiä seuraaville tahoille. Tarkkojen hälytyskriteereiden määrittely jätettiin työn ulkopuolelle, mutta loimme erinäisiä testihälytyksiä tehtyjen Moodlen ja Nagioksen monitorointien pohjalta sekä testasimme niiden toimintaa. Testauksen tuloksena luotiin hälytykset, joiden toiminta varmistettiin sekä yksittäisten henkilöiden että erikseen määriteltyjen ryhmien pohjalta. Varmistimme toiminnan myös niin, että hälytyksiä on mahdollista lähettää suoraan helpdeskin ja järjestelmien asiantuntijoiden yhteisessä käytössä olevaan toiminnanohjausjärjestelmään.

Lisäsimme myös suurimman osan Haaga-Helian virtuaalipalvelimista Nagioksen monitoroinnin piiriin. Monitoroinnin piiriin kuului lopulta yhteensä 764 kohdetta.

Työn aikana pohdimme kuinka Haaga-Helia voisi jatkokehittää monitorointia tulevaisuudessa sekä esitimme konkreettisia kehitysehdotuksia monitoroinnin yhdistämiseksi osaksi prosesseja. Näitä kehitysehdotuksia on esitetty tarkemmin luvussa 17.5.

# 17 Pohdinta

Projektin tavoitteena oli valita Haaga-Helialle sopiva monitorointiratkaisu ja toteuttaa sillä Moodlen testipalvelimen monitorointia. Tehtävänantona tavoite kuulosti korviimme verrattain yksinkertaiselta, joskin tiedostimme sen vaativan melko runsaasti teknistä perehtymistä, sillä meillä ei ollut aiempaa kokemusta Haaga-Helian kokoisen organisaation laajuuden kattavista monitorointityökaluista.

Työtä aloittaessamme ymmärsimme kuitenkin nopeasti, että monitorointiin olisi perehdyttävä myös ajatuksen tasolla ja lisäksi tutustuttava hyviksi todettuihin käytäntöihin. Tämän myötä ymmärsimme myös hieman omaksi yllätykseksemme prosessilähtöisen ajattelutavan merkityksen. Aiemmat henkilökohtaiset kokemuksemme monitoroinnista ovat olleet hyvin suorituskykylähtöisiä, siinä missä Haaga-Helian suuruisessa yrityksessä liiketoiminnan kannalta oleellisia palveluita ja palvelimia on runsaasti ja toisaalta niihin liittyy hyvin suuri käyttäjämäärä, joka osaltaan asettaa vaatimuksen myös kokonaisvaltaisemmalle monitoroinnille. Monitorointi ei siis ole pelkkä työkalu vaan parhaimmillaan osa päivittäisiä prosesseja.

Prosessikuvausta tehdessämme yllätyimme myös siitä, kuinka moneen sidosryhmään monitoroinnin hyödyt voivat vaikuttaa. Tämä taas puolestaan johti siihen, että jouduimme pohtimaan opinnäytetyön lähestymistapaa hieman uudelleen, jotta prosessilähtöisyyden konkreettinen hyöty välittyisi myös lopullisesta dokumentaatiosta.

## 17.1 Haasteet

Projektin yhdeksi merkittävimmäksi haasteeksi osoittautui jo melko alkuvaiheessa toteutettu monitorointityökalujen vertailu. Emme olleet osanneet varautua siihen, että monien verrattain suurtenkin toimijoiden tuotteistaan tarjoama dokumentaatio olisi niin vaihtelevan tasoista. Paikoitellen tuotteista tarjottu dokumentaatio oli jopa lähes saavuttamattomissa ilman, että palveluun olisi sitoutunut, jolloin jouduimme luottamaan kolmansien osapuolten kuvauksiin tuotteen toiminnasta. Teimme virheellisen oletuksen siitä, että tuotteet, joiden tarkoituksena on monitoroida arkkitehtuuriltaan hyvin laajojakin palvelinympäristöjä, olisivat sellaisia, että myös niiden toiminta ja tekniikat olisivat kuvattu melko kattavasti jo ennen ostopäätöstä. Mielestämme onnistuimme kuitenkin koostamaan vertailluista palveluista tarpeisiin suhteutettuna tarpeeksi kattavan kokonaisuuden, mutta työhön kului huomattavasti pidempi aika, kuin mitä olimme projektia suunnitellessamme ajatelleet.

Mikäli ryhtyisimme toteuttamaan vastaavaa projektia nyt, tekisimme erityisesti yhden asian toisin. Kuten luvussa aiemmin keskusteltiin, tuli monitoroinnin prosessilähtöisyys meille yllätyksenä vasta sen jälkeen, kun opinnäytetyöprojekti oli jo hyvässä vauhdissa. Jälkiviisaina tiedostamme, että tietohallintoyksikön järjestelmien asiantuntijat olisi ollut kannattavaa ottaa projektiin selkeämmin mukaan jo sen alkuvaiheessa, ennen kuin varsinaista ratkaisua alettiin edes toteuttaa. Aktiivisempi kommunikaatioyhteys olisi mahdollistanut sen, että kyseiset järjestelmien asiantuntijat olisivat päässeet paremmin sisälle monitorointiin jo projektin aikana, jonka myötä myös sen jatkokehitys olisi osaltaan saanut lisävarmistusta. Samalla olisimme ehkä voineet saada juuri Haaga-Helian ympäristöön sopivia ideoita myös heidän puoleltaan.

Kolmantena haasteena työn lopputulokselle oli työn laajuuden kasvaminen sen tekovaiheessa. Alkuperäinen ajatuksemme lopputuloksesta painottui selkeämmin tekniseen toteutukseen, mutta kuten aiemmin totesimme, ymmärsimme monitoroinnin prosessilähtöisyyden vasta myöhemmin. Tämä oivallus puolestaan kasvatti tarvetta tietoperustalle, mutta työlle varaamamme aika pysyi silti samana. Tästä syystä työstä jäi pois joitain sellaisia teoriataustoja, joilla olisimme voineet paremmin tukea joitain tekemiämme valintoja. Esimerkiksi tuotteiden vertailuun olisi ollut mahdollista hyödyntää myös investointilaskennan mallia ja toisaalta mikäli olisimme ymmärtäneet työn tutkimusaspektin aiemmin, olisimme voineet liittää työhön myös tutkimuksen teoriaa. Käytännössä opinnäytetyön tyypin määritelmä hämärtyi sen edetessä puhtaasti toiminnallisesta työstä kattamaan myös tutkimuksen piirteitä.

### 17.2 Lopputulos

Mielestämme projekti vastasi sille määriteltyyn toimeksiantoon. Otimme käyttöön monitorointiohjelmisto Nagioksen onnistuneesti ja toteutimme siihen sellaisia konfiguraatioita, joita ajattelemme voitavan hyödyntää myös muiden palvelinten monitoroinnissa. Tehdyt konfiguraatiot ovat jo sellaisenaan käyttökelpoisia osana tietohallintoyksikön prosesseja.

Ajatuksena oli, että raportissa esiteltäisiin tehdyt konfiguraatiot niin, että niiden pohjalta olisi helppo jatkokehittää monitorointia. Tästä syystä päätimme, että vaikka toimeksiantajan kanssa oli sovittu Moodlen testipalvelimen monitoroinnista, emme kuitenkaan keskittyisi dokumentissa raportoimaan tehtyjä konfiguraatioita liikaa vain Moodlen näkökulmasta, sillä ajatuksenahan oli pystyä toistamaan ja soveltamaan konfiguraatioita myös muille palvelimille. Tähän päätökseen vaikutti myös se, että pystyimme näin säilyttämään työn sellaisena, ettei se anna ilmi liikaa yksityisiä tietoja organisaation tietoverkoista ja palvelinympäristöstä.

Opinnäytetyö on dokumentaation osalta saatettu päätökseen, mutta varsinainen työ on Haaga-Helian näkökulmasta vasta alkamassa. Monitorointiprosessin jatkuvuuden ja kehityksen varmistamiseksi jatkamme yhteistyötä Haaga-Helian kanssa vielä erillisen projektin esittelyn sekä mahdollisten työpajojen muodossa. Tarkoituksena on käydä läpi monitoroinnin periaatteita yhdessä Haaga-Helian tietohallintoyksikön työntekijöiden kanssa sekä tutustua Nagioksen käyttöön käytännössä, sen tarjoamiin mahdollisuuksiin ja varsinaiseen konfiguraatioon.

### 17.3 Ajankohtaisuus

Uskomme monitorointiprojektin hyödyttävän Haaga-Heliaa jatkossa ja olevan sen myötä merkityksellinen. Organisaatiossa viimeisen vuoden aikana tapahtuneet laajat järjestelmäuudistukset luovat omalta osaltaan uudenlaisen paineen monitoroinnin kehitykselle, jotta uusien järjestelmien toimintaa saadaan tuettua ja varmistettua. Lisäksi oma historiamme työharjoittelijoina Haaga-Helian tietohallintoyksikössä on tuonut meille kokemuksen siitä, kuinka laajalti järjestelmien erilaiset häiriöt vaikuttavat loppukäyttäjiin ja kuinka niihin reagoidaan. Monitoroinnin hyötyjä ovat muun muassa suunnittelemattomien käyttökatkojen ehkäisy jo ennakoivasti sekä avoimuuden lisäys järjestelmähäiriöistä, jonka erityisesti uskomme olevan tervetullut uudistus Haaga-Heliassa.

Tämä opinnäytetyö kattoi kuitenkin vain pienen osan siitä monitoroinnista, jota Haaga-Helia voisi ympäristössään toteuttaa. Sen lisäksi, että organisaatiossa on tehty suuria järjestelmämuutoksia, on aihe erityisen ajankohtainen myös hallinnollisten muutosten näkökulmasta. Tietohallintopalveluiden yksikkö koki opinnäytetyön toteutusajankohtana muutoksia henkilöstöpalkkausten ja uuden tietohallintopäällikön osalta. Siinä missä prosessien kehitys on tietysti mahdollista ilman henkilöstömuutoksiakin, ajattelemme tällaisten muutosten usein luovan uudenlaisen mahdollisuuden muutokselle sekä antavan tilaa kehitykselle, kun prosesseja käydään läpi tuorein silmin.

### 17.4 Oppimistavoitteet

Etukäteen työlle asettamamme oppimistavoitteet olivat vahvasti sidoksissa varsinaisen monitorointiratkaisun toteutukseen. Toivoimme saavamme konkreettista osaamista monitoroinnin toteutuksesta ja siihen käytettävistä tekniikoista. Tältä osin oppimistavoite toteutui, ja meillä on nyt vahvempi käsitys siitä, kuinka monitorointia voisi lähteä kehittämään myös muissa ympäristöissä.

Varsinaisen teknisen osaamisen lisäksi opinnäytetyö ohjasi meitä ajattelemaan myös tekniikkaa pidemmälle ja uskomme tämän olevan arvokas taito, kun jatkossa suunnittelemme erilaisiin ympäristöihin teknisiä ratkaisuja. Kestävien ja toimivien ratkaisujen toteuttamiseksi on tärkeää pohtia, kuinka ratkaisu vaikuttaa eri tahoihin ja sidosryhmiin.

Kolmanneksi oppimistilanteeksi koimme toimeksiantajan kanssa käydyt neuvottelut ja niiden pohjalta tehdyn työn rakenteen ja sisällön suunnittelun, jotta työ vastasi toimeksiantajan toiveita. On eri asia tehdä työtä, jonka ensisijainen lukija tai hyödyntäjä on tuntematon tai jos työtä tehdään pääasiassa omaa tarvetta varten. Uskomme projektin antavan meille uudenlaista kokemusta tulevaisuutta varten nimenomaan siksi, että projekti toteutettiin todelliselle yritykselle.

## 17.5 Kehitysehdotukset

Opinnäytetyöprojektin aikana käydyissä keskusteluissa heräsi monia eri ideoita siitä, kuinka monitorointiratkaisua voitaisiin vielä kattavammin hyödyntää osana Haaga-Helian prosesseja. Työn aikataulullisten rajoitusten vuoksi kaikkea ei kuitenkaan voitu tämän projektin aikana toteuttaa. Kokosimme yhteen muutamia kehitysehdotuksia, jotka voisivat mielestämme hyödyntää Haaga-Heliaa tulevaisuudessa. Kuten dokumentissa aiemmin todettiin, monitorointi on prosessi, joka ei välttämättä valmistu koskaan, vaan uusia ideoita syntyy sitä mukaa kun ympäristö ja sen käyttäjät kehittyvät. Tämä ei kuitenkaan tarkoita sitä, etteikö monitoroinnista voisi saada suurta hyötyä jo pienelläkin panostuksella.

### 17.5.1 Monitoroinnin kehitys

Tämän projektin tarkoituksena oli toteuttaa monitorointi pääasiassa Haaga-Helian käytössä olevan Moodle-testipalvelimen osalta, niin sanottuna soveltuvuusselvityksenä. Haaga-Helialla on kuitenkin useita palvelimia ja palveluita, joiden monitorointia ei vielä ole toteutettu. Jotta monitoroinnista saadaan paras hyöty irti, tulisikin seuraavaksi keskittyä monitoroinnin laajentamiseen niin, että se kattaa myös muut palvelimet ja palvelut.

Monitoroinnin perustana ovat erilaiset palveluiden ja palvelimien toiminnalle määritellyt raja-arvot. Raja-arvot voivat vaihdella paljonkin riippuen siitä mitä monitoroidaan ja kuinka kriittinen monitoroitava kohde on liiketoiminnan kannalta. Opinnäytetyön aikana emme keskittyneet määrittelemään kyseisiä arvoja, sillä niiden määrittely vaatisi laajempaa tuntemusta järjestelmistä ja niiden tarpeista. Monitoroinnin raja-arvojen määrittely onkin asia, johon olisi oleellista tarttua jo varhaisessa vaiheessa monitoroinnin jatkokehitystä. Oleellista on myös se, että määrittelyyn osallistuisi mahdollisimman moni taho, jotta arvot ovat aidosti hyödyllisiä, sillä raja-arvot toimivat perustana myös hälytysten määritykselle. On

oleellista, että hälytykseen johtavat kriteerit ovat tarkoin määriteltyjä niin, että syntyvät hälytykset vastaavat todellisia ongelmatilanteita, eikä turhia hälytyksiä syntyisi.

Käytännössä raja-arvojen määrittely voi olla alkuun pidempiaikainen prosessi, eikä tarkoituksena olekaan saada kerralla täydellistä määrittelyä. Monitoroinnin tarpeetkin voivat ajan myötä muuttua. Määrittelyprosessin aikana voidaan testata ja harkita miten kuhunkin raja-arvojen ylitykseen on kannattavaa reagoida, sillä kaikesta ei välttämättä ole tarpeellista luoda hälytystä, vaikka arvot ylittyisivätkin.

Nagios XI tukee myös mahdollisuutta lähettää sähköpostin lisäksi, tai sen sijaan, hälytyksiä tekstiviestitse. Tekstiviestien lähetys ei sisältynyt opinnäytetyön sisältöön, mutta tekstiviestejä voitaisiin hyödyntää joissain erittäin kriittiseksi laskettavissa tilanteissa. Tämä tarkoittaisi tilanteita, joissa olisi erittäin tärkeää, että tieto ongelmatilanteesta saavuttaa järjestelmien ylläpitäjät heti, eikä esimerkiksi vasta seuraavana arkipäivänä. Jotta tällaisista hälytyksistä voitaisiin saada irti paras hyöty, tulisi organisaation todennäköisesti miettiä myös sitä, kuinka saatavilla työntekijöiden tulisi olla työaikojen ulkopuolella. Monitoroinnin prosessien läpikäyminen on siis muutakin kuin suoranaisten konfiguraatioiden ja työaikaan sisältyvien toimenpiteiden läpikäyntiä.

### 17.5.2 Prosessien kehitys monitoroinnin näkökulmasta

Mielestämme on tärkeää, että monitorointi on avointa ja saavutettavaa kaikkien niiden henkilöiden välillä, jotka järjestelmiä ylläpitävät tai jotka vastaavat tiedottamisesta eteenpäin asiakkaille. Näin myös varmistetaan se, että monitorointi on tarpeeksi monipuolista ja kattaa useamman käyttäjäryhmän tarpeet. Samalla myös vastuu järjestelmien vikatiloihin reagoimisesta jakautuu tasaisesti useamman asiantuntijan välille, eikä esimerkiksi palvelimen häiriöön reagoiminen ole vain yhden henkilön varassa. Tällä hetkellä, kun monitorointia ollaan vasta ottamassa käyttöön, ei monitorointiin osallistu vielä kovinkaan moni henkilö ja se onkin selkeä osa-alue monitoroinnin jatkokehittämiseksi osaksi jo olemassa olevia prosesseja.

Monitoroinnin kehittyessä mahdollistuu myös kattavan raportoinnin hyödyntäminen. Erityisesti erilaisten ongelmatilanteiden tai niiden jälkipuinnin yhteydessä voitaisiin hyötyä monitoroinnin tarjoamasta datasta. Raporttien perusteella voitaisiin selvittää mitä on tapahtunut ja miksi tai kuinka ongelmaan reagoitiin. Raporttien läpikäynti voisi olla hyödyllistä sekä yksikön sisällä, että myös silloin kun yksikön toiminnasta raportoidaan ylemmälle tasolle. Raportit voisi liittää myös osaksi yksikön viikoittaisia palavereja, joiden myötä myös

monitoroinnin rooli osana prosesseja vahvistuisi ja tämän myötä saattaisi parantaa kokemusta siitä, että kaikilla on mahdollisuus osallistua monitoroinnin sekä järjestelmien kehitykseen.

### 17.5.3 Web-sivusto

Toimeksiantajan kanssa keskusteltiin mahdollisuudesta toteuttaa erillinen web-sivusto, josta voitaisiin nopealla vilkaisulla tarkistaa palveluiden toiminta. Sivuston saama data pohjautuisi Nagioksen keräämään monitorointidataan ja se koostettaisiin sivulle esimerkiksi selkeästi tulkittavana liikennevalomallina. Sivustoon voitaisiin valita Haaga-Helian ulkopuolelle näkyvän toiminnan kannalta oleellisimpia palveluita, kuten esimerkiksi julkinen verkkosivu, Moodle- ja Peppi-palvelimet tai jokin muu opiskelun kannalta oleellinen palvelu. Opiskelija tai henkilöstön jäsen voisi sivuston kautta helposti tarkistaa palveluiden kunkin hetkisen tilanteen. Hyöty tulisi esille erityisesti tilanteessa, jossa henkilö ei saa jotakin palvelua auki. Tällöin hän voisi itse tarkistaa onko järjestelmässä jokin jo tiedossa oleva häiriö. Tiedon jakaminen avoimemmin palveluiden loppukäyttäjien tietoon saattaisi osaltaan myös helpottaa tietohallintoyksikön asiakaspalvelua, sillä vikatilanteissa toiminnanohjausjärjestelmä ja puhelinpäivystys ruuhkautuvat herkästi vikailmoituksista.

## Lähteet


Adato, L. 2018. Monitoring 101: A primer to the philosophy, theory, and fundamental concepts involved in systems monitoring. Luettavissa: https://www.solarwinds.com/-/media/solarwinds/swdc/resources/whitepaper/monitoring-101-ebook.ashx. Luettu: 12.3.2019.

Barth W. 2005. Nagios: System and Network Monitoring. No Starch Press. San Francisco.

Barth W. 2009. Nagios, 2nd edition. System and Network Monitoring. No Starch Press. San Francisco.

Blomberg, O., Milne, K., Palislamovic, S. & Sonderegger, J. 2009. JUNOS High Availability: Best Practices for High Network Uptime. O'Reilly Media, Inc. Sebastopol.

Brazil, B. 2018. Prometheus: Up & Running: Infrastructure and Application Performance Monitoring. O'Reilly Media Inc. Sebastopol.

Canon, Inc. 2015. Monitoring and Controlling the Machine with SNMP. Luettavissa: https://oip.manual.canon/USRMA-0336-zz-SS-enUS/contents/SS729_network_226monitoringandcontrollingthemachinewithsnmp.html. Luettu 3.5.2019.

Cisco Systems, Inc. 2007. Management Information Base Overview. Luettavissa: https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/pgw/7/mibs/guide/7MIB_Ch1.html. Luettu: 10.3.2019.

Cisco Systems, Inc. 2015. SNMP Configuration Guide, Cisco IOS XE Release 3SE (Cisco WLC 5700 Series). Luettavissa: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/snmp/configuration/xe-3se/5700/snmp-xe-3se-5700-book/nm-snmp-snmpv3.pdf. Luettu: 10.3.2019.

Curtis, M., Wilkinson, J & Hettich, S. 2016. Site Reliability Engineering: How Google Runs Production Systems. O'Reilly Media, Inc. Sebastopol.

Dissmeyer, J. & Dissmeyer J. 2013. SolarWinds Orion Network Performance Monitor. Packt Publishing, Limited. Birmingham.

Donaldson, T. & Preston, L. 1995. The Stakeholder Theory of the Corporation: Concepts, Evidence, and Implications.

Elasticsearch B.V. 2019. Basic Concepts. Luettavissa: https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started-concepts.html Luettu: 9.3.2019.

Evans, K. 2019. Cloud Native Computing Foundation Announces Prometheus Graduation. Luettavissa: https://www.cncf.io/announcement/2018/08/09/prometheus-graduates/. Luettu: 14.3.2019.

Haaga-Helia ammattikorkeakoulu. 2019. Vuosikertomus 2018. Haaga-Helia ammattikorkeakoulu Oy. Luettavissa: https://www.haaga-helia.fi/sites/default/files/Kuvat-ja-liitteet/vuosikertomus_2018.pdf. Luettu: 5.3.2019.

Icinga. 2019a. Icinga2 GitHub. Luettavissa: https://github.com/Icinga/icinga2. Luettu: 27.2.2019.

Icinga. 2019b. Service Monitoring. Luettavissa: https://icinga.com/docs/icinga2/latest/doc/05-service-monitoring/. Luettu: 27.2.2019.

Icinga. 2019c. Icinga Support Matrix. Luettavissa: https://icinga.com/support/support-details/. Luettu: 27.2.2019.

Icinga. 2019d. Icinga 2 API. Luettavissa: https://icinga.com/docs/icinga2/latest/doc/12-icinga2-api/. Luettu: 27.2.2019.

IDERA, Inc. 2016. The Truth about Agent vs. Agentless Monitoring: A Short Guide to Choosing the Right Monitoring Solution. IDERA, Inc. Luettavissa: https://www.idera.com/~/media/Corporate/Files/WhitePapers/IderaWP_Agent_vs_Agentless_Monitoring.ashx. Luettu: 13.3.2019.

InfluxData, Inc. 2019a. Introduction to the InfluxData Platform. Luettavissa: https://docs.influxdata.com/platform/introduction. Luettu: 25.2.2019.

InfluxData, Inc. 2019b. Telegraf 1.10 documentation. Luettavissa: https://docs.influxdata.com/telegraf/v1.10/. Luettu: 25.2.2019.

InfluxData, Inc. 2019c. InfluxDB 1.7. documentation. Luettavissa: https://docs.influxdata.com/influxdb/v1.7/. Luettu: 25.2.2019.

InfluxData, Inc. 2019d. Kapacitor 1.5. documentation. Luettavissa: https://docs.influxdata.com/kapacitor/v1.5/. Luettu: 25.2.2019.

InfluxData, Inc. 2019e. Chronograf 1.7. documentation. Luettavissa: https://docs.influxdata.com/chronograf/v1.7/. Luettu: 25.2.2019.

InfluxData, Inc. 2019f. Telegraf. Luettavissa: https://www.influxdata.com/time-series-platform/telegraf/. Luettu: 25.2.2019.

Josephsen, D. 2016. 5 Lines I Couldn't Draw'. Monitorama PDX 2016. Avoimen lähdekoodin konferenssi & hackathon. Portland. Katsottavissa: https://vimeo.com/173610048. Katsottu: 22.5.2019.

Julian, M. 2017. Practical Monitoring: Effective Strategies for the Real World. O'Reilly Media, Inc. Sebastopol.

Kennedy, J. & Satran, M. 2018. Windows Management Instrumentation – Windows applications. Luettavissa: https://docs.microsoft.com/en-us/windows/desktop/wmisdk/wmi-start-page. Luettu: 13.3.2019.

Ligus, S. 2012. Effective Monitoring and Alerting. O'Reilly Media, Inc. Sebastopol.

Mason, A. & Newcomb, M. 2001. Cisco Secure Internet Security Solutions. Cisco Press. Indianapolis.

McCabe, J. 2011. Network Management Know It All. Morgan Kaufmann Publishers. Burlington.

Mitchell, R., Agle, B. & Wood, D. 1997. Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts. Luettavissa: https://www.syrialearning.org/system/files/content/resource/files/main/259247.pdf. Luettu: 10.5.2019.

Mobily, T. 2012. Nagios Vs. Icinga: the real story of one of the most heated forks in free software. Luettavissa: http://freesoftwaremagazine.com/articles/nagios_and_icinga/. Luettu: 27.2.2019.

Nagios Enterprises. 2017a. Using SSL/TLS with Active Directory / LDAP. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Using_SSL_with_XI_Active_Directory_Component.pdf. Luettu: 28.4.2019.

Nagios Enterprises. 2017b. Installing the Windows Agent: NSClient++. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Installing-The-Windows-Agent-NSClient++-for-Nagios-XI.pdf. Luettu: 25.4.2019.

Nagios Enterprises. 2017c. How To Monitor Devices Using The NCPA Agent and Wizard. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Monitoring-Devices-Using-The-NCPA-Agent-And-Nagios-XI.pdf. Luettu: 25.4.2019.

Nagios Enterprises. 2017d. Installing the Linux NRPE Agent. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Installing_The_XI_Linux_Agent.pdf. Luettu: 25.4.2019.

Nagios Enterprises. 2017e. Monitoring JMX with Nagios XI. https://assets.nagios.com/downloads/nagiosxi/docs/Monitoring-JMX-with-Nagios-XI.pdf. Luettu: 29.4.2019.

Nagios Enterprises. 2018a. Hardware requirements. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Nagios-XI-Hardware-Requirements.pdf. Luettu: 16.4.2019.

Nagios Enterprises. 2018b. Best Practices. https://assets.nagios.com/downloads/nagiosxi/docs/Nagios-XI-Best-Practices.pdf. Luettu: 16.4.2019.

Nagios Enterprises. 2018c. Installing Nagios XI Manually on Linux. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Installing-Nagios-XI-Manually-on-Linux.pdf. Luettu: 22.4.2019.

Nagios Enterprises. 2018d. Backing up and restoring your Nagios XI system. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Backing-Up-And-Restoring-Nagios-XI.pdf. Luettu: 22.4.2019.

Nagios Enterprises. 2018e. How To Monitor A Websensor EM08. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Monitoring-A-Websensor-EM08-with-Nagios-XI.pdf Luettu: 8.5.2019.

Nagios Enterprises. 2018f. Monitoring Websites with Nagios XI. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Monitoring-Websites-With-Nagios-XI.pdf. Luettu: 29.4.2019.

Nagios Enterprises. 2018g. How to Configure Email and Text Notifications. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Configuring-Email-And-Text-Notifications-in-Nagios-XI.pdf. Luettu 26.2.2019.

Nagios Enterprises. 2018h. Object Configuration Overview. Luettavissa: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/configobject.html. Luettu: 12.4.2019.

Nagios Enterprises. 2018i. Determining Status and Reachability of Network Hosts. Luettavissa: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/networkreachability.html. Luettu: 12.4.2019.

Nagios Enterprises. 2018j. Host Checks. Luettavissa: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/hostchecks.html. Luettu: 12.4.2019.

Nagios Enterprises. 2018k. Service Checks. Luettavissa: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/servicechecks.html. Luettu: 12.4.2019.

Nagios Enterprises. 2018l. Nagios Plugins. Luettavissa: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/plugins.html. Luettu: 13.5.2019.

Nagios Enterprises. 2018m. NCPA Overview. Luettavissa: https://www.nagios.org/ncpa/help.php#configuration. Luettu: 13.5.2019.

Nagios Enterprises. 2018n. NRPE – Agent and Plugin Explained. Luettavissa: https://support.nagios.com/kb/article/nrpe-agent-and-plugin-explained-612.html. Luettu: 13.5.2019.

Nagios Enterprises. 2019a. Frequently Asked Questions. Luettavissa: https://www.nagios.com/products/nagios-xi/#faqs. Luettu 26.2.2019.

Nagios Enterprises. 2019b. Active Checks. Luettavissa: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/activechecks.html. Luettu 26.2.2019.

Nagios Enterprises. 2019c. Passive Checks. Luettavissa: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/passivechecks.html. Luettu 26.2.2019.

Nagios Enterprises. 2019d. Agent Installation Instructions. Luettavissa: https://assets.nagios.com/downloads/ncpa/docs/Installing-NCPA.pdf. Luettu: 25.4.2019.

Nagios Enterprises. 2019e. Monitoring VMware With Nagios XI. Luettavissa: https://assets.nagios.com/downloads/nagiosxi/docs/Monitoring-VMware-With-Nagios-XI.pdf Luettu: 5.5.2019

NSClient. 2018. About NSClient++. Luettavissa: https://docs.nsclient.org/. Luettu: 13.5.2019.

Olups, R. 2016. Zabbix Network Monitoring – Second Edition. Packt Publishing. Birmingham.

Orange SA. 2015. sysDescr. Luettavissa: http://www.oid-info.com/get/1.3.6.1.2.1.1.1. Luettu: 10.3.2019.

Paessler AG. 2010. Architecture: PRTG Core Server, PRTG Probe and the User Interfaces. Luettavissa: http://prtg.interlake.net/help/system_architecture.htm. Luettu: 4.3.2019.

Paessler AG. 2019a. Frequently Asked Questions. Luettavissa: https://www.paessler.com/support/faqs. Luettu: 4.3.2019.

Paessler AG. 2019b. PRTG Manual: Introduction: Monitoring with PRTG. Luettavissa: https://www.paessler.com/manuals/prtg/introduction_monitoring_with_prtg. Luettu: 4.3.2019.

Paessler AG. 2019c. PRTG Manual: Clustering. Luettavissa: https://www.paessler.com/manuals/prtg/clustering. Luettu: 4.3.2019.

Paul, A. 2009. ITIL Heroes Handbook. CreateSpace. Paramount. Luettavissa: https://www.manageengine.com/products/service-desk/itil-whitepaper.html. Luettu 19.5.2019.

Prometheus. 2019a. Overview: What is Prometheus? Luettavissa: https://prometheus.io/docs/introduction/overview/. Luettu: 6.3.2019.

Prometheus. 2019b. Prometheus GitHub. Luettavissa: https://github.com/prometheus/pushgateway. Luettu: 6.3.2019.

Prometheus. 2019c. Exporters and Integrations. Luettavissa: https://prometheus.io/docs/instrumenting/exporters/. Luettu: 6.3.2019.

SNMP Research International, Inc. 2019. SNMPv3 with Security and Administration. Luettavissa: http://www.snmp.com/snmpv3/snmpv3_intro.shtml. Luettu: 10.4.2019.

SolarWinds, Inc. 2018. Getting Started Guide Network Performance Monitor Version 12.4 Part 1 of 2: Get Started. Luettavissa: https://documentation.solarwinds.com/archive/pdf/npm/NPM_Getting_Started_Guide_1_Get_Started.pdf. Luettu: 8.3.2019.

SolarWinds, Inc. 2019. Administrator Guide Network Performance Monitor Version 12.4. Luettavissa: https://documentation.solarwinds.com/archive/pdf/npm/NPM_Administrator_Guide.pdf. Luettu: 8.3.2019.

Turnbull J. 2016. The Art of Monitoring. Turnbull Press. Brooklyn.

White, S. 2008. Process Modeling Notations and Workflow Patterns. The Workflow Patterns initiative. Luettavissa: https://www.omg.org/bpmn/Documents/Notations_and_Workflow_Patterns.pdf. Luettu: 9.5.2019.

Zabbix LLC. 2017a. Zabbix features. Luettavissa: https://www.zabbix.com/documentation/4.0/manual/introduction/features?rev=1503319531. Luettu: 13.3.2019.

Zabbix LLC. 2017b. Notifications & Automatic actions. Luettavissa: https://www.zabbix.com/notification. Luettu: 13.3.2019.

Zoho Corp. 2019. What is SNMP? Luettavissa: https://www.site24x7.com/network/what-is-snmp.html. Luettu: 10.4.2019.

# Liitteet

## Liite 1. Työnjako

| Tehtävä | Tekijä |
|---|---|
| Tiivistelmä | Matilda Sinervo |
| Sanasto | Matilda Sinervo |
| Johdanto | Matilda Sinervo ja Joonas Valsta |
| Haaga-Helia toimeksiantajana | Matilda Sinervo |
| Monitoroinnin hyödyt Haaga-Heliassa | Matilda Sinervo |
| Sidosryhmät | Matilda Sinervo |
| Prosessikuvaus | Matilda Sinervo |
| Monitoroinnin tuottamat mittarit | Matilda Sinervo ja Joonas Valsta |
| Monitoroinnista yleisesti | Joonas Valsta |
| Monitoroinnin periaatteet | Joonas Valsta |
| Datan keruu | Joonas Valsta |
| Datan säilöntä | Joonas Valsta |
| Datan visualisointi | Joonas Valsta |
| Analytiikka ja raportointi | Joonas Valsta |
| Hälytykset | Joonas Valsta |
| Monitoroinnin tarkoitus | Joonas Valsta |
| Laatikkotestaukset monitoroinnissa | Joonas Valsta |
| Black-box monitorointi | Joonas Valsta |
| White-box monitorointi | Joonas Valsta |
| Agentiton monitorointi | Matilda Sinervo |
| Simple Network Management Protocol | Joonas Valsta |
| Management Information Base ja Object Identifier | Joonas Valsta |
| SNMP versiot | Joonas Valsta |
| SNMP komennot | Joonas Valsta |
| Windows Management Instrumentation | Matilda Sinervo |
| Monitorointiratkaisu palveluna | Joonas Valsta |
| Monitorointi prosessien tukena | Joonas Valsta |
| Palvelujen monitorointi | Matilda Sinervo |
| Monitorointiratkaisuja | Matilda Sinervo |
| Nagios XI | Matilda Sinervo |
| SolarWinds NPM | Matilda Sinervo |
| PRTG | Matilda Sinervo |
| Zabbix | Matilda Sinervo |
| Icinga 2 | Matilda Sinervo |
| Prometheus | Matilda Sinervo |
| TICK Stack | Matilda Sinervo |
| Monitorointiratkaisujen vertailu | Matilda Sinervo ja Joonas Valsta |

| Tehtävä | Tekijä |
|---|---|
| Valittu monitorointiratkaisu | Matilda Sinervo |
| Nagios XI:n laitevaatimukset | Matilda Sinervo |
| Käytössä oleva laitteisto | Matilda Sinervo |
| Nagios XI asennus ja konfigurointi | Matilda Sinervo ja Joonas Valsta |
| Esivalmistelut | Matilda Sinervo ja Joonas Valsta |
| Nagios XI asennus | Matilda Sinervo ja Joonas Valsta |
| Nagios XI:n konfigurointi | Matilda Sinervo ja Joonas Valsta |
| Aktiivihakemiston integraatio | Joonas Valsta |
| Monitorointi toimeksiantajan palveluympäristössä | Matilda Sinervo ja Joonas Valsta |
| Monitorointi Nagios XI:ssä | Joonas Valsta |
| Monitoroitavien kohteiden verkkohierarkia Nagios XI:ssä | Joonas Valsta |
| Nagios XI agentit | Matilda Sinervo |
| Palvelimien monitorointi | Joonas Valsta |
| Windows-palvelimien monitorointi | Matilda Sinervo |
| Nagios XI agentin asennus Windows-palvelimelle | Matilda Sinervo |
| Nagios XI agentin konfigurointi ja käyttöönotto Windows-palvelimella | Matilda Sinervo |
| Linux palvelimien monitorointi | Joonas Valsta |
| Nagios XI agentin asennus Linux-palvelimelle | Joonas Valsta |
| Nagios XI agentin konfigurointi ja käyttöönotto Linux-palvelimella | Joonas Valsta |
| VMwaren virtualisointialustat | Joonas Valsta |
| Muut monitoroitavat kohteet | Joonas Valsta |
| Moodlen monitorointi | Matilda Sinervo ja Joonas Valsta |
| Moodlen testipalvelimien komponenttien monitorointi | Matilda Sinervo ja Joonas Valsta |
| Moodlen verkkosivun monitorointi | Matilda Sinervo ja Joonas Valsta |
| Hälytykset | Joonas Valsta |
| Sähköposti | Matilda Sinervo ja Joonas Valsta |
| Web-käyttöliittymä hälytyksien tukena | Matilda Sinervo |
| Hälytysten testaus | Matilda Sinervo ja Joonas Valsta |
| Nagios XI:n web-käyttöliittymän näkymät | Matilda Sinervo |
| Nagios XI:n raportointimahdollisuudet | Joonas Valsta |
| Tulokset | Matilda Sinervo ja Joonas Valsta |
| Matka tulokseen | Matilda Sinervo ja Joonas Valsta |
| Tulos ja tuloksen hyöty toimeksiantajalle | Matilda Sinervo ja Joonas Valsta |
| Pohdinta | Matilda Sinervo |
| Haasteet | Matilda Sinervo ja Joonas Valsta |
| Lopputulos | Matilda Sinervo ja Joonas Valsta |
| Ajankohtaisuus | Matilda Sinervo ja Joonas Valsta |
| Oppimistavoitteet | Matilda Sinervo ja Joonas Valsta |
| Kehitysehdotukset | Matilda Sinervo |
| Monitoroinnin kehitys | Matilda Sinervo ja Joonas Valsta |
| Prosessien kehitys monitoroinnin näkökulmasta | Matilda Sinervo ja Joonas Valsta |
| Web-sivusto | Matilda Sinervo ja Joonas Valsta |

## Liite 2. Prosessi uimaratamallina esitettynä