

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Tanel Saar 193866IAAB

**Haldusmuudatuste automatiseerimislahenduse  
loomine Windows rakendusserverite  
operatsioonisüsteemi paikamiseks suuretevõtte  
näitel**

Bakalaureusetöö

Juhendaja: Reimo Lepp  
Teadusmagister

Tallinn 2022

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Tanel Saar

16.05.2022

## **Annotatsioon**

Antud bakalaureusetöö eesmärk on kasutusele võtta lahendus, mille abil saab automatiseerida domeeniväliste Windows rakendusserverite operatsioonisüsteemi paikamise protsessi.

Hetkel kasutusel olev paikamise protsessi puudumine tekitab olukorra, kus administraator peab suurt hulka käsitööd tegema, mis pole kuluefektiivne, kuna tuleb ületundide näol töötajatele rohkem palka maksta. Suure hulga käsitööga võivad lisaks tekkida inimvead.

Töö teoreetilises osas tuuakse välja ja kaardistatakse ettevõttes praegu kasutusel oleva domeeniväliste serverite paikamislahenduse puudused ja nende olulisus. Viiakse läbi küsitlus, kasutades autori isiklike kontakte, kes tegelevad antud valdkonnas. Luuakse küsitluse abil esmane valim ja analüüsitakse valimis olevaid lahendusi.

Töö praktilises osas viiakse analüüsi käigus kolme parima lahendusega testid ja tehakse järeldused. Peale testimise järeldusi võetakse uus lahendus kasutusele ja räägitakse selle juurutamisest.

Lõputöö tulemusena juurutati keskse konfiguratsioonihaldus andmebaasis oleval infol baseeruv Windows rakendusserverite operatsioonisüsteemi jaoks automaatne paikamislahendus.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 33 leheküljel, 8 peatükki, 22 joonist, 4 tabelit.

## **Abstract**

### **Creating an Automated Solution for Administrative Changes on the Example of Windows Application Server Patching**

The purpose of this bachelor's thesis is to implement a solution that can be used to automate the process of patching the operating system of non-domain Windows application servers.

The current lack of a patching process causes the administrator to have a lot of manual work, which is not cost-effective. A large amount of manual administration can cause human error. In addition, employees have to be paid more in the form of overtime.

In the theoretical part of the thesis, the shortcomings of the current non-domain application server patching solution used in the company and its importance are highlighted and mapped. A survey will be conducted using the author's personal contacts in the field of server patching. A primary sample for to-be solutions is created through the survey and the solutions in the sample are analyzed.

In the practical part of the thesis, the three best solutions from the sample are tested and conclusions are drawn. After the conclusions of the testing, a new solution is implemented.

The dissertation introduced an automatic patching solution for the Windows application server operating system based on the information in the central configuration management database.

The thesis is in estonian and contains 33 pages of text, 8 chapters, 22 figures, 4 tables.

## Lühendite ja mõistete sõnastik

*nix	UNIX-i laadsed operatsioonisüsteemid
Agent	Programm, mis vahendab teist programmi
Agentless	Lahendus, mis ei kasuta agent-e
CMDB	Configuration Management Database
Critical ja security paigad	Kriitilise- ja turva taseme paigad
DMZ	Demilitarized Zone, füüsiline ja/või loogiline alamvõrk, mis-eraldab usaldatava võrgu ebausaldatavast
Domeeniväline server	Server, mis ei ole lisatud Microsoft-i aktiivkataloogi
Hüperviisor	Funktsionaalüksus, mis loob ja käitab virtuaalmasinaid
IIS	Internet Information Services, Microsoft-i poolt arendatud veebiserver
ITIL	Information Technology Infrastructure Library, Infotehnoloogia taristu teek
ITPA	Information Technology Process Automation, Protsesside seeria, mis hõlbustab protsesside korraldamist ja integreerimist automatiseeritud töövoogude kaudu
Jinja2	Mallide loomise tarkvara
PaaS	Platform as a Service, Platvorm teenusena
Playbook	Automatiseerimisülesannete plaan
PoC	Proof of Concept, Prototüüp
Pre- ja post skript	Mingile tegevusele eelnevad ja järgnevad skriptid
RPC	Remote Procedure Call, kaugprotseduurikutse
SCCM	System Center Configuration Manager
SLA	Service Level Agreement, Teenustaseme lepe
WinRM	Windows Remote Management, Windows operatsioonisüsteemi kaughaldusprotokoll
WMI	Windows Management Instrumentation, Windows operatsioonisüsteemi andme- ja haldusinfrastruktuur
WSUS	Windows Server Update Services, Windows operatsioonisüsteemide paikade haldusliides
Autodiscovery	Automaatleidmine, olemite automaatse avastamise funktsioon
Runbook	Azure Automation-i jaoks loodud PowerShell skriptide töövoogude segu

# Sisukord

1 Sissejuhatus .....	11
1.1 Teema ajakohasus.....	11
1.2 Paikamise hetkeseis maailmas.....	12
1.3 Teadaolevad probleemid paikamise valdkonnas .....	12
1.4 Lõputöö eesmärk .....	13
1.5 Lõputöö metoodika.....	13
1.6 Lõputöö struktuur .....	14
2 Eesti Energia AS rakendusserverite paikamise protsess .....	15
2.1 Ettevõtte Eesti Energia AS lühikirjeldus .....	15
2.2 Domeeniväliste serverite paikamine.....	16
2.3 Domeenisestest serverite paikamine.....	16
3 Probleemi püstitus .....	18
3.1 Eesti Energia AS nõuded uuele lahendusele .....	18
4 Paikamise tarkvarade valimi koostamine .....	21
4.1 Küsitlused ettevõtete IT ekspertidega paikamise teemal.....	21
4.1.1 Ehituse valdkonna järelused .....	22
4.1.2 Telekomide valdkonna järelused.....	22
4.1.3 Avaliku sektori järelused .....	23
4.1.4 Küsitluste kokkuvõte .....	23
4.2 Lõplik paikamishalduse lahenduste valim.....	24
5 Lahenduste võrdlused .....	25
5.1 SolarWinds Patch Manager .....	25
5.2 Microsoft System Center Configuration Manager .....	26
5.3 Ansible konfiguratsioonihalduse tööriist.....	28
5.4 Azure Automation Update Management.....	30
5.5 PowerShell skriptimiskeel .....	31
6 Lahenduse valimine.....	33
6.1 Võrdlustabel esmase valimi kohta .....	33
6.2 Valimis osalenud lahenduste hindamine .....	35
7 Lahenduse kasutuselevõtmine .....	37
7.1 Arhitektuuriline ülevaade .....	37

7.2 Ansible serveri loomine .....	37
7.3 Igakuine inventory ja cron tööde loomise töö .....	38
7.3.1 Inventory loomise töö.....	38
7.3.2 Cron tööde loomise automatiseerimine .....	40
7.4 Rakendusserverite ette valmistamine lahenduse jaoks .....	40
7.4.1 WinRM seadistamine serveril .....	40
7.4.2 Mandaadid .....	41
7.5 Paikamise Ansible roll.....	41
7.5.1 WSUS serveri määramine .....	41
7.5.2 Serverite hooldusesse määramine monitooringusüsteemis .....	42
7.5.3 Paikade alla laadimine ja paigaldamine .....	42
7.5.4 Paikamise lõplik playbook .....	42
7.6 Töövoo testimine .....	43
7.7 Järeldused .....	43
8 Kokkuvõte .....	44
Kasutatud kirjandus .....	45
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	49
Lisa 2 – Valimis osalenud lahenduste hindamismatriks .....	50
Lisa 3 Näide Ansible töö abil genereeritud inventory failist.....	51
Lisa 4 Cron tööde loomise Ansible töö, Pythoni skript ja genereeritud crontab fail .....	52
Lisa 5 – Paroolihoidlast mandaatide pärimine ja nende kasutamine Ansible-s .....	55
Lisa 6 – Serveri paikamise rollid ja tööd .....	56
Lisa 7 – PoC-id ja järeldused kolme parima lahenduse kohta.....	58
Lisa 8 – Nõuete täitmise hindamine .....	60

## Tabelite loetelu

Joonis 1 Domeeni liidetud serverite paikamise protsess .....	16
Joonis 2 Paikamislahenduse tarkvara nõuded koos osakaaludega .....	20
Joonis 3 SolarWinds Patch Manager haldusliidese vaade ( <a href="https://www.solarwinds.com/fr/patch-manager/use-cases/remote-patch-management,2017">https://www.solarwinds.com/fr/patch-manager/use-cases/remote-patch-management,2017</a> ).....	25
Joonis 4 Kuvatõmmis SCCM haldusliidest (kuvatõmmis on võetud autori isiklikust kogumist).....	27
Joonis 5 Kuvatõmmis Ansible playbook-i koodilõigust ( <a href="https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html">https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html</a> ).....	29
Joonis 6 Azure Automation Update Management haldusliidese vaade ( <a href="https://docs.microsoft.com/th-th/azure/automation/update-management/view-update-assessments">https://docs.microsoft.com/th-th/azure/automation/update-management/view-update-assessments</a> ) .....	30
Joonis 7 Ansible paikamislahenduse arhitektuuriline ülevaade (kuvatõmmis on võetud autori isiklikust kogumist).....	37
Joonis 8 Ettevõtte CMDB, Jira Insight-i päringulause .....	38
Joonis 9 Ansible inventory faili loomise töö.....	39
Joonis 10 Inventory faili loomise Jinja2 mall.....	39
Joonis 11 WinRM HTTPS sertifikaaditaotluse sisu .....	40
Joonis 12 Rakendusserverite paikamise playbook .....	42
Joonis 13 Ansible töö abil genereeritud inventory faili sisu (kuvatõmmis on võetud autori isiklikust kogumist).....	51
Joonis 14 Crontab faili loomise Ansible töö .....	52
Joonis 15 Crontab faili loomise Python skript .....	53
Joonis 16 Genereeritud crontab faili sisu (kuvatõmmis on võetud autori isiklikust kogumist).....	54
Joonis 17 Paroolihoidlast mandaatide pärimine ja muutujatesse kirjutamine.....	55
Joonis 18 Ühenduse loomise näide Ansible dünaamilise muutuja abil.....	56
Joonis 19 WSUS serveri määramine rakendusserveritele .....	56
Joonis 20 Rakendusserveri hooldustöö staatuse määramine monitooringusüsteemis....	57



Joonis 21 Rakendusserveri turvapaikamise Ansible tööd .....	57
Joonis 22 Azure Automation Update Manager võrgutopoloogia kohaliku andmekeskuse serverite jaoks ( <a href="https://docs.microsoft.com/en-us/azure/azure-monitor/agents/log-analytics-agent#installation-options">https://docs.microsoft.com/en-us/azure/azure-monitor/agents/log-analytics-agent#installation-options</a> ).....	59

## Jooniste loetelu

Tabel 1 Paikamislahenduste võrdlus 1/2 .....	34
Tabel 2 Paikamislahenduste võrdlus 2/2 .....	35
Tabel 3 Valimis osalenud lahenduste hindamistabel 1/2 .....	50
Tabel 4 Valimis osalenud lahenduste hindamistabel 2/2 .....	50

# 1 Sissejuhatus

Tänapäeva IT-infrastruktuur on keerulisem kui kunagi varem ja selle keerukus ainult kasvab ajaga. Kasutusele võetakse rohkelt rakendusi, erinevaid servereid, nii füüsilisi, kui ka virtuaalseid ja see tekitab halduse poole pealt suure probleemi – kuidas seda kõike ära hallata? Ettevõtted ei saa endale lubada seda probleemi pidevalt inimeste juurde värbamisega lahendada, vaid peaksid leidma viisi, kuidas seda kõike tõhusamalt juhtida. Organisatsioonidel on sageli probleeme dokumenteerimata või aegunud protsessidega, mille tulemusel võivad töötajad muudatusi sisse viies hoopis kahju tekitada.

## 1.1 Teema ajakohasus

Perioodilised haldusprotseduurid on üksluised ja aeganõudvad, kui neid tehakse käsitsi. Paikamistega seotud tööülesanded on tavaliselt planeeritud, hästi mõistetava protsessiga ja annavad etteaimatava tulemuse ja seega langevad samuti üksluisete ja aeganõudvate haldusprotseduuride alla. Just seda tüüpi tööülesanded sobivad ITIL-i järgi automatiseerimiseks [1]. Kuna paikamisi viiakse läbi suure hulga serveritega, siis on see käsitöö näol väga aeganõudev protsess, mille automatiseerimisel oleks mitmeid plusse. Üheks suurimaks plussiks oleks ületundide näol raha säästmine. Suuremat hulka servereid, mis mingeid teenuseid enda peal jooksutavad, ei saa tööpäeva sees paigata, kuna need peavad olema töös. Seega tuleb neid teha töövälisel ajal, millal Eestis kehtiva Töölepingu seaduse (2009), § 44 lg 7 sõnastusest lähtuvalt tuleb töötajale maksta 1,5-kordset töötasu. Arvestades, et serverite hulk on suur ja paikamised võivad aega võtta mitu tundi, siis on see üpriski suur kulu. Raha säästmine võibolla ei ole tähtis selleks, et enda kasumit suurendada, vaid et seda saaks investeerida muul moel – näiteks uute inimeste värbamine ja panustamine enda töötajate arengusse ja kasvatada kompetentsi koolituste näol.

## 1.2 Paikamise hetkeseis maailmas

Paigad on tarkvara ja operatsioonisüsteemi värskendused, mis kõrvaldavad toodetega seotud teatud probleeme ja turvanõrkusi või annavad tootele juurde lisafunktsionaalsusi.

Kui vaadata minevikku, siis paikade haldamine ei olnud küberturvalisuse probleem, vaid pigem oli see IT-probleem. Alles peale ühe esimese suurema pahatahtliku viiruse, Code Red-i [2] ilmumist 2001. aastal hakkas Microsoft välja andma paikasad, et parandada oma tarkvara turvaauke. Sellised sarnased suuremad juhtumid panid aluse regulaarsete paikamishalduse tsüklite laialdaseks kasutuselevõtuks ettevõtete ja tarkvaratarnijate seas. Enne regulaarseid haldustsükleid esinesid peamiselt juhuslikud ja reaktiivsed parandused [3].

Kui rääkida tänapäeval paikamisest, siis on tegu üpriski standardse tegevusega, mida on võimalik väga erinevatel viisidel läbi viia. Turul on pakkuda palju kommertstarkvara, mis teevad paikamise üpriski lihtsaks. Kahjuks on liigne valik ka üheks tarkvara valimisel negatiivseks küljeks. Ühe ja sama asja lahendamiseks on liiga palju erinevaid tööriistaid ja selle raames tekib valiku ülekoormus ja võib põhjustada otsuste – isegi oluliste – vastuvõtmise edasilükkamist, sest meile pakutavate paljude valikute arvestamine koormab liialt meie kognitiivset mõtlemist [4]. Lisaks on ka erinevaid tasuta lahendusi turul, mis võibolla funktsionaalsuste poole pealt ei paku nii palju, kui tasulised tarkvarad, kuid on täiesti töövõimelised.

Üha enam soovitakse liikuda PaaS mudeli poole, kus antakse haldus üle tagasi teenusepakkuja kätte [5]. Antud mudeli poole tüürides võtavad aina enam ettevõtteid kasutusele erinevaid pilveteenusetarnijaid ja migreerivad enda infrastruktuuri pilve ja seda arvestades võib järeltada, et tulevikutrend on pilv [6]. Kuna PaaS mudelis on ka operatsioonisüsteemi tase teenusepakkuja halduses, siis vastutab ta ka operatsioonisüsteemi ja hüperviisori paikamiste eest, mis teeb administraatoril töö praegusest oluliselt lihtsamaks [7].

## 1.3 Teadaolevad probleemid paikamise valdkonnas

Paikade paigaldamine võib olla üks põhi turvalisuse aspektidest, kuid see ei tähenda, et seda oleks praktikas alati lihtne teha. Paikamine võtab aega, võib olla korduv ja tüütu töö kui seda teha käsitsi. Ideaalis peaks paikamishalduse protsess olema ettevõttes hästi

dokumenteeritud ja vähemalt mingil määral automatiseeritud. Eriti tähtis on teada oma süsteemide kohta ajakohast infot, et selle kaudu oleks võimalik turvariskidele reageerida ja skoope hoomata. Lisaks, ideaalis võiks enne nende täielikku paikade paigaldust need läbi testida, mis võib aidata võimalike probleemide avastamisel [8].

Kahjuks nõuab kõik ülaltoodu juurutamine päris suurt hulka raha nii töötundide kui ka tarkvara poole pealt, mille pärast kiputakse lihtsama vastupanu teed minema ja sellega mitte tegelema. Töötundide koha pealt on suureks probleemiks ka töötajate enda mugavustsoon. Nimelt ei taheta olemasolevaid lahendusi ja protsesse muuta, isegi, kui need pole ajakohased, kuna kardetakse, et miski praeguses lahenduses läheb katki ja uus ei tööta [9].

## **1.4 Lõputöö eesmärk**

Antud bakalaureusetöö eesmärk on tõsta domeeniväliste rakendusserverite turvalisuse baashügieeni läbi uue töövahendi, ning selle raames ka teha administraatorite tööd lihtsamaks ja kuluefektiivsemaks. Selle teostamiseks tuleb välja tuua ja kaardistada ettevõttes praegu kasutusel oleva domeeniväliste serverite paikamislahenduse puudused ja nende olulisus, mille raames saab määrata uuele lahendusele nõuded. Töö praktilises osas soovitakse erinevate lahenduste vastavaust analüüsis välja kujunenud nõuetele hinnata ja parim lahendus reaalselt kasutusele võtta.

## **1.5 Lõputöö metoodika**

Lõputöö eesmärgi täitmiseks teostatakse analüüs, mille abil kirjeldatakse ettevõtte uue juurutatava lahenduse jaoks. Tehakse küsitlus teiste ettevõtete spetsialistidega, et koostada uute lahenduste valim. Analüüsitakse valimis olevaid lahendusi, kas ja kui hästi need vastavad töös paika pandud nõuetele. Analüüsi käigus kuulutatakse välja kolm parimat lahendust, millega tehakse katsetused, kus testitakse praktiliselt nõudeid ettevõtte kontekstis. Peale testimisi valitakse välja parim, mille abil realiseeritakse uus paikamislahendus.

## 1.6 Lõputöö struktuur

Bakalaureusetöö on jaotatud kaheksaks peatükiks. Esimeses peatükis antakse sissejuhatav ülevaade lõputöös käsitletavast valdkonnast, eesmärgist ja metoodikast. Teises peatükis tutvustatakse hetkel kasutusel olevaid Eesti Energia AS rakendusserverite paikamise protsesse. Kolmandas peatükis püstitatakse käsitletav probleem ja räägitakse lähemalt ettevõtte nõuetest uuele lahendusele. Neljandas peatükis viiakse läbi teiste ettevõtete spetsialistidega küsitlus, mille raames koostatakse lõputöös käsitletavate, uute lahenduste valim. Viendas peatükis viiakse läbi võrdlused ja analüüsid koostatud valimis olevate lahendustega. Kuuendas peatüki raames viiakse läbi võrdluse käigus parima kolme lahendusega praktilised katsetused, ning nende seast valitakse välja parim lahendus. Seitsmendas peatükis kirjeldatakse parima lahenduse kasutuselevõttu. Kaheksandas peatükis võetakse töö kokku ja räägitakse saavutatud tulemustest.

Lõputöö põhilised allikad on erinevad erialased artiklid ja internetiväljaanded. Lisaks allikatena on ka kasutatud küsitluse raames teiste eriala spetsialistide teadmisi.

## **2 Eesti Energia AS rakendusserverite paikamise protsess**

Käesolevas peatükis annab autor ülevaate Eesti Energia AS-st ja lahendustest, mida praeguseni on kasutatud. Kuna töö keskendub domeeniväliste serverite skoobis paikamise probleemi lahendamisele, mille haldus on ettevõttes käsitöö näol lahendatud, räägib ka autor lisaks domeeni liidetud serverite paikamisest. Infoturbe põhjustel on autor kohustatud protsessist suhteliselt pealiskaudselt rääkima.

### **2.1 Ettevõtte Eesti Energia AS lühikirjeldus**

Eesti Energia AS on Eesti riigile kuuluv energiaettevõtte, mis tegutseb elektri- ja gaasimüügiga ning rahvusvahelisel vedelkütuste turul. Lisaks Eestile tegeleb ka Eesti Energia teiste Baltikumi riikide, Soome, Rootsi ja Poola turgudel [10]. Kontserni kuuluvad suured ettevõtted nagu Enefit Connect, mis haldab erinevaid elektrivõrke, kiire interneti võrku, kaasaegset elektriautode laadimisvõrku ning suurt osa Eesti tänavavalgustusest [11], Enefit Power, mis osutab hooldusteenuseid ja tehnoloogilisi lahendusi kontserni elektri jaamadele, õlitehastele ja kaevandustele, [12] ja Elektrilevi, mis on Eesti suurim võrguettevõtte ja ehitab ning haldab peaaegu kogu Eesti elektrivõrku [13]. Arvestades antud ettevõtte ja kontserni suurust võib mõista, et tegu on elutähtsa teenusepakkujaga Eestis, mille IT osakond ja tema vastutus võivad olla suhteliselt suured. Autor töötab ise ettevõttes IT halduse osakonnas, IT platvormide valdkonnas, mis on jaotatud kolmeks alamtiimiks:

- Windows serverite eest vastutav tiim
- \*nix serverite eest vastutav tiim
- Andmebaasi administraatorite tiim

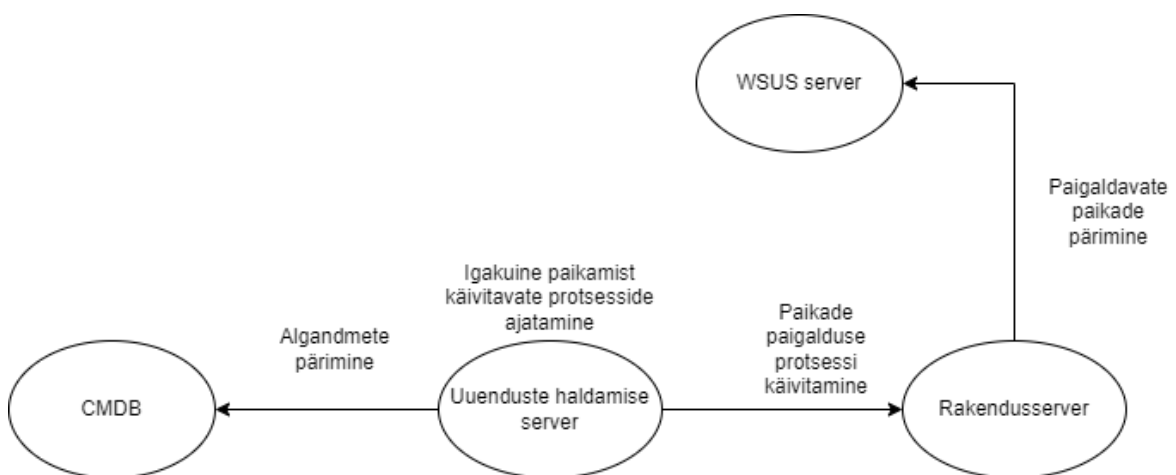
Autori alamtiim on Windows serverite eest vastutav tiim.

## 2.2 Domeeniväliste serverite paikamine

Administraator peab kasutama keskset algandmete infoallikat, mille põhjal servereid paigatakse. Selleks on ettevõttes kasutusel CMDB, vara- ja konfiguratsioonihaldustarkvara Jira Insight [14], kus on ära kirjeldatud vajalik info serverite kohta. Kõige tähtsam serveri kohta käiv info on tema hooldusakna aeg. Hooldusaknad on äripooltega kokku lepitud ja määratud, kuna iga teenuse tööd ei ole võimalik suvalisel ajal katkestada tema tähtsuse või töö tõttu. Administraator peab olema valmis just õigel ajal paikamist käivitama, et mahtuda ära ajaaknasse. Kuna tegu on domeenivälise serveriga, peab administraator ka kasutama ajakohaseid mandaate. Iga serveri kohta on mandaadid kirjas keskses paroolihoidlas, mis roteerib paroole turvalisuse pärast. Peale paikade paigalduse käivitamist tuleb hoida silma peal, et paigaldused läksid korrektselt läbi, mille järgselt saab taastada rakenduste töö serveril.

## 2.3 Domeenisiseste serverite paikamine

Ettevõttes on suurusjärgus 500 Windows operatsioonisüsteemiga serverit, millest umbes 450 on domeeni liidetud. Kindlasti ei ole mõttekas antud suurusjärgus servereid käsitsi paigata ja seega on juurutatud ettevõttele kohandatud lahendus nende serverite automaatseks paikamiseks. Praegune protsess koosneb peamiselt PowerShell skriptidest (vt. joonis 1.).



Joonis 1 Domeeni liidetud serverite paikamise protsess

Paikade paigalduse käivitamiseks on kasutusele võetud üks kindel server, mille kaudu viiakse läbi haldustegevused. Iga kuu käivitatakse haldusmuudatusi läbiviival serveril skript, mis pärib infot CMDB-st. CMDB-s on kajastatud erinevate serverite kohta info



tema võrgu, domeeni, operatsioonisüsteemi versiooni ja kõige tähtsama – tema hooldusakna kohta. Sealt päritud andmete põhjal loob igakuine skript iga plaanitud tööde ajaakna kohta uue ajastatud töö. Kui õige aeg kätte jõuab, käivitab haldusmuudatuste server vastu rakendusserverit käsu, mille abil käivitatakse paikamine. Paikade enda info pärimiseks kasutab rakendusserver eelkonfigureeritud WSUS serverit. WSUS on Microsoft poolt pakutav tasuta lisarakendus, mille kaudu saab alla laadida ja hallata Windows Serveri operatsioonisüsteemide värskendusi ja paikasid. [15]. Juhul, kui rakendusserverile on lisatud funktsionaalsus lisaskripti näol, on paikade paigalduse käivitamise skripti abil võimalik ka hallata serveril oleva rakenduse tööd, et seda näiteks ilma vigadeta sulgeda ja käivitada saaks.

### **3 Probleemi püstitus**

Automatiseerimisel on juba praegu märkimisväärne mõju sellele, kuidas ettevõtted oma tööd läbi viivad. Parimad ülesanded automatiseerimiseks on need, mis on korduvad ja inimtöötajate jaoks igavad või tüütud. Automaatikaid läbi viivatel skriptidel või rakendustel ei hakka igav – nad suudavad kiiresti ja täpselt läbi töödelda suurt andmehulka, mille läbimiseks võib inimesel kuluda terve või isegi mitu tööpäeva [16].

Ettevõttes praegu kasutusel olev protsess ei ole esiteks üldse kuluefektiivne, kuna peame maksma inimestele töövälise aja eest lisapalka. Rakendusserverite paikamisi ei saa ka päevastele aegadele tõsta, kuna rakendused ise näevad tööpäeva sees kasutust ning SLA-d ei paku piisavalt suurt ajaakent, et saaks sellel ajal võimalikult suure katkestuse teha. Isegi, kui leiaks mitme-tunnise ajaakna ja suudaks äripoollega kokku leppida katkestuse tööpäeva sees, siis jõuame teise suure probleemini. Paikamiste õigeaegne käivitamine nõuab ettevalmistust ja sellega raiskame administraatori tööaega, mille ajal oleks võimalik tal erinevate teiste arendustöödega tegeleda. Lisaks kõigele, võib ka probleeme tekitada administraatori enda tähelepanematus - võib tekkida olukord, kus paikamisi käivitatakse vales ajal ja sellega kaasneb rakenduse planeerimata katkestus.

Töö eesmärgiks on leida lahendus, mis suudaks vastavalt välja toodud nõuetele (v.t peatükk 3.1) skoobis olevaid servereid paigata ja luua automatiseeritud töövoog või protsess, mis säästaks administraatori väärtuslikku tööaega, suudaks vältida inimlikke lohaku vigu ja lisaks ületundide näol ettevõttele raha säästa.

#### **3.1 Eesti Energia AS nõuded uuele lahendusele**

Käesolevas peatükis on välja toodud lahenduse tellija poolsed nõuded. Autor tegi lõputöö vältel koostööd ettevõtte rakendusservereid haldava tiimiga ja infoturbe osakonnaga, et paika panna lahenduse jaoks vajalikud nõuded.

Allolevas loendis on ettevõtte poolt välja toodud nõuete seletused, koos nõuete kaaludega. Igale lahendusele määratakse nõude hinnang viie palli süsteemis, mis

korrutatakse läbi nõude kaaluga. Mida rohkem punkte lahendusele vastavalt nõuetele määratakse, seda paremaga on tegu ettevõtte silmis. Nõude hinnangu määramisel on kasutatud ettevõtte poolt määratud ülem- ja alam määrade seletused (v.t lisa 8).

- Kuna eelarve vahendid on antud lõputöö raames piiratud, on eelistatud lahendused, millega kaasnevad võimalikud väikesed rahalised kulud.
- Turvalisuse all antud töö kontekstis on mõeldud, et paikamise paigaldust juhtiv protsess peab rakendusserveritega ühenduma üle turvalise kanali. Turvaline kanal on tähtis, kuna töös käsitletakse domeeniväliseid servereid, mis võivad omada ühendust välismaailmaga ja seega võivad olla infoturbe riskiallikeks.
- *CMDB* kesksus ehk võimalus kasutada ettevõttes kasutusel olevat *CMDB*-d, Jira Insight-i [17] paikamise protsessis keskse infoallikana. *CMDB* kasutamine protsessis on tähtis, kuna see pakuks andmete hoidla näol täpset ja täielikku informatsiooni serverite kohta.
- Protsessi automatiseerimise võimalus ehk tarkvara võimaldaks paikade paigaldamisi ja sellega kaasnevat tegevusi automatiseerida. See säästaks administraatori töö aega, kuna suurem osa paikamisi toimub töövälisel ajal, kui serveritel olevad rakendused ei nõua palju kasutust.
- Teenuste haldus ehk võimalus teenustega seotud tegevusi põimida protsessiga. Paljud rakendused vajavad, et neid sulgetaks kindlal viisil ja nende ebakorrektnel sulgemine võib rikkuda nende edasist tööd. Seega on vajadus, et paikamishalduse tööriist võimaldaks ka automatiseerimisprotsessi lisada rakendustega seotud sulgemis või käivitamise tegevusi.
- Lahenduse lihtsus ehk kui kompleksne tundub lahenduse juurutamine ja haldamine ettevõttes. Paljud tarkvarad on enda loomult väga suure õpikõveraga ja nõuavad spetsialistilt enne pikemat kogemust, et neid saaks asjalikult igapäevases töös kasutada. Kuna lihtsama lahenduse kasutuselevõtmine säästaks administraatori tööaega, oleks see ettevõttele veelgi kuluefektiivsem. Nõude hindamisel võetakse arvesse ka juurutamise ja edasiarenduse võimalusi. Nimelt ei taheta, et ettevõtte administraator oleks lahenduse ainus edasi arendaja.

Nõuded	Punktid
Lahenduse maksumus	5
Turvalisus	5
Võimalus integreeruda <i>CMDB</i> -ga	4
Protsessi automatiseerimise võimalus	4
Rakenduste halduse võimalus	3
Lahenduse lihtsus	3

Joonis 2 Paikamislahenduse tarkvara nõuded koos osakaaludega

## **4 Paikamise tarkvarade valimi koostamine**

Käesolevas peatükis toob autor välja erinevad tarkvarad ja miks need valimisse sattusid. Peatükk koosneb kahest osast, millest esimeses tuuakse välja erinevate ettevõtete IT ekspertide küsitlused ja nende järeldused ja teises tuuakse välja küsitluse põhjal valimisse sattunud tarkvarade kohta kokkuvõte.

### **4.1 Küsitlused ettevõtete IT ekspertidega paikamise teemal**

Lõputöö raames viis autor läbi erinevate Eesti ettevõtete IT ekspertidega küsitluse, et võtta arvesse nende kogemused ja arvamused erinevate paikamislahenduste kohta. Küsitlust sooviti läbi viia 7 ettevõtte eksperdiga, millest 6-ga sai reaalselt läbi viidud. Peamiselt olid küsitletavad antud valdkonna spetsialistid ja pikema kogemusega süsteemiadministraatorid. Lisaks olemasolevate lahenduste kohta küsimisele sooviti ka erinevate lahenduste kohta infot pärida, mis aja vältel on ettevõttes kasutusel olnud või proovitud, aga ei osutunud lõplikuks lahenduseks. Ettevõtete valdkonnad on üldistatud ja kuna antud lõputöö on avalik, on ettevõtete ja ekspertide nimed anonümiseeritud.

Lõputöö autori poolt on asutused jagatud kolme kategooriasse:

- Ehituse valdkond
- Telekomi valdkond
- Riikliku taustaga ettevõtte

Küsimused, mida ekspertide käest küsiti:

- Mis paikamislahendus ettevõttes kasutusel on?
  - Mis on praeguse lahenduse plussid ja miinused?
- Mis nõuded on ettevõttel enda lahendusele?
- Milliseid alternatiivseid lahendusi on ettevõttes kaalutud?

- Põhjus, miks kaalutud lahendused ei osutunud valituks?
- Kas ettevõttes on kasutusel *CMDB*?
  - Juhul kui on, siis kas ja kuidas on see põimitud paikamise protsessiga?
  - Juhul kui ei ole, siis kuidas on vajalik informatsioon paikamise kohta dokumenteeritud?
- Kas ettevõttes on paikamise protsess automatiseeritud?
  - Kas ja kuidas automaatika on põimitud *CMDB* (eeldusel et see on kasutusel) andmetega?
- Kui protsess ei ole automatiseeritud, siis miks?
  - Kas ja miks mingis skoobis olevaid servereid paigatakse käsitsi?

#### **4.1.1 Ehituse valdkonna järeldused**

Ehituse valdkonna ettevõtte kohal ei toodud väga kindlaid põhjuseid, miks just antud paikamislahendus kasutusel on. Lisaks ei toodud ka välja plusse, mille najale saaks toetuda. Peamiselt tehti paikamisi käsitsi, kuna ei soovitud olemasolevat protsessi ümber teha. Serverid saavad info paikade ja värskenduste kohta WSUS-st, mis on tasuta tarkvara ja ei paku erilisi automatiseerimise funktsionaalsusi. Serverite endi kohta üritatakse hoida ajakohast infot käsitsi muudetavates dokumentides, mis viitavad igasugusele automaatika puudumisele. Ettevõttes on kaalutud ka erinevaid tasulisi lahendusi, kuid raha säästmise põhjusel ei ole neid kasutusele võetud.

Antud valdkonna kohta erilisi järeldusi teha ei saanud, kuna käsitsi paikamisest soovib ettevõtte antud momendil lahti saada. Vanad ja „kindlad“ protsessid ei õigusta käsitööd ja antud lähenemine ei sobi autori ettevõttele alampeatükis 1.1 välja toodud põhjustel.

#### **4.1.2 Telekomi valdkonna järeldused**

Erinevalt ehituse valdkonnale olid telekomi valdkonnas suhteliselt kindlad nõuded kasutatud lahendustele. Peamine mittefunktsionaalne nõue oli, et oleks tegu turul laialdaselt kasutatava ja tuntud tehnoloogilise platvormiga. Mõlemas küsitletud ettevõttes oli protsess automatiseeritud ja kasutati *CMDB*-d baasinfoallikana. Funktsionaalsete

nõuete koha pealt toodi välja, et tähtis oleks, et lahendus saaks *CMDB* kaudu kasutada kindlaid määratud ajaaknaid automaatikas ja et oleks võimalik uuendusi kas lubada või keelata vastavalt vajadusele. Kindluse pärast ei ole kõik serverid automaatse paikamisega kaetud ja eri nõudmistega lahendusi, nagu andmebaasi ja mailiservereid, paigatakse käsitsi, kuna need nõuavad rakendustes käsitsi sekkumist.

Ühes ettevõttes sobis ülaltoodud nõuetele Microsoft-i enda poolt pakutav Azure Automation Update Management tööriist, millega on võimalik nii pilves kui ka kohalikus andmekeskuses olevaid servereid hallata. Teises telekomi valdkonna ettevõttes oli antud nõuded lahendatud SolarWinds Patch Manager abil. Lisaks võib ka välja tuua, et mõlemas ettevõttes oli proovitud ka sarnaselt autori ettevõttega maja siseselt arendatud skriptide näol paikamisi läbi viia. Proovitud on ka Ansible konfiguratsioonihaldustööriista kogukondade poolt kirjutatud moodulite abil probleemi lahendada. Enda skriptid ja Ansible olid kasutusel, kuna neid on võimalik suhteliselt igasse süsteemi integreerida. Vaatamata sellele, põhjus, miks enda poolt kirjutatud skriptid või Ansible ei osutunud välja valituks oli, et need nõudsid kompetentsi omandamiseks, juurutamiseks ja edasiarenduseks liialt töötunde.

#### **4.1.3 Avaliku sektori järeldused**

Avaliku sektori küsitluste põhjal saadi peamiselt teada seda, et soovitakse Microsoft-i enda tööriistadega probleeme lahendada ja et lahendus oleks keskselt hallatav. Kaks ettevõtet kolmest tõid välja Microsoft System Center Configuration Manager ehk SCCM ja viimane käsitsi paikamise. Viimase ettevõtte põhjendus käsitsi paikamisele oli, et skoobis olevaid servereid on liiga vähe, et eritarkvara osta või kasutusele võtta. Ainuke pluss, mis käsitsi paikamisel välja toodi, oli, et siis on olemas kindlus, et paigad saavad paigaldatud.

#### **4.1.4 Küsitluste kokkuvõte**

Ettevõtetega küsitluste läbiviimise käigus sai töö autor laialdast infot erinevate Eestis kasutusel olevate lahenduste kohta. Autori jaoks tuli see üllatusena, et ka teistes suuremates ettevõtetes on veel suure hulga serverite paikamiseks kasutusel käsitöö. Autori jaoks ei olnud väljatoodud põhjendused piisavalt ammendavad, et kaaluda käsitöö edasi kasutamist ja selle lisamist valimi nimistusse. Lisaks saadi teada, et pigem ei soovita kolmanda osapoolle tarkvara kasutada ja usaldatakse Microsoft-i enda tooteid.

Autor soovis ettevõtetes kasutusel olevaid lahendusi just nende plusside või põhjenduste pärast valimisse lisada, isegi, kui need ei osutunud ettevõttel lõplikuks lahenduseks. Küsitluste põhjal lisati valimisse järgmised lahendused:

- SolarWinds Patch Management
- Microsoft System Center Configuration Manager (SCCM)
- Ansible konfiguratsioonihalduse tööriist
- Azure Automation Update Management

## **4.2 Lõplik paikamishalduse lahenduste valim**

Lisaks valitud tarkvaradele lisati valimisse ka ettevõttes hetkel kasutusel olev domeenisest serverite paikamiseks kasutusel olev tööriist PowerShell skriptimiskeel, et võrrelda, kas antud lahenduse edasi arendamine domeeniväliste serverite haldamiseks ennast õigustab. Käesoleva lõputöö raames on paikamishalduse tarkvarade lõplik valim järgmine:

- SolarWinds Patch Management
- Microsoft System Center Configuration Manager
- Ansible konfiguratsioonihalduse tööriist
- Azure Automation Update Management
- PowerShell skriptimiskeel

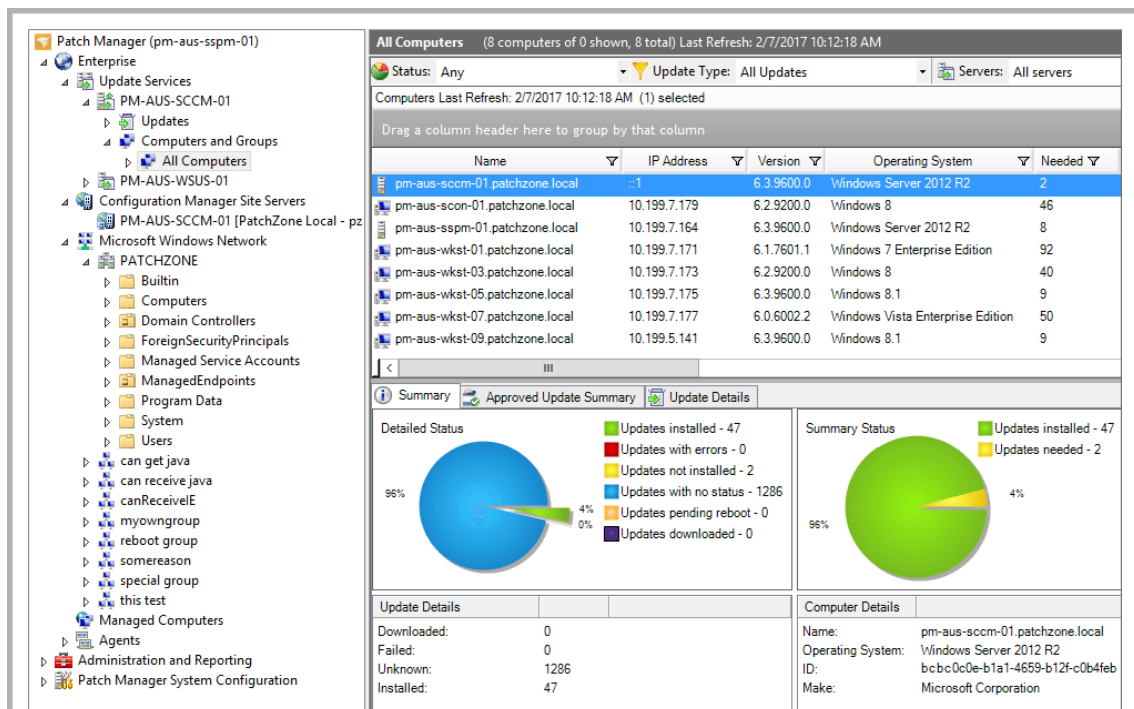


## 5 Lahenduste võrdlused

Käesolevas peatükis toob autor välja valimisse sattunud paikamishalduse tööriistade võrdlused. Iga alampeatükk keskendub ühele kindlale tööriistale, kus autor tutvustab erinevaid lahendusi ja nendega kaasnevaid plusse ja miinuseid.

### 5.1 SolarWinds Patch Manager

SolarWinds on Ameerikas asuv IT tarkvara arenduse ettevõtte. Peamiselt on nende poolt loodud tarkvarade abil võimalik hallata võrke ja muud infotehnoloogilist infrastruktuuri. SolarWinds-i peakorter asub Texases Austinis [18]. Ettevõtte kohta uurides jäi autoril silma, et 2020 aastal oli SolarWinds-i monitoringutarkvara, Orion, fookuseks küberturbe rünnakule, mis jättis Ameerika Ühendriikide valitsuse IT taristu ohustatuks vähemalt üheksaks kuuks [19]. Teades eeltoodut ja arvestades, et autori ettevõtte on samuti elutähtsa teenuse pakkuja, ei ole SolarWinds-i kaubamärk väga usaldusväärne. Peamiste funktsioonide, miinuste ja plusside kohta on autor punktid välja toonud allolevas loendis.



Joonis 3 SolarWinds Patch Manager haldusliidese vaade (<https://www.solarwinds.com/fr/patch-manager/use-cases/remote-patch-management>, 2017)

Peamised funktsioonid [20]:

- Intuiitiivne veebihaldusliides
- Paikade vastavusraportite genereerimise võimalus
- Tsentraliseeritud Windows serverite paikamise võimalus
- Võimalus paigata nii virtuaal- kui ka füüsilisi servereid
- Laialt levinud ja testitud kolmanda osapoole tarkvarade paikamise võimalus
- Kohandatud paikamispakkide võimekus ehk on võimalik serveritele paigaldada ka koos operatsioonisüsteemi paikadega erinevaid rakendusi.

Plussid:

- Tegu suure ettevõttega, kes pakub mitmeid erinevaid tooteid
- Kvaliteetne kasutajatugi
- Paikade raportite abil on võimalik hoida enda süsteemide üle ajakohast infot

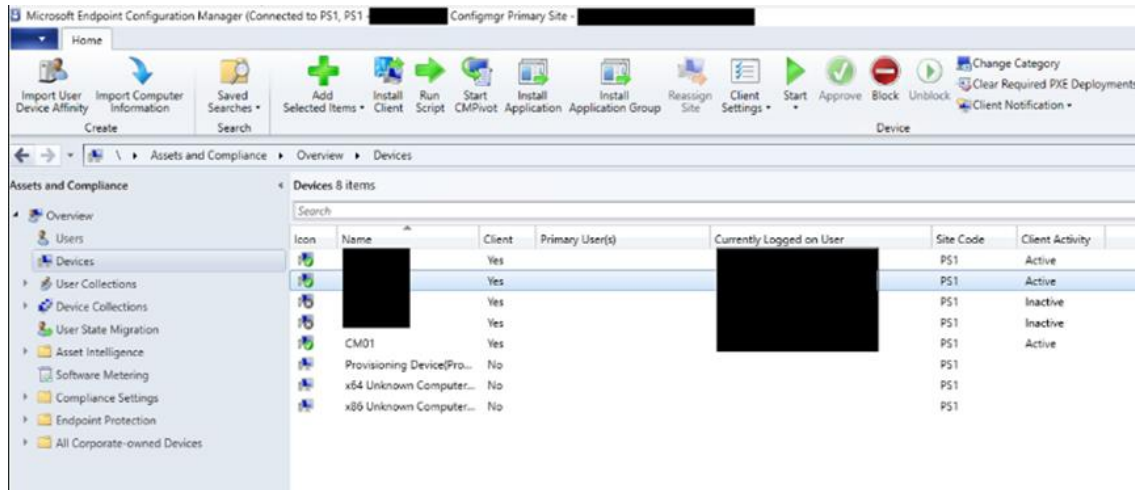
Miinused:

- Üles seadmine võtab palju aega
- Kasutamine võib olla alguses keerukas
- Maksumus

## **5.2 Microsoft System Center Configuration Manager**

Microsoft on suurim arvutitarkvara pakkuja maailmas. Samuti on tegu juhtiva pilveteenusetarnijaga, arvutiriistvara ja -mängude tootjaga. Microsofti ettevõtte peakorter asub Redmondis, Washingtonis ja omab kontoreid enam kui 60 riigis [21]. Microsoft on samuti antud töös käsitletava Windows Server brändiga operatsioonisüsteemide arendaja ja tarnija. Microsoft SCCM on integreeritud lahendus Windows operatsioonisüsteemiga

tööjaamade ja serverite halduseks, ning all toodud nimistus on tema peamised funktsioonid, plussid ja miinused.



Joonis 4 Kuvatõmmis SCCM haldusliidesest (kuvatõmmis on võetud autori isiklikust kogumist)

Peamised funktsioonid [22]:

- Võrgus olevate seadmete uuenduste ja turvapaikade paigaldamine
- Võrgus olevate seadmete inventeerimine
- Rakenduste paigaldamine seadmetele

Plussid:

- Integreerub sujuvalt Windowsi süsteemidega - Microsofti tootena integreerub SCCM väga hästi Windowsi süsteemide ja teiste Microsofti toodetega.
- Microsoft on regioonis üks suurimaid teenusepakkujaid ja seega on tegu usaldusväärse tootega
- SCCM-i juhitakse suhteliselt lihtsa haldusliidese kaudu, mis tähendab, et seda on lihtsam õppida ja rakendada kui teisi konfiguratsioonihaldus tööriistasid, nagu Chef ja Puppet.
- SCCM on kindlaskujunenud ja tasuline Microsofti teenus ja seega pakub ka head tuge kogukonna kanalite ja Microsofti enda kaudu

Miinused:

- Ehitatud peamiselt Windows süsteemide haldamiseks – piiratud haldusvõimalused Linux ja MacOS seadmetel
- Kõrged omandamise ja käitamise kulud – SCCM-i müüakse tavaliselt Microsofti suurema tarkvarakomplekti osana ja võib olla ülemäära kallis
- Kolmanda osapoole rakenduste uuendamine on piiratud
- Puudub pilves olevate seadmete haldustugi

### **5.3 Ansible konfiguratsioonihalduse tööriist**

Ansible on avatud lähtekoodiga haldusmuudatuste läbiviimiseks mõeldud tööriist. Red Hat, eduka mainega avatud lähtekoodiga tarkvara pakkuv ettevõtte omandas aastal 2015 Ansible [23]. Ansible on kirjutatud Python-i programmeerimiskeeles, mis on üks populaarsemaid, kui mitte kõige populaarsem hetkel maailmas ja seega näeb kogukonna näol head tuge [24]. Ansible abil on võimalik kirjutada valmis playbook-id, mis on automatiseerimisülesannete kavandid. Ansible playbook-e, mis käivitatakse hostirühmade pihta, mis moodustavad Ansible inventory.

```

---
- name: Update web servers
  hosts: webservers
  remote_user: root

  tasks:
  - name: Ensure apache is at the latest version
    ansible.builtin.yum:
      name: httpd
      state: latest
  - name: Write the apache config file
    ansible.builtin.template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf

- name: Update db servers
  hosts: databases
  remote_user: root

  tasks:
  - name: Ensure postgresql is at the latest version
    ansible.builtin.yum:
      name: postgresql
      state: latest
  - name: Ensure that postgresql is started
    ansible.builtin.service:
      name: postgresql
      state: started

```

Joonis 5 Kuvatõmmis Ansible playbook-i koodilõigust  
([https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_intro.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html))

Peamised funktsioonid [25]:

- Haldusmuudatuste automatiseerimine playbook-ide abil

Plussid

- Agentless – tööriist ei nõua haldamiseks süsteemidesse agentide paigaldamist, mis toob endaga kaasa vähem hoolduskulusid
- Suure kogukonna poolt arendatud avatud lähtekoodiga tööriist
- Omab kesket hoidlat kogukonna poolt kirjutatud moodulite jaoks
- Tasuta tarkvara

- Võimalik valmis kirjutatud koodi ja mooduleid taaskasutada
- Võimalik kirjutatud koodi versioonihalduses hoida, mille abil on hea ülevaade muudatustest
- Haldusvõimalustel pole piire
- Deklaratiivne ehk väljendatakse, mida programm peab saavutama, mitte kuidas ta seda saavutab

## Miinused

- Palju algset käsitööd moodulite ja playbook-ide kirjutamisel
- Graafilise kasutajaliidese puudumine
- Nõuab hallatavate serverite poolt eelseadistusi turvaliseks ühenduseks

## 5.4 Azure Automation Update Management

Sarnaselt eelnevalt välja toodud tarkvara SCCM-ga, on Azure Automation Update Management suurkorporatsiooni Microsoft-i toode.

The screenshot displays the Azure Automation Update Management interface. At the top, it shows the account name 'MAIC-AA-Pri' and the 'Update management' section. A search bar is available. The main dashboard provides a summary of update status:

- Non-compliant machines:** 1 out of 4
- Machines need attention (2):** 1 Critical and security, 1 Other, 0 Not assessed
- Missing updates (9):** 1 Critical, 2 Security, 6 Others
- Failed update deployments:** 0 out of 1 in the past six months

Below the summary, there is a table of updates:

Update name	Classification	Machines missing updates	Operating system	Information link
AzureConnectedMachineAgent - July 2020 Release	Critical updates	1	Windows	KB4573905
2020-07 Cumulative Update for Windows Server 2019 (1809) for x64-bas...	Security updates	1	Windows	KB4558998
2020-06 Security Update for Adobe Flash Player for Windows Server 2019...	Security updates	1	Windows	KB4561600
2020-07 Cumulative Update Preview for .NET Framework 3.5, 4.7.2 and 4...	Updates	1	Windows	KB4567327
Security Intelligence Update for Microsoft Defender Antivirus - KB226760...	Definition updates	2	Windows	KB2267602
2020-01 Update for Windows Server 2019 for x64-based Systems (KB449...	Updates	1	Windows	KB4494174
2020-03 Cumulative Update for Windows Server 2019 (1809) for x64-bas...	Updates	1	Windows	KB4541331

Joonis 6 Azure Automation Update Management haldusliidese vaade (<https://docs.microsoft.com/th-th/azure/automation/update-management/view-update-assessments>)

Peamised funktsioonid [26]:

- Linux ja Windows operatsioonisüsteemide paikamine

- Uuenduste paigalduseelne hindamine
- Uuenduste planeerimine

#### Plussid

- Toetab nii pilves kui ka kohalikus andmekeskuses olevate serverite operatsioonisüsteemide haldust
- Tasuta tarkvara

#### Miinused

- Lisafunktsioonidega võivad kaasneda suured kulud
- Serverisse peab paigaldama agendi, mida peab eraldi haldama hakkama

## 5.5 PowerShell skriptimiskeel

PowerShell on Microsofti poolt pakutav haldusmuudatusteks ja automatiseerimiseks mõeldud skriptimiskeel. Alates aasta 2016 augustist on PowerShell avatud lähtekoodiga ja on toetatud Windows, Linux ja macOS operatsioonisüsteemide poolt [27].

Peamised funktsioonid [28]:

- Haldusmuudatuste automatiseerimine skriptide abil

Plussid:

- Tasuta tarkvara
- Haldusvõimalustel pole piire
- Suur kogukond skriptide näidete jaoks

Miinused:

- Nõuab hallatavate serverite poolt eelseadistusi turvaliseks ühenduseks
- Puudub graafiline liides ja skriptide kirjutamine nõuab suurt käsitööd

- Ettevõtte jaoks kohandatud skriptide loomine on puhas käsitöö ja nõuab rohkelt aega



## 6 Lahenduse valimine

Käesolevas peatükis toob autor välja, milliste kriteeriumite alusel valimis välja toodud paikamislahendusi ja tööriistu omavahel võrreldakse.

Antud peatüki eelifoks on peatükis 4 „Paikamise tarkvarade valimi koostamine“ koostatud lahenduste valim, mida võrreldakse vastavalt alampeatükis 3.1 „Eesti Energia AS nõuded uuele lahendusele“ välja toodud ettevõtte nõuetele. Võrdluse raames tekib punktiline pingerida lahendustele ja valitakse välja kolm parimat ja testitakse *Proof of Concept* näol testkeskkonnas, mille abil on võimalik katsetada ja hinnata, kas lahendused ka päriselt sobivad ettevõttesse kasutusele. Peale katsetusi tehakse analüüs testitud lahenduste kohta ja järelduste põhjal valitakse välja parim, mis võetakse reaalselt ettevõttes kasutusele.

### 6.1 Võrdlustabel esmase valimi kohta

Antud alampeatükis toob autor välja ettevõtte funktsionaalsed ja mittefunktsionaalsed nõuded lahendustele ja analüüsib, kas valimis olevad lahendused vastavad antud nõuetele. Ettevõtte jaoks tähtsaim nõue lahendusele abil oleks võimalik automaatselt ja keskselt servereid paigata. Teisalt on tähtis, et sellega kaasneksid võimalikult väiksed kulud. Antud nõude põhjuseks on, et puudub kindel eelarve lahenduse soetamiseks ja arvestades, et serveripargi suurus võib tulevikus tõusta, ei soovita riskida lahenduse juurutamisega, mis võib tuua kaasa endaga palju lisakulusid.

Tingimus/ Toode	Lahenduse maksumus	Turvalisus	Võimalik integreeruda <i>CMDB</i> -ga
Azure Automation Update Management	Tasuta tarkvara, kuid võivad kaasneda lisakulud	Suhtleb serverisse paigaldatud agendi abil üle <i>TLS 1.2</i> [29]	Ettevõtte <i>CMDB</i> -ga pole võimalik liidestada
Microsoft SCCM	\$3607 Datacenter redaktsiooni eest, kuid hind võib tõusta vastavalt SCCM serveri ressursside võimekusele [30]	HTTPS RPC toetus	Võimalik Jira Insight <i>CMDB</i> -ga liidestada
PowerShell	Tasuta tarkvara	Toetab <i>HTTPS WinRM</i> - i	Enda kirjutatud skripti abil võimalik pärida ja kirjutada <i>CMDB</i> -sse infot
Ansible	Tasuta tarkvara	Toetab <i>HTTPS WinRM</i> - i	Enda kirjutatud moodulite abil võimalik pärida ja kirjutada <i>CMDB</i> -sse infot
SolarWinds Patch Manager	\$1909 aastas või \$3750 ühekordne ost [31]	Võimalik kasutada agente või <i>WMI</i> -d, dokumentatsioonis ei ole mainitud kas võimalik liiklust krüpteerida [32]	Ei ole võimalik ettevõttes oleva <i>CMDB</i> - ga liidestada, kuid SolarWinds pakub eraldi tootena enda versiooni <i>CMDB</i> -st

Tabel 1 Paikamislahenduste võrdlus 1/2

Tingimus/ Toode	Protsessi automatiseerimise võimalus	Rakenduste halduse võimalus	Lahenduse lihtsus
Azure Automation Update Management	Käsitsi paikamise ajutamise võimalus	Toetab paikamise eelseid ja järgseid skripte	Microsoft-i poolt loodud karbitoode, mis ühildub hästi sama ettevõtte poolt tarnitava operatsioonisüsteemiga
Microsoft SCCM	Käsitsi paikamise ajutamise võimalus	Võimalik PowerShell skriptide abil rakendusi hallata	Microsoft-i poolt loodud karbitoode, mis ühildub hästi sama ettevõtte poolt tarnitava operatsioonisüsteemiga
PowerShell	Kuna integreerub <i>CMDB</i> -ga, siis on võimalik kirjutada täisautomaatne protsess	Rakendusi võimalik skriptide näol hallata	Peab kõik skriptid ise kirjutama ja arvestama erinevate serverite versioonidega
Ansible	Kuna integreerub <i>CMDB</i> -ga, siis on võimalik kirjutada täisautomaatne protsess	Rakendusi võimalik moodulite või skriptide näol hallata	Võimalik ära kasutada kogukonna poolt kirjutatud mooduleid, kuid peab integreerima ettevõttega
SolarWinds Patch Manager	Võimalik SolarWinds enda <i>CMDB</i> abil paikamine automatiseerida	Toetab paikamise eelseid ja järgseid PowerShell skripte	Tegu karbitootega, mis lahendab hästi paikamise probleemi

Tabel 2 Paikamislahenduste võrdlus 2/2

## 6.2 Valimis osalenud lahenduste hindamine

Antud alampeatükis summeeritakse kokku vastavalt alampeatükis 3.1 „Eesti Energia AS nõuded uuele lahendusele“ välja toodud osakaaludele iga lahenduse punktisummad. Lisaks on hindamisel arvesse võetud Peatükis 5 „Lahenduste võrdlused“ välja toodud plusse ja miinuseid. Lahendused said igale nõudele viie punkti skaalas hinde, mis korrutati läbi iga nõude osakaaluga. Punktisummade põhjal selgitatakse välja kolm parimat kandidaati, mida käsitletakse täpsemalt ja tehakse edasised testkeskkonna katsetused. Hindamiskriteeriumite põhjal on loodud suur hindamistabel, mis on välja toodud lõputöö Lisas 2 „Valimis osalenud lahenduste hindamistabel“.

Hindamistabelis välja toodud võrdluste põhjal said võimalikust saja kahekümnest punktist parimad kolm endale tulemused:

- Ansible – 104 punkti
- Microsoft SCCM – 77 punkti
- Azure Automation Update Management – 75 punkti

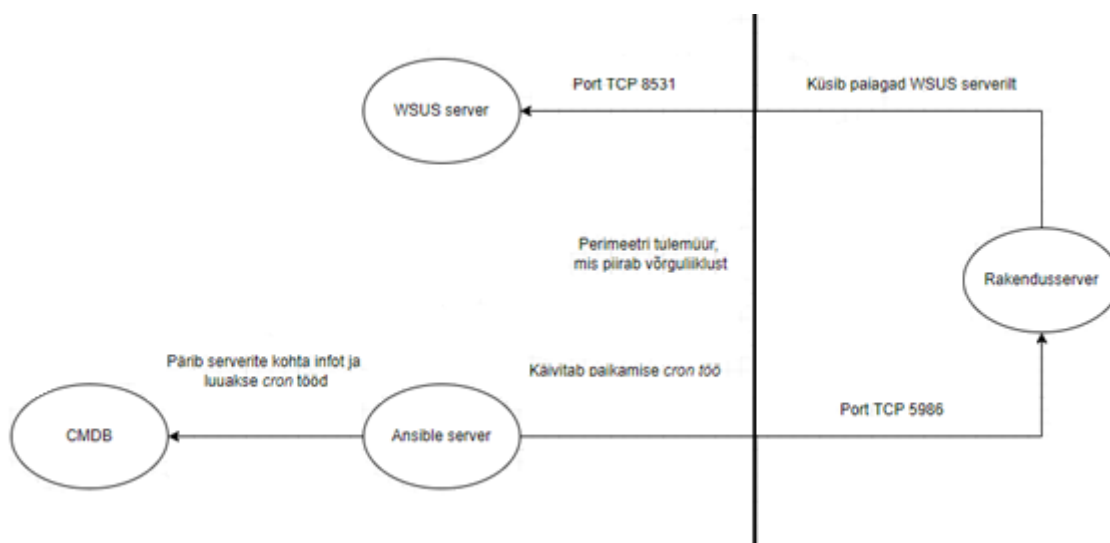
Lõputöö autor koos enda tiimis olevate kolleegidega koostöös analüüsid ja katsetades (v.t lisa 7) leidis, et lõpptulemusena on Ansible konfiguratsioonihalduse tööriist parim, mida saaks rakendada domeeniväliste serverite operatsioonisüsteemi paikamiseks ettevõttes.

## 7 Lahenduse kasutuselevõtmine

Antud peatükis kirjutatakse Eesti Energia AS domeeniväliste rakendusserverite uue paikamislahenduse arhitektuurilisest ülevaatest ja lahenduse kasutusele võtmise erinevatest tegevustest. Juurutamisel on serverite, keskkondade ja muude komponentide nimed üldistatud, et kaitsta ettevõtte spetsiifilisi infoturbe aspekte.

### 7.1 Arhitektuuriline ülevaade

Antud lahenduse paremaks mõistmiseks on autor loonud arhitektuurilise joonise (v.t joonis 7), kus on välja toodud erinevad komponendid ja lihtsustatud tegevused. Komponentide tegevustest on pikemalt räägitud edasistes alampeatükkides.



Joonis 7 Ansible paikamislahenduse arhitektuuriline ülevaade (kuvatõmmis on võetud autori isiklikust kogumist)

### 7.2 Ansible serveri loomine

Kuna Ansible nõuab enda jaoks Linux serverit või Linux binaaride käitamise jaoks loodud Windows-i ühilduvuskihti, WSL-i [33], siis tuli luua uus keskhalduseks mõeldud Linux server. Serveri loomiseks pöörduiti ettevõttes Linux servereid haldava tiimi poole,

kes sai vajaliku serveri luua ja anda ligipääsud töö autorile, et oleks võimalik sinna keskkond üles seada.

## 7.3 Igakuine inventory ja cron tööde loomise töö

Ansible kasutab mitme serveriga haldamiseks serverite gruppe, mis omakorda koostavad Ansible inventory (v.t lisa 3). Serverite grupid sai teha just CMDB-s kirjas olevate hooldusakende põhised. Lisaks, et automatiseerida failide loomised, tehti eraldi Ansible töö, mis tekitab iseennast genereeriva crontab faili.

### 7.3.1 Inventory loomise töö

Selleks, et infot *CMDB* käest küsida, sai ära kasutada tema rakendusliidest. Jira Insight toetab päringute tegemist, mille abil on võimalik kindlale kriteeriumitele vastavaid objekte küsida. Meie soov on, et Ansible saaks kätte kolmele kriteeriumile vastavad serverid:

- Operatsioonisüsteem on Windows Server
- Tegu on domeenivälise serveriga
- Serverile on määratud hooldusaken

Nende kriteeriumite alusel oli võimalik luua päringulause ja üle *CMDB* rakendusliidese sai serverid ja nendega kaasneva info pärida:

```
"Operating System" LIKE "Windows Server" AND "Domain name" = WORKGROUP  
AND "Maintenance Window" IS NOT empty'
```

Joonis 8 Ettevõtte CMDB, Jira Insight-i päringulause

Et oleks võimalik automatiseerida inventory loomise, kirjutas autor Ansible töö (v.t joonis 9).

```

---
- name: Query existing maintenance windows from CMDB
  uri:
    url: "{{ CMDB url }}objecttypeattribute/{{ maintenanceWindowId }}"
    method: get
    force_basic_auth: yes
    body_format: json
    user: "{{ CMDB api user }}"
    password: "{{ CMDB api password }}"
  register: queriedMaintenanceWindows

- name: Query workgroup computers from Insight
  uri:
    url: "{{ CMDB url }}iql/objects?iql={{ serverQueryIql | urlencode() }}"
    method: get
    force_basic_auth: yes
    body_format: json
    user: "{{ CMDB api user }}"
    password: "{{ CMDB api user password }}"
  register: queriedServers

- name: Create inventory file
  template:
    src: hosts.j2
    dest: "{{ inventoryFilePath }}"

```

Joonis 9 Ansible inventory faili loomise töö

Töö pärib CMDB-st päringulause (v.t joonis 9) alusel serverid ja genereerib Jinja2 malli (v.t joonis 10) abil inventory faili (v.t lisa 3).

```

[windows]
{% for server in queriedServers['json']['objectEntries'] %}
{{ server['name'] }}
{% endfor %}

{% for window in queriedMaintenanceWindows['json']['options'].split(",") %}
[{{ window }}]
{% for server in queriedServers['json']['objectEntries'] %}
{% for attribute in server['attributes'] %}
{% if attribute['objectAttributeValues'][0]['displayValue'] == window %}
{{ server['name'] }}
{% endif %}
{% endfor %}
{% endfor %}
{% endfor %}

```

Joonis 10 Inventory faili loomise Jinja2 mall

### 7.3.2 Cron tööde loomise automatiseerimine

Autor lõi iseennast genereeriva crontab faili, mis käivitatakse iga kuu esimesel kuupäeval (v.t Lisa 4). *Cron* töö küsib *CMDB*-st info, annab selle autori poolt kirjutatud Python skriptile (v.t Lisa 4) edasi, mis genereerib uue *crontab* faili (v.t Lisa 4). Genereeritud failis on ajaakende kaupa välja toodud uue kuu paikamise töö käivitamiste *cron* tööd.

## 7.4 Rakendusserverite ette valmistamine lahenduse jaoks

Et oleks võimalik rakendusservereid hallata Ansible-ga, nõudsid need lisategevusi. Nimelt tuli ära defineerida, üle mis kanali ja kuidas Ansible autendib ennast vastu rakendusserverit. Antud alampeatükis on ühe rakendusserveri ette valmistamise tegevused välja toodud.

### 7.4.1 WinRM seadistamine serveril

Ansible ja rakendusserveri vahelist ühendust sooviti teha üle *WinRM* kaughalduse protokoll. *WinRM* kasutab autentimiseks *CredSSP* protokoll. Et serveril *CredSSP*-d kasutada, tuli see eraldi lubada käsuga:

```
Enable-WSManCredSSP -Role Server -Force
```

Kuna soov oli, et ühendus käiks üle krüpteeritud kanali, tuli enne *WinRM* kuulaja konfigureerimist ettevõtte domeeni sertifitseerimiskeskuse kaudu genereerida *HTTPS* sertifikaat. Sertifikaadi loomiseks kirjutas autor sertifikaaditaotluse (v.t Joonis 12).

```
[ req ]
default_bits      = 2048
distinguished_name = req_distinguished_name
req_extensions    = req_ext
prompt            = no
[ req_distinguished_name ]
countryName       = CountryName
stateOrProvinceName = State name
localityName      = Localilty name
oranzationName    = Organization name
commonName        = server1.domain.local
[ req_ext ]
subjectAltName = @alt_names
[ alt_names ]
DNS.1 = server1.domain.local
```

Joonis 11 WinRM HTTPS sertifikaaditaotluse sisu



Sertifikaaditaotluse loomisel oli tähtis, et alternative names listis oleks sama *DNS* nimekirje, millega Ansible server ühendub serveri poole. Mis tähendab, et kui serveri sertifikaaditaotluses on kirje `server1.domain.local`, siis tuleb ühenduda `server1.domain.local` aadressile, mitte lihtsalt `server1` poole. Sertifikaaditaotluse abil genereeriti domeeni sertifitseerimiskeskuses uus sertifikaat. Sertifikaat, koos loodud võtmepaariga tuli import-ida rakendusserverisse. Peale sertifikaadi paigaldust serverisse, sai lõpuks konfigureerida *WinRM* kuulaja. Kuulaja konfigureerimiseks sai kasutada käsku:

```
New-Item -Path WSMAN:\LocalHost\Listener -Transport HTTPS -Address * -  
CertificateThumbPrint $certificate_thumbprint -Force
```

#### **7.4.2 Mandaadid**

CredSSP protokoll serveriga ühendudes vajab ühendust loodava serveri lokaalset kasutajanime ja parooli. Turvalisuse mõttes hoitakse domeeniväliste serverite lokaalseid kontosid keskses paroolihoidlas, mis roteerib määratud ajavahemiku tagant konto parooli serveris. Et Ansible jaoks ei peaks staatilist paroolide faili tegema, kirjutas autor Ansible rolli (v.t lisa 5), mille abil on paroolihoidlast võimalik pärida mingi kindla ajaakna serverite mandaadid. Päritud mandaadid kirjutatakse Ansible-s serveri nimelistesse muutujasse, mida on võimalik dünaamiliselt vastavalt serveri nimele ühenduse loomiseks kasutada (v.t lisa 5 - ühenduse loomise näide dünaamilise muutujaga).

### **7.5 Paikamise Ansible roll**

Paikamiste ja sellega kaasnevate lisatööde jaoks kirjutas lõputöö autor erinevad Ansible rollid ja tööd.

#### **7.5.1 WSUS serveri määramine**

Rakendusserverite paigaldatavate paikade pärimiseks kasutatakse ära ettevõtte domeenisisest WSUS serverit. Rakendusserveris tuli määrata registrikirje abil WSUS serveri aadress. Registrikirje muutmiseks lisati paikamise rolli lisatöö (v.t lisa 6).

### 7.5.2 Serverite hooldusesse määramine monitooringusüsteemis

Rakendusserverite tööd monitooritakse ja juhul, kui töös esinevad vead, saavad vajalikud administraatorid alarmi. Et hooldusakna ajal alarme ei saadetak, tuli kirjutada paikamise rolli lisatöö (v.t lisa 6), mis üle monitooringusüsteemi rakendusliides määrab rakendusserverile hooldustöö staatuse.

### 7.5.3 Paikade alla laadimine ja paigaldamine

Paikade paigalduste jaoks kirjutatakse autor eraldi tööd (v.t lisa 6), mis laeb alla WSUS serverilt critical ja security tasemega paigad. Juhul, kui enne paikamist on vaja serverit taaskäivitada, siis tehakse seda. Peale taaskäivitusi paigaldatakse alla laetud paigad, mille järel tehakse vajadusel jälle taaskäivitused rakendusserveril.

### 7.5.4 Paikamise lõplik playbook

Paikamise lõplik playbook sisaldab endas kolme Ansible rolli. Esimene on mandaatide pärimine rakendusserveritega ühenduse loomiseks. Teiseks on hooldusakna määramine monitooringu serveris, et ebavajalikke alarme ei tuleks. Viimane on paikamise paigalduse käivitamine rakendusserveril paroolihoidlast päritud mandaatide abil.

```
---
- name: Create ansible variables for Windows hosts
  hosts: localhost
  roles:
    - credentialsQuery

- name: Create maintenance for windows hosts
  hosts: localhost
  roles:
    - maintenance

- name: Install Windows patches and restart servers if needed
  hosts: "{{ host_group }}"
  vars:
    ansible_user: Administrator
    ansible_password: "{{
hostvars['localhost'][inventory_hostname.split('.')[0]] }}"
    ansible_connection: winrm
    ansible_winrm_scheme: https
    ansible_winrm_transport: credssp
    ansible_winrm_ca_trust_path: "{{ rootCaCertPath }}"
  roles:
    - Update
```

Joonis 12 Rakendusserverite paikamise playbook

## 7.6 Töövoog testimine

Rakendusserverite paikamine toimub kord kuus. Autor soovis näha igakuist *inventory* faili loomist ja paikamise õigeaegset käivitust. Testimise eesmärgil loodi uus, domeeni lisamata server. Administraatori konto sisestati paroolihoidlasse, serverile määrati *CMDB*-s hooldusaken ja jäädi uut kuud ootama. Igakuine *inventory* loomise töö (v.t lisa 4) suutis genereerida faili, kus oli õige hooldusakna nimelise serverigrupi sees uus rakendusserver. Peale hooldusakent uuris autor Ansible väljundite logist, kus oli näha, et paikamisel ei ole vigu tekkinud ja töövoog on edukalt läbinud. Lisaks vaatas autor eraldi testimiseks loodud serveri peal olevaid paikade paigalduste listi käsu `wmic qfe list` abil. Listist oli näha, et määratud hooldusakna sees olid serverile paigaldatud uued security paigad.

## 7.7 Järeldused

Ansible nõudis alguses aega, et selle funktsionaalsusteega tutvuda – eriti turvalise ühenduse loomine rakendusserveriga. Tulevikus nõuab see ka aega platvormi edasiarendamise näol, kuid autor on võtnud seda arvesse ja näeb seda kui investeringut, kuna tööriista saab kasutada erinevatel otstarvetel. Peale lahenduse kasutuselevõttu tõendab autor, et vajalikud nõuded said täidetud. Autor näeb, et lõputöös loodud lahendust saaks parendada edasiarenduste näol. Võimalikud edasiarendused antud tööle oleksid:

- Serverite automaatne ettevalmistus paikamiseks
- Ansible logid tõsta kesksesse logimissüsteemi
- Monitooring paikamise protsessi jaoks

## 8 Kokkuvõte

Käesolev bakalaureusetöö võrdles Windows operatsioonisüsteemi paikamise lahendusi. Töö eesmärgiks oli välja valida ja juurutada ettevõtte nõuetele vastav lahendus. Võrdluseks püstitati erinevad nõuded, mis tulenesid ettevõtte vajadustest. Autor suhtles erinevate Eesti ettevõtete IT spetsialistidega paikamise teemal, ning viis läbi küsitluse, mille tagajärjel valmis lahenduste valim. Valimis olevaid lahendusi võrreldi vastavalt nõuetele, selgitati välja kolm parimat ja tehti katsetused. Katsetuste järelduste põhjal valiti ettevõttele sobivaim lahendus, milleks oli konfiguratsioonihalduse tööriist Ansible.

Praktilises osas realiseeris lõputöö autor Ansible abil töövoos, mis suudab ettevõtte haldusmuudatuste andmebaasis olevate andmete põhjal paigata domeenivälise Windows rakendusserverite operatsioonisüsteeme. Juurutatud lahendus täitis vajalikud nõuded praktikas.

## Kasutatud kirjandus

- [1] C. Miles, „ITIL 4 and automation – improvement and transformation,“ Axelos, 19 05 2020. [Võrgumaterjal]. Available: <https://www.axelos.com/resource-hub/blog/itil-4-automation-improvement-and-transformation>. [Kasutatud 10 03 2022].
- [2] L. Asher-Dotan, „What is Code Red,“ cyberreason, 4 12 2017. [Võrgumaterjal]. Available: <https://www.cybereason.com/blog/what-is-code-red-worm>. [Kasutatud 11 04 2022].
- [3] S. Mukkamala, „The Evolution of Patch Management: How and When It Got So Complicated,“ DARKReading, 22 01 2022. [Võrgumaterjal]. Available: <https://www.darkreading.com/vulnerabilities-threats/the-evolution-of-patch-management-how-and-when-it-got-so-complicated>. [Kasutatud 16 03 2022].
- [4] A. Chernev, U. Bockenholt ja J. Goodman, „Choice Overload: A Conceptual Review and Meta-Analysis,“ Journal of Consumer Psychology, Chicago, 2015.
- [5] L. Rosencrance, „The state of the PaaS business model and market in 2022,“ TechTarget, 31 01 2022. [Võrgumaterjal]. Available: <https://www.techtarget.com/searchcloudcomputing/feature/The-state-of-the-PaaS-business-model-and-market>. [Kasutatud 25 03 2022].
- [6] Gartner, „Gartner Says 28 Percent of Spending in Key IT Segments Will Shift to the Cloud by 2022,“ Gartner, 18 09 2018. [Võrgumaterjal]. Available: <https://www.gartner.com/en/newsroom/press-releases/2018-09-18-gartner-says-28-percent-of-spending-in-key-IT-segments-will-shift-to-the-cloud-by-2022>. [Kasutatud 07 04 2022].
- [7] B. Kirsch, „What IT admins need to know about cloud application patching,“ TechTarget, 16 09 2019. [Võrgumaterjal]. Available: <https://www.techtarget.com/searchitoperations/tip/What-IT-admins-need-to-know-about-cloud-application-patching>. [Kasutatud 01 04 2022].
- [8] E. W, „The problems with patching,“ National Cyber Security Centre, 10 07 2019. [Võrgumaterjal]. Available: <https://www.ncsc.gov.uk/blog-post/the-problems-with-patching>. [Kasutatud 13 03 2022].
- [9] J. Pichardo, „If it works ... don't touch it?,“ DEV Community, 29 05 2019. [Võrgumaterjal]. Available: <https://dev.to/pichardoj/if-it-works--dont-thouch-it-4ij7>. [Kasutatud 18 03 2022].
- [10] E. Energia, „Üheskoos klientidega oleme teekonnal nulli - Eesti Energia,“ Eesti Energia, [Võrgumaterjal]. Available: <https://www.energia.ee/et/ettevottest>. [Kasutatud 2022 04 20].
- [11] E. Energia, „Enefit Connect- Eesti Energia,“ Eesti Energia, [Võrgumaterjal]. Available: <https://www.energia.ee/enefit-connect>. [Kasutatud 20 04 2022].
- [12] E. Energia, „Enefit Solutions - Eesti Energia,“ Eesti Energia, [Võrgumaterjal]. Available: <https://www.energia.ee/en/ettevottest/tehnoloogia/enefit-solutions>. [Kasutatud 22 04 2022].

- [13] Elektrilevi, „Tutvustus - Elektrilevi,“ Elektrilevi, [Võrgumaterjal]. Available: <https://www.elektrilevi.ee/et/ettevottest/elektrilevi-tutvustus>. [Kasutatud 20 04 2022].
- [14] Atlassian, „Insight - Asset Management | Atlassian Marketplace,“ Atlassian, [Võrgumaterjal]. Available: <https://marketplace.atlassian.com/apps/1212137/insight-asset-management>. [Kasutatud 21 02 2022].
- [15] Techopedia, „What is Windows Server Update Services (WSUS)? - Definition from Techopedia,“ Techopedia, [Võrgumaterjal]. Available: <https://www.techopedia.com/definition/13917/windows-server-update-services-wsus>. [Kasutatud 29 03 2022].
- [16] S. Technology, „The Age of Automation and How it Impacts Business and Workflow,“ Spade Technology, [Võrgumaterjal]. Available: <https://www.spadetechnology.com/the-age-of-automation-and-how-it-impacts-business-and-workflow/>. [Kasutatud 18 03 2022].
- [17] Atlassian, „Insight - Asset Management,“ Atlassian, [Võrgumaterjal]. Available: <https://marketplace.atlassian.com/apps/1212137/insight-asset-management>. [Kasutatud 17 03 2022].
- [18] T. Lind, „SolarWinds blows into Post Falls > Spokane Journal of Business,“ Journal of Business, 11 09 2011. [Võrgumaterjal]. Available: <https://www.spokanejournal.com/local-news/solarwinds-blows-into-post-falls/>. [Kasutatud 19 03 2022].
- [19] I. Jibilian ja K. Canales, „What Is the SolarWinds Hack and Why It Is a Big Deal,“ Insider, 15 04 2021. [Võrgumaterjal]. Available: <https://www.businessinsider.com/solarwinds-hack-explained-government-agencies-cyber-security-2020-12>. [Kasutatud 19 03 2022].
- [20] SolarWinds, „SolarWinds Patch Manager datasheet,“ 2016. [Võrgumaterjal]. Available: [https://www.solarwinds.com/-/media/solarwinds/swdvcv2/licensed-products/patch-manager/resources/datasheets/sw\\_pm\\_datasheet.ashx?rev=af9555906000462ead232303842359de](https://www.solarwinds.com/-/media/solarwinds/swdvcv2/licensed-products/patch-manager/resources/datasheets/sw_pm_datasheet.ashx?rev=af9555906000462ead232303842359de). [Kasutatud 11 03 2022].
- [21] A. Patrizio ja S. J. Bigelow, „What is Microsoft?,“ TechTarget, 02 2022. [Võrgumaterjal]. Available: <https://www.techtarget.com/searchwindowserver/definition/Microsoft>. [Kasutatud 20 03 2022].
- [22] M. Stewart, A. Czechowski, S. Mandalika ja D. Eby, „Features and capabilities - Configuration Manager,“ Microsoft, 10 05 2021. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/changes/features-and-capabilities>. [Kasutatud 11 03 2022].
- [23] N. Jordan, „Source: Red Hat is buying Ansible for more than \$100M,“ VentureBeat, 15 10 2015. [Võrgumaterjal]. Available: <https://venturebeat.com/2015/10/15/source-red-hat-is-buying-ansible-for-more-than-100m/>. [Kasutatud 01 04 2022].
- [24] „The Most Popular Programming Languages – 1965/2022 – New Update,“ Statistics & Data, 01 2022. [Võrgumaterjal]. Available: <https://statisticsanddata.org/data/the-most-popular-programming-languages-1965-2022-new-update/>. [Kasutatud 29 03 2022].

- [25] R. Hat, „Learning Ansible basics,“ Red Hat, 2 11 2020. [Võrgumaterjal]. Available: <https://www.redhat.com/en/topics/automation/learning-ansible-tutorial>. [Kasutatud 12 03 2022].
- [26] S. Sudhir, D. Brown, B. Wren, S. Dhanwada ja D. Coulter, „Azure Automation Update Management overview,“ Microsoft, 08 04 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/azure/automation/update-management/overview>. [Kasutatud 10 03 2022].
- [27] arstechnica, „PowerShell is Microsoft’s latest open source release, coming to Linux, OS X,“ arstechnica, 16 08 2016. [Võrgumaterjal]. Available: <https://arstechnica.com/information-technology/2016/08/powershell-is-microsofts-latest-open-source-release-coming-to-linux-os-x/>. [Kasutatud 30 03 2022].
- [28] S. Wheeler, H. Jacobson ja S. Koon, „What is PowerShell? - Powershell,“ Microsoft, 16 02 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.2>. [Kasutatud 09 03 2022].
- [29] Microsoft, „Azure Automation update management - Azure Architecture Center,“ Microsoft, [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/azure/architecture/hybrid/azure-update-mgmt>. [Kasutatud 10 04 2022].
- [30] Microsoft, „System Center - Licensing & Pricing | Microsoft,“ Microsoft, [Võrgumaterjal]. Available: <https://www.microsoft.com/en-us/system-center/pricing>. [Kasutatud 10 04 2022].
- [31] SaaSwothy, „SolarWinds Patch Manager Pricing: Cost and Pricing plans,“ SaaSwothy, 01 08 2020. [Võrgumaterjal]. Available: <https://www.saaswothy.com/product/solarwinds-patch-manager/pricing>. [Kasutatud 10 04 2022].
- [32] SolarWinds, „Patch Manager 2020.2 System Requirements,“ SolarWinds, 02 2020. [Võrgumaterjal]. Available: [https://documentation.solarwinds.com/en/success\\_center/patchman/content/system\\_requirements/patchman\\_2020-2\\_system\\_requirements.htm](https://documentation.solarwinds.com/en/success_center/patchman/content/system_requirements/patchman_2020-2_system_requirements.htm). [Kasutatud 10 04 2022].
- [33] Ansible, „Installing Ansible,“ Ansible, 29 03 2022. [Võrgumaterjal]. Available: [https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html). [Kasutatud 30 03 2022].
- [34] L. S. Sterling, The Art of Agent-Oriented Modeling, London: The MIT Press, 2009.
- [35] P. Vihalemm, „Eksperiment | Sotsiaalse Analüüsi Meetodite ja Metodoloogia õpibaas,“ 2014. [Võrgumaterjal]. Available: <https://samm.ut.ee/eksperiment>. [Kasutatud 28 03 2022].
- [36] A. R. Dennis ja J. S. Valacich, „"Conducting Experimental Research in Information Systems" by Alan R. Dennis and Joseph S. Valacich,“ 29 01 2001. [Võrgumaterjal]. Available: <https://doi.org/10.17705/1CAIS.00705>. [Kasutatud 28 03 2022].
- [37] C. Cimpanu, „90% of Companies Get Attacked with Three-Year-Old Vulnerabilities,“ Bleepingcomputer, 24 08 2017. [Võrgumaterjal]. Available: <https://www.bleepingcomputer.com/news/security/90-percent-of-companies-get-attacked-with-three-year-old-vulnerabilities/>. [Kasutatud 03 04 2022].

- [38] M. Davis, „ansible.windows.win\_updates module – Download and install Windows updates,“ Ansible, 20 04 2022. [Võrgumaterjal]. Available: [https://docs.ansible.com/ansible/latest/collections/ansible/windows/win\\_updates\\_module.html](https://docs.ansible.com/ansible/latest/collections/ansible/windows/win_updates_module.html). [Kasutatud 20 04 2022].



# Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>

Mina, Tanel Saar

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “ Haldusmuudatuste automatiseerimislahenduse loomine Windows rakendusserverite operatsioonisüsteemi paikamiseks suurettevõtte näitel“, mille juhendaja on Reimo Lepp
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

16.05.2022

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Valimis osalenud lahenduste hindamismaatriks

	Punktid		
	Protsessi automatiseerimise võimalus	Rakenduste halduse võimalus	Lahenduse lihtsus?
SolarWinds Patch Manager	12	9	12
Ansible	20	15	9
Powershell	16	12	3
Azure Automation Update Management	12	9	15
Microsoft SCCM	12	9	15

Tabel 3 Valimis osalenud lahenduste hindamistabel 1/2

	Punktid		
	Lahenduse maksumus	Turvalisus	Võimalik integreeruda CMDB-ga
SolarWinds Patch Manager	5	10	8
Ansible	25	15	20
Powershell	15	10	16
Azure Automation Update Management	15	20	4
Microsoft SCCM	5	20	16

Tabel 4 Valimis osalenud lahenduste hindamistabel 2/2

## Lisa 3 Näide Ansible töö abil genereeritud inventory failist

```
[windows]
server1.domain.local
server2.domain.local
server3.domain.local
server4.domain.local
[Every_2nd_Saturday_0300-6500]
[Every_2nd_Sunday_0300-6500]
[Every_3rd_Tuesday_0400-0630]
[Every_3rd_Tuesday_0900-1100]
[Every_3rd_Tuesday_1000-1200]
[Every_3rd_Tuesday_1200-1400]
[Every_3rd_Tuesday_2000-2200]
[Every_3rd_Tuesday_2200-0000]
[Every_3rd_Wednesday_1000-1200]
[Every_3rd_Wednesday_1200-1400]
[Every_3rd_Wednesday_2000-2200]
[Every_3rd_Wednesday_2200-0000]
[Every_4th_Tuesday_1000-1200]
[Every_4th_Tuesday_1200-1400]
server1.domain.local
[Every_4th_Tuesday_1600-1800]
[Every_4th_Tuesday_1900-2100]
[Every_4th_Tuesday_2000-2200]
[Every_4th_Tuesday_2200-0000]
[Every_4th_Wednesday_1000-1200]
[Every_4th_Wednesday_1100-1300]
[Every_4th_Wednesday_1200-1400]
server2.domain.local
server4.domain.local
[Every_4th_Wednesday_1700-1900]
[Every_4th_Wednesday_2000-2200]
server3.domain.local
[Every_4th_Wednesday_2200-0000]
[Every_4th_Friday_1200-1400]
```

Joonis 13 Ansible töö abil genereeritud inventory faili sisu (kuvatõmmis on võetud autori isiklikust kogumist)

## Lisa 4 Cron tööde loomise Ansible töö, Pythoni skript ja genereeritud crontab fail

```
---
- name: Query existing maintenance windows from CMDB
  uri:
    url: "{{ CMDB uri }}objecttypeattribute/{{
maintenanceWindowObjectTypeAttributeId }}"
    method: get
    force_basic_auth: yes
    body_format: json
    user: "{{ CMDB user }}"
    password: "{{ CMDB password }}"
    register: queriedMaintenanceWindows

- name: Ensure that cronjob creation script is present on host
  template:
    src: cronjobs.py
    dest: "{{ crontabPythonScript }}"

- name: Run cronjob creation script with queried maintenance windows as
parameter
  script: /usr/bin/python3 {{ crontabPythonScript }} "{{
queriedMaintenanceWindows['json']['options'] }}"

- name: Correct permissions for crontab file
  file:
    path: "{{ crontabFilePath }}{{ crontabUser }}"
    owner: "{{ crontabUser }}"
    group: crontab
    mode: '0600'
```

Joonis 14 Crontab faili loomise Ansible töö

```

from datetime import timedelta, date
import sys

def nth_weekday(the_date, nth_week, week_day):
    temp = the_date.replace(day=1)
    adj = (week_day - temp.weekday()) % 7
    temp += timedelta(days=adj)
    temp += timedelta(weeks=nth_week - 1)
    return temp

def getMaintenanceWindowCron(maintenanceWindow):
    maintenanceWindowTemp = maintenanceWindow.split("_")
    hour = maintenanceWindowTemp[3][0:2]
    weekdays = {
        "Monday": 0,
        "Tuesday": 1,
        "Wednesday": 2,
        "Thursday": 3,
        "Friday": 4,
        "Saturday": 5,
        "Sunday": 6,
    }
    curDate = date.today()
    nthWeekDay = int(maintenanceWindowTemp[1][0])
    weekDay = int(weekdays[maintenanceWindowTemp[2]])
    patchDay = nth_weekday(curDate, nthWeekDay, weekDay).day
    cron = f"0 {hour} {patchDay} * *"

    return cron

if __name__ == '__main__':
    cronTabFolder = "/var/spool/cron/crontabs/"
    cronUser = "ansibleupdater"
    cronFile = f"{cronTabFolder}{cronUser}"
    patchPlaybookName = "win_patching.yaml"
    prePatchPlaybookName = "prepatchjobs.yaml"
    ansiblePath = "/etc/ansible"
    everyFirstDayOfMonthCron = "0 6 1 * *"

    maintenanceWindows = sys.argv[1].split(",")
    with open(cronFile, "w") as f:
        cronLine = f"{everyFirstDayOfMonthCron} ansible-playbook
{ansiblePath}/{prePatchPlaybookName}\n"
        f.write(cronLine)
        for window in maintenanceWindows:
            cronLine = f"{getMaintenanceWindowCron(window)} ansible-playbook
{ansiblePath}/{patchPlaybookName} -e \"host_group={window}\"\\n"
            f.write(cronLine)

```

Joonis 15 Crontab faili loomise Python skript

```

0 6 1 * * ansible-playbook /etc/ansible/prepatchjobs.yaml
0 03 9 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_2nd_Saturday_0300-6500"
0 03 10 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_2nd_Sunday_0300-6500"
0 24 19 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Tuesday_0400-0630",
0 09 19 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Tuesday_0900-1100"
0 10 19 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Tuesday_1000-1200"
0 12 19 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Tuesday_1200-1400"
0 20 19 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Tuesday_2000-2200"
0 22 19 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Tuesday_2200-0000"
0 10 20 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Wednesday_1000-1200"
0 12 20 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Wednesday_1200-1400"
0 20 20 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Wednesday_2000-2200"
0 22 20 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_3rd_Wednesday_2200-0000"
0 10 26 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Tuesday_1000-1200"
0 12 26 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Tuesday_1200-1400"
0 16 26 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Tuesday_1600-1800"
0 19 26 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Tuesday_1900-2100",
0 20 26 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Tuesday_2000-2200",
0 22 26 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Tuesday_2200-0000"
0 10 27 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Wednesday_1000-1200"
0 11 27 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Wednesday_1100-1300"
0 12 27 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Wednesday_1200-1400"
0 17 27 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Wednesday_1700-1900",
0 20 27 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Wednesday_2000-2200"
0 22 27 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Wednesday_2200-0000"
0 12 22 * * ansible-playbook /etc/ansible/win_patching.yaml -e "host_group=Every_4th_Friday_1200-1400"

```

Joonis 16 Genereeritud crontab faili sisu (kuvatõmmis on võetud autori isiklikust kogumist)

## Lisa 5 – Paroolihoidlast mandaatide pärimine ja nende kasutamine Ansible-s

---

- name: Get Auth token from password manager
  - uri:
    - url: "{{ password manager auth api }}"
    - method: post
    - body\_format: form-urlencoded
    - headers:
      - Accept: application/json
      - content-type: application/x-www-form-urlencoded
    - body:
      - username: "{{ password manager api user }}"
      - password: "{{ password manager api user password }}"
      - grant\_type: 'password'
  - register: tokenData
  
- name: Find server secret ID from password manager
  - uri:
    - url: "{{ password manager api }}/secrets/?filter.HeartbeatStatus=1&filter.includeRestricted=true&filter.searichtext={{ item }}"
    - method: get
    - body\_format: json
    - headers:
      - Authorization: Bearer {{ tokenData['json']['access\_token'] }}
      - content-type: application/json
  - register: serverIdData
  - with\_items: "{{ groups[host\_group] }}"
  
- name: Get server password field from password manager
  - uri:
    - url: "{{ password manager api }}/secrets/{{ item['json']['records'][0]['id'] }}/fields/password"
    - method: get
    - body\_format: json
    - headers:
      - Authorization: Bearer {{ tokenData['json']['access\_token'] }}
      - content-type: application/json
  - register: serverPasswordData
  - with\_items: "{{ serverIdData['results'] }}"
  
- name: Build facts
  - set\_fact:
    - "{{ item['item']['item'].split('.')[0] }}": "{{ item['json'] }}"
  - with\_items: "{{ serverPasswordData['results'] }}"

Joonis 17 Paroolihoidlast mandaatide pärimine ja muutujatesse kirjutamine

```

---
- name: Create ansible variables for Windows hosts
  hosts: localhost
  roles:
    - credentialsQuery

- name: Install Windows patches and restart servers if needed
  hosts: "{{ host_group }}"
  vars:
    ansible_user: Administrator
    ansible_password: "{{
hostvars['localhost'][inventory_hostname.split('.')[0]] }}"
    ansible_connection: winrm
    ansible_winrm_scheme: https
    ansible_winrm_transport: credssp
    ansible_winrm_ca_trust_path: "{{ rootCaCertPath }}"
  roles:
    - update

```

Joonis 18 Ühenduse loomise näide Ansible dünaamilise muutuja abil

## Lisa 6 – Serveri paikamise rollid ja tööd

```

---
- name: Add or update WSUS server registry for application server
  win_regedit:
    path: HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate
    name: WUServer
    data: "{{ WSUS server ip }}:8530"

- name: Add or update WSUS server registry for application server
  win_regedit:
    path: HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate
    name: WUStatusServer
    data: "{{ WSUS server ip }}:8530"

```

Joonis 19 WSUS serveri määramine rakendusserveritele



```

---
- name: Create monitoring maintenance window for hosts
  monitoring_application_maintenance:
    name: Maintenance for "{{ item }}"
    host_name: "{{ item }}"
    state: present
    minutes: 150
    server_url: "{{ monitoring uri }}"
    login_user: "{{ monitoring api user }}"
    login_password: "{{ monitoring api user password }}"
    validate_certs: no
  with_items: "{{ groups[host_group] }}"

- name: Pause playbook for 10 seconds after maintenance window creation
  pause:
    seconds: 10

```

Joonis 20 Rakendusserveri hooldustöö staatuse määramine monitooringusüsteemis

```

---
- name: Check for missing updates.
  win_updates:
    state: searched
    category_names:
      - CriticalUpdates
      - SecurityUpdates
    register: update_count

- name: Reboot if needed
  win_shell: Restart-Computer -Force
  when: update_count.reboot_required

- name: Install all security, critical, and rollup updates
  win_updates:
    category_names:
      - CriticalUpdates
      - SecurityUpdates
    state: installed
    reboot: yes

- name: Reboot if needed
  win_shell: Restart-Computer -Force
  when: update_count.reboot_required

```

Joonis 21 Rakendusserveri turvapaikamise Ansible tööd

## Lisa 7 – PoC-id ja järeldused kolme parima lahenduse kohta

Antud lisas on välja toodud autori poolt tehtud kolme parima lahenduse PoC-id, testimised ja nende järeldused vastavalt lahenduste nõuetele (v.t alampeatükk 3.1). Lisaks on ka lisas autori meeskonnas olevate, teiste administraatorite tagasiside lahendustele.

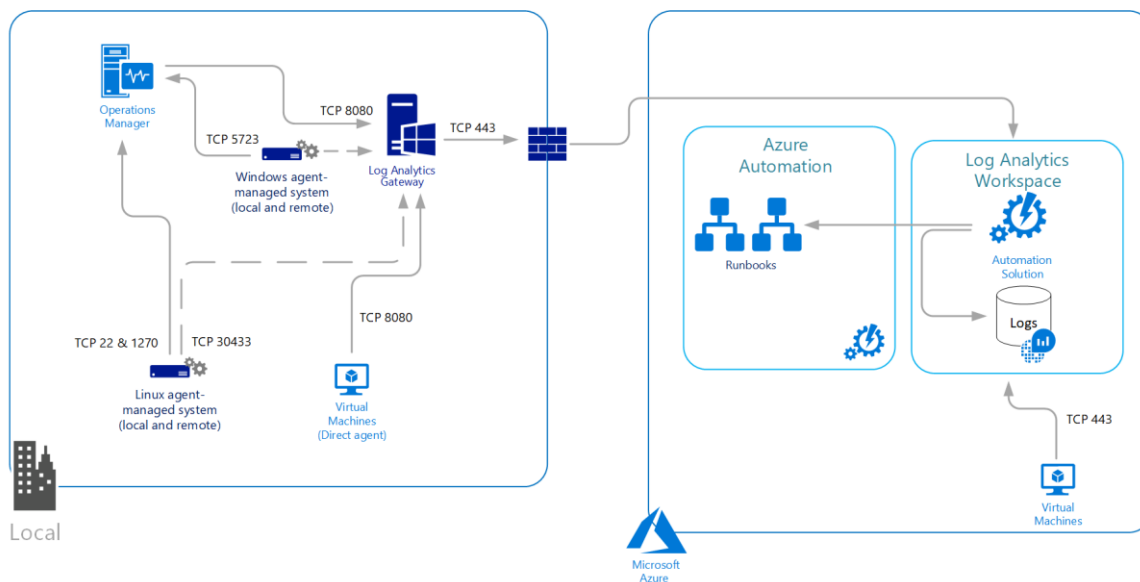
### **Ansible**

Ansible konfiguratsioonihaldustööriist on tasuta tarkvara ja selle kasutusele võtmiseks tuleb ainult alguses üles seadmisel investeerida töötunde. Ansible abil oli võimalik üle krüpteeritud *HTTPS* kanali suhelda domeenivälise serveriga. Ansible suutis domeenivälise serveriga autentida kasutajanime ja parooli või avaliku- ja privaatvõtme paari abil. Ansible kasutab ühenduste defineerimiseks inventory faili, kus on välja toodud serverite grupid ja serverid. Kuna antud faili on võimalik genereerida Ansible enda abil, kasutades ära API päringuid vastu ettevõtte *CMDB*-d, on see ülimalt *CMDB* keskne. Inventory faili genereerimine ja paikamiste tööde käivitamine oli võimalik automatiseerida cron tööde abil. Teenuste haldamiseks oli võimalik käivitada Ansible abil rakendusserveris lokaalne Powershell skript, mis teeb vajalikud tegevused rakenduste sulgemiseks või käivitamiseks. Lahendust arendab suur kommuun ja sellega katsetuste läbi viimine oli lihtne. Lisaks on Ansible juba ettevõttes mujal tiimides kasutusel ja oli vajadusel võimalik teiste tiimide käest abi küsida. Kokkuvõtteks oli Ansible-t väga lihtne üles seada ja sellega oli võimalik peaaegu kõiki tegevusi rakendusserveriga teha.

### **Azure Automation Update Management**

Azure Automation Update Management on tasuta tarkvara, kuid sellega kaasnevad erinevad lisafunktsionaalsused võivad tekitada märkimisväärset kulu. Näiteks logide hoiustamine Azure Log Analytics jaoks maksab eraldi raha. Update Management kasutab kohalikus andmekeskuses olevate serveritega suhtlemiseks gateway serverit. Gateway server omakorda pöördub rakendusserveri poole rakendusserverisse paigaldatud agent-i abil. Kogu liiklus on *TLS 1.2* turbeprotokolliga krüpteeritud ja agent ei nõua autentimiseks kasutajanime ja parooli, mis teeb halduse natuke mugavamaks. Update Management-i ja ettevõtte *CMDB* sidumine piirdus sellega, et Update Management-i oli võimalik serverid tekitada selle abil, kui serverile vajalik agent paigaldati. Paikamise ajaaknad tuli eraldi käsitsi määrata Update Management haldusliideses. Rakendusserveris olevaid teenuseid oli võimalik *runbook* sisse kirjutatud Powershell käskude abil hallata.

*Runbook*-ide kirjutamine oli autori arvates ebaintuiitivne ja tülikas. Lahenduse enda üles seadistamine võttis aega ja autoril jäi katsetuste käigus teenuste halduse teema segaseks. Kokkuvõtteks oli Update Management lahenduse esmane seadistus väga ebaintuiitivne, eriti *runbook*-ide kirjutamine. Peamiseks mureks antud lahendusel olid *CMDB* liidestuse piirangud.



Joonis 22 Azure Automation Update Manager võrgutopoloogia kohaliku andmekeskuse serverite jaoks (<https://docs.microsoft.com/en-us/azure/azure-monitor/agents/log-analytics-agent#installation-options>)

## Microsoft SCCM

Microsoft SCCM on tasuline tarkvara ja Datacenter iteratsioon tarkvarast maksab \$3607 [30]. SCCM ja rakendusserveri vaheline suhtlus toimub rakendusserverisse paigaldatud agent-i abil. Suhtlus oli võimalik seadistada *HTTPS* krüpteeritud kanali peale. Serverid tekivad SCCM haldusliidesesse paigaldatud agent-i abil, mis raporteerib oma staatust SCCM serverile. Ettevõtte *CMDB*-ga ei olnud võimalik SCCM-i liidestada ja pigem tundus, et SCCM on ise *CMDB* laadne toode. Protsessi sai automatiseerida, kui serverile hooldusaken käsitsi SCCM haldusliidesest määrata. Teenuseid oli võimalik hallata orkestratsioonigruppide pre- ja post skriptide näol. Tarkvara üles seadmine oli intuiitivne ja haldusliidese abil sai funktsionaalsusi testida. Kokkuvõtteks ei soovitud lahendust tema maksumuse pärast kasutusele võtta. Lisaks olid ettevõtte *CMDB*-ga liidestamise piirangud suureks puuduseks.

## Lisa 8 – Nõuete täitmise hindamine

Nõue	Hinnang	Hinnangu kriteerium
Lahenduse maksumus		
	Hinne 1	Lahenduse ühekordne ostusumma ületab 2000€ või aastane ostusumma ületab 1000€
	Hinne 5	Tasuta ja vabavaraline tarkvara
Turvalisus		
	Hinne 1	Suhtlus lahenduse ja rakendusserveri vahel toimub üle krüpteerimata kanali
	Hinne 5	Suhtlus lahenduse ja rakendusserveri vahel toimub üle krüpteeritud kanali ja on võimalik ära kasutada paroolihoidlas olevaid mandaate
CMDB integreeritavus		
	Hinne 1	Lahendusel puudub võimalus ära kasutada <i>CMDB</i> -s olevat infot
	Hinne 5	Lahendus suudab <i>CMDB</i> kaudu dünaamiliselt pärida rakendusservereid ja käivitada vastavalt määratud ajaaknale paikamised
Protsessi automatiseerimise võimalus		
	Hinne 1	Lahenduses tuleb käivitada paikamised käsitsi
	Hinne 5	Lahendus suudab peale rakendusserveri ettevalmistamist automaatselt <i>CMDB</i> info põhjal teda paigata
Rakenduse halduse võimalus		
	Hinne 1	Lahendus ei suuda üldse rakendusserveril olevaid teenuseid hallata
	Hinne 5	Lahenduse abil on võimalik kõiki rakendusserveri teenuseid lihtsalt ja intuitiivselt hallata

Lahenduse lihtsus		
	Hinne 1	Lahenduse juurutamine, haldus ja arendus on ainult ettevõtte administraatori teha
	Hinne 5	Lahenduse juurutamine ja arendus on kolmanda osapoole teha ja edasine haldus on administraatori jaoks intuiitivne ja tarnija poolt dokumenteeritud