TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Konstantin Dzyuba 206063IAAB

Veebiteenuse infrastruktuuri kaasajastamine ja arendus Euroland.com AS näitel

Bakalaureusetöö

Juhendaja: Margus Sumla
Magistrikraad

Tallinn 2023

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: Konstantin Dzyuba

24.04.2023

Annotatsioon

Käesoleva töö eesmärgiks on võrrelda ja valida ülesande lahendamiseks sobivaim riistvara veebiserverite, koormusjaoturite ja veebirakenduse tulemüüride paigaldamiseks, kontrollida süsteemi jõudlust erineva koormuse all ning, automatiseerida või poolautomatiseerida infrastruktuuri konfigureerimist ja võimaldada horisontaalset skaleerimist. Töö teoreetiline osa koosneb olemasoleva infrastruktuuri analüüsist, olemasoleva infrastriktuuri probleemide kirjeldusest ja uue laheduse valimisest. Samuti käsitletakse võimalikke lahendusi nii riistvara kui ka tarkvara osas. Tuuakse välja nii valitud lahenduses kasutatava riistvara ja tarkvara puudused kui ka eelised. Töö praktilises osas paigaldatakse ja konfigureeritakse serveri riistvara serverite majutuses, konfigureeritakse serverite tarkvara ja operatsioonisüsteemi, paigaldatakse võrgujaoturi tarkvara ja viiakse läbi koormustestid. Tulemuseks oli uus süsteem, mis vastab kaasaegsetele standarditele, suurendab süsteemi kättesaadavust ja pakub paremat teenuse kvaliteeti. Antud infrastruktuuri uuendus, kaasajastamine, arendus ja süsteemi projekteerimine on väga oluline ettevõtte jaoks, kuna antud ettevõtte kontekstis on veebiserveritel paiknev teenus äri põhieesmärk.

Antud bakalaureusetöö võtmesõnadeks on infrastruktuuri uuendus, kaasajastamine, arendus, süsteemi projekteerimine, riistvara valik ja analüüs.

Lõputöö on kirjutatud eesti keeles keeles ning sisaldab teksti 26 leheküljel, 7 peatükki, 17 joonist, 4 tabelit.

Abstract
Modernization and Development of Web Service Infrastructure on the Example of Euroland.com AS

The aim of this thesis is to compare and select the most appropriate hardware for the installation of web servers, load balancers and web application firewalls, check system performance under different loads, automate or semi-automate infrastructure configuration and apply horizontal scaling.

The theoretical part of the work consists of the analysis of the existing infrastructure, the description of the problems of the existing infrastructure and the selection of a new solution. It also discusses possible solutions both in terms of hardware and software. The disadvantages and advantages of the hardware and software used in the selected solution are pointed out.

In the practical part of the work, the server hardware is installed and configured, the server software and operating system are configured, load balancing software is installed, and load tests are carried out. The result was a new system that meets modern standards, increases system availability, and offers better service quality. The innovation, modernization, development, and system design of the given infrastructure is very important for the company, because in the context of the given company, the service located on the web servers is the main business goal.

The thesis is in Estonian and contains 26 pages of text, 7 chapters, 17 figures, 4 tables.

Lühendite ja mõistete sõnastik

ARR1	*Application Request Routing 1*, Päringute marsruutimine 1
ARR2	*Application Request Routing 2*, Päringute marsruutimine 2
BIOS	*Basic Input-Output System*, Põhimine sisend-väljundsüsteem
CSV	*Comma-separated values*, Komaeraldusega väärtused
DNS	*Domain name system*, Domeeninimede süsteem
HTTP	*HyperText Transfer Protocol*, Hüperteksti edastuse protokoll
HTTPS	*Hypertext Transfer Protocol Secure*, Turvaline hüperteksti edastuse protokoll
iDRAC	*Integrated Dell Remote Access Controller*
IIS	*Internet Information Services*
IP	*Internet Protocol*, Interneti protokoll
IR	*Investor Relations*, Investorsuhted
IT	*Information technology*, Infotehnoloogia
LB1	*Load balancer 1*, Koormusjaotur 1
LB2	*Load balancer 2*, Koormusjaotur 2
MAC	*Medium access control*, Meediumipääsu reguleerimine
NAXSI	*Nginx Anti XSS & SQL Injection*
NLB	*The Network Load Balancing*, Võrgu koormusjaotur
OWASP	*The Open Worldwide Application Security Project*
SLA	*Service level agreement*, Teenusetasemelepe
SQL	*Structured query language*, Struktureeritud päringute keel
TLS	*Transport layer security*, Transpordikihi turve
TMG	*Microsoft Forefront Threat Management Gateway*, Ohuhalduse lüüs
URL	*Uniform Resource Locator*, Ühtne ressursilokaator
XSS	*Cross-site scripting*, Murdskriptimine

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Tänapäeval on ettevõtte infotehnoloogia infrastruktuur keeruline ja mahukas, see koosneb mitmest osast, mis on tihti üksteisest eraldatud. Erinevad süsteemid ja võrgud vajavad pidevat kontrolli, uuendamist ja kaasajastamist. Erinevad meeskonnad töötavad selle nimel, et ettevõtte infrastruktuur töötaks koordineeritult ja nii efektiivselt kui võimalik.

Ärikeskkonnas on IT infrastruktuurist saanud ettevõtete jaoks ülioluline vahend, mida nad kasutavad oma igapäevaste tegevuste juhtimiseks ja toimimiseks. Võrreldes minevikuga, kus IT infrastruktuuri roll piirdus arvutite ja serverite haldamisega, on tänapäeva IT infrastruktuur palju laiem ja keerukam mõiste. See hõlmab andmekeskuseid, pilveteenuseid, võrke, tarkvara ning palju muud.

IT infrastruktuuri kaasajastamine on muutunud ettevõtetele hädavajalikuks, sest see tagab nende konkurentsivõime ning võimaldab neil püsida kiiresti muutuvas ärikeskkonnas. Üks IT infrastruktuuri kaasajastamise olulisemaid aspekte on automatiseerimise võimaluse juurutamine, mis aitab vähendada inimlike vigade tõenäosust ning suurendab ettevõtte efektiivsust. Lisaks sellele võimaldab süsteemi kaasajastamine ja skaleerimine ettevõtetel kiiresti kohaneda muutuvate vajadustega ning suurendada nende võimekust teenindada suuremat hulka kliente. Täiendavalt aitab IT infrastruktuuri kaasajastamine ettevõtetel säästa kulusid ja ressursse. Kui infrastruktuur on vananenud, võib see nõuda rohkem hooldust ning olla tõrkeallikaks, mis võib lõpuks kaasa tuua kulukaid katkestusi ja probleeme, nagu raskusi konfiguratsiooni paindlikuses vananenud tulemüüri liideses.

IT infrastruktuuri kaasajastamine võib olla üsna keeruline protsess, kuid selle tulemusena on võimalik saavutada mitmeid eeliseid. Näiteks võimaldab kaasajastamine ettevõtetel kasutada uusi ja innovaatilisi tehnoloogiaid, mis aitavad suurendada nende efektiivsust ja parandada nende teenuste kvaliteeti.

Seega võib öelda, et IT infrastruktuur on tänapäeva ettevõtete jaoks eluliselt oluline ning selle kaasajastamine on vajalik samm, mida ettevõtted peaksid astuma, et tagada nende konkurentsivõime ning efektiivne toimimine.

2 Probleemi kirjeldus ja eesmärk

Käesoleva peatüki ja selles sisalduvate alapeatükkide eesmärk on selgitada ja analüüsida probleemi taustsüsteemi. Tuuakse välja konkreetse organisatsiooni infrastruktuuri ning kirjeldatakse autori rolli antud projektis. Lisaks tuvastatakse ja analüüsitakse protsesse, mis takistavad ärieesmärkide tõhusat saavutamist.

2.1 Ettevõtte tutvustus

Euroland IR [1] on 1999. aastal asutatud Rootsi ettevõte, mis pakub klientidele erinevaid haldusvahendeid, millega nad saavad investorisuhet oma ettevõttega piisaval tasemel hoida. Üle 150 spesialisti 9-s riigis tegelevad teenuste arendusega ja ülalhoidmisega.

Euroland IR on aastaid abistanud ettevõtteid nende investorsuhete parandamisel, pakkudes parimate tavade töövahendeid ja silmapaistvat ööpäevaringset teenust. Keskendudes oma klientide veebipõhise IR-suhtluse täiustamisele, suudab ettevõtte pakkuda valdkonna tehnoloogiliselt arenenumaid ja kasutajasõbralikumaid lahendusi.

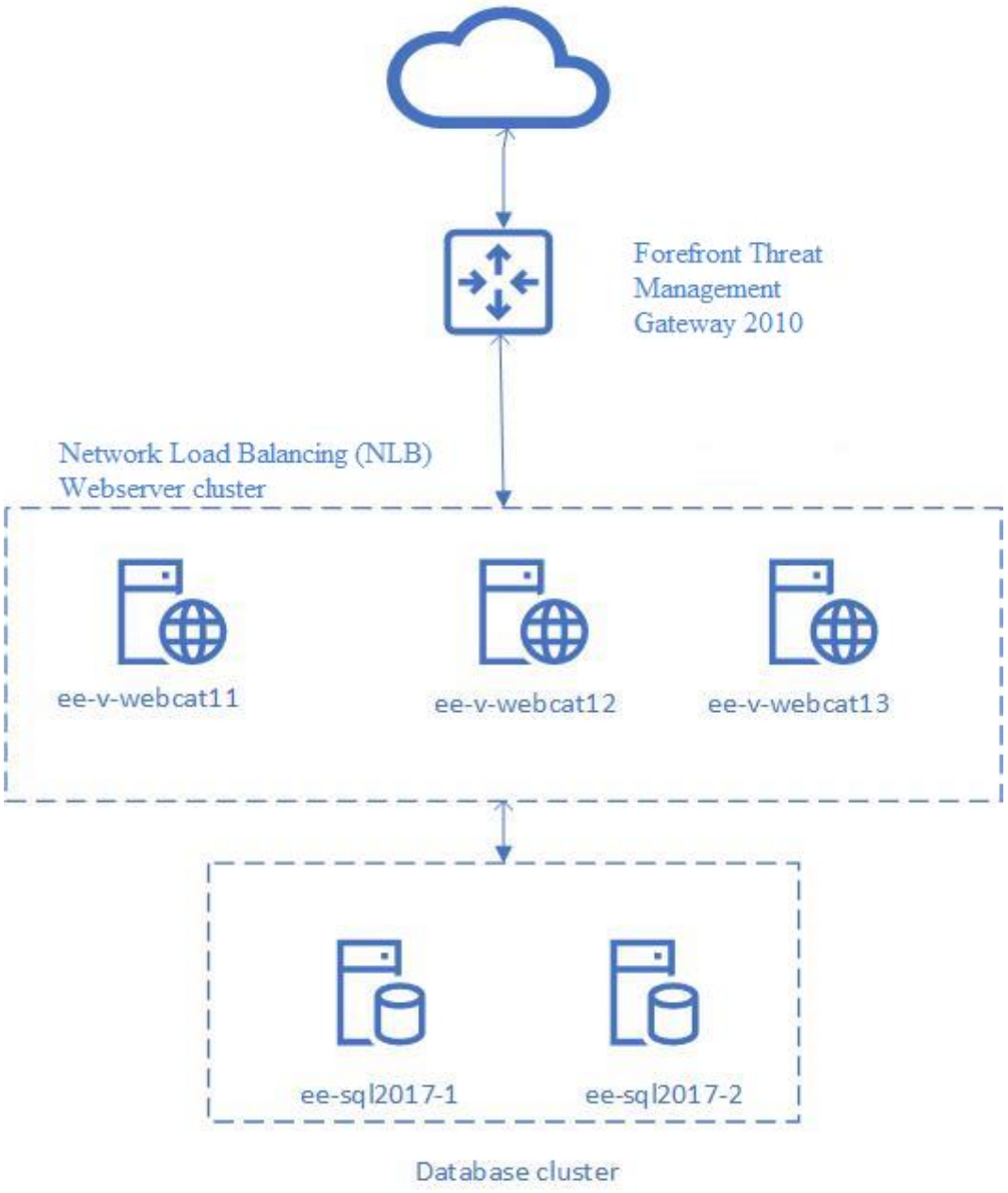
Pakkutavate teenuste intuitiivne disain võimaldab igakülgselt mõista klientide aktsiate lugu. See on peamine digitaalsete tööriistade plaan. Teenuste juurdepääsetavaks ja interaktiivseks muutmine julgustab kasutajaid antud firma finantsajaloos rohkem kaasa lööma. EurolandIR teenused aitavad kaasata investoreid. Püüdes alati olla arengu ees, töötab EurolandIR meeskond väsimatult selle nimel, et firma investoritega suhtlemist ja sidusrühmadega suhete loomist uuesti leiutada ja täiustada. Juba üle 25 aasta on Euroland IR töötanud investorsuhete kogemuse täiustamise nimel.

2.2 Olemasoleva veebiserverite infrastruktuuri kirjeldus

Käesoleva ettevõtte infrastruktuuris on kasutuses vananenud tarkvara ja riistvara lahendused, mis ei võimalda enam sissetulevate päringutega ja liiklusega hakkama saama.

Joonisel [Joonis 1] saab näha, kuidas täpsemalt olemasolev lahendus töötas. Esialgu, kui olemasoleva süsteemi alles püstitati, katsetati TMG sisseehitatud koormusjaotur, aga

selle katsetamine ebaõnnestus, sest tihtipeale kadus ühendus veebiserveritega ning see näitas süsteemi ebastabiilsust. Nii saigi kasutusele võetud NLB [2] nimeline koormusjaotur. Antud koormusjaotur töötas paremini ja oli palju stabiilsem kui enne proovitud TMG koormusjaotur.



Joonis 1 Olemasoleva infrastruktuuri joonis

Joonisel kirjeldatud skeem töötas probleemideta kuni tehti otsus lahti saada füüsilisel riistvaral olevatest veebiserveritest ja teha nendest virtuaalmasinad. Probleem tekkis

koormusjaoturiga, kuna masinate klastri tööks on vaja, et klastris olevate masinate võrguadapter võimaldaks enda MAC-aadressi muutmist. [3] Vastasel juhul klaster keeldub edasi töötamast. Lahendus sai varsti leitud *Hyper-V* masina konfiguratsioonis pidi lubama MAC-aadressi teesklemist. [4]

Kirjeldatud süsteem töötas probleemideta juba pikemat aega aga, kuid nüüdseks on see vananenud. Tarkvara osa on saavutanud oma eluea lõpu, riistvara poole ka tükk aega uuendatud ja vajab väljavahetamist.

2.3 Lähtetingimused lahenduste valimiseks

Käsitletava projekti arengu käigus täpsustati lähtetingimusi, võttes arvesse klientide arvu ja koormuse suurendamist lähitulevikus. Projekti eesmärgiks on tõsta toodangukeskkonna infrastruktuuri jõudlust ja läbilaskevõimet. Esimeseks sammuks projekteerimises oleks TLS mahalaadimis protsessi eraldamine üld süsteemist. See annaks meile rohkem võimalusi ja ruumi rakenduskihi reeglite majandamisega. Serverite tellimisel, tuleb veenduda, et tellitud serveritel on 10-gigabitised võrgukaardid. Kiiremate võrgukaartide nõue tuleneb toodangukeskonna süsteemi kiire taastamise vajadusest varukoopiast.

Üldised nõuded:

- Võimalikult kaasaegsed serverid
- Valitud tarkvara peab saama pidevalt turva uuendusi
- Peavad olema 10-gigabitised võrgukaardid
- Eraldada TLS dešifreerimise koormust eraldi serverisse
- Veebitulemüür ja päringute marsruutimise tarkvara võivad esineda ühes serveris

Uue koormusjaoturi tarkvara peab vastama järgmistele nõuetele:

- Vabavaraline
- Võimalusel, peavad tarkvara komponendid olema avatud lähtekoodiga

- Kasutusele võetava tarkvara dokumentatsioon peab olema avalik
- Stabiilsus - süsteem peab olema valmis haldama kliente tootmis infrastruktuuris
- Tarkvaras peab esinema seire ja silumise võimalus
- Silumine võimalikult lihtne ja probleemid lihtsasti otsitavad veebis
- Automatiseerimisvõimalus

Uue koormusjaoturi riistvara peab vastama järgmistele nõuetele:

- Tuleb tellida serveri muutmälu vastavalt praegusele läbilaskevõimele ja arvestada tuleviku klientide kasvuga
- Serveri protsessor peab olema võimalikult viimasest põlvkonnast ja surema sagedusega

Uue veebirakenduse tulemüüri riistvara peab vastama järgmistele nõuetele:

- Võimalusel kasutada vabavaralist tarkvara
- Võimalusel peab tarkvara olema tuntud tootja poolt tehtud
- Paindlik konfiguratsioon
- Seire võimalus

2.4 Töösuhe ja autoripositsiooni roll projektis

Käsitletava projekti arendus- ja haldusmeeskond, mille liige on ka autor, vastutab infrastruktuuri kaasajastamise, käideldavuse ja ülalhoiu, uute rakenduste juurutamise ning, tööprotsesside ja süsteemi alamkomponentide automatiseerimise eest.

Antud Bakalaureusetöös kirjeldatakse uue projekti infrastruktuuri loomise protsessi. Bakalaureusetöö autor on andnud oma panuse antud projektis infrastruktuuri ehitusse. Bakalaureusetöös kirjeldatu on aga üksnes osa tervest projektist. Bakalaureusetöö autor on lisanud antud töö skoopi olemasoleva süsteemi kirjeldust ja selle süsteemi nõrgad küljed. Samuti autor uuris ning viis läbi turul olevate koormusjaoturite ja veebitulemüüride analüüsi. Tehnilisest osast on bakalaureusetöö skoopi lisatud valitud lahenduse arendus ja projekteerimine. Kui valmis lahendus on kontrollitud, siis autor integreerib süsteemi olemasolevasse infrastruktuuri, konfigureerib ja loob täieliku tootmis keskkonna.

Bakalaureusetöö skoopi ei kulu:

- Uue süsteemi seire
- Uute veebiserverite loomine ja konfigureerimine
- Azure Traffic Manager konfigureerimine
- Andmebaasiserverite kaasajastamine
- Veebirakenduse tulemüüri seadistamine
- Võrgu riistvara rakendamine ja konfigureerimine

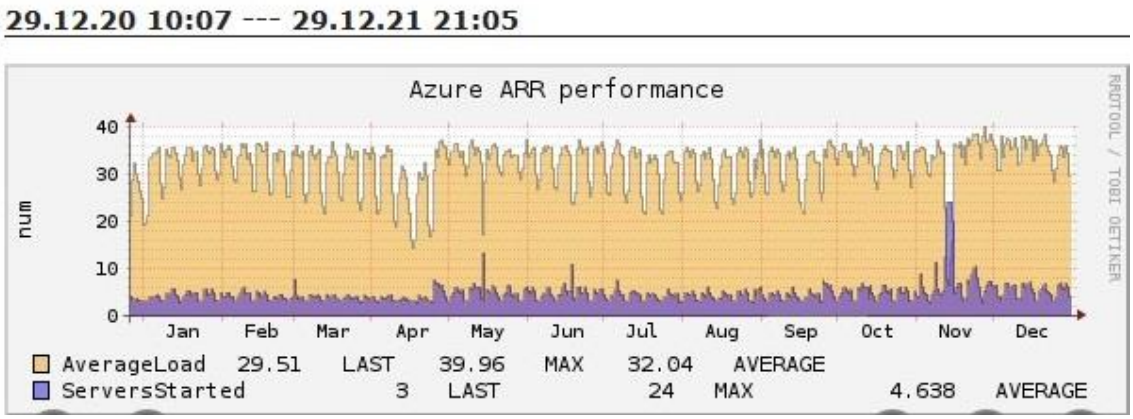
3 Olemasoleva infrastruktuuri kirjeldus ja analüüs

Käesoleva peatüki ja selles sisalduvate alapeatükkide eesmärk on kirjeldada ja analüüsida olemasolevat lahendust ja selle puudusi. Kirjeldatakse, milline tarkvara, operatsioonisüsteem ja serverite riistvara oli kasutuses

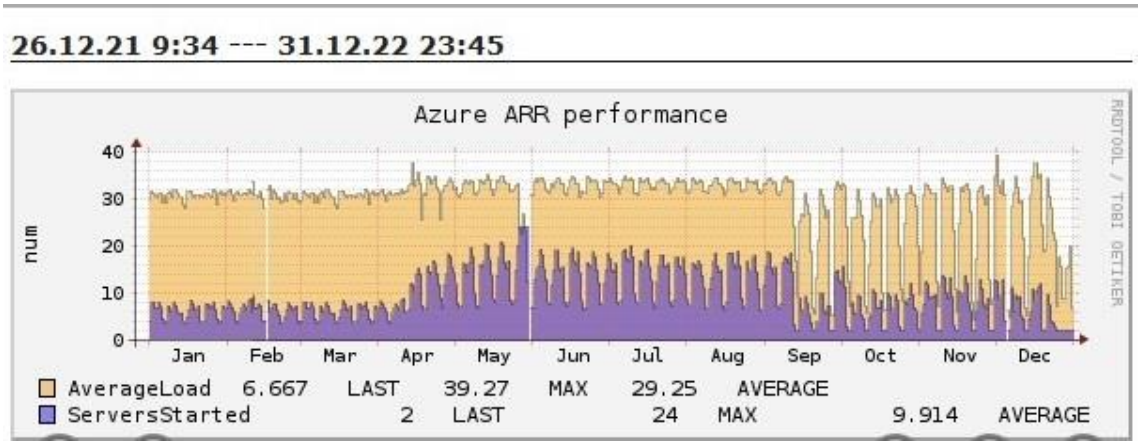
3.1 Olemasoleva riistvara puudused

Olemasolevas infrastruktuuris on kasutuses *Dell PowerEdge R210* ja *Dell PowerEdge R410* serverid aastast 2015. Kuna viimastel aastatel klientide arv ja koormus suurenes piisavas mahus [Joonis 2, Joonis 3], uuendati esimese hooga andmebaasiservereid.

Alltoodud joonistel [Joonis 2, Joonis 3] on näidatud, kuidas täpsemalt aastatega koormus suurenes meie toodangukeskkonna infrastriktuuris. Kui võrrelda koormust aastal 2021 [Joonis 2] ja aastal 2022 [Joonis 3], on märgata, et keskmine koormus 2021. aastal oli madalam ning, aprillis 2022 toimunud koormuse tõus sundis tõstma serverite arvu. Nagu näidatud joonistel, oli käivitatud serverite keskmine arv 2021. aastal 4.638. Aastaga hiljem see arv kahekordistus ja nüüd on 9.914.



Joonis 2 Koormuse graafik aastal 2021



Joonis 3 Koormuse graafik aastal 2022

Antud projekti plaaniks on lahendada võrgupoolset ja veebiliikluse turva probleemi, kuna olemasolevas infrastruktuuris on puudu veebirakenduse tulemüürid ja kaasaegne koormusjaoturi lahendus.

Võrgu riistvara poolest olid kasutuses *Dell PowerEdge R210* mille peal oli *Forefront Threat Management Gateway (TMG) 2010*. Antud tarkvara tulemüür juba saavutas enda eluea lõpu ja nõudis väljavahetamist. Osa koormusest ja tulemüüri reeglitest, mis on olemasolevas tulemüüris konfigureeritud, planeetitakse jagada uue süsteemi ja peatulemüüri vahel. Veebiliiklus hakkab toimuma uue süsteemi kaudu ja muu liiklus suunatakse läbi peatulemüüri.

Üks kaasajastamise osa on ka võrgu uuendamine, sest olemasoleval riistvaral on ainult 1-gigabitised võrgukaardid. Uuete serverite ostmisega on plaanis minna üle 10-gigabitise võrgu kiiruse peale. Kuna tegu on tootmis keskkonnaga, on tähtis kindlustada süsteemi kiire taastamine varukoopiast tõrke korral.

3.2 Olemasoleva tarkvara puudused

Olemasoleva lahenduse tarkvaralises osas oli kasutuses oma eluea lõpu saavutanud *Microsoft Forefront Threat Management Gateway* ning koormusjaoturi ülesandeid täitis *Windows Network Load Balancing*.

Kuna veebiserverid esinesid virtuaalmasinatena, oli tähtis hoida püsival tasemel süsteemi stabiilsust. Tarkvaral, mis täitis koormusjaoturi ülesandeid, oli programmiviga, mis tekkitas võrgus ebavajalikku liiklust [4] ja võis tekitada ebastabiilseid olukordi süsteemis.

Veel üheks puuduseks selles infrastruktuuris on veebitulemüüri puudus, mis tekkitas ohtliku olukorda sisevõrgus. Antud projekti eesmärgiks oli see olikord lahendada.

Olemasoleva süsteemi veebiserverid töötasid *Windows Server 2012* operatsioonisüsteemi peal ning, projekti raames on mõistlik ka need uuendada viimasele operatsioonisüsteemile. *Windows Server 2012* saavutab oma eluea lõpu oktoobris 2023 ning pole mõtet uuendamist hilisemaks jätta.

4 Ülevaade võimalikest lahendustest

Käesoleva peatüki ja selles sisalduvate alapeatükkide eesmärk on vaadata läbi turul olevad lahendused ja võrrelda neid.

4.1 Uue riistvara valimine ja analüüs

Üldisteks nõueteks füüsiliste serverite valimisel olid serveri *rack* kujutegur, *rack density*, milleks on 1U [5] ja protsessoriks Intel. Koormusjaoturi serveri mudeliks sai valitud *Dell PowerEdge R240* [6]. Väike server, mis on suunatud maksimaalse protsessori jõudluse kasutamiseks [7]. Serveri konfiguratsiooni valimisel tähtsaim koht antud projekti jaoks on protsessori valik. Alltoodud tabelis [Tabel 1] on toodud mitu protsessorit mille hulgast saab serveri tellimisel valida. Kuna koormusjaotur hakkab töötama litsentsi mitte vajaval operatsioonisüsteemil, tehti otsus valida valikust kõige võimsam, *Intel(R) Xeon(R) E-2288G* mudeliga protsessor, millel on kaheksa tuuma, nominaal sagedus on kõrgeim võrreldes teiste valikus olevate protsessoritega ja maksimaalne sagedus on samuti kõrgeim.

Tabel 1 Koormusjaoturi protsessorite valiku tabel

Protsessori mudel	E-2274G	E-2276G	E-2278G	E-2286G	E-2288G
Sagedus (MHz)	4000	3800	3400	4000	3700
Maksimaalne sagedus(MHz)	4900	4900	5000	4900	5000
L1 puhver	256	384	512	384	512
L2 puhver (KB)	1024	1536	2048	1536	2048
L3 puhver (KB)	8192	12288	16384	12288	16384
Maksimaalne temperatuur (°C)	69.3	73	73	67.3	67.3
TDP (Watt)	83	80	80	95	95
Tuumade arv	4	6	8	6	8
Lõimede arv	8	12	16	12	16

Veebitulemüüridest langes valik *Dell PowerEdge R640* [8] serveritele. Veebitulemüüriks ja päringute marsruuteriks sobis see server paremini tema loodud tarkvarapõhise salvestusruumi ja kõrgjõudlusega andmetöötluse pärast. Alltoodud tabelis [Tabel 2] on toodud mitu protsessorit, mida saab serveri konfiguratsiooni lisada. *Intel(R) Xeon(R) Gold 6226R* on parim valik antud juhul. Õige litsentseerimise põhimõttega meie jaoks on kõige parim, kui tuumade arv jaguneks 16-ga [9]. Samuti on valikus olemas protsessori mudel *Intel(R) Xeon(R) Gold 6246R* mis on parem ja võimsam ning tuumade arv jääb samaks, kuigi hind kohe kahekordistus. Teised protsessorid on kas suurema tuumade arvuga või väiksema võimsusega.

Tabel 2 Veebitulemüüri protsessorite valiku tabel

Protsessori mudel	6226	6226R	6242R	6246R	6250	6256

Sagedus (MHz)	2700	2900	3100	3400	3900	3600
Maksimaalne sagedus(MHz)	3700	3900	4100	4100	4500	4500
L1 puhver	768	1024	1280	1024	512	768
L2 puhver (KB)	12288	16384	20480	16384	8192	12288
L3 puhver (KB)	19712	22528	36608	36608	36608	33792
Maksimaalne temperatuur (°C)	86	85	76	75	60	64
TDP (Watt)	125	150	205	205	185	205
Tuumade arv	12	16	20	16	8	12
Lõimede arv	24	32	40	32	16	24
Hind	1713$	1804$	3063$	4073$	4506$	4578$

Koormusjaoturiteks valiti kaks *Dell PowerEdge R240* koos *Intel(R) Xeon(R) E-2288G* ja 16GB mäluga. Veebitulemüürideks valiti kaks *Dell PowerEdge R640* koos *Intel(R) Xeon(R) Gold 6226R* ja 32GB mäluga.

4.2 Veebirakenduse tulemüüride analüüs

Tabelis [Tabel 1] ei ole välja toodud *Azure* veebirakenduse tulemüüri. Autori uuringute ja kasutajate tagasiside alusel järeldas autor, et *Azure* veebirakenduse tulemüür [10] projekti ei sobi, kuna kasutab aegunud turvareegleid versioonist 3.2 [11] kui uuem versioon 3.3.4 on juba kättesaadav [12]. Samuti on see kallim kui muud vababara lahendused enda serveritel. Azure veebirakenduse tulemüüri asemel võrdlen *Cloudflare* veebirakenduse tulemüüri [13] ja teen selle SWOT analüüsi. Palun pöörata tähelepanu, et eesolevates tabelites asub IT SWOT analüüs. IT SWOT analüüs asub eesolevates tabelites [Tabel 1, Tabel 2, Tabel 3, Tabel 4].

Nginx'i põhine NAXSI nimeline veebirakenduse tulemüür ei ole võimeline tuvastama enamust rünnakutest ja turvareeglid on suunatud pigem *Nginx* serveri kaitsmisele. Tagasiside järgi NAXSI on umbes 30% aeglasem kui *OWASP ModSecurity* [14].

Selliste teenuste valik on koostatud tarkvara kuulsuse ja mugavuse põhjal. Tabelis analüüsitud teenustel on oma eripärad ja nende meie infrastruktuuri integreerimiseks vajalik töö on erinev nii töö mahu kui ka töö raskuse poolest. Antud projekti raames oli väga tähtis ka selgitada, kui lihtne on valitud tarkvara süsteemi juurutada, sest teenuste stabiilne töö peab olema meie poolt kindlustatud ja klient peab jääma rahule.

Tabel 3 Veebitulemüüride SWOT tabel

SWOT analüüs	OWASP ModSecurity	AQTRONiX WebKnight	Cloudflare Web Application Firewall
Tugevused	1. Omab enda koostatud reeglid 2. Rakenduspõhised välistused 3. Täielik HTTP liikluse logimine 4. Rünnaku käigus on võimalik reaalajas kitsendada turvareeglite skoopi ja HTTP-funktsioone nagu sisutüübid ja päringumeetodid	1. Saab lihtsasti integreerida IIS keskkonda 2. Lihtne ja mugav kasutajaliides 3. Paindlik turvareeglite seadistamine 4. Logimise kautajaliides ja Syslog protokolli toetus	1. Toetab SLA'd 2. Väga lihtne seadistamine 3. Ei pea muretsema turvareeglite uuendamise pärast 4. Kindel kõrgkäideldavus 5. Mugav ja mahukas näidikulaud
Nõrkused	1. Tähendab veel ühte kihti meie süsteemis 2. Reeglid annavad	1. Nõuab kasutaja sekkumist 2. Reeglid annavad	1. Ei sobi meie infrastruktuuri 2. Nõuab DNS nimeserverite

	väärpositiivseid tulemusi	väärpositiivseid tulemusi	ümberkonfigureerimist 3. Tasuta versioonis ei ole avalikke turvareegleid
Võimalused	1. Paranoia tase mis lülitab sisse täiendavad ranged päringute kontrollid 2. Kasutaja määratud paranoia tase 3. Rakenduspõhised välistused	1. SQL, XSS vastu turvareeglid 2. Kodeeritud sisuga päringute kontroll 3. Turva reeglid robotite vastu	1. OWASP reeglid 2. Cloudflare tehtud reeglid 3. Võimalik muuta ja seadistada enda süsteemi jaoks eraldi reeglid 4. Varastatud paroolide tuvastamine 5. Tundlike andmete tuvastamine

4.3 Koormusjaoturite analüüs

Selles IT SWOT analüüsis vaatame üle ja valime antud projekti jaoks sobilikkuma koormusjaoturi lahenduse. Kõige tuntumad nendest on *Haproxy* ja *NGINX* [15]. Kolmandaks koormusjaotusiks oli valitud *Traefik*, mis on suunatud rohkem konteinerite dünaamilisele koormuse jaotamisele [16]. Tabelis [Tabel 2] võrreldi omavahel läbilaskevõimet, päringule vastamise aega ja avalikult internetis leitavaid koormusteste [17].

Tabel 4 Koormusjaoturite SWOT tabel

SWOT analüüs	NGINX	HaProxy	Traefik
Tugevused	1. Tasuta 2. Parim pöördproksi tarkvara turul 3. Sõltub vähem võrgu stabiilsusest 4. Paindlik veebipuhver	1. Tasuta 2. Seire lihtsasti integreeritav 3. Status lehel on rohkem üksikasju 4. Kerge ja mugav seire 5. Paindlik konfiguratsioon 6. Parim tasuta koormusjaotur turul 7. Jooksutab tagasüsteemi kontrolli	1. Tasuta 2. Orienteeritud mikroteenustele 3. Võimaldab dünaamilist konfigureerimist 4. Sisseehitatud LetsEncrypt [18] teenus 5. Mugav ja mahukas näidikulaud
Nõrkused	1. Tasuta tarkvaras antud projekti ülesannete lahenduseks vajalikud	1. Ei toeta mõningaid protokolle 2. Mitmelõimelisus ei ole piisaval tasemel	1. Läbilaskevõime on madal 2. Keeruline seadistamine ja paigaldamine

	funktsioonid puuduvad 2. Tasuta tarkvara toetab ainult mõningaid protokolle: HTTP, HTTPS, Email protokollid 3. Puudub staatuse leht 4. Tasuta versioonis seire on kättesaamatu. Meetrika eksport puudub.		
Võimalused	1. Saab töötada ka tavalise veebiserverina 1. Pöördproksi veebipuhveriga 2. WebSocket toetus	2. Veebipuhver 3. Paindlik ja võimas logimine 4. Mitmelõimelisus 5. TLS koormuse vabastus	1. Orienteeritud mikroteenustele 2. Dünaamiline konfigureerimine 3. Lihtsasti ühilduv kõigi suuremate klastritehnoloogiatega

	3. Võimaldab hoida kuni 10000 üheaegset ühendust		

5 Valitud lahenduse analüüs

Käesoleva peatüki ja selles sisalduvate alapeatükkide eesmärk on võtta kokku eelnevalt võrreldud tarkvara ja riistvara ning luua topoloogia diagrammi.

5.1 Valitud lahenduse kirjeldus

Projekti eesmärgi lahendamiseks oli vaja valida paindlik tarkvara mis oskaks hästi töötada veebiserveritega ja võimalusel oleks ühe tootja poolt tehtud mis tähendaks lihtsat integreerimist süsteemi. Selle aluseks valiti koormusjaoturi tarkvaraks Haproxy. Haproxy on lihtne paigaldada ja seadistada, kuid seadistamiseks on vaja konfiguratsiooni süntaksi baas-teadmisi. Dokumentatsioon on kergesti kättesaadav ja tarkvara pidevalt uuendatakse, parandatakse turvaauke ja arendatakse edasi. Tähtis on ka selline aspekt, nagu litsentsid. Kuna Haproxy on tasuta ja avatud lähtekoodiga ning seda saab paigaldada Linux operatsioonisüsteemile, ei pea muretsema muu operatsioonisüsteemi litsentsi ostmise pärast, mis on ka tähtis ja välditakse täiendavat finantskulu.

Veebitulemüüriks valiti AQTRONiX WebKnight. Suurt rolli selle veebitulemüüri valikus mängis integreerimise võimalus IIS veebiserveriga. See tähendaks ühe kihi kaotamist meie süsteemis. Turvareeglite poolest, on nad standardsed ja konfigureeritavad, on olemas sisseehitatud logimine ja põhjalik päringute skaneerimine turvariskide osas.

Füüsiliseks riistvaraks valiti Dell serverid. Kuna infrastruktuuris olid ka enne kasutuses ainult Dell PowerEdge serverid, tehti ettepanek kasutada neid ka edasi ja mitte proovida midagi muud. Koormusjaoturiteks valiti kaks PowerEdge R240 koos Intel(R) Xeon(R) E-2288G protsessoriga ja 16GB mäluga. Antud protsessorit valiti tema kiiruse pärast,

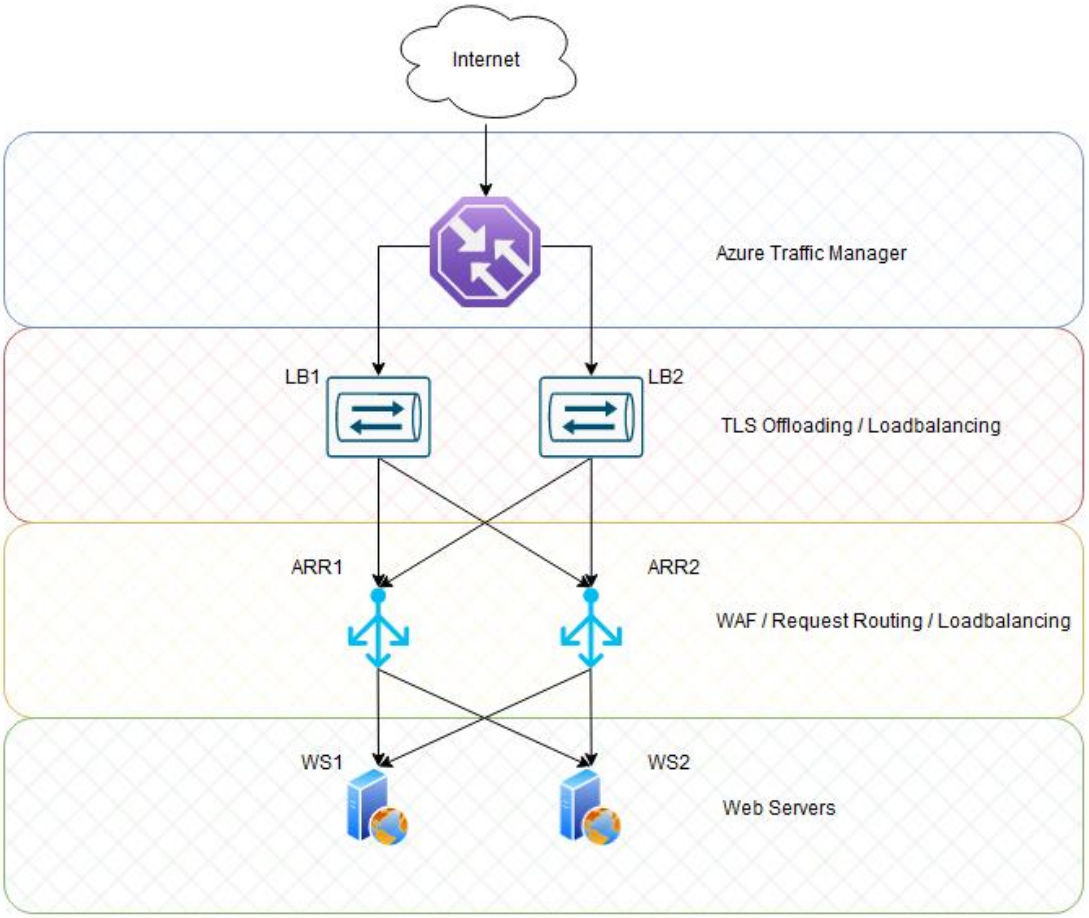
baas sagedusega 3,7GHz ja kuni 5GHz. Mälu puhul võeti 16GB, millest piisab mitmekümne tuhande konkureeriva ühenduse ülalhoidmiseks ja saab veel dešifreerida TLS ühendusi.

Kuna veebitulemüüri ja päringute marsruutimise on võimalik juurutada IIS veebiserveri peale, tehti otsus juurutada need ühte füüsilisse serverisse. Ostmiseks valiti kaks Dell PowerEdge R640 koos Intel(R) Xeon(R) Gold 6226R ja 32GB mäluga. Antud protsessorit valiti tema tuumade arvu pärast ja mugava litsentseerimise jaoks, kuna Microsoft müüb litsentse protsessori tuuma arvu alusel. Tellimuse ajal oli see kõige mõistlikum valik ning teised pakutud protsessorid olid juba suurema või väiksema tuumade arvuga.

5.2 Valitud lahenduse topoloogia

Pärast pikemat arutlemist ja topoloogia muutmist tehti valmis järgmine topoloogia alloleval joonisel [Joonis 4]. Välistest teenusest võeti kasutusse Azure Traffic Manager, mis teeb DNS-põhilist liikluse koormusjaotust. Azure Traffic Manager konfigureeriti järgmiselt: loodi uus profiil, kuhu konfigureeriti kaks lõppseadet (LB1 ja LB2). LB1 ja LB2 on kaks Haproxy serverit, mis dešifreerivad TLS liiklust ja töötavad ka koormusjaoturitena. Edasi jõuab dešifreeritud võrguliiklus järgmisesse etappi (serverid ARR1 ja ARR2), kus enne kui uus päring jõuab veebiserveriteni, kontrollib esimesena päringut veebitulemüür. Kui veebitulemüüri turvareeglid ei vallandu, siis järgmise etapina kontrollitakse ja käiakse üksteise järel läbi kõik reeglid, mis on olemas päringute marsruutimise konfiguratsioonis. Lõppetapiks on vastavalt päringu masruutimise tabelile õige serveri või serveri pordil asuva aplikatsiooni valik ja päringu edasi saatmine.

Samuti arutleti antud süsteemi arenduse üle lähitulevikus. Projekteerimisel on arvesse võetud tulevaste klientide arv ja vastavalt koormuse suurenemine süsteemile. Lõpptulemusena omab projekteeritud süsteem horisontaalse skaleerimise võimalust ja selle kiire juurutamist. Horisontaalne skaleerimine on tänapäeval levinud hea tava, millega arvestatakse süsteemide projekteerimisel [19].



Joonis 4 Planeeritava süsteemi topoloogia diagramm

Kirjeldatud topoloogia on kõige mõistlikum antud ettevõte infrastruktuuris ja lahendab järgmiseid probleeme:

- Tõrkesiire ja kõrgkäideldavus
- Horisontaalse skaleerimise arendus
- Paindlikkus ja uuete süsteemide integreerimine
- Juurutamine ja konfiguratsiooni muutmine

Eeltoodust tulenevalt võetigi vastu otsus liikuda edasi antud projekti topoloogiaga kuna see vastab nõuetele ja ei vaja lisa arengut või raha kulu. Tõrke puhul oskavad serverid üks-teist asendada ja see ei vaja IT administraatori poolt mingisugust panust.

6 Tehniline juurutamise protsess

Käesoleva peatüki ja selles sisalduvate alapeatükkide eesmärk on kirjeldada uue süsteemi juurutamisprotsessi olemasolevasse infrastruktuuri ja viia läbi koormustestid.

6.1 Riistvara paigaldamine

Enne riistvara viimist serverite majutusse on tähtis eelkonfigureerida mõned üksikasjad nagu iDRAC, IP aadressid, serveri domeeni lisamine ja BIOS.

iDRAC'i puhul on tähtis konfigureerida IP aadresi, võimalusel panna sisseehitatud kasutajale pikem parool või lisada autentimine läbi domeeni serveri. Kontrollida, kas töötavad erinevad iDRAC'i osad, nagu virtuaalne konsool. Veebitulemüüri serverite puhul lülitame BIOS konfiguratsiooni ajal välja loogilised protsessorid ning toitehalduse all valida säte nimega „maksimaalne jõudlus" [20]. See säte hoiab serveri protsessori kindlal sagedusel. Nii lahendame probleemi, kui protsessor ei ole valmis suurt osa päringuid kohe vastu võtma.

Kui iDRAC'i osa on valmis, tuleb aeg eelkonfigureerida operatsioonisüsteemi. Esimeseks sammuks on serveri domeeni lisamine. Serveri nimi tuleb panna vastavalt ettevõtte standardile, kus esimene osa on riik, kus asub server, siis täht, mis identsifitseerib, kas server on virtuaalmasin või füüsiline masin. Lõpus aga kas juhuslik nimi, mille järgi meeskond saab tuvastada serveri, või teenuse nimi, mis hakkab jooksma serveris. Serveri IP aadress tuleb teha staatiliseks ja konfigureerida käsitsi. Kui IP aadressi konfiguratsioon on tehtud, tuleb teha uus DNS kiri selleks, et ei peaks IP aadressi meeles hoidma. Järgmiseks etapiks on turvatarkvara paigaldamine. Selleks kasutame *Windows Defender for Business*. Paigaldamiseks on vaja laadida alla „*batch*" skript *Windows Defender* haldamise keskkonnast ja käivitada see. Skript paigaldab vajalike teeke, muudab registrit ja lisab vajalike litsentse. Kui eelseadistaime õnnestus, saab tegutseda serveri otsese ülesande tarkvara paigaldamisega.

6.2 Koormusjaoturi paigaldamine

Koormusjaoturi tööd hakkab tegema Haproxy nimeline tarkvara. Kui Haproxy on juba edukalt paidaldatud, mille täpne protsess on kirjeldatud lisas [Lisa 2], tuleb pöörata

tähelepanu ainult ühele aspektile. Kuna antud süsteem hakkab töötama tootmis keskkonnas, on kriitiline saada kõik viimased tarkvara uuendused. Kuna Debian'i ametlik hoidla ei oma viimast Haproxy versiooni ja seda hoidlat nii tihti ei uuendata, siis tuleb kasutada Haproxy enda tehtud hoidlat kust saab värskema tarkvara versiooni ja uuendused jõuavad kiiremini kasutajateni.

Haproxy konfiguratsioonis on kasutatud mõningaid eri-funktsioone, nagu TLS sertifikaadi teede hoidmine ühes teksti-failis. Seda on mugav kasutada juhul, kui on mitu TLS sertifikaati kasutuses. Kasutame „crt-list" seadet, kus määrame teksti faili ja seal failis on kirjutatud teed sertifikaadi failideni ja nende juures järgmistena asuvad kandilised sulud, mille sisse saab kirjutada parameetrid. Sulgude kõrval on DNS nimed mille kaudu tarkvara saab aru, millele see viitab.

```
/etc/haproxy/certs/public-foo.pem [] *.foo.com
/etc/haproxy/certs/public-bar.pem [] *.bar.com
/etc/haproxy/certs/public-lorem.pem [] *.lorem.com
/etc/haproxy/certs/public-ipsum.pem [] *.ipsum.com
```

Joonis 5 Näidis konfiguratsioon TLS sertifikaatide loetelust ühes failis

Selleks, et Haproxy mitme tuumalise protsessoriga paremini töötaks, on võimalik seadistada Haproxy spetsiaalselt kasutada kõiki protsessori tuumi. Selleks on mitu eraldi seadet mitmelõimelisuse konfigureerimise jaoks, mis on kirjeldatud Haproxy dokumentatsioonis [21]. Antud juhul kasutasime "nbthread 8" seadet. Kuna kasutuses oli kaheksa tuumaline protsessor, siit tuligi number kaheksa lõppu.

Tähtis on ka mitte unustada sisse lülitada Haproxy statistika lehe. See võimaldab koguda täpsemat informatsiooni. Antud statistika leht võimaldab genereerida CSV failivormingus vajalikke andmeid, mida saab mugavamalt lugeda seire tarkvara abil milleks on Grafana ja Prometheuse pinu. Grafanas saab ise teha näidikulaua või kasutada avalike valmis lahendusi Grafana veebilehel [22].

Turvalisuse poolest on hädavajalik kasutada kas sisseehitatud tulemüüri või õigesti konfigureerida IP aadressid mille peal teenused hakkavad töötama teatud pordil. Antud juhul olid konfigureeritud eesliides „http-in" ja „https-in" välise IP aadressiga ja statistika leht sisemise IP aadressiga. Nii me väldime tulemüüri kasutamist ja teenuste töötamist iga konfigureeritud IP aadressil.

Koormusjaoturi paigaldamise ja konfigureerimise protsess on kirjeldatud lisades [Lisa 2, Lisa 3].

6.3 Veebitulemüüri paigaldamine

Veebitulemüüriks sai valitud AQTRONiX WebKnight. Kuna antud veebitulemüüri paigaldamise protsess sisaldab ainult paigaldamis tarkvara käivitamist, loetakse tähtsaimaks osaks veebitulemüüri konfigureerimise protsessi.

Kuna ettevõttes on kasutuses mitu erinevat DNS nime, mis hakkavad läbi minema sellest süsteemist, on vajalik need kõik veebitulemüüri konfiguratsiooni sisse kirjutada. Tuleb veel tähele panna, et logimine ja seire on vajalik ning antud tarkvara seda võimaldab. Logimise all on vajalik seadistada logimise serveri IP aadress ja port. Tulevikus tuleb veel muuta URL skaneerimise reeglid, kuhu lisada teed veebiteenusteni, mida veebitulemüür ei pea kontrollima.

Kõrgkäideldavuse põhimõttega veebitulemüüre on tehtud mitu ja tekiks probleem kui konfiguratsioon oleks erinev igas serveris. Probleemi saab lahendada, kui kopeerida konfiguratsiooni fail peaserverist, kasutades „*robocopy*“, mis on sisseehitatud Windows operatsioonisüsteemi. Joonisel [Joonis 6] saab näha skripti, mida käivitatakse ülesannete planeerijaga kohe pärast operatsioonisüsteemi käivitamist. Parameetriteks on „/R“, mis tähendab korduskatsete arvu juhul, kui eelmine ebaõnnestub. Parameeter „/W“ määrab ooteaja korduskatsete vahel. Parameeter „/MON“ on siin tähtis, kuna hoiab „*robocopy*“ protsessi tagaplaanil lahti, jälgib kausta ja käivitab faili kopeerimise juhul kui tuvastab faili muudatust kaustas. Parameeter „/MOT“ jälgib kausta ja käivitab faili kopeerimise kindla aja pärast. Alltoodud joonisel [Joonis 6] oleval skriptil käivitatakse kolm käsku mis jälgivad kolme erinevat kausta tagaplaanil, ja juhul, kui peaserveri kaustas tekib muudatus, kopeerib alluvserver muudatuse endale. Antud serveri puhul kopeeritakse IIS konfiguratsiooni, veebitulemüüri konfiguratsiooni ja TLS sertifikaadid.

```
@echo off
start robocopy \\ee-arr1\iis_shared_config C:\inetpub\iis_shared_config /R:3 /W:10 /MON:1 /MOT:1
start robocopy "\\ee-arr1\AQTRONIX Webknight" "C:\Program Files\AQTRONIX Webknight" *.xml /R:3 /W:10 /MON:1 /MOT:1
start robocopy \\ee-arr1\centralcert C:\inetpub\centralcert /R:3 /W:10 /MON:1 /MOT:1 /MIR
```

Joonis 6 Skript, millega kopeeritakse konfiguratsiooni failid

Veebitulemüüri paigaldamise ja konfigureerimise täielik protsess on kirjeldatud lisas [Lisa 4].

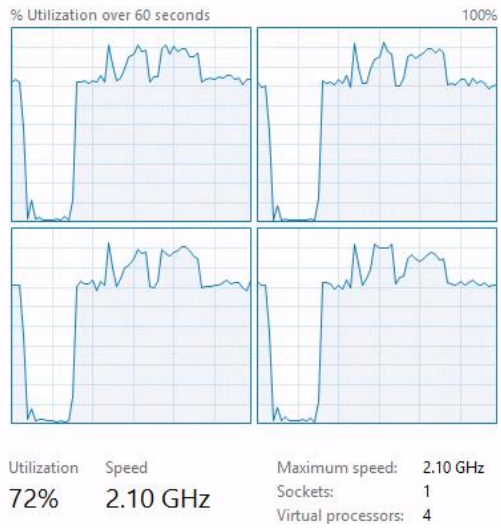
6.4 Serverite süsteemi koormustestid

Koormustestide eesmärgiks on testida erinevad koormusjaoturi profiile ja konfiguratsioone. Koormustestiks kasutame „*vegeta*“ nimelist tarkvara [23]. Testiks kasutame virtuaalmasinat millele on konfigureeritud 16 protsessori tuuma.

Esimese hooga kontrollime, milline läbilaskevõime on ühel veebiserveril. Selleks teeme valmis HTTP päringu mis pärib otse veebiserverist, ning koormusjaoturi antud juhul ei kasuta. Selleks, et kontrollida veebiserveri ja andmebaasi omavahelist tööd, saadame veebiserverile sellise päringu mille vastuseks veebiserver peaks andmed võtma andmebaasi serverist. Koormustesti tulemused on kajastatud alloleval joonisel [Joonis 7].

```
10:24:45 root@VEGETA:~# echo "GET http://ee-v-alpha.euroland.com/tools/Ticker/Scrolling/GetGraphIntradayData/?sid=0.6
Time&defaultNumberFormat=%23%2C%23%230.00&companycode=uk-rdsa&getCleanData=false&v=1" | vegeta -cpus=16 attack -durat
Requests      [total, rate, throughput]         300000, 4999.77, 4999.65
Duration      [total, attack, wait]             1m0s, 1m0s, 1.471ms
Latencies     [min, mean, 50, 90, 95, 99, max]  433.103µs, 2.948ms, 1.581ms, 6.631ms, 10.034ms, 19.816ms, 115.953ms
Bytes In      [total, mean]                     28200000, 94.00
Bytes Out     [total, mean]                     0, 0.00
Success       [ratio]                           100.00%
Status Codes  [code:count]                      200:300000
Error Set:
```

Joonis 7 Koormustesti tulemus ühe veebuserveri vastu

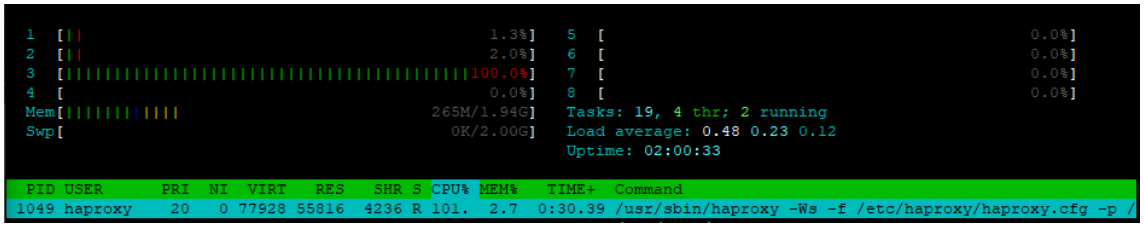


Joonis 8 Veebiserveri protsessori kasutamine koormustesti ajal

Kõrval-olevatel joonistel [Joonis 7, Joonis 8] on näha, et 5000 päringu sekundis puhul on ühe veebiserveri protsessori kasutamine 70 %. Koormustesti tarkvara näitas, et keskmine vastuse aeg oli umbes 3 ms ja maksimaalne vastuse ooteaeg oli 115 ms. Veebiserver töötles läbi ja vastas edukalt kõikidele päringutele, mida olime saatnud.

Kuna mitte šifreeritud liiklusega saavad veebiserverid hakkama, on meie peamiseks eesmärgiks parim läbilaskevõime ja minimaalne jõudluse ja aja tarbimine iga šifreeritud ühenduse kohta. Nüüd tuleb kontrollida ja seadistada koormusjaoturi tööd, mis hakkab tegelema liikluse dešifreerimisega. Selleks, et võrrelda, kuidas töötab vaikekonfiguratsioon, teeme esimese testi ja vaatame tulemusi. Vaikekonfiguratsioonis

olid muutetud ainult serverite IP aadressid õigete vastu. Vaikekonfiguratsiooni saab leida lisas [Lisa 3].



Joonis 9 Koormusjaotur kasutab ainult ühe tuma

```
11:29:50 root@VEGETA:~# echo "GET https://tools.euroland.com/tools/Ticker/Scrolling/GetGraphIntradayData/?sid=0.62239
991512733387&instrumentID=20634&lang=en-GB&decimalMarket=.&thousandGroupMarker=%2C&timezone=W.%20Europe%20Standart%20T
ime&defaultNumberFormat=%23%2C%23%230.00&companycode=uk-rdsa&getCleanData=false&v=1" | vegeta -cpus=16 attack -durati
on=10s -rate=500 -workers=50 -timeout=10s | tee reports.bin | vegeta report
Requests      [total, rate, throughput]         5000, 500.10, 79.48
Duration      [total, attack, wait]             12.72s, 9.998s, 2.722s
Latencies     [min, mean, 50, 90, 95, 99, max]  49.601µs, 1.977s, 105.593µs, 6.172s, 6.489s, 7.286s, 10.001s
Bytes In      [total, mean]                     95034, 19.01
Bytes Out     [total, mean]                     0, 0.00
Success       [ratio]                           20.22%
Status Codes  [code:count]                      0:3989  200:1011
Error Set:
```

Joonis 10 Koormustesti tulemused koormusjaoturi vaikekonfiguratsiooniga

Joonisel [Joonis 10] olev koormustesti tulemus näitab, et koormusjaotur kasutab ainult ühe protsessori tuuma [Joonis 9] päringute töötlemiseks. Viiesaja päringuga sekundis ei suutnud Haproxy töödelda läbi 80% sissetulevatest päringutest. Olukorra parandamiseks on vaja konfigureerida mitmelõimelisust. [24] [25]

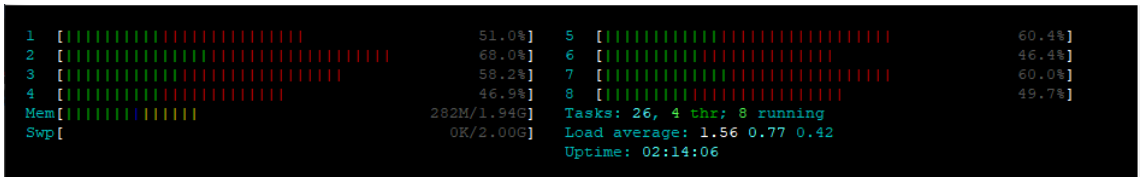
Eelolevad testid olid tehtud mitte häälestatud Haproxy konfiguratsiooniga. Konfigureerides mitmelõimelisust on tähtis kasutada sätted nagu „nbproc“, „nbthread“ ja „cpu-map“ [21], kus „nbproc“ reguleerib protsesside arvu, „nbthread“ võimaldab määrata, mitu lõime protsessi kohta käivitatakse, ja „cpu-map“ vastavalt konfiguratsioonile kleebib Haproxy protsessi lõimed serveri protsessori külge.

Esimeses testis kontrollime kuidas mitmelõimelisus töötab. Konfigureerime „nbproc“ nii, et käivitaks kaheksa Harpxy protsessi iga protsessori tuuma kohta. Esialgu teeme koormustesti 1000 päriguga sekundis.

```
11:38:03 root@VEGETA:~# echo "GET https://tools.euroland.com/tools/Ticker/Scrolling/GetGraphIntradayData/?sid=0.62239
991512733887&instrumentID=20634&lang=en-GB&decimalMarket=.&thousandGroupMarker=%2C&timezone=W.%20Europe%20Standart%20T
ime&defaultNumberFormat=%23%2C%23%230.00&companycode=uk-rdsa&getCleanData=false&v=1" | vegeta -cpus=16 attack -durati
on=10s -rate=1000 -workers=50 -timeout=60s | tee reports.bin | vegeta report
Requests      [total, rate, throughput]         10000, 1000.09, 897.20
Duration      [total, attack, wait]             10.001s, 9.999s, 1.981ms
Latencies     [min, mean, 50, 90, 95, 99, max]  45.804µs, 150.112ms, 2.288ms, 627.931ms, 1.466s, 1.685s, 2.037s
Bytes In      [total, mean]                     843462, 84.35
Bytes Out     [total, mean]                     0, 0.00
Success       [ratio]                           89.73%
Status Codes  [code:count]                      0:1027  200:8973
Error Set:
Get "https://tools.euroland.com/tools/Ticker/Scrolling/GetGraphIntradayData/?sid=0.6223999151273387&instrumentID=2063
4&lang=en-GB&decimalMarket=.&thousandGroupMarker=%2C&timezone=W.%20Europe%20Standart%20Time&defaultNumberFormat=%23%2
C%23%230.00&companycode=uk-rdsa&getCleanData=false&v=1": dial tcp 0.0.0.0:0->192.168.101.102:443: socket: too many op
en files
```

Joonis 11 Koormustesti tulemus konfigureeritud mitmelõimelisusega

Koormustesti tulemused on head, paremad kui vaikekonfiguratsiooniga, aga ei ole veel ideaalsed. Selleks, et simuleerida päris olukorda ja tuntavalt koormata protsessorit, käivitame testi 5000 päringuga sekundis.



Joonis 12 Protsessori läbilaskevõimsuse koormustest

```
11:42:56 root@VEGETA:~# echo "GET https://tools.euroland.com/tools/Ticker/Scrolling/GetGraphIntradayData/?sid=0.62239
991512733887&instrumentID=20634&lang=en-GB&decimalMarket=.&thousandGroupMarker=%2C&timezone=W.%20Europe%20Standart%20T
ime&defaultNumberFormat=%23%2C%23%230.00&companycode=uk-rdsa&getCleanData=false&v=1" | vegeta -cpus=16 attack -durati
on=60s -rate=5000 -workers=50 -timeout=10s | tee reports.bin | vegeta report
Requests      [total, rate, throughput]         300000, 5000.01, 4943.19
Duration      [total, attack, wait]             1m0s, 1m0s, 8.17ms
Latencies     [min, mean, 50, 90, 95, 99, max]  40.401µs, 13.133ms, 8.938ms, 18.3ms, 22.287ms, 46.765ms, 1.551s
Bytes In      [total, mean]                     27883314, 92.94
Bytes Out     [total, mean]                     0, 0.00
Success       [ratio]                           98.88%
Status Codes  [code:count]                      0:3369  200:296631
Error Set:
Get "https://tools.euroland.com/tools/Ticker/Scrolling/GetGraphIntradayData/?sid=0.6223999151273387&instrumentID=2063
4&lang=en-GB&decimalMarket=.&thousandGroupMarker=%2C&timezone=W.%20Europe%20Standart%20Time&defaultNumberFormat=%23%2
C%23%230.00&companycode=uk-rdsa&getCleanData=false&v=1": dial tcp 0.0.0.0:0->192.168.101.102:443: socket: too many op
en files
```

Joonis 13 Koormustesti tulemus 5000 päringuga sekundis

Ülalolevatel joonistel oleva koormustesti tulemus näitas, et protsessor on võimeline töötlema läbi 5000 päringut sekundis [Joonis 12]. Kahjuks tulemus ei ole ikka veel ideaalne. Joonisel [Joonis 13] saab näha, et 3369 päringut ei saanud vastust koormusjaoturist. Kui proovida veel konfiguratsioooni muuta ja käivitada veel koormusteste, võib leida parima konfiguratsiooni mis annab parima tulemuse koormustesti käivitades.

Koormustestide ajal tuli välja järgmine eripära. Serveri protsessor ei olnud konfigureeritud töötama nominaal sagedusega terve aja. See tähendab, et koormustesti alguses, kui päringud hakatakse saatma, ei ole koormusjaotur valmis neid vastu võtma,

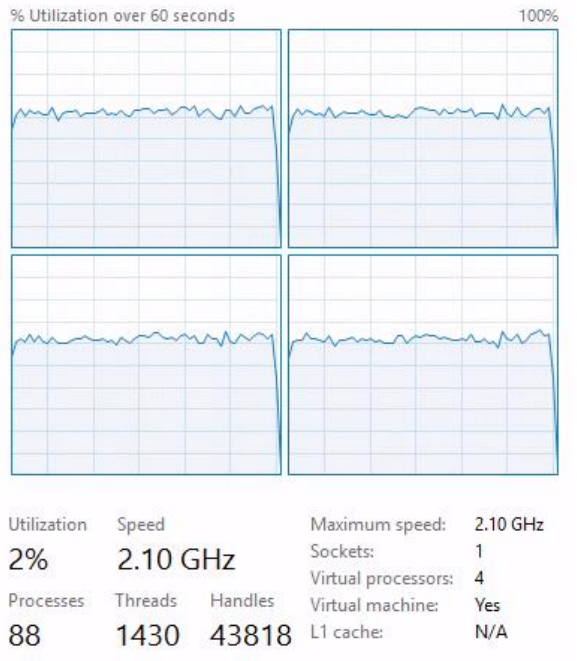
sest serveri protsessor töötab minimaalse sagedusega. Siit tulevad üksikud tõrked päringute töötlemises koormustesti alguses. Parima jõudluse saavutamiseks tuleb samuti õigesti konfigureerida operatsioonisüsteemi tuum. [26]

Proovides mitu erinevat konfiguratsiooni, leiti parima millega lõpptulemuseks koormustestid näitasid parema tulemuse.

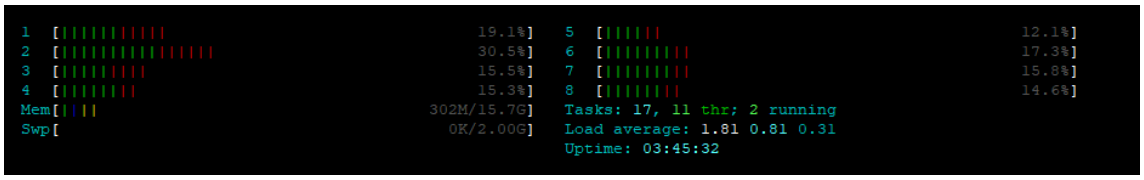
```
group haproxy
daemon
nbproc 1
nbthread 8
cpu-map auto:1/1-8 0-7
```

Joonis 14 Haproxy mitmelõimelisuse konfiguratsioon

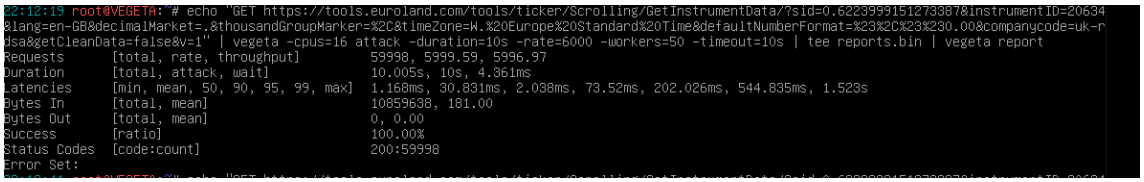
Joonisel [Joonis 14] säte „nbproc 1“ tekitab ühe peaprotsessi ja „nbthreat 8“ loob kaheksa lõime. „cpu-map“ on „auto“ seisundis ja jagab koormust võrdselt kõikide protsessori tuumade vahel.



Joonis 15 Veebiserveri protsessori koormus parima koormustesti tulemuse ajal



Joonis 16 Koormusjaoturi protsessori koormus parima koormustesti tulemuse ajal



Joonis 17 Parim koormustest

Ülalolevatel joonistel [Joonis 15, Joonis 16, Joonis 17] saab näha parimat tulemust viimase koormustesti läbiviimise ajal. Optimeeritud koormusjaoturi operatsioonisüsteemi tuuma konfiguratsioon aitas vähendada koormust protsessorile [Joonis 16]. Veebiserveri protsessori [Joonis 15] kasutamine on umbes 60%. Antud konfiguratsioon [Lisa 5] näitas paremat tulemust ja see on ka kasutusele võetud.

7 Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli võrrelda ja valida ülesande lahendamiseks sobivaim riistvara ja tarkvara veebiserverite, koormusjaoturite ja veebirakenduse tulemüüride paigaldamiseks.

Bakalaureusetöös lahendatavaks probleemiks oli uue infrastruktuuri projekteerimine ja ehitus. Töö käigus kirjeldati lahendusi mis võivad sobida probleemi lahendamiseks. Autor kirjeldas ja analüüsis olemasoleva infrastruktuuri probleeme ja puudusi ning nende parandamine ja vältimine projekteerimise käigus oli üks eesmärkidest.

Teoreetilises osas analüüsiti koormusjaoturite ja veebirakenduse tulemüüride turgu. Turuanalüüsi põhjal valiti sobivaim lahendus SWOT analüüsi põhjal. Põhiliseks nõudeks olid hästi dokumeteeritud tarkvara ja piisav tuntus, mis võimaldab probleemide tuvastamisel need interneti abil kiiresti lahendada. Koormusjaoturiks sai valitud Haproxy ja veebitulemüüriks tuli AQTRONiX WebKnight.

Praktilises osas paigaldati ja konfigureeriti uus süsteem. Seadistati serveri riistvara ja tarkvara ning viidi läbi koormustestid.

Töö käigus viidi tõhusalt ellu projekt, millel on paljutõotavad tulemused ja suur potentsiaal, et seda saab edukalt edasi arendada. Nimelt on plaanis veebiserveritest lahti saada ja seal olevad veebiteenused mahutada konteineritesse mis on kergesti skaleeritavad ja lihtsasti kontrollitavad. Veebiteenuste konteineriseerimine ei mõjuta autori tehtud tööd, kuna muudatust tehtakse ainult veebiserverite kihil uue süsteemi topoloogia skeemil [Joonis 4], ning ainuke muudatus, mis tuleb teha veebitulemüüri kihis, on tagaserveri IP aadressi muutmine veebitulemüüri konfiguratsioonis. Lisaks, klientide arvu tõusuga on plaanis lisada veel servereid, mille ülesandeks saab TLS koormuse vabastus ja koormuse jaotamine. Valmis ehitatud projekti kasu kajastus kohe tuntavalt lihtsamas seires ja halduses. Uuemate tehnoloogiatega saab täpsemini jälgida olukorda ja üksikasju, mis lisas projektile väärtust ja võtab vähem meeskonna tööaega probleemide avastamisel.

Bakalaureusetöös püstitatud eesmärk sai täidetud. Töö tulemuste hulka kuuluvad uue pinu ehitus, kõrgkäideldavus, horisontaalse skaleerimise võimalus, koormuse vähendamine võimsamate serverite ostmisega ja ettevõte teenuste stabiilne töö.

Kasutatud kirjandus

[1] EurolandIR. [Võrgumaterjal]. Available: https://services.euroland.com/about/.

[2] „Network Load Balancing," [Võrgumaterjal]. Available: https://learn.microsoft.com/en-us/windows-server/networking/technologies/network-load-balancing.

[3] R. Cooper. [Võrgumaterjal]. Available: https://www.loadbalancer.org/blog/windows-nlb-wnlb-and-its-disadvantages/.

[4] J. Marlin. [Võrgumaterjal]. Available: https://techcommunity.microsoft.com/t5/failover-clustering/deploying-network-load-balancing-nlb-and-virtual-machines-on/ba-p/371631.

[5] EDN, „1U vs. 2U vs. 3U: Rack Units Explained," [Võrgumaterjal]. Available: https://www.edn.com/what-does-1u-2u-or-3u-mean/.

[6] „PowerEdge R240 Rack Server," [Võrgumaterjal]. Available: https://www.dell.com/en-us/shop/enterprise-products/r240-1ru-server-intel/spd/poweredge-r240.

[7] „POWEREDGE R240 SPEC SHEET," [Võrgumaterjal]. Available: https://i.dell.com/sites/csdocuments/Product_Docs/en/poweredge-r240-spec-sheet.pdf.

[8] „PowerEdge R640 Rack Server," [Võrgumaterjal]. Available: https://www.dell.com/en-us/shop/servers-storage-and-networking/poweredge-r640-rack-server/spd/poweredge-r640/pe_r640_tm_vi_vp_sb.

[9] „Introduction to Microsoft Core licensing models," [Võrgumaterjal]. Available: https://www.microsoft.com/en-us/licensing/product-licensing/windows-server.

[10] „What is Azure Web Application Firewall on Azure Application Gateway?," [Võrgumaterjal]. Available: https://learn.microsoft.com/en-us/azure/web-application-firewall/ag/ag-overview.

[11] „Web Application Firewall CRS rule groups and rules," [Võrgumaterjal]. Available: https://learn.microsoft.com/en-us/azure/web-application-firewall/ag/application-gateway-crs-rulegroups-rules?tabs=owasp32.

[12] „OWASP ModSecurity Core Rule Set," [Võrgumaterjal]. Available: https://coreruleset.org/.

[13] „Cloudflare Web Application Firewall," [Võrgumaterjal]. Available: https://www.cloudflare.com/en-gb/waf/.

[14] M. Kozlowski, „modsecurity-vs-naxsi.md," [Võrgumaterjal]. Available: https://gist.github.com/marcinguy/3a106991d3a84995efacc473f8db21a9.

[15] H. Jethva, „HAProxy vs Nginx – What's the Difference?," [Võrgumaterjal]. Available: https://cloudinfrastructureservices.co.uk/haproxy-vs-nginx-whats-the-difference/.

[16] „HAProxy vs Traefik: What are the differences?," [Võrgumaterjal]. Available: https://stackshare.io/stackups/haproxy-vs-traefik.

[17] G. Dillon, „Benchmarking 5 Popular Load Balancers: Nginx, HAProxy, Envoy, Traefik, and ALB," [Võrgumaterjal]. Available:

https://www.loggly.com/blog/benchmarking-5-popular-load-balancers-nginx-haproxy-envoy-traefik-and-alb/.

[18] „Let's Encrypt,“ [Võrgumaterjal]. Available: https://letsencrypt.org/.

[19] H. Ashtari, „Horizontal vs. Vertical Cloud Scaling: Key Differences and Similarities,“ [Võrgumaterjal]. Available: https://www.spiceworks.com/tech/cloud/articles/horizontal-vs-vertical-cloud-scaling/.

[20] „Configuring the System Profile to Performance in BIOS,“ [Võrgumaterjal]. Available: https://www.dell.com/support/kbdoc/en-us/000132984/dell-technologies-configuring-the-system-profile-to-performance-in-bios.

[21] H. C. Manual. [Võrgumaterjal]. Available: http://cbonte.github.io/haproxy-dconv/2.5/configuration.html.

[22] „Prometheus HAProxy Prometheus Dashboard,“ [Võrgumaterjal]. Available: https://grafana.com/grafana/dashboards/2428-haproxy/.

[23] „Vegeta github page,“ [Võrgumaterjal]. Available: https://github.com/tsenart/vegeta.

[24] C. Faulet. [Võrgumaterjal]. Available: https://www.haproxy.com/blog/multithreading-in-haproxy/.

[25] S. Malhotra, „How we fine-tuned HAProxy to achieve 2,000,000 concurrent SSL connections,“ [Võrgumaterjal]. Available: https://www.freecodecamp.org/news/how-we-fine-tuned-haproxy-to-achieve-2-000-000-concurrent-ssl-connections-d017e61a4d27/.

[26] „Linux Kernel Tuning for Haproxy,“ [Võrgumaterjal]. Available: https://www.haproxy.com/documentation/hapee/latest/getting-started/system-tuning/.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Konstantin Dzyuba

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Veebiteenuse infrastruktuuri kaasajastamine ja arendus Euroland.com AS näitel“, mille juhendaja on Margus Sumla
   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.

24.04.2023

Lisa 2 – Haproxy paigaldamine Debian operatsioonisüsteemi

Kuna paigaldame Haproxy tarkvara mitte Debiani pakettide hoidlast, tuleb konfigureerida süsteemi Haproxy hoidlat.

Teeme lahti konfiguratsiooni faili ja kirjutame sisse hoidla informatsiooni:
vi /etc/apt/sources.list.d/haproxy.list

deb [signed-by=/usr/share/keyrings/haproxy.debian.net.gpg] http://haproxy.debian.net bullseye-backports-2.5 main

Paigaldame järgmise paketi:

apt install pgp

Kasutades pgp dešifreerime hoidla võtit:

curl https://haproxy.debian.net/bernat.debian.org.gpg | gpg --dearmor > /usr/share/keyrings/haproxy.debian.net.gpg

Ja paigaldame nüüd Haproxy:

apt update

*apt-get install haproxy=2.5.**

Lisa 3 – Haproxy koormusjaoturi eelkonfigureerimine

Eelseadistamise protsess kujutab endast statistika lehe sisselülitamist ning ees ja tagasüsteemide konfigureerimist.

Statistika lehe konfiguratsioon, kus määrame IP aadressi, URI ja kasutajanime koos parooliga, näeb välja järgmisena:

listen stats

bind 192.168.168.111:1936

mode http

stats enable

stats hide-version

stats realm LoadBalanced\ Servers

stats uri /haproxy?stats

stats auth admin:admin

Ees- ja tagasüsteemide konfigureerimine:

frontend http-in

mode tcp

bind 192.168. 168. 111:80

bind 192.168. 168. 111:443 ssl crt /etc/ssl/private/crtkey.pem

default_backend def

backend def

server 1 192.168. 168.20:80 check inter 500 fall 3 rise 2

Vaikekonfiguratsioon muudetud IP aadressidega:

global

daemon

maxconn 2048

stats   timeout 30s

user haproxy

group haproxy

tune.ssl.default-dh-param 2048

ssl-default-bind-options no-sslv3 no-tls-tickets

ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS

defaults

log global

mode tcp

option tcplog

timeout connect 5000

timeout client 10000

timeout server 10000

retries 3

errorfile 400 /etc/haproxy/errors/400.http

errorfile 403 /etc/haproxy/errors/403.http

errorfile 408 /etc/haproxy/errors/408.http

errorfile 500 /etc/haproxy/errors/500.http

errorfile 502 /etc/haproxy/errors/502.http

errorfile 503 /etc/haproxy/errors/503.http

errorfile 504 /etc/haproxy/errors/504.http

listen stats

bind 192.168.168.111:1936

mode http

stats enable

stats hide-version

stats realm LoadBalanced\ Servers

stats uri /haproxy?stats

stats auth admin:admin

frontend http-in

mode tcp

bind 192.168.168.111:80

bind 192.168.168.111:443 ssl crt /etc/ssl/private/crtkey.pem

default_backend def

backend def

mode tcp

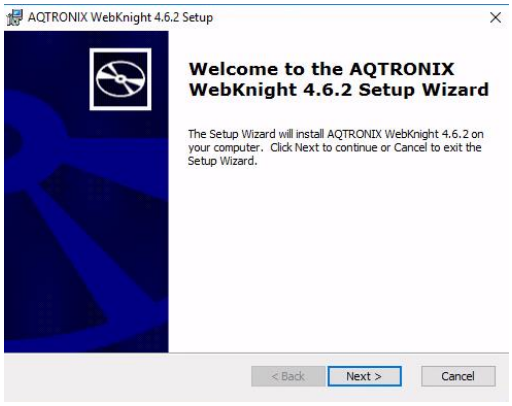
balance roundrobin

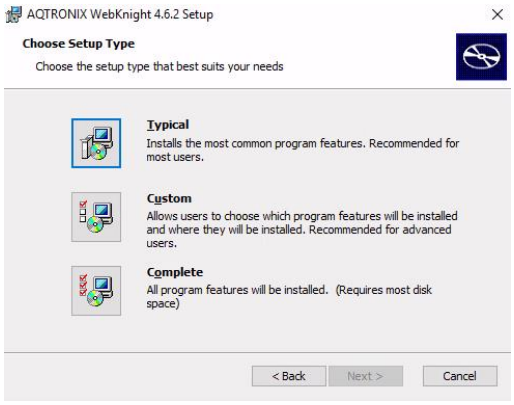
server 1 192.168.168.210:80 check inter 500 fall 3 rise 2

server 2 192.168.168.211:80 check inter 500 fall 3 rise 2

Lisa 4 – Veebitulemüüri paigaldamise ja konfigureerimise protsess

Veebitulemüüri paigaldamise protsess näeb välja järgmisena:
Kui litsents on ostetud, tuleb allalaadida paigaldamis fail ja teha see lahti.
Tuleb pöörata tähelepanu, kui IIS'is on sisselülitatud „jagatud konfiguratsioon", siis see tuleb väljalülitada enne veebitulemüüri paigaldamist.





Kui paigaldamise ajal ei tekkinud vigu, on veebitulemüür edukalt paigaldatud.

Lisa 5 – Haproxy koormusjaoturi stabiilne konfiguratsioon

```
global
        log /dev/log len 65535 local0
        log /dev/log    local1 notice
        chroot /var/lib/haproxy
        stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
        stats timeout 30s
        user haproxy
        group haproxy


                daemon
        nbproc 1
        nbthread 8
        cpu-map auto:1/1-8 0-7
                maxconn             100000
                hard-stop-after      1m
                tune.ssl.default-dh-param      2048
                tune.bufsize 32768


        # Default SSL material locations
        ca-base /etc/ssl/certs
        crt-base /etc/ssl/private

        ssl-default-bind-ciphers
ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES
:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS
        ssl-default-bind-options no-sslv3 ssl-min-ver TLSv1.2 no-tls-tickets

defaults
        log     global
        mode    http
                log-format
'{"host":"%H","ident":"haproxy","pid":%pid,"time":"%Tl","haproxy":{"name":{"frontend":
"%ft","backend":"%b","server":"%s"},"conn":{"act":%ac,"fe":%fc,"be":%bc,"srv":%sc},"q
ueue":{"backend":%bq,"srv":%sq},"time":{"response":%Td,"CRS":%Tq,"QW":%Tw,"TTW"
:%Tc,"R":%Tr,"ST":"%Tt"},"termination_state":"%tsc","retries":%rc,"network":{"client_ip
":"%ci","client_port":%cp,"frontend_ip":"%fi","frontend_port":%fp},"ssl":{"version":"%ssl
v","ciphers":"%sslc"},"request":{"method":"%HM","uri":"%[capture.req.uri,json(utf8s)]","
http_version":"%HV","header":{"host":"%[capture.req.hdr(0),json(utf8s)]","xforwardfor":"
%[capture.req.hdr(1),json(utf8s)]","user-
agent":"%[capture.req.hdr(2),json(utf8s)]"}},"response":{"status_code":%ST,"header":{"xr
equestid":"%[capture.res.hdr(0),json(utf8s)]"}},"bytes":{"request_size":%U,"response_size"
:"%B"}}}'
        timeout connect 5000
        timeout client  50000
        timeout server  50000
```

```
listen stats
    bind  192.168.168.111:1936
    mode http
    stats enable
    stats refresh 30
    stats admin if TRUE
    stats show-legends
    stats hide-version
    stats realm LoadBalanced\ Servers
    stats uri /haproxy?stats
    stats auth admin:admin
    timeout client 5000
    timeout connect 5000
    timeout server 5000



frontend frontend-public
    bind            125.122.11.55:80 name 125.122.11.55:80   tfo
    bind            2011:f90:fd10:1000::55:80 name 2011:f90:fd10:1000::55:80   tfo
    mode             http
    log             global
    option           logasap
    option           http-keep-alive
    option           forwardfor
    http-request capture req.hdr(Host) len 1000
    http-request capture req.hdr(Referer) len 1000
    http-request set-header      X-Forwarded-Proto http
    maxconn          100000
    timeout client       30000

    default_backend arr_backend


frontend frontendSSL-public-euroland-merged
    bind            125.122.11.55:443 name 125.122.11.55:443   ssl crt-list
/etc/haproxy/certs/frontend-cert-list.txt  tfo alpn h2,http/1.1
    bind            2011:f90:fd10:1000::55:443 name 2011:f90:fd10:1000::55:443   ssl
crt-list /etc/haproxy/certs/frontend-cert-list.txt  tfo alpn h2,http/1.1
    mode             http
    log             global
    option           logasap
    option           http-keep-alive
    option           forwardfor
    maxconn          100000
    timeout client       30000
    http-request capture req.hdr(Host) len 1000
    http-request capture req.hdr(Referer) len 1000
            http-request set-header      X-Forwarded-Proto https
```

```
    http-request set-header        X-Forwarded-Https on

    default_backend arr_backend


backend arr_backend
    mode                 http
    id                   100
    balance              roundrobin
    timeout connect      30000
    timeout server       30000
    retries              3
    option               httpchk
    http-check send meth GET uri /dynect.txt hdr Host tools.euroland.com hdr Accept */*
    http-check expect status 200
    http-request set-header X-Forwarded-Port %[dst_port]
    http-request set-header X-Client-IP %[src]

    server          ee-v-arr3 10.10.15.232:80 id 111 check inter 2000  weight 20
    server          webcat13 10.10.15.163:80 id 115 check inter 2000 disabled
    server          ee-v-webcat15 10.10.15.171:8080 id 116 check inter 2000 disabled
maxconn 1000
    server          ee-v-webcat16 10.10.15.172:8080 id 117 check inter 2000 disabled
maxconn 1000
    server          webcat11 10.10.15.160:80 id 101 check inter 2000 disabled
    server          webcat12 10.10.15.161:80 id 102 check inter 2000 disabled
    server          ee-arr1_80 10.10.15.230:80 id 109 check inter 2000  weight 40
    server          ee-v-arr1-1 10.10.15.231:80 id 118 check inter 2000  weight 40
```