

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Teele Kaal 206614IAAB

**Koormusjaoturi teenuse automatiseerimine
GitOps
printsiipe kasutades Eesti Energia AS näitel**

Bakalaureusetöö

Juhendaja: Siim Vene
MSc

Kaasjuhendaja: Mariliis Häälme
BSc

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: Teele Kaal

15.05.2023

Annotatsioon

Ettevõtte F5 Networks, Inc. toodete seas leidub koormusjaoturite teenuse pakkumiseks loodud platvorm Big-IP Local Traffic Manager. Eesti Energia AS soovib uurida, kuidas on võimalik Big-IP platvormi haldustegevusi automatiseerida ning komponentide hoiustamise struktuur platvormil üles seada nii, et saaks jagada rollipõhiseid ligipääsusi erinevatele meeskondadele, kes on seotud platvormil leiduvate ressurssidega.

Käesoleva bakalaureusetöö eesmärk on uurida, kuidas on võimalik läbi GitOps'i printsiipide rakendamise automatiseerida F5 Big-IP platvormi haldus ning uuele komponentide hoiustamise struktuurile üle liikumise protsess. Töö koosneb analüüsist, näidislahenduse teostusest ja järeldest. Analüüsi käigus tutvustatakse GitOps'i printsiipe, seatakse paika lõputöö skoop ning valitakse välja sobilik töövahend lahenduse loomiseks. Analüüsi käigus osutub sobivaimaks automatiseerimisvahendiks Ansible, millega luuakse näidislahendus.

Lahenduse teostuse osas kirjeldatakse selle valmimise protsessi ning testitakse uuele struktuurile ülemineku protsessi katsetamiseks loodud ressurssidega, kolimise ajal mõõdetakse selle mõju testressursside poolt pakutavale teenusele. Testimise tulemused on positiivsed, sest komponendid liigutati edukalt uuele struktuurile, nende funktsionaalsus säilis ja teenus oli protsessist mõjutatud ligikaudu 60 sekundit. Lõputöö järeldeste osas tuuakse näidislahenduse tulemustena välja automatiseerimise õnnestumine ning keskses koodihoidlas lähtekoodi hoiustamise abil versioonihalduse ja meeskondade tulemusliku koostöö tagamine Atlassian Bitbucketi näitel. Samuti tehakse ettepanekud lahenduse edasi arendamiseks läbi CI/CD vahendite kasutuselevõtu muudatuste automaatseks juurutamiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 34 leheküljel, 5 peatükki, 13 joonist, 1 tabelit.

Abstract

Automating Load Balancer Service Management Based on GitOps Principles by the Example of Eesti Energia AS

Big-IP Local Traffic Manager by F5 Networks, Inc. is a platform that makes it possible to provide load balancing services within organizations. Eesti Energia AS has requested research and a proof-of-concept solution about automating the management procedures of Big-IP platform and setting up the resource management setup on the platform in a way where role-based access control could be used.

The aim of this thesis is to create a proof-of-concept solution for automating the management of F5 Big-IP platform and the process of moving platform's components to the new management setup by following GitOps principles. The thesis consists of three main parts: analysis, creation of the proof-of-concept solution and results. During the analysis, an overview of GitOps principles as well as the scope of the thesis is provided and Ansible is chosen as the most suitable Infrastructure as Code tool.

Creation of the solution part describes the functionality of the solution and the testing procedure. The testing is performed by moving demonstration resources to a new management setup while monitoring the availability of the service provided by the test resources. The testing was successful as the demonstration components were moved to the new setup without any problems and the service was affected for around 60 seconds.

The results section brings out the success of automation as well as providing version management and more effective ways for teamwork by maintaining the source code in a code repository based on the example of Atlassian Bitbucket as the main results of the thesis. Additionally, using CI/CD tools for automated deployments is brought out as an option for future development.

The thesis is in Estonian and contains 34 pages of text, 5 chapters, 13 figures, 1 table.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendusliides
CI/CD	<i>Continuous Integration/Continuous Delivery</i> , muudatuste järjepidev ja automatiseeritud juurutamine
DDoS	<i>Distributed Denial-of-Service</i> , hajutatud teenustökestusrünne
DevOps	Tarkvaraarenduse kultuur, mis ühendab tarkvaraarenduse ja haldustegevused
GitOps	DevOpsist arenenud mõttemaailm, mis väärtustab infrastruktuuri koodina hoiustamist ühtses Git repositooriumis, meeskondade vahelise koostöö suurendamist läbi versioonihaldustarkvara ning järjepidevate muudatuste implementeerimist
JSON	<i>JavaScript Object Notation</i> , andmevahetusvorming
LTM	<i>Local Traffic Manager</i> , Big-IP platvormi osa koormusjaoturite seadistamiseks
Pipeline	CI/CD töövahendite poolt kasutatav tegevuste jada, mida kasutatakse tavaliselt erinevate muudatuste järjepidevaks juurutamiseks
Playbook	Ansible poolt sooritatavate ülesannete kirjeldamise fail
Play	Ülesandeid defineeriv ja teostav osa <i>playbook</i> failist
Provider	HashiCorp Terraformi poolt pakutud vahend teatud platvormi API'ga suhtlemiseks
REST	<i>Representational State Transfer</i> , tarkvaraarhitektuuri tüüp
URL	<i>Uniform Resource Locator</i> , võrguaadress
YAML	<i>Yet Another Markup Language</i> , inimloetav andmete hoiustamise keel

Sisukord

1 Sissejuhatus	10
2 Analüüs.....	12
2.1 GitOps'i mõttemaailm	12
2.2 Ülevaade F5 Big-IP platvormist ja selle kasutusest Eesti Energias	13
2.3 Eesti Energia poolt seatud nõuded ja ootused näidislahendusele	15
2.4 Võrdlev analüüs: HashiCorp Terraform vs Ansible	15
2.4.1 HashiCorp Terraform	15
2.4.2 Ansible.....	16
2.4.3 F5 Big-IP LTM ressursside haldamise võimaluste võrdlus	17
2.5 Näidislahenduse funktsionaalsuse ja skoobi määramine	19
3 Näidislahenduse teostus.....	21
3.1 Ansible <i>playbook</i> 'ide üldine ülesehitus	21
3.2 F5 Big-IP LTM ressursside haldamine Ansible abil	21
3.2.1 Rollide üldine struktuur ja funktsionaalsus	22
3.2.2 Aktiivse Big-IP serveri õla tuvastamine ja muutujasse salvestamine	24
3.2.3 LTM ressursside haldamine	25
3.2.4 LTM ressursside haldamiseks loodud <i>playbook</i>	27
3.3 Ansible abiga uuele struktuurile üleminemine	28
3.3.1 Rollid olemasolevate LTM komponentide seadistusfailide loomiseks	29
3.3.2 Rollid komponentide Common partitsioonist teise liigutamiseks.....	30
3.3.3 Ressursside Common partitsioonist teise liigutamiseks loodud <i>playbook</i>	32
3.4 Ansible näidislahenduse testressurssidega katsetamine	33
3.5 Lähtekoodi ja seadistusfailide keskses koodihoidlas hoiustamine	35
4 Järeldused	40
4.1 Loodud näidislahenduse tulemused.....	40
4.2 Ettepanekud lahenduse edasi arendamiseks	41
5 Kokkuvõte	42
Kasutatud materjalid.....	44

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	48
Lisa 2 - Aktiivse Big-IP öla tuvastamine Ansible'ga	49
Lisa 3 – Ressursside haldamiseks loodud main.yaml koormusjaoturite näitel	50
Lisa 4 – Ressursside YAML formaadis seadistusfailid koormusjaoturite näitel.....	51
Lisa 5 – Ülesanded ressursside haldamiseks koormusjaoturite näitel	52
Lisa 6 – Monitoride haldamiseks loodud ülesanded	53
Lisa 7 – Ressursside haldamise rolle täitev <code>playbook.yaml</code>	54
Lisa 8 – <i>Playbook'i</i> <code>playbook.yaml</code> struktuur.....	55
Lisa 9 – Olemasolevate komponentide andmete kogumine koormusjaoturite näitel.....	56
Lisa 10 – Olemasolevate komponentide seadistuste kirjutamine failidesse koormusjaoturite näitel.....	57
Lisa 11 – Ressursside Common partitsioonist teise partitsiooni liigutamiseks loodud ülesannete fail koormusjaoturite näitel.....	58
Lisa 12 – Ressursside Common partitsioonist teise liigutamine koormusjaoturite näitel	59
Lisa 13 – Ressursside liigutamiseks loodud <code>migration_playbook.yaml</code>	61
Lisa 14 - Ressursside liigutamiseks loodud <code>playbook'i</code> struktuur	62
Lisa 15 – Common partitsiooni katsetamiseks loodud ühend test-web	63
Lisa 16 – Katsetamiseks loodud test-web ressursside kättesaadavuse hindamiseks kasutatav http monitor	64
Lisa 17 – Common partitsiooni katsetamiseks loodud koormusjaotur test-web.....	65
Lisa 18 – Testimiseks täiendatud roll komponentide seadistusfailide genereerimiseks koormusjaoturite näitel.....	66
Lisa 19 – Katsetamiseks loodud <code>playbook migration_playbook_for_demo.yaml</code>	67
Lisa 20 – Pythoni skript <code>curl_test.py</code>	68
Lisa 21 – <i>Playbook'i</i> täitmise vältel terminali kuvatud informatsioon	69
Lisa 22 – <i>Playbook'i</i> jooksumise käigus loodud komponentide seadistusfailid	73
Lisa 23 – Skripti <code>curl_test.py</code> terminali tagastatud tulemused.....	75
Lisa 24 – Partitsiooni ansible-test liigutatud ressursid Big-IP kasutajaliideses	76

Jooniste loetelu

Joonis 1 LTM ressursside halduseks loodud rollide üldine loogika.....	26
Joonis 2 Monitoride haldamiseks loodud rolli loogika	27
Joonis 3 Olemasolevate LTM komponentide seadistuse kogumise rollide loogika.....	30
Joonis 4 LTM ressursside Common partitsioonist teise kolimiseks loodud rollide loogika	32
Joonis 5 Lähtekoodi hoiustamine Bitbucketi koodihoidlas	36
Joonis 6 Tõmbetaotluse koostamine.....	36
Joonis 7 Tõmbetaotlus vajab ülevaatamist enne selle master haruga ühendamist	37
Joonis 8 Tõmbetaotluse käigus muudetavate andmete kontrollimine	37
Joonis 9 Ülevaataja poolt heaks kiidetud tõmbetaotlust on võimalik ühendada teise haruga	37
Joonis 10 testing-changes-via-pull-request haru ühendamine master haruga	38
Joonis 11 Muudatused on peale harude ühendamist lisatud harusse master	38
Joonis 12 Bitbucketi repositooriumi poolt pakutav versioonihaldus.....	39
Joonis 13 GitOps'i printsiipide rakendamine komponentide haldamises.....	41

Tabelite loetelu

Tabel 1 Võrdlus HashiCorp Terraformi ja Ansible abil F5 Big-IP ressursside haldamisest	17
---	----

1 Sissejuhatus

IT süsteemide haldamises ja nende poolt pakutavate teenuste töökindluse tagamises on tänu DevOps'i mõttemaailma laiale levikule märgata aina rohkem lahendusi, kus infosüsteemide ülesehitamine ning edasine haldamine on viidud koodi kujule ja üha enam on hägustumas piirid arendus- ja haldusmeeskondade vahel [1]. DevOps'i liikumisest on välja kujunenud ka GitOps'i mõttemaailm, mis paneb rõhku infrastruktuuri koodi kujul keskselt hoiustamisele ning CI/CD vahenditega muudatuste järjepidevale juurutamisele [2]. GitOps'i printsiipide meeskondades kasutusele võtmine võimaldab tagada läbipaistva ning efektiivse töökeskkonna, mida iseloomustab infosüsteemide järjepidev areng [2].

Ettevõtte F5 Networks, Inc. toodete seas leidub koormusjaoturite teenuse pakkumiseks loodud platvorm Big-IP Local Traffic Manager [3]. Eesti Energia AS soovib uurida, millised on võimalused Big-IP platvormi automatiseeritud haldamiseks, et hoiduda manuaalsetest haldustegevustest. Tahetakse leida sobilikud haldusmeetodid, mis võimaldavad pidada järge muudatuste ajalool ning teha automatiseeritud muudatusi, mida saab vajadusel kiirelt tagasi keerata. Samuti soovitakse uurida, milliseid rollipõhiseid juurdepääsu reguleerimise võimalusi Big-IP pakub läbi platvormil komponentide hoiustamise struktuuri muutmise.

Käesoleva lõputöö eesmärgiks on luua näidislahendus sellest, kuidas on võimalik GitOps'i printsiipe osaliselt kasutusele võtta Big-IP platvormi haldamises ning automatiseeritult üle liikuda uuele komponentide hoiustamise struktuurile, mis võimaldab platvormi ressurssidega seotud meeskondadele rollipõhiseid ligipääsusi jagada. Teema on aktuaalne, sest Eesti Energias kehtib kokkulepe versioonihalduse tagamiseks hoiustada infrastruktuuri koodi kujul. Samuti aitab GitOps'i printsiipide rakendamine automatiseerida korduvaid tööülesandeid ja suurendada koormusjaoturi teenuse töökindlust.

Näidislahenduse planeerimiseks teostatakse esmalt analüüs, mille käigus seatakse paika lahenduse funktsionaalsus ja skoop ning valitakse eesmärkide täitmiseks sobilik

töövahend. Seejärel kirjeldatakse näidislahenduse teostust ja ülesehitust, mida katsetatakse testimiseks loodud ressurssidega. Lõpuks tuuakse välja näidislahenduse ja selle testimise tulemused ning tehakse ettepanekud lahenduse edasiseks arendamiseks.

2 Analüüs

Analüüsi peatükis tutvustatakse GitOps'i mõttemaailma ja lähenemisi, mida plaanitakse lõputöös käsitletud lahenduses rakendada. Seejärel antakse ülevaade F5 Big-IP Local Traffic Manageri (edaspidi LTM) kasutusjuhtudest ning tuuakse välja nõuded, mis seatakse Eesti Energia poolt näidislahendusele. Viimaks teostatakse võrdlev analüüs HashiCorp Terraformi ja Ansible kohta, mille käigus valitakse välja kõige sobilikum töövahend, ning seatakse paika lõputöö lahenduse funktsionaalsus ja skoop.

2.1 GitOps'i mõttemaailm

GitOps on kogum tavasid, mis liidab ühte DevOps'i metodoloogiad ja töövahendeid nii, et arendajatel on võimalik rohkem panustada IT haldamise ja operatiivsete tegevustega seotud protsessidesse. Sisuliselt on GitOps IT süsteemide infrastruktuuri koodi kujul hoiustamine, mis kasutab Git koodihoidlat ainsa tõeallikana ning samuti töövahendina meeskondade vahelise koostöö edendamiseks – kõikidel tiimidel on läbi tõmbetaoluste võimalik lähtekoodi muuta, kui soovitatav muudatus on vastutavate meeskondade poolt heakskiidu saanud [2]. GitOps kombineerib arendustsüklites juba kasutusel olevad automatiseerimise ning pideva muudatuste voo põhimõtted infrastruktuuri haldamisega, et muuta infrastruktuuri loomist ja muutmist kiiremaks ning järjepidevaks [4].

Infrastruktuur kui kood on üks GitOps'i alustaladest, mis aitab infrastruktuuri ehitamise ja haldamise muuta töökindlamaks ning kvaliteetsemaks. Selle saavutamiseks on defineeritud järgnevad kolm põhimõtet: iga infrastruktuuri komponent tuleb koodina defineerida, muudatusi tuleb järjepidevalt testida ja juurutada ning infrastruktuur peab olema jaotatud väiksemateks osadeks, mida on võimalik eraldiseisvalt muuta. Kogu infrastruktuuri koodina defineerimine aitab meeskondadel efektiivsemalt muudatusi juurutada, sest see võimaldab uute ressursside loomisel varasemat koodi taaskasutada, lähtekoodi abil ehitamine tagab infrastruktuurile alati ühesugused omadused ning kood toimib ka dokumentatsioonina, sest see kirjeldab ressursside ülesehitust ja seadistust.

Järjepidev muudatuste sisseviimine ja testimine aitab tiimidel süsteeme järk-järgult parendada ning seejuures kontrollida, et nõutud funktsionaalsus säilib. Seda toetab ka infrastruktuuri väiksemateks osadeks jaotamine – tehtavad muudatused on piiratud teatud ressursiga, mida on võimalik isoleeritult katsetada. [5]

GitOps'i teiseks oluliseks komponendiks on koodi hoiustamine Git koodihoidlas, mis võimaldab tagada versioonihalduse, parandada meeskondade vahelist koostööd ning luua infrastruktuuri ülesehitusele üheainsa tõeallika. Keskne repositoorium lähtekoodi hoiustamiseks garanteerib tööprotsesside läbipaistvuse – kõiki varasemaid tegevusi on võimalik muudatuste ajaloost uurida ja kontrollida. See pakub meeskondadele ka kindlustunnet muudatuste juurutamiseks, sest vigade tekkimisel on võimalik eelmine versioon koheselt taastada ning hilisemate probleemide ilmnemisel on võimalik vaadata, milliseid muudatusi on süsteemis hiljuti tehtud. [2]

GitOps'i puhul on keskne koodihoidla ainsaks tõeallikaks ning kasutusele on võetud automaatika, mis tagab repositooriumi sisu reaalse vastavuse infrastruktuuriga. Selle jaoks kasutatakse tavaliselt CI/CD töövahendeid, mis ehitatud *pipeline*'ide abil vajalikke muudatusi süsteemidesse sisse viivad. *Pipeline* 'i kasutamine võimaldab kiirelt muudatusi teha ning vajadusel eelmist seisu automatiseeritult taastada, samuti on võimalik lisada loogika vajaliku funktsionaalsuse säilimise kontrollimiseks. [2]

GitOps'i põhimõtete kasutusele võtmine F5 Big-IP LTM ressursside haldamises võimaldab pakkuda koormusjaoturite seadistamist teenuse osana, andes erinevatele meeskondadele võimaluse vastavalt vajadusele infrastruktuuri muuta. Lisaks võimaldab Big-IP LTM ressursside haldamise koodi kujule viimine ning lähtekoodi keskselt hoiustamine tagada koormusjaoturi teenuse suuremat töökindlust, sest koodihoidlas on alati näha infrastruktuuri hetkeolukord ja versioonihaldus võimaldab saada ülevaadet süsteemi ajaloost ning selle eelmistest seisudest.

2.2 Ülevaade F5 Big-IP platvormist ja selle kasutusest Eesti Energias

F5 Big-IP on ettevõtte F5 Networks, Inc. poolt pakutav rakenduste käideldavuse haldamise platvorm, mis võimaldab mõjutada rakendustele suunatud võrguliiklust ja selle turvalisust [6]. Big-IP tarkvara jaguneb kaheks osaks: Big-IP DNS suure liikluse ja DDoS rünnakute puhul rakenduste kõrge käideldavuse tagamiseks [7] ning Big-IP LTM

üleüldiseks rakenduste kõrge käideldavuse tagamiseks läbi koormuse jaotamise, võrguliikluse suunamise ja ressursside poolt pakutavate teenuste kättesaadavuse monitoorimise [3].

Big-IP LTM platvormil on kolm põhikomponenti, millega on võimalik rakenduste koormuse jaotamist seadistada: *Virtual Servers* ehk koormusjaoturid, *Nodes* ehk serverid, mis rakendust käitavad ja mis koormuse jaotamist vajavad ning *Pools* ehk ühendid, mis koondavad ühe koormusjaoturi taha servereid kokku [8]. Big-IP LTM platvormi koormusjaoturid on sisuliselt virtuaalsed IP aadressid, mille poole kasutajad teenuse kasutamiseks pöörduda saavad [9]. Koormusjaoturite seadistustes määratakse neile kasutamiseks ühendid, mis hõlmavad endas sama teenust pakkuvaid servereid, millele suunatakse edasi koormusjaoturile saabunud võrguliiklus.

Veel ühtedeks olulisteks Big-IP LTM komponentideks on *Monitors* ehk monitorid, mille abil on võimalik hinnata teiste LTM komponentide kättesaadavust. Monitorid kontrollivad ühenditesse koondatud serverite kättesaadavust ning vajadusel märgistavad probleemse serveri ära, et koormusjaotur ei suunaks liiklust serverile, mille poolt pakutav teenus pole antud ajamomendil kättesaadav [10]. F5 Big-IP platvormil komponentide hoiustamine ja nende ligipääsu jagamine toimub ressursi *Partitions* ehk partitsioonide läbi. Vaikimisi on Big-IP LTM platvormil olemas üks partitsioon Common, milles hoiustatavatele ressurssidele on kõikidele platvormi kasutajatele ligipääs alati tagatud. Erinevate komponentide ja nende ligipääsu eraldamiseks saab luua uusi partitsioone, millele ligisaavaid kasutajaid on võimalik piirata [11]. Seega on ka partitsioonid tihedalt seotud eelnevalt mainitud F5 Big-IP LTM platvormi ressursside haldamisega.

Eesti Energias kasutatakse F5 Big-IP LTM platvormi peamiselt ettevõtte siseste veebirakenduste ning arendajate tööd toetavate platvormide kõrge käideldavuse tagamiseks. Rakenduste jaoks on loodud koormusjaoturid, mis kasutavad ühenditesse koondatud füüsilisi või hüperviisoril jooksvaid virtuaalseid servereid, mis kindlat teenust ettevõtte töötajatele pakuvad. F5 Big-IP LTM kasutamine võimaldab tagada teenuste suuremat kättesaadavust probleemide tekkimise korral ning teha muudatusi või hooldustöid serverites nii, et kasutajate jaoks pole teenuse kättesaadavuses katkestusi.

2.3 Eesti Energia poolt seatud nõuded ja ootused näidislahendusele

Eesti Energia poolt on lõputöös kirjeldatavale lahendusele seatud järgnevad nõuded, mida tuleb silmas pidada:

- Big-IP LTM halduse koodi kujule viimiseks tuleb kasutada kas HashiCorp Terraformi või Ansible't ehk tuleb valida nende seast sobivaim töövahend.
- Näidislahendus peab toetama kõikide punktis 2.2 väljatoodud ressursitüüpide automatiseeritud haldamist.
- Versioonihalduse tagamiseks tuleb lähtekoodi hoiustada keskses koodihoidlas, mis võimaldab pidada järge muudatuste ajalool ning kirjeldab F5 Big-IP LTM platvormi komponentide kõige hiljutisemat seadistust.
- Lahendus peab ühtima kasutusel olevate Big-IP versioonidega.
- Automatiseeritud haldustegevustest tingitud koormusjaoturi teenuse katkestused peavad olema minimaalsed, et platvormil oleks vajadusel võimalik hallata ressursse, millega seatud teenuste käideldavus peab olema üle 80% ööpäevast.

2.4 Võrdlev analüüs: HashiCorp Terraform vs Ansible

Üheks Eesti Energia poolt projektile seatud tingimuseks on Ansible ja HashiCorp Terraformi vahel sobivaima automatiseerimisvahendi valimine. Nii Ansible kui ka HashiCorp Terraform pakuvad vahendeid F5 Big-IP ressursside haldamiseks läbi platvormi REST API'ga suhtlemise, järelkult on mõlema tööriista kasutamine teoreetiliselt võimalik. Sobiliku töövahendi välja valimiseks teostatakse analüüs, mis iseloomustab töövahendite omadusi ning nende sobivust F5 Big-IP LTM ressursside haldamise protsessidega ja võrreldakse Ansible ning HashiCorp Terraformi poolt F5 Big-IP LTM haldamiseks pakutavaid vahendeid.

2.4.1 HashiCorp Terraform

Terraform on ettevõtte HashiCorp poolt loodud vabavaraline töövahend, mis võimaldab infrastruktuuri koodi kujul hallata ning luua. Terraformi kasutades on võimalik defineerida meelepärane infrastruktuur HashiCorp Configuration Language keeles, mis realiseeritakse läbi vastavate platvormide API'dega suheldes. Terraformiga on võimalik hallata ressursse kõikidel platvormidel, millel on API ning mille jaoks on loodud *provider*

ehk komponent, mis oskab Terraformi failide sisu muundada antud API'ga suhtlemiseks vajalikeks päringuteks. [12]

HashiCorp Terraform on deklaratiivne töövahend, mille tõttu ei pea Terraformi failides defineerima kindlat järjekorda ja tegevusi, mida Terraformi abiga soovitakse teha, vaid failides defineeritakse vajalik tulemus ning selle täpne saavutamise viis jääb töövahendi enda ülesandeks. HashiCorp Terraform peab järke infrastruktuuri kõige hiljutisemast seisukorrast, mis võimaldab töövahendil otsustada, milliseid tegevusi tuleb soovitu saavutamiseks läbi viia. Sellest tingituna on Terraformi failide sisust alati võimalik saada täpset ülevaadet infrastruktuuri hetkeolukorrast. Deklaratiivsete töövahendite tegevuste jäiga olemuse tõttu on need kõige sobilikumad süsteemide üles ehitamiseks ja struktuuris muudatuste tegemiseks, mitte süsteemide poolt pakutavate teenuste seadistamiseks. [12]

Terraform lähtub muutmatu infrastruktuuri põhimõtetest. See tähendab, et kasutaja poolt Terraformi failides defineeritud muudatusi ei viida paljudel juhtudel läbi juba eksisteerivate ressursside peal, vaid olemasolevad ressursid asendatakse uutega, mille omadused on vastavuses Terraformi failides defineerituga. Muutmatu infrastruktuuri põhimõtete jälgimine võimaldab Terraformiga alati tagada ühesuguse süsteemide ülesehituse, sest ressursid luuakse sama šablooni põhjal ehk varasemate komponentide peal sooritatud tegevused või seadistused ei mõjuta lõpptulemust. Ressursside taasloomine võib teatud juhtudel olla üsna ajakulukas ning teenuseid oluliselt mõjutada. Seega on muutmatu infrastruktuuri põhimõtteid jälgivad töövahendid sobilikud olukordadeks, kus süsteemide ühetaoline ülesehitus on väga oluline ning kus ressursside vahepealne kustutamine ja seejärel taasloomine ei ole vastuolus teenustele seatud käideldavuse nõuetega. [12]

2.4.2 Ansible

Ansible on automatiseerimisvahend, mis on eelkõige mõeldud infrastruktuuri konfiguratsiooni koodina haldamiseks. Ansible võimaldab läbi erinevate moodulite infrastruktuuri seadistada ning luua, mis muudab selle paindlikuks ning võimsaks töövahendiks. Ansible kasutab YAML formaadis *playbook* faile, milles defineeritu põhjal viiakse infrastruktuuri peal vajalikke tegevusi läbi. [13]

Ansible on protseduuriline töövahend, mille tõttu tuleb *playbook* failides defineerida täpne järjekord ja vajalikud tegevused, mida Ansible infrastruktuuri peal läbi viib. Lisaks

ei pea protseduurilised töövahendid vaikimisi järke infrastruktuuri hetkeolukorrast tervikuna, vaid tööriista kasutaja peab ise teadma, milliseid tegevusi on varasemalt läbi viidud ning uusi muudatusi planeerides nendega arvestama. See muudab protseduurilised tööriistad kõige sobilikumateks olukordades, kus süsteemide hetkeolukord ei mõjuta oluliselt soovitud eesmärkide täitmist või on tarvis paindlikku automatiseerimise vahendit. Näiteks võimaldavad need mugavalt servereid seadistada ja nendele erinevat tarkvara paigaldada, kuid infrastruktuuri ülesehituse haldamine on raskendatud, sest vajalike tegevuste planeerimine eeldab täpset ülevaadet hetkeolukorrast. [12]

Ansible lähtub muudetava infrastruktuuri põhimõtetest, seega failides defineeritud muudatused ja tegevused viiakse üldiselt läbi juba eksisteerivate ressursside peal. Muudetava infrastruktuuri põhimõtetest kinni pidavad töövahendid on seega heaks abivahendiks süsteemide teenuste sobilikul moel ülesseadmiseks ja haldamiseks, kus komponendi täielik taasloomine ei ole eesmärkide saavutamiseks vajalik. [12]

2.4.3 F5 Big-IP LTM ressursside haldamise võimaluste võrdlus

F5 Big-IP LTM ressursside koodi kujul haldamiseks peab valitud töövahend toetama Big-IP API'ga suhtlemist. Nii Terraformil kui ka Ansible'1 on see funktsionaalsus olemas – Terraformi valikus on F5Networks Big-IP *provider* koos erinevate ressurssitüüpidega [14] ja Ansible valikus on moodulite kogum F5Networks.F5_Modules [15], mis võimaldab LTM ressursse hallata. Terraformi ja Ansible poolt F5 Big-IP LTM ressursside haldamiseks pakutavate vahendite võrdlus on esitatud tabelis 1.

Tabel 1 Võrdlus HashiCorp Terraformi ja Ansible abil F5 Big-IP ressursside haldamisest

LTM ressurss	Ansible	HashiCorp Terraform
Koormusjaoturid	Leidub moodul, mis võimaldab luua ja hallata koormusjaotureid ning neid täies mahus seadistada. [16]	F5 Big-IP <i>provider</i> valikus leidub ressurss, mis võimaldab Terraformiga koormusjaotureid hallata. Ansible mooduliga võrreldes leidub vähem parameetreid, mida on võimalik seadistada. Puudu on järgmised parameetrid: <i>firewall_enforced_policy</i> ,

		<i>firewall_staged_policy, policies, rate_limit, rate_limit_dst_mask, rate_limit_mode, rate_limit_src_mask, security_nat_policy</i> ja <i>type</i> . [17]
Serverid	Serverite jaoks leidub kaks moodulit: üks moodul serverite jaoks, mis ei kuulu mõne ühendi koosseisu ja teine ühendisse kuuluvate serverite loomiseks. Terraformiga võrreldes ei ole Ansible moodulitega võimalik seadistada parameetri <i>dynamic_ratio</i> väärtust. [18, 19]	Serverite jaoks eksisteerib kaks Terraformi ressursi: üks serveri ühendisse lisamiseks ja teine ühendisse mitte kuuluvate serverite haldamiseks. Ansible moodulitega võrreldes leidub vähem seadistusparameetreid, puudu on järgmised parameetrid: <i>availability_requirements, ip_encapsulation, monitors, fqdn_auto_populate</i> . [20, 21]
Ühendid	Ühendite haldamiseks on loodud moodul, millega on võimalik ühendeid täies ulatuses seadistada. [22]	Ühendite haldamiseks leidub Terraformi ressurss, millel on Ansible mooduliga võrreldes puudu seadistusparameeter <i>monitor type</i> . [23]
Monitorid	Monitoride puhul on eraldi moodulid erinevatele monitoride tüüpidele, kuid kõikidele Big-IP LTM valikus olevatele monitoride tüüpidele ei ole mooduleid. Moodulid on olemas järgnevatele monitoride tüüpidele: HTTP, HTTPS, ICMP, LDAP, MySQL, Oracle,	Monitoride haldamiseks on loodud Terraformi ressurss, mis võimaldab hallata kõiki Big-IP LTM ressursside valikus leiduvaid monitoride tüüpe. [24]

	SMTP, SNMP DCA, TCP, TCP echo, TCP haf-open ja UDP. [15]	
Partitsioonid	Partitsioonide haldamiseks on olemas moodul, millega saab kõiki parameetreid seadistada [25].	Partitsioonide haldamiseks puudub Terraformil ressurss ehk neid pole võimalik Terraformiga luua ega hallata [14].

Tabeli põhjal saab järeldada, et nii Terraformi kui ka Ansible puhul leidub viise F5 Big-IP LTM ressursside haldamiseks, kuid enamikel juhtudel võimaldavad Ansible moodulid ressursse täpsemini seadistada. Lisaks eksisteerib Ansible moodul partitsioonide haldamiseks, mida Terraform ei toeta. Ansible monitoride moodulid jäävad Terraformi poolt pakutule alla, sest eri tüüpe monitore saab hallata nende spetsiifiliste moodulitega ning kõikide monitoride tüüpide jaoks ei ole mooduleid loodud. Terraformi puhul on võimalik sama ressursiga hallata kõiki eksisteerivaid monitoride tüüpe.

Terraformi eeliseks on selle punktis 2.3.1 kirjeldatud deklaratiivne olemus, mis võimaldab Terraformi failidest alati saada täpset ülevaadet infrastruktuuri seisust. Lisaks nõuavad teatud muudatused eksisteeriva komponendi kustutamist ning seejärel taasloomist. Näiteks on selline vajadus ressursi Common partitsioonist teise liigutamisel [11] ning komponendi ümber nimetamisel [26]. Terraformi kasutades toimub varasema ressursi kustutamine ning uue taasloomine automaatselt. Ansible puhul tuleb süsteemi seisukorra talletamise ja komponentide vahepealse kustutamise loogika ise luua.

Kahe töövahendi võrdlemisel kerkivad esile järgnevad Ansible tugevad küljed: LTM komponentide täpsemad seadistamise võimalused ja võimalus hallata kõiki ressursse, mis on üheks ettevõtte poolt näidislahendusele seatud nõudeks. Selle tõttu kasutatakse lõputöös käsitletava lahenduse loomisel Ansible't.

2.5 Näidislahenduse funktsionaalsuse ja skoobi määramine

Käesolevas lõputöös kirjeldatava lahenduse eesmärgiks on automatiseerida F5 Big-IP LTM ressursside haldus GitOps'i printsiipidele tuginedes, keskendudes ressursside halduse koodi kujule viimisele, mida on võimalik hakata hoiustama keskses koodihoidlas.

Lisaks on plaan tulevikus platvormil rakendada rollipõhiste ligipääsude jagamist, mis eeldab selleks sobiliku komponentide hoiustamise struktuuri kasutusele võtmist. Seega tegeletakse lõputöö raames kahe protsessiga – LTM ressursside halduse koodi kujule viimisega ning uuele struktuurile ülemineku automatiseerimisega.

F5 Big-IP LTM ressursside haldamise koodi kujule viimine hõlmab endas mitut aspekti. Kõigepealt tuleb paika seada, kuidas on Ansible abiga võimalik erinevaid LTM komponente luua ning redigeerida. Seejärel tuleb olemasolevad komponendid ja nende seadistus viia koodi kujule, et edasisi muudatusi oleks võimalik teha läbi lähtekoodi muutmise. Seega koosneb LTM ressursside haldamise koodikujuliseks muutmine automatiseeritud haldamise loogika paika seadmisest ning olemasolevate komponentide halduse koodi kujule viimisest.

Ühelt komponentide hoiustamise struktuurilt teisele üle liikumiseks tuleb kõigepealt viia olemasolevate ressursside konfiguratsioon koodi kujule, seejärel valmistada ette uus hoiustamise ülesehitus, millele olemasolevad komponendid migreeritakse. Big-IP platvormil on ressursse võimalik jaotada erinevatesse partitsioonidesse, mille abil saab piirata seda, millised kasutajad antud komponentidele ligi saavad ning millised ressursid saavad platvormil üksteisega suhelda. Uue komponentide hoiustamise struktuuri ehk olemasolevate ressursside erinevatesse partitsioonidesse eraldamise täpne ülesehitus jääb lõputöö skoobist välja. Eesmärgiks on uuele ülesehitusele ülemineku protsess automatiseerida nii, et see toetaks tulevikus erinevaid lähenemisi uue struktuuri paika panekuks.

3 Näidislahenduse teostus

Kolmandas peatükis antakse ülevaade Ansible *playbook*'ide üldisest ülesehitusest ja funktsionaalsusest. Lisaks kirjeldatakse näidislahenduse valmimist, mille käigus kirjeldatakse automatiseeritult ressursside haldamise, olemasolevate ressursside halduse koodi kujule viimise ning uuele komponentide hoiustamise ülesehitusele migreerimise protsesse. Seejärel katsetatakse kirjeldatud lahendust testimiseks loodud ressurssidega ning antakse ülevaade lähtekoodi keskses koodihoidlas hoiustamise võimalustest.

3.1 Ansible *playbook*'ide üldine ülesehitus

Ansible kasutab YAML formaadis *playbook* faile, mille sees on defineeritud hulk *play*'sid. *Play*'d koosnevad ülesannetest, mille põhjal viib Ansible erinevate moodulitega soovitud tegevused täide. Moodulid on oma olemuselt kogum skripte, mida on võimalik Ansible pakettidena alla tõmmata ning mis võimaldavad erinevaid tegevusi defineeritud serverite peal läbi viia. Linuxi distributsioonidele loodud Ansible moodulid on kirjutatud Python programmeerimise keeles, seega on Ansible kasutamiseks vajalik Pythoni olemasolu nii selle poolt hallatavates serverites kui ka *playbook*'e käivitavas serveris. [13]

Ansible *playbook*'ides tuleb defineerida serverid, mille peal ülesanded läbi viiakse. Ülesandeid on võimalik sooritada nii teiste serverite peal kui ka lokaalselt serveris, milles *playbook*'e käivitatakse. Lisaks on ülesannete eraldamiseks ning nende hilisemaks taaskasutuseks võimalik luua rolle, mis täidavad selle piires defineeritud ülesandeid. Loodud rolle on võimalik *playbook*'ides välja kutsuda, mis muudab *playbook*'id kergemini loetavamaks ning võimaldab vastavalt soovile teostatavaid ülesandeid grupeerida. [13]

3.2 F5 Big-IP LTM ressursside haldamine Ansible abil

Punktis 2.4 teostatud analüüsist selgub, et Ansible pakub erinevaid moduleid, millega on võimalik F5 Big-IP LTM ressursse hallata. Iga ressursitüübi jaoks on eraldi moodul, millega saab antud tüüpi komponente seadistada. Analüüsist selgub, et ainsaks erandiks

on monitorid – erinevatele monitoride tüüpidele on loodud eraldiseisvad moodulid, kuid kõikidele Big-IP platvormil leiduvatele monitoride variantidele ei ole mooduleid loodud. Seega tuleb monitoride haldamiseks leida lahendus, mis võimaldaks siiski kõikvõimalikke monitoride tüüpe Ansible abiga hallata.

Ansible'ga ressursside haldamiseks peab olema võimalik uusi komponente luua, olemasolevaid muuta ning neid ka soovi korral kustutada. Lisaks on F5 Big-IP platvormi käitavad kaks serverit seadistatud klastrisse, millest üks õlg on aktiivne ja teine on ootel. Klastrisisese seadistuste konflikti vältimiseks tuleb muudatusi teha aktiivse õla peal ning need kantakse teatud aja jooksul üle ka ootel õlale. Seega peab Ansible enne ressurssidega toimetama asumist ka määrama, millisel õlal tuleks muudatused sisse viia.

Ressursside haldamiseks ja aktiivse Big-IP õla tuvastamiseks luuakse eraldi rollid, et grupeerida kokku sama ressursitüübiga seotud ülesanded. See võimaldab hoida *playbook*'i lühikese ja arusaadavana, sest põhiloogika on kirjutatud rollidesse. Seega luuakse LTM ressursside haldamiseks kuus rolli – *determine_active_bigip_node*, *partitions*, *pools*, *monitors*, *virtual_servers* ja *nodes*.

3.2.1 Rollide üldine struktuur ja funktsionaalsus

Punktis 2.4 teostatud analüüsis kirjeldatud Ansible Big-IP moodulid on oma ülesehituselt võrdlemisi sarnased – moodulid suhtlevad platvormi imperatiivse API'ga [27], ressursitüüp on defineeritud mooduliga, erinevad konfiguratsiooni parameetrid võimaldavad komponente seadistada ning *provider* parameeter võimaldab määrata F5 Big-IP API'ga suhtlemiseks vajaliku informatsiooni [15]. Kuna F5 Big-IP LTM moodulid suhtlevad platvormi REST API'ga ehk nende raames sooritatakse HTTP päringuid ja Ansible'ga Big-IP platvormi serveritesse ühendumiseks pole vajadust, siis defineeritakse *playbook*'is parameetri *host* väärtuseks „localhost“, mis viitab *playbook*'i käivitavale serverile. Lokaalsele käivitamisele viitamiseks seatakse ka rollides parameetrite *connection* ja *delegate_to* väärtusteks vastavalt „local“ ja „localhost“, nagu on kirjeldatud ka Big-IP Ansible kollektsiooni dokumentatsioonis [28].

Big-IP platvormil leiduvate komponentide seadistuste kirjeldamiseks tuleb luua lahendus, mis võimaldab platvormi ülesehitust ja seadistust talletada nii, et Ansible saab andmeid ülesannetes kasutada. Ansible pluginate valikus leidub pistikprogramm *lookup*, mis võimaldab kasutada mõnest välisest infoallikast pärinevaid andmeid. Näiteks saab *lookup*

pistikprogrammiga lugeda informatsiooni erinevatest failidest [13]. Ansible pakub ka erinevaid filtreid, mille abil on võimalik sisendeid töödelda teistesse formaatidesse [13]. Ansible paigaldusel tuleb vaikimisi kaasa filter *from_yaml*, mis konverteerib sisendina saadud YAML struktuuriga sõne sõnastiku kujul muutujaks, mille sisu on võimalik võtme-väärtus paaride abil pärida [29]. Seega on F5 Big-IP LTM komponentide seadistusi võimalik hoiustada YAML failides, milles leiduvat informatsiooni on võimalik ülesannetes kasutada tänu *lookup* pluginale ning *from_yaml* filtrile.

Punktis 2.4 kirjeldatud analüüsi tulemustest selgub, et monitoride haldamiseks tuleb leida teistest ressurssidest erinev lahendus, sest kõikidele Big-IP platvormil eksisteerivatele monitoride liikidele ei leidu mooduleid. Big-IP API toetab kõikide monitoride tüüpide haldamist [30], seega saab kõikide monitoride liikide koodi kujul haldamiseks kasutada F5 Big-IP monitoride moodulite asemel *ansible.builtin.uri* moodulit, mis võimaldab Ansible'1 Big-IP REST API'ga läbi HTTP päringute suhelda. [31].

Et muuta rollid dünaamilisemaks ja taaskasutatavaks, kasutatakse rollides Ansible muutujaid, mis võimaldavad samu ülesandeid erinevate omadustega ressursside peal kasutada. Lisaks on muutujate abil võimalik vältida korduvate andmete erinevatesse rollidesse sisse kirjutamist, sest kattuva informatsiooni defineerimine muutujatena loob keskse tõeallika, kust kõikide rollide ülesanded saavad vajalikke andmeid lugeda. Informatsiooni muutujates talletamine teeb ka nende väärtuste muutmise lihtsamaks – vajalik muudatus tuleb teha ainult muutuja defineerimisel. Näidislahenduse kattuvad muutujad defineeritakse *group_vars* kataloogis asuvate failide abil, mis on üks Ansible muutujate kataloogide vaikeväärtustest [13]. Lisaks kasutatakse rollides jooksvalt *playbook*'i tegevuste käigus defineeritud muutujaid.

F5 Big-IP platvormi teatud versioonides esineb programmiviga, mis ei võimalda muuta Big-IP REST API päringute ajalõpu 60-sekundilist vaikeväärtust. Selle tõttu võib API'le saadetud suuremate päringutega esineda probleeme, kus Big-IP poolset vastust oodates ületatakse ajalimiiti ning päring lõpeb veateatega. Tegelikult võib päring olla edukas, kuid API ei jõua defineeritud ajapiirangu jooksul päringu õnnestumise vastust tagastada [32]. Antud programmiviga mõjutab ka näidislahenduses käsitletavat Big-IP versiooni. Ainsaks lahenduseks probleemile on Big-IP versiooni muutmise, mis on vastuolus lahendusele seatud nõudega selle ühtivusest kasutusel oleva Big-IP versiooniga. Selle tõttu kasutatakse kõikides rollides ülesandeid defineerides parameetrit *until*, mis

võimaldab seadistada tingimuse ülesande kordamiseks seni, kuni ülesanne õnnestub. Korduste arvu on võimalik piirata parameetri *retries* abil, et vältida ülesande igavesti kordamist [13]. Parameetri *until* kasutamine on sobilik hädalahendus API ajalõpu limiidile, sest kui üks ülesande täitmise katse lõpeb ajalimiidi ületamise tõttu veateatega, siis kordusega saadetud päringu vastus saab juba API poolt kinnitava vastuse ja Ansible jätkab *playbook*'i tegevustega.

3.2.2 Aktiivse Big-IP serveri õla tuvastamine ja muutujasse salvestamine

Big-IP platvormi käitamiseks on kasutusel kaks klatriks ühendatud serverit ning muudatusi tuleb teha aktiivse õla peal. Big-IP platvormi kohta informatsiooni kogumiseks eksisteerib moodul *f5networks.f5_modules.bigip_device_info*, mille parameetriga *gather_subset* on võimalik defineerida, millist informatsiooni soovitakse platvormilt saada ning Big-IP REST API tagastab vastavad andmed JSON formaadis [33]. Ansible võimaldab tagastatud vastuse parameetriga *register* muutujasse salvestada, tänu millele on ühe ülesande poolt tagastatud vastust võimalik kasutada mõnes teises *playbook*'i ülesandes [13].

Ansible'1 on erinevaid viise *playbook*'i tegevuste käigus jooksvalt muutujate loomiseks. Üheks valikuvariandiks on moodul *ansible.builtin.set_fact*, mille ülesandes kasutamine võimaldab defineerida muutuja, mis kehtib terve edasise *playbook*'i jooksmise vältel ehk ühes rollis *set_fact* mooduliga defineeritud muutujat on võimalik kasutada kõikides järgnevatel rollides [13]. Järelikult saab *f5networks.f5_modules.bigip_device_info* ja *set_fact* moodulite kombinatsiooniga koguda Big-IP platvormilt vajalikku informatsiooni aktiivse õla kohta ning registreerida see muutujasse, mida saab seejärel ka kõikides teistes rollides Big-IP serveri määramiseks kasutada.

Big-IP platvormi aktiivse õla tuvastamiseks ja salvestamiseks loodud roll *determine_active_bigip_node* (vt Lisa 2, lk 49) koosneb kolmest ülesandest. Esimene ülesanne kogub mooduli *f5networks.f5_modules.bigip_device_info* abil informatsiooni platvormi käitavate serverite kohta [33], saadud vastus registreeritakse muutujasse *device_facts*. Platvormi käest informatsiooni küsimisel võib Ansible suhelda nii ootel kui aktiivse õlaga, selle tõttu võib ülesandes serveri defineerimisel kasutada mõlema serveri domeeninime või IP aadressi.

Teine ülesanne töötleb esimeses ülesandes defineeritud muutuja *device_facts* sisu ning sorteerib sellest välja aktiivse Big-IP öla, mille väärtus salvestatakse *set_fact* mooduli abil muutujasse *bigip_active_node*. Muutujat *bigip_active_node* saab ressursside haldamiseks loodavates rollides kasutada Big-IP serveri defineerimiseks, et Ansible teeks muudatusi aktiivse öla peal. Rolli kolmas ülesanne kasutab Ansible moodulit *ansible.builtin.debug*, et kuvada kasutajale käsurea terminali muutuja *bigip_active_node* väärtus [13].

3.2.3 LTM ressursside haldamine

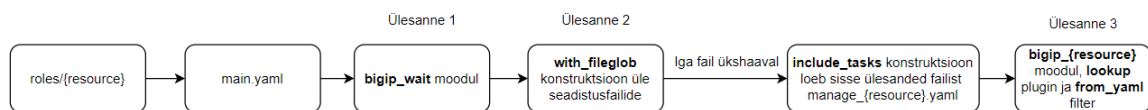
Koormusjaoturite, serverite, ühendite ja partitsioonide haldamiseks luuakse eraldi rollid, mis kasutavad nendele vastavat F5 Big-IP Ansible moodulit, et komponente luua, seadistada ja soovi korral kustutada. Nagu kirjeldatud punktis 3.2.1, hoiustatakse ressursside andmeid YAML failides, millest ülesanded vajaliku informatsiooni kätte saavad. Selleks, et rollis defineeritud ülesandeid kõikide sama liiki komponentide seadistusfailidega läbi viia, kasutatakse Ansible *with_fileglob* konstruktsiooni, mis võimaldab iteratiivselt faile ülesannetes kasutada [13].

Vaikimisi töötavad Ansible iteratiivsed ülesannete täitmised nii, et kõigepealt täidetakse üks ülesanne kõikide elementidega ning seejärel liigutakse järgmiste ülesannete juurde, mis samuti viiakse täide kõikide komponentidega enne edasi liikumist. Selleks, et ressursside haldamiseks vajalikud ülesanded viia järjestikuselt läbi ühe komponendi peal, on võimalik kasutada *with_fileglob* iteratsiooni kombinatsioonis *include_tasks* konstruktsiooniga. Konstruktsioon *include_tasks* võimaldab ühe ülesande raames välja kutsuda ülesandeid, mis on defineeritud mõnes teises ülesannete failis [13] ning kõik sisse loetud ülesanded sooritatakse järjestikuselt neid välja kutsuva ülesande raames.

F5 Big-IP LTM ressursside haldamiseks loodud rollid *nodes*, *virtual_servers*, *partitions* ja *pools* on oma ülesehituselt sarnased – rolli raames täidetavad ülesanded on defineeritud kahes failis, millest esimene käib iteratiivselt üle komponentide YAML formaadis seadistusfailide ning täidab nendega teises failis kirjeldatud ülesandeid, mis ressursitüübile vastava mooduliga neid haldab. Ressursside haldamiseks loodud ülesanded kasutavad punktis 3.2.1 kirjeldatud *lookup* pluginat ja *from_yaml* filtrit, et erinevate parameetrite väärtuseid seadistusfailidest kätte saada.

Igas rollis on komponentide iteratiivselt haldamiseks loodud ülesannete fail *main.yaml* (vt Lisa 3, lk 50), mis koosneb kahest ülesandest. Esimene ülesanne kasutab moodulit

f5networks.f5_modules.bigip_wait, et enne API'le ressursside haldamiseks saadetavate päringute saatmist veenduda, et Big-IP platvorm on saadaval ning valmis päringute põhjal muudatusi tegema [34]. Teine ülesanne viib *with_fileglob* konstruktsiooniga kogutud failidega läbi ülesanded, mis saadakse *include_tasks* konstruktsiooni abil teisest failist. Konstruktsioon *with_fileglob* käib iteratiivselt üle kataloogi, mis hoiustab komponentide YAML seadistusfaile. YAML failides defineeritavad parameetrid sõltuvad ressursside liigist, näiteks koosnevad koormusjaoturite seadistusfailid kahekümne ühest parameetrist (vt Lisa 4, lk 51). Neid parameetreid kasutatakse *include_tasks* konstruktsiooniga välja kutsutud ülesannetes, mis vastavalt saadud väärtustele komponente haldavad (vt Lisa 5, lk 52). Joonis 1 kujutab LTM ressursside haldamiseks loodud loogikat.

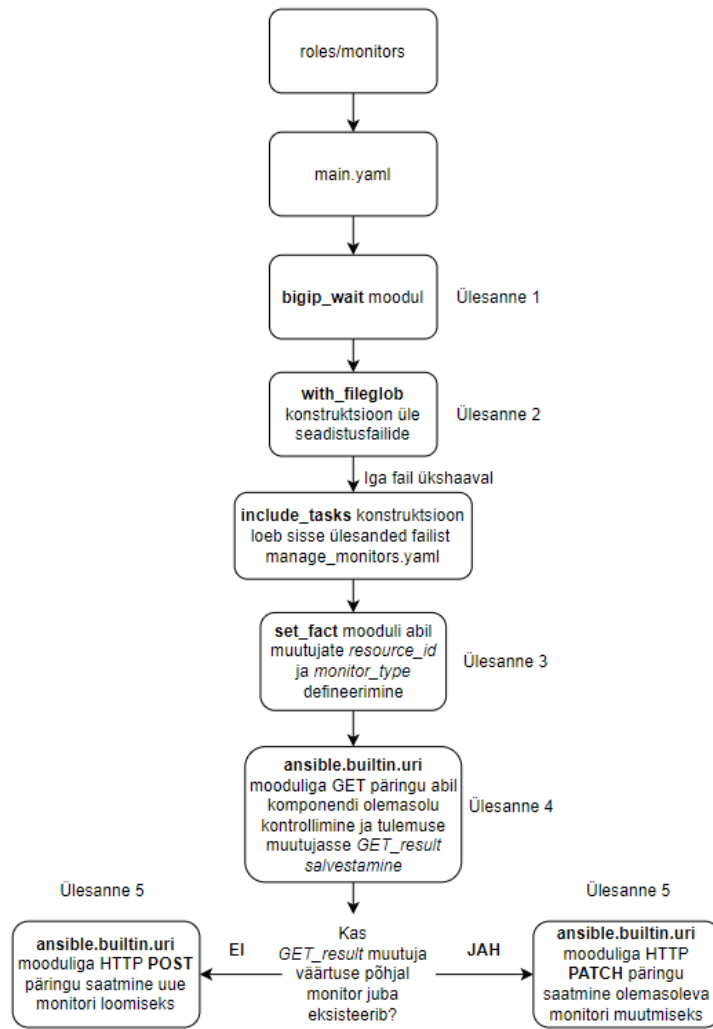


Joonis 1 LTM ressursside halduseks loodud rollide üldine loogika

Monitoride puhul kasutatakse nende loomiseks ja haldamiseks *ansible.builtin.uri* moodulit, et REST API'le vajalikke HTTP päringuid saata ning kõikide erinevate monitoride liikide haldamist ühe rolliga toetada. Selle tõttu on monitoride rolli ülesehitus veidi erinev teiste LTM ressursside haldamiseks loodud rollidest. Valminud roll *monitors* koosneb samuti kahest ülesandest kirjeldavast failist, millest esimene kutsub teiste rollidega sarnaselt välja ülesandeid teisest failist, mida viiakse iteratiivselt *with_fileglob* konstruktsiooniga määratud seadistusfailidega läbi.

Monitoride haldamiseks loodud ülesannete fail (vt Lisa 6, lk 53) koosneb neljast ülesandest. Esimene ülesanne defineerib *set_fact* mooduli abil komponendi ID [35] muutujasse *resource_id* ning monitori tüübi muutujasse *monitor_type*. Monitori tüübist sõltub HTTP päringu saatmiseks kasutatav URL ning ID abil saab API komponente üksteisest eristada. Teine ülesanne kontrollib HTTP GET päringu abil, kas kirjeldatud ressurss juba eksisteerib platvormil. Tulemus talletatakse parameetri *register* abil muutujasse *GET_result*, mida kasutatakse järgnevate ülesannete täitmise tingimuse määramiseks. Kui komponenti veel platvormil ei eksisteeri, tehakse kolmanda ülesande raames HTTP POST päring ressursi loomiseks. Kui kirjeldatud monitor juba eksisteerib,

siis tehakse kolmanda ülesande raames seadistuste muutmiseks HTTP PATCH päring. Joonis 2 kujutab monitoride haldamiseks loodud rolli *monitors* loogikat.



Joonis 2 Monitoride haldamiseks loodud rolli loogika

3.2.4 LTM ressursside haldamiseks loodud *playbook*

LTM ressursside haldamiseks loodud rolle on võimalik kasutada *playbook*'is. Loodud *playbook.yaml* (vt Lisa 7, lk 54) kutsub rollid eraldi *play*'dena välja ning igale *play*'le on lisatud ka parameeter *tag*, mis võimaldab soovi korral *playbook.yaml* käivitades piirata, milliseid rolle Ansible täidab [13]. Roll *determine_bigip_active_node* tuleb käivitada igal *playbook*'i jooksumise korral, sest selle rolli ülesannete käigus defineeritakse muutuja *bigip_active_node*, mida teised rollid kasutavad Big-IP serveri määramiseks. Kui *determine_bigip_active_node* rolli ei käivitata, siis muutujat *bigip_active_node* ei

määrata ning ülejäänud ülesanded ebaõnnestuvad. Joonis *playbooki* struktuurist on välja toodud lõputöö lisades (vt Lisa 8, lk 55).

Loodud *playbook* koosneb kuuest *play*'st, mida käivitades on võimalik erinevaid Big-IP LTM ressurssidega seotud haldamise ülesandeid automatiseeritult läbi viia. Fail *playbook.yaml* kasutab muutujaid, kõikides rollides kasutatavad muutujad *bigip_user* ja *bigip_password* ning rollis *determine_active_bigip_node* kasutatav muutuja *bigip_server* loetakse sisse kataloogis *group_vars* asuvas muutujate failist *all.yaml*, ülejäänud muutujad defineeritakse dünaamiliselt *playbook*'i jooksmise käigus.

3.3 Ansible abiga uuele struktuurile üleminemine

Teiseks näidislahenduses käsitletavaks protsessiks on automatiseeritult uuele komponentide hoiustamise struktuurile üleminek Ansible abiga ehk olemasolevate LTM ressursside kolimine uuele ülesehitusele. Big-IP platvormil hoiustatakse komponente partitsioonides ning ressursside eraldamiseks on võimalik neid erinevatesse partitsioonidesse liigutada [11]. Punktis 2.4 teostatud võrdlevast analüüsist selgub, et Big-IP platvormil on vaikumisi olemas partitsioon Common, milles asuvatele ressurssidele on kõikidel kasutajatel ligipääs ning lisaks saavad Common partitsioonis asuvatele komponentidele ligi kõik teistes partitsioonides paiknevad ressursid. Common partitsioonis asuvaid komponente ei ole võimalik liigutada teise partitsiooni, vaid need tuleb kõigepealt Common partitsioonist kustutada ja seejärel meelepärases asukohas taastada [11]. Seega koosneb komponentide uude partitsiooni liigutamise protsess järgnevatest osadest: ressursi Common partitsioonist kustutamine ning ressursi uues partitsioonis loomine samasuguse seadistusega. Ainult ühendite ressursse on võimalik uues partitsioonis luua ilma olemasoleva kustutamiset [11].

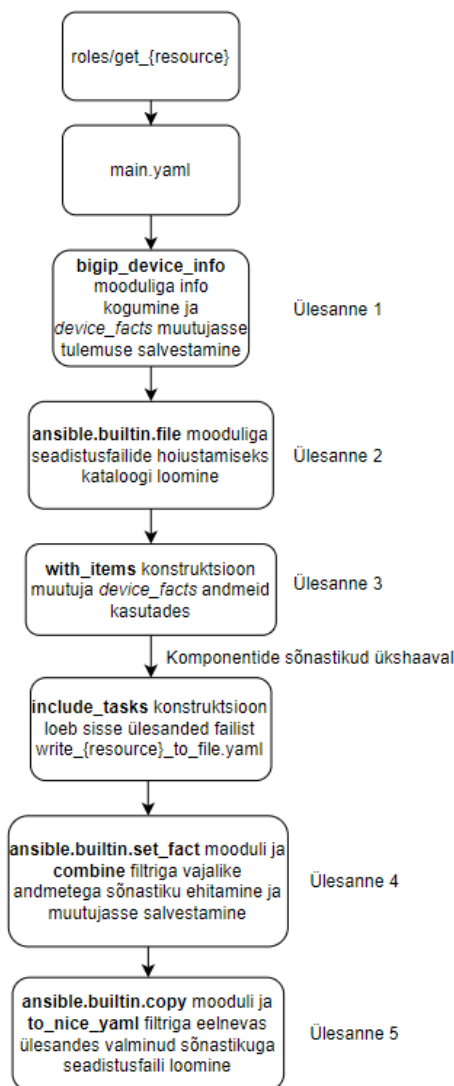
Ressursside Common partitsioonist uuele kolimiseks on võimalik kasutada sama loogikat nagu punktis 3.2 kirjeldatud LTM ressursside haldamiseks loodud rollide puhul – komponendid ja nende seadistused on kirjeldatud YAML formaadis seadistusfailidega, mille põhjal Ansible Big-IP API'ga suhtleb. Olemasolevate komponentide seadistuste saamise jaoks tuleb luua rollid, mis võimaldavad Ansible abiga genereerida YAML formaadis seadistusfaile, mida saab komponentide liigutamise ülesannetes kasutada. Lisaks tuleb kolimisprotsessi jooksvalt kontrollida, et vältida olukorda, kus olemasolevad komponendid eemaldatakse Common partitsioonist, kuid uusi ei õnnestu asemele luua.

3.3.1 Rollid olemasolevate LTM komponentide seadistusfailide loomiseks

Ansible F5 Big-IP moodulite valikus on moodul *f5networks.f5_modules.bigip_device_info*, mida on kasutatud punktis 3.2.2 kirjeldatud Big-IP aktiivse õla määramise rollis. Antud mooduliga on võimalik koguda informatsiooni ka platvormil leiduvate ressursside kohta, kui parameetriga *gather_subset* defineerida huvipakkuv ressursitüüp [33]. API tagastab ressursside kohta informatsiooni JSON formaadis ja iga komponendi seadistused on sõnastiku kujul [33], millest on võimalik vajalikke andmeid välja sorteerida. Ansible võimaldab *to_nice_yaml* filtriga sõnastikke YAML kujule töödelda [36] ehk vajalikud komponentide andmed on võimalik koguda kokku sõnastikku, mille saab YAML formaati töödelda ning seejärel faili kirjutada.

Olemasolevate LTM ressursside seadistusfailide kirjutamiseks on loodud neli rolli *get_monitors*, *get_nodes*, *get_pools* ja *get_virtual_servers*, mille raames sooritatavad ülesanded on defineeritud kahe ülesannete faili abil. Esimene ülesannete fail *main.yaml* (vt Lisa 9, lk 56) koosneb kolmest ülesandest. Esimene ülesanne kasutab moodulit *f5networks.f5_modules.bigip_device_info*, et koguda Big-IP platvormilt rollile vastava ressursitüübi kohta andmeid ning salvestab API poolt tagastatud informatsiooni parameetri *register* abil muutujasse. Teises ülesandes veendutakse *ansible.builtin.file* mooduliga [37], et on olemas kataloog, kus komponentide seadistusfaile hoiustada. Kolmanda ülesande raames kutsutakse *include_tasks* konstruktsiooniga välja teises ülesannete failis defineeritud ülesanded, mis viiakse iteratiivselt ükshaaval läbi esimeses ülesandes muutujasse salvestatud komponentide sõnastikega.

Teises ülesannete failis (vt Lisa 10, lk 57) defineeritud esimene ülesanne töötleb ressursside andmeid sisaldavat sõnastikku ning loob selle põhjal filtriga *ansible.builtin.combine* [38] uue sõnastiku, mis sisaldab komponendi taasloomiseks vajalikku informatsiooni. Seejärel kasutatakse moodulit *ansible.builtin.copy* [39] ja *to_nice_yaml* filtrit, et esimeses ülesandes koostatud sõnastik viia YAML formaati ning kirjutada saadud sisu faili. Komponentide kohta välja sorteeritud informatsioon ja selle põhjal loodud YAML seadistusfaili ülesehitus on samasugune nagu punktis 3.2.3 kirjeldatud konfiguratsioonifailide sisu (vt Lisa 4, lk 51). Joonis 3 illustreerib LTM olemasolevate komponentide seadistusfailide loomiseks tehtud rollide loogikat.



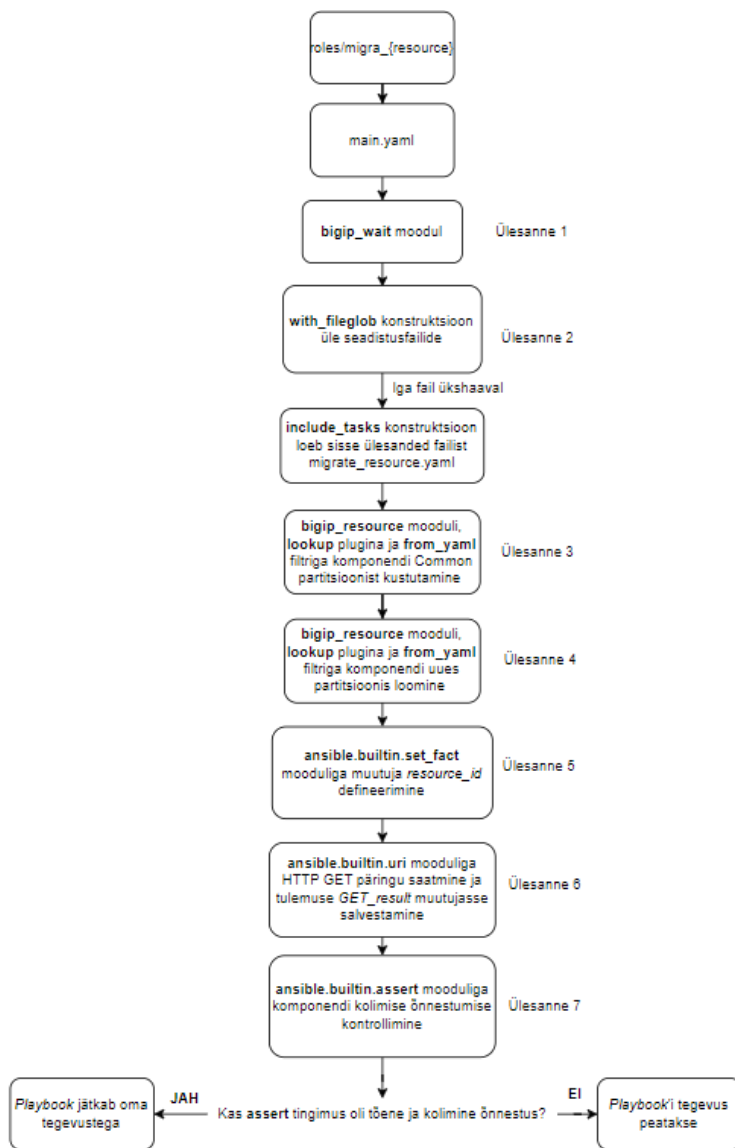
Joonis 3 Olemasolevate LTM komponentide seadistuse kogumise rollide loogika

3.3.2 Rollid komponentide Common partitsioonist teise liigutamiseks

Big-IP platvormil rollipõhiste ligipääsuõiguste rakendamiseks tuleb komponendid eraldada erinevatesse partitsioonidesse ehk kolida need Common partitsioonist välja. LTM ressursside Common partitsioonist uude liigutamise rollid *migra_monitor*, *migra_node* ja *migra_virtual_server* on oma ülesehituselt sarnased punktis 3.2.3 kirjeldatud LTM komponentide haldamiseks loodud rollidega – esimene ülesannete fail *main.yaml* (vt Lisa 11, lk 58) käib iteratiivselt üle komponentide seadistusfailide ning täidab nendega ükshaaval konstruktsiooniga *include_tasks* teisest ülesannete failist sisse loetud ülesandeid. Teises failis (vt Lisa 12, lk 59) leiduvad ülesanded kasutavad ressursitüübile vastavat moodulit ja punktis 3.3.1 kirjeldatud rollide abil loodud komponentide seadistusfaile, et ressurss kõigepealt Common partitsioonist kustutada

ning seejärel teises partitsioonis uuesti luua. Sarnaselt monitoride haldamise rolliga, kasutab monitoride uuele partitsioonile liigutamise roll samuti *ansible.builtin.uri* moodulit, et toetada kõiki monitoride liike.

Peale ressursside teises partitsioonis loomise ülesande täitmist kasutatakse *ansible.builtin.uri* moodulit, et saata HTTP GET päring ressursi liigutamise õnnestumise kontrollimiseks. GET päringu tulemus salvestatakse *register* parameetri abil muutujasse, mida kasutatakse viimases ülesandes kolimise protseduuri õnnestumise hindamiseks mooduliga *ansible.builtin.assert* [40]. Kui GET päring on edukas, siis kuvatakse terminali kolimise õnnestumise teade ja jätkatakse *playbook*'i jooksutamisega. Kui GET päring ebaõnnestub, kuvatakse terminali teade komponendi liigutamise ebaõnnestumise kohta ning *playbook*'i jooksutamine peatatakse. Ressursside kolimise regulaarne kontrollimine võimaldab uuele struktuurile ülemineku protsessi muuta turvalisemaks, sest ühe ressursiga probleemide tekkimisel peatatakse koheselt Ansible tegevused, et vältida teiste ressurssidega probleemide juurde tekkimist. Joonis 4 kujutab ressursside Common partitsioonist teise kolimise loogikat.



Joonis 4 LTM ressursside Common partitsioonist teise kolimiseks loodud rollide loogika

Nagu eelnevalt välja toodud, ei nõua ühendite Common partitsioonist teise liigutamine eksisteeriva komponendi kustutamist. Seega on ühendite puhul ressursside uues partitsioonis loomise ja Common partitsioonist eemaldamise ülesanded eraldatud kahte rolli *migra_new_pool* ja *migra_delete_pool*, mis oma funktsionaalsuse poolest on sarnased teiste ressursside liigutamise rollidega – Big-IP Ansible ühendite mooduli abiga luuakse või kustutatakse vastav komponent.

3.3.3 Ressursside Common partitsioonist teise liigutamiseks loodud *playbook*

Komponentide Common partitsioonist teise liigutamiseks loodud rolle on võimalik kasutada *playbook*'is, mis automatiseeritult kolimise läbi viib. Ressursside liigutamise

protsess näeb välja järgmine: aktiivse Big-IP õla määramine, olemasolevate komponentide seadistuse YAML failidesse kirjutamine, uue partitsiooni loomine, ühendite loomine uues partitsioonis, ülejäänud ressursitüüpide liigutamine Common partitsioonist uude partitsiooni, ühendite Common partitsioonist kustutamine ning platvormilt uuesti komponentide seadistuste pärimine konfiguratsioonifailide õigete andmetega üle kirjutamiseks.

Näidislahenduse raames loodud *migration_playbook.yaml* (vt Lisa 13, lk 61) koosneb ühest *play*'st, mis kasutab punktides 3.3.1 ja 3.3.2 kirjeldatud rolle, et liigutada ressursid Common partitsioonist uude partitsiooni ansible-test. Sarnaselt LTM ressursside haldamiseks loodud *playbook*'iga *playbook.yaml*, kasutatakse *migration_playbook.yaml* raames *group_vars* kataloogis paiknevat *all.yaml* faili muutujate *bigip_user*, *bigip_password* ja *bigip_server* väärtuste saamiseks, teised muutujad defineeritakse dünaamiliselt ülesannete läbimise käigus. *Playbook*'i *migration_playbook.yaml* struktuur on välja toodud lõputöö lisades (vt Lisa 14, lk 62).

3.4 Ansible näidislahenduse testressurssidega katsetamine

Näidislahenduse testimiseks kasutatakse punktis 3.3 kirjeldatud LTM ressursside Common partitsioonist teise liigutamiseks loodud rolle, sest kolimise protseduur võimaldab katsetada mõlemat lõputöös käsitletavat protsessi – ressursside haldamise rollidest pärinev loogika on kasutusel komponentide kustutamisel ja taasloomisel ning testressurssidega on võimalik testida komponentide liigutamise mõju teenusele. Katsetamiseks luuakse spetsiaalsed ressursid, mis võimaldavad lahendust katsetada ilma teisi platvormi komponente mõjutamata ning mille teenuse kättesaadavust on võimalik kolimise ajal monitorida.

Katsetamiseks luuakse kaks Oracle Linux operatsioonisüsteemiga virtuaalserverit *test-web01* ja *test-web02*, mille IP aadressid on vastavalt 192.168.145.114 ja 192.168.145.115 ning millele on paigaldatud veebiserver, mis ootab pordil 80 HTTP päringuid. Virtuaalserverite jaoks luuakse Common partitsiooni ühend *test-web*, mis koondab enda liikmeteks serveritele *test-web01* ja *test-web02* vastavad *Node* ressursitüübi instantsid, mille nimed on seadistatud IP aadresside järgi (vt Lisa 15, lk 63). Ühendi liikmete kättesaadavuse hindamiseks määratakse neile monitoriks Common partitsioonil asuv *http*

monitor, mis saadab 5-sekundiliste intervallidega ühendi liikmetele HTTP GET päringu (vt Lisa 16, lk 64). Big-IP platvormile luuakse koormusjaotur *test-web*, mille IP aadressiks määratakse 192.168.145.46 ning mis seadistatakse kasutama ühendit *test-web* (vt Lisa 17, lk 65). Seega on võimalik koormusjaoturi *test-web* poole pöördudes kasutada veebiteenust, mida pakuvad virtuaalserverid *test-web01* ning *test-web02*.

Loodud Ansible rollide ainult testressursside peal rakendamise piiramiseks lisatakse punktides 3.3.1 kirjeldatud rollidele parameetri *when* abil filtreeriv tingimus (vt Lisa 18, lk 66), mis sorteerib platvormil leiduvate ressursside seast välja katsetamiseks loodud *test-web* instantsid. Katsetamise läbiviimiseks luuakse eraldi rollid, millele on seatud vastavad filtreerivad tingimused, ja *playbook*, mis vastavaid rolle välja kutsub. Komponentide liigutamise testimisest jäetakse välja *http* monitori liigutamine teise partitsiooni, sest selle Common partitsiooni jätmisega saab katsetada teisest partitsioonist Common partitsioonis paiknevatele ressurssidele viitamist, mis võimaldab ühtivaid ressursse erinevate partitsioonide vahel jagada [25].

Katsetamiseks loodud *playbook migration_playbook_for_demo.yaml* (vt Lisa 19, lk 67) kasutab testimiseks tehtud rolle, et luua komponentide seadistusfaile, liigutada ressursid Common partitsioonist uude partitsiooni ansible-test ning seejärel kirjutada seadistusfailid uute andmetega üle. *Playbook*'i jooksmise ajal käivitatakse käsurealt Pythoni programm *curl_test.py* (vt Lisa 20, lk 68), mis kasutab Pythoni moodulit *requests* [41], et saata 15-sekundilise intervalliga GET päringud *test-web* koormusjaoturile. Programm *curl_test.py* aitab kolimise protsessi käigus monitoorida testressursside poolt pakutava veebiteenuse kättesaadavust ja selle läbi hinnata liigutamise mõju teenusele.

Playbook'i *migration_playbook_for_demo.yaml* käivitamisel terminali kuvatud informatsioonist (vt Lisa 21, lk 69) on näha, et kõigepealt tuvastatakse aktiivne Big-IP õlg ning seejärel jätkatakse testressursside seadistusfailide loomisega. Ülesannete käigus jäetakse vahele kõik komponendid, mis ei kuulu testressursside hulka. Kui informatsioon on Big-IP platvormilt edukalt kogutud, luuakse uus partitsioon ansible-test ja selles uus ühend *test-web* ning kasutatakse Ansible'ga loodud seadistusfaile (vt Lisa 22, lk 73), et Common partitsioonis eksisteerinud testressursid taasluua ansible-test partitsioonis. Kui kõik komponendid on uude partitsiooni liigutatud, kustutatakse *test-web* ühend Common

partitsioonist ning käivitatakse taas ülesanded komponentide seadistusfailide andmete üle kirjutamiseks.

Playbook'i *migration_playbook_for_demo.yaml* jooksutamine võttis aega ligikaudu 13 minutit, millest suurem osa kulus Big-IP platvormilt ressursside kohta info pärimiseks ning saadud andmete töötlemiseks. *Playbook*'i jooksutamise alguses käivitati terminalis Pythoni skript *curl_test.py*, et hinnata testressursside poolt pakutava teenuse kättesaadavust. Skripti poolt terminali tagastatud informatsiooni (vt Lisa 23, lk 75) põhjal on näha, et teenus muutus kättesaadamatuks momendil, kui *test-web01* ja *test-web02* virtuaalserverid kustutati partitsioonist Common ja muutus taas kättesaadavaks siis, kui koormusjaotur *test-web* oli partitsioonis ansible-test taasloodud. Skripti poolt 15-sekundilise intervalliga tehtud päringutest selgub, et liigutamise ajal ebaõnnestusid neli järjestikust päringut ning teenuse kättesaadavus oli komponentide kolimisest mõjutatud ligikaudu 60 sekundit.

Peale *playbook*'i jooksmise lõppu on võimalik Big-IP kasutajaliidesest kontrollida, kas uus partitsioon ansible-test ja sellele liigutatud testressursid on ilmunud nähtavale. Sooritatud kontrolli põhjal (vt Lisa 24, lk 76) on näha, et *playbook*'i *migration_playbook_for_demo.yaml* abiga tehtud automatiseeritud kolimine oli edukas, sest testressursid paiknevad uues partitsioonis ansible-test ning on protsessi käigus säilitanud oma seadistuse ja funktsionaalsuse.

3.5 Lähtekoodi ja seadistusfailide keskses koodihoidlas hoiustamine

Üheks Eesti Energia poolt näidislahendusele seatud ootuseks on versioonihalduse tagamiseks hoiustada valminud lähtekoodi keskses koodihoidlas, mis toetab ka GitOps'i printsiipi kasutada Git repositooriumi ainsa tõeallikana. Lõputöö raames katsetatakse koodi hoiustamist Atlassian Bitbucketis, kuhu luuakse uus repositoorium Ansible-Intern, millesse lükatakse näidislahendusega seotud failid. Joonis 5 illustreerib projekti ülesehitust.

Source

master ... Ansible-Intern / Browse Filter

Source	Description	Size	Last Modified
demo_conf_files			
demo_roles			
group_vars			
hosts			
request_script			
roles			
.gitignore	testing pools	20 B	12 Aug 2022
ansible.cfg	changed cfg file	273 B	31 Aug 2022
Jenkinsfile	script for GET requests	2.04 KB	06 Sep 2022
migration_playbook.yaml	some modifications	568 B	02 Sep 2022
migration_playbook_for_demo.yaml	fixing file writings	696 B	02 Sep 2022
playbook.yaml	minor edits	1008 B	23 Aug 2022
README.md	updated README	227 B	02 Sep 2022

Joonis 5 Lähtekoodi hoiustamine Bitbucketi koodihoidlas

Bitbucketis erinevate meeskondade vahelise koostöö edendamiseks on võimalik luua uusi harusid, millesse saab üles ehitada eraldiseisvaid struktuure ning seejärel tõmbetaotlustega avaldada soovi kahe haru omavahel ühendamiseks. Näiteks olukorras, kus soovitakse luua Big-IP platvormile uus koormusjaotur ja ühend, on võimalik vajalikud seadistusfailid lisada uuel harul *testing-changes-via-pull-request* paikneva konfiguratsioonifailide hoiustamise kataloogi *demo_conf_files*. Seejärel saab alustada tõmbetaotluse koostamisega (vt Joonis 6), mille raames avaldatakse soovi *testing-changes-via-pull-requests* haru ühendamiseks *master* haruga.

Create pull request

Collaborate on code by choosing teammates to review your changes to a branch.

Source

A Ansible / Ansible-Intern testing-changes-via-pull-request

Teete Kaal authored 9f923a179e3 A moment ago

Destination

A Ansible / Ansible-Intern master

Teete Kaal authored 0329b475aa8 23 mins ago

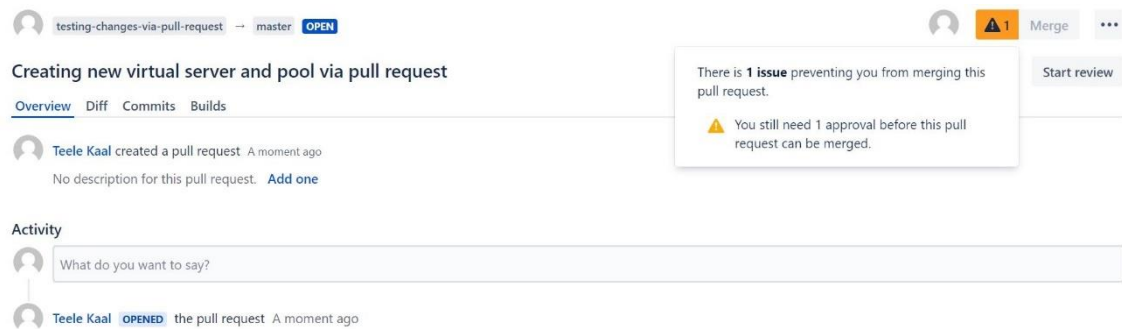
Continue

Diff Commits

Author	Commit	Message	Commit date	Builds
Teete Kaal	9f923a179e3	Creating new virtual server and pool via pull request	A moment ago	

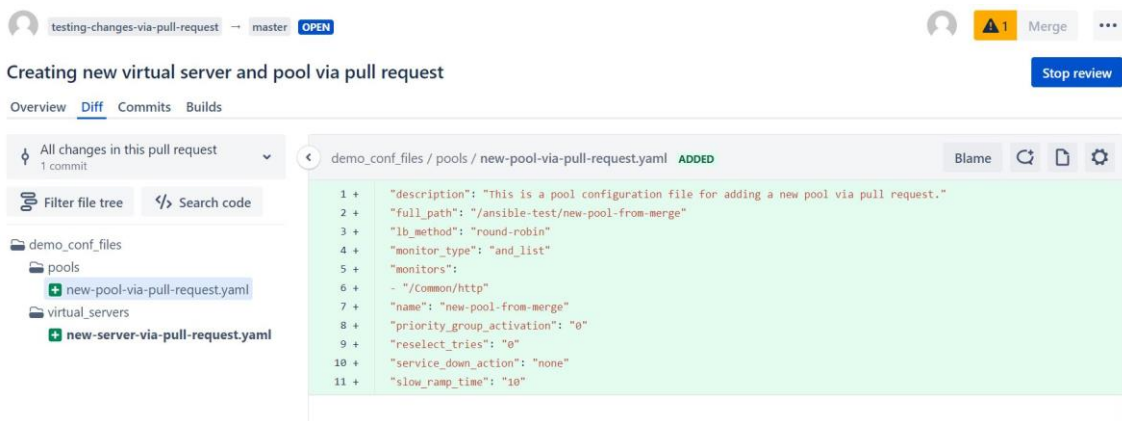
Joonis 6 Tõmbetaotluse koostamine

Kui tõmbetaotlus on esitatud, siis pole selle ühendamine *master* haruga võimalik enne, kui päring on ühe või mitme Big-IP platvormi administraatori poolt heakskiidu saanud (vt Joonis 7).



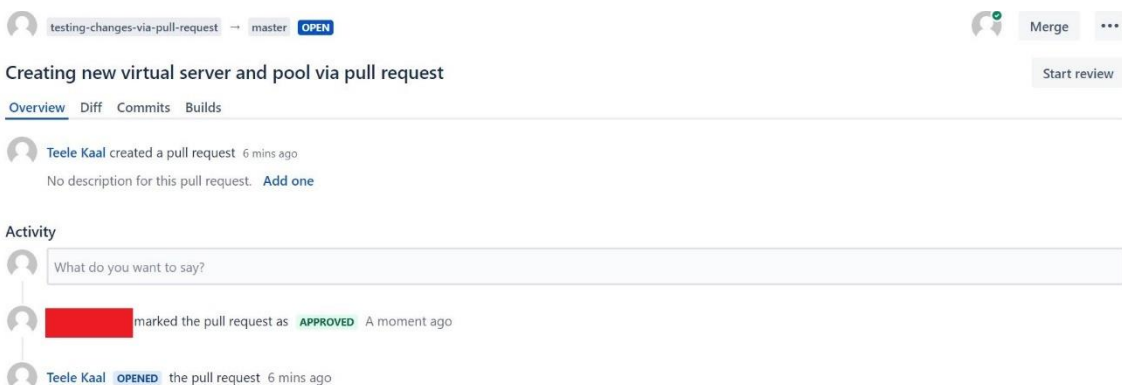
Joonis 7 Tõmbetaotlus vajab ülevaatamist enne selle master haruga ühendamist

Tehtud tõmbetaotluse vaates on ülevaatajal võimalik kontrollida, milliseid muudatusi soovitakse lähtekoodi lisada (vt Joonis 8).

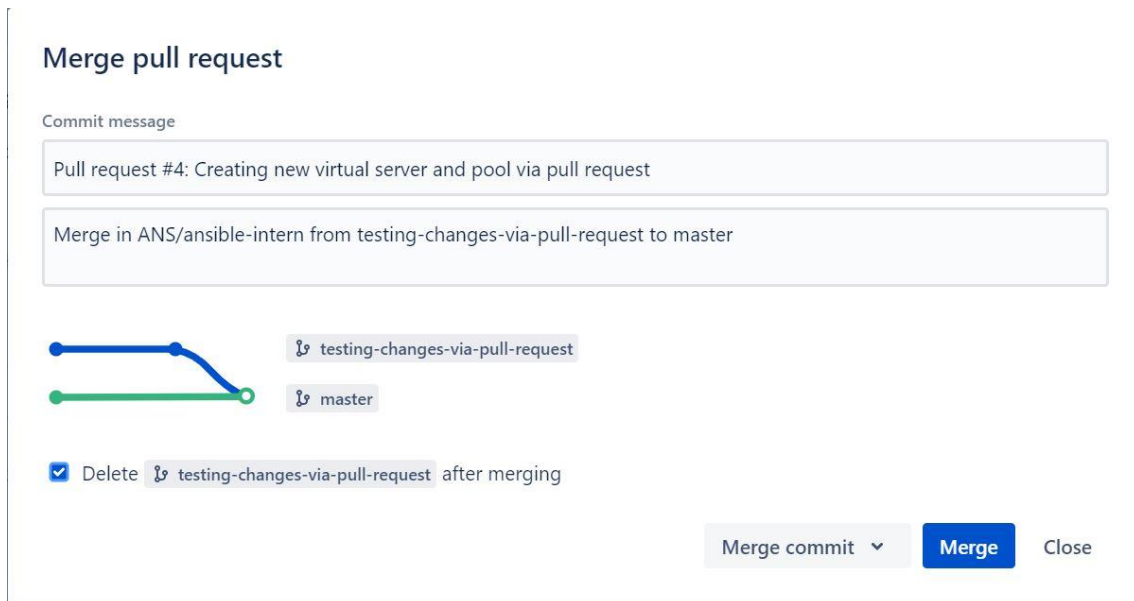


Joonis 8 Tõmbetaotluse käigus muudetavate andmete kontrollimine

Kui päring on ülevaataja poolt heaks kiidetud (vt Joonis 9), siis on võimalik *testing-changes-via-pull-request* haru ühendada haruga *master* (vt Joonis 10) ning soovitud muudatused lisatakse *master* harusse (vt Joonis 11).



Joonis 9 Ülevaataja poolt heaks kiidetud tõmbetaotlust on võimalik ühendada teise haruga



Joonis 10 testing-changes-via-pull-request haru ühendamise master haruga



Joonis 11 Muudatused on peale harude ühendamist lisatud harusse master




Kirjeldatud näide iseloomustab, kuidas on koodi keskses koodihoidlas hoiustamise ning erinevate harude kasutamise võimalik soodustada tiimide vahelist koostööd ning pakkuda uute ressursside loomist teenuse osana. Erinevate meeskondade jaoks saab luua eraldi repositooriumid, mis hõlmavad endas meeskonna vastutuses olevate teenuste Big-IP komponente.

Repositooriumile ligipääsejatel on võimalik teistesse harudesse omapoolseid muudatusi lisada ning seejärel avaldada tõmbetaoluste abil soovi lisada tehtud muudatused põhiharusse. Põhiharule on seadistatud kindlad ülevaatajad, kes vastutavad Big-IP platvormi eest ning kes saavad päringutele heakskiitust andes ja Ansible *playbook*'e haldusserveris käivitades muudatusi infrastruktuuris juurutada. Juurutamise automatiseerimiseks on võimalik kasutada ka CI/CD töövahendeid, mis heakskiidu korral koheselt muudatused ellu viivad. Lisaks saab kesksest koodihoidlast kontrollida

muudatuste ajalugu (vt Joonis 12), mis võimaldab tehtud muudatusi vajaduse korral kiirelt tagasi keerata ning tagab versioonihalduse.

Commits

🔗 master ▾ ⋮

Author	Commit	Message	Commit date	Builds
 Teele Kaal	@f1bbb462e1 M	Pull request #4: Creating new virtual server and pool via pull request Merge in ANS/ansible-intern...	2 mins ago	
 Teele Kaal	9f923a179e3	Creating new virtual server and pool via pull request	12 mins ago	
 Teele Kaal	0329b475aa8	renamed dir	35 mins ago	

Joonis 12 Bitbucketi repositooriumi poolt pakutav versioonihaldus

4 Järeldused

Neljandas peatükis tuuakse välja lõputöös kirjeldatava näidislahenduse tulemused ning ettepanekud lahenduse edasiseks arendamiseks.

4.1 Loodud näidislahenduse tulemused

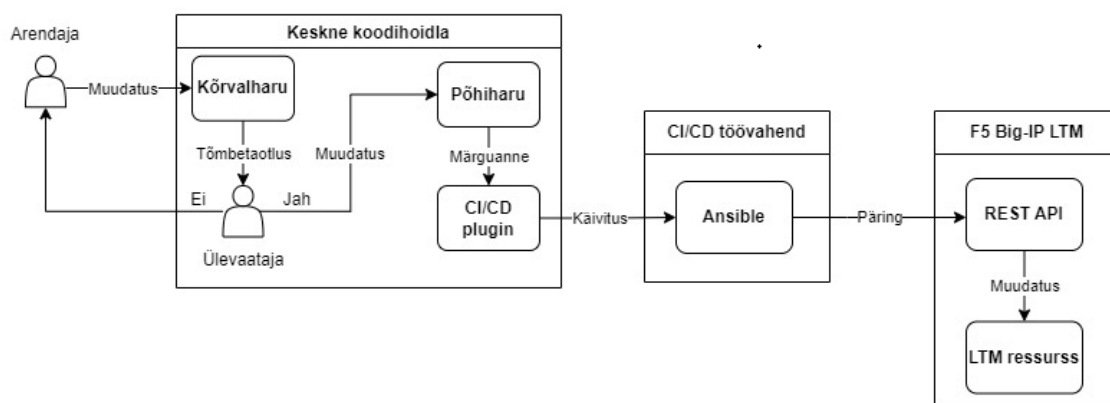
Lahenduse testimine oli edukas ehk loodud näidislahendus võimaldab Ansible abiga Big-IP platvormil LTM ressursse hallata ning neid automatiseeritult ühest partitsioonist teise liigutada. Lisaks demonstreerib lahendus Atlassian Bitbucketi näitel, kuidas on võimalik lähtekoodi keskses koodihoidlas hoiustada ja selle põhiharu Big-IP LTM komponentide konfiguratsiooni ainsa tõeallikana kasutada. Repositooriumisse uute harude loomine võimaldab pakkuda koormusjaoturite seadistamist teenuse osana ehk erinevatel meeskondadel on vastavalt vajadusele võimalik teha päringuid lähtekoodi muutmiseks, mis heakskiidu saamisel põhiharusse lisatakse. Kirjeldatud koostöö tõstab koormusjaoturite loomise protsessi tootlikkust – erinevatel tiimidel pole vajadust koostada piletit Big-IP platvormi haldavale meeskonnale ning oodata, millal vajalikud koormusjaoturid luuakse. Tõmbetaotlustega saab uute ressursside loomist sooviv meeskond ise vajalikud seadistusfailid luua, mis sobivuse korral lisatakse põhiharusse, milles paiknevaid andmeid kasutab Ansible Big-IP platvormil muudatuste juurutamiseks.

Eesti Energia poolt näidislahendusele seatud ootus versioonihalduse tagamiseks läbi koodi keskses koodihoidlas hoiustamise sai täidetud, sest kõik näidislahendustega seotud failid on hoiustatud Bitbucketi Ansible-Intern repositooriumis. Bitbucketi muudatuste ajaloo vaade võimaldab näha, milliseid muudatusi on kellegi poolt erinevatel ajahetkedel tehtud. Versioonihaldus muudab ka Big-IP platvormi teenuse pakkumise töökindlamaks – uute muudatustega tekkinud probleeme on võimalik kiirelt lahendada, kui taastada muudatuste ajaloost kõige hiljutisem töötav versioon.

4.2 Ettepanekud lahenduse edasi arendamiseks

Üheks oluliseks GitOps'i printsiibiks on muudatuste automatiseeritud juurutamine CI/CD töövahendite abil. CI/CD töövahendid aitavad tagada, et Git repositooriumis kirjeldatu vastab infrastruktuuri reaalsele olukorrale, sest põhiharu uuenedes käivitatakse automaatselt *pipeline*, mis teostab vajalikud muudatused. Näidislahendust saab edasi arendada läbi CI/CD töövahendi kasutuselevõtu, mis juurutaks automaatselt põhiharusse lisatud muudatused, kontrolliks muudatuste järgselt vajaliku funktsionaalsuse säilimist ning probleemide tekkimise korral taastaks infrastruktuuri eelneva seisuga. Kirjeldatud edasiarendus kaotab ära vajaduse läbi Git päringute koodihoidlast muudatuste serverisse lisamiseks ning selles *playbook* ide käsitsi käivitamiseks.

Joonis 13 illustreerib Big-IP LTM komponentide haldamist GitOps'i printsiipidele tuginedes, kui lõputöös kirjeldatud lahendusele lisada CI/CD töövahend muudatuste juurutamiseks. Joonisel kujutatud protsess kasutab keskse koodihoidla põhiharu ainsa tõeallikana. Big-IP platvormil muudatuste tegemiseks tehakse soovitud muudatus koodihoidla kõrvalharus ning esitatakse tõmbetaotlus muudatuse lisamiseks projekti põhiharusse. Tõmbetaotluse heakskiidu korral ühendatakse harud, mille tulemusena käivitub automaatselt CI/CD töövahend, mis kasutab Ansible't põhiharus defineeritud seadistuste realiseerimiseks Big-IP platvormil.



Joonis 13 GitOps'i printsiipide rakendamine komponentide haldamises

5 Kokkuvõte

Käesoleva lõputöö eesmärgiks oli luua näidislahendus GitOps'i printsiipide osalisest juurutamisest koormusjaoturi platvormi F5 Big-IP halduses ning automatiseeritult uuele komponentide hoiustamise ülesehitusele liikumise protsessist. Lahenduse planeerimisel teostati kõigepealt analüüs, mille raames kirjeldati GitOps'i mõttemaailma ning printsiipe, seati paika näidislahenduse funktsionaalsus ja skoop ning võrreldi HashiCorp Terraformi ja Ansible poolt pakutavaid võimalusi Big-IP LTM ressursside haldamiseks. Analüüsi käigus osutus lahenduse jaoks sobivaimaks töövahendiks Ansible.

Näidislahenduse valmimisel automatiseeriti Ansible vahenditega kaks protsessi – Big-IP LTM ressursside haldamine ning platvormil leiduvate komponentide liigutamine erinevatesse partitsioonidesse rollipõhiste ligipääsude jagamise võimaldamiseks. Töö käigus valmisid Ansible rollid ja *playbook*'id, mida katsetati testimiseks loodud ressurssidega. Lahenduse katsetamiseks liigutati Common partitsioonist kaks veebiserverit koos koormusjaoturiga partitsiooni ansible-test, protsessi mõju veebiteenusele monitooriti terve protseduuri vältel Pythoni skriptiga. Testimine oli edukas, sest veebiserverite liigutamisel probleeme ei tekkinud ning teenuse funktsionaalsus säilis, kolimine mõjutas teenuse kättesaadavust ligikaudu 60 sekundit. Näidislahenduse lähtekoodi hoiustamiseks keskses koodihoidlas loodi uus Atlassian Bitbucketi repositoorium ning demonstreeriti, kuidas on Bitbucketi tõmbetaotlustega võimalik pakkuda koormusjaoturite seadistamist teenuse osana.

Lõputöös kirjeldatud näidislahendust on võimalik kasutusele võtta Big-IP platvormiga seotud tööprotsesside kiirendamiseks – erinevatel meeskondadel kaob vajadus teha pilet platvormi eest vastutavale tiimile, kes esimesel võimalusel manuaalselt vajalikud muudatused ellu viib. Samuti tagab Big-IP platvormi ülesehituse koodi kujul Bitbucketis hoiustamine versioonihalduse ehk on võimalik kontrollida, milliseid muudatusi kellegi poolt erinevatel ajamomentidel tehtud on ning uute muudatuste tõttu probleemide ilmnemisel taastada infrastruktuuri eelmine seisukord. Kirjeldatud näidislahenduse üheks

edasi arendamise variandiks on CI/CD töövahendi kasutusele võtmine muudatuste juurutamiseks, mis võimaldaks läbi *pipeline*'i veenduda, et Bitbucketi repositooriumis kirjeldatu vastab tõepoolest Big-IP platvormi ressursside hetkeseisule.

Kasutatud materjalid

- [1] Atlassian kodulehekülg. *What Is DevOps?*, [Võrgumaterjal]. Loetud aadressil: <https://www.atlassian.com/devops>. Kasutatud: 12.04.2023.
- [2] Atlassian Bitbucketi kodulehekülg. *Is GitOps the next big thing in DevOps?*. [Võrgumaterjal]. Loetud aadressil: <https://www.atlassian.com/git/tutorials/gitops>. Kasutatud: 01.04.2023.
- [3] F5 Networks Inc., kodulehekülg. *BIG-IP Local Traffic Manager* [Võrgumaterjal]. Loetud aadressil: <https://www.f5.com/products/big-ip-services/local-traffic-manager>. Kasutatud: 25.03.2023.
- [4] GitLab, *A Beginner's Guide to GitOps*. [E-raamat]. Loetud aadressil: <https://about.gitlab.com/topics/gitops/>. Kasutatud: 15.04.2023
- [5] K. Morris, *Infrastructure as Code: Dynamic Systems for the Cloud Age. Second Edition*. O'Reilly, 2021.
- [6] F5 Networks Inc. kodulehekülg. *Explore BIG-IP application services* [Võrgumaterjal]. Loetud aadressil: <https://www.f5.com/products/big-ip-services>. Kasutatud: 25.03.2023.
- [7] F5 Networks Inc. kodulehekülg. *Big-IP DNS*. [Võrgumaterjal]. Loetud aadressil: <https://www.f5.com/products/big-ip-services/big-ip-dns>. Kasutatud: 25.03.2023.
- [8] F5 Big-IP kasutajatoe kodulehekülg MyF5. *BIG-IP Local Traffic Management: Basics. Introduction to Local Traffic Management* [Võrgumaterjal]. Loetud aadressil: <https://techdocs.f5.com/en-us/bigip-14-1-0/big-ip-local-traffic-management-basics-14-1-0/introduction-to-local-traffic-management.html>. Kasutatud: 25.03.2023.
- [9] F5 Big-IP kasutajatoe kodulehekülg MyF5. *BIG-IP Local Traffic Management: Basics. About Virtual Servers*. [Võrgumaterjal]. Loetud aadressil: <https://techdocs.f5.com/en-us/bigip-14-1-0/big-ip-local-traffic-management-basics-14-1-0/about-virtual-servers.html#GUID-AA2EDFF7-B5D5-4F6F-8535-A30CD0360CF4>. Kasutatud: 25.03.2023.
- [10] F5 Big-IP kasutajatoe kodulehekülg MyF5. *BIG-IP Local Traffic Management: Basics. About Pools*. [Võrgumaterjal]. Loetud aadressil: <https://techdocs.f5.com/en-us/bigip-14-1-0/big-ip-local-traffic-management-basics-14-1-0/about-pools.html#GUID-E3F736FF-8C72-4B9F-8413-343E3BD868AD>. Kasutatud: 25.03.2023
- [11] F5 Big-IP dokumentatsioon. *BIG-IP LTM. BIG-IP TMOS: Concepts. Working with partitions*. [Võrgumaterjal]. Loetud aadressil: https://techdocs.f5.com/kb/en-us/products/big-ip_ltm/manuals/product/tmos-concepts-11-5-0/8.html. Kasutatud: 20.03.2023.
- [12] Y. Brikman, *Terraform Up & Running: Writing Infrastructure as Code*. O'Reilly, 2019.
- [13] B. Meijer, L. Hochstein ja R. Moser, *Ansible Up & Running: Automating Configuration Management and Deployment the Easy Way*. O'Reilly, 2022.
- [14] HashiCorp Terraform dokumentatsioon. *F5Networks provider*. [Võrgumaterjal]. Loetud aadressil: <https://registry.terraform.io/providers/F5Networks/bigip/latest/docs>. Kasutatud: 18.03.2023.

- [15] Ansible dokumentatsioon. *F5Networks.F5_Modules*. [Võrgumaterjal]. Loetud aadressil:
https://docs.ansible.com/ansible/latest/collections/f5networks/f5_modules/index.html.
Kasutatud: 18.03.2023.
- [16] Ansible dokumentatsioon. *f5networks.f5_modules.bigip_virtual_server module – Manage LTM virtual servers on a BIG-IP*. [Võrgumaterjal]. Loetud aadressil:
https://docs.ansible.com/ansible/latest/collections/f5networks/f5_modules/bigip_virtual_server_module.html#ansible-collections-f5networks-f5-modules-bigip-virtual-server-module.
Kasutatud: 18.03.2023.
- [17] HashiCorp Terraform dokumentatsioon. *bigip_ltm_virtual_server*. [Võrgumaterjal].
Loetud aadressil:
https://registry.terraform.io/providers/F5Networks/bigip/latest/docs/resources/bigip_ltm_virtual_server.
Kasutatud: 19.03.2023.
- [18] Ansible dokumentatsioon. *f5networks.f5_modules.bigip_node module – Manages F5 BIG-IP LTM nodes*. [Võrgumaterjal]. Loetud aadressil:
https://docs.ansible.com/ansible/latest/collections/f5networks/f5_modules/bigip_node_module.html#ansible-collections-f5networks-f5-modules-bigip-node-module.
Kasutatud: 18.03.2023.
- [19] Ansible dokumentatsioon. *f5_modules.bigip_pool_member module – Manages F5 BIG-IP LTM pool members*. [Võrgumaterjal]. Loetud aadressil:
https://docs.ansible.com/ansible/latest/collections/f5networks/f5_modules/bigip_pool_member_module.html#ansible-collections-f5networks-f5-modules-bigip-pool-member-module.
Kasutatud: 18.03.2023.
- [20] HashiCorp Terraform dokumentatsioon. *bigip_ltm_pool_attachment*. [Võrgumaterjal]. Loetud aadressil:
https://registry.terraform.io/providers/F5Networks/bigip/latest/docs/resources/bigip_ltm_pool_attachment.
Kasutatud: 19.03.2023.
- [21] HashiCorp Terraform dokumentatsioon. *bigip_ltm_node*. [Võrgumaterjal]. Loetud aadressil:
https://registry.terraform.io/providers/F5Networks/bigip/latest/docs/resources/bigip_ltm_node.
Kasutatud: 19.03.2023.
- [22] Ansible dokumentatsioon. *f5networks.f5_modules.bigip_pool module – Manages F5 BIG-IP LTM pools*. [Võrgumaterjal]. Loetud aadressil:
https://docs.ansible.com/ansible/latest/collections/f5networks/f5_modules/bigip_pool_module.html#ansible-collections-f5networks-f5-modules-bigip-pool-module.
Kasutatud: 19.03.2023.
- [23] HashiCorp Terraform dokumentatsioon. *bigip_ltm_pool* [Võrgumaterjal]. Loetud aadressil:
https://registry.terraform.io/providers/F5Networks/bigip/latest/docs/resources/bigip_ltm_pool.
Kasutatud: 19.03.2023.
- [24] HashiCorp Terraform dokumentatsioon. *bigip_ltm_monitor*. [Võrgumaterjal]. Loetud aadressil:
https://registry.terraform.io/providers/F5Networks/bigip/latest/docs/resources/bigip_ltm_monitor.
Kasutatud: 19.03.2023.
- [25] Ansible dokumentatsioon. *f5networks.f5_modules.bigip_partition module – Manage BIG-IP partitions*. [Võrgumaterjal]. Loetud aadressil:
https://docs.ansible.com/ansible/latest/collections/f5networks/f5_modules/bigip_partition_module.html#ansible-collections-f5networks-f5-modules-bigip-partition-module.

- ion_module.html#ansible-collections-f5networks-f5-modules-bigip-partition-module. Kasutatud: 19.03.2023.
- [26] F5 Big-IP kasutajatoe kodulehekülg MyF5. *K75181207: Renaming BIG-IP configuration objects*. [Võrgumaterjal]. Loetud aadressil: <https://my.f5.com/manage/s/article/K75181207>. Kasutatud: 20.03.2023.
- [27] F5 Big-IP dokumentatsioon, CloudDocs Home. *F5 Big-IP Ansible Collections* [Võrgumaterjal]. Loetud aadressil: <https://clouddocs.f5.com/products/orchestration/ansible/devel/>. Kasutatud: 02.04.2023.
- [28] F5 Big-IP dokumentatsioon, CloudDocs Home. *Example Playbook and Setup in F5 Imperative Collection*. [Võrgumaterjal]. Loetud aadressil: https://clouddocs.f5.com/products/orchestration/ansible/devel/f5_modules/playbook_tutorial.html. Kasutatud: 07.04.2023.
- [29] Ansible dokumentatsioon. *ansible.builtin.from_yaml filter – Convert YAML string into variable structure* [Võrgumaterjal]. Loetud aadressil: https://docs.ansible.com/ansible/devel/collections/ansible/builtin/from_yaml_filter.html. Kasutatud: 02.04.2023.
- [30] F5 Big-IP REST API dokumentatsioon. *APIRef_tm_ltm_monitor* [Võrgumaterjal]. Loetud aadressil: https://clouddocs.f5.com/api/icontrol-rest/APIRef_tm_ltm_monitor.html. Kasutatud: 02.04.2023.
- [31] Ansible dokumentatsioon. *ansible.builtin.uri module – Interacts with webservices* [Võrgumaterjal]. Loetud aadressil: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/uri_module.html. Kasutatud: 02.04.2023.
- [32] F5 kasutajatoe kodulehekülg, Bug tracker. *Bug ID 679901: The iControl REST timeout value is not configurable*. [Võrgumaterjal]. Loetud aadressil: <https://cdn.f5.com/product/bugtracker/ID679901.html>. Kasutatud: 07.04.2023.
- [33] Ansible dokumentatsioon. *f5networks.f5_modules.bigip_device_info module – Collect information from F5 BIG-IP devices*. [Võrgumaterjal]. Loetud aadressil: https://docs.ansible.com/ansible/latest/collections/f5networks/f5_modules/bigip_device_info_module.html#ansible-collections-f5networks-f5-modules-bigip-device-info-module. Kasutatud: 05.04.2023.
- [34] Ansible dokumentatsioon. *f5networks.f5_modules.bigip_wait module – Wait for a BIG-IP condition before continuing* [Võrgumaterjal]. Loetud aadressil: https://docs.ansible.com/ansible/latest/collections/f5networks/f5_modules/bigip_wait_module.html#ansible-collections-f5networks-f5-modules-bigip-wait-module. Kasutatud: 08.04.2023.
- [35] F5 Networks Inc. kodulehekülg, *iControl REST API User Guide*. [Võrgumaterjal]. Loetud aadressil: <https://cdn.f5.com/websites/devcentral.f5.com/downloads/icontrol-rest-api-user-guide-14-1-0.pdf>. Kasutatud: 09.04.2023.
- [36] Ansible dokumentatsioon. *Using filters to manipulate data*. [Võrgumaterjal]. Loetud aadressil: https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_filters.html. Kasutatud: 09.04.2023.
- [37] Ansible dokumentatsioon. *ansible.builtin.file module – Manage files and file properties*. [Võrgumaterjal]. Loetud aadressil:

- https://docs.ansible.com/ansible/latest/collections/ansible/builtin/file_module.html.
Kasutatud: 09.04.2023.
- [38] Ansible dokumentatsioon. *ansible.builtin.combine filter – combine two dictionaries*. [Võrgumaterjal]. Loetud aadressil:
https://docs.ansible.com/ansible/latest/collections/ansible/builtin/combine_filter.html.
Kasutatud: 09.04.2023.
- [39] Ansible dokumentatsioon. *ansible.builtin.copy module – Copy files to remote locations*. [Võrgumaterjal]. Loetud aadressil:
https://docs.ansible.com/ansible/latest/collections/ansible/builtin/copy_module.html.
Kasutatud: 09.04.2023.
- [40] Ansible dokumentatsioon. *ansible.builtin.assert module – Asserts given expressions are true*. [Võrgumaterjal]. Loetud aadressil:
https://docs.ansible.com/ansible/latest/collections/ansible/builtin/assert_module.html.
Kasutatud: 09.04.2023.
- [41] PyPi kodulehekülg. *Python HTTP for Humans* [Võrgumaterjal]. Loetud aadressil:
<https://pypi.org/project/requests/>. Kasutatud: 10.04.2023.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Teele Kaal

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Koormusjaoturi teenuse automatiseerimine GitOps printsiipe kasutades Eesti Energia AS näitel“, mille juhendaja on Siim Vene ja kaasjuhendaja on Mariliis Häälme
 - 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.

15.05.2023

¹ 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 - Aktiivse Big-IP õla tuvastamine Ansible'ga

```
- name: Collecting BIG-IP information...
bigip_device_info:
  gather_subset:
    - devices
  provider:
    user: "{{ bigip_user }}"
    password: "{{ bigip_password }}"
    server: "{{ bigip_server }}"
  delegate_to: localhost
  register: device_facts
  until: device_facts is not failed
  retries: 3

- name: Determine active BIG-IP node
ansible.builtin.set_fact:
  bigip_active_node: "{{ item.name }}"
when: (item.failover_state == "active") and (item.management_address is match("192.168.115.*"))
with_items:
  - "{{ device_facts['ansible_facts']['ansible_net_devices'] }}"

- name: Print name of the active BIG-IP node
debug:
  msg: Active node is "{{ bigip_active_node }}"
```

Lisa 3 – Ressursside haldamiseks loodud main.yaml koormusjaoturite näitel

```
---
- name: Wait for BIG-IP to be ready to take configuration
  bigip_wait:
  provider:
    password: "{{ bigip_password }}"
    server: "{{ bigip_active_node }}"
    user: "{{ bigip_user }}"
  delegate_to: localhost
  register: ready
  until: ready is not failed
  retries: 3

- name: Manage virtual_servers
  include_tasks: manage_virtual_servers.yaml
  with_fileglob:
    - "/path/to/files/virtual_servers/*"
```

Lisa 4 – Ressursside YAML formaadis seadistusfailid

koormusjaoturite näitel

```
"auto_last_hop": "default"
"description": "Example virtual server configuration fire."
"destination": "192.168.145.50"
"enabled_vlans":
- "/Common/Test"
"full_path": "/Common/example"
"ip_protocol": "tcp"
"irules": []
"mirror": "no"
"name": "example"
"pool": "/Common/example-pool"
"port": "80"
"port_translation": "yes"
"profiles":
- "context": "all"
  "full_path": "/Common/http-lan-optimized-nocaching"
  "name": "http-lan-optimized-nocaching"
- "context": "all"
  "full_path": "/Common/tcp"
  "name": "tcp"
"rate_limit": "0"
"rate_limit_dst_mask": "0"
"rate_limit_mode": "object"
"snat": "automap"
"source": "0.0.0.0/0"
"source_port": "preserve"
"type": "standard"
```

Lisa 5 – Ülesanded ressurside haldamiseks koormusjaoturite näitel

```
---
- name: Manage virtual server "{{ (lookup('template', '{{ item }}')|from_yaml).name }}"
  bigip_virtual_server:
    state: "{{ (lookup('template', '{{ item }}')|from_yaml).state | default('present') }}"
    partition: "{{ (lookup('template', '{{ item }}')|from_yaml).full_path.split('/')[0:-1] | join('/') }}"
    name: "{{ (lookup('template', '{{ item }}')|from_yaml).name }}"
    port: "{{ (lookup('template', '{{ item }}')|from_yaml).port }}"
    pool: "{{ (lookup('template', '{{ item }}')|from_yaml).pool }}"
    destination: "{{ (lookup('template', '{{ item }}')|from_yaml).destination }}"
    description: "{{ (lookup('template', '{{ item }}')|from_yaml).description | default('') }}"
    ip_protocol: "{{ (lookup('template', '{{ item }}')|from_yaml).ip_protocol }}"
    type: "{{ (lookup('template', '{{ item }}')|from_yaml).type }}"
    profiles: "{{ (lookup('template', '{{ item }}')|from_yaml).profiles }}"
    enabled_vlans: "{{ (lookup('template', '{{ item }}')|from_yaml).enabled_vlans }}"
    port_translation: "{{ (lookup('template', '{{ item }}')|from_yaml).port_translation | default('yes') }}"
    snat: "{{ (lookup('template', '{{ item }}')|from_yaml).snat | default('automap') }}"
    rate_limit_mode: "{{ (lookup('template', '{{ item }}')|from_yaml).rate_limit_mode | default('object') }}"
    rate_limit: "{{ (lookup('template', '{{ item }}')|from_yaml).rate_limit }}"
    rate_limit_dst_mask: "{{ (lookup('template', '{{ item }}')|from_yaml).rate_limit_dst_mask }}"
    source: "{{ (lookup('template', '{{ item }}')|from_yaml).source | default('0.0.0.0/0') }}"
    source_port: "{{ (lookup('template', '{{ item }}')|from_yaml).source_port | default('preserve') }}"
    auto_last_hop: "{{ (lookup('template', '{{ item }}')|from_yaml).auto_last_hop }}"
    irules: "{{ (lookup('template', '{{ item }}')|from_yaml).irules }}"
    provider:
      server: "{{ bigip_active_node }}"
      user: "{{ bigip_user }}"
      password: "{{ bigip_password }}"
  register: create_server
  until: create_server is not failed
  retries: 3
  delegate_to: localhost
```

Lisa 6 – Monitoride haldamiseks loodud ülesanded

```
---
- name: Set resource ID and monitor type
  ansible.builtin.set_fact:
    resource_id: "{{ (lookup('template', '{{ item }}')|from_yaml).fullPath.split('/') | join('~') }}"
    monitor_type: "{{ item.split('/')[-2] }}"

- name: Check if monitor already exists to decide if HTTP POST or PATCH request should be used
  ansible.builtin.uri:
    url: "https://{{ bigip_active_node }}/mgmt/tm/ltn/monitor/{{ monitor_type }}/{{ resource_id }}"
    user: "{{ bigip_user }}"
    password: "{{ bigip_password }}"
    method: GET
    status_code: [200, 404]
  register: GET_result
  no_log: True
  until: GET_result is not failed
  retries: 3

- name: Create a new monitor with HTTP POST request
  ansible.builtin.uri:
    url: "https://{{ bigip_active_node }}/mgmt/tm/ltn/monitor/{{ monitor_type }}"
    user: "{{ bigip_user }}"
    password: "{{ bigip_password }}"
    method: POST
    body: "{{ (lookup('template', '{{ item }}') | from_yaml | to_json) }}"
    body_format: json
  when: GET_result.status == 404
  no_log: True
  register: POST_result
  until: POST_result is not failed
  retries: 3

- name: Modify existing monitor with HTTP PATCH request
  ansible.builtin.uri:
    url: "https://{{ bigip_active_node }}/mgmt/tm/ltn/monitor/{{ monitor_type }}/{{ resource_id }}"
    user: "{{ bigip_user }}"
    password: "{{ bigip_password }}"
    method: PATCH
    body: "{{ (lookup('template', '{{ item }}') | from_yaml | to_json) }}"
    body_format: json
  when: GET_result.status == 200
  no_log: True
  register: PATCH_result
  until: PATCH_result is not failed
  retries: 3
```

Lisa 7 – Ressursside haldamise rolle täitev `playbook.yaml`

```
---
- name: Determine active BIG-IP node
  hosts:
    - localhost
  connection: local
  roles:
    - determine_active_bigip_node
  tags: active_node

- name: Manage partitions
  hosts:
    - localhost
  connection: local
  gather_facts: no
  roles:
    - partitions
  tags: partitions

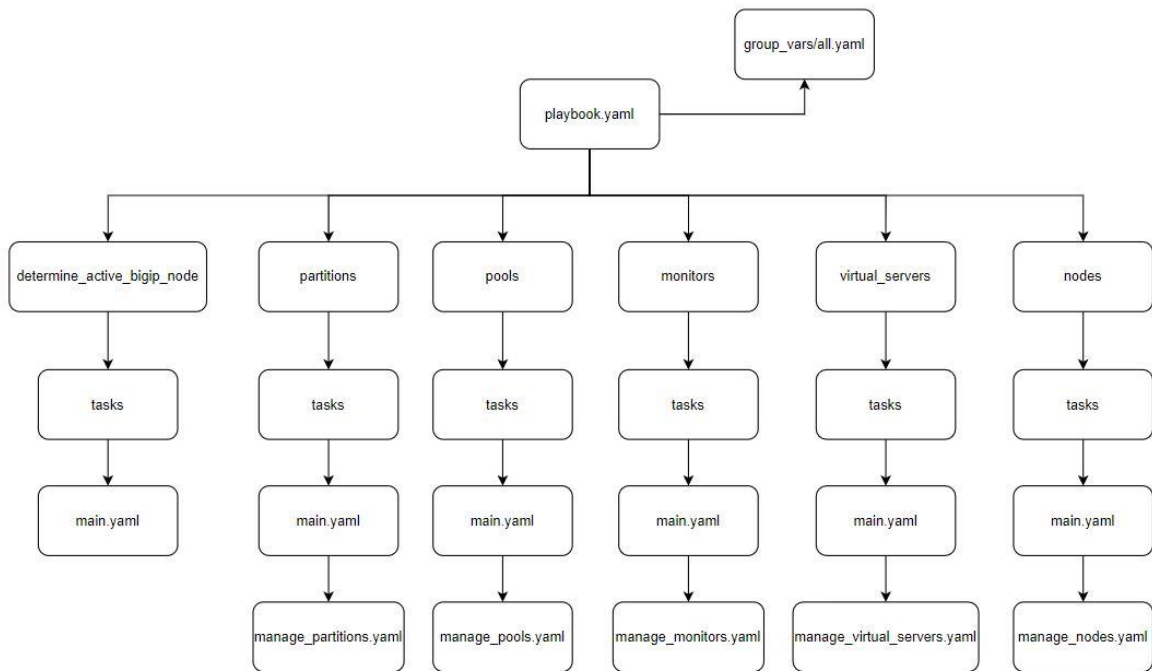
- name: Manage pools
  hosts:
    - localhost
  connection: local
  gather_facts: no
  roles:
    - pools
  tags: pools

- name: Manage monitors
  hosts:
    - localhost
  connection: local
  gather_facts: no
  roles:
    - monitors
  tags: monitors

- name: Manage virtual servers
  hosts:
    - localhost
  connection: local
  gather_facts: no
  roles:
    - virtual_servers
  tags: virtual_servers

- name: Manage nodes
  hosts:
    - localhost
  connection: local
  gather_facts: no
  roles:
    - nodes
  tags: nodes
```

Lisa 8 – *Playbook*'i *playbook.yaml* structuur



Lisa 9 – Olemasolevate komponentide andmete kogumine koormusjaoturite näitel

```
---
- name: Gather data from "{{ bigip_active_node }}" about virtual servers
  bigip_device_info:
    gather_subset:
      - virtual-servers
    provider:
      user: "{{ bigip_user }}"
      password: "{{ bigip_password }}"
      server: "{{ bigip_active_node }}"
    delegate_to: localhost
    register: device_facts
    until: device_facts is not failed
    retries: 3

- name: Make sure that the directory for virtual servers configuration files exists
  file:
    path: "/path/to/configuration-files-directory/virtual_servers"
    state: directory

- name: Create YAML configuration files for every virtual server
  include_tasks: write_servers_to_file.yaml
  with_items:
    - "{{ device_facts['ansible_facts']['ansible_net_virtual_servers'] }}"
```


Lisa 10 – Olemasolevate komponentide seadistuste kirjutamine failidesse koormusjaoturite näitel

```
---
- name: Filter out necessary data for YAML configuration file
  ansible.builtin.set_fact:
    server_dictionary: "{{ server_dictionary | default({}) | combine ({{ server_conf.key : server_conf.value }}) }}"
  with_items:
    - {key: "name", value: "{{ item.name }}" }
    - {key: "port", value: "{{ item.destination_port }}" }
    - {key: "destination", value: "{{ item.destination_address }}" }
    - {key: "pool", value: "{{ item.default_pool | default('') }}" }
    - {key: "description", value: "{{ item.description | default('') }}" }
    - {key: "ip_protocol", value: "{{ item.protocol }}" }
    - {key: "type", value: "{{ item.type }}" }
    - {key: "profiles", value: "{{ item.profiles }}" }
    - {key: "enabled_vlans", value: "{{ item.vlans | default('[]') }}" }
    - {key: "port_translation", value: "{{ item.translate_port }}" }
    - {key: "snat", value: "{{ item.snat_type }}" }
    - {key: "source_port", value: "{{ item.source_port_behavior }}" }
    - {key: "mirror", value: "{{ item.connection_mirror_enabled }}" }
    - {key: "irules", value: "{{ item.irules | default('[]') }}" }
    - {key: "rate_limit", value: "% if item.rate_limit == -1 %0% else %}item.rate_limit{% endif %}" }
    - {key: "rate_limit_mode", value: "{{ item.rate_limit_mode }}" }
    - {key: "rate_limit_dst_mask", value: "{{ item.rate_limit_destination_mask }}" }
    - {key: "source", value: "{{ item.source_address }}" }
    - {key: "auto_last_hop", value: "{{ item.auto_lasthop }}" }
    - {key: "full_path", value: "{{ item.full_path }}" }
  loop_control:
    loop_var: server_conf
- name: Write virtual servers data to virtual servers config file
  copy:
    content: "{{ server_dictionary | to_nice_yaml(indent=2, width=1337, default_style='\"') }}"
    dest: "/path/to/configuration-files-directory/virtual_servers/{{ server_dictionary['name'] }}.yaml"
```

Lisa 11 – Ressursside Common partitsioonist teise partitsiooni liigutamiseks loodud ülesannete fail koormusjaoturite näitel

```
---
- name: Wait for BIG-IP to be ready to take configuration
  bigip_wait:
    provider:
      password: "{{ bigip_password }}"
      server: "{{ bigip_active_node }}"
      user: "{{ bigip_user }}"
    delegate_to: localhost
  register: ready
  until: ready is not failed
  retries: 3

- name: Migrate virtual servers
  include_tasks: /etc/ansible-intern/roles/migra_virtual_server/tasks/migrate_virtual_servers.yaml
  with_fileglob:
    - "/path/to/configuration-files/virtual_servers/*"
```

Lisa 12 – Ressursside Common partitsioonist teise liigutamine

koormusjaoturite näitel

```
---
- name: Delete virtual servers from partition "Common"
  bigip_virtual_server:
    state: "absent"
    partition: "Common"
    name: "{{ (lookup('template', '{{ item }}')|from_yaml).name }}"
    port: "{{ (lookup('template', '{{ item }}')|from_yaml).port }}"
    pool: "{{ (lookup('template', '{{ item }}')|from_yaml).pool }}"
    destination: "{{ (lookup('template', '{{ item }}')|from_yaml).destination }}"
    description: "{{ (lookup('template', '{{ item }}')|from_yaml).description | default('') }}"
    ip_protocol: "{{ (lookup('template', '{{ item }}')|from_yaml).ip_protocol }}"
    type: "{{ (lookup('template', '{{ item }}')|from_yaml).type }}"
    profiles: "{{ (lookup('template', '{{ item }}')|from_yaml).profiles }}"
    enabled_vlans: "{{ (lookup('template', '{{ item }}')|from_yaml).enabled_vlans }}"
    port_translation: "{{ (lookup('template', '{{ item }}')|from_yaml).port_translation | default('yes') }}"
    snat: "{{ (lookup('template', '{{ item }}')|from_yaml).snat | default('automap') }}"
    rate_limit_mode: "{{ (lookup('template', '{{ item }}')|from_yaml).rate_limit_mode | default('object') }}"
    rate_limit: "{{ (lookup('template', '{{ item }}')|from_yaml).rate_limit }}"
    rate_limit_dst_mask: "{{ (lookup('template', '{{ item }}')|from_yaml).rate_limit_dst_mask }}"
    source: "{{ (lookup('template', '{{ item }}')|from_yaml).source | default('0.0.0.0/0') }}"
    source_port: "{{ (lookup('template', '{{ item }}')|from_yaml).source_port | default('preserve') }}"
    auto_last_hop: "{{ (lookup('template', '{{ item }}')|from_yaml).auto_last_hop }}"
    irules: "{{ (lookup('template', '{{ item }}')|from_yaml).irules }}"
    provider:
      server: "{{ bigip_active_node }}"
      user: "{{ bigip_user }}"
      password: "{{ bigip_password }}"
  register: delete_server
  until: delete_server is not failed
  retries: 3
  delegate_to: localhost

- name: Create virtual server "{{ (lookup('template', '{{ item }}')|from_yaml).name }}" in partition "ansible-test"
  bigip_virtual_server:
    state: "present"
    partition: "ansible-test"
    name: "{{ (lookup('template', '{{ item }}')|from_yaml).name }}"
    port: "{{ (lookup('template', '{{ item }}')|from_yaml).port }}"
    pool: "/ansible-test/{{ (lookup('template', '{{ item }}')|from_yaml).pool.split('/')[2:] | join('/') }}"
    destination: "{{ (lookup('template', '{{ item }}')|from_yaml).destination }}"
    description: "{{ (lookup('template', '{{ item }}')|from_yaml).description | default('') }}"
    ip_protocol: "{{ (lookup('template', '{{ item }}')|from_yaml).ip_protocol }}"
    type: "{{ (lookup('template', '{{ item }}')|from_yaml).type }}"
    profiles: "{{ (lookup('template', '{{ item }}')|from_yaml).profiles }}"
    enabled_vlans: "{{ (lookup('template', '{{ item }}')|from_yaml).enabled_vlans }}"
    port_translation: "{{ (lookup('template', '{{ item }}')|from_yaml).port_translation | default('yes') }}"
    snat: "{{ (lookup('template', '{{ item }}')|from_yaml).snat | default('automap') }}"
    rate_limit_mode: "{{ (lookup('template', '{{ item }}')|from_yaml).rate_limit_mode | default('object') }}"
    rate_limit: "{{ (lookup('template', '{{ item }}')|from_yaml).rate_limit }}"
    rate_limit_dst_mask: "{{ (lookup('template', '{{ item }}')|from_yaml).rate_limit_dst_mask }}"
    source: "{{ (lookup('template', '{{ item }}')|from_yaml).source | default('0.0.0.0/0') }}"
    source_port: "{{ (lookup('template', '{{ item }}')|from_yaml).source_port | default('preserve') }}"
    auto_last_hop: "{{ (lookup('template', '{{ item }}')|from_yaml).auto_last_hop }}"
    irules: "{{ (lookup('template', '{{ item }}')|from_yaml).irules }}"
    provider:
      server: "{{ bigip_active_node }}"
      user: "{{ bigip_user }}"
      password: "{{ bigip_password }}"
  register: create_server
  until: create_server is not failed
  retries: 3
  delegate_to: localhost
```

```

- name: Set resource ID
  ansible.builtin.set_fact:
    resource_id: "~ansible-test~{{ (lookup('template', '{{ item }}')|from_yaml).full_path.split('/')[2:] | join('~') }}"

- name: Check if virtual server "{{ (lookup('template', '{{ item }}')|from_yaml).name }}" was created successfully
  ansible.builtin.uri:
    url: "https://{{ bigip_active_node }}/mgmt/tm/ltm/virtual/{{ resource_id }}"
    user: "{{ bigip_user }}"
    password: "{{ bigip_password }}"
    method: GET
    status_code: [200, 404]
  register: GET_result
  until: GET_result is not failed
  retries: 3
  no_log: True

- name: Assert that new virtual server was created
  ansible.builtin.assert:
    that:
      - GET_result.status == 200
    fail_msg: "Virtual server {{ (lookup('template', '{{ item }}')|from_yaml).name }} was not created!"
    success_msg: "Virtual server {{ (lookup('template', '{{ item }}')|from_yaml).name }} was created successfully!"

```

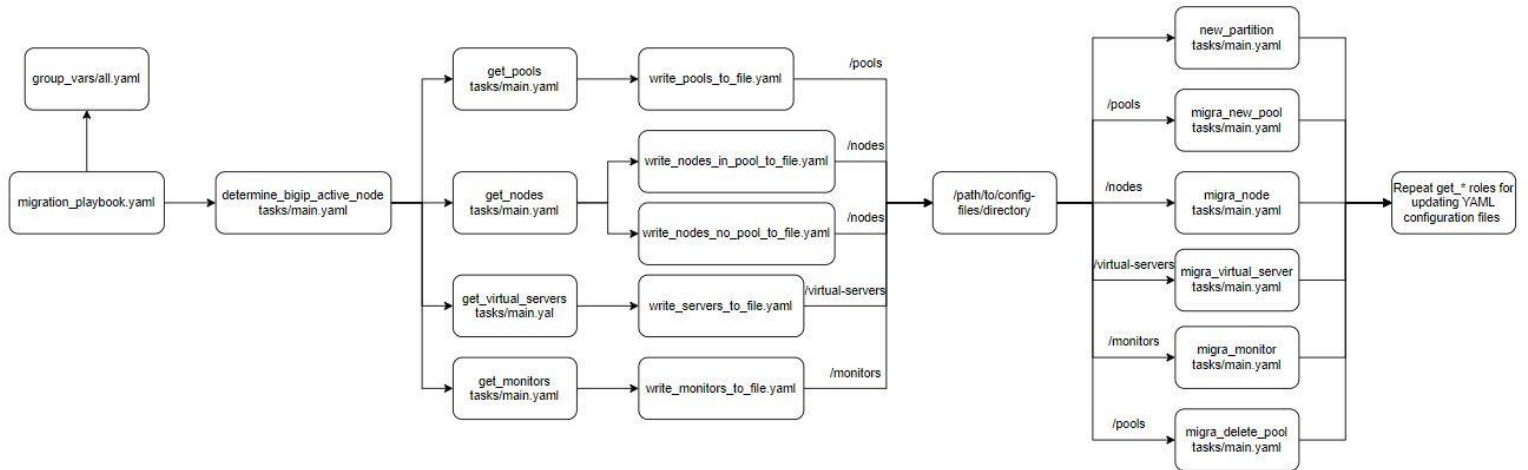
Lisa 13 – Ressursside liigutamiseks loodud

migration_playbook.yaml

```
---
- name: Determine active node, gather information and migrate all LTM resources
  hosts: localhost
  connection: local
  roles:
    - determine_active_bigip_node
    - get_pools
    - get_nodes
    - get_virtual_servers
    - get_monitors
    - new_partition
    - migra_new_pool
    - migra_node
    - migra_virtual_server
    - migra_monitor
    - migra_delete_pool
  tags: migra_all

- name: Update configuration files with latest data
  hosts: localhost
  connection: local
  roles:
    - get_pools
    - get_nodes
    - get_virtual_servers
    - get_monitors
```

Lisa 14 - Ressursside liigutamiseks loodud *playbook*'i struktuur



Lisa 15 – Common partitsiooni katsetamiseks loodud ühend test-web

Pools : Pool List >> test-web

Members Statistics

Method: Round Robin

Activation: Disabled

Add...

Member	Address	Service Port	FQDN	Ephemeral	Ratio	Priority Group	Connection Limit	Partition / Path
192.168.145.114:80	192.168.145.114	80		No	1	0 (Active)	0	Common
192.168.145.115:80	192.168.145.115	80		No	1	0 (Active)	0	Common

Lisa 16 – Katsetamiseks loodud test-web ressursside kättesaadavuse hindamiseks kasutatav http monitor

Local Traffic » Monitors » http			
⚙️	Properties	Instances	Test
General Properties			
Name	http		
Partition / Path	Common		
Type	HTTP		
Configuration: <input type="text" value="Advanced"/>			
Interval	5 seconds		
Up Interval	Disabled		
Time Until Up	0 seconds		
Timeout	16 seconds		
Manual Resume	No		
Send String	GET /\r\n		

Lisa 17 – Common partitsiooni katsetamiseks loodud koormusjaotur test-web

Local Traffic » Virtual Servers : Virtual Server List » test-web

⚙️ Properties Resources Statistics ↗

General Properties

Name	test-web
Partition / Path	Common
Description	<input type="text"/>
Type	Standard ▾
Source Address	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text" value="0.0.0.0/0"/>
Destination Address/Mask	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text" value="192.168.145.46"/>
Service Port	<input checked="" type="radio"/> Port <input type="radio"/> Port List <input type="text" value="80"/> HTTP ▾
Notify Status to Virtual Address	<input checked="" type="checkbox"/>

Local Traffic » Virtual Servers : Virtual Server List » test-web

⚙️ Properties Resources Statistics ↗

Load Balancing

Default Pool	test-web ▾
Default Persistence Profile	None ▾
Fallback Persistence Profile	None ▾

Lisa 18 – Testimiseks täiendatud roll komponentide seadistusfailide genereerimiseks koormusjaoturite näitel

```
---
- name: Filter out necessary data for YAML configuration file
  ansible.builtin.set_fact:
    server_dictionary: "{{ server_dictionary | default({}) | combine ({ server_conf.key : server_conf.value }) }}"
  with_items:
    - {key: "name", value: "{{ item.name }}" }
    - {key: "port", value: "{{ item.destination_port }}" }
    - {key: "destination", value: "{{ item.destination_address }}" }
    - {key: "pool", value: "{{ item.default_pool | default('') }}" }
    - {key: "description", value: "{{ item.description | default('') }}" }
    - {key: "ip_protocol", value: "{{ item.protocol }}" }
    - {key: "type", value: "{{ item.type }}" }
    - {key: "profiles", value: "{{ item.profiles }}" }
    - {key: "enabled_vlans", value: "{{ item.vlans | default('[]') }}" }
    - {key: "port_translation", value: "{{ item.translate_port }}" }
    - {key: "snat", value: "{{ item.snat_type }}" }
    - {key: "source_port", value: "{{ item.source_port_behavior }}" }
    - {key: "mirror", value: "{{ item.connection_mirror_enabled }}" }
    - {key: "irules", value: "{{ item.irules | default('[]') }}" }
    - {key: "rate_limit", value: "% if item.rate_limit == -1 %}0{% else %}item.rate_limit{% endif %}" }
    - {key: "rate_limit_mode", value: "{{ item.rate_limit_mode }}" }
    - {key: "rate_limit_dst_mask", value: "{{ item.rate_limit_destination_mask }}" }
    - {key: "source", value: "{{ item.source_address }}" }
    - {key: "source_port", value: "{{ item.source_port_behavior }}" }
    - {key: "auto_last_hop", value: "{{ item.auto_lasthop }}" }
    - {key: "full_path", value: "{{ item.full_path }}" }
  loop_control:
    loop_var: server_conf
  when: item.name == "test-web"

- name: Write virtual servers data to virtual servers config file
  copy:
    content: "{{ server_dictionary | to_nice_yaml(indent=2, width=1337, default_style='') }}"
    dest: "/etc/ansible-intern/demo_config_files/virtual_servers/{{ server_dictionary['name'] }}.yaml"
  when: item.name == "test-web"
```

Lisa 19 – Katsetamiseks loodud *playbook*

migration_playbook_for_demo.yaml

```
---
- name: Determine active node, gather information and migrate all LTM resources
  hosts: localhost
  connection: local
  roles:
    - demo_roles/demo_determine_active_bigip_node
    - demo_roles/demo_get_pools
    - demo_roles/demo_get_nodes
    - demo_roles/demo_get_virtual_servers
    - demo_roles/demo_new_partition
    - demo_roles/demo_new_pool
    - demo_roles/demo_migra_node
    - demo_roles/demo_migra_virtual_server
    - demo_roles/demo_delete_pool

- name: Get latest configurations from Big-IP and update configuration files
  hosts: localhost
  connection: local
  roles:
    - demo_roles/demo_get_pools
    - demo_roles/demo_get_nodes
    - demo_roles/demo_get_virtual_servers
```

Lisa 20 – Pythoni skript curl_test.py

```
import datetime
import time

import requests
from requests import ConnectionError

def check_web_page():

    while True:
        try:
            requests.get("INSERT_URL")
            with open('get_results.txt', 'a') as result_file:
                result_file.writelines(f"Get request time: {datetime.datetime.now()} Web service is available.")
            print(f"Get request time: {datetime.datetime.now()} Web service is available.")
        except ConnectionError:
            with open('get_results.txt', 'a') as result_file:
                result_file.writelines(f"Get request time: {datetime.datetime.now()} Web service is not available!")
            print(f"Get request time: {datetime.datetime.now()} Web service is not available!")
            time.sleep(15)

if __name__=="__main__":
    check_web_page()
```

Lisa 21 – *Playbook*'i täitmise vältel terminali kuvatud informatsioon

Playbook'i jooksmise käigus terminali kuvatud informatsiooni hulk oli suur, sest komponentide kohta Big-IP platvormilt informatsiooni pärimisel töödeldi läbi andmed kõikide platvormil leiduvate ressursside kohta. Seega on lisasse välja toodud testressurssidega seotud informatsioon koos *playbook*'i käivitamise alguse ja lõpuga. Lisaks on kuvatõmmistel peidetud tundliku loomuga informatsioon.

```
[2022-09-07T10:25:48.048Z] PLAY [Determine active node, gather information and migrate all LTM resources] ***
[2022-09-07T10:25:48.079Z]
[2022-09-07T10:25:48.079Z] TASK [Gathering Facts] *****
[2022-09-07T10:25:49.893Z] ok: [localhost]
[2022-09-07T10:25:49.958Z]
[2022-09-07T10:25:49.958Z] TASK [demo_roles/demo_determine_active_bigip_node : Collecting BIG-IP information...] ***
[2022-09-07T10:25:57.275Z] ok: [localhost -> localhost]
[2022-09-07T10:25:57.305Z]

[2022-09-07T10:25:57.454Z] TASK [demo_roles/demo_determine_active_bigip_node : Print name of the active BIG-IP node] ***
[2022-09-07T10:25:57.485Z] ok: [localhost] => {
[2022-09-07T10:25:57.485Z]   "msg": "Active node is \"[REDACTED]\""
[2022-09-07T10:25:57.485Z] }
[2022-09-07T10:25:57.559Z]

[2022-09-07T10:26:45.838Z] TASK [demo_roles/demo_get_pools : Gather necessary configuration data from [REDACTED] about pools and write that data to a file]
***
[2022-09-07T10:26:45.954Z] ok: [localhost] => (item={'key': 'name', 'value': 'test-web'})
[2022-09-07T10:26:46.013Z] ok: [localhost] => (item={'key': 'full_path', 'value': '/Common/test-web'})
[2022-09-07T10:26:46.013Z] ok: [localhost] => (item={'key': 'description', 'value': ''})
[2022-09-07T10:26:46.013Z] ok: [localhost] => (item={'key': 'lb_method', 'value': 'round-robin'})
[2022-09-07T10:26:46.053Z] ok: [localhost] => (item={'key': 'monitor_type', 'value': 'and_list'})
[2022-09-07T10:26:46.053Z] ok: [localhost] => (item={'key': 'monitors', 'value': ['/Common/http']})
[2022-09-07T10:26:46.117Z] ok: [localhost] => (item={'key': 'priority_group_activation', 'value': '0'})
[2022-09-07T10:26:46.117Z] ok: [localhost] => (item={'key': 'resselect_tries', 'value': '0'})
[2022-09-07T10:26:46.126Z] ok: [localhost] => (item={'key': 'service_down_action', 'value': 'none'})
[2022-09-07T10:26:46.126Z] ok: [localhost] => (item={'key': 'slow_ramp_time', 'value': '10'})
[2022-09-07T10:26:46.230Z]
[2022-09-07T10:26:46.230Z] TASK [demo_roles/demo_get_pools : Write pools data to pools config file] *****
[2022-09-07T10:26:47.121Z] changed: [localhost]
[2022-09-07T10:26:47.226Z]

[2022-09-07T10:27:29.529Z] TASK [demo_roles/demo_get_nodes : Gather necessary node information into dictionary] ***
[2022-09-07T10:27:29.631Z] ok: [localhost] => (item={'key': 'address', 'value': '192.168.145.114'})
[2022-09-07T10:27:29.693Z] ok: [localhost] => (item={'key': 'connection_limit', 'value': '0'})
[2022-09-07T10:27:29.693Z] ok: [localhost] => (item={'key': 'dynamic_ratio', 'value': '1'})
[2022-09-07T10:27:29.769Z] ok: [localhost] => (item={'key': 'description', 'value': 'test-web01. [REDACTED]'})
[2022-09-07T10:27:29.769Z] ok: [localhost] => (item={'key': 'full_path', 'value': '/Common/192.168.145.114'})
[2022-09-07T10:27:29.769Z] ok: [localhost] => (item={'key': 'monitors', 'value': []})
[2022-09-07T10:27:29.769Z] ok: [localhost] => (item={'key': 'name', 'value': '192.168.145.114'})
[2022-09-07T10:27:29.828Z] ok: [localhost] => (item={'key': 'monitors', 'value': []})
[2022-09-07T10:27:29.828Z] ok: [localhost] => (item={'key': 'rate_limit', 'value': '0'})
[2022-09-07T10:27:29.828Z] ok: [localhost] => (item={'key': 'ratio', 'value': '1'})
[2022-09-07T10:27:29.834Z] ok: [localhost] => (item={'key': 'state', 'value': 'enabled'})
[2022-09-07T10:27:29.961Z]
[2022-09-07T10:27:29.961Z] TASK [demo_roles/demo_get_nodes : Write data of nodes to YAML file] *****
[2022-09-07T10:27:30.526Z] changed: [localhost]
[2022-09-07T10:27:30.647Z]

[2022-09-07T10:27:30.647Z] TASK [demo_roles/demo_get_nodes : Gather necessary node information into dictionary] ***
[2022-09-07T10:27:30.761Z] ok: [localhost] => (item={'key': 'address', 'value': '192.168.145.115'})
[2022-09-07T10:27:30.777Z] ok: [localhost] => (item={'key': 'connection_limit', 'value': '0'})
[2022-09-07T10:27:30.818Z] ok: [localhost] => (item={'key': 'dynamic_ratio', 'value': '1'})
[2022-09-07T10:27:30.818Z] ok: [localhost] => (item={'key': 'description', 'value': 'test-web02. [REDACTED]'})
[2022-09-07T10:27:30.900Z] ok: [localhost] => (item={'key': 'full_path', 'value': '/Common/192.168.145.115'})
[2022-09-07T10:27:30.903Z] ok: [localhost] => (item={'key': 'monitors', 'value': []})
[2022-09-07T10:27:30.903Z] ok: [localhost] => (item={'key': 'name', 'value': '192.168.145.115'})
[2022-09-07T10:27:30.903Z] ok: [localhost] => (item={'key': 'monitors', 'value': []})
[2022-09-07T10:27:30.951Z] ok: [localhost] => (item={'key': 'rate_limit', 'value': '0'})
[2022-09-07T10:27:30.952Z] ok: [localhost] => (item={'key': 'ratio', 'value': '1'})
[2022-09-07T10:27:30.952Z] ok: [localhost] => (item={'key': 'state', 'value': 'enabled'})
[2022-09-07T10:27:31.095Z]
[2022-09-07T10:27:31.095Z] TASK [demo_roles/demo_get_nodes : Write data of nodes to YAML file] *****
[2022-09-07T10:27:31.638Z] changed: [localhost]
```

```

[2022-09-07T10:28:50.792Z] TASK [demo_roles/demo_get_nodes : Add "pool" field to nodes that belong to a pool] ***
[2022-09-07T10:28:51.013Z] included: /etc/ansible-intern/demo_roles/demo_get_nodes/tasks/demo_add_pool_field_to_config_file.yaml for localhost => (item={'name':
'192.168.145.114:80', 'partition': 'Common', 'address': '192.168.145.114', 'ephemeral': 'no', 'logging': 'no', 'ratio': 1, 'connection_limit': 0, 'dynamic_ratio': 1,
'full_path': '/Common/192.168.145.114:80', 'inherit_profile': 'yes', 'priority_group': 0, 'rate_limit': 'no', 'fqdn_autopopulate': 'yes', 'monitors': [], 'real_session':
'monitor-enabled', 'real_state': 'up', 'state': 'present'})
[2022-09-07T10:28:51.013Z] included: /etc/ansible-intern/demo_roles/demo_get_nodes/tasks/demo_add_pool_field_to_config_file.yaml for localhost => (item={'name':
'192.168.145.115:80', 'partition': 'Common', 'address': '192.168.145.115', 'ephemeral': 'no', 'logging': 'no', 'ratio': 1, 'connection_limit': 0, 'dynamic_ratio': 1,
'full_path': '/Common/192.168.145.115:80', 'inherit_profile': 'yes', 'priority_group': 0, 'rate_limit': 'no', 'fqdn_autopopulate': 'yes', 'monitors': [], 'real_session':
'monitor-enabled', 'real_state': 'up', 'state': 'present'})
[2022-09-07T10:28:51.127Z]
[2022-09-07T10:28:51.127Z] TASK [demo_roles/demo_get_nodes : Get configuration file name] *****
[2022-09-07T10:28:51.172Z] ok: [localhost]
[2022-09-07T10:28:51.291Z]
[2022-09-07T10:28:51.291Z] TASK [demo_roles/demo_get_nodes : Get configurations from existing YAML file and add pool and port fields to the dictionary] ***
[2022-09-07T10:28:52.257Z] changed: [localhost]
[2022-09-07T10:28:52.399Z]
[2022-09-07T10:28:52.399Z] TASK [demo_roles/demo_get_nodes : Get configuration file name] *****
[2022-09-07T10:28:52.442Z] ok: [localhost]
[2022-09-07T10:28:52.573Z]
[2022-09-07T10:28:52.573Z] TASK [demo_roles/demo_get_nodes : Get configurations from existing YAML file and add pool and port fields to the dictionary] ***
[2022-09-07T10:28:53.528Z] changed: [localhost]

[2022-09-07T10:30:25.114Z] TASK [demo_roles/demo_get_virtual_servers : Filter out necessary data for YAML configuration file] ***
[2022-09-07T10:30:25.241Z] ok: [localhost] => (item={'key': 'name', 'value': 'test-web'})
[2022-09-07T10:30:25.283Z] ok: [localhost] => (item={'key': 'port', 'value': '80'})
[2022-09-07T10:30:25.283Z] ok: [localhost] => (item={'key': 'destination', 'value': '192.168.145.46'})
[2022-09-07T10:30:25.303Z] ok: [localhost] => (item={'key': 'pool', 'value': '/Common/test-web'})
[2022-09-07T10:30:25.442Z] ok: [localhost] => (item={'key': 'description', 'value': ''})
[2022-09-07T10:30:25.442Z] ok: [localhost] => (item={'key': 'ip_protocol', 'value': 'tcp'})
[2022-09-07T10:30:25.442Z] ok: [localhost] => (item={'key': 'type', 'value': 'standard'})
[2022-09-07T10:30:25.442Z] ok: [localhost] => (item={'key': 'profiles', 'value': [{'name': 'http-lan-optimized-nocaching', 'context': 'all', 'full_path': '/Common/http-lan-
optimized-nocaching'}, {'name': 'tcp', 'context': 'all', 'full_path': '/Common/tcp'}]})
[2022-09-07T10:30:25.443Z] ok: [localhost] => (item={'key': 'enabled_vlans', 'value': ['/Common/Test']})
[2022-09-07T10:30:25.443Z] ok: [localhost] => (item={'key': 'port_translation', 'value': 'yes'})
[2022-09-07T10:30:25.463Z] ok: [localhost] => (item={'key': 'snat', 'value': 'automap'})
[2022-09-07T10:30:25.510Z] ok: [localhost] => (item={'key': 'source_port', 'value': 'preserve'})
[2022-09-07T10:30:25.510Z] ok: [localhost] => (item={'key': 'mirror', 'value': 'no'})
[2022-09-07T10:30:25.530Z] ok: [localhost] => (item={'key': 'irules', 'value': []})
[2022-09-07T10:30:25.550Z] ok: [localhost] => (item={'key': 'rate_limit', 'value': '0'})
[2022-09-07T10:30:25.576Z] ok: [localhost] => (item={'key': 'rate_limit_mode', 'value': 'object'})
[2022-09-07T10:30:25.667Z] ok: [localhost] => (item={'key': 'rate_limit_dst_mask', 'value': '0'})
[2022-09-07T10:30:25.667Z] ok: [localhost] => (item={'key': 'source', 'value': '0.0.0.0'})
[2022-09-07T10:30:25.667Z] ok: [localhost] => (item={'key': 'source_port', 'value': 'preserve'})
[2022-09-07T10:30:25.667Z] ok: [localhost] => (item={'key': 'auto_last_hop', 'value': 'default'})
[2022-09-07T10:30:25.668Z] ok: [localhost] => (item={'key': 'full_path', 'value': '/Common/test-web'})
[2022-09-07T10:30:25.884Z]
[2022-09-07T10:30:25.884Z] TASK [demo_roles/demo_get_virtual_servers : Write virtual servers data to virtual servers config file] ***
[2022-09-07T10:30:26.539Z] changed: [localhost]

[2022-09-07T10:30:36.818Z] TASK [demo_roles/demo_new_partition : Creating new partition] *****
[2022-09-07T10:30:45.352Z] changed: [localhost -> localhost]
[2022-09-07T10:30:45.549Z]
[2022-09-07T10:30:45.549Z] TASK [demo_roles/demo_new_partition : Check if partition "ansible-test" was created successfully] ***
[2022-09-07T10:30:49.264Z] ok: [localhost]
[2022-09-07T10:30:49.485Z]
[2022-09-07T10:30:49.485Z] TASK [demo_roles/demo_new_partition : Assert that new partition was created] ***
[2022-09-07T10:30:49.536Z] ok: [localhost] => {
[2022-09-07T10:30:49.536Z]   "changed": false,
[2022-09-07T10:30:49.536Z]   "msg": "Partition ansible-test was created successfully!"
[2022-09-07T10:30:49.536Z] }
[2022-09-07T10:30:50.146Z]
[2022-09-07T10:30:50.146Z] TASK [demo_roles/demo_new_pool : Wait for BIG-IP to be ready to take configuration] ***
[2022-09-07T10:30:56.287Z] ok: [localhost -> localhost]
[2022-09-07T10:30:56.481Z]
[2022-09-07T10:30:56.482Z] TASK [demo_roles/demo_new_pool : Create new pools] *****
[2022-09-07T10:30:56.711Z] included: /etc/ansible-intern/demo_roles/demo_new_pool/tasks/create_demo_pools.yaml for localhost => (item=/etc/ansible-
intern/demo_config_files/pools/test-web.yaml)
[2022-09-07T10:30:57.103Z]
[2022-09-07T10:30:57.104Z] TASK [demo_roles/demo_new_pool : Create pool] *****
[2022-09-07T10:31:05.760Z] changed: [localhost -> localhost]
[2022-09-07T10:31:05.960Z]
[2022-09-07T10:31:05.960Z] TASK [demo_roles/demo_new_pool : Set resource ID] *****
[2022-09-07T10:31:06.435Z] ok: [localhost]
[2022-09-07T10:31:07.035Z]
[2022-09-07T10:31:07.035Z] TASK [demo_roles/demo_new_pool : Check if pool "test-web" was created successfully] ***
[2022-09-07T10:31:10.060Z] ok: [localhost]
[2022-09-07T10:31:10.251Z]
[2022-09-07T10:31:10.251Z] TASK [demo_roles/demo_new_pool : Assert that new pool was created] *****
[2022-09-07T10:31:11.004Z] ok: [localhost] => {
[2022-09-07T10:31:11.004Z]   "changed": false,
[2022-09-07T10:31:11.004Z]   "msg": "Pool test-web was created successfully!"
[2022-09-07T10:31:11.004Z] }

```

```

[2022-09-07T10:31:11.576Z] TASK [demo_roles/demo_migra_node : Wait for BIG-IP to be ready to take configuration] ***
[2022-09-07T10:31:17.608Z] ok: [localhost -> localhost]
[2022-09-07T10:31:17.802Z]
[2022-09-07T10:31:17.803Z] TASK [demo_roles/demo_migra_node : Migrate nodes] *****
[2022-09-07T10:31:18.054Z] included: /etc/ansible-intern/demo_roles/demo_migra_node/tasks/migrate_demo_nodes.yaml for localhost => (item=/etc/ansible-
intern/demo_config_files/nodes/192.168.145.114.yaml)
[2022-09-07T10:31:18.064Z] included: /etc/ansible-intern/demo_roles/demo_migra_node/tasks/migrate_demo_nodes.yaml for localhost => (item=/etc/ansible-
intern/demo_config_files/nodes/192.168.145.115.yaml)
[2022-09-07T10:31:18.903Z]
[2022-09-07T10:31:18.903Z] TASK [demo_roles/demo_migra_node : Delete node "192.168.145.114" with pool membership from partition "Common"] ***
[2022-09-07T10:31:33.643Z] changed: [localhost -> localhost]
[2022-09-07T10:31:34.430Z]
[2022-09-07T10:31:34.430Z] TASK [demo_roles/demo_migra_node : Delete node "192.168.145.114" without pool membership from partition "Common"] ***
[2022-09-07T10:31:34.803Z] skipping: [localhost]
[2022-09-07T10:31:35.493Z]
[2022-09-07T10:31:35.494Z] TASK [demo_roles/demo_migra_node : Create node "192.168.145.114" without pool membership in partition "ansible-test"] ***
[2022-09-07T10:31:46.097Z] changed: [localhost -> localhost]
[2022-09-07T10:31:46.887Z]
[2022-09-07T10:31:46.887Z] TASK [demo_roles/demo_migra_node : Add node "192.168.145.114" to specified pool in partition "ansible-test"] ***
[2022-09-07T10:31:55.465Z] changed: [localhost -> localhost]
[2022-09-07T10:31:55.660Z]
[2022-09-07T10:31:55.660Z] TASK [demo_roles/demo_migra_node : Set resource ID] *****
[2022-09-07T10:31:56.015Z] ok: [localhost]
[2022-09-07T10:31:56.490Z]
[2022-09-07T10:31:56.490Z] TASK [demo_roles/demo_migra_node : Check if node "192.168.145.114" was created successfully] ***
[2022-09-07T10:31:59.600Z] ok: [localhost]
[2022-09-07T10:31:59.792Z]
[2022-09-07T10:31:59.792Z] TASK [demo_roles/demo_migra_node : Assert that new node was created] *****
[2022-09-07T10:32:00.528Z] ok: [localhost] => {
[2022-09-07T10:32:00.528Z]   "changed": false,
[2022-09-07T10:32:00.528Z]   "msg": "Node 192.168.145.114 was created successfully!"
[2022-09-07T10:32:00.528Z] }
[2022-09-07T10:32:01.345Z]
[2022-09-07T10:32:01.345Z] TASK [demo_roles/demo_migra_node : Delete node "192.168.145.115" with pool membership from partition "Common"] ***
[2022-09-07T10:32:15.688Z] changed: [localhost -> localhost]
[2022-09-07T10:32:16.468Z]
[2022-09-07T10:32:16.468Z] TASK [demo_roles/demo_migra_node : Delete node "192.168.145.115" without pool membership from partition "Common"] ***
[2022-09-07T10:32:16.843Z] skipping: [localhost]
[2022-09-07T10:32:17.544Z]
[2022-09-07T10:32:17.544Z] TASK [demo_roles/demo_migra_node : Create node "192.168.145.115" without pool membership in partition "ansible-test"] ***
[2022-09-07T10:32:28.767Z] changed: [localhost -> localhost]
[2022-09-07T10:32:29.567Z]
[2022-09-07T10:32:29.567Z] TASK [demo_roles/demo_migra_node : Add node "192.168.145.115" to specified pool in partition "ansible-test"] ***
[2022-09-07T10:32:37.878Z] changed: [localhost -> localhost]
[2022-09-07T10:32:38.068Z]
[2022-09-07T10:32:38.068Z] TASK [demo_roles/demo_migra_node : Set resource ID] *****
[2022-09-07T10:32:38.512Z] ok: [localhost]
[2022-09-07T10:32:39.083Z]
[2022-09-07T10:32:39.083Z] TASK [demo_roles/demo_migra_node : Check if node "192.168.145.115" was created successfully] ***
[2022-09-07T10:32:41.849Z] ok: [localhost]
[2022-09-07T10:32:42.041Z]
[2022-09-07T10:32:42.041Z] TASK [demo_roles/demo_migra_node : Assert that new node was created] *****
[2022-09-07T10:32:42.640Z] ok: [localhost] => {
[2022-09-07T10:32:42.640Z]   "changed": false,
[2022-09-07T10:32:42.641Z]   "msg": "Node 192.168.145.115 was created successfully!"
[2022-09-07T10:32:42.641Z] }
[2022-09-07T10:32:43.200Z]
[2022-09-07T10:32:43.200Z] TASK [demo_roles/demo_migra_virtual_server : Wait for BIG-IP to be ready to take configuration] ***
[2022-09-07T10:32:49.547Z] ok: [localhost -> localhost]
[2022-09-07T10:32:49.741Z]
[2022-09-07T10:32:49.742Z] TASK [demo_roles/demo_migra_virtual_server : Migrate virtual servers] *****
[2022-09-07T10:32:49.979Z] included: /etc/ansible-intern/demo_roles/demo_migra_virtual_server/tasks/migrate_demo_virtual_servers.yaml for localhost => (item=/etc/ansible-
intern/demo_config_files/virtual_servers/test-web.yaml)
[2022-09-07T10:32:50.352Z]
[2022-09-07T10:32:50.352Z] TASK [demo_roles/demo_migra_virtual_server : Delete virtual servers from partition "Common"] ***
[2022-09-07T10:33:04.244Z] changed: [localhost -> localhost]
[2022-09-07T10:33:05.046Z]
[2022-09-07T10:33:05.047Z] TASK [demo_roles/demo_migra_virtual_server : Create virtual server "test-web" in partition "ansible-test"] ***
[2022-09-07T10:33:20.857Z] changed: [localhost -> localhost]
[2022-09-07T10:33:21.064Z]
[2022-09-07T10:33:21.065Z] TASK [demo_roles/demo_migra_virtual_server : Set resource ID] *****
[2022-09-07T10:33:21.374Z] ok: [localhost]
[2022-09-07T10:33:21.851Z]
[2022-09-07T10:33:21.851Z] TASK [demo_roles/demo_migra_virtual_server : Check if virtual server "test-web" was created successfully] ***
[2022-09-07T10:33:24.985Z] ok: [localhost]
[2022-09-07T10:33:25.185Z]
[2022-09-07T10:33:25.185Z] TASK [demo_roles/demo_migra_virtual_server : Assert that new virtual server was created] ***
[2022-09-07T10:33:25.934Z] ok: [localhost] => {
[2022-09-07T10:33:25.935Z]   "changed": false,
[2022-09-07T10:33:25.935Z]   "msg": "Virtual server test-web was created successfully!"
[2022-09-07T10:33:25.935Z] }
[2022-09-07T10:33:26.494Z]
[2022-09-07T10:33:26.494Z] TASK [demo_roles/demo_delete_pool : Delete pool] *****
[2022-09-07T10:33:36.231Z] changed: [localhost -> localhost] => (item=/etc/ansible-intern/demo_config_files/pools/test-web.yaml)
[2022-09-07T10:33:36.911Z]
[2022-09-07T10:33:36.911Z] PLAY [Get latest configurations from Big-IP and update configuration files] ****
[2022-09-07T10:33:37.102Z]
[2022-09-07T10:33:37.103Z] TASK [Gathering Facts] *****
[2022-09-07T10:33:38.534Z] ok: [localhost]
[2022-09-07T10:33:39.101Z]
[2022-09-07T10:33:39.101Z] TASK [demo_roles/demo_get_pools : Gather data from "██████████" about pools] ***
[2022-09-07T10:33:56.303Z] ok: [localhost -> localhost]
[2022-09-07T10:33:56.544Z]
[2022-09-07T10:33:56.544Z] TASK [demo_roles/demo_get_pools : Make sure that directory for node configuration files exists] ***
[2022-09-07T10:33:56.839Z] ok: [localhost]
[2022-09-07T10:33:57.027Z]

```

```
[2022-09-07T10:34:33.280Z] TASK [demo_roles/demo_get_pools : Gather necessary configuration data from "██████████" about pools and write that data to a file]
***
[2022-09-07T10:34:33.372Z] ok: [localhost] => (item={'key': 'name', 'value': 'test-web'})
[2022-09-07T10:34:33.404Z] ok: [localhost] => (item={'key': 'full_path', 'value': '/ansible-test/test-web'})
[2022-09-07T10:34:33.434Z] ok: [localhost] => (item={'key': 'description', 'value': ''})
[2022-09-07T10:34:33.434Z] ok: [localhost] => (item={'key': 'lb_method', 'value': 'round-robin'})
[2022-09-07T10:34:33.477Z] ok: [localhost] => (item={'key': 'monitor_type', 'value': 'and_list'})
[2022-09-07T10:34:33.477Z] ok: [localhost] => (item={'key': 'monitors', 'value': ['/Common/http']})
[2022-09-07T10:34:33.503Z] ok: [localhost] => (item={'key': 'priority_group_activation', 'value': '0'})
[2022-09-07T10:34:33.541Z] ok: [localhost] => (item={'key': 'reselect_tries', 'value': '0'})
[2022-09-07T10:34:33.542Z] ok: [localhost] => (item={'key': 'service_down_action', 'value': 'none'})
[2022-09-07T10:34:33.556Z] ok: [localhost] => (item={'key': 'slow_ramp_time', 'value': '10'})
[2022-09-07T10:34:33.747Z]
[2022-09-07T10:34:33.747Z] TASK [demo_roles/demo_get_pools : Write pools data to pools config file] *****
[2022-09-07T10:34:34.318Z] changed: [localhost]
[2022-09-07T10:34:34.887Z]
```

```
[2022-09-07T10:35:43.896Z] TASK [demo_roles/demo_get_nodes : Gather necessary node information into dictionary] ***
[2022-09-07T10:35:43.989Z] ok: [localhost] => (item={'key': 'address', 'value': '192.168.145.114'})
[2022-09-07T10:35:44.030Z] ok: [localhost] => (item={'key': 'connection_limit', 'value': '0'})
[2022-09-07T10:35:44.030Z] ok: [localhost] => (item={'key': 'dynamic_ratio', 'value': '1'})
[2022-09-07T10:35:44.049Z] ok: [localhost] => (item={'key': 'description', 'value': 'test-web01.██████████'})
[2022-09-07T10:35:44.142Z] ok: [localhost] => (item={'key': 'full_path', 'value': '/ansible-test/192.168.145.114'})
[2022-09-07T10:35:44.142Z] ok: [localhost] => (item={'key': 'monitors', 'value': []})
[2022-09-07T10:35:44.142Z] ok: [localhost] => (item={'key': 'name', 'value': '192.168.145.114'})
[2022-09-07T10:35:44.142Z] ok: [localhost] => (item={'key': 'monitors', 'value': []})
[2022-09-07T10:35:44.142Z] ok: [localhost] => (item={'key': 'rate_limit', 'value': '0'})
[2022-09-07T10:35:44.177Z] ok: [localhost] => (item={'key': 'ratio', 'value': '1'})
[2022-09-07T10:35:44.177Z] ok: [localhost] => (item={'key': 'state', 'value': 'enabled'})
[2022-09-07T10:35:44.384Z]
[2022-09-07T10:35:44.384Z] TASK [demo_roles/demo_get_nodes : Write data of nodes to YAML file] *****
[2022-09-07T10:35:44.941Z] changed: [localhost]
[2022-09-07T10:35:45.133Z]
[2022-09-07T10:35:45.133Z] TASK [demo_roles/demo_get_nodes : Gather necessary node information into dictionary] ***
[2022-09-07T10:35:45.214Z] ok: [localhost] => (item={'key': 'address', 'value': '192.168.145.115'})
[2022-09-07T10:35:45.233Z] ok: [localhost] => (item={'key': 'connection_limit', 'value': '0'})
[2022-09-07T10:35:45.233Z] ok: [localhost] => (item={'key': 'dynamic_ratio', 'value': '1'})
[2022-09-07T10:35:45.232Z] ok: [localhost] => (item={'key': 'description', 'value': 'test-web02.██████████'})
[2022-09-07T10:35:45.232Z] ok: [localhost] => (item={'key': 'full_path', 'value': '/ansible-test/192.168.145.115'})
[2022-09-07T10:35:45.232Z] ok: [localhost] => (item={'key': 'monitors', 'value': []})
[2022-09-07T10:35:45.232Z] ok: [localhost] => (item={'key': 'name', 'value': '192.168.145.115'})
[2022-09-07T10:35:45.232Z] ok: [localhost] => (item={'key': 'monitors', 'value': []})
[2022-09-07T10:35:45.232Z] ok: [localhost] => (item={'key': 'rate_limit', 'value': '0'})
[2022-09-07T10:35:45.232Z] ok: [localhost] => (item={'key': 'ratio', 'value': '1'})
[2022-09-07T10:35:45.232Z] ok: [localhost] => (item={'key': 'state', 'value': 'enabled'})
[2022-09-07T10:35:45.594Z]
[2022-09-07T10:35:45.594Z] TASK [demo_roles/demo_get_nodes : Write data of nodes to YAML file] *****
[2022-09-07T10:35:46.162Z] changed: [localhost]
```

```
[2022-09-07T10:37:12.609Z] TASK [demo_roles/demo_get_nodes : Add "pool" field to nodes that belong to a pool] ***
[2022-09-07T10:37:12.860Z] included: /etc/ansible-intern/demo_roles/demo_get_nodes/tasks/demo_add_pool_field_to_config_file.yaml for localhost => (item={'name': '192.168.145.114:80', 'partition': 'ansible-test', 'address': '192.168.145.114', 'description': 'test-web01.██████████', 'ephemeral': 'no', 'logging': 'no', 'ratio': 1, 'connection_limit': 0, 'dynamic_ratio': 1, 'full_path': '/ansible-test/192.168.145.114:80', 'inherit_profile': 'yes', 'priority_group': 'no', 'rate_limit': 'no', 'fqdn_autopopulate': 'yes', 'monitors': [], 'real_session': 'monitor-enabled', 'real_state': 'up', 'state': 'present'})
[2022-09-07T10:37:12.866Z] included: /etc/ansible-intern/demo_roles/demo_get_nodes/tasks/demo_add_pool_field_to_config_file.yaml for localhost => (item={'name': '192.168.145.115:80', 'partition': 'ansible-test', 'address': '192.168.145.115', 'description': 'test-web02.██████████', 'ephemeral': 'no', 'logging': 'no', 'ratio': 1, 'connection_limit': 0, 'dynamic_ratio': 1, 'full_path': '/ansible-test/192.168.145.115:80', 'inherit_profile': 'yes', 'priority_group': 'no', 'rate_limit': 'no', 'fqdn_autopopulate': 'yes', 'monitors': [], 'real_session': 'monitor-enabled', 'real_state': 'up', 'state': 'present'})
[2022-09-07T10:37:13.055Z]
[2022-09-07T10:37:13.055Z] TASK [demo_roles/demo_get_nodes : Get configuration file name] *****
[2022-09-07T10:37:13.093Z] ok: [localhost]
[2022-09-07T10:37:13.280Z]
[2022-09-07T10:37:13.281Z] TASK [demo_roles/demo_get_nodes : Get configurations from existing YAML file and add pool and port fields to the dictionary] ***
[2022-09-07T10:37:14.174Z] changed: [localhost]
[2022-09-07T10:37:14.366Z]
[2022-09-07T10:37:14.366Z] TASK [demo_roles/demo_get_nodes : Get configuration file name] *****
[2022-09-07T10:37:14.402Z] ok: [localhost]
[2022-09-07T10:37:14.592Z]
[2022-09-07T10:37:14.592Z] TASK [demo_roles/demo_get_nodes : Get configurations from existing YAML file and add pool and port fields to the dictionary] ***
[2022-09-07T10:37:15.473Z] changed: [localhost]
[2022-09-07T10:37:16.034Z]
```

```
[2022-09-07T10:38:41.118Z] TASK [demo_roles/demo_get_virtual_servers : Filter out necessary data for YAML configuration file] ***
[2022-09-07T10:38:41.214Z] ok: [localhost] => (item={'key': 'name', 'value': 'test-web'})
[2022-09-07T10:38:41.233Z] ok: [localhost] => (item={'key': 'port', 'value': '80'})
[2022-09-07T10:38:41.449Z] ok: [localhost] => (item={'key': 'destination', 'value': '192.168.145.46'})
[2022-09-07T10:38:41.449Z] ok: [localhost] => (item={'key': 'pool', 'value': '/ansible-test/test-web'})
[2022-09-07T10:38:41.449Z] ok: [localhost] => (item={'key': 'description', 'value': ''})
[2022-09-07T10:38:41.449Z] ok: [localhost] => (item={'key': 'ip_protocol', 'value': 'tcp'})
[2022-09-07T10:38:41.533Z] ok: [localhost] => (item={'key': 'type', 'value': 'standard'})
[2022-09-07T10:38:41.533Z] ok: [localhost] => (item={'key': 'profiles', 'value': [{'name': 'http-lan-optimized-nocaching', 'context': 'all', 'full_path': '/Common/http-lan-optimized-nocaching'}, {'name': 'tcp', 'context': 'all', 'full_path': '/Common/tcp'}]})
[2022-09-07T10:38:41.533Z] ok: [localhost] => (item={'key': 'enabled_vlans', 'value': ['/Common/Test']})
[2022-09-07T10:38:41.533Z] ok: [localhost] => (item={'key': 'port_translation', 'value': 'yes'})
[2022-09-07T10:38:41.655Z] ok: [localhost] => (item={'key': 'snat', 'value': 'automap'})
[2022-09-07T10:38:41.655Z] ok: [localhost] => (item={'key': 'source_port', 'value': 'preserve'})
[2022-09-07T10:38:41.655Z] ok: [localhost] => (item={'key': 'mirror', 'value': 'no'})
[2022-09-07T10:38:41.655Z] ok: [localhost] => (item={'key': 'irules', 'value': []})
[2022-09-07T10:38:41.663Z] ok: [localhost] => (item={'key': 'rate_limit', 'value': '0'})
[2022-09-07T10:38:41.663Z] ok: [localhost] => (item={'key': 'rate_limit_mode', 'value': 'object'})
[2022-09-07T10:38:41.669Z] ok: [localhost] => (item={'key': 'rate_limit_dst_mask', 'value': '0'})
[2022-09-07T10:38:41.726Z] ok: [localhost] => (item={'key': 'source', 'value': '0.0.0.0/0'})
[2022-09-07T10:38:41.726Z] ok: [localhost] => (item={'key': 'source_port', 'value': 'preserve'})
[2022-09-07T10:38:41.742Z] ok: [localhost] => (item={'key': 'auto_last_hop', 'value': 'default'})
[2022-09-07T10:38:41.753Z] ok: [localhost] => (item={'key': 'full_path', 'value': '/ansible-test/test-web'})
[2022-09-07T10:38:41.955Z]
[2022-09-07T10:38:41.955Z] TASK [demo_roles/demo_get_virtual_servers : Write virtual servers data to virtual servers config file] ***
[2022-09-07T10:38:42.525Z] changed: [localhost]
[2022-09-07T10:38:43.240Z]
[2022-09-07T10:38:43.240Z] PLAY RECAP *****
[2022-09-07T10:38:43.240Z] localhost : ok=855 changed=26 unreachable=0 failed=0 skipped=1288 rescued=0 ignored=0
[2022-09-07T10:38:43.241Z]
```


Lisa 22 – *Playbook*'i jooksutamise käigus loodud komponentide seadistusfailid

Ühend test-web:

```
"description": ""
"full_path": "/Common/test-web"
"lb_method": "round-robin"
"monitor_type": "and_list"
"monitors":
- "/Common/http"
"name": "test-web"
"priority_group_activation": "0"
"reselect_tries": "0"
"service_down_action": "none"
"slow_ramp_time": "10"
```

Virtuaalserver test-web01:

```
"address": "192.168.145.114"
"connection_limit": "0"
"description": "test-web01. [REDACTED]"
"dynamic_ratio": "1"
"full_path": "/Common/192.168.145.114:80"
"monitors": []
"name": "192.168.145.114:80"
"partition": "Common"
"rate_limit": "0"
"ratio": "1"
"state": "enabled"
"pool": "/Common/test-web"
```

Virtuaalserver test-web02:

```
"address": "192.168.145.115"
"connection_limit": "0"
"description": "test-web02. [REDACTED]"
"dynamic_ratio": "1"
"full_path": "/Common/192.168.145.115:80"
"monitors": []
"name": "192.168.145.115:80"
"partition": "Common"
"rate_limit": "0"
"ratio": "1"
"state": "enabled"
"pool": "/Common/test-web"
```

Koormusjaotur test-web:

```
"auto_last_hop": "default"
"description": ""
"destination": "192.168.145.46"
"enabled_vlans":
- "/Common/Test"
"full_path": "/Common/test-web"
"ip_protocol": "tcp"
"irules": []
"mirror": "no"
"name": "test-web"
"pool": "/Common/test-web"
"port": "80"
"port_translation": "yes"
"profiles":
- "context": "all"
  "full_path": "/Common/http-lan-optimized-nocaching"
  "name": "http-lan-optimized-nocaching"
- "context": "all"
  "full_path": "/Common/tcp"
  "name": "tcp"
"rate_limit": "0"
"rate_limit_dst_mask": "0"
"rate_limit_mode": "object"
"snat": "automap"
"source": "0.0.0.0/0"
"source_port": "preserve"
"type": "standard"
```

Lisa 23 – Skripti curl_test.py terminali tagastatud tulemused

```
Get request time: 2022-09-07 13:31:10.689650 Web service is available.
Get request time: 2022-09-07 13:31:25.730770 Web service is available.
Get request time: 2022-09-07 13:31:40.767792 Web service is available.
Get request time: 2022-09-07 13:31:55.806922 Web service is available.
Get request time: 2022-09-07 13:32:10.843463 Web service is not available!
Get request time: 2022-09-07 13:32:25.882773 Web service is not available!
Get request time: 2022-09-07 13:32:40.922895 Web service is not available!
Get request time: 2022-09-07 13:32:55.959361 Web service is not available!
Get request time: 2022-09-07 13:33:26.031503 Web service is available.
Get request time: 2022-09-07 13:33:41.069427 Web service is available.
Get request time: 2022-09-07 13:33:56.100274 Web service is available.
```

Lisa 24 – Partitsooni ansible-test liigutatud ressursid Big-IP kasutajaliideses

Local Traffic » Virtual Servers : Virtual Server List » test-web

Properties Resources Statistics

General Properties

Name	test-web
Partition / Path	ansible-test
Description	<input type="text"/>
Type	Standard
Source Address	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text" value="0.0.0.0/0"/>
Destination Address/Mask	<input checked="" type="radio"/> Host <input type="radio"/> Address List <input type="text" value="192.168.145.46"/>
Service Port	<input checked="" type="radio"/> Port <input type="radio"/> Port List <input type="text" value="80"/> HTTP

Local Traffic » Virtual Servers : Virtual Server List » test-web

Properties Resources Statistics

Load Balancing

Default Pool	test-web
Default Persistence Profile	None
Fallback Persistence Profile	None

Pools : Pool List » test-web

Members Statistics

Method: Round Robin
 Activation: Disabled

Member	Address	Service Port	FQDN	Ephemeral	Ratio	Priority Group	Connection Limit	Partition / Path
	192.168.145.114:80	192.168.145.114 80		No	1	0 (Inactive)	0	ansible-test
	192.168.145.115:80	192.168.145.115 80		No	1	0 (Inactive)	0	ansible-test