

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Martin Vool

# **Kubernetese laborikeskkond Entigo OÜ uue koolituse tarbeks**

Bakalaureusetöö

Juhendaja: Siim Vene  
Magister  
Rein Remmel  
Rakenduslik  
kõrgharidus

Tallinn 2022

## **Autorideklaratsioon**

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

Autor: Martin Vool

24.04.2022

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärgiks on uurida erinevaid Kubernetese kobarate käitustööriistu Entigo OÜ uue koolituse laborikeskkonna tarbeks ja praktilises osas realiseerida valitud tööriistaga koolituse laborite läbiviimise keskkond.

Seni kasutusel olnud laborikeskkonnad ei rahulda planeeritud koolituse vajadusi. Diplomitöös uuritavate Kubernetese kobarate käitustööriista juures on oluline maksumus, sobivus Entigo OÜ olemasolevate praktikatega ja võimekus simuleerida toodangu keskkonnaga sarnaseid Kubernetese kobarate topoloogiaid laborikeskkonnas.

Töö fookuses on tööriist, mis aitaks käidelda mitut Kubernetese kobarat laborikeskkonnas, et seal saaks viia läbi koolituse harjutusi. Uurimise osas püstitatakse tööriistale nõuded, teostatakse erinevate tööriistade võrdlust ja hindamist nõuetele, kus lähtutakse tööriistade dokumentatsioonist ning tehakse mõõtmised.

Töö praktilises osas realiseeritakse teoreetilises osas välja valitud tööriistaga Entigo OÜ koolituse laborikeskkond, et leida kinnitust selle sobivuses. Selle jaoks paigaldatakse Entigo OÜ teistes koolitustes kasutuses olev näidisrakendus loodud Kubernetese kobarasse, kasutades välja kujunenud tööriistu ja praktikaid. Selle praktilise töö raames ei looda uut koolitust või selles kasutatavaid harjutusi. Luuakse sobiv laborikeskkond, kus neid saaks tulevikus teha.

Töö teoreetilise osa tulemusena valiti välja kõige paremini sobiv tööriist koolituse laborikeskkonna jaoks. Praktilises osas välja valitud tööriista kasutades õnnestus luua soovitud koolituse laborikeskkond ja kontrolliti praktikas tööriista vastavust teoreetilises osas püstitatud nõuetele.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 46 leheküljel, 5 peatükki, 21 joonist, 8 tabelit.

## **Abstract**

### **Kubernetes Laboratory Environment for Entigo OÜ's New Training**

This bachelor thesis investigates different Kubernetes clusters operation tools to be used for Entigo OÜ's new training laboratory environment and in the practical part realizes the laboratory environment of the training with the chosen tool.

The laboratory environments that are used until now do not satisfy the needs of the new training. The important aspects of the Kubernetes clusters operation tool are cost, suitability with Entigo OÜ's existing practices and ability to simulate production-like environment topologies in a laboratory environment.

The focus of the thesis is a tool that would help to operate multiple Kubernetes clusters in a laboratory environment, so it could be used to do training exercises. Requirements are set for the tool. Different tools are compared and graded according to set requirements. For grading the tools documentation is used and measurements are done.

In the practical part of the thesis, the laboratory environment for Entigo OÜ's training is created using the tools chosen in the research to validate the suitability of the tool.

For that Entigo OÜ's demonstration application that was used in the previous training is installed into the Kubernetes cluster, using developed tools and practices. In this thesis, new training or exercises are not developed. A suitable laboratory environment is created where exercises could be developed in the future.

As the result of the research, the most suitable tool was chosen for the training laboratory environment. In the practical part, the creation of the laboratory environment was successful with the tool and the compliance with the requirements was checked.

The thesis is in Estonian and contains 46 pages of text, 5 chapters, 21 figures, 8 tables.

## Lühendite ja mõistete sõnastik

API server	<i>API Server</i> , Kubernetese juhttasandi rakendusliidese komponent [1]
Argo CD	Tööriist, mis koodihoidlat kasutades võimaldab pidevvalmidust Kubernetese kobarates
AWS	<i>Amazon Web Services</i>
AWS EC2	<i>Amazon Web Services Elastic Compute Cloud</i>
AWS ECR	<i>Amazon Web Services Elastic Container Registry</i>
AWS EKS	<i>Amazon Web Services Elastic Kubernetes Service</i>
CKA	<i>Certified Kubernetes Administrator</i>
CKAD	<i>Certified Kubernetes Application Developer</i>
CNCF	<i>Cloud Native Computing Foundation</i>
CRD	<i>Custom Resource Definition</i>
Etd andmebaas	<i>Etd Database</i> , Võtmeväärtuste andmebaas, kus Kubernetes hoiab andmeid [1]
<i>Ingress</i>	Kubernetese objekt ühtsete ressursilokaatorite vastavusse seadmiseks teenustega
Juhtsõlm	<i>Control node</i> , Kubernetese sõlm, millel töötavad Kubernetese juhttasandi komponendid
Kaun	<i>Pod</i> , Kubernetese kobaras töötav rakenduse konteiner või konteinerid, mis paiknevad ühel sõlmel.
Kontrollerite haldaja	<i>Controller Manager</i> , Kubernetese juhttasandi komponent, mis juhib erinevate Kubernetese objektide tööd [1]
Kubernetese juhttasand	<i>Kubernetes Control Plane</i> , Kubernetese tööks vajalikud komponendid, mis töötavad juhtsõlmedel. [1] [2]
Koormusjaoturi tüüpi teenus	<i>Loadbalancer type Service</i>
LXC	<i>Linux Containers</i>
Planeerija	<i>Scheduler</i> , Kubernetese juhttasandi komponent, mis paigutab kaunad sõlmedele [1]
Snapcraft	Rakenduste pakihaldur [3]

SSH	<i>Secure Shell</i> , Protokoll, millega luuakse turvaline ühendus kahe arvuti või serveri vahel
Sõlm	<i>Node</i> , Kubernetese sõlm, mis võib olla nii juhtsõlm kui töötajasõlm
Töötajasõlm	<i>Worker node</i> , Kubernetese sõlm, millel töötavad rakendused
USD	<i>United States dollar</i> , Ameerika Ühendriikide dollar

## Sisukord

Autorideklaratsioon .....	2
Annotatsioon.....	3
Abstract Kubernetes Laboratory Environment for Entigo OÜ's New Training .....	4
Lühendite ja mõistete sõnastik .....	5
Sisukord .....	7
Jooniste loetelu .....	9
Tabelite loetelu .....	10
1 Sissejuhatus .....	11
1.1 Taust ja probleem .....	12
1.2 Ülesande püstitus .....	13
1.3 Ülevaade tööst .....	13
2 Metoodika.....	15
2.1 Nõuete täitmise hindamine .....	15
2.2 Tööriista omaduste mõõtmine .....	16
3 Töö teoreetilised alused .....	18
3.1 Olemasolevate laborikeskkondade kirjeldus .....	18
3.1.1 Kubernetese põhialused 1 .....	18
3.1.2 Kubernetese põhialused 2.....	19
3.2 Nõuete analüüs .....	19
3.3 Tööriistade ülevaade.....	22
3.4 Tööriistade omaduste mõõtmise tulemused .....	22
3.5 Tööriistade nõuete täitmise analüüs .....	24
3.5.1 Minikube.....	24
3.5.2 K3d .....	25
3.5.3 Microk8s.....	27
3.5.4 Kind .....	28
3.6 Tööriistade võrdluse tulemused.....	29
4 Laborikeskkonna loomine .....	31
4.1 Virtuaalmasinate loomine ja tarkvara konteinerite puhverserver.....	31

4.2 Kubernetese kobarate loomine Kind tööriistaga .....	33
4.3 Entigo näidisrakenduse paigaldus .....	35
4.4 Tõrkesiirde test .....	38
4.5 Järeldused .....	39
5 Kokkuvõte .....	41
Kasutatud kirjandus .....	42
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	47
Lisa 2 – Skript kobara omaduste mõõtmiseks .....	48
Lisa 3 – Minikube tööriista omaduste mõõtmise käsud .....	49
Lisa 4 – Microk8s tööriista omaduste mõõtmise käsud .....	50
Lisa 5 – K3d tööriista omaduste mõõtmise käsud .....	52
Lisa 6 – Kind tööriista omaduste mõõtmise käsud.....	53
Lisa 7 – Laborikeskkonna ekraanitõmmised .....	54



## Jooniste loetelu

Joonis 1. Terraformiga loodud komponentide skeem .....	32
Joonis 2. Koolitusel osaleja virtuaalmasinasse paigaldatud komponendid ja ligipääs...	33
Joonis 3. Virtuaalmasinasse tööriistaga loodav kobarate arhitektuur .....	34
Joonis 4. Rakenduste loomine Argo CD tööriistaga.....	36
Joonis 5. Teenustele ligipääsu tekitamine avalikust internetist.....	38
Joonis 6. AWS EC2 koolituse laborikeskkonnas kasutatavad virtuaalmasinad.....	54
Joonis 7. Harbor tarkvara konteinerite registri kasutajaliidese vaade .....	54
Joonis 8. AWS veebiliidese vaade Route53 teenusesse loodud nimeserveri kanded.....	55
Joonis 9. Ligipääs virtuaalmasinasse ja koodihoidlast pärit failide kuvamine.....	55
Joonis 10. Kubernetese kobarate loomine Kind tööriistaga .....	56
Joonis 11. Kubernetese kobaratega suhtluse testimine kubectl tööriistaga.....	56
Joonis 12. Kubernetese juhttasandi komponentide töötamine juhtsõlmede kaunades ...	57
Joonis 13. Tööriista loodud kobara sõlmede töötamine konteinerites .....	57
Joonis 14. Entigo näidisrakenduse objektide kuvamine.....	58
Joonis 15. Koormusjaotur tüüpi teenuse objektid ja võrguühenduste suunamine.....	58
Joonis 16. Entigo näidisrakendus „test“ kobaras.....	59
Joonis 17. Entigo näidisrakendus „prod“ kobaras .....	59
Joonis 18. Kubernetese sõlmed enne tõrkesiirde testi .....	60
Joonis 19. Entigo näidisrakenduse kaunad enne tõrkesiirde testi.....	60
Joonis 20. Kubernetese sõlmed pärast tõrkesiirde testi .....	60
Joonis 21. Entigo näidisrakenduse kaunad pärast tõrkesiirde testi.....	60

## Tabelite loetelu

Tabel 1. Nõuete täitmise hindamine .....	16
Tabel 2. Tööriistade omaduste mõõtmise tulemused .....	23
Tabel 3. Minikube nõuete täitmine.....	24
Tabel 4. K3d nõuete täitmine .....	25
Tabel 5. Microk8s nõuete täitmine .....	27
Tabel 6. Kind nõuete täitmine .....	28
Tabel 7. Laborikeskkondade Kubernetese kobarate maksumused .....	39
Tabel 8. Kind tööriista nõuete täitmine praktikas.....	40

# 1 Sissejuhatus

Käesoleva töö autor töötab Entigo OÜ-s ja on viimased 4 aastat läbi viinud Kubernetese põhialuste koolitusi. Koolitused said alguse vajadusest anda Entigo OÜ klientidele üle loodud tehnilised lahendused, et tagada klientidele võimekus nende lahenduste kasutamise ja hooldamisega iseseisvalt toime tulla. Lisaks on koolitused vajalikud teadmiste ühtlustamiseks, et klientidel oleks võimalus ühiste projektide raames Kubernetese ja rakenduskonteineritega seotud tehnilistel teemadel kaasa rääkida.

2021. aastal on viis töö autor läbi 9 koolitust, neis osales keskmiselt 10 inimest, seega kokku 90 inimest. Klientide seas on nii Eesti riigiasutusi, kui ka globaalseid ja kohalikke eraettevõtteid.

Olemasolev koolitusprogramm põhineb Cloud Native Computing Foundation (edaspidi CNCF) Certified Kubernetes Application Developer (edaspidi CKAD) ja Certified Kubernetes Administrator (edaspidi CKA) ametlikul õppekaval. Koolitus koosneb kolmest osast: “Rakenduse konteinerid”, “Kubernetes põhialused 1”, “Kubernetes põhialused 2”. Eelmainitud sertifikaatide eksamid koosnevad ainult praktilistest harjutustest ja sellest tulenevalt on oluline koolitusel praktiliste harjutuste maht ning laborikeskkonna võimekus harjutusi teostada [4] [5].

Praeguse koolitusprogrammi kogu kestus on viis tööpäeva ja on väga tugeva fookusega praktilistel harjutustel, mida tehakse Entigo OÜ poolt ette valmistatud laborikeskkondades. Igal koolituse osal on sellele vastav laborikeskkond, mis on sobiv õpetatava praktikaga.

Koolitusel on juba käsitletud rakenduse konteinereid, erinevaid Kubernetese kobarate paigaldamise meetodeid ja rakenduse käideldavuse ning kättesaadavuse tagamise viise Kubernetese objektide abil. Põgusalt käsitletakse ka seiret, tõrkesiiret, automatiseerimist ja mitme Kubernetese kobara halduseks vajalike vahendeid ning võimalusi, sest CKA ja CKAD sertifitseerimisel neid teemasid süvitsi ei nõuta [6] [7].

Koolitusel osalejad on tundnud huvi, kuidas Kubernetese tehnoloogiat igapäevatoos kasutusele võtta ja töös hoida. Sealhulgas kuidas kasutada Kubernetest tarkvara tarneprotsesside automatiseerimiseks, kõrgkäideldavuse tagamiseks ja kuidas korraldada Kubernetese infrastruktuuri seire ja hooldus.

Entigo OÜ-s planeeritakse järgmise koolituse väljatöötamist, mis on mõeldud edasijõudnutele ja oleks loogiline jätk Kubernetese põhitõdede koolituse läbinud inimestele. Ka sellele koolituse osale on vaja välja töötada sobilik laborikeskkond.

## 1.1 Taust ja probleem

Olemasolevad koolitused toimuvad keskmiselt korra kuus. Kogu koolitusprogramm kestab viis päeva. Laborikeskkonnad on vaja kustutada ja uuesti käivitada kuna eelnevalt koolitusest ei tohi jääda muudatusi. Lisaks ei ole rahaliselt mõistlik koolituse laborikeskkondasid pidevalt töös hoida. Kõik varasemad laborikeskkonnad on umbes poole tunniga võimalik käivitada ja sulgeda Amazon Web Services (edaspidi AWS) pilvekeskkonnas kasutades Terraform tööriista. Entigo OÜ on AWS Partner ja soovib kasutada koolituse laborikeskkondade jaoks AWS pilveteenuseid. Planeeritav koolituse laborikeskkond peab töötama AWS keskkonnas ning olema lihtsalt käivitatav ja suletav.

Koolitusprogrammi olemasolevates osades on lähtuvalt harjutuse vajadustest õpilastele kättesaadavad erinevad ressursid, näiteks ainult üks virtuaalmasin, AWS Elastic Kubernetes Service (edaspidi EKS) kobar või oma Kubernetese kobara ehitamiseks vajalikud virtuaalmasinad ja koormusjaoturid. Senised laborikeskkonnad kasutavad AWS spetsiifilisi komponente, aga koolituste laborites lahendatavad harjutused on tehtud universaalselt kasutatavaks - ehk need ei ole AWS spetsiifilised. Näiteks kasutatakse AWS EKS seal, kus koolitatakse ei tegele Kubernetese kobara loomisega vaid on selle tarbija rollis, samas kui on vaja rakenduskonteineri register luua, siis ei kasutata AWS *Elastic Container Registry* (edaspidi ECR) vaid kasutatakse vabavaralist Harbor tarkvara, mis on infrastruktuuri pakkujast sõltumatu [8]. Seda sama põhimõtet peab jätkama ka loodav laborikeskkond. Seega ei tohi arendatav laborikeskkond olla ühe pilveteenuse pakkuja keskkonna spetsiifiline, kui see on oluline õpetatava teema juures.

Uuele koolitusele on eelkõige oodata juba varasematel koolitustel osalenud inimesi. Sellest kõigest tulenevalt jäävad ka loodava laborikeskkonna nõuded osaleja arvutile

samaks eelnevatega. Koolitus laborites osalemiseks peab osalejatel olema oma sülearvuti, millega laborikeskkonda ühenduda. Ühel koolitusel võib osaleda erinevatest asutustest inimesi, siis on ka kõigil sageli erinevate parameetritega raudvara, operatsioonisüsteemid ja asutuse spetsiifilised seadistused. Sülearvutis on vajalik veebibrauser, *Secure Shell* (edaspidi SSH) terminali klient ja võrguühenduses lubatud teha internetti TCP ühendusi 443, 80, 8080 ja 22 portidel. Eelkirjeldatud nõudmistega seni kõik osalejad saanud hakkama.

Olemasolevates laborikeskkondades ei soovita uue koolituse harjutusi teha, kuna seal on teistsugused vajadused. „Kubernetes põhialused 1“ laborikeskkonna puuduseks on Kubernetese kobarate paindlikus ning loomise ja kustutamise kiirus. „Kubernetes põhialused 2“ laborikeskkonna puuduseks on laborikeskkonna maksumus ja seal kasutatava tööriista omadused. Tuleb leida parem lahendus uue koolituse laborikeskkonnale.

## **1.2 Ülesande püstitus**

Käesoleva diplomitöö eesmärgiks on selgitada välja planeeritud koolituse laborikeskkonna nõuded ja leida sobivaim tööriist planeeritud koolituse laborikeskkonnas Kubernetese kobarate käitamiseks. Praktilises osas on eesmärgiks kontrollida valitud tööriista sobivust süvitsi.

Diplomitöö peamiseks eesmärkideks on:

- Selgitada välja nõuded planeeritud koolituse laborikeskkonnale
- Analüüsida erinevate tööriistade vastavust püstitatud nõuetele
- Realiseerida laborikeskkond valitud tööriistaga

## **1.3 Ülevaade tööst**

Esmalt sõnastatakse meetodika, kus pannakse paika, kuidas nõuded püstitatakse ja nõuete täitmist kontrollitakse. Seejärel analüüsitakse vastavalt meetodikale, kus leitakse tööriistade nõuded. Kui nõuded on leitud, siis kontrollitakse erinevate tööriistade

vastavust nõuetele. Valitakse välja kõige paremini sobiv tööriist ja sellega realiseeritakse praktilises osas laborikeskkond, et kontrollida valitud tööriista sobivust.

## **2 Metoodika**

Püstitatud ülesande lahendamiseks kasutab autor nelja aastast koolitus- ja töökogemust Kubernetesega Entigo OÜ-s ja IT süsteemide administreerimise õppekavas omandatud teadmisi ja töövahendeid.

Töö teoreetilises osas esmalt analüüsitakse nõudeid, mida Kubernetese kobaraid käitev tööriist peab täitma loodava koolituse laborikeskkonnas. Seda tehakse kirjeldades planeeritavat koolitust ja sinna planeeritavaid harjutusi. Analüüsi tulemusena pannakse paika nõuded tööriistale.

Seejärel hinnatakse erinevate Kubernetese kobarate paigaldamise tööriistade võimet täita nõudeid. Tööriistad, mida hakatakse uurima on välja valitud Kubernetese enda dokumentatsioonist, blogipostitustest ja veebilehtedel olevatest võrdlustest teiste tööriistadega. Nõuete kontrollimiseks analüüsitakse tööriistade dokumentatsiooni alusel vastavust esitatud nõuetele ja teostatakse mõõtmine.

Lõpuks valitakse välja kõige sobivaim tööriist kasutades nõuete täitmisel saadud tulemusi ja töö praktilises osas realiseeritakse sellega laborikeskkond, kus pannakse käima näidisrakendus kasutades Entigo OÜ-s levinud praktikaid.

### **2.1 Nõuete täitmise hindamine**

Parima tööriista valimiseks tehakse iga tööriista kohta tabel, kus hinnatakse seatud nõuete täitmist nelja punkti skaalal, vt tabel 1. Töös esitatud nõuded on ühe kaaluga. Parima tööriista valimiseks liidetakse tööriista nõuete täitmisel antud hinded kokku. Suurem summa tähendab, et tööriist sobib kõige paremini laborikeskkonna nõuete täitmiseks.

Tabel 1. Nõuete täitmise hindamine

Hinne	Nõudele vastavuse selgitus
0	Üldse ei täida
1	Täidab paljude puudustega
2	Täidab väikse puudusega
3	Täidab täielikult

## 2.2 Tööriista omaduste mõõtmine

Kuigi enamike tööriistade dokumentatsioonis on näiteid mälu kasutusest ja käivituse kiirusest, siis need ei ole omavahel võrreldavad, sest seal ei ole kirjeldatud, millises keskkonnas ja mis seadistuses need mõõtmised on tehtud. Selle töö raames tehtud mõõtmistes võrreldakse Kubernetese kobaraid, mis koosnevad kolmest juhtsõlmest (*control node*) ja viiest töötajasõlmest (*worker node*). Kui tööriist ei võimalda kolme juhtsõlme loomist, siis nende asemel kasutatakse töötajasõlmi. Mõõtmised on vaja teha, sest antud töö fookuses on suurema sõlmede arvuga kobarad, mille puhul tööriistade dokumentatsioonist ei ole võimalik tuletada vajaminevat mälu kogust või kuluvat aega.

Kõik mõõtmised teostakse AWS *Elastic Compute Cloud* (edaspidi AWS EC2) m5.large virtuaalmasinatel. Seda tüüpi virtuaalmasinal on 2 tuuma ja 8GB mälu [9]. Need virtuaalmasinad tagavad ühtlase jõudluse [10].

Kasutatakse AWS Frankfurdi regiooni, sest seda on kasutatud ka varasemates koolitustes. Igale tööriistale käivitatakse oma AWS EC2 virtuaalmasin ja sellele paigaldatakse ka Docker tarkvara.

Iga tööriista jaoks on kirjutatud skript, mis mõõdab:

1. Tööriista paigaldamiseks kulunud aega sekundites.
2. Tööriistaga Kubernetese kobara loomiseks kulunud aega sekundites.
3. Aega mis kulub kõikide Kubernetese kobarate sõlmede valmisolekuni.



4. Mälukasutuse muutust megabaitides operatsioonisüsteemis alates tööriista paigaldusest kuni kõikide sõlmede valmisolekuni.
5. Aega mis kulub tööriistal Kubernetese kobara kustutamiseks.

Tööriista käivitatakse kahel korral, et näha, kas tööriistal on optimeeringuid, mis teevad teistkordse kobara loomise kiiremaks. Kui tööriistaga teisel korral Kubernetese kobara loomine on kiirem, siis tuleb arvesse võtta parema tulemuse saanud mõõtmise. Teisel käivitamisel ei arvestata tööriista paigaldamiseks kulunud aega, kobara kustutamiseks kulunud aega ega mälukasutuse muutumist.

Skript näitab väljundisse Kubernetese juhttasandi komponentide kaunad(*Pods*), kui need on olemas. Skriptid on kõik ühesugused, erinevad ainult tööriista väljakutsumise sammudes, sest need on tööriista spetsiifilised, vt. Lisa 2.

### 3 Töö teoreetilised alused

Kirjeldatakse olemasolevaid laborikeskkondasid ja uues koolituses lahendatavaid harjutusi. Seejärel analüüsitakse uue laborikeskkonna vajadusi ja püstitatakse vajaduste põhjal nõuded. Antakse lühike ülevaade valitud tööriistadest. Teostatakse tööriistade omaduste mõõtmine ja püstitatud nõuete analüüs. Valitakse välja kõige paremini sobiv tööriist, millega realiseerida praktilises osas laborikeskkond.

#### 3.1 Olemasolevate laborikeskkondade kirjeldus

Koolitustel on kasutusel kaks erinevat laborikeskkonda. Esimene Kubernetest kasutatav koolituse laborikeskkond on seotud “Kubernetese põhialused 1” koolitusega, mis põhineb CKAD õppekaval [7]. Seal keskendutakse Kubernetese kobara rakendustele, mitte Kubernetese kobara haldamisele [11]. Teise koolituse labor on seotud “Kubernetese põhialused 2” koolitusega, mis põhineb CKA õppekaval [6]. Seal keskendutakse ühe Kubernetese kobara paigaldamisele, uuendamisele ja haldusele [12].

##### 3.1.1 Kubernetese põhialused 1

Selles laborikeskkonnas kasutatakse AWS EKS teenust ja *Managed node groups* teenust. Selles koolituses ei tegeleta Kubernetese kobara loomise, uuendamise või haldamisega, seetõttu ei ole koolitatava jaoks oluline, millise tööriistaga Kubernetese kobar on loodud ja kas koolitatav pääseb ligi Kubernetese juhttasandi komponentidele.

Kogu laborikeskkond käivitatakse kasutades Terraformi, mis loob AWS teenuseid kasutades laborikeskkonna. Isiklik kogemus varasematest koolitustes on näidanud, et EKS teenuse loomine võtab 12-20 minutit ja iga töötajasõlme lisamine veel 2 minutit. See aeg on välja toodud ka Githubi probleemipüstituses [13].

Kuna EKS on AWS poolt hallatud teenus, siis puudub lõppkasutajal ligipääs juhtsõlmedele ja enamikele Kubernetese juhttasandi komponentidele [14] [15]. Selle tõttu ei saa kõiki seadistusi muuta, ega tekitada tõrkesiirdeid. Lisaks ei oleks koolitataval ülevaadet, mis toimub Kubernetese juhttasandi komponentidega. AWS EKS teenuse puhul ei ole võimalik kasutada MetalLB tarkvara oma koormusjaoturi tüüpi teenuse (*Loadbalancer type Service*) loomiseks [16].

### 3.1.2 Kubernetese põhialused 2

Selles laborikeskkonnas kasutatakse AWS EC2 t3.large virtuaalmasinaid. Kogu laborikeskkond käivitatakse kasutades Terraformi ja võtab aega umbes 3 minutit iga sõlme kohta.

CKA õppekavas on nõutud Kubeadm tööriista kasutamise oskus [6]. Seega Kubernetese kobar paigaldatakse koolitavate poolt iseseisvalt seda tööriista kasutades. Kubeadm tööriista eesmärk on ainult kobara tarkvara seadistus ja käivitus, mitte virtuaalmasinate või tarkvara konteinerite töötajasõlmede säte. Sellest tulenevalt puudub tööriistal endal ka mitme kobara keskne halduse võimekus [17]. Lisaks AWS EC2 virtuaalmasinatele ei ole võimalik kasutada MetalLB koormusjaoturi tarkvara [16].

Kubeadm miinimum nõueteks on 2GB mälu ja 2 tuuma [18]. Kõige odavam AWS virtuaalmasin mis seda nõuet täidab on t3.small. Frankfurdis maksab see 0.024 USD tunnis [9]. Ööpäeva maksumus kolmele juhtsõlmele on 1.728 USD ja viiele töötajasõlmele 2.88USD. Ühe kobara kulu ööpäevas on 4.61 USD. Sellises seadistuses Kubernetese kobara maksumus on võrreldav antud töös loodavate teiste kobaratega.

### 3.2 Nõuete analüüs

Tuuakse välja uue koolituse vajadused koos põhjendustega. Nendest tuletatakse vajalikud nõuded tööriistale.

1. Seniste koolituste tagasisides on osalejad soovinud, et laborites käsitletakse pideva tarne(*continious delivery*) ahelate ehitamist üle mitme keskkonna, kus eri keskkonnad töötavad erinevatel Kubernetese kobaratel [19]. Käsitleda tuleb ka olukorda, kus sama rakendus töötab ühes keskkonnas, aga kahes erinevas Kubernetese kobaras. Seega on ühel osalejal vaja luua ja kustutada mitu Kubernetese kobarat laborikeskkonnas.
2. Senise praktika käigus on väga sageli olnud vajadus eraldada Kubernetese kobara töötajasõlmi kindlatele rakendustele. Näiteks on erineva otstarbega seire, andmebaasi või koormusjaoturi töötajasõlmed, kuhu teised rakendused ei ole lubatud [20] [21]. Nende oskuste harjutamiseks tuleb luua erineva topoloogiaga Kubernetese kobaraid. Seega labori tööriist peab olema võimeline looma

eriotstarbega töötajasõlmi. Eriotstarbega töötajasõlmele on määratud *label* ja *taint* [22].

3. Senine autori kogemus nii füüsilise kui pilveinfrastruktuuriga on näidanud, et elame maailmas, kus asjad lähevad katki või ei tööta nii nagu ette nähtud. Planeeritud on kõrgkäideldavuse ja tõrkesiirde harjutused. Eesmärgiks on simuleerida sõlmede katkestusi ja juhttasandi komponentide tõrkesiirdeid. Kubernetese juhttasandi kõrgkäideldavus on tagatud vähemalt kolme juhtsõlme olemasolul [23]. Seega Kubernetese juhttasand peab koosnema vähemalt kolmest juhtsõlmest.
4. Kõrgkäideldavuse ja tõrkesiirde harjutustes on oluline, et koolitatavatel oleks mugav ligipääs kõikidele Kubernetese juhttasandi komponentidele. Kubernetese juhttasandi komponendid *etcd* andmebaas(*etcd database*), *API server*(*API server*), planeerija(*scheduler*) ja kontrolleriite haldaja(*controller manager*) peavad töötama Kubernetese kobara kaunades.
5. Kubernetese pilveteenustes on võimalik luua koormusjaoturi tüüpi teenus. Selleks, et koolitusel õpitav oleks rakendatav ka organisatsioonides, mis kasutavad töös enda riistvarale ehitatud Kubernetese kobaraid, soovime kasutada *MetaLB* koormusjaoturi tarkvara, et luua keskkonnast sõltumatu koormusjaoturi tüüpi teenus. Tegemist on tarkvaraga, mis ei kasuta pilveteenusepakkuja valmis lahendust või teenust [24]. Seega peab valitud tööriist võimaldama kasutada *MetaLB* tarkvara.
6. Koolituse harjutuste käigus tahetakse luua ja kustutada Kubernetese kobaraid, et proovida erinevaid topoloogiaid. Koolituse keskmes ei ole Kubernetese tarkvara paigaldamine, selle jaoks on juba olemasolev koolitus [12]. Tööriistaga peab kobarate käivitamine ja kustutamine toimuma lühikese aja jooksul, et harjutuste aeg oleks hästi kasutatud. Kubernetese kobara loomine loetakse kiireks, kui see toimub alla viie minuti ja kustutamine alla minuti.
7. Konteinerite kasutamine virtuaalmasinate asemel on oluliselt efektiivsema mälu- ja arvutusressursi kasutusega, kui virtuaalmasinad [25]. Koolitatavatel on konteinerite põhiste lahenduste kasutamine juba varasemas koolituses kogetud [26]. Ühtlasi ei ole virtuaalmasinad koolituses käsitletav teema. Seega tööriista

poolt loodud Kubernetese sõlmed peavad töötama tarkvara konteinerites ühe virtuaalmasina piires. See võimaldab hõlpsamini koolitatavatel õpitud tööriista hiljem oma töös kasutada.

8. On soov vähendada koolituse laborikeskkonna haldusele kuluvat aega. Tööriist peab olema lihtsasti kasutatav ja nõuded peab täitma tööriist või selle dokumentatsioon iseseisvalt, ilma lisaarenduseta Entigo OÜ töötajate poolt.
9. Koolituste laborikeskkonna maksumus mõjutab koolituselt teenitud tulu. Senine kogemus laborikeskkondadega on näidanud, et virtuaalmasinaid tehakse suuremaks mäluksutuse, mitte protsessori võimsuse tõttu. Kobarad peavad olema võimalikult väikese mäluksutusega, et hoida kulud madalal. Mäluksutus loetakse väikeseks, kui mäluksutus jääb alla kolme gigabaidi, sest siis on võimalik 8 gigabaidi mäluuga virtuaalmasinasse käivitada kaks kobarat.

Järgnevalt on toodud kokkuvõtte eelnevas analüüsis defineeritud nõuetest tööriistadele. Tööriistade võrdluses hinnatakse nende vastavust loetletud nõuetele.

1. Luua ja kustutada mitut Kubernetese kobarat.
2. Kubernetese kobaras luua eriotstarbega töötajasõlmi.
3. Luua kolme juhtsõlmega Kubernetese kobar.
4. Kubernetese juhttasandi komponendid töötavad kaunades.
5. Kasutada MetalLB koormusjaoturi tarkvara.
6. Kubernetese kobara loomine ja kustutamine käib kiirelt.
7. Kubernetese sõlmed töötavad virtuaalmasina rakenduskonteinerites.
8. Nõuete täitmine peab olema realiseeritud tööriista enda omadustega.
9. Kubernetese kobarad peavad olema väikese mäluksutusega.

### 3.3 Tööriistade ülevaade

Valikusse on võetud need tööriistad, mida tutvustatakse, kui väikese jalajäljega ja kohalikus tööjaamas Kubernetese käituseks sobivad tööriistad. Tööriistad, mida hakatakse uurima on välja valitud Kubernetese enda dokumentatsioonist, blogipostitustest ja tööriistade võrdlustest teiste tööriistadega. Tööriistad, mida selles töös uuritakse on Minikube, Microk8s, Kind ja K3d. [27] [28] [29].

**Minikube** on mõeldud Kubernetese käivitamiseks ühe masina piires. Välja on toodud peamiste omadustena kõige uuemate Kubernetese versioonide tugi, toetab nii Linuxit, Maci kui Windowsit. Töötab nii rakenduse konteineris, virtuaalmasinas kui raudvara peal [30]. Githubis on projektil 721 panustajat ja see on loodud 2016 aprillis [31].

**K3d** on tööriist, mis omakord kasutab K3s tööriista, eesmärgiga luua Kubernetese kobar rakenduse konteinerites ja võimaldab ühes arvutis käivitada mitut Kubernetese kobarat [32]. K3s on Rancheri poolt pakutav õhuke Kubernetese distributsioon, mis on mõeldud servtööluse jaoks [33]. Projektil on 77 panustajat ja see on loodud 2019 aprillis [34].

**Microk8s** on Canonical poolt arendatav projekt, mis lubab teha Kubernetese kergesti kasutatavaks ja kõrgkäideldavaks. Välja on toodud ka nende tugi Raspberry Pi jaoks, mis viitab väiksele jalajäljele [35]. Ennast võrdlevad nad Minikube ja K3s projektiga [29]. Projektil on 137 panustajat ja see on loodud 2018 mais [36].

**Kind** on mõeldud Kubernetese kobarate käitamiseks ühe arvuti piires. Eelkõige arendatud selleks, et testida Kubernetese kobaraid, aga võimalik kasutada ka tarkvara arenduseks [37]. Projektil on 230 panustajat ja see on loodud 2018 septembris [38].

### 3.4 Tööriistade omaduste mõõtmise tulemused

Mõõtmised teostati 26. Märts 2022 ja tulemused on toodud tabel 2. Mõõtmistel kasutatud infrastruktuuri kood ja skriptid on laetud Github koodihoidlasse [39]. Mälukasutuse erinevused võib lugeda marginaalseteks, sest kõige enam ja vähem tarbiva tööriista mälukasutus erineb natuke üle poole gigabaidi. Seejuures tuleb arvestada, et Minikube puhul ei olnud võimalik luua kolme juhtsõlme, mis võib põhjendada selle tööriista väikseimat mälukasutust võrreldes teiste tööriistadega.

Tabel 2. Tööriistade omaduste mõõtmise tulemused

Omadus	Minikube	Microk8s	K3d	Kind	Ühik
Töötajasõlmi	7	5	5	5	Tükki
Juhtsõlmi	1	3	3	3	Tükki
Tööriista versioon	V1.25.2	1.23	V5.3.0	V0.11.1	
Tööriista paigaldus	1	16	2	1	Sekundit
Kobara loomine	275	1314	83	232	Sekundit
Kobara sõlmed on valmisolekus	9	21	2	30	Sekundit
Kobara kustutamine	15	32	3	4	Sekundit
Kobara loomine teist korda	213	1150	77	184	Sekundit
Kobara sõlmed on valmisolekus teist korda	10	25	0	29	Sekundit
Operatsioonisüsteemis vaba mälu vähenemine pärast kobara loomist	1758	2513	1884	2317	Megabaiti
Esmane kobara loomine kokku	285	1351	87	263	Sekundit
Teise korra kobara loomine kokku	223	1175	77	213	Sekundit

### 3.5 Tööriistade nõuete täitmise analüüs

Kirjeldused uuritud tööriistade kasutamisest ja tekkinud probleemidest. Iga tööriista kohta on tehtud tabel, kus on antud igale nõudele hinne. Hinde põhjendamiseks on vajadusel lisatud kommentaare ja viiteid. Saadud hinned on kokku liidetud tabeli lõpus.

#### 3.5.1 Minikube

Käesoleva töö autori hinnangul oli tööriista lihtne kasutada. Kobara loomiseks piisas ühest käsust, vt. Lisa 3. Töö käigus selgus viga dokumentatsioonis. Dokumentatsioonis mainitud *node add* käsu lipud ei tööta. Viga on raporteeritud projekti veahaldussüsteemis Github [40]. Hetkel puudu olevate nõuete täitmine paraneb ilmselt tulevikus, sest mitme juhtsõlme ja eriotstarbega töösõlmede toetamine on neil tulevikuplaanides [41]. Tegemist on kõige suurema panustajate arvuga ja vanima projektiga, mida töös analüüsitakse. Nõuete täitmise analüüs on tabelis 3.

Tabel 3. Minikube nõuete täitmine

Nõue	Hinne	Kommentaari
Luu ja kustutada mitut Kubernetese kobarat	3	Selleks tuleb kasutada <i>--profile</i> lipukest [42].
Kubernetese kobaras luua eriotstarbega töötajasõlmi	1	Saab määrata mitu töötajasõlme, aga neile ei saa tööriistas määrata eriotstarvet [43].
Luu kolme juhtsõlmega Kubernetese kobar	0	Dokumentatsiooni järgi on võimalik lisada kasutades käsku <i>minikube node add</i> koos parameetriga <i>--control-plane</i> [44]. Githubi probleemipüstituse järgi see ei tööta [40] [41].
Kubernetese juhttasandi komponendid töötavad kaunades	3	Jah, seda kinnitas kobara omaduste mõõtmiskriipti väljund.
Kasutada MetalLB koormusjaoturi tarkvara	3	Kasutades tööriista omadusi [45].



Kubernetesi kobarade loomine ja kustutamine käib kiirelt	3	Loomiseks kulus alla viie minuti ja kustutamiseks alla minuti, vt. Tabel 2.
Kubernetesi sõlmed töötavad virtuaalmasina rakenduskonteinerites	3	Kasutab Docker tarkvara konteinereid, vt Lisa 3.
Nõuete täitmine peab olema tööriista enda omadus	0	Puuduseks on eriotstarbega töötajasõlmede loomine ja kolmest sõlmest koosnev Kubernetesi juhttasand. Viimast nõuet ei saa täita tööriista väliselt.
Kubernetesi kobarad peavad olema väikese mälukasutusega	3	Kobar vajab alla 3 GB mälu, vt. Tabel 2.
<b>Punkte kokku</b>	<b>19</b>	

### 3.5.2 K3d

Käesoleva töö autori hinnangul oli tööriista lihtne kasutada. Kobarade loomiseks ja kustutamiseks piisas ühest käsust ja seadistusfailist, vt. Lisa 5. Antud uurimuses esitatud nõudeid ei saa see tööriist täielikult täita, sest põhineb K3s tööriistal. K3s arhitektuuri puhul ei tööta Kubernetesi juhttasandi komponendid kaunades [46]. Arvesse tuleks võtta ka seda, et tegemist on kõige uuema ja kõige väiksema arendajate arvuga tööriistaga, mida siin uurimuses uuritakse. Nõuete täitmise analüüs on tabelis 5.

Tabel 4. K3d nõuete täitmine

Nõue	Hinne	Kommentaar
Luu ja kustutada mitut Kubernetesi kobarat	3	Selleks tuleb määrata kobarale nimi [47].

Kubernetes kobaras luua eriotstarbega töötajasõlmi	3	Kasutades tööriista sõlme seadistuses K3d parameetrit koos <i>nodeLabels</i> ja <i>extraArgs</i> lipukestega [48].
Luu kolme juhtsõlme K Kubernetes kobar	3	Tööriista seadistuses määrata <i>servers</i> parameetrile juhtsõlmede arv [49].
Kubernetes juhttasandi komponendid töötavad kaunades	0	Ei tööta kaunades ja seda kinnitas kobara omaduste mõõtmiskriipti väljund. K3B on erinev süsteem tavapärasest Kubernetesest. Kubernetes juhttasandi komponendid ei ole <i>kubectl</i> käsuga ligipääsetavad ning ei tööta kaunades [46].
Kasutada MetalLB koormusjaoturi tarkvara	2	Ei ole tööriista enda dokumentatsiooni osa, aga on võimalik tööle panna kolmanda osapoole juhendi abil [50].
Kubernetes kobara loomine ja kustutamine käib kiirelt	3	Loomiseks kulus alla viie minuti ja kustutamiseks alla minuti, vt. Tabel 2.
Kubernetes sõlmed töötavad virtuaalmasina rakenduskonteinerites	3	Kasutab Docker tarkvara konteinereid [32].
Nõuete täitmine peab olema tööriista enda omadus	2	MetalLB koormusjaoturi tuge tööriist ise ei oma. Seda on võimalik siiski realiseerida tööriistaväliselt.
Kubernetes kobarad peavad olema väikese mälukasutusega	3	Kobar vajab alla 3 GB mälu, vt. Tabel 2.
<b>Punkte kokku</b>	<b>22</b>	

### 3.5.3 Microk8s

Käesoleva töö autori hinnangul selle tööriistaga tarkvara konteinerites Kubernetese kobara loomine ei olnud lihtne, selleks oli vajalik sisestada 25 käsku, mis teiste tööriistadega võrreldes on 25 korda rohkem, vt. Lisa 4. Tuli kombineerida kahte erinevat dokumenti, et saavutada soovitud tulemus [51] [52]. See viitab sellele, et tarkvara konteinerites Kubernetese kobarate käitus ei ole selle tööriista eesmärk. Autori hinnangul on sellest tingitud suur osa teisi tekkinud puudujääke. Nende nõuete paremaks täitmiseks võiks kasuks tulla sarnase projekti teke, nagu on K3s jaoks loodud K3d. Nõuete täitmise analüüs on tabelis 4.

Tabel 5. Microk8s nõuete täitmine

Nõue	Hinne	Kommentaar
Luu ja kustutada mitut Kubernetese kobarat	0	Dokumentatsioonist ei ole sellist võimalust [51]. Ei ole tööriista enda omadus. Selleks tuleks tekitada tööriista väliselt eraldi tarkvara konteinerid ja moodustada Kubernetese kobarad.
Luu kolme juhtsõlmega Kubernetese kobar	0	Dokumentatsioonist ei ole ühtegi lisa parameetrit mida saaks anda liitumisel. Iga arvutusõlm tuleb eraldi lisada [51].
Luu Kubernetese kobar, milles on kolm juhtsõlme	3	Sõlme lisamisel saab määrata juhtsõlmeks [53].
Kubernetese juhttasandi komponendid töötavad kaunades	0	Ei tööta kaunades ja seda kinnitas kobara omaduste mõõtmiskriipti väljund.
Kasutada MetalLB koormusjaoturi tarkvara	3	On tööriista enda omadus [54].
Kubernetese kobara loomine ja kustutamine käib kiirelt	1	Loomiseks kulus üle viie minuti ja kustutamiseks alla minuti, vt. Tabel 2.

Kubernetesi sõlmed töötavad virtuaalmasina rakenduskonteinerites	1	Dokumentatsioonis on kirjeldatud LXC ( <i>Linux Containers</i> ) kasutamine, aga nende rakenduskonteinerite tekitamine ei ole tööriista enda omadus. Dokumentatsiooni näide ei sisalda mitme töötajasõlme kasutust [52].
Nõuete täitmine peab olema tööriista enda omadus	0	Enamikes punktides on tööriistal puudused, mis on tingitud tööriista enda võimetusest.
Kubernetesi kobarad peavad olema väikese mälu kasutusega	3	Kobar vajab alla 3 GB mälu, vt. Tabel 2.
<b>Punkte kokku</b>	<b>11</b>	

### 3.5.4 Kind

Käesoleva töö autori hinnangul oli tööriista lihtne kasutada. Kobara loomiseks ja kustutamiseks piisas ühest käsust ja seadistusfailist, vt Lisa 6. Antud uurimistöö nõudeid täitis see tööriist kõige paremini. Üks punkt kaotati MetalLB nõude juures, sest mitmel teisel tööriistal on selle aktiveerimiseks käsk. Samas on antud tööriista dokumentatsioonis detailsed juhised, kuidas MetalLB on võimalik kasutusele võtta. Nõuete täitmise analüüs on tabelis 6.

Tabel 6. Kind nõuete täitmine

Nõue	Hinne	Kommentaari
Luu ja kustutada mitu Kubernetesi kobarat	3	Tuleb kasutada lipukest <i>--name</i> [55].
Kubernetesi kobaras luua eriotstarbega töötajasõlmi	3	Kasutades tööriista sõlme seadistuses <i>Kubeadm Config Patches</i> parameetrit koos <i>node-labels</i> ja <i>register-with-taints</i> lipukestega [56].

Luua kolme juhtsõlmega Kubernetese kobar	3	Kasutades tööriista sõlme seadistuses sõlme rolli <i>control-plane</i> [57].
Kubernetese juhttasandi komponendid töötavad kaunades	3	Jah, seda kinnitas kobara omaduste mõõtmiskriipti väljund.
Kasutada MetalLB koormusjaoturi tarkvara	3	Tööriista dokumentatsiooni järgi saab [58].
Kubernetese kobara loomine ja kustutamine käib kiirelt	3	Loomiseks kulus alla viie minuti ja kustutamiseks alla minuti, vt. Tabel 2.
Kubernetese sõlmed töötavad virtuaalmasina rakenduskonteinerites	3	Kasutab Docker tarkvara konteinereid [37].
Nõuete täitmine peab olema tööriista enda omadus	2	MetalLB kasutamine ei käi läbi tööriista, aga tööriista dokumentatsioon kirjeldab detailset seadistust.
Kubernetese kobarad peavad olema väikese mälukasutusega	3	Kobar vajab alla 3 GB mälu, vt. Tabel 2.
<b>Punkte kokku</b>	<b>26</b>	

### 3.6 Tööriistade võrdluse tulemused

Hindamise hetkel olemas olnud funktsionaalsuse ja mõõtmise tulemusel osutus kõige sobivamaks tööriistaks Kind. Tuleb märkida, et teised tööriistad on aktiivses arenduses, mille tulemusena võib nende sobivus tulevikus muutuda. Käesolevas töös realiseeritakse praktiline lahendus analüüsis parima tulemuse saanud Kind tööriistaga.

Kuigi teistel tööriistadel on tugevaid külgi, siis ülejäänud kolmel tööriistal on kõigil mõni nõue, mis ei ole üldse täidetud.

Tööriist k3d on väga võimekas tööriist, kui koolitusel ei oleks oluline Kubernetese juhttasandi komponentide ligipääs, siis võiks see tööriist olla ka väga hea valik. MetalLB

toe puudumine tööriista poolt on väiksem probleem, sest tehniliselt on võimalik seda tarkvara kasutada [50]. Peale nende kahe puuduse ei ole tööriistal teisi puudusi.

Minikube tööriistal on kindlasti tugevaks küljeks väga lihtne kasutamine. Võib oletada, et tulevikus parandatakse selle tööriista suurim puudus, milleks on mitme juhtsõlme toe puudumine.

Microk8s puhul sai üheks suurimaks takistuseks aeglus. Tööriista dokumentatsioonile tuginedes LXC konteinerites Snapcraft pakihalduri kasutamine tegi kobarate paigalduse aeglaseks. Dokumentatsioonis ei olnud ka välja toodud, kuidas käitada mitme sõlmega kobarat ühes virtuaalmasinas.

## 4 Laborikeskkonna loomine

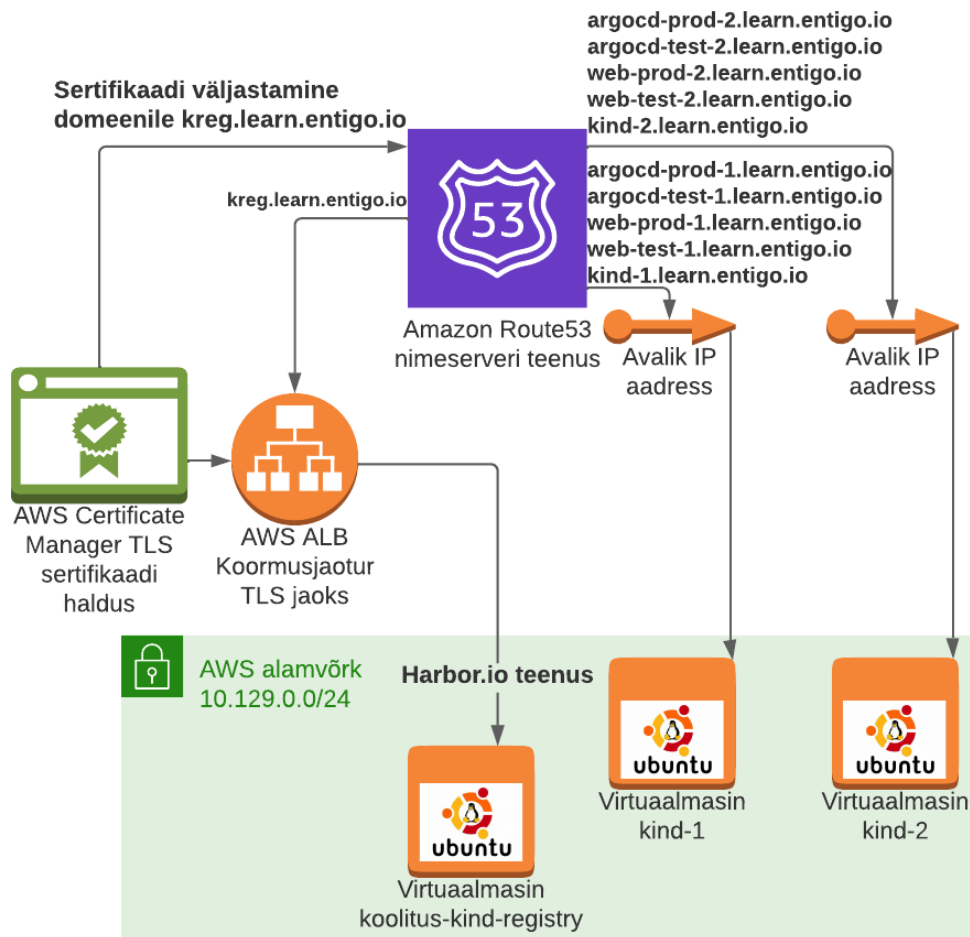
Kogu laborikeskkonna loomise kood laetakse Github koodihoidlasse [59]. Tööriist pannakse käima AWS EC2 virtuaalmasina tüübil t3.large, millel on 8GB mälu ja 2 protsessori tuuma [9]. Selle virtuaalmasina ööpäeva maksumus Frankfurdis on 2.3 USD [9]. Virtuaalmasinasse luuakse kaks Kubernetese kobarat, ühe kobara maksumus 24 tunni kohta on 1.15 USD. Mõlemas kobaras saab olema kolm juhtsõlme, kaks eriotstarbega arvutusõlme ja kolm üldkasutatavat arvutusõlme. Kui tööriist on kobarad loonud, siis nende sees olevaid objekte hallatakse Argo CD tarkvara abil, mis ise töötab kobara juhtsõlmedel. Eriotstarbega sõlmed teenindavad koormusjaoturi tarkvara MetallB ja HAProxy. Üldkasutatavatel töötajasõlmedel pannakse tööle Entigo näidisrakendus, mida kasutatakse ka teistes koolitustes. Näidisrakendus ja Argo CD on ligipääsetavad koolitatava veebibrauseri kaudu tänu MetallB ja HAProxy tarkvara kooslusele. Testitakse sõlmede tõrkesiirdeid praktikas.

### 4.1 Virtuaalmasinate loomine ja tarkvara konteinerite puhverserver

Osaliselt taaskasutati eelnevates koolitustes kasutusel olnud Terraformi infrastruktuuri loomise kood. Selles on funktsionaalsus, mis loob konteinerite registri ja koolitusel osalejatele virtuaalmasinad AWS EC2 teenuseid kasutades [60]. Konteinerite registri loomiseks kasutatakse Docker tarkvara konteinereid ja Harbor tarkvara. Konteinerite registri teenuse TLS jaoks kasutatakse AWS Application Loadbalancer (edaspidi AWS ALB) ja AWS Certificate Manager (edaspidi AWS CM) teenust. TLS sertifikaat valideeritakse nimeserveri kande alusel, mis on loodud AWS Route53 teenust kasutades. Konteinerite register on tööle pandud puhverserverina hub.docker.com suunal, et avaliku registri allalaadimise piiranguid mitte ületada [61]. Registri paigaldus ja seadistus on automatiseeritud ja teostatakse virtuaalmasina loomisel [62].

Koolitusel osalejate arvu saab määrata Terraformi muutuja *students* abil [60]. Antud töös seadistati kaks osalejat. Osalejad nummerdatakse alates number ühest kuni maksimaalse osalejate arvuni ja iga osaleja saab individuaalse numbri. Igal virtuaalmasinal on avalik IP-aadress, millele on loodud nimeserveris vajalikud nimed kasutades AWS Route53 teenust. Igale virtuaalmasina avalikule IP-aadressile on loodud mitu nimeserveri kannet, et oleks võimalik nime alusel eristada, millise teenusega soovitakse suhelda. Nimeserveri

kanded sisaldavad endas koolitatava individuaalset numbrit. Tulemürist on lubatud kõik ühendused avalikust internetist. Olulisemad AWS teenused on illustreeritud joonisel 1. Ekraanitõmmised loodud virtuaalmasinatest on joonisel 6, tarkvara konteinerite registri Harbor kasutajaliidesest joonisel 7 ja nimeserveri kanded AWS Route53 teenuses joonisel 8.

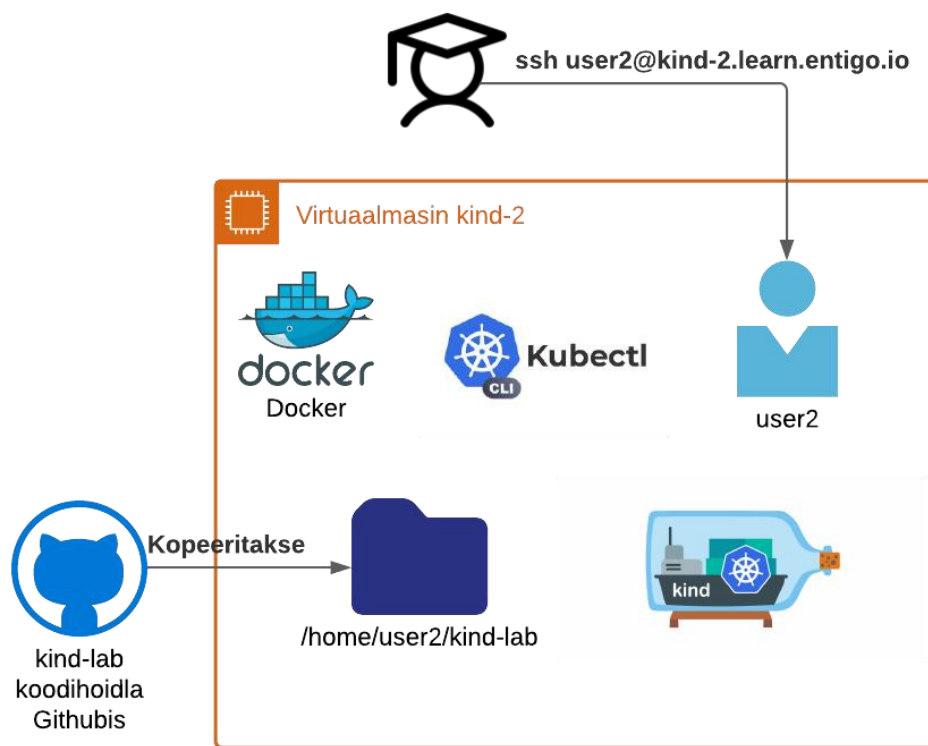


Joonis 1. Terraformiga loodud komponentide skeem

Igasse virtuaalmasinasse luuakse automaatselt kasutaja koos parooliga, mis sisaldab koolitatava individuaalset numbrit. Igale koolitatavale loodud virtuaalmasinasse paigaldatakse automaatselt Kind tööriist, Docker tarkvara ja kobarate seadistamiseks vajalikud failid laetakse alla Github koodihoidlast kasutaja kodukausta [63]. Koolitatav saab ühenduda talle loodud virtuaalmasinasse SSH terminali kliendi abil. Virtuaalmasina



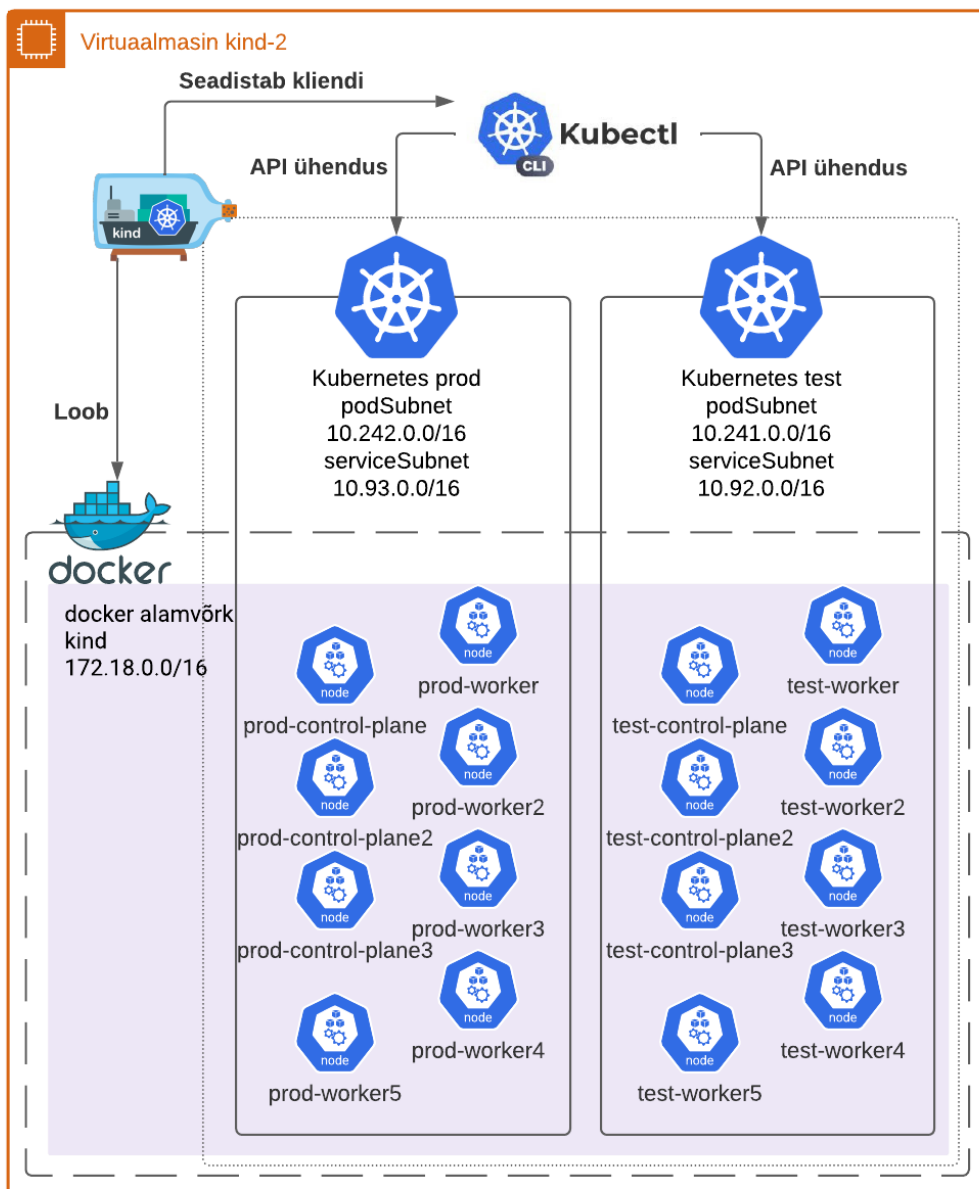
seadistus on illustreeritud joonisel 2. Ekraanitõmmised virtuaalmasinasse sisenemisest ja seal olevatest failidest on joonisel 9.



Joonis 2. Koolitusel osaleja virtuaalmasinasse paigaldatud komponendid ja ligipääs

## 4.2 Kubernetes kobarate loomine Kind tööriistaga

Virtuaalmasinas kasutatakse skripti, mis esimeses osas kutsub välja Kind tööriista ja loob kaks kobarat [64]. Kummagi kobara jaoks on loodud seadistusfailid, mille omavaheline erinevus seisneb kobara nime vahetamises, *podSubnet* ja *serviceSubnet* väärtustamises erinevate alamvõrkudega. Sellises seadistus ei teki konflikti võrgu või nime kasutusel. Seadistatud on konteineri registri puhverserveri kasutamine ja kaks eriotstarbega sõlme määrates neile *label* ja *taint* [65] [66] [22]. Tööriistaga loodavate Kubernetes kobarate arhitektuur on illustreeritud joonisel 3. Ekraanitõmmis Kubernetes kobarate loomisest on joonisel 10.



Joonis 3. Virtuaalmasinasse tööriistaga loodav kobarate arhitektuur

Kubernetesi klienti *kubectl* kasutades kontrollitakse loodud kobarate vastavust nõuetele. Kasutades *--context* lipukest saab suhelda mõlema kobaraga, seega mitme kobara loomine õnnestus, vt joonis 11. Leiti, et loodud kolm juhtsõlme ja nende peal töötavad Kubernetesi juhttasandi komponendid kaunades, vt joonis 12. Loodud kobara sõlmed töötavad konteinerites, vt joonis 13. Kubernetesi kobarate kustutamiseks on loodud skript [67].

Kobarate loomine õnnestus t3.large virtuaalmasina tüübiga, aga selleks et järgmises sammus paigaldada rakendused oli vaja suurendada t3.xlarge virtuaalmasina tüübile, millel on 16GB mälu ja 4 protsessori tuuma [9]. Vastasel juhul jäi puudu nii mälust, kui protsessori tuumadest. Selle virtuaalmasina ööpäeva maksumus Frankfurdis on 4.6 USD [9]. Seega ühe kobara maksumus 24 tunni kohta on 2.3 USD. Tuleb silmas pidada, et varasemates laborikeskkondade maksumuse arvutuses ei ole arvestatud rakenduste käivitamisega vaid ainult kobarate loomiseks vajalike miinimum nõuetega, seega tuleks laborikeskkondade maksumuse võrdlemisel võtta aluseks t3.large tüüpi virtuaalmasina maksumus, mis on kirjeldatud 4. peatüki alguses.

### 4.3 Entigo näidisrakenduse paigaldus

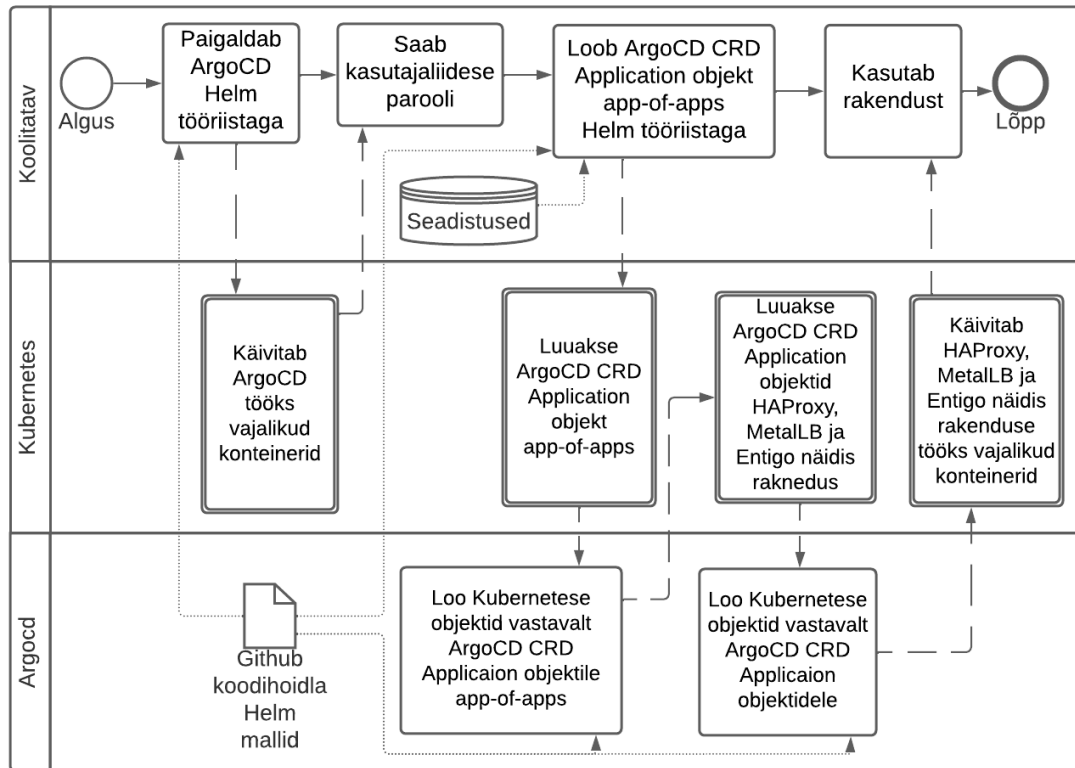
Antud töö eesmärk ei ole uurida või põhjendada Entigo OÜ poolt kasutatavaid praktikaid, seega kirjeldatakse lühidalt kuidas näidisrakendus paigaldati. Pärast kobarate loomist paigaldab skript Argo CD tarkvara mõlemasse kobarasse kasutades Helm pakihaldurit ja Github koodihoidlast kasutaja kodukasuta laetud Helm malle [64]. Skript ootab kuni Argo CD tarkvara on käivitatud ja kuvab ekraanile mõlema kobara Argo CD haldusliidese administraatori parooli.

Seejärel luuakse mõlemasse kobarasse Argo CD *Custom Resource Definition* (edaspidi CRD) *Applications* tüüpi objekt Helm pakihalduri abil. Objekti sees on Helm malli seadistused ja viide Githubi koodihoidla kausta *argocd-applications* [68]. Seadistuses määratakse rakendustele vajalikud parameetrid, milleks on koolitatava individuaalne number ja kobara nimetus. Kobara nimetused on muutujas *runenv* ja väärtused on vastavalt *test* või *prod*. Koolitatava individuaalne number on muutujas *number*.

Argo CD tarkvara loeb Argo CD CRD *Applications* objektis seadistatud koodihoidlast Helm tarkvara malle ja eelnevalt paika pandud seadistusi, et luua järgmisi Kubernetese objekte. Selle tulemusena luuakse järgmised Argo CD CRD *Applications* objektid [69]. Nendele objektidele pärandatakse edasi seadistused. Iga loodud objekt vastab ühele rakendusele ja igal rakendusel on *argocd-yaml* alamkaustades oma Helm mallid.

Nende objektide seadistuse abil laeb Argo CD omakorda rakenduste Helm mallid teistest Github koodihoidla kaustadest ja loob rakenduste käitamiseks vajalikud Kubernetese objektid. Eelkirjeldatud viisil luuakse MetalLB, HAProxy ja Entigo näidisrakendus [70].

Antud protsessi kujutamiseks kasutati *Business Process Mapping and Notation 2.0* notatsiooni, mis on näha joonisel 4 [71].



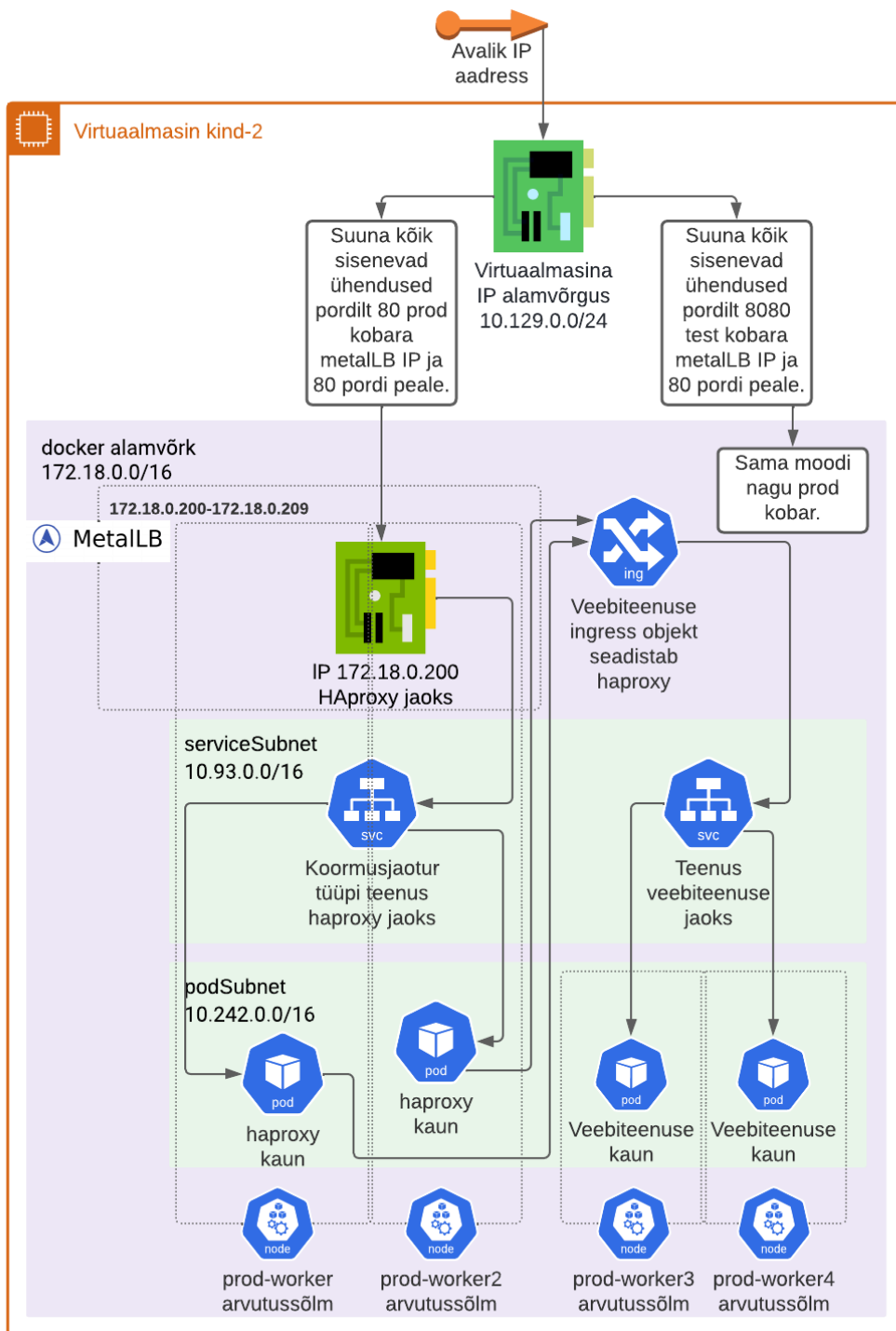
Joonis 4. Rakenduste loomine Argo CD tööriistaga

Entigo näidisrakendus koosneb kolmest komponendist. Redis andmebaas, mida kasutatakse kasutajate sessioonide talletamiseks. API mis vastutab rakenduse dünaamilise sisu eest, sealhulgas ka sessioonide talletamisega andmebaasi. Veebiliides, mis kuvab kasutajasõbralikult API vastuseid. Ekraanitõmmis loodud objektides on näha joonisel 14. Joonisel on näha ka *test* ja *prod* kobarate *Ingress* (Kubernetes objekt ühtsete ressursilokaatorite (*Uniform Resource Locator*) vastavusse seadmiseks teenustega) objektide erinevus, mis on saavutatud tänu Helm mallide ja Argo CD CRD *Application* objektide eelnevalt kirjeldatud seadistusele.

MetalLB ülesanne on võimaldada Kubernetes koormusjaoturi tüüpi teenuste objekte kasutada. MetalLB tarkvara pannakse käima kõrgkäideldavas seadistuses kahe eriotstarbega töötajasõlme peal. *Prod* ja *test* kobaratele eraldatakse erinevad IP vahemikud Docker'i alamvõrgust [72]. Nendest vahemikest antakse koormusjaoturi tüüpi teenustele IP-aadresse, mille poole on võimalik pöörduda ka väljastpoolt Kubernetese

kobarat, aga mitte internetist. Mõlema kobara internetist kättesaadavaks tegemiseks suunatakse skriptis *iptables* käsu abil virtuaalmasinasse internetist tulevad ühendused kummagi MetalLB poolt eraldatud HAProxy koormusjaotur tüüpi teenuse IP-aadressile. *Prod* kobara puhul TCP 80 port ja *test* kobara puhul TCP 8080 port. [64]. Koormusjaotur tüüpi teenuste ja ühenduste suunamise ekraanitõmmis on joonisel 15.

HAProxy kontrolleri ülesanne on tõlgendada Kubernetese *Ingress* tüüpi objektid HAProxy seadistusteks, et tekitada ühtse ressursilokaatori väärtuste järgi ligipääsu reegleid erinevatele Kubernetese kobara teenustele [73]. Antud juhul on kolm *Ingressi* objekti kasutatavat teenust: Argo CD, Entigo näidisrakenduse veebiliides ja Entigo näidisrakenduse API. HAProxy ise kasutab koormusjaoturi tüüpi teenust, et selle poole saaks pöörduda väljastpoolt kobarat. Kogu eelnevalt kirjeldatud ahel on illustreeritud joonisel 5. Koolitatavatel on võimalik *test* kobara Entigo näidisrakendust avada veebibrauseris <http://web-test-2.learn.entigo.io:8080/> aadressil, vt joonis 16 ja *prod* kobara oma avada veebibrauseris <http://web-prod-2.learn.entigo.io:80/> aadressil, vt joonis 17. Samamoodi on võimalik ka avada Argo CD veebiliidest <http://argocd-test-2.learn.entigo.io:8080/> ja <http://argocd-prod-2.learn.entigo.io:80/> aadressidelt.



Joonis 5. Teenustele ligipääsu tekitamine avalikust internetist

#### 4.4 Tõrkesiirde test

Mitme juht- ja töötajasõlme nõuded põhinevad tõrkesiirde ja kõrgkäideldavuse harjutamise eesmärgil. Selleks, et veenduda tarkvara konteinerites töötavate Kubernetese

sõlmede, juhttasandi ja muude objektide tõrkesiirde toimimises teostati test. Testi käigus pandi seisma üks juhtsõlm, üks eriotstarbega töötajasõlm ja üks üldkasutatav töötajasõlm. Kobara sõlmede ja Entigo näidisrakenduse kaunade ekraanitõmmised enne katkestust on joonis 18 ja 19.

Pärast kobara sõlmede seiskamist oli kobara juhttasand töötav, sest Kubernetese klient oli kasutatav. See tähendab et Kubernetese juhttasandi API server komponent töötab. Entigo näidisrakendus jätkuvalt avanes veebibrauseris, mis kinnitab, et HAProxy ja MetalLB töötavad. Kubernetese kobar reageeris katkestusele oodatult. Peatatud sõlmede seisund muutus *Not Ready* (mitte valmis) staatusesse, vt joonis 20. Entigo näidisrakenduse kaunad, mis olid peatatud sõlmel on pandud tööle teistel sõlmedel, vt joonis 21. Viimane kinnitab ka Kubernetese juhttasandi planeerija ja kontrolleriite haldaja töötamist.

## 4.5 Järeldused

Tänu valitud tööriistale õnnestus luua Kubernetese kobarad neli korda odavamalt, kui „Kubernetese põhialused 2“ laborikeskkonnas kasutatud vahenditega. Tuleb arvestada, et koolitustel osaleb kümme või enam koolitatavat, koolitus kestab mitu päeva ja planeeritavas koolituses hakatakse kasutama mitmeid kobaraid, vt Tabel 7. Seega hoitakse koolitusel kokku märkimisväärne summa.

Tabel 7. Laborikeskkondade Kubernetese kobarate maksumused

Keskkond	Kubernetese põhialused 2	Antud töös loodud laborikeskkond
Ööpäevane virtuaalmasinate kulu ühe Kubernetese kobara käitamiseks.	4.61 USD	1.15 USD
Kümne osalejaga kolme päevase koolituse virtuaalmasinate maksumus, kus igale osalejale on tagatud nelja Kubernetese kobara käitamise võimekus.	553.2 USD	138 USD

Tööriistaga oli võimalik rakendada Entigo OÜ-s laialt levinud praktikaid ja paigaldada Entigo näidisrakendus. Tõrkesiirde ja kõrgkäideldavuse test oli edukas. Laborikeskkonna

kasutamiseks on osalejatel vajalikud samad vahendid, mis varasemate koolituste laborikeskkondades. Tööriistale püstitatud nõuete täitmine leidis praktilise kinnituse, vt tabel 8.

Tabel 8. Kind tööriista nõuete täitmine praktikas

Nõue	Kommentaar
Luu ja kustutada mitut Kubernetese kobarat	Täidetud, vt. peatükk 4.1
Kubernetese kobaras luua eriotstarbega töötajasõlmi	Täidetud, vt. peatükk 4.2
Luu kolme juhtsõlmeiga Kubernetese kobar	Täidetud, vt. peatükk 4.2
Kubernetese juhttasandi komponendid töötavad kaunades	Täidetud, vt. peatükk 4.2
Kasutada MetalLB koormusjaoturi tarkvara	Täidetud, vt. peatükk 4.3
Kubernetese kobara loomine ja kustutamine käib kiirelt	Täidetud, vt. tabel 2
Kubernetese sõlmed töötavad virtuaalmasina rakenduskonteinerites	Täidetud, vt. peatükk 4.2
Nõuete täitmine peab olema tööriista enda omadus	Täidetud, vt. tabel 2
Kubernetese kobarad peavad olema väikese mälukasutusega	Täidetud, vt. tabel 2



## 5 Kokkuvõte

Käesolev töö võrdles Kubernetese kobarate käitustööriistu. Töö eesmärgiks oli välja valida sobivaim tööriist koolituse laborikeskkonna kobarate halduseks. Võrdluseks püstitati mitmeid nõudeid, mis tulenesid koolituse vajadustest. Seejärel hinnati nelja tööriista sobivust nendele nõuetele ja valiti kõige rohkem nõudeid täitev tööriist, milleks oli Kind.

Realiseeriti Kind tööriistaga laborikeskkond. Laborikeskkonda paigaldati Entigo näidisrakendus kasutades asutuses levinud praktikaid. Loodud laborikeskkonnas täitis tööriist püstitatud nõuded ka praktikas. Laborikeskkonnas kobarate käitamise maksumus tuli neli korda soodsam Kind tööriistaga, kui „Kubernetese põhialused 2“ kasutatavate vahenditega. Koolitusel osalejate arvutitele esitatavad nõuded jäid samaks, mis varasemates koolitustes.

Praktilises osas loodud keskkonna alusel saab koolituse arendusega edasi minna ja luua harjutusi planeeritava koolituse tarbeks.

## Kasutatud kirjandus

- [1] The Kubernetes Authors, „Kubernetes Components,“ 21 Veebruar 2022. [Võrgumaterjal]. Available: <https://kubernetes.io/docs/concepts/overview/components/>. [Kasutatud 12 Märts 2022].
- [2] Cybernetica, „ANDMEKAITSE JA INFOTURBE LEKSIKON: control plane,“ [Võrgumaterjal]. Available: <https://akit.cyber.ee/term/5079-control-plane>. [Kasutatud 12 Märts 2022].
- [3] Canonical Ltd., „About Snaps,“ 2022. [Võrgumaterjal]. Available: <https://snapcraft.io/about>. [Kasutatud 21 Aprill 2022].
- [4] The Linux Foundation, „Certified Kubernetes Administrator (CKA),“ 2022. [Võrgumaterjal]. Available: <https://training.linuxfoundation.org/certification/certified-kubernetes-administrator-cka/>. [Kasutatud 13 Märts 2022].
- [5] The Linux Foundation, „Certified Kubernetes Application Developer (CKAD),“ 2022. [Võrgumaterjal]. Available: <https://training.linuxfoundation.org/certification/certified-kubernetes-application-developer-ckad/>. [Kasutatud 14 Märts 2022].
- [6] Cloud Native Computing Foundation, „Certified Kubernetes Administrator (CKA) Exam Curriculum,“ 2022. [Võrgumaterjal]. Available: [https://github.com/cncf/curriculum/blob/master/CKA\\_Curriculum\\_v1.22.pdf](https://github.com/cncf/curriculum/blob/master/CKA_Curriculum_v1.22.pdf). [Kasutatud 12 Märts 2022].
- [7] Cloud Native Computing Foundation, „Certified Kubernetes Application Developer Exam Curriculum,“ 2022. [Võrgumaterjal]. Available: [https://github.com/cncf/curriculum/blob/master/CKAD\\_Curriculum\\_v1.23.pdf](https://github.com/cncf/curriculum/blob/master/CKAD_Curriculum_v1.23.pdf). [Kasutatud 17 Märts 2022].
- [8] Harbor Authors, „Harbor,“ 2022. [Võrgumaterjal]. Available: <https://goharbor.io/>. [Kasutatud 12 Märts 2022].
- [9] Amazon Web Services, Inc., „Amazon EC2 On-Demand Pricing,“ 2022. [Võrgumaterjal]. Available: <https://aws.amazon.com/ec2/pricing/on-demand/>. [Kasutatud 12 Märts 2022].
- [10] J. Barr, „M5 – The Next Generation of General-Purpose EC2 Instances,“ 2022. [Võrgumaterjal]. Available: <https://aws.amazon.com/blogs/aws/m5-the-next-generation-of-general-purpose-ec2-instances/>. [Kasutatud 3 Aprill 2022].
- [11] Entigo OÜ, „Kubernetes põhialused 1,“ 2022. [Võrgumaterjal]. Available: <https://www.entigo.com/et/koolitused/kubernetes-pohialused-1/>. [Kasutatud 19 Märts 2022].
- [12] Entigo OÜ, „Kubernetes põhialused 2,“ 2022. [Võrgumaterjal]. Available: <https://www.entigo.com/et/koolitused/kubernetes-pohialused-2/>. [Kasutatud 19 Märts 2022].

- [13] K. Chandak, „Reduction in EKS cluster creation time,“ 2022. [Võrgumaterjal]. Available: <https://github.com/aws/containers-roadmap/issues/1227>. [Kasutatud 19 Märts 2022].
- [14] Amazon Web Services, Inc., „What is Amazon EKS?,“ 2022. [Võrgumaterjal]. Available: <https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html>. [Kasutatud 12 Märts 2022].
- [15] Amazon Web Services, Inc., „Amazon EKS clusters,“ 2022. [Võrgumaterjal]. Available: <https://docs.aws.amazon.com/eks/latest/userguide/clusters.html>. [Kasutatud 19 Märts 2022].
- [16] The MetalLB Contributors, „Cloud Compability,“ The Linux Foundation, 2022. [Võrgumaterjal]. Available: <https://metallb.universe.tf/installation/clouds/>. [Kasutatud 19 Märts 2022].
- [17] The Linux Foundation, „Kubernetes Documentation Setup tools Kubeadm,“ 2022. [Võrgumaterjal]. Available: <https://kubernetes.io/docs/reference/setup-tools/kubeadm/>. [Kasutatud 18 Aprill 2022].
- [18] The Kubernetes Authors, „Installing kubeadm,“ The Linux Foundation, 2022. [Võrgumaterjal]. Available: <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/#before-you-begin>. [Kasutatud 20 Märts 2022].
- [19] Anonüümne, „Entigo Training Evaluation Form 7. Juuni 2021,“ Entigo OÜ, Tallinn, 2021.
- [20] Entigo OÜ, „Fleetcomplete infrastructure as code repository,“ 2021. [Võrgumaterjal]. Available: [https://<salajane>/DevOps/PlatformOne/P1-Core/-/blob/master/ng\\_monitoring.tf#L5](https://<salajane>/DevOps/PlatformOne/P1-Core/-/blob/master/ng_monitoring.tf#L5). [Kasutatud 20 Märts 2022].
- [21] Entigo OÜ, „Akzonobel infrastructure as code repository,“ 2021. [Võrgumaterjal]. Available: <https://<salajane>/devops/azure-ansible/blob/master/playbooks/os/roles/kube-taint/tasks/main.yml>. [Kasutatud 20 Märts 2022].
- [22] The Kubernetes Authors, „Assigning Pods to Nodes,“ 2022. [Võrgumaterjal]. Available: <https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/>. [Kasutatud 22 Aprill 2022].
- [23] The Kubernetes Authors, „Options for Highly Available Topology,“ 17 Jaanuar 2022. [Võrgumaterjal]. Available: <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/ha-topology/>. [Kasutatud 20 Märts 2022].
- [24] The MetalLB Contributors, „MetalLB,“ 2021. [Võrgumaterjal]. Available: <https://metallb.universe.tf/>. [Kasutatud 10 Aprill 2022].
- [25] S. Proietti Conti ja R. Rogers, „Virtual Machines versus containers,“ IBM IT Economics Consulting & Research, Veebruar 2021. [Võrgumaterjal]. Available: <https://www.ibm.com/downloads/cas/POANK8YE>. [Kasutatud 9 Aprill 2022].
- [26] Entigo OÜ, „Rakenduse konteinerite koolitus,“ 2020. [Võrgumaterjal]. Available: <https://www.entigo.com/et/koolitused/rakenduse-konteinerid/>. [Kasutatud 20 Märts 2022].
- [27] The Kubernetes Authors, „Kubernetes Documentation Install Tools,“ The Linux Foundation, 2022. [Võrgumaterjal]. Available: <https://kubernetes.io/docs/tasks/tools/>. [Kasutatud 12 Märts 2022].

- [28] A. E. Amri, „k3d vs. k3s vs. Kind vs. microk8s vs. minikube,“ 2022. [Võrgumaterjal]. Available: <https://faun.pub/k3d-vs-k3s-vs-kind-vs-microk8s-vs-minikube-6949ebb93d18>. [Kasutatud 13 Märts 2022].
- [29] Canonical Ltd, „MicroK8s vs K3s vs minikube,“ 2022. [Võrgumaterjal]. Available: <https://microk8s.io/compare>. [Kasutatud 14 Märts 2022].
- [30] The Kubernetes Authors, „Minikube Documentation Welcome,“ Veebruar 2022. [Võrgumaterjal]. Available: <https://minikube.sigs.k8s.io/docs/>. [Kasutatud 21 Märts 2022].
- [31] The Kubernetes Authors, „Github kubernetes/minikube,“ 2022. [Võrgumaterjal]. Available: <https://github.com/kubernetes/minikube>. [Kasutatud 21 Märts 2022].
- [32] k3d Authors, „What is k3d?,“ 2022. [Võrgumaterjal]. Available: <https://k3d.io/v5.3.0/#what-is-k3d>. [Kasutatud 12 Märts 2022].
- [33] K3s Project Authors, „K3s: Lightweight Kubernetes,“ 2022. [Võrgumaterjal]. Available: <https://k3s.io/>. [Kasutatud 18 Märts 2022].
- [34] k3d Authors, „Github k3d-io/k3d,“ 2022. [Võrgumaterjal]. Available: <https://github.com/k3d-io/k3d>. [Kasutatud 14 Märts 2022].
- [35] Canonical Ltd., „Microk8s Zero-ops Kubernetes for developers, edge and IoT,“ 2022. [Võrgumaterjal]. Available: <https://microk8s.io/>. [Kasutatud 12 Märts 2022].
- [36] Canonical Ltd., „Github canonical/microk8s,“ 2022. [Võrgumaterjal]. Available: <https://github.com/canonical/microk8s>. [Kasutatud 13 Märts 2022].
- [37] The Kubernetes Authors, „Kind,“ 2021. [Võrgumaterjal]. Available: <https://kind.sigs.k8s.io/>. [Kasutatud 13 Märts 2022].
- [38] The Kubernetes Authors, „Github kubernetes-sigs/kind,“ 2022. [Võrgumaterjal]. Available: <https://github.com/kubernetes-sigs/kind>. [Kasutatud 13 Märts 2022].
- [39] M. Vool, „Github martivo/kind-minikube-k3d-microk8s,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-minikube-k3d-microk8s>. [Kasutatud 10 Aprill 2022].
- [40] S. Elgamel, „Github adding more control-plane nodes doesn't work,“ 17 Märts 2021. [Võrgumaterjal]. Available: <https://github.com/kubernetes/minikube/issues/10382#issuecomment-801315518>. [Kasutatud 26 Märts 2022].
- [41] S. Elgamel, „Github Allow multi control plane clusters by knowing which node is the api server,“ 7 Aprill 2020. [Võrgumaterjal]. Available: <https://github.com/kubernetes/minikube/issues/7461>. [Kasutatud 26 Märts 2022].
- [42] The Kubernetes Authors, „How can I create more than one cluster with minikube?,“ 2022. [Võrgumaterjal]. Available: <https://minikube.sigs.k8s.io/docs/faq/#how-can-i-create-more-than-one-cluster-with-minikube>. [Kasutatud 26 Märts 2022].
- [43] The Kubernetes Authors, „Minikube Using Multi-Node Clusters,“ 2022. [Võrgumaterjal]. Available: [https://minikube.sigs.k8s.io/docs/tutorials/multi\\_node](https://minikube.sigs.k8s.io/docs/tutorials/multi_node). [Kasutatud 25 Märts 2022].
- [44] The Kubernetes Authors, „Minikube Documentation Commands node,“ 2022. [Võrgumaterjal]. Available: <https://minikube.sigs.k8s.io/docs/commands/node/>. [Kasutatud 25 Märts 2022].

- [45] V. Jakkula, „MetalLB Configuration in Minikube — To enable Kubernetes service of type “LoadBalancer”,“ 26 Mai 2020. [Vörgumaterjal]. Available: <https://faun.pub/metallb-configuration-in-minikube-to-enable-kubernetes-service-of-type-loadbalancer-9559739787df>. [Kasutatud 25 Märts 2022].
- [46] Rancher, „Rancher K3s Architecture“, 2022. [Vörgumaterjal]. Available: <https://rancher.com/docs/k3s/latest/en/architecture/>. [Kasutatud 25 Märts 2022].
- [47] k3d Authors, „K3d cluster create“, 17 September 2021. [Vörgumaterjal]. Available: [https://k3d.io/v5.3.0/usage/commands/k3d\\_cluster\\_create/](https://k3d.io/v5.3.0/usage/commands/k3d_cluster_create/). [Kasutatud 25 Märts 2022].
- [48] k3d Authors, „K3d Using Config Files“, 22 Veebruar 2022. [Vörgumaterjal]. Available: <https://k3d.io/v5.3.0/usage/configfile/>. [Kasutatud 25 Märts 2022].
- [49] k3d Authors, „K3d Creating multi-server clusters“, 17 September 2021. [Vörgumaterjal]. Available: <https://k3d.io/v5.3.0/usage/multiserver/>. [Kasutatud 25 Märts 2022].
- [50] K. Lee, „Github keunlee/k3d-metallb-starter-kit“, 2021. [Vörgumaterjal]. Available: <https://github.com/keunlee/k3d-metallb-starter-kit>. [Kasutatud 25 Märts 2022].
- [51] Canonical Ltd., „Microk8s Command reference“, 2022. [Vörgumaterjal]. Available: <https://microk8s.io/docs/command-reference#heading--microk8s-join>. [Kasutatud 25 Märts 2022].
- [52] Canonical Ltd., „MicroK8s in LXD“, 2020. [Vörgumaterjal]. Available: <https://microk8s.io/docs/lxd>. [Kasutatud 25 Märts 2022].
- [53] Canonical Ltd., „Create a MicroK8s cluster“, 2022. [Vörgumaterjal]. Available: <https://microk8s.io/docs/clustering>. [Kasutatud 25 Märts 2022].
- [54] Canonical Ltd., „Microk8s Add on: MetalLB“, 2021. [Vörgumaterjal]. Available: <https://microk8s.io/docs/addon-metallb>. [Kasutatud 25 Märts 2022].
- [55] The Kubernetes Authors, „Kind Interacting With Your Cluster“, 24 Veebruar 2022. [Vörgumaterjal]. Available: <https://kind.sigs.k8s.io/docs/user/quick-start/#interacting-with-your-cluster>. [Kasutatud 25 Märts 2022].
- [56] The Kubernetes Authors, „Kind Configuration“, 2022. [Vörgumaterjal]. Available: <https://kind.sigs.k8s.io/docs/user/configuration/#kubeadm-config-patches>. [Kasutatud 25 Märts 2022].
- [57] The Kubernetes Authors, „Quick Start controlplane-ha“, 2022. [Vörgumaterjal]. Available: <https://kind.sigs.k8s.io/docs/user/quick-start/#controlplane-ha>. [Kasutatud 25 Märts 2022].
- [58] The Kubernetes Authors, „Kind LoadBalancer“, 2022. [Vörgumaterjal]. Available: <https://kind.sigs.k8s.io/docs/user/loadbalancer/>. [Kasutatud 25 Märts 2022].
- [59] M. Vool, „Github martivo/kind-lab“, 2022. [Vörgumaterjal]. Available: <https://github.com/martivo/kind-lab>. [Kasutatud 9 Aprill 2022].
- [60] M. Vool, „Github martivo/kind-lab/main.tf“, 2022. [Vörgumaterjal]. Available: <https://github.com/martivo/kind-lab/blob/master/main.tf>. [Kasutatud 9 Aprill 2022].
- [61] Docker Inc., „Docker docs download rate limit“, [Vörgumaterjal]. Available: <https://docs.docker.com/docker-hub/download-rate-limit/>. [Kasutatud 14 Aprill 2022].

- [62] M. Vool, „Github martivo/kind-lab/install-registry.sh,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/blob/master/install-registry.sh>. [Kasutatud 9 Aprill 2022].
- [63] M. Vool, „Github martivo/kind-lab/install-dependencies.sh,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/blob/master/install-dependencies.sh>. [Kasutatud 9 Aprill 2022].
- [64] M. Vool, „Github martivo/kind-lab/up.sh,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/blob/master/up.sh>. [Kasutatud 9 Aprill 2022].
- [65] M. Vool, „Github martivo/kind-lab/kind-prod.yaml,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/blob/master/kind-prod.yaml>. [Kasutatud 9 Aprill 2022].
- [66] M. Vool, „Github martivo/kind-lab/kind-test.yaml,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/blob/master/kind-test.yaml>. [Kasutatud 9 Aprill 2022].
- [67] M. Vool, „Github martivo/kind-lab/down.sh,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/blob/master/down.sh>. [Kasutatud 9 Aprill 2022].
- [68] M. Vool, „Github martivo/kind-lab/argocd-yaml/app-of-apps/templates/argocd-applications.yaml,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/blob/master/argocd-yaml/app-of-apps/templates/argocd-applications.yaml>. [Kasutatud 9 Aprill 2022].
- [69] M. Vool, „Github martivo/kind-lab/argocd-applications/templates,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/tree/master/argocd-applications/templates>. [Kasutatud 9 Aprill 2022].
- [70] M. Vool, „Github martivo/kind-lab/argocd-yaml,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/tree/master/argocd-yaml>. [Kasutatud 9 Aprill 2022].
- [71] A. ŞURA, „Basics: How to use BPMN 2.0 for your business in 2021,“ 26 Märts 2021. [Võrgumaterjal]. Available: <https://ebs-integrator.com/blog/bpmn-how-to-guide-2021/>. [Kasutatud 23 Aprill 2022].
- [72] M. Vool, „Github martivo/kind-lab/argocd-yaml/metallb/templates/configmap.yaml,“ 2022. [Võrgumaterjal]. Available: <https://github.com/martivo/kind-lab/blob/master/argocd-yaml/metallb/templates/configmap.yaml>. [Kasutatud 9 Aprill 2022].
- [73] The Kubernetes Authors, „Ingress,“ 2022. [Võrgumaterjal]. Available: <https://kubernetes.io/docs/concepts/services-networking/ingress/>. [Kasutatud 10 Aprill 2022].

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

Mina, Martin Vool

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Kubernetese laborikeskkond Entigo OÜ uue koolituse tarbeks“, mille juhendaja on Rein Remmel ja Siim Vene.
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

24.04.2022

---

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

## Lisa 2 – Skript kobara omaduste mõõtmiseks

```
#!/bin/bash
MEMORY_AVAIL=`free -m | grep Mem | awk '{print $7}'`
echo "Tööriista paigaldus"
START_TIME=$SECONDS

#Siia pannakseööriista paigaldamise käsud.

echo "Tööriista paigaldus võttis $(( $SECONDS - $START_TIME )) sekundit"
echo "Tööriistaga kobara loomine."
START_TIME=$SECONDS
free -m

#Siia pannakseööriistaga Kubernetese kobara loomise käsud

echo "Tööriista käivitamine võttis $(( $SECONDS - $START_TIME )) sekundit"
echo "Kontroll kas kõik sõlmed on valmis."
START_TIME=$SECONDS
foo=$(kubectl describe nodes | grep KubeletReady | wc -l)
while [ "$foo" -lt "8" ]
do
    sleep 1
    foo=$(kubectl describe nodes | grep KubeletReady | wc -l)
    echo "$foo nodes ready, waiting."
done
kubectl get nodes
kubectl get pods -n kube-system
echo "Kõik sõlmed on valmis $(( $SECONDS - $START_TIME )) sekundiga."
MEM_NOW=`free -m | grep Mem | awk '{print $7}'`
echo "Mälukasutuse muutus on $(( $MEMORY_AVAIL - $MEM_NOW ))"
free -m
START_TIME=$SECONDS

#Siia pannakseööriistaga kobara kustutamise käsud

echo "Kobara kustutamine võttis $(( $SECONDS - $START_TIME )) sekundit."
```



## Lisa 3 – Minikube tööriista omaduste mõõtmise käsud

```
#Tööriista paigaldus
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-
linux-amd64
sudo install minikube-linux-amd64 /usr/bin/minikube

#Tööriistaga kobara loomine
minikube start -p myminikube -n 8 --container-runtime docker --driver=docker

#Tööriistaga kobara kustutamine
minikube delete -p myminikube
```

## Lisa 4 – Microk8s tööriista omaduste mõõtmise käsud

```
#Tööriista paigaldus
sudo apt -y install lxd
sudo lxd init --auto --storage-backend=dir
lxc profile create microk8s
wget
https://raw.githubusercontent.com/ubuntu/microk8s/master/tests/lxc/microk8s.p
rofile -O microk8s.profile
cat microk8s.profile | lxc profile edit microk8s
```

```
#Tööriistaga kobara loomine
lxc launch -p default -p microk8s ubuntu:20.04 microk8s-m1
lxc launch -p default -p microk8s ubuntu:20.04 microk8s-m2 &
lxc launch -p default -p microk8s ubuntu:20.04 microk8s-m3 &
lxc launch -p default -p microk8s ubuntu:20.04 microk8s-w1 &
lxc launch -p default -p microk8s ubuntu:20.04 microk8s-w2 &
lxc launch -p default -p microk8s ubuntu:20.04 microk8s-w3 &
lxc launch -p default -p microk8s ubuntu:20.04 microk8s-w4 &
lxc launch -p default -p microk8s ubuntu:20.04 microk8s-w5 &
while [ "$(lxc list | grep RUNNING | wc -l)" -ne 8 ]
do
    sleep 0.1
done
lxc exec microk8s-m1 -- snap install microk8s --classic
lxc exec microk8s-m2 -- snap install microk8s --classic
lxc exec microk8s-m3 -- snap install microk8s --classic
lxc exec microk8s-w1 -- snap install microk8s --classic
lxc exec microk8s-w2 -- snap install microk8s --classic
lxc exec microk8s-w3 -- snap install microk8s --classic
lxc exec microk8s-w4 -- snap install microk8s --classic
lxc exec microk8s-w5 -- snap install microk8s --classic
lxc exec microk8s-m1 -- microk8s status --wait-ready
lxc exec microk8s-m2 -- microk8s status --wait-ready
lxc exec microk8s-m3 -- microk8s status --wait-ready
join=$(lxc exec microk8s-m1 -- microk8s add-node --format short | head -1)
lxc exec microk8s-m2 -- $join --controlplane
join=$(lxc exec microk8s-m1 -- microk8s add-node --format short | head -1)
lxc exec microk8s-m3 -- $join --controlplane
join=$(lxc exec microk8s-m1 -- microk8s add-node --format short | head -1)
lxc exec microk8s-w1 -- $join --worker
join=$(lxc exec microk8s-m1 -- microk8s add-node --format short | head -1)
lxc exec microk8s-w2 -- $join --worker
join=$(lxc exec microk8s-m1 -- microk8s add-node --format short | head -1)
lxc exec microk8s-w3 -- $join --worker
join=$(lxc exec microk8s-m1 -- microk8s add-node --format short | head -1)
lxc exec microk8s-w4 -- $join --worker
join=$(lxc exec microk8s-m1 -- microk8s add-node --format short | head -1)
lxc exec microk8s-w5 -- $join --worker
```

```
#Tööriistaga kobara kustutamine
lxc delete microk8s-m1 microk8s-m2 microk8s-m3 microk8s-w1 microk8s-w2
microk8s-w3 microk8s-w4 microk8s-w5 --force
```

## Lisa 5 – K3d tööriista omaduste mõõtmise käsud

```
#Tööriista paigaldus
curl -s https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh | bash

#Tööriistaga kobara loomine
k3d cluster create mycluster --config run_k3d.yaml

#Tööriistaga kobara kustutamine
k3d cluster delete mycluster

#Seadistusfail run_k3d.yaml
apiVersion: k3d.io/v1alpha4 # this will change in the future as we make
everything more stable
kind: Simple # internally, we also have a Cluster config, which is not yet
available externally
metadata:
  name: mycluster
servers: 3
agents: 5
options:
  k3d:
    wait: true
    timeout: "600s"
  k3s:
    nodeLabels:
      - label: ingress=true
      nodeFilters:
        - agent:0,1
    extraArgs: # additional arguments passed to the `k3s server|agent`
command; same as `--k3s-arg`
      - arg: --node-taint=ingress=true:NoSchedule
      nodeFilters:
        - agent:0,1
```

## Lisa 6 – Kind tööriista omaduste mõõtmise käsud

```
#Tööriista paigaldus
curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.11.1/kind-linux-amd64
chmod +x ./kind
sudo mv ./kind /usr/bin/kind
```

```
#Tööriistaga kobara loomine
kind create cluster --config run_kind.yaml
```

```
#Tööriistaga kobara kustutamine
kind delete cluster --name multi-a
```

```
#Seadistusfail run_kind.yaml
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
name: multi-a
nodes:
- role: control-plane
- role: control-plane
- role: control-plane
- role: worker
  kubeadmConfigPatches:
  - |
    kind: JoinConfiguration
    nodeRegistration:
      kubeletExtraArgs:
        node-labels: "ingress=true"
        register-with-taints: "ingress=true:NoSchedule"
- role: worker
  kubeadmConfigPatches:
  - |
    kind: JoinConfiguration
    nodeRegistration:
      kubeletExtraArgs:
        node-labels: "ingress=true"
        register-with-taints: "ingress=true:NoSchedule"
- role: worker
- role: worker
- role: worker
```

## Lisa 7 – Laborikeskkonna ekraanitõmmised

**Instances (3)** [Info](#)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	koolitus-kind-registry	i-0d259817e3b2e9b5f	Running	t3.small
<input type="checkbox"/>	kind-1	i-0e5715fa69373ae1f	Running	t3.large
<input type="checkbox"/>	kind-2	i-001d263f990fb6b54	Running	t3.large

Joonis 6. AWS EC2 koolituse laborikeskkonnas kasutatavad virtuaalmasinad

Harbor

Search Harbor...

Projects

Logs

Administration

- Users
- Robot Accounts
- Registries
- Replications
- Distributions
- Labels
- Project Quotas
- Interrogation Services

Projects

- Private
- Public
- Total

+ NEW PROJECT X DELETE

<input type="checkbox"/>	Project Name	Access Level	Role	Type
<input type="checkbox"/>	hub	Public	Project Admin	Proxy Cache
<input type="checkbox"/>	library	Public	Project Admin	Project
<input type="checkbox"/>	test	Public	Project Admin	Project

Joonis 7. Harbor tarkvara konteinerite registri kasutajaliidese vaade

**Records (16)** [Info](#)

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

🔍 *Filter records by property or value* Type

<input type="checkbox"/>	Record name	Type	Routin...	Differ...	Value/Route traffic to
<input type="checkbox"/>	argocd-prod-1.learn.entigo.io	A	Simple	-	18.184.112.10
<input type="checkbox"/>	argocd-test-1.learn.entigo.io	A	Simple	-	18.184.112.10
<input type="checkbox"/>	kind-1.learn.entigo.io	A	Simple	-	18.184.112.10
<input type="checkbox"/>	web-prod-1.learn.entigo.io	A	Simple	-	18.184.112.10
<input type="checkbox"/>	web-test-1.learn.entigo.io	A	Simple	-	18.184.112.10
<input type="checkbox"/>	kind-mgmt.learn.entigo.io	A	Simple	-	18.184.113.127
<input type="checkbox"/>	argocd-prod-2.learn.entigo.io	A	Simple	-	3.120.36.206
<input type="checkbox"/>	argocd-test-2.learn.entigo.io	A	Simple	-	3.120.36.206
<input type="checkbox"/>	kind-2.learn.entigo.io	A	Simple	-	3.120.36.206
<input type="checkbox"/>	web-prod-2.learn.entigo.io	A	Simple	-	3.120.36.206
<input type="checkbox"/>	web-test-2.learn.entigo.io	A	Simple	-	3.120.36.206
<input type="checkbox"/>	_c0d6bb7452ee5c675a5210...	CNAME	Simple	-	_f50c05f2c1a17859cda57fe19
<input type="checkbox"/>	kreg.learn.entigo.io	A	Simple	-	kind-registry-472389375.eu-c

Joonis 8. AWS veebiliidese vaade Route53 teenusesse loodud nimeserveri kanded

```

martivo@x1carb:~/ttu/loputoo/kind-lab$ ssh user2@kind-2.learn.entigo.io
Warning: Permanently added 'kind-2.learn.entigo.io,3.120.36.206' (ECDSA) to the list of known hosts.
user2@kind-2.learn.entigo.io's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1021-aws x86_64)

user2@ip-10-129-0-216:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
user2@ip-10-129-0-216:~$ ls -l
total 4
drwxr-xr-x 5 root root 4096 Apr 11 12:06 kind-lab
user2@ip-10-129-0-216:~$ ls -l kind-lab/
total 44
drwxr-xr-x 3 root root 4096 Apr 11 12:06 argocd-applications
drwxr-xr-x 7 root root 4096 Apr 11 12:06 argocd-yaml
-rwxr-xr-x 1 root root 135 Apr 11 12:06 down.sh
-rw-r--r-- 1 root root 1517 Apr 11 12:06 install-dependencies.sh
-rw-r--r-- 1 root root 3847 Apr 11 12:06 install-registry.sh
-rw-r--r-- 1 root root 1674 Apr 11 12:06 kind-prod.yaml
-rw-r--r-- 1 root root 1674 Apr 11 12:06 kind-test.yaml
-rw-r--r-- 1 root root 9204 Apr 11 12:06 main.tf
-rwxr-xr-x 1 root root 2416 Apr 11 12:06 up.sh
user2@ip-10-129-0-216:~$

```

Joonis 9. Ligipääs virtuaalmasinasse ja koodihoidlast pärit failide kuvamine

```

user2@ip-10-129-0-216:~/kind-lab$ ./up.sh
Creating cluster "test" ...
  ✓ Ensuring node image (kreg.learn.entigo.io/hub/kindest/node:v1.21.1) 🚧
  ✓ Preparing nodes 📦 📦 📦 📦 📦 📦 📦 📦 📦 📦
  ✓ Configuring the external load balancer 📡
  ✓ Writing configuration 📄
  ✓ Starting control-plane 🚦
  ✓ Installing CNI 🚦
  ✓ Installing StorageClass 🗄️
  ✓ Joining more control-plane nodes 📡
  ✓ Joining worker nodes 📦
Set kubectl context to "kind-test"
You can now use your cluster with:

kubectl cluster-info --context kind-test

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community 😊
Creating cluster "prod" ...
  ✓ Ensuring node image (kreg.learn.entigo.io/hub/kindest/node:v1.21.1) 🚧
  ✓ Preparing nodes 📦 📦 📦 📦 📦 📦 📦 📦 📦 📦
  ✓ Configuring the external load balancer 📡
  ✓ Writing configuration 📄
  ✓ Starting control-plane 🚦
  ✓ Installing CNI 🚦
  ✓ Installing StorageClass 🗄️
  ✓ Joining more control-plane nodes 📡
  ✓ Joining worker nodes 📦
Set kubectl context to "kind-prod"
You can now use your cluster with:

kubectl cluster-info --context kind-prod

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community 😊
*      kind-prod  kind-prod  kind-prod
      kind-test  kind-test  kind-test

```

Joonis 10. Kubernetes kobarate loomine Kind tööriistaga

```

user2@ip-10-129-0-216:~/kind-lab$ kubectl get nodes --context kind-test -o wide
NAME                STATUS    ROLES                    AGE      VERSION    INTERNAL-IP
test-control-plane  Ready    control-plane,master    5m45s   v1.21.1   172.18.0.2
test-control-plane2 Ready    control-plane,master    5m12s   v1.21.1   172.18.0.8
test-control-plane3 Ready    control-plane,master    4m15s   v1.21.1   172.18.0.11
test-worker         Ready    <none>                  3m56s   v1.21.1   172.18.0.7
test-worker2        Ready    <none>                  3m56s   v1.21.1   172.18.0.5
test-worker3        Ready    <none>                  3m56s   v1.21.1   172.18.0.4
test-worker4        Ready    <none>                  3m55s   v1.21.1   172.18.0.3
test-worker5        Ready    <none>                  3m56s   v1.21.1   172.18.0.12
test-worker6        Ready    <none>                  3m56s   v1.21.1   172.18.0.6
test-worker7        Ready    <none>                  3m56s   v1.21.1   172.18.0.9
user2@ip-10-129-0-216:~/kind-lab$ kubectl get nodes --context kind-prod -o wide
NAME                STATUS    ROLES                    AGE      VERSION    INTERNAL-IP
prod-control-plane  Ready    control-plane,master    3m23s   v1.21.1   172.18.0.20
prod-control-plane2 Ready    control-plane,master    2m57s   v1.21.1   172.18.0.23
prod-control-plane3 Ready    control-plane,master    2m30s   v1.21.1   172.18.0.21
prod-worker         Ready    <none>                  2m10s   v1.21.1   172.18.0.13
prod-worker2        Ready    <none>                  2m9s    v1.21.1   172.18.0.15
prod-worker3        Ready    <none>                  2m10s   v1.21.1   172.18.0.22
prod-worker4        Ready    <none>                  2m9s    v1.21.1   172.18.0.16
prod-worker5        Ready    <none>                  2m9s    v1.21.1   172.18.0.17
prod-worker6        Ready    <none>                  2m10s   v1.21.1   172.18.0.19
prod-worker7        Ready    <none>                  2m9s    v1.21.1   172.18.0.14

```

Joonis 11. Kubernetes kobaratega suhtluse testimine kubectl tööriistaga



```

user2@ip-10-129-0-216:~/kind-lab$ kubectl get pods -n kube-system --context kind-test -o wide | grep "api\|schedule"
NAME READY STATUS RESTARTS AGE IP NODE
etcd-test-control-plane 1/1 Running 0 13m 172.18.0.2 test-control-plane
etcd-test-control-plane2 1/1 Running 0 13m 172.18.0.8 test-control-plane2
etcd-test-control-plane3 1/1 Running 0 12m 172.18.0.11 test-control-plane3
kube-apiserver-test-control-plane 1/1 Running 0 13m 172.18.0.2 test-control-plane
kube-apiserver-test-control-plane2 1/1 Running 1 13m 172.18.0.8 test-control-plane2
kube-apiserver-test-control-plane3 1/1 Running 1 12m 172.18.0.11 test-control-plane3
kube-controller-manager-test-control-plane 1/1 Running 1 13m 172.18.0.2 test-control-plane
kube-controller-manager-test-control-plane2 1/1 Running 0 13m 172.18.0.8 test-control-plane2
kube-controller-manager-test-control-plane3 1/1 Running 0 11m 172.18.0.11 test-control-plane3
kube-scheduler-test-control-plane 1/1 Running 1 13m 172.18.0.2 test-control-plane
kube-scheduler-test-control-plane2 1/1 Running 0 13m 172.18.0.8 test-control-plane2
kube-scheduler-test-control-plane3 1/1 Running 0 11m 172.18.0.11 test-control-plane3
user2@ip-10-129-0-216:~/kind-lab$ kubectl get pods -n kube-system --context kind-prod -o wide | grep "api\|schedule"
NAME READY STATUS RESTARTS AGE IP NODE
etcd-prod-control-plane 1/1 Running 0 11m 172.18.0.20 prod-control-plane
etcd-prod-control-plane2 1/1 Running 0 10m 172.18.0.23 prod-control-plane2
etcd-prod-control-plane3 1/1 Running 0 10m 172.18.0.21 prod-control-plane3
kube-apiserver-prod-control-plane 1/1 Running 0 11m 172.18.0.20 prod-control-plane
kube-apiserver-prod-control-plane2 1/1 Running 0 10m 172.18.0.23 prod-control-plane2
kube-apiserver-prod-control-plane3 1/1 Running 0 10m 172.18.0.21 prod-control-plane3
kube-controller-manager-prod-control-plane 1/1 Running 0 11m 172.18.0.20 prod-control-plane
kube-controller-manager-prod-control-plane2 1/1 Running 0 10m 172.18.0.23 prod-control-plane2
kube-controller-manager-prod-control-plane3 1/1 Running 0 10m 172.18.0.21 prod-control-plane3
kube-scheduler-prod-control-plane 1/1 Running 0 11m 172.18.0.20 prod-control-plane
kube-scheduler-prod-control-plane2 1/1 Running 0 10m 172.18.0.23 prod-control-plane2
kube-scheduler-prod-control-plane3 1/1 Running 0 10m 172.18.0.21 prod-control-plane3

```

Joonis 12. Kubernetese juhttasandi komponentide töötamine juhtsõlmede kaunades

```

user2@ip-10-129-0-216:~/kind-lab$ docker ps --format "{{.ID}}\t{{.Image}}\t{{.Names}}"
987fc675cf04 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-worker6
8dc86bb6e1b6 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-control-plane3
2149477967ea kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-control-plane
6ed6207b9efc kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-worker3
8be7cdb8c226 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-control-plane2
507eac96c618 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-worker7
968188dce044 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-worker5
25cf3553e055 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-worker4
a20f5a306f12 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-worker
609203384a2d kreg.learn.entigo.io/hub/kindest/node:v1.21.1 prod-worker2
43394ba2c6f1 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-worker7
e24c5613eb4d kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-worker
032e5b7fb53c kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-control-plane3
e3d2775e4d77 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-worker6
7528c8231408 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-worker5
7da2300db81a kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-control-plane2
3a3270c5fc00 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-worker2
ca7604c76444 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-worker4
eff9d2c9abfa kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-control-plane
eef136a72ea0 kreg.learn.entigo.io/hub/kindest/node:v1.21.1 test-worker3

```

Joonis 13. Tööriista loodud kobara sõlmede töötamine konteinerites

```

user2@ip-10-129-0-216:~/kind-lab$ kubectl --context=kind-prod get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)        AGE
api           ClusterIP    10.93.59.92   <none>         80/TCP         2d4h
kubernetes    ClusterIP    10.93.0.1     <none>         443/TCP        2d4h
redis         ClusterIP    10.93.101.20  <none>         6379/TCP       2d4h
web           ClusterIP    10.93.218.79  <none>         80/TCP         2d4h
user2@ip-10-129-0-216:~/kind-lab$ kubectl --context=kind-prod get deploy
NAME    READY    UP-T0-DATE    AVAILABLE    AGE
api     2/2      2              2            2d4h
web     2/2      2              2            2d4h
user2@ip-10-129-0-216:~/kind-lab$ kubectl --context=kind-prod get sts
NAME    READY    AGE
redis   1/1      2d4h
user2@ip-10-129-0-216:~/kind-lab$ kubectl --context=kind-prod get pod
NAME          READY    STATUS    RESTARTS    AGE
api-b9b77c48c-lb9jk    1/1     Running   2           2d4h
api-b9b77c48c-zvpfd    1/1     Running   2           2d4h
redis-0           1/1     Running   0           2d4h
web-547f7967bf-mwbrv   1/1     Running   0           2d4h
web-547f7967bf-xknxt   1/1     Running   0           2d4h
user2@ip-10-129-0-216:~/kind-lab$ kubectl --context=kind-prod get ingress
NAME    CLASS    HOSTS          ADDRESS          PORTS    AGE
web     <none>   web-prod-2.learn.entigo.io  172.18.0.200    80       2d4h
user2@ip-10-129-0-216:~/kind-lab$ kubectl --context=kind-test get ingress
NAME    CLASS    HOSTS          ADDRESS          PORTS    AGE
web     <none>   web-test-2.learn.entigo.io  172.18.0.210    80       2d4h

```

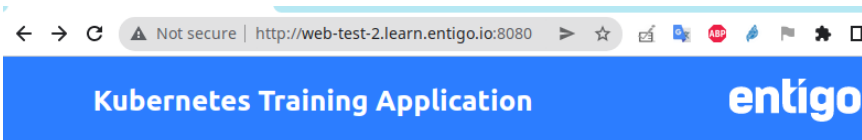
Joonis 14. Entigo näidisrakenduse objektide kuvamine

```

user2@ip-10-129-0-216:~/kind-lab$ kubectl --context=kind-prod get svc -n haproxy
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)
haproxy-ingress    LoadBalancer  10.93.26.167   172.18.0.200   80:31276/TCP,443:32517/TCP,1024:31887/TCP
haproxy-ingress-default-backend  ClusterIP    10.93.112.153  <none>         8080/TCP
user2@ip-10-129-0-216:~/kind-lab$ kubectl --context=kind-test get svc -n haproxy
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)
haproxy-ingress    LoadBalancer  10.92.145.114  172.18.0.210   80:30486/TCP,443:30369/TCP,1024:31471/TCP
haproxy-ingress-default-backend  ClusterIP    10.92.122.248  <none>         8080/TCP
user2@ip-10-129-0-216:~/kind-lab$ sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 1553 packets, 93020 bytes)
pkts bytes target      prot opt in     out    source          destination
 3   180 DNAT      tcp  --  ens5   *     0.0.0.0/0       0.0.0.0/0          tcp dpt:8080 to:172.18.0.210:80
 6   360 DNAT      tcp  --  ens5   *     0.0.0.0/0       0.0.0.0/0          tcp dpt:80 to:172.18.0.200:80

```

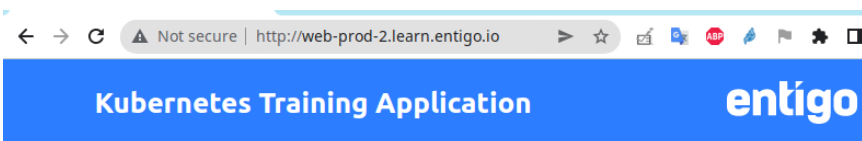
Joonis 15. Koormusjaotur tüüpi teenuse objektid ja võrguühenduste suunamine



Counter: 4 Pod: api-b9b77c48c-tvp6r Namespace: default  
Time: Tue Apr 19 2022 14:26:03 GMT+0000 (Coordinated  
Universal Time) Client IP: ::ffff:10.241.7.2 X-forwarded-  
for: 185.46.20.35

```
[info] Received a message from /api: Counter: 4  
Pod: api-b9b77c48c-tvp6r  
Namespace: default  
Time: Tue Apr 19 2022 14:26:03 GMT+0000 (Coordinated Universal Time)  
Client IP: ::ffff:10.241.7.2  
X-forwarded-for: 185.46.20.35
```

Joonis 16. Entigo näidisrakendus „test“ kobaras



Counter: 1 Pod: api-b9b77c48c-zvpfd Namespace: default  
Time: Tue Apr 19 2022 14:27:11 GMT+0000 (Coordinated  
Universal Time) Client IP: ::ffff:10.242.7.2 X-forwarded-  
for: 185.46.20.35

```
[info] Received a message from /api: Counter: 1  
Pod: api-b9b77c48c-zvpfd  
Namespace: default  
Time: Tue Apr 19 2022 14:27:11 GMT+0000 (Coordinated Universal Time)  
Client IP: ::ffff:10.242.7.2  
X-forwarded-for: 185.46.20.35
```

Joonis 17. Entigo näidisrakendus „prod“ kobaras

```

user2@ip-10-129-0-216:~/kind-lab$ kubectl get nodes --context=kind-prod
NAME                                STATUS    ROLES                                AGE      VERSION
prod-control-plane                 Ready    control-plane,master                2d5h    v1.21.1
prod-control-plane2                Ready    control-plane,master                2d5h    v1.21.1
prod-control-plane3                Ready    control-plane,master                2d5h    v1.21.1
prod-worker                         Ready    <none>                              2d5h    v1.21.1
prod-worker2                       Ready    <none>                              2d5h    v1.21.1
prod-worker3                       Ready    <none>                              2d5h    v1.21.1
prod-worker4                       Ready    <none>                              2d5h    v1.21.1
prod-worker5                       Ready    <none>                              2d5h    v1.21.1

```

Joonis 18. Kubernetese sõlmed enne tõrkesiirde testi

```

user2@ip-10-129-0-216:~/kind-lab$ kubectl get pods --context=kind-prod -o wide
NAME                                READY    STATUS    RESTARTS   AGE    IP             NODE
api-b9b77c48c-lb9jk                 1/1     Running   2          2d5h   10.242.5.2    prod-worker4
api-b9b77c48c-zvpfd                 1/1     Running   2          2d5h   10.242.6.2    prod-worker3
redis-0                              1/1     Running   0          2d5h   10.242.5.3    prod-worker4
web-547f7967bf-mwbrv                1/1     Running   0          2d5h   10.242.6.3    prod-worker3
web-547f7967bf-xknxt                1/1     Running   0          2d5h   10.242.3.2    prod-worker5

```

Joonis 19. Entigo näidisrakenduse kaunad enne tõrkesiirde testi

```

user2@ip-10-129-0-216:~/kind-lab$ kubectl get nodes
NAME                                STATUS    ROLES                                AGE      VERSION
prod-control-plane                 NotReady  control-plane,master                2d5h    v1.21.1
prod-control-plane2                Ready     control-plane,master                2d5h    v1.21.1
prod-control-plane3                Ready     control-plane,master                2d5h    v1.21.1
prod-worker                         NotReady  <none>                              2d5h    v1.21.1
prod-worker2                       Ready     <none>                              2d5h    v1.21.1
prod-worker3                       NotReady  <none>                              2d5h    v1.21.1
prod-worker4                       Ready     <none>                              2d5h    v1.21.1
prod-worker5                       Ready     <none>                              2d5h    v1.21.1

```

Joonis 20. Kubernetese sõlmed pärast tõrkesiirde testi

```

user2@ip-10-129-0-216:~/kind-lab$ kubectl get pods --context=kind-prod -o wide
NAME                                READY    STATUS    RESTARTS   AGE    IP             NODE
api-b9b77c48c-9rdb2                 1/1     Running   2          2m7s   10.242.3.3    prod-worker5
api-b9b77c48c-lb9jk                 1/1     Running   2          2d5h   10.242.5.2    prod-worker4
api-b9b77c48c-zvpfd                 1/1     Terminating 2        2d5h   10.242.6.2    prod-worker3
redis-0                              1/1     Running   0          2d5h   10.242.5.3    prod-worker4
web-547f7967bf-mwbrv                1/1     Terminating 0        2d5h   10.242.6.3    prod-worker3
web-547f7967bf-p6fsw                1/1     Running   1          2m7s   10.242.5.4    prod-worker4
web-547f7967bf-xknxt                1/1     Running   0          2d5h   10.242.3.2    prod-worker5

```

Joonis 21. Entigo näidisrakenduse kaunad pärast tõrkesiirde testi