

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Anneli Väli 206532IAAB

# **Seire- ja logimislahenduste integreerimine ettevõttele Datel AS**

Bakalaureusetöö

Juhendaja: Margus Sumla  
MSc

Tallinn 2024

## **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: Anneli Väli

13.05.2024

## **Annotatsioon**

Käesolevas bakalaureusetöös käsitletakse Datel AS-i IT-infrastruktuuri parandamiseks mõeldud seire- ja logimissüsteemi arendamist ja juurutamist. Datel AS on spetsialiseerunud keerukatele tarkvaraarendusprojektidele ja kosmosetehnoloogiatele. Peamine probleem seisneb selles, et olemasolev süsteem, mis põhineb Grafana ja Prometheusel, on puudulik ja ei suuda kiiresti arenevas tehnoloogilises keskkonnas nõutavaid seireülesandeid täita, mis toob kaasa sagedased tõrked ja kõrged operatsioonikulud. Lõputöö eesmärk on luua uuendatud, intuitiivne ja kasutajasõbralik seiresüsteem, mis tagab ettevõtte teenuste usaldusväarsuse ja kättesaadavuse, võimaldades reaalajas seiret. Töös kohandatud analüüs hõlmab sobiva seire- ja logimistarkvara analüüsi ja valikut ning selle integreerimist olemasolevasse infrastruktuuri, kasutades kaasaegseid tehnoloogiaid nagu Kubernetes, Rancher ja Helm. Lõputöö tulemuste eesmärk on oluliselt vähendada süsteemi riketele reageerimise aega ja ressursikasutust, tagades sellega ettevõtte IT-infrastruktuuri stabiilsema toimimise.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 6 peatükki, 16 joonist, 2 tabelit.

## **Abstract**

### **Integration of Monitoring and Logging Solution for Datel AS Company**

This Bachelor's thesis deals with the development and deployment of a monitoring and logging system designed to improve Datel AS IT infrastructure. Datel AS specialises in complex software development projects and space technologies. The main problem is that the existing system, based on Grafana and Prometheus, is deficient and unable to perform the monitoring tasks required in a rapidly evolving technological environment, leading to frequent failures and high operating costs. The aim of the thesis is to create an updated, intuitive and user-friendly monitoring system that ensures the reliability and availability of the company's services, enabling real-time monitoring. Work-tailored analysis includes the analysis and selection of appropriate monitoring and logging software and its integration into existing infrastructure using modern technologies such as Kubernetes, Rancher and Helm. The results of the final work aim to significantly reduce the time and resource use of the system's failure response, thereby ensuring a more stable functioning of the company's IT infrastructure.

The thesis is in Estonian and contains 28 pages of text, 6 chapters, 16 figures, 2 tables.

## Lühendite ja mõistete sõnastik

API	<i>Application programming interface</i> , rakendusliides
AWS	<i>Amazon Web Services</i> , pilveteenuste platvorm
CD	<i>Continuous Deployment</i> , pidevtarne
CI	<i>Continuous Integration</i> , pidev integratsioon
GCS	<i>Google Cloud Storage</i> , pilvesalvestus
HTTP	<i>HyperText Transfer Protocol</i> , võrguprotokoll andmete edastamiseks arvutivõrkudes
K8s	<i>Kubernetes</i> , konteinerihalduse süsteem
RDBMS	<i>Relational database management system</i> , objekt-relatsiooniline andmebaaside haldussüsteem
S3	<i>Simple Storage Service</i> , objektisalvestusteenus
URL	<i>Uniform resource locator</i> , internetiaadress

## Sisukord

1 Sissejuhatus .....	10
1.1 Probleemi taust .....	10
1.2 Probleemi püstitus .....	11
1.3 Eesmärk .....	12
2 Nõuded .....	13
2.1 Ärinõuded .....	13
2.2 Funktsionaalsed nõuded .....	13
2.3 Mittefunktsionaalsed nõuded.....	13
3 Analüüs.....	15
3.1 Olemasoleva süsteemi analüüs .....	15
3.2 Mõõdustiku talletamise tehnoloogiate analüüs.....	15
3.2.1 Cortex .....	16
3.2.2 Mimir.....	16
3.2.3 Thanos .....	17
3.2.4 Mõõdustiku talletamise tehnoloogiate võrdlus.....	17
3.2.5 Tehnoloogia valik.....	18
3.3 Logide talletamise tehnoloogiate analüüs.....	18
3.3.1 Loki.....	18
3.3.2 Elasticsearch.....	20
3.3.3 Logide talletamise tehnoloogiate võrdlus.....	21
3.3.4 Tehnoloogia valik.....	21
4 Lahenduse realiseerimine .....	22
4.1 Mõõdustiku talletamise tehnoloogia paigaldamine .....	23
4.1.1 Mimir'i paigaldamisprotsess .....	23
4.1.2 Mimir'i kollektori paigaldamine .....	23
4.2 Tsentraalse logimislahenduse paigaldamine .....	24
4.2.1 Loki paigaldamisprotsess .....	24
4.2.2 Loki kollektori paigaldamine .....	25
4.3 Grafana seiresüsteemi seadistamine ja paigaldamine.....	25

4.3.1 Grafana seiresüsteemi paigaldamisprotsess .....	25
4.3.2 Postgres andmebaasi paigaldamisprotsess .....	26
4.4 Grafana seadistamine.....	26
4.4.1 Andmeallika lisamine .....	26
4.4.2 Grafana seirevaade seadistamine.....	27
4.4.3 Seire-paneelide lisamine ja seadistamine .....	27
4.5 Grafana teavituste seadistamine .....	34
5 Tulemuste analüüs .....	36
6 Kokkuvõte .....	37
Kasutatud kirjandus .....	38
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	41
Lisa 2 – Cortex'i arhitektuur .....	42
Lisa 3 – Mimir'i arhitektuur mikroteenuse reziimis.....	43
Lisa 4 – Thanose arhitektuur .....	44
Lisa 5 – Loki klasteri arhitektuur .....	45
Lisa 6 – Mimir helm chart'i näide .....	46
Lisa 7 – Loki helm chart'i näide.....	47
Lisa 8 –Lõputöös loodud lahenduse arhitektuur .....	48

## Jooniste loetelu

Joonis 1. Loki logimiskogum .....	19
Joonis 2. Elasticsearch'i arhitektuur .....	20
Joonis 3. Andmeallika lisamine grafanas .....	27
Joonis 4. Grafana seirepaneeli loomine .....	28
Joonis 5. Protsessori tuumade arv serveris .....	28
Joonis 6. Süsteemi töötamise aeg .....	28
Joonis 7. Serveri operatiivmälu kasutuse graafik .....	29
Joonis 8. Paneel, mis näitab serveri kettaruumi kasutust .....	29
Joonis 9. Serveri süsteemi koormuse paneel graafikuna .....	30
Joonis 10. Protsessori kasutamine graafikuna .....	30
Joonis 11. Serveri operatiivmälu kasutamine graafikuna .....	31
Joonis 12. Serveri valimine seirevaade jaoks .....	31
Joonis 13. Logide seirevaade .....	32
Joonis 14. Klatri operatiivmälu kasutus .....	33
Joonis 15. Klatri protsessori kasutus .....	34
Joonis 16. Teavitused paneelide jaoks .....	35



## **Tabelite loetelu**

Tabel 1. Mõõdustiku talletamise tehnoloogiate võrdlus.....	17
Tabel 2. Logide talletamise tehnoloogiate võrdlus.....	21

# 1 Sissejuhatus

Bakalaureusetöö keskendub Datel AS seire- ja logimissüsteemide arendamisele ja integreerimisele. IT-sektori kiire areng nõuab süsteeme, mis võimaldavad ettevõtetel reaalsajas jälgida ja analüüsida oma IT-taristu toimimist, aidates kaasa järjepidevale ja katkematule tegevusele.

Seire võimaldab organisatsioonidel hallata infosüsteeme ja tagada nende tõrgeteta toimimine. Süsteemide jälgimine tuvastab probleeme enne nende tekkimist ja võimaldab lahendada neid ettenägelikult, et vältida katkestusi ja andmekadusid, mis mõjutavad tegevust ja mainet. IT-seire optimeerib ressursikasutust ja tuvastab kitsaskohti, mis parandab jõudlust ja võimaldab IT-meeskondadel teha kiireid otsuseid süsteemide täiustamiseks [1].

Samuti teavituste süsteem on oluline, sest see võimaldab meeskondadel tuvastada ja reageerida potentsiaalsetele probleemidele enne, kui need põhjustavad tõsisemaid süsteemirikkeid. Õigeaegsed teavitused aitavad vältida tööseisakuid ja kiirendada probleemide lahendamist, mis omakorda vähendab ettevõttele tekkivat kahju ning aitab säilitada süsteemi stabiilsust ja parandada kasutajakogemust, tagades teenuste pideva kättesaadavuse [2].

Lisaks seirele on oluline osa süsteemide efektiivses haldamises logimislahendus. Logide kiire ja mugav kättesaamine on nii arendajatele, kui ka süsteemi administraatoritele hädavajalik tõhusate rakenduste arendamiseks ja uute tehnoloogiate kasutusele võtmiseks.

## 1.1 Probleemi taust

Datel AS, asutatud 1990. aastal Eestis, on IT-ettevõtte, mis on spetsialiseerunud keerukatele tarkvaraarendusprojektidele, IT-infrastruktuurilahendustele ja kosmosetehnoloogiatele, sealhulgas uuenduslikule Sille.space teenusele [3]. Ettevõtte kasutab mitmesuguseid kaasaegseid tehnoloogilisi lahendusi, nagu Kubernetes, Rancher,

Helm ja GitLab CI/CD, mis aitavad kaasa tõhusale koodihaldusele ja võimaldavad pidevat integratsiooni ja levitamist (CI/CD).

Ettevõtte praegune Grafana-põhine lahendus on vananenud ja ei vasta enam ettevõtte kiiresti arenevatele nõudmistele. See puudujääk ilmneb eriti selgelt, kui on vaja integreerida uusi tehnoloogiaid. Süsteemi ebatõhusus on põhjustanud mitmeid tehnilisi rikkeid, mida on avastatud liiga hilja, põhjustades ootamatuid seisakuid ja suurendades operatsioonikulusid.

Uue IT-seiresüsteemi arendamine on muutunud kriitiliseks, et toetada ettevõtte laienemist ja suurendada süsteemide tõhusust. Datel AS vajab süsteemi, mis võimaldaks mitte ainult praeguse infrastruktuuri jälgimist, vaid oleks ka piisavalt paindlik, et toime tulla pidevalt areneva tehnoloogia maastikuga. Uus süsteem peaks pakkuma paremaid andmeanalüüsi tööriistu, kiiremat häirete tuvastamist ja tõhusamat ressursside haldust.

Uuendatud seire- ja logimislahenduse süsteem aitaks Datel AS-l jätkata oma tegevust kõrgel tasemel, pakkudes stabiilsust ja usaldusväarsust, mis on hädavajalik kiiresti arenevas tehnoloogiakeskkonnas. Lisaks sellele tagaks see, et ettevõtte suudab jätkuvalt pakkuda tipptasemel teenuseid oma klientidele, hoides samal ajal kulud kontrolli all, samas minimeerides võimalike tõrgete mõju.

## **1.2 Probleemi püstitus**

Datel AS kasutab oma IT-infrastruktuuri seireks vananenud Grafana ja Prometheus'e süsteeme, mis praegusel kujul on puudulikud. Nimelt puuduvad süsteemides vajalikud seirevaated ja funktsionaalsuse seadistus, mistõttu on IT-meeskonnal raskusi teenuste seisukorra ülevaatega. Selline olukord on juba põhjustanud tõsiseid probleeme, sealhulgas süsteemide kokkukukkumisi ülekoormuse tõttu, mille tagajärjel on meeskond sunnitud süsteeme taastama ja uuesti seadistama.

Ettevõtte IT-taristu hõlmab kaasaegseid tehnoloogiaid nagu Kubernetes, Rancher, Git, Gitlab CI/CD, Grafana, Prometheus ja Helm, kuid praegune seiresüsteem ei ole piisavalt integreeritud ega konfigureeritud, et tagada nende tehnoloogiate efektiivne jälgimine ja haldamine. Eriti oluline on see, et süsteem suudaks mõõta ja jälgida ressursside kasutust iga teenuse puhul, mis praegu on võimalik ainult käsitsi.

Riskid, nagu süsteemide ülekoormus, on endiselt suur mure, sest olemasolev süsteem ei suuda neid tõhusalt tuvastada ega nendele proaktiivselt reageerida. See puudujääk seab ohtu kogu IT-taristu stabiilsuse ja võib põhjustada tõsiseid häireid ettevõtte tegevuses.

Võttes arvesse praeguseid probleeme IT-infrastruktuuriga, on selge, et Datel AS vajab uut, kaasaegsemat ja täiustatud seiresüsteemi ja logimislahendust, mis oleksid kohandatud praeguste ja tulevaste tehnoloogiate jälgimiseks. Uus süsteem peaks olema skaleeritav, paindlik ja kohanduv, et see saaks kasvada koos ettevõtte vajadustega. Oluline on ka uue süsteemi kujundamisel arvestada nii töötajate, kui ka teenuste nõudmistega, tagades, et kõik kriitilised komponendid on jälgitavad ja hallatavad, mis vähendab süsteemi ülekoormuse ja muude probleemide tekkimise riski.

### **1.3 Eesmärk**

Eesmärk on välja töötada ja seadistada Datel AS-i vajadustele vastav seire- ja logimissüsteem, mis tagab ettevõtte teenuste töökindluse ja kättesaadavuse ning võimaldab proaktiivselt reageerida võimalikele süsteemi rikketele. See hõlmab kõige tõhusamate ja sobivamate tehnoloogiliste lahenduste valimist ja nende rakendamist, mis võimaldavad reaalajas seiret ja logimist. Lisaks pannakse rõhku süsteemi kasutajaliidese intuiivsusele ja kasutusmugavusele, et tagada IT-meeskonna efektiivne töö süsteemiga. Eesmärkide saavutamise protsess hõlmab seire- ja logimistarkvarade analüüsi, valikut ja integreerimist, et tagada süsteemi optimaalne jõudlus ja vastavus ettevõtte nõuetele.

Lisaks sellele seatakse konkreetne eesmärk vähendada süsteemirikketele reageerimise aega ja kulutatud ressursse. On soov optimeerida seiresüsteemi, et varajane avastamine ja sekkumine vähendaks sarnaste intsidentide lahendamiseks kuluvat aega ning vähendaks kulusid. Selline parandus mitte ainult ei tõsta IT-infrastruktuuri usaldusväärsust, vaid toob ka märkimisväärsed kokkuhoiu tulemusi, mis parandab üldist projekti- ja äri efektiivsust.

## 2 Nõuded

Lõputöö käigus arendatav seire- ja logimissüsteem peab vastama erinõuetele, et see oleks edukalt integreeritud Datel AS-i olemasolevasse IT-taristusse ning vastaks ettevõtte operatiiv- ja tehnilistele vajadustele. Süsteemi põhinõuded on esitatud allpool.

### 2.1 Ärinõuded

Lahendus peab kasutama avatud lähtekoodiga tehnoloogiaid, et minimeerida litsentsi- ja hoolduskulusid. Siiski tuleb hinnata igat tehnoloogiat selle tegelike omamis- ja kasutuskulude põhjal, et tagada süsteemi tõeline kulutõhusus.

### 2.2 Funktsionaalsed nõuded

- **Reaalajas andmete kuvamine:** Süsteem peab tagama reaalajas andmete töötluste ja kuvamise.
- **Ressursikasutuse jälgimine:** Süsteem peab suutma kuvada serverite ressursikasutust, nagu operatiivmälu ja protsessori koormus. Et hõlbustada koormuse jälgimist ja optimeerimist on vaja luua seirevaated ja -paneelid, mis kuvavad kõiki nõutud andmeid.
- **Skaleeritavus:** Süsteem peab olema võimeline kohanduma suureneva koormusega, ilma et selle jõudlus väheneks. See tähendab, et süsteemi arhitektuur ja komponendid peavad toetama skaleeritavust.

### 2.3 Mittefunktsionaalsed nõuded

- **Integreerimise võimalus Grafanaga:** Süsteem peab olema sujuvalt integreeruv Prometheus- ja Grafanaga, et visualiseerida logisid ja mõõdustikke ühest kohast.
- **Kubernetese klatri integreerimine:** Kogu süsteem peab olema juurutatav ja integreeritav Kubernetese klatriga, mis tagab skaleeritavuse, paindlikkuse ja klatri mugava haldamise.

- **Salvestamine keskse asukohta:** Lahendus peab tagama kõigi logide ja mõõdustike salvestamist kesksesse asukohta, kasutades selleks S3 andmesalvestus tehnoloogiat.
- **Pikaajaline mõõdustiku ja logide säilitamine:** Lahendus peab olema võimeline talletada mõõdustiku ja logisid vähemalt kuus kuud.

Need nõuded on hoolikalt valitud ja koostatud, et tagada arendatava seire- ja logimissüsteemi vastavus Datel AS-i operatiivsetele vajadustele ning tehnilistele nõuetele.

## **3 Analüüs**

Käesolevas osas keskendutakse Datel AS-i praeguse seire- ja logimissüsteemi analüüsile ning uuritakse ja võrreldakse erinevaid mõõdustike ja logide talletamise tehnoloogiaid, et leida ettevõtte vajadustele kõige optimaalsem lahendus. Eesmärk on mõista olemasoleva süsteemi tugevusi ja nõrkusi, selgitada välja peamised arenguvajadused ning leida kõige sobivamad tehnoloogilised lahendused, mis tagavad süsteemi efektiivsema toimimise, skaleeritavuse ja parema integreerimise olemasolevasse IT-taristusse.

### **3.1 Olemasoleva süsteemi analüüs**

Hetkel on meeskonnal puudulik seirelahendus, samuti on kasutusel olev logimislahendus vananenud ja ebamugav nii haldus- kui ka kasutajakogemuse mõttes. Lisaks ei ole olemasoleval seiresüsteemil asjakohaseid seirevaateid põhiteenuste kohta ning teateid nii probleemide ennetamise, kui ka süsteemirikete kohta.

Üheks näiteks selles lõputöös on olukord, kus Kubernetes'e klatri sõlmede ressursside ülekoormuse tõttu toimus klatri terviklikkuse rikkumine. Seetõttu peatusid arendajate ja klientide töövood mitmeks tunniks, enne kui probleem lahendati. Kuna seiresüsteem oli puudulik ja hoiatused olid puudu, avastati probleem mõne tunni pärast. Probleemi uurimiseks ja süsteemi taastamiseks kulus umbes 20 tundi ning see põhjustas ligikaudu 4500€ suuruse rahalise kahju.

### **3.2 Mõõdustiku talletamise tehnoloogiate analüüs**

Mõõdustikute salvestamiseks vajalike tehnoloogiate valikul võetakse arvesse mitmeid kriteeriume, nagu andmete kogumise tõhusus, haldamise lihtsus, analüüsivõime ja integreerimine teiste süsteemidega. Uuritakse ja võrreldakse erinevaid lahendusi, sealhulgas Mimir, Cortex ja Thanos, et hinnata nende sobivust Datel AS-i nõuetega. Iga tehnoloogia puhul analüüsitakse selle arhitektuuri, jõudlust suurte andmehulkade korral, päringute täitmise kiirust, süsteemi halduskulude aspekte ja nende laiaulatusliku kogukonna, mis aitab kaasa järjepidevatele uuendustele. Mimir, Cortex ja Thanos on

valitud nende eriliste omaduste, nagu skaleeritavus, võimalus talletada andmed mitmeks kuuks ja salvestada andmed kesksesse asukohta, tagades kõrge jõudlus reaalajas andmetöötluses.

### 3.2.1 Cortex

Cortex on välja töötatud, et pakkuda suure jõudlusega ja horisontaalselt skaleeritavat mõõdustiku talletus- ja päringusüsteem, mis suudab efektiivselt hallata suuri andmekogumeid. See süsteem on eriti sobiv suurte organisatsioonide jaoks, kellel on keerukad IT-taristud [4]. Cortex kasutab jaotatud andmebaasi, et tagada andmete kõrge kättesaadavus ja vastupidavus, olles hädavajalik suurte ja keerukate IT-taristutega organisatsioonidele. Cortex on loodud töötama suurel skaalal, pakkudes mitmekihilist turvalisust, automaatset skaleerimist ja sujuvat integreeritavust teiste süsteemidega [5]. Lisa 2 illustreerib Cortex'i arhitektuuri, mis sisaldab järgmisi põhikomponente:

- **Compaktor:** Optimeerib andmete salvestust, vähendades andmemahtu ja parandades päringute tõhusust.
- **Distributor:** Vastutab sisenevate mõõdikute vastuvõtmise ja nende edastamise eest ingesteritele.
- **Store Gateway:** Tagab juurdepääsu salvestatud andmetele, toimides andmete ja päringute töötlejate vahelise liidese rollis.
- **Ingester:** Hoiab mõõdikuid ajutiselt mälus, enne kui need pikaajalisse salvestusse kirjutatakse, aidates kaasa andmete kiirele töötlemisele.
- **Querier:** Võimaldab päringute tegemist salvestatud andmetele, kasutades PromQL keelt, mis on Prometheus päringukeel.
- **Ruler:** Töötleb andmete vastu reegleid, mis hõlmavad teavitamisi ja andmete automaatset töötlemist.

### 3.2.2 Mimir

Mimir, hiljuti integreeritud osa Grafana ökosüsteemist, on mõeldud suurtes kogustes ajalooliste mõõdustikute tõhusaks säilitamiseks ja päringuteks [6]. See põhineb Cortexi arhitektuuril, millele on lisatud olulisi parandusi jõudluses ja skaleeritavuses [7]. Mimir



toetab PromQL keelt, võimaldades kiireid ja tõhusaid päringuid suurtele andmekogumitele, mis on oluline reaalajas andmeanalüüsi ja otsuste tegemise protsessides ettevõtetes [8]. Mimir suudab integreeruda erinevate andmeallikatega, pakkudes laia valikut andmetöötluse võimalusi, alates lihtsatest päringutest kuni keerukate analüütiliste aruanneteni [9]. Lisa 3 näitab Mimir'i arhitektuuri mikroteenuse režiimis, kus on kasutusel Cortex'iga sarnased komponendid [10] [11].

### 3.2.3 Thanos

Thanos on välja töötatud, et lahendada suure hulga andmete pikaajalise säilitamise ja kättesaadavuse probleeme, töötades tihedas koostöös Prometheusega [12]. See lahendus kasutab andmete kompakteerimist ja deduplitseerimist, et optimeerida salvestusruumi kasutust, samal ajal pakkudes kõrget kättesaadavust ja vastupidavust andmekadudele läbi jaotatud andmesalvestussüsteemi [13] [14]. Thanos pakub võimalust hallata andmeid suurel skaalal, integreerides pilvetechnoloogiaid ja -platvorme, nagu AWS ja Google Cloud Storage [15] [16]. Lisaks funktsionaalsele paindlikkusele on Thanos kavandatud toetama kriitilise tähtsusega rakendusi, tagades andmete järjepideva ja usaldusväärse kättesaadavuse [17] [18]. Lisa 4 illustreerib Thanos'e arhitektuuri, kus komponendid [19].

### 3.2.4 Mõõdustiku talletamise tehnoloogiate võrdlus

Selleks, et valida, millist tehnoloogiat kasutada, oli otsustatud teha võrdlustabel ja vaadata, milline tehnoloogia vastab kõige paremini Datel AS nõuetele [20] [15] [21] (Tabel 1).

Tabel 1. Mõõdustiku talletamise tehnoloogiate võrdlus

Omadus/Tehnoloogia	Mimir	Cortex	Thanos
Skaleeritavus	Kõrgelt skaleeritav	Kõrgelt skaleeritav	Kõrgelt skaleeritav
Kogukond ja Tugi	Väga lai kogukond, osa Grafana ökosüsteemist	Toetab lai kogukond	Toetab lai kogukond
Avatud lähtekood	Jah	Jah	Jah
Paigaldamine K8s klastris	Jah	Jah	Jah
Kaugsalvestuse võimalus (S3)	Jah	Jah	Jah

Pikaajaline säilitamine	Jah	Jah	Jah
Pidev arendamine	Jah	Osaliselt	Osaliselt

### 3.2.5 Tehnoloogia valik

Pärast põhjalikku analüüsi ja võrdlust otsustati kasutada Mimir'it, kuna see kõige rohkem vastab ettevõtte ja meeskonna nõuetele. Mimir'i valimise peamised põhjused on selle skaleeritavus ja kuluefektiivsus, mis teeb sellest ideaalse lahenduse ettevõtte praeguste ja tulevaste vajaduste rahuldamiseks. Lisaks pakub Mimir'i tugev integratsioon Prometheus- ja Grafanaga ning võime pakkuda suurt kättesaadavust ja tuge. On oluline ka märkida, et praegu on Mimir uusim võrreldud tehnoloogiatest, mis pidevalt areneb ja täiustub.

## 3.3 Logide talletamise tehnoloogiate analüüs

Logide talletamise tehnoloogiate valik keskendub sellistele kriteeriumidele nagu andmete indekseerimise kiirus, otsingu efektiivsus, halduse lihtsus ja võime integreerida olemasolevate seiresüsteemidega. Loki ja Elasticsearch, laialdaselt kasutatavad lahendused logide kogumiseks, talletamiseks ja analüüsimiseks, on võrdluses. Võrreldavates tehnoloogiates analüüsitakse nende arhitektuuri, suurte andmekogumite töötlemise võime ning halduskulude põhjal. Loki ja Elasticsearch on valitud nende omaduste tõttu, mis võimaldavad kiiret logide indekseerimist, paindlikke otsinguvõimalusi, integreerimist Grafanaga, nende platvormide tugev kogukonnatugi ja pidev arendustegevus.

### 3.3.1 Loki

Loki on horisontaalselt skaleeritav, kõrge kättesaadavusega, mitme rentnikuga logide kogumise süsteem, mis on inspireeritud Prometheusest ja on spetsiaalselt loodud töötama sujuvalt Grafana ökosüsteemiga [6]. Erinevalt teistest logisüsteemidest ei indekseeri Loki terveid logisõnumeid; selle asemel keskendub see identifikaatoritele ja ajatemplitele, indekseerides iga logivoogu ainult siltide kogumi. See lähenemine muudab andmete salvestamise ökonoomsemaks ja päringute täitmise kiiremaks. Loki, mis loodi 2018. aastal Grafana Labs'is, on tuntud oma kulutõhususe ja lihtsa halduse poolest [22] [23]. Süsteemi peamised eelised hõlmavad lihtsat integreerimist Grafanaga, madalaid halduskulusid ning võimalust logisid visuaalselt Grafanas esitada, muutes Loki

atraktiivseks valikuks organisatsioonidele, kes juba kasutavad Prometheus ja Grafana ökosüsteeme [24]. Joonis 1. Kuvab Loki logimiskogum.



Joonis 1. Loki logimiskogum

Loki logimiskogum koosneb kolmest komponendist [22]:

- **Agent** - Agent või klient, mis kraabib logisid, muudab logid voogudeks, lisades silte, ja tõukab vood Loki serverisse läbi HTTP API.
- **Loki** - Peamine server, vastutab logide salvestamise ja päringute töötlemise eest.
- **Grafana** - Logide pärimiseks ja kuvamiseks.

Lisa 5. kuvab Loki arhitektuuri.

Loki komponendid ja nende eesmärgid:

- **Distributor** - Vastutab logide vastuvõtmise ja neid mitmele *ingester*'ile edastamise eest, tagades andmete jaotuse klastris.
- **Ingester** - Hoiab ajutiselt mälus vastuvõetud logiandmeid, enne kui need salvestatakse püsivasse andmelattu, tagades kiire andmesisestuse.
- **Querier** - Vastutab salvestatud logide päringute töötlemise eest, võimaldades kasutajatel teha otsinguid ja analüüsi.
- **Query-Frontend** - Optimeerib ja haldab päringuid, tehes koostööd *querier*'itega, et parandada päringute efektiivsust.
- **Ruler** - Võimaldab defineerida ja rakendada reegleid logide üle, automatiseerides teatud toiminguid põhinedes logide sisule.

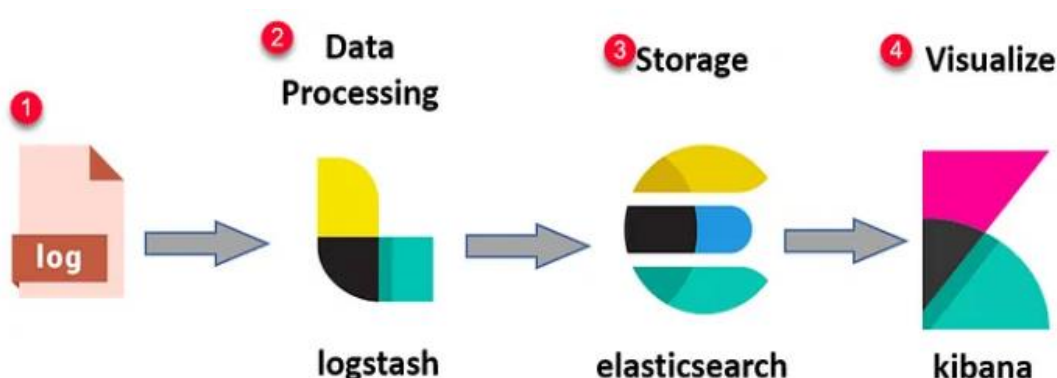
- **Compactor** - Optimeerib salvestusruumi, kompakteerides ja indekseerides logid, et parandada päringute töötlemise kiirust ja vähendada salvestuskulusid.

### 3.3.2 Elasticsearch

Elasticsearch on väga võimas ja paindlik otsingumootor, mis on optimeeritud kiireks tekstipõhiseks otsinguks ja logide analüüsiks reaajas. See võimaldab keerulisi päringuid, andmete filtreerimist ja agregatsiooni. Elasticsearch paistab silma oma võimega toime tulla suurte andmemahtudega, pakkudes samas kiireid ja täpseid otsingutulemusi. Elasticsearch'il on ka lai ökosüsteem, mis sisaldab Kibana logi visualiseerimise tööriista ja Logstash Log Collection Tool'i, pakkudes seega terviklikku lahendust logide haldamiseks [25]. Joonis 2. Elasticsearch'i arhitektuuri.

Elasticsearchi komponendid ja nende eesmärgid:

- **Logs** - Tuvastatakse serverilogid, mida tuleb analüüsida.
- **Logstash** - Kogub logisid ja sündmuste andmeid. Samuti analüüsib ja transformeerib andmeid.
- **ElasticSearch** - Logstashist transformeeritud andmed salvestatakse, otsitakse ja indekseeritakse.
- **Kibana** - Kibana kasutab Elasticsearch andmebaasi, et uurida, visualiseerida ja jagada.



Joonis 2. Elasticsearch'i arhitektuur

### 3.3.3 Logide talletamise tehnoloogiate võrdlus

Selleks, et valida, millist tehnoloogiat kasutada, oli otsustatud teha võrdlustabel ja vaadata, milline tehnoloogia vastab kõige paremini Datel AS nõuetele [26] (Tabel 2).

Tabel 2. Logide talletamise tehnoloogiate võrdlus

Omadus/Tehnoloogia	Loki	Elasticsearch
Põhiline kasutusjuht	Logide kogumine	Täistekstiotsing, logi- ja sündmusandmete salvestamine, analüüs
Avatud lähtekood	Jah	Jah
Skaleeritavus	Kõrgelt skaleeritav	Kõrgelt skaleeritav
Kaugsalvestuse võimalus (S3)	Jah	Ei
Pikaajaline säilitamine	Jah	Jah
Kogukond ja tugi	Väga lai kogukond, osa Grafana ökosüsteemist	Suur kogukond
Paigaldamine K8s klastri	Jah	Jah
Visualiseerimise võimalus Grafanas	Jah	Jah
Päringukeel	LogQL	Kibana Query Language
Integratsioon	Integratsioon Grafana- ja Prometheus'ga	Kibana visualiseerimiseks, Beats ja Logstash andmete kogumiseks
Logide indekseerimine	Loodud hoidma indekseerimist madalal	Indekseerib kõik logiread

### 3.3.4 Tehnoloogia valik

Pärast tähelepanelikku uurimist ja võrdlemist otsustati eelistada Loki't, mis vastab kõige paremini ettevõtte ja meeskonna nõuetele. Loki kasuks otsustamine oli tingitud mitmest võtmetegurist: selle erakordne ühilduvus Kubernetes'iga ja kõrge efektiivsus logide töötlemisel. Loki ja Grafana koostöö pakub mugavaid võimalusi logide visualiseerimiseks ja jälgimiseks, võimaldades meeskonnal andmeid analüüsida. Samuti väärib märkimist Loki ümbruse aktiivne areng ja selle pidev täiustamine, mis tagab juurdepääsu viimastele uuendustele ja uutele funktsioonidele.

## 4 Lahenduse realiseerimine

Peatükk keskendub Datel AS-i valitud seirelahenduse juurutamisele K8s klastris, kus kasutatakse Loki lahendust logide haldamiseks ja Mimirit andmete talletamiseks. Järgnevalt käsitletakse, kuidas neid komponente integreeritakse, et luua tõhus ja skaleeritav seiresüsteem, mis vastab ettevõtte nõudmistele.

Esiteks tuuakse sisse Mimir, mis on valitud mõõdustiku andmete pikaajalise säilitamise ja töötlemise vahendiks. Mimir valiti pärast põhjalikku võrdlust tänu oma skaleeritavusele ja kuluefektiivsusele, mis on ideaalsed Datel AS-i kasvavate vajaduste rahuldamiseks. Mimir pakub tugevat integratsiooni Prometheus ja Grafanaga, mis võimaldab suurendada süsteemi kättesaadavust ja toetust. Mimir on eriti oluline oma võime poolest hallata suuri andmemahte, pakkudes tõhusaid võimalusi ajaloolise andmete analüüsi ja haldamise jaoks.

Seejärel antakse ülevaade Loki kasutamisest logide kogumiseks ja haldamiseks. Loki, mille valik põhines selle ühilduvusel Kubernetes'i keskkonnaga ja tõhususel logide töötlemisel, on oluline vahend logiandmete korraldamiseks ja analüüsiks. See on integreeritud Grafana'ga, mis võimaldab logide visualiseerimist ja jälgimist, toetades sellega meeskonna operatiivset ja analüütilist tööd.

Järgmisena kirjeldatakse tehnilisi samme, mis on vajalikud Loki ja Mimir komponentide paigaldamiseks ja seadistamiseks K8s klastris. Arutletakse paigaldamise protsesside ja konfiguratsioonide üle, mis tagavad lahenduste optimaalse toimivuse ja vastupidavuse. Pärast käsitletakse Grafana seiresüsteemi seadistamist ja paigaldamist, kasutades Helm paketi haldurit. See samm on oluline, et tagada andmete tõhus visualiseerimine ja seire. Grafana pakub võimsaid tööriistu andmevoogude, nagu Loki logid ja Mimir'iga hallatav mõõdustiku andmed, visuaalseks analüüsiks ja jälgimiseks, mis on hädavajalik kiirete ja informeeritud otsuste tegemiseks.

## 4.1 Mõõdustiku talletamise tehnoloogia paigaldamine

Peatükk keskendub Mimir tehnoloogia paigaldusprotsessile Datel AS-i Kubernetes klastris, kasutades Gitlab CI/CD ja Helm *chart*i. Antakse ülevaate tehnilistest etappidest, mis on vajalikud Mimir süsteemi edukaks paigaldamiseks ja konfigureerimiseks, tagades, et see vastab Datel AS-i operatsioonilistele nõuetele ja eesmärkidele [27].

### 4.1.1 Mimir'i paigaldamisprotsess

Mimir pakub mitmeid töörežiime, nagu ühe instantsi (*single-instance*) ja mikroteenuste (*microservices*) režiimid [10]. Suurema skaleeritavuse ja efektiivsema halduse eesmärgil valiti mikroteenuste režiim. See režiim lubab Mimir komponente eraldi hallata, mis võimaldab paremat koormuse jaotust ja tõhusamat andmetöötlust suure hulga meetrikate puhul. Mikroteenuste režiim pakub võimalust süsteemi paindlikult laiendada, lisades vajadusel rohkem ressursse, mis on kasulik nii suurenevatele ettevõtetele, kui ka arenevatele projektidele. Allpool on loetletud käsud Mimir'i paigaldamise jaoks. Lisa 6 kuvab Mimir'i helm *chart*'i näidet.

1. Grafana helm *chart* repositooriumi lisamine:

```
helm repo add grafana https://grafana.github.io/helm-charts
```

2. Repositooriumite värskendamine, et tagada kõige uuemate versioonide kättesaadavust:

```
helm repo update
```

3. Mimir'i paigaldamine, kasutades *values.yaml* faili:

```
helm upgrade -i mimir grafana/mimir-distributed -f values.yaml --create-namespace -n mimir
```

### 4.1.2 Mimir'i kollektori paigaldamine

Mimir vajab meetrikate kogumiseks kollektorit, mille valikul on mitmeid võimalusi, nagu Grafana Agent, Prometheus agent ja Telegraf. Igäihel neist on oma spetsiifilised eelised, kuid Mimir'i jaoks valiti Prometheus agent. Erinevalt tavalisest Prometheus serverist, mis kogub ja salvestab metriikate andmed, töötab Prometheus agent režiimis, kus see keskendub peamiselt andmete kogumisele ja edastamisele, mitte salvestamisele. See muudab agent režiimi ressursisäästlikumaks ja vähem koormavaks. Allpool on loetletud käsud Prometheus agent'i paigaldamise jaoks.

1. Prometheus'e helm *chart* repositooriumi lisamine:

```
helm repo add prometheus-community https://prometheus-  
community.github.io/helm-charts
```

2. Repositooriumite värskendamine, et tagada kõige uuemate versioonide kättesaadavust:

```
helm repo update
```

3. Prometheus'e paigaldamine, kasutades *values.yaml* faili:

```
helm upgrade -i prometheus prometheus-community/prometheus -f values.yaml --  
create-namespace -n prometheus
```

## 4.2 Tsentraalse logimislahenduse paigaldamine

Peatükk keskendub Loki tehnoloogia paigaldusprotsessile Datel AS-i Kubernetes klastris, kasutades Gitlab CI/CD ja Helm *chart*'i. Kirjeldatakse Loki paigaldamis protsessi tehnilistest sammudest alates Helm *chart*'i ettevalmistamisest kuni süsteemi käivitamiseni.

### 4.2.1 Loki paigaldamisprotsess

Loki pakub mitmeid töörežiime, sealhulgas ühe instantsi (*single-instance*) ja mikroteenuste (*microservices*) režiime [22]. Suurema skaleeritavuse ja paindlikkuse tagamiseks valiti mikroteenuste režiim. See režiim võimaldab Loki komponente eraldi hallata, võimaldades seeläbi tõhusamat koormuse jaotamist ja andmetöötlust suurel hulgal logide puhul. Mikroteenuste režiimis saab süsteemi hõlpsasti laiendada, lisades rohkem ressursse vastavalt vajadusele, mis teeb selle ideaalseks ettevõttele ja kiiresti projektile [28]. Lisa 7 näitab Loki helm *chart*'i näidet.

Allpool on loetletud käsud Loki-distributed paigaldamise jaoks.

1. Grafana helm *chart* repositooriumi lisamine:

```
helm repo add grafana https://grafana.github.io/helm-charts
```

2. Repositooriumite värskendamine, et tagada kõige uuemate versioonide kättesaadavust:



```
helm repo update
```

3. Loki paigaldamine, kasutades *values.yaml* faili:

```
helm upgrade -i loki grafana/loki-distributed -f values.yaml --create-namespace -n loki
```

#### 4.2.2 Loki kollektori paigaldamine

Grafana Loki on logide haldussüsteem, mis vajab logide kogumiseks kollektorit. Selleks on mitmeid võimalusi, sealhulgas Fluentd, Logstash ja Promtail. Promtail valiti peamiselt seetõttu, et see on välja töötatud ja optimeeritud just Loki jaoks, pakkudes tihedat integreeritust ja lihtsustades seadistamist. Promtail jälgib lokaalseid logifaile ja edastab need Loki serveritele. See tööriist suudab tõhusalt töödelda suures koguses logisid, lisades metainfot ja toetades logide filtreerimist. Allpool on loetletud käsud Promtail'i paigaldamise jaoks.

1. Grafana helm *chart* repositooriumi lisamine:

```
helm repo add grafana https://grafana.github.io/helm-charts
```

2. Repositooriumite värskendamine, et tagada kõige uuemate versioonide kättesaadavust:

```
helm repo update
```

3. Promtail'i paigaldamine, kasutades *values.yaml* faili:

```
helm upgrade -i promtail grafana/promtail -f values.yaml --create-namespace -n promtail
```

### 4.3 Grafana seiresüsteemi seadistamine ja paigaldamine

Grafana on analüüsi- ja interaktiivse visualiseerimise platvorm [29]. See pakub rikkalikku valikut diagramme, graafikuid ja hoiatusi ning ühendub paljude toetatud andmeallikatega nagu Prometheus, ajaseeriad andmebaasid või tuntud RDBMS-id. See võimaldab päringuid teha, visualiseerida ja luua hoiatusi oma mõõdikute kohta sõltumata sellest, kus need on salvestatud [30] [31].

#### 4.3.1 Grafana seiresüsteemi paigaldamisprotsess

Grafana paigaldamiseks kasutakse

1. Grafana helm *chart* repositooriumi lisamine:

```
helm repo add grafana https://grafana.github.io/helm-charts
```

2. Repositooriumite värskendamine, et tagada kõige uuemate versioonide kättesaadavust:

```
helm repo update
```

3. Grafana seiresüsteemi paigaldamine, kasutades *values.yaml* faili:

```
helm upgrade -i grafana grafana/Grafana -f values.yaml --create-namespace -n grafana
```

### 4.3.2 Postgres andmebaasi paigaldamisprotsess

PostgreSQL, tuntud ka kui Postgres, on avatud lähtekoodiga objekt-relatsiooniline andmebaasisüsteem, mis on tuntud oma jõudluse, skaleeritavuse ja paindlikkuse poolest. See pakub mitmeid arenenud funktsioone nagu tehingute tugi ja talletatud protseduurid. PostgreSQL sobib hästi kasutamiseks koos Grafana'ga. Allpool on loetletud käsud Postgres'i paigaldamise jaoks [32] [33].

1. Postgres andmebaasi paigaldamine, kasutades *values.yaml* faili:

```
helm upgrade -i postgresql oci://registry-1.docker.io/bitnamicharts/postgresql -f values.yaml --create-namespace -n postgresql
```

Kõik ülaltoodud sammud on lisatud GitLab CI/CD pipeline'idesse. Selleks on loodud *.gitlab-ci.yml* fail, et käskudel oleks oma roll *pipeline*'is erinevates etappides.

## 4.4 Grafana seadistamine

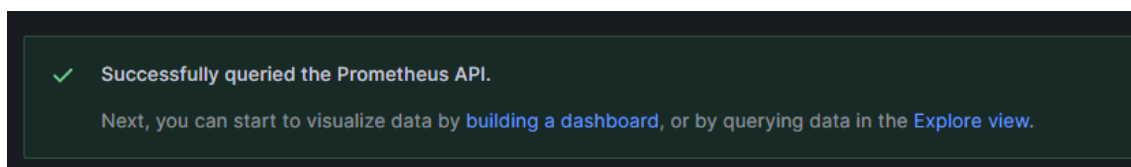
Enne alustamist, tuleb märkida, et andmeallikate hulka on oluline lisada Mimir ja Loki. Grafana pakub paindlikku süsteemi erinevate andmeallikate integreerimiseks.

### 4.4.1 Andmeallika lisamine

Grafana toetab erinevaid andmeallikaid läbi "*Data sources*" sektsiooni, kus on võimalik Prometheus tüüpi andmeallikaid hõlpsasti hallata. Mimir'i andmeallikaga töötamisel on keskne osa selle konfigureerimisel, mille käigus tuleb täita Mimir'i URL, andmeallika tüüp ning Mimir'i versiooni väljad. Ühenduse edukuse kindlustamiseks pakub süsteem

"Save & Test" funktsionaalsust, mis kontrollib sisestatud andmete õigsust ja tagab andmeallika nõuetekohase toimimise.

Loki puhul tuleb sarnased sammud läbi teha, "Data sources" sektsioonis on vaja täpsustada loki URL ning vajutada nuppu "Save and Test" Joonis 3. Näitab eduka andmeallika lisamist graffanas.



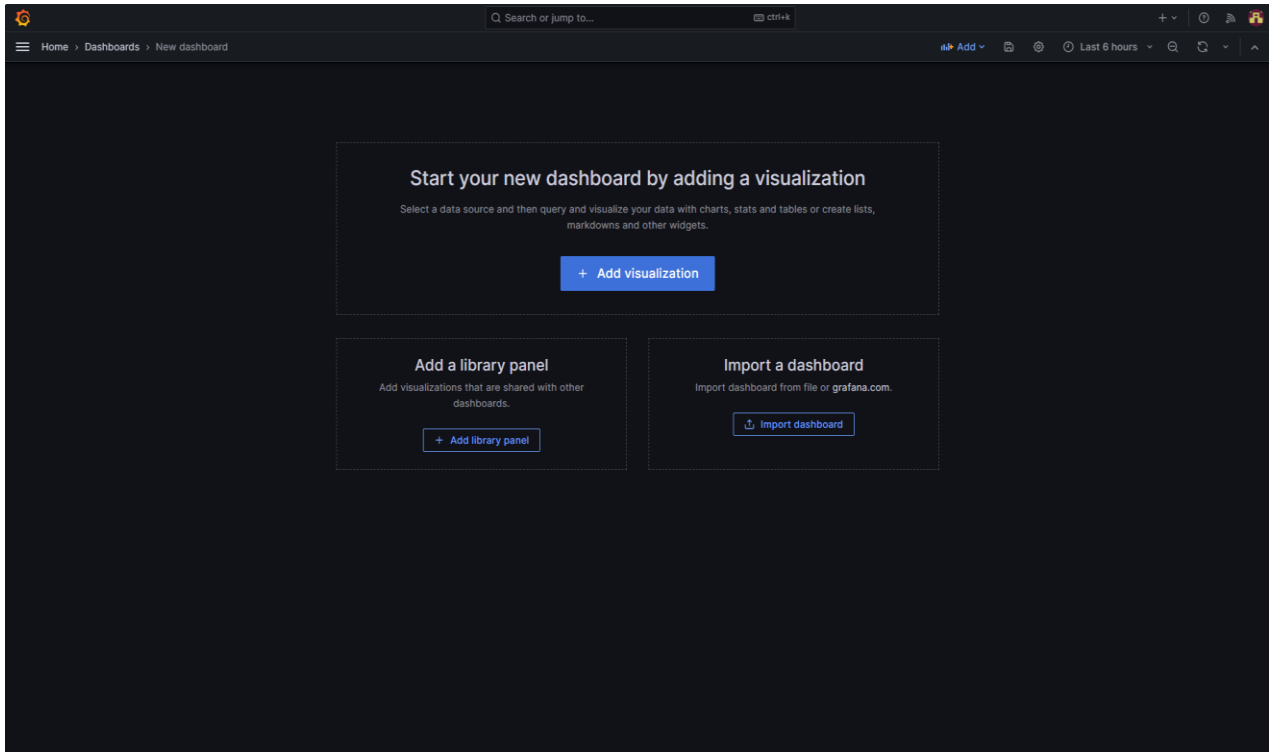
Joonis 3. Andmeallika lisamine graffanas

#### 4.4.2 Grafana seirevaade seadistamine

Seirevaate loomisel on andmeallika eelnev lisamine ja testimine kriitilise tähtsusega, et tagada seiretava info reaalajas kuvamine. Pärast Mimir andmeallika edukat integreerimist avaneb võimalus liikuda edasi seirevaate seadistamisega. Selles faasis kujundatakse seirevaade, mis koondab endas paneelid, näiteks protsessori tuumade arvu, süsteemi tööaja, operatiivmälu kasutuse ning teisi kriitilisi näitajaid kajastavaid graafikuid. Grafana dünaamiline süsteem toetab erinevate visualiseerimisviiside rakendamist, mis võimaldab kasutajatel luua joon-, tulp- ja ringdiagramme, et edastada andmeid selgelt ja mõistetavalt.

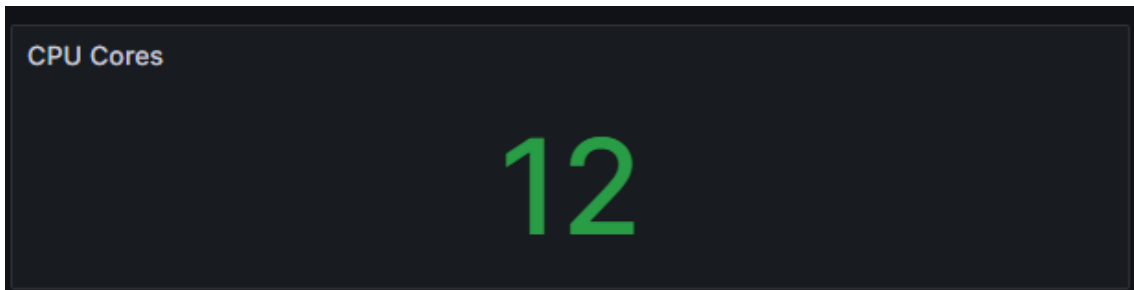
#### 4.4.3 Seire-paneelide lisamine ja seadistamine

Uue paneeli lisamisel Grafana seirevaatesse on "Add visualization" nupp, mis juhatab kasutajat paneelide loomise protsessi. Joonis 4. kuvab grafana seirepaneeli loomist. Oluline on paneelidele päringute genereerimine, mis on kohandatud vastavalt jälgitavatele meetrikatele. Näiteks protsessori tuumade arvu visualiseerimiseks kasutatakse spetsiifilist päringut, mis arvutab protsessori tuumade koormust süsteemi erinevates olukordades. Selline lähenemine võimaldab jälgida mitte ainult üksikute protsessori tuumade, vaid kogu süsteemi üldist jõudlust. Seirevaaded olid osaliselt imporditud ametlikust Grafana veebilehelt ning uuendatud vastavalt nõuetele [34].



Joonis 4. Grafana seirepaneeli loomine

Joonis 5. Näitab mitu protsessori tuuma on olemas serveril.



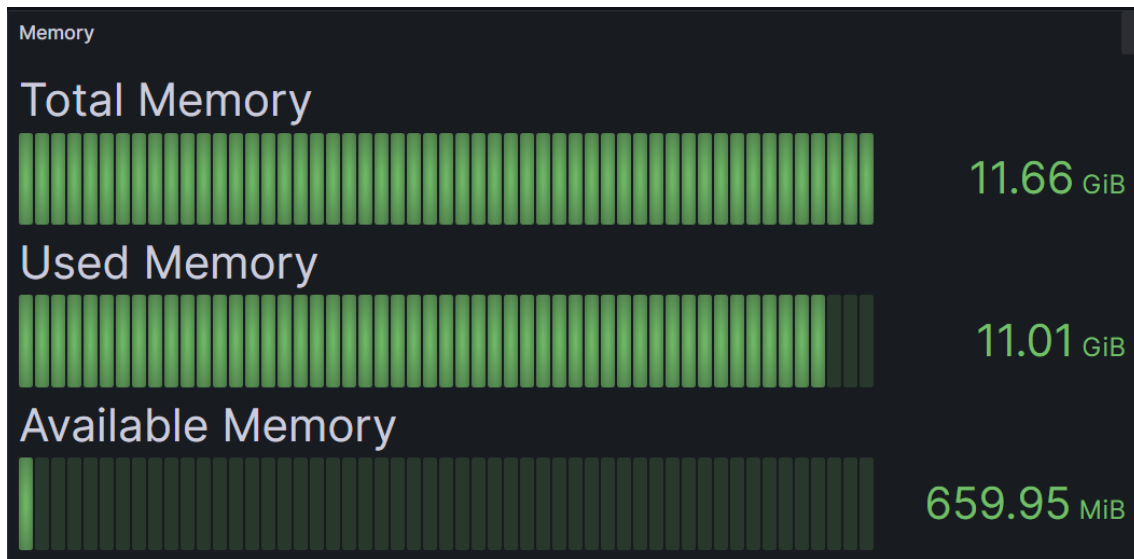
Joonis 5. Protsessori tuumade arv serveris

Joonis 6. Kuvab paneeli, mis näitab süsteemi töötamisaega.



Joonis 6. Süsteemi töötamise aeg

Joonis 7. Kuvab paneeli, mis nätab operatiivmälu kasutust, kui palju on üldse operatiivmälu ja kui palju operatiivmälu on vaba.



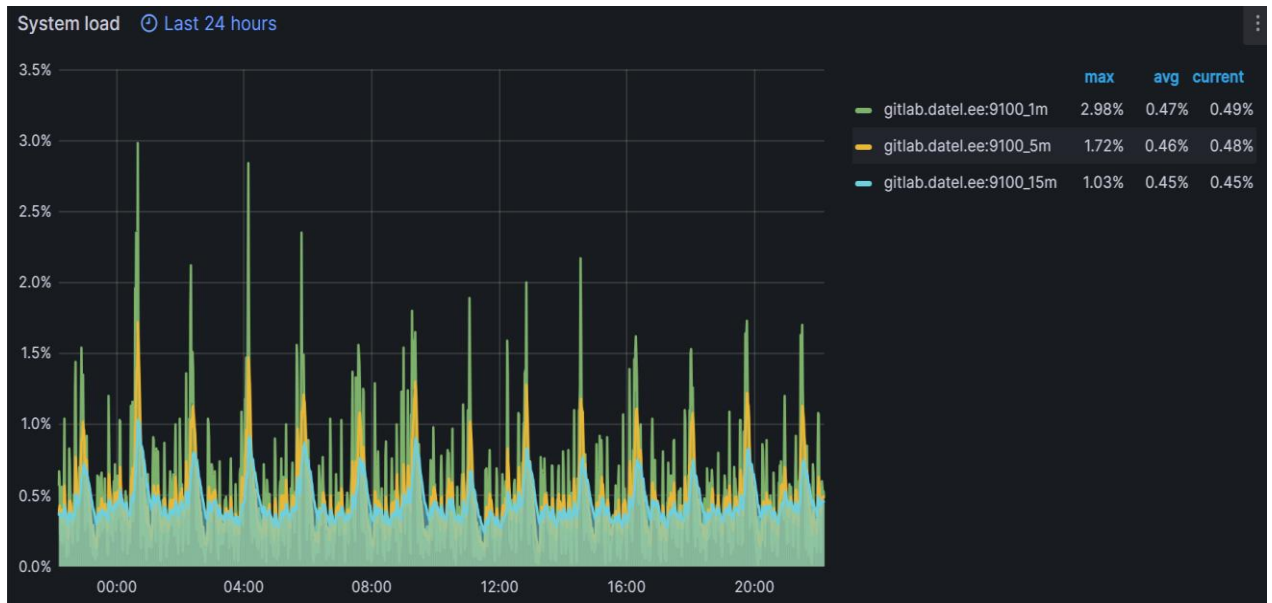
Joonis 7. Serveri operatiivmälu kasutuse graafik

Joonis 8. Kuvab paneeli, mis näitab kettaruumi kasutust.

Disk Space Used Basic(EXT4/XFS)						
Filesystem	IP	Mounted on	Size	Available	Used Space	Used
ext4	gitlab.datel.ee:9100	/opt	39.2 GiB	24.3 GiB	13.1 GiB	33.30%
ext4	gitlab.datel.ee:9100	/	191.3 GiB	143.1 GiB	40.3 GiB	21.07%
ext4	gitlab.datel.ee:9100	/boot	973.4 MiB	696.5 MiB	209.7 MiB	21.55%

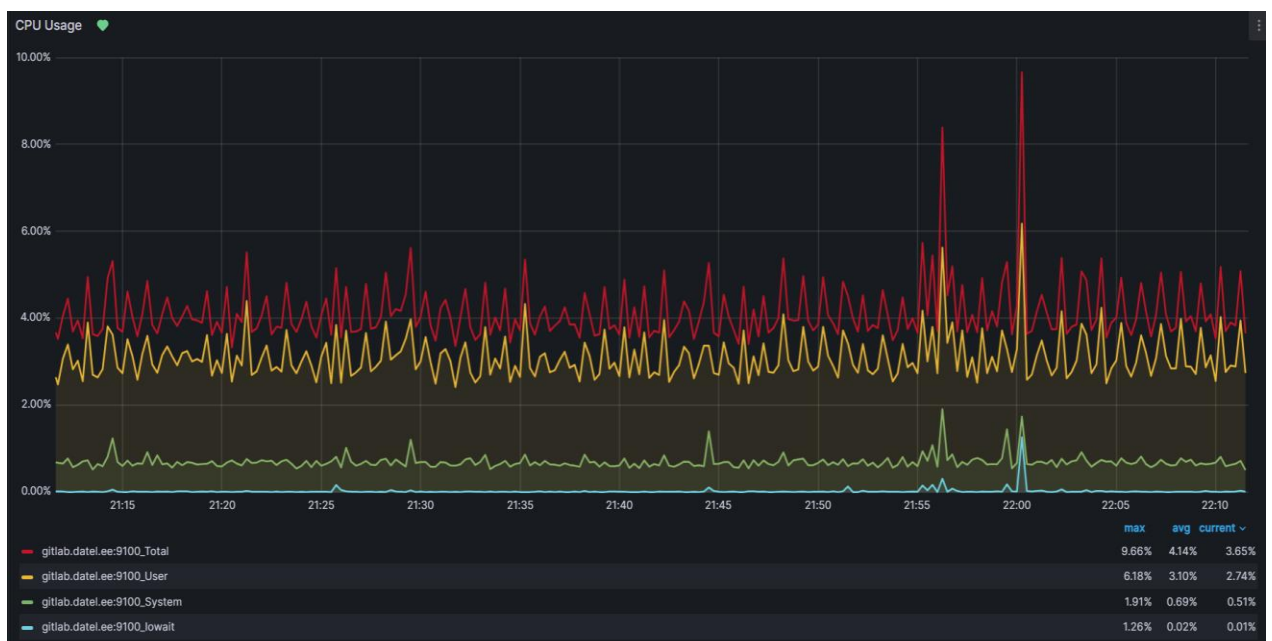
Joonis 8. Paneel, mis näitab serveri kettaruumi kasutust

Joonis 9. Kuvab paneeli, mis näitab süsteemi koormust graafikuna, kus on kolm peamist graafikut, süsteemi koormus ühe minuti jooksul, viie minuti jooksul ja 15 minuti jooksul.



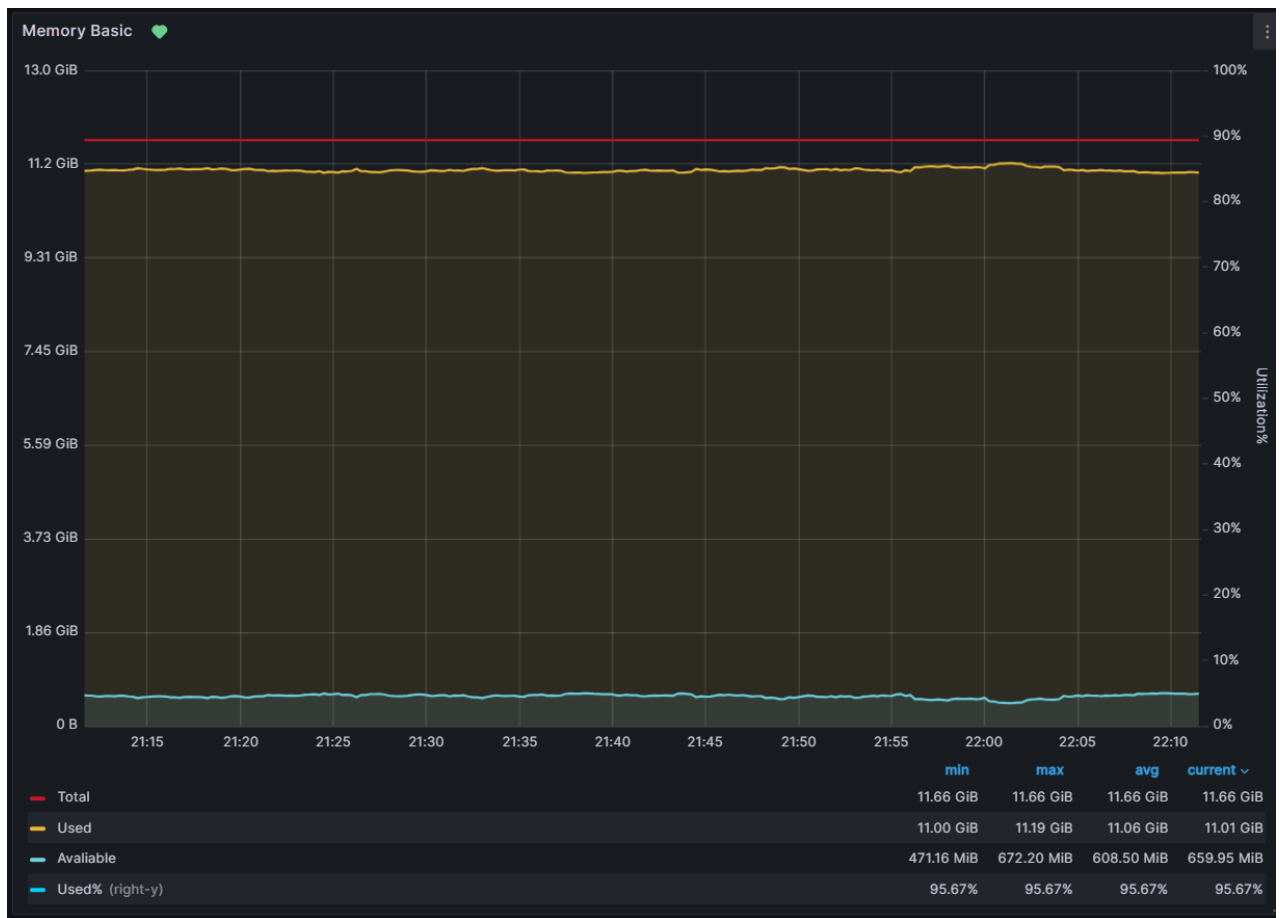
Joonis 9. Serveri süsteemi koormuse paneel graafikuna

Joonis 10. Kuvab paneeli, mis näitab protsessori kasutust graafikuna.



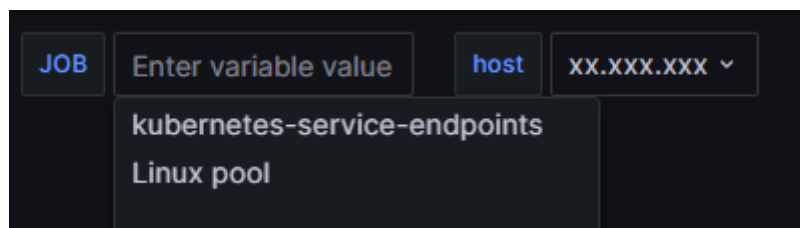
Joonis 10. Protsessori kasutamine graafikuna

Joonis 11. Kuvab paneeli, mis näitab serveri operatiivmälu kasutust graafikuna.



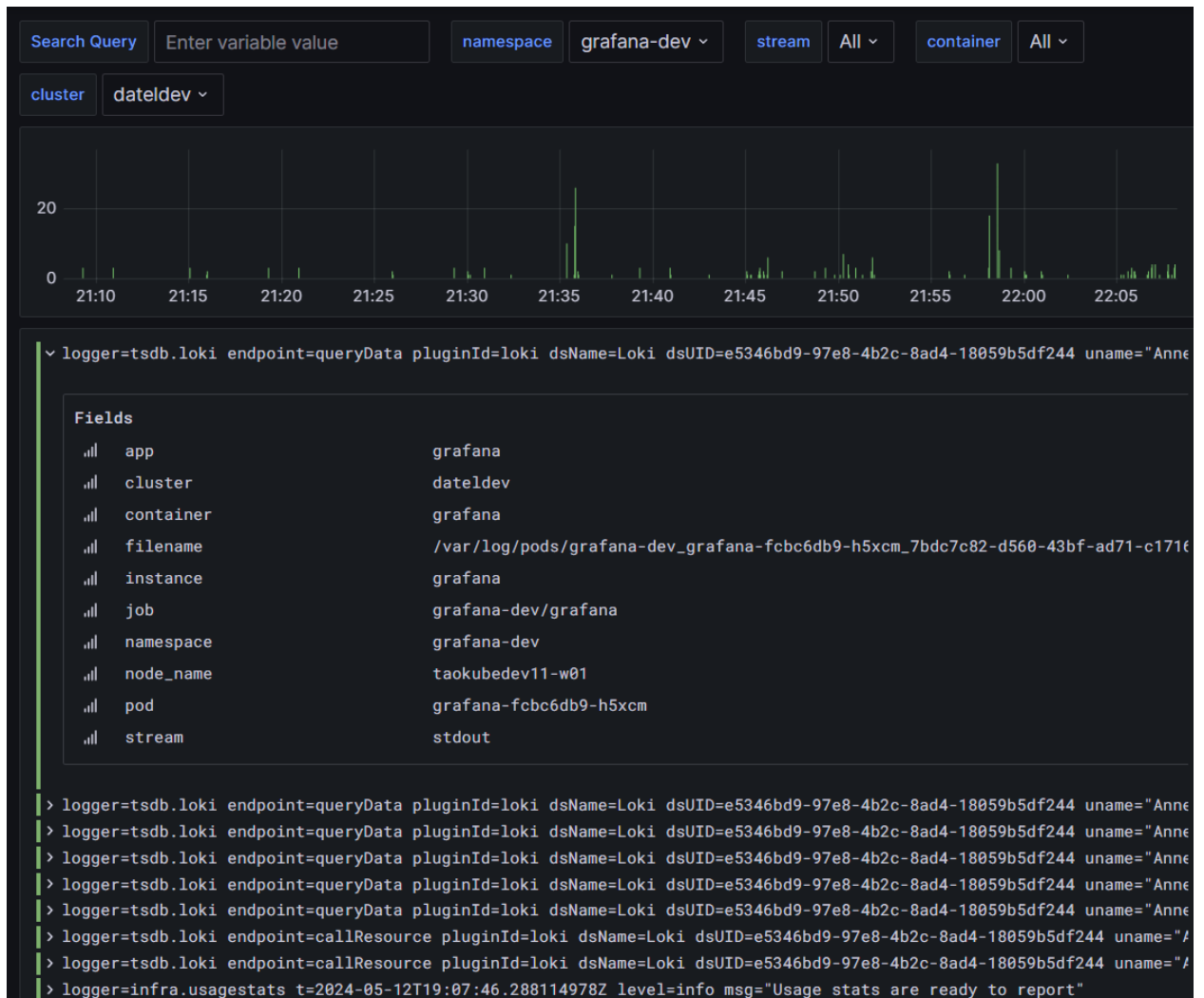
Joonis 11. Serveri operatiivmälu kasutamine graafikuna

Lisaks on seirepaneeli jaoks tehtud muutujad et oleks võimalus kuvada infot sama paneelides aga erinevate serverite kohta. Joonis 12. Näitab serveri valikut seirevaade jaoks.



Joonis 12. Serveri valimine seirevaade jaoks

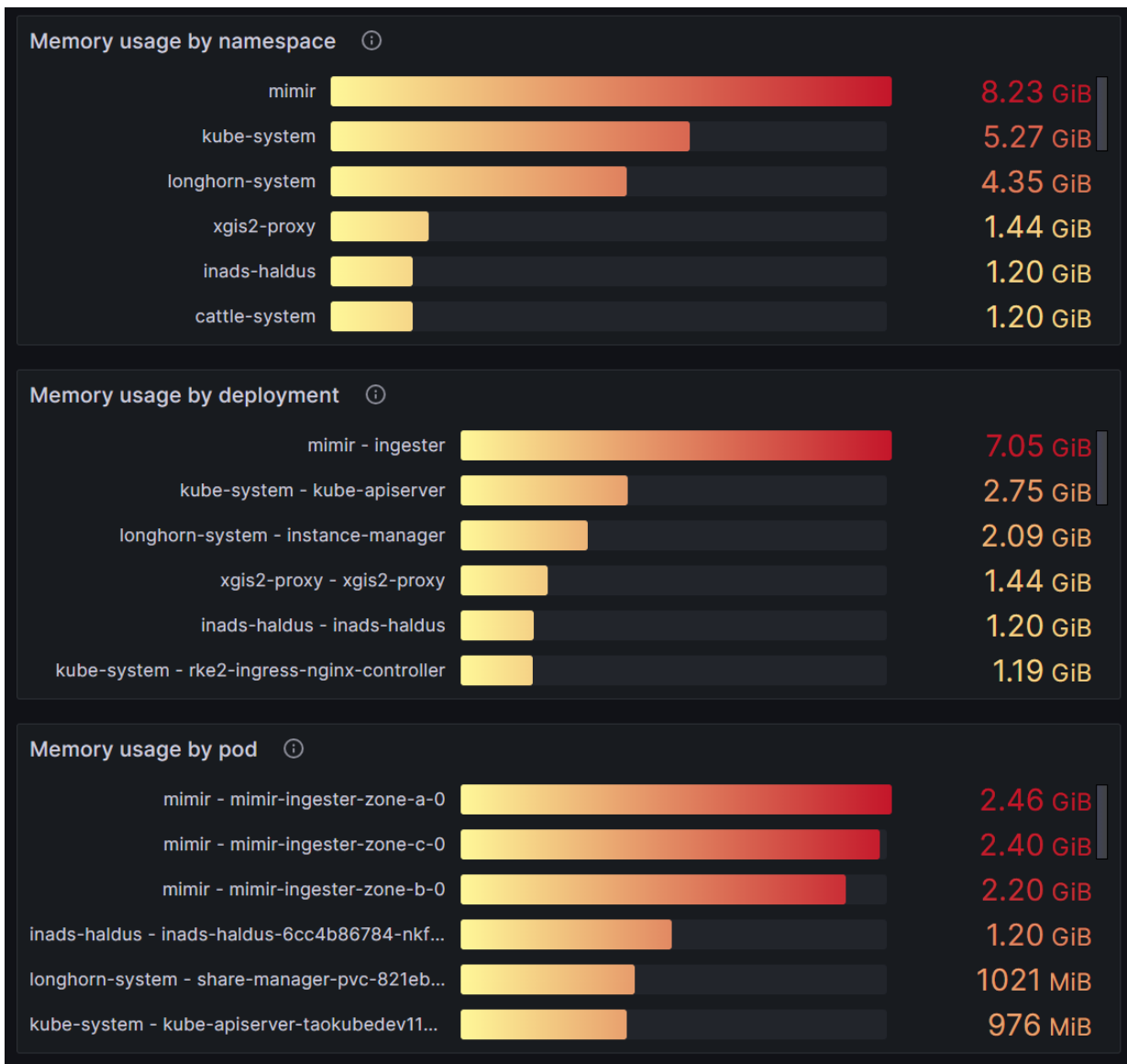
Loki jaoks on tehtud eraldi seirevaade, mis annab võimaluse K8s klasteri logid filtreerida ja vaadata. Joonis 13. Kuvab logide seirevaadet.



Joonis 13. Logide seirevaade

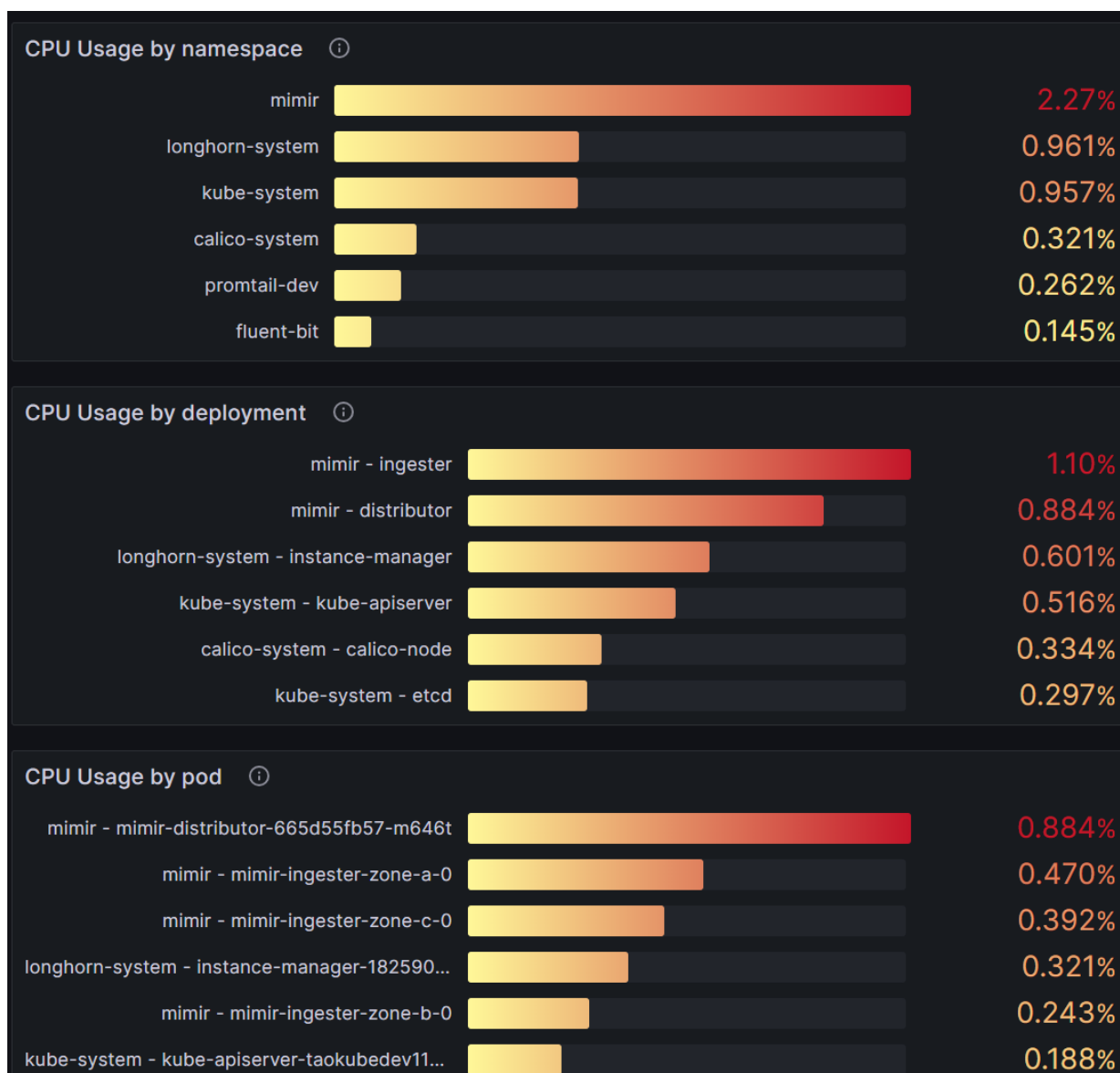
Kubernetes'e jaoks on loodud seirevaade, mis näitab operatiivmälu ja protsessori kasutust iga nimeruumi, juurutuse ja *pod*'i kohta, lihtsustades ressursside jälgimist. Joonis 14. Kuvab klasteri operatiivmälu kasutust.





Joonis 14. Klastri operatiivmälu kasutus

Joonis 15. Klastri protsessori kasutust.



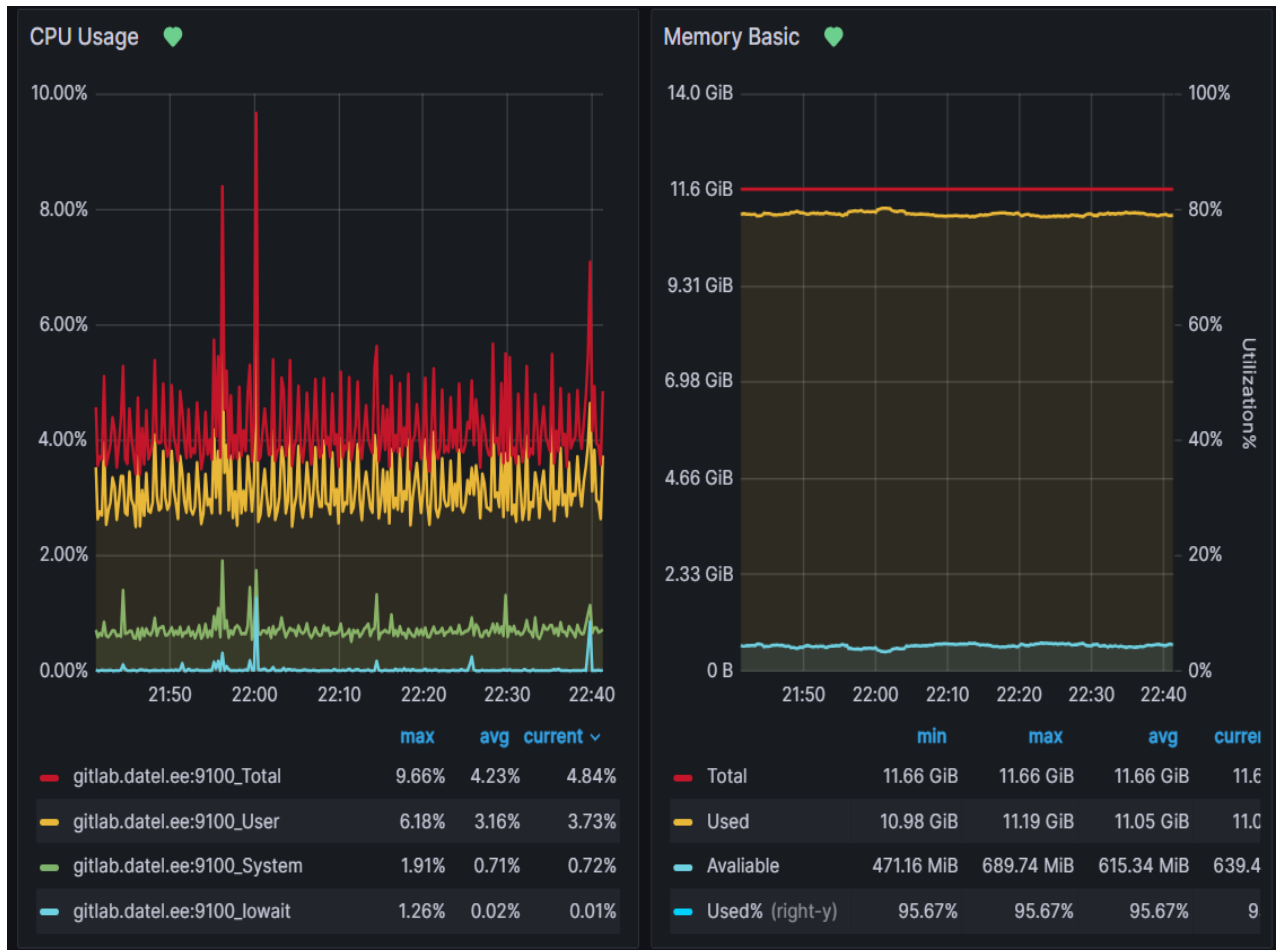
Joonis 15. Klastri protsessori kasutus

## 4.5 Grafana teavituste seadistamine

Selleks, et luua Grafanas efektiivseid teavitusi, tuleb kasutada ajareaga graafikut (*Time Series Graph*). Antud juhul loodi kaks kriitilise tähtsusega teavitus, mis on seotud protsessori koormuse ja operatiivmälu kasutusega. Teavituse loomise protsess algab sellest, et tuleb graafikul vajutada *Alerts* nupule. Seejärel tuleb seadistada teavitus vastavalt vajalikele parameetritele. Näiteks oli sel juhul määratud, et hoiatus saadetakse, kui protsessori kasutus ületab 80 protsenti. Operatiivmälu puhul aga saadetakse hoiatus, kui kasutus on 17 GB 20-st võimalikust. Lisaks sellele dubleeritakse kõik hoiatused

automaatselt ka Teamsi kanalisse, et teavitada projekti meeskonna olukorrast. See võimaldab kiiresti reageerida võimalikele süsteemi ülekoormustele või muudele tehnilistele probleemidele.

Kui teavitused on edukalt seadistatud, siis Grafana seirevaate paneelil saab näha rohelist südame, mis näitab, et teavitus toimib korrektselt. Joonis 16. Kuvab kahte seirevaade paneele, millele on tehtud teavitused.



Joonis 16. Teavitused paneelide jaoks

## 5 Tulemuste analüüs

Püstitatud seire- ja logimissüsteem koosneb Mimir ja Grafana Loki komponentidest, mis vastutavad vastavalt mõõdustikute kogumise ja logimise eest, salvestades andmed S3 hoidlasse. Kõik osad on ühendatud Grafana'ga, mis tagab nii mõõdustikute, kui ka logide visualiseerimist. Lisa 8. kuvab loodud arhitektuuri.

Uue seiresüsteemi teavituste kaudu on võimalik varasemaid intsidente kiiresti tuvastada ja neid ennetada. Grafana seirevaate ja teavituste abil on loodud reaalajas toimiva ülevaate kriitilistest sündmustest, mis võimaldab probleemidele ennetavalt reageerida. Samuti loodud seirevaated kuvavad serverite ressursikasutust, nagu operatiivmälu ja protsessori koormus.

Võrreldes intsidendiga, kus ressursi ülekoormusest tingitud süsteemi kokkukukkumise tõttu katkesid tööprotsessid ja süsteemi taastamine võttis aega umbes 20 tundi, tekitades ligikaudu 4500 euro suurust kahju, on uus süsteem loodud selliselt, et sarnaseid olukordi oleks võimalik õigeaegselt märgata ja vältida, vähendades seeläbi märkimisväärselt võimalikku kahju. Täiustatud seire- ja logimissüsteemi kasutuselevõtt on märkimisväärne samm ettevõtte IT-infrastruktuuri haldamise efektiivsuse tõstmisel.

## 6 Kokkuvõte

Käesoleva bakalaaurusetöö eesmärgiks oli välja töötada ja seadistada Datel AS-i vajadustele vastav seire- ja logimissüsteem. Töö keskendus tõhusate ja sobivate tehnoloogiliste lahenduste valimisele ja nende võrdlemisele ja reaalajas seire ja logimise võimaldamisele. Eriti rõhutati, et süsteem oleks avatud lähtekoodiga ja tagades kasutajaliidese intuiitiivsust ja kasutusmugavust, mis tagab IT-meeskonna efektiivse töö süsteemiga.

Praktilises osas kirjeldati samm-sammult lahenduse loomist ja seadistamist, sealhulgas Mimir ja Loki tehnoloogiate paigaldamist ning Grafana seiresüsteemi paigaldamist ja integreerimist. Eesmärkide saavutamise protsess hõlmas seire- ja logimistarkvarade põhjalikku analüüsi, valikut ja integreerimist, mis tagas süsteemi optimaalse jõudluse ja vastavuse ettevõtte nõuetele. Lahenduse analüüs kinnitas, et süsteem vastab kõikidele ettevõtte nõuetele ning on edukalt integreeritud ettevõtte infrastruktuuriga, parandades süsteemi stabiilsust ja tagades projektide sujuva toimimise.

Konkreetsena, mis põhines süsteemirikele reageerimise aja ja kulude vähendamisele, näitas lahendus olulist paranemist. Algne olukord, kus süsteemi taastamiseks kulus umbes 20 töötundi ja 4500 eurot, optimeeriti varajase avastamise ja sekkumise kaudu, mis vähendas sarnaste intsidentide lahendamiseks kuluvat aega ja kulusid märkimisväärselt. See tõstis IT-infrastruktuuri usaldusväärsust ja tõi kaasa kokkuhoiu, mis parandas üldist projekti ja äri efektiivsust.

Kokkuvõttes on loodud seirelahendus, logide ja mõõdustiku talletamise tehnoloogiad efektiivsed, tagades Datel AS-i süsteeme töökindluse ja kättesaadavuse. Lahendused on näidanud end oluliste vahenditena süsteemide seisundi ja toimimise reaalajas jälgimisel, pakkudes vajalikku toe ning suurendades ettevõtte projektide juhtimise edukust.

## Kasutatud kirjandus

- [1] H. BARANES, „5 Reasons Why You Need Monitoring Tools,“ SysAid, 8 Oktoober 2019. [Võrgumaterjal]. Available: <https://www.sysaid.com/blog/service-desk/5-reasons-why-you-need-monitoring-tools>. [Kasutatud 8 Mai 2024].
- [2] A. S. Mahapatra, „Setting up Service Monitoring — The Why’s and What’s,“ Medium, 22 Mai 2021. [Võrgumaterjal]. Available: <https://amitosh.medium.com/the-whys-and-what-s-of-setting-up-service-monitoring-cc1c165ee088>. [Kasutatud 9 Mai 2024].
- [3] DATEL, „30+ years of empowering smart communities,“ AS DATEL, 2024. [Võrgumaterjal]. Available: <https://www.datel.ee/et/>. [Kasutatud 25 Märts 2024].
- [4] Cortex, „Documentation,“ Cortex, [Võrgumaterjal]. Available: <https://cortexmetrics.io/docs/>. [Kasutatud 20 Märts 2024].
- [5] Cortex, „Cortex Architecture,“ Cortex, [Võrgumaterjal]. Available: <https://cortexmetrics.io/docs/architecture/>. [Kasutatud 2 Aprill 2024].
- [6] J. Elliott, F. Liao, C. Toven, J. Villa ja E. Welch, „LGTM: Scale observability with Mimir, Loki, and Tempo,“ Grafana Labs, 2 November 2022. [Võrgumaterjal]. Available: <https://grafana.com/go/observabilitycon/2022/lgtm-scale-observability-with-mimir-loki-and-tempo/>. [Kasutatud 1 Aprill 2024].
- [7] Grafana Labs, „Mimir overview,“ Grafana Labs, [Võrgumaterjal]. Available: <https://grafana.com/oss/mimir/>. [Kasutatud 20 Märts 2024].
- [8] M. Tan, „A year in Mimir: Massive scale, new metrics formats, increased adoption,“ Grafana Labs, 27 Märts 2023. [Võrgumaterjal]. Available: <https://grafana.com/blog/2023/03/27/a-year-in-mimir-massive-scale-new-metrics-formats-increased-adoption/>. [Kasutatud 23 Märts 2024].
- [9] T. Wilkie, „Announcing Grafana Mimir, the most scalable open source TSDB in the world,“ Grafana Labs, 30 Märts 2022. [Võrgumaterjal]. Available: <https://grafana.com/blog/2022/03/30/announcing-grafana-mimir/>. [Kasutatud 4 Aprill 2024].
- [10] Grafana Labs, „Grafana Mimir deployment modes,“ Grafana Labs, [Võrgumaterjal]. Available: <https://grafana.com/docs/mimir/latest/references/architecture/deployment-modes/>. [Kasutatud 2 Aprill 2024].
- [11] Grafana Labs, „Grafana Mimir architecture,“ Grafana Labs, [Võrgumaterjal]. Available: <https://grafana.com/docs/mimir/latest/get-started/about-grafana-mimir-architecture/>. [Kasutatud 4 Aprill 2024].
- [12] Last9, „What is Thanos and How Does it Scale Prometheus?,“ Last9, 12 Oktoober 2023. [Võrgumaterjal]. Available: <https://last9.io/blog/what-is-thanos-and-how-does-it-scale-prometheus/>. [Kasutatud 24 Märts 2024].
- [13] „Quick Tutorial,“ Thanos, [Võrgumaterjal]. Available: <https://thanos.io/tip/thanos/quick-tutorial.md/>. [Kasutatud 4 Aprill 2024].

- [14] OpsRamp, „Scaling Prometheus Using Thanos,“ OpsRamp, [Võrgumaterjal]. Available: <https://www.opsramp.com/guides/prometheus-monitoring/prometheus-thanos/>. [Kasutatud 5 Aprill 2024].
- [15] I. Horak, „5 recommendations when running Thanos and Prometheus,“ zapier, 13 Jaanuar 2023. [Võrgumaterjal]. Available: <https://zapier.com/blog/five-recommendations-when-running-thanos-and-prometheus/>. [Kasutatud 29 Märts 2024].
- [16] A. Dunand, „Thanos : Long-Term Storage of Prometheus Metrics,“ enix, 4 Oktoober 2023. [Võrgumaterjal]. Available: <https://enix.io/en/blog/prometheus-thanos/>. [Kasutatud 8 Märts 2024].
- [17] SYSRANT, „Getting started with Prometheus and Thanos,“ SYSRANT, 29 Juuli 2020. [Võrgumaterjal]. Available: <https://sysrant.com/posts/getting-started-with-prometheus-and-thanos-in-kubernetes/>. [Kasutatud 3 Aprill 2024].
- [18] Last9, „Prometheus vs Thanos,“ Last9, 12 Juuni 2023. [Võrgumaterjal]. Available: <https://last9.io/blog/prometheus-vs-thanos/>. [Kasutatud 3 Aprill 2024].
- [19] Thanos, „Architecture Overview,“ Thanos, [Võrgumaterjal]. Available: <https://thanos.io/v0.6/thanos/getting-started.md/#architecture-overview>. [Kasutatud 1 Aprill 2024].
- [20] B. C. Gain, „The Great Grafana Mimir and Cortex Split,“ TheNewStack, 10 August 2022. [Võrgumaterjal]. Available: <https://thenewstack.io/the-great-grafana-mimir-and-cortex-split/>. [Kasutatud 9 Märts 2024].
- [21] S. Khan, „Thanos vs Cortex,“ Last9, 16 Detsember 2022. [Võrgumaterjal]. Available: <https://last9.io/blog/thanos-vs-cortex/>. [Kasutatud 11 Märts 2024].
- [22] Grafana Labs, „Loki overview,“ Grafana Labs, 2022. [Võrgumaterjal]. Available: <https://grafana.com/docs/loki/latest/get-started/overview/>. [Kasutatud 4 Aprill 2024].
- [23] R. Kadikar, „Log Monitoring and Alerting with Grafana Loki,“ InfraCloud, 2 Juuni 2020. [Võrgumaterjal]. Available: <https://www.infracloud.io/blogs/grafana-loki-log-monitoring-alerting/>. [Kasutatud 29 Märts 2024].
- [24] E. Welch ja C. Toven, „Introduction to Loki: Like Prometheus, but for Logs,“ Grafana Labs, 2019. [Võrgumaterjal]. Available: <https://grafana.com/go/webinar/intro-to-loki-like-prometheus-but-for-logs/>. [Kasutatud 2 Aprill 2024].
- [25] K. K. R, „ELK Stack Architecture and How Does It Works,“ medium, 29 Oktoober 2022. [Võrgumaterjal]. Available: <https://medium.com/yavar/elk-stack-architecture-and-how-does-it-works-bda6382b3c83>. [Kasutatud 1 Aprill 2024].
- [26] A. Anand, „Loki vs Elasticsearch - Which tool to choose for Log Analytics?,“ SigNoz, 22 Jaanuar 2024. [Võrgumaterjal]. Available: <https://signoz.io/blog/loki-vs-elasticsearch/#:~:text=The%20biggest%20difference%20between%20Loki,more%20cost%2Deffective%20and%20performant..> [Kasutatud 23 Märts 2024].
- [27] Grafana Labs, „Deploy Mimir with Helm,“ Grafana Labs, 2023. [Võrgumaterjal]. Available: <https://grafana.com/docs/mimir/latest/setup/helm-chart/>. [Kasutatud 3 Aprill 2024].
- [28] Grafana Labs, „Install Grafana Loki with Helm,“ Grafana Labs, [Võrgumaterjal]. Available: <https://grafana.com/docs/loki/latest/setup/install/helm/>. [Kasutatud 1 Aprill 2024].

- [29] Skedler Team, „Everything You Need to Know About Grafana,“ Skedler, 27 August 2021. [Vörgumaterjal]. Available: <https://www.skedler.com/blog/everything-you-need-to-know-about-grafana/>. [Kasutatud 2 Aprill 2024].
- [30] „What is Grafana?,“ Red Hat, Inc., 13 Mai 2022. [Vörgumaterjal]. Available: <https://www.redhat.com/en/topics/data-services/what-is-grafana>. [Kasutatud 25 Märts 2024].
- [31] D. Lee, „10 Grafana features you need to know for effective monitoring,“ Red Hat, Inc., 18 Veebruar 2020. [Vörgumaterjal]. Available: <https://opensource.com/article/20/2/grafana-features>. [Kasutatud 1 Aprill 2024].
- [32] Bitnami, „Bitnami package for PostgreSQL GitHub,“ Bitnami, Mai 2024. [Vörgumaterjal]. Available: <https://github.com/bitnami/charts/tree/main/bitnami/postgresql>. [Kasutatud 10 Aprill 2024].
- [33] Bitnami, „Bitnami package for PostgreSQL,“ bitnami by vmware, Mai 2024. [Vörgumaterjal]. Available: <https://bitnami.com/stack/postgresql/helm>. [Kasutatud 10 Aprill 2024].
- [34] Grafana Labs, „Dashboards,“ Grafana Labs, 2024. [Vörgumaterjal]. Available: <https://grafana.com/grafana/dashboards/>. [Kasutatud 20 Aprill 2024].



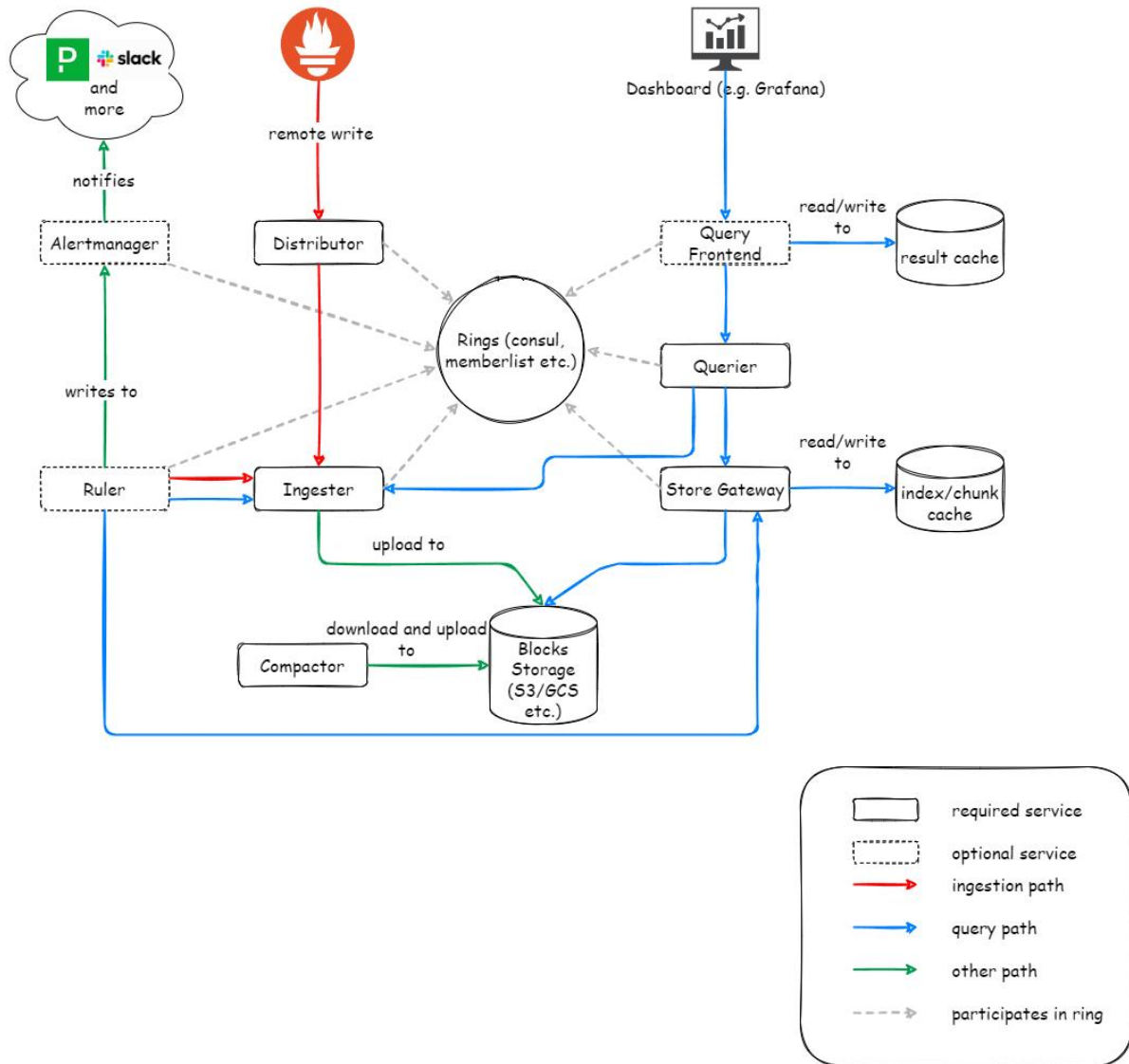
## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, Anneli Väli

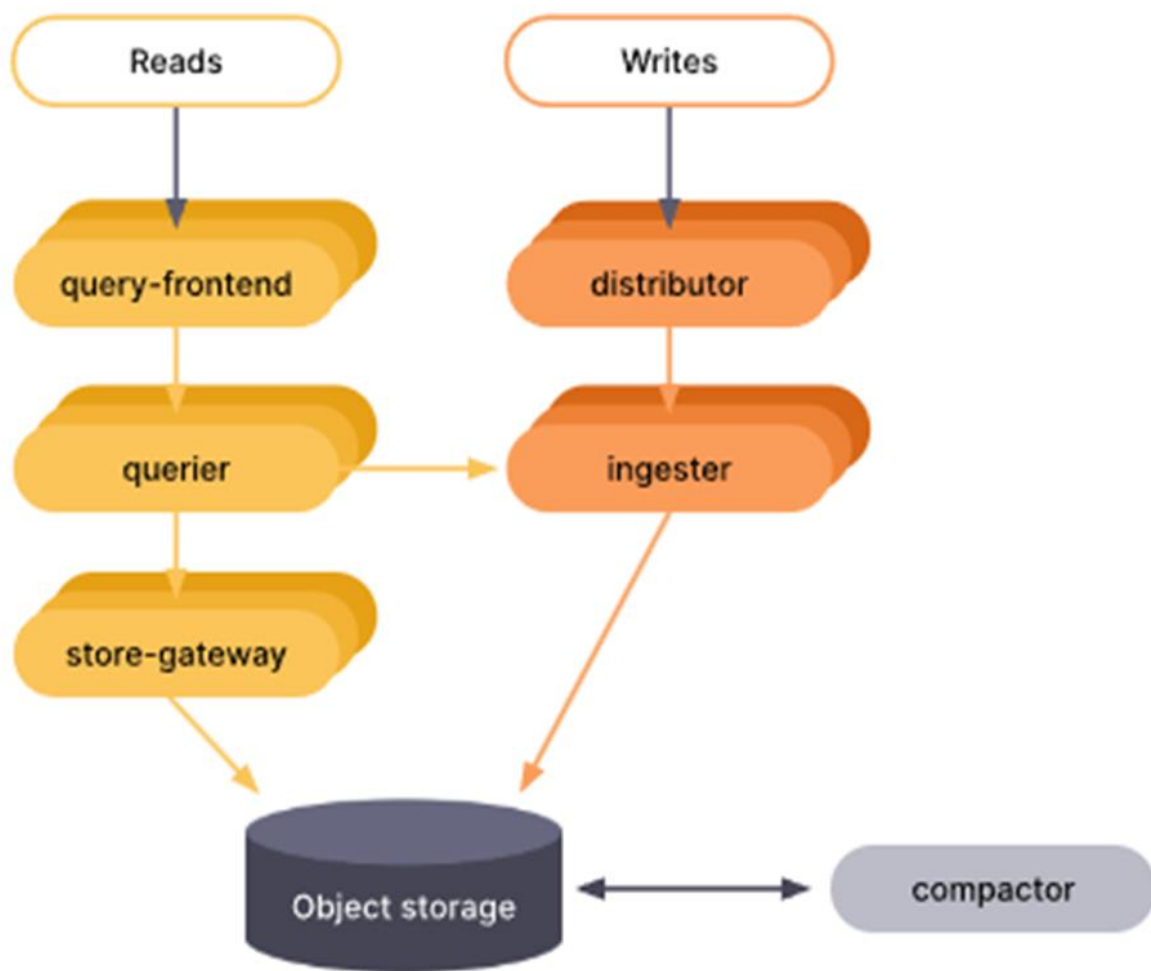
1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Seire- ja logimislahenduste integreerimine ettevõttele Datel AS”, mille juhendaja on Margus Sumla
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

13.05.2024

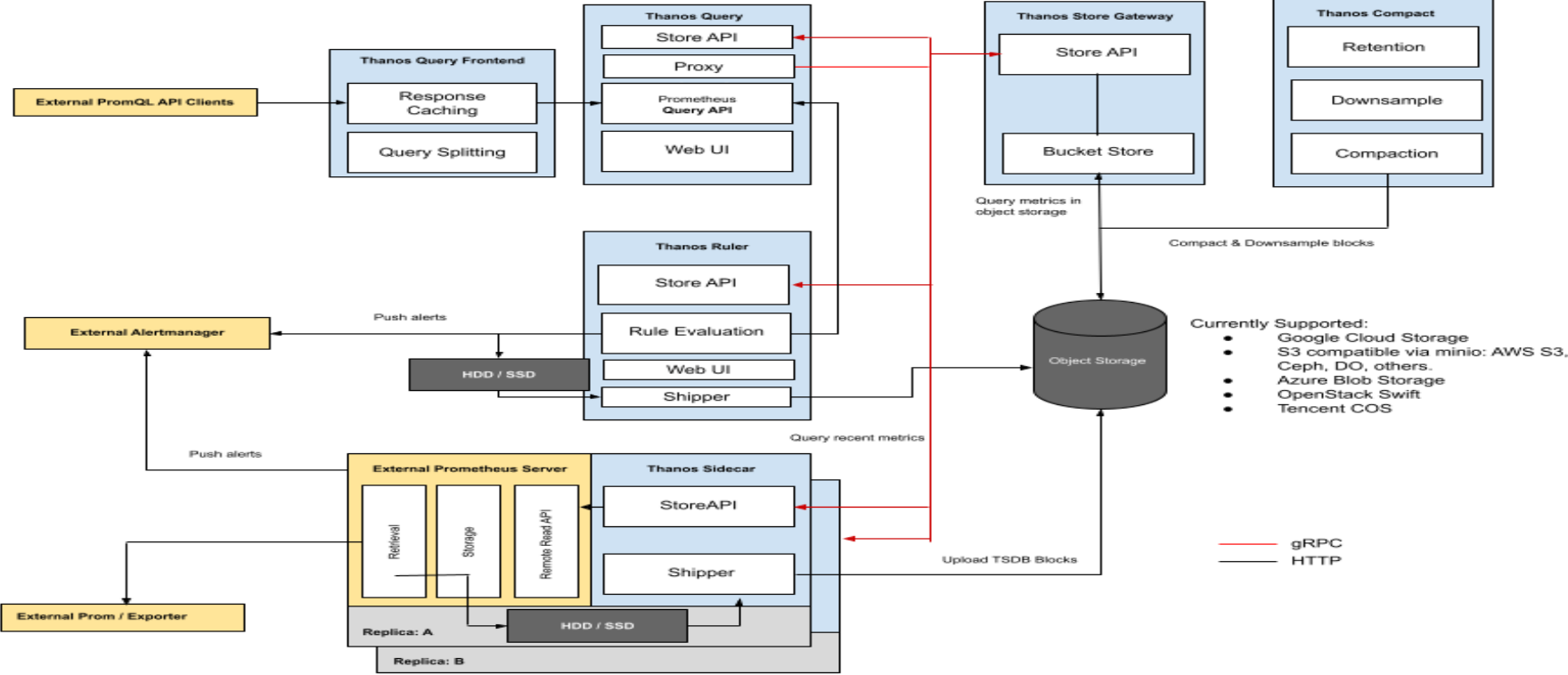
## Lisa 2 – Cortex'i arhitektuur



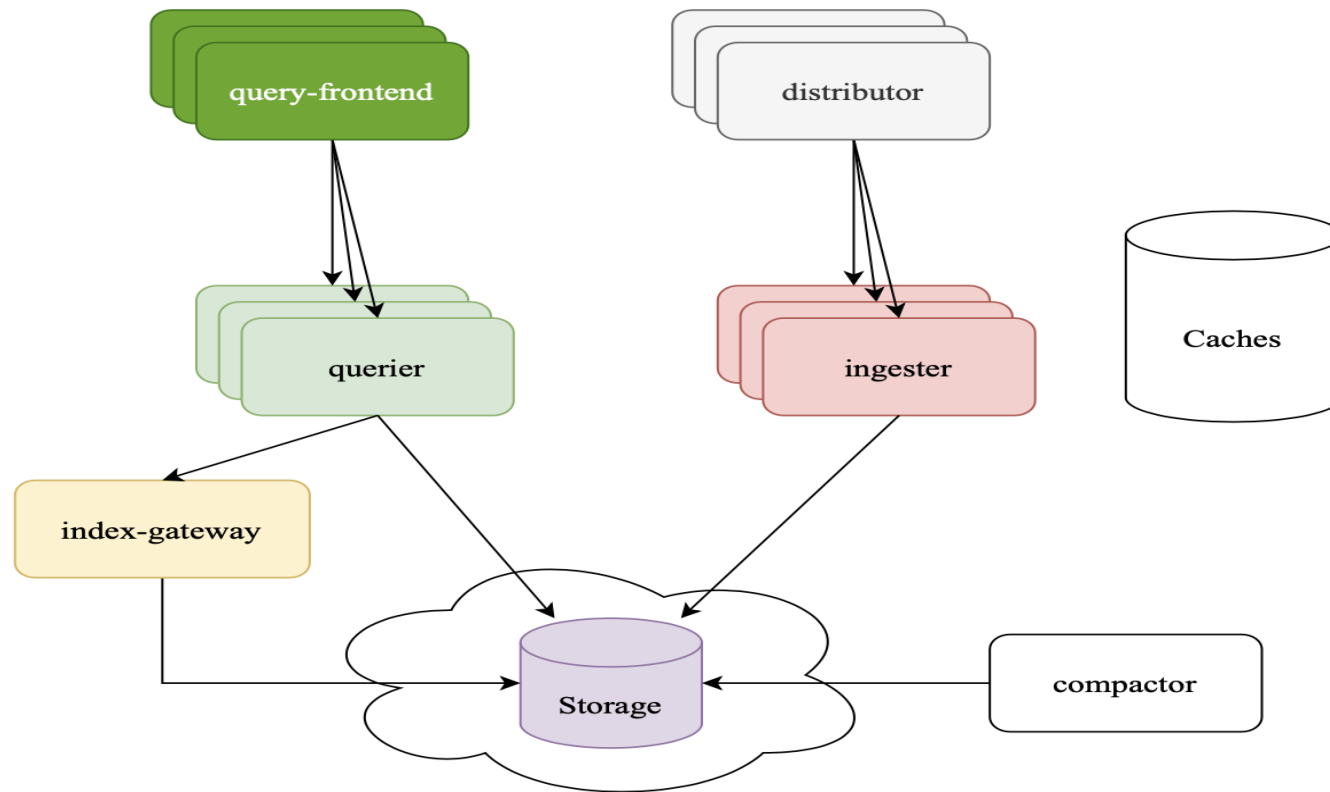
### Lisa 3 – Mimir'i arhitektuur mikroteenuse režiimis



# Lisa 4 – Thanose arhitektuur



## Lisa 5 – Loki klastru arhitektuur



## Lisa 6 – Mimir helm chart'i näide

```
mimir:
  structuredConfig:
    common:
      storage:
        backend: s3
        s3:
          endpoint: s3.x.x
          region: ""
          secret_access_key: "${access_key}"
          access_key_id: "${access_key_id}"
distributor:
  replicas: 1
  resources:
    requests:
      cpu: 100m
      memory: 512Mi
ingester:
  replicas: 3
  resources:
    requests:
      cpu: 100m
      memory: 1024Mi
  persistentVolume:
    size: 20Gi
ruler:
  enabled: true
  replicas: 1
  resources:
    requests:
      cpu: 100m
      memory: 128Mi
query_frontend:
  replicas: 1
  resources:
    requests:
      cpu: 100m
      memory: 128Mi
query_scheduler:
  enabled: true
  replicas: 2
  resources:
    requests:
      cpu: 100m
      memory: 128Mi
```

## Lisa 7 – Loki helm chart'i näide

```
loki:
  structuredConfig:
    common:
      storage:
        s3:
          bucketnames: loki
          endpoint: https://s3.x.x
          region: ""
          access_key_id: "${access_key_id}"
          secret_access_key: "${secret_access_key}"
          s3forcepathstyle: true

gateway:
  enabled: true
  replicas: 1
  ingress:
    enabled: true
    hosts:
      - host: loki.x.x
    paths:
      - path: /
```

## Lisa 8 –Lõputöös loodud lahenduse arhitektuur

