Open source monitoring and observability stack on Kubernetes

Student Team

Pooja Srinivasan (1BM18CS069)
Anusree Manoj K (1BM18CS017)
Niha (1BM18CS060)
Shikha N (1BM18CS149)

Faculty Mentor

Dr. Nandini Vineeth
Assistant Professor

HPE Mentors

Divakar Padiyar
Sonu Sudhakaran

Hewlett Packard
Enterprise

B.M.S. COLLEGE OF ENGINEERING
ESTD. 1946
BANGALORE - 560019

kubernetes

Agenda

- Introduction
- Abstract
- What is observability
- Pillars of observability
- Observability vs Monitoring
- Observability Use cases
- Observability solution with Prometheus, Grafana, Loki and Jaeger
- Demo
- Learnings
- Next steps

Introduction

- **Aim** - Open source monitoring and observability stack on kubernetes.
- **Kubernetes** - open-source container orchestration platform.
- Designed to automate the **deployment**, **scaling**, and **management** of containerized applications.
- **Three pillars of observability** - **logs, metrics, and traces**
  When combined, they provide sufficient insights to monitor software at any scale.
- The project covers Monitoring, Alerting/Visualization, Log Aggregation/analytics, and Distributed systems tracing infrastructure which collectively make up observability.

Abstract

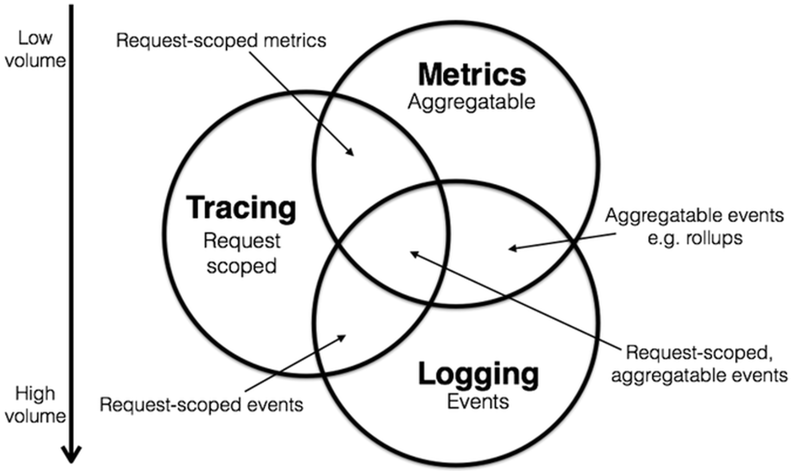
- Physical Server vs Virtual Machine vs Containers.
- Containerization and role of Kubernetes.
- Observability and Monitoring with respect to K8s.
- Setting up kubernetes tools - Micro K8s, Minikube, K3s.
- Using Grafana and Prometheus stack on the cluster to observe metrics of cluster.
- Configuring Alertmanager to receive alerts on slack channel.
- Logging and Tracing using Loki and Jaeger.

What is Observability?

- A way to get insights into the whole infrastructure.
- Can explain any questions about what is happening on the inside of the system just by observing the outside of the system.
- Helps developers understand multi-layered architectures: what’s slow, what’s broken, and what needs to be done to improve performance.
- Creates an insight through an actionable knowledge of the whole environment by assembling all fragments from logs, monitoring tools and organizing them.

Three Pillars Of Observability

- **Metrics:**
  These are numeric representation of data measured over intervals of time.
- **Logging:**
  They are discrete events and data in a structured textual form.
- **Tracing:**
  Represents consecutive events which reflect an end-to-end request path in a distributed system.



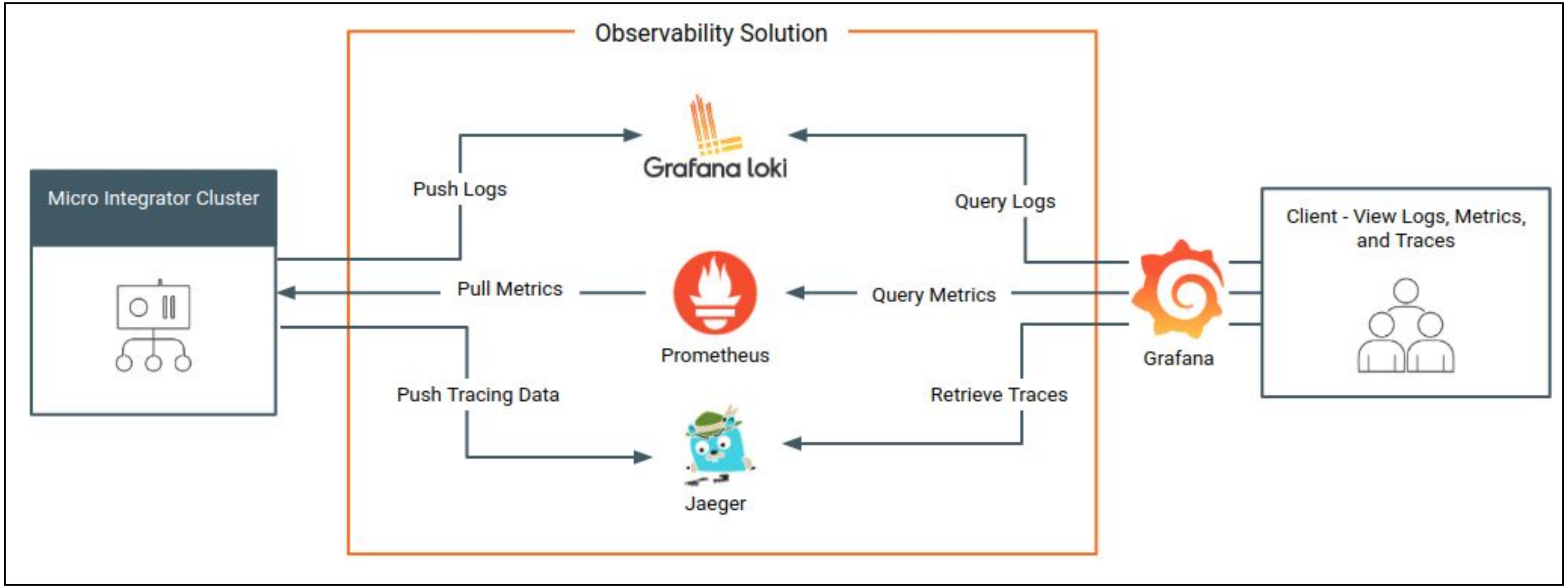
Observability vs Monitoring

Observability	**Monitoring**
What is the system doing?	Is the system working?
Tells us **why** something goes wrong	Tells us **when** something went wrong
Proactive in nature	**Reactive** in nature
Reduces the duration and impact of incidents	Enables **quick response** when an incident occurs
Gain understanding **actively**	Consume information **passively**
Build to tame dynamic environments with changing complexity	Built to maintain static environments with little variation
Preferred by developers of systems with variability and **unknown permutations**	Used by developers of systems with little change and **known permutation**

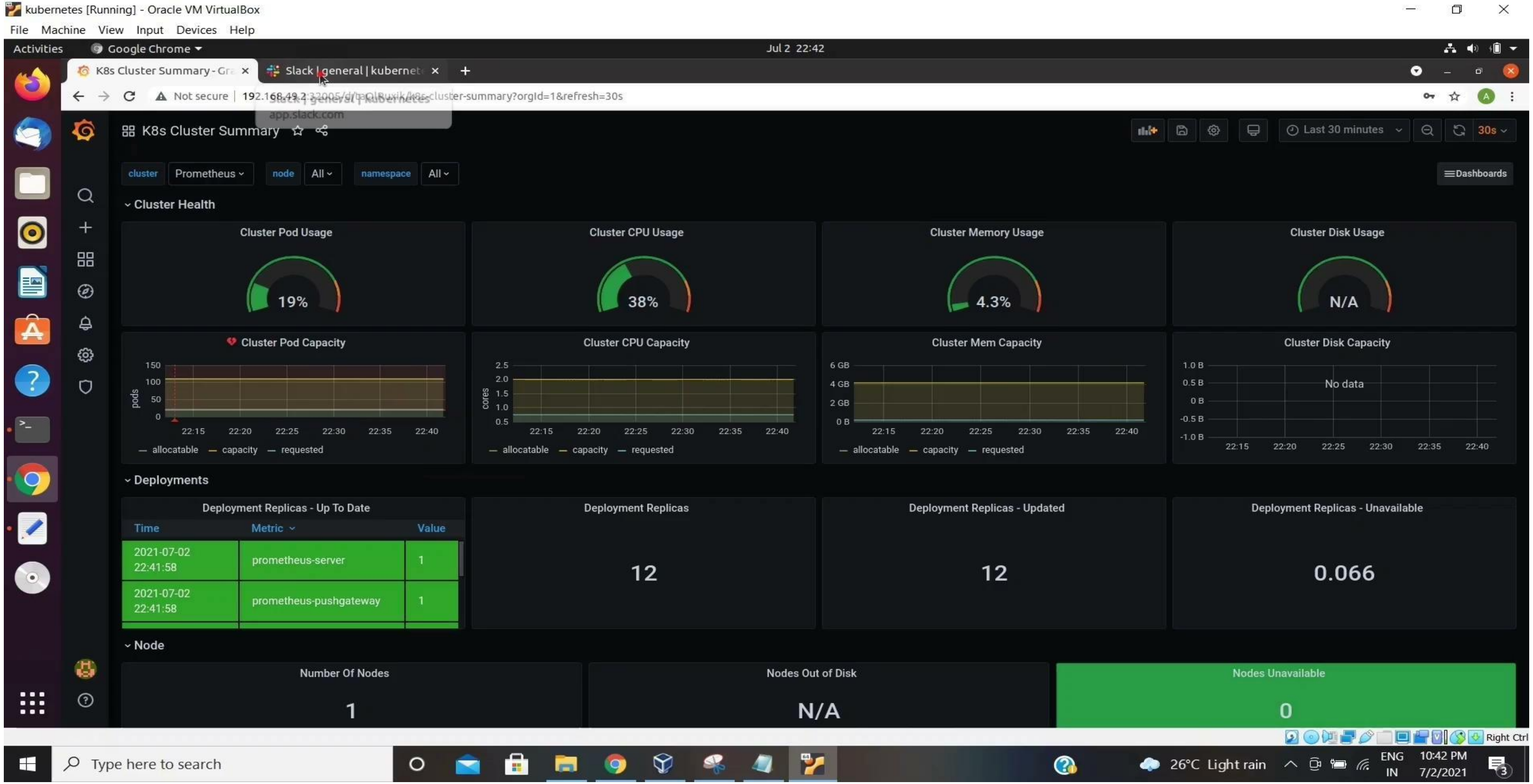
Observability Use cases

- As an Infrastructure Admin, I would like the "Platform" services are monitored constantly for application runtime errors, re-curing operational / API / UI failures and generate alerts so that assigned / scheduled Support team gets notified immediately and is able to troubleshoot the error through Operations Console.
- As an Infrastructure Admin, I would like to configure Monitoring in the "Platform" to send Monitoring alerts to Operations console for any persistent failures in the System for timely response from the Operations team.
- As an SRE, I would like to monitor a gradual but consistent degradation of the Platform services performance / response times so that I can rectify any hardware resource related or scaling issues with the application and prevent missing any SLA(s).
- As an IT Operator, I would like to monitor cluster kube state and node metrics of Platform.
- As an IT Operator, I would like to monitor all the namespaces of the Platform.

Observability solution with Prometheus, Grafana, Loki and Jaeger



Demo



K8s Cluster Summary

cluster: Prometheus · node: All · namespace: All

Cluster Health

Cluster Pod Usage: 19%

Cluster CPU Usage: 38%

Cluster Memory Usage: 4.3%

Cluster Disk Usage: N/A

Cluster Pod Capacity · Cluster CPU Capacity · Cluster Mem Capacity · Cluster Disk Capacity

Deployments

Deployment Replicas - Up To Date

Time	Metric	Value
2021-07-02 22:41:58	prometheus-server	1
2021-07-02 22:41:58	prometheus-pushgateway	1

Deployment Replicas: 12

Deployment Replicas - Updated: 12

Deployment Replicas - Unavailable: 0.066

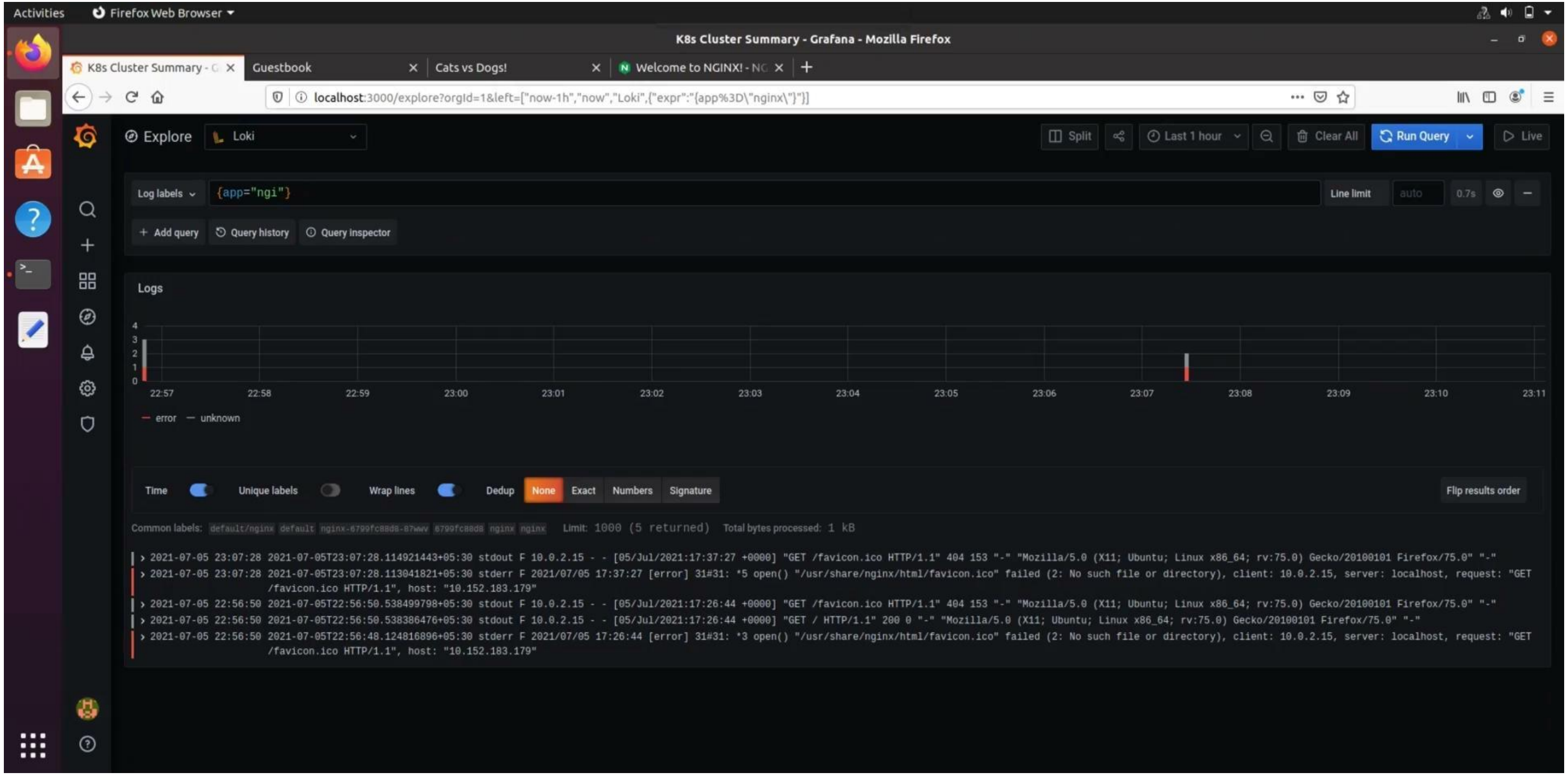
Node

Number Of Nodes: 1

Nodes Out of Disk: N/A

Nodes Unavailable: 0

Demo



localhost:3000/explore?orgId=1&left=["now-1h","now","Loki",{"expr":"{app%3D\"nginx\"}"}]

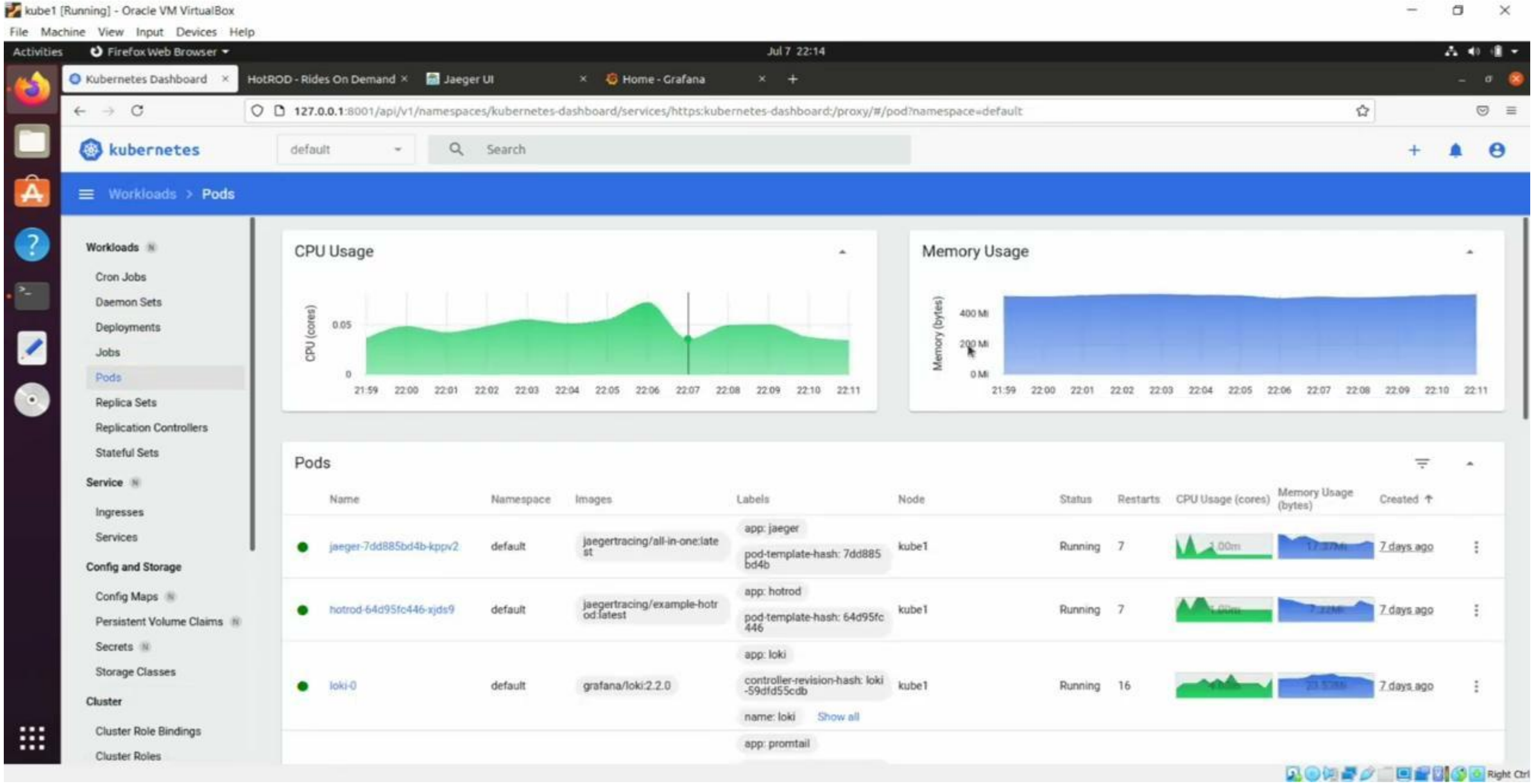
`{app="ngi"}`

Time ● Unique labels ○ Wrap lines ● Dedup [✓ None] Exact Numbers Signature

Common labels: default/nginx default nginx-6799fc88d8-87wwv 6799fc88d8 nginx nginx Limit: 1000 (5 returned) Total bytes processed: 1 kB

```
› 2021-07-05 23:07:28 2021-07-05T23:07:28.114921443+05:30 stdout F 10.0.2.15 - - [05/Jul/2021:17:37:27 +0000] "GET /favicon.ico HTTP/1.1" 404 153 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0" "-"
› 2021-07-05 23:07:28 2021-07-05T23:07:28.113041821+05:30 stderr F 2021/07/05 17:37:27 [error] 31#31: *5 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 10.0.2.15, server: localhost, request: "GET
                      /favicon.ico HTTP/1.1", host: "10.152.183.179"
› 2021-07-05 22:56:50 2021-07-05T22:56:50.538499798+05:30 stdout F 10.0.2.15 - - [05/Jul/2021:17:26:44 +0000] "GET /favicon.ico HTTP/1.1" 404 153 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0" "-"
› 2021-07-05 22:56:50 2021-07-05T22:56:50.538386476+05:30 stdout F 10.0.2.15 - - [05/Jul/2021:17:26:44 +0000] "GET / HTTP/1.1" 200 0 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0" "-"
› 2021-07-05 22:56:50 2021-07-05T22:56:48.124816896+05:30 stderr F 2021/07/05 17:26:44 [error] 31#31: *3 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 10.0.2.15, server: localhost, request: "GET
                      /favicon.ico HTTP/1.1", host: "10.152.183.179"
```

Demo



	Name	Namespace	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
●	jaeger-7dd885bd4b-kppv2	default	jaegertracing/all-in-one:latest	app: jaeger pod-template-hash: 7dd885bd4b	kube1	Running	7	1.00m	17.37Mi	7 days ago
●	hotrod-64d95fc446-xjds9	default	jaegertracing/example-hotrod:latest	app: hotrod pod-template-hash: 64d95fc446	kube1	Running	7	1.00m	7.22Mi	7 days ago
●	loki-0	default	grafana/loki:2.2.0	app: loki controller-revision-hash: loki-59dfd55cdb name: loki Show all	kube1	Running	16	[illegible]	[illegible]	7 days ago
				app: promtail						

Learnings

- Importance of containerization and container orchestration.
- Importance of using Kubernetes.
- Using different tools to set up kubernetes clusters.
- Importance of observability and monitoring.
- Exploring different tools for observability and monitoring.
- Deploying multiple applications on a kubernetes cluster.
- Practical implementation of three pillars of observability i.e metrics, logging and tracing.
- Exploring multi-cluster observability.

Next steps

- Creating a central dashboard for visualizing metrics of multiple clusters.
- Using Spark to process Big Data on Kubernetes.
- Implementation of load balancers using traefik.
- Monitoring and observability using EFK Stack.
- Observability using Open Telemetry.
- Setting up a High availability cluster using K3d.
- AI/ML stack on Kubernetes.

CHALLENGES FACED

- Working remotely
- Insufficient hardware
- Ramp up time in working with various tools

Any Questions?

Thank You!