



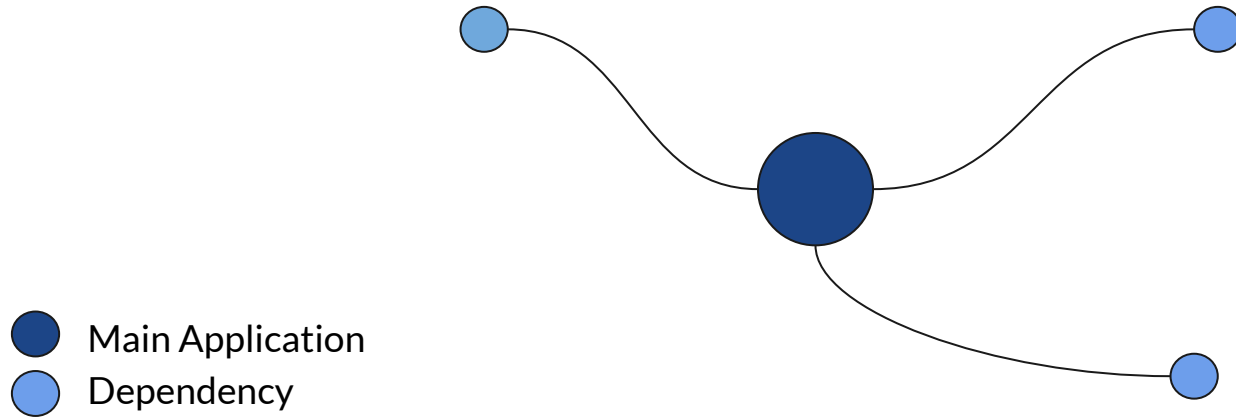
# *Symphonize*

## Kubernetes Dashboard

Bharti Garde  
Chinmay Disale

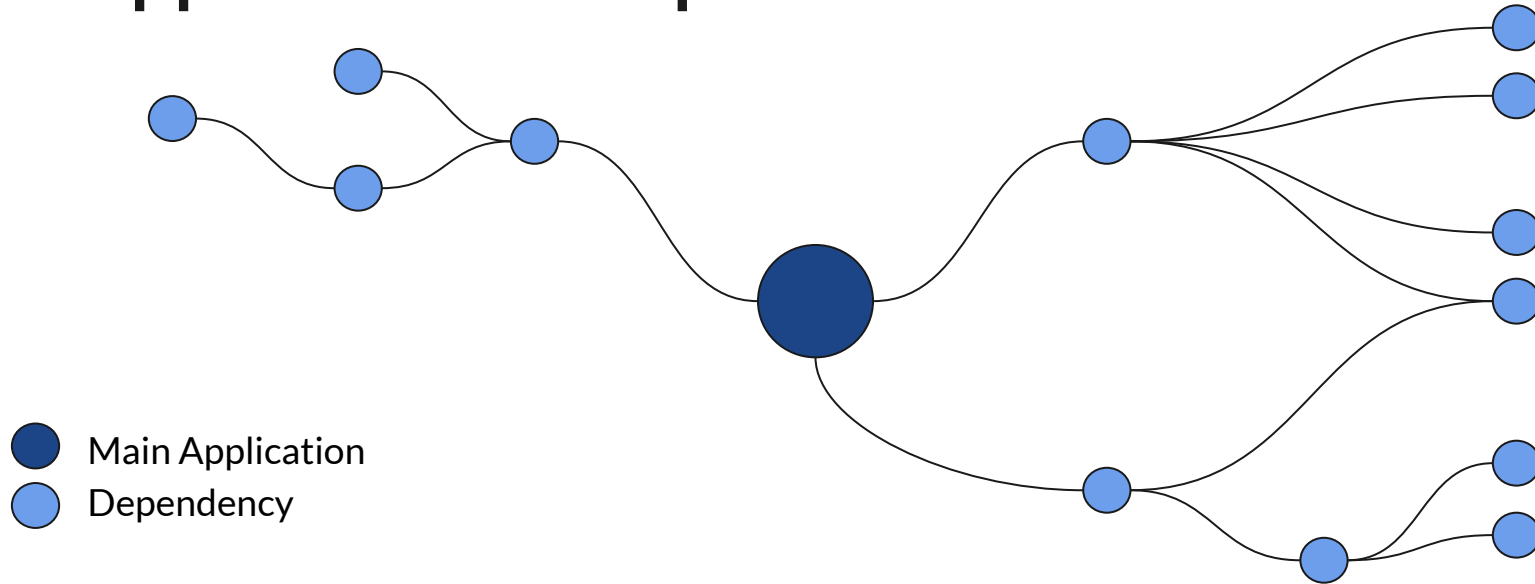


# Applications & Dependencies

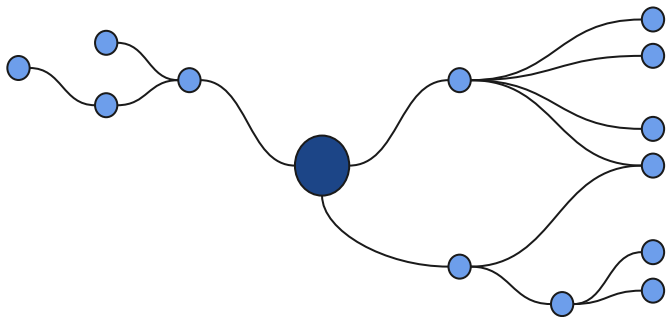




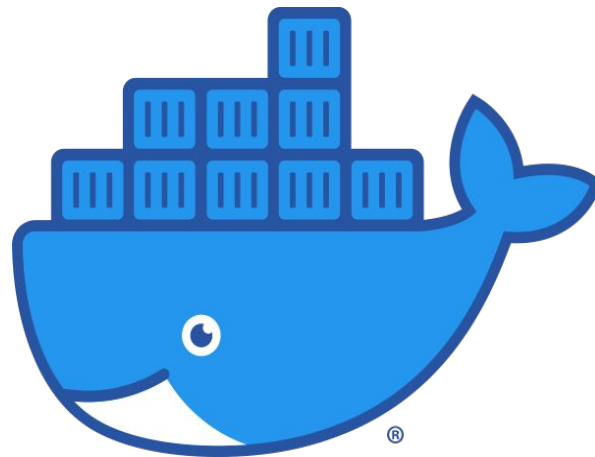
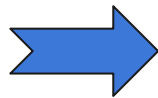
# Applications & Dependencies



# Containerization



Traditional Software



Containerized Software



# How modern apps are built?

- How apps were built previously?
- Problems while deploying
- Possible solutions - microservices approach

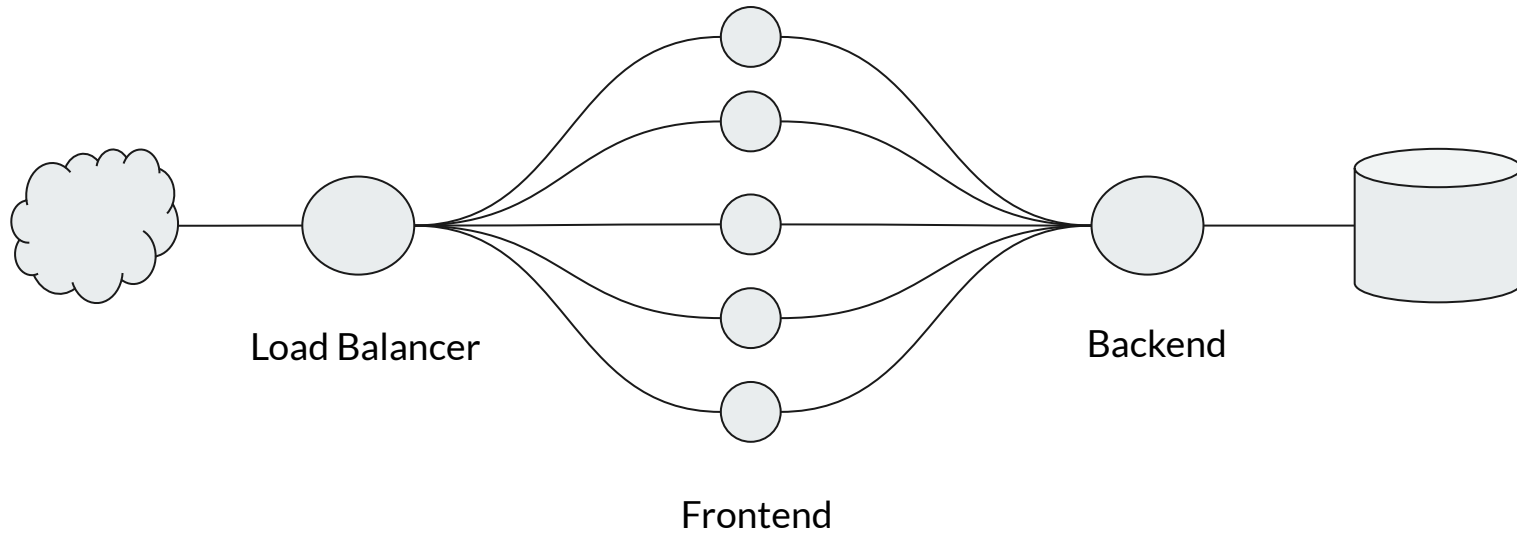


# Example





# Example





# What do these Orchestration tools do?

- Services Discovery
- Auto Scaling
- Cluster Wide Job Scheduling
- Rollouts and Rollbacks





## Comparing Kubernetes & Docker Swarm

Parameters	Docker Swarm	Kubernetes
Scaling	No Autoscaling	Autoscaling
Load Balancing	Cannot configure	Can configure
Logging & Monitoring	3rd party support	Built-in support
Availability	Health checks	Health checks + Restarts
Node support	Upto 5000 nodes	Around 2000 nodes



# Kubernetes Resources

1. Pods
2. Deployments
3. Services
4. Jobs
5. Cron Jobs
6. Config Maps
7. Secrets
8. And many more like Volumes, Daemon sets, etc



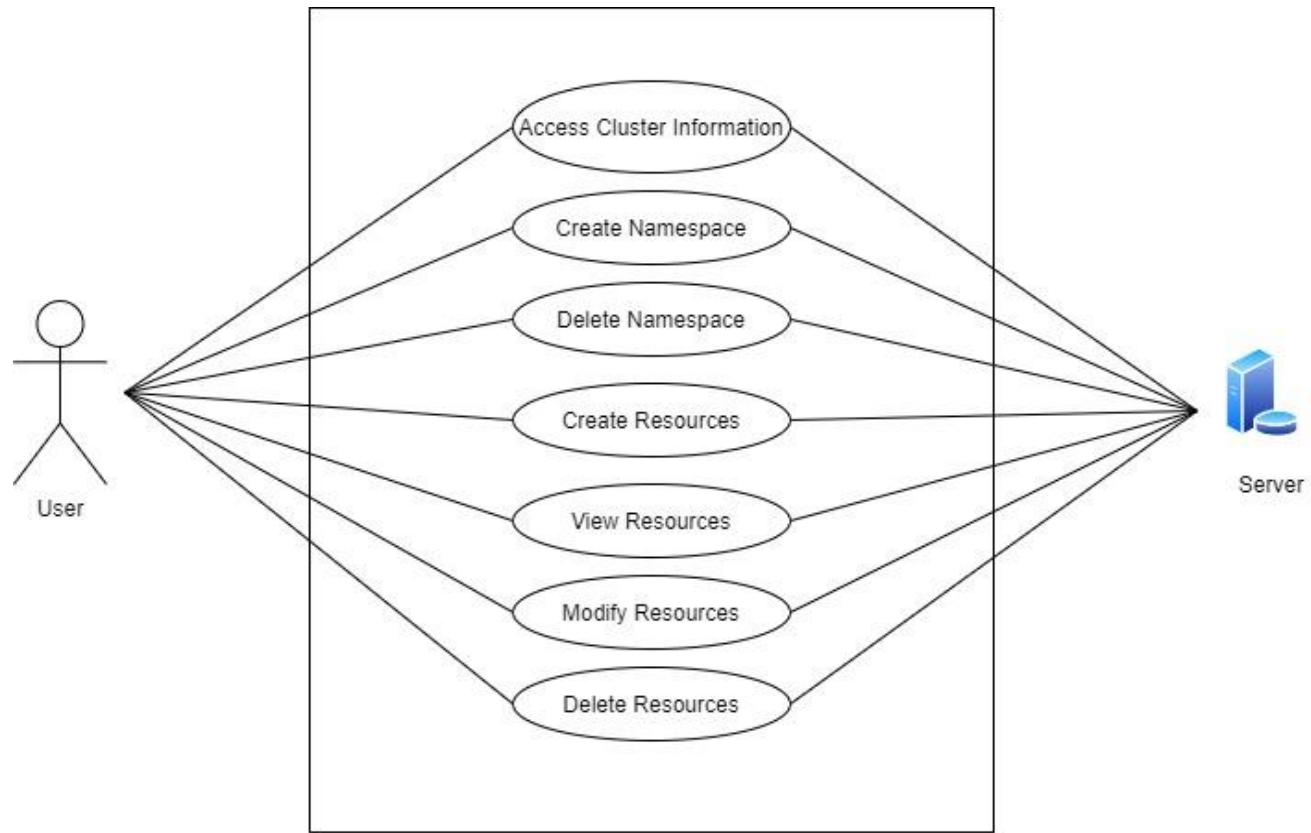
# What we are building?

- Existing ways of managing Kubernetes:
  - CLI
  - YAML
  - JSON
- Existing graphical tools are mainly aimed towards monitoring the cluster.
- There is a need for a tool which makes deploying apps on cluster easy.

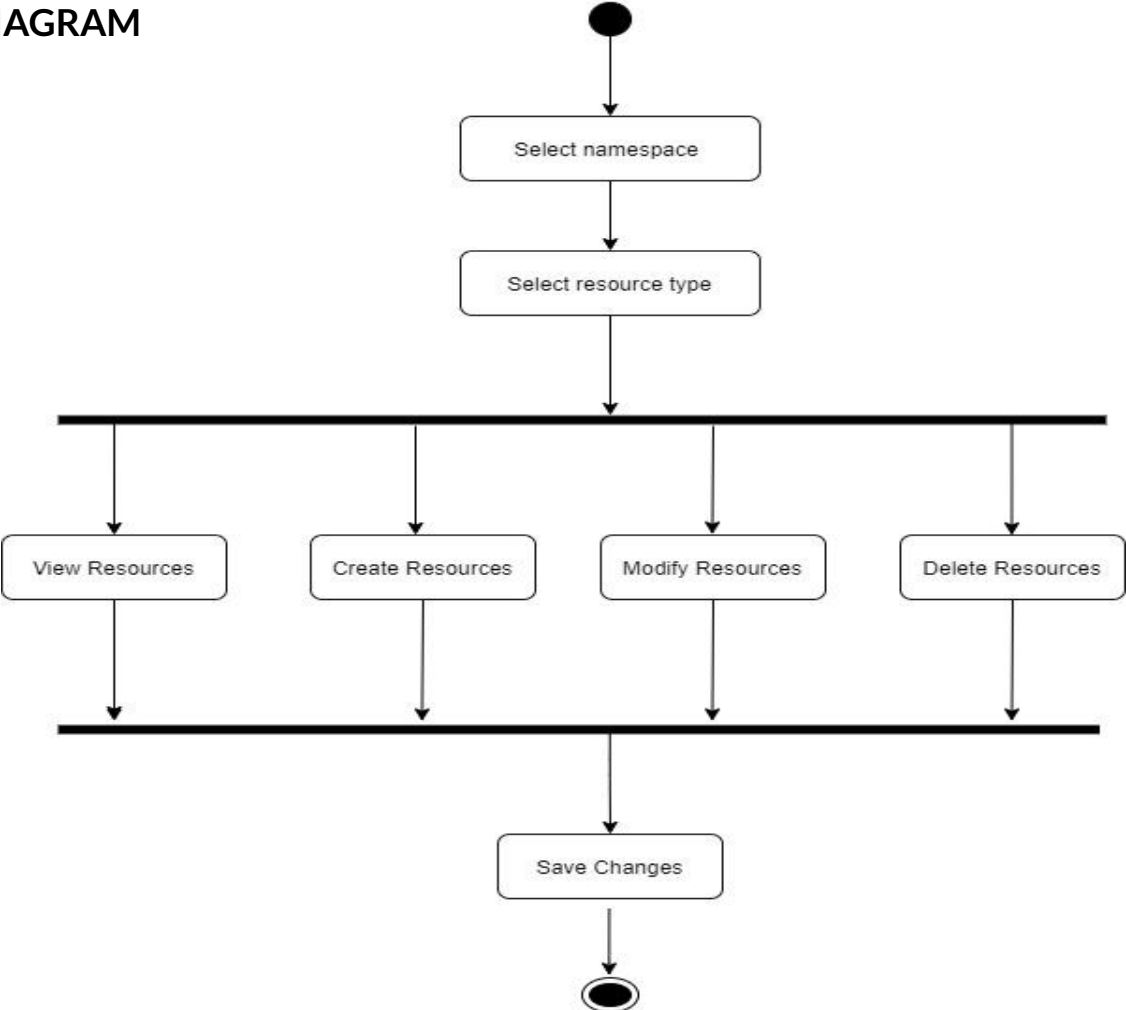
---

# UML Diagrams

## USE CASE DIAGRAM



ACTIVITY DIAGRAM





## Future Enhancements

- Enabling cross orchestration tool migration.
- Adding support for more niche resources.
- Adding a metrics system to analyze traffic across various points of cluster.
- Adding feature to check pod log information.
- Adding feature to SSH into a pod.

---

**THANK YOU!!**