Kubernetes at INCD

Our Journey

Miguel Viana
Samuel Bernardo
João Martins

October 28, 2024

$ whoami

Miguel Viana
DevOps Engineer @ LIP and INCD

Background in HPC: Started my career in high-performance computing cluster administration, where I gained foundations in Linux systems management.

Containerization and Kubernetes: Over time, transitioned into the exciting world of containerization and orchestration with Kubernetes.

Certifications: Currently CKA certified and actively working towards the CKS certification.

Cybersecurity Focus: In addition to my technical skills, I've developed a strong interest in cybersecurity. Taking the Google Cybersecurity Professional Certificate. Regular participation in CTF events and attended CERN's Cybersecurity Summer School.

Our K8s team

Jorge Gomes

João Machado

Samuel Bernardo

João Martins

César Ferreira

João Pina

Zacarias Benta

Miguel Viana

Our timeline

Building Kubernetes for:

A. INCD internal services
B. INCD external services
C. INCD end users

Late 2019

Kubernetes PoC on Idle Blades

Initial Kubernetes proof of concept on unused hardware

Early 2020

Cloud Migration with Kubespray

Started migrating Kubernetes cluster to the cloud using Kubespray

Mid 2020

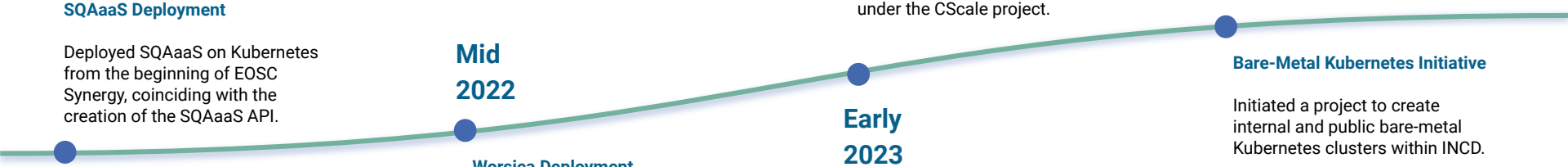
First Production-Grade Kubernetes Cluster

Successfully deployed the first production-grade Kubernetes cluster on the cloud without the need for a load balancer

Late 2020

Vanadium Supports the 1st Public Service and a ELK stack monitoring

Vanadium began supporting our GitLab instance as public service. Simultaneously, the ELK stack was deployed on K8s for log aggregation, and prometheus for Lustre and Slurm monitoring.



Early 2022

SQAaaS Deployment

Deployed SQAaaS on Kubernetes from the beginning of EOSC Synergy, coinciding with the creation of the SQAaaS API.

Mid 2022

Worsica Deployment

Deployed Worsica on Kubernetes in August 2022, towards the end of the EOSC Synergy project.

Early 2023

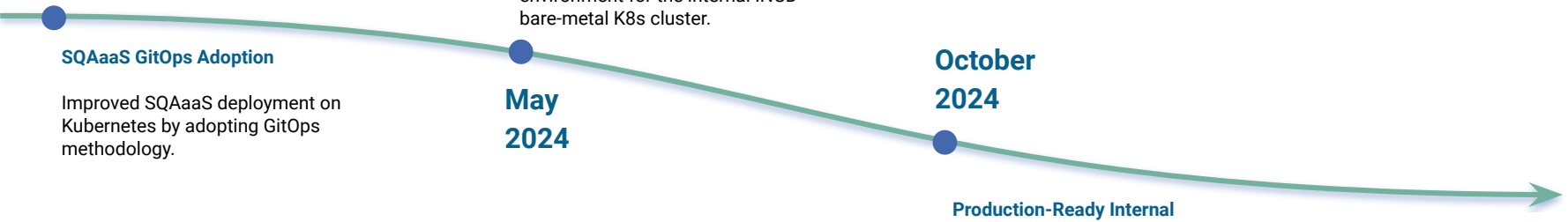
OpenEO Deployment

Deployed OpenEO on Kubernetes under the CScale project.

Mid 2023

Bare-Metal Kubernetes Initiative

Initiated a project to create internal and public bare-metal Kubernetes clusters within INCD.



Early 2024

SQAaaS GitOps Adoption

Improved SQAaaS deployment on Kubernetes by adopting GitOps methodology.

May 2024

Internal bare-metal cluster testing

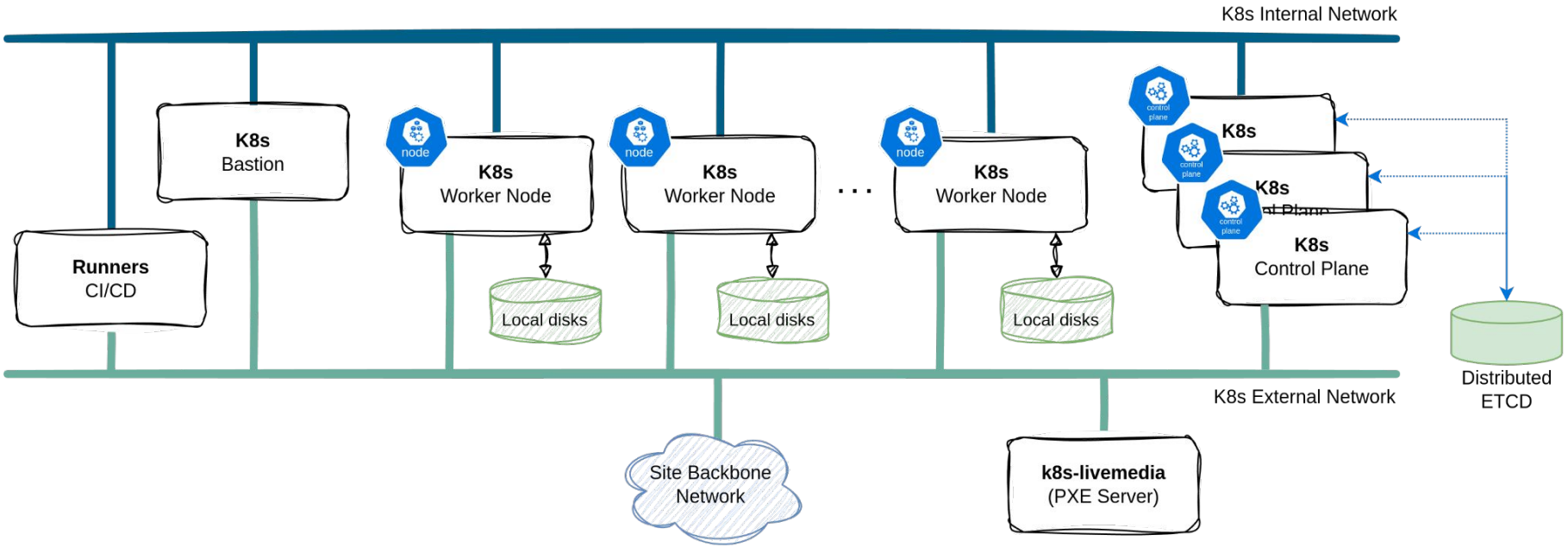
Started testing the preview environment for the internal INCD bare-metal K8s cluster.

October 2024

Production-Ready Internal Cluster

Achieved a production-ready state for the internal INCD bare-metal Kubernetes cluster.

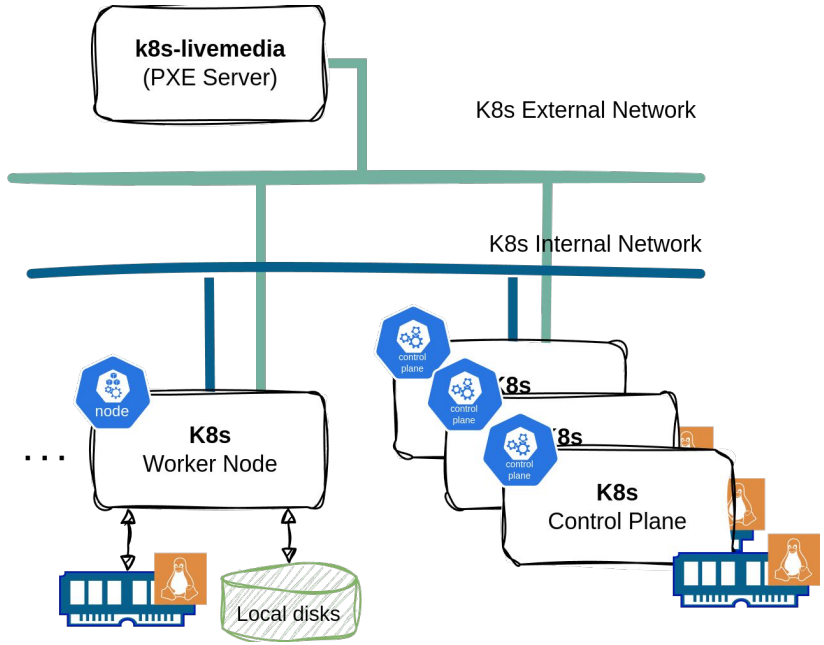
K8s infrastructure



K8s node setup

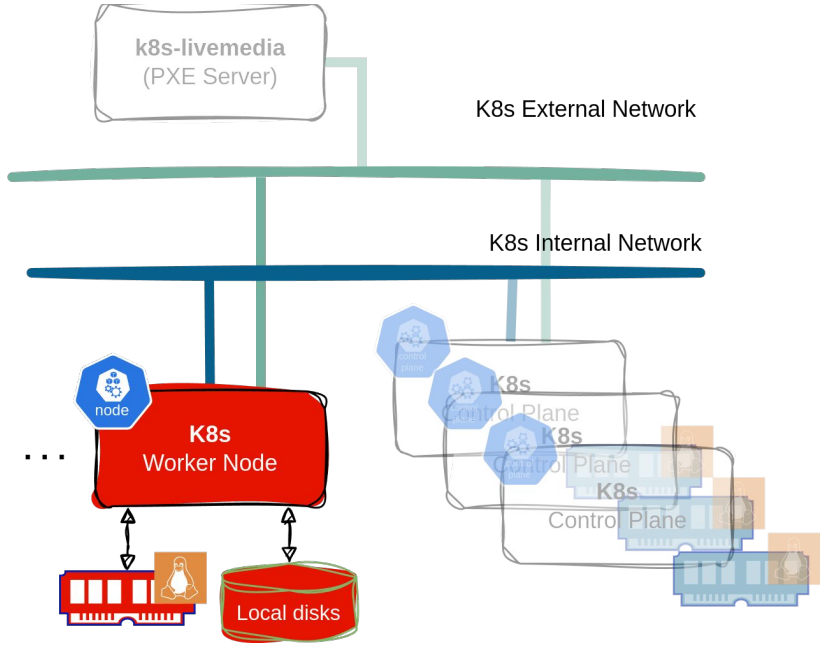
Each node runs a specialized and optimized image based on AlmaLinux OS which is designed to consume the least possible resources, further enhancing efficiency.

AlmaLinux



K8s node setup

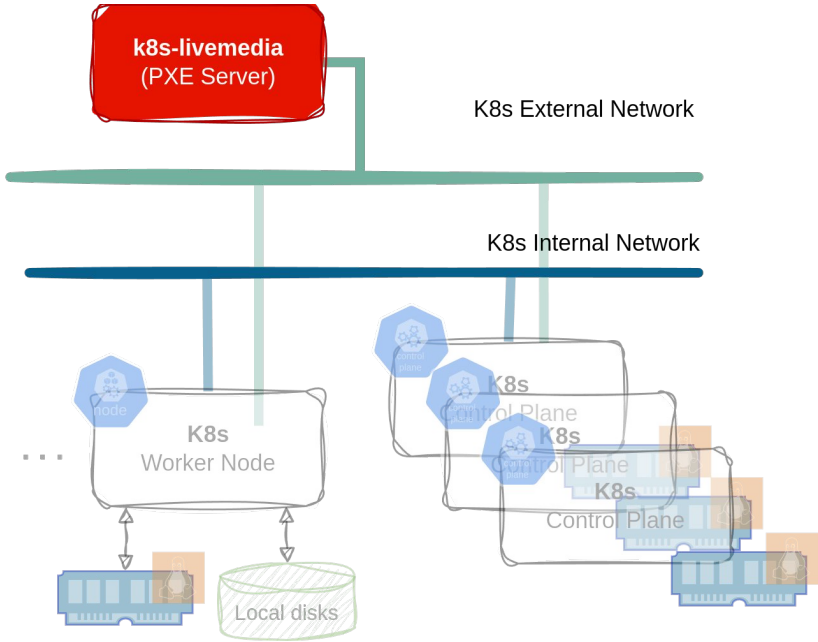
For **enhanced security** and **performance**, nodes run the OS mainly on RAM, with essential data such as containers, configs, logs, and write-intensive directories stored on ZFS for data persistence and minimizing RAM memory usage.



K8s node setup

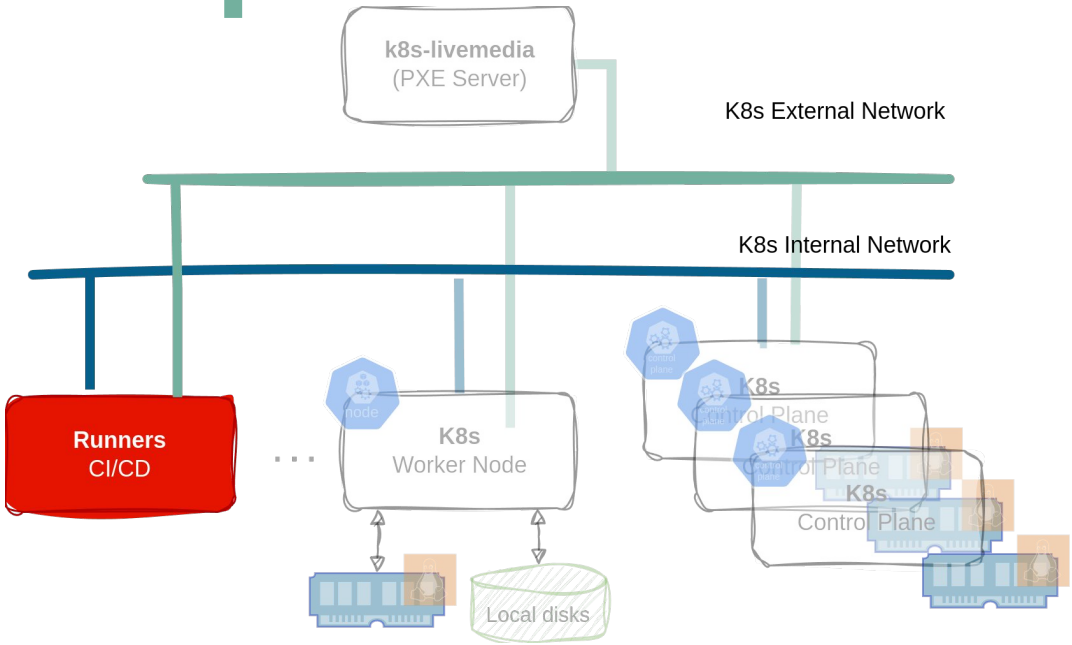
The PXE server provides a **centralized mechanism** for deploying and managing node images.

OS updates and configuration **changes can be easily applied** by simply rebooting the node.



K8s node setup

K8s nodes can be quickly deployed, configured and managed using a Gitlab CI pipeline, ensuring **consistency** and **reducing manual intervention**.

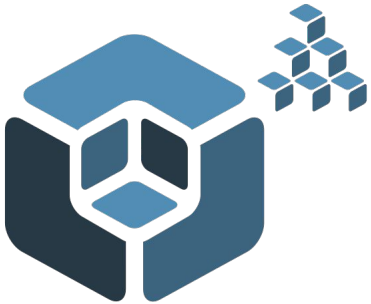


K8s deployment

Manually installation of a Kubernetes cluster can be painful and is very prone to error.

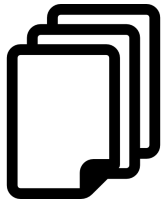
Tools such as Kubespray **simplifies the Kubernetes deployment by automating complex tasks, making this process almost effortless**.

It efficiently installs essential components, generates certificates, configures networking, joins nodes to the cluster, … saving time and effort.



K8s deployment

Kubespray comes with Ansible playbooks to perform all kind of operations over a K8s cluster. We just need to provide the configuration values we want, and we are ready to go!

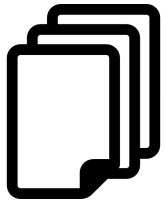


values files

inventory

K8s deployment

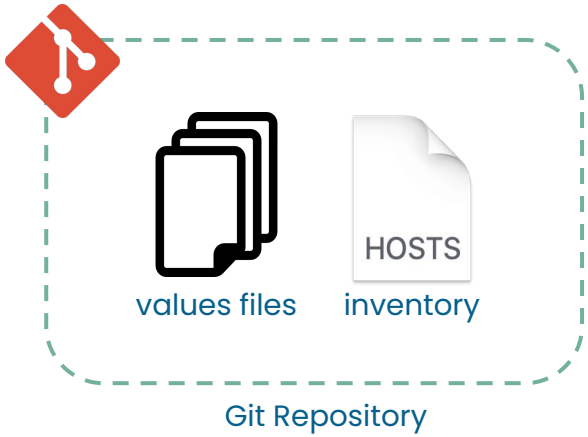
Kubespray comes with Ansible playbooks to perform all kind of operations over a K8s cluster. We just need to provide the configuration values we want, and we are ready to go!



values files inventory

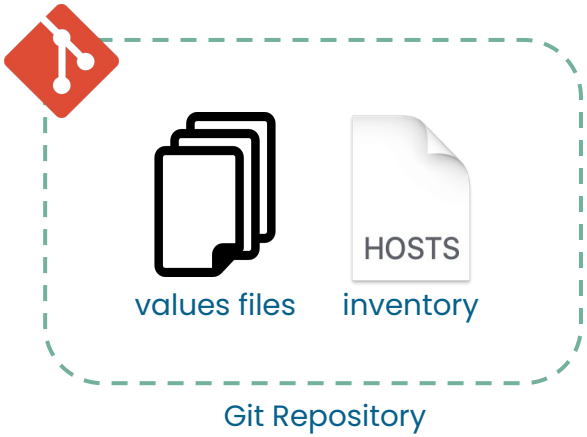
How can we effectively manage and version control those Ansible values and inventory files?

K8s deployment



- Improves the collaboration between team members.
- Creates a historical record of changes.
- Allows to easily revert/rollback to the last known good configuration in case of issues.
- Integrates with CI/CD pipelines

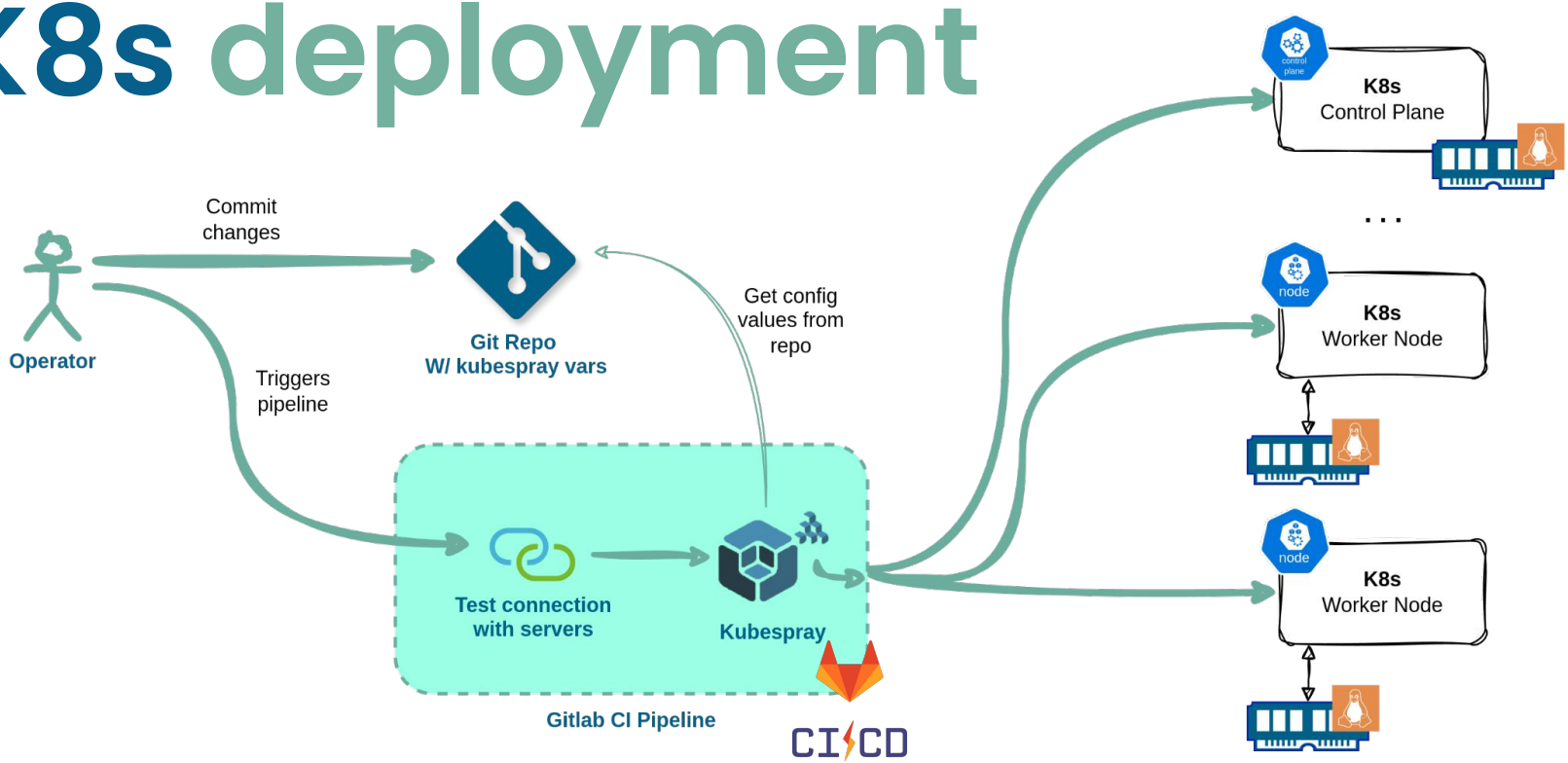
K8s deployment



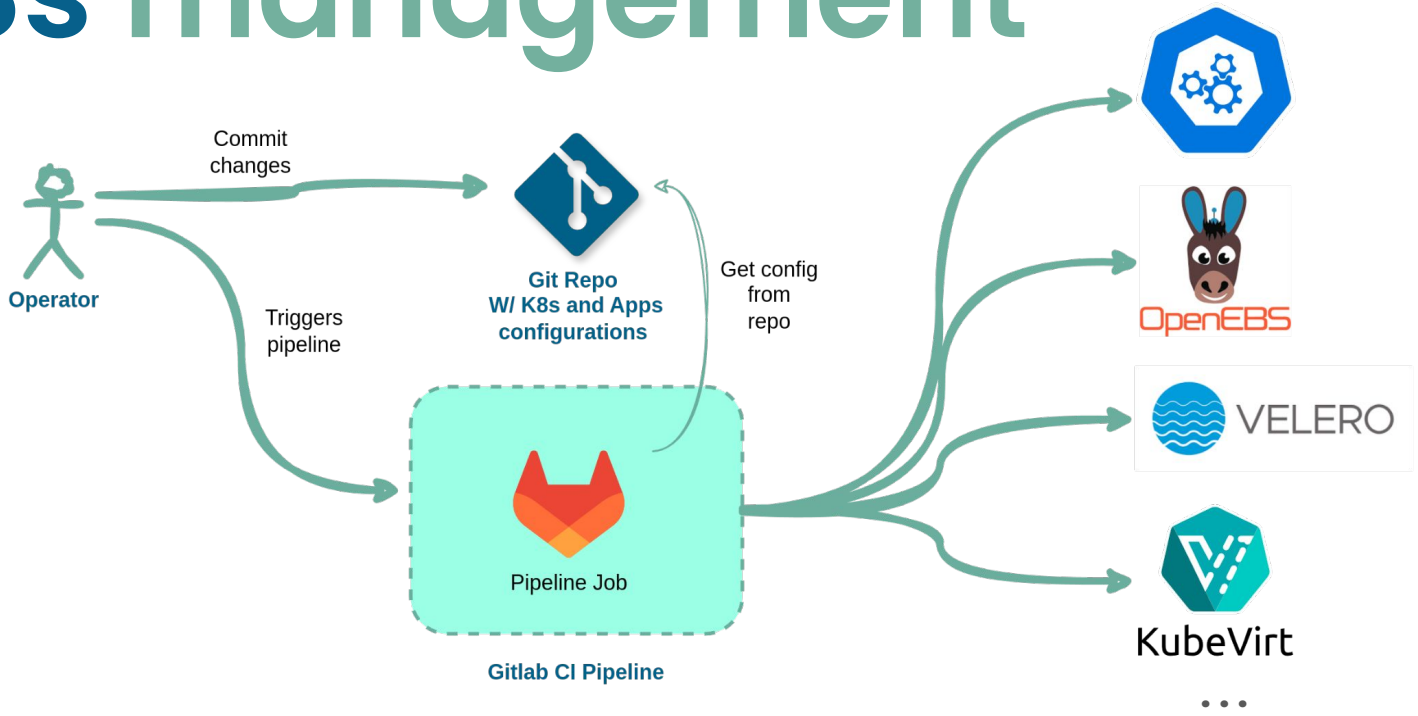
+

CI/CD

K8s deployment



K8s management



Applications landscape

Storage: Mayastor for high-performance, persistent storage.

Ingress Controllers: Nginx and Traefik for managing traffic and exposing applications to the external networks.

Backup and Restore: Velero for automated backups and disaster recovery.

Load Balancing: MetalLB

Applications landscape

Logging and Monitoring: Fluent Bit, OpenTelemetry, Prometheus

CI/CD and Application Deployment: ArgoCD and FluxCD

Secret Management: HashiCorp Vault, External Secrets

Developer Tools: Telepresence

K8s security

We prioritize security in our Kubernetes environment through a layered approach:

Network Security: Network policies

Access Control: RBAC with least privilege principle.

Security Tools: Wazuh, Falco, and Trivy for real-time monitoring and threat detection.

K8s security

Container Security: AppArmor for container hardening.

Security Contexts: Enforcing security contexts for user namespaces.

Centralized Security Operations Center (SOC): Collecting and analyzing security data from various tools and logs.

Thank you
for your attention

Feel free to contact me at mviana@lip.pt