



Astronomical Archive as a Service: A Microservices-Based Hyperconverged Infrastructures

Bologna 26/28 Febbraio 2025 - "Archives and Data management Systems"



Da dove partiamo?

Dove vogliamo arrivare?



Stato dei software scientifici da usare

Know how + Men Power



Tempo a disposizione

Come fare?



Approccio a Step e Fasi

Motivazioni

Step 1

Esigenze

Creare un sistema che consenta di utilizzare varie tecnologie software da mettere insieme per gestire grandi quantità di dati in arrivo dai telescopi.

Step 1

Esigenze

- Gestione Utente
- Simulazioni
- Riduzioni (pipeline e non solo)
- Ingestione in archivio
- Accesso ai dati generati
- Poco tempo per la realizzazione
- Poco personale (1/2 pax)



Step 2

Hardware disponibile

- 12 server
- CPU: 48 core ognuno
- DISCO: 250 TB ognuno
- RAM: 380 GB ognuno
- NETWORK: 2 if da 10 Gb + 2 da 1Gb
- IPMI
- Tape Library (LT05+6)

Step 3

Scelta Architetture

- curva di apprendimento bassa
- **stabilità**
- **ridondanza**
- **scalabilità**
- performance
- backup
- disaster recovery



Step 3

Scelta Architettuale

Sistema Iperconvergente



Step 4

Componenti Software

- Operating System
- Storage Manager
- Configuration Manager
- Virtualization Manager
- Compute Manager
- Monitoring/Alerting

Step 4

Componenti Software

Operating System

- RHEL **
 - server fisici
- Rocky Linux **
 - virtual machines



Step 4

Componenti Software

Storage Manager

- Ceph
- Lustre
- GlusterFS + Ganesha **



Step 4

Componenti Software

Virtualization Manager

- KVM/QEMU



Step 4

Componenti Software

Configuration Manager

- Ansible



Step 4

Componenti Software

Compute Manager

- OpenHPC



Step 4

Componenti Software

Monitoring/Alert

- Zabbix
- Grafana+Prometheus **



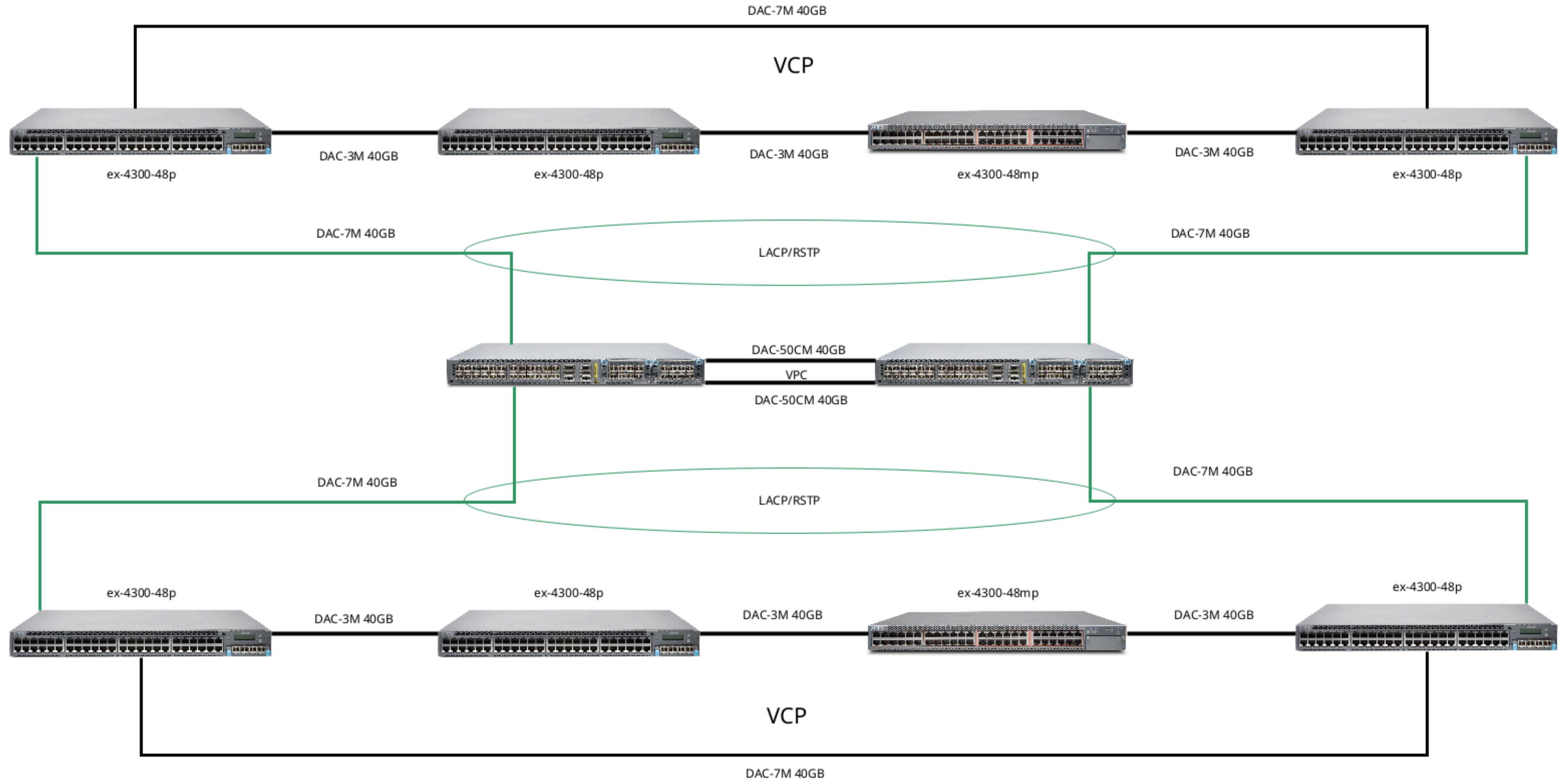
Step 5

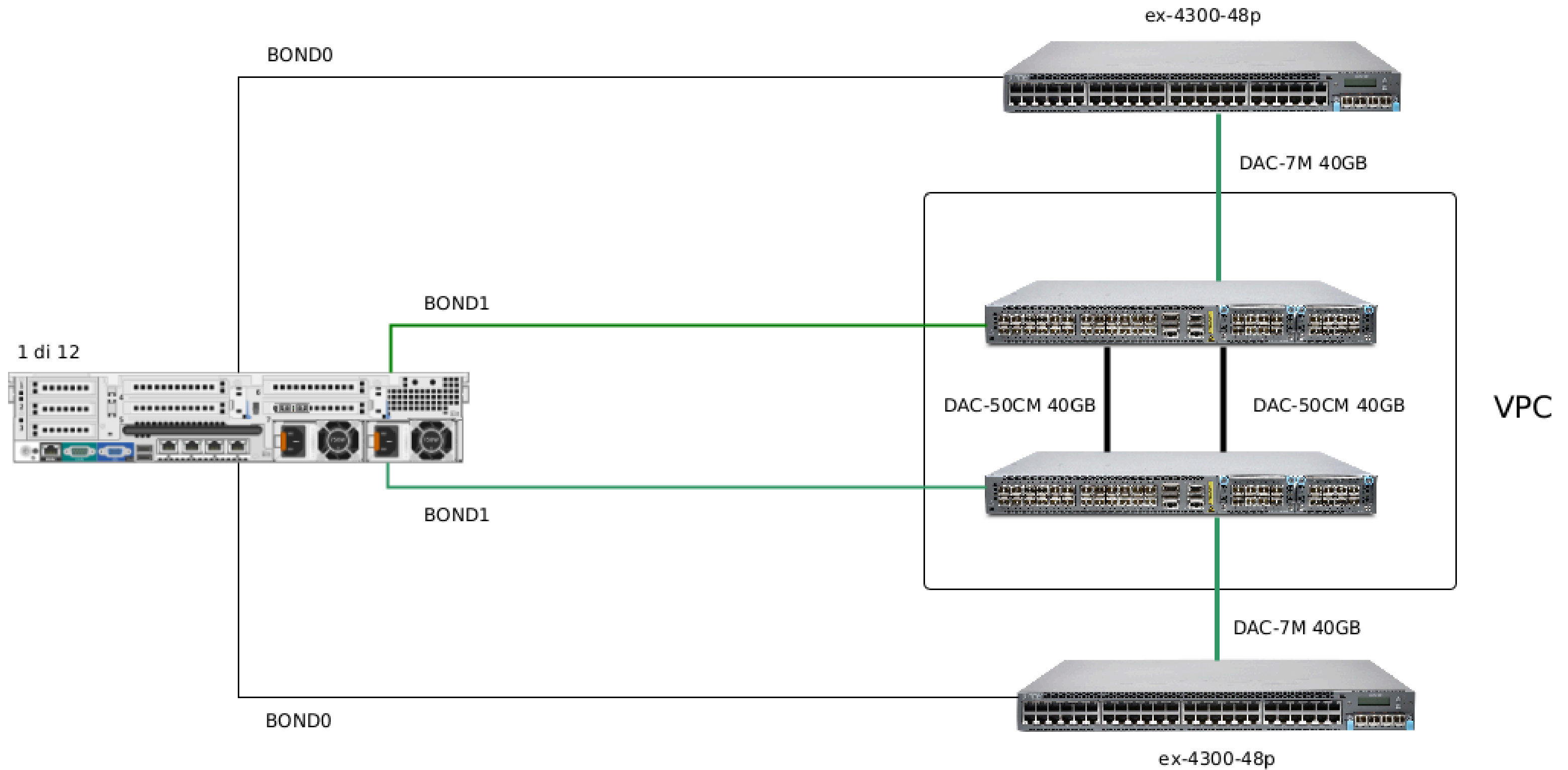
Networking

Acquisti Hardware Networking

Switch Juniper

- 6x EX4300-48P
- 2x EX4300-48MP
- 2x EX4600-40F







Step 5

Networking

VLANs

- storage communications
- storage exports
- OS management
- service 1
- service 2
- PXE
- Compute



Step 6

Valutazioni

Dopo aver testato tutti i software sul campo, dopo innumerevoli reinstallazioni fisiche, abbiamo deciso con quale configurazione patire.



Fase 1

Infrastruttura Tradizionale

- VM
- Software usabile tramite moduli



Fase 2

Testing e Usabilità

- Verifica delle funzionalità
- Usabilità
- Pro/Contro
- Aggiustamenti
- rilascio solo per alcuni utenti



Fase 3

Produzione

- rilascio a tutti gli utenti

Fase 4

Container

- Creazione Cluster Kubernetes (VM)
- pubblicazione prime web-app
 - Grafana+Prometheus
 - Web Pages
 - Airflow
 - Primi Test migrazione container
 - ascisoft
 - heasoft



Noi siamo arrivati qui!

Fase 5

Microservizi

- Migrare tutti i software di riduzione/simulazione su microservizi utilizzando pipeline automatizzate che avviano e distruggono i container a lavoro concluso.
- YAML su GIT
- Parallelizzazione dei run
- IAM

Fase 6

Rebuild

- Una volta passati i software su pod, possiamo ricreare il cluster in modalità fisica e non più in VM, eliminando Slurm, **tutto in una giornata**.
- Mantenendo solo lo storage Iperconvergente



Fase 7

Scale Up

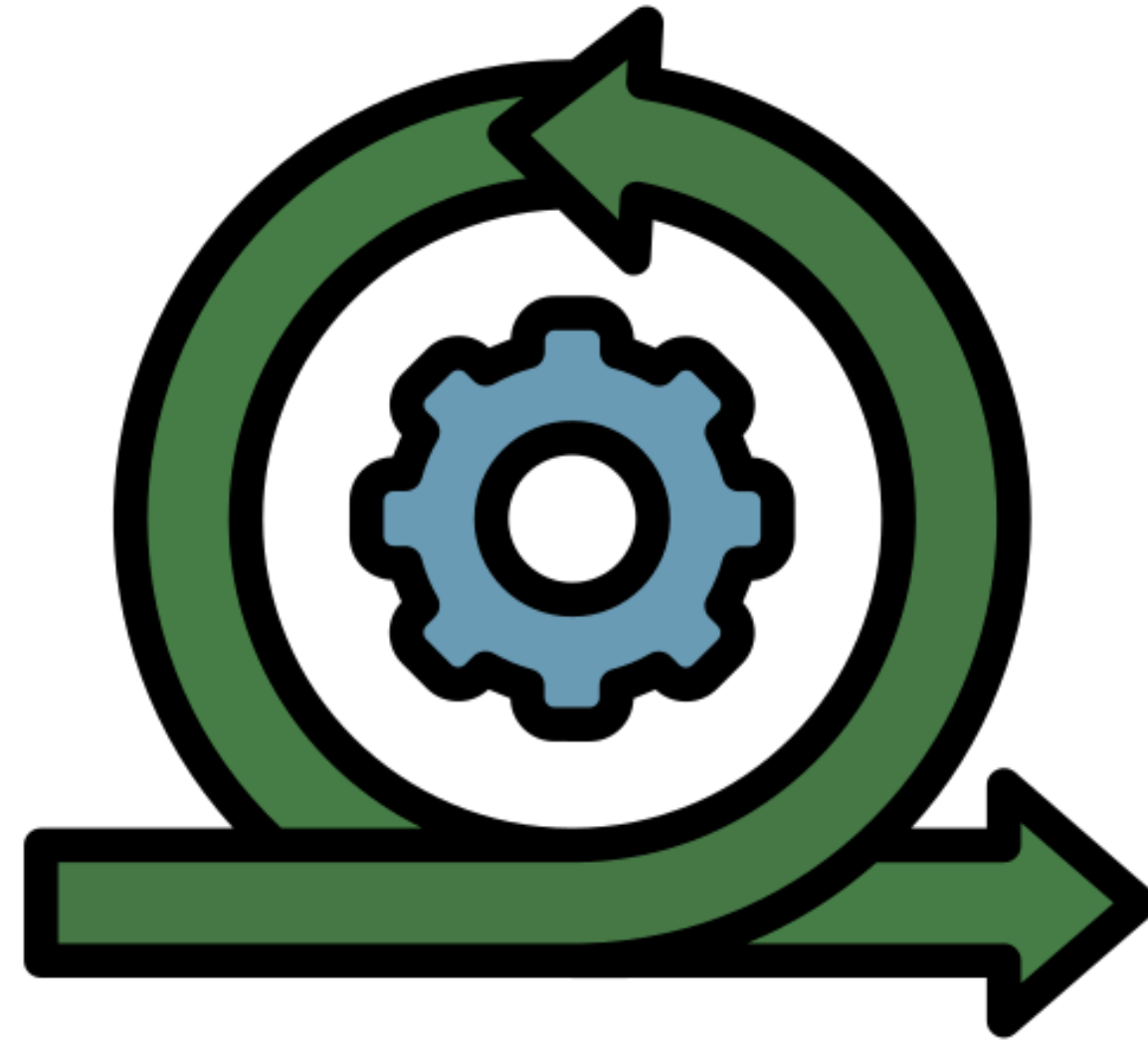
Aggiungendo uno Storage esterno come un fibre-channel si potrebbe eliminare GlusterFS per aumentare le performance ed avere delle aree di scratch locali



Fase 7

Scale Up

- Aggiunta di server fisici
- Aggiunta di Cluster Kubernetes
- Con dCache si potrebbe semplificare il trasferimento da e per eventuali altri cluster



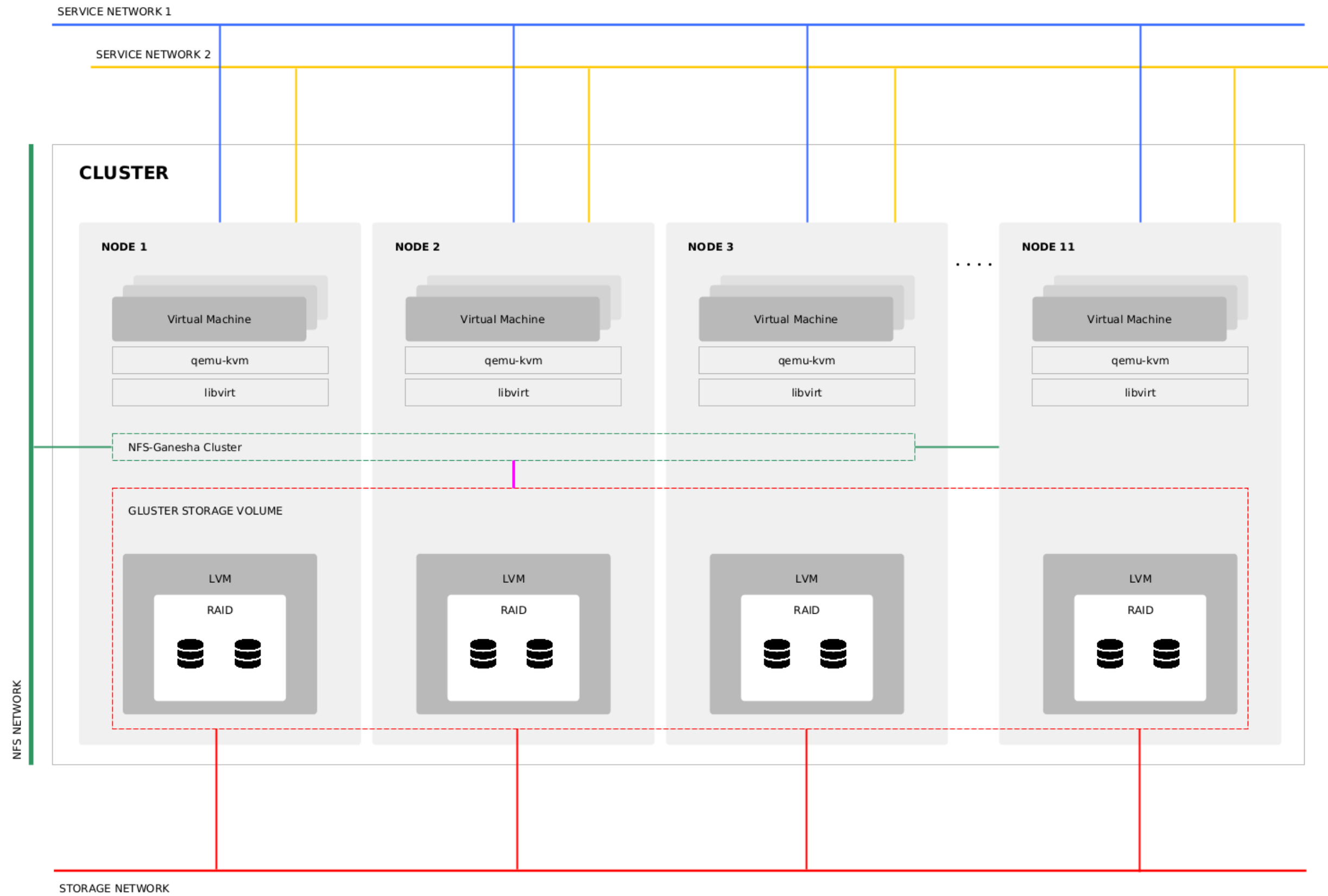
Processo Iterativo!

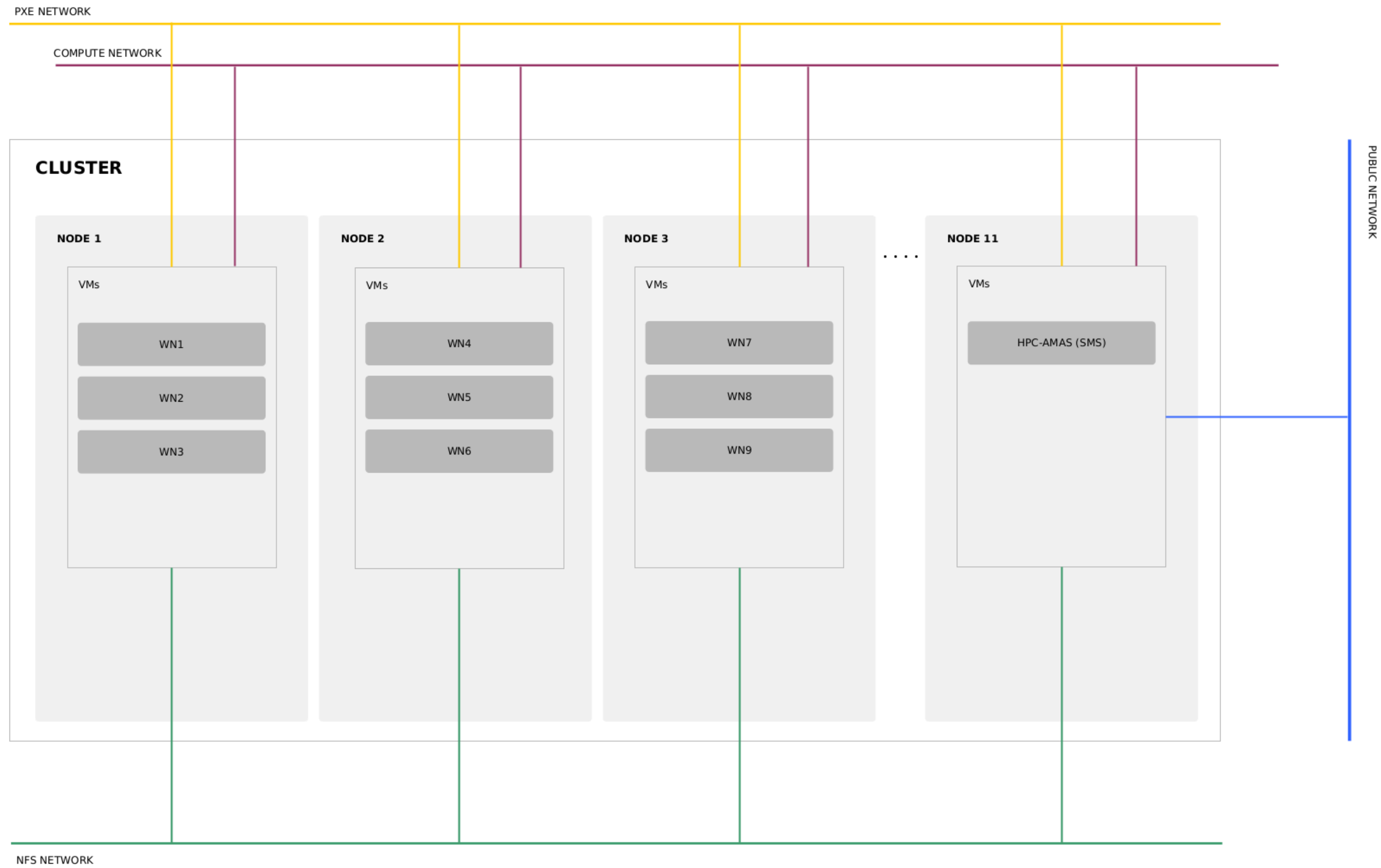


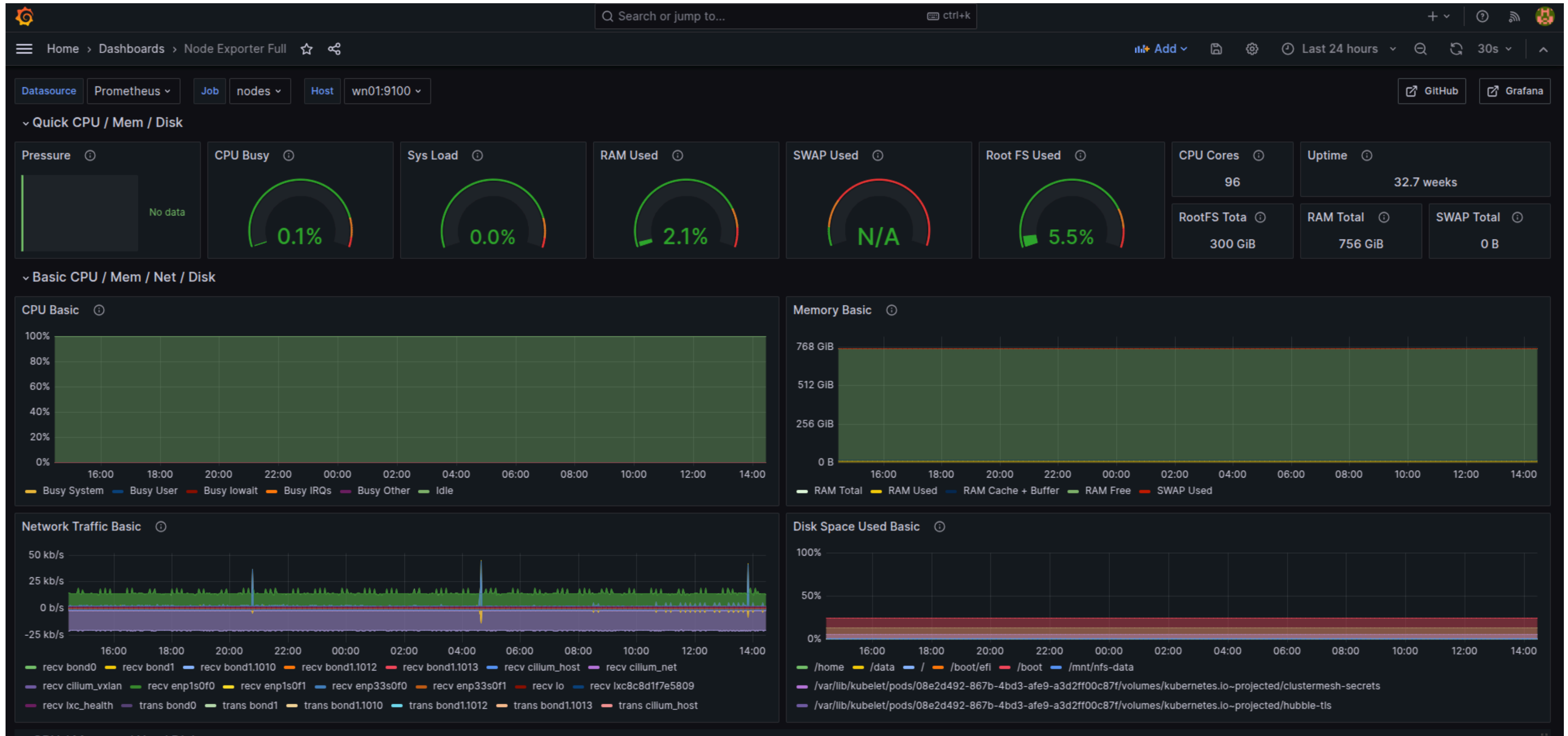
Domande?

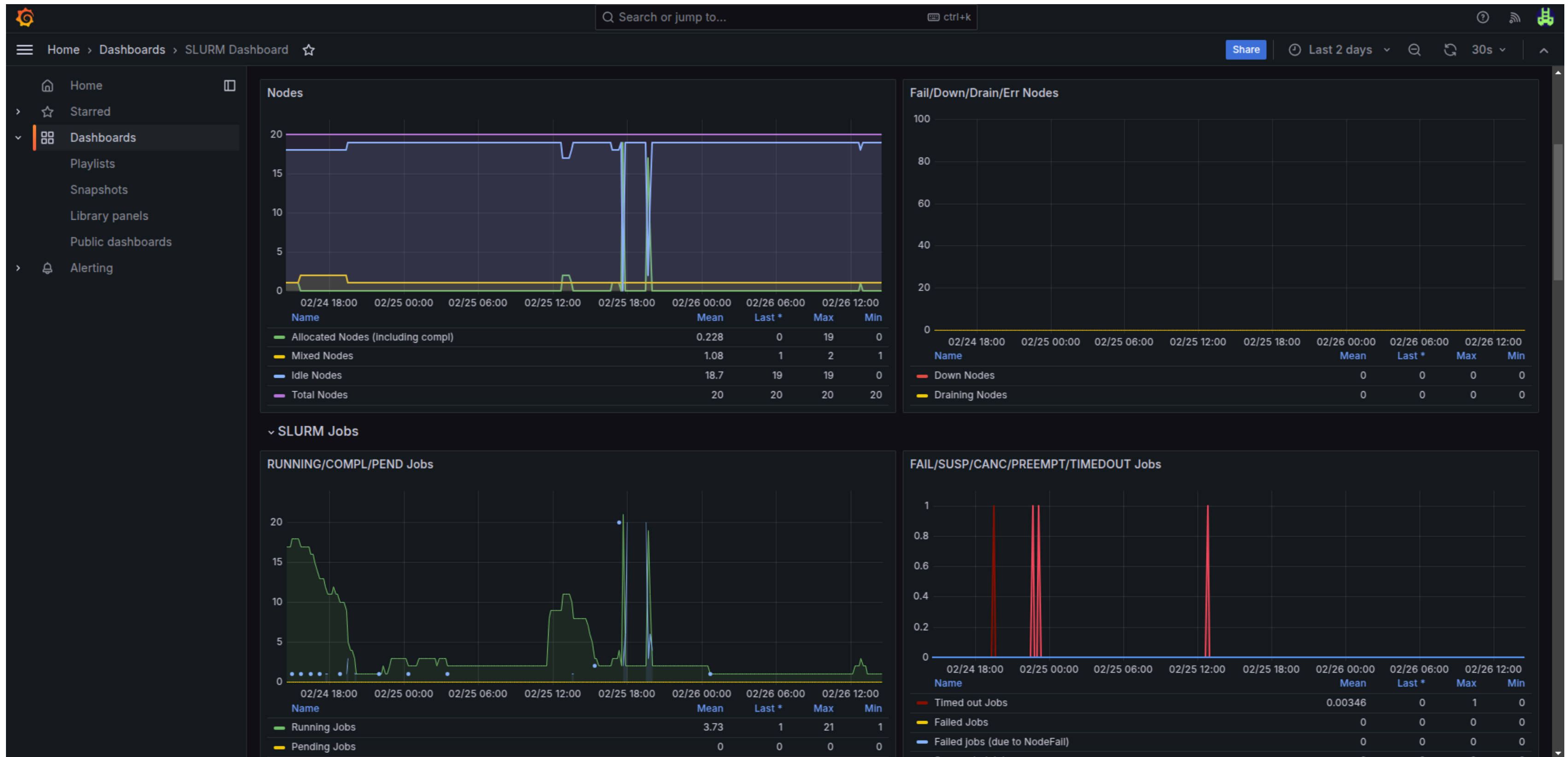


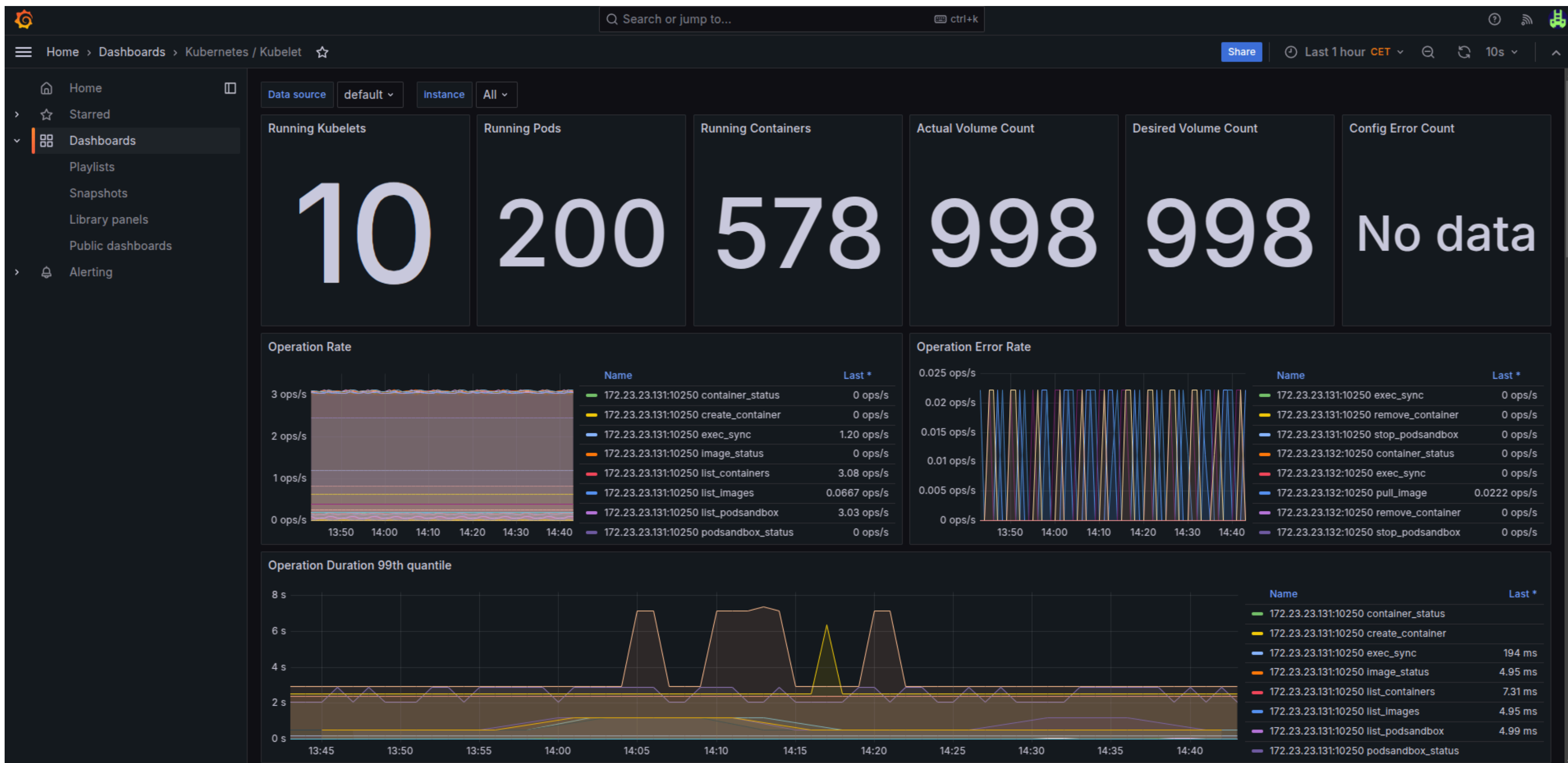
Altre Info













Grazie!