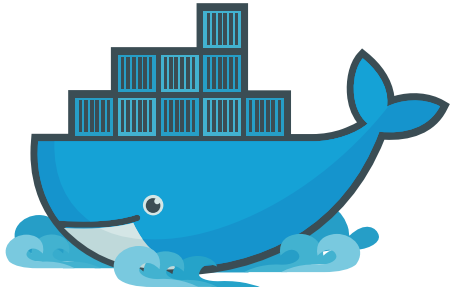
SYSTEM TESTING CLOUD SERVICES USING

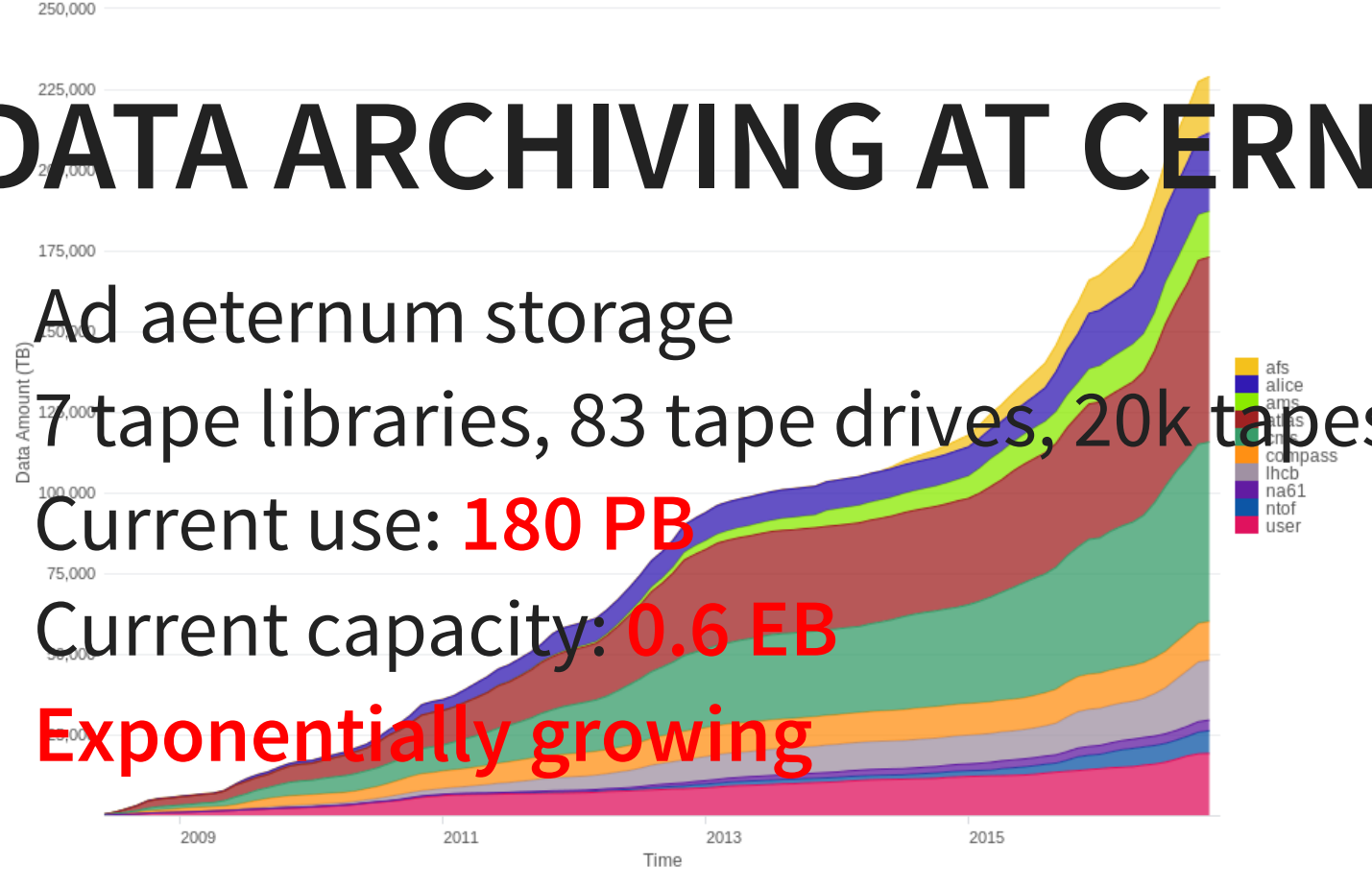


EOS + CTA DEVELOPMENT USE-CASE

Julien Leduc from **IT ST**orage group CERN

DATA ARCHIVING AT CERN

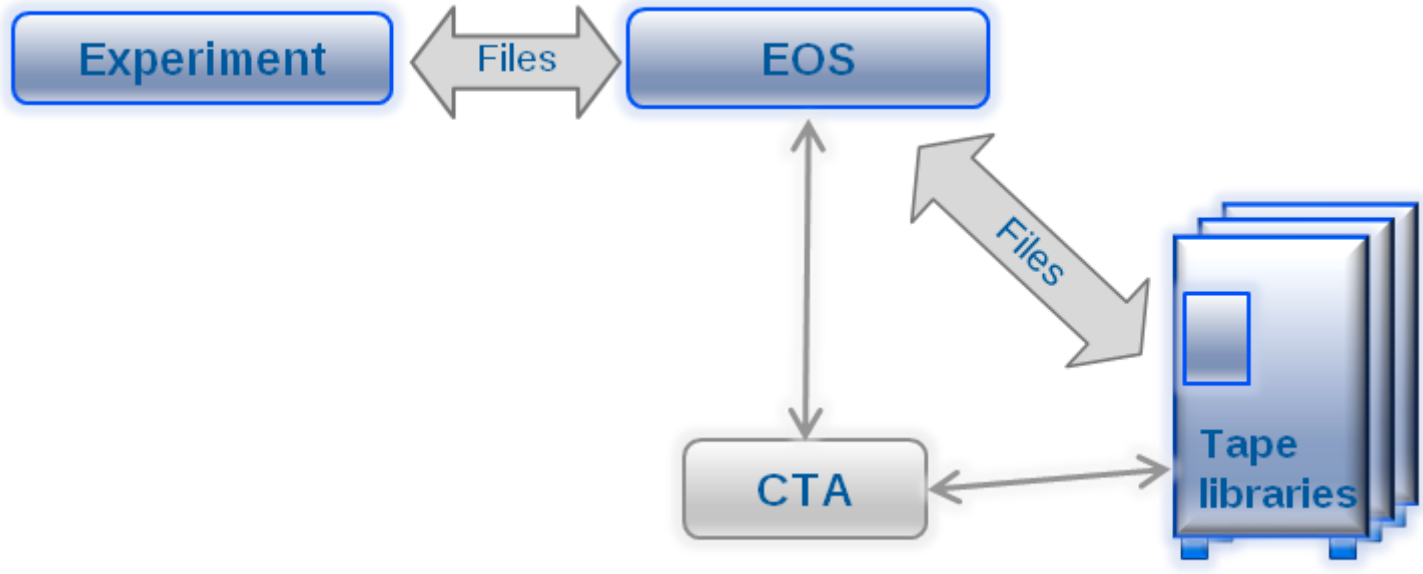
- Ad aeternum storage
- 7 tape libraries, 83 tape drives, 20k tapes
- Current use: **180 PB**
- Current capacity: **0.6 EB**
- **Exponentially growing**



DATA ARCHIVING AT CERN

EVOLUTION

- EOS + tapes...
  - EOS is CERN strategic storage platform
  - tape is the strategic long term archive medium
- EOS + tapes = ♥
  - Meet CTA: CERN Tape Archive
  - Streamline data paths, software and infrastructure



- CTA is glued to the back of EOS
- EOS manages CTA tape files as replicas
- CTA contains a catalogue of all tape files
- CTA provides optimised, preemptive scheduling

CTA DEVELOPMENT TIMELINE

- End 2016: First functional prototype release
- April 2017: First release for additional copy use cases
- 2018: Production-ready version

Easy migration path from CASTOR to EOS+CTA: only metadata need to be migrated CASTOR tape format will be reused.

CTA + EOS DEVELOPMENTS

This involves tightly coupled development in the intial phase for both software, and extensive testing to quickly catch regressions.

CASTOR INTEGRATION TESTS

- Easy situation:
  - all components are within **one git repository**
  - `Puppet` deploys development instances on VMs
  - **Limited external dependencies** per instance: 1 database, 1 virtual tape library

CASTOR INTEGRATION TESTS

- But several issues:
  - deploying a developer instance from scratch takes **loooonnng time...**
  - code changes in CASTOR often require Puppet **manifest change**
  - **real tape hardware** tests are way further down the road in separate hostgroups, environments...
    - which implies **ad hoc** developer tests...

CTA+EOS INTEGRATION TESTS

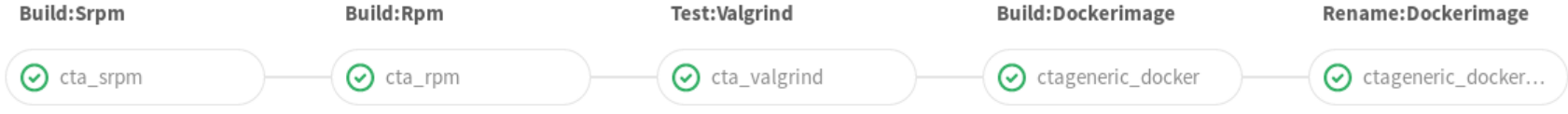
- Complex situation:
  - **2 distinct software projects**
  - **More external dependencies** per instance: 1 database, 1 virtual tape library, 1 objectstore

CTA+EOS INTEGRATION TESTS

- How to fix everything?
  - I am **lazy** and **impatient**
    - no manual operation → CI
    - make it fast
  - Must allow similarly easy beta testing deployments for administrators/users (simple and bulletproof)
  - How to test real tape hardware?

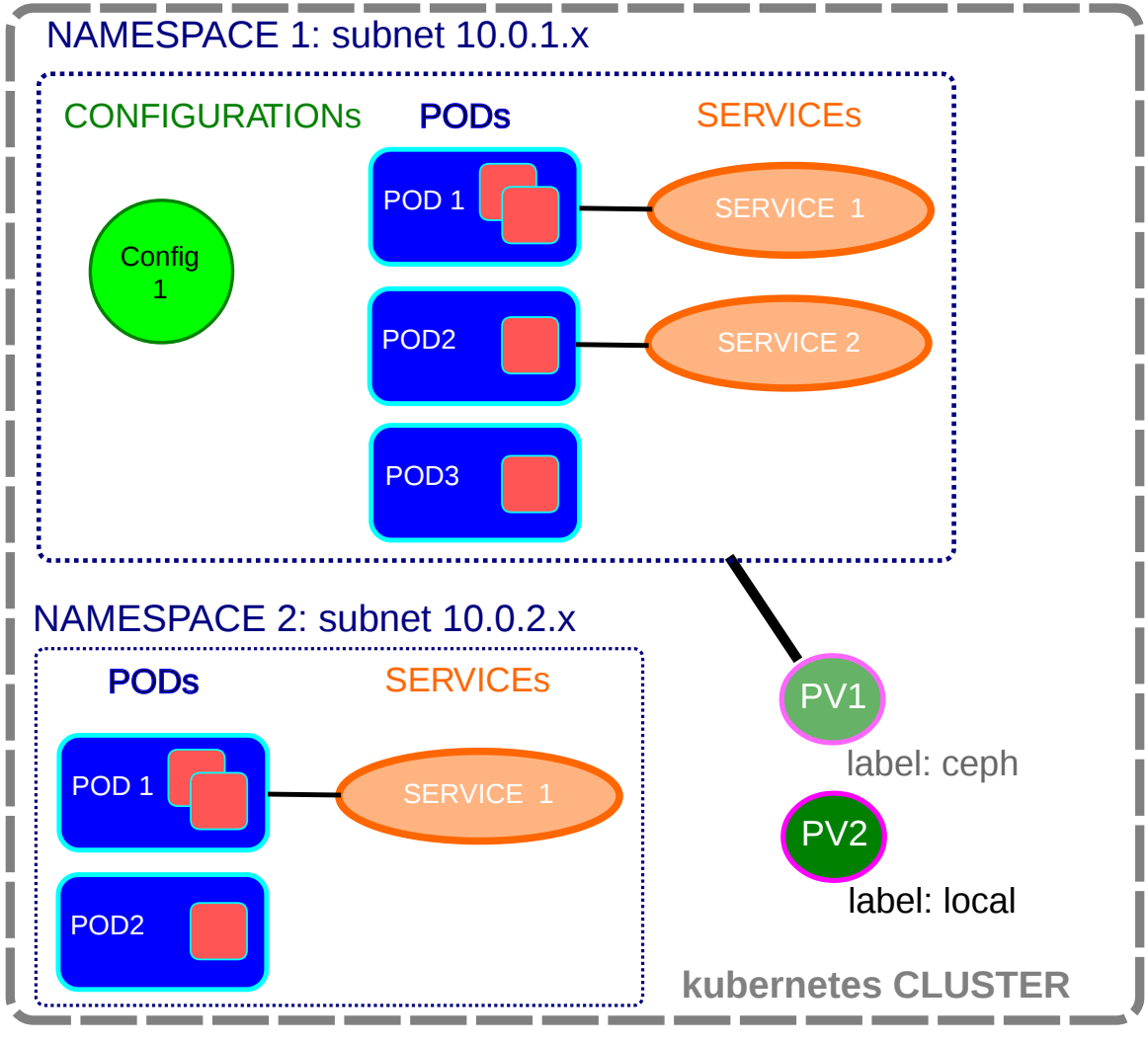
CTA CI

Implemented in CERN Gitlab instance



- Build software: CTA RPMs available as **artifacts**
- Build and publish a **generic Docker image** in gitlab registry
  - Contains **all required RPMs for instantiation** (CTA artifacts, specific EOS version, specific XROOTD version)
- Run **system tests** in custom `kubernetes` cluster

BASIC KUBERNETES CONCEPTS

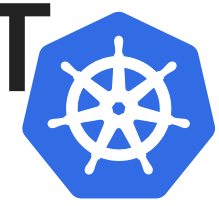


KUBERNETES RESOURCES

System tests on dedicated kubernetes clusters

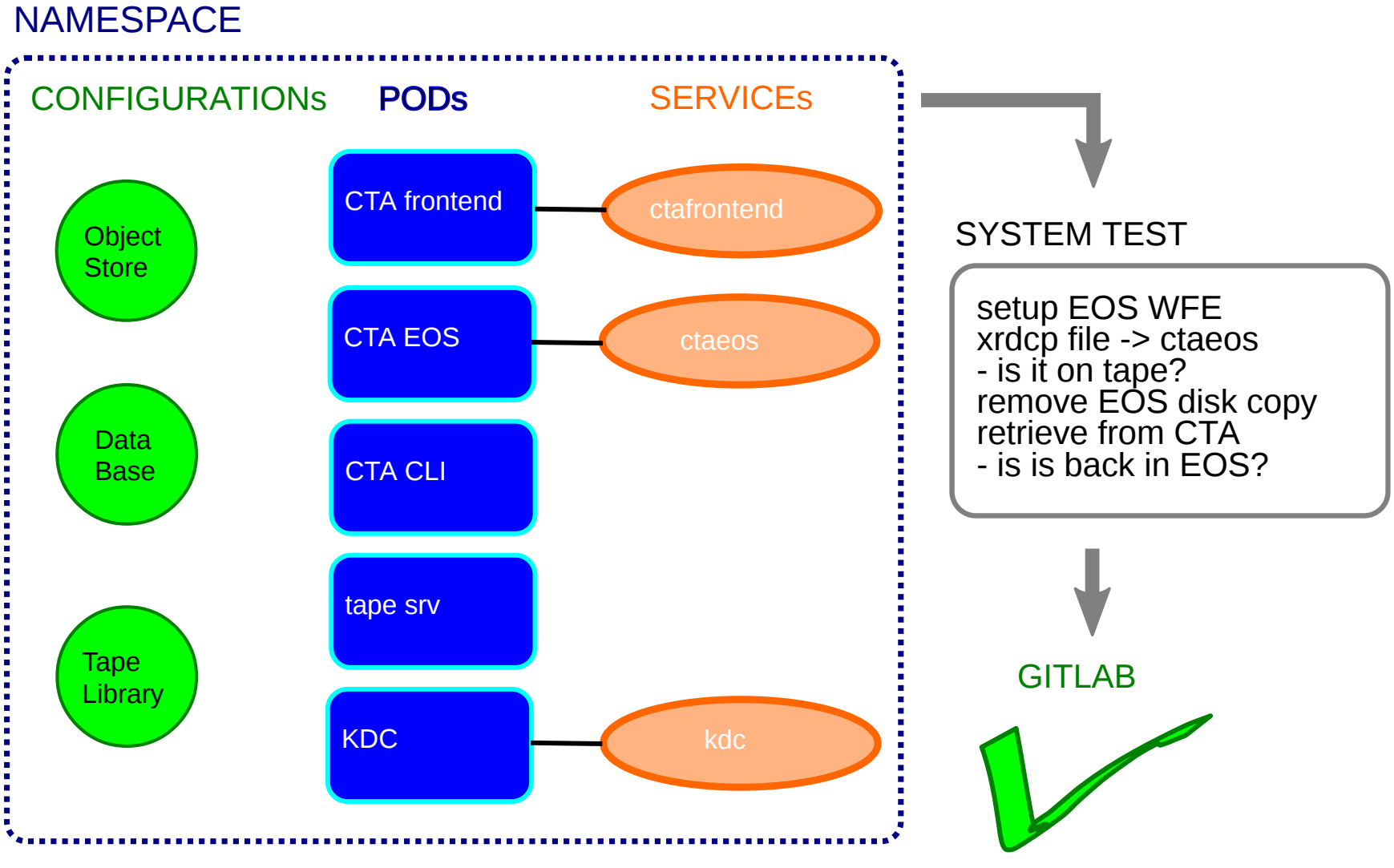
- One Puppet deployed kubernetes cluster per developer on one VM
- Kubernetes resources per cluster:
  - 1 Oracle database (+ unlimited sqlite accounts)
  - 1 Ceph objectstore (+ unlimited local objectstores)
  - 10 Virtual tape libraries: 2 tape drives, 10 tapes

INSTANTIATING A TEST



- Create k8 Namespace
- Instantiate all Services in the namespace
- Consumable resources are implemented as Persistent Volumes
  - Issue a Persistent Volume Claim with selector
  - Instantiate associated Configuration in the Namespace
- Instantiate all the Pods with their associated containers to implement all the services
- Wait for all the pods to be ready

INSTANTIATING A TEST



REAL TAPE DRIVE TESTS

- Deploy Puppet manifest on real hardware
- Add physical tape library resources in hiera
- Increase timeouts for system tests

VOILÀ!

We can deploy the same kubernetes instance on real tape hardware and run exactly the same system tests.

THE END

- Very powerful approach addresses all our use cases
- Fast, flexible, isolated and self contained in software repository

TO DO

- Write more system tests (using jtest? scripts?...)
- Bulletproofing reproducibility for regression tests
- Evaluate possible production use ☺

QUESTIONS?