

Julien Leduc, Cédric Caffy, Eric Cano, German Cancio Melia, Michael Davis, Steven Murray, Vladimir Bahyl - CERN IT

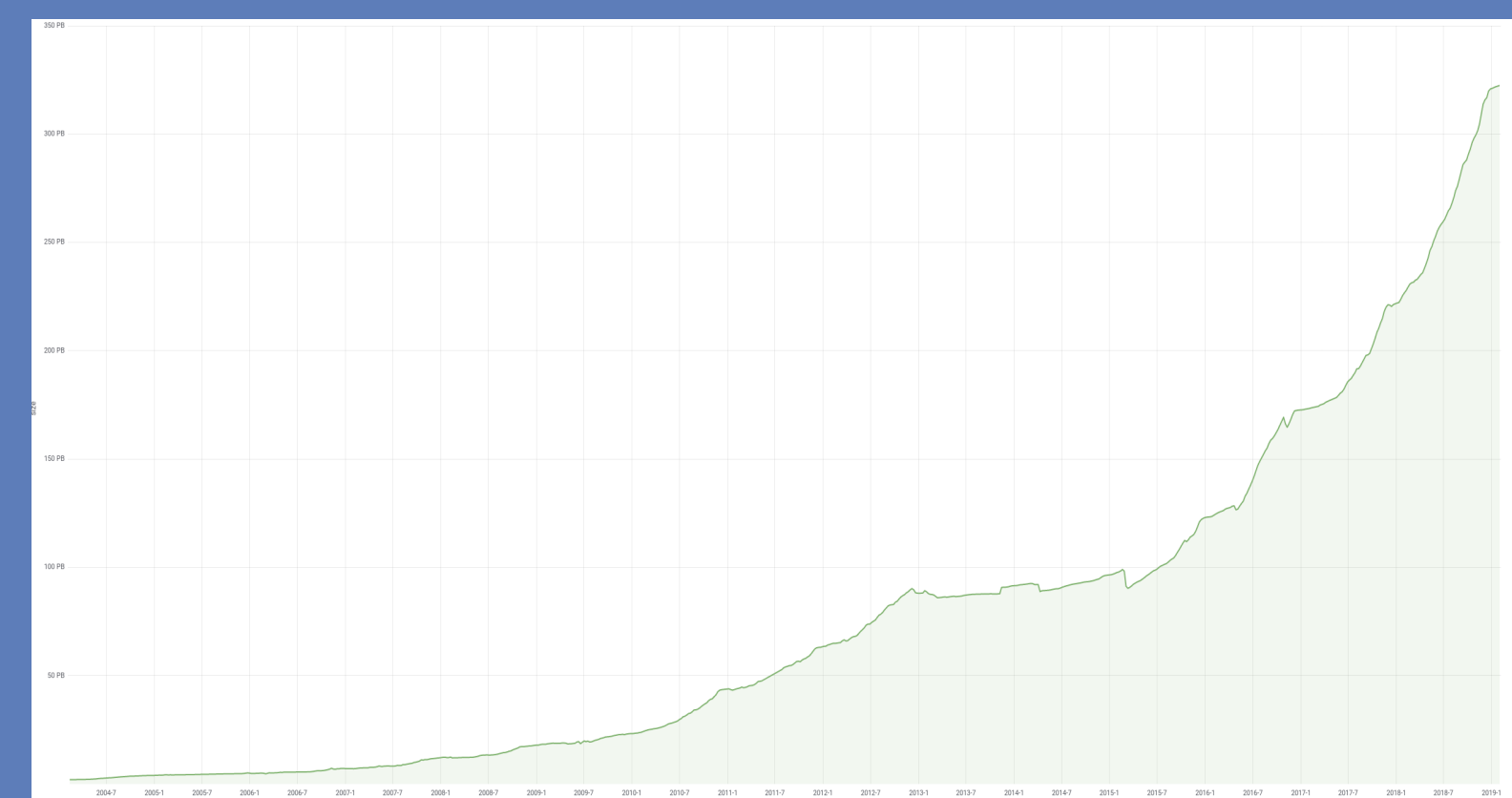
CERN storage architecture is evolving to address run3 and run4 challenges. CTA and EOS integration requires parallel development of features that needs to be synchronized and systematically tested on a specific distributed development infrastructure for each commit in the code base. CTA Continuous Integration development initially started to run functional system tests against the freshly built software. But its importance grew over time to include all development, testing and deployment aspects.

CERN Tape Archive

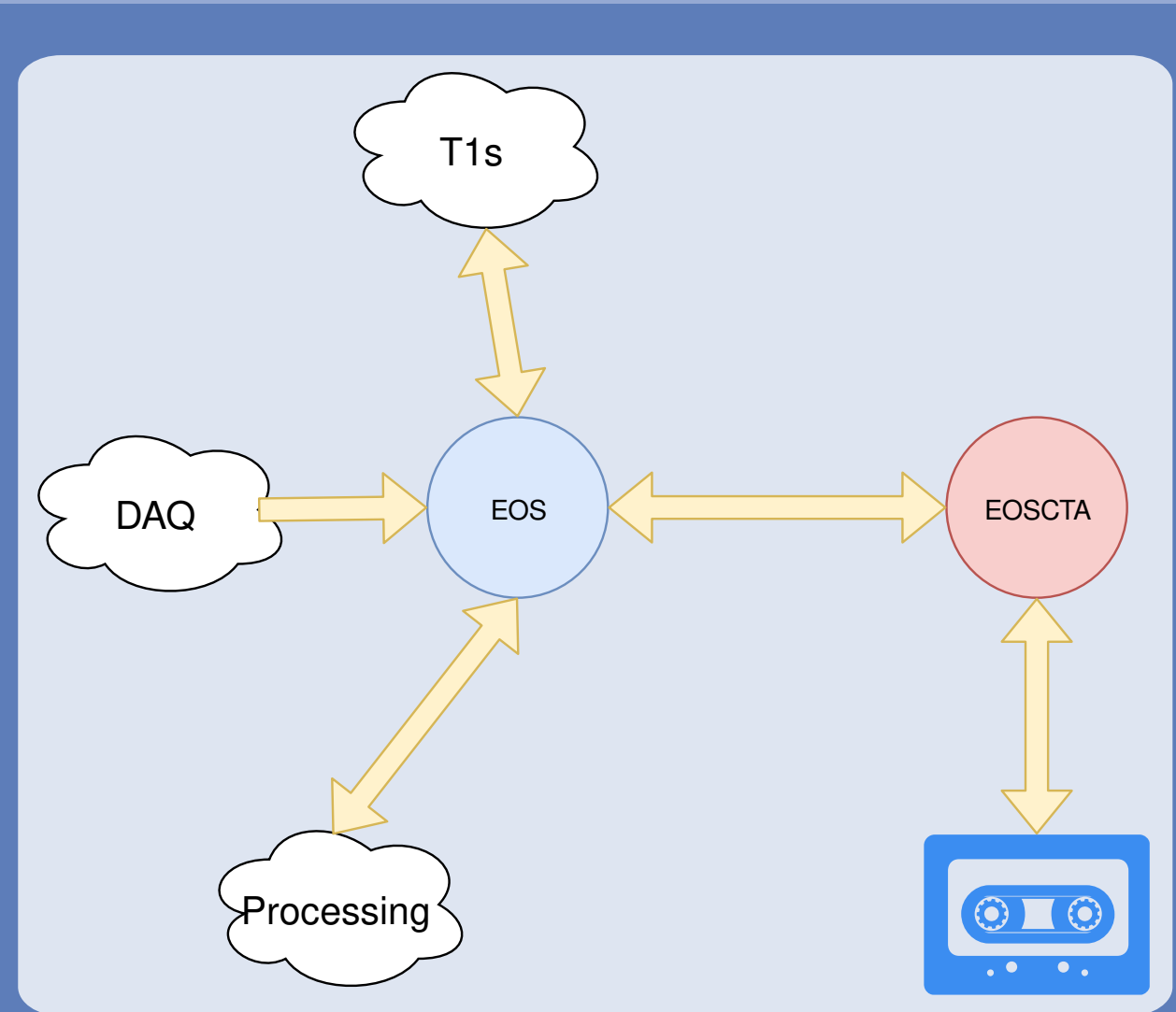
- Data archiving at CERN
- **ad aeternum** storage
- Current use: **340 PB**
- **Exponentially growing**

Run2: 7 tape libraries, 83 tape drives, 30k tapes

Run3: 4-5 tape libraries, 160+ tape drives, 150PB+/year, >40GB/s



CERN Physics Tape Archive volume history



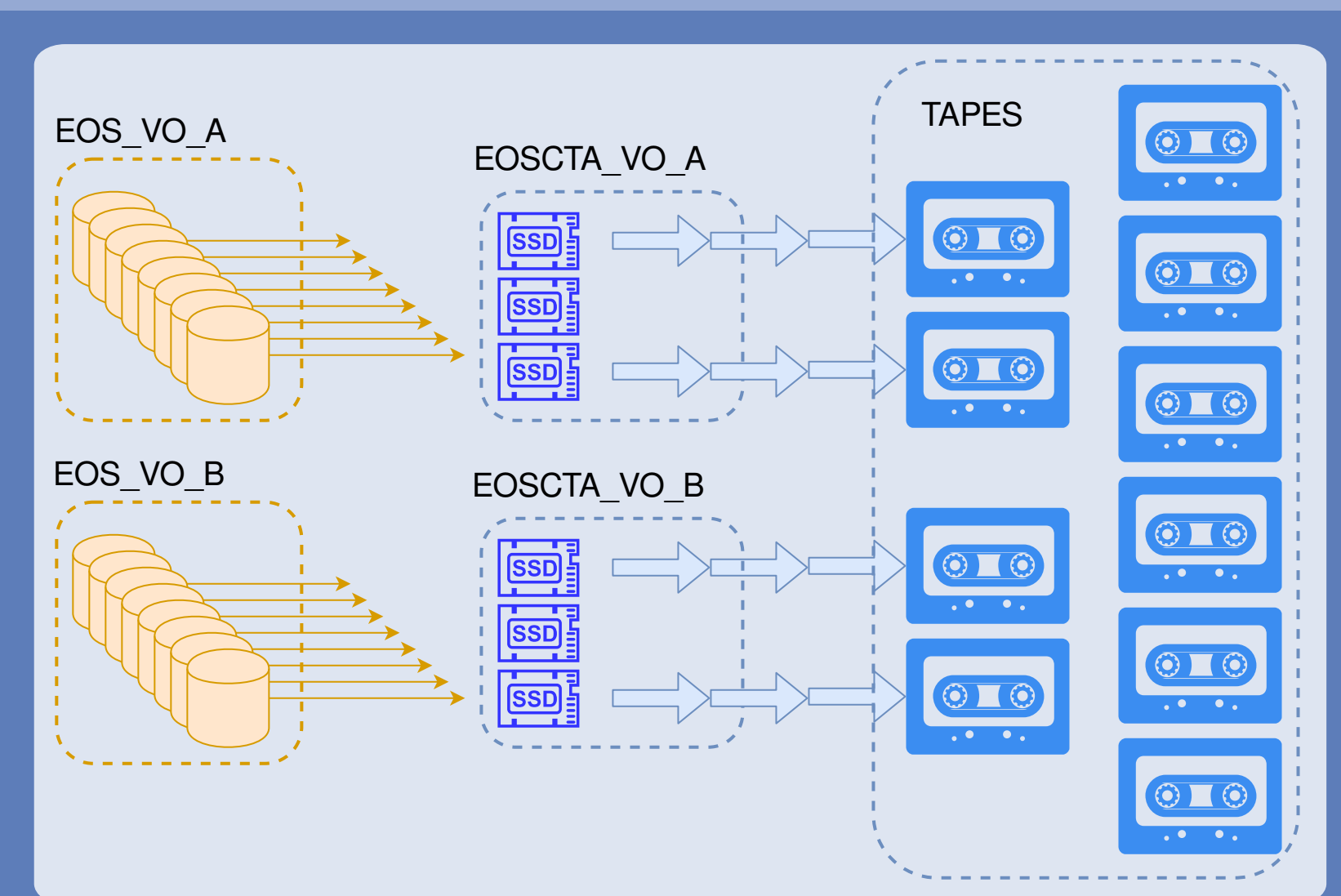
Data archiving at CERN Evolution

EOS + tapes...

- EOS is CERN strategic storage platform
- tape is the strategic archive medium

EOS+ tapes = ❤️

- Meet in CTA: CERN Tape Archive
- Streamline data paths, software and infrastructure



EOS+CTA Architecture

EOSCTA is a **pure tape system**

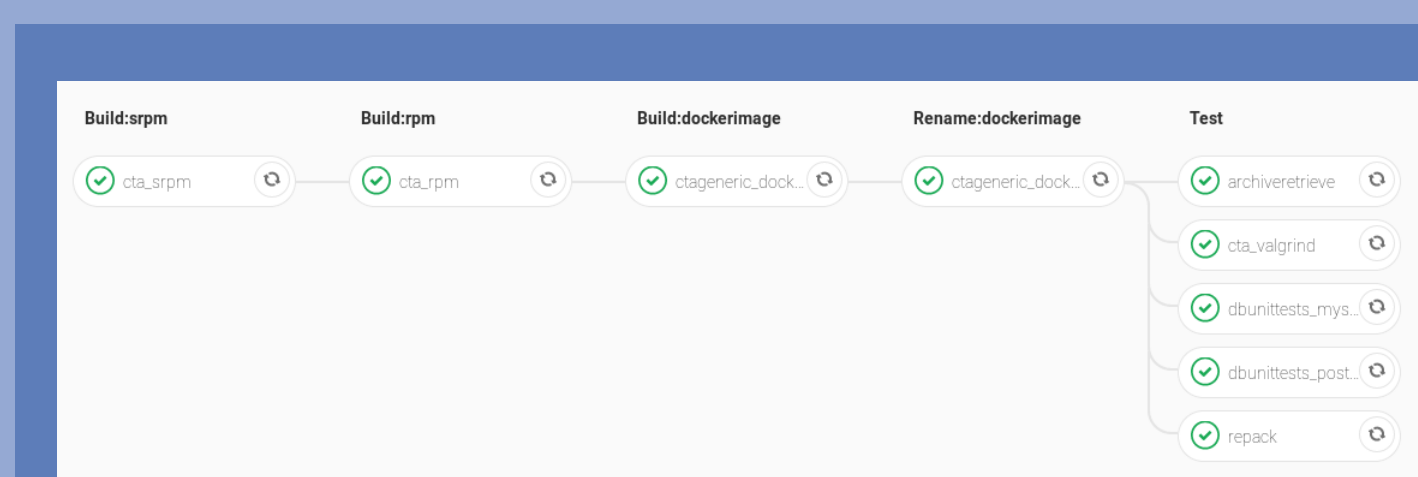
Disk cache is consolidated in each experiment EOS instance

Operating tape drive at full speed full time efficiently requires a **SSD based buffer**

CONTINUOUS INTEGRATION

Implemented in **CERN Gitlab instance**

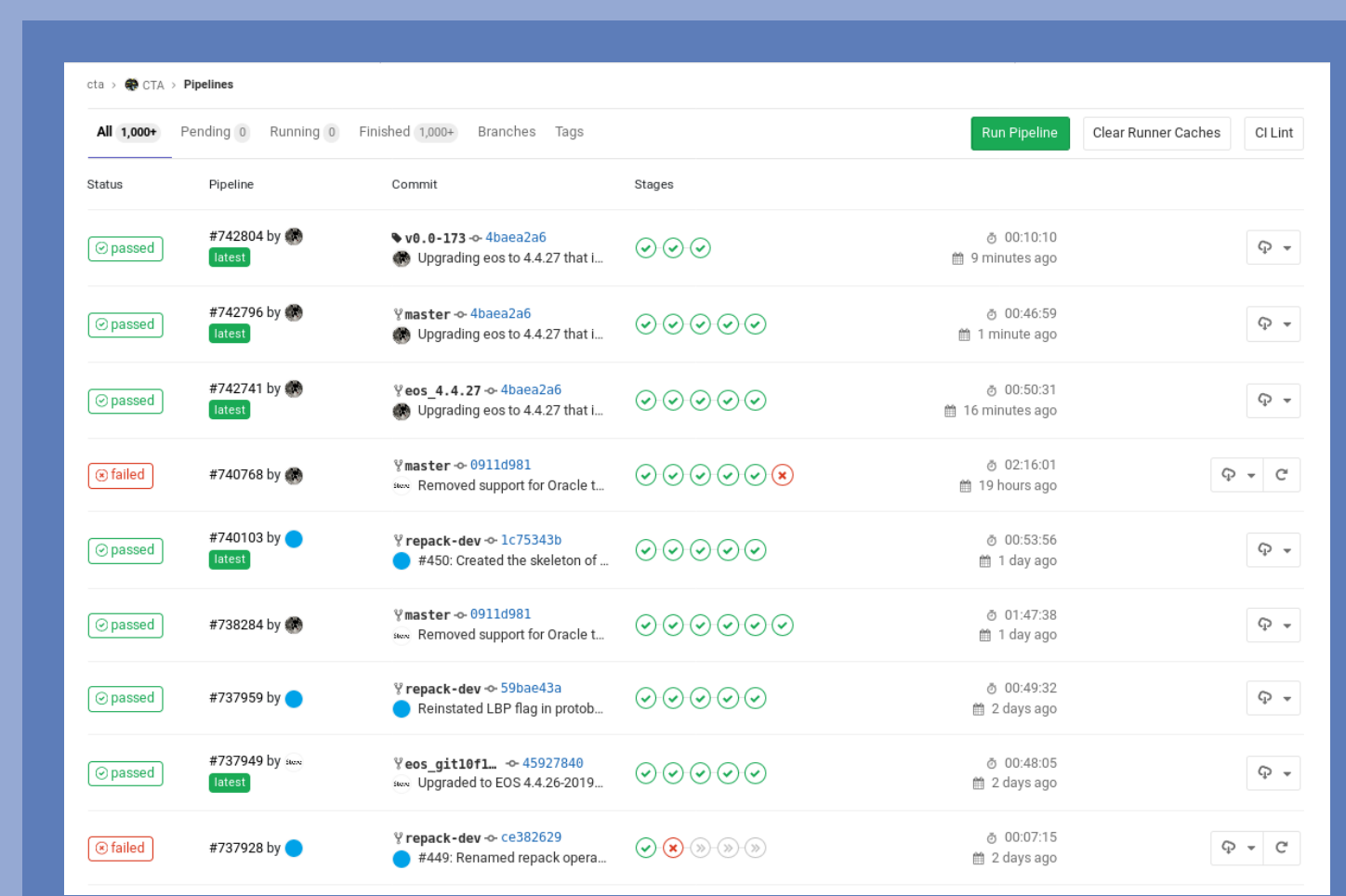
- Deploys Kubernetes framework on a custom gitlab runner
- When instance ready run a test that:
 - archive 10k files to EOSCTA
 - delete the 10k disk copies
 - retrieve 10k files from tape



Full Gitlab CI pipeline

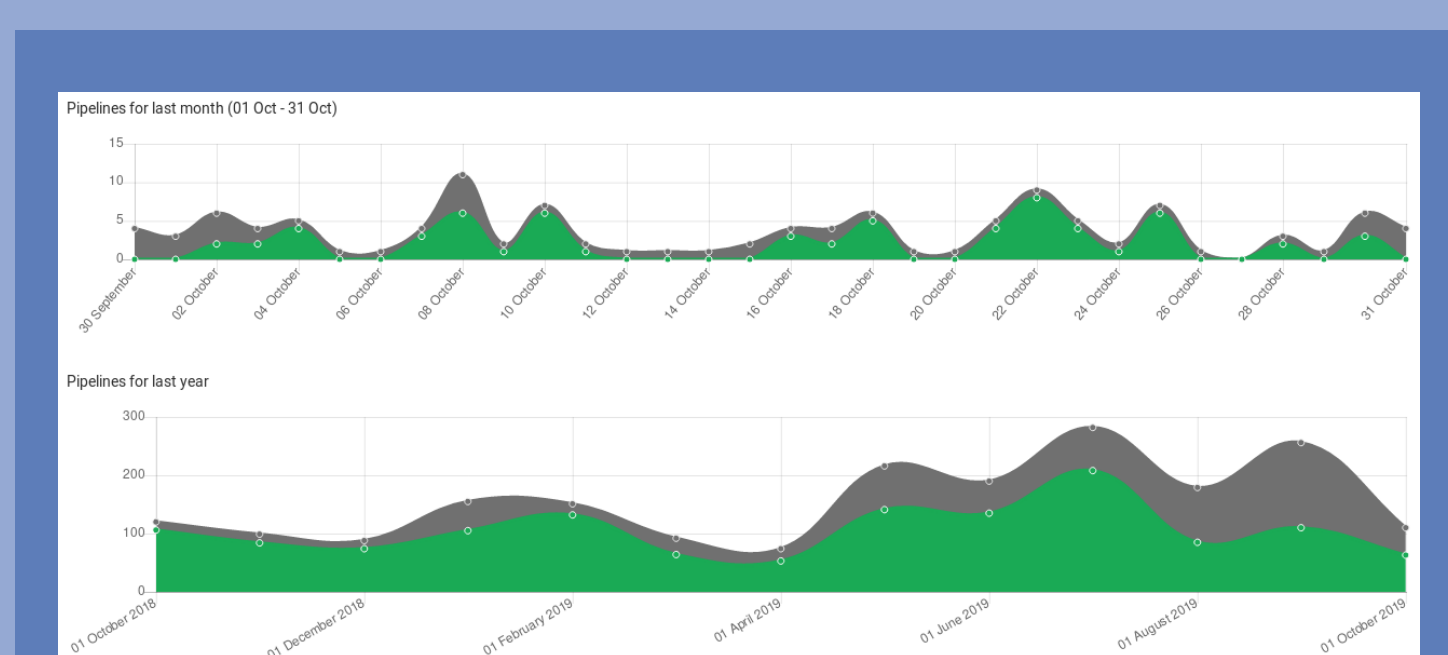
The full continuous integration pipeline runs at **every commit** and exercises its code along with its installation procedures until the final archival+retrieval test.

All git branches are tested allowing developers to **continuously test their branch** and the merged result.

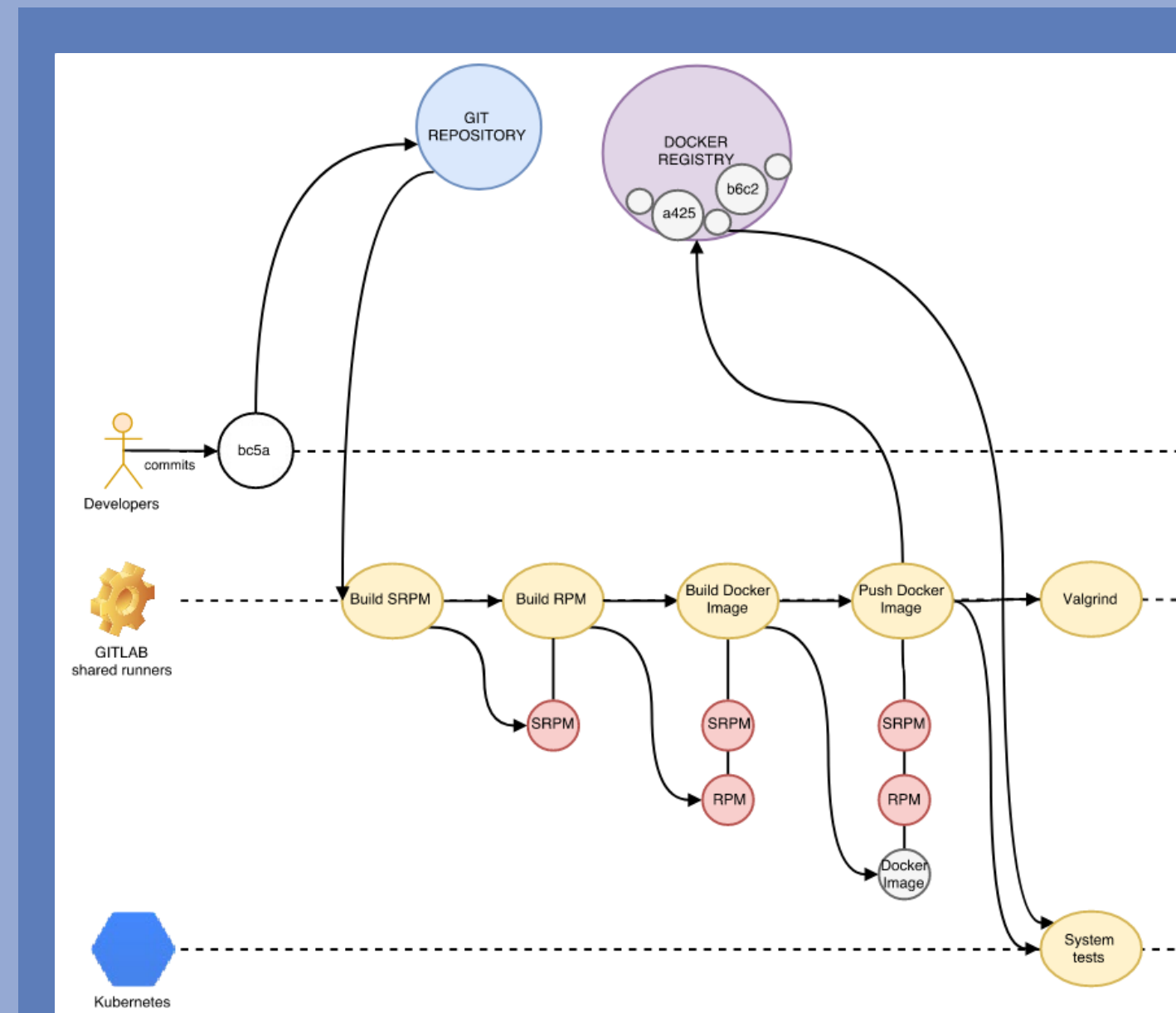


Several commits and corresponding CI

Approximately **5000 pipelines** ran since continuous integration is in place

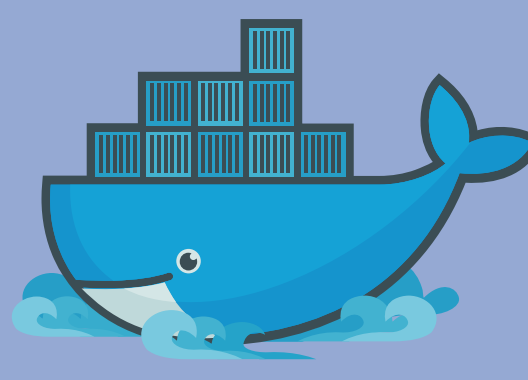


Several commits and corresponding CI



Full workflow for a single commit

EOSCTA Kubernetes Instance

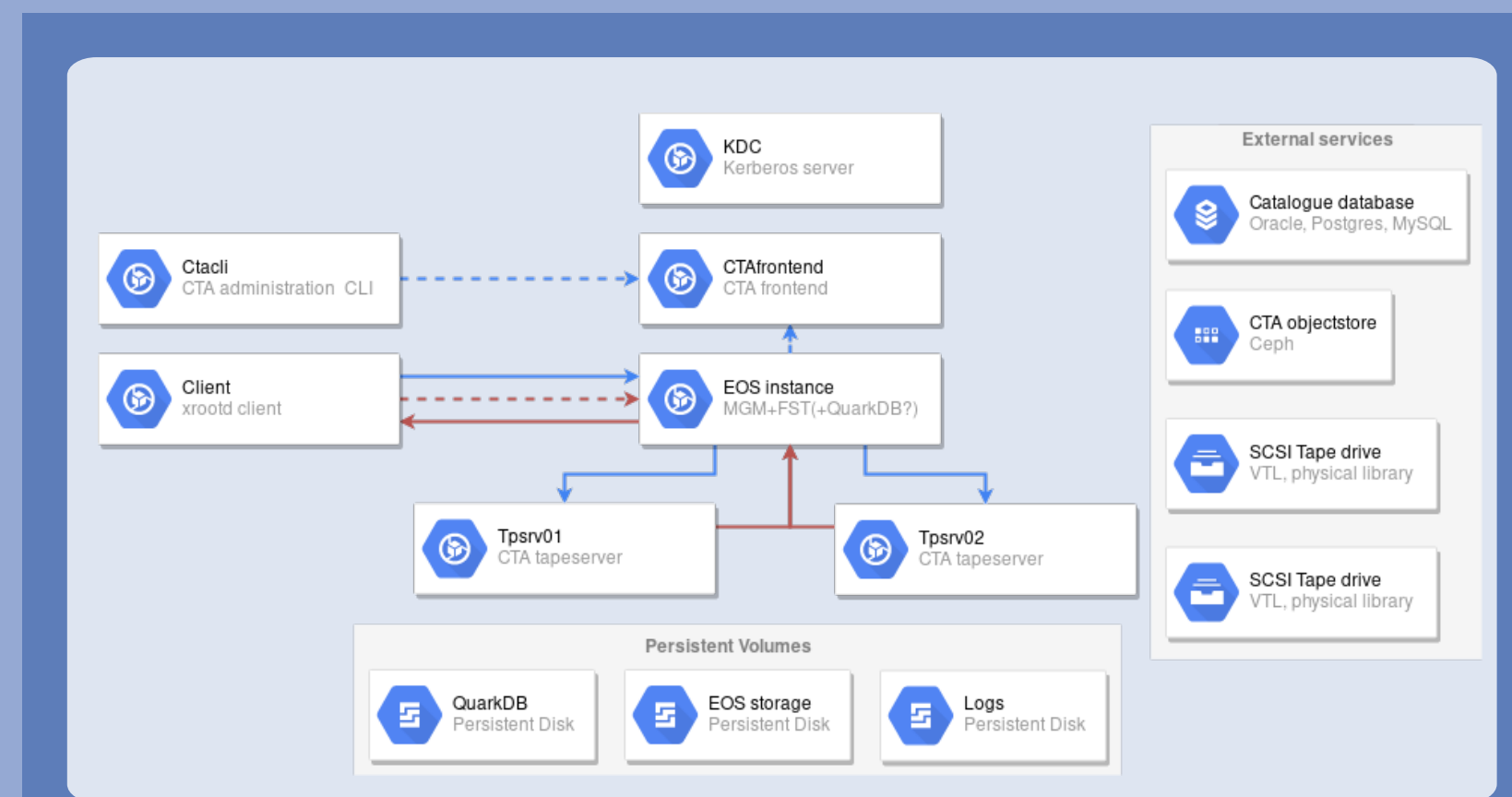


Tightly coupled software → **tightly coupled developments**

Extensive and systematic testing is paramount to limit regressions



- Implement a framework based on a **single generic docker image**
- Use **Kubernetes** to build an EOS CTA instance out of it
- Flexible enough to accomodate any supported resource (database, objectstore, tape library)
- **Part of CTA code repository:** CI tests are evolving along with CTA code.



EOSCTA generic Kubernetes instance

OTHER USE CASES

DEVELOPERS

Entirely runs on an **offline developer laptop**

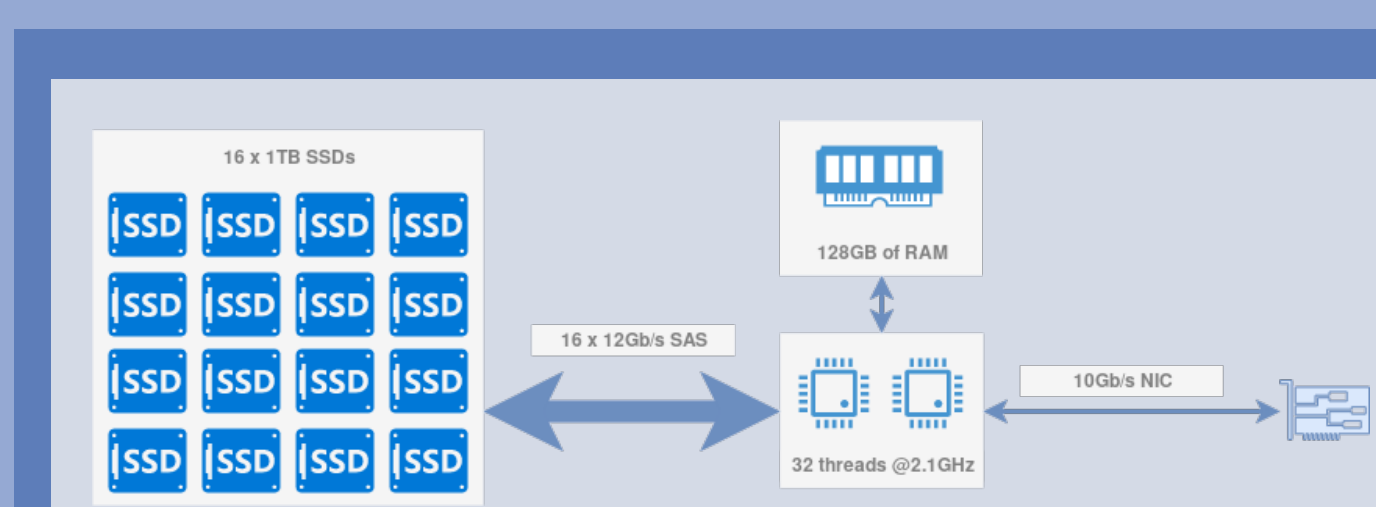
- Instantiates kubernetes framework in a CentOS7 Virtual Machine
- Offline resources only:
 - local Postgres instance
 - local file based objectstore
 - MHVTL

When instance ready run **specific developer test**

Strengths

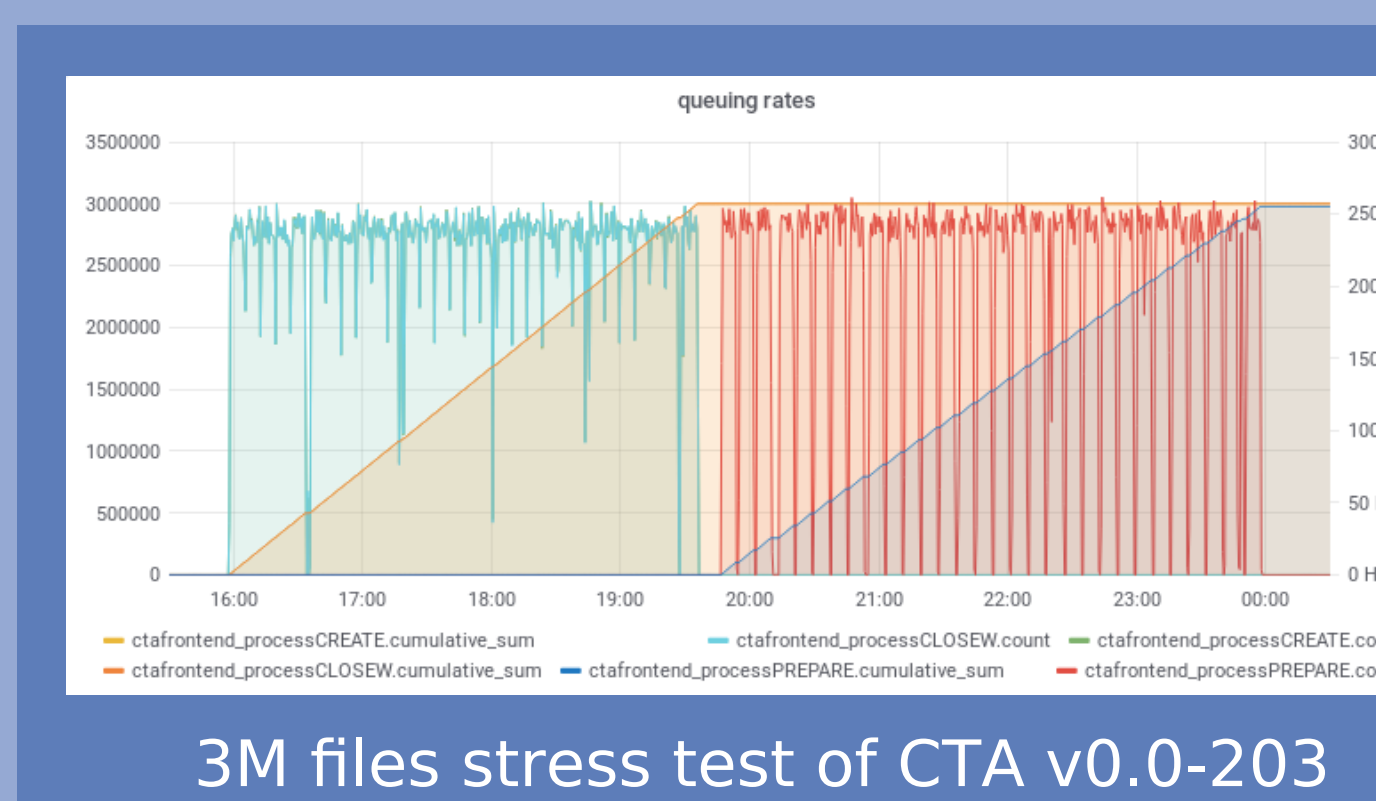
- **Much shorter learn curve** for new comers that can focus on their work
- **Best deployment practices** included.
- Successfully used for:
 - Objectstore developments
 - Database catalogue backend developments (`MySQL` , `Postgres`)

STRESS TESTING RELEASE CANDIDATE



Hyperconverged server for stress tests

Fast turnover environment that allows to quickly **reproduce a bug again and again under various conditions**

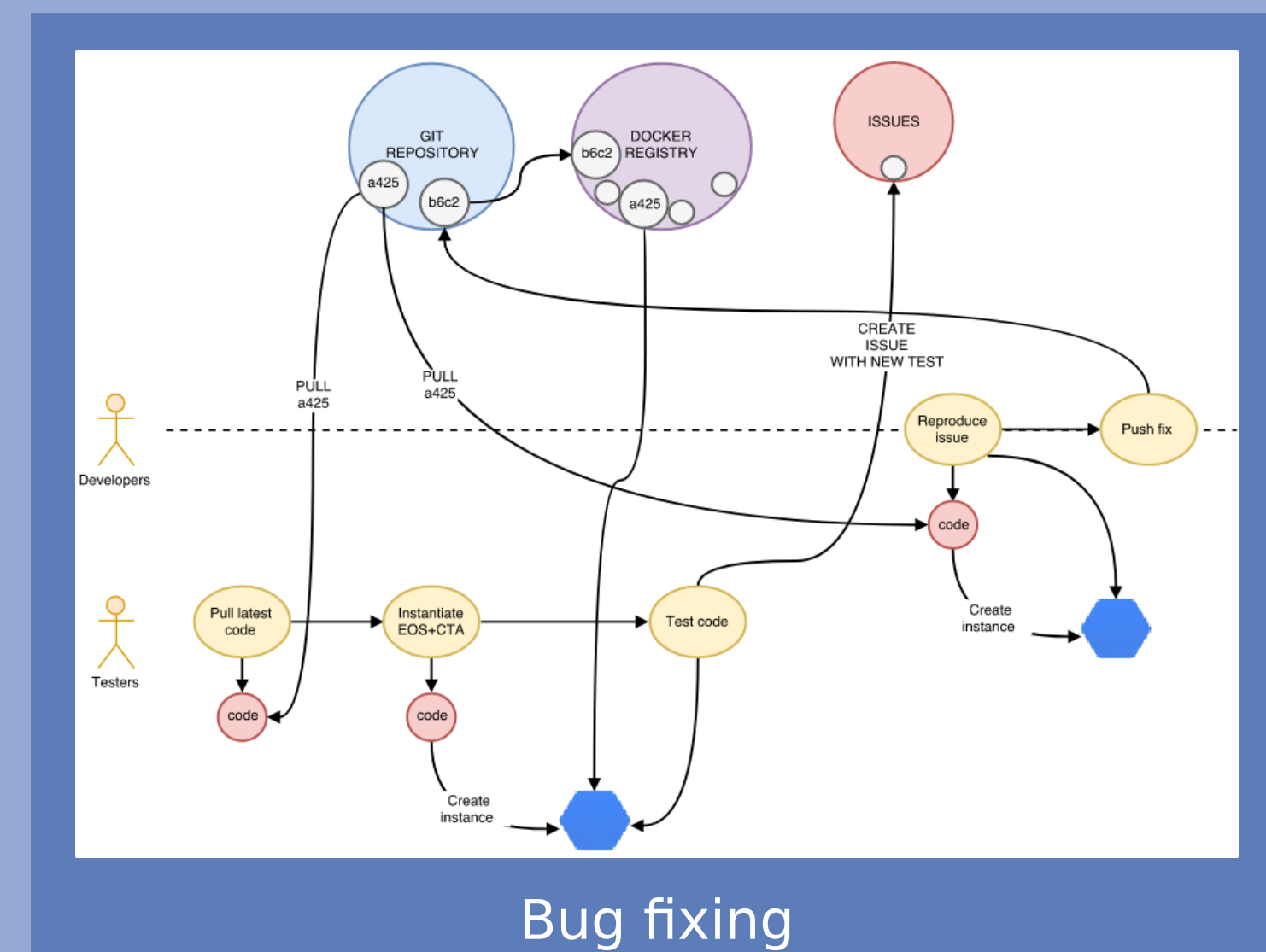


3M files stress test of CTA v0.0-203

Implements EOSCTA kubernetes framework on hyperconverged server:

- **Plenty of IOPS** for file rate tests
- **Plenty of bandwidth** to simulate a sizable CTA instance (equivalent to 10 physical tape servers + 6 disk servers).

Launch a CI stress test that archive and retrieve 3M files to EOSCTA.



Bug fixing

MORE USE CASES

Very powerful approach that **addresses and federates all our development/testing use cases**
Fast, flexible, isolated and self contained in software repository

EXTERNAL COLLABORATIONS

With developers: IHEP added MySQL support to CTA developing with kubernetes framework.
With T1s: self contained environment that provides deployment recipes that can be adapted by T1s.
With experiments: add FTS and RUCIO inside kubernetes?

PRODUCTION USAGE?

Deploy CERN production EOSCTA instances inside kubernetes?