
# **BigHPC**: Management Framework for Consolidated Big Data and High-Performance Computing

Speaker: Samuel Bernardo (LIP) on behalf of BigHPC consortium

BIG HPC HIGH PERFORMANCE COMPUTING

Cofinanciado por: COMPETE 2020 Lisb@20[20] PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

FCT Fundação para a Ciência e a Tecnologia UTAustin Portugal

# Outline

- The BigHPC consortium
- Overview of BigHPC platform and development challenges
- Why GitOPS and how to get it in practice
- GitOPS applied to BigHPC
- Next steps

# BigHPC consortium

## Austin, USA

UT Austin

TACC

## Portugal

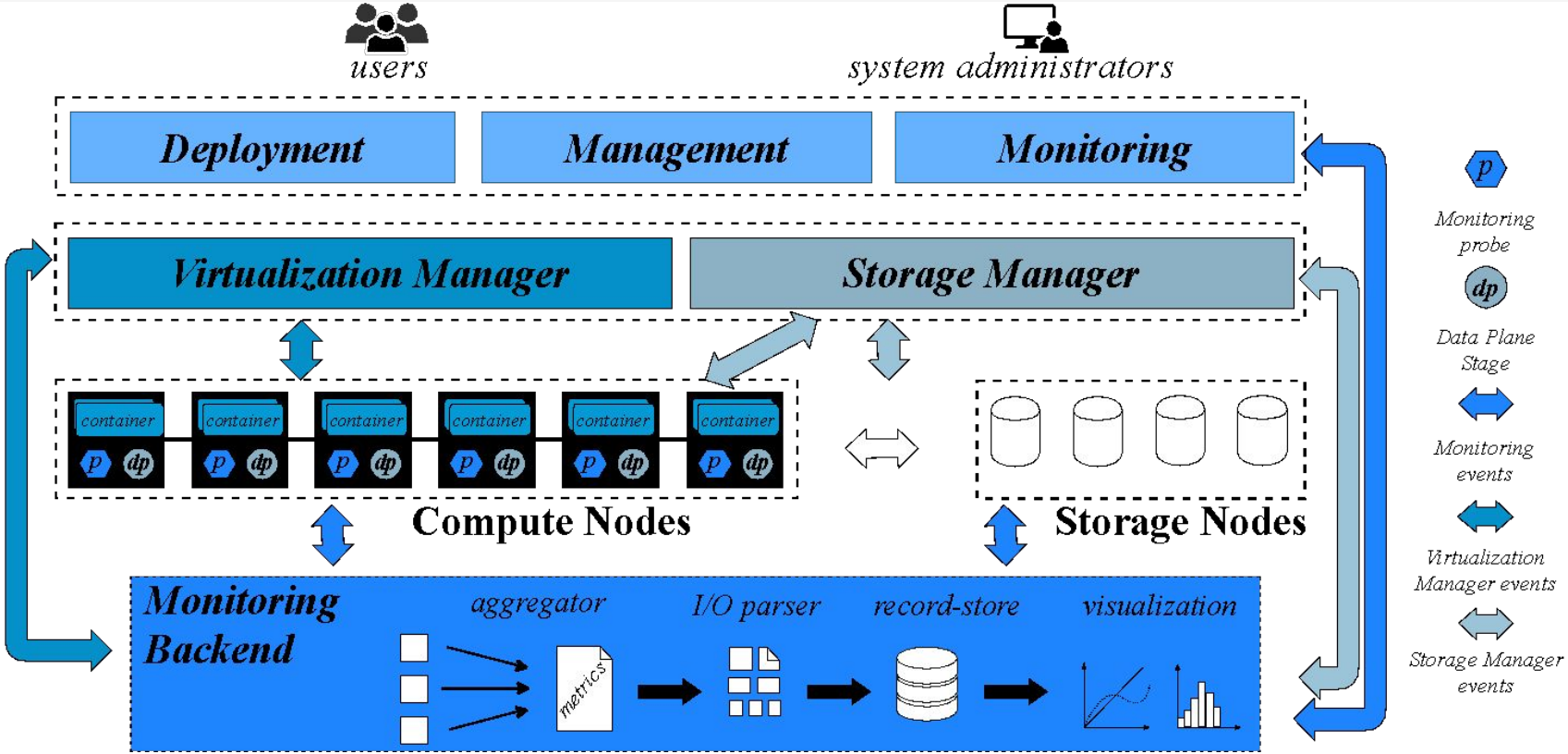
Wavecom

INESC TEC

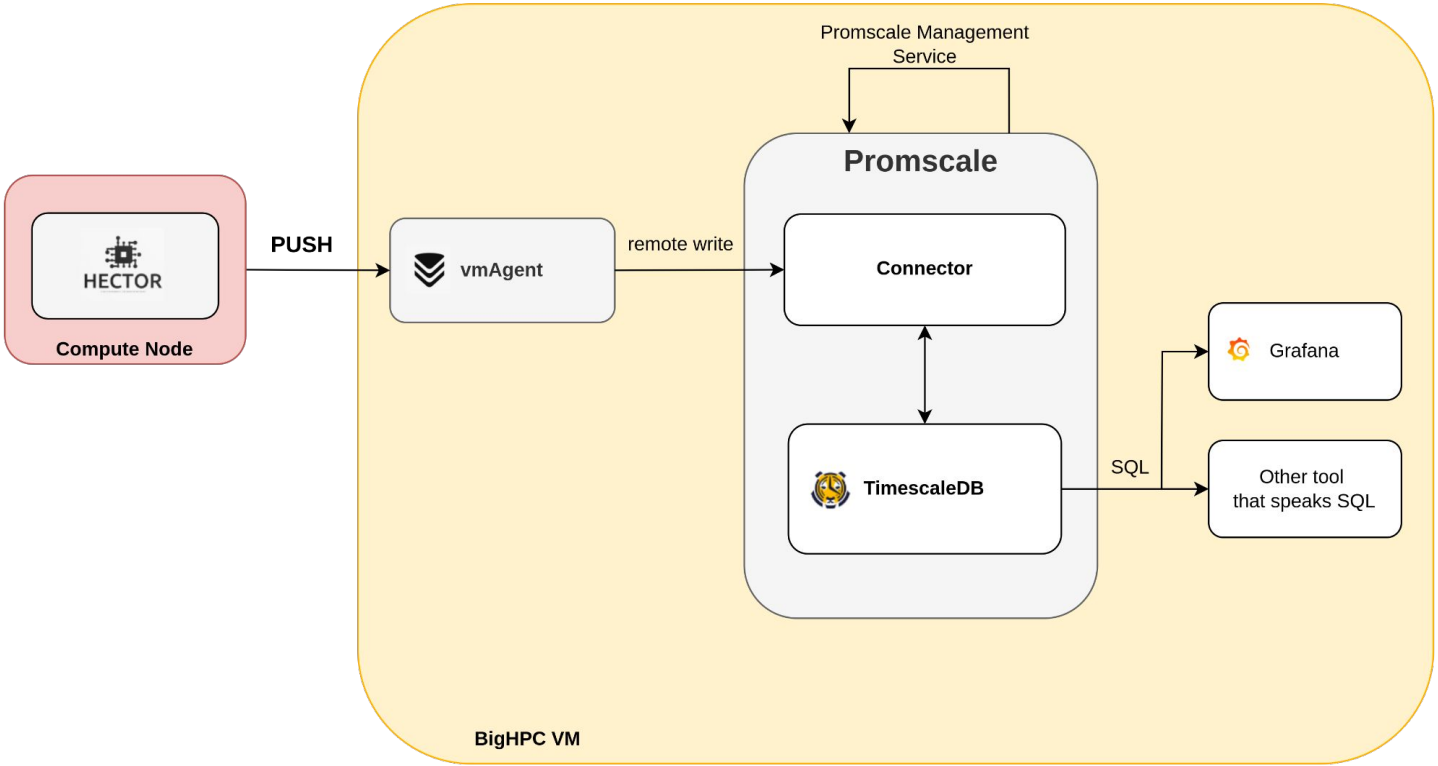
LIP

MACC

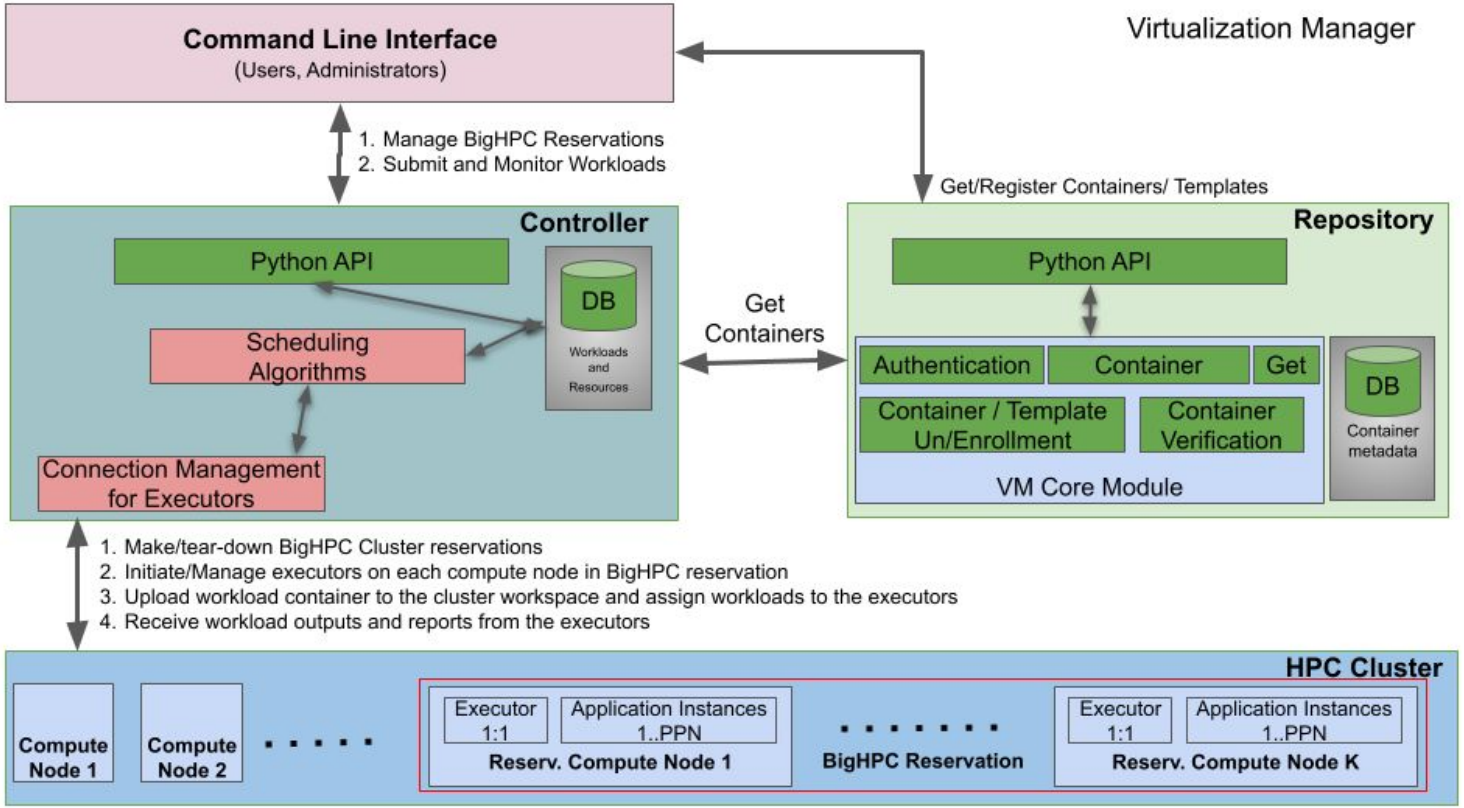
# BigHPC platform overview



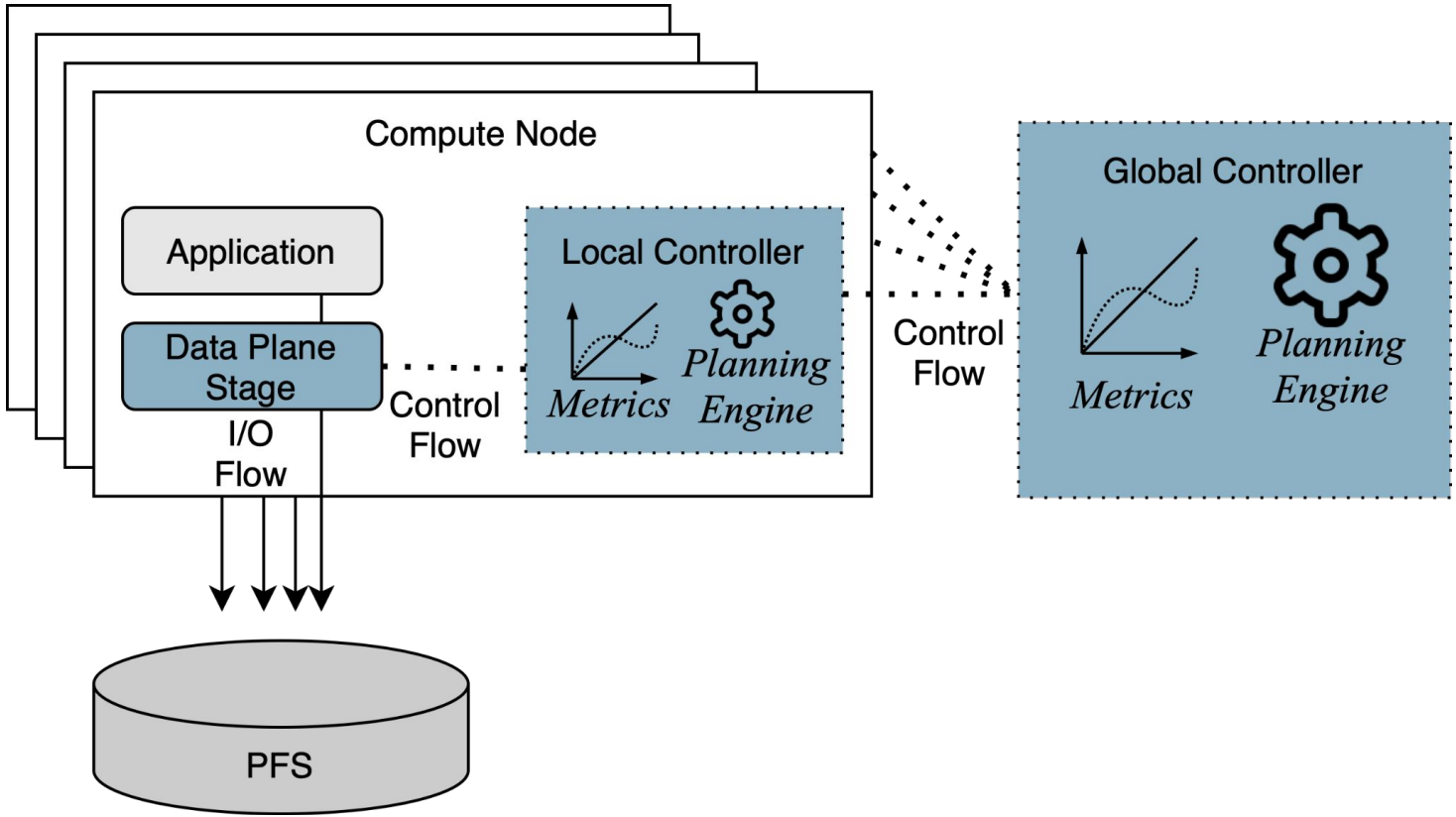
# BigHPC platform: Monitoring Backend



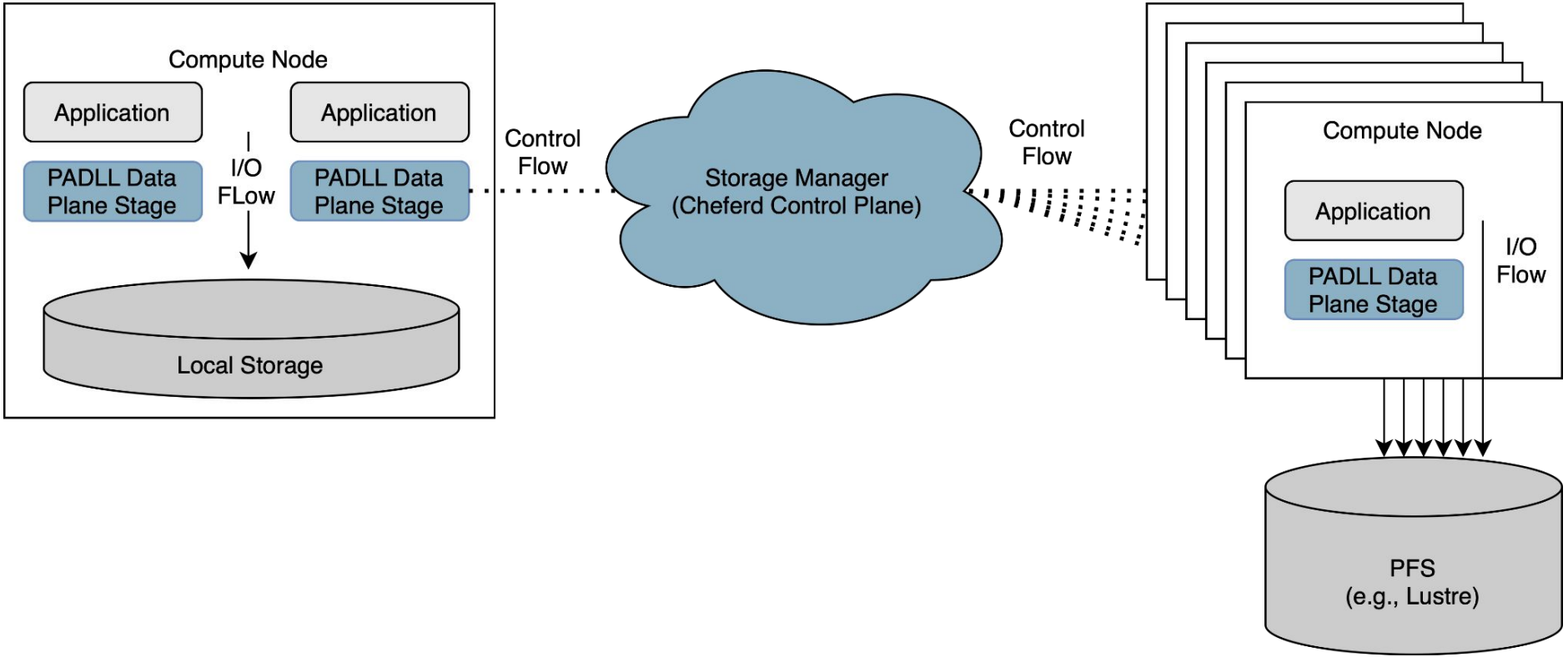
# BigHPC platform: Virtualization Manager



# BigHPC platform: Storage Manager Control Plane



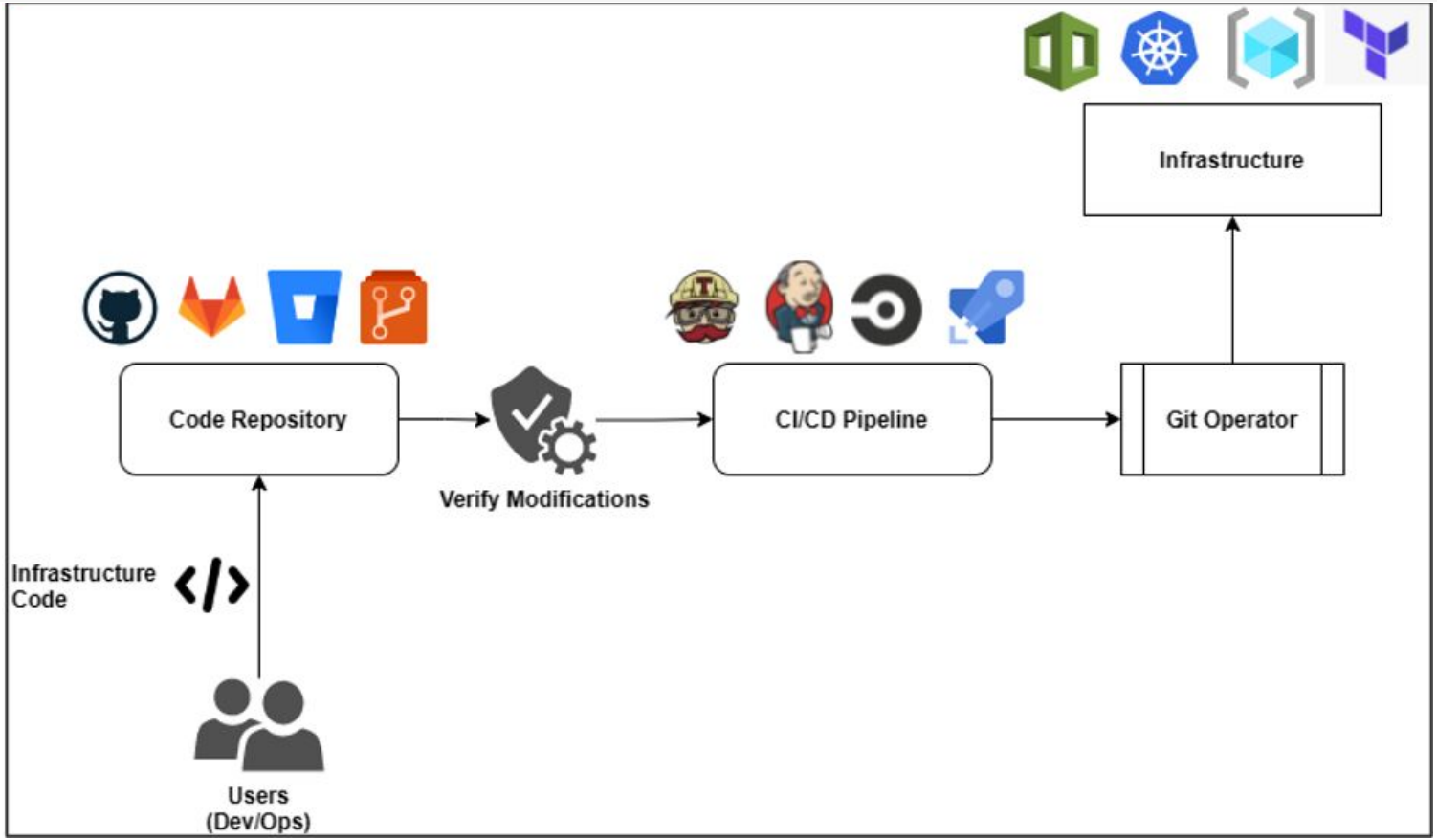
# BigHPC platform: Storage Manager Data Plane



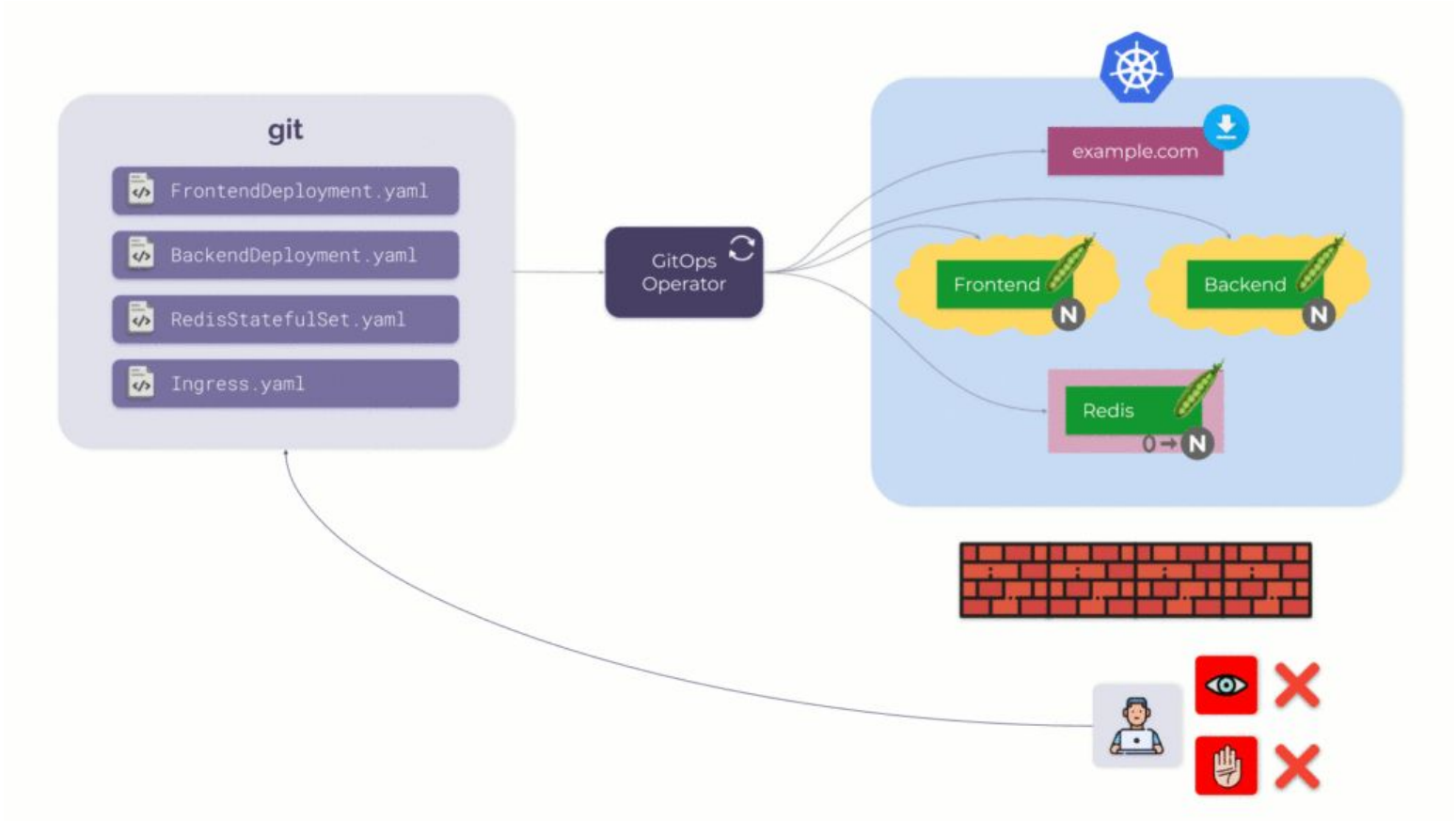
# Overview of BigHPC development challenges

- Software quality
- Components integration
- Deploy more often
- Easy error recovery
- Keep teams together in same direction
- Share knowledge between teams
- Documented deliveries with complete changes history

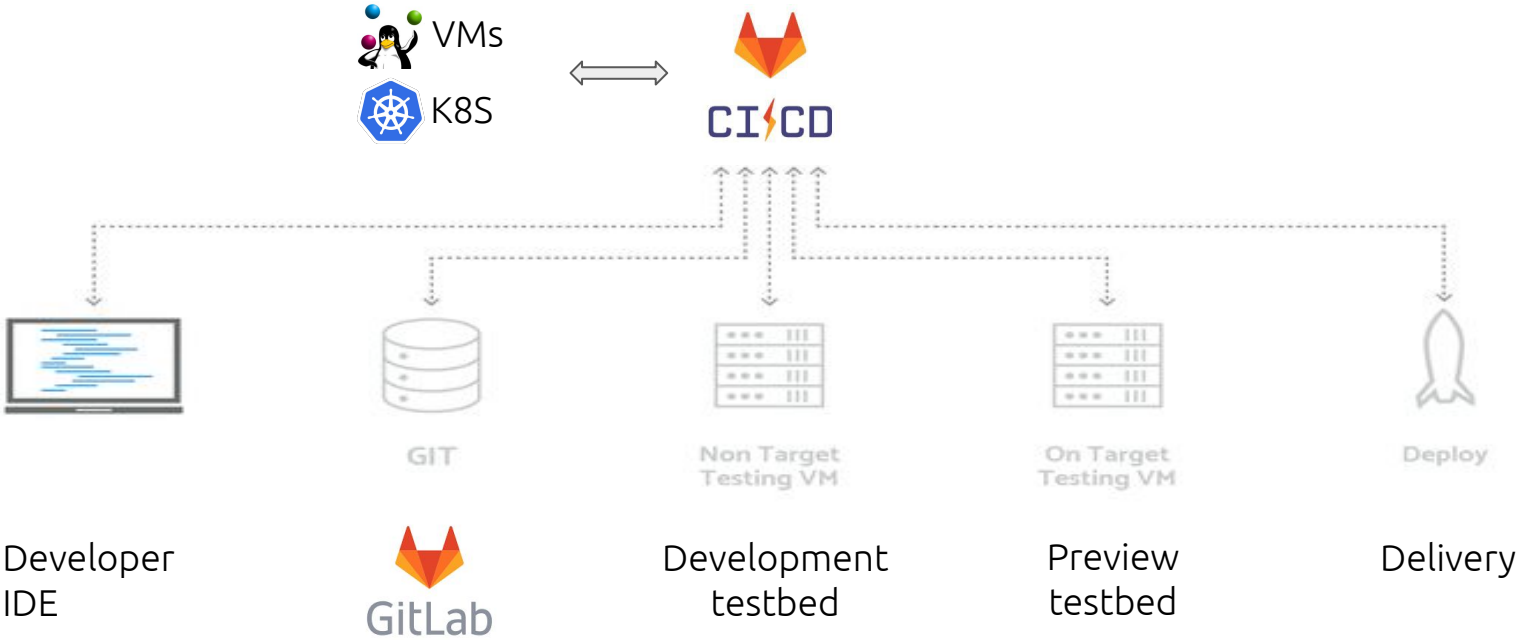
# GitOPS workflow



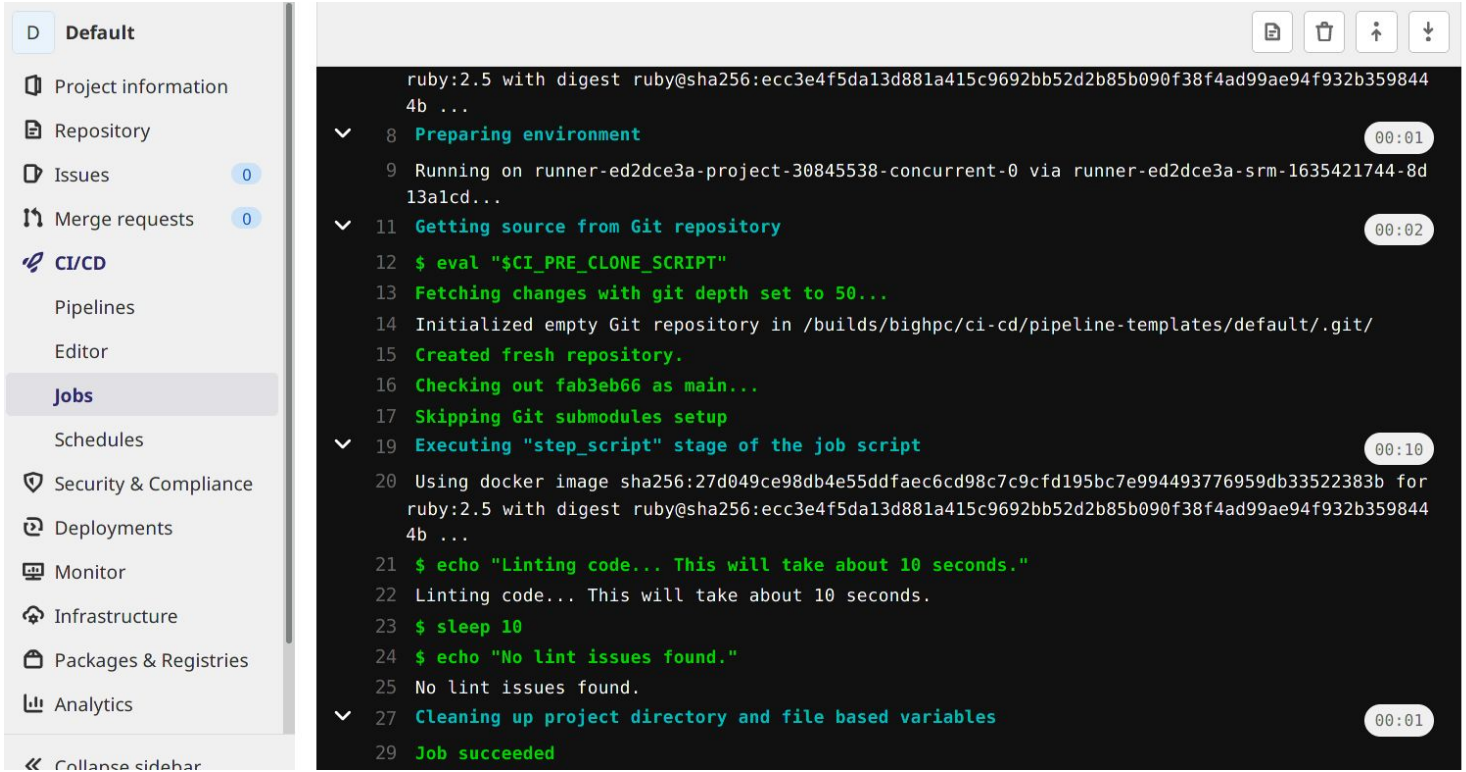
# Why GitOPS


git
FrontendDeployment.yaml
BackendDeployment.yaml
RedisStatefulSet.yaml
Ingress.yaml
GitOps Operator
example.com
Frontend
Backend
Redis

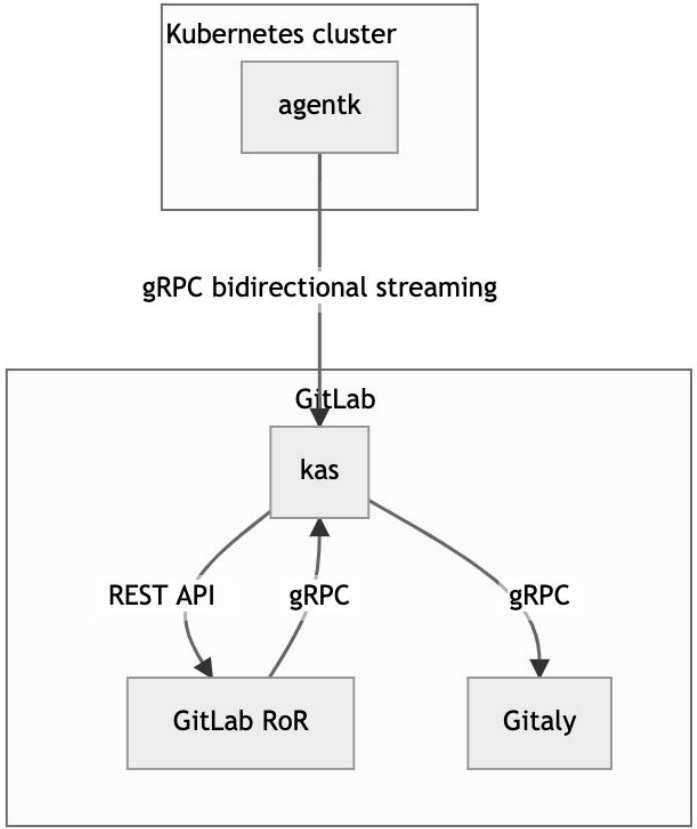
# Current status: implemented workflow



# Review Job log directly from web interface

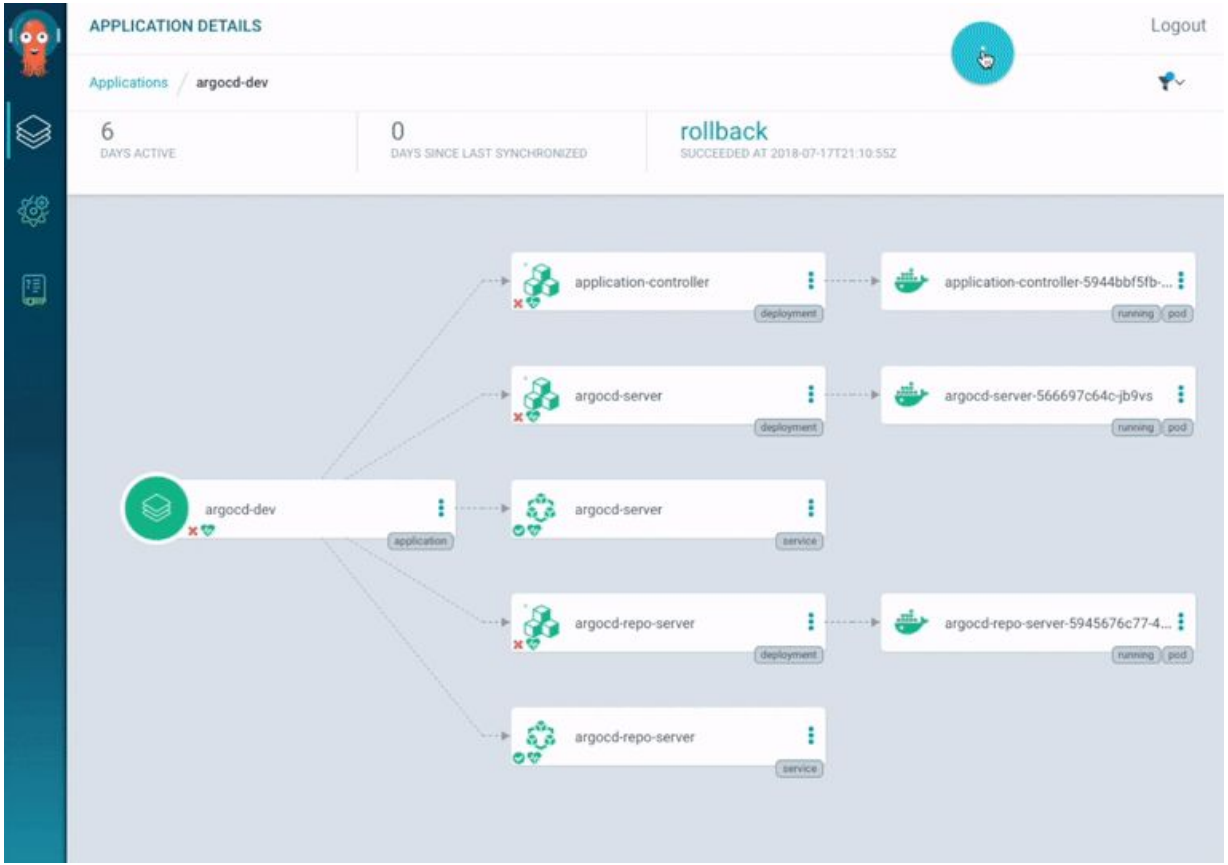


# Attach the Environment with the Infrastructure




- GitOPS push based strategy is implemented using Gitlab Agent
- agentk communicates to the Gitlab Agent Server (KAS) to perform GitOPS operations
- Changes applied to Kubernetes cluster anytime developer changes a manifest file managed within Gitlab
- This approach keeps Gitlab platform isolated from real infrastructure

# GitOPS aware of infrastructure state




- Argo CD provides a web user interface that allows to check application real time activity
- Health status analysis of application resources

# Next steps

- GitOPS insight
  - Improve GitOPS implementation, extending Gitlab CI/CD agent with kubernetes
  - Improve team collaboration using Gitlab platform
  - Test deployment error recovery taking the advantage of git
- Pilot
  - Test platform components with real use cases
  - Identify application performance issues

# Questions and Answers time

**Thank you for your attention!**

More details at: https://bighpc.wavecom.pt/

Contact: info.bighpc@wavecom.pt

# BIG HPC HIGH PERFORMANCE COMPUTING

## Partners:

INESCTEC

LIP

TEXAS
The University of Texas at Austin

MACC
Minho Advanced Computing Center

TACC

wavecom

## Funding:

Cofinanciado por:
COMPETE 2020

Lisb@20[20]

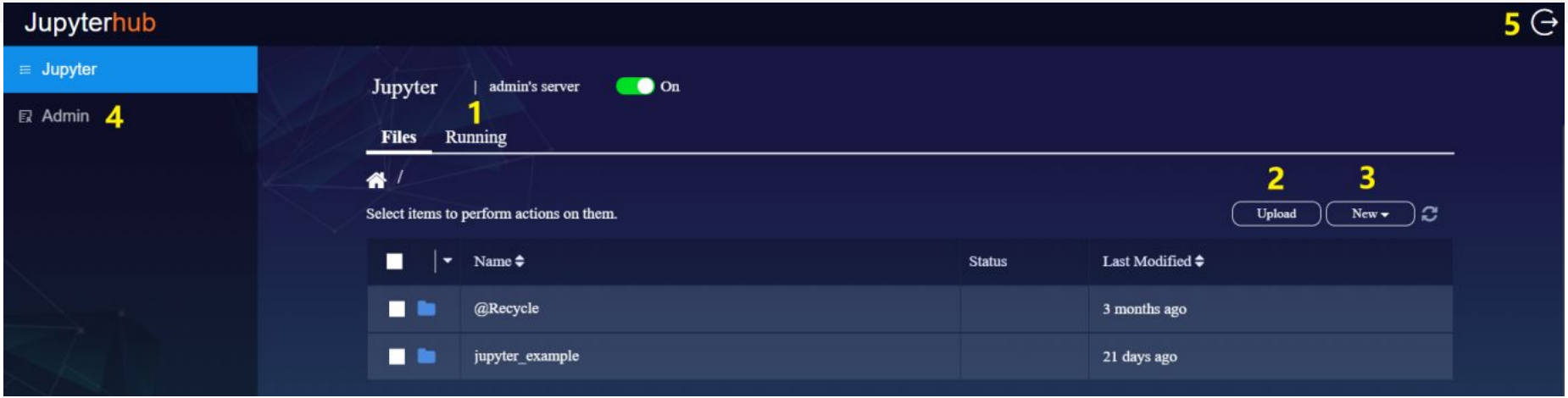
PORTUGAL 2020

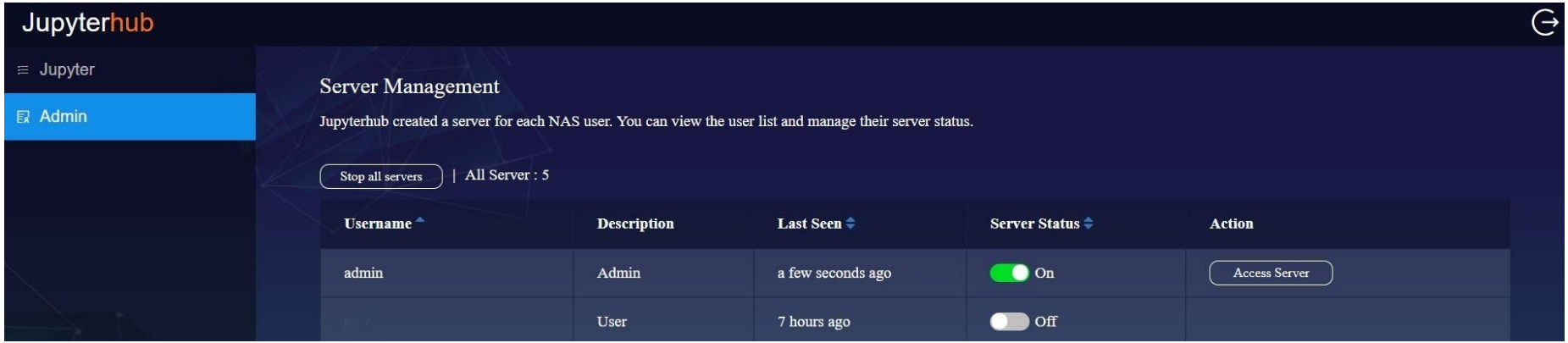
UNIÃO EUROPEIA
Fundo Europeu de Desenvolvimento Regional

FCT Fundação para a Ciência e a Tecnologia

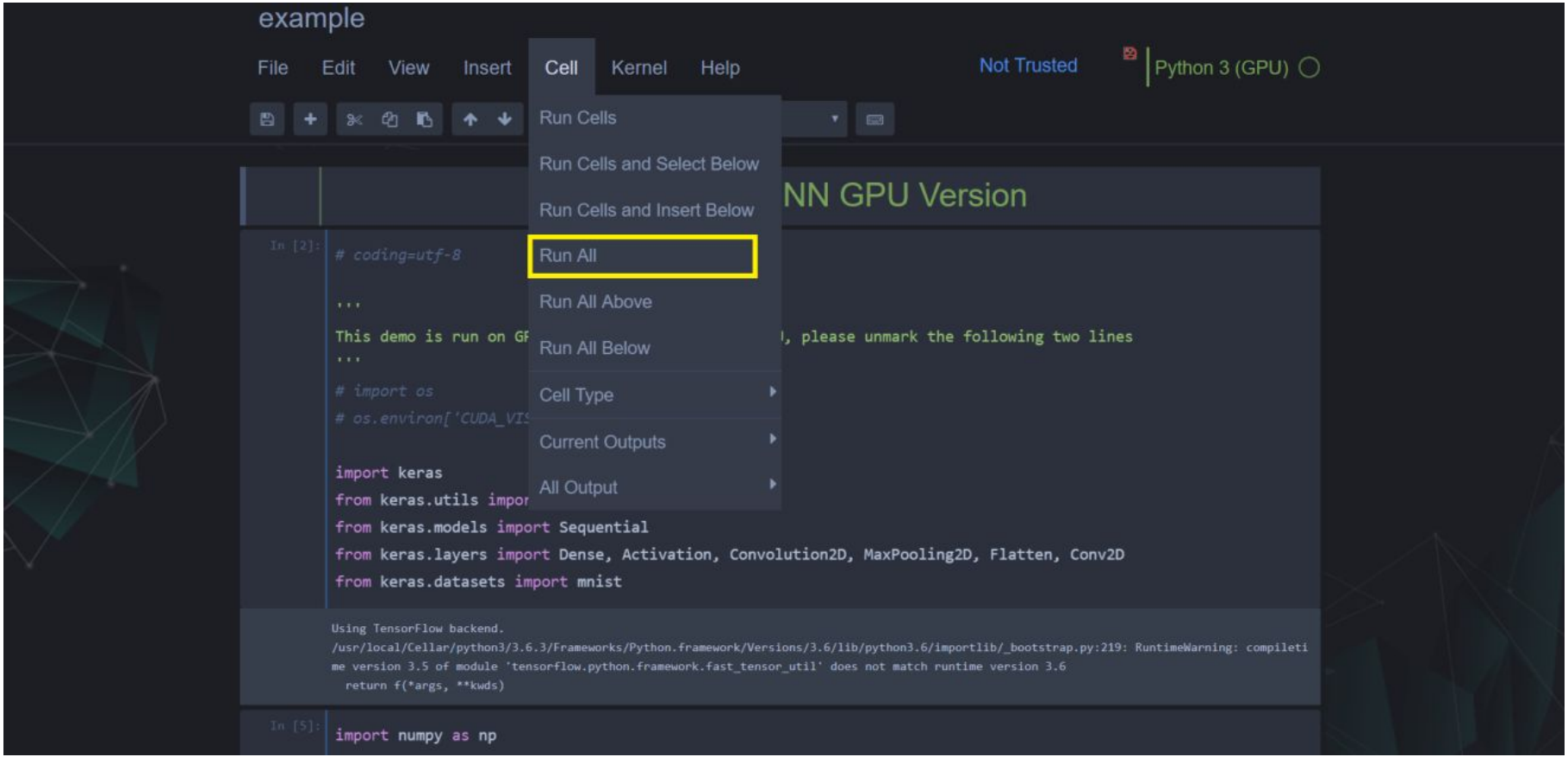
UTAustin Portugal

# User and administrator interfaces

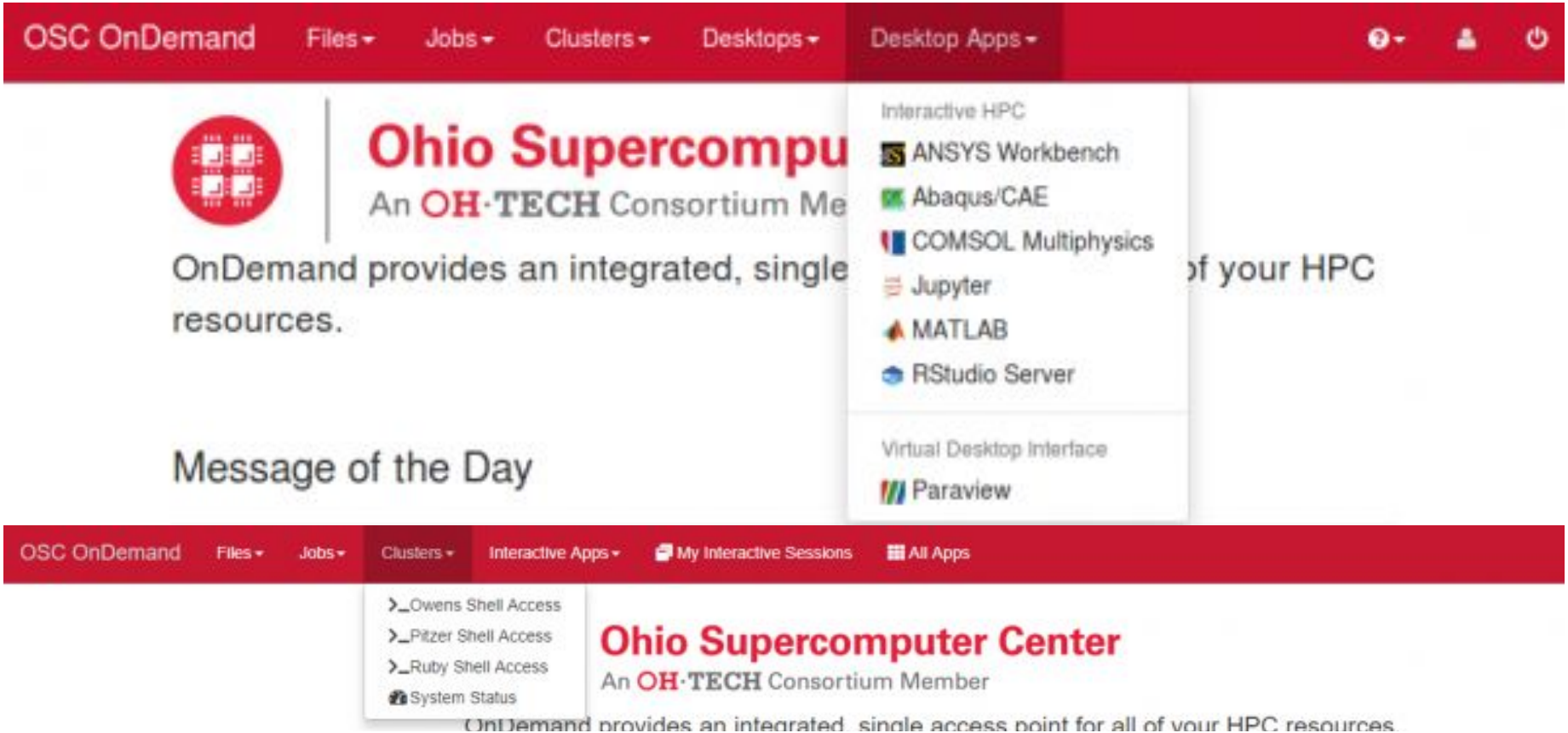


# Manage code and do debugging



# OnDemand: extend access to applications



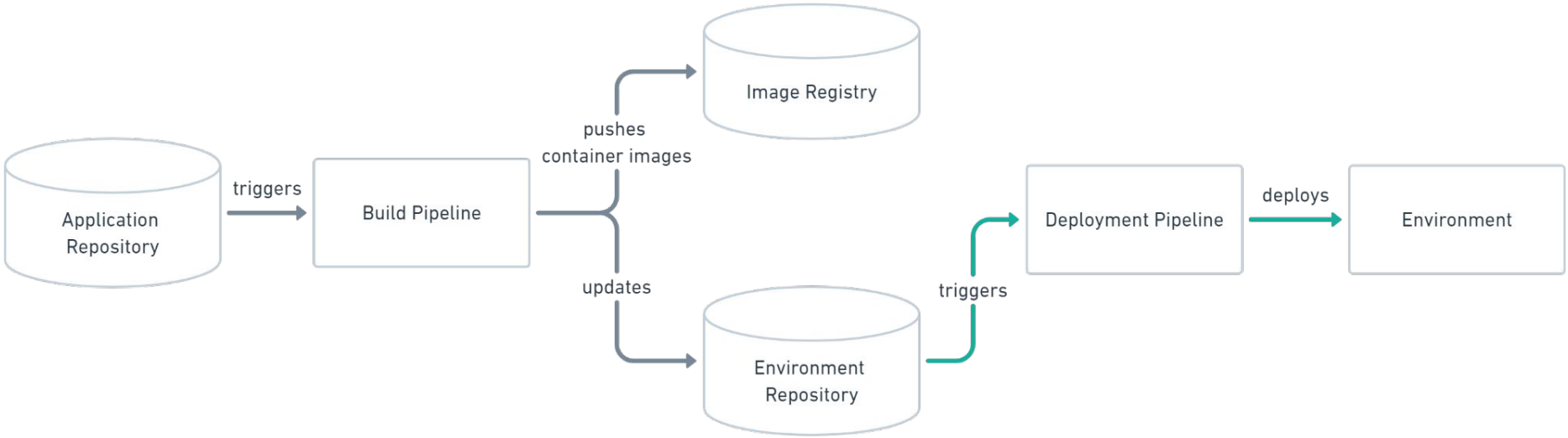
# What is GitOPS

- Its inception in 2017 by Weaveworks
- Uses developer common tools:
    - git version control system to track code changes
    - continuous deployment tools
- Have a git repository that contains infrastructure declarative description
- Production environment match the described state in the repository
- Deployment and updates means push changes to a git repository

# Why GitOPS

- Always use same procedure to deploy applications without the burden of switching between required tools
- Lighter learning curve since everything happens on git with less effort
- Error recovery as easy as git revert
- Easier credential management since only required access to git repository
- No need to give developers direct access to the deployed endpoints
- Complete description of what is deployed
  - every changes goes through the git repository
  - complete history of every change made to the system
- Share knowledge with great commit messages where everybody can reproduce and find examples how to set up new systems

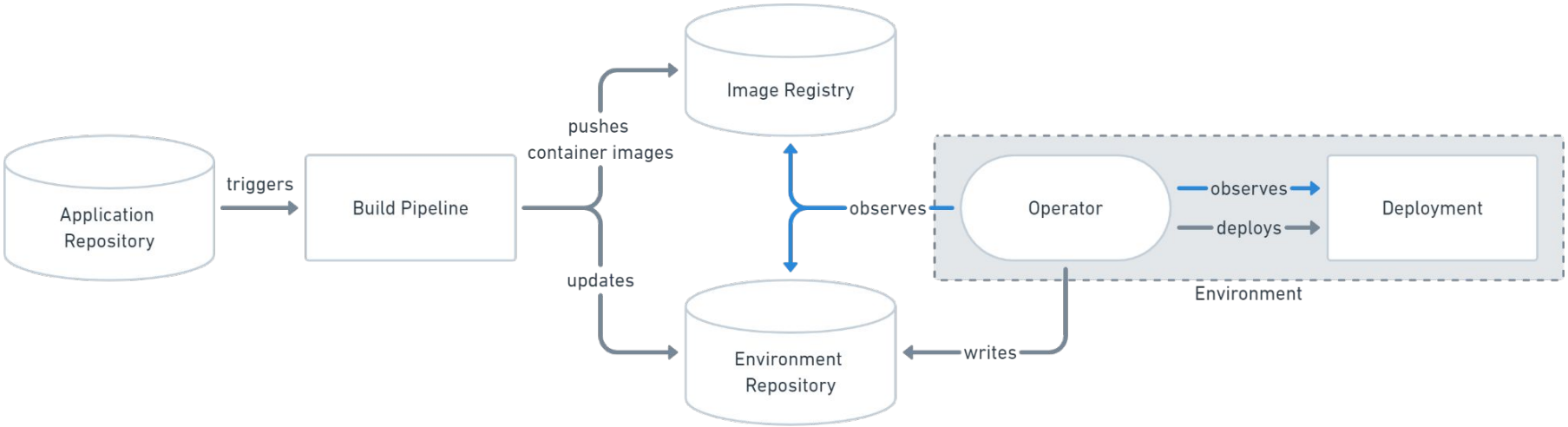
# Deployment strategy: push based




- Implemented by Gitlab CI/CD, Jenkins, Circle CI or Travis CI
- Code updates trigger pipeline execution (external event)

- Environment configuration repository is updated with new deployment descriptors
- misses automatic check of environment desired state

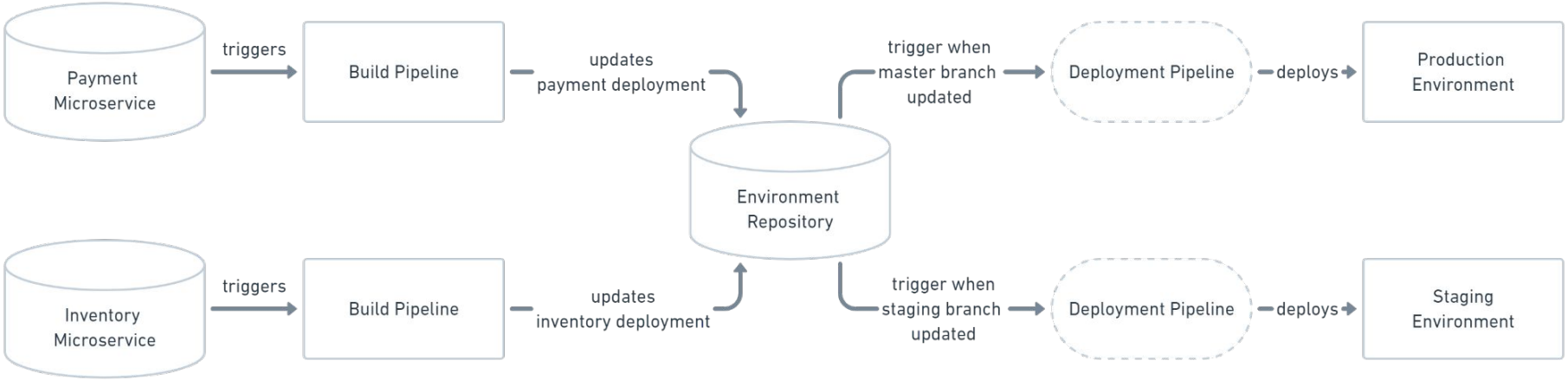
# Deployment strategy: pull based




- Operator is introduced comparing with previous strategy
- Implemented by Argo CD, Flux CD and Gitlab starting from v15

- Operator checks consistency between deployed infrastructure, environment repository and image registry

# Multiple Applications and Environments




- GitOPS also support multiple build pipelines from different repositories
- Changes in each project generate an update to environment repository
- Multiple environments with GitOPS means separate branches in environment repository
- Operator will react on changes and run the associated environment pipeline

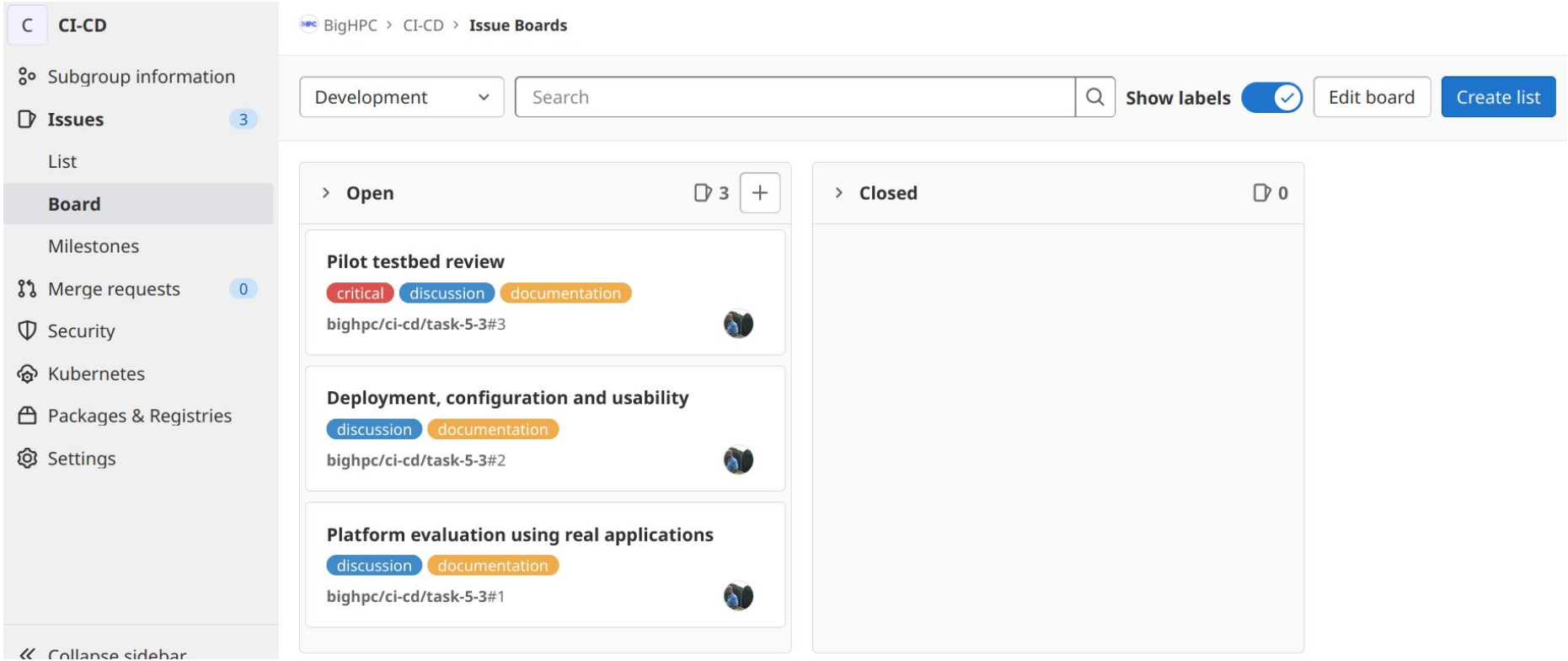
# BigHPC Gitlab platform adoption

- Adopting GitOPS using Gitlab platform
  - Implement continuous deployment applying software quality best practices
  - Well documented deliveries with complete history of system changes
  - Promote shared knowledge and get teams together
  - Easy to reproduce the thought process of changing infrastructure
  - Get the examples to setup new systems

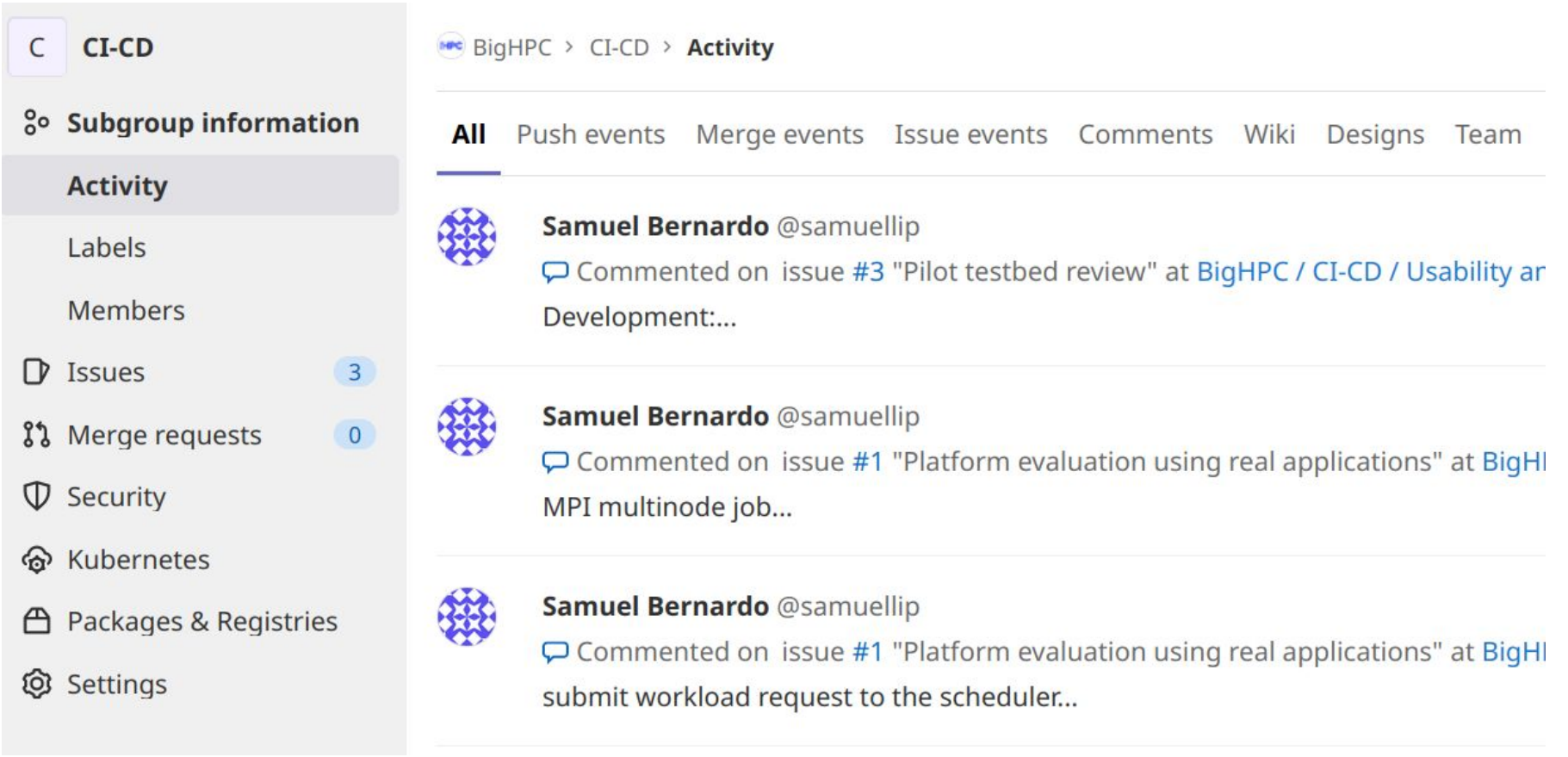
# BigHPC environments

- Testbeds and Pilot
  - Review the hardware profiles and application requirements
  - Define execution environments for the required testbeds
  - Gather applications that covers the demanded use case requirements
  - Prepare the pilot to test platform components with collected applications over HPC
  - Collect the metrics and get the report from jobs submission

# Current status: team collaboration



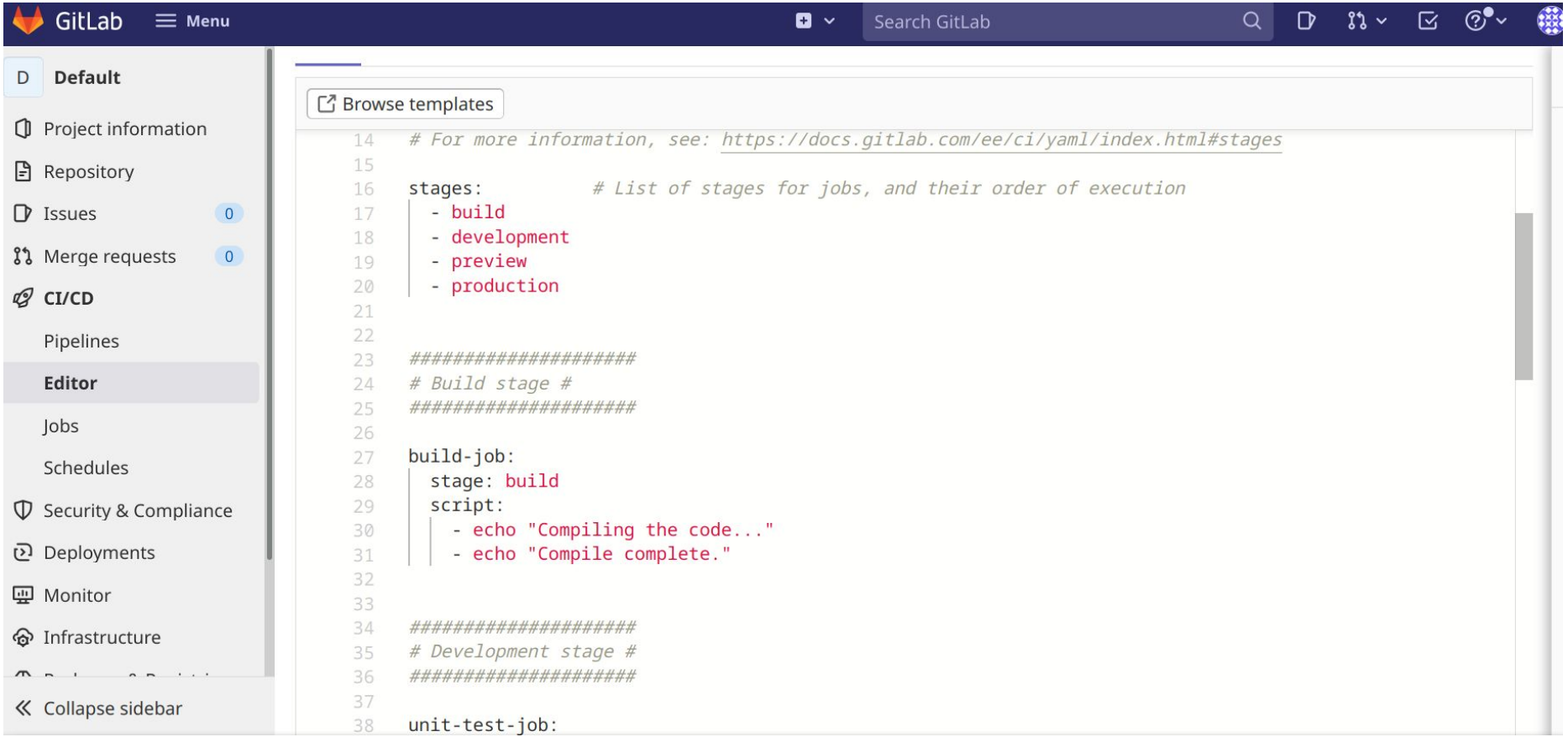
# Current status: team collaboration



# Gitlab stages for the project

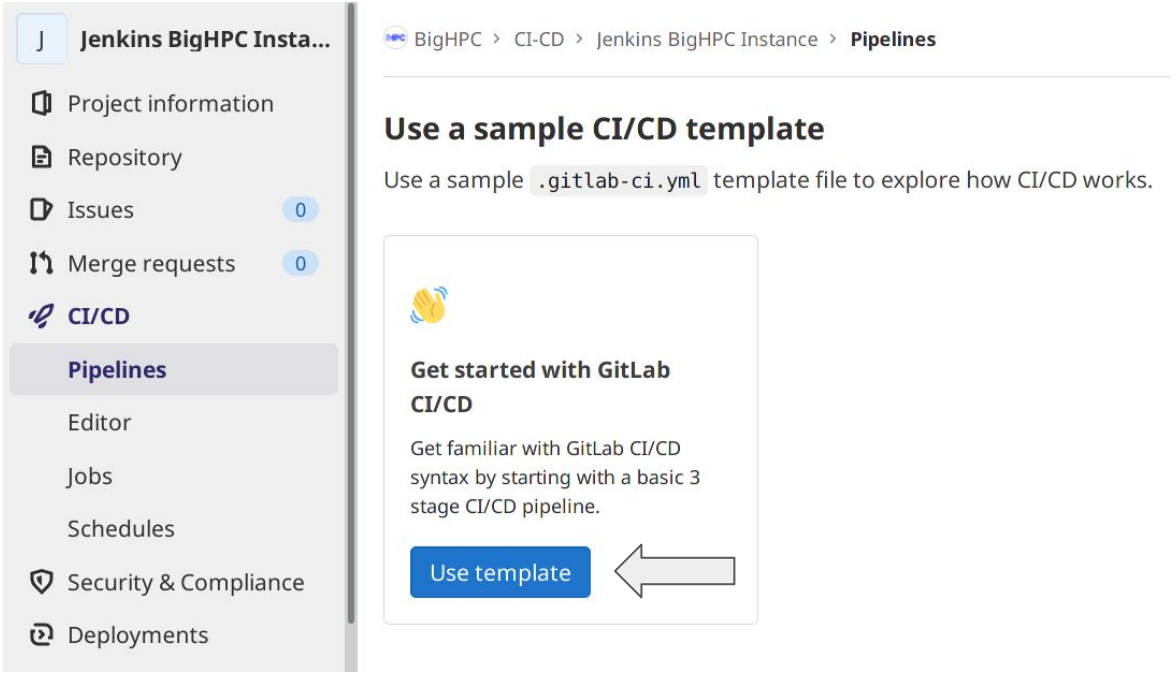
- Development testbed
- Preview testbed
- Delivery to production (create a release)

# Create the pipeline configuration



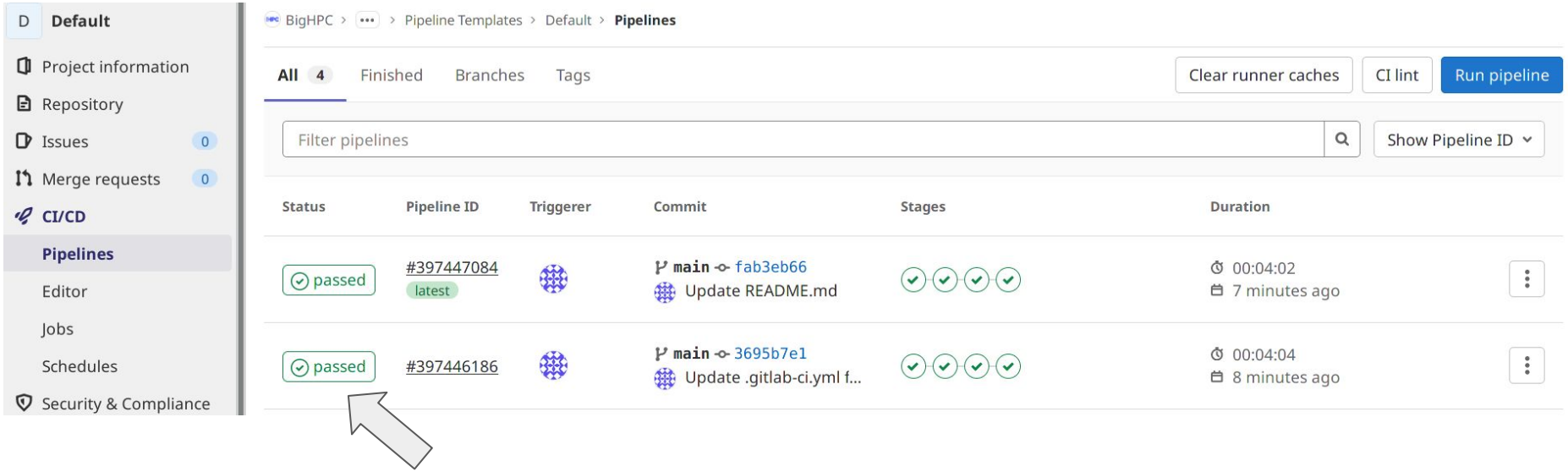
# Developers integration task

- Create a new project in Gitlab
- Add pipeline from template

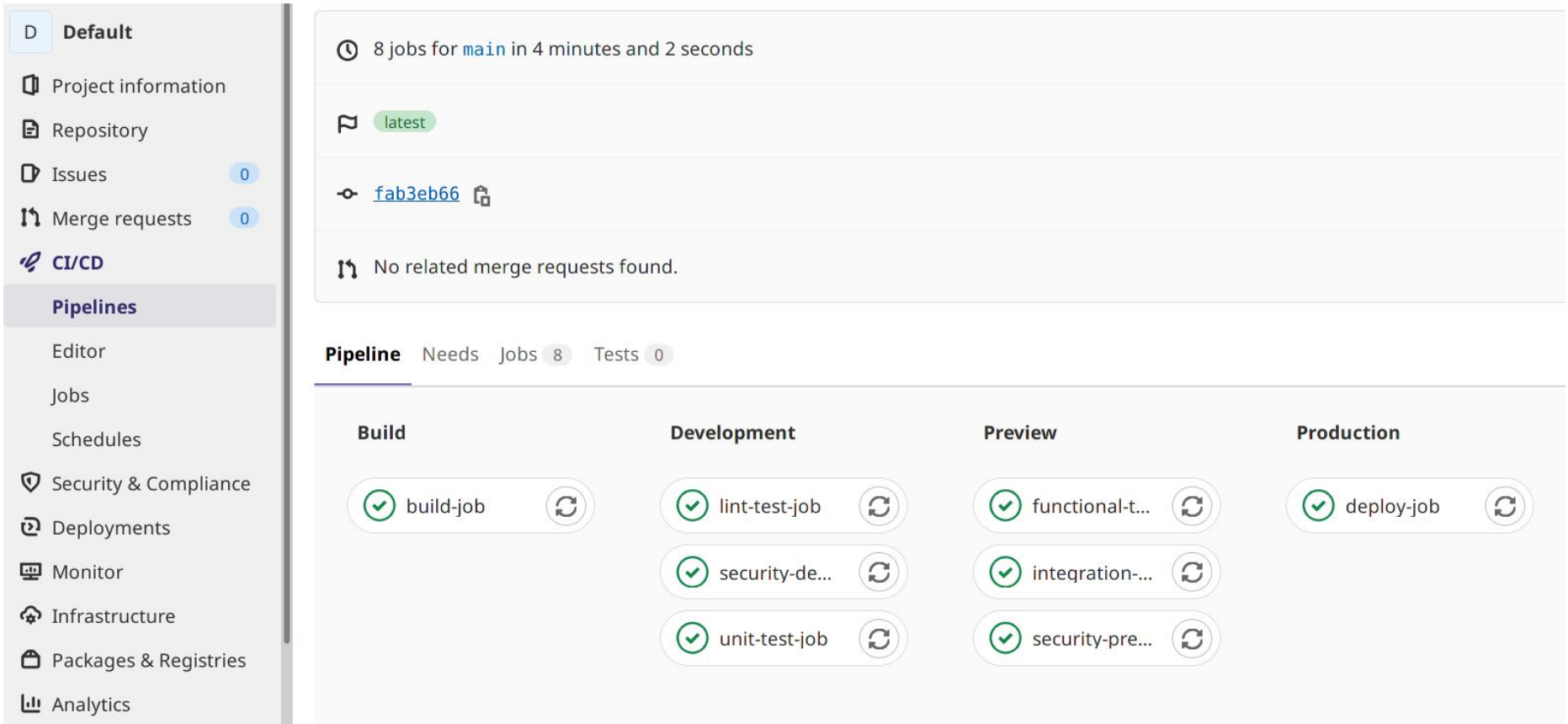


# Check pipeline results

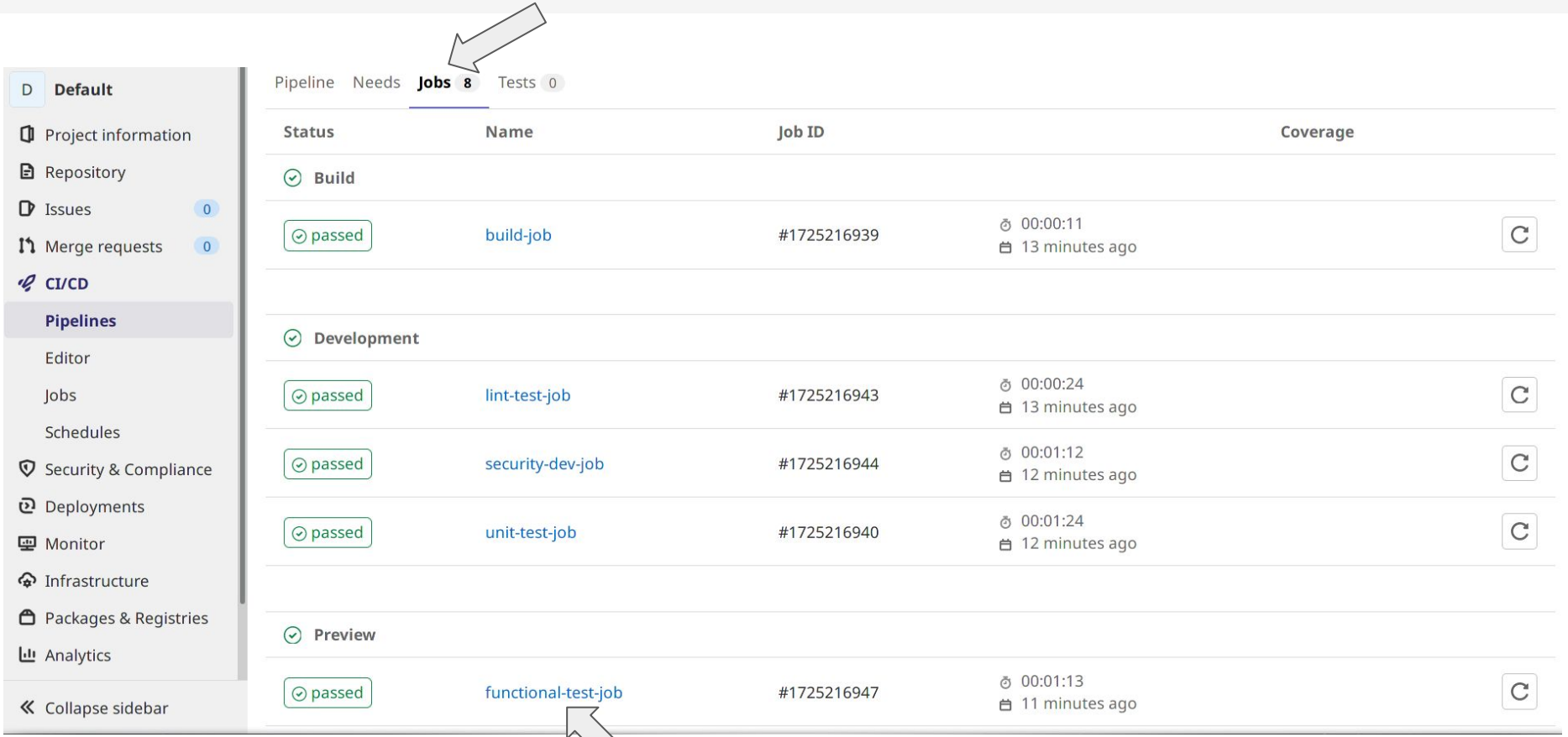
Click over status of pipeline after passing the tests



# Look into pipeline stages

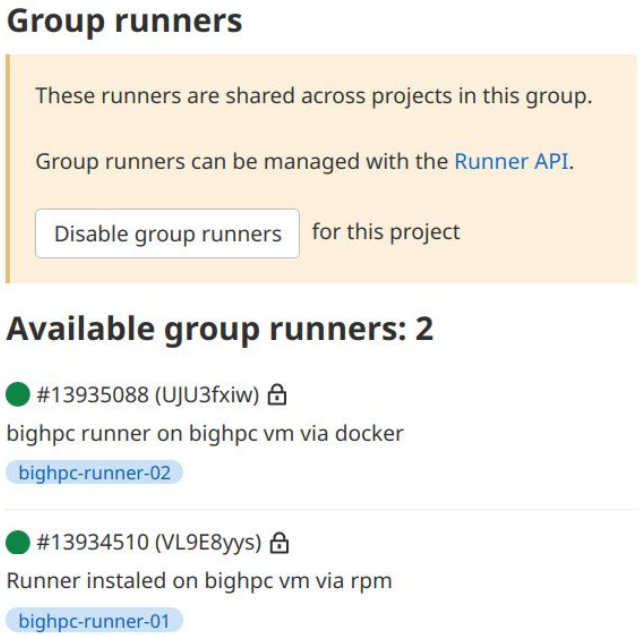


# Check jobs state

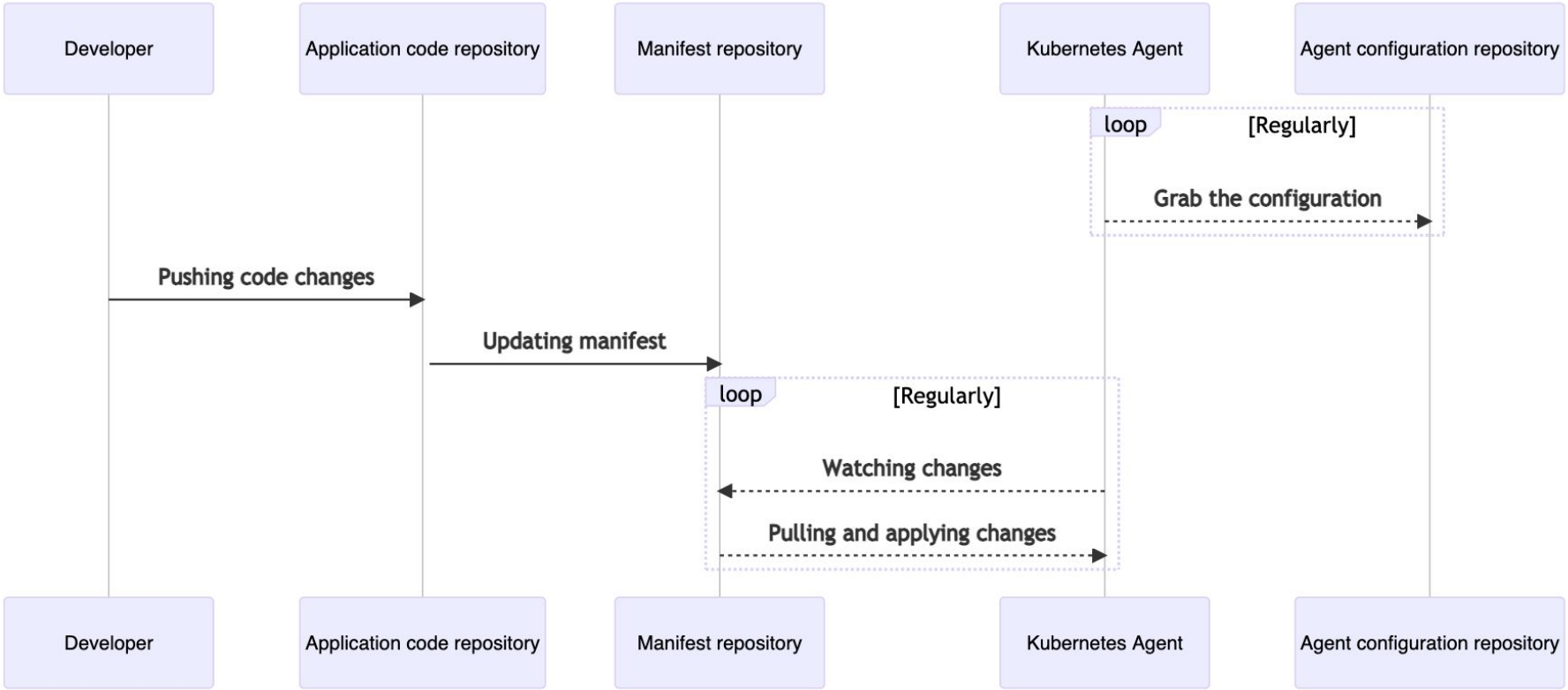


# Attach the Environment with the Infrastructure

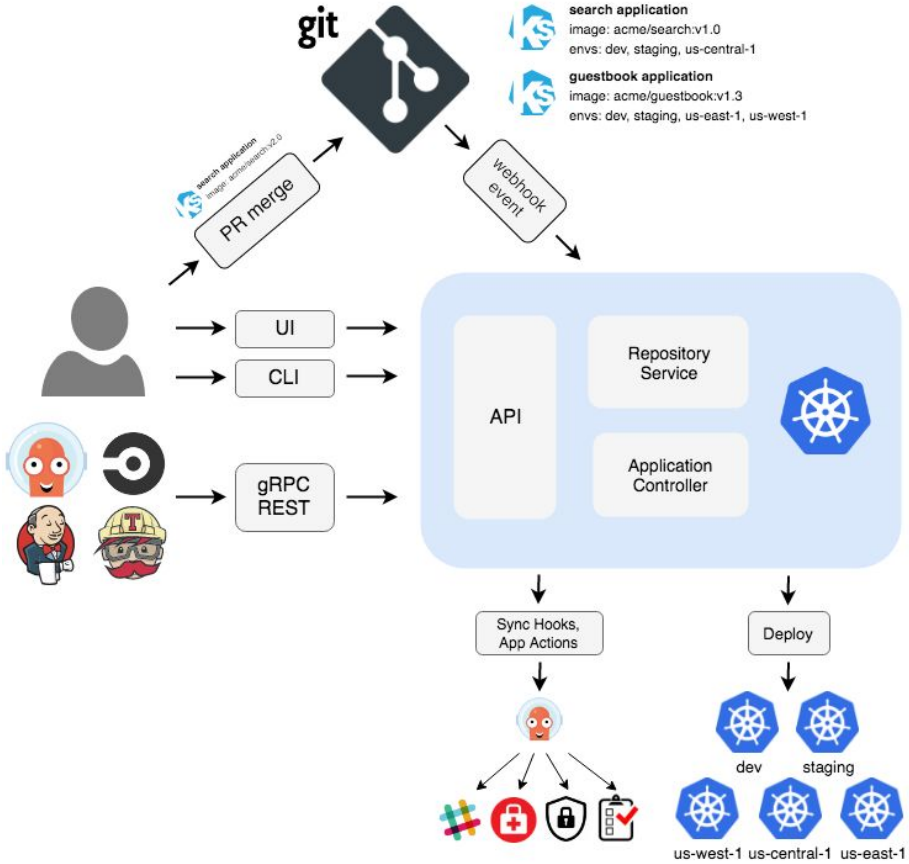

- Gitlab Runners are agents that run CI/CD jobs from Gitlab
- Gitlab Runner implements executors used to run builds in different environments
- Environments comprehends different Operating Systems and available tools, such as docker and kubernetes
- Each Gitlab Runner can have different access policies and can be limited to pipelines asking for specific tag



# Attach the Environment with the Infrastructure



# GitOPS aware of infrastructure state




- Argo CD implements the GitOPS pull based strategy
- This not only follows the repositories activity, but also the deployment state in the infrastructure
- Webhook integration with Gitlab
- Support of multiple config management and templating tools (Kustomize, Helm, …)

# Background execution behind the scene

- Develop Gitlab CI/CD configuration for the required pipelines
- Create a Gitlab project and submit the code for the defined testbeds
- Connect Gitlab project with Gitlab runner instance, providing the reports in Gitlab
- Create docker images to pack the required code
- Do the required experimental validation checking the automated results over the testbeds (development and preview)
- Deploy the release to the production infrastructure
- Present the Pilot for further validation