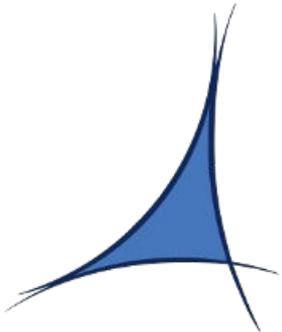
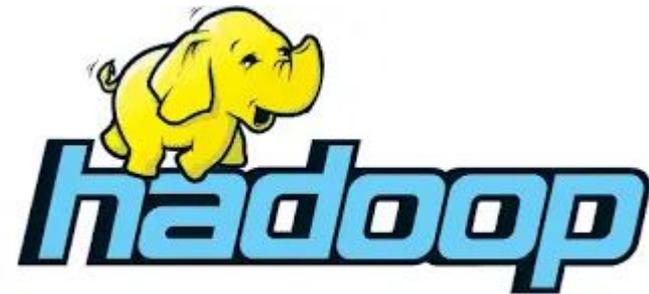


# Development and automated deployment of a custom Hadoop distribution at PIC using CI/CD



**PIC**  
port d'informació  
científica

Agustin Bruzzese - PIC  
bruzzese@pic.es



red.es



**Ciemat** Centro de Investigaciones  
Energéticas, Medioambientales  
y Tecnológicas



**IFAE**

**R** Institut de Física  
d'Altes Energies



# Contents

---

## 1. Introduction

- Research topic and problem
- Challenges and Problems
- Objectives

## 2. Contribution and results

- PIC's Hadoop distribution
- Distributed deployment
- Hadoop Client
- Summary of Main contributions

## 3. Conclusions & Project future steps

# Contents

---

## 1. Introduction

- Research topic and problem
- Challenges and Problems
- Objectives

## 2. Contribution and results

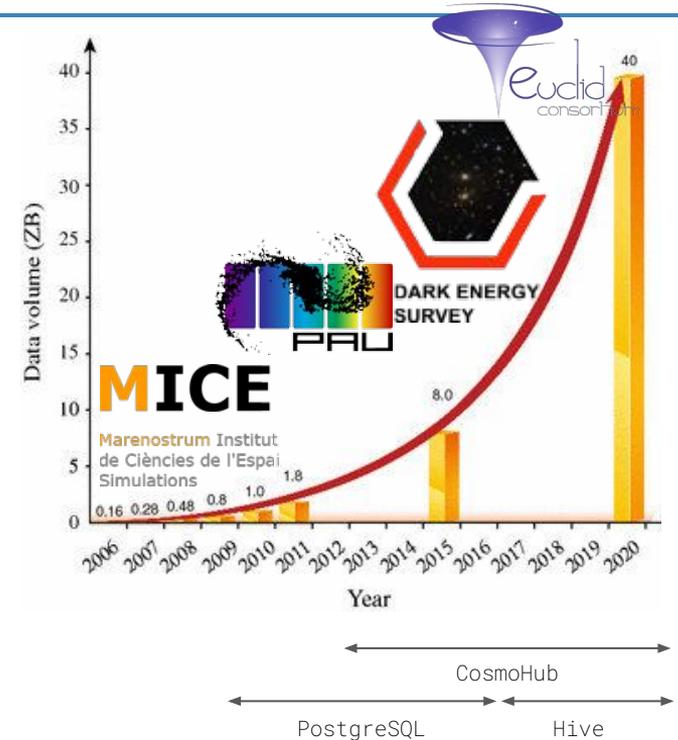
- PIC's Hadoop distribution
- Distributed deployment
- Hadoop Client
- Summary of Main contributions

## 3. Conclusions & Project future steps

# 1. Introduction

## Project topic

- **Data volume is growing at an exponential rate**
  - also becomes more diverse and complex
- **Exploiting all this data requires**
  - new analysis algorithms/frameworks
  - new computing platforms/paradigms
- **Apache Hadoop is an**
  - open source platform
  - for distributed storage and processing
  - based on commodity hardware
- **Hadoop provides three main services/abstractions**
  - Distributed and scalable storage: HDFS
  - Processing framework: MapReduce
  - Resource scheduling: YARN



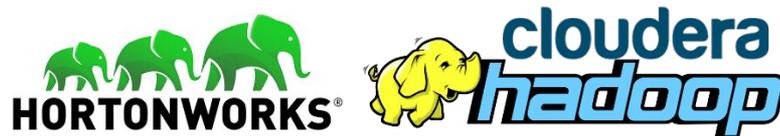
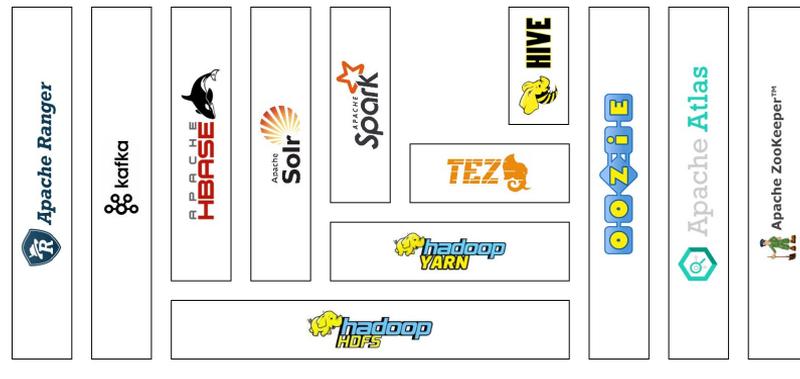
Global growth trend of data volume, 2006–2020.  
Image modified from [researchgate](https://www.researchgate.net/publication/312211111)



# 1. Introduction

## Project topic

Component	Description
Atlas	Apache <b>Atlas</b> provides <b>metadata management</b> and governance capabilities for organizations to build, categorize, and govern their data assets on Hadoop clusters, by representing metadata as <b>types</b> and <b>entities</b> .
HBASE	Apache <b>HBase</b> is a column-oriented <b>non-relational database</b>
Hive	<b>HIVE</b> is a <b>data warehousing component</b> which performs reading, writing and managing large data sets in a distributed environment using <b>SQL-like interface</b> .
Kafka	Apache <b>Kafka</b> is a <b>message brokering</b> system based on the publisher/subscriber model
Oozie	<b>Oozie</b> is a <b>workflow scheduler</b> system to manage Apache Hadoop jobs.
Ranger	Apache <b>Ranger</b> is a framework to enable, monitor and manage comprehensive data <b>security</b> across the Hadoop platform.
Solr	Apache <b>Solr</b> (stands for Searching On Lucene w/ Replication) is a free, open-source <b>search engine</b> based on the Apache Lucene library.
Spark	<b>Spark</b> is a framework for real time data <b>analytics</b> in a distributed computing environment.
Tez	<b>Tez</b> is usually running under MapReduce, so it's just a <b>MapReduce optimized with less and compacted steps</b> .
Zookeeper	<b>Zookeeper</b> coordinates with various services in a distributed environment.
Kerberos	<b>Kerberos</b> is a computer network authentication protocol, which provides a secure Single Sign On(SSO) based on a trusted third-party mutual authentication service.



# 1. Introduction

## Challenges and Problems

Component	Description
Atlas	Apache <b>Atlas</b> provides <b>metadata management</b> and governance capabilities for organizations to build, categorize, and govern their data assets on Hadoop clusters, by representing metadata as <b>types</b> and <b>entities</b> .
HBASE	Apache <b>HBase</b> is a column-oriented <b>non-relational database</b>
Hive	<b>HIVE</b> is a <b>data warehousing component</b> which performs reading, writing and managing large data sets in a distributed environment using <b>SQL-like interface</b> .
Kafka	Apache <b>Kafka</b> is a <b>message brokering</b> system based on the publisher/subscriber model
Oozie	<b>Oozie</b> is a <b>workflow scheduler</b> system to manage Apache Hadoop jobs.
Ranger	Apache <b>Ranger</b> is a framework to enable, monitor and manage comprehensive data <b>security</b> across the Hadoop platform.
Solr	Apache <b>Solr</b> (stands for Searching On Lucene w/ Replication) is a free, open-source <b>search engine</b> based on the Apache Lucene library.
Spark	<b>Spark</b> is a framework for real time data <b>analytics</b> in a distributed computing environment.
Tez	<b>Tez</b> is usually running under MapReduce, so it's just a <b>MapReduce optimized with less and compacted steps</b> .
Zookeeper	<b>Zookeeper</b> coordinates with various services in a distributed environment.
Kerberos	<b>Kerberos</b> is a computer network authentication protocol, which provides a secure Single Sign On(SSO) based on a trusted third-party mutual authentication service.

### Cloudera Vendor Lock-in.

We need to Explore the possibility to develop our own distribution

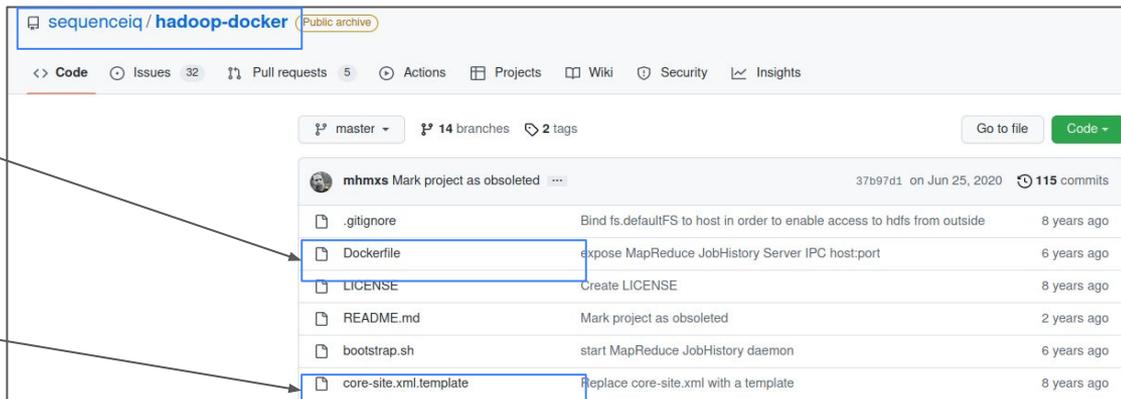


# 1. Introduction

## Challenges and Problems

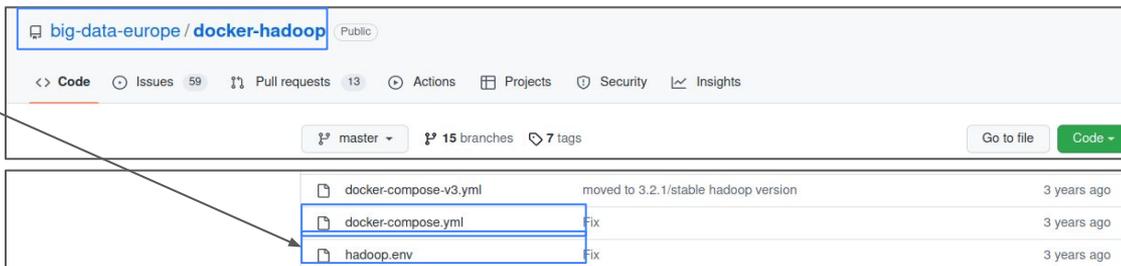
### Other Hadoop Distributions:

- Following the state of the art of hadoop distributions, our distribution has been made **based on docker**.
- We also have some generic **configuration template files** per component that are part of the hadoop distribution ecosystem.
- In addition to a **file of configuration variables** that just configured the generic configuration templates of the hadoop components



### Downside:

- Use **outdated versions** of the required components. [Link](#)
- All of them **do not use half as many components** as we need to meet the requirements of a cluster on the PIC



```
# download native support
RUN mkdir -p /tmp/native
RUN curl -L https://github.com/sequenceiq/docker-hadoop-build/releases/download/v2.7.1/hadoop-native-64-2.7.1.tgz | tar -xz -C /tmp/native
```

# 1. Introduction

## Objectives

---

- **Motivations:** Given the above problems, **develop** a big data analytics platform based on **Hadoop on distributed** scale-out storage system to achieve massive performance, scalability and fault tolerance.
    - gain knowledge
    - to be able to do upgrades at our own will
    - fixed patches/modifications/bug fixes
    - flexibility in the combination of components
- 
- **Objectives:**
    1. Develop a **Hadoop distribution** with all necessary components for big data processing
    2. **Different environments** to ensure proper testing
    3. A **workflow** to deploy PIC's hadoop distribution

# Contents

---

## 1. Introduction

- Research topic and problem
- Challenges and Problems
- Objectives

## 2. Contribution and results

- PIC's Hadoop distribution
- Distributed deployment
- Hadoop Client
- Summary of Main contributions

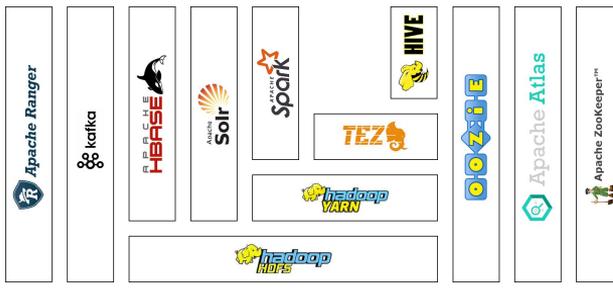
## 3. Conclusions & Project future steps



## 2. Contributions and results: PIC's Hadoop distribution

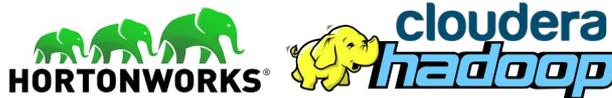
### 1. Architecture and Features:

- We develop a full hadoop distribution that mimics the Cloudera ecosystem at the production cluster at PIC in a **single docker image**.



### 2. Management:

- Cluster managing **bins** offers an even more abstract way for the administrator to **set up, start, stop** and **status** to get details of each component of the hadoop ecosystem
- It can be run with **systemctl**



Vs



Component	Versions (HDP 3.1.4)	Versions (PIC 1.0.0)
Atlas	1.1.0	2.2.0
Hadoop	3.1.1	3.2.3
HBASE	2.0.2	2.2.6
Hive	3.1.0	3.1.2
Kafka	2.0.0	2.5.0
Oozie	4.3.1	5.2.1
Ranger	1.2.0	2.1.0
Solr	7.7.0	6.5.1
Spark	2.3.2	3.1.2
Tez	0.9.1	0.10.0
Zookeeper	3.4.6	3.7.1
Kerberos	MIT KDC	FreeIPA

```
agus@agus-laptop:~/Git/hadoop-cluster$ hdfs.datanode
'/home/agus/hadoop-conf-files/hdp/data' found and now copying files, please wait ...
no options provided...
```

Hadoop 1.0

Usage: hdp [Options]

## 2. Contributions and results: PIC's Hadoop distribution

### 1. Considerations

- Incompatibilities between version
- Lack of **documentation**. [Link](#)

### 2. Testing:

- Test the **functionality/changes** of configurations.
- Test the **versions upgrade**.
- Test configuration, functionality and **compatibility** between ecosystem components
- Tests can be **extended** at any point
- **Bats Documentation**: [Link](#)

	HBase-0.94.x	HBase-0.98.x (Support for Hadoop 1.1+ is deprecated.)	HBase-1.0.x (Hadoop 1.x is NOT supported)	HBase-1.1.x	HBase-1.2.x	HBase-1.3.x
Hadoop-1.0.x	X	X	X	X	X	X
Hadoop-1.1.x	S	NT	X	X	X	X
Hadoop-0.23.x	S	X	X	X	X	X
Hadoop-2.0.x-alpha	NT	X	X	X	X	X
Hadoop-2.1.0-beta	NT	X	X	X	X	X
Hadoop-2.2.0	NT	S	NT	NT	X	X
Hadoop-2.3.x	NT	S	NT	NT	X	X

Compatible version of hadoop and hbase. Image taken from [researchgate](#)



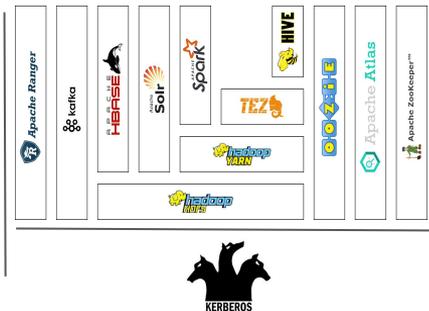
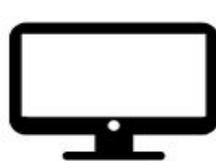
```
bats /opt/tools/tests/hdfs.bats
1..5
ok 1 HDFS 01: non admin create folder
ok 2 HDFS 02: non admin copy file to hdfs
ok 3 HDFS 03: hdfs admin user initialization
ok 4 HDFS 04: Check hdfs web services
ok 5 HDFS 05: Check YARN web services
Cleaning up project directory and file based variables
Job succeeded
```

Bats for testing configurations and versions compatibilities in Hadoop ecosystem. [Link](#)

## 2. Contributions and results: PIC's Hadoop distribution

- **CI/CD Pipeline:**

- Automated processing of **building**, and **pushing** image.
- Check changes at the **templates configuration files (\*.xml)**
- Performs **components tests** on builded image
- **Deploy** at **test environment** (pseudo distributed deploy).



Single node deployment of Hadoop ecosystem for testing

The screenshot shows a pipeline interface with the following details:

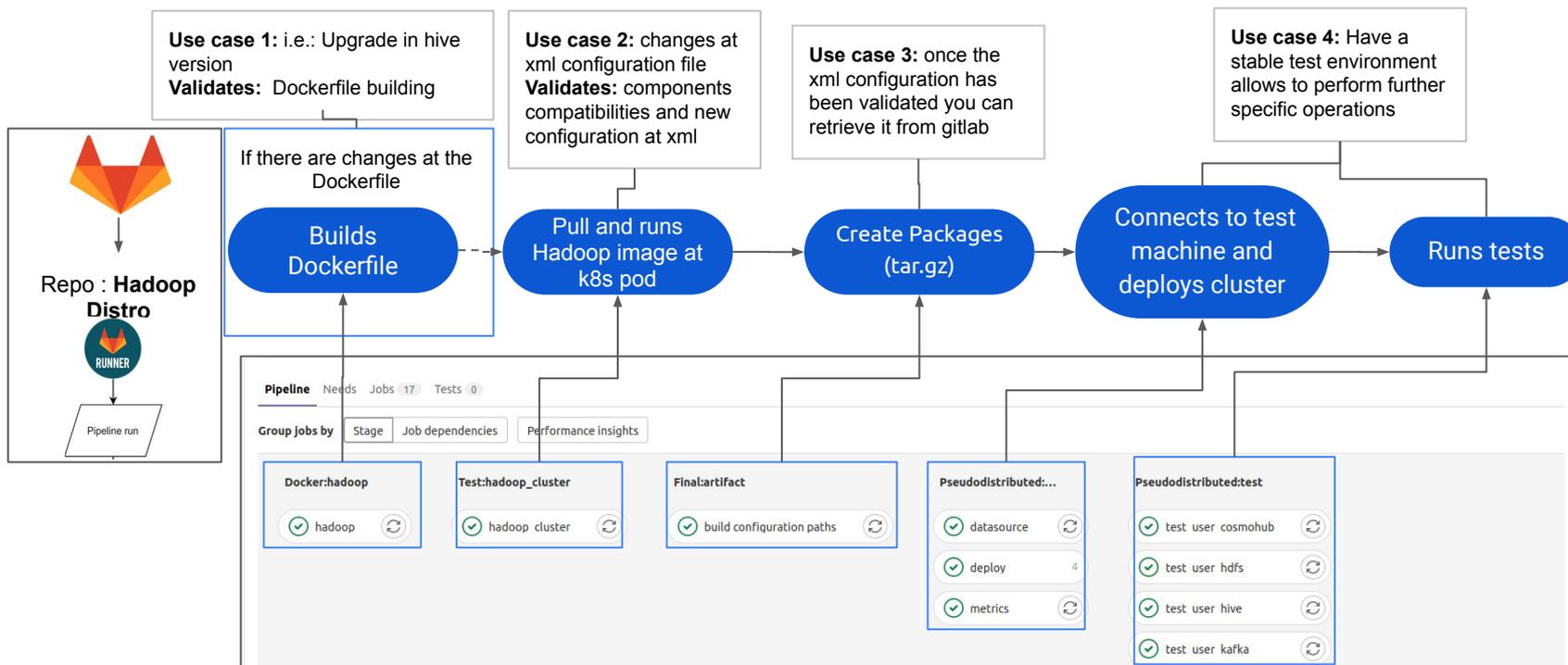
- Pipeline:** Needs 0, Jobs 17, Tests 0.
- Group jobs by:** Stage, Job dependencies, Performance insights.
- Stages and Jobs:**
  - Docker:hadoop:** hadoop (status: success)
  - Test:hadoop\_cluster:** hadoop cluster (status: success)
  - Final:artifact:** build configuration paths (status: success)
  - Pseudodistributed...:** datasource, deploy (4), metrics (status: success)
  - Pseudodistributed:test:** test user cosmo hub, test user hdfs, test user hive, test user kafka (status: success)

Pipeline schema with each step involved in the hadoop distribution project

## 2. Contributions and results: PIC's Hadoop distribution

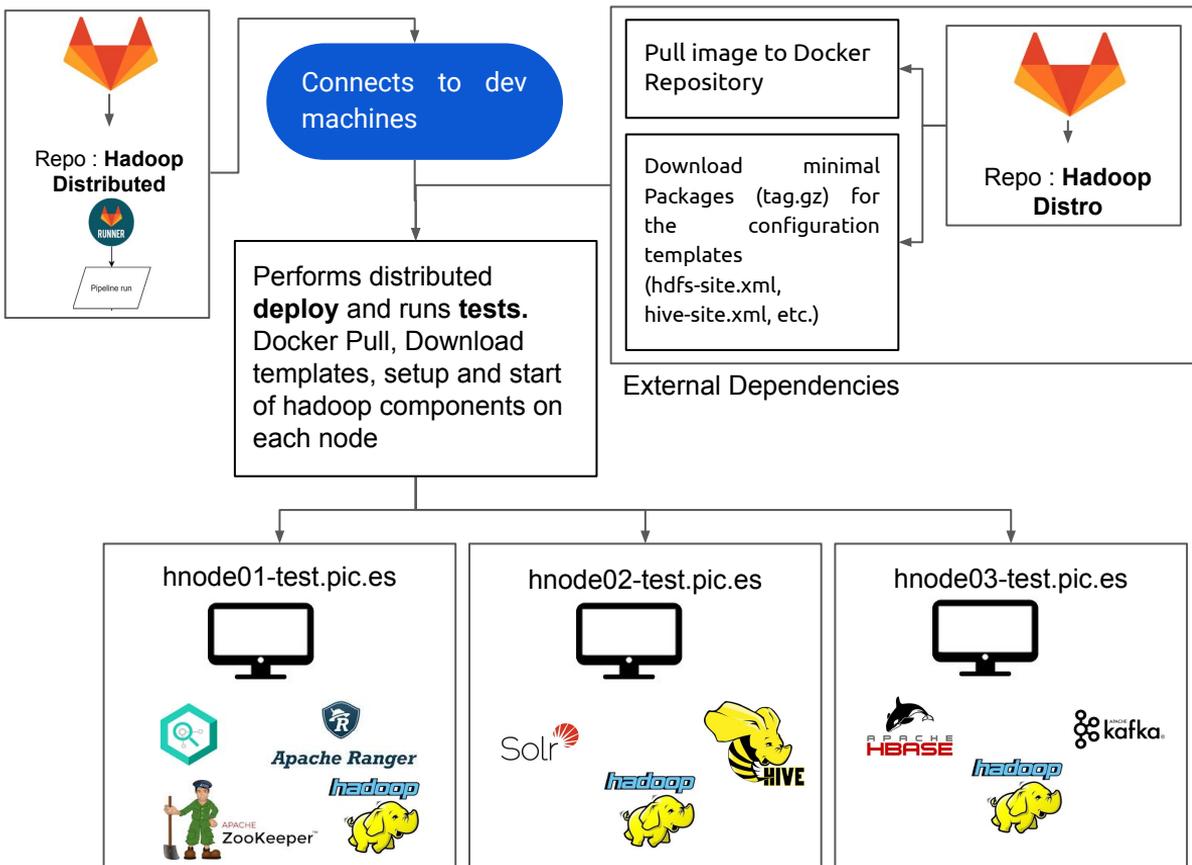
- **Hadoop Distribution pipeline Summary:**

- **Hadoop Distribution:** [Link](#)



## 2. Contributions and results: Distributed deployment

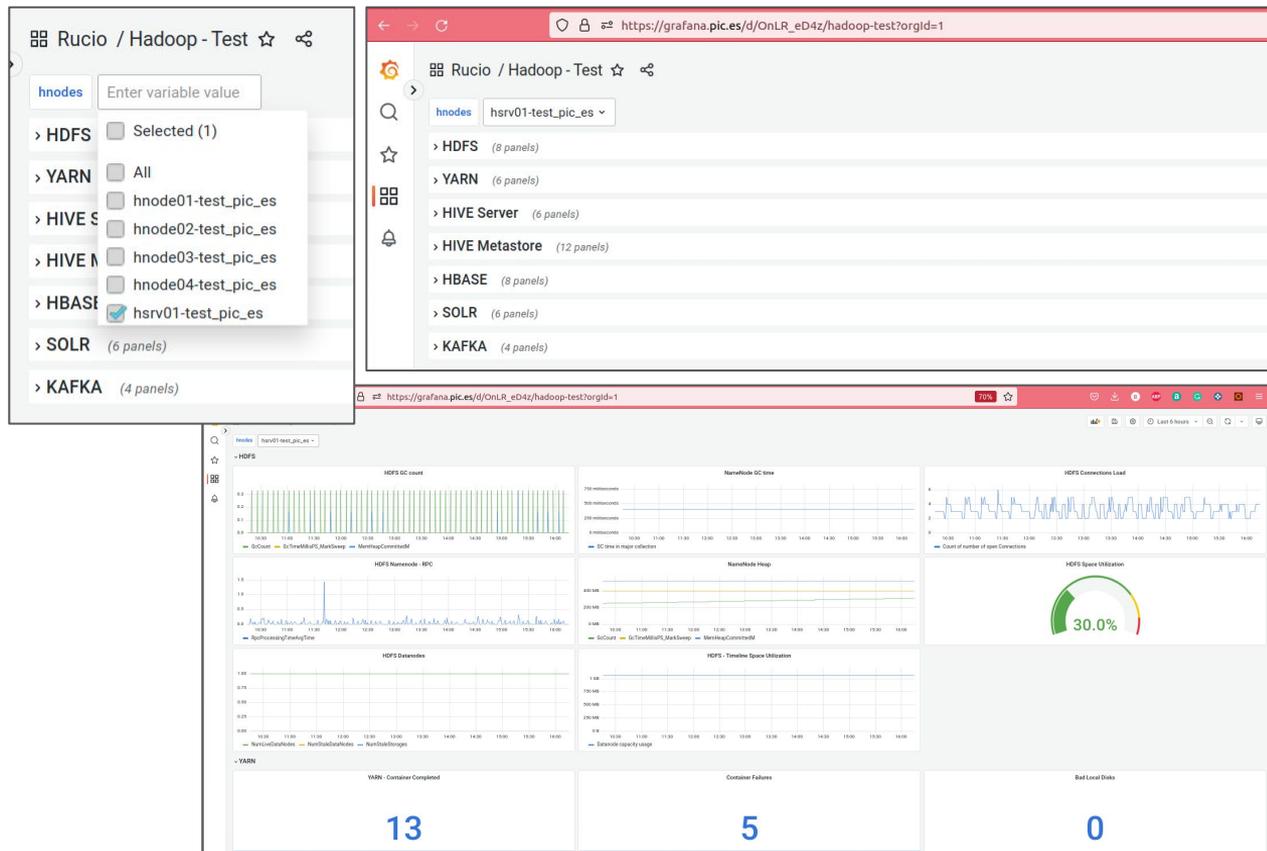
- **Hadoop Distributed Deployment repository:**
- Our hadoop distribution **supports distributed deployment mode.**
  - In this mode, each component of the ecosystem will be started on a arbitrary machine and communicate with each other to distribute the data across
- **Pulls Hadoop image**(generated upon hadoop distro CI/CD)
- It uses a **configuration file** (hadoop.env) in order to **deploy** a service to host , and **load the specific configurations.**
- **Hadoop Distributed:** [Link](#)



## 2. Contributions and results: PIC's Hadoop distribution

### Hadoop Monitoring:

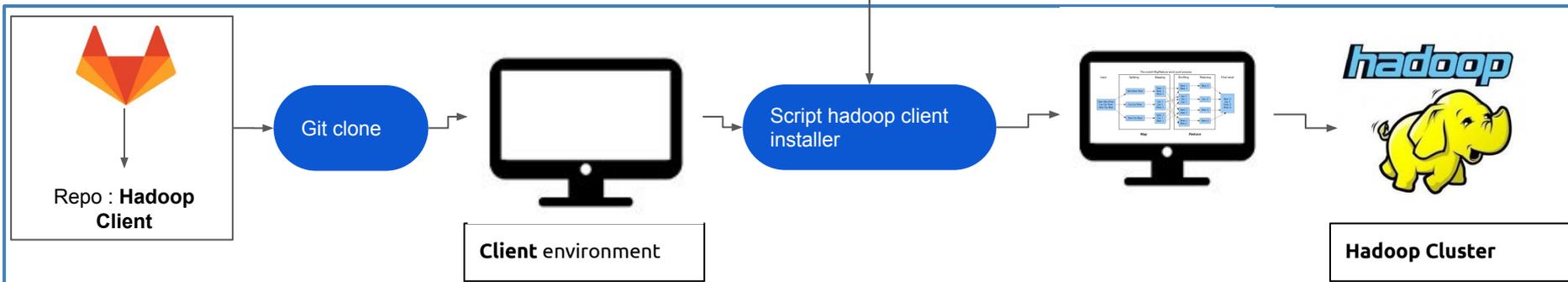
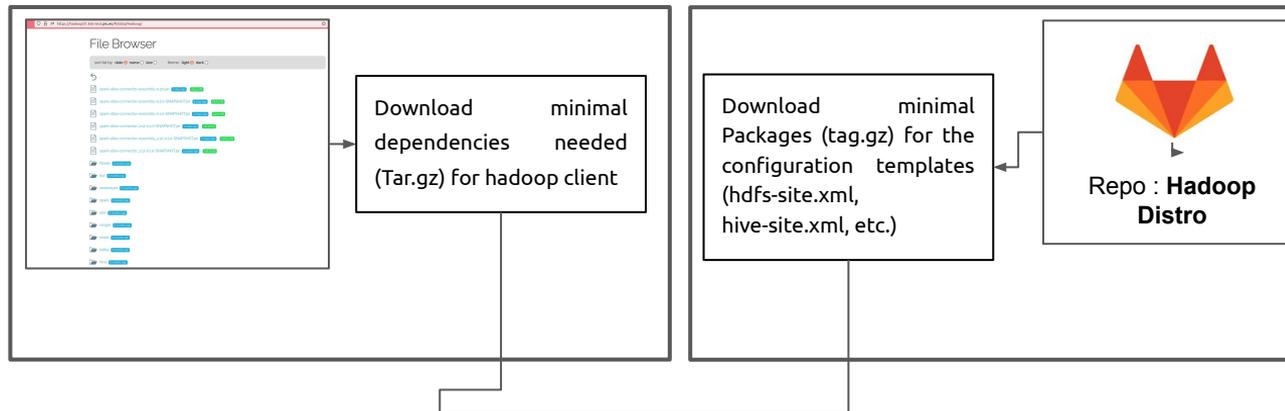
- Even so, there is **no agreed-upon or standardized way** to monitor a Hadoop cluster.
- The **monitoring** of the different environments (**test**, and **dev**) of the hadoop clusters is achieved by exposing a specific **JMX port**. JMX is a particular Java protocol.
- Hence the initials **Java Management Extensions (JMX)**.
- The port exposes all the metrics for each hadoop service, then a **collectd** process is in charge of collecting them, exporting them to **graphite** and painting them in graph.



## 2. Contributions and results: Hadoop Client

- **Client repository:**
- **Minimal installation** of the hadoop cluster (hdfs, mapred, yarn, tez, hive, spark)
- It uses the packaged with all the configuration templates
- Easy to activate through **submodule package** and **facilitate, simplify access to files from outside**
- **Hadoop Client:** [Link](#)

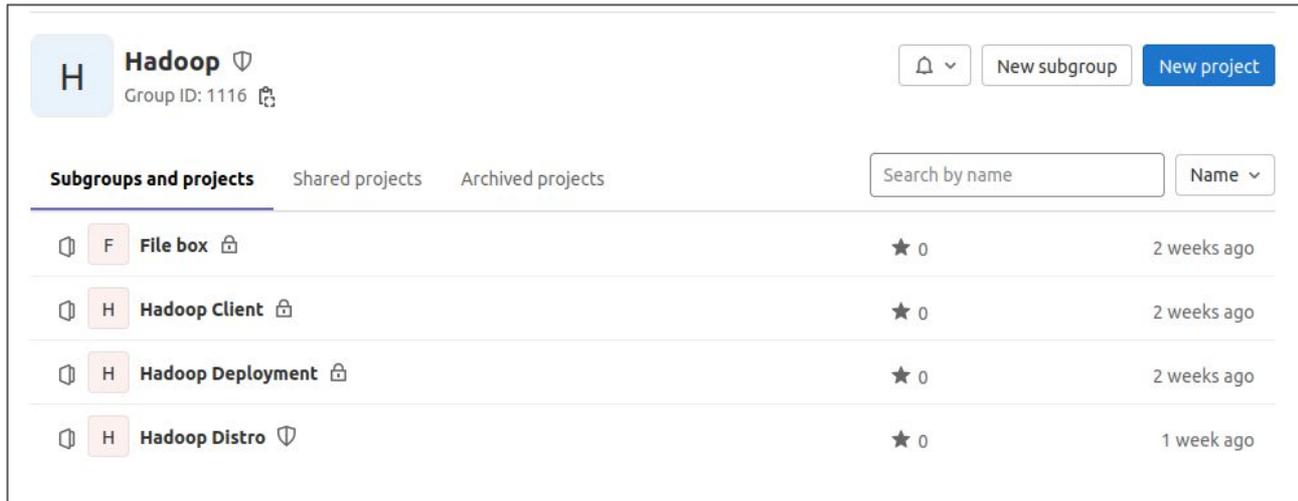
### External Dependencies



## 2. Contributions and results: Summary of Main contributions

- The hadoop distribution project is composed of the following repositories:

- Hadoop distro:** main repository. Here are the configuration files, the dockerfile to build the image, and makes a pseudo-distributed deploy in the test environment.
- Hadoop distributed:** it is used to make a distributed deploy from the dev environment (to be deprecated).
- Hadoop client:** contains the installer for a minimal local installation of the hadoop client.



The screenshot shows the GitHub group page for 'Hadoop'. At the top, there is a group header with the letter 'H' in a blue square, the name 'Hadoop' with a shield icon, and 'Group ID: 1116' with a lock icon. To the right are buttons for 'New subgroup' and 'New project', and a notification bell icon. Below the header are three tabs: 'Subgroups and projects' (selected), 'Shared projects', and 'Archived projects'. A search bar 'Search by name' and a dropdown 'Name' are on the right. The main content is a list of four subgroups:

Icon	Letter	Name	Lock	Stars	Last Update
📁	F	File box	🔒	★ 0	2 weeks ago
📁	H	Hadoop Client	🔒	★ 0	2 weeks ago
📁	H	Hadoop Deployment	🔒	★ 0	2 weeks ago
📁	H	Hadoop Distro	🛡️	★ 0	1 week ago

# Contents

---

## 1. Introduction

- Research topic and problem
- Challenges and Problems
- Objectives

## 2. Contribution and results

- PIC's Hadoop distribution
- Distributed deployment
- Hadoop Client
- Summary of Main contributions

## 3. Conclusions & Project future steps



- The present work has two main contributions:
  - Proposing a complete ecosystem **Hadoop distribution** on distributed scale-out storage system to achieve low latency, massive scalability and fault tolerance **in a single docker image**.
  - Because of its hard management and version upgrade/downgrade can lead to incompatibilities between versions:
    - Proposing a workflow to enable fast, reproducible and testable suitcase for **CI/CD of the present hadoop distribution**:
      - **Test** (pseudo-distributed)
      - **Dev** (distributed upon three machines)



# Project Future steps

---

- Future steps:
  - **Update versions of hadoop distribution components:** [Apache Spark](#) version to 3.2.4, and [apache Solr](#)
  - **Improve resource utilization: Backfilling of idle resources** using [opportunistic containers](#)  
One of the problems with large Hadoop clusters is that we can hardly get the cluster to reach a relatively high resource utilization. Opportunistic containers allocate processes depending on the real-time utilization of the cluster and can even queue processes.
  - **Simplify administrator interfaces:** Implementation of [apache Knox](#): this acts as a secure entry point for the Hadoop cluster, providing an additional layer of security. Knox takes care of authenticating and authorizing user requests before they reach the cluster.
- **Public repository available:**
  - To foster collaboration and accessibility, we have made our Hadoop distribution image and repository publicly available. This allows interested individuals or organizations to access and utilize our customized distribution for their own projects and requirements.
    - [Hadoop distribution image: https://hub.docker.com/r/bruzzese/hadoop-cluster](https://hub.docker.com/r/bruzzese/hadoop-cluster)
    - [Hadoop distribution repository: https://gitlab.pic.es/hadoop/hadoop-distro](https://gitlab.pic.es/hadoop/hadoop-distro)

## PIC staff

- Carles Acosta as technical support for Hadoop configuration
- Dario Graña as technical support for Hadoop in Rocky 8
- Esther Accion as technical support for Freeipa, and Hadoop cluster synchronization
- Jordi Casals as technical support for Hadoop monitoring
- Agustín Bruzzese (Hadoop scripts configuration and development) - [bruzzese@pic.es](mailto:bruzzese@pic.es)
- Pau Tallada (Hadoop distribution coordinator) - [tallada@pic.es](mailto:tallada@pic.es)
- Jorge Carretero (project owner) - [carretero@pic.es](mailto:carretero@pic.es)