

# HPC working seminar for physicists

Scientific Computing Department at HIM

Dr. Dalibor Djukanovic

Dr. Peter-Bernd Otte

bi-weekly meeting – 15.3.2022



# Today's Topics

1. Storage upgrade theory partition
2. TBit link
3. your questions / discussion / requests to the maintainers
4. CI with gitlab.rlp.net

- compact in time (15mins + user questions/discussion).
- bring people together tackling the same problems
- minutes: <https://www.hi-mainz.de/research/computing/hpc-working-seminar/>

# Hot Topics we are working on

- Theory partition:
  - + 900TB, quota updates this afternoon
  - For the next days: before copying TBs, please contact.

# Hot Topics we are working on

- Theory partition:
  - + 900TB, quota updates this afternoon
  - For the next days: before copying TBs, please contact.
- Lustre mount GSI <-> HIM via T-Bit Link
  - Test IP-connection with 10GBit/s
  - Fixed user and group mapping for first tests
  - Who else wants to participate (GSI account necessary)?
- Status:
  - link broke 2 weeks ago, repaired now
  - new computer set up on Mainz site
  - Next: mount GSI lustre on Himster partition

your questions / discussion /  
requests to the maintainers?

# CI with gitlab.rlp.net

Continuous Integration:

first steps

advanced BES3 example

# gitlab.rlp.net

- GitLab:
  - extensive web service for source code management
  - git, wiki, issue tracking, continuous integration & deployment
- details: <https://www.zdv.uni-mainz.de/gitlab/>

- Status: 27.1.2022

# gitlab.rlp.net

- GitLab:
  - extensive web service for source code management
  - git, wiki, issue tracking, continuous integration & deployment
- details: <https://www.zdv.uni-mainz.de/gitlab/>
- Limits:
  - maximal 100 projects/user
  - max. 10 GB /project
  - gitlab runner:
    - 3 VMs, each 2vCPU and 4GB RAM
    - 1h timeout
  - Shared with all on campus
- Status: 27.1.2022



# Why?

- improve code quality:
  - no sudden breaking changes
  - consistent test results

# Why?

- improve code quality:
  - no sudden breaking changes
  - consistent test results

## How?

- checks and tests
- generate docs, container images, precompiled binaries

## EASY!

- add “.gitlab-ci.yml” file

# Key Concepts @ gitlab

- **Job:** ← a set of commands to execute
- **Pipeline** ← a set of jobs organized in stages
- **Environments & Environment Variables**
- **Job Artifacts** ← Outputs
- **GitLab Runner** ← runs actual job

# Jenkins / Gitlab / Github

## Terminology:

gitlab	github.com	Jenkins
CI	Actions	Automation
Group	Organisation	Directory
Project / Repository	Repository	Job
Pipeline	Workflow	Pipeline
Stage	Job	Stage
Job	Step	Step

(→ naming clearly not created by nature.)

# Use YAML

- Human readable data serialisation language
- Simple concept (↔ compare XML!)

- design objectives: only

- scalars,
- arrays and
- dictionarys

- example

```
receipt: Oz-Ware Purchase
my_array:
  - 1
  - 2
items:
  Pride and Prejudice: Alice
  Great Expectations: John
```

# Simple test

- Tests automatically run, when file “.gitlab-ci.yml” exists.

Fails:

```
image: alpine:latest
```

```
test_simple:
```

```
script:
```

```
- exit 1
```

works:

```
image: alpine:latest
```

```
test_simple:
```

```
script:
```

```
- exit 0
```

# Python test (1/2)

- Use gitlab example
- Your repository gets automatically included

```
image: python:latest

# Change pip's cache directory to be inside the project directory since we can
# only cache local items.
variables:
  PIP_CACHE_DIR: "$CI_PROJECT_DIR/.cache/pip"

# If you want to also cache the installed packages, you have to install
# them in a virtualenv and cache it as well.
cache:
  paths:
  - .cache/pip
  - venv/

before_script:
  - python --version # For debugging
  - pip install virtualenv
  - virtualenv venv
  - source venv/bin/activate

test:
  script:
  - python hello.py
```

# Python test (2/2)

- Dependent on return value of hello.py

CI fails:

```
import sys  
  
print("Hello")  
  
sys.exit(1)
```

CI works:

```
import sys  
  
print("Hello")  
  
sys.exit(0)
```



# Advanced example: CI for BES3

## Ingredients:

1. Docker image (CERN CentOS 7 + build environment + BES3-CVMFS)  
<https://gitlab.rlp.net/bes3-mainz/CI/bossdocker/>
2. CI definition file builds the docker image (standard)
3. CI definition file in analysis repository (build generator + )  
[https://gitlab.rlp.net/bes3-mainz/generator/phokhara/-/blob/CI\\_integration/.gitlab-ci.yml](https://gitlab.rlp.net/bes3-mainz/generator/phokhara/-/blob/CI_integration/.gitlab-ci.yml)

Work by Riccardo Aliberti & Thomas Lenz

## dockerfile

```
#CC7
FROM cern/cc7-base

RUN yum -y install --
disableplugin=fastestmirror sudo which make
git redhat-lsb-core

RUN yum -y install --
disableplugin=fastestmirror glibc-devel glibc-
devel.i686 libstdc++-devel.i686

RUN yum -y install --
disableplugin=fastestmirror motif-devel libXpm
mesa-libGLU-devel xz-devel perl boost-devel

RUN list=`find /usr/lib64/ -name "lib*.so.*"`;
for f in $list; do fname=$( basename $f );
newfname=$( echo $fname | cut -d . -f 1 ).so;
if [ ! -f /usr/lib64/$newfname ]; then echo
"ln -s $f /usr/lib64/$( echo $fname | cut -d .
-f 1 ).so"; ln -s $f /usr/lib64/$( echo $fname
| cut -d . -f 1 ).so; fi; done

RUN yum -y install --
disableplugin=fastestmirror
https://ecsft.cern.ch/dist/cvmfs/cvmfs-
release/cvmfs-release-latest.noarch.rpm
RUN yum -y install --
disableplugin=fastestmirror
https://ecsft.cern.ch/dist/cvmfs/cvmfs-
config/cvmfs-config-default-latest.noarch.rpm

RUN yum -y install --
disableplugin=fastestmirror cvmfs cvmfs-config

RUN yum clean all && rm -rf /var/cache/yum/*

COPY cvmfs_default.local
/etc/cvmfs/default.local
COPY cvmfs_ihep_domain.conf
/etc/cvmfs/domain.d/ihep.ac.cn.conf
RUN mkdir -p /etc/cvmfs/keys/ihep.ac.cn/
COPY cvmfs_key_ihep.ac.cn.pub
/etc/cvmfs/keys/ihep.ac.cn/ihep.ac.cn.pub

RUN adduser user
RUN echo "user ALL=(ALL) NOPASSWD: ALL" >>
/etc/sudoers

WORKDIR /home/user

USER user

# No need to have autofs process running, we
mount manually
ENTRYPOINT for DIR in $( grep "REPOSITORIES"
/etc/cvmfs/default.local | cut -d '=' -f 2 |
sed 's|,|\n|g' ); do sudo mkdir -p /cvmfs/$DIR
&& sudo mount -t cvmfs $DIR /cvmfs/$DIR; done
&& /bin/bash
```

## .gitlab-ci.yml for analysis repo

```
image: registry.gitlab.rlp.net/bes3-  
mainz/ci/bossdocker:latest
```

```
build:
```

```
stage: build
```

```
before_script:
```

```
- mkdir -p $CI_PROJECT_DIR/ci-build  
- cd $CI_PROJECT_DIR/ci-build  
- cp $CI_PROJECT_DIR/CI/*.sh .  
- chmod +x setupBES3.sh buildModule.sh  
- ./setupBES3.sh $CI_PROJECT_DIR/ci-build
```

```
script:
```

```
- cd $CI_PROJECT_DIR/ci-build/boss  
- source SetupBoss.sh  
- cd workarea  
- $CI_PROJECT_DIR/ci-build/buildModule.sh
```

```
$CI_PROJECT_DIR
```

```
artifacts:
```

```
paths:
```

```
- $CI_PROJECT_DIR/ci-build/
```

```
expire_in: 1 day
```

```
test:
```

```
stage: test
```

```
before_script:
```

```
- cd $CI_PROJECT_DIR/ci-build  
- source boss/SetupBoss.sh  
- mkdir run  
- cd run  
- cp $PHOKHARAROOT/share/* .
```

```
script:
```

```
- sed -i  
"s|//RootCnvSvc.digiRootOutputFile|RootCnvSvc.di  
giRootOutputFile|g" Phokhara_gensim.txt  
- echo "ApplicationMgr.EvtMax = 10;" >>  
Phokhara_gensim.txt  
- echo "MessageSvc.OutputLevel = 5;" >>  
Phokhara_gensim.txt  
- echo 'DatabaseSvc.DbType="SQLITE";' >>  
Phokhara_gensim.txt  
- echo  
'DatabaseSvc.SqliteDbPath="/cvmfs/boss.cern.ch/d  
atabase/";' >> Phokhara_gensim.txt  
- cat Phokhara_gensim.txt  
- boss.exe Phokhara_gensim.txt
```

```
artifacts:
```

```
paths:
```

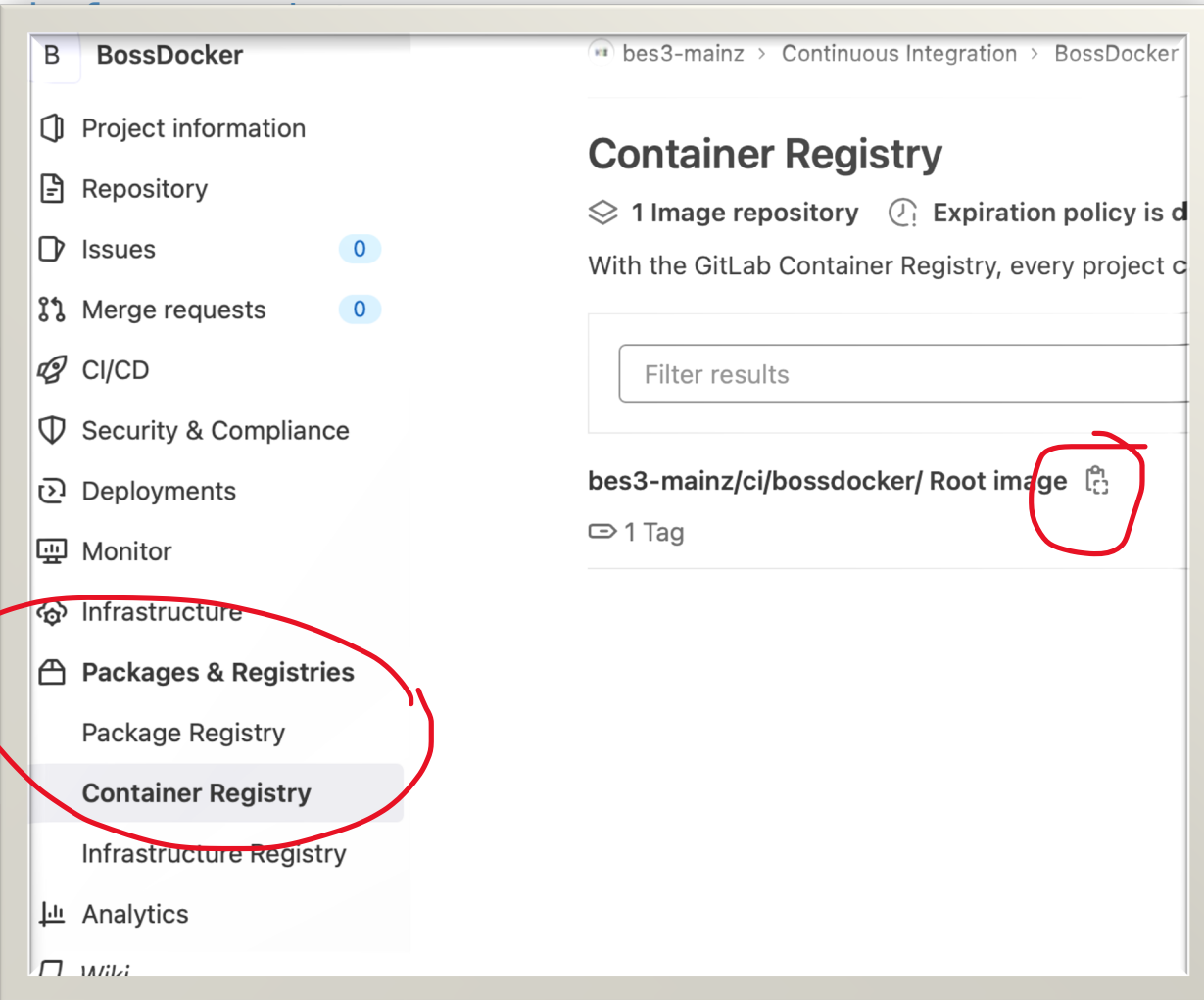
```
- $CI_PROJECT_DIR/ci-build/run/  
expire_in: 1 day
```

## .gitlab-ci.yml for analysis repo

```
image: registry.gitlab.rlp.net/bes3-mainz/ci/bossdocker:latest
```

```
build:  
stage: build
```

```
- cd $CI_PROJECT_DIR/ci-build  
- source boss/SetupBoss.sh  
- mkdir run  
- cd run  
- cp -r $CI_PROJECT_DIR/share/* .
```



Look out for:

Package and Registry > Container registry,  
Copy the path of your image

```
'DatabaseSvc.SQLiteDbPath= /cvmfs/boss.cern.ch/d  
atabase/";' >> Phokhara_gensim.txt  
- cat Phokhara_gensim.txt  
- boss.exe Phokhara_gensim.txt  
artifacts:  
paths:  
- $CI_PROJECT_DIR/ci-build/run/  
expire_in: 1 day
```

```
before_script:
```

# Who else uses CI?

- Who else is using CI @ HIM?
- SPEC/F/BES 3 group, analysis framework
- unix group @ ZDV
- SDE group @ GSI

# Next meeting, 29th March 2022

- Only technical / short meetings during semester break
- hand in your topics!