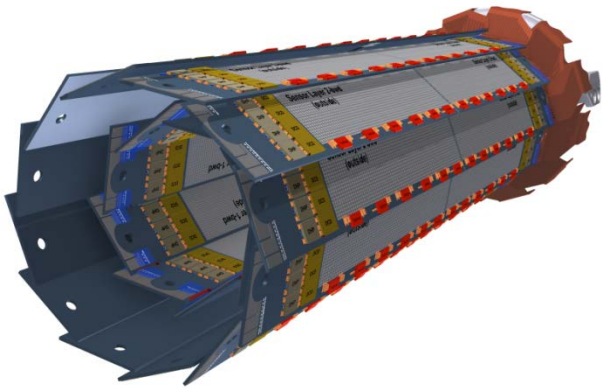
Software issues
Setup of PC with SL7

Common Effort of
Jakob Haidl, Christian Koffmane, Philipp Leitl, Florian Lütticke, Felix Müller,
Botho Paschen, Eduard Prinker, *Martin Ritter*, *Thorsten Röder*,
Harrison Schreeck, Benjamin Schwenker, Philipp Wieduwilt

10th Belle II VXD Workshop
Santander – Sep 14-16, 2016



Structure of Setup PC

Git repositories (DESY stash):

Install scripts, DHH Software, Measurement and Analysis scripts

RPM Packages:

EPICS for DHH, PS

Webpage

CS-Studio Snapshot: https://sussrv01.ziti.uni-heidelberg.de/~ritzert/PXD/CSS4/

DHH Firmware: https://www.e18.physik.tu-muenchen.de/belle/firmware/

Install script took care of almost everything except the egit client in CS-Studio

Starting from scratch

1) Download iso-Image

   either create a DVD or a USB Stick ($ `dd if=/path/to/SL7.iso of=/dev/sdb`)

2) Get a GridKA certificate (see appendix)

3) Get a personal DESY account (see appendix)

4) Talk to your IT to grant access to desy.stash.de:7999 (**PORT 7999**)

5) Checkout git repository (on your personal PC)

   ($ `git clone ssh://git@stash.desy.de:7999/b2g/pxd_sc_testsetup.git`)

from scratch

Therefore, you need your private key. So install your private key and put it to desy stash (See appendix)

Option1:

Copy the repository to the TestPC (scp -R pxd_sc_testsetup.git testpc:~/.)

Option 2:

Clone the git repository from your personal PC to the TestPC

$ `git clone MYUSERNAME@testpc:~/pxd_sc_testsetup`

And change origin to desy stash

$ `git remote set-url origin ssh://git@stash.desy.de:7999/b2g/pxd_sc_testsetup.git`

Starting from scratch

6) run install script

```
$ cd pxd_sc_teststup/system-config
$ bash install.sh
```

7) run network configuration script

```
$ cd network
$ bash install.sh eth1 (choose your network adapter)
```

or

```
$ bash install_separate.sh eth1 eth2 eth3 (PS, DHH_SC, DHH_data)
```

Installation should be completed, now one has to configure cs-studio

See `$ vim pxd_sc_testsetup/system-config/README.md`

Permissions and commits

The key which is provided already in the pxd_sc_testsetup/system-config folder only allows to read, no commits are possible

Since multiple users are using the same testsetup, institute keys have been introduced, i.e. every institute has its own key granting read/write access

(How to setup your institute key: see appendix)

create a .gitconfig (due to the shared institute key of the PCs, it is impossible to see the author of the commits if one forgets --author “MYUSERNAME”)

```
$ vim .gitconfig
    [user]
    └──────email = chk@hll.mpg.de
    └──────name = HLL Munich (pxdtest6)
```

example

`git add` adds file which should be commited (by default None)

`git commit --author “MYSUERNAME“` commits to the local repository

`git commit --author “MYSUERNAME <EMAIL>“` required the first time

`git push` commits to desy.stash.de

git repositories

ssh://git@stash.desy.de:7999/b2g/pxd_sc_css.git

ssh://git@stash.desy.de:7999/b2g/pxd_sc_dhh.git

ssh://git@stash.desy.de:7999/b2g/pxd_sc_testsetup.git

ssh://git@stash.desy.de:7999/b2g/pxd_sc_configdb.git

In pxd_sc_css there will be a lab_framework

This lab_framework contains the cleanup of the code

- `basics` (plot a frame, no mapping etc)
- `calibrations` (perform all the measurements and analysis like gated-mode, delays, etc.)
- `devices` (control commercial devices, like PowerSupplies, PulseGenerator, Chiller, SMU)
- `lib` (libraries like epics_utils, dhp_utils, functions that will be used in multiple measurement and analysis scripts)
- `testbeam_2016_04` (special scripts for the last testbeam, includes DHC)

Conclusion

We are physicists not computer experts

Read this presentation as a kind of manual

Complicated structure (access keys, different locations etc.)

Many simplifications by install script (does almost everything except of accessing via SSH (key) and setting up CS-Studio)

Takes approximately half a day to install from scratch

APPENDIX

Git migration

Where: stash.desy.de - Server which hosts all the files from the svn repositories

Port: **7999** (talk to your IT regarding firewalls)

How:

- With a personal account
  - https://confluence.desy.de/display/BI/Belle+II+Registration+Procedure
    - Register at GridKA (fill out form and sent it online & fill out form, get sign from responsible person at your institute (director, …) and send it by **mail** (not email) to GridKa (IMPORTANT: use your personal computer because only with this web browser you use to fill out the form, you will be able to get the certificate)
    - Receive email and follow instructions
    - Install GridKa certificate in your web browser
    - Get an DESY account by identifying yourself with GridKa
      - https://belle2-request.desy.de/
- With a PC account (for test setup PCs)
  - …

How to create the personal key?

Go to your personal PC:

```
$ ssh-keygen –t rsa
```

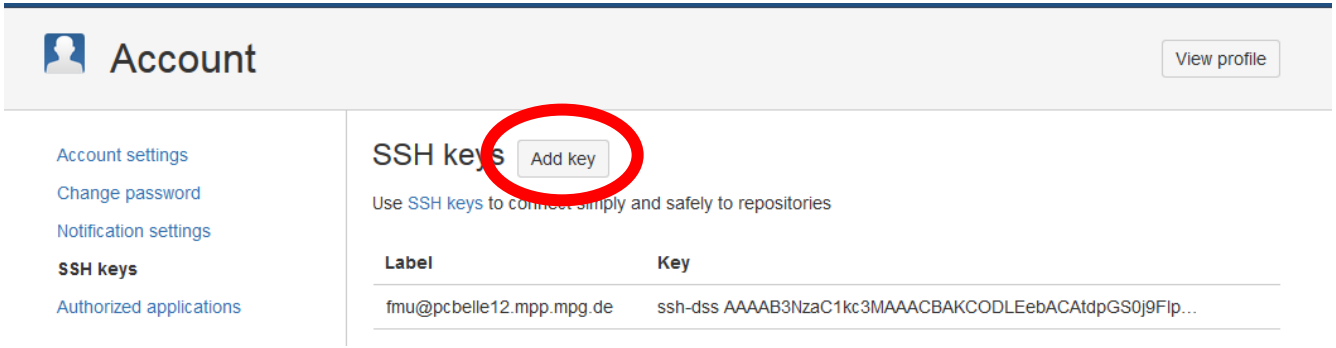
Name: you do not need a specific name, you can easily confirm by return

It will create “`id_rsa`” and “`id_rsa.pub`” (private and public key)

```
$ vim ~.ssh/config
```

```
Host stash.desy.de
     Hostname stash.desy.de
     Port 7999
     IdentityFile ~/.ssh/id_rsa
```

Webbrowser: `https://stash.desy.de/plugins/servlet/ssh/account/keys`



Click add key and copy your public key there, you will get the **public** key by

```
$cat ~/.ssh/id_rsa.pub
```

Try whether you have access:

```
$ git ls-remote ssh://git@stash.desy.de:7999/b2g/pxd_sc_css.git
```

How to create the institute keys?

Go to one of the test setups:

```
$ ssh-keygen –t rsa
```

Name: “desystash_INSTITUTE”, e.g. “desystash_MPP”

It will create “desystash_<institute>” and “desystash_<institute>.pub” (private and public key)

```
$ vim ~.ssh/config
```

```
Host stash.desy.de
     Hostname stash.desy.de
     Port 7999
     IdentityFile ~/.ssh/stashkey_MPP
```

configures which file/key will be used

$ send the “desystash_<institute>.pub” to fmu@mpp.mpg.de

Having received the confirmation you can test the connection by:

```
$ git ls-remote ssh://git@stash.desy.de:7999/b2g/pxd_sc_css.git
```