



cern.ch/allpix-squared

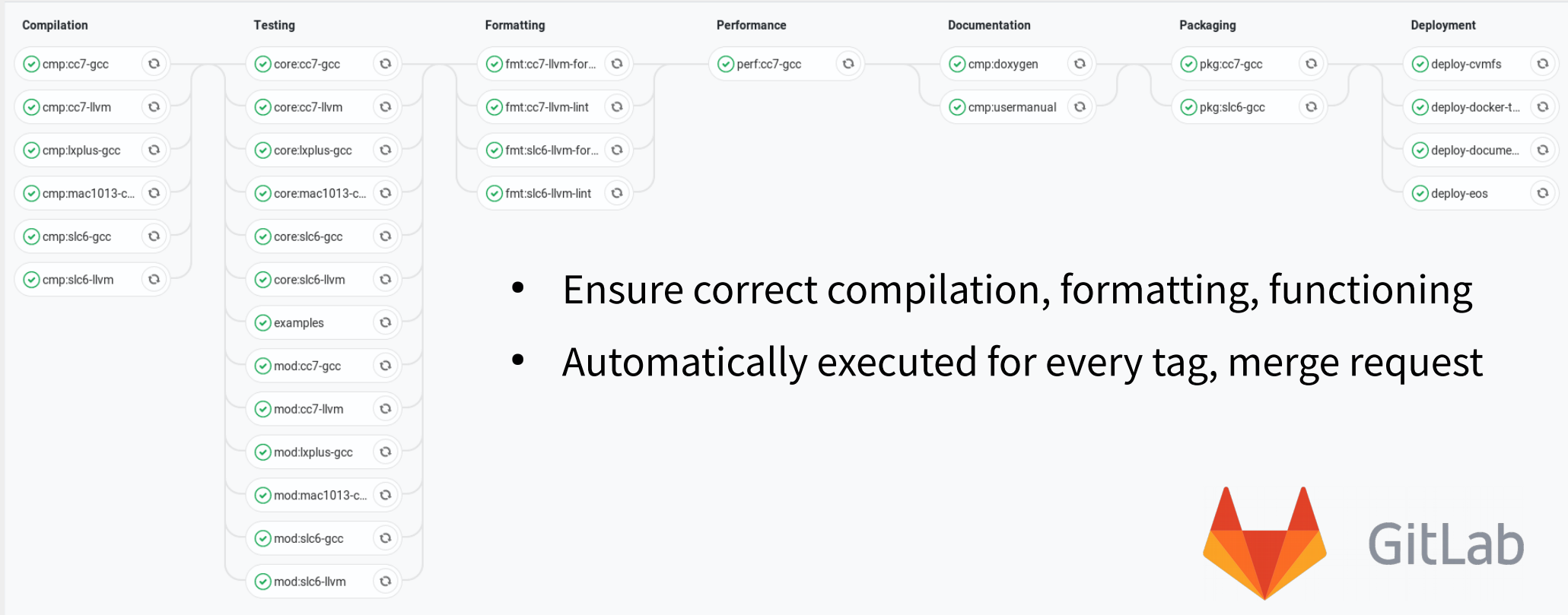
Installing Allpix Squared

A little 101 on the different options

Simon Spannagel, CERN

1st Allpix Squared User Workshop
CERN, 26 November 2018

Excursion: Continuous Integration



- Ensure correct compilation, formatting, functioning
- Automatically executed for every tag, merge request



CI Stages

- **Compilation**
builds the framework from source on Scientific Linux 6, CentOS7, and Mac OS X using GCC, Clang, AppleClang
- **Testing**
executes functionality tests for the core (correct parsing of configs, coordinate transformations...) and modules (constant behavior, no change to physics)
- **Formatting**
makes sure that changes concern code, not its style (white spaces, tabs...) and ensures proper usage of recent C++ features (linting, best practices)
- **Performance**
Execute a few tests with several thousand events, measure execution time

CI Stages

- Documentation
generate user manual from LaTeX sources, generate Doxygen code reference
- Packaging
generate tarballs for deployment/publication
- Deployment
 - Publish new version on CVMFS
 - Publish new Docker image in registry
 - Publish user manual and code reference on website
 - Publish binary tarballs on website

Me? How?

- Enable runners (to execute jobs)

Settings

- General
- Members
- Integrations
- Repository
- CI / CD**
- Audit Events

Runners

Register and see your runners for this project.

A Runner is a process which runs a job. You can setup as many Runners as you need. Runners can be placed on separate users, servers, and even on your local machine.

Each Runner can be in one of the following states:

- active** - Runner is active and can process any new jobs
- paused** - Runner is paused and will not receive any new jobs

- Pipeline fails: do read output of the failing job!

```
3 warnings treated as errors
/builds/allpix-squared/allpix-squared/src/core/geometry/Detector.cpp:185:29: error:
parameter 'field' is passed by value and only copied once; consider moving it to avoid
unnecessary copies [performance-unnecessary-value-param,-warnings-as-errors]
    electric_field.setGrid(field, sizes, scales, offset, thickness_domain);
                        ^
                        std::move( )
/builds/allpix-squared/allpix-squared/src/core/geometry/Detector.cpp:191:33: error:
parameter 'function' is passed by value and only copied once; consider moving it to avoid
unnecessary copies [performance-unnecessary-value-param,-warnings-as-errors]
    electric_field.setFunction(function, thickness_domain, type);
                              ^
                              std::move( )
/builds/allpix-squared/allpix-squared/src/core/geometry/DetectorField.hpp:51:27: error:
member initializer for 'field_type_' is redundant [modernize-use-default-member-init,-
warnings-as-errors]
    DetectorField() : field_type_(FieldType::NONE){};
                    ^
```

⊗ Pipeline #582017 from fieldparser

formatting

- ⊗ **fmt:cc7-llvm-lint**
- ⊗ **fmt:slc6-llvm-lint**
- ✓ **fmt:cc7-llvm-format**
- ✓ **fmt:slc6-llvm-format**

The Options

- CVMFS Installation
- Docker Images
- Binary Tarballs
- Compiling from source

CVMFS – CernVM File System

“provides a scalable, reliable and low-maintenance software distribution service. It was developed to assist High Energy Physics (HEP) collaborations to deploy software on the worldwide-distributed computing infrastructure used to run data processing applications”

<https://cernvm.cern.ch/portal/filesystem>

- Central installation of software for SLC6 and CC7
- On any machine with CVMFS, simply *source* corresponding script and use the SW
- Many packages available: ROOT, Geant4, LCIO, Delphes, FastJet, ...

... Allpix Squared

CVMFS – CernVM File System

- Using project space of CLICdp at `/cvmfs/clicdp.cern.ch/software/allpix-squared/`
- All versions since v1.1 available
Nightly build of *master* in “latest”
- Each version built for SLC6 and CC7:
`/1.3/x86_64-centos7-gcc7-opt/`
`/1.3/x86_64-slc6-gcc7-opt/`
- Load all dependencies, C++ libraries & set up \$PATH using `setup.sh` file:

```
2018-11-22 14:42 simonspa@lxplus079:~$ ll /cvmfs/clicdp.cern.ch/software/allpix-squared/
total 4.5K
drwxrwxr-x. 4 cvmfs cvmfs 4 Jan 11 2018 1.1.0/
drwxrwxr-x. 4 cvmfs cvmfs 4 May 4 2018 1.1.1/
drwxrwxr-x. 4 cvmfs cvmfs 4 May 4 2018 1.1.2/
drwxrwxr-x. 4 cvmfs cvmfs 4 Jun 13 14:44 1.2/
drwxrwxr-x. 4 cvmfs cvmfs 4 Aug 2 10:28 1.2.1/
drwxrwxr-x. 4 cvmfs cvmfs 4 Sep 10 11:22 1.2.2/
drwxrwxr-x. 4 cvmfs cvmfs 4 Nov 13 17:56 1.2.3/
drwxrwxr-x. 4 cvmfs cvmfs 4 Nov 21 15:02 1.3/
drwxrwxr-x. 4 cvmfs cvmfs 5 Nov 21 06:33 latest/
```

```
$ source /cvmfs/clicdp.cern.ch/software/allpix-squared/1.3/x86_64-centos7-gcc7-opt/setup.sh
$ allpix --version
Allpix Squared version v1.3
built on 2018-11-21, 14:00:57 UTC
```



“Docker is a computer program that performs operating-system-level virtualization, also known as *containerization*”

[https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))

- Not a traditional virtualization, where full machine is emulated
- Run software in container, encapsuled from operating system
 - All dependencies met inside Docker container
 - Very little performance penalty compared to native execution
- Software distributed via “Docker images”
 - Hosted in “docker registries”

- CERN's GitLab instance provides Docker registries for each project:
https://gitlab.cern.ch/allpix-squared/allpix-squared/container_registry
- How to use this? The user manual has it:

- Start an interactive shell inside the Docker:

```
$ docker run --interactive --tty \
  --volume "$(pwd)":/data \
  --name=allpix-squared \
  gitlab-registry.cern.ch/allpix-squared/allpix-squared:v1.3 \
  bash
```

- Directly start a simulation:

```
$ docker run --tty --rm \
  --volume "$(pwd)":/data \
  --name=allpix-squared \
  gitlab-registry.cern.ch/allpix-squared/allpix-squared:v1.3 \
  "allpix -c my_simulation.conf"
```

Cheat Sheet:

<code>--tty</code>	Allocate a pseudo-TTY
<code>--rm</code>	Automatically remove container when it exits
<code>--interactive</code>	Keep STDIN open even if not attached
<code>--volume</code>	Bind mount a volume
<code>--name</code>	Assign a name to the container

Binary Tarballs

- Are produced anyway, as by-product of the Continuous Integration
- We collect them, put them on the web:
<https://cern.ch/allpix-squared/releases/>
- Available for SLC6, CC7

Compiling from Source #1 – Satisfy Dependencies

- Allpix Squared depends on
 - ROOT6 - objects, object history, storing/reading from/to files
 - Eigen - library for fast algebra, used by some modules
- Standard installation will require
 - Geant4 - particle interaction with matter & tracking
- On SLC6/CC7 machines, all is ready:

\$ source etc/scripts/setup_lxplus.sh

Compiling from Source #2 – Configuring the Build

- We use CMake to configure the build
 - Cross-platform, same scripts on Linux & Mac OS
 - Weird-to-write but fairly easy to use/run
- Prefer out-of-source builds: make new folder to compile in

```
$ cmake /path/to/allpix-squared/
```

- Options can be set using “-D...=...” e.g.

```
$ cmake -DBUILD_VisualizationGeant4=OFF /path/to/allpix-squared/
```

- Graphical/ncurses tools can aid in configuration, try:

```
$ ccmake /path/to/allpix-squared/
```

Compiling from Source #3 – Building

- Well...

```
$ make -j20  
$ make install
```

(assuming you have a 20-core machine...)

- Wait a bit
- Run.

A Note on Visualization

- We currently use the Qt visualization viewer by Geant4
- Requires the Geant4 version used to be built with options

```
-DGEANT4_USE_QT=ON  
-DGEANT4_USE_OPENGL_X11=ON
```

enabled

- Unfortunately currently not available on CVMFS installation of Geant4
- Working on resolving this issue

