

# Container Technology for Phenomenology Tools



INDIGO - DataCloud

RIA-653549



**CSIC**

CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS



LABORATÓRIO DE INSTRUMENTAÇÃO E  
FÍSICA EXPERIMENTAL DE PARTICULAS

**Isabel Campos, Alvaro López and Pablo Orviz - CSIC (Spain)**

**Jorge Gomes – LIP (Portugal)**

**Sven Heinemeyer CSIC (Spain)**

**Emanuele Bagnaschi DESY (Germany)**



INDIGO-DataCloud is co-funded by the  
Horizon 2020 Framework Programme

# Outline



- Motivation: Containers and Phenomenology codes
- A pilot case: Mastercode
- Building a container for *Mastercode*
- Automation tools for building and maintaining containers
- Running Mastercode everywhere: *udocker*
  - On your desktop
  - On a Linux Cluster
  - On the Grid

# Containers - definition

**“Linux Containers” is a technology provided by the Linux kernel, to “contain” a group of processes in an independent execution environment: this is called a “container”.**

- The “group of processes” can be for example a software package
- Containers are the future of software provisioning both in industry and academia:  
eg. see <https://access.redhat.com/articles/1353593>

## **Advantages of Containers versus classical Virtual Machines**

- Containers are much more light-weighted than a Virtual Machine: they start faster, run faster, and many more fit on the same hardware host machine
- They provide an enormous simplification of the software deployment processes in any Linux platform with a kernel advanced enough to support container technology (kernel version > 3.8.x)

## **We have available nowadays two types of virtualization:**

- Virtual Machines with Hypervisors: (Xen, KVM) i.e. another OS instance by Virtualization
- Containers: Lightweight Process level handled by the linux kernel

# Containers - docker



- To build containers, we use the system software provided by “Docker”
- Docker is an Open Source Linux container engine.
  - <https://www.docker.io/>
  - git repository at: <https://github.com/dotcloud/docker.git>
- Docker is optimized for the deployment of applications as containers
  - Other options like the available lxc are more oriented to deployment of system software/sysadmin
- One can generate a completely tailored “docker image” of any Linux Operating System, with all the required libraries, compilers, source codes,...

## “docker image” ≠ “VM image”

- It is a Snapshot of the OS which has been modified to run as a container
- In particular, when comparing with an image for a hypervisor (VM), it has no Kernel.
- Still they are large, minimum 200MB (essentially libraries).
- Those “images” will not run when launched with a hypervisor !



cern.ch/mastercode



- **Collaborative effort** between Experimentals and Theorist in High Energy Particle Physics

The code takes as input experimental data coming from Particle Accelerators and Astrophysics observations, in order to build a consistent model of Nature explaining the experimental data

- ✓ It focuses on the search for Supersymmetric models

It is supported by an ERC Advanced Grant:  
***“Exploring the Terauniverse with the LHC, Astrophysics and Cosmology”*** (J. Ellis)

See <http://johne.web.cern.ch/johne/>

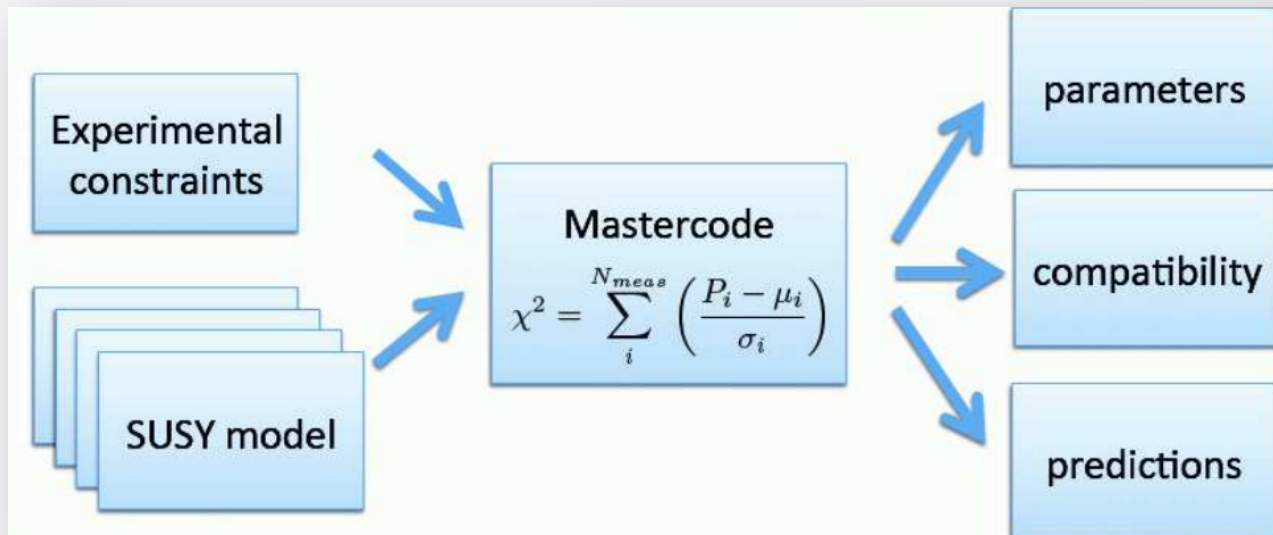
### Members of the MasterCode collaboration

- E. Bagnaschi, O. Buchmüller, R. Cavanaugh, M. Citron, A. De Roeck, M. Dolan, J. Ellis, H. Flücher, S. Heinemeyer, G. Isidori, S. Malik, D. Martinez, K. Olive and K. Sakurai, K. de Vries, G. Weiglein

### Codes

- RGE running: SoftSUSY
- Higgs,  $(g-2)_\mu$ : FeynHiggs
- Higgs observables: HiggsBounds and HiggsSignal
- B-physics: SuFla
- B-physics: SuperIso
- EWPO: FeynWZ
- Dark matter observables: Micromegas
- Dark matter observables: DarkSUSY
- Recast of LHC searches: Atom
- Recast of LHC searches: Scorpion
- SUSY decay modes: SDECAY

# Mastercode computing model



**Monday, 3.40pm:**

*New SUSY Fits with MasterCode*  
(Sven Heinemeyer)

**Tuesday, 4.30pm:**

*SUSY Predictions for Dark Matter Searches from MasterCode*  
(Matt Dolan)

# A pilot case: Mastercode



- Mastercode is an “über-code” written in C++ which connects all the different codes
- It does **parametric runs**: scanning through large parameter spaces, in **single core** mode
- Those original codes are treated as subroutines, or sub-codes
- The sub-codes are written in C++ or Fortran. Many different authors, often legacy code is there....
- **The “über-code” is not public...**
- **Perfect candidate to be “containerized”**: problems to run the code in current computing centers because it is difficult to install (library dependences and compatibility issues with local software, etc...)
- One of the main problems they have: finding a computing center where it works!

# Building a container for Mastercode

Description of the automated process via github + dockerhub:

1. We have created a project in github: <https://github.com/indigo-dc/docker-mastercode>
2. The project contains just a Dockerfile to build a fedora-based container that includes all the libraries and system software needed to run Mastercode
3. We then created a hub in docker for the container to be built automatically by docker <https://hub.docker.com/r/indigodatacloud/docker-mastercode>
4. Finally, we connected github-docker to automatically build the container

# Github project: Dockerfile

indigo-dc / docker-mastercode

<> Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Docker image for Phenomenology tools — Edit

14 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

isabel-campos-plasencia master update Latest commit 39d4433 on May 5

<a href="#">Dockerfile</a>	master,update	2 months ago
<a href="#">LICENSE</a>	Initial commit	3 months ago
<a href="#">README.mastercode</a>	README.mastercode added	3 months ago
<a href="#">README.md</a>	Create README.md	3 months ago

[README.md](#)

## phenotools

### Docker Image adapted to the usage of Phenomenology Tools

indigo-dc / docker-mastercode

<> Code Issues 0 Pull requests 0 Wiki Pulse Graphs

Branch: master docker-mastercode / Dockerfile

isabel-campos-plasencia master update

1 contributor

56 lines (43 sloc) | 1.23 KB

```
1 FROM fedora:23
2
3 ENV sqlite_version="3100200"
4
5
6 RUN dnf install -y \
7     autoconf \
8     automake \
9     git \
10    gcc \
11    gcc-c++ \
12    gcc-gfortran \
13    libX11-devel \
14    make \
15    patch \
16    rpm-build \
17    tar \
18    wget \
19    which
```

# Connect dockerhub ↔ github




PUBLIC | AUTOMATED BUILD


indigodatacloud/docker-mastercode ★

Last pushed: 2 months ago

Repo Info Tags Dockerfile Build Details Build Settings Collaborators Webhooks Settings


Short Description 

Environment container for phenomenology tools

Full Description 


phenotools

Docker Image adapted to the usage of Phenomenology Tools


Docker Pull Command 

```
docker pull indigodatacloud/docker-
```

Owner

 indigodatacloud

Source Repository

 [indigo-dc/docker-mastercode](#)

We created a project in **dockerhub**, which reads as **source repository** our **github** Mastercode repository.

In particular dockerhub aware of the changes we may commit to the Dockerfile

Dockerhub provides a service to build Containers automatically on their cloud Facilities.

We use this service to maintain up-to-date The container for Mastercode

# Connect dockerhub ↔ github



A change in the dockerfile, triggers automatically a new build of the container

PUBLIC | AUTOMATED BUILD

indigodatacloud/docker-mastercode ★

Last pushed: 2 months ago

Repo Info Tags Dockerfile **Build Details** Build Settings Collaborators Webhooks Settings

Build Settings

When active, builds will happen automatically on pushes.

The build rules below specify how to build your source into Docker images. The name can be a string or a regex. The Docker Tag name may contain variables. We currently support {sourcerefer}, which refers to the source branch/tag name. [Show more](#)

Source Repository  
indigo-dc/docker-mastercode

Type	Name	Dockerfile Location	Docker Tag Name	
Branch	master	/	latest	+ <b>Trigger</b>
Branch	devel	/	Same as branch	- <b>Trigger</b>

Save Changes

PUBLIC | AUTOMATED BUILD

indigodatacloud/docker-mastercode ★

Last pushed: 2 months ago

Repo Info Tags Dockerfile **Build Details** Build Settings Collaborators Webhooks Settings

Status	Tag	Created	Last Updated
🚀 Success	latest	2 months ago	2 months ago
✓ Success	latest	2 months ago	2 months ago
✓ Success	latest	2 months ago	2 months ago
✓ Success	devel	2 months ago	2 months ago
! Error	devel	2 months ago	2 months ago
✓ Success	latest	2 months ago	2 months ago

# Using the Mastercode container with docker

On your local Linux box, **with root user privileges**:

1. Install docker with your favourite package manager (eg. "**# yum install docker**")
2. Start the docker daemon: **# systemctl start docker**
3. Download the container: **# docker pull indigodatacloud/docker-mastercode**
4. *Run it*

```
# docker run -t -i -v $HOME \  
-w $HOME/mcpp-master \  
indigodatacloud/docker-mastercode \  
/bin/bash
```

← mounts your HOME directory when running the container  
← Mastercode sources directory is set as working directory  
← Name of the container to run  
← Get a bash shell inside the container

**Notice:** the container does not contain the code, only the environment necessary to run it:

- Developers keep doing their own modifications to Mastercode in their private copy.
- In **MacOS**, download the Docker Tool Box for Mac, and start from (3).
- Yes, you can run Mastercode on your Mac !

# On MacOS running boot2docker:



```
isabel — @db926fad41cc:/ — docker — 113x23
cscdiica@fs6802:~
@db926fad41cc:/
Using default tag: latest
^Cgaepc26:~ isabel$ docker pull indigodatacloud/docker-mastercode
Using default tag: latest
latest: Pulling from indigodatacloud/docker-mastercode
a3ed95caeb02: Pull complete
236608c7b546: Pull complete
75c50b6518be: Pull complete
e9058dcbc910: Pull complete
7265dc1bc8b1: Pull complete
0912b41dded9: Pull complete
977aca495a74: Pull complete
ebfcb6c1d599: Pull complete
fa489049f716: Pull complete
f2d0727f9b49: Pull complete
92d8d3220efd: Pull complete
Digest: sha256:fbcdc2d79981564905132535ccb1c3d9ba3f290cb5b00d96c7ad7840bccc8f18
Status: Downloaded newer image for indigodatacloud/docker-mastercode:latest
gaepc26:~ isabel$ docker run -t -i indigodatacloud/docker-mastercode /bin/bash
[root@db926fad41cc /]# sw_vers -productVersion
bash: sw_vers: command not found
[root@db926fad41cc /]# cat /etc/fedora-release
Fedora release 23 (Twenty Three)
[root@db926fad41cc /]#
```

Boot2docker uses virtualization (VirtualBox) to run the a Tiny Linux system where the containers are executed. It does NOT provide native execution.

# On a Linux Cluster / HPC system



- Adoption of docker is being very slow in HPC centers
- Thus the typical situation is that docker is not installed and one cannot run containers without some support from the system software.
- In general Docker adoption will be slow in any computing farm or interactive linux system shared by many users.
  - It will take time for sysadmins to overcome the concerns of their security teams.
  - It is yet another service to maintain...
  - .... you name it.

# We have developed “udocker”



- **tool to execute content of docker containers in user space** when docker is not available
  - enables download of docker containers from dockerhub
  - enables execution of docker containers by non-privileged users
- **can be used to execute the content of docker containers in Linux batch systems and interactive clusters managed by others**
- **wrapper around other tools to mimic docker capabilities**
  - current version uses **proot** to provide a chroot like environment without privileges (runs on CentOS 6, CentOS 7, Fedora, Ubuntu)
- **More info and downloads at:**
  - <https://www.gitbook.com/book/indigo-dc/udocker/details>
  - [https://indigo-dc.gitbooks.io/udocker/content/doc/user\\_manual.html](https://indigo-dc.gitbooks.io/udocker/content/doc/user_manual.html)

# We have developed “udocker”



- Examples:

```
# download, but could also import or load a container exported/save by docker
```

```
$ udocker.py pull fedora:latest
```

```
$ udocker.py create --name=myfed fedora:latest
```

```
# install Firefox, and run it accessing /dev, for audio (feature not supported in docker!)
```

```
$ udocker.py run myfed /bin/bash yum install -y firefox
```

```
$ udocker.py run --bindhome --hostauth --hostenv -v /sys -v /proc \  
-v /var/run -v /dev --user=myusername --dri myfed firefox
```

**udocker is NOT an alternative to docker:** we need the container image built by docker.

**Is a tool to handle and run containers with regular user privileges** and/or when docker is not available for some reason: **it is very convenient to access clusters and Grid resources**

# We have developed “udocker”

## Basic description:



- Everything is stored in the `$HOME` or some other directory belonging to the user (tunable parameter).
- Container layers are download to the above specified directory
- Directory trees can be created/extracted from these container layers
- **proot** uses the debugger **ptrace mechanism** to change pathnames and execute transparently inside a directory tree
- No impact on read/write or execution, only impact on system calls using pathnames (ex. open, chdir, etc)
- Does not require installation of software in the host system:
  - *udocker* is a python script
  - *proot* is statically compiled

# Running Mastercode with udocker: Linux Cluster



1. Download udocker from: <https://github.com/indigo-dc/udocker>
2. Download the container: `$ ./udocker.py pull indigodatacloud/docker-mastercode`
3. Make sure you have enough space to uncompress it by pointing to a proper directory (default is \$HOME): `export UDOCKER_DIR=/MY_LARGE_FILESYSTEM/userabc/udocker`
4. Create the container directory tree on your user space  
`$ ./udocker.py create indigodatacloud/docker-mastercode  
bb889c79-2872-37f3-adad-cd9e937dd6f0`
5. You probably want to give it a nicer name:  
`$ ./udocker.py name bb889c79-2872-37f3-adad-cd9e937dd mymastercode`

```
userabc@headnode:~/udocker-master$ ls -lrt $UDOCKER_DIR
total 20
drwxrwxr-x 3 isabel isabel 4096 Sep 30  2015 lib
drwxrwxr-x 2 isabel isabel 4096 Sep 30  2015 bin
drwxrwxr-x 3 isabel isabel 4096 May 15 17:14 repos
drwxrwxr-x 2 isabel isabel 4096 May 15 17:16 layers
drwxrwxr-x 3 isabel isabel 4096 May 15 17:18 containers
```

# Running Mastercode with udocker: Linux Cluster: batch script



```
export MASTERDIR=/gpfs/csic_users/userabc/mastercode
export UDOCKER_DIR=$MASTERDIR/.udocker
```

```
../udocker-master/udocker.py run --hostauth \  
-v /home/csic/cdi/ica/mcpp-master \  
-v /home/csic/cdi/ica \  
-user=userabc \  
-w /home/csic/cdi/ica/mcpp-master mastercode \  
'/bin/bash -c "pwd; ./udocker-mastercode.sh"'
```

-hostauth : to use the /etc/passwd of the host machine

-v makes directories available inside the container

-user: your userid

-w working directory from where the commands will be issued

Where **udocker-mastercode.sh** is the (usual) command line sequency to execute Mastercode

```
./mc_point.py --run-mode mc-cmssm --predictors all --inputs 500 600 0 10 --print-mc-spectrum > output.txt
```

# Conclusions

- We have shown how it is now possible to automatize container building for a complex case of a Phenomenology code with the help of docker
- We have developed a tool to run Mastercode without docker, and without worrying about modifying the system software
  - On a desktop (Linux / MacOS)
  - On a Linux Cluster
  - On the Grid
- We are very confident that this will be a huge step towards the facilitation of the usage of Mastercode.

# Questions?

*“Using udocker we can run the container as an unprivileged user  
Without needing any additional system software”*

