3nd ASTERICS-OBELICS Workshop

23-25 October 2018, Cambridge, UK.

H2020-Astronomy ESFRI and Research Infrastructure Cluster
(Grant Agreement number: 653477).

Data and Software Preservation through Containerisation

Tamás Gál
3rd ASTERICS – OBELICS Workshop
23-26 October 2018, Postdoc Centre, Eddington

KM3NeT

ERLANGEN CENTRE
FOR ASTROPARTICLE
PHYSICS

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Overview

- About KM3NeT
- Motivation
- Containerisation
- Docker and Singularity
- Singularity in Production Pipelines
- Continuous Integration
- Data Provenance

About KKM3NeT

- Thousands of digital optical modules (DOMs), each holding 31 photomultiplier tubes will be deployed in the Mediterranean within the next few years to observe neutrino interactions
- In its final configuration it will comprise 6 building blocks with 115 strings each (string = vertical line with 18 DOMs), i.e. 12420 DOMs in total.
- The raw data rate of a single building block is roughly 25 Gb/s
- Letter of intent for KM3NeT 2.0 https://doi.org/10.1088/0954-3899/43/8/084001

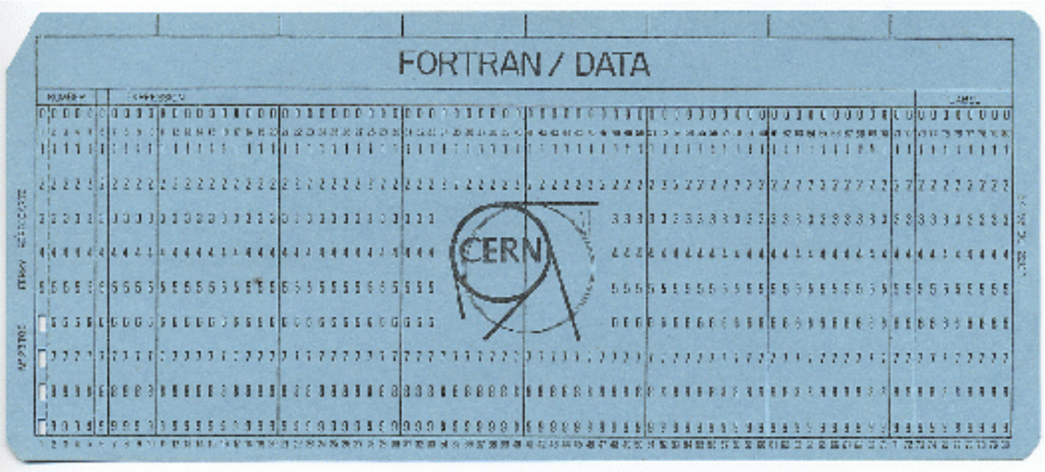
Computing Model of KM3NeT – Processing Chain

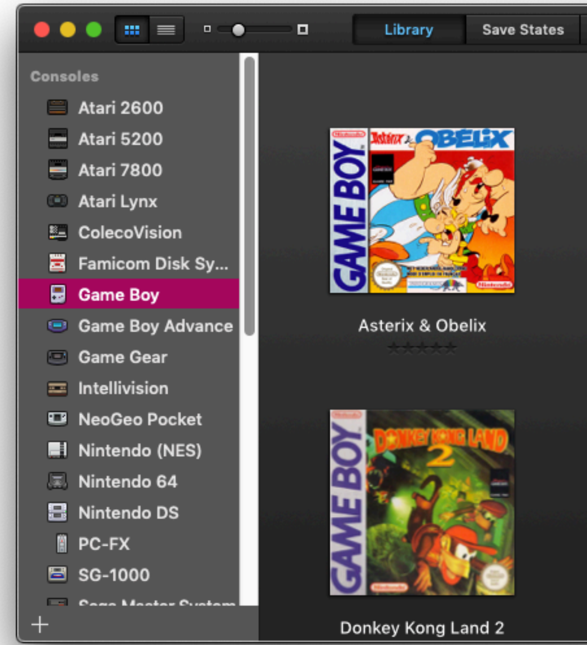
- "All data to shore" approach
- Events are triggered at detector site (Tier 0)
- Calibration and reconstruction on HPC centres (Tier 1)
- High level analysis on HPC, local computing clusters or desktops (Tier 2)

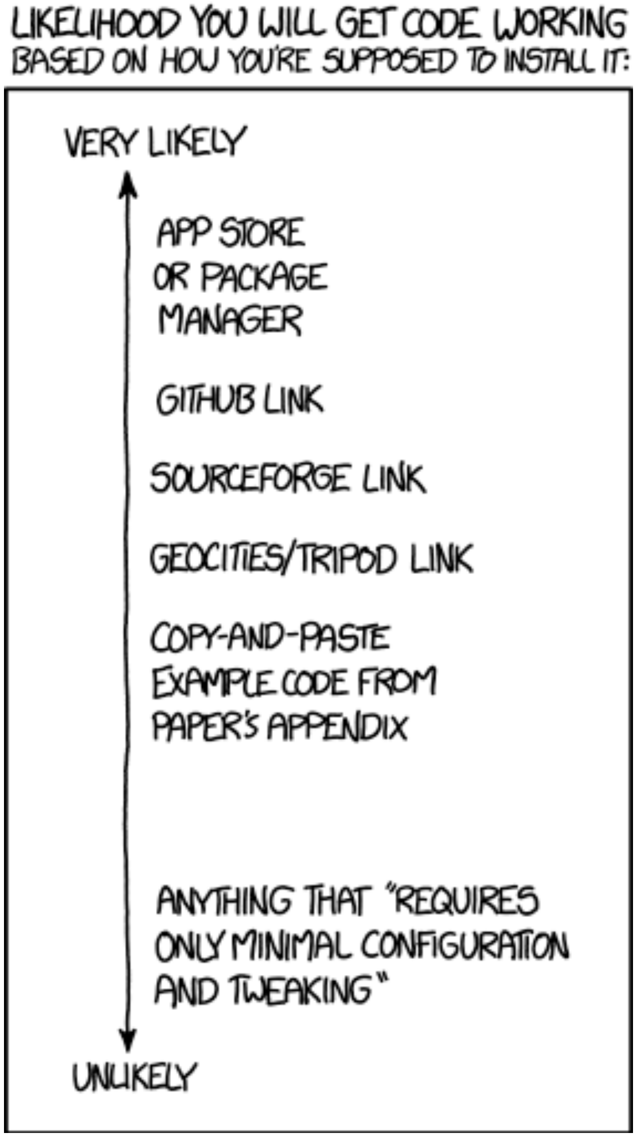
- There is a need for robust processing chains for these complex tasks

Motivation of Data and Software Preservation

- Reusing existing data and software
- Reproducing scientific results
- Being independent of the processing infrastructure (cloud computing, GRID, etc.)
- *Minimise the efforts to achieve these tasks*






LIKELIHOOD YOU WILL GET CODE WORKING BASED ON HOW YOU'RE SUPPOSED TO INSTALL IT:
VERY LIKELY
APP STORE OR PACKAGE MANAGER
GITHUB LINK
SOURCEFORGE LINK
GEOCITIES/TRIPOD LINK
COPY-AND-PASTE EXAMPLE CODE FROM PAPER'S APPENDIX
ANYTHING THAT "REQUIRES ONLY MINIMAL CONFIGURATION AND TWEAKING"
UNLIKELY

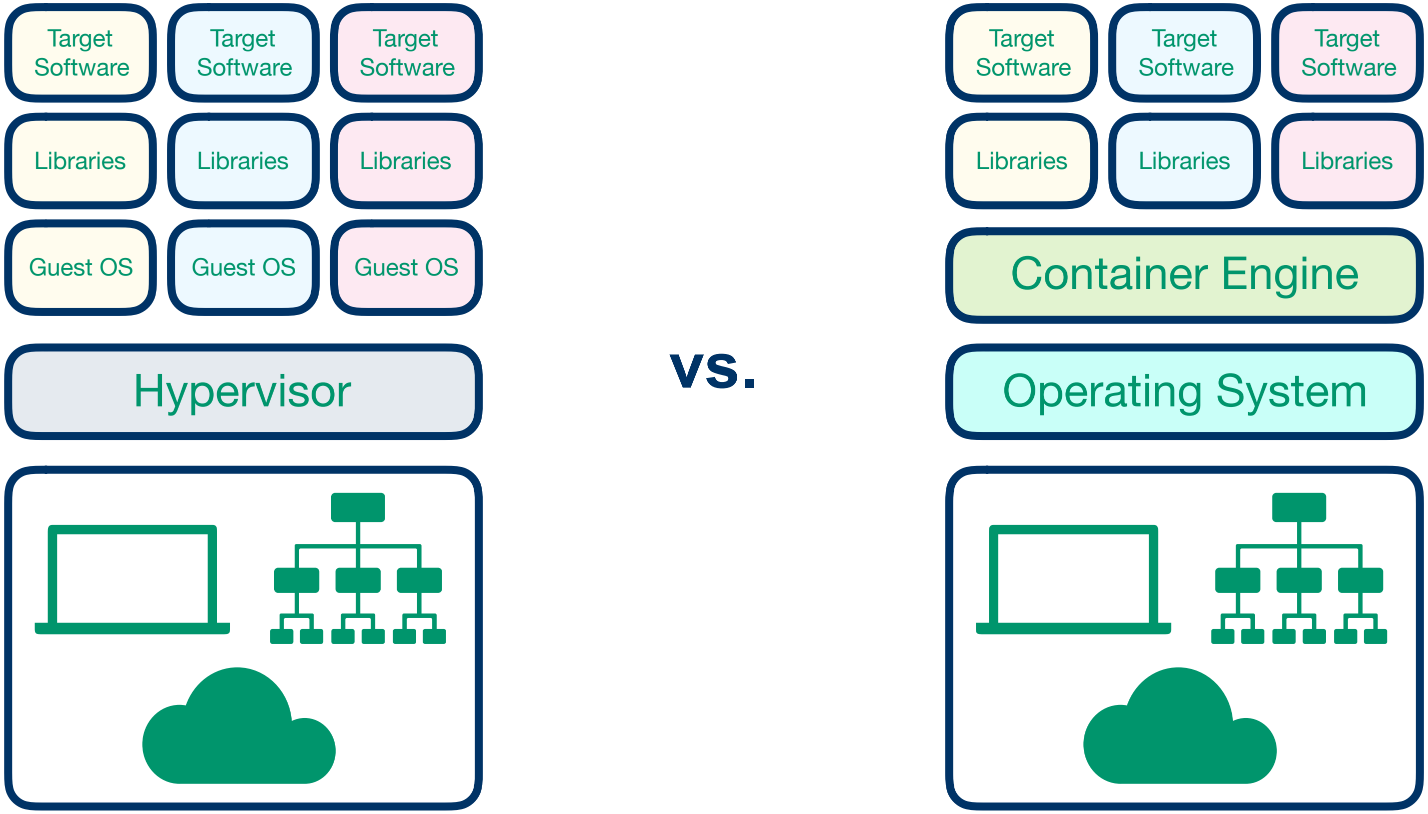
https://xkcd.com/1742

Containerisation

- Encapsulation of a system environment
- Primarily designed for micro-service virtualisation
- Enterprise solution for getting the most out of powerful servers
  - "lots of (idling) containers" vs. "bare metal solutions"
  - flexible ressource allocation and prioritisation

- For science, the use-case is different:
  - Maximum performance needed
  - No root access
  - One container to utilise all available host sources
  - No need for resource isolation

Virtual Machines vs. Containerisation



Target Software | Target Software | Target Software

Libraries | Libraries | Libraries

Guest OS | Guest OS | Guest OS

Hypervisor

vs.

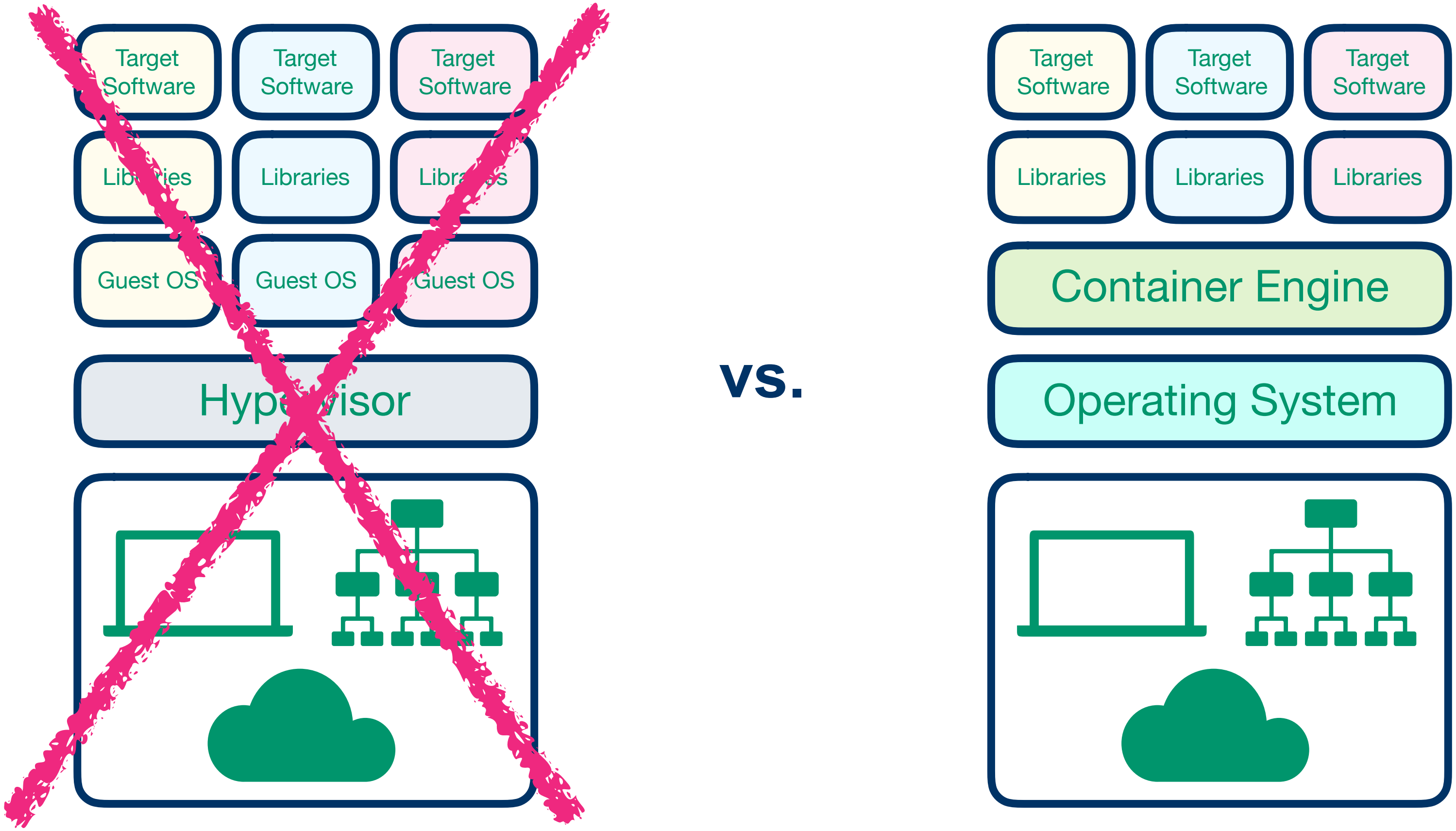
Target Software | Target Software | Target Software

Libraries | Libraries | Libraries

Container Engine

Operating System

Virtual Machines vs. Containerisation



~~Target Software~~ ~~Target Software~~ ~~Target Software~~

~~Libraries~~ ~~Libraries~~ ~~Libraries~~

~~Guest OS~~ ~~Guest OS~~ ~~Guest OS~~

~~Hypervisor~~

vs.

Target Software Target Software Target Software

Libraries Libraries Libraries

Container Engine

Operating System

Docker vs. Singularity (from the science perspective)

Docker

- Great (and designed) for micro services, development and continuous integration
- Images consist of "layers" which can be reused
- Easy sharing of images and layers (e.g. DockerHub)
- Containers may however not be fully reproducible due to complex layer dependencies (which may change over time)
- Very high level of encapsulation – feels like a Virtual Machine with native performance
- Docker is not meant and allowed (I haven't seen one so far) to run on HPC (e.g. it requires root permissions)
- https://docker.com

Docker vs. Singularity (from the science perspective)

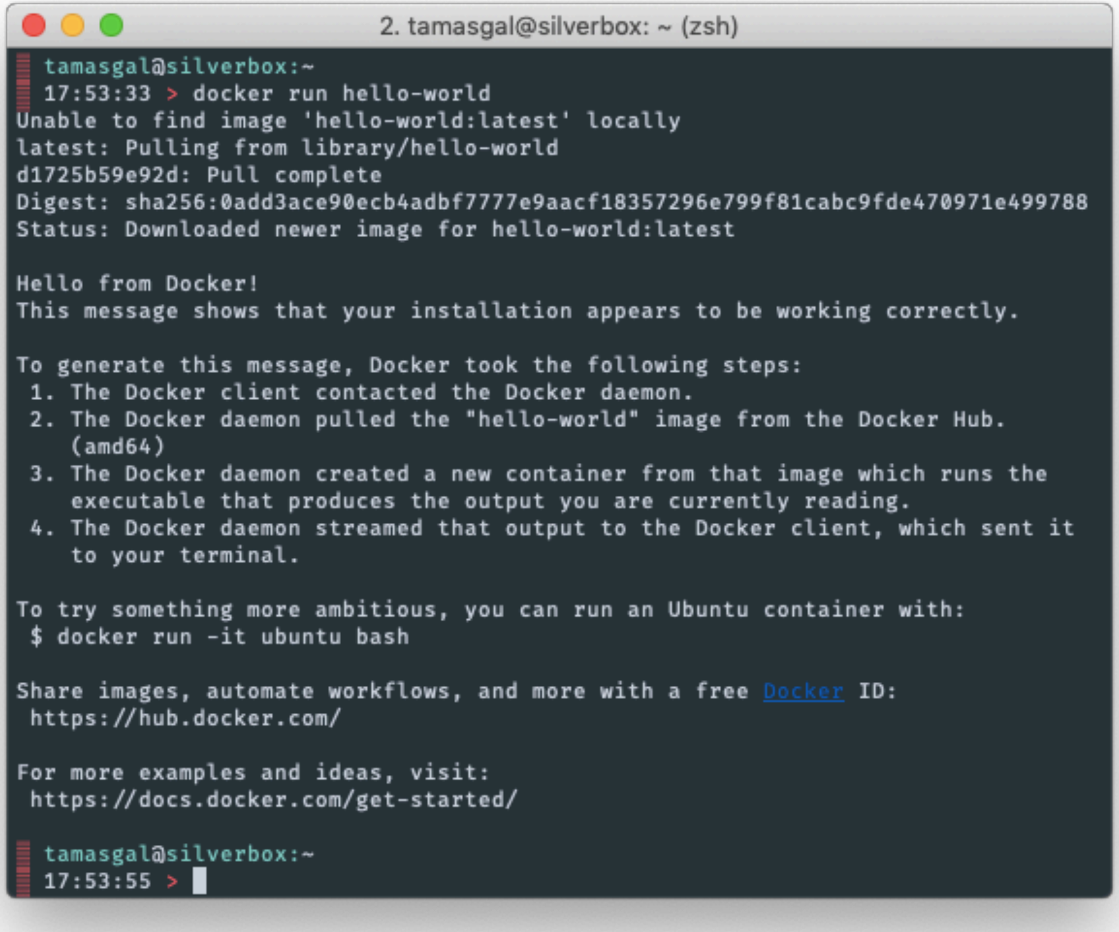
Singularity

- Created for and by HPC engineers, scientists and Linux developers
- No root-owned base container daemon needed, it runs under the initial user's privileges

  It even blocks privilege escalation out of the container via the kernel feature `pr_set_no_new_privs`
- /home, /dev, /sys and /proc are mounted from the host machine
- Direct usage of host resources (like network, filesystems, devices etc.)
- Safe for HPC! (and even runs MPI and can natively use GPUs!)
- Singularity is compatible to other container solutions (especially Docker)
- Single-file based images (just needs Singularity installed on the target)
- More performant than Docker (not designed around micro-service process isolation => minimum number of namespaces necessary)
- https://singularity.lbl.gov/

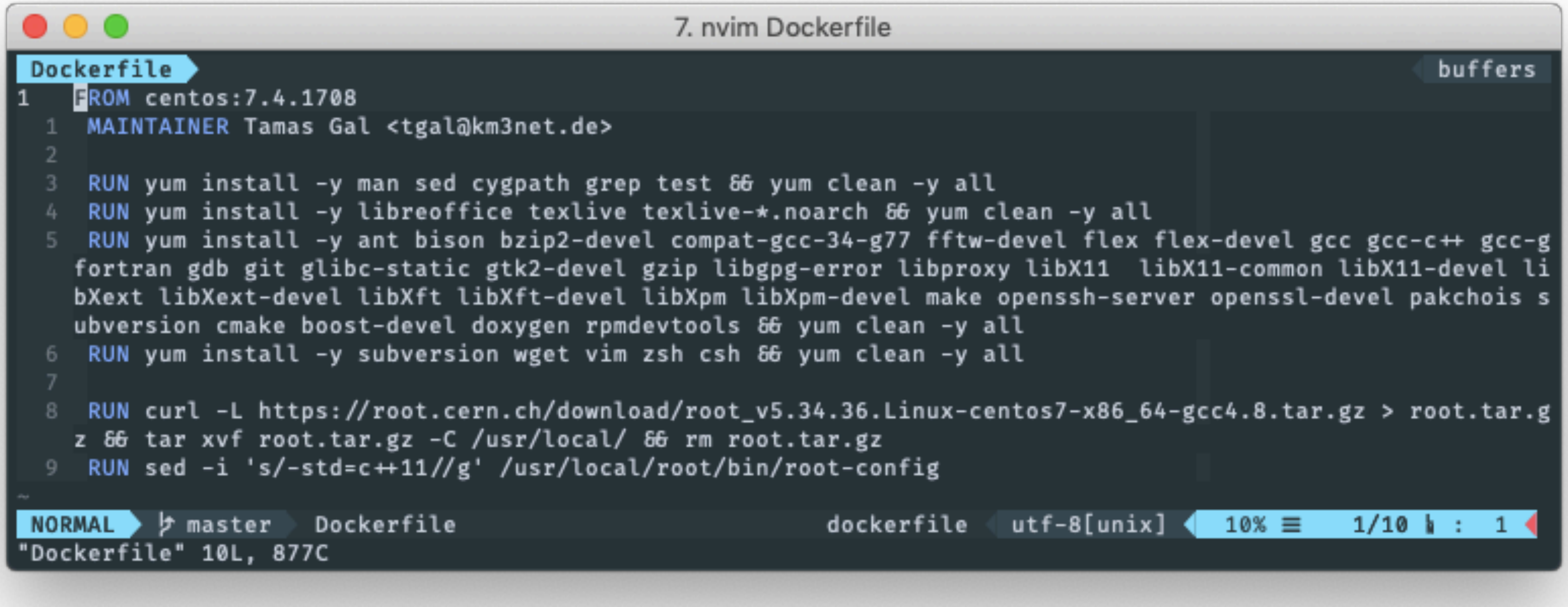
Docker

```
# Dockerfile
FROM scratch
COPY hello /
CMD ["/hello"]
```



```
tamasgal@silverbox:~
17:53:33 > docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
d1725b59e92d: Pull complete
Digest: sha256:0add3ace90ecb4adbf7777e9aacf18357296e799f81cabc9fde470971e499788
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

tamasgal@silverbox:~
17:53:55 > 
```

Docker – Dockerfile example (used in KM3NeT)



```
FROM centos:7.4.1708
MAINTAINER Tamas Gal <tgal@km3net.de>

RUN yum install -y man sed cygpath grep test && yum clean -y all
RUN yum install -y libreoffice texlive texlive-*.noarch && yum clean -y all
RUN yum install -y ant bison bzip2-devel compat-gcc-34-g77 fftw-devel flex flex-devel gcc gcc-c++ gcc-gfortran gdb git glibc-static gtk2-devel gzip libgpg-error libproxy libX11  libX11-common libX11-devel libXext libXext-devel libXft libXft-devel libXpm libXpm-devel make openssh-server openssl-devel pakchois subversion cmake boost-devel doxygen rpmdevtools && yum clean -y all
RUN yum install -y subversion wget vim zsh csh && yum clean -y all

RUN curl -L https://root.cern.ch/download/root_v5.34.36.Linux-centos7-x86_64-gcc4.8.tar.gz > root.tar.gz && tar xvf root.tar.gz -C /usr/local/ && rm root.tar.gz
RUN sed -i 's/-std=c++11//g' /usr/local/root/bin/root-config
```

```
"Dockerfile" 10L, 877C
```

Singularity – Usage Example (using an ANTARES image)

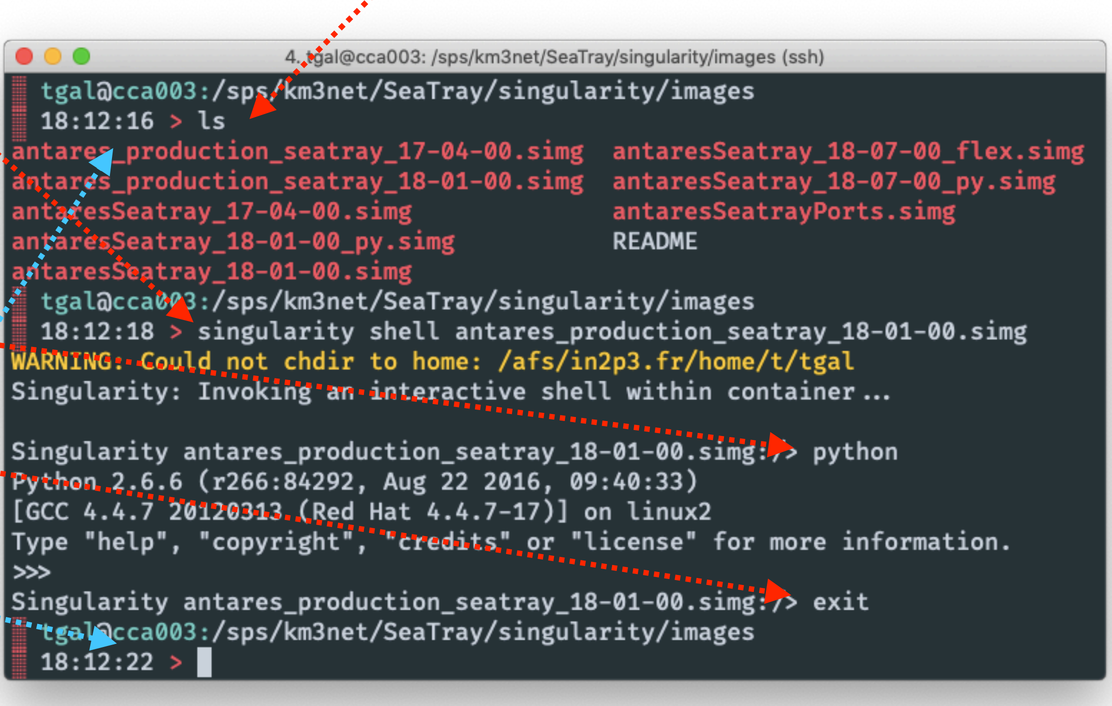
available images (prebuilt)

launch a shell in a container from an image

run e.g. python

exit...

all in ~6 seconds
(I cannot type faster ;)



4. tgal@cca003: /sps/km3net/SeaTray/singularity/images (ssh)

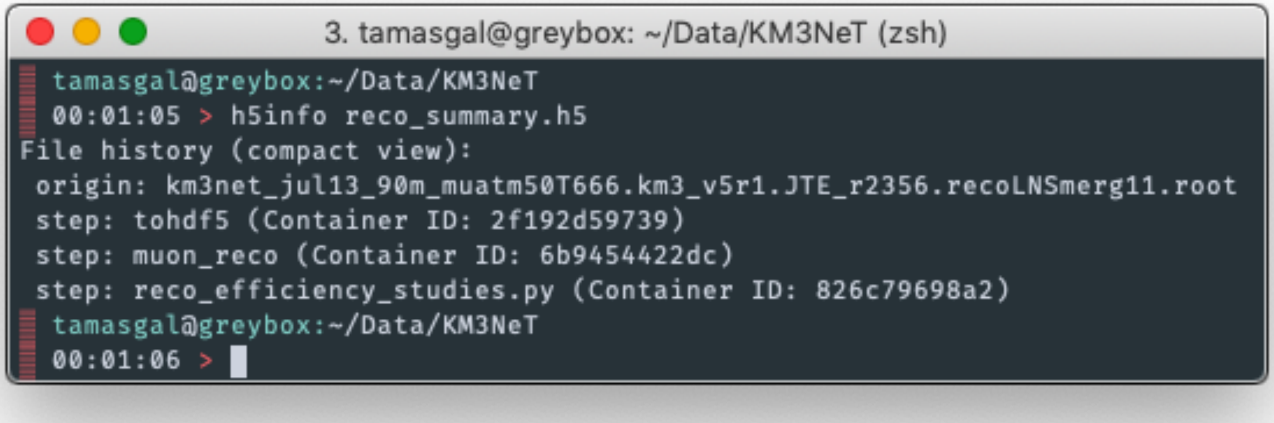
```
 tgal@cca003:/sps/km3net/SeaTray/singularity/images
 18:12:16 > ls
antares_production_seatray_17-04-00.simg  antaresSeatray_18-07-00_flex.simg
antares_production_seatray_18-01-00.simg  antaresSeatray_18-07-00_py.simg
antaresSeatray_17-04-00.simg              antaresSeatrayPorts.simg
antaresSeatray_18-01-00_py.simg           README
antaresSeatray_18-01-00.simg
 tgal@cca003:/sps/km3net/SeaTray/singularity/images
 18:12:18 > singularity shell antares_production_seatray_18-01-00.simg
WARNING: Could not chdir to home: /afs/in2p3.fr/home/t/tgal
Singularity: Invoking an interactive shell within container...

Singularity antares_production_seatray_18-01-00.simg:~> python
Python 2.6.6 (r266:84292, Aug 22 2016, 09:40:33)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-17)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
Singularity antares_production_seatray_18-01-00.simg:~> exit
 tgal@cca003:/sps/km3net/SeaTray/singularity/images
 18:12:22 >
```

Singularity Images in Production Pipelines

- Seamless integration into the file system of the host
  - Processes running within the container take data from the host as input and write their output to the host
  - Runs like native software
  - No "source env.sh nightmare"
  - Transparent user privileges
- Completely independent of the host system (as long as Singularity is installed)
- OS upgrades on HPC are no-brainer
- Perfect fit for Grid computing with heterogeneous systems

Data Provenance

- File history is part of the data preservation
- Provides historical records of the data and its origins, like
  - container IDs (e.g. Singularity Hub)
  - or specific software and library versions
  - and additional parameters
    (e.g. command line arguments or configuration files)
- Data and Software stored together in a single image file

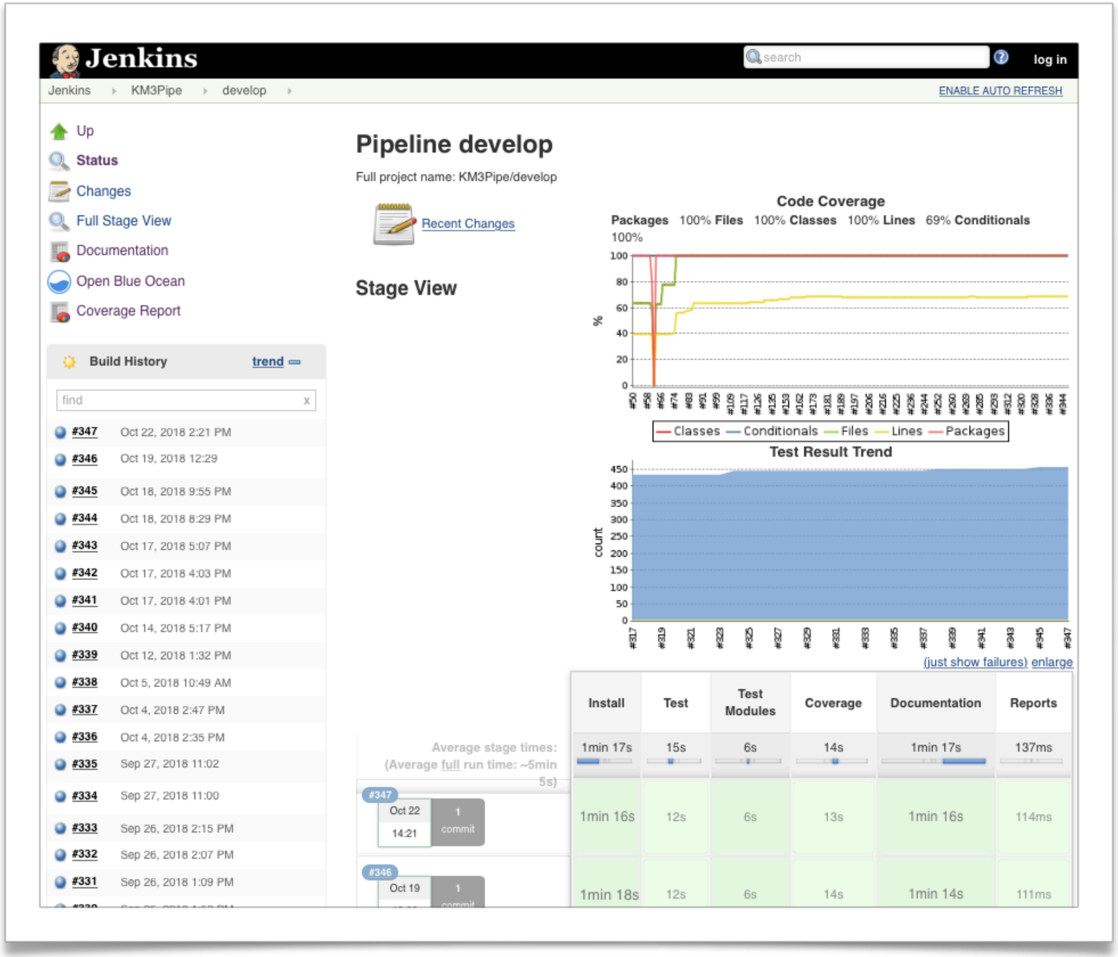


3. tamasgal@greybox: ~/Data/KM3NeT (zsh)

```
 tamasgal@greybox:~/Data/KM3NeT
 00:01:05 > h5info reco_summary.h5
File history (compact view):
 origin: km3net_jul13_90m_muatm50T666.km3_v5r1.JTE_r2356.recoLNSmerg11.root
 step: tohdf5 (Container ID: 2f192d59739)
 step: muon_reco (Container ID: 6b9454422dc)
 step: reco_efficiency_studies.py (Container ID: 826c79698a2)
 tamasgal@greybox:~/Data/KM3NeT
 00:01:06 > 
```

Continuous Integration using Jenkins and Docker (1/2)

- Containerisation as part of the software development process
- Software projects utilise Jenkins to automatise steps for build, test, documentation etc.
- Docker is used to give developers full access to the container creation process



Jenkins — search — log in

Jenkins › KM3Pipe › develop › — ENABLE AUTO REFRESH

Up
Status
Changes
Full Stage View
Documentation
Open Blue Ocean
Coverage Report

Build History — trend

find

#347	Oct 22, 2018 2:21 PM
#346	Oct 19, 2018 12:29
#345	Oct 18, 2018 9:55 PM
#344	Oct 18, 2018 8:29 PM
#343	Oct 17, 2018 5:07 PM
#342	Oct 17, 2018 4:03 PM
#341	Oct 17, 2018 4:01 PM
#340	Oct 14, 2018 5:17 PM
#339	Oct 12, 2018 1:32 PM
#338	Oct 5, 2018 10:49 AM
#337	Oct 4, 2018 2:47 PM
#336	Oct 4, 2018 2:35 PM
#335	Sep 27, 2018 11:02
#334	Sep 27, 2018 11:00
#333	Sep 26, 2018 2:15 PM
#332	Sep 26, 2018 2:07 PM
#331	Sep 26, 2018 1:09 PM

Pipeline develop

Full project name: KM3Pipe/develop

Recent Changes

Stage View

Code Coverage

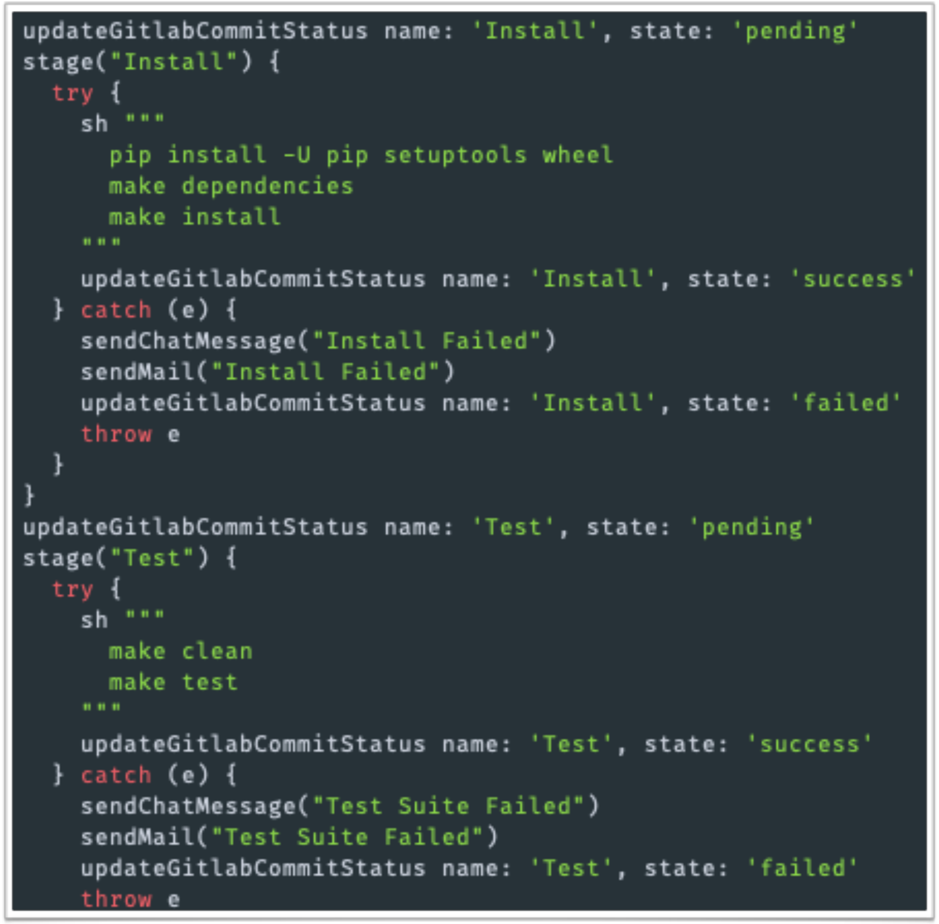
Packages 100% Files 100% Classes 100% Lines 69% Conditionals

(just show failures) enlarge

	Install	Test	Test Modules	Coverage	Documentation	Reports
Average stage times: (Average full run time: ~5min 5s)	1min 17s	15s	6s	14s	1min 17s	137ms
#347 Oct 22 14:21 1 commit	1min 16s	12s	6s	13s	1min 16s	114ms
#346 Oct 19 1 commit	1min 18s	12s	6s	14s	1min 14s	111ms

Continuous Integration using Jenkins and Docker (2/2)

- The Jenkinsfile contains a full recipe for each stage (building, testing, generating documentation etc.)
- Everything is executed in a specific (isolated) Docker container
- The same environment can be (re)used by Singularity to build the single-file image



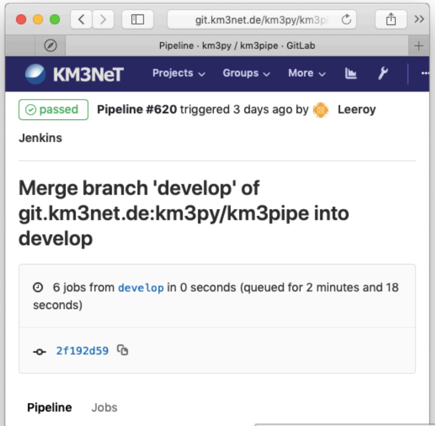
```
updateGitlabCommitStatus name: 'Install', state: 'pending'
stage("Install") {
  try {
    sh """
      pip install -U pip setuptools wheel
      make dependencies
      make install
    """
    updateGitlabCommitStatus name: 'Install', state: 'success'
  } catch (e) {
    sendChatMessage("Install Failed")
    sendMail("Install Failed")
    updateGitlabCommitStatus name: 'Install', state: 'failed'
    throw e
  }
}
updateGitlabCommitStatus name: 'Test', state: 'pending'
stage("Test") {
  try {
    sh """
      make clean
      make test
    """
    updateGitlabCommitStatus name: 'Test', state: 'success'
  } catch (e) {
    sendChatMessage("Test Suite Failed")
    sendMail("Test Suite Failed")
    updateGitlabCommitStatus name: 'Test', state: 'failed'
    throw e
```

KM3NeT

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

Continuous Integration using Jenkins, Docker and GitLab

- Commits to the GitLab repository trigger Jenkins pipelines
- Build procedure and test suite are run and the documentation is updated continuously
- Each tagged release triggers a Docker (or Singularity) image creation

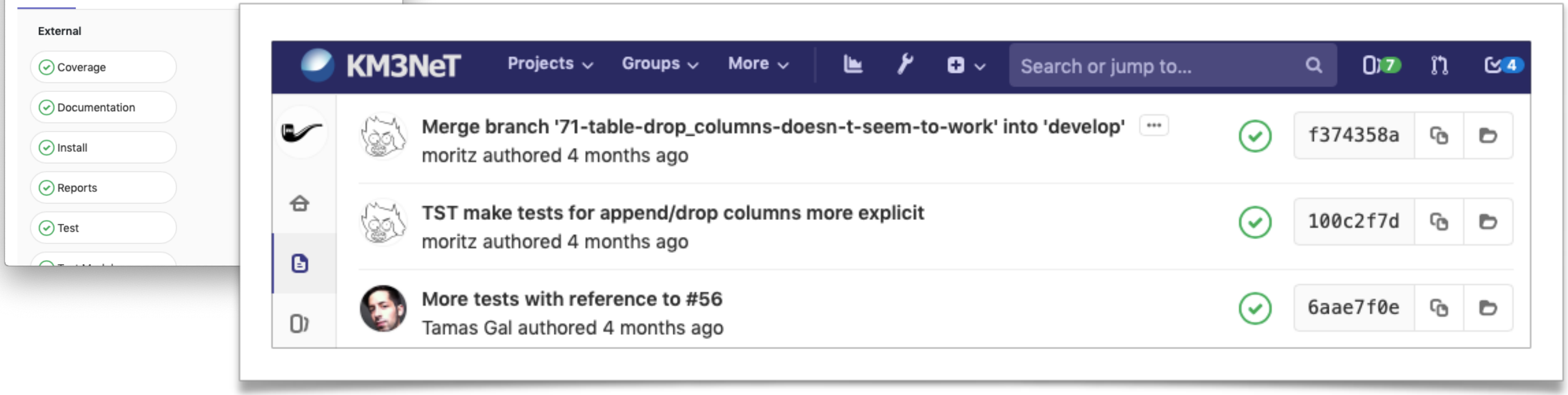


✓ passed **Pipeline #620** triggered 3 days ago by **Leeroy Jenkins**

Merge branch 'develop' of git.km3net.de:km3py/km3pipe into develop

6 jobs from develop in 0 seconds (queued for 2 minutes and 18 seconds)

2f192d59



✓ Coverage

✓ Documentation

✓ Install

✓ Reports

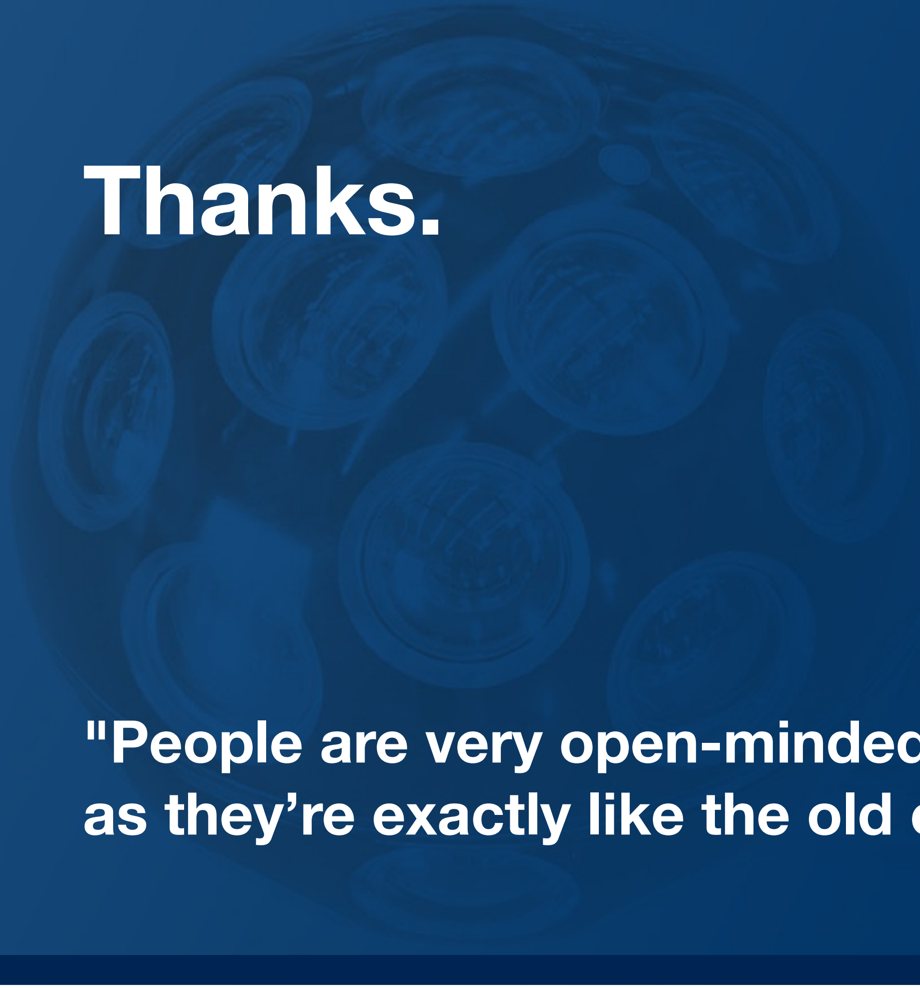
✓ Test

Commit	Status	SHA
Merge branch '71-table-drop_columns-doesn-t-seem-to-work' into 'develop' ··· moritz authored 4 months ago	✓	f374358a
TST make tests for append/drop columns more explicit moritz authored 4 months ago	✓	100c2f7d
More tests with reference to #56 Tamas Gal authored 4 months ago	✓	6aae7f0e

Summary

- First Aim: Data and Software Preservation
  - Selection of appropriate method: containerisation with Singularity (and Docker)
  - Implementation of use case for KM3NeT
- Derivation I: Preservation of full processing workflow
  - Implementation of ANTARES code to run at HPC
  - Implementation for KM3NeT: WIP
- Derivation II: Preservation & Optimisation of Development and Analysis
  - Containerisation as part of software development and integration
  - Reproducibility of results through containerised analysis chain

- ...fully implemented in KM3NeT



Thanks.

"People are very open-minded about new things — as long as they’re exactly like the old ones." - Charles F. Kettering

KM3NeT

ERLANGEN CENTRE
FOR ASTROPARTICLE
PHYSICS

FAU
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Acknowledgement

- H2020-Astronomy ESFRI and Research Infrastructure Cluster (Grant Agreement number: 653477).