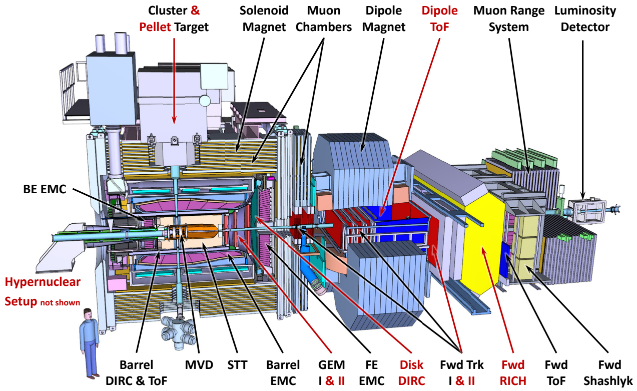
Deployment for the PANDA Detector Control System

EPICS Collaboration Meeting, 09/19/2022

Florian Feldbauer

Ruhr-Universität Bochum - Experimentalphysik I AG

The $\overline{\text{P}}$ANDA Detector



$\overline{\text{P}}$ANDA physics program:

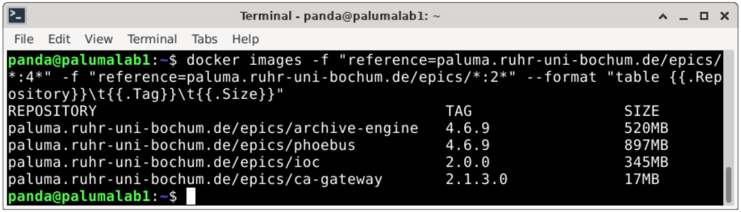
- Hadron spectroscopy
- Hadron structure
- Hadrons in medium
- Hypernuclear physics

Challenges for the Detector Control System

- Still in development and prototyping phase
- Detectors are developed all over the world
- Each subsystem should develop their own DCS partition
- Large diversity in used operating system at the different sites
- Large diversity in skills of “DCS-experts”

⇒ Control system should be easy to deploy for everyone

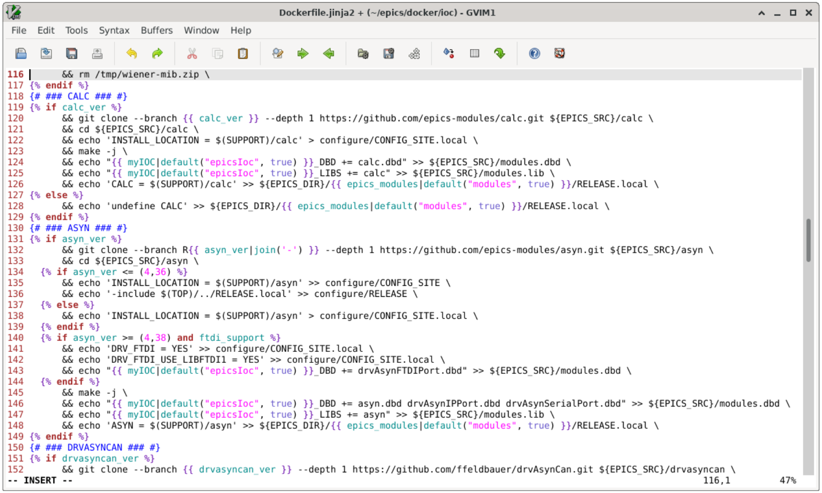
⇒ Container Virtualization (docker)


Terminal - panda@palumalab1: ~
File Edit View Terminal Tabs Help

```
panda@palumalab1:~$ docker images -f "reference=paluma.ruhr-uni-bochum.de/epics/
*:4*" -f "reference=paluma.ruhr-uni-bochum.de/epics/*:2*" --format "table {{.Rep
ository}}\t{{.Tag}}\t{{.Size}}"
REPOSITORY                                         TAG                 SIZE
paluma.ruhr-uni-bochum.de/epics/archive-engine     4.6.9               520MB
paluma.ruhr-uni-bochum.de/epics/phoebus            4.6.9               897MB
paluma.ruhr-uni-bochum.de/epics/ioc                2.0.0               345MB
paluma.ruhr-uni-bochum.de/epics/ca-gateway         2.1.3.0             17MB
panda@palumalab1:~$ 
```

Version Management

Dockerfiles written as Jinja2 templates



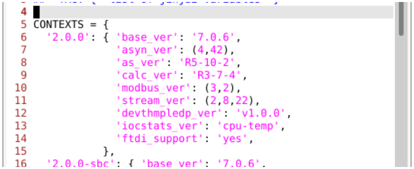
Dockerfile.jinja2 + (~/epics/docker/ioc) - GVIM1
File Edit Tools Syntax Buffers Window Help

```
      && rm /tmp/wiener-mib.zip \
{% endif %}
{# ### CALC ### #}
{% if calc_ver %}
      && git clone --branch {{ calc_ver }} --depth 1 https://github.com/epics-modules/calc.git ${EPICS_SRC}/calc \
      && cd ${EPICS_SRC}/calc \
      && echo 'INSTALL_LOCATION = $(SUPPORT)/calc' > configure/CONFIG_SITE.local \
      && make -j \
      && echo "{{ myIOC|default("epicsIoc", true) }}_DBD += calc.dbd" >> ${EPICS_SRC}/modules.dbd \
      && echo "{{ myIOC|default("epicsIoc", true) }}_LIBS += calc" >> ${EPICS_SRC}/modules.lib \
      && echo 'CALC = $(SUPPORT)/calc' >> ${EPICS_DIR}/{{ epics_modules|default("modules", true) }}/RELEASE.local \
{% else %}
      && echo 'undefine CALC' >> ${EPICS_DIR}/{{ epics_modules|default("modules", true) }}/RELEASE.local \
{% endif %}
{# ### ASYN ### #}
{% if asyn_ver %}
      && git clone --branch R{{ asyn_ver|join('-') }} --depth 1 https://github.com/epics-modules/asyn.git ${EPICS_SRC}/asyn \
      && cd ${EPICS_SRC}/asyn \
  {% if asyn_ver <= (4,36) %}
      && echo 'INSTALL_LOCATION = $(SUPPORT)/asyn' >> configure/CONFIG_SITE \
      && echo '-include $(TOP)/../RELEASE.local' >> configure/RELEASE \
  {% else %}
      && echo 'INSTALL_LOCATION = $(SUPPORT)/asyn' > configure/CONFIG_SITE.local \
  {% endif %}
  {% if asyn_ver >= (4,38) and ftdi_support %}
      && echo 'DRV_FTDI = YES' >> configure/CONFIG_SITE.local \
      && echo 'DRV_FTDI_USE_LIBFTDI1 = YES' >> configure/CONFIG_SITE.local \
      && echo "{{ myIOC|default("epicsIoc", true) }}_DBD += drvAsynFTDIPort.dbd" >> ${EPICS_SRC}/modules.dbd \
  {% endif %}
      && make -j \
      && echo "{{ myIOC|default("epicsIoc", true) }}_DBD += asyn.dbd drvAsynIPPort.dbd drvAsynSerialPort.dbd" >> ${EPICS_SRC}/modules.dbd \
      && echo "{{ myIOC|default("epicsIoc", true) }}_LIBS += asyn" >> ${EPICS_SRC}/modules.lib \
      && echo 'ASYN = $(SUPPORT)/asyn' >> ${EPICS_DIR}/{{ epics_modules|default("modules", true) }}/RELEASE.local \
{% endif %}
{# ### DRVASYNCAN ### #}
{% if drvasyncan_ver %}
      && git clone --branch {{ drvasyncan_ver }} --depth 1 https://github.com/ffeldbauer/drvAsynCan.git ${EPICS_SRC}/drvasyncan \
-- INSERT --                                                                                    116,1          47%
```

RUHR UNIVERSITÄT BOCHUM

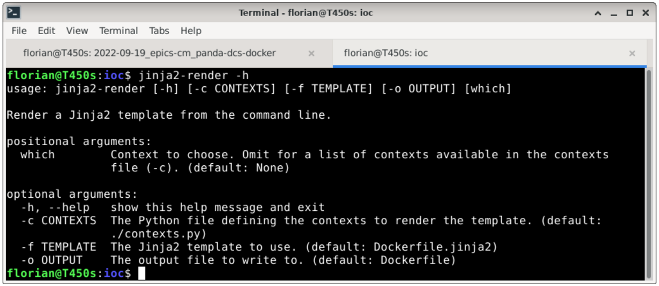
Version Management

Module versions set in context file



```
CONTEXTS = {
  '2.0.0': { 'base_ver': '7.0.6',
             'asyn_ver': (4,42),
             'as_ver': 'R5-10-2',
             'calc_ver': 'R3-7-4',
             'modbus_ver': (3,2),
             'stream_ver': (2,8,22),
             'devthmpledp_ver': 'v1.0.0',
             'iocstats_ver': 'cpu-temp',
             'ftdi_support': 'yes',
           },
  '2.0.0-sbc': { 'base_ver': '7.0.6',
```

Create Dockerfile from template with jinja2-render:

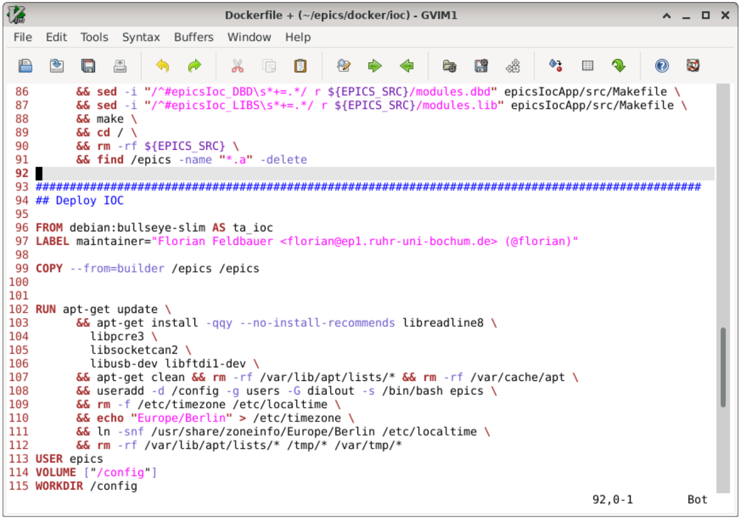


Terminal - florian@T450s: ioc
File Edit View Terminal Tabs Help
florian@T450s: 2022-09-19_epics-cm_panda-dcs-docker
florian@T450s: ioc

```
florian@T450s:ioc$ jinja2-render -h
usage: jinja2-render [-h] [-c CONTEXTS] [-f TEMPLATE] [-o OUTPUT] [which]

Render a Jinja2 template from the command line.

positional arguments:
  which        Context to choose. Omit for a list of contexts available in the contexts
               file (-c). (default: None)

optional arguments:
  -h, --help   show this help message and exit
  -c CONTEXTS  The Python file defining the contexts to render the template. (default:
               ./contexts.py)
  -f TEMPLATE  The Jinja2 template to use. (default: Dockerfile.jinja2)
  -o OUTPUT    The output file to write to. (default: Dockerfile)
florian@T450s:ioc$ 
```

Decrease Image Size

Using multistage builds to reduce size of final images


Dockerfile + (~/epics/docker/ioc) - GVIM1
File Edit Tools Syntax Buffers Window Help

```
      && sed -i "/^#epicsIoc_DBD\s*+=.*/ r ${EPICS_SRC}/modules.dbd" epicsIocApp/src/Makefile \
      && sed -i "/^#epicsIoc_LIBS\s*+=.*/ r ${EPICS_SRC}/modules.lib" epicsIocApp/src/Makefile \
      && make \
      && cd / \
      && rm -rf ${EPICS_SRC} \
      && find /epics -name "*.a" -delete

###############################################################################################
## Deploy IOC

FROM debian:bullseye-slim AS ta_ioc
LABEL maintainer="Florian Feldbauer <florian@ep1.ruhr-uni-bochum.de> (@florian)"

COPY --from=builder /epics /epics


RUN apt-get update \
      && apt-get install -qqy --no-install-recommends libreadline8 \
        libpcre3 \
        libsocketcan2 \
        libusb-dev libftdi1-dev \
      && apt-get clean && rm -rf /var/lib/apt/lists/* && rm -rf /var/cache/apt \
      && useradd -d /config -g users -G dialout -s /bin/bash epics \
      && rm -f /etc/timezone /etc/localtime \
      && echo "Europe/Berlin" > /etc/timezone \
      && ln -snf /usr/share/zoneinfo/Europe/Berlin /etc/localtime \
      && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
USER epics
VOLUME ["/config"]
WORKDIR /config
                                                                    92,0-1        Bot
```

Pros and Cons

Pros:

- Easy to create images for multiple architectures

```
docker buildx build \
  --platform=linux/amd64,linux/arm64,linux/arm/v7 ...
```

- Easy to deploy at remote labs

```
docker pull paluma.ruhr-uni-bochum.de/epics/ioc
```

- Easy to setup EPICS as a service

```
docker run -dit --restart always ...
```

Cons:

- Easy to gain root access on host system
  Workarounds available, e.g. Singularity, pottman
- "Pinches" holes into firewall
  Docker has (and needs) access to iptables
- Need a lot of flags for container creation to access hardware

```
--device <DEV>, --group-add <GROUP>
```