

# Experiencing ecFlow - Docker

It is possible to experience ecFlow in a [Docker](#) container.

Here is an example of a Dockerfile to generate an image. The image size is large, and it can be used to test, evaluate and learn the ecFlow in a protected environment (behind a firewall).

## ecFlow Dockerfile

```
FROM debian:jessie
MAINTAINER map

RUN apt-get -y update && \
    apt-get -y upgrade && \
    apt-get update && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y locales && \
    sed -i -e 's/# en_US.UTF-8 UTF-8/en_US.UTF-8 UTF-8/' /etc/locale.gen && \
    dpkg-reconfigure --frontend=noninteractive locales && \
    update-locale LANG=en_US.UTF-8 && \
    apt-get install -y build-essential cmake python-dev qtbase5-dev \
    libmotif-dev libx11-dev libxext-dev libxpm-dev vim fvwm libxt-dev \
    xvfb wget && \
    apt-get install -qqy x11-apps

WORKDIR /tmp

# Environment flags used for compilation (they can be removed after we build)
ENV WK=/tmp/ecflow_build/ecFlow-4.7.0-Source \
    BOOST_ROOT=/tmp/ecflow_build/boost_1_53_0 \
    HTTP=https://confluence.ecmwf.int/download/attachments/8650755 \
    TE=ecFlow-4.7.0-Source.tar.gz \
    TB=boost_1_53_0.tar.gz \
    COMPILE=1

COPY fix_regex.sh /tmp/

RUN mkdir -p ${WK}/build && \
    cd /tmp/ecflow_build && \
    wget --output-document=${TE} ${HTTP}/${TE}?api=v2 && \
    wget --output-document=${TB} ${HTTP}/${TB}?api=v2 && \
    tar -zxvf ${TE} && \
    tar -zxvf ${TB}

RUN test ${COMPILE} -eq 1 && /tmp/fix_regex.sh && \
    cd ${BOOST_ROOT} && ./bootstrap.sh && \
    ${WK}/build_scripts/boost_1_53_fix.sh && \
    ${WK}/build_scripts/boost_build.sh

RUN cd ${WK}/build && cmake .. -DENABLE_GUI=ON -DENABLE_UI=ON && \
    make -j2 && make install # && make test && cd /tmp && rm -rf *

# environment variables for ecFlow server
ENV ECFLOW_USER=ecflow \
    ECF_PORT=2500 \
    ECF_HOME=/home/ecflow \
    HOME=/home/ecflow \
    HOST=ecflow \
    LANG=en_US.UTF-8 \
    PYTHONPATH=/usr/local/lib/python2.7/site-packages

EXPOSE ${ECF_PORT}

RUN groupadd --system ${ECFLOW_USER} && \
    useradd --create-home --system --gid ${ECFLOW_USER} ${ECFLOW_USER} && \
    chown ecflow /home/ecflow && chgrp ecflow /home/ecflow

USER ecflow
WORKDIR /home/ecflow
ENV DISPLAY=:0
```

It is possible to run the GUI `ecflow_ui` from there.

```
XSOCK=/tmp/.X11-unix
XAUTH=/tmp/.docker.xauth
xauth nlist :0 | sed -e 's/^.../ffff/' | xauth -f $XAUTH nmerge -
VARS="-v $XSOCK:$XSOCK -v $XAUTH:$XAUTH -e XAUTHORITY=$XAUTH"
# xhost + # might be needed
docker run -it -rm $VARS ecflow ecflow_ui
```



notebook.tgz

A [Jupyter](#) notebook is present in the following tar file as an alternative way to learn ecFlow.

[notebook](#)

Docker images are also available

- [alpine](#)
- [debian](#)