

- Configuration
 - Docker Image
 - Centos 6
 - Centos 7
 - Centos 8
 - Docker Compose
 - Docker Compose .env
 - Makefile
 - Example directory tree
- Execution

Docker Image

Centos 6

centos6/Dockerfile

```
FROM centos:centos6.10

ARG USER_NAME
ARG GROUP_NAME
ARG USER_ID
ARG GROUP_ID

#General packages
RUN yum -y install vim redhat-lsb-core sudo dbus-x11

#Install Git and Git LFS
RUN yum -y install epel-release
RUN yum -y install https://download1.rpmfusion.org/free/el/rpmfusion-free-release-6.noarch.rpm
RUN yum -y install git
RUN curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.rpm.sh | bash
RUN yum -y install git-lfs

#Install Java 8
RUN yum -y install java-1.8.0-openjdk-devel
#RUN echo 1| alternatives --config java_sdk_openjdk

#Install Java 11
#RUN yum -y install java-11-openjdk-devel
#RUN echo 2| alternatives --config java_sdk_openjdk

#Prepare for Java
RUN mkdir -p /usr/java
RUN ln -s /usr/lib/jvm/java-openjdk /usr/java/default

#Install ACS required packages
RUN yum -y install ksh gcc gcc-c++ gcc-gfortran libX11-devel libffi-devel perl readline-devel bzip2 bzip2-devel
openssl-devel openldap-devel libxml2-devel freetype-devel libxslt-devel sqlite-devel expat-devel tcl-devel tk-
devel bison flex autoconf unzip dos2unix

#ARCHIVE
RUN yum -y install libtool libdb-devel

#ICD
RUN yum -y install cmake blas-devel cfitsio-devel wcslib-devel lapack-devel

#CONTROL
RUN yum -y install binutils-devel libmemcached-devel

#CORRCommon
RUN yum -y install kernel kernel-devel procmail

#TELCAL
RUN yum -y install motif-devel texlive

#RTI DDS
#RUN cd / && curl http://****:****@webdav.sco.alma.cl/restricted/proprietary/RTI-RHEL6-x86_64.tar.bz2 | tar -xj

#Debugging Tools
RUN yum -y install gdb net-tools

#User configuration
RUN groupadd -r -g ${GROUP_ID} ${GROUP_NAME}
RUN useradd -r -m -u ${USER_ID} -g ${GROUP_NAME} ${USER_NAME}
RUN groupadd sudo
RUN usermod -aG sudo ${USER_NAME}
RUN echo new2me |passwd ${USER_NAME} --stdin
RUN echo "%sudo          ALL=(ALL)          ALL" >> /etc/sudoers
RUN echo "%sudo          ALL=(ALL)          NOPASSWD: ALL" >> /etc/sudoers
USER ${USER_NAME}
```

Centos 7

centos7/Dockerfile

```
FROM centos:centos7.7.1908

ARG USER_NAME
ARG GROUP_NAME
ARG USER_ID
ARG GROUP_ID

#General packages
RUN yum -y install vim redhat-lsb-core sudo dbus-x11

#Install Git and Git LFS
RUN yum -y install epel-release
RUN yum -y install https://download1.rpmfusion.org/free/el/rpmfusion-free-release-7.noarch.rpm
RUN yum -y install git
RUN curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.rpm.sh | bash
RUN yum -y install git-lfs

#Install Java 11
RUN yum -y install java-11-openjdk-devel

#Prepare for Java
RUN mkdir -p /usr/java
RUN ln -s /usr/lib/jvm/java-openjdk /usr/java/default

#Install ACS required packages
RUN yum -y install ksh gcc gcc-c++ gcc-gfortran libX11-devel libffi-devel perl readline-devel bzip2 bzip2-devel
openssl-devel openldap-devel libxml2-devel freetype-devel libxslt-devel sqlite-devel expat-devel tcl-devel tk-
devel bison flex autoconf unzip dos2unix

#ARCHIVE
RUN yum -y install libtool libdb-devel

#ICD
RUN yum -y install cmake blas-devel cfitsio-devel wcslib-devel lapack-devel

#CONTROL
RUN yum -y install binutils-devel libmemcached-devel

#CORRCommon
RUN yum -y install kernel kernel-devel procmail

#TELCAL
RUN yum -y install motif-devel texlive

#RTI DDS
#RUN cd / && curl http://****:****@webdav.sco.alma.cl/restricted/proprietary/RTI-RHEL6-x86_64.tar.bz2 | tar -xj

#Debugging Tools
RUN yum -y install gdb net-tools

#User configuration
RUN groupadd -r -g ${GROUP_ID} ${GROUP_NAME}
RUN useradd -r -m -u ${USER_ID} -g ${GROUP_NAME} ${USER_NAME}
RUN groupadd sudo
RUN usermod -aG sudo ${USER_NAME}
RUN echo new2me |passwd ${USER_NAME} --stdin
RUN echo "%sudo          ALL=(ALL)          ALL" >> /etc/sudoers
RUN echo "%sudo          ALL=(ALL)          NOPASSWD: ALL" >> /etc/sudoers
USER ${USER_NAME}
```

Centos 8

centos8/Dockerfile

```
FROM centos:centos8.1.1911

ARG USER_NAME
ARG GROUP_NAME
ARG USER_ID
ARG GROUP_ID

#General packages
RUN dnf -y install vim redhat-lsb-core sudo dbus-x11 rsync dnf-plugins-core
RUN dnf config-manager --set-enabled PowerTools

#Install Git and Git LFS
RUN dnf -y install epel-release
RUN dnf -y install https://download1.rpmfusion.org/free/el/rpmfusion-free-release-8.noarch.rpm
RUN dnf -y install git
RUN curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.rpm.sh | bash
RUN dnf -y install git-lfs

#Install Java 11
RUN dnf -y install java-11-openjdk-devel
#RUN echo 1| alternatives --config java_sdk_openjdk

#Prepare for Java
RUN mkdir -p /usr/java
RUN ln -s /usr/lib/jvm/java-openjdk /usr/java/default

#Install ACS required packages
RUN dnf -y install ksh gcc gcc-c++ gcc-gfortran libX11-devel libffi-devel perl readline-devel bzip2 bzip2-devel
openssl-devel openldap-devel libxml2-devel freetype-devel libxslt-devel sqlite-devel expat-devel tcl-devel tk-
devel bison flex autoconf unzip dos2unix

#ARCHIVE
RUN dnf -y install libtool libdb-devel

#ICD
RUN dnf -y install cmake blas-devel cfitsio-devel wcstools-devel lapack-devel

#CONTROL
RUN dnf -y install binutils-devel libmemcached-devel

#CORRCommon
RUN dnf -y install kernel kernel-devel procmail

#TELCAL
RUN dnf -y install motif-devel texlive

#RTI DDS
#RUN cd / && curl http://*****:****@webdav.sco.alma.cl/restricted/proprietary/RTI-RHEL6-x86_64.tar.bz2 | tar -xj

RUN dnf -y install gdb net-tools

#User configuration
RUN groupadd -r -g ${GROUP_ID} ${GROUP_NAME}
RUN useradd -r -m -u ${USER_ID} -g ${GROUP_NAME} ${USER_NAME}
RUN groupadd sudo
RUN usermod -aG sudo ${USER_NAME}
RUN echo new2me |passwd ${USER_NAME} --stdin
RUN echo "%sudo          ALL=(ALL)          ALL" >> /etc/sudoers
RUN echo "%sudo          ALL=(ALL)          NOPASSWD: ALL" >> /etc/sudoers
USER ${USER_NAME}
```

Docker Compose

docker-compose.yml

```
version: "3.4"

x-args: &args
  USER_NAME: ${USER_NAME}
  GROUP_NAME: ${GROUP_NAME}
  USER_ID: ${USER_ID}
  GROUP_ID: ${GROUP_ID}

x-env: &env
  - DISPLAY=${DISPLAY}

x-volumes: &volumes
  - ${REPO_DIR}:/Repos:z
  - ${ALMA_DIR}/${CONTAINER_DIR}:/alma:z
  - ${HOME_DIR}/${CONTAINER_DIR}:/home/${USER_NAME}:z
  - /tmp/.X11-unix:/tmp/.X11-unix:z

x-acsc-images: &common
  build:
    dockerfile: Dockerfile
    context: ./${IMAGE_OS}
    args:
      *args
  security_opt:
    - seccomp:unconfined
  image: acsdev-${IMAGE_OS}
  environment:
    *env
  volumes:
    *volumes
  working_dir: /home/${USER_NAME}
  networks:
    - acs-docker

services:
  centos6:
    <<: *common
  centos7:
    <<: *common
  centos8:
    <<: *common

networks:
  acs-docker:
    driver: bridge
```

Docker Compose .env

- USER_NAME: The container user name
- GROUP_NAME: The container user group
- USER_ID: Id of local user executing docker instance
- GROUP_ID: Group id from local user executing docker instance
- ALMA_DIR: Path that has all the directories that will be mounted as /alma in the container, for instance in my examples is '/external/docker/'
 - Inside this directory there's a set of directories for each OS, each will be mounted as /alma for the different images/containers. For instance '/external/docker/centos7' will be mounted as /alma in centos7 containers
- HOME_DIR: Path that will be mounted as the /home/\$USER directory in the machine. The \$HOME_DIR/\$CONTAINER_OS will be mapped in the container to /home/almamgr with the default configuration
- REPO_DIR: Path to the Git Repository. In my case inside this path (/external/Repos) I have 'almasw' and 'worktrees', which allow to have several worktrees for building without affecting other parts of the system
- IMAGE_OS: Here I define the default OS to be used. This can be overridden on command line if other OS is going to be used, but the default one should be defined here
- CONTAINER_DIR: Here I define the default DIR to be used. This can be overridden on command line if other DIR is going to be used, but the default one should be defined here

An example of the .env file:

.env example

```
USER_NAME=almamgr
GROUP_NAME=almamgr
USER_ID=1000
GROUP_ID=1000
ALMA_DIR=/external/docker/alma
HOME_DIR=/external/docker/home
REPO_DIR=/external/Repos
IMAGE_OS=centos7
CONTAINER_DIR=centos7
```

Makefile

Makefile

```
images:=$(wildcard centos*)
build: $(images)
.PHONY: $(images)
all: $(images)
$(images):
    @IMAGE_OS=$@ CONTAINER_DIR=$@ docker-compose build $@
$(addprefix up-, $(images)):
    @IMAGE_OS=$(subst up-,,$@) CONTAINER_DIR=$(subst up-,,$@) docker-compose up $(subst up-,,$@)
$(addprefix down-, $(images)):
    @IMAGE_OS=$(subst down-,,$@) CONTAINER_DIR=$(subst down-,,$@) docker-compose down $(subst down-,,$@)
```

Example

```
make centos7
```

Example directory tree

```
.
├── centos6
│   └── Dockerfile
├── centos7
│   └── Dockerfile
├── centos8
│   └── Dockerfile
├── docker-compose.yml
├── .env
└── Makefile
```

Bash

```
#Executing centos7
docker-compose run centos7 bash
#Executing centos8 requires to define the CONTAINER_OS variable
IMAGE_OS=centos8 CONTAINER_DIR=centos8 docker-compose run centos8 bash
```