Install docker and CVMFS

Check if docker is install on the computer and if you have permission to run it by running `id` command. If docker is a part of the ouput it is ok. If not, ask to the IT departement to install docker and add you to the docker group.

Test docker with the command :

```
docker run -it hello-world
```

If a error occur, ask to the IT departement.

Check if CVMFS is install and configure correctly by running the command `cd /cvmfs/aosis.opensciencegrid.org`. If the command is execute it is ok. If not ask to the IT departement to install CVMFS and setup the aosis.opensciencegrid.org folder.

Download clas12software docker

Create folder : `mkdir /vol0/mywork` and `cd /vol0/mywork` (`/vol0` is the physical disk of your compute, you can write on it)

1. Add your localhost to the list of accepted X11 connections with one of these two commands (if the first doesn't work, try the second one):

   ```
   xhost 127.0.0.1
   xhost local:root
   ```

2. Export the env variable DISPLAY:

   ```
   export DISPLAY=:0
   ```

3. Run the command using your local x11 tmp dir:

   ```
   docker run -it --rm -v /cvmfs:/cvmfs -v /tmp/.X11-unix:/tmp/.X11-unix
   -v /vol0/mywork:/jlab/work/mywork -e DISPLAY=$DISPLAY
   jeffersonlab/clas12software:production /bin/bash
   ```

Generate ALERT geometry inside the docker

Inside the docker, create `script_install.sh` in `mywork` folder and open the file for editing with nano.

```
echo "remove java-1.8.0"
dnf remove java-1.8.0-openjdk-headless.x86_64 -y

echo "install java-11"
dnf install java-11-openjdk-devel -y

echo "install maven"
wget https://www-us.apache.org/dist/maven/maven-3/3.6.3/binaries/apache-
maven-3.6.3-bin.tar.gz -P /tmp
tar xf /tmp/apache-maven-3.6.3-bin.tar.gz -C /opt
ln -s /opt/apache-maven-3.6.3 /opt/maven

export JAVA_HOME=/usr/lib/jvm/jre-openjdk
export M2_HOME=/opt/maven
export MAVEN_HOME=/opt/maven
export PATH=${M2_HOME}/bin:${PATH}

echo "Set python as alternative for python3"
alternatives --set python /usr/bin/python3

echo "groovy install"
curl -s get.sdkman.io | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
sdk install groovy
```

Run the script for install good version of java, maven, groovy and setup python3 as python.

```
. script_install.sh
```

Clone the `clas12-offline-software` repository in `mywork` with `git clone`.

```
git clone https://github.com/JeffersonLab/clas12-offline-software
```

Switch to Alert branch :

```
cd clas12-offline-software && git checkout Alert
```

And build `clas12-offline-software` with available script `build-coataja.sh`.

```
./build-coatjava.sh
```

Change directory to `mywork` and clone `detectors` repository.

```
cd /jlab/work/mywork && git clone https://github.com/gemc/detectors
```

Generate AHDC geometry with `run-groovy` command and `factory_ahdc.groovy` script and copy it into `alert/AHDC_geom` folder.

```
./../../clas12-offline-software/coatjava/bin/run-groovy
alert/AHDC_geom/factory_ahdc.groovy --variation rga_fall2018 --runnumber 11
&& cp ahdc__* alert/AHDC_geom/
```

Generate ATOF geometry with `run-groovy` command and `factory_atof.groovy` script and copy it into `alert/ATOF_geom` folder.

```
./../../clas12-offline-software/coatjava/bin/run-groovy
alert/ATOF_geom/factory_atof.groovy --variation rga_fall2018 --runnumber 11
&& cp atof__* alert/ATOF_geom/
```

Build AHDC detector with `ahdc.pl` script.

```
cd alert/AHDC_geom && ./ahdc.pl config.dat
```

Change line `detector_name: myatof` to `detector_name: atof` in `ATOF_geom/config.dat`with nano editor and then build ATOF detector with `atof.pl` script.

```
cd ../ATOF_geom && ./atof.pl config.dat
```

Go to `mywork` folder and clone `clas12Tags` repository and change directory to `clas12Tags/4.4.0/source`

```
cd /jlab/work/mywork && git clone https://github.com/gemc/clas12Tags && cd
clas12Tags/4.4.0/source
```

Build GEMC from source with SCons.

```
scons -j4 OPT=1
```

Create a `alert.gcard` on source folder and open the file for editing.

```
<gcard>

    <!-- Implementation ahdc, example -->
    <detector
name="/jlab/work/mywork/detectors/clas12/alert/AHDC_geom/ahdc"
factory="TEXT" variation="default"/>
    <detector
name="/jlab/work/mywork/detectors/clas12/alert/ATOF_geom/myatof"
factory="TEXT" variation="default"/>

    <option name="BEAM_P" value="e-, 10.0*GeV, 20*deg, 20*deg"/>
    <option name="SPREAD_P" value="2.0*GeV, 20*deg, 180*deg, flat"/>

    <option name="BEAM_V" value="(0, 0, -2.0)cm"/>
    <option name="SPREAD_V" value="(0.0, 0.0)cm"/>

    <option name="OUTPUT" value="txt, out.txt"/>
    <option name="N" value="1"/>


</gcard>
```

And then run gemc with the gcard

```
./gemc alert.gcard
```