TUTORIAL 01:
CREATE THE WORKING ENVIRONEMENT

In this tutorial, we will review some basic instructions to **prepare the working environment**. We will work on the CHPC **cluster** called LENGAU. In this tutorial we will review the steps to log onto the super-computer. Then, we will copy some CROCO directories that are already stored on the super-computer (instead of downloading some archives from the CROCO website https://www.croco-ocean.org). At last, we will test our working environment.

STEP 1: Logging onto the Lengau HPC cluster

→ To connect to the CHPC you must write, from a terminal/console, the following instruction:

```
ssh -X login@lengau.chpc.ac.za
```

✋ Replace **login** with your corresponding account number.

→ When entering the LENGAU super-computer, you will see something similar to this in your terminal:

```
Last login: Mon Sep 12 13:28:23 2022 from XX.XXX.XX.XXX

Welcome to LENGAU

################################################################################
#                                                                              #
# In order to receive notifications via email from the CHPC all users should   #
# be subscribed to the CHPC user distribution list. If you are not part of the #
# distribution list you can subscribe at the following link:                   #
# https://lists.chpc.ac.za/mailman/listinfo/chpc-users                         #
#                                                                              #
################################################################################
[login@login2 ~]$
```

➥ This will take you to your directory on your `home` directory on Lengau: **/home/login**

✋ You should **NOT** use your `home` directory to save large datasets or run CROCO!
You should go to your directory on `lustre` (there is a link to it in your **/home/login** directory) to do all the processing and saving of data.

→ To list the files and directories inside your home directory, just type the Linux command **ls**:

```
[login@login2 ~]$ ls
lustre
[login@login2 ~]$
```

➥ If the `lustre` symbolic link does not exist in your home directory, you have to create it with the Linux command **ln**:

```
[login@login2 ~]$ ln -sf /mnt/lustre/users/login lustre
[login@login2 ~]$ ls -l
lustre -> /mnt/lustre/users/login/
[login@login2 ~]$
```

→ To know where your home directory is, just type the Linux command **pwd**:

```
[login@login2 ~]$ pwd
/home/login
[login@login2 ~]$
```

→ Go to your `lustre` directory by executing the Linux change directory command **cd**:

```
[login@login2 ~]$ cd lustre
[login@login2 lustre]$ pwd
/home/login/lustre
[login@login2 lustre]$ ls
[login@login2 lustre]$
```

STEP 2: Requesting some interactive nodes to manipulate large files

On the Lengau cluster, you can **NOT** manipulate large files or perform calculous with MATLAB/python/nco softwares directly on the login node.

➥ You need to **request a node** on the compute server that will be reserved for your personal use. You need to execute your processes on these allocated nodes. If you execute a process on the frontal note it will be killed by the system.

➥ You need to open an interactive **pbs** session to request a node on the compute server. This can be done by typing the following in the command line:

```
qsub -X -I -l select=1:ncpus=4:mpiprocs=4 -q serial -l walltime=4:00:00 -P WCHCP
```

➥ In the command above, we are requesting **4** processors on one node for **4** hours. We also have to specify the project we belong to, i.e. **WCHPC**.

To make things easier, I created two aliases: **qsubi1** and **qsubi4** to reserve 1 or 4 nodes respectively. These aliases are defined in my **.bashrc** script. The latter is a hidden file in your **home** directory (that you can see by adding the option **-a** to the **ls** command).

→ Replace your **.bashrc** file with mine:

```
[login@login2 lustre]$ cd
[login@login2 ~]$ cp /mnt/lustre/users/sillig/CROCO_TRAINING_Week1/3_Some_files/.bashrc ~
[login@login2 ~]$ source ~/.bashrc
[login@login2 ~]$ cd lustre
```

→ Request one node with the alias command **qsubi1** (it can take a few seconds):

```
[login@login2 ~]$ qsubi1
qsub: waiting for job 4416950.sched01 to start
qsub: job 4416950.sched01 ready
[login@cnode0220 lustre]$ pwd
/home/login
[login@cnode0220 lustre]$
```

NODES

When you will execute **qsubi1 (qsubi4)**, it sends you **back to your home directory**. You need to navigate back to the directory where you want to be in.

→ Go back into your lustre directory:

```
[login@cnode0220 ~]$ cd lustre
[login@cnode0220 lustre]$ ls
[login@cnode0220 lustre]$
```

NODES

STEP 3: Copying the source code of the CROCO model (croco_v1.2.1)

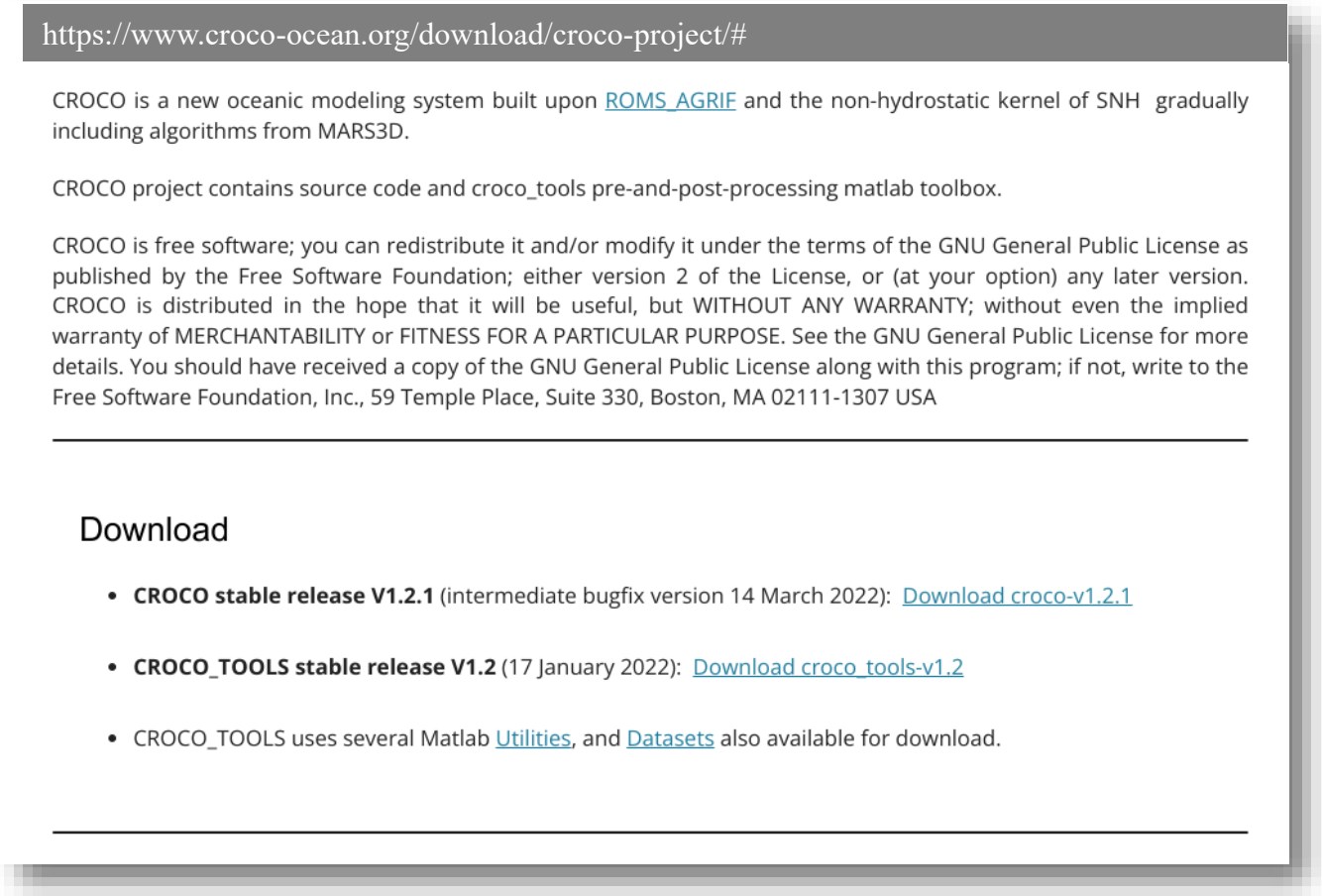
We are going to copy of the CROCO code pre-release version v1.3 (released in October 2022) in our lustre directory.

CROCO code source can be freely downloaded from the CROCO website https://www.croco-ocean.org/download/croco-project (at https://data-croco.ifremer.fr/CODE_ARCHIVE/croco-v1.2.1.tar.gz).

➥ This file has already been downloaded and uncompressed on the Lengau cluster.

→ To copy the code in your lustre directory, execute the following command in the terminal:

```
cp -r /mnt/lustre/users/sillig/CROCO_TRAINING_Week1/1_CROCO_code/croco .
```


https://www.croco-ocean.org/download/croco-project/#

CROCO is a new oceanic modeling system built upon ROMS_AGRIF and the non-hydrostatic kernel of SNH gradually including algorithms from MARS3D.

CROCO project contains source code and croco_tools pre-and-post-processing matlab toolbox.

CROCO is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. CROCO is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Download

- **CROCO stable release V1.2.1** (intermediate bugfix version 14 March 2022): Download croco-v1.2.1
- **CROCO_TOOLS stable release V1.2** (17 January 2022): Download croco_tools-v1.2
- CROCO_TOOLS uses several Matlab Utilities, and Datasets also available for download.

→ Using the command **ls**, you can now see that the directory **croco** has been copied in your lustre directory:

```
[login@cnode0220 lustre]$ ls
croco
[login@cnode0220 lustre]$
```

→ You can go into the **croco** directory and list its content, using the following Linux commands: **cd** and **ls**:

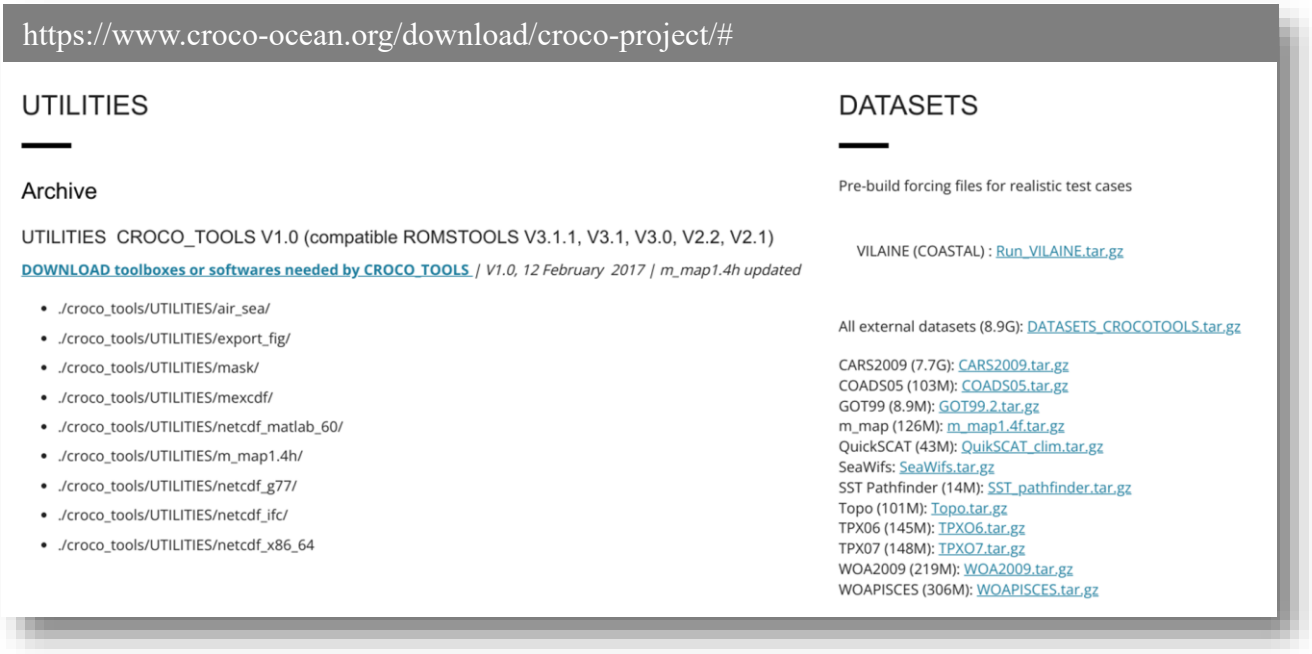
```
[login@cnode0220 lustre]$ cd croco
[login@cnode0220 croco]$ ls
 AGRIF                 DOC_SPHINX                OCEAN       SCRIPTS
 create_config.bash    MPI_NOLAND_preprocessing  PISCES      TEST_CASES
 CVTK                  MUSTANG                   README.md   XIOS
[login@cnode0220 croco]$ cd ..
[login@cnode0220 lustre]$
```

STEP 4: Copying the source code of the CROCO tools (crocotools_v1.2)

Then, we are going to get a copy of the CROCO_TOOLS (pre-release version v1.3, released in October 2022) in our lustre directory.

CROCO_TOOLS, along with some useful data sets and utilities can be freely downloaded from the CROCO website https://www.croco-ocean.org/download/croco-project (at https://data-croco.ifremer.fr/CODE_ARCHIVE/croco_tools-v1.2.tar.gz; https://www.croco-ocean.org/download/datasets/; and https://www.croco-ocean.org/download/utilities/).

➥ These files have already been downloaded and uncompressed on the Lengau cluster.



→ To copy the CROCO_TOOLS code that we will use, execute the following command:

```
cp -r /mnt/lustre/users/sillig/CROCO_TRAINING_Week1/2_CROCO_tools/croco_tools .
```

→ Using the Linux command **ls**, you can now see that the directory **croco_tools** has been copied in your lustre directory:

```
[login@cnode0220 lustre]$ ls
croco croco_tools
[login@cnode0220 lustre]$
```

NODES

→ You can go into the **croco_tools** directory and list its content:

```
[login@cnode0220 lustre]$ cd croco_tools
[login@cnode0220 croco_tools]$ ls
Aforc_CFSR            Diagnostic_tools              Preprocessing_tools
Aforc_ECMWF           example_job_prepro_matlab.pbs readme_version_croco_tools.txt
Aforc_ERA5            Forecast_tools                Rivers
Aforc_NCEP            job_prepro_matlab.pbs         RUNOFF_DAI
Aforc_QuikSCAT        Nesting_tools                 start.m
Coupling_tools        oct_start.m                   Tides
croco_pyvisu          Oforc_OGCM                    Town
crocotools_param.m    Opendap_tools                 UTILITIES
DATASETS_CROCOTOOLS   Opendap_tools_no_loaddap      Visualization_tools
[login@cnode0220 croco_tools]$ cd ..
[login@cnode0220 lustre]$
```

NODES

STEP 5: Testing MATLAB Software

The CROCO_TOOLS are an ensemble of routines written to help you create input files that are needed to run your CROCO simulations (pre-processing) and make some analyses of the model outputs (post-processing). This toolbox uses the software MATLAB. Let's test if MATLAB works correctly on your account:

→ Using the Linux command **cp**, copy two test files in your lustre directory:

```
cp /mnt/lustre/users/sillig/CROCO_TRAINING_Week1/3_Some_files/TP0* .
```

→ List the content of your lustre directory:

```
[login@cnode0220 lustre]$ ls
croco croco_tools TP0_test_file.nc TP0_test_script.m
[login@cnode0220 lustre]$
```

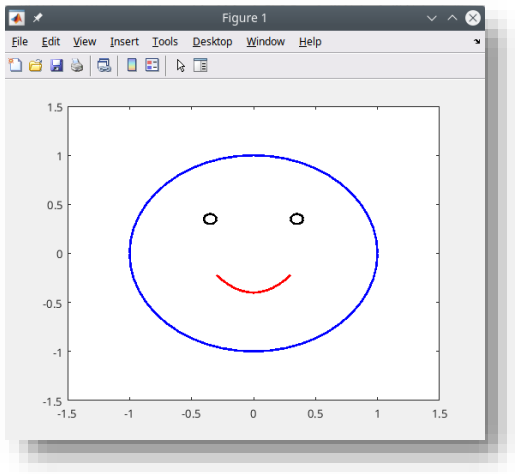
→ Launch MATLAB software by typing **matlab -nodesktop** (or the alias **mat**)

```
[login@cnode0220 lustre]$ matlab -nodesktop
[login@cnode0220 lustre]$
```

→ When MATLAB is launched, you will see something similar to this in your terminal:

```
                            < M A T L A B (R) >
                  Copyright 1984-2020 The MathWorks, Inc.
                R2020a Update 8 (9.8.0.1873465) 64-bit (glnxa64)
                               February 3, 2022

To get started, type doc.
For product information, visit www.mathworks.com.
>>
```



→ Inside MATLAB, you can list the file using the MATLAB command ls:

```
>> ls
croco croco_tools
TP0_test_file.nc TP0_test_script.m
>>
```

→ Execute the Matlab script **TP0_test_script**

```
>> TP0_test_script
>>
```

→ The script opens a NetCDF file and plot its content. You should get a first plot revealing a smiling face.

```
>> exit
```

→ You can remove the **TP0_test_*** files:

```
[login@cnode0220 ~]$ rm TP0_test_*
rm: remove regular file 'TP0_test_file.nc'? y
rm: remove regular file 'TP0_test_script.m'? y
[login@cnode0220 ~]$ ls
croco-v1.2.1 croco_tools
[login@cnode0220 ~]$
```

STEP 6: Exiting

→ Give back the compute node and logout from Lengau:

```
[login@cnode0220 ~]$ exit
logout
qsub: job 4416950.sched01 completed
[login@login2 ~]$ exit
```