Intra-ACP Climate Service and related applications (ClimSA)

Climate Station

Installation Manual

Version 1.1.1

August 2022

Date: 10/08/2022
Ref: ClimSA/DOC/C-Station/InstallationManual

Abstract / Résumé
This document lists all the products available on Climate Station (or C-Station)

	Name	Position
Prepared by	Marco Clerici	JRC-EC responsible for ClimSA
Contributions/Reviews by	Christophe Lavaysse Jurriaan Van't Klooster Vijay Charan Venkatachalam Dario Simonetti Fabrizio Cappucci	JRC-EC thematic Expert for ClimSA IT-GIS Specialist IT-GIS Specialist IT-GIS Specialist IT-GIS Specialist

	Details	Date
Version history		
Version 1.1.0	Initial Release	May 2022
Version 1.1.1	Update the procedure with changes, namely in climatestation.sh	August 2022

Contents

List of Figures and Tables

ACRONYMS and DEFINITIONS

RAM	Random Access Memory
CPU	Central Processing Unit
OS	Operating System
CS	Climate Station
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
Jupyter	Julia, Python and R
JWT	JSON Web Tokens
SFTP	Secure File Transfer Protocol
JRC	Joint Research Centre
GIS	Geographical Information System

1. INTRODUCTION

This document presents the software installation procedures of the Climate Station. It serves the system administrator to configure their server to install the climate station software. This section also contains the hardware and software requirements

1.1 HARDWARE REQUIREMENT

The Linux server dedicated for the Climate Station, physical server or a virtual machine, should have the following minimum requirements:

1.1.1 Minimum Size

Component	requirement
CPU	8 core
RAM	16GB
Disk storage space	2 TB

Table 1: Minimum Size – Hardware requirement

1.1.2 Medium Size

Component	requirement
CPU	Xeon-Gold 5115 (2x)
Core	20 /40 threads
RAM	512 GB
Disk storage space	OS + application software >= 2 x 250GB SSD in RAID1 Data >= 5/6 disks 1.2TB SAS HDD (RAID5 or RAID6 configuration, respectively) 4.8 TB space storage
Estimated Price	8000 – 12000 euro

Table 2: Medium Size – Hardware requirement

1.1.3 Large Size Server

Component	requirement
CPU	Xeon-Gold 6138 2.5 GHz (2x)
Core	40/80 threads
RAM	1024 GB
Disk storage space	OS + application software >= 2 x 250GB SSD in RAID1 Data >= 8 disks 1.8TB SAS HDD (RAID5 or RAID6 configuration) 12 Tb space in RAID5 – 10.8 Tb in RAID6
Estimated Price	15000 – 30000 euro

Table 3: Large Size Server – Hardware requirement

1.2 Software Requirement

The Climate Station (CS) installer needs the following software packages to be installed on the host machine:

- Docker engine (version 19.03+)
- Docker compose (version 1.29+)
- Git (version 2.22+)

2. Preparation of the Installation

2.1 User Definition

The user on the host machine that installs the software requirements must have root privileges for this. This can be done in two ways:

- Have your system administrator install these requirements as root.
- Give sudo (super user) rights to install the requirements to the CS user created on the host machine (by your system administrator, see below)[1].

On the host machine, it is considered best practice to not use the “root” user for installing an application, but to create a CS user with “sudo” rights.

Create a user (e.g. adminuser) and give this user sudo rights:

```
$ adduser adminuser sudo
$ usermod -aG sudo adminuser
```

[1] https://www.digitalocean.com/community/tutorials/how-to-create-a-sudo-user-on-ubuntu-quickstart
https://docs.docker.com/engine/install/linux-postinstall/

2.2 Installation of Docker and Docker compose

Install all packages as root or as a user with sudo rights. All following commands are done by the CS user with sudo.

Docker Engine
Please follow the installation instructions for the OS on your host machine:
https://docs.docker.com/engine/install/

Docker-compose
Installation instructions:
https://docs.docker.com/compose/install/
https://pypi.org/project/docker-compose/

Install the required packages and dependencies:

CentOS:

```
$ sudo yum install python3-pip
$ sudo yum install rust
$ sudo pip3 install –upgrade pip
$ sudo pip3 install setuptools
$ sudo pip3 install setuptools-rust
```

Ubuntu:

```
$ sudo apt-get install python3-pip
$ sudo apt-get install rustc
$ sudo pip3 install –upgrade pip
$ sudo pip3 install setuptools
$ sudo pip3 install setuptools-rust
```

Install docker-compose:

```
$ sudo pip3 install docker-compose
```

Once the installation is completed, check if it is installed fine by checking its version in the command prompt as follows:

- Docker engine → docker --version

```
$ docker --version
Docker version 20.10.14, build a224086
```

- Docker Compose → docker-compose --version

```
$ docker-compose --version
docker-compose version 1.28.5, build unknown
```

Important:
The new user created in the previous step (e.g. adminuser), needs to be able to run docker without sudo. To do so, add the user to the docker group:

```
$ su - adminuser
$ sudo groupadd docker
$ sudo usermod -aG docker adminuser
$ newgrp docker
```

2.3 INSTALLATION OF GIT

The Climate Station installation package is made available in the git, so in order to install it you have to install git[2] in your machine either as root user or as a user (eg. adminuser) with sudo rights. For example if you're on a Debian-based distribution, such as Ubuntu, try apt:

```
$ sudo apt install git
```

Once installation is completed, check if it is installed fine by checking its version in command prompt as below:

```
$ git --version
git version 1.8.3.1
```

2.4 CLONING THE CLIMATESTATION CODE FROM GITHUB

To download the code of the Climate Station you will have to clone the climatestation repository from github on your local machine.

After you installed Git on your computer, open a Terminal and run the following commands:

- First move to the directory where you want to create the clone. This will be the root directory of the installation, indicated as <climatestation_dir>:

  ```
  $ cd  <climatestation_dir> (eg. /opt or /home/adminuser)
  ```

- Execute git-clone:

  ```
  $ git clone https://github.com/eStation2/climatestation.git climatestation
  Cloning into "climatestation" ...
  remote: Enumerating objects: 17901, done.
  remote: Counting objects: 100% (17901/17901), done.
  remote: Total 25110 (delta 3689), reused 17689 (delta 3527), pack-reused 7209
  Receiving objects: 100% (25110/25110), 173.65MiB | 11.68 MiB/s, done.
  Resolving deltas: 100% (6625/6625), done.
  Checking out files: 100% (34362/34362), done.
  ```

- Check the content of the directory where the clone has been created and it should contain the following files and directories:

```
              $ cd climatestation
              $  ls -sla
0     drwxrwxr-x  8    adminuser adminuser   321     May    4    11:39    .
4     drwx------  14   adminuser adminuser   4096    May    5    16:07    ..
0     drwxrwxr-x  8    adminuser adminuser   106     May    4    10:02    build-docker
20    -rwxrwxr-x  1    adminuser adminuser   20341   May    4    1:39     climatestation.sh
0     drwxrwxr-x  2    adminuser adminuser   40      May    4    10:02    doc
4     -rw-rw-r--  1    adminuser adminuser   3466    May    4    11:39    docker-compose.yml
4     -rw-rw-r--  1    adminuser adminuser   897     May    4    10:02    .env.template
4     -rwxrwxr-x  1    adminuser adminuser   211     May    4    10:02    get-ownership.sh
0     drwxrwxr-x  8    adminuser adminuser   247     May    9    11:32    .git
4     -rw-rw-r--  1    adminuser adminuser   17      May    4    10:02    .gitattributes
4     -rw-rw-r--  1    adminuser adminuser   288     May    4    10:02    .gitignore
4     -rw-rw-r--  1    adminuser adminuser   132     May    4    10:02    .gitmodules
8     -rw-rw-r--  1    adminuser adminuser   5061    Apr    6    15:57    README.md
4     drwxrwxr-x  7    adminuser adminuser   4096    May    4    10:02    src
```

[2] https://git-scm.com/download/linux
https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

3. Installation of the Climate Station

3.1 Checking the current installation and settings

Running the command below for the first time, will check the current installation version and settings.

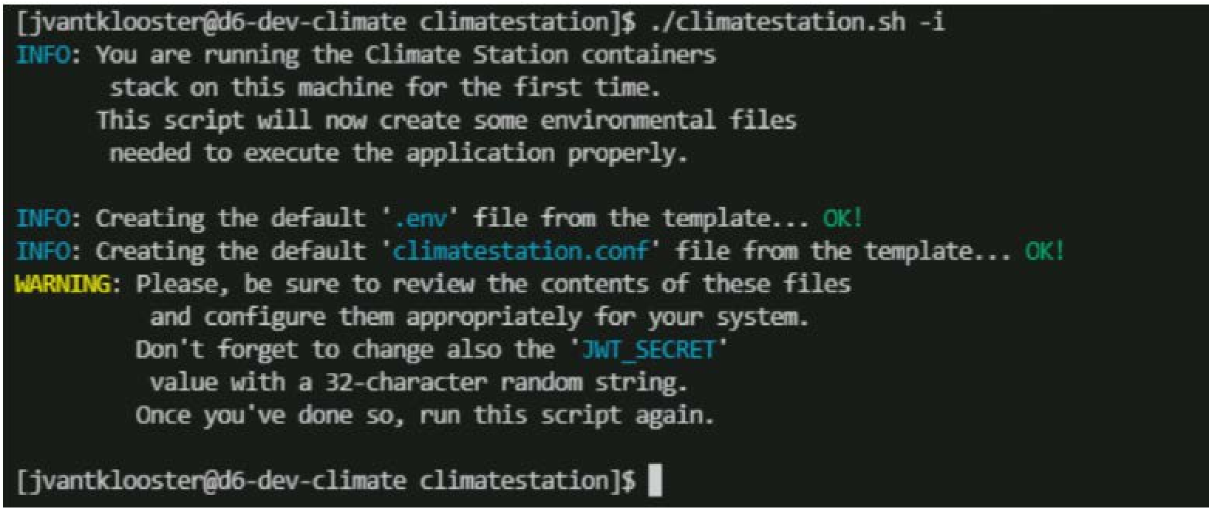
Make sure that the internet connection is stable!

Open a Terminal and run the following commands:

$ cd <climatestation_dir> *(e.g. /opt/climatestation)*

$ ./climatestation.sh -i ***[do not run it as root]***

You will be asked to change the settings in the .env file (see Figure 1).



```
[jvantklooster@d6-dev-climate climatestation]$ ./climatestation.sh -i
INFO: You are running the Climate Station containers
       stack on this machine for the first time.
      This script will now create some environmental files
       needed to execute the application properly.

INFO: Creating the default '.env' file from the template... OK!
INFO: Creating the default 'climatestation.conf' file from the template... OK!
WARNING: Please, be sure to review the contents of these files
          and configure them appropriately for your system.
         Don't forget to change also the 'JWT_SECRET'
          value with a 32-character random string.
         Once you've done so, run this script again.

[jvantklooster@d6-dev-climate climatestation]$
```

Figure 1: Initial notification to change the variable in the .env file

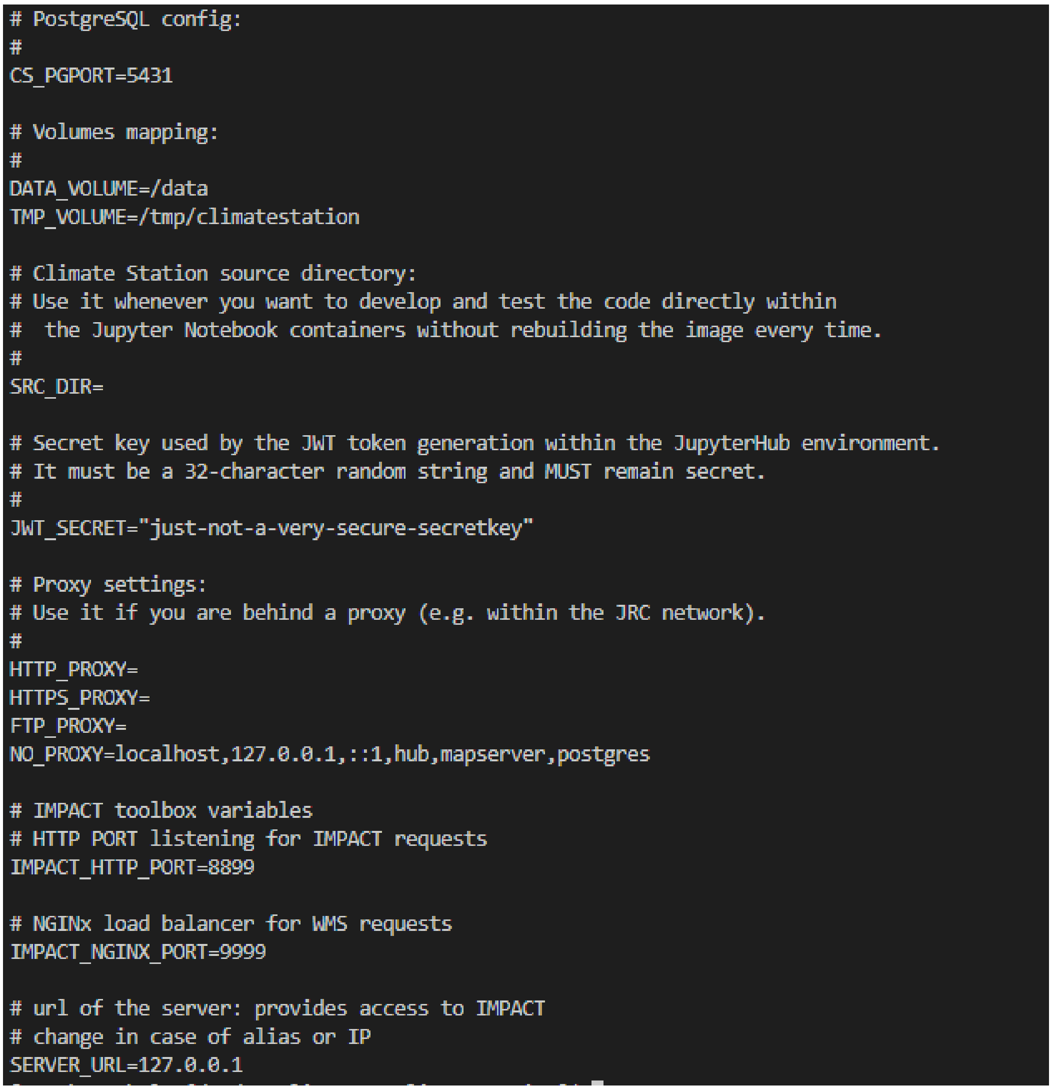
3.2 Customize User settings

For the installation of the Climate Station, there are some important definitions that might need to be updated before continuing the installation, e.g the working directories on the host machine, the proxy settings and others. All these variables are defined in the .env file, which is in the root directory of the installation after you have run ./climatestation.sh the first time (see Figure 1).

You can use ‘vi’ (or another editor like nano or gedit) - to modify definitions in the .env file.

$ cd <climatestation_dir> (e.g. /opt/climatestation)

$ vi .env



```
# PostgreSQL config:
#
CS_PGPORT=5431

# Volumes mapping:
#
DATA_VOLUME=/data
TMP_VOLUME=/tmp/climatestation

# Climate Station source directory:
# Use it whenever you want to develop and test the code directly within
#  the Jupyter Notebook containers without rebuilding the image every time.
#
SRC_DIR=

# Secret key used by the JWT token generation within the JupyterHub environment.
# It must be a 32-character random string and MUST remain secret.
#
JWT_SECRET="just-not-a-very-secure-secretkey"

# Proxy settings:
# Use it if you are behind a proxy (e.g. within the JRC network).
#
HTTP_PROXY=
HTTPS_PROXY=
FTP_PROXY=
NO_PROXY=localhost,127.0.0.1,::1,hub,mapserver,postgres

# IMPACT toolbox variables
# HTTP PORT listening for IMPACT requests
IMPACT_HTTP_PORT=8899

# NGINx load balancer for WMS requests
IMPACT_NGINX_PORT=9999

# url of the server: provides access to IMPACT
# change in case of alias or IP
SERVER_URL=127.0.0.1
```

Figure 2: The list of parameters to be customized

A number of parameters (Figure 2) can be customized to match the User’s environment, in terms of volume mapping, proxy definition.

Optional:

- DATA_VOLUME is the base directory for the installation of the data, both static data and datasets (default /data).

- TMP_VOLUME is a working directory for temporary files, e.g. intermediate steps of computation (default /tmp/climatestation).
- The 4 PROXY definitions (HTTP_PROXY, HTTPS_PROXY, FTP_PROXY, NO_PROXY) have to be used in case the host machine operates behind a proxy, and are needed to reach the internet.
- CS_PGPORT by default is 5431 but if you have the postgres already running in that port you can modify it.

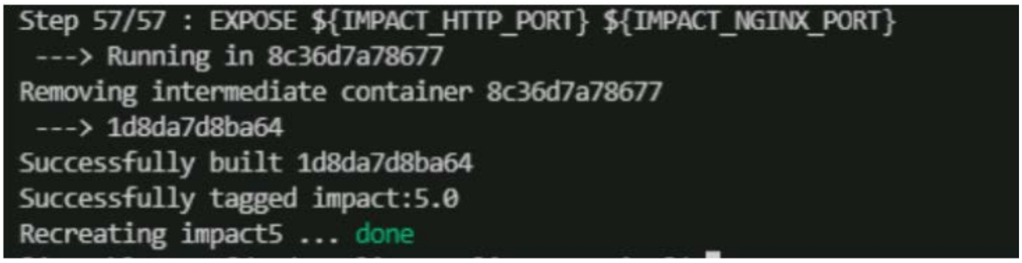
3.3 Building and starting the Climate station

Now that the user settings have been corrected (if needed), we can build and start the Climate Station. Make sure that the internet connection is stable.

Open a Terminal and run the following commands:

```
$ cd <climatestation_dir> (e.g. /opt/climatestation)
$ ./climatestation.sh -i     [do not run it as root]
```

The first time you build the Climate Station might take up to 20 minutes.
Output when build has finished:



```
Step 57/57 : EXPOSE ${IMPACT_HTTP_PORT} ${IMPACT_NGINX_PORT}
 ---> Running in 8c36d7a78677
Removing intermediate container 8c36d7a78677
 ---> 1d8da7d8ba64
Successfully built 1d8da7d8ba64
Successfully tagged impact:5.0
Recreating impact5 ... done
```

Figure 3: The final lines of the build

4. Post Installation Operations and checks

4.1 Check if the Climate station is running well

How to check if the Climate Station is running well?

Open a web browser and go to:

http://localhost:8080

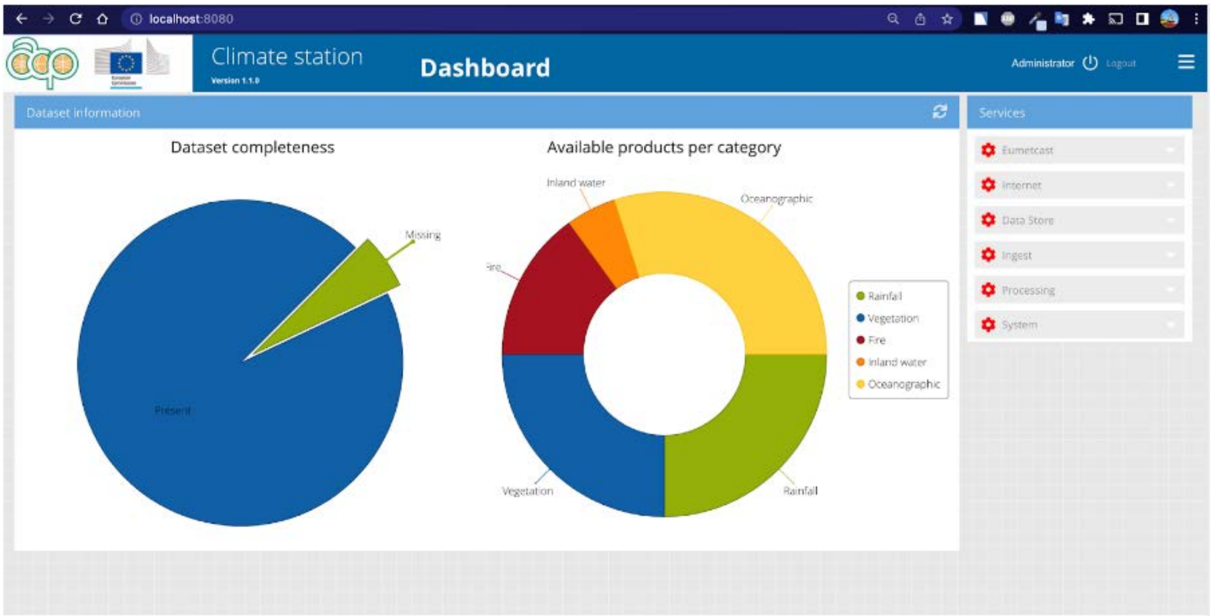


Figure 4: Climate Station dashboard

4.2 CHECKING THE DATA AND OTHER DIRECTORIES

Check the existence of the following directories under the DATA directory indicated in the .env file, by default the /data directory on your host machine.

On the first build and start of the Climate Station, using ./climatestation.sh (see section 3.3), the sub directories under the /data directory will be created as follows:

The "data" directory will contain the directories:

+ processing
+ ingest
+ ingest.wrong
+ static_data

The "static_data" sub directory will contain the following directories:

+ completeness_bars
+ config_cds
+ config_iri
+ db_dump
+ docs
+ get_lists
+ layers
+ log
+ logos
+ requests
+ settings

4.3 CONTROLLING THE CLIMATE STATION APPLICATION

The Climate Station hosts a bash script that controls the execution of the containers, and is meant to be used for their initial creation, their update (see section 5) and switching them ON or OFF. Here below the main commands to control the status.

- Move to the directory where the script is located:

  $ cd <climatestation_dir> (e.g. /opt/climatestation)

- Stop the Climate Station

  $ ./climatestation.sh -s

- Start the Climate Station

  $ ./climatestation.sh

- Restart the Climate Station

  $ ./climatestation.sh -s
  $ ./climatestation.sh

- Rebuild the Climate Station using Docker cache (faster)

  $./climatestation.sh -i

- Rebuild the Climate Station without using Docker cache (containers rebuild from scratch)

  $./climatestation.sh -i -C

5. Upgrading the Climate Station

For upgrading the Climate Station version, the same script for its creation and control is used. You need to go into the directory where the Climate Station has been cloned, shutdown the containers and then pull the upgrade from git and re-build CS, through the operations indicated below.

- Navigate to climate station folder where is source code

  $ cd <climatestation_dir> (e.g. /opt/climatestation)

- Shutdown the Climate Station

  $ ./climatestation.sh -s

- Record the changes if any and revert to clean current working directory

  $ git stash

- Pull the latest ClimateStation code

  $ git pull origin master

- Build the new version

  $ ./climatestation.sh –i

- You will be asked to change the settings in the .env file (see Figure 1) in some case if the configuration is changed. Follow the section 3.2 in order to modify them.

- After changing the configurations build it again

  $ ./climatestation.sh –i

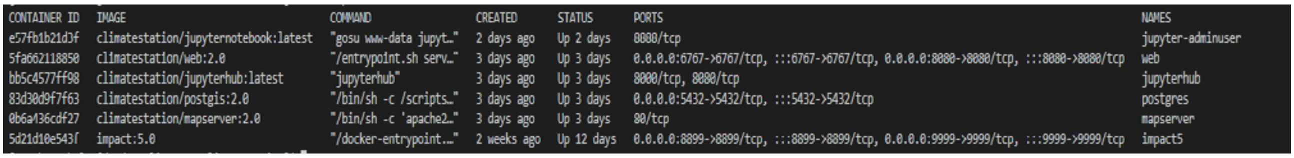
6. ANNEX

This Annex contains some additional indications on how to operate, from the command line, on the containers of the Climate Station and operate on the host machine.

6.1 WORKING WITH DOCKER CONTAINERS

- Check the status of the containers. Are they up?

$ docker ps



```
CONTAINER ID   IMAGE                                   COMMAND                  CREATED       STATUS        PORTS                                                                      NAMES
e57fb1b21d3f   climatestation/jupyternotebook:latest   "gosu www-data jupyt…"   2 days ago    Up 2 days     8000/tcp                                                                   jupyter-adminuser
5fa662118850   climatestation/web:2.0                  "/entrypoint.sh serv…"   3 days ago    Up 3 days     0.0.0.0:6767->6767/tcp, :::6767->6767/tcp, 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   web
bb5c4577ff98   climatestation/jupyterhub:latest        "jupyterhub"             3 days ago    Up 3 days     8000/tcp, 8080/tcp                                                         jupyterhub
83d30d9f7f63   climatestation/postgis:2.0              "/bin/sh -c /scripts…"   3 days ago    Up 3 days     0.0.0.0:5432->5432/tcp, :::5432->5432/tcp                                  postgres
0b6a436cdf27   climatestation/mapserver:2.0            "/bin/sh -c 'apache2…"   3 days ago    Up 3 days     80/tcp                                                                     mapserver
5d21d10e543f   impact:5.0                              "/docker-entrypoint.…"   2 weeks ago   Up 12 days    0.0.0.0:8899->8899/tcp, :::8899->8899/tcp, 0.0.0.0:9999->9999/tcp, :::9999->9999/tcp   impact5
```

Figure 5: List of containers available

How do I operate on a container?

- Open a terminal inside a container/attach a shell to a container

$ docker exec -it web bash
(base) root@f4db624c4e4a:/var/www/climatestation#

- Run a script that is inside a container

$ docker exec -ti postgres sh -c "/install_update_db.sh"

- Run a python script inside a container

$ docker exec -ti web bash python apps/acquisition/test/test_get_internet.py

6.2 Connecting data, database and code

- Where are the data?

How do I see the CS data from the host machine?

You can find the CS data under the DATA directory indicated in the .env file, by default the data directory is /data both on the host and in the web container.

The actual data is under the directory:

/data/processing

View the data in your favorite GIS tool (like QGIS) directly from the /data/processing directory on the host.

- Where is the Database, and how to reach it?

The PostgreSQL database data is in a docker volume called ***cs-docker-postgresql12-volume***.

You can work directly on the database using the tool pgAdmin4.
Install pgadmin4 desktop version:
https://www.pgadmin.org/download/

Install pgadmin4 in docker container:
https://hub.docker.com/r/dpage/pgadmin4/

Open pgAdmin4 and define a server as follows:

Host	: localhost
Port	: 5432
DB	: estationdb
Username	: estation
Password	: mesadmin

- Where is the code?

You can find the code in the ‘src’ directory under the directory where the Climate Station has been cloned. By default:

/opt/climatestation/src

How to check if the code is up-to-date?

Open a Terminal and type in the following commands:

```
$ cd /opt/climatestation
$ git pull origin master
```