Global Monitoring for Environment and Security (GMES) and Africa

eStation 3.0

Installation Manual for Linux

Version 1.3.2

February 2025

Date: 21/02/2025
Ref: GMES/DOC/eStation3.0/EN/InstallationManual

Abstract / Résumé
This document lists all the products available on eStation

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

	Details	Date
Version history		
Version 1.2.0	Initial Release	October 2023
Version 1.2.1		September 2024
Version 1.3.2	Better explanation of the parameters in .env file.	February 2025

Contents

ACRONYMS and DEFINITIONS

RAM	Random Access Memory
CPU	Central Processing Unit
OS	Operating System
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 eStation. It serves the system administrator to configure their server to install the eStation software. This section also contains the *hardware and software requirements.*

1.1 OVERVIEW OF THE INSTALLATION

The installation of the eStation consists of the following steps

- Find a suitable computer/server as host machine (see 1.2 for some indications)
- Identify or create a user with administrative rights on the computer (see 2.1)
- Install Docker and Git on the host machine (see 2.2 and 2.3)
- Clone the Git repository containing the installation script (see 2.4)
- Execute the installation script to pull Docker images, and customize settings (see 3)
- Perform post installation checks, and control the application (see 4)

The current document also contains instructions on how to upgrade the eStation to the latest version (see 5) and an Annex with some insights on the application.

1.2 HARDWARE REQUIREMENT

The Linux server dedicated for the eStation, physical server or a virtual machine should have the following requirements. Three different options are indicated, according to the equipment availability and the expected workload for the platform.

1.2.1 Minimum Size

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

Table 1: Minimum Size Server

1.2.2 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 Server

1.2.3 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

1.3 SOFTWARE REQUIREMENT

The machine where you want to install the eStation (also called host-machine) needs the following software packages to be installed:

- 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, used to install 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 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 user with "sudo" rights.

Create a user, or identify and existing one, (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 a user with sudo rights. All following commands are done by the 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 eStation installation package is made available on github, 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 ESTATION INSTALLER FROM GITHUB

To download the installer of the eStation you will have to clone the Installer 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 <Installer_root_dir>:

  $ ***cd <cs_installer_root_dir>*** *(eg. /opt or /home/adminuser)*

- Execute git-clone:

  $ git clone https://github.com/eStation2/CS-Installer.git
  git clone https://github.com/eStation2/CS-Installer.git
  Cloning into 'CS-Installer'...

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

  $ cd CS-Installer
  $ ls -sla

0	drwxrwxr-x	8	adminuser adminuser	321	May	4	11:39	.
4	drwx------	14	adminuser adminuser	4096	May	4	16:07	..
4	-rw-rw-r--	1	adminuser adminuser	9835	May	4	11:39	cs_install.sh
20	-rwxrwxr-x	1	adminuser adminuser	936	May	4	1:39	.env.template
0	drwxrwxr-x	2	adminuser adminuser	416	May	4	10:02	.git
4	-rw-rw-r--	1	adminuser adminuser	9835	May	4	11:39	es_install.sh
4	-rw-rw-r--	1	adminuser adminuser	2298	May	4	11:39	docker-compose.yml

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

3. INSTALLATION OF THE ESTATION

3.1 PULLING THE DOCKER IMAGES

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 <CS-Installer_dir> *(e.g. /home/adminuser/CS-Installer)*

$ ./es_install.sh -p

Flags description:
-p Pull the eStation docker images and update if any.

The first time you pull and build the eStation might take up to 10 minutes, depending on your internet connection.

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

```
[adminuser@s-jrciprap262p CS-Installer]$ ./cs_install.sh -p
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!
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

[adminuser@s-jrciprap262p CS-Installer]$
```

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

3.2 CUSTOMIZE USER SETTINGS

For the installation of the eStation, 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 ./es_installer.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 <CS-Installer-dir> *(e.g. /home/adminuser/CS-Installer)*

$ vi .env

```
# PostgreSQL config:
#
CS_PGPORT=5431

# WEB port config:
#
CS_WEBPORT=8080

# 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

# 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, port definition and server domain name.

Optional:

- The PostgreSQL database port, CS_PGPORT by default is 5431 in your installation, but if that port is already used by another service on your server, you can modify it.

  For example:

  CS_PGPORT=5432

- The port used by the Climate Station web application, CS_WEBPORT by default is 8080 in your installation, but if that port is already used by another service, you can modify it.

  For example if you want your installation to work under port 80, set this parameter as follows:

  CS_WEBPORT=80

- 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 four PROXY definitions (HTTP_PROXY, HTTPS_PROXY, FTP_PROXY, NO_PROXY) have to be used in case the host machine operates behind a proxy in your network, and are needed to reach the internet.

  Example proxy settings for proxy server address http://10.168.200.71:8012 :
  HTTP_PROXY=http://10.168.200.71:8012
  HTTPS_PROXY=http://10.168.200.71:8012
  FTP_PROXY=http://10.168.200.71:8012

  Under the NO_PROXY you can add specific IP addresses to exclude from using the proxy server.

  For example:
  NO_PROXY=localhost,127.0.0.1,::1,hub,mapserver,postgres,10.169.11.29,10.169.11.35,139.191.149.56

  Always keep under the NO_PROXY the values:
  localhost,127.0.0.1,::1,hub,mapserver,postgres

- SERVER_URL indicates the url of the server and provides access to the IMPACT Toolbox Change in case your server has a dedicated IP address or domain name.

  For example:
  SERVER_URL=climatestation.institutionname.org

3.3 STARTING THE ESTATION

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

Open a Terminal and run the following commands:

$ cd <CS-Installer-dir> *(e.g. /home/adminuser/CS-Installer)*

$ ./es_install.sh

Output when build has finished:

INFO: No changes to configuration files.

[+] Running 4/4

✓ Network cs-installer_default	Created	0.1s
✓ Container postgres	Healthy	0.6s
✓ Container jupyterhub	Started	1.0s
✓ Container web	Started	2.0s

Climate Station is up

INFO: Waiting for the database containers to be ready to install updates… Ready

2023-10-11 07:21 Database structure already exists. Continue

2023-10-11 07:21 DB estationdb already up-to-date. Continue

Figure 3: The final lines of the build

4. POST INSTALLATION OPERATIONS AND CHECKS

4.1 CHECK IF THE ESTATION IS RUNNING WELL

How to check if the eStation is running well?

Open a web browser and go to:

http://localhost:8080

If you have changed the CS_WEBPORT in the .env file to port 80, go to:

http://localhost

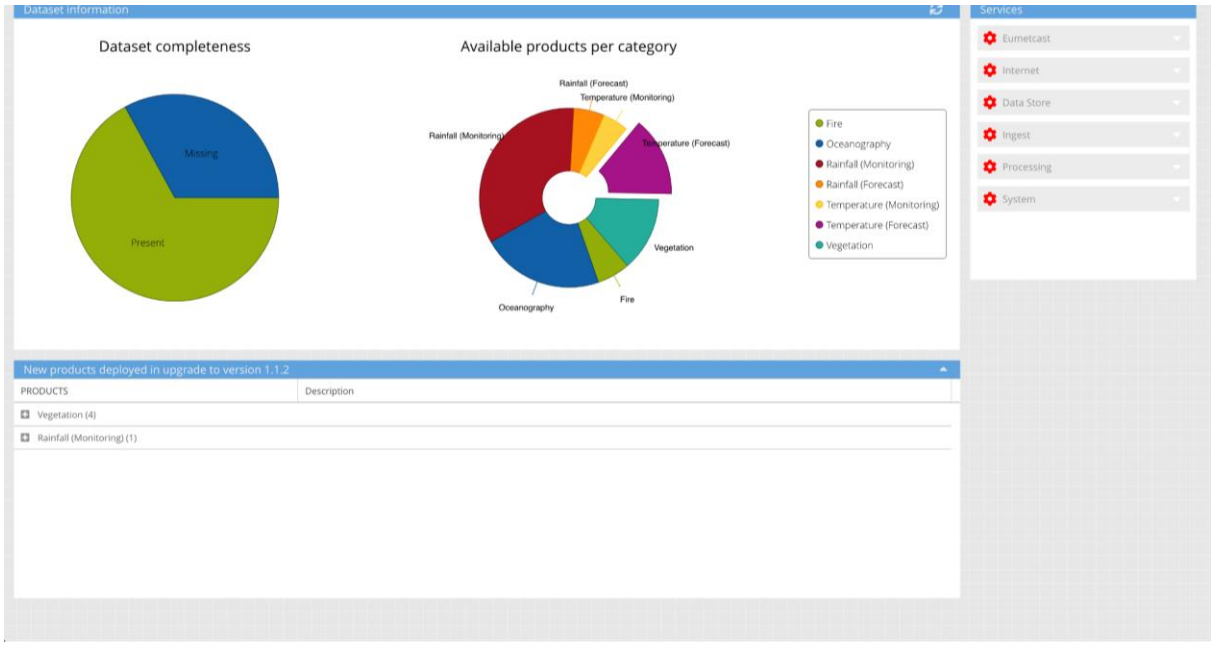

Figure 4: eStation dashboard

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 eStation, using ./***es_install***.sh (see section 3.3), the sub directories under the /data directory will be created as follows:

The "data" directory will contain at least the following directories (additional ones are created by running the application):

+ 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 ESTATION APPLICATION

The eStation 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 <CS-Installer_dir> (e.g. /home/adminuser/CS-Installer)

- Stop the eStation

  $ ./es_install.sh down

- Start the eStation

  $ ./es_install.sh up

- Restart the eStation

  $ ./es_install.sh down
  $ ./es_install.sh up

- Update the eStation (see next section for the correct upgrade procedure)

  $ git pull
  $./es_install.sh -p

5. UPGRADING THE ESTATION

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

- Navigate to eStation folder where is source code

  $ cd <CS-Installer_dir> (e.g. /home/adminuser/CS-Installer)

- Update the eStation

  $ git pull
  $./es_install.sh -p

6. ANNEX

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

6.1 WORKING WITH DOCKER CONTAINERS

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

$ docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
20f5e9d69713	mydronedocker/impact5:latest	"/docker-entrypoint.…"	30 hours ago	Up 30 hours	0.0.0.0:8899->8899/tcp	impact5
614191885f03	climatestation/cstation:latest	"/entrypoint.sh serv…"	30 hours ago	Up 30 hours	0.0.0.0:6767->6767/tcp, 0.0.0.0:8080->8080/tcp	web
1725444771fd	climatestation/cstation:latest	"/entrypoint.sh jupy…"	30 hours ago	Up 30 hours	0.0.0.0:59258->8000/tcp, 0.0.0.0:59257->8080/tcp	jupyterhub
3bfb56f0bfbc	climatestation/postgis:2.0	"/bin/sh -c /scripts…"	30 hours ago	Up 30 hours	0.0.0.0:5431->5432/tcp	postgres

Figure 5: List of containers available

How do I operate on a container?

2. Open a terminal inside a container/attach a shell to a container

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

3. Run a script that is inside a container

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

4. 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 is the data?

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

You can find the eStation data under the DATA_VOLUME directory indicated in the .env file. By default the data directory is /data on the host machine (as well as in in the web container).

The <DATA_VOLUME> is the root directory, whose contents in described in 4.2; the actual datasets of the eStation are under the directory:

/<DATA_VOLUME>/processing

You can visualize the data by using your favorite GIS tool (like QGIS) directly from this directory on the host machine.

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, which is available for download at

https://www.pgadmin.org/download/

The is also the option to install pgadmin4 in the docker container, as explained at:
https://hub.docker.com/r/dpage/pgadmin4/

In order to access the eStation DB, you can open pgAdmin4 and define a server as follows:

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

The direct access to the eStation DB is meant only for advanced users, since all relevant changes in the configuration of the eStation can be done from the user interface.