Copernicus Maintenance Team

GSS COTS Installation

Role/Title	Name	Signature	Date
Authors	Giuseppe Presta, Iryna Miatka		09/12/2024
Verified/Approved	Elisabetta Giuliani, Giulia Carosi		09/12/2024

Change Register

Version/Rev.	Date	Description
1.0	13/02/2023	First version of the document
1.1	22/03/2023	Updated Version
1.2	17/10/2023	Updated Version
1.3	28/05/2024	• Section 1.3: updated document applicability • Section 1.5: updated GSS SDD and Administration Manual versions • Section 2.5.3: added procedure to manage contingency case of no space left on Solr and Zookeeper containers
1.4	09/12/2024	• Section 2.6.1: added how to manage a contingency case of Zookeeper instances running on the same machine. • Section 3: added Kafka configuration for the RedHat OP

Table of Contents

1. Introduction

1.1 Scope

This document applies to the GAEL Store Service (GSS) and is maintained within the service "Collaborative Data Hub Software Maintenance and Evolution Services for Digital Twin Earth" hereinafter called "the Collaborative service".

1.2 Purpose

This document aims to detail step-by-step instructions to install, configure and use all the software necessary to properly run the GSS.

1.3 Document applicability

Please note that this document is referring to DHS#7 of GSS, according to GSS Administration Manual as per RD-1

Document version	Component	DHS Release
2.1.1	GSS	DHS#7

1.4 Document structure

The document is structured as follows:

- Section 1 (this section) contains scope and purpose, providing document structure, reference documents and definitions/acronyms.
- Section 2 contains an overview of the GSS COTS Installation with detailed description of all its components.
- Section 3 contains the procedure for the GSS COTS installation for the RedHat based OS.

1.5 Reference documents

Table 1 - Reference Documents

Ref.	Title	Reference and Version
RD-1.	Collaborative Data Hub Software GSS Administration Manual	GAEL-P311-GSS-CDH-Administration Manual, 2.1.1
RD-2.	Collaborative Data Hub Software GSS Software Design Document	GAEL_P311 – GSS-CDH-SDD, v2.1.1

1.6 Acronysm and Abbreviations

Table 2 - Acronyms and Abbreviations

Acronym	Definition
GSS	GAEL Store Service
COTS	Commercial-Off-The-Shelf

2. GSS COTS Installation

2.1 Overview

This is a list of all the software necessary for the installation of the GSS with the related installation procedures, grouped in different tables for each software to install.

2.2 Docker engine installation

This procedure is referred to 24.0.5 Docker engine version.

Docker Engine version required by GSS: 20.10.12 and after

Table 3 Docker engine installation procedure

Step ID	Step Description	Expected Results
1.	As Administrator, access via SSH to the VM where the docker engine should be installed. Log in as root user, and perform the following steps to install the docker engine: 1) apt update 2) apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common 3) curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add - 4) add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable" 5) apt update 6) apt-cache policy docker-ce 7) apt install docker-ce 8) systemctl status docker 9) docker --version	The "docker --version" reports correctly the docker version installed.
2.	Execute the following step post-installation: 1) groupadd docker 2) usermod -a -G docker dhs 3) apt install gnupg2 pass	All commands successfully performed.

2.3 Docker compose installation

This procedure is referred to 1.29.2 docker compose version.

Docker compose version required by GSS: 1.29.0 and after

Table 4 Docker compose installation procedure

Step ID	Step Description	Expected Results
1	As Administrator, access via SSH to the VM where the docker compose should be installed. Log in as root user, and perform the following steps to install the docker compose: 1) curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose 2) chmod +x /usr/local/bin/docker-compose 3) docker-compose –version	The "docker-compose --version" reports correctly the docker compose version installed.

2.4 Postgres installation

2.4.1 Pre-Requisite

Before the Postgres installation in docker mode it is necessary the Java installation.

This procedure is referred to the Java installation.

Java version required by GSS: 17 and after (the procedure below is for version 17).

Table 5 Java installation procedure

Step ID	Step Description	Expected Results
1	As Administrator, access via SSH to the VM where Java should be installed. Log in as root user. The first step is to remove the old Java version: **Public Repository** 1) apt-get remove openjdk* 2) apt-get remove --auto-remove openjdk* 3) apt-get purge openjdk* **Private Repository** 1. apt-get --autoremove jdk*	The old Java version is removed

2	Download the Java SE Development Kit 17 package and install it using one of these procedures: **Private Repository** • wget https://download.oracle.com/java/17/archive/jdk-17_linux-x64_bin.deb • apt install ./jdk-17_linux-x64_bin.deb **Public Repository** • wget https://download.java.net/java/GA/jdk17/0d483333a00540d886896bac774ff48b/35/GPL/openjdk-17_linux-x64_bin.tar.gz • tar -xvzf openjdk-17_linux-x64_bin.tar.gz • cd jdk-17 • mkdir -p /usr/lib/jvm/openjdk-17 • mv * /usr/lib/jvm/openjdk-17	Java 17 is installed
3	Configure the Java environment: **Private Repository** From terminal, execute: • cat <<EOF | sudo tee /etc/profile.d/jdk.sh export JAVA_HOME=/usr/lib/jvm/jdk-17/ export PATH=\$PATH:\$JAVA_HOME/bin EOF **Public Repository** From terminal, execute: • export JAVA_HOME=/usr/lib/jvm/openjdk-17 • export PATH=$JAVA_HOME/bin:$PATH • source ~/.bashrc	Java env is configured
4	Installation Check: **Private Repository** From the terminal, execute: 1) source /etc/profile.d/jdk.sh 2) java -version **Public Repository** From the terminal, execute: 1. java –version 2. echo $JAVA_HOME	The Java software is correctly installed and the exposed version is the expected one.

2.4.2 Installation

This procedure is referred to 13.4 Postgres version.

Postgres version required by GSS: 10.12 and after

Table 6 Postgres installation procedure

Step ID	Step Description	Expected Results
1	To install and configure Postgres 13.4 pull the Postgres docker image: • docker pull postgres:13.4 To check if Postgres docker image has been pulled successfully, execute: • docker image list To run Postgres docker image, execute: • docker run --name postgres_13.4 -e POSTGRES_PASSWORD=<password> -d -p 5432:5432 postgres:13.4 -N 2100 Finally, to install Postgres, execute: • sudo apt -y install postgresql postgresql-client	Postgres is correctly installed and configured
2	To create the first Database execute the following commands on terminal: 1) psql -h localhost -U postgres (insert the password to access) 2) CREATE DATABASE gss; 3) exit	The Postgres database is correctly created.

2.5 SOLR installation

2.5.1 Pre-Requisite

Before the SOLR installation in docker mode it is necessary the Java installation.

The Java installation procedure is highlighted in 2.4.1.

2.5.2 Installation

This procedure applies to Solr 9.0.0

Solr version required by GSS: 8.0.0 and after

Table 7 Solr installation procedure

<table>
<tr><th>Step ID</th><th>Step Description</th><th>Expected Results</th></tr>
<tr><td>1</td><td>

As Administrator, access via SSH to the VM where the SOLR should be installed.

Following these steps:

- Open the “docker-compose.yml” compose file (otherwise, create it with the command “vi docker-compose.yml”) and configure it as follows:

```
version: '3.7'
services:
        solr-1:
                image: solr:9.0.0
                container_name: solr-1
                volumes:
                        - <path_to_folder>/solr-data:/var/solr
                ports:
                        - 8983:8983
                environment:
                        - SOLR_HOST=<IP>
                        - ZK_HOST=zoo-1
                        - SOLR_OPTS=-Dsolr.autoCommit.maxTime=6000 -Dsolr.autoSoftCommit.maxTime=3000
                depends_on:
                        - zoo-1
        zoo-1:
                image: zookeeper:3.8
                container_name: zoo-1
                restart: always
                hostname: zoo-1
                volumes:
                        - <path_too_folder>/zoo1:/data
                ports:
                        - 2181:2181
                environment:
                        ZOO_MY_ID: 1
                        ZOO_SERVERS: server.1=10.21.2.10:2888:3888;2181
```

</td><td>SOLR configuration file created. SOLR and Zookeeper configurations set correctly</td></tr>
</table>

2	Run, sequentially, the following commands to complete the installation: 1 docker-compose -f docker-compose.yml up -d zoo-1 2 docker-compose -f docker-compose.yml up -d solr-1	Zookeeper and SOLR installation are performed successfully
3	Create the new collection on SOLR with the command: 1 docker exec solr-1 solr create_collection -c gss	The new collection is correctly created on SOLR
4	For the SOLR initialization use the following commands: 1 docker exec -u root -it solr-1 /bin/bash 2 wget -O /opt/solr/server/solr-webapp/webapp/WEB-INF/lib/jts-core-1.19.0.jar https://repo1.maven.org/maven2/org/locationtech/jts/jts-core/1.19.0/jts-core-1.19.0.jar 3 exit 4 docker restart <container_solr>	SOLR is correctly initialized

2.5.3 Contingency: No space left on containers

It could happen that Solr and Zookeeper saturate the space in the container where they are writing. As consequence, the restart of both applications fails and a manual intervention is needed.

In order to clean device space, the following command can be executed:

```
docker system prune -a -f
```

If this does not improve the situation, the involved containers shall be cleaned and Docker shall be uninstalled, by following the Procedure below.

Table 8 Resolution no space left on container procedure

Step ID	Step Description	Expected Results
1	As Administrator, access via SSH to the VM where the docker compose is installed. Log in as root user and delete the content of the following folders: 1) cd /var/lib/docker 2) rm -rf containers/* 3) rm -rf overlay2/*	The content of both folders is correctly deleted.
2	Delete all the docker components present in the machine. 1) apt-get purge -y docker-engine docker docker.io docker-ce docker-ce-cli docker-compose-plugin docker-ce-rootless-extras golang-docker-credential-helpers docker-buildx-plugin 2) apt-get autoremove -y docker-engine docker docker.io docker-ce docker-ce-cli docker-compose-plugin docker-ce-	All the docker components are correctly deleted.

	rootless-extras golang-docker-credential-helpers docker-buildx-plugin	
3	Delete all the images, containers, volumes, or user created configuration files on your host. 1) rm -rf /var/lib/docker /etc/docker 2) rm /etc/apparmor.d/docker 3) groupdel docker 4) rm -rf /var/run/docker.sock 5) rm -rf /var/lib/containerd	All the images, containers, volumes, or user created configuration files are correctly deleted

2.6 Kafka installation

This procedure applies to the latest Kafka version.

Kafka version required by GSS: 3.3.1 and after.

Table 9 Kafka installation procedure

<table>
<tr><th>Step ID</th><th>Step Description</th><th>Expected Results</th></tr>
<tr><td>1</td><td>

As Administrator, access via SSH to the VM where Kafka should be installed.

Following these steps:

- Open the “docker-compose.yml” compose file (otherwise, create it with the command “vi docker-compose.yml”) and configure it as follows:

```
version: '3'
services:
        zookeeper:
                image: bitnami/zookeeper:latest
                container_name: zookeeper
                ports:
                        - "2181:2181"
                environment:
                        - ALLOW_ANONYMOUS_LOGIN=yes
        kafka:
                image: bitnami/kafka:latest
                container_name: kafka
                ports:
                        - "9092:9092"
                environment:
                        - KAFKA_BROKER_ID=1
                        - KAFKA_NUM_PARTITIONS=10
KAFKA_CFG_ADVERTISED_LISTENERS=PLAINTEXT://<IP_MACHINE>:2181
                        - KAFKA_CFG_ZOOKEEPER_CONNECT=zookeeper:2181
                        - ALLOW_PLAINTEXT_LISTENER=yes
                depends_on:
```

</td><td>Kafka configuration file created.
Zookeeper and Kafka configurations set correctly</td></tr>
</table>

	- zookeeper	
2	Run, sequentially, the following commands to complete the installation: 3 docker-compose -f docker-compose.yml up -d zookeeper 4 docker-compose -f docker-compose.yml up -d kafka	Zookeeper and Kafka installation are performed successfully

2.6.1 Contingency: port conflict and session refusals

If the Zookeeper instances for Solr and Kafka are running on the same machine you should adapt one of the instances and configure it with the different port. This change will prevent the conflict of using the same port.

When facing the issues with session refusals error messages from a ZooKeeper server (for example, "Refusing session request for client…"), it may indicate that Kafka clients are not connecting properly to the Kafka instance due to misconfiguration.

Please, ensure that KAFKA_CFG_ADVERTISED_LISTENERS points to the Kafka instance and not the ZooKeeper instance, using the appropriate port. For example, changing the port from 2181 to 9092 fixes the problem.

3. Annex – Installation for RedHat based OS

The instructions in the GSS Administration Manual are not applicable for the RedHat based OS. For this case you may follow the instructions described in this annex.

3.1 Docker engine and compose installation

Docker Engine version should be 20.10.12

Table 3.1 Docker engine installation procedure

Step ID	Step Description	Expected Results
1.	As Administrator, access the VM where the docker engine should be installed. Log in as root user, and perform the following steps to install the docker engine: 1) sudo yum install -y yum-utils 2) sudo yum-config-manager –add-repo https://download.docker.com/linux/rhel/docker-ce.repo 3) sudo dnf config-manager –add-repo https://download.docker.com/linux/centos/docker-ce.repo 4) sudo dnf install docker-ce-20.10.12 docker-ce-cli-20.10.12 containerd.io 5) Alternative: sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y 6) sudo systemctl start docker 7) sudo systemctl enable docker	The "docker --version" reports correctly the docker version installed.

2.	Execute the following step post-installation: 1) sudo usermod -aG docker $USER 2) newgrp docker 3) sudo yum install gnupg2 pass	All commands successfully performed.
3.	To install the docker compose: 1) sudo curl -L “https://github.com/docker/compose/releases/download/1.29.2/docker-compose-\protect\TU\ textdollar(uname -s)-$(uname -m)” -o /usr/local/bin/docker-compose 2) sudo chmod +x /usr/local/bin/docker-compose	The “docker-compose --version” reports correctly the docker compose version installed.

3.2 Java and Postgres installation

Before the Postgres installation in docker mode it is necessary to install Java.

Java version required by GSS: 17 and after. If Java 17 is not available in the default repositories of RedHat, you may download it directly from the OpenJDK website and install it manually.

Postgres version required by GSS: 10.12 and after.

Table 3.2 Java and Postgres installation procedure

Step ID	Step Description	Expected Results
4.	As Administrator, access via SSH to the VM where Java should be installed. Log in as root user. Download the Java SE Development Kit 17 package and install it: • cd /usr/local • sudo curl -O https://download.java.net/java/GA/jdk17/0d483333a00540d886896bac774ff48b/35/GPL/ openjdk-17_linux-x64_bin.tar.gz • sudo tar xvf openjdk-17_linux-x64_bin.tar.gz • sudo vim /etc/profile • export JAVA_HOME=/usr/local/jdk-17 • export PATH=$JAVA_HOME/bin:$PATH • source /etc/profile • modify /etc/bashrc exactly as done in /etc/profile, also modify ~/.bash_profile	The Java software is correctly installed, and the exposed version is the expected one.

4a.	If Java is still pointing to the old JAVA version, you can follow the steps: • export JAVA_HOME=/usr/local/jdk-17 • export PATH=$JAVA_HOME/bin:$PATH • source ~/.bashrc By editing the ~/.bashrc file, you're setting the JAVA_HOME and PATH variables for your user only. This will not affect other users in the system. If you need to use Java 17 from another user or from a system service, please set the JAVA_HOME and PATH variables in a way that it's available system-wide (like in /etc/profile or /etc/environment) or set it specifically for that user or service. Check first the JAVA version before running or installing anything for another user.	
5.	To install and configure Postgres 13.4 pull the Postgres docker image: • docker pull postgres:13.4 To check if Postgres docker image has been pulled successfully, execute: • docker image list To run Postgres docker image, execute: • docker run --name postgres_13.4 -e POSTGRES_PASSWORD=<password> -d -p 5432:5432 postgres:13.4 -N 2100 Finally, to install Postgres, execute: • sudo dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm • sudo dnf -qy module disable postgresql • sudo dnf install -y postgresql13-server	Postgres is correctly installed and configured
6.	To create the first Database, execute the following commands on terminal: • psql -h localhost -U postgres • CREATE DATABASE gss; \q	The Postgres database is correctly created
7.	Check port 5432: • sudo iptables -L INPUT -n -v | grep 5432 If there is no output, or if the output is different from this line 0 0 ACCEPT tcp – * * 0.0.0.0/0 0.0.0.0/0 tcp dpt:5432 use the following command to open the port 5432: • sudo iptables -A INPUT -p tcp –dport 5432 -j ACCEPT	The port 5432 is opened

3.3 Solr and Kafka installation

For the RedHat based OS the following modifications should be added to the docker-compose.yml file:

Table 3.3 Postgres installation procedure

Step ID	Step Description	Expected Results
	For Solr: • docker-compose -f docker-compose.yml up -d zoo-1 • mkdir /root/solr-data • sudo chown 8983:8983 /root/solr-data/ • docker-compose -f docker-compose.yml up -d solr-1 • docker exec solr-1 solr create_collection -c gss • docker exec -u root -it solr-1 /bin/bash • wget -O /opt/solr/server/solr-webapp/webapp/WEB-INF/lib/jts-core-1.19.0.jar https://repo1.maven.org/maven2/ • org/locationtech/jts/jts-core/1.19.0/jts-core-1.19.0.jar • exit	
	For Kafka: • docker-compose -f docker-compose.yml up -d zookeeper • docker-compose -f docker-compose.yml up -d kafka	

In case you have an error related to iptables after starting a docker container, please restart docker and start one by one all the docker containers in this sequence:

– docker start postgres_13.4

– docker start zoo-1

– docker start solr-1

– docker start zookeeper

– docker start kafka