COPtool

COPtool INSTALLATION GUIDE

Pedro Montero, Garbiñe Ayensa, Silvia Allen-Perkins, Alberto Gómez

Co-funded by the European Union Civil Protection

ACKNOWLEDGEMENT

The work described in this report was supported by the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) of the European Union through the Grant Agreement number 101004912 - MANIFESTS – UCPM-2020-PP-AG, corresponding to the Call objective “Enhancing prevention and protection from the effects of maritime disasters” under priority 1: “Developing response capacity for marine pollution”.

DISCLAIMER

The content of this document represents the views of the author only and is his/her sole responsibility; it cannot be considered to reflect the views of the European Commission and/or the Directorate-General for European Civil Protection and Humanitarian Aid Operations (DG-ECHO) or any other body of the European Union. The European Commission and the DG-ECHO is not responsible for any use that may be made of the information it contains.

Co-funded by the European Union Civil Protection

Project Acronym	**MANIFESTS**
Project Full Title	**MAN**aging risks and **I**mpacts **F**rom **E**vaporating and gaseous **S**ubstances To population **S**afety
Gant Agreement Nr.	**101004912**
Project Website	**https://www.manifests-project.eu/**

Deliverable Nr.	**D5.2 MANIFESTS DSS_Instalation guide**
Status (Final/Draft/Revised)	**Draft**
Work Package	**5**
Task Number	**5.4**
Responsible Institute	**INTECMAR**
Author/s	**P- Montero, G. Ayensa, S. Allen-Perkins, A. Gómez**
Recommended Citation	
Dissemination Level	

Document History			
Version	Date	Modification Introduced	
		Modification Reason	Modified by
1.0	31/01/2023		

Content

Co-funded by the European Union Civil Protection

1. Background

In the event of a maritime accident involving HNS, maritime authorities must take numerous decisions to organize the best response strategy, i.e., one that minimizes risks to human health (including incident response teams, crew members and coastal communities), to the marine environment, for maritime safety and for socio-economic activities and facilities. While many key decisions and considerations are prescribed in national or regional contingency plans, operational response activities will generally need ongoing adjustment or review to reflect the most recent information available as the contamination event evolves. In such a rapidly changing situation, an efficient exchange of information between competent decision-making authorities and response teams on the ground can greatly facilitate both decision-making processes and organizational processes.

The purpose of work package 5 of the Manifests project (Manifests decision support system) is to develop an efficient information system that helps (1) decision makers understand the situation at stake and its likely evolution in the coming hours and days; (2) identify the population, ecosystems and socio-economic assets at risk and (3) share useful information with response teams deployed at sea, in the air or on the coast.

Building on the experience gained and development carried out during the previous HNS-MS and MARINER projects, the MANIFESTS decision support system (DSS) will integrate several services, including the DSS Common Operational Picture (COPtool).

This COPtool refers to a system designed so that during a contingency, the exchange of information that occurs between the maritime authorities and the different response teams (sea, coast, air) is carried out in the most efficient way possible, ensuring that all actors involved in the crisis committee and response teams can access the same data. These can be standard reports (such as the Standard Pollution Observation Report of the Bonn Agreement), images, videos and any other georeferenced data collected by response teams, as well as satellite observations, model simulation results, exclusion areas, location of response media, requests for new response actions shared by the crisis.

The COPtool documentation is composed by:

- Installation guides (this document).
- User Manual.
- DSS Implementation report.

2. Services

The platform consists of the following services:

- API: Consisting of a REST API that will allow the consultation, creation, update, and deletion of data. This service will be in charge of user authentication.
- WEB: This service will be in charge of serving the resources and managing the redirection of requests to the data service.

2.1. Software Requirements

- Operating System: Windows/Linux
- PostgreSQL https://www.postgresql.org/ (>9.5)
- PostGIS: https://postgis.net/
- Java/OpenJDK 1.8
- Apache with mod_proxy_ajp, mod_ssl, mod_rewrite https://httpd.apache.org/
- JDBC authentication driver

2.2. Files

- API service resources: *https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/API_Service*
- WEB service resources: *https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/WEB_Service*
- Database backup (PostgreSQL): *https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/Database*
- *JDBC driver: https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/JDBC_driver*
- *WinSW software: https://github.com/MANIFESTS-DSS/COPTOOL/tree/main/WinSW*

3. Database

The database requires postgreSQL 9.5 or higher with the PostGIS extension (created with PostGis v3.1).

Create the user and the database

```
CREATE USER coptool WITH SUPERUSER PASSWORD [yourpassword];
CREATE DATABASE "coptool" WITH owner=coptool;
```

Edit pg_hba.conf to trust local addresses.

Import into the database the data structure and default data from the backup **coptool.sql**. This backup was created with PostGis v3.1.

```
pg_restore -h localhost -p 5432 -U coptool -d coptool -v coptool.sql
```

The connection to the database will be configured in the properties file application.yml of the API service.

4. Setting API Service

1- Get API service code from GitHub.

2- Edit application.yml from \BOOT-INF\classes to suit your application environment.

```
server:
  port: ${port:9000}
spring:
  http:
    multipart:
      enabled: true
      location: /path/to/temp
      max-file-size: 200MB
      max-request-size: 200MB
  application:
    name: coptool-api
  datasource:
    platform: postgres
    url: jdbc:postgresql://localhost:5432/${db:coptool}
    username: coptool
    password: [yourpassword]
arcopol:
  visor-url: http://url/to/viewer/
```

3- Compress META-INF, BOOT-INF and org folders into a zip file, with no compression.

4- Rename the compressed file to coptool-api.jar

5. Setting WEB Service

1- Get WEB service code from GitHub.

2- Edit application.yml from \BOOT-INF\classes to suit your application environment.

```
server:
  port: 8080
tomcat:
  ajpEnabled: true
  ajpPort: 8010
  ajpSecretRequired: false
zuul:
  routes:
    api:
      service-id: api
      path: /api/**
      custom-sensitive-headers: true
      sensitive-headers: Cookie,Set-Cookie
      url: http://localhost:9000
    pub:
      service-id: pub
      path: /pub/**
      custom-sensitive-headers: true
      sensitive-headers: Cookie,Set-Cookie
      url: http://localhost:9000/pub
    auth:
      service-id: auth
      path: /auth/**
      custom-sensitive-headers: true
      sensitive-headers: Cookie,Set-Cookie
      url: http://localhost:9000/auth
```

3. In the root folder, compress META-INF, BOOT-INF and org folders into a zip file, with no compression

4. Rename the compressed file to coptool-web.jar

6. Services deployment (LINUX)

A folder must be created for each service. Copy the *.jar files inside them.

```
/path/to/coptool/api (API)
    -   coptool-api.jar
/path/to/coptool/web (WEB)
    -   coptool-web.jar
```

Create a repository and temp folder

```
/path/to/coptool/repository
/path/to/coptool/temp
```

Set permissions in jar files

```
chmod +x /path/to/coptool/api/coptool-api.jar
chmod +x /path/to/coptool/web/coptool-web.jar
```

Set system services

```
# /etc/init.d
ln -s /path/to/coptool/api/coptool-api.jar /etc/init.d/coptool-api
ln -s /path/to/coptool/web/coptool-web.jar /etc/init.d/coptool-web
```

Start services

```
systemctl start coptool-api
systemctl start coptool-web
```

7. Service deployment (Windows)

A folder must be created for each service. Copy the *.jar files inside them.

```
C:\path\to\coptool\api (API)
    -   coptool-api.jar
C:\path\to\coptool\web (WEB)
    -   coptool-web.jar
```

Create a repository and temp folder

```
C:\path\to coptool\repository
C:\path\to coptool\temp
```

Open two consoles (one for each of the folders) and test the execution with the following commands

```
C:\path\to coptool\api\java –jar coptool-api.jar
       (..)
C:\path\to coptool\api\java –jar coptool-web.jar
```

Opening a browser the service should be accessible locally at the URL: localhost:8080

Install as a service

In windows environment, it is recommended that the services run as a system service. For this task, WinSW tool can be used. Some template files are provided in GitHub folder.

https://github.com/winsw/winsw

coptool-api.xml sample

```
<service>
    <id>CopTool-API</id>
    <name>CopTool-API</name>
    <description>CopTool API REST</description>
    <env name="MYAPP_HOME" value="%BASE%"/>
    <executable>C:\Program Files\Java\jdk1.8.0_45\bin\java</executable>
    <arguments>-Xmx512m -jar "C:\path\to\coptool\api\coptool-api.jar"</arguments>
    <logmode>rotate</logmode>
</service>
```

8. Apache web server

It is mandatory for the service to be published through a secure protocol in order to geolocation functionalities work properly.

Verify SSL context: \Apache\conf\extra\httpd-ssl.conf

```
<VirtualHost _default_:443>

ProxyRequests On
ProxyPass / ajp://localhost:8010/ retry=0
ProxyPassReverse / ajp://localhost:8010/
Timeout 600
ProxyTimeout 600
Header set Access-Control-Allow-Origin "*"
Header always set Access-Control-Allow-Headers "*"
SSLCertificateFile "${SRVROOT}/conf/server.crt"
SSLCertificateKeyFile "${SRVROOT}/conf/server.key"
SSLCACertificateFile "${SRVROOT}/conf/ssl/ca_bundle.crt"
```

Verify settings in \Apache\conf\httpd.conf

```
Listen 80

LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule headers_module modules/mod_headers.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so

<IfModule ssl_module>
Include conf/extra/httpd-ssl.conf
```

9. Publishing service

External access should be through a subdomain e.g. coptool.plancamgal.gal under HTTPS. Proxy, firewall, etc must be configured so that port 443 is accessible from the outside.