Sentinels Reprocessing Preparation Package, Installation & Deployment Manual

Reference RRPP-TN-0011-CS

Issue/Revision 1.0

Date of Issue 10/05/2021

|  |  |  |
| --- | --- | --- |
| Description | Version | Date |
| Issued | 1.0 | 10/05/2021 |
|  |  |  |
|  |  |  |
|  |  |  |

**Change Log:**

Table of contents



[1. INTRODUCTION 5](#_Toc75184498)

[1.1 Purpose and scope 5](#_Toc75184499)

[1.2 Documents 5](#_Toc75184500)

[1.2.1 Reference documents 5](#_Toc75184501)

[1.2.2 Applicable documents 5](#_Toc75184502)

[1.3 Terminology 6](#_Toc75184503)

[1.3.1 Definitions 6](#_Toc75184504)

[1.3.2 List of Acronyms 6](#_Toc75184505)

[2. SENTINELS REPROCESSING PREPARATION PACKAGE 7](#_Toc75184506)

[2.1 Overview 7](#_Toc75184507)

[2.2 Sentinels Reprocessing Preparation Package 7](#_Toc75184508)

[3. SENTINELS REPROCESSING PREPARATION PACKAGE INTERFACE CONTEXT 8](#_Toc75184509)

[4. INSTALLATION AND DEPLOYMENT MANUAL 9](#_Toc75184510)

[4.1 Overview 9](#_Toc75184511)

[4.2 Installation and Deployment requirements 9](#_Toc75184512)

[4.3 Build docker images 9](#_Toc75184513)

[4.4 Deployment as docker-compose applications 10](#_Toc75184514)

[5. APPENDICES 11](#_Toc75184515)

[5.1 AUXIP POM file 11](#_Toc75184516)

[5.2 AUXIP docker-compose file 11](#_Toc75184517)

# INTRODUCTION

## Purpose and scope

This document details the installation and the deployment of Sentinels Reprocessing Preparation package.

The reprocessing Preparation package contains three main tasks:

* The Reprocessing Configuration Baseline API.
* The AUXIP API.
* The Reprocessing Data Baseline API.

## Documents

### Reference documents

1. OData Documentation <http://www.odata.org/documentation>
2. OData Protocol <http://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part1-protocol.html>
3. The OAuth 2.0 Authorization Framework – Authorization Code Grant, October 2012,Internet Engineering Task Force (IETF) [https://tools.ietf.org/html/rfc6749#section-4.1](https://tools.ietf.org/html/rfc6749%23section-4.1)

### Applicable documents

1. Auxiliary Data Interface Delivery Point Specification\_v1.1

## Terminology

### Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **API** | A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service. |
| **AUX\_File** | Auxiliary file used in a satellite processing chain |
|  |  |

### List of Acronyms

|  |  |
| --- | --- |
| Acronym | Signification |
| AD | Applicable Document |
| UUID | Universally Unique Identifier – Allows to identify an instance |
| JSON | JavaScript Object Notation, commonly used format in web services |
| URI | Uniform Resource Identifier |
| JPA | Java Persistence API |
| POM | Project object model |

# SENTINELS REPROCESSING PREPARATION PACKAGE

## 2.1 Overview

Sentinels Reprocessing Preparation Package prepares the future Sentinel-1/2/3 reprocessing activities by ensuring that all information required to reprocess historical data is available and ready to use in the frame of a reprocessing operations service.

The main activities of Sentinels Reprocessing Preparation Package are the following tasks:

* **Task 1: Build-up of a Reprocessing Configuration Baseline**

The scope of this task is to identify for each satellite unit and product type, the set of auxiliary files to be used for their reprocessing using the latest available data processor version. The information shall be recorded in a “Reprocessing Configuration Baseline Table”.

* **Task 2: Set up a reprocessing auxiliary files interface ( AUXIP API)**

The Auxiliary Data Interface Delivery Point ( AUXIP ) is the pick-up point for Sentinel Auxiliary products. The AUXIP allows clients to discover and retrieve available products through standard the OData RESTful API interface.

* **Task 3: Build-up of a Reprocessing Data Baseline**

The scope of this task is to identify for each satellite unit, and Level-0 product name and Level-1/2 product type, the applicable up to date set of auxiliary files to be used for their reprocessing as well as the latest applicable data processor version.

## Sentinels Reprocessing Preparation Package

ReproBaseAPI

AUXIP

ReprocessingDataBaseline

Compose

*build\_all.bash*

RollingArchive

Figure 1 : Reprocessing Preparation Package

# SENTINELS REPROCESSING PREPARATION PACKAGE INTERFACE CONTEXT

Sentinels Reprocessing Preparation package is a set of OData based RESTful API reinforced with an API gateway (Kong) and an openID Connect Provider (Keycloak) allowing the security improvement.

Figure 2 : Reprocessing preparation package interfaces

ReproBaseline

(Task 1)

API Gateway

(Kong)

OpenID Connect

(Keycloak)

Internet

End User (client)

Authentication Flow

Token introspection

Token introspection

Protected Network

AUXIP API

(Task 2)

Reprocessing Data Baseline

(Task 3)

* All requests sent to the AUXIP API passes through the **Kong gateway API** to check the client credentials.
* If clients are not authenticated, they will be redirected the **Keycloak interface** to create an account or to be authenticated.
* Client should retrieve the **access token** via a Keycloak interface in order to be able to send requests to reprocessing preparation package services.

# INSTALLATION AND DEPLOYMENT MANUAL

## Overview

All services forming the reprocessing preparation package are implemented as Java projects object models (POM) and all their dependencies and buildings are managed and simplified using Maven tool. See the AUXIP API pom.xml file in the appendices.

The installation process of the reprocessing preparation package is also simplified by dockerizing each service to be able to run it in the adequate environment making all dependencies available by using the docker-compose tool.

As mentioned before in the interface context, kong and keycloak shall be used to reinforce the security system, and Docker Compose will be used to build and connect all related containers.

At the end, the following containers are built and managed via a docker-compose.yml file:

* Container for each service of the reprocessing preparation package.
* Postgresql Database container for each service of the reprocessing preparation package.
* Kong container
* Keycloak container
* Nginx container (this container can be omitted by using directly the kong one with an addition configuration.)

## Installation and Deployment requirements

The Docker engine is mandatory to be able to run the following installation and deployment process; you should have been installing it in your target machine.

## Build docker images

The first step of the installation process is to build all necessary docker images. The reprocessing preparation package comes with a *build\_all.bash* script which allows doing so.

* *./build\_all.bash ( will build docker images for all services)*

**Kong Dockerfile**:

To be able to establish a secure connection between kong and Keycloak , a Kong OpenId connect plugin should be available in the kong docker image, to do so , the following dockerfile will be used to generate a final kong image based on the official one :

*FROM kong:2.3.3-alpine*

*LABEL description="Alpine + Kong 2.3.3 + kong-oidc plugin"*

*USER root*

*RUN apk update && apk add git unzip luarocks*

*RUN luarocks install kong-oidc*

*RUN luarocks install lua-resty-openidc 1.7.4 --force*

*USER kong*

**Keycloak and postgres images:**

Official images available in the Dockerhub will be used.

## Deployment as docker-compose applications

The operational reprocessing preparation package comes with a compose component which contains the docker-compose file (see it in appendices) of the package for all services and a configuration data folder for the initial configuration of kong, keycloak and nginx containers.

Deploy the reprocessing preparation package services using:

* *cd compose/kongkeycloak*
* *docker-compose --env-file path\_to\_env\_file up up -d*

The docker-compose contains environment variables to passwords that will have to be fixed, these passwords can be retrieved from an environment file to be passed as *--env-file option* when launching docker-compose up. (See an example in appendices)

# APPENDICES

## AUXIP POM file

[https://github.com/reprocessing-reference/preparation/blob/release-0.0.3/AUXIP\_OLINGO/pom.xm*l*](https://github.com/reprocessing-reference/preparation/blob/release-0.0.3/AUXIP_OLINGO/pom.xml)

## AUXIP docker-compose file

[*https://github.com/reprocessing-reference/preparation/blob/release-0.0.3/compose/kongkeycloak/docker-compose.yml*](https://github.com/reprocessing-reference/preparation/blob/release-0.0.3/compose/kongkeycloak/docker-compose.yml)

## Environments file

*KEYCLOAK\_PASSWORD="\*\*\*KEYCLOAK\_PASSWORD\*\*\*"*

*REPROBASE\_POSTGRES\_PASSWORD="\*\*POSTGRES\_PASSWORD\*\*"*

*AUXIP\_POSTGRES\_PASSWORD="\*\*AUXIP\_POSTGRES\_PASSWORD\*\*"*

*PGADMIN\_DEFAULT\_PASSWORD="\*\* PGADMIN\_DEFAULT\_PASSWORD\*\*"*

*KEYCLOAK\_POSTGRES\_PASSWORD="\*\*KEYCLOAK\_POSTGRES\_PASSWORD\*\*"*

*AUXIP\_S3\_ACCESS\_KEY=JC6XSDLKDPZ1MGLDNTD*

*AUXIP\_S3\_SECRET\_KEY=7lmKSLSKS1oqY4WMmAVmfrQVKSGHSeqGJFT4y1*

*ROLLING\_S3\_ACCESS\_KEY=HQWODLDJDMWOPK687JDT1JA*

*ROLLING\_S3\_SECRET\_KEY=Bu6btWhY4emc0HDldkdbsZDbtJFG0AwWtytHFis*

*REPROCESSINGDATABASELINE\_POSTGRES\_PASSWORD="\*\* REPROCESSINGDATABASELINE\*\*"*