|  |
| --- |
| **Title** |
| SD-10-2100-1 S1PRO Installation Manual IM V3.32.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prepared (Author)** | | | | |
|  | **Signature :** |  | **Date :** |  |
|  | **Name / Function** | **S1PPRO Development Team** |  |  |
|  |  |  |  |  |
|  | | | | |
| **Approved** | | | | |
|  | **Signature :** |  | **Date :** |  |
|  | **Name / Function** | **Torben KESSLER *S1PRO Product Owner*** |  |  |
|  |  |  |  |  |
|  | | | | |
| **Released** | | | | |
|  | **Signature :** |  | **Date :** |  |
|  | **Name / Function** | **Faisal RAFI**  *S1PRO Project Manager* |  |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Document type | Nb WBS | Keywords |
| IM | 2100 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CMMI-SVC | SGN |  |  |

|  |
| --- |
| SUMMARY |

This document aims to provide the installation procedure of the cloud environment of Sentinel-1 Payload Data Ground Segment (S1 PDGS) project

|  |
| --- |
| CHANGE LOG |

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue/Revision** | **Date** | **Modified pages** | **Observations** |
| 1/0 | 12 Feb 2018 | All | Initial version |
| 1/1 | 27 Feb 2018 | All | Sprint 2 version |
| 1/2 | 28 Mar 2018 | All | Sprint 3 version |
| 1/3 | 16 Apr 2018 | All | Sprint 4 version |
| 1/5 | 07 Jun 2018 | All | Sprint 5 version |
| 1/6 | 03 Jul 2018 | All | Sprint 7 version |
| 1/7 | 30 Jul 2018 | All | Sprint 8 version |
| 1/8 | 04 Aug 2018 | All | Sprint 9 version |
| 1/9 | 18 Sep 2018 | All | Sprint 10 version |
| 2/0 | 08 Nov 2018 | All | Finalization of document |
| 2/1 | 25 Jun 2019 | All | V0.4.0 Adding installation procedure for request repository and compression service, L2 Wrapper, L2 Job Generator,L2 scaler |
| 03 | 01 Oct 2019 | All | Updating installation procedure for S1PRO-195, S1PRO-178 |
| 04 | 25 Oct 2019 | All | Updating installation procedure |
| 05 | 13 Nov 2019 | All | IM to be managed in confluence S1PRO0-764 |
| 06 | 16 Nov 2019 | ANNEX1 , Cluster Configuration | S1PRO-935, S1PRO-956 |
| 07 | 15 Apr 2020 | Front Page | V1 Delivery |
| 08 | 12 May 2020 |  | S1PRO-1407 |
| 09 | 22 Jul 2020 | Cluster Initialisation |  |
| 10 | 07 Aug 2020 | Pre-Requirements, Cluster Configuration | Pre-Requirements and installation procedure for S1PRO-1661 |
| 11 | 21 Sep 2020 | Insallation Verification | S1PRO-1765 , S1PRO-1774 |
| 12 | 16 Oct 2020 | Insallation Verification,  Cluster Configuration | Updated for AUXIP interface implementation |
| 13 | 28 Oct 2020 | Cluster Configuration,  Installation Verification | Updated for EDIP interface implementation |
| 14 | 28 Jan 2021 | Cluster Configuration | Added section for Elasticsearch and Kafka configuration |
| 15 | 09 Mar 2021 | Installation Verification | Added Information for MyOcean Cleaner |
| 15 | 09 Mar 2021 |  | V3.30.0 Version |
| 16 | 20 May 2021 | Cluster Initialisation  Installation Verification | Added new QCSS Pods |
| 17 | 08 Jun 2021 | ANNEX 1 | Updated EDIP configuration and added EDIP Mocks |
| 18 | 19 Jul 2021 |  | V3.32.0 Version |
| 19 | 31 Aug 2021 | Cluster Configuration,  Cluster initialisation,  Installation Verification | Removed sections about manual secret creation, added sections for s3.sh, mongo.sh and deploy.sh in Cluster Configuration  Updated component lists in Cluster Initialisation and Installation Verification |

|  |
| --- |
| TABLE OF CONTENTS |

[SUMMARY 2](#_Toc256000000)

[CHANGE LOG 3](#_Toc256000001)

[TABLE OF CONTENTS 5](#_Toc256000002)

[LIST OF FIGURES 8](#_Toc256000003)

[LIST OF TABLES 8](#_Toc256000004)

[1 Introduction 9](#_Toc256000005)

[1.1 SCOPE OF THE DOCUMENT 9](#_Toc256000006)

[1.2 PRESENTATION OF THE DOCUMENT STRUCTURE AND CONTENTS 9](#_Toc256000007)

[2 Applicable and reference documents, acronyms 10](#_Toc256000008)

[2.1 APPLICABLE AND REFERENCE DOCUMENTS 10](#_Toc256000009)

[2.1.1 Applicable documents 10](#_Toc256000010)

[2.1.2 Reference documents 10](#_Toc256000011)

[2.2 ACRONYMS 10](#_Toc256000012)

[3 Pre-Requirements 11](#_Toc256000013)

[3.1 Kubernetes 11](#_Toc256000014)

[3.1.1 Specific nodes 11](#_Toc256000015)

[3.2 Kafka & Zookeeper 11](#_Toc256000016)

[3.3 MongoDB 12](#_Toc256000017)

[3.4 PostgreSQL 12](#_Toc256000018)

[3.5 Elasticsearch 12](#_Toc256000019)

[3.6 Object Storage 12](#_Toc256000020)

[3.7 Master server 12](#_Toc256000021)

[3.8 Helm 12](#_Toc256000022)

[3.9 S3CMD 13](#_Toc256000023)

[3.10 CFI 13](#_Toc256000024)

[4 Cluster Configuration 14](#_Toc256000025)

[4.1 OBS Configuration 14](#_Toc256000026)

[4.2 Configuration For S1pro Scripts 14](#_Toc256000027)

[4.2.1 environment-specific.sh 14](#_Toc256000028)

[4.2.2 s3.sh 17](#_Toc256000029)

[4.2.3 kafka.sh 20](#_Toc256000030)

[4.2.4 mongo.sh 27](#_Toc256000031)

[4.2.5 deploy.sh 27](#_Toc256000032)

[5 Cluster Initialisation 31](#_Toc256000033)

[5.1 DEPLOYMENT PREPARATION 31](#_Toc256000034)

[5.2 Pod Initialisation 33](#_Toc256000035)

[5.2.1 Contingency Ingestion 33](#_Toc256000036)

[6 6 Patch Deployment 34](#_Toc256000037)

[6.1 deploy script 34](#_Toc256000038)

[7 Verification of Deployment on Kubernetes 36](#_Toc256000039)

[7.1 Automatic Verification of Deployment on Kubernetes 36](#_Toc256000040)

[7.2 Manual Verification of Deployment on Kubernetes 36](#_Toc256000041)

[7.2.1 Applicative catalog 36](#_Toc256000042)

[7.2.2 Metadata catalog 36](#_Toc256000043)

[7.2.3 XBIP 36](#_Toc256000044)

[7.2.4 AUXIP 37](#_Toc256000045)

[7.2.5 EDIP 37](#_Toc256000046)

[7.2.6 Ingestion 37](#_Toc256000047)

[7.2.7 Plan and Report Ingestion 38](#_Toc256000048)

[7.2.8 Compression 38](#_Toc256000049)

[7.2.9 Request Repository 38](#_Toc256000050)

[7.2.10 Queue Watcher 38](#_Toc256000051)

[7.2.11 SPP MBU 38](#_Toc256000052)

[7.2.12 SPP OBS 39](#_Toc256000053)

[7.2.13 AIO 39](#_Toc256000054)

[7.2.14 ASP 39](#_Toc256000055)

[7.2.15 L1 40](#_Toc256000056)

[7.2.16 L2 40](#_Toc256000057)

[7.2.17 PRIP 40](#_Toc256000058)

[7.2.18 Contingency Ingestion 40](#_Toc256000059)

[7.2.19 Dissemination 40](#_Toc256000060)

[7.2.20 Validation 41](#_Toc256000061)

[7.2.21 On-Demand Interface Provider 41](#_Toc256000062)

[7.2.22 Request Respository 41](#_Toc256000063)

[7.2.23 Report 41](#_Toc256000064)

[7.2.24 MyOcean Service 42](#_Toc256000065)

[7.2.25 Data Lifecycle Services 42](#_Toc256000066)

[7.2.26 QCSS Pods 43](#_Toc256000067)

[7.2.27 Prometheus Adapter 43](#_Toc256000068)

[8 Annex 1 44](#_Toc256000069)

[8.1 Values.yaml 44](#_Toc256000070)

[8.2 ReplicaCount.yaml 52](#_Toc256000071)

[9 Annex 2 55](#_Toc256000072)

[9.1 Keystore 55](#_Toc256000073)

[9.1.1 Private key and certificate creation 55](#_Toc256000074)

[9.1.2 Private key and certificate import 55](#_Toc256000075)

[9.2 Truststore 56](#_Toc256000076)

[9.3 Encoding Key-/Truststores as Base64 56](#_Toc256000077)

[DISTRIBUTION LIST 57](#_Toc256000078)

**LIST OF FIGURES**

**No table of figures entries found.**

**LIST OF** **TABLES**

[Table 1 Document Structure 9](#_Toc256000079)

[Table 2 Applicable Documents 10](#_Toc256000080)

[Table 3 Reference Documents 10](#_Toc256000081)

[Table 4 s3.sh 19](#_Toc256000082)

[Table 5 kafka.sh 23](#_Toc256000083)

[Table 6 es.sh 27](#_Toc256000084)

[Table 7 mongo.sh 27](#_Toc256000085)

[Table 8 deploy.sh 30](#_Toc256000086)

# Introduction

## SCOPE OF THE DOCUMENT

This document aims to provide the installation procedure for deploying the Sentinel-1 PDGS in cloud environment.

## PRESENTATION OF THE DOCUMENT STRUCTURE AND CONTENTS

This document is structured as follow Chapter number Title Objectives and contents

|  |  |  |
| --- | --- | --- |
| **Chapter number** | **Title** | **Objectives and contents** |
| 1 | INTRODUCTION | This chapter presents the scope of the document and introduces its contents, in order to provide the reader with a road map into the document |
| 2 | APPLICABLE AND REFERENCE DOCUMENTS, ACRONYMS | This chapter contains references to the project documents and used acronyms. |
| 3 | PRE-REQUIREMENTS | This chapter contains the requirements on the K8S architecture and the steps to proceed for preparing the installation |
| 4 | CLUSTER CONFIGURATION | This chapter contains S1PRO related configuration needs to performed for the installation. |
| 5 | CLUSTER INITIALISATION | This chapter describes the steps for deploying the S1PRO software on the K8S . |
| 6 | PATCH DEPLOYMENT | This chapter describes the steps for deploying a S1PRO Patch Delivery. |
| 7 | INSTALLATION VERIFICATION | This chapter contains the information regarding verification of  S1PRO related cluster installation. |
| Annex 1 | ENVIRONMENT SPECIFIC CONFIGURATION | This annex contains the details of environment specific configuration |
| Annex 2 | KEY AND TRUSTSTORE CONFIGURATION | This annex describes the details for configuration of Keystore and Trustore |

Table 1 Document Structure

# Applicable and reference documents, acronyms

## APPLICABLE AND REFERENCE DOCUMENTS

### Applicable documents

Applicable documents are the documents that have been necessary formal inputs to the writing of this document.

|  |  |
| --- | --- |
| **Document ID** | **Document Title** |
| ECSS-M‐ST‐40C | Configuration and information management |
| S1PD-CR-ADST- 1000530589 | S1PD-CR-ADST-1000530589 |

Table 2 Applicable Documents

### Reference documents

|  |  |
| --- | --- |
| **Document Reference** | **Document Title** |
| SD-10-2100-2 | S1PRO User Manual |
| SD-133-2100 | S1 Core PDGS Glossary |

Table 3 Reference Documents

## ACRONYMS

The specific terminology and acronyms in this document are defined in SD-133-2100 S1PRO Glossary.

# Pre-Requirements

For setting up the initial structure of the cluster for the processing chain a script is provided. It will automatically deploy a set of helm packages in the cluster to ensure that everything is been installed and also handling the initial configuration of the services. However before this script can be executed it is required to have a set of preparations that needs to be addressed before starting with the automatic installation.

These prerequisites will be handled in the next sections.

## Kubernetes

It is expected that the Kubernetes cluster had been setup already on the worker servers as well as having a correctly configured kubernetes client on the installation host (e.g. the bastion server). The minimal version of Kubernetes that expected to operate the environment is v1.8.7.

### Specific nodes

#### L0, L1 and L2 nodes

The nodes dedicated to execution of L0 Wrapper, L1 Wrapper and L2 wrapper must be configured with the system disk on a SSD disk. These machines must be identified with a specific label.

|  |
| --- |
| # kubectl label nodes <node> wrapperconfig=l0 (respectively l1 and L2)  where node = name of the node dedicated to L0 (respectively L1 and L2) |

To be persistent, the file “/etc/systemd/system/kubelet.service.d/10-kubeadm.conf”. This file must be modified by adding “--node-labels=wrapperconfig=l0,wrapperstate=used” (respectively l1 and l2) at the KUBELET\_EXTRA\_ARGS.

#### Compression nodes

The nodes dedicated to execution of the compression activities must be labels as well in order to allow the compression service to be deployed upon them. These machines must be identified with a specific label.

|  |
| --- |
| # kubectl label nodes <node> wrapperconfig=zip  where node = name of the node dedicated to compression |

To be persistent, the file “/etc/systemd/system/kubelet.service.d/10-kubeadm.conf”. The file must be modified by adding “--node-labels=wrapperconfig=zip”) at the KUBELET\_EXTRA\_ARGS.

## Kafka & Zookeeper

It is expected that KAFKA and Zookeeper have been already installed, up and running. The minimal version of Kafka is 2.11-0.10.2.0 and the minimal version of Zookeeper is 3.4.10.

It is mandatory that Kafka needs to forbid the automatic creation of topics by setting the parameter "auto.create.topics.enabled" to "false".

By default it is expected that Kafka is available as pod "kafka-0" and available as service "kafka-headless" under port 9092. If these expectations are not matched, it is required to change them in the environment specific configuration accordingly. Further information regarding this subject can be found in the Annex.

It is expected that Kafka and Zookeeper pods are deployed and running in the cluster already. These packages are maintained and provided by S1MIS and it is expected that their pods had been deployed to the environment before the automatic installation is executed. Please be sure that the pods are deployed and at least one  pod exists in the cluster and having installed "kafka-topics.sh" script that will be called by S1PRO cluster deployment script in order to initialize the topics.

## MongoDB

It is expected that MongoDB has been already installed, up and running within the cluster. The expected version of MongoDB is 4.2.0.

## PostgreSQL

It is expected that PosgreSQL has been already installed, up and running within the cluster.

## Elasticsearch

It is expected that Elasticsearch has been already installed, up and running within the cluster. The expected version of ElasticSearch is 6.5.4. The initialization of the database is done during the automatic installation. By default it is expected that ElasticSearch is available as pod defined in the environmental variable ELASTICSEARCH\_POD and the service is available under ELASTICSEARCH\_URL (e.g. elasticsearch-processing-trace-elasticsearch-coordinating-only:9200). If this is the case, there are no further configuration required. Further information regarding this can be found in the Annex.

## Object Storage

It is expected that an object storage compatible with the S3 API is already provided and accessible within the cluster. When using the Cloud Orange environment this can be easily setup within the technical console. In this case, you just need to specify the endpoint of the object store as well as the credentials.

## Master server

In order to install the configuration correctly, it is required to have access to the Kubernetes Master server or any other system in the same Cluster that does have access to the cluster via the kubernetes client. The minimal version of the Kubernetes client is v1.8.5. The system must be able to perform certain kubectl commands in the cluster in order to execute commands within some pods.

## Helm

It is expected that the cluster does contain a helm installation with a tiller service running in the namespace "kube-system". It is expected that the helm client can be used to deploy local packages into the cluster successfully.

## S3CMD

It is assumed that the command line tool "s3cmd" is available within the environment in order to access the Object Storage. This can be done by installing the tool "s3cmd" via yum and setting up the login credentials from orange cloud in the file .s3cfg.

## CFI

Following IPF are delivered as CFI and integrated in S1PRO.

|  |  |
| --- | --- |
| **Processor** | **Version** |
| L0 AIO IPF | 5.1.0 |
| L0 ASP IPF | 5.1.0 |
| L1 IPF | 3.4.0 |
| L2 IPF | 3.4.0 |
| MBU | 2.0.0 |
| OBS | 1.11.0 |

# Cluster Configuration

## OBS Configuration

The OBS needs to be configured in the file "configuration-s3.sh" where an endpoint configuration must be provied.

Please setup the variables "storage\_endpoint" to the location of your API and select with "storage\_region" the region it is located.

The related credentials will be stored in a secret. For this, the credentials need to be configured in the file "environment-specific.sh".

Please note also that is not required to create the buckets manually anymore as they will be automatically setup during the automatic installation. If you want to modify the names of the buckets that are created during the installation, this can be done in the file "configuration-s3.sh" as well.

To use OVH S3 object storage, you have to change configuration/template-s3.sh as follows:

|  |
| --- |
| export S3\_ENDPOINT="http://storage.gra.cloud.ovh.net";  export S3\_REGION="gra";  export S3\_DISABLE\_CHUNKED\_ENCODING="true"; |

By default these configuration are set for Orange environment.

|  |
| --- |
| export S3\_ENDPOINT="http://oss.eu-west-0.prod-cloud-ocb.orange-business.com/";  export S3\_REGION="eu-west-0";  export S3\_DISABLE\_CHUNKED\_ENCODING="false"; |

## Configuration For S1pro Scripts

### environment-specific.sh

In the environment specific configuration folder (derived from env-template-config repository) , open the file "environment-specific.sh" and adjust the following environment variables according to cluster.

|  |
| --- |
| #!/bin/sh  #############################################  ## PRIVATE DATA  #############################################  CLEAR\_DATA=$(gpg --batch --passphrase-file ~/gpg\_pass\_werum -d $(dirname ${BASH\_SOURCE})/**private**.gpg);  $CLEAR\_DATA    #############################################  ## MISCELEANOUS  #############################################  export NAMESPACE\_PROCESSING="processing";    export DOCKER\_REGISTRY\_SECRET\_NAME="harbor-tools"  export DOCKER\_REGISTRY\_ADRESS="registry.tools.s1pdgs.eu"  #export DOCKER\_REGISTRY\_USER="git\_ivv";  export DOCKER\_REGISTRY\_USER="ops\_harbor";    export HELM\_REPO\_NAME="local"  export HELM\_REPO\_URL=""  #export HELM\_REPO\_NAME="werum\_harbor"  #export HELM\_REPO\_URL="https://registry.tools.s1pdgs.eu/chartrepo/werum"  export HELM\_REPO\_USER="git\_ivv";    export OPENSTACK\_SECRET\_NAME="openstack";  export OPENSTACK\_USER=$( sudo cat /mnt/s1pdgs/k8s/cloud-config | grep 'username' | head -n 1 | cut -d '=' -f 2 | sed 's/"//g' | sed 's/ //g');  export OPENSTACK\_PASS=$( sudo cat /mnt/s1pdgs/k8s/cloud-config | grep 'password' | head -n 1 | cut -d '=' -f 2 | sed 's/"//g' | sed 's/ //g');    export S3\_SECRET\_NAME="obs";  export S3\_ACCESS\_KEY=$( cat ~/.s3cfg | grep access\_key | cut -d '=' -f 2 | sed 's/ //g');  export S3\_SECRET\_KEY=$( cat ~/.s3cfg | grep secret\_key | cut -d '=' -f 2 | sed 's/ //g');    export MONGODB\_SECRET\_NAME="mongodb";  export MONGODB\_ROOT\_USER="root";  export MONGODB\_ROOT\_PASS=$(kubectl get secret -n infra mongodb -o json | jq -r '.data."mongodb-root-password"' | base64 -d);  export MONGODB\_USER="s1pdgs";    export AUXIP\_SECRET\_NAME="auxip";  export AUXIP\_USER="s1pdgs1";    export EDIP\_PEDC\_SECRET\_NAME="edip-pedc";  export EDIP\_PEDC\_USER="esaclient01";    export EDIP\_BEDC\_SECRET\_NAME="edip-bedc";  export EDIP\_BEDC\_USER="esaclient01";    export QCSS\_SECRET\_NAME="qcss";  export QCSS\_USER="s1pdgs";  export QCSS\_PASS="XXXXX"    export XBIP\_01\_SECRET\_NAME="xbip-cgs01";  export XBIP\_02\_SECRET\_NAME="xbip-cgs02";  export XBIP\_03\_SECRET\_NAME="xbip-cgs03";  export XBIP\_04\_SECRET\_NAME="xbip-cgs04";  export XBIP\_05\_SECRET\_NAME="xbip-cgs05";  export XBIP\_10\_SECRET\_NAME="xbip-cgs10";  export XBIP\_01\_USER="s1pdgs";  export XBIP\_02\_USER="sentinel1";  export XBIP\_03\_USER="s1pdgs\_intaddp";  export XBIP\_04\_USER="esacopas";  export XBIP\_10\_USER="xbip\_airbus";    export AMALFI\_SECRET\_NAME="amalfi";  export AMALFI\_DB\_URL="postgresql-quality-pgpool.processing.svc.cluster.local";  export AMALFI\_DB\_USER="amalfi";    export KONGPLUGIN\_JSON=$(kubectl -n infra get kongplugin kong-oidc-plugin -o json | jq -r '.metadata.annotations."kubectl.kubernetes.io/last-applied-configuration"' | sed 's/\\"/"/g')    export KEYCLOAK\_OIDC\_CLIENT\_NAME=$( echo ${KONGPLUGIN\_JSON} | jq -r '.config.client\_id');  export KEYCLOAK\_OIDC\_CLIENT\_SECRET=$( echo ${KONGPLUGIN\_JSON} | jq -r '.config.client\_secret');  export KEYCLOAK\_OIDC\_DISCOVERY\_URL=$( echo ${KONGPLUGIN\_JSON} | jq -r '.config.discovery');  export KEYCLOAK\_OIDC\_SESSION\_SECRET=$( echo ${KONGPLUGIN\_JSON} | jq -r '.config.session\_secret');  export KEYCLOAK\_OIDC\_IPA\_GROUP\_ALLOWED=$( echo ${KONGPLUGIN\_JSON} | jq -r '.config.groups\_authorized\_paths[0].group\_name');    export KEYCLOAK\_OIDC\_PERMISSIONS=" - group\_authorized\_paths:  - /odata  group\_name: operations  - group\_authorized\_paths:  - /odata  group\_name: centreexpert  - group\_authorized\_paths:  - /odata  group\_name: op\_manager  - group\_authorized\_paths:  - /  group\_name: sysadmin  - group\_authorized\_paths:  - /odata  group\_name: customer  - group\_authorized\_paths:  - /odata  group\_name: b2b"    #############################################  ## OPENSTACK  #############################################  export OS\_ENDPOINT="https://iam.eu-west-0.prod-cloud-ocb.orange-business.com/v3";  export OS\_DOMAIN\_ID="XXXXX";  export OS\_PROJECT\_ID="XXXXX";  export OS\_USERNAME="XXXXX";  export OS\_PASSWORD="XXXXX";  export OS\_AZ="eu-west-0a";  export OS\_NETWORK="XXXXX";  export OS\_SECURITY\_GROUP="XXXXX";  export OS\_FLOATING\_NETWORK="XXXXXXXXXXX";    #############################################  ## K8S  #############################################  export K8S\_MASTER=$(cat ~/.kube/config | grep server: | awk '{print $2}');  export K8S\_NAMESPACE="default";  export K8S\_USERNAME="kubernetes-admin";  export K8S\_CLIENT\_KEY=$(cat ~/.kube/config | grep client-key-data: | awk '{print $2}')  export K8S\_CLIENT\_CERT\_DATA=$(cat ~/.kube/config | grep client-certificate-data: | awk '{print $2}');    #############################################  ## Debug  #############################################  export LOG4J\_CONFIG="log/log4j2.yml"    #############################################  ## Werum  #############################################  export WAITING\_FROM\_INGESTION\_IN\_SECONDS="1800"    export WERUM\_USE\_MOCK\_WEBDAV="false"  export WERUM\_USE\_MOCK\_DISSEMINATION="true"    export WERUM\_IGNORE\_INGESTION\_PV="false";    export WERUM\_IGNORE\_SECRETS="false";  export WERUM\_IGNORE\_DEPLOY="false";    export WERUM\_IGNORE\_S3="true";  export WERUM\_IGNORE\_ES="false";  export WERUM\_IGNORE\_MONGO="false";  export WERUM\_IGNORE\_KAFKA="true";    export WERUM\_FAST\_KAFKA\_INIT="false";  export WERUM\_SKIP\_KAFKA\_CLEAN="false";    #############################################  ## END  ############################################# |

### s3.sh

The s3.sh script contains the Amazon S3 object storage relevant configuration. For example the OBS connection properties and the bucket names to be used.

|  |
| --- |
| export S3\_ENDPOINT="http://oss.eu-west-0.prod-cloud-ocb.orange-business.com";  export S3\_REGION="eu-west-0";  export S3\_DISABLE\_CHUNKED\_ENCODING="false"  export S3\_CONFIGURATION\_FILE\_PATH="/share/.s3cfg"    export S3\_CLEAN\_HARD="rb --force --recursive";  export S3\_CLEAN\_SOFT="rm -r --force --recursive";  export S3\_CLEAN="${S3\_CLEAN\_HARD}";    export S3\_PREFIX="ops-c1"; # ex: "ivv-c2"  export S3\_SUFFIX\_ZIP="zip";  export S3\_EXCEPTION\_CLEAN="nothing"; # Example: "session|auxiliary". Set to "nothing" if no exception required (empty string will fail)  TMP\_S3\_EXCEPTION\_RESET="";  # Example: "bucket\_name:filter1|filter2|...". 1 bucket per line. No line if no filter.  # "${S3\_PREFIX}-auxiliary-files:auxiliary-files  # ${S3\_PREFIX}-auxiliary-files-zip:auxiliary-files  # ${S3\_PREFIX}-session-files:raw"    export S3\_BUCKET\_AUX="${S3\_PREFIX}-auxiliary-files";  export S3\_BUCKET\_SESSIONS="${S3\_PREFIX}-session-files";    export S3\_BUCKET\_L0\_SEGMENTS="${S3\_PREFIX}-l0-segments";  export S3\_BUCKET\_L0\_SLICES="${S3\_PREFIX}-l0-slices";  export S3\_BUCKET\_L0\_ACNS="${S3\_PREFIX}-l0-acns";    export S3\_BUCKET\_L1\_SLICES="${S3\_PREFIX}-l1-slices";  export S3\_BUCKET\_L1\_ACNS="${S3\_PREFIX}-l1-acns";  export S3\_BUCKET\_L2\_SLICES="${S3\_PREFIX}-l2-slices";  export S3\_BUCKET\_L2\_ACNS="${S3\_PREFIX}-l2-acns";    export S3\_BUCKET\_L0\_BLANKS="${S3\_PREFIX}-l0-blanks";  export S3\_BUCKET\_SPP="${S3\_PREFIX}-spp";  export S3\_BUCKET\_SPP\_MBU="${S3\_PREFIX}-spp-mbu";  export S3\_BUCKET\_INVALID="${S3\_PREFIX}-invalid";  export S3\_BUCKET\_GHOST="${S3\_PREFIX}-ghost";  export S3\_BUCKET\_DEBUG="${S3\_PREFIX}-debug";  export S3\_BUCKET\_FAILED\_WORKDIR="${S3\_PREFIX}-failed-workdir";  export S3\_BUCKET\_SESSION\_RETRANSFER="${S3\_PREFIX}-session-retransfer";  export S3\_BUCKET\_PLANS\_AND\_REPORTS="${S3\_PREFIX}-plans-and-reports";    export S3\_BUCKET\_ZIP\_AUX="${S3\_BUCKET\_AUX}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_L0\_SEGMENTS="${S3\_BUCKET\_L0\_SEGMENTS}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_L0\_SLICES="${S3\_BUCKET\_L0\_SLICES}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_L0\_ACNS="${S3\_BUCKET\_L0\_ACNS}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_L1\_SLICES="${S3\_BUCKET\_L1\_SLICES}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_L1\_ACNS="${S3\_BUCKET\_L1\_ACNS}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_L2\_SLICES="${S3\_BUCKET\_L2\_SLICES}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_L2\_ACNS="${S3\_BUCKET\_L2\_ACNS}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_L0\_BLANKS="${S3\_BUCKET\_L0\_BLANKS}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_SPP="${S3\_BUCKET\_SPP}-${S3\_SUFFIX\_ZIP}";  export S3\_BUCKET\_ZIP\_PLANS\_AND\_REPORTS="${S3\_BUCKET\_PLANS\_AND\_REPORTS}-${S3\_SUFFIX\_ZIP}";    TMP\_BUCKETS\_INPUT="  ${S3\_BUCKET\_AUX}  ${S3\_BUCKET\_SESSIONS}  ${S3\_BUCKET\_PLANS\_AND\_REPORTS}  ${S3\_BUCKET\_SESSION\_RETRANSFER}  ";    TMP\_BUCKETS\_PRODUCED="  ${S3\_BUCKET\_L0\_SEGMENTS}  ${S3\_BUCKET\_L0\_SLICES}  ${S3\_BUCKET\_L0\_ACNS}    ${S3\_BUCKET\_L1\_SLICES}  ${S3\_BUCKET\_L1\_ACNS}  ${S3\_BUCKET\_L2\_SLICES}  ${S3\_BUCKET\_L2\_ACNS}    ${S3\_BUCKET\_L0\_BLANKS}  ${S3\_BUCKET\_SPP}  ${S3\_BUCKET\_SPP\_MBU}  ${S3\_BUCKET\_INVALID}  ${S3\_BUCKET\_GHOST}    ${S3\_BUCKET\_DEBUG}  ${S3\_BUCKET\_FAILED\_WORKDIR}  ";    TMP\_BUCKETS\_COMPRESSED="  ${S3\_BUCKET\_ZIP\_AUX}  ${S3\_BUCKET\_ZIP\_L0\_SEGMENTS}  ${S3\_BUCKET\_ZIP\_L0\_SLICES}  ${S3\_BUCKET\_ZIP\_L0\_ACNS}  ${S3\_BUCKET\_ZIP\_L1\_SLICES}  ${S3\_BUCKET\_ZIP\_L1\_ACNS}  ${S3\_BUCKET\_ZIP\_L2\_SLICES}  ${S3\_BUCKET\_ZIP\_L2\_ACNS}  ${S3\_BUCKET\_ZIP\_L0\_BLANKS}  ${S3\_BUCKET\_ZIP\_SPP}  ${S3\_BUCKET\_ZIP\_PLANS\_AND\_REPORTS}  ";    TMP\_BUCKETS\_ALL="  ${TMP\_BUCKETS\_INPUT}  ${TMP\_BUCKETS\_PRODUCED}  ${TMP\_BUCKETS\_COMPRESSED}  ";    # Delete empty lines  export BUCKETS\_INPUT=$( echo "${TMP\_BUCKETS\_INPUT}" | grep -v -e '^$');  export BUCKETS\_PRODUCED=$( echo "${TMP\_BUCKETS\_PRODUCED}" | grep -v -e '^$');  export BUCKETS\_COMPRESSED=$(echo "${TMP\_BUCKETS\_COMPRESSED}" | grep -v -e '^$');  export BUCKETS\_ALL=$( echo "${TMP\_BUCKETS\_ALL}" | grep -v -e '^$');  export S3\_EXCEPTION\_RESET=$(echo "${TMP\_S3\_EXCEPTION\_RESET}" | grep -v -e '^$'); |

Table 4 s3.sh

### kafka.sh

The kafka.sh script contains kafka relevant configuration. For example the kafka cluster connection properties and the topics to be initialized when the cluster is initialized.

|  |
| --- |
| export KAFKA\_NAMESPACE="infra";  export KAFKA\_POD\_PREFIX="kafka";  export KAFKA\_POD=$(kubectl -n ${KAFKA\_NAMESPACE} get pods | grep ${KAFKA\_POD\_PREFIX} | grep Running | head -1 | awk '{print $1}');  export KAFKA\_CONTAINER="kafka";  export KAFKA\_SCRIPT\_TOPIC="/opt/kafka/bin/kafka-topics.sh";  export KAFKA\_SCRIPT\_CONFIG="/opt/kafka/bin/kafka-configs.sh";  export KAFKA\_RUN\_CLASS\_SCRIPT\_TOPIC="/opt/kafka/bin/kafka-run-class.sh";  export KAFKA\_URL="kafka-headless.${KAFKA\_NAMESPACE}.svc.cluster.local:9092";  export ZOOKEEPER\_LIST="zookeeper-0.zookeeper-headless.${KAFKA\_NAMESPACE}.svc.cluster.local:2181,zookeeper-1.zookeeper-headless.${KAFKA\_NAMESPACE}.svc.cluster.local:2181,zookeeper-2.zookeeper-headless.${KAFKA\_NAMESPACE}.svc.cluster.local:2181"  export KAFKA\_TOPIC\_ERROR="t-pdgs-errors";    # Set KAFKA\_EXCEPTION if you want to select a subset of topics (regex) to be NOT handled by the cluster configuration script (eg. modules/kafka.sh configuration/configuration.sh init)  # Set to "nothing" if no exception required (empty string will fail)  export KAFKA\_EXCEPTION="nothing"    # Set KAFKA\_FILTER if you want to select only a subset of topics (regex) to be handled by the cluster configuration script (eg. modules/kafka.sh configuration/configuration.sh init)  # Example: export KAFKA\_FILTER="t-pdgs-"  export KAFKA\_FILTER="t-pdgs-"    TMP\_TOPIC\_LIST\_DETAILS="  t-pdgs-ingestion-jobs-s1pro-ingestion-0 2 259200000  t-pdgs-ingestion-jobs-s1pro-ingestion-1 2 259200000  t-pdgs-ingestion-jobs-s1pro-ingestion-2 2 259200000  t-pdgs-ingestion-jobs-s1pro-ingestion-3 2 259200000  t-pdgs-ingestion-jobs-s1pro-ingestion-4 2 259200000  t-pdgs-ingestion-jobs-s1pro-ingestion-5 2 259200000  t-pdgs-ingestion-jobs-s1pro-ingestion-6 2 259200000  t-pdgs-ingestion-jobs-s1pro-ingestion-7 2 259200000  t-pdgs-ingestion-jobs-s1pro-ingestion-8 2 259200000  t-pdgs-ingestion-jobs-s1pro-ingestion-9 2 259200000  t-pdgs-ingestion-jobs-xbip-nominal 20 259200000  t-pdgs-ingestion-jobs-xbip-retransfer 20 259200000  t-pdgs-ingestion-jobs-auxip-nominal 10 259200000  t-pdgs-ingestion-jobs-edip-pedc-nominal 6 259200000  t-pdgs-ingestion-jobs-edip-pedc-retransfer 6 259200000  t-pdgs-ingestion-jobs-edip-bedc-nominal 6 259200000  t-pdgs-ingestion-jobs-edip-bedc-retransfer 6 259200000  t-pdgs-contingency-ingestion-jobs 2 259200000  t-pdgs-plan-and-report-ingestion-jobs 2 259200000  t-pdgs-qcss-ingestion-jobs 2 259200000  t-pdgs-qcss-ingestion-zip-jobs 2 259200000  t-pdgs-aux-ingestion-events 2 259200000  t-pdgs-session-file-ingestion-events 2 259200000  t-pdgs-contingency-ingestion-events 2 259200000  t-pdgs-plan-and-report-ingestion-events 2 259200000  t-pdgs-plan-and-report-zip-ingestion-events 2 259200000  t-pdgs-auxip-aux-ingestion-events 2 259200000  t-pdgs-auxip-plan-and-report-ingestion-events 2 259200000  t-pdgs-metadata-extraction-jobs 10 259200000  t-pdgs-session-file-catalog-events 2 259200000  t-pdgs-l0-segment-catalog-events-fast 2 259200000  t-pdgs-l0-segment-catalog-events-nrt 2 259200000  t-pdgs-l0-segment-catalog-events-pt 2 259200000  t-pdgs-l0-segment-catalog-events-operator-demand 2 259200000  t-pdgs-l0-slice-catalog-events-fast 2 259200000  t-pdgs-l0-slice-catalog-events-nrt 2 259200000  t-pdgs-l0-slice-catalog-events-pt 2 259200000  t-pdgs-l0-slice-catalog-events-operator-demand 2 259200000  t-pdgs-plan-and-report-catalog-events 2 259200000  t-pdgs-other-catalog-events 2 259200000  t-pdgs-aio-preparation-jobs 2 259200000  t-pdgs-aio-execution-jobs 6 259200000  t-pdgs-aio-l0-segment-production-events 2 259200000  t-pdgs-aio-l0-blank-production-events 2 259200000  t-pdgs-l0asp-preparation-jobs-fast 2 259200000  t-pdgs-l0asp-preparation-jobs-nrt 2 259200000  t-pdgs-l0asp-preparation-jobs-pt 2 259200000  t-pdgs-l0asp-preparation-jobs-operator-demand 2 259200000  t-pdgs-l0asp-preparation-jobs-late 2 259200000  t-pdgs-l0asp-execution-jobs-fast 6 259200000  t-pdgs-l0asp-execution-jobs-nrt 6 259200000  t-pdgs-l0asp-execution-jobs-pt 6 259200000  t-pdgs-l0asp-execution-jobs-operator-demand 6 259200000  t-pdgs-l0asp-execution-jobs-late 6 259200000  t-pdgs-l0asp-l0-acn-production-events-fast 2 259200000  t-pdgs-l0asp-l0-acn-production-events-nrt 2 259200000  t-pdgs-l0asp-l0-acn-production-events-pt 2 259200000  t-pdgs-l0asp-l0-acn-production-events-operator-demand 2 259200000  t-pdgs-l0asp-l0-acn-production-events-late 2 259200000  t-pdgs-l0asp-l0-slice-production-events-fast 2 259200000  t-pdgs-l0asp-l0-slice-production-events-nrt 2 259200000  t-pdgs-l0asp-l0-slice-production-events-pt 2 259200000  t-pdgs-l0asp-l0-slice-production-events-operator-demand 2 259200000  t-pdgs-l0asp-l0-slice-production-events-late 2 259200000  t-pdgs-l1-preparation-jobs-fast 2 259200000  t-pdgs-l1-preparation-jobs-nrt 2 259200000  t-pdgs-l1-preparation-jobs-pt 2 259200000  t-pdgs-l1-preparation-jobs-operator-demand 2 259200000  t-pdgs-l1-preparation-jobs-late 2 259200000  t-pdgs-l1-execution-jobs-fast 50 259200000  t-pdgs-l1-execution-jobs-nrt 50 259200000  t-pdgs-l1-execution-jobs-pt 50 259200000  t-pdgs-l1-execution-jobs-operator-demand 50 259200000  t-pdgs-l1-execution-jobs-late 50 259200000  t-pdgs-l1-acn-production-events-fast 2 259200000  t-pdgs-l1-acn-production-events-nrt 2 259200000  t-pdgs-l1-acn-production-events-pt 2 259200000  t-pdgs-l1-acn-production-events-operator-demand 2 259200000  t-pdgs-l1-acn-production-events-late 2 259200000  t-pdgs-l1-slices-production-events-fast 2 259200000  t-pdgs-l1-slices-production-events-nrt 2 259200000  t-pdgs-l1-slices-production-events-pt 2 259200000  t-pdgs-l1-slices-production-events-operator-demand 2 259200000  t-pdgs-l1-slices-production-events-late 2 259200000  t-pdgs-l2-preparation-jobs-fast 2 259200000  t-pdgs-l2-preparation-jobs-nrt 2 259200000  t-pdgs-l2-preparation-jobs-pt 2 259200000  t-pdgs-l2-preparation-jobs-operator-demand 2 259200000  t-pdgs-l2-preparation-jobs-late 2 259200000  t-pdgs-l2-execution-jobs-fast 20 259200000  t-pdgs-l2-execution-jobs-nrt 20 259200000  t-pdgs-l2-execution-jobs-pt 20 259200000  t-pdgs-l2-execution-jobs-operator-demand 20 259200000  t-pdgs-l2-execution-jobs-late 20 259200000  t-pdgs-l2-acn-production-events-fast 2 259200000  t-pdgs-l2-acn-production-events-nrt 2 259200000  t-pdgs-l2-acn-production-events-pt 2 259200000  t-pdgs-l2-acn-production-events-operator-demand 2 259200000  t-pdgs-l2-acn-production-events-late 2 259200000  t-pdgs-l2-slices-production-events-fast 2 259200000  t-pdgs-l2-slices-production-events-nrt 2 259200000  t-pdgs-l2-slices-production-events-pt 2 259200000  t-pdgs-l2-slices-production-events-operator-demand 2 259200000  t-pdgs-l2-slices-production-events-late 2 259200000  t-pdgs-obs-preparation-jobs 2 259200000  t-pdgs-obs-execution-jobs 2 259200000  t-pdgs-obs-production-events 2 259200000  t-pdgs-compression-jobs-fast 50 259200000  t-pdgs-compression-jobs-nrt 50 259200000  t-pdgs-compression-jobs-pt 50 259200000  t-pdgs-compression-jobs-operator-demand 40 259200000  t-pdgs-compression-jobs-late 40 259200000  t-pdgs-uncompression-jobs 10 259200000  t-pdgs-uncompression-jobs-pt 10 259200000  t-pdgs-uncompression-jobs-data-request 2 259200000  t-pdgs-compression-events 2 259200000  t-pdgs-uncompression-events-data-request 2 259200000  t-pdgs-publishing-jobs 10 259200000  t-pdgs-errors 2 259200000  t-pdgs-eviction-management-jobs 2 259200000  t-pdgs-eviction-events 2 259200000  t-pdgs-data-request-jobs 2 259200000  t-pdgs-data-request-events 2 259200000  t-pdgs-lta-download-events 2 259200000  t-pdgs-aio-production-report-events 2 259200000  t-pdgs-l0asp-production-report-events 2 259200000  t-pdgs-l1-production-report-events 2 259200000  t-pdgs-l2-production-report-events 2 259200000  t-pdgs-operator-demand-events 10 259200000  t-pdgs-myocean-jobs 2 259200000  t-pdgs-mbu-preparation-jobs 2 259200000  t-pdgs-mbu-execution-jobs 2 259200000  t-pdgs-mbu-production-events 2 259200000  t-pdgs-mbu-dissemination-jobs 2 259200000  ";    # Delete empty lines and choose columns  export TOPIC\_LIST\_DETAILS=$( echo "${TMP\_TOPIC\_LIST\_DETAILS}" | grep -E "${KAFKA\_FILTER}" | grep -E -v "${KAFKA\_EXCEPTION}" | grep -v -e '^$' | awk '{print $1":"$2":"$3}');  export TOPIC\_LIST=$( echo "${TMP\_TOPIC\_LIST\_DETAILS}" | grep -E "${KAFKA\_FILTER}" | grep -E -v "${KAFKA\_EXCEPTION}" | grep -v -e '^$' | awk '{print $1}'); |

Table 5 kafka.sh

es.sh

The es.sh script contains Elasticsearch relevant configuration. For example the Elasticsearch cluster connection properties and the indices to be initialized when the cluster is initialized.

|  |
| --- |
| export ES\_NAMESPACE="monitoring";  export ES\_SVC="elasticsearch-trace-elasticsearch-coordinating-only.${ES\_NAMESPACE}.svc.cluster.local";  export ES\_PORT="9200";  export ES\_POD="elasticsearch-trace-elasticsearch-master-0";  export ES\_CLUSTER\_NAME="elasticsearch-trace";    # Set ES\_EXCEPTION if you want to select a subset of indices (regex) to be NOT handled by the cluster configuration script (eg. modules/es.sh configuration/configuration.sh init)  # Set to "nothing" if no exception required (empty string will fail)  # Example: export ES\_EXCEPTION="session|raw|aux|mpl|msk|landmask"  export ES\_EXCEPTION="nothing"    # Set ES\_FILTER if you want to select only a subset of indices (regex) to be handled by the cluster configuration script (eg. modules/es.sh configuration/configuration.sh init)  # Example: export ES\_FILTER="session|raw|aux|mpl|msk|mask|prip|lifecycle|plan\_and\_report|spp|l0|l1|l2"  export ES\_FILTER=""    TMP\_ES\_INDICES\_INPUT="  mpl\_orbpre  mpl\_orbres  mpl\_orbsct  aux\_obmemc  aux\_poeorb  aux\_preorb  aux\_resorb  aux\_cal  aux\_pp1  aux\_pp2  aux\_ins  aux\_wnd  aux\_ice  aux\_wav  raw  session  msk\_ew\_slc  msk\_\_land\_  msk\_ocean\_  msk\_ovrpas  aux\_ece  aux\_scs  plan\_and\_report  spp\_mbu  spp\_obs  ";    TMP\_ES\_INDICES\_TMP="  aux\_att  ";    TMP\_ES\_INDICES\_SEGMENTS="  l0\_segment  ";    TMP\_ES\_INDICES\_PRODUCTS="  l0\_slice  l0\_acn  l1\_slice  l1\_acn  l2\_slice  l2\_acn  ";    TMP\_ES\_INDICES\_EWSLCMASK="  ewslcmask  ";    TMP\_ES\_INDICES\_LANDMASK="  landmask  ";    TMP\_ES\_INDICES\_OCEANMASK="  oceanmask  ";    TMP\_ES\_INDICES\_OVERPASSMASK="  overpassmask  ";    TMP\_ES\_INDICES\_PRIP="  prip  ";    TMP\_ES\_INDICES\_DATA\_LIFECYCLE\_METADATA="  data-lifecycle-metadata  ";    TMP\_ES\_ALL="  ${TMP\_ES\_INDICES\_INPUT}  ${TMP\_ES\_INDICES\_TMP}  ${TMP\_ES\_INDICES\_SEGMENTS}  ${TMP\_ES\_INDICES\_PRODUCTS}  ${TMP\_ES\_INDICES\_EWSLCMASK}  ${TMP\_ES\_INDICES\_LANDMASK}  ${TMP\_ES\_INDICES\_OCEANMASK}  ${TMP\_ES\_INDICES\_OVERPASSMASK}  ${TMP\_ES\_INDICES\_PRIP}  ${TMP\_ES\_INDICES\_DATA\_LIFECYCLE\_METADATA}  ";    # Delete empty lines  export ES\_INDICES\_INPUT=$( echo "${TMP\_ES\_INDICES\_INPUT}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_INDICES\_TMP=$( echo "${TMP\_ES\_INDICES\_TMP}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_INDICES\_SEGMENTS=$( echo "${TMP\_ES\_INDICES\_SEGMENTS}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_INDICES\_PRODUCTS=$( echo "${TMP\_ES\_INDICES\_PRODUCTS}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_INDICES\_EWSLCMASK=$( echo "${TMP\_ES\_INDICES\_EWSLCMASK}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_INDICES\_LANDMASK=$( echo "${TMP\_ES\_INDICES\_LANDMASK}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_INDICES\_OCEANMASK=$( echo "${TMP\_ES\_INDICES\_OCEANMASK}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_INDICES\_OVERPASSMASK=$( echo "${TMP\_ES\_INDICES\_OVERPASSMASK}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_INDICES\_PRIP=$( echo "${TMP\_ES\_INDICES\_PRIP}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_INDICES\_DATA\_LIFECYCLE\_METADATA=$( echo "${TMP\_ES\_INDICES\_DATA\_LIFECYCLE\_METADATA}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$');  export ES\_ALL=$( echo "${TMP\_ES\_ALL}" | grep -E "${ES\_FILTER}" | grep -E -v "${ES\_EXCEPTION}" | grep -v -e '^$'); |

Table 6 es.sh

### mongo.sh

The mongo.sh script contains MongoDB relevant configuration. For example the MongoDB cluster connection properties.

|  |
| --- |
| export MONGO\_NAMESPACE="infra";  export MONGO\_APP="s1pro-mongodb";  export MONGO\_POD\_PREFIX="mongodb-primary-";  export MONGO\_SVC="mongodb-headless.infra.svc.cluster.local";  export MONGO\_HOST="mongodb-primary-0.mongodb-headless.infra.svc.cluster.local,mongodb-secondary-0.mongodb-headless.infra.svc.cluster.local,mongodb-secondary-1.mongodb-headless.infra.svc.cluster.local";  export MONGO\_PORT="27017";  export MONGO\_DB="s1pdgs"; |

Table 7 mongo.sh

### deploy.sh

The deploy.sh script contains Helm relevant configuration. For example, this includes the chart version as well as the possiblity to bypass the deployment of individual charts.

|  |
| --- |
| HELM\_CHART\_VERSION="3.32.0";    HELM\_LIST\_TMP="  00 YES s1pro-s1pdgs-base ${HELM\_CHART\_VERSION} helm/00\_s1pdgs\_base/values.yaml  30 YES s1pro-app-catalog ${HELM\_CHART\_VERSION} helm/30\_app\_catalog/values.yaml  31 YES s1pro-metadata-catalog-trigger ${HELM\_CHART\_VERSION} helm/31\_metadata\_catalog\_trigger/values.yaml  32 YES s1pro-metadata-catalog-worker ${HELM\_CHART\_VERSION} helm/32\_metadata\_catalog\_worker/values.yaml  33 YES s1pro-disseminator-fos ${HELM\_CHART\_VERSION} helm/33\_disseminator\_fos/values.yaml  33 YES s1pro-disseminator-pod ${HELM\_CHART\_VERSION} helm/33\_disseminator\_pod/values.yaml  34 YES s1pro-disseminator-mp ${HELM\_CHART\_VERSION} helm/34\_disseminator\_mp/values.yaml  34 YES s1pro-disseminator-unav ${HELM\_CHART\_VERSION} helm/34\_disseminator\_unav/values.yaml  36 YES s1pro-disseminator-errmat ${HELM\_CHART\_VERSION} helm/36\_disseminator\_errmat/values.yaml  41 YES s1pro-l0-aio-production-trigger ${HELM\_CHART\_VERSION} helm/41\_l0\_aio\_production\_trigger/values.yaml  42 YES s1pro-l0-aio-ipf-preparation-worker ${HELM\_CHART\_VERSION} helm/42\_l0\_aio\_ipf\_preparation\_worker/values.yaml  43 YES s1pro-l0-asp-production-trigger ${HELM\_CHART\_VERSION} helm/43\_l0\_asp\_production\_trigger/values.yaml  44 YES s1pro-l0-asp-ipf-preparation-worker ${HELM\_CHART\_VERSION} helm/44\_l0\_asp\_ipf\_preparation\_worker/values.yaml  45 YES s1pro-l1-production-trigger ${HELM\_CHART\_VERSION} helm/45\_l1\_production\_trigger/values.yaml  46 YES s1pro-l1-ipf-preparation-worker ${HELM\_CHART\_VERSION} helm/46\_l1\_ipf\_preparation\_worker/values.yaml  47 YES s1pro-l2-production-trigger ${HELM\_CHART\_VERSION} helm/47\_l2\_production\_trigger/values.yaml  48 YES s1pro-l2-ipf-preparation-worker ${HELM\_CHART\_VERSION} helm/48\_l2\_ipf\_preparation\_worker/values.yaml  52 YES s1pro-l0-aio-ipf-execution-worker ${HELM\_CHART\_VERSION} helm/52\_l0\_aio\_ipf\_execution\_worker/values.yaml  54 YES s1pro-l0-asp-ipf-execution-worker ${HELM\_CHART\_VERSION} helm/54\_l0\_asp\_ipf\_execution\_worker/values.yaml  56 YES s1pro-l1-ipf-execution-worker ${HELM\_CHART\_VERSION} helm/56\_l1\_ipf\_execution\_worker/values.yaml  58 YES s1pro-l2-ipf-execution-worker ${HELM\_CHART\_VERSION} helm/58\_l2\_ipf\_execution\_worker/values.yaml  59 YES s1pro-obs-production-trigger ${HELM\_CHART\_VERSION} helm/59\_obs\_production\_trigger/values.yaml  60 YES s1pro-obs-ipf-preparation-worker ${HELM\_CHART\_VERSION} helm/60\_obs\_ipf\_preparation\_worker/values.yaml  61 YES s1pro-obs-ipf-execution-worker ${HELM\_CHART\_VERSION} helm/61\_obs\_ipf\_execution\_worker/values.yaml  62 YES s1pro-contingency-ingestion ${HELM\_CHART\_VERSION} helm/62\_contingency\_ingestion/values.yaml  63 YES s1pro-ingestion ${HELM\_CHART\_VERSION} helm/63\_ingestion/values.yaml  64 YES s1pro-ingestion-xbip-cgs01-trigger ${HELM\_CHART\_VERSION} helm/64\_ingestion\_xbip\_cgs01\_trigger/values.yaml  64 YES s1pro-ingestion-xbip-cgs02-trigger ${HELM\_CHART\_VERSION} helm/64\_ingestion\_xbip\_cgs02\_trigger/values.yaml  64 YES s1pro-ingestion-xbip-cgs03-trigger ${HELM\_CHART\_VERSION} helm/64\_ingestion\_xbip\_cgs03\_trigger/values.yaml  64 YES s1pro-ingestion-xbip-cgs04-trigger ${HELM\_CHART\_VERSION} helm/64\_ingestion\_xbip\_cgs04\_trigger/values.yaml  64 NO s1pro-ingestion-xbip-cgs05-trigger ${HELM\_CHART\_VERSION} helm/64\_ingestion\_xbip\_cgs05\_trigger/values.yaml  64 YES s1pro-ingestion-xbip-cgs10-trigger ${HELM\_CHART\_VERSION} helm/64\_ingestion\_xbip\_cgs10\_trigger/values.yaml  65 YES s1pro-ingestion-xbip-worker ${HELM\_CHART\_VERSION} helm/65\_ingestion\_xbip\_worker/values.yaml  66 YES s1pro-ingestion-auxip-trigger ${HELM\_CHART\_VERSION} helm/66\_ingestion\_auxip\_trigger/values.yaml  67 YES s1pro-ingestion-auxip-worker ${HELM\_CHART\_VERSION} helm/67\_ingestion\_auxip\_worker/values.yaml  68 YES s1pro-plan-and-report-ingestion ${HELM\_CHART\_VERSION} helm/68\_plan\_and\_report\_ingestion/values.yaml  70 YES s1pro-data-request-worker ${HELM\_CHART\_VERSION} helm/70\_data\_request\_worker/values.yaml  70 YES s1pro-datalifecycle-trigger ${HELM\_CHART\_VERSION} helm/70\_datalifecycle\_trigger/values.yaml  70 YES s1pro-eviction-management-worker ${HELM\_CHART\_VERSION} helm/70\_eviction\_management\_worker/values.yaml  74 YES s1pro-compression-trigger ${HELM\_CHART\_VERSION} helm/74\_compression\_trigger/values.yaml  75 YES s1pro-compression-worker ${HELM\_CHART\_VERSION} helm/75\_compression\_worker/values.yaml  76 YES s1pro-prip-trigger ${HELM\_CHART\_VERSION} helm/76\_prip\_trigger/values.yaml  77 YES s1pro-prip-worker ${HELM\_CHART\_VERSION} helm/77\_prip\_worker/values.yaml  78 YES s1pro-prip-frontend ${HELM\_CHART\_VERSION} helm/78\_prip\_frontend/values.yaml  80 YES s1pro-myocean-trigger ${HELM\_CHART\_VERSION} helm/80\_myocean\_trigger/values.yaml  81 YES s1pro-myocean-worker ${HELM\_CHART\_VERSION} helm/81\_myocean\_worker/values.yaml  82 NO s1pro-myocean-cleaner ${HELM\_CHART\_VERSION} helm/82\_myocean\_cleaner/values.yaml  83 NO s1pro-report ${HELM\_CHART\_VERSION} helm/83\_report/values.yaml  83 YES s1pro-ingestion-edip-pedc-trigger ${HELM\_CHART\_VERSION} helm/84\_ingestion\_edip\_pedc\_trigger/values.yaml  85 YES s1pro-ingestion-edip-pedc-worker ${HELM\_CHART\_VERSION} helm/85\_ingestion\_edip\_pedc\_worker/values.yaml  84 YES s1pro-ingestion-edip-bedc-trigger ${HELM\_CHART\_VERSION} helm/84\_ingestion\_edip\_bedc\_trigger/values.yaml  85 YES s1pro-ingestion-edip-bedc-worker ${HELM\_CHART\_VERSION} helm/85\_ingestion\_edip\_bedc\_worker/values.yaml  86 YES s1pro-ingestion-edip-worker ${HELM\_CHART\_VERSION} helm/86\_ingestion\_edip\_worker/values.yaml  90 YES s1pro-on-demand-interface-provider ${HELM\_CHART\_VERSION} helm/90\_on\_demand\_interface\_provider/values.yaml  91 YES s1pro-request-repository ${HELM\_CHART\_VERSION} helm/91\_request\_repository/values.yaml  92 YES s1pro-queue-watcher ${HELM\_CHART\_VERSION} helm/92\_queue\_watcher/values.yaml  93 YES s1pro-validation ${HELM\_CHART\_VERSION} helm/93\_validation/values.yaml  96 YES s1pro-qcss-online-disseminator ${HELM\_CHART\_VERSION} helm/96\_qcss\_online\_disseminator/values.yaml  96 YES s1pro-qcss-online-disseminator-geo ${HELM\_CHART\_VERSION} helm/96\_qcss\_online\_disseminator\_geo/values.yaml  96 YES s1pro-qcss-online-ingestion-trigger ${HELM\_CHART\_VERSION} helm/96\_qcss\_online\_ingestion\_trigger/values.yaml  96 YES s1pro-qcss-online-ingestion-worker ${HELM\_CHART\_VERSION} helm/96\_qcss\_online\_ingestion\_worker/values.yaml  96 YES s1pro-qcss-webserver-disseminator ${HELM\_CHART\_VERSION} helm/96\_qcss\_webserver\_disseminator/values.yaml  97 NO s1pro-prometheus-adapter ${HELM\_CHART\_VERSION} helm/97\_prometheus\_adapter/values.yaml  98 NO s1pro-mock-webdav-cgs01 ${HELM\_CHART\_VERSION} helm/98\_mock\_webdav\_cgs01/values.yaml  98 NO s1pro-mock-webdav-cgs02 ${HELM\_CHART\_VERSION} helm/98\_mock\_webdav\_cgs02/values.yaml  98 NO s1pro-mock-webdav-cgs03 ${HELM\_CHART\_VERSION} helm/98\_mock\_webdav\_cgs03/values.yaml  98 NO s1pro-mock-webdav-cgs04 ${HELM\_CHART\_VERSION} helm/98\_mock\_webdav\_cgs04/values.yaml  98 NO s1pro-mock-webdav-cgs05 ${HELM\_CHART\_VERSION} helm/98\_mock\_webdav\_cgs05/values.yaml  98 NO s1pro-mock-webdav-cgs10 ${HELM\_CHART\_VERSION} helm/98\_mock\_webdav\_cgs10/values.yaml  99 YES s1pro-mock-dissemination ${HELM\_CHART\_VERSION} helm/99\_mock\_dissemination/values.yaml  "    export HELM\_CONF\_GLOBAL="$(dirname ${BASH\_SOURCE})/values.yaml";  export HELM\_CONF\_REPLICAS="$(dirname ${BASH\_SOURCE})/replicaCount.yaml";    export HELM\_LIST=$(echo "${HELM\_LIST\_TMP}" | grep -v -e '^$') |

Table 8 deploy.sh

This environmentals will modify the behavior of the automatic installation script in order to access the pods in your cluster correctly. Whenever possible, it is recommended to stick to the default values.

# Cluster Initialisation

## DEPLOYMENT PREPARATION

In order to use the automatic installation script a few things needs to be setup on the master server. Please check out the S1pro configuration project containing the base configuration that are shared by all systems:

|  |
| --- |
| cd ~  git clone https://gitlab.s1pdgs.eu/werum/processing/s1pro-configuration.git --branch <TAG-To-Deploy> |

Automatic deployment

After the prerequisites from the previous sections had been setup correctly, you can perform the automatic installation in order to finish the setup of the environment and launch the pods.

Before starting with the deployment, setup your environment by editing the files s1pro-configuration/configuration. It is highly recommended to copy these files to another location. If you already have a suitable configuration for your environment, please use a diff on it to verify if additional modifications are required.

To run the actual deployment, please execute the following command.

|  |
| --- |
| # ./wrapper.sh $PATH\_TO\_WRAPPER\_CONFIGURATION init |

In order to verify that the deployment was successfully, please use:

|  |
| --- |
| # ./wrapper.sh $PATH\_TO\_WRAPPER\_CONFIGURATION check |

In order to clean the environment, please use:

|  |
| --- |
| # ./wrapper.sh $PATH\_TO\_WRAPPER\_CONFIGURATION clean |

After deployment following pods should be up and running:

|  |
| --- |
| $ kubectl -n processing get pods  NAME READY STATUS RESTARTS AGE  s1pro-app-catalog-0 1/1 Running 0 2d14h  s1pro-compression-trigger-0 2/2 Running 0 2d14h  s1pro-compression-worker-0 2/2 Running 0 2d14h  s1pro-contingency-ingestion-0 3/3 Running 0 2d14h  s1pro-data-request-worker-0 2/2 Running 0 2d14h  s1pro-datalifecycle-trigger-0 2/2 Running 0 2d14h  s1pro-disseminator-errmat-0 2/2 Running 0 2d14h  s1pro-disseminator-fos-0 2/2 Running 0 2d14h  s1pro-disseminator-mp-0 2/2 Running 0 2d14h  s1pro-disseminator-pod-0 2/2 Running 0 2d14h  s1pro-disseminator-unav-0 2/2 Running 0 2d14h  s1pro-eviction-management-worker-0 2/2 Running 0 2d14h  s1pro-ingestion-0 3/3 Running 0 2d14h  s1pro-ingestion-1 3/3 Running 0 2d14h  s1pro-ingestion-2 3/3 Running 0 2d14h  s1pro-ingestion-auxip-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-auxip-worker-0 2/2 Running 0 2d14h  s1pro-ingestion-edip-bedc-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-edip-bedc-worker-0 2/2 Running 0 2d14h  s1pro-ingestion-edip-pedc-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-edip-pedc-worker-0 2/2 Running 0 2d14h  s1pro-ingestion-xbip-cgs01-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs02-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs03-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs04-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs05-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs10-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-worker-0 2/2 Running 0 2d14h  s1pro-ingestion-xbip-worker-1 2/2 Running 0 2d14h  s1pro-l0-aio-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-l0-aio-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-l0-aio-production-trigger-0 2/2 Running 0 2d14h  s1pro-l0-asp-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-l0-asp-ipf-execution-worker-1 2/2 Running 0 2d14h  s1pro-l0-asp-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-l0-asp-production-trigger-0 2/2 Running 0 2d14h  s1pro-l1-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-l1-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-l1-production-trigger-0 2/2 Running 0 2d14h  s1pro-l2-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-l2-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-l2-production-trigger-0 2/2 Running 0 2d14h  s1pro-mbu-dissemination-trigger-0 2/2 Running 0 2d14h  s1pro-mbu-dissemination-worker-0 2/2 Running 0 2d14h  s1pro-mbu-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-mbu-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-mbu-production-trigger-0 2/2 Running 0 2d14h  s1pro-metadata-catalog-trigger-0 2/2 Running 0 2d14h  s1pro-metadata-catalog-worker-0 2/2 Running 0 2d14h  s1pro-mock-dissemination-0 2/2 Running 0 2d14h  s1pro-mock-edip-bedc-0 1/1 Running 0 2d14h  s1pro-mock-edip-pedc-0 1/1 Running 0 2d14h  s1pro-mock-webdav-cgs01-0 1/1 Running 0 2d14h  s1pro-mock-webdav-cgs02-0 1/1 Running 0 2d14h  s1pro-mock-webdav-cgs03-0 1/1 Running 0 2d14h  s1pro-mock-webdav-cgs04-0 1/1 Running 0 2d14h  s1pro-mock-webdav-cgs05-0 1/1 Running 0 2d14h  s1pro-mock-webdav-cgs10-0 1/1 Running 0 2d14h  s1pro-myocean-trigger-0 2/2 Running 0 2d14h  s1pro-myocean-worker-0 2/2 Running 0 2d14h  s1pro-obs-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-obs-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-obs-production-trigger-0 2/2 Running 0 2d14h  s1pro-on-demand-**interface**-provider-0 1/1 Running 0 2d14h  s1pro-plan-and-report-ingestion-0 3/3 Running 0 2d14h  s1pro-prip-frontend-0 1/1 Running 0 2d14h  s1pro-prip-trigger-0 2/2 Running 0 2d14h  s1pro-prip-worker-0 2/2 Running 0 2d14h  s1pro-prometheus-adapter-0 1/1 Running 0 2d14h  s1pro-qcss-online-disseminator-0 2/2 Running 0 2d14h  s1pro-qcss-online-disseminator-geo-0 2/2 Running 0 2d14h  s1pro-qcss-online-ingestion-trigger-0 1/1 Running 0 2d14h  s1pro-qcss-online-ingestion-worker-0 2/2 Running 0 2d14h  s1pro-qcss-webserver-disseminator-0 2/2 Running 0 2d14h  s1pro-queue-watcher-0 2/2 Running 0 2d14h  s1pro-report-ftp-0 2/2 Running 0 2d14h  s1pro-request-repository-0 1/1 Running 0 2d14h  s1pro-validation-0 1/1 Running 0 2d14h |

## Pod Initialisation

Most of the pods/containers are automatically initialized and are ready to go after the automatic deployment. Pods/Containers that need specific initialisation are listed here.

### Contingency Ingestion

The Ingestion-Trigger container of Contingency Ingestion  need to have inbox directoriers configured and present in the file system from where it picks up the files to ingest.

For EDRS sessions ingestion there need to be a specific directory structure created in the inbox directory. Eg. if the inbox directory is /data/inbox/ the directory structure for EDRS sessions need to be as follows:

|  |
| --- |
| [inbox path]/[4 letter station name]/[3 letter satellite ID]  eg.:    /data/inbox/WILE/S1A  /data/inbox/WILE/S1B  /data/inbox/MPS\_/S1A  /data/inbox/MPS\_/S1B  /data/inbox/SGS\_/S1A  /data/inbox/SGS\_/S1B  /data/inbox/MTI\_/S1A  /data/inbox/MTI\_/S1B |

This is not relevant for other files and the directory structure can be flat.

# 6 Patch Deployment

Please read the SRN of the Patch Delivery for a more detailed description of the patch provided. In the SRN it is written which helm charts need to be re-deployed for the patch and if the persistence layer is affected or not.

For deployment of single charts use the script "deploy" (see below).

## deploy script

$ deploy

Synopsis: INSTALL|UNINSTALL|INSTALL\_LOCAL|UNINSTALL\_LOCAL [CHART\_DIR]  
Option CHART\_DIR allows to (un-)deploy a single service  
 INSTALL, i Installs referred helm chart(/charts) for configured chart repository (harbor)  
 INSTALL\_LOCAL, il Installs helm chart from given local chart directory  
 UNINSTALL, u Uninstalls referred helm chart(/charts) for configured chart repository (harbor)  
 UNINSTALL\_LOCAL, ul Uninstalls helm chart from given local chart directory

The script itself uses the generic scripts as described in https://wiki.s1pdgs.eu/x/AQBFBQ,

modules/deploy.sh $ENV\_DIR/configuration/template-wrapper.sh init "CHART\_DIR" [LOCAL\_INST]

Option CHART\_DIR allows to (un-)deploy a single service  
Option LOCAL\_INST specifies whether the locally available charts shall be installed (default: false)

**Examples:**

$ deploy UNINSTALL 78\_prip\_frontend

Undeploying service 78\_prip\_frontend  
  
#################################  
# HELM PROCESSING CLEANING ...  
#################################  
Using /home/sysadmin1/s1pro-configuration/helm to clean services ...  
Undeploying ... s1pro-prip-frontend  
try to delete s1pro-prip-frontend  
release "s1pro-prip-frontend" uninstalled  
  
s1pro-prip-frontend PURGED   
done.

$ deploy INSTALL 78\_prip\_frontend

Deploying service 78\_prip\_frontend  
  
#################################  
# HELM PROCESSING INIT ...  
#################################  
Updating helm repository werum (<https://registry.tools.s1pdgs.eu/chartrepo/werum>)  
Hang tight while we grab the latest from your chart repositories...  
...Successfully got an update from the "werum" chart repository  
Update Complete. ⎈ Happy Helming!⎈   
Using /home/sysadmin1/s1pro-configuration/helm to deploy services ...  
Deploying ... prip-frontend  
wait for all spawned 'helm install' processes to finish  
done  
  
s1pro-prip-frontend DEPLOYED   
done.

# Verification of Deployment on Kubernetes

## Automatic Verification of Deployment on Kubernetes

In order to to an automatic verification of the deployment on your cluster, go into the folder s1pro-configuration you checked out and perform the following command:

|  |
| --- |
| # ./wrapper.sh $PATH\_TO\_WRAPPER\_CONFIGURATION check |

## Manual Verification of Deployment on Kubernetes

### Applicative catalog

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-app-catalog-0 1/1 Running 0 2d14h  … |

### Metadata catalog

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-metadata-catalog-trigger-0 2/2 Running 0 2d14h  s1pro-metadata-catalog-worker-0 2/2 Running 0 2d14h  … |

### XBIP

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-ingestion-xbip-cgs01-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs02-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs03-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs04-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs05-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-cgs10-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-xbip-worker-0 2/2 Running 0 2d14h  s1pro-ingestion-xbip-worker-1 2/2 Running 0 2d14h  … |

### AUXIP

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-ingestion-auxip-trigger-0 1/1 Running 0 41m  s1pro-ingestion-auxip-worker-0 2/2 Running 0 41m  … |

### EDIP

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-ingestion-edip-bedc-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-edip-bedc-worker-0 2/2 Running 0 2d14h  s1pro-ingestion-edip-pedc-trigger-0 1/1 Running 0 2d14h  s1pro-ingestion-edip-pedc-worker-0 2/2 Running 0 2d14h  … |

### Ingestion

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-ingestion-0 3/3 Running 0 2d14h  s1pro-ingestion-1 3/3 Running 0 2d14h  s1pro-ingestion-2 3/3 Running 0 2d14h  … |

### Plan and Report Ingestion

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-plan-and-report-ingestion-0 3/3 Running 0 2d14h  … |

### Compression

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-compression-trigger-0 2/2 Running 0 2d14h  s1pro-compression-worker-0 2/2 Running 0 2d14h  … |

### Request Repository

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  request-repository-0 1/1 Running 0 2d14h  … |

### Queue Watcher

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  queue-watcher-0 1/1 Running 0 2d14h  … |

### SPP MBU

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-mbu-dissemination-trigger-0 2/2 Running 0 2d14h  s1pro-mbu-dissemination-worker-0 2/2 Running 0 2d14h  s1pro-mbu-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-mbu-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-mbu-production-trigger-0 2/2 Running 0 2d14h  … |

### SPP OBS

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-obs-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-obs-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-obs-production-trigger-0 2/2 Running 0 2d14h  … |

### AIO

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-l0-aio-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-l0-aio-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-l0-aio-production-trigger-0 2/2 Running 0 2d14h  … |

### ASP

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-l0-asp-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-l0-asp-ipf-execution-worker-1 2/2 Running 0 2d14h  s1pro-l0-asp-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-l0-asp-production-trigger-0 2/2 Running 0 2d14h  … |

### L1

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-l1-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-l1-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-l1-production-trigger-0 2/2 Running 0 2d14h  … |

### L2

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-l2-ipf-execution-worker-0 2/2 Running 0 2d14h  s1pro-l2-ipf-preparation-worker-0 2/2 Running 0 2d14h  s1pro-l2-production-trigger-0 2/2 Running 0 2d14h  … |

### PRIP

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-prip-frontend-0 1/1 Running 0 2d14h  s1pro-prip-trigger-0 2/2 Running 0 2d14h  s1pro-prip-worker-0 2/2 Running 0 2d14h  … |

### Contingency Ingestion

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-contingency-ingestion-0 3/3 Running 0 2d14h  … |

### Dissemination

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-disseminator-errmat-0 2/2 Running 0 2d14h  s1pro-disseminator-fos-0 2/2 Running 0 2d14h  s1pro-disseminator-mp-0 2/2 Running 0 2d14h  s1pro-disseminator-pod-0 2/2 Running 0 2d14h  s1pro-disseminator-unav-0 2/2 Running 0 2d14h  … |

### Validation

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-validation-0 1/1 Running 0 2d14h  … |

### On-Demand Interface Provider

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-on-demand-**interface**-provider-0 1/1 Running 0 2d14h  … |

### Request Respository

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-request-repository-0 1/1 Running 0 2d14h  … |

### Report

Before launching the pods the environment variables can be configured (see Annex1)

Deploy the servers for publications:

|  |
| --- |
| # kubectl create -f init\_configuration/K8s\_conf/83\_report.yaml |

Verify

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  output-report-ftp-0 2/2 Running 0 2d14h  … |

### MyOcean Service

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-myocean-trigger-0 2/2 Running 0 2d14h  s1pro-myocean-worker-0 2/2 Running 0 2d14h  s1pro-job-myocean-cleaner-1615201200-cm2tp 1/1 Running 0 2m4s  … |

The MyOcean Cleaner is realized as a Kubernetes cronjob that will spawn jobs at the configured schedule. The jobs will create a pod with the cleaner software and execute it. The pod only exists when created by the job and will stay for 10 minutes before it vanishes. During this 10 minutes the pod is visible per kubectl get pods.

|  |
| --- |
| # kubectl get cronjobs  NAME SCHEDULE SUSPEND ACTIVE LAST SCHEDULE AGE  s1pro-job-myocean-cleaner 0 \* \* \* \* False 0 18m 13h    # kubectl get jobs  NAME COMPLETIONS DURATION AGE  s1pro-job-myocean-cleaner-1614762000 0/1 18m 18m |

### Data Lifecycle Services

|  |
| --- |
| # kubectl get pods NAME                                    READY   STATUS        RESTARTS   AGE …                s1pro-data-request-worker-0             2/2     Running       0          13hs1pro-datalifecycle-trigger-0           2/2     Running       0          13h s1pro-eviction-management-worker-0      2/2     Running       0          13h … |

### QCSS Pods

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-qcss-online-disseminator-0 2/2 Running 0 5m  s1pro-qcss-online-disseminator-geo-0 2/2 Running 0 4m59s  s1pro-qcss-online-ingestion-trigger-0 1/1 Running 0 4m58s  s1pro-qcss-online-ingestion-worker-0 2/2 Running 0 4m58s  s1pro-qcss-webserver-disseminator-0 2/2 Running 0 4m57s  … |

### Prometheus Adapter

|  |
| --- |
| # kubectl get pods  NAME READY STATUS RESTARTS AGE  …  s1pro-prometheus-adapter-0 1/1 Running 0 2d14h  … |

|  |
| --- |
|  |

# Annex 1

In order to deploy the software successfully into a cluster, it is required to setup the environment specific aspects of the environment correctly.

These settings are saved "s1pro-configuration"  repository under the templates director:

|  |
| --- |
| https://gitlab.s1pdgs.eu/werum/processing/s1pro-configuration/tree/master/templates |

These templates provide basic layout that can be used to create an environment specific configuration for the S1PRO.

## Values.yaml

This file contains default configuration parameters for S1PRO that shall be reviewed and modified as per need. The following table code block the configuration parameters that are available:

|  |
| --- |
| # Shall helm deploy the simulator images?  simulator:  simulateL0AioIpf: **false**  simulateL0AspIpf: **false**  simulateL1Ipf: **false**  simulateL2Ipf: **false**  simulateMbuIpf: **false**    image:  imagePullSecrets: harbor-tools  registry: registry.tools.s1pdgs.eu  repository: werum/processing  tag: develop    mqiImage:  registry: registry.tools.s1pdgs.eu  repository: werum/processing  name: s1pro-core-mqi-server  tag: develop    mqi:  persistence:  messagePersistenceStrategy: AppCatalogMessagePersistence # InMemoryMessagePersistence | AppCatalogMessagePersistence    gateway:  toleration:  key: node-role.kubernetes.io/gateway  effect: NoSchedule  label:  key: node-role.kubernetes.io/gateway  value:  net: infra  std: std  edrs:  pedc: edrs-pedc  bedc: edrs-bedc    myocean:  toleration:  key: node-role.kubernetes.io/myocean  effect: NoSchedule  label:  key: node-role.kubernetes.io/myocean  value: ftp    qcss:  toleration:  key: node-role.kubernetes.io/qcss  effect: NoSchedule  label:  key: node-role.kubernetes.io/worker  value: infra    mbu:  toleration:  key: node-role.kubernetes.io/mbu  effect: NoSchedule  label:  key: node-role.kubernetes.io/worker  value: infra    worker:  label:  key: node-role.kubernetes.io/worker  value: infra    processor:  toleration:  key: node-role.kubernetes.io/processor  effect: NoSchedule  label:  key: node-role.kubernetes.io/processor  value:  aio: aio  asp: asp  l1: l1  l2: l2  zip: zip    kongplugin:  clientId: "s1pdgs\_client"  clientSecret: ""  discovery: "https://c1.ops.s1pdgs.eu/auth/realms/s1pdgs/.well-known/openid-configuration"  sessionSecret: ""  introspection\_endpoint: "https://c1.ops.s1pdgs.eu/auth/realms/s1pdgs/protocol/openid-connect/token/introspect"  groupsAuthorizedPaths:  group:  - name: "operations"  path: "/odata"  - name: "centreexpert"  path: "/odata"  - name: "op\_manager"  path: "/odata"  - name: "sysadmin"  path: "/odata"  - name: "customer"  path: "/odata"  - name: "b2b"  path: "/odata"    metadataCatalog:  restApiNbRetry: 3  restApiTempoRetryMs: 1000    auxip:  serviceRootUri: "https://aux1.s1pdgs.eu/odata/v1"  sslValidation: "false"  authType: "oauth2"  oauthAuthUrl: "https://aux1.s1pdgs.eu/auth/realms/s1pdgs/protocol/openid-connect/token"  creationDateAttributeName: "PublicationDate"  productNameAttrName: "Name"  idAttrName: "Id"  contentLengthAttrName: "ContentLength"  start: "2021-02-22T08:30:00.000000"  timeWindowSec: 240 #3600 # The time window to check at each request  timeWindowOverlapSec: 240 # the current time window will have some overlap in seconds with the predecessing time window  # **this** is just to have the possibility to add some savety margin, making sure nothing is overseen between time windows  offsetFromNowSec: 240 # time window will not reach now but now minus offset-from-now-sec  # **this** is **for** savety too, giving the queried system some grace period to provide **new**/current data  maxPageSize: 500 # aux metadata will be splitted into pages with **this** size    xbip:  bufferInputs: **false**  ignoreFilesBeforeDate:  cgs01: "2021-02-23T10:00:00.000Z"  cgs02: "2021-02-23T10:00:00.000Z"  cgs03: "2021-02-23T10:00:00.000Z"  cgs04: "2021-02-23T10:00:00.000Z"  cgs05: "2021-02-23T10:00:00.000Z"  cgs10: "2021-02-23T10:00:00.000Z"  stationRetentionTime:  cgs01: 7  cgs02: 7  cgs03: 7  cgs04: 7  cgs05: 7  cgs10: 7    edip:  bufferInputs: **false**  pedc:  ignoreFilesBeforeDate: "2021-02-23T10:00:00.000Z"  stationRetentionTime: 7  host: 10.21.10.1  path: /  port: 21  enablePasv: **true**  explicitFtps: **true**  disableCertificateCheck: **false**  enableHostnameVerification: **false**  enableDataChannelEncryption: **false**  sslProtocol: TLSv1.2  connectTimeoutSec: 60  directory: "ftp://${pedc.host}:${pedc.port}${pedc.path}NOMINAL/"  matchRegex: '^([A-Za-z\_]{4}/)?([0-9A-Za-z\_]{2})([0-9A-Za-z\_]{1})/([0-9A-Za-z\_]+)/(ch[0|\_]?[1-2]/)?(DCS\_[0-9]{2}\_([a-zA-Z0-9\_]\*)\_ch([12])\_(DSDB|DSIB).\*\.(raw|aisp|xml|RAW|AISP|XML))$'  topic: "t-pdgs-ingestion-jobs-edip-pedc-retransfer"  family: "SESSION\_RETRANSFER"  keystore\_data: ""  keystore\_pass: "changeit"  truststore\_data: "xxxxxxxxxx"  truststore\_pass: changeit  bedc:  ignoreFilesBeforeDate: "2021-02-23T10:00:00.000Z"  stationRetentionTime: 7  host: 10.22.10.1  path: /  port: 21  enablePasv: **true**  explicitFtps: **true**  disableCertificateCheck: **false**  enableHostnameVerification: **false**  enableDataChannelEncryption: **false**  sslProtocol: TLSv1.2  connectTimeoutSec: 60  directory: "ftp://${pedc.host}:${pedc.port}${pedc.path}NOMINAL/"  matchRegex: '^([A-Za-z\_]{4}/)?([0-9A-Za-z\_]{2})([0-9A-Za-z\_]{1})/([0-9A-Za-z\_]+)/(ch[0|\_]?[1-2]/)?(DCS\_[0-9]{2}\_([a-zA-Z0-9\_]\*)\_ch([12])\_(DSDB|DSIB).\*\.(raw|aisp|xml|RAW|AISP|XML))$'  topic: "t-pdgs-ingestion-jobs-edip-bedc-retransfer"  family: "SESSION\_RETRANSFER"  keystore\_data: ""  keystore\_pass: "changeit"  truststore\_data: "xxxxxxxxxx"  truststore\_pass: changeit    dataLifecycle:  retentionTimeUpdate:  offsetInHoursL1: 6    disseminator:  target:  fos\_hk\_s1a: "fos-hk-s1a" # mock or fos-hk-s1a  fos\_hk\_s1b: "fos-hk-s1b" # mock or fos-hk-s1a  pod\_gp: "pod-gp" # mock or pod-gp  pod\_l0a: "pod-l0a" # mock or pod-l0a  mp: "mp" # mock or mp  unav: "unav" # mock or unav  errmat: "errmat" # mock or errmat  myocean: "myocean"  mbu: "mbu"  qcss\_app: "qcss-app"  qcss\_web: "qcss-web"  outboxes:  mock:  protocol: sftp  hostname: s1pro-mock-dissemination-svc.processing.svc.cluster.local  port: 22  username: foo  password: pass  path: /outboxes  qcss:  app:  path: /data/in  chmod-script-path: /app/chmod/script.sh  web:  protocol: sftp  hostname: qcss-web-server-svc.processing.svc.cluster.local  port: 22  username: foo  password: pass  path: /data/in  ftp-pasv: **true**  truststore\_pass: changeit  mp:  protocol: sftp  hostname: s1mis-ingestor-mp.monitoring.svc.cluster.local  port: 22  username: foo  password: pass  path: /in  ftp-pasv: **true**  truststore\_pass: changeit  unav:  protocol: sftp  hostname: s1mis-ingestor-unav.monitoring.svc.cluster.local  port: 22  username: foo  password: pass  path: /in  ftp-pasv: **true**  truststore\_data:  fos:  hk\_s1a:  protocol: sftp  hostname: senxser.esoc.ops.esa.**int**  port: 22  username: s1apdgs  key-data: ""  path: /home/s1apdgs/fromPDGS  path-evaluator: ISIP  ftp-pasv: **true**  truststore\_data:  hk\_s1b:  protocol: sftp  hostname: senxser.esoc.ops.esa.**int**  port: 22  username: s1bpdgs  key-data: ""  path: /home/s1bpdgs/fromPDGS  path-evaluator: ISIP  pod:  gp:  protocol: ftps  hostname: cpodopsftp.gmv.com  port: 21  username: s1pdgs  password:  path: /s1pdgs/gps\_l0\_products  ftp-pasv: **true**  l0a:  protocol: ftps  hostname: cpodopsftp.gmv.com  port: 21  username: s1pdgs  password:  path: /s1pdgs/sar\_l0\_annotations\_products  ftp-pasv: **true**  errmat:  protocol: ftps  hostname: aux1.s1pdgs.eu  port: 2121  username: s1pdgs  password: "xxxxxx"  path: /  ftp-pasv: **true**  truststore\_data: "xxxxxxxxxx"  truststore-file: "/app/ssl/truststore.jks"  truststore-pass: "changeit"  myocean:  protocol: ftps  hostname: ops-c1-ftp-0001.cluster.local  port: 21  username: "s1pdgs"  password: "CC8IiE0qwUSG"  path: /**public**  path-evaluator: myocean  ftp-pasv: **true**  truststore\_data: "xxxxxxxxxx"  truststore-file: "/app/ssl/truststore.jks"  truststore-pass: "changeit"  mbu:  protocol: ftp  hostname: "s1pro-mock-dissemination-svc"  port: 21  username: s1pdgs  password: "PDGS5!"  path: **public**/METEO  path\_evaluator: mbu  ftp\_pasv: **false**  keystore\_data: ""  keystore-file: "/path/to/keystore.jks"  keystore-pass: "changeit"  truststore\_data: ""  truststore-file: "/path/to/truststore.jks"  truststore-pass: "changeit"    directorycleaner:  myocean:  schedule: "0 \* \* \* \*"  protocol: ftp  path: **public**  port: 21  user: s1pdgs  pass: "xxxxxxxx"  host: s1pro-mock-dissemination-svc  ftp\_pasv: **true**  retention\_time\_in\_days: 7  keystore\_data: ""  keystore\_pass: "changeit"  truststore\_data: ""  truststore\_pass: "changeit"    preparator:  aio:  minimalWaitingTimeSec: 36000  timeoutSec:  cgs1: 86400 # 7200  cgs2: 86400 # 7200  cgs3: 86400 # 7200  cgs4: 86400 # 7200  cgs5: 86400 # 7200  edrs: 86400 # 7200  descramble:  cgs1: "yes"  cgs2: "yes"  cgs3: "yes"  cgs4: "no"  cgs5: "yes"  edrs: "yes"  rsEncode:  cgs1: "yes"  cgs2: "yes"  cgs3: "yes"  cgs4: "yes"  cgs5: "yes"  edrs: "yes"  asp:  late-topic-active: **true**  fast:  minimal: 3 # 3  nominal: 36 # 20  late-after-milliseconds: 86400000  nrt:  minimal: 3 # 1  nominal: 36 # 2  late-after-milliseconds: 10800000  pt:  late-after-milliseconds: 10800000  l1:  late-topic-active: **true**  fast:  late-after-milliseconds: 86400000  nrt:  late-after-milliseconds: 10800000  pt:  late-after-milliseconds: 10800000  l2:  late-topic-active: **true**  fast:  late-after-milliseconds: 86400000  nrt:  late-after-milliseconds: 10800000  pt:  late-after-milliseconds: 10800000    appCatalog:  mqi:  cleaningOld: "172800000" # 2 day (1 day = 86400000)  jobs:  cleaningJobsTerminatedFixedRateMs: "604800000" # 7 days  cleaningJobsInvalidFixedRateMs: "604800000" # 7 days  edrsSessions:  maxAgeJobMs:  waiting: "604800000" # 7 days  dispatching: "604800000"  generating: "604800000"  terminated: "604800000"  levelSegments:  maxAgeJobMs:  waiting: "604800000"  dispatching: "604800000"  generating: "604800000"  terminated: "604800000"  levelProducts:  maxAgeJobMs:  waiting: "604800000"  dispatching: "604800000"  generating: "604800000"  terminated: "604800000"    prometheusAdapter:  prometheusUrl: http://prometheus-operator-prometheus.monitoring.svc.cluster.local:9090  securePort: 6443  metricsRelistInterval: 1m  targets:  xbip\_worker:  minReplicas: 2  maxReplicas: 4  scaleDown:  stabilizationWindowSeconds: 300  selectPolicy: Max  scaleUp:  stabilizationWindowSeconds: 10  selectPolicy: Max |

## ReplicaCount.yaml

This file can be used to define the amount of replica that should be set for the instance of the cluster and enforce a specific amount of replica. The file needs to be adjusted as per need.

|  |
| --- |
| replicaCount:  app\_catalog: 1  metadata\_catalog\_trigger: 5  metadata\_catalog\_worker: 10  disseminator\_fos: 1  disseminator\_pod: 1  disseminator\_qcss: 2  disseminator\_mp: 1  disseminator\_unav: 1  disseminator\_errmat: 1  l0\_aio\_production\_trigger: 1  l0\_aio\_ipf\_preparation\_worker: 1  l0\_asp\_production\_trigger: 1  l0\_asp\_ipf\_preparation\_worker: 1  l1\_production\_trigger: 1  l1\_ipf\_preparation\_worker: 1  l2\_production\_trigger: 1  l2\_ipf\_preparation\_worker: 1  l0\_aio\_ipf\_execution\_worker: 3  l0\_asp\_ipf\_execution\_worker: 3  l1\_ipf\_execution\_worker: 25  l2\_ipf\_execution\_worker: 20  obs\_production\_trigger: 1  obs\_ipf\_preparation\_worker: 1  obs\_ipf\_execution\_worker: 1  contingency\_ingestion: 1  ingestion: 10  ingestion\_qcss: 1  ingestion\_xbip\_cgs01\_trigger: 1  ingestion\_xbip\_cgs02\_trigger: 1  ingestion\_xbip\_cgs03\_trigger: 1  ingestion\_xbip\_cgs04\_trigger: 1  ingestion\_xbip\_cgs05\_trigger: 0  ingestion\_xbip\_cgs10\_trigger: 1  ingestion\_xbip\_worker: 20  ingestion\_auxip\_trigger: 0  ingestion\_auxip\_worker: 0  plan\_and\_report\_ingestion: 1  data\_request\_worker: 1  datalifecycle\_trigger: 1  eviction\_management\_worker: 1  compression\_trigger: 1  compression\_worker: 50  prip\_trigger: 1  prip\_worker: 2  prip\_frontend: 3  myocean\_trigger: 1  myocean\_worker: 1  report: 1  ingestion\_edip\_pedc\_trigger: 0  ingestion\_edip\_pedc\_worker: 0  ingestion\_edip\_bedc\_trigger: 0  ingestion\_edip\_bedc\_worker: 0  mbu\_production\_trigger: 0  mbu\_ipf\_preparation\_worker: 0  mbu\_ipf\_execution\_worker: 0  on\_demand\_interface\_provider: 1  request\_repository: 1  queue\_watcher: 1  validation: 1  mbu\_dissemination\_trigger: 0  mbu\_dissemination\_worker: 0  mock\_webdav\_cgs01: 1  mock\_webdav\_cgs02: 1  mock\_webdav\_cgs03: 1  mock\_webdav\_cgs04: 1  mock\_webdav\_cgs05: 1  mock\_webdav\_cgs10: 1  mock\_edip\_bedc: 0  mock\_edip\_pedc: 0  mock\_dissemination: 1  mock\_qcss\_online\_subsystem: 1  mock\_qcss\_webserver\_subsystem: 1  prometheus\_adapter: 1 |

In this case, it will enforce to use just one replica for metadata-catalog and compression.

# Annex 2

For encryption purposes (e. g. dissemination via FTPS), certificates, a truststore and optionally a keystore are needed.

This document will begin with the certificate creation for a keystore. If only a truststore is going to be used and certificates are already present, please skip over to the truststore section.

Finally, the key- or truststore is base64 encoded, so it can be used as configuration parameter value.

## Keystore

### Private key and certificate creation

The keystore is used to keep private keys and certificates. It is needed for FTPS encryption. If no certificate exists a certificate can be created.

**Generate a Java keystore and key pair:**

keytool -genkey -alias <ALIAS\_NAME> -keyalg RSA -keystore <PATH\_TO\_KEYSTORE.JKS> -keysize 2048

**Generate a certificate signing request (CSR) for an existing Java keystore:**

keytool -certreq -alias <ALIAS\_NAME> -keystore <PATH\_TO\_KEYSTORE.JKS> -file <PATH\_TO\_CERTIFICATE.CSR>

Forward this CSR to your CA and let them sign it.

**Import a signed primary certificate to an existing Java keystore:**

keytool -import -trustcacerts -alias <ALIAS\_NAME> -file <PATH\_TO\_CERTIFICATE.CRT> -keystore <PATH\_TO\_KEYSTORE.JKS>

It is of course also possible to use a self-signed certificate.

**Generate a keystore and self-signed certificate:**

keytool -genkey -keyalg RSA -alias <ALIAS\_NAME> -keystore <PATH\_TO\_KEYSTORE.JKS> -storepass <PASSWORD> -validity 360 -keysize 2048

### Private key and certificate import

If a private key and certificate are already available, but not stored in a JKS keystore, it can be converted with the following steps:

**Create PKCS12 keystore from private key and public certificate:**

openssl pkcs12 -export -name <NAME> -in <PATH\_TO\_CERTIFICATE.CRT> -inkey <PATH\_TO\_CERTIFICATE.KEY> -out <PATH\_TO\_KEYSTORE.P12>

**Convert PKCS12 keystore into a JKS keystore:**

keytool -importkeystore -destkeystore <PATH\_TO\_KEYSTORE.JKS> -srckeystore <PATH\_TO\_KEYSTORE.P12> -srcstoretype pkcs12 -alias <ALIAS\_NAME>

## Truststore

The truststore keeps all trusted Certificate Authorities (CA) and certificates. It is necessary to import the CA or certificate for the trusted FTPS servers.

If multiple FTPS servers are used, these have to be configured accordingly. If several certificates or CAs are imported, the alias has to be altered, as this is a unique identifier.

**Import a CA or certificate into a truststore:**

keytool -import -file <CERTIFICATE\_OR\_CA.PEM> -alias <ALIAS\_NAME> -keystore <TRUSTSTORE.JKS> -storepass <PASSWORD>

## Encoding Key-/Truststores as Base64

Key- or truststore are binary data. Before they can be passed as a configuration parameter value, they have to be encoded as a base64 string first.

**Convert a key-/truststore to a base64 encoded string:**

base64 -w0 <KEY\_OR\_TRUSTSTORE.JKS>

|  |
| --- |
| DISTRIBUTION LIST |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Corporate use | Project Limited |  | Configured |
| Internal Classification |  |  |  | X |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ESA | Info. | Action |  | AIRBUS D&S | Info. | Action |
| DESBOUILLONS Franck |  | X |  | ALACEVICH Roberto |  | X |
| GRIMONT Patrick | X |  |  | BUTHEAU Grégory | X |  |
| ROSICH Betlem | X |  |  | GUEYTRON Aline (ASTEK) |  | X |
|  |  |  |  | CUQ Pierre |  | X |
|  |  |  |  | GAVA Virginie |  | X |
|  |  |  |  | LEBRET Emmanuel |  | X |
|  |  |  |  | LECONTE Nicolas |  | X |
|  |  |  |  | MAURY Roland |  | X |
|  |  |  |  | MOILLARD Emmanuelle |  | X |
|  |  |  |  | MUNIER Bernard | X |  |
|  |  |  |  | PARMENTIER Mylène |  | X |
|  |  |  |  | THOMAS Kévin |  | X |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| C-S | Info. | Action |  | WERUM | Info. | Action |
| BOUISSON Cyrille |  | X |  | BERNHARDT Knut |  |  |
| BALDUCCI Adrien |  | X |  | COLAK Birol |  |  |
| DEVIGNOT Olivier | X |  |  | EDELMANN Jan-Marco |  |  |
| RECULEAU Serge | X |  |  | KAISER Fritz |  | X |
|  |  |  |  | KESSLER Torben |  | X |
|  |  |  |  | RAFI Faisal |  | X |
|  |  |  |  | STIEGE Lutz |  | X |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |