TED-SWS Architecture Overview

Editors Eugeniu Costetchi <eugen@meaningfy.ws>
Dragos Paun <dragos.paun@meaningfy.ws>

Version 0.9

Date 30/07/2022

Contents

Glossary

The official AWS glossary is available here.

The official Archimate business layer glossary and conventions are found here.

Introduction

The TED Semantic Web Service (TED SWS) is a pipeline system that continuously converts the public procurement notices (in XML format) available on the TED Website into RDF format, and publishes them into CELLAR. This is done so that, the produced RDF notices are made available to the public through CELLAR's SPARQL endpoint.

Purpose of the document

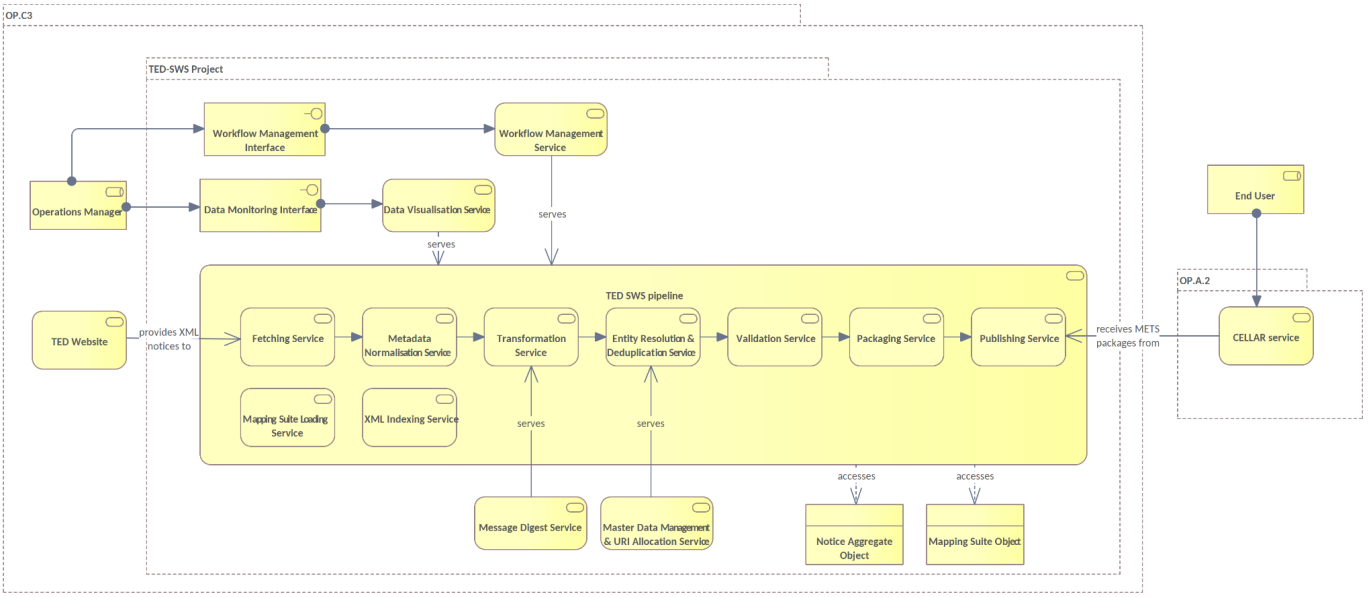
The purpose of this document is to give an overview of the infrastructure architecture of the TED-SWS system. The intention of this document is to be updated regularly by the development team.

Intended audience

This document is mainly intended for technical persons involved in the project and willing to have information on the infrastructure architecture of the system.

Application Architecture

The diagram below depicts the application architecture of the TED-SWS system using the Archimate business layer shape conventions and colour code.



There are three contexts depicted in the diagram. Two of them are organisational: the OP.C3 and OP.A2 units, and the central one is the TED-SWS project context situated within the OP.C3.

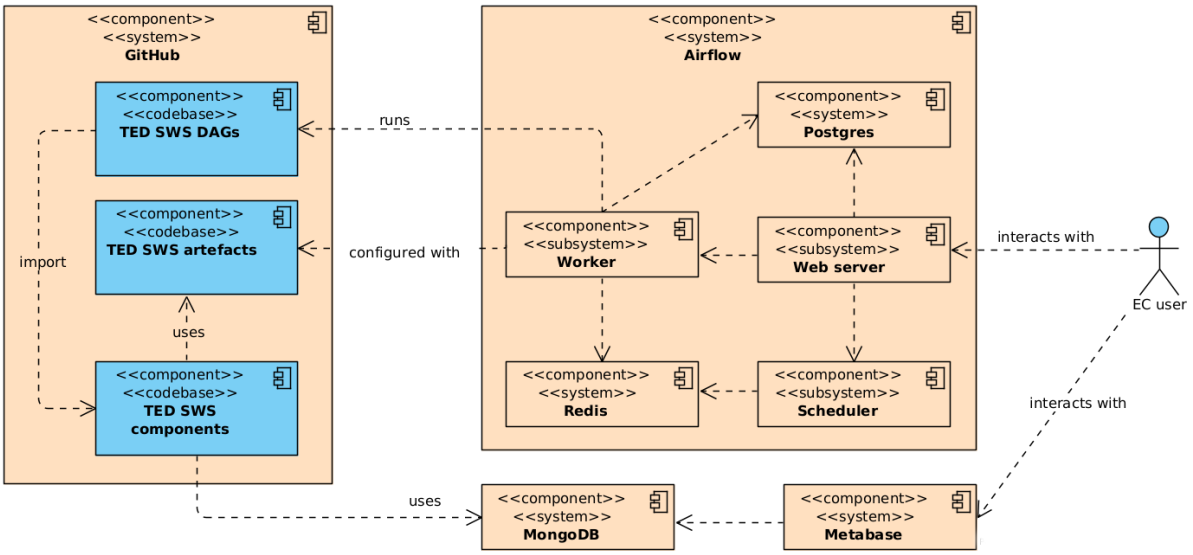
At the centre of the diagram is depicted the TED SWS pipeline as a service, which fetches XML notices from the TED website, processes the notices in a series of steps, and publishes them into Cellar RDF notices packages as METS packages.

To interact with the TED-SWS, the system provides the user interface for the Operations Manager role within the OP.C3 unit. The two main concerns are: (a) workflow management and (b) monitoring and status checking.

The workflow management is realised by the Airflow system, while the data monitoring is realised by the Metabase system. This is addressed in the next section.

Deployment architecture

This section describes what needs to be deployed and how.



The TED-SWS system is conceptualised as a set of *components* which must be executed in a well-established order forming a process, which is realised as a Directed Acyclic Graph (DAG). So there is a set of DAGs which import the components and trigger their execution. The codebase of the DAGS and components resides on GitHub. In addition, a set of configuration artefacts is necessary for the execution of the TED SWS system, which is handled as a separate repository on GitHub.

The Airflow system is meant to serve as an execution platform for DAGs. So it means that the DAGs, the components, and the artefacts must be deployed inside Airflow. So that it is possible to execute the DAGs, which will run as processes using the configurations.

Airflow is a set of a few components: Worker, Schedule and Web server, and they all need a Redis and a Postgres database. The worker is the component responsible for executing the DAGs, the scheduler governs when the executions shall take place, while the web server offers a control panel for the entire system.

The TED-SWS components are designed to use MongoDB for storing and retrieving information. So before executing any DAG, a MongoDB instance needs to be available and the connection parameters provided as environment configurations.

Metabase is an open-source analytics platform. It offers a view into the process outcome, available as data in MongoDB.

The user needs to communicate with two interfaces: the Airflow webserver to control processes and the Metabase to view the state of the processed data.

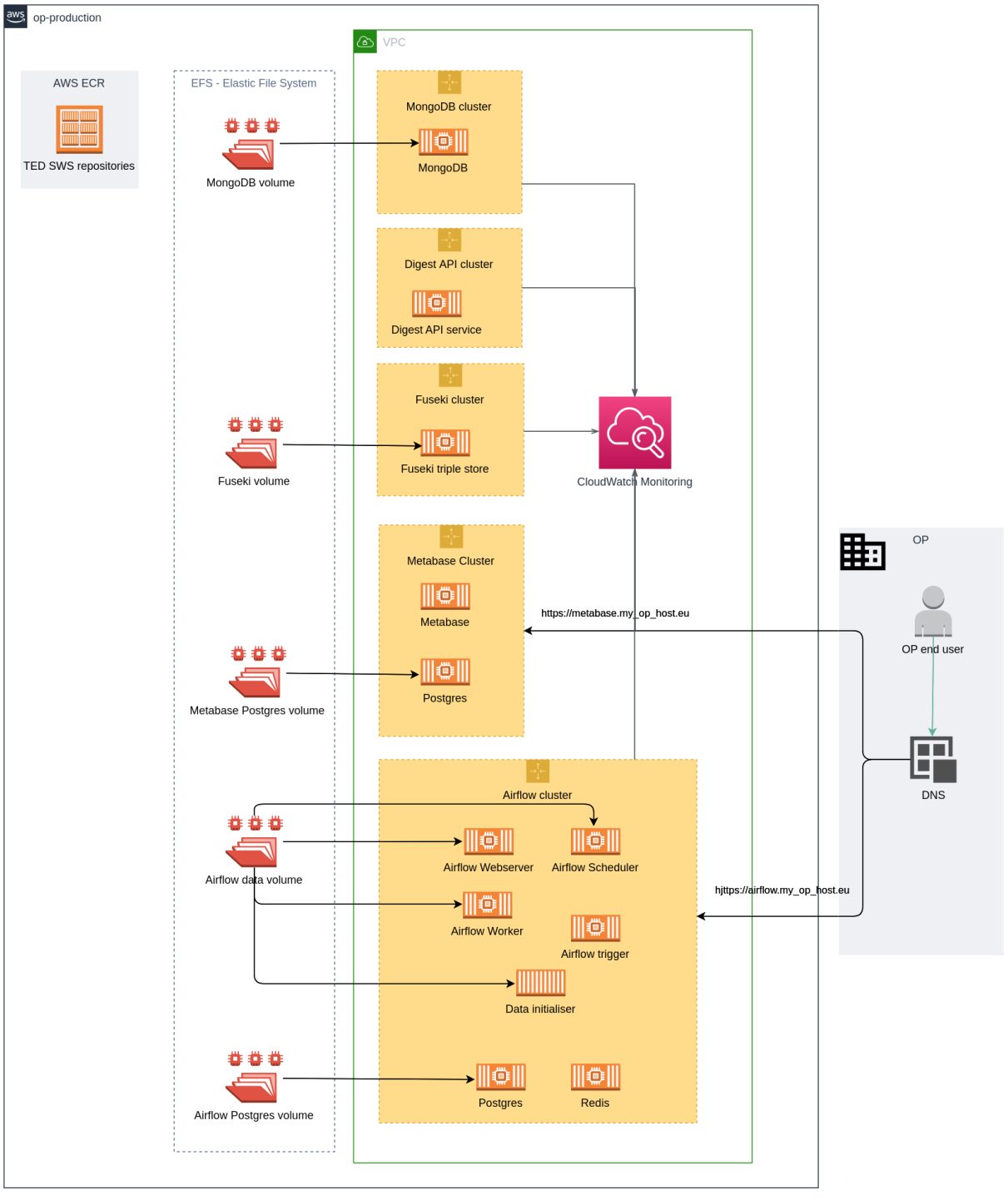
Infrastructure Architecture

The diagram below depicts the infrastructure architecture of the TED-SWS system. It is written with a limited understanding of the constraints and requirements of the OP AWS infra team.

There are three contexts depicted:

- op-production: the Publications Office production environment,
- DIGIT AWS Cloud,
- DIGIT organisational environment, and in particular, the OP environment, where the end-users of the TED-SWS system are expected to be.

The two contexts on the right are meant for controlling the DNS and the domain address (using Route53 and other DNS technologies), where the TED-SWS services will be made available to the end users. The central context depicts the TED-SWS system deployment and will be further the focus of the description.



The next set of diagram fragments is meant as an attention aid when describing each section of the overall diagram.

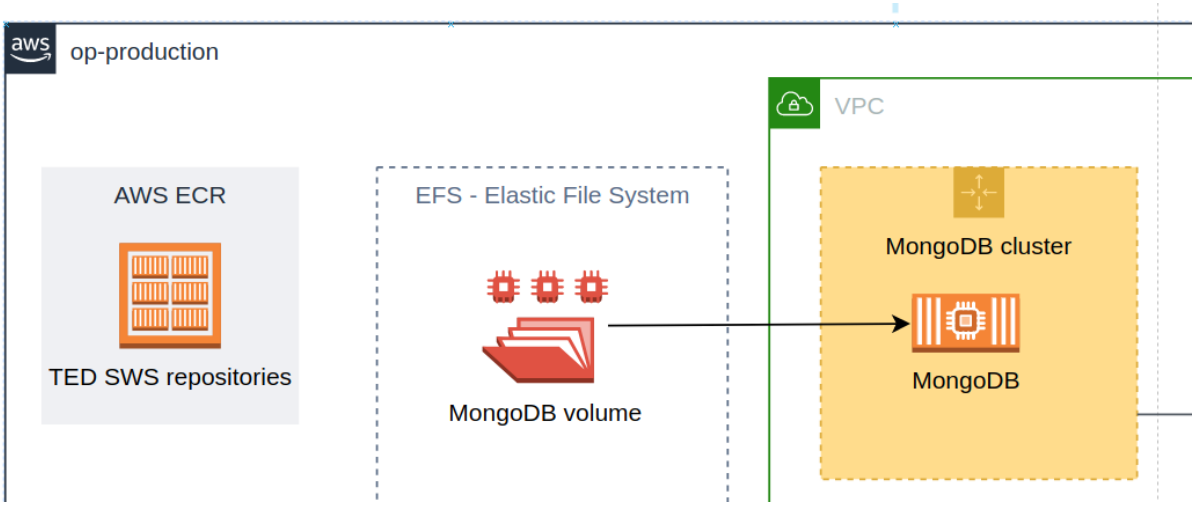
The TED SWS system has been developed using *docker-compose* technology. The docker-compose setup used for development has been ported to AWS cloud services. Naturally, the most solicited service is the Amazon Elastic Container Service (ECS), supported by a suite of *Docker images* (built and) pushed into a dedicated Elastic Container Registry (ECS). To monitor the state of each container, we connect the CloudWatch service.

Clusters

We foresee four clusters deployed in two availability zones perfectly mirroring one another.

MongoDB cluster. It is formed of one container of MongoDB database. Currently this database is run in an ECS cluster rather than using a managed service.

Currently, we cannot switch to Amazon DocumentDB (which is the correspondent of MongoDB) as this solution has not yet been tested, and the main development was performed with MongoDB. As this implies additional development costs, we deploy this cluster on a single large instance. An EFS volume is attached to the container.



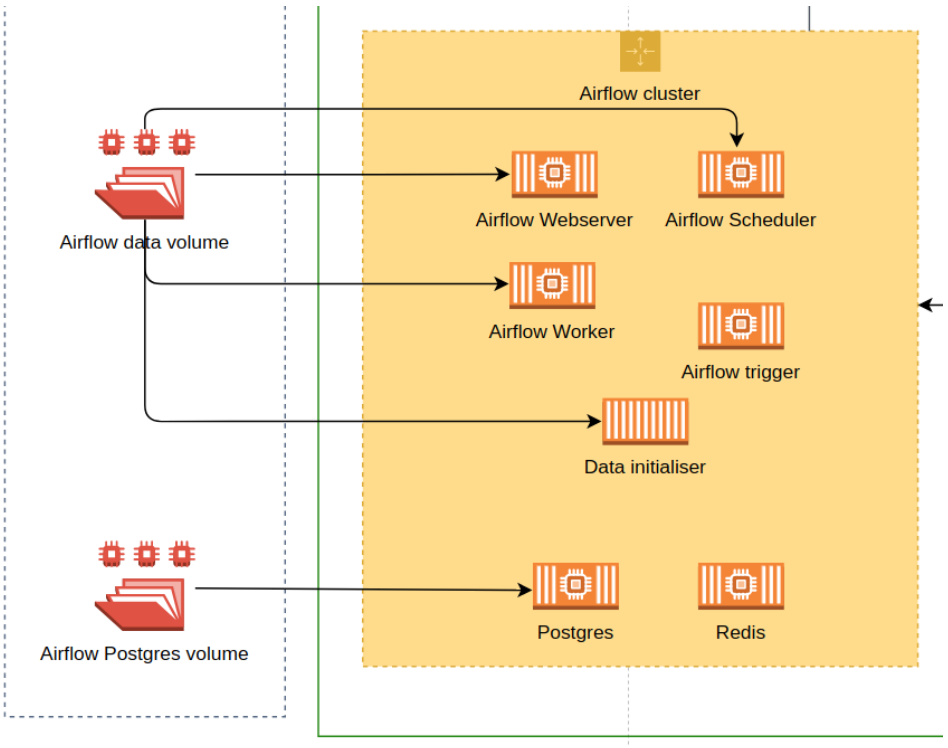
Airflow cluster. Airflow is a suite of several containers:

- Airflow components: webserver, scheduler, worker
- Initialisation containers: trigger & data initialiser
- Dependent services: Redis & Postgres

The Airflow components need to share a common data volume containing:

- The DAGs code
- The TED-SWS code
- The environment variables configuration

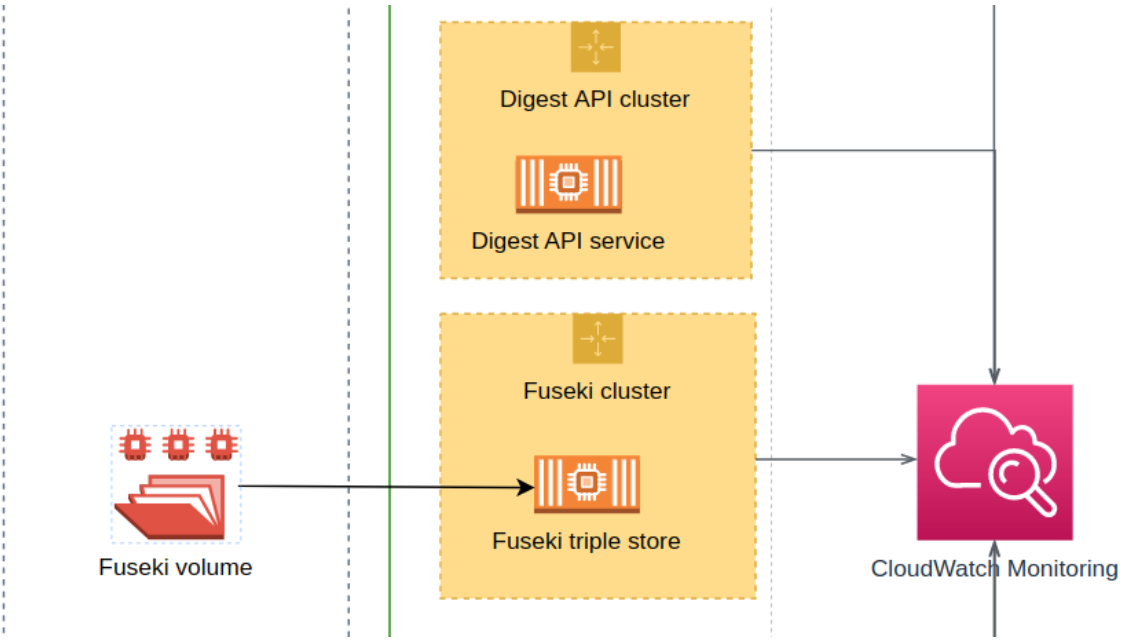
Postgres is a service that all Airflow components depend on. It runs in a container with a dedicated EFS drive attached.



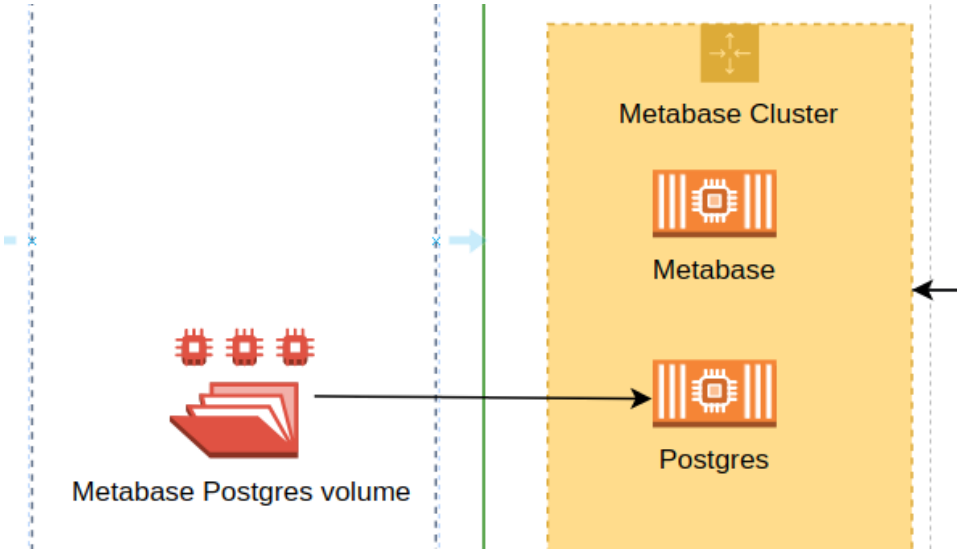
Fuseki cluster. It is a single container cluster running Fuseki triple store. A dedicated EFS drive is attached to it.

Digest API cluster. Digest API is a simple stateless service, yet vital for the XML2RDF transformation process implemented in the TED-SWS.

It needs to be exposed on a fixed domain address. When the TED-SWS-artefacts are loaded into the TED-SWS database, the assigned domain address must be injected into the RML rules.



Metabase Cluster. It is a cluster with two images, one is the Metabase application and the other one is a Postgres database.



Communication

All clusters need to communicate with one another in a common network. This is indicated in the overview diagram with a green VPN bounding box. In addition, two essential services must be accessible by end users: Airflow Webservice and Metabase user interface. This is represented in the diagram below. On the right side, an end used placed on the premises of the OP organisation. A DNS that binds correctly to the needed services to a dedicated domain name. The user accesses the services at a specified address.