# Synthetic procedure to replicate the INFN Open Access Repository
(based on Invenio v3 and Zenodo)

M. Fargetta[1)], R. Rotondo[1)], R. Barbera[2,1)]

1) Italian National Institute of Nuclear Physics, Division of Catania - Italy
2) Department of Physics and Astronomy “E. Majorana” of the University of Catania - Italy

## Pre-requisites

This guide is intended for people already acquainted with web programming and having a previous knowledge of how GitHub software repositories, Docker containers and Kubernetes clusters work. A knowledge of how to set secure (digital-certificate-enabled) web transactions is also recommended.

Furthermore, to be able to follow in detail all the steps described in this document, readers must make sure they already have: (i) accounts both on GitHub and Docker Hub and (ii) access either to a real or a virtualised infrastructure.

Finally, in order to ensure the compliance of your repository with FAIR principles, it would be a big plus is your organisation holds a Digital Object Identifier (DOI) prefix to be assigned to the repository, released either by DataCite or any other equivalent provider.

## Introduction

This short guide briefly outlines the steps to be taken to configure and deploy a clone of the INFN Open Access Repository (OAR), which is in turn based on Invenio version 3 and Zenodo software.

Please, note that two separate installations have to be done and maintained:

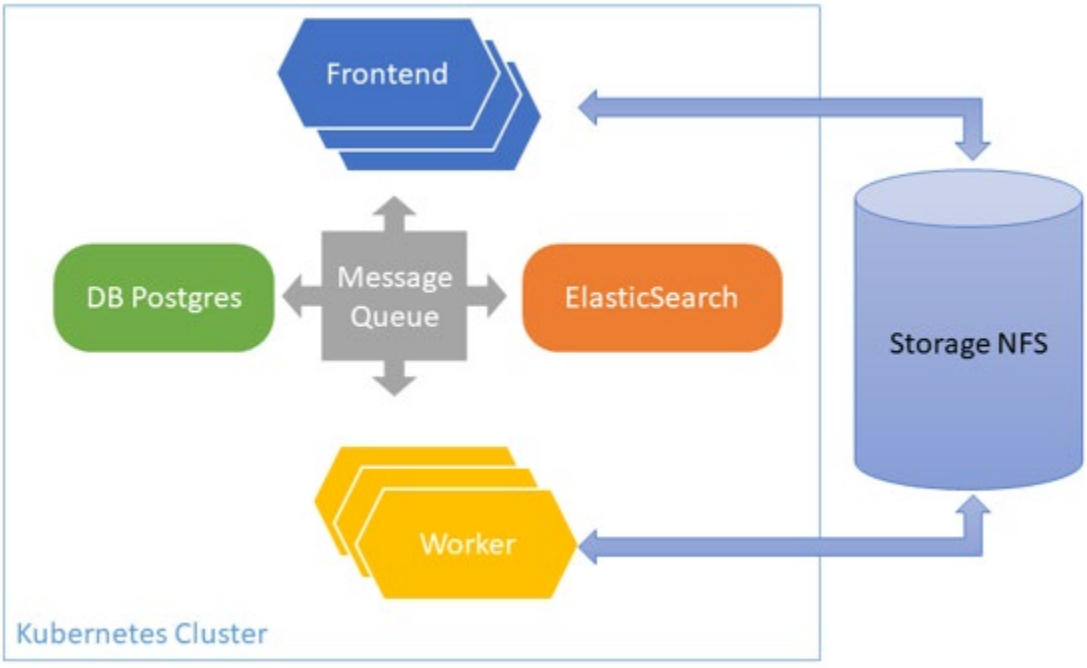
- First, the installation of a development instance of INFN OAR on a local computer (even a laptop or notebook) to configure/customise the graphic theme and layout of the repository.
- Then, the deployment of the production-level containerised version of the properly configured/customised development instance on a Kubernetes cluster.

The workflow to configure, operate and maintain your clone of the INFN OAR is indeed the following:

1) Configure/customise/correct things (graphic theme, e.g. header and footer, CSS, colours, logos, etc.) in the locally installed development version until you are satisfied with the changes.
2) Create/update the Docker containers of the repository and store it on Docker Hub.
3) Deploy the containers from Docker Hub on the Kubernetes cluster.
4) Start/restart the container to run the new/latest version of the software and make the repository available to all end-users.
5) Go to 1) in case a new change is needed, or an error is spotted.

## Step 1 – Create and deploy a Kubernetes cluster

- Install Kubernetes (see https://kubernetes.io/docs/setup/independent/install-kubeadm/) and Docker (see https://kubernetes.io/docs/setup/cri/) as container runtime interface on a (either real or virtualised) cluster. The deployment layout of INFN OAR is shown in the figure below, so in order to cope with its level of complexity make sure that your Kubernetes cluster has 1 master and at least 3 nodes.



- Connect both master and nodes to a NFS server (see, for example, https://wiki.archlinux.org/index.php/NFS).
- Note: if human resources allow it, this step may go in parallel with Step 2 below.

## Step 2 – Install and configure the development instance

- Clone the software repository https://github.com/osct/zenodo and use this guide https://zenodo.readthedocs.io/en/latest/ to install and run the development instance on a local computer at your premises.
- Configure/customise the development instance. As examples:
  - All files related to the graphic theme are under:
    - https://github.com/osct/zenodo/tree/master/zenodo/modules/theme
  - Header and footer of pages can be customised modifying these files:
    - https://github.com/osct/zenodo/tree/master/zenodo/modules/theme/templates/zenodo_theme
  - The INFN OAR logo to be replaced with yours can be found in this folder:
    - https://github.com/osct/zenodo/tree/master/zenodo/modules/theme/static/img
- Note: if human resources allow it, this step may go in parallel with Step 1 above.

## Step 3 – Create the Docker container from the development instance

- Clone the software repository https://github.com/osct/zenodo-apache-shibboleth-container
- Customise it and build the container of your repository with Docker (read the documentation at https://docs.docker.com/).
  - The submodules should be modified to include your customised version of Zenodo.
- Upload and publish the container of your repository on your account on Docker Hub.

## Step 4 – Deploy the containers on the Kubernetes cluster

- Clone the software repository https://github.com/osct/zenodo-kubernetes
- Create/configure resources and microservices to deploy your repository on Kubernetes using yaml configuration files you will find in the above software repository.
- Configurations need to be adapted to your infrastructure. As examples:
  - The correct values of IP addresses, ports, storage size and other parameters need to be specified in the following files:

        - https://github.com/osct/zenodo-kubernetes/blob/master/zenodo-frontend-deployment.yaml
        - https://github.com/osct/zenodo-kubernetes/blob/master/zenodo-worker-deployment.yaml
        - https://github.com/osct/zenodo-kubernetes/blob/master/zenodo-worker-low-index-deployment.yaml
    - The Zenodo configuration file:
        - https://github.com/osct/zenodo-kubernetes/blob/master/config/zenodo-config.yaml

      should be customised with the correct values of your instance (administration and support email addresses, default community, DOI prefix, etc.).
- Make sure frontend and worker containers use the Docker container created during Step 3.
- Connect to frontend to complete the repository initialisation as described in this guide https://zenodo.readthedocs.io/en/latest/installation.html#initialization

## Conclusions

Should you have any problems in installing and deploying your repository using the present guide, support can be sought by writing at librarian@openaccessrepository.it.