Remote Workflow Enactment using Docker and the Generic Execution Framework in EUDAT

Asela Rajapakse
Max Planck Institute for Meteorology

EUDAT receives funding from the European Union's Horizon 2020 programme - DG CONNECT e-Infrastructures. Contract No. 654065

The Team

- Emanuel Dima (Uni Tuebingen)
- Pascal Dugenie (CINES)
- Weu Qiu (Uni Tuebingen)
- Asela Rajapakse (MPI-M)
- Luca Trani (KNMI)

Presentation outline

A problem facing European Research Infrastructures

- The trend is towards proliferation and diversification of Research Infrastructures in Europe.

- All RIs are facing data management challenges:

  - How to handle the growing amount of data?
  - How to find data sets?
  - How to make the most of the data?

Stakeholders can benefit from pan-European solutions and synergies can be exploited!

EUDAT's objective

- EUDAT (short for European Data) is to fulfill the vision of the High Level Expert Group on Scientific Data[2] and aims

"to provide an integrated, cost-effective, and sus-tainable pan-European solution for sharing, preser-ving, accessing, and performing computations with primary and secondary research data."

– ***EUDAT2020 Description of Work***

[2] High Level Expert Group on Scientific Data:
Riding the wave: How Europe can gain from the rising tide of scientific data (2010)
Final report to the European Commission

EUDAT

EUDAT2020 - 35 Partners

CSC

UNINETT Sigma2

CLARIN

Barcelona Supercomputing Center

Science & Technology Facilities Council

Max Planck Computing & Data Facility

CERFACS

UCL

Jisc

CINECA

Trust-IT Services Ltd

grnet

BioSense

CINES

KIT

DKRZ

LIBER

CERN

umweltbundesamt

University of Helsinki

Eberhard Karls Universität Tübingen

DANS

EMBL

epcc

escience data factory

SNIC

INGV

Lund University

KTH

Jülich

PSNC

Royal Netherlands Meteorological Institute

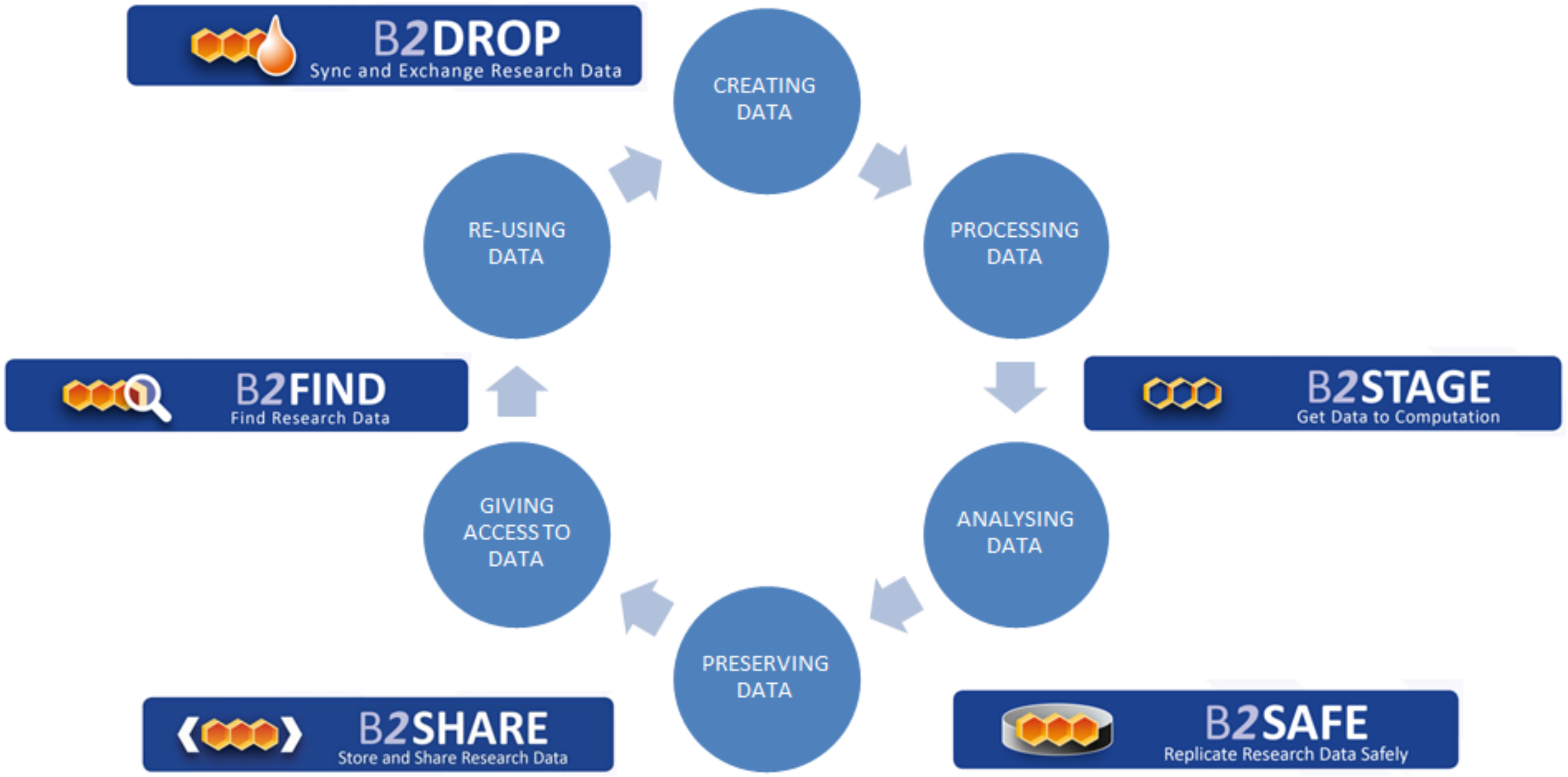
Max-Planck-Institut für Meteorologie

SURF SARA

GFZ

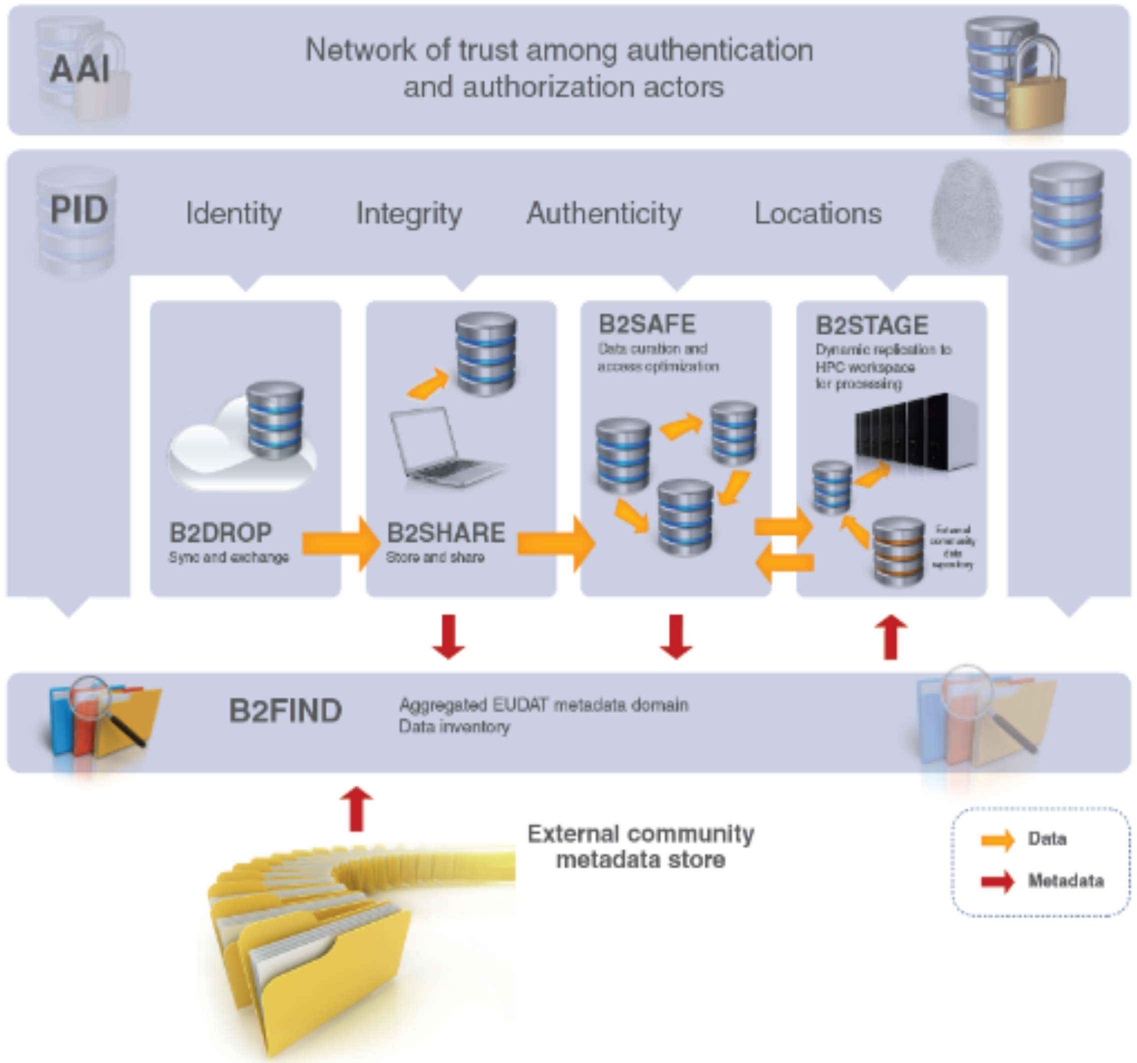
The EUDAT Service Suite

- Meant to solve the basic data service requirements of participating RIs.



EUDAT2020 Service Suite Overview: https://www.eudat.eu/services

The EUDAT Service Suite


AAI
Network of trust among authentication and authorization actors
PID
Identity
Integrity
Authenticity
Locations
B2DROP
Sync and exchange
B2SHARE
Store and share
B2SAFE
Data curation and access optimization
B2STAGE
Dynamic replication to HPC workspace for processing
External community data repository
B2FIND
Aggregated EUDAT metadata domain
Data inventory
External community metadata store
Data
Metadata

Diagrams taken from the official EUDAT presentation.

EUDAT2020 Service Suite Overview: https://www.eudat.eu/services

The EUDAT Service Suite

Although the basic services are being taken up by communities, they are still under constant develop-ment.

EUDAT and the GEF

- The Generic Execution Framework started as the topic of a research activity in EUDAT 1.
- GEF development is continuing in EUDAT 2 and is conducted in:
  - Work package 5: Service Building
  - Work package 8: Data Life Cycle across Communities, necessitating research for possible GEF extensions

The ideas behind the GEF

The GEF is meant to address the following problem:

- With growing data size it becomes increasingly costly and time consuming to transfer data from their storage to a data processing location.

The ideas behind the GEF

Move the tools, not the data!

- Minimize data transfer cost when no heavy computation is required by containerizing the software tools required for data processing and moving them closer to the data.

Reproduce results more easily by reusing containers!

- The same container and software tool templates can be used over and over ensuring the software environment and the processing tools remain the same.

The technology powering the GEF: Docker

“Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, […] system tools, system libraries.” **Docker Website**

- Docker containers are based on Docker images which can be viewed as templates for container instances.
- Every Docker image is built on top of a base image, e.g. Ubuntu or BusyBox.

We will look at an example later!

The technology powering the GEF: Docker

- Containers are run by the Docker Engine on a virtualized Linux system
- Containers share the host kernel
- Versions for Mac OS and Windows are available now, but they also run a virtualized Linux system

If it can be installed on a Linux system, chances are good it can be containerized with Docker.

We will look at an example later!

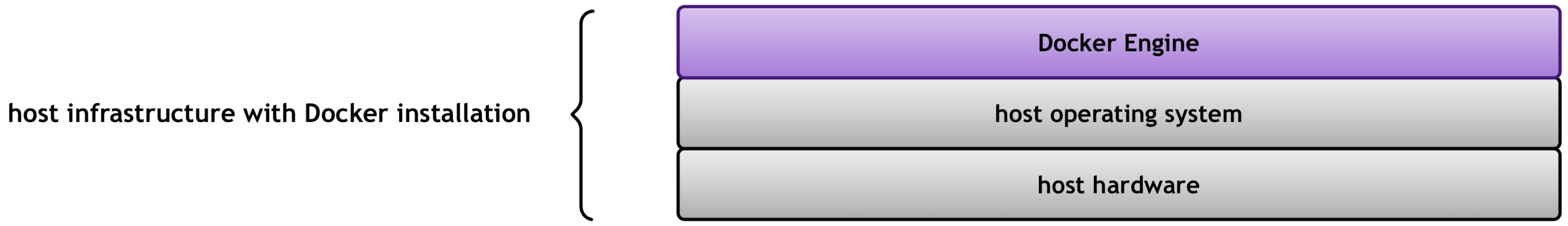
GEF services/Docker containers

The GEF relies on so-called GEF services that are customized by the user to perform the required tasks:

- Docker images specifically annotated to allow handling by the GEF.
- GEF service instances are Docker containers that are spun up for execution close to the data.
- User communities are solely responsible for the contents of their images.

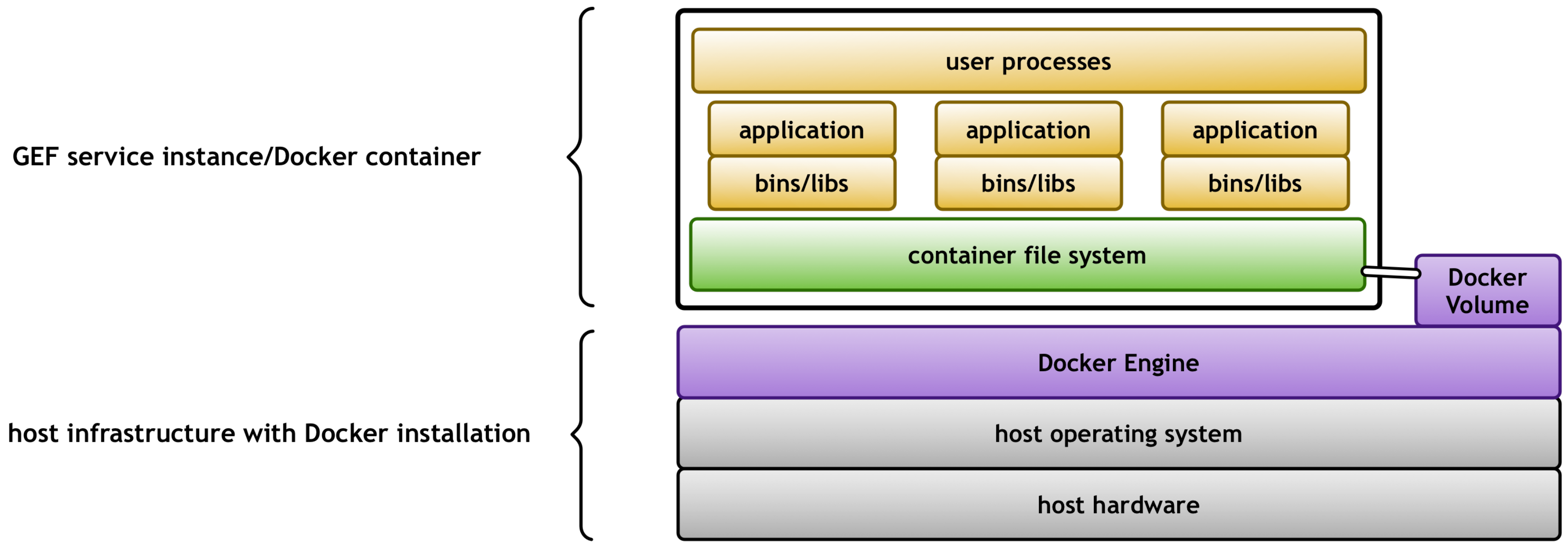
GEF services/Docker containers

The schematic view of a GEF service instance running on a Dockerized host.



GEF services/Docker containers

The schematic view of a GEF service instance running on a Dockerized host.



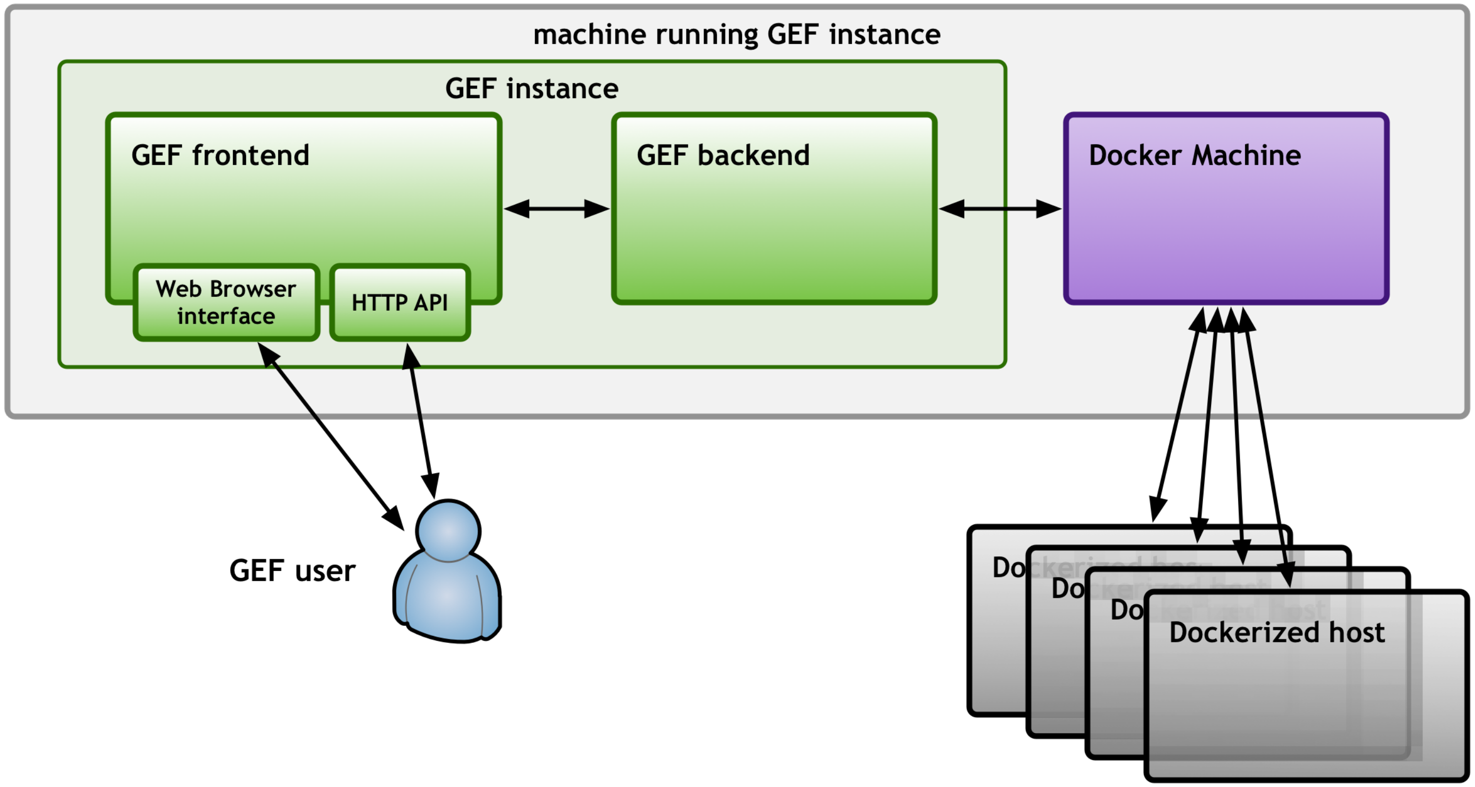
GEF services/Docker containers

- Processing results from the container can be stored in a Docker Volume managed by the Docker Engine.
- Docker Volumes can be shared between containers and even be containerized to move them to a different node in a cluster.

EUDAT

A GEF instance

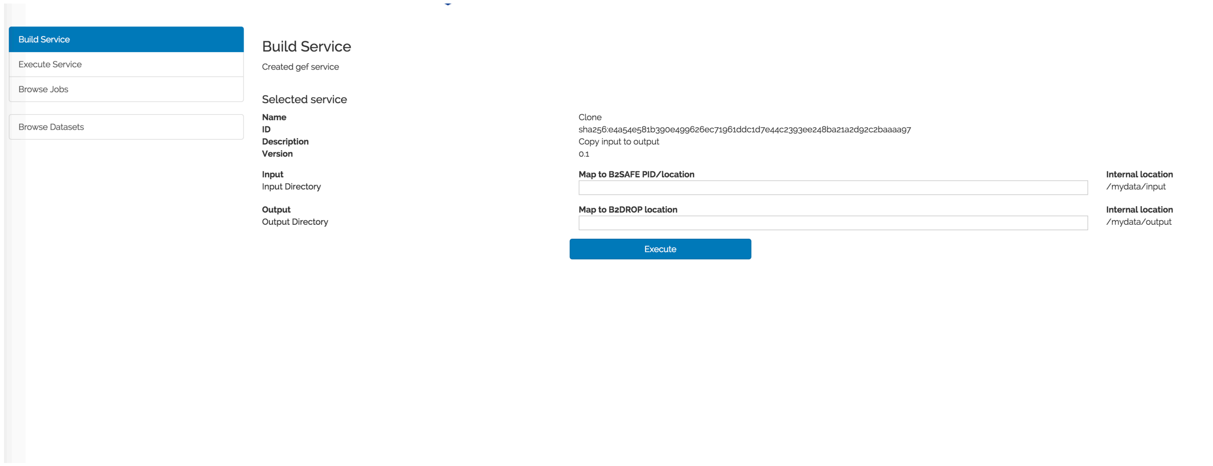
The container/GEF service invocations on the hosts are controlled by a Docker Machine integrated with a GEF instance.



Workflow integration

Users have two methods to communicate with a GEF instance:

- through a graphical user interface



Build Service

Created gef service

Selected service

Name	Clone
ID	sha256:e4a54e581b390e499626ec71961ddc1d7e44c2393ee248ba21a2d92c2baaaa97
Description	Copy input to output
Version	0.1

Input	**Map to B2SAFE PID/location**	**Internal location**
Input Directory		/mydata/input
Output	**Map to B2DROP location**	**Internal location**
Output Directory		/mydata/output

- or access programmatically through an HTTP API.

Reproducibility of processing results

The capability to reproduce the results of a data processing job is of major concern to scientists:

- Instantiation from a single GEF service image is to make this easier.
- In the future, using a central repository for all GEF service images will foster reuse of existing images.

The GEF image repository

In the future, all GEF service images are to be stored in a central repository for trusted and tested images:

- This will be a dedicated Docker Hub repository.
- The GEF will only accept images from this repository.
- The service images will be immutable once stored in the repository (versions are possible).
- The repository is publicly accessible and all stored images are open to the peer-review process.

The GEF image repository

For now, during development and once the GEF becomes a full prototype, GEF service images are not public yet and access to the GEF prototype will be restricted.

The Dockerization of a workflow

Using a branch of ObsPy, a Python framework for processing seismological data, to extract metadata from an archive of waveform data.

- Install the mSEED-QC branch of ObsPy.
- Configure the extraction through a JSON file.
- Run the WFCatalogCollector.py script.

Thanks to Luca Trani and Mathijs Koymans from KNMI (Royal Dutch Meteorological Institute)

EUDAT

The Dockerization of a workflow

The bulk of the work is creating the image, most easily through a Dockerfile.

```
 # Ubuntu 14.04 base image
FROM ubuntu

MAINTAINER Mathijs Koymans

# Create an app directory
RUN mkdir -p /usr/src/collector
WORKDIR /usr/src/collector

# Copy the application source to this directory
COPY . /usr/src/collector

# Get necessary software (e.g. git to fetch
# the branch, vim for interactive editing)
RUN apt-get update
RUN apt-get install -y git
RUN apt-get install -y libxml2-dev libxslt1-dev
RUN apt-get install -y python-pip
RUN apt-get install -y libfreetype6-dev
RUN apt-get install -y pkg-config
RUN apt-get install -y vim
```

EUDAT

The Dockerization of a workflow

```
# Numpy needs to be installed manually
# Also include PyMongo if the database is used
RUN pip install numpy
RUN pip install pymongo

# Fetch the branch from ltrani@GitHub
RUN git clone https://github.com/ltrani/obspy.git

# Set the work directory to inside the fetched ObsPy source
WORKDIR /usr/src/collector/obspy

# Switch to and install the branch
RUN git checkout mseed-qc
RUN pip install -e .

# Set the workdir back to the main source
WORKDIR /usr/src/collector
```

The Docker image is named and built.

```
$ docker build -t collector:1 .
```

This takes care of installation. We now only need to run the container and start the script.

EUDAT

The Dockerization of a workflow

The container is started with a /mnt directory linking it to the archive on the host.

```
$ docker run --name collector_container -v /path/to/archive:/mnt -d -t
collector:1
```

We open an interactive terminal to the container and can modify the configuration file for our extraction.

```
$ docker exec -it collector_container bash
root@50e3f57e16e6:/#  vi config.json
```

Similarily, the Python script inside the container is executed with the link to the archive.

```
$ docker exec collector_container bash -c "python WFCatalogCollector.py --
dir /mnt/~"
```

The Dockerization of a workflow

This merely was to give you a basic idea of what Dockerization looks like.

Security through guidelines and trust

Many people are still concerned about Docker security.

- When configured appropriately Docker is reason-ably secure.
- We will provide a set of guidelines or best practices to users to ensure that Docker images created by them do not pose security risks.

These guidelines and the measure of trust associated with accredited EUDAT users will guarantee an acceptable level of security.

The current status of the integration with existing services

GEF service containers are to obtain their data through EUDAT services. For now,

- we are developing dedicated access containers to connect to B2SHARE and to B2DROP in our current implementation of a first use case,
- and we aim for full authorization and authentication through B2ACCESS.

Collaboration with Work Package 8

Results from the Joint Research Activity in Work Package 8 will feed into the GEF development as they appear along the way. We envision work on support for

- a provenance model,
- directives,
- dynamic data,
- and semantic annotation.

The next steps

The next major goals:

- Set up a prototype instance that can be used for our use cases.
- Further addressing security concerns (repository and best practices for prospective users).

Come forward and scrutinize it!

The GEF is being designed after community requirements. We are still looking for new requirements and use cases.

Asela Rajapakse
asela.rajapakse@mpimet.mpg.de

EUDAT

Thank you for your attention!

Questions?