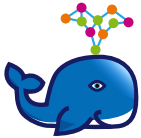


Dockerizing Linked Data

Knowledge Base Shipping to the Linked Open Data Cloud

Natanael Arndt, Markus Ackermann,
Martin Brümmer and Thomas Riechert

Semantics, Vienna 2015

September 16th 2015

Introduction

Current Situation

- Low availability of popular SPARQL endpoints
- Setting up a local SPARQL endpoint is a complex task
+ Our approach could help to improve reproducibility of experiments with/on RDF data

Introduction

Setting Up a SPARQL Endpoint

1. Installation of a triple store
2. Loading the data to be published into the triple store
3. Setting up a (publicly available) SPARQL endpoint
4. Providing a presentation application to support the exploration of the knowledge base
5. Ensuring de-referencability of IRIs occurring in the published knowledge base
6. Maintain the setup and ensure its availability

Introduction

Linked Data Stack

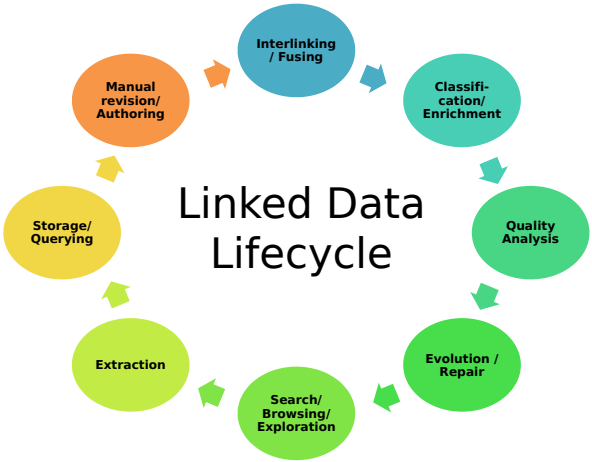


Figure: The Linked Data Stack `http://stack.linkeddata.org`

Introduction

Linked Data Stack

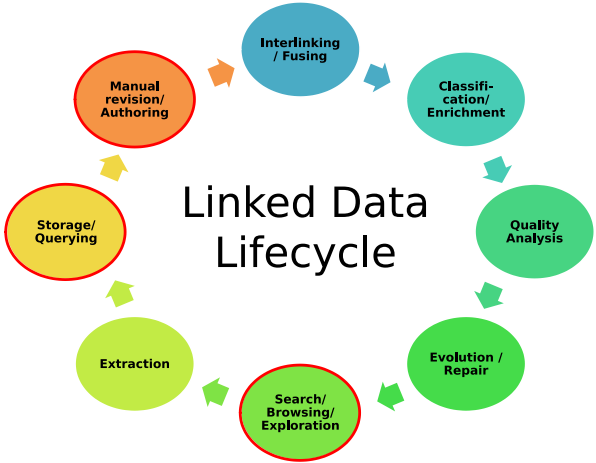


Figure: Position in the Linked Data Stack `http://stack.linkeddata.org`

Introduction

Setup a Local SPARQL Endpoint

- The setup is a time consuming task
- Requires a certain level of technical knowledge
- Need of DevOp-competences or an administrator

Introduction

Other Approaches

- What about a hosted Linked Data publishing infrastructure e.g. `http://dydra.com/`

Introduction
Other Approaches

- What about a hosted Linked Data publishing infrastructure e.g. `http://dydra.com/`
  - Still depend on the availability of the service
  - No controlled environment
  - Privacy issues?

Introduction
Other Approaches

- What about a hosted Linked Data publishing infrastructure e.g. `http://dydra.com/`
  - Still depend on the availability of the service
  - No controlled environment
  - Privacy issues?
- Why aren't we just using any packaging system?

Introduction

Other Approaches

- What about a hosted Linked Data publishing infrastructure e.g. `http://dydra.com/`
  - Still depend on the availability of the service
  - No controlled environment
  - Privacy issues?
- Why aren't we just using any packaging system?
  - limits in platform choice, portability

Introduction
The DLD Approach

- Containerised Linked Data publication infrastructure
- Benefits regarding the maintainability and ease of setup
- Modularisation of individual components, following the micro service principle

Background and Tooling

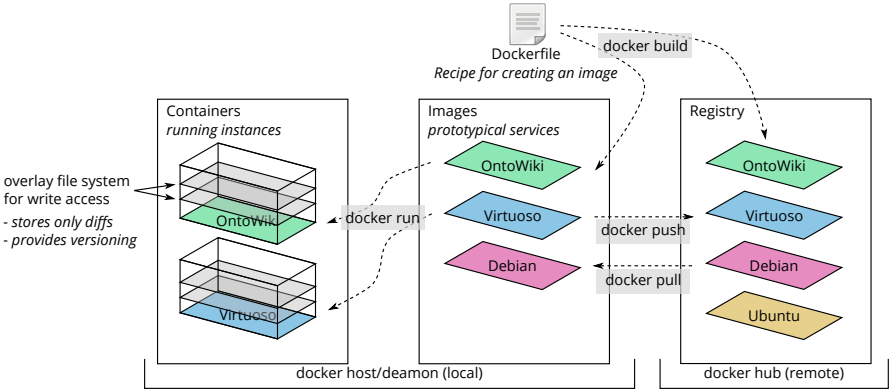
Container-Based Virtualisation

- Provides an isolated execution environment for applications
- Keep all necessary dependencies together to run a process
- Share the complete hardware and core Operation System (OS, in contrast to full- and para-virtualisation)
- Still each container has its own filesystem
- Can allow reproducibility of execution environments

Examples of container-based virtualisation:
FreeBSD jail, Linux Containers (LXC), Docker

Background and Tooling

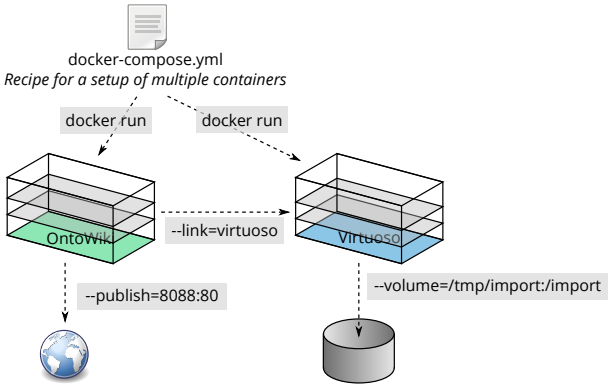
Docker



Docker is a container-based virtualisation system running on mayor OS: Linux, Mac OS X and Windows and on mayor cloud platforms (e.g. Amazon EC2, Microsoft Azure, Google Cloud Platform)

Background and Tooling

Docker Compose



Docker Compose (former known as *fig*), allows to combine multiple micro services to a bigger service

Architecture

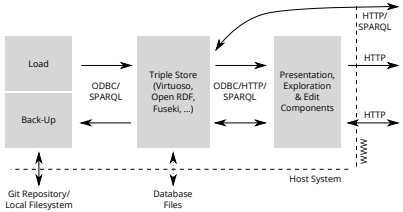


Figure: Architecture and data-flow of the containerised micro services for publishing knowledge bases

The following roles:

- Triple Storage
- Load & Back-Up
- Presentation & Publication

Architecture

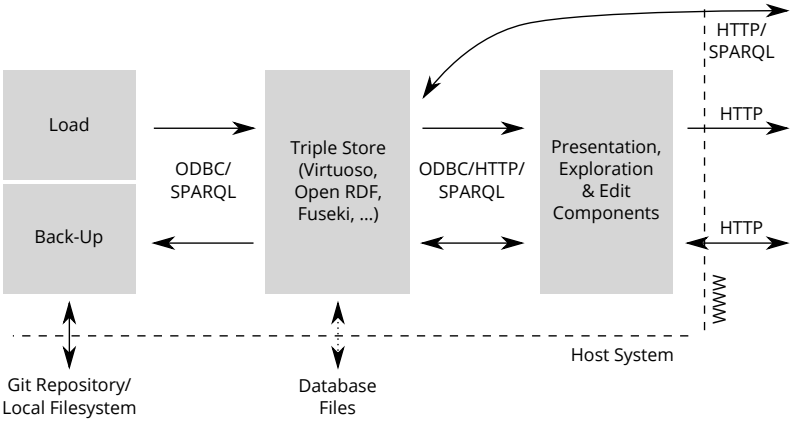


Figure: Architecture and data-flow of the containerised micro services for publishing knowledge bases

Architecture

Triple Storage

- Persistent storage is implemented using volumes
- Provides SPARQL interface
- In the current architecture the is the limitation of only one Triple Store

Load & Back-Up

- A load component is responsible for pre-loading the store
- Data is taken from a local volume or an online source
- Backup Containers dump the data to a local volume or an online repository

Presentation & Publication

- Provide exploration interfaces,
- Editing/curation interfaces or
- Domain specific views

Architecture

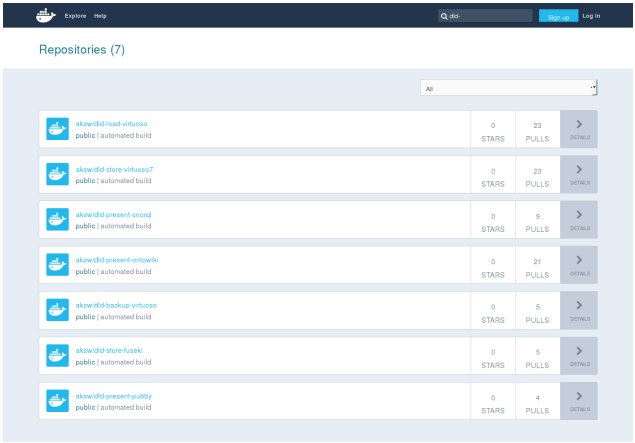


Figure: DLD containers at the docker hub, searching to “dld-”
`https://hub.docker.com/search/?q=dld-&isAutomated=0`

Architecture

Provisioning of a Container Setup

```
datasets:
    dbpedia-2015-endpoint-some-set:
        graph_name: "http://dbpedia.org"
        file: "dbp-2015-a.rdf"
    dbpedia-2015-endpoint-some-other-set:
        graph_name: "http://dbpedia.org"
        location: "http://dbpedia.org/files/dbp-2015-b¬
          .rdf"

environment_global:
    DEFAULT_GRAPH: "http://dbpedia.org"
```

Listing 1: A DLD configuration file for publishing DBpedia data ...

Architecture

Provisioning of a Container Setup

```
components:
    load: aksw/dld-load-virtuoso
    store:
        image: aksw/dld-store-virtuoso7
    present:
        ontowiki:
            image: aksw/dld-present-ontowiki
            ports: ["8088:80"]
```

Listing 2: … using a setup with Virtuoso triple store, a suitable load component and OntoWiki for presentation/editing

Architecture

Provisioning of a Container Setup

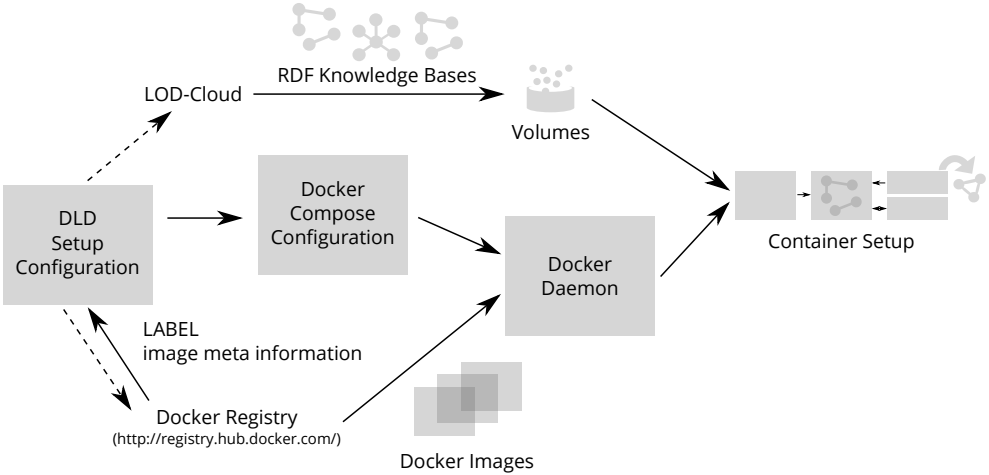


Figure: Workflow for creating a container setup

Discussion and Conclusion

- Provide an Infrastructure for bootstrapping Linked Data publication infrastructure
- Lower the entry barrier for setting up a SPARQL endpoint
- Improved portability of Linked Data setups to transferring a `dld.yml` file

Future Work

- Embed the container and setup descriptions in DataID [1]
- Integration into the Linked Data Stack `http://stack.linkeddata.org`
- Provide more images and ready to run setups
- Improve the usability for domain experts
- Provide a UI for configuring the DLD-Setup-Tool

Discussion and Conclusion

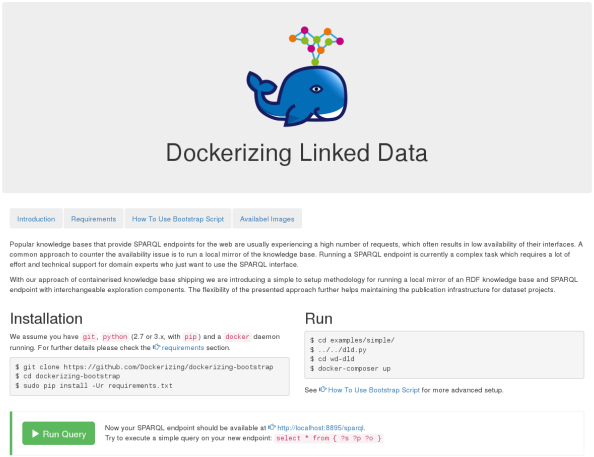


Figure: Homepage with an example at: `http://dld.aksw.org/`

Thank you for your attention!

```
$ ./dld.py
$ cd wd-dld
$ docker-compose up
```


M. Brümmer, C. Baron, I. Ermilov, M. Freudenberg, D. Kontokostas, and S. Hellmann.
DataID: Towards semantically rich metadata for complex datasets.
In *Proceedings of the 10th International Conference on Semantic Systems, SEM '14*, pages 84–91. ACM, 2014.