Docker2RDF: Lifting the Docker Registry Hub into RDF

Conference Paper · June 2017

DOI: 10.1109/SERVICES.2017.15

Ahmed Ben Ayed
University of Tunis El Manar
7 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Frederique Laforest
Université Jean Monnet
113 PUBLICATIONS **455** CITATIONS
SEE PROFILE

Wajdi Louati
University of Sfax
36 PUBLICATIONS **702** CITATIONS
SEE PROFILE

Ahmed Hadj Kacem
Faculté des Sciences Économiques et de Gestion de Sfax
192 PUBLICATIONS **595** CITATIONS
SEE PROFILE

Docker2RDF: Lifting the docker Registry Hub into RDF

Ahmed Ben Ayed*,†, Julien Subercaze*, Frederique Laforest*, Tarak Chaari†, Wajdi Louati†, Ahmed Hadj Kacem†
*Univ Lyon, UJM Saint-Étienne, CNRS, Laboratoire Hubert Curien UMR 5516, France. Email: firstname.lastname@univ-st-etienne.fr
†Univ Sfax, ReDCAD, Sfax, Tunisia. Email: firstname.lastname@redcad.org

***Abstract*—Docker is the most popular open-source software for managing software containers. Docker is widely used in the industry and is becoming the de-facto standard. In this paper, we tackle the issue of bringing Docker into the Linked Open Data Cloud to enhance docker images descriptions and search. We first present the construction of the ontology that describes Docker images. This ontology is interconnected with DBPedia and reuses SIOC and prov vocabularies. We then describe the automated process that automatically extracts data from the Docker Hub Registry to populate the aforementioned ontology. We also present a Web platform that provides a SPARQL endpoint as well as analytical insights into the dataset.**

***Index Terms*—Docker, RDF, Datalift, LOD Cloud, dataset.**

I. Introduction

Component-based software engineering promotes components reuse in the software development process for economical and rapidity reasons [6]. The last revolution in components reuse is the arrival of Docker [10]. Docker is a popular open-source program that enables a Linux application and its dependencies to be packaged into containers. Containers isolate applications from each other on a shared operating system (OS). Docker containers are created using images. A Docker image is a read-only template used to create Docker containers. The large adoption of Docker in companies and industry makes it the defacto industrial standard. Recent statistics show that over 460.000 applications have been dockerized [1]. Docker is also a perfect tool for researcher in order to foster research reproducibility [4].

To ease the reuse of Docker images, the Docker community has built a shared repository called the Docker Hub. The Docker Hub is a free open source service that freely stores and distributes built images. From the same statistics source, more than 4 billion containers have been pulled from the Docker Hub. The Docker Hub offers a service for images search that is based on images names. Browsing among the available images is also possible, but no advanced tool is proposed. It would be of the utmost practical interest to integrate Docker images descriptions into the Linked Open Data Cloud (LOD Cloud) so that searching Docker images could benefit from the Semantic Web powerful competencies. For example, searching for all DBMS images today requires the user to make many keyword searches and browsings. With the semantic Web and the LOD Cloud, semantic search allows to identify one concept that can be automatically enriched with linked concepts. The aim of this work is to provide a semantic RDF description of each Docker Hub entry so that the end user can benefit from the Semantic Web stack. The LOD Cloud has gained extensible since the adoption of RDFa and microformats [3] and companies are more ready than ever to adopt Semantic Web technologies.

In this paper, we present a complete approach to unlock the Docker images silo and to integrate it in the LOD Cloud, by connecting to DBpedia, the most linked resource in the LOD Cloud [12]. In Section II, we describe Docker Images, the Docker Hub and give insights into Docker and its ecosystem. We also provide a large set of statistics on the dataset, following the current LOD trend [2], [8].

The images stored in the Docker Hub are accompanied with a set of metadata provided by the authors. These metadata include some general information about the content of the image and also information on how to reuse the image. Based on the structure of these metadata, we define a Docker Ontology that is connected to `DBPedia` and uses `SIOC` and `prov` ontologies. This process is described in Section III. We then devise Web Extraction techniques to structure data from the Docker Hub in order to populate the ontology defined previously. We describe our solution in Section IV. Section V presents the online presence of our research, this includes a SPARQL endpoint, a Web interface to access statistics as well as to navigate and download the dataset. Section **??** presents sample queries and usage of the dataset. Finally, Section VI summarizes the work and provides opportunities and future directions.

II. Docker Hub

The Docker Hub[2] is a Web application that stores and distributes Docker images. It is the most convenient way to find and discover existing docker images. The Docker Hub is open-source, under the Apache license. As shown in Figure 1, each web page of the Docker Hub describes a docker image. It is composed of several elements:

1) *image name*: it is usually chosen with care as it is the only item presented in the Docker Hub List Page.
2) *star*: The image name is associated with a star the user can highlight when this image is one of his favorites.

[1] http://www.coscale.com/blog/docker-usage-statistics-increased-adoption-by-enterprises-and-for-production-use

[2] https://hub.docker.com/

3) *date of publication*: the date when the current version of this image has been made available on the Docker Hub (creation date or update date).
4) *short description*: the short description is a small text (less than 100 characters) where the image provider gives a summary description of the image. The content of this short text is carefully written by authors as it is the second level contact of potential users, after the image name.
5) *full description*: the full description is a larger text where the image provider describes in details the image, provides links to download and install the image and gives details on installation parameters like port number as well as related docker images.
6) *links*: in the full description, one can find links to Docker files[3]: they give the exact location to the docker image files.
7) *comments*: a set of elements in the page provide users comments. They can be divided into several pages, accessible through the page navigation area. Comments are the collaborative part of the page, where visitors can discuss the image, give their opinions, propose improvements etc.
8) *comment author's nickname*: for each comment, the nickname of the user who posted the comment is provided.
9) *date of comment publication*: for each comment, the date of comment publication is given.
10) *comment contents*: user generated content, that contain either opinion or user experience with the image, that can be very valuable for other users.
11) *docker pull command*: the command line that allows to download an image.

These elements are the source of information to build automatically RDF descriptions of images. The next section gives the target RDF model for images semantic descriptions. Section IV explains the extraction process we defined to populate the ontology.

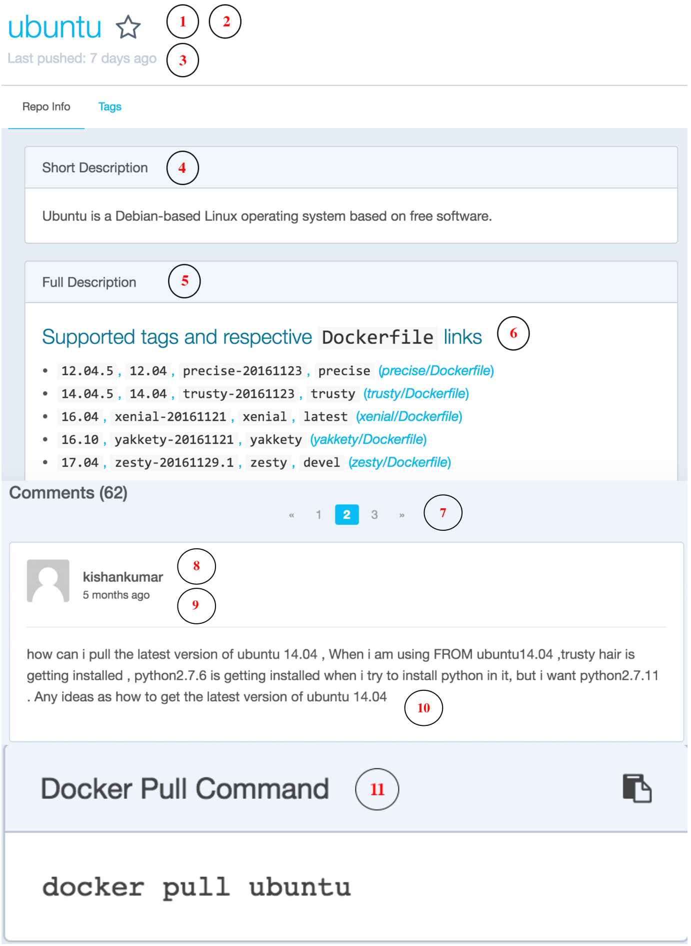
III. The *VirtualComponent* Ontology

In the previous Section, we presented the structure of Docker Hub Web pages, each page describing a Docker image. From these pages, we identified three different categories of information that compose a Docker image description: (1) information on the docker image itself: its name and how to execute it, (2) context data such as links to Wikipedia pages, and (3) the social inputs materialized by stars and users' comments. The first category of data provides information on the Docker image itself. To the best of our knowledge, no ontology today can describe this concept. Based on the structure of the pages as shown in Figure 1, we have defined the *VirtualComponent* ontology that represents a docker image.

The second category provides context data under the form of external links and additional information about the docker image. We use these metadata to enrich the description of the docker image. In our system, these links are enriched with links to the LOD through DBPedia URIs. Doing so, we augment the LOD cloud with docker images nodes. It allows the development of applications that both include Docker descriptions and take advantage of the LOD cloud. The third category concerns the social inputs. It links the docker image to the social Web. Structuring the ratings and the user comments will allow applications to take profit of the information it contains. In our system, we rank the images [7], [13] and perform a large scale sentiment analysis on the dataset using the works described in [11]. The dataset we build enables deeper social and semantic analysis.

Prefix	Namespace
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns
prov:	http://www.w3.org/ns/prov
Sioc:	http://rdfs.org/sioc/ns
docterms:	http://purl.org/dc/terms/
VirtualComponent:	http://telecom-st-etienne.fr/VirtualComponent

TABLE I
Prefixes used



ubuntu

Last pushed: 7 days ago

Short Description

Ubuntu is a Debian-based Linux operating system based on free software.

Full Description

Supported tags and respective `Dockerfile` links

- `12.04.5`, `12.04`, `precise-20161123`, `precise` (precise/Dockerfile)
- `14.04.5`, `14.04`, `trusty-20161123`, `trusty` (trusty/Dockerfile)
- `16.04`, `xenial-20161121`, `xenial`, `latest` (xenial/Dockerfile)
- `16.10`, `yakkety-20161121`, `yakkety` (yakkety/Dockerfile)
- `17.04`, `zesty-20161129.1`, `zesty`, `devel` (zesty/Dockerfile)

Comments (62)

kishankumar
5 months ago

how can i pull the latest version of ubuntu 14.04 , When i am using FROM ubuntu14.04 ,trusty hair is getting installed , python2.7.6 is getting installed when i try to install python in it, but i want python2.7.11 . Any ideas as how to get the latest version of ubuntu 14.04

Docker Pull Command

```
docker pull ubuntu
```

Fig. 1. Description of the Ubuntu image in its Docker Hub web page

We created the *VirtualComponent* Ontology that structures the descriptions of Docker images and that allows to link them to the Linked Open Data Cloud. For this purpose, we have

[3] https://docs.docker.com/engine/reference/builder/

defined several classes and properties that are described below. Prefixes used are defined in table I.

- *VirtualComponent:imageName*: The name of the image.
- *VirtualComponent:lastPushed*: The date of publication of the image or of the last update
- *VirtualComponent:imageShortDescription*: A summary description.
- *VirtualComponent:dockerPullCommand*: The command line to download the image.
- *VirtualComponent:dockerRunCommand*: The command line to use the image to build Docker on the host.
- *dbo:wikiPageRedirects*: A link to a dbpedia concept to better understand the role of the image.
- *prov:wasDerivedFrom*: A link to a wikipedia page describing the software component contained in the image.
- *VirtualComponent:dockerVersionSupported*: Docker version the image can support.
- *VirtualComponent:ImageGitLink*: The dockerfile download link from github.
- *VirtualComponent:ImageVersion*: version of the image.
- *VirtualComponent:LinkedDockerImages*: list of related Docker images.
- *sioc:Post*: object describing user comments with the following properties.
- *dcterms:created*: comment date.
- *sioc:has_container*: an object that this post belongs to.
- *sioc:has_creator*: the sioc:UserAccount object describing the comment author.
- *sioc:UserAccount*: object that defines the comment author account.
- *sioc:content*: Comment textual content.

The ontology reuses parts of well-known ontologies, namely PROV-O and SIOC. The element ”Prov:wasDerivedFrom” from the PROV Ontology [9] refers to the DBPedia pages that describe images and their usage. The SIOC ontology [5] is used to describe user comments and ratings. The reader can also notice that the generated dataset is connected to the Linked Open Data Cloud through links to DBPedia concepts.

Figure 2 provides an example for the description of the Wordpress image, with contents as extracted from the Docker Hub. For example, the *prov:wasDerivedFrom property* has value ”wikipedia.org/wiki/WordPress”. The pull command is given as ”docker pull wordpress”, and the image is linked to *MySQL*.

IV. Docker2RDF: Converting Docker image Web pages to RDF

In order to populate the knowledge base, we have developed Docker2RDF, a tool that automatically converts HTML pages from the Docker Hub into instances of our VirtualComponent Ontology.

Docker2RDF uses the HTML structure of the Docker Hub pages to identify the fields corresponding to our ontology. For this purpose we make an extensive use of CSS selectors to extract the contents of the relevant fields.

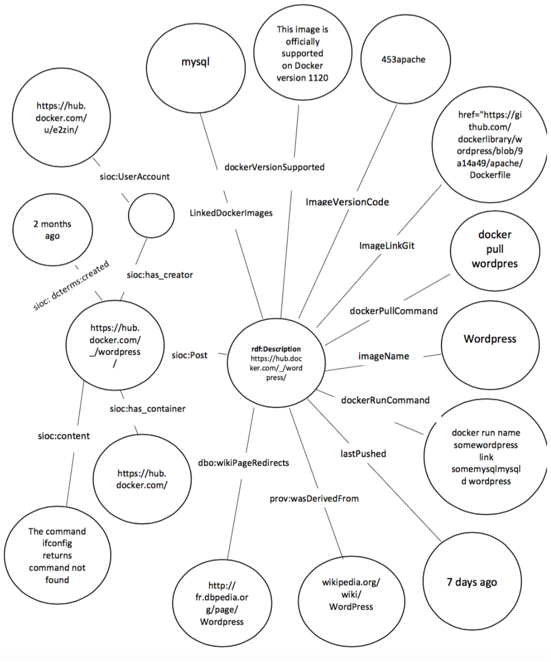

mysql
This image is officially supported on Docker version 1120
453apache
https://hub. docker.com/ u/e2zin/
href="https://github.com/dockerlibrary/wordpress/blob/9a14a49/apache/Dockerfile
sioc:UserAccount
dockerVersionSupported
LinkedDockerImages
ImageVersionCode
2 months ago
ImageLinkGit
docker pull wordpres
sioc:has_creator
sioc: dcterms:created
dockerPullCommand
https://hub. docker.com/ _/wordpress /
sioc:Post
rdf:Description
https://hub.docker.com/_/wordpress/
imageName
Wordpress
dockerRunCommand
sioc:has_container
sioc:content
docker run name somewordpress link somemysqlmysql d wordpress
lastPushed
https://hub. docker.com/
dbo:wikiPageRedirects
prov:wasDerivedFrom
The command ifconfig returns command not found
http:// fr.dbpedia.org/page/ Wordpress
wikipedia.org/ wiki/ WordPress
7 days ago

Fig. 2. Example: Wordpress image description

In this extraction process, we also analyse the contents of these fields to interlink the entities. For instance, the following Docker command :

```
docker run wordpress --link
 --name some-some-mysql: mysql -p
 8080: 80 -d wordpress
```

describes how to start the Wordpress Docker image. This command contains an explicit link to an instance of MySQL. We therefore create a link between the Wordpress image and the MySQL image. Since it is very common to have multiple and complex dependencies, the ontology structure is very helpful for users.

Docker2RDF crawls the Docker Hub, extracts information identified on Figure 1 and uses extracted data to populate the ontology and link entities. It outputs the resulting dataset into an RDF file that constitutes our dataset. In the following subsection, we give some figures characterizing the dataset.

We have run Docker2RDF on the Docker Hub in December 2016. The generated RDF file can be downloaded from the DockerWebStat application [4]. The size of the file is 610 KB. At that date, 120 images were available on the Docker Hub. Each image is described following the VirtualComponent ontology format. The total number of comments is 1833, which corresponds to 15 comments per image in average.

V. DockerWebStats : A Web application to exploit Docker2RDF

In order to make our dataset accessible, we developed DockerWebStats[5], a web application that presents our dataset and its characteristics.

[4] https://161.3.199.17/DockerWebStats

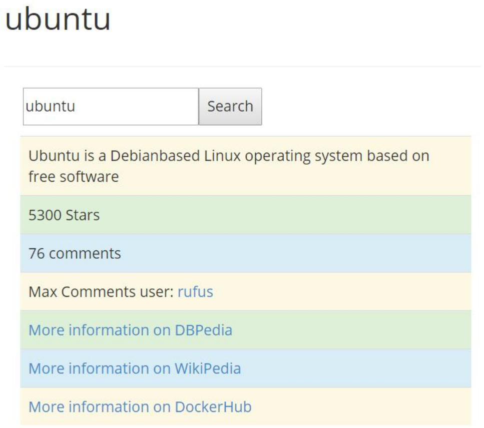
[5] http://161.3.199.17/DockerWebStats/

This Web application presents some statistics on our dataset. It also gives Docker users a fast and accurate description of the Docker images that are made available on the Docker Hub. Among all, it lists the best images rated by users, giving an idea of the point of view of people who have used these images. For semantic Web practitioners, it offers a SPARQL endpoint and also comes along with illustrative predefined queries. We here describe the different features offered today by DockerWebStats.

a) Image Stats : Statistics on a chosen image: The *image stats* service of DockerWebStats provides statistics on each image independently.

By selecting an image, 3 blocks appear on the page.

In the first block, as illustrated on Figure 3, a table shows the short description of the image, the number of stars it received from users, the overall number of comments and the user who commented most. It also provides links to the DBPedia and Wikipedia pages that describe the software included in the image. With the number of stars and comments, this table gives an idea on the popularity and on the perceived quality of the image.



ubuntu

ubuntu

Ubuntu is a Debianbased Linux operating system based on free software

5300 Stars

76 comments

Max Comments user: rufus

More information on DBPedia

More information on WikiPedia

More information on DockerHub

Fig. 3. DockerWebStats: information on an image

The second block is a graph that shows the number of comments the image received each month.

The third block of the page is also a graph. It describes the users opinion on this image, using a sentiment analysis algorithm f the literature called ”sentiment.vivekn”

A majority dislike on an image may indicates that the image is of poor quality, but it may also indicate that its description misses some important information. Reading the comments can help arbitrating.

b) General stats: The *General stats* service provides general statistics on the dataset. It first provides the top-10 starred images, i.e. the images that have been tagged as star by the largest number of users.

c) Query on RDF file: The *Query* service proposes predefined queries. It also opens access to a SPARQL endpoint with which proficient users can write any SPARQL query on the dataset. We follow best practices to publish our dataset [1], this SPARQL endpoint makes it a first class LOD cloud citizen.

VI. Conclusion

In this paper, we have proposed Docker2RDF, a system that extracts information from the Docker Hub description pages and builds a semantic description of Docker images. We have defined an ontology for Docker images descriptions that enhanced the Hub pages with links to the LOD cloud. We have designed and implemented Docker2RDF, it builds a dataset under the form of an RDF document. We also propose a Web Application called DockerWebStats that allows to interact with these semantic descriptions. Docker2RDF extends the scope of Docker Hub to new types of resources and interlinks using the standard RDF semantic data format. It allows the use of full SPARQL queries on the dataset.

In short term perspectives, other analyses on the extracted information should be made, so as to take full profit of the information provided by images authors. We should allow to propose the automatisation of some steps in the process of Docker containers deployment. Second, we should extend our work so as it can encompass or be compatible with other types of components or virtual components.

Lastly, the links with wikipedia and DBpedia show promises in many directions. At the level of the software design process, many questions remain: how and when should we include the search and query of Docker containers? Will wikipedia and DBpedia demonstrate their power in the component selection process? On another axis, we require also a federated querying system that can execute queries on multiple endpoints in a user-transparent way.

References

[1] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing linked datasets. In *LDOW*, 2009.

[2] Sören Auer, Jan Demter, Michael Martin, and Jens Lehmann. Lodstats–an extensible framework for high-performance dataset analytics. In *EKAW*, 2012.

[3] Christian Bizer, Kai Eckert, Robert Meusel, Hannes Mühleisen, Michael Schuhmacher, and Johanna Völker. Deployment of rdfa, microdata, and microformats on the web–a quantitative analysis. In *ISWC*, pages 17–32. Springer, 2013.

[4] Carl Boettiger. An introduction to docker for reproducible research. *ACM SIGOPS*, 49(1):71–79, 2015.

[5] John G Breslin, Stefan Decker, Andreas Harth, and Uldis Bojars. Sioc: an approach to connect web-based communities. *IJWBC*.

[6] Xia Cai, Michael R. Lyu, Kam-Fai Wong, and Roy Ko. Component-based software engineering: technologies, development frameworks, and quality assurance schemes. In *Asia-Pacific Software Engineering Conference*, page 372. IEEE Computer Society, 2000.

[7] Xin Dong, Alon Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In *SIGMOD, pages=85–96, year=2005*.

[8] Ivan Ermilov, Michael Martin, Jens Lehmann, and Sören Auer. Linked open data statistics: Collection and exploitation. In *KESW*, 2013.

[9] Olaf Hartig and Jun Zhao. Publishing and consuming provenance metadata on the web of linked data. In *International Provenance and Annotation Workshop*, pages 78–90, 2010.

[10] Dirk Merkel. Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239), March 2014.

[11] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[12] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the linked data best practices in different topical domains. In *ISWC*, pages 245–260. Springer, 2014.

[13] Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudré-Mauroux, and Karl Aberer. Trank: Ranking entity types using the web of data. In *ISWC*, pages 640–656, 2013.

[5] http://sentiment.vivekn.com