Exploring Complexity in Health: An Interdisciplinary Systems Approach
A. Hoerbst et al. (Eds.)

doi:10.3233/978-1-61499-678-1-643

Simplified Deployment of Health Informatics Applications by Providing Docker Images

Matthias LÖBE[*,a,1], Thomas GANSLANDT[*,b], Lydia LOTZMANN[a], Sebastian MATE[b], Jan CHRISTOPH[b], Benjamin BAUM[c], Murat SARIYAR[d], Jie WU[d] and Sebastian STÄUBERT[a]

[a] *Institute for Medical Informatics, Statistics and Epidemiology (IMISE), Leipzig University, Germany*
[b] *Chair of Medical Informatics, Friedrich-Alexander-University of Erlangen-Nuremberg, Germany*
[c] *Department of Medical Informatics, University Medical Center Göttingen, Germany*
[d] *TMF – Technology, Methods, and Infrastructure for Networked Medical Research, Berlin, Germany*

Abstract. Due to the specific needs of biomedical researchers, in-house development of software is widespread. A common problem is to maintain and enhance software after the funded project has ended. Even if many tools are made open source, only a couple of projects manage to attract a user basis large enough to ensure sustainability. Reasons for this include complex installation and configuration of biomedical software as well as an ambiguous terminology of the features provided; all of which make evaluation of software laborious. Docker is a para-virtualization technology based on Linux containers that eases deployment of applications and facilitates evaluation. We investigated a suite of software developments funded by a large umbrella organization for networked medical research within the last 10 years and created Docker containers for a number of applications to support utilization and dissemination.

Keywords. DevOps, Docker, deployment, IT infrastructure, microservices, PaaS, cloud, container

1. Introduction

Biomedical informatics aims at supporting medicine-related research and health care with innovative tools and services. Developing new software, extending existing applications with new features and assembling software modules to a sophisticated solution-driven architecture are common tasks in clinical and translational settings. Fast development cycles, limited resources, heterogeneous data sources, short project durations as well as small user groups that distract commercial providers are typical characteristics in the field. Public funding for software development frequently leads to open source software that is not maintained once the project has ended. In practice, this often means “abandonware” with no support available for interested third parties. Even

[1] Corresponding Author: matthias.loebe@imise.uni-leipzig.de | * Equal contribution

if some applications were initially developed with a bigger picture in mind, further public funding just for hardening and evolving existing tools is rare. One main problem is that evaluating the potentials of existing but unfamiliar software can be difficult, if it is not disseminated widely. Installing and especially customizing new software is usually a complex and time-consuming task. As a result, many researchers start to develop new solutions from scratch, having full control over the development at first, but leading to a redundant clone with a limited life expectation later on.

In order to harden and disseminate tools in the biomedical community, the TMF – Technology, Methods, and Infrastructure for Networked Medical Research – hosts and extends many tools. TMF is the umbrella organization for networked medical research in Germany. Its members are academic biomedical research networks, in the majority of cases funded by the Federal Ministry of Education and Research, and medical research facilities such as Clinical Trials Centers or University hospitals. The work of the TMF focuses on organizational, legal/ethical and technological solutions for modern medical research, including the customization of existing and the development of new IT applications. The TMF has a sub-committee on reviewing the IT landscape in Germany as worldwide, publishing its recommendations annually [1].

An important requirement for TMF projects is that results should fit for more than one concrete use case; for that reason, e.g., grant applications will have to verify support from about five TMF members scientific community. Examples for generic IT solutions developed in last years are data protection compliant ID management and pseudonymization services [2], tools for anonymizing patient-related data [3], software for data management of longitudinal cohorts [4], for managing clinical trial participants [5], for creating clinical data marts [6], for supporting data integration [7] and semantic interoperability [8].

To overcome the problem of complex installation and configuration of third-party software and to simplify evaluation and comparison by providing easy deployable demo instances, we propose a new approach based on *Docker*, a Linux container technology. Docker is similar to virtualization, but in contrast to Virtual Machines (VM), it does not require a full-scale operation system for every target instance, but enables sharing of one Linux kernel (see figure 1). Docker uses proven Linux kernel features like cgroups for allocating resources (CPU usage, amount of RAM, I/O operations), namespacing for security (process isolation, separating and hiding processes, mount points or devices) and the Union file systems (overlay file systems that operate by creating layers to minimize redundancy).

The basic idea of Docker is to provide a container template for every service endpoint, e.g. a database, a web application or a directory service. Splitting monolithic architectures into smaller, easier to maintain modules leads to *microservices,* which can be regarded as a refinement of service-oriented architectures (SOA), which is in turn based on the pattern of orchestrated services.

A *Docker image* is a template for creating Docker containers. A *Docker contain*er is an instance of an image and combines certain services and applications in an executable runtime environment. Multiple containers can be run from one image. A typical image would be the Apache web server. Images are usually based on other images, e.g. Apache will inherit from some Linux image like Debian. They are stored in a *registry*. Before an image can be used locally, it must be pulled from a registry. The default registry is called *Docker Hub* and contains images for most well-known operation systems, services and applications. Docker images can exist in multiple versions using *tags* as distinct annotations. The most current image is tagged *latest*,

other tag names usually refer to the software version contained, e.g. *apache:2.4*. New Docker images can be created by starting a container, making the desired modifications and committing the changes to a new image. However, to make such changes in a transparent and traceable way, it is best practice to use a text script called *Dockerfile*. A Dockerfile has to conform to the syntax of a domain specific language and allows reproducing the steps required to build a Docker image on any other host (plus auxiliary files, if necessary).

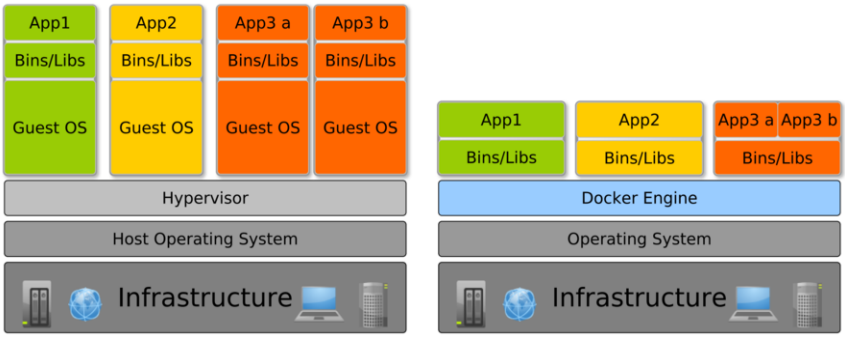

Figure 1. Docker containers (right) compared to traditional VMs (left).

2. Methods

Our project aimed at two principal goals: first, to test and review the Docker technology with regards to suitability for clinical/translational research and healthcare applications and second, to prepare strategic recommendations for general rules of deployment for TMF-supported software development projects. To accomplish the first objective, we exemplarily chose three tools that have seen widespread use in recent years:

- the TMF PID generator [9], a tool for generating and managing a master patient index, e.g., for clinical trial participants;
- OpenClinica [10], a data management system that can access the PID generator via Web Services, and
- i2b2 [11], a data warehouse platform for cohort discovery and hypothesis testing on clinical data originating from sources such as OpenClinica [7].

To accomplish the second objective, we reviewed the complete list of TMF products and services (http://www.tmf-ev.de/EnglishSite/ProductsServices.aspx) and argued for or against containerization, depending on the type of product, complexity of installation, software life cycle, platform requirements and synergy effects.

3. Results

We created Docker files and Docker images for all of the three tools. The TMF PID generator is a rather old development and requires a 32-bit architecture for which no official Ubuntu Docker image exists, but an alternative image could be found. This proved to be the first real-life advantage since the Docker host runs on a 64-bit Ubuntu. The TMF PID generator further depends on a PostgreSQL database server, which was

realized externally as a second image to avoid redundancy, since OpenClinica and i2b2 use PostgreSQL as well. OpenClinica relies on Java and Tomcat. To prevent unintentional data loss, all documents (configuration files, exports, case report forms, rules declarations) are saved on a special data volume. OpenClinica was linked to the PostgreSQL database mentioned before, which had to be downgraded to version 8.4 to comply with the official OpenClinica system requirements. i2b2 is based on JBoss and supports PostgreSQL as well as Oracle™, so a dedicated Oracle XE container was created. For all applications, basal functional checks were performed to ensure operability. Finally, a Docker Hub account was created and all images pushed.

For the TMF product list, we reviewed 73 existing entries and 7 grant applications in progress. We further added 14 external products that were in use in TMF member networks and that are representative for certain fields of application, among them tranSMART for biological data exploration, openBIS for managing experimental data, Mirth Connect as a common HL7 message router and OpenMRS for electronic health records. From those 94 entries, 48 were removed because their focus was not software, but guidelines or expert's reports. The remaining 46 software applications were further analyzed with regards to their usage scenario. 6 tools were excluded for being intended as singular installations (e.g. the central German registry of biobanks), 10 tools were supposed to be run locally without a web interface (which vastly reduces the benefits of Docker as a server technique). Of the 30 tools left, 5 already had a Docker image available in the registry. We argued that the 25 remaining tools are suitable for dockerization, so Dockerfiles should be developed and a TMF "dockerbank" established. At the time of writing, 4 more containers are going to be implemented. We further demanded that Docker containers should be made mandatory for future software development projects.

4. Conclusions

The Docker technology has become broadly accepted since its first production release in June 2014, especially in Cloud environments. With the introduction of Windows Server 2016™, Microsoft will also support Docker container on Windows™ and its Azure™ cloud platform. In the medium term, Docker may complement traditional virtualization in the IT departments of most University computer centers.

Docker containers have, compared to VMs, a very low footprint (see figure 1). This makes it easy to provide scalability (new instances from generic templates) and high availability (fast startup). They are easily distributable (due to their small size) and portable (pre-built ready to use software packages). Most major software vendors already provide Docker images or support Docker with dedicated tools.

For software developers in biomedical research and health care, the Docker architecture combines modern and interesting features: it's application-centric and service-oriented, easy to understand (in principle, a Docker container is just a directory in the file system), easy to create (Dockerfile as a receipt for build automation), supports version control (tagging, diffs, history, commits) and central, maintained repositories. But Docker is not only intended for development and testing, Docker can also foster the dissemination because it accelerates installation and evaluation of software, especially when documentation is only available in local languages and not in English. Additionally, containers can be configured to contain example files and demo data that demonstrate the concepts and features of an application. Currently, the TMF is

planning to establish an IT service and information portal to boost its national and international visibility. It would be advisable to rely on the experiences made in this project: to better invest resources in proven tools rather than beginning from scratch all over again.

Docker can furthermore be used in a number of medical informatics scenarios, e.g. in fast-changing environments for saving and archiving a certain state, for instance when computing data for scientific publications. It can support interoperability efforts like IHE Connectathons with its virtualization features. Docker could in the long run even have an impact on product qualification in regulated environments (Good Automated Manufacturing Practice) since it facilitates automated software tests.

Acknowledgements

The research leading to these results has received funding from the TMF – Technology, Methods, and Infrastructure for Networked Medical Research, Germany through the Federal Ministry of Education and Research, MethInfraNet (grant number 01GI1003) and i:DSem - Integrative data semantics in systems medicine (grant number 031L0026).

References

[1] J. Drepper, S.C. Semler (Eds.), IT-Infrastrukturen in der patientenorientierten Forschung: Aktueller Stand und Handlungsbedarf 2015. Verfasst und vorgelegt vom IT-Reviewing-Board der TMF, AKA, Berlin, 2016.

[2] M. Lablans, A. Borg, F. Ückert, A RESTful interface to pseudonymization services in modern web applications, *BMC Medical Informatics and Decision Making* **15** (2015) 2.

[3] R. Lautenschläger, F. Kohlmayer, F. Prasser, K.A. Kuhn, A generic solution for web-based management of pseudonymized data, *BMC Med Inform Decis Mak* **15** (1) (2015) 795.

[4] M. Bialke, T. Bahls, C. Havemann, J. Piegsa, K. Weitmann, T. Wegner, W. Hoffmann, MOSAIC – A Modular Approach to Data Management in Epidemiological Studies, *Methods Inf Med* **54** (4) (2015) 364–371.

[5] J. Schwanke, O. Rienhoff, T.G. Schulze, S.Y. Nussbeck, Suitability of Customer Relationship Management Systems for the Management of Study Participants in Biomedical Research, *Methods Inf Med* **52** (4) (2013) 340–350.

[6] T. Ganslandt, S. Mate, K. Helbing, U. Sax, H.U. Prokosch, Unlocking Data for Clinical Research – The German i2b2 Experience, *ACI* **2** (1) (2011) 116–127.

[7] Bauer, C R K D, T. Ganslandt, B. Baum, J. Christoph, I. Engel, M. Löbe, S. Mate, S. Stäubert, J. Drepper, H.-U. Prokosch, A. Winter, U. Sax, Integrated Data Repository Toolkit (IDRT). A Suite of Programs to Facilitate Health Analytics on Heterogeneous Medical Data, *Methods of information in medicine* **54** (6) (2015).

[8] J. Stausberg, M. Löbe, P. Verplancke, J. Drepper, H. Herre, M. Löffler, Foundations of a Metadata Repository for Databases of Registers and Trials, in: *Medical Informatics in a United and Healthy Europe - Proceedings of MIE 2009, The XXIInd International Congress of the European Federation for Medical Informatics*, Sarajevo, Bosnia and Herzegovina, Agust 30 - September 2, 2009, IOS Press, 2009, pp. 409–413.

[9] A. Faldum, K. Pommerening, An optimal code for patient identifiers, *Computer Methods and Programs in Biomedicine* **79** (1) (2005) 81–88.

[10] H. Leroux, S. McBride, S. Gibson, On selecting a clinical trial management system for large scale, multi-centre, multi-modal clinical research study, *Studies in health technology and informatics* **168** (2011) 89–95.

[11] S.N. Murphy, M. Mendis, K. Hackett, R. Kuttan, W. Pan, L.C. Phillips, V. Gainer, D. Berkowicz, J.P. Glaser, I. Kohane, H.C. Chueh, Architecture of the open source clinical research chart from Informatics for Integrating Biology and the Bedside, *AMIA Annual Symposium proceedings / AMIA Symposium.* AMIA Symposium (2007) 548–552.