Moving the sciebo Sync and Share Cloud Service to State-of-the-Art Cloud Technology

Holger Angenent, Marcel Wunderlich, Raimund Vogl
Westfälische Wilhelms-Universität Münster, WWU IT, Germany
holger.angenent@uni-muenster.de, wunderlich@uni-muenster.de, rvogl@uni-muenster.de

Abstract

The campuscloud sciebo is a widely used service for higher education in North Rhine-Westphalia (NRW) in compliance with German data protection legislation. The project has started in February 2015. Due to the high demand, especially because of the Corona crisis, there are currently about 200.000 registered users alone. Modernizing the infrastructure for future demands is one of the current major challenges of sciebo. With the concept of the second project phase, a modernisation of the setup became necessary. Not only for scalability, but also for the closer integration with other services the shift to a Kubernetes-based platform is an obvious step. The modernized setup consists of two synchronously mirrored server sites each with 5 PB net capacity Spectrum Scale file system and two Kubernetes clusters running the ownCloud software.

1 Project History

The campuscloud sciebo – reported at EUNIS before (Rudolph et al., 2016; Vogl et al., 2015) – started in early 2015 after a conceptual design phase of two years, and to date, the number of registered users has grown to about 200,000 (see predictions in Vogl, Angenent, Rudolph, Stieglitz, & Meske, 2016). The main drivers of the projects were and still are the need for a secure file syncing and sharing solution and large on premise data storage capacities for researchers. Most important for global acceptance is an easy to use interface with focus on intuitive usability.

An evaluation of available software solutions in 2013/2014 led to the decision that ownCloud was covering all features required for the project. At the time, OwnCloud was uniquely positioned as an open source solution. In the meantime, the fork Nextcloud and other Open Source solutions appeared on the market, too. However, the long term vision of ownCloud of delivering a stable and scalable product with their next generation ownCloud Infinite Scale is still best suited for the future of sciebo.

For the first project phase of five years (2015-2020), as a legal framework for the operation of sciebo, a consortium was formed. The consortium as operator of the sciebo service was replaced by the

University of Münster in the course of updating the contract framework for GDPR, now being the only supplier of the service, with all other participating institutions acting as customers. The end users now conclude the terms of service not with their own institution, but directly with the University of Münster.

Some coordination effort was necessary to enable all participating universities to join the Shibboleth-based Authentication and Authorization Infrastructure, operated by German Research Network (DFN-AAI), which is required for the envisioned self-enrolment portal. During the first project phase, this portal has been extended with additional features as the invitation for external guest users and the creation of so-called project boxes. Project boxes offer larger accounts for groups and individuals with higher storage demand.

2 sciebo ng – The Next Generation of sciebo

At the heart of sciebo lies a classical LAMP-stack which runs ownCloud. Currently there are three hosting sites, one at the University of Bonn, one at the University Duisburg-Essen and one at the University of Muenster. Each site consists of an IBM Spectrum Scale cluster for storage, a MariaDB Galera cluster, a fleet of webservers, assorted services for session persistence, user authentication as well as some central services like OnlyOffice, which run only in Münster and are accessible from all sites.

Abiding the KISS principle, deployment and management happens mainly via Ansible and ssh, avoiding complex interactions as much as possible. This has yielded a very reliable and stable deployment. It also reflects the best practices from the time of sciebo's inception and is a testament to their validity.

However, the industry has moved forward and new best practices have found their way into production: Version controlled configurations and deployments, review processes, automated testing and vulnerability scans – things that once have been wishful thinking are starting to become requirements, not only for developers, but also for operators. An ever increasing demand for feature velocity from our own users as well as the quest for more interoperability in the scientific community furthermore pushes the limits in adaptability and maintainability of our old deployment. With a central service such as file storage, usability and user satisfaction depend not only on availability, but also on low latency. To ensure this, more modern monitoring solutions are required, which not only tell you if the service is healthy, but allow for more quantified measurements.

These factors require a more effective deployment strategy, standardized interfaces, and APIs exposing operational functionality not only to operators, but also developers of associated services and automation tools. There are several technology stacks enabling this move, among which Kubernetes stands out as a comparatively lightweight solution: Building containers with dockerfiles comes natural and is far more transparent than the provisioning of virtual machines with other solutions. As containers are just Linux processes, it is still possible to gain introspection in the behaviour of involved components with familiar tools.

For sciebo ng (the technological next generation rehaul) we devised a Kubernetes platform that serves as a thin layer between the hardware and the application itself. Building upon the existing expertise in the team and playing well along with the IBM Spectrum Scale storage, the bare metal nodes are provisioned via xcat. Ansible is used to deploy the necessary tools for Kubernetes, like kubeadm, a container runtime and some minor adjustments which might change during lifetime of the cluster, like keepalived configurations for internet facing loadbalancers and the control-plane. Kubernetes itself is installed with kubeadm. Inside the Kubernetes cluster then Prometheus operator, rook-ceph and NGinx ingresses loadbalanced by HAProxy are deployed for monitoring, persistent storage of data that is not user data and ingress networking respectively. This allows to bootstrap a whole site quickly and might even serve as starting point for a generalized administrative platform of webservices or as addition on existing hardware.

A well-known buzzword is “cloud-native”, a term summarizing several principles which make an application suitable to be run on Kubernetes. Those are principles like statelessness, microservice architecture, rollback capabilities, principles that were not of concern or even diametral to deployment goals (e.g. avoiding microservice sprawl) at the time of sciebo’s first inception. Also ownCloud (and consequently its fork Nextcloud) were not conceptualized as cloud-native services. Because of this, we had to put considerable effort into making it work in this new environment. In some instances this was even beneficial for the existing sciebo deployment. For example only the ephemeral nature of Kubernetes pods made errors in session persistence apparent, whereas the stability of the existing deployment hid those bugs from our view.

We have now reduced the toil of setting up an ownCloud running on multiple servers with its own database cluster to a handful of kubectl and helm commands. We run two physically separate sites in Münster, with the IBM Spectrum Scale filesystem mirrored synchronously. Moving an ownCloud installation from one site to the other is as fast as waiting for DNS caches to flush. We are now in the process of migrating the productive instances from the old hardware to the new hardware, and adjusting parameters and monitoring according to the corner cases that only appear in production with thousands of active users.

3 References / Citations

Vogl, R., Angenent, H., Rudolph, D., Thoring, A., Schild, C., Stieglitz, S., & Meske, C. (2015). sciebo — theCampuscloud for NRW. In: *21st EUNIS Congress, Dundee, Schottland*, p. 15-26.

Rudolph, D., Vogl, R., Angenent, A., Thoring, A., Wilmer, A., & Schild, S. (2016). Challenging Dropbox: The Adoption and Usage of New Cloud Storage Service "sciebo" at German Universities. In: *EUNIS 2016: Crossroads where the past meets the future. EUNIS 22nd Annual Congress, Book of Proceedings*. Thessaloniki, p. 19-29.

Vogl, R., Angenent H., Rudolph, D., Stieglitz S., & Meske, C. (2016). Predictions on Service Adoption and Utilization Meet Reality. First Results from the sciebo (science box) Project. In: Zaphiris, P., Iannou, A. (Eds.): *Learning and Collaboration Technologies. Third International Conference, LCT 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings.* Springer International Publishing, p. 639-649. doi: 10.1007/978-3-319-39483-1_58.

4 Author Biographies

H. Angenent has studied physics at the Technical University of Ilmenau and the University of Münster (Germany). He worked four years as a research assistant at the institute of Theoretical Physics in Münster. Since 2010 he is a research assistant at the WWU IT (the university computing centre) of the University of Münster and leads the eScience infrastructure group, responsible for high performance computing systems and cloud services.

R. Vogl holds a Ph.D. in elementary particle physics from the University of Innsbruck (Austria). After completing his Ph.D. studies in 1995, he joined Innsbruck University Hospital as IT manager for medical image data solutions and moved on to be deputy head of IT. He served as a lecturer in medical informatics at UMIT (Hall, Austria) and as managing director for a medical image data management software company (icoserve, Innsbruck) and for a center of excellence in medical informatics (HITT, Innsbruck). Since 2007 he has been director of the WWU IT (the university computing centre) of the University of Münster (Germany). His research interests focus on management of complex information systems and information infrastructures.

M. Wunderlich has studied mathematics at the University of Münster (Germany) with focus on differential geometry and always worked in administration and development on the side. He earned his PhD in 2019, when he also joined the eScience infrastructure group of the WWU IT. There he supports the sciebo team in the day to day business and developed the platform for the next generation of sciebo. His research focus lies on the balancing act of providing scalable yet maintainable infrastructure, which allows researchers and educators to excel.