ScienceBox 2.0: Evolving the demonstrator package for CERN storage and analysis services

Enrico Bocchi[1]*, *Jimil* Desai[1], and *Samuel* Alfageme Sainz[2]

[1]CERN, Esplanade des Particules 1, 1211 Geneva 23, Switzerland
[2]SWITCH, Werdstrasse 2, 8004 Zürich, Switzerland

Abstract. With containers being the de-facto standard to package, distribute, and run applications, Helm charts are on the rise for application deployment in managed clusters (e.g., Kubernetes, OpenShift), providing developers and operators with a rich ecosystem of tools to utilize, as well as the means to configure applications and roll changes out in a programmatic way.
This paper describes the reboot of the ScienceBox project: The containerized software bundle providing the ability to deploy CERN storage and analysis services on any cloud or local infrastructure. While the service offering provided through ScienceBox remains unchanged, we evolved the original implementation to make use of Helm charts across the entire stack and incorporated a major architectural update to CERNBox, replacing the previous PHP backend with a distributed microservices architecture. ScienceBox maintains the flexibility to be deployed in a few clicks for demonstration purposes, or scaled out on managed clusters for the provisioning of a multi-user service. It also removes external dependencies by deploying additional services (e.g., IDP, DBs) and facilitates the configuration of the service stack by providing automated configuration scripts, making the package fully self-contained and easy to use.

1 Introduction

In the era of containerization [1], where encapsulation and seamless distribution of software have become a cornerstone of modern computing, Helm charts [2] have risen to prominence as a go-to solution for deploying applications within managed container clusters. Charts offer a powerful ecosystem of tools, ultimately providing administrators with the flexibility to configure applications and orchestrate deployments with ease and efficiency.

We here provide an update to [3] and describe how the ScienceBox project was restarted to use modern widely-adopted container technologies, to improve maintainability and modularity, and to ease contributions to the package. While the goal of ScienceBox has not changed, its implementation has been re-architected from scratch: ScienceBox continues to be the demonstrator package for CERN-developed storage and analysis services and allows for the deployment of a fully-fledged replica of CERN production services on external premises, i.e., private, hybrid, or commercial clouds, or even a single VM for demonstrative purposes. Its implementation has been reworked to make extensive use of Helm charts so as to reutilize existing charts from services that are part of the package and to allow for the

*e-mail: enrico.bocchi@cern.ch

deployment of single components or the addition of new ones. ScienceBox is designed to be flexible with respect to the available infrastructure and can be deployed on a single node, a cluster of hosts, or on managed Kubernetes clusters provided by commercial clouds. Previous exercises have demonstrated that ScienceBox can be deployed on mixed bare-metal and virtualized nodes, OpenStack-based clouds, as well as on commercial platforms including Amazon Web Services (AWS) and the Helix Nebula Science Cloud (HNSciCloud).

This paper is structured as follows: i) Sec.2 describes ScienceBox in its early days, the use cases where it proved to be useful, and the infrastructure type and size where it was deployed; ii) Sec.3 details the new implementation based on Helm charts and explains how better maintainability was achieved by partitioning ScienceBox into service silos and expressing dependencies against them for a fully-fledge deployment (Sec. 3.1), which components were added to achieve full portability (Sec. 3.2), and how ScienceBox can be easily deployed on a single node in a few clicks, or rather scaled-out on a cluster of nodes (Sec. 3.3); and iii) Sec.4 reports on future plans for ScienceBox and its new fields of application.

2 ScienceBox Heritage

The ScienceBox project started in 2017 and has provided a sandbox to non-CERN users for evaluating and using the storage, sharing, and analysis services developed and operated at CERN since then. In addition to demonstration purposes at conferences and workshops, ScienceBox has proven useful in the context of education and HEP data analysis.

The development of ScienceBox started in the context of the European Project Up To University (Up2U)[1], which aimed at bridging the gap between high schools and higher education. As a technical partner, CERN's objective was to provide tools and technologies commonly used in academia and big science to secondary school students in order to prepare them for their future careers. This led to the first implementation of ScienceBox, first as a set of containers running on a single host with `docker-compose`, then as a collection of templated `YAML` files and `bash` wrappers for scalable multi-host deployments in Kubernetes.

ScienceBox was deployed at the Poznan Supercomputing and Networking Center (PSNC) in Poland for the Up2U production service, and integrated with the Project Identity Provider (IDP) and Learning Management System (LMS), i.e., `Moodle`[2]. The deployment ran for over 2 years, fostering the use of sync&share and data analysis platforms in schools and providing learning materials to more than 200 students, teachers, and scientists. In addition to the production service for Up2U, ScienceBox had a leading role for education in response to the COVID-19 lockdown [4]: In April 2020, ScienceBox was deployed, together with `Moodle` and `eduMEET`[3] (a video-conferencing platform developed by GÉANT), on AWS as a free-access remote learning platform for European students.

ScienceBox also proved useful while investigating Big Data analysis tools for the TOTal Elastic and diffractive cross section Measurement (TOTEM)[4] experiment. An installation on the HNSciCloud [5] comprising 2,400 CPUs, 10 TB of memory, and 22 TB of storage featured ScienceBox in Kubernetes and an Apache Spark[5] cluster to scale-out computations massively. Access to the platform was granted to a group of TOTEM scientists who actively used SWAN [6] in place of the production service at CERN. With this setup, TOTEM scientists demonstrated that it is possible to reduce the time required for the final analysis steps by a factor of 200x and they successfully validated the obtained physics results [7].

[1] up2university.eu: Bridging the gap between schools and universities through informal education.

[2] moodle.org

[3] edumeet.org

[4] totem.web.cern.ch: The TOTEM Experiment.

[5] spark.apache.org

3 ScienceBox Reimagined

ScienceBox continues to provide a container-based version of open-source technologies developed and operated at CERN: i) **EOS** [8], exabyte-scale distributed storage hosting LHC physics data and users' files; ii) **CERNBox** [9], sync&share platform for science leveraging EOS for cross-platform file access and collaboration; iii) **SWAN** [6], Jupyter Notebooks with EOS storage and massive computing power from Spark clusters and GPUs; and iv) **CVMFS** [10], the HTTP-based Content Delivery Network (CDN) for distributing software powering the Worldwide LHC Computing Grid (WLCG). Readers are referred to [3] for further details on the cloud services included in ScienceBox.

Since its first implementation (which dates back to 2017, see Sec. 2), new tooling has become standard practice. As a result, ScienceBox has been re-implemented from scratch to reduce this technological gap and adopt modern, cloud-native technologies. With respect to its previous releases, ScienceBox now boasts a revised architecture based on Helm [2], a tool for the deployment, configuration, and lifecycle management of containerized applications. The choice of Helm is driven by the need for modularity and maintainability: Helm allows the description of services (and the resources they require) by defining packages of `YAML` templates, which can be combined to deploy complex service stacks. Specifically to ScienceBox, Helm allows decomposing the package into multiple sub-services, each of which has its own development and release cycle. As a result, it is now possible to deploy single components (e.g., the EOS stack or the CVMFS client), perform rolling upgrades of the desired service (e.g., following a bug-fix or the implementation of a new feature), or integrate ScienceBox with external services by looking at the interfaces defined in the `YAML` descriptors.

In what follows, we describe the new ScienceBox architecture based on Helm (Sec. 3.1), the additional components required to make it entirely autonomous (Sec. 3.2), and how to get started with a one-click demonstrator or deploy a fully scaled-out installation (Sec. 3.3).

3.1 Chart-based Architecture

Fig. 1 describes the ScienceBox architecture based on recently developed Helm charts. While ScienceBox stands at the root of the tree, its chart only expresses dependencies against other charts, making it an *umbrella* chart in Helm parlance. The blue oval shapes in the upper part of the diagram are dependent charts needed to make ScienceBox entirely autonomous (further details are provided in Sec. 3.2), while the three colored blocks in the bottom part show the core services:

- EOS, in red, is a collection of 3 required charts (`mgm`, `qdb`, and `fst`) and one optional chart (`mq`), each responsible for deploying one component of the EOS service. More specifically, the `mgm` chart deploys a stateless management node responsible for the hierarchical namespace, access control, quota, etc. The `qdb` chart deploys a set of 3 QuarkDB (a high-performance key-value store based on RocksDB[6]) instances with RAFT consensus, where the namespace and the configuration of the EOS instance are stored. The `fst` chart deploys the daemon writing files to persistent storage (either physical block devices or virtualized storage, including dynamically provisioned storage offered by cloud providers) and is horizontally scalable, allowing for the addition of storage on when required.
- CERNBox, in green, is a collection of 8 charts, 6 of which make use of the `revad` chart available in the CS3 Organization[7]. All 6 of the charts run the `revad`[8] binary, each providing a specialized backend integration service according to the configuration enforced via

[6] rocksdb.org

[7] cs3community.org

[8] reva.link

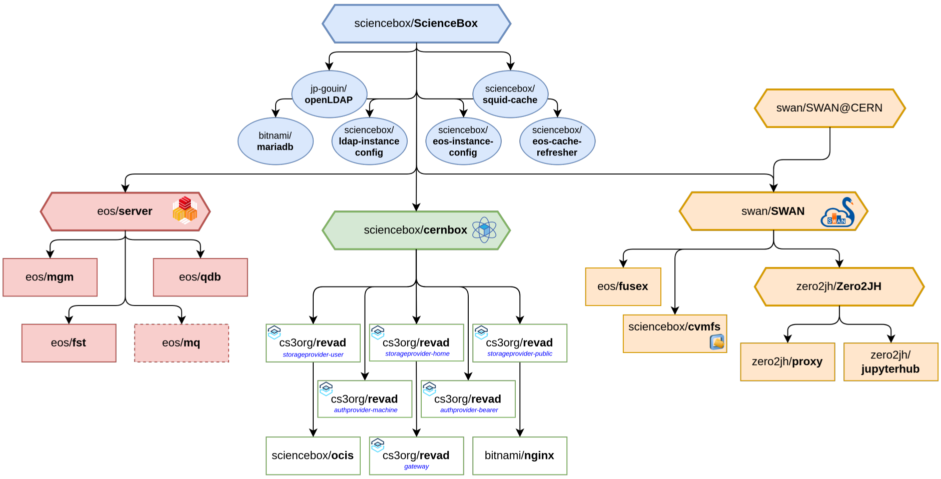

Figure 1. Revised ScienceBox Architecture based on Helm Charts.
Stretched hexagons represent umbrella charts, i.e., charts expressing dependencies against other charts; Rectangles represent component charts, i.e., charts describing the resources required to run one component; Ovals represent additional components developed to make ScienceBox fully self-standing, i.e., component charts providing commodity functionalities like caches, databases, identity providers, tooling for automated configuration, etc. The name preceding the slash indicates the project providing the chart, while the one following is the name of the specialized component deployed through the chart.

Helm, including access to EOS storage (i.e., `storageprovider-user/home/public`), user authentication (i.e., `authprovider-machine/bearer`), and request routing (i.e., `gateway`). In addition to `revad`, CERNBox incorporates the `ocis` chart running a subset of the ownCloud Infinite Scale (oCIS) extensions[9], and a Bitnami[10] `NGINX` chart serving the web interface with customized theme and assets.

- SWAN, in orange, relies on the upstream “Zero to JupyterHub with Kubernetes”[11] project, which encapsulates all the required components to deploy a multi-user Jupyter service. A few extra charts are added for CERN integrations: `fusex` for accessing EOS storage and `cvmfs` for pulling software libraries from centrally-curated repositories. All the charts mentioned are re-used to run the production SWAN service at CERN, demonstrating Helm’s usefulness in writing flexible templates, ultimately reducing duplication of work.

3.2 Additional Charts for Complete Portability

For production at CERN, the core services previously described rely on centrally-maintained IDP and Lightweight Directory Access Protocol (LDAP) servers, databases, monitoring and alerting, network load-balancers, secret stores, etc. While Kubernetes provides some of these functionalities, several extra services remain to be deployed to make ScienceBox fully portable, including:

- An LDAP server (i.e., `openLDAP`) storing the user account information and backing the authentication and authorization mechanisms implemented in each application. LDAP is also

[9] owncloud.dev/ocis
[10] bitnami.com
[11] z2jh.jupyter.org

```yaml
name: sciencebox
type: application
version: 0.0.8
description: The chart to deploy and configure ScienceBox

dependencies:
#
# --core--
# EOS
- name: server
  alias: eos
  version: 0.1.5
  repository: "oci://registry.cern.ch/eos/charts"
# CERNBox
- name: cernbox
  version: 0.0.5
  repository: "oci://registry.cern.ch/sciencebox/charts"
# SWAN
- name: swan
  version: 0.1.11
  repository: "oci://registry.cern.ch/swan/charts"
#
#
# --extras--
- name: nginx
- name: mariadb
- name: openldap
...
```

Table 1. `YAML` snippet of the `ScienceBox` chart.
The chart does not template resource definitions for service deployment but rather uses external charts by expressing them as dependencies. Core (i.e., EOS, CERNBox, SWAN) and extra (e.g., IDP, etc.) services are visually grouped in two blocks. One can selectively update single charts to newer versions or exclude entire services to be replaced with externally provisioned ones.

used to assign users with stable Unix User and Group identifiers (`uid`, `gid`) for accessing EOS storage from the upper-layer applications CERNBox and SWAN.

- A relational database storing the CERNBox sharing information, currently implemented with a MariaDB backend (i.e., `mariadb`).
- A HTTP cache (i.e., `squid-cache`) to accelerate access to CVMFS repositories and to reduce traffic to upstream servers in case of large-scale ScienceBox deployments.

Similarly to the core services, these are deployed by means of Helm charts reusing publicly-available charts freely available in upstream repositories.

3.3 From One-click Demo to Scaled-out Deployment

ScienceBox can be installed on diverse infrastructures, ranging from personal laptops to managed Kubernetes clusters on commercial providers' clouds. While it maintains the two deployment modes (i.e., one-click one-node demonstrator and multi-node scalable installation) described in [3], relevant changes have been implemented to improve maintainability and reduce the number of custom scripts in the container images.

The previous one-click demonstrator with `docker-compose` has been deprecated in favor of a new implementation based on `minikube`[12], which emulates a fully-fledged Kubernetes cluster on one node. This allows the use of Helm charts to deploy ScienceBox in its demonstrator form and removes the need to maintain specialized `YAML` for `compose`. To ease the installation, additional charts have been created to automatically configure ScienceBox. Such charts (shown in the top part of Fig. 1, i.e., `*-instance-config`) are pods of type *Job* used to pre-provision local users and configure the EOS instance with storage spaces and authentication methods. The demonstrator package also comes with scripts to automate the installation of `minikube`, Helm, Kubernetes command-line utilities, and a container runtime. Refer to sciencebox.web.cern.ch for up-to-date links to code repositories and instructions.

Secondly, and more importantly, the container images have been subject to a major rationalization and clean-up: All the custom setup scripts used for configuration have been moved to *Init Containers*, removing the need to run multiple processes (typically controlled by `supervisord`) in the application container and leaving it with the application binary only. This is a step forward towards improved maintainability, ease of debugging, and recommended best practices for deploying applications in containers.

As a result, it is now possible to reuse the same images and charts to deploy ScienceBox on a large set of resources over a multi-node cluster. Scalability is achieved horizontally by increasing the number of replicas of relevant containers, e.g., storage capacity can be added by running additional `eos/fst` pods backed by block devices or virtualized storage. Similarly, an increased population of SWAN Notebooks users can be supported by adding computing resources (i.e., increasing the number of worker nodes where users' sessions can be scheduled) and by replicating the containers providing access to EOS and CVMFS. In both cases, the process is automated through Helm and can be achieved with no downtime.

In the case of scaled-out deployments, one may want to integrate ScienceBox with existing on-premise IDPs or LDAP servers. This is possible by factoring out the dependencies expressed in the ScienceBox umbrella chart (see Tab. 1) and updating the relevant configuration elements to reach the existing services. Similarly, it is possible to extend the list of ScienceBox dependencies by adding extra charts to provide, for instance, additional monitoring, external storage, or job submission capabilities to specialized computing clusters.

4 Future Outlook

This new iteration of ScienceBox provides a major upgrade of the package to bring the underlying technology to the industry standards of containerized applications. At the same time, it updates the versions of the individual services part of ScienceBox as well as the configuration required to interconnect them. As part of the upgrade, the authors invested in a new project page and documentation (sciencebox.web.cern.ch) to lower the entry barrier. Overall, this was an ambitious leap from previous work [3] towards enabling and extending the package and improving accessibility for both new users and contributors.

Proof of this fact is the interest aroused during the 2023 CS3 Conference [11] for the work presented in this paper: The current state of ScienceBox provides a base for application developers to integrate their own, leveraging on the flexibility of containerized development and standard APIs implemented by services part of the package. For instance, various popular collaborative applications, e.g., `CodiMD`[13], `draw.io`[14], can be connected to CERNBox via the CS3-developed `WOPI` server[15] to provide collaborative document-editing capabilities.

[12] minikube.sigs.k8s.io

[13] github.com/hackmdio/codimd

[14] drawio.com

[15] github.com/cs3org/wopiserver

ScienceBox has also recently been used by CERN's Onboard project[16], which fosters the utilization of open-source analysis tools for reproducible research and the exploration of open data. By a custom deployment on our OpenStack cloud, ScienceBox allowed for the provisioning of dedicated project spaces to Onboard users and for the sharing of analysis notebooks across the pool of users. In this case, ScienceBox has been integrated with CERN single sign-on service to leverage its support for external user identities.

Lastly, many WLCG tiers demonstrated increasing interest in storing physics data on existing storage systems while maintaining the data processing paradigms proper of the HEP community. In practical terms, tier sites would benefit from a thin layer composed of CVMFS for the analysis software and EOS storage for Grid storage to be plugged on top of their existing IT infrastructures. As demonstrated in [12], ScienceBox fulfills this need thanks to its modular and allows for the reuse of CVMFS and EOS charts in combination with additional charts for batch processing, accounting, and logging. The feature set in ScienceBox charts does not provide yet full coverage for tier site needs: While ScienceBox focuses fuse access to local users, a tier site will likely require native `xrootd` access, third-party-copy support, and certificate-based authentication and authorization. Support for these features is being developed with the authors of [12] and will be available in future releases of ScienceBox.

Acknowledgements

The work described in this paper has been produced in the context of the CS3MESH4EOSC Project. CS3MESH4EOSC received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 863353.

References

[1] K. Matthias, S. Kane, *Docker Up & Running: Shipping Reliable Containers in Production* (O'Reilly Media, Sebastopol, California, United States, 2015)

[2] S. Gokhale, R. Poosarla, S. Tikar, S. Gunjawate, A. Hajare, S. Deshpande, S. Gupta, K. Karve, *Creating Helm Charts to ease deployment of Enterprise Application and its related Services in Kubernetes*, in *2021 International Conference on Computing, Communication and Green Engineering (CCGE)* (2021), pp. 1–5

[3] E. Bocchi, L. Canali, D. Castro, P. Kothuri, H. Labrador, M. Malawski, J. Moscicki, P. Mrowczynski, *ScienceBox: Converging to Kubernetes containers in production for on-premise and hybrid clouds for CERNBox, SWAN, and EOS*, EPJ Web of Conferences **245**, 07047 (2020)

[4] *CERN technologies contribute to openUp2U, a learning platform for schools in Europe*, CERN News (2020), accessed: 2023-11-20

[5] M. Gasthuber, H. Meinhard, R. Jones, *HNSciCloud - Overview and technical Challenges*, Journal of Physics: Conference Series **898**, 052040 (2017)

[6] D. Piparo, E. Tejedor, P. Mato, L. Mascetti, J. Moscicki, M. Lamanna, *SWAN: A service for interactive analysis in the cloud*, Future Generation Computer Systems **78**, 1071 (2018)

[7] V. Avati, M. Blaszkiewicz, E. Bocchi, L. Canali, D. Castro, J. Cervantes, L. Grzanka, E. Guiraud, J. Kaspar, P. Kothuri et al., *Big Data Tools and Cloud Services for High Energy Physics Analysis in TOTEM Experiment*, in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)* (2018), pp. 5–6

[16] onboard.web.cern.ch

[8] A. Peters, E. Sindrilaru, G. Adde, *EOS as the present and future solution for data storage at CERN*, Journal of Physics: Conference Series **664**, 042042 (2015)

[9] L. Mascetti, H.G. Labrador, M. Lamanna, J. Mościcki, A. Peters, *CERNBox + EOS: end-user storage for science*, Journal of Physics: Conference Series **664**, 062037 (2015)

[10] J. Blomer, C. Aguado-Sánchez, P. Buncic, A. Harutyunyan, *Distributing LHC application software and conditions databases using the CernVM file system*, Journal of Physics: Conference Series **331**, 042003 (2011)

[11] S. Alfageme Sainz, *Driving the ScienceBox package into the future*, CS3 2023 - Cloud Storage Synchronization and Sharing (2023), `https://cds.cern.ch/record/2855381`

[12] R. Taylor, J.R. Albert, F.H. Barreiro Megino (ATLAS), *A grid site reimagined* (2023)