

KUBERNETES TESTBED CLUSTER FOR THE LIGHTWEIGHT SITES PROJECT

Iuliia Gavrilenko^{1, a}, Mayank Sharma^{2, b}, Maarten Litmaath^{2, c}

¹ *Plekhanov Russian University of Economics, 36 Stremyanny per., Moscow, 117997, Russia*

² *CERN, CH-1211 Geneva 23 Switzerland*

E-mail: ^a yulya952@rambler.ru, ^b mayank.sharma@cern.ch, ^c maarten.litmaath@cern.ch

The Worldwide LHC Computing Grid (WLCG) is a global collaboration of more than 170 computing centres in 42 countries and the number of sites is expected to grow in the next years. However, provisioning of the resources (compute, network, storage) at new sites to support WLCG workloads still is not a straightforward task and often requires significant assistance from WLCG experts. Recently, a large fraction of the WLCG community has expressed that such overheads could be reduced at their sites through the use of prefab Docker containers or OpenStack VM images, along with the adoption of popular tools like Puppet for configuration. In 2017, the Lightweight Sites project was initiated to construct shared community repositories providing such building blocks. These repositories are governed by a single Lightweight Site Specification Document which describes a modular way to define site components such as Batch Systems, Compute Elements, Worker Nodes, Networks etc. Implementation of the specification is based on popular orchestration technologies – Docker Swarm, Kubernetes and possibly others. Here we discuss how the use of Kubernetes was pioneered on a testbed cluster for deploying Lightweight grid sites. The research was mainly focused on controlling the lifecycle of containers for compute element, batch system and worker nodes. Also, some parameters for benchmarking and evaluation of the performance of different implementations were introduced.

Keywords: WLCG, SIMPLE Grid project, Kubernetes, Docker Swarm, container, Master Node, Worker Node.

© 2018 Iuliia Gavrilenko, Mayank Sharma, Maarten Litmaath

Today, the WLCG consists of more than 170 computing centres running a great variety of service flavours, configurations and software packages that define this global infrastructure. As a result, there are no easy recipes to that will work for many institutes. Instead, setting up and maintaining sites may take a lot of time and experience from site admins. Many years ago, attempts were already made to solve this problem. For example, in 2004 the Go2Grid project was launched [3] - a web-oriented tool to simplify service installation and configuration on the grid sites at the time, but as its framework removed autonomy from the sites, it did not become popular and the project finally was stopped. Today, modern technologies used in the Lightweight Sites project allow sites to set up and run grid services more easily, yet retain full autonomy.

Within the Lightweight Sites project a framework has been designed to simplify the installation and management of WLCG sites: SIMPLE - Solution for Installation, Management and Provisioning of Lightweight Elements. It aims to support diversity in WLCG sites with minimal oversight and operation efforts. It intends to keep the grid service layout and functionality the same, but easier for site admins to set up and maintain. The SIMPLE framework architect and main developer is Mayank Sharma. [2] It is currently being co-developed and tested using the infrastructure resources mainly at CERN and CBPF (Brazil). At the same time, admins and developers from other WLCG sites are encouraged to engage in this open source and community-driven effort to benefit from its outcomes.

In the framework's ecosystem, grid services like compute elements, batch systems etc. are represented through component repositories. These are essentially GitHub repositories that contain a Dockerfile for the containerized grid service. For instance, in the Cream-CE repository, the installation of configuration information for the Cream Compute Element is described in its Dockerfile [7]. The framework utilizes information provided by site admins to start and configure containers for several grid services required to set up a WLCG site. Modern container orchestration systems like Docker Swarm and Kubernetes can highly simplify deployment and maintenance of such container clusters which run grid services. For the first release of the framework, Docker Swarm was chosen for orchestration of the grid service containers because of its close integration with the Docker ecosystem.

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications [1] developed at Google. It has become highly popular among some of the largest distributed computing projects in recent years and currently is the market leader among its competitors. In the context of particle physics experiments it has for example been used in the BigPanDA self-monitoring alarm system for ATLAS [4] and in the cloud-based computing services at IHEP for web-based data analysis in the LHAASO [5] experiment. The SIMPLE framework's modular architecture supports plug and play of different container orchestration technologies without disrupting its overall functionality. For these reasons, adding support for Kubernetes is both desirable and feasible.

Before Kubernetes can be integrated into the SIMPLE Grid Framework, it is important to investigate the extent to which the capabilities of Kubernetes can be leveraged by the framework. To perform this investigation, we needed to probe the complexity of instantiating and running the existing containers for Cream-CE, Torque Batch System and Torque Worker through Kubernetes in a development testing environment. We started looking into the various possibilities for installation and configuration of a Kubernetes cluster. One of the easiest ways to achieve this was using Minikube [6], a tool to run a single-node Kubernetes cluster inside a VM. To scale such a cluster to multiple nodes, however, it was necessary to install and configure Kubernetes in a master-worker configuration on multiple hosts. For a multi-node cluster, Kubernetes allows a host to act either as a master or a worker. The master nodes are set up for High Availability (HA) and help achieve high performance as well as failover in order to reduce downtimes. A worker usually runs Docker containers inside Kubernetes Pods. A pod is a group of one or more containers that share their storage and network resources. In order to keep the site deployment close to real-world implementations and the initial container deployments achieved via Docker Swarm, we chose to use single-container pods, one deployed on each worker. In a more practical setting, however, the number of containers in a pod and the number of pods per worker can vary depending on the requirements and resource availability at the site. Under these constraints, the installation of Kubernetes was manually performed on a cluster of three nodes on VMs provisioned via the CERN OpenStack infrastructure. After several trials, a working installation of Kubernetes in master-worker configuration was attained. In order to automate

the sequence of actions performed to achieve the kube cluster, we developed Ansible playbooks that can run the process for future installations. Ansible is a popular configuration management tool from RedHat to automate IT infrastructure configuration. The Ansible playbooks developed for the use of Kubernetes in the SIMPLE framework are available in the project area on GitHub [8].

With the kube cluster set up and configured, we were able to deploy grid services on top of it. The Torque Worker Node container was successfully deployed in a Kubernetes pod on the workers in our testbed cluster. For the time being, the container internally uses the legacy grid service configuration framework YAIM [9] to configure itself at boot time through an Entrypoint script called `init.sh`, mounted inside the configuration directory of the container which is mounted into the container before it starts. By design, the entrypoint is automatically executed when the container starts up. However, in our case, we first had to get the configuration files for YAIM into the Kubernetes pod that hosts the container. To obtain a configured container, we had to execute the Kubernetes deployment, start the container, mount the configuration files into the pod and then execute the entrypoint script. This sequence of operations differs from the Docker Swarm implementation, where the notion of pods does not exist. The exact process of setting up the worker node container and the Kubernetes deployment file ultimately used were captured in yet another Ansible Playbook.

While all this was achieved in less than a month, our immediate follow-up tasks are to deploy the Cream-CE and Torque Batch System container onto the testbed and to establish an optimum networking strategy to connect the pods hosting the compute element, batch system and worker node containers. The former task is relatively simple, given that the startup process for a Cream-CE container is similar to that of the Torque worker node container. The latter task is being investigated, to determine the best choice from the different network drivers and networking strategies offered by Kubernetes. In the near future, we will publish the Ansible playbooks that will install and configure the Kubernetes cluster. This would enable anyone to deploy and configure the grid service containers with a few Ansible commands. That is when other components of the SIMPLE Grid Framework can start integrating the playbooks to allow site admins to set up and configure their grid site with Kubernetes as the container orchestration tool used under the hood.

The figure below describes how this integration would look.

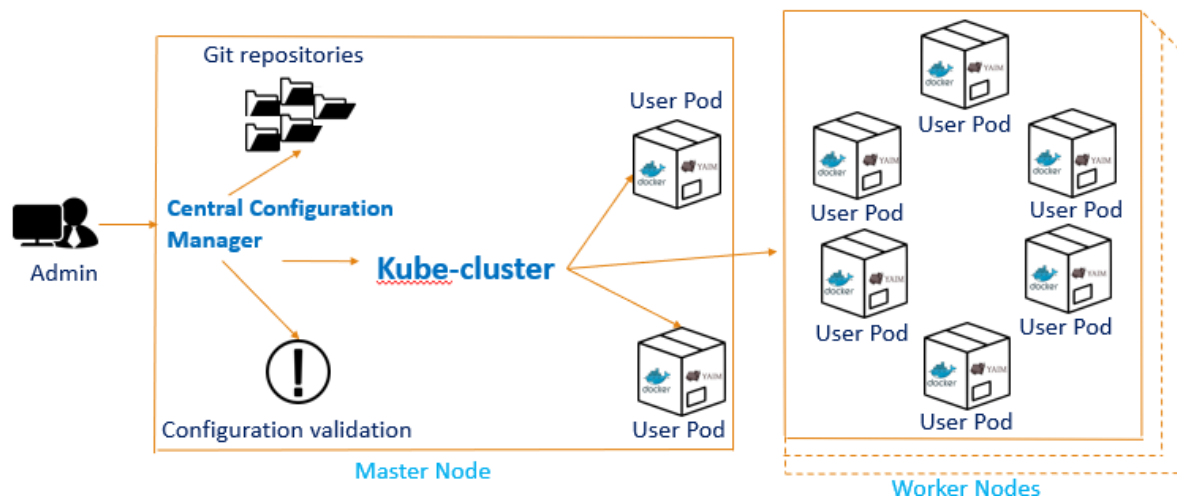


Figure 4 Kubernetes in the SIMPLE Grid project

The site admin provides the SIMPLE framework with a description of the site infrastructure and required grid services through a single YAML file. Within the framework, the Central Configuration Manager fetches the component repositories for the grid services. It then executes the Ansible Roles and Playbooks that will be produced as the output of our work, to set up a Kubernetes cluster and deploy the containers for each of the grid services, resulting in a functional WLCG site.

In conclusion, we have described the purpose of the Lightweight Sites project, the SIMPLE deployment framework, and how Kubernetes is foreseen to play a major role in it. We have successfully finished the first steps toward the integration of Kubernetes into the framework and laid out the subsequent tasks currently being worked on. We expect Kubernetes to become a popular technology also within the SIMPLE framework!

References

- [1] Kubernetes. Production-Grade Container Orchestration [The orchestration platform]. Available at: <https://kubernetes.io>. (accessed 23.07.2018).
- [2] Mayank Sharma. The SIMPLE Grid project. Available at: <https://wlcg-lightweight-sites.github.io>. (accessed 25.09.2018).
- [3] A. Retico, A. Usai, G. Diez-Andino Sancho, I. Tkachev, M. Schulz, O. Keeble, P. Nyczyk. GoToGrid - A Web-Oriented Tool in Support to Sites for LCG Installations, proceedings of CHEP'2004 conference, 27 September - 1 October 2004, Interlaken, Switzerland, <https://cds.cern.ch/record/865759>.
- [4] A. Alekseev, T. Korchuganova, S. Padolski. The BigPanDA self-monitoring alarm system for ATLAS, to appear in proceedings of GRID'2018 conference, 10 – 14 September 2018, Dubna, Russia.
- [5] Qiulan Huang, Weidong Li, Haibo Li, Yaodong Cheng, Tao Cui, Jingyan Shi, Qingbao Hu. Cloud-based Computing for LHAASO experiment at IHEP, to appear in proceedings of GRID'2018 conference, 10 – 14 September 2018, Dubna, Russia.
- [6] dlorenc, luxas et. al. Minikube (Version 0.30.0) [Software] available at <https://github.com/kubernetes/minikube/tree/v0.30.0> . (accessed 23.09.2018).
- [7] Mayank Sharma. Dockerfile for Cream-CE. `wlcg_lightweight_site_cream_ce` [Software] available at https://github.com/WLCG-Lightweight-Sites/wlcg_lightweight_site_ce_cream/blob/master/yaim/Dockerfile . (accessed 22.09.2018).
- [8] Iuliia Gavrilenko. Kubernetes cluster. Available at: https://github.com/WLCG-Lightweight-Sites/simple_grid_kube_cluster/pull/2/files (accessed 04.11.2018)
- [9] Maarten Litmaath. The description of YAIM version 4. Available at: <https://twiki.cern.ch/twiki/bin/view/LCG/YaimGuide400> (accessed 04.11.2018)