14th International Workshop on Science Gateways (IWSG 2022), 15-17 June 2022

Abstract: Everyday orchestration with Docker on Kubernetes

James DesLauriers[1], Resmi Arjun[1], Jozsef Kovacs[1,2], Tamas Kiss[1]

[1]Centre For Parallel Computing
University of Westminster, London UK
{j.deslauriers, r.arjun, j.kovacs, t.kiss}@westminster.ac.uk

[2]ELKH SZTAKI
Budapest, Hungary
jozsef.kovacs@sztaki.hu

Abstract

Many different industries and applications can benefit from the run-time management of containerised applications that is offered by a container orchestrator such as Kubernetes [1]. Orchestrators can provide one-click deployment of a complex microservices architecture, smart scheduling across a cluster of compute nodes, self-healing abilities, automated scaling, zero-downtime rolling updates and much more.

Science Gateways providing an interface for researchers looking to deploy applications in either the long- or short-term are an ideal candidate for this technology. However, deploying a Kubernetes cluster and writing Kubernetes manifests for each microservice are technically complex operations with a high learning curve. An initial investment of effort to prepare the cluster is equalled by the effort required for day-to-day cluster management and the constant need to author new manifests for new applications. Developers of Science Gateways may struggle with efficiently deploying and maintaining a backend Kubernetes cluster and likewise, users of Science Gateways cannot be expected to develop the native Kubernetes manifests required to deploy their applications.

We propose a pair of tools aimed at empowering developers of Science Gateways with easier access to the benefits afforded by container orchestration: DocKubeADT and MiCADO [2].

MiCADO is a one-click Kubernetes cluster deployed with Ansible. On top of the standard Kubernetes offering, MiCADO adds vendor-free auto-scaling for both containers and the underlying virtual machines, as well as improved security and monitoring. MiCADO offers a REST API for the submission, update and deletion of Application Description Templates (ADT) [3], which define the size of the Kubernetes cluster and the applications to schedule across it. The ADT is based heavily on the generic TOSCA [4] format proposed by OASIS.

DocKubeADT is a new tool that translates a Docker-Compose file to MiCADO's native ADT format. Docker container owners no longer need to learn an orchestrator specific format but can instead rely on the familiar toolchain they use when deploying their container locally. DocKubeADT compiles the Docker descriptor to a native Kubernetes manifest with Kompose[1], and then further compiles it to an ADT, ensuring correctness. Properly placed in the submission pipeline, DocKubeADT can be used to submit a common Docker-Compose file directly to MiCADO, which in turn realises the Kubernetes deployment of all services described in the Docker-Compose, ready to benefit from the self-healing and smart scheduling of Kubernetes.

DocKubeADT is already being used in the DIGITbrain platform [5], which empowers manufacturing SMEs with easy access to Digital Twins via a cloud-based marketplace. Future work will see DocKubeADT embedded in a Science Gateway for PITHIA-NRF [6], as part of an e-Science Centre for Space researchers.

Acknowledgement

This work was funded by the following projects: DIGITbrain - Digital twins bringing agility and innovation to manufacturing SMEs, by empowering a network of DIHs with an integrated digital platform that enables Manufacturing as a Service, project, No. 952071, European Commission (EU H2020); and PITHIA-NRF - Plasmasphere Ionosphere Thermosphere Integrated Research Environment and Access services: a Network of Research Facilities, project no. 101007599, European Commission (EU H2020).

Keywords – container orchestration, Docker, Kubernetes, cloud-based services, MiCADO.

References

[1] "Kubernetes, 'Production-Grade Container Orchestration.'" https://kubernetes.io/ (accessed Mar. 07, 2022).

[2] J. DesLauriers *et al.*, "Cloud apps to-go: Cloud portability with TOSCA and MiCADO," in *Concurrency and Computation* , 2020, p. e6093, doi: 10.1002/cpe.6093

[3] G. Pierantoni, T. Kiss, G. Terstyanszky, J. DesLauriers, G. Gesmier, and H. Van Dang, "Describing and Processing Topology and Quality of Service Parameters of Applications in the Cloud," *J. Grid Comput.*, vol. 18, no. 4, pp. 761–778, Dec. 2020, doi: 10.1007/s10723-020-09524-0.

[4] "Topology and Orchestration Specification for Cloud Applications Version 1.0." http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html (accessed Mar. 10, 2022).

[5] J.DesLauriers, J. Kovacs, T.Kiss, Dynamic Composition and Automated Deployment of Digital Twins for Manufacturing, IWSG 2021, 10-11 June 2021, on-line event.

[6] G. Pierantoni, T. Kiss, A. Bolotov, D. Kagialis, J. DesLauriers, A. Ullah, H. Chen, D. Chan You Fee, H. Dang, J. Kovacs, A. Belehaki, T. Herekakis, I. Tsagouri, S. Gesing: Towards a Reference Architecture based Science Gateway Framework with Embedded E-Learning Support, Concurrency and Computation, Practice and Experience, Wiley, 2022, https://doi.org/10.1002/cpe.6872

[1] Kompose. https://kompose.io/