2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)

Transparent deployment of scientific workflows across clouds – Kubernetes approach

Michal Orzechowski*, Bartosz Balis*, Krystian Pawlik*, Maciej Pawlik* and Maciej Malawski*
*AGH Universtity of Science and Technology
Krakow, Poland
Email: {morzech,balis,malawski}@agh.edu.pl

***Abstract*—We present an end-to-end solution for automation of scientific workflow deployment and execution on distributed computing infrastructures. The solution integrates de-facto standard and widely adopted tools, including Terraform and Kubernetes, with our HyperFlow workflow management system. In such a solution, infrastructure providers are abstracted away thanks to generic Kubernetes layer. However, we also support other computing infrastructures, both containerized such as Kubernetes or Amazon ECS, and non-containerized, e.g. Amazon Lambda, in a single unified approach. The resulting solution enables execution of hybrid workflows that utilize multiple computing infrastructures, and significantly lowers the complexity related to management of repeatable infrastructures for the execution of scientific workflows and conducting scientific workflow research.**

I. Introduction

Deployment and execution of applications on diverse and heterogeneous infrastructures remain a significant challenge, notably for scientific applications which have often the form of complex workflows consisting of multiple tasks, requiring specialized software packages or libraries, and using a Workflow Management Systems (WMS). Technologies such as virtualization, application containers, or infrastructure automation, provide means to mitigate this complexity.

Cloud providers supply advanced tools for provisioning of the infrastructure and deployment of applications. However the diversity of cloud solutions (e.g. OpenStack, AWS, or GCP) makes it difficult to avoid spending significant resources in order to avoid vendor lock-in. Kubernetes[1] solves this problem by providing a mature set of well-standardized principles and practices for running software in distributed virtualized environment. Just recently, a number of articles[2] refer to Kubernetes as a *Cloud Native Operating System* while major cloud providers and popular platforms started offering *one-click* Kubernetes deployments.

In this paper, we report on a work in progress solution for automation for scientific workflow deployment and execution. Unlike existing solutions, we utilize Kubernetes, together with infrastructure automation tool Terraform,[3] to abstract away differences between infrastructure providers. However we go beyond Kubernetes and support both containerized and non-containerized infrastructures. The solution is integrated with our HyperFlow workflow management system [1]. The main contributions can be summarized as follows:

- we provide an initial implementation of Kubernetes native support for running HyperFlow and scientific workflows with hybrid infrastructure support – workflows can transparently run on multiple cloud infrastructures;
- we describe a *unified solution* for both containerized (managed and non-managed Kubernetes, Amazon ECS) and non-containerized (AWS Lambda) infrastructures;
- we present an *end-to-end solution* that automates full workflow lifecycle, including provisioning, deployment, execution, and monitoring.

II. Related Work

Many existing workflow management systems utilize proprietary tools for infrastructure provisioning, e.g. Occopus (WS-PGRADE) [2] or PRECIP (Pegasus) [3]. However, the cost of development and maintenance of such tools is too high, so they only support a small fraction of available clouds and quickly become outdated. A few WMS integrate with Kubernetes on various levels. In Nextflow [4], when deploying a workflow a pod running the main workflow driver application is deployed, and it orchestrates and submits the execution of workflow tasks as separate pods. The Galaxy [5] workflow system can be deployed inside Kubernetes in an automated fashion through the use of Helm[4] Charts. The Galaxy Kubernetes job runner allows Galaxy to offload jobs to Kubernetes, either when Galaxy runs inside the Kubernetes or when deployed on the outside infrastructure. Pachyderm[5], a workflow system for creating distributed and reproducible pipelines is built natively on top of Kubernetes and uses Kubernetes Go library to deploy itself and the workflows on the orchestration cluster. Overall, no WMS fully exploits standard infrastructure automation tools in order to provide an *end to end* solution for infrastructure provisioning, workflow deployment, execution and monitoring.

III. Solution description

A high-level overview of the proposed solution is depicted in Fig. 1. The solution automates the entire workflow lifecyle,

[1]https://kubernetes.io
[2]https://blogs.oracle.com/cloud-infrastructure/kubernetes-a-cloud-and-data-center-operating-system
[3]https://www.terraform.io
[4]https://helm.sh/
[5]http://pachyderm.io

978-1-7281-0359-4/18/$31.00 ©2018 IEEE
DOI 10.1109/UCC-Companion.2018.00020

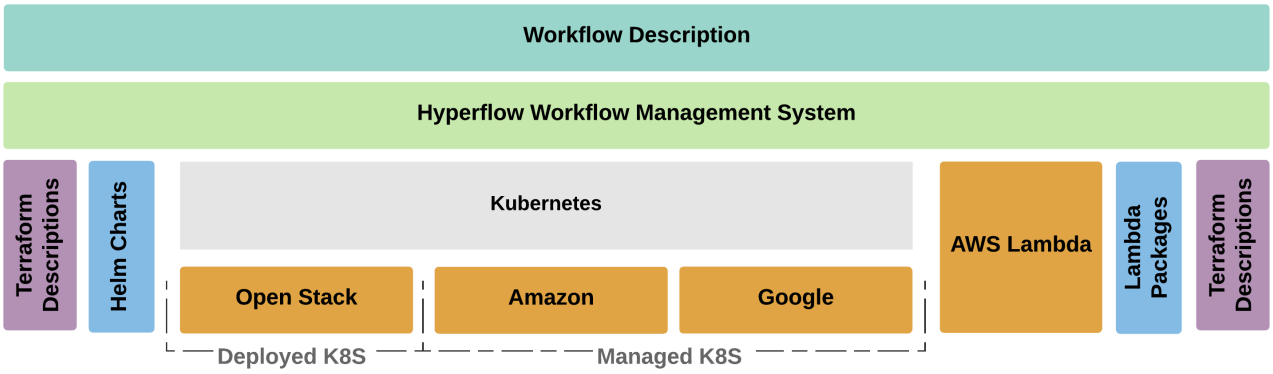

Fig. 1. Workflow provisioning, deployment and execution with HyperFlow.

including provisioning of the infrastructure, workflow deployment, and workflow execution/monitoring. The single entry-point to provisioning and deployment is the Terraform tool. We provide Terraform descriptions describing infrastructure configurations for deployment of the HyperFlow WMS and workflows on both containerized and non-containerized infrastructures. Consequently, Terraform can create Kubernetes instances either by provisioning the virtual machines and then the cluster as in the case of OpenStack, or create infrastructure resources on managed Kubernetes services of Amazon AWS and Google Cloud. Next, Helm Charts are used to deploy Hyperflow on the Kubernetes cluster. In the case of AWS Lambda, in turn, Terraform scripts deploy workflow components using Lambda deployment packages.

Once the infrastructure is ready, the user can run workflows using the HyperFlow engine. A HyperFlow workflow is a graph of tasks expressed in a JSON format [1]. Importantly, the workflow description is infrastructure-independent. For each workflow, a *workflow function* (implemented in JavaScript) needs to be provided. This function is where the mapping to the underlying infrastructure is done. To this end, HyperFlow provides *executors* enabling execution of jobs on remote resources. For example, on Kubernetes we use the HyperFlow Master-Worker executor, while AWS Lambda is supported by a dedicated HyperFlow Lambda executor. Another aspect automated by HyperFlow is workflow monitoring and auto-scaling (as long as it is supported by the infrastructure). Infrastructure-specific monitoring components are deployed to collect metrics. If supported, auto-scaling rules are also defined and controlled by user-defined thresholds. We are also working on a generic two-level Kubernetes auto-scaling mechanism. Depending on the monitoring data supplied by the native Kubernetes monitoring system, HyperFlow will scale up or down the number of pods (level 1), or the number of nodes (level 2) running in the Kubernetes cluster.

IV. Conclusion

Infrastructure automation reduces complexity related to creation of repeatable infrastructures for scientific workflows and contributes to the reproducibility of scientific experiments. It also facilitates automation of workflow research that frequently involves running of experiments in different computing infrastructures and in diverse configurations. We presented a solution for automation of the entire scientific workflow lifecycle: infrastructure provisioning, workflow deployment, and workflow execution/monitoring. This solution integrates standard tools, including Kubernetes and Terraform, with the HyperFlow workflow management system. Thanks to separation of workflow description from infrastructure description and provisioning, we support hybrid workflows that can utilize multiple computing infrastructures. Introducing Kubernetes to act as de facto cloud native operating system, further simplifies the infrastructure provisioning process.

Acknowledgment

This work was supported by the National Science Centre, Poland, grant 2016/21/B/ST6/01497. HyperFlow is also partially supported by the AGH statutory Grant no. 11.11.230.337. MO is also grateful for his doctoral grant at AGH.

References

[1] B. Balis, "Hyperflow: A model of computation, programming approach and enactment engine for complex distributed workflows," *Future Generation Computer Systems*, vol. 55, pp. 147 – 162, 2016.

[2] P. Kacsuk, G. Kecskemeti, A. Kertesz, Z. Nemeth, A. Visegradi, and M. Gergely, "Infrastructure aware scientific workflows and their support by a science gateway," in *Science Gateways (IWSG) Workshop*. IEEE, 2015, pp. 22–27.

[3] S. Azarnoosh, M. Rynge, G. Juve, E. Deelman, M. Niec, M. Malawski, and R. F. Da Silva, "Introducing precip: an API for managing repeatable experiments in the cloud," in *2013 IEEE 5th CloudCom Conference*. IEEE, 2013, pp. 19–26.

[4] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, "Nextflow enables reproducible computational workflows," *Nature Biotechnology*, vol. 35, pp. 316 EP –, 04 2017.

[5] E. Afgan *et al.*, "The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update," *Nucleic Acids Research*, vol. 46, no. W1, pp. W537–W544, 2018.