Application Note

Galaxy-Kubernetes integration: scaling bioinformatics workflows in the cloud

Pablo Moreno[a], Luca Pireddu[b], Pierrick Roger[c], Nuwan Goonasekera[f], Enis Afgan[g], Marius van den Beek[l], Sijin He[a], Anders Larsson[f], Daniel Schober[d], Christoph Ruttkies[d], David Johnson[h], Philippe Rocca-Serra[h], Ralf JM Weber[i], Björn Gruening[j], Reza M Salek[k], Namrata Kale[a], Yasset Perez-Riverol[a], Irene Papatheodorou[a], Ola Spjuth[e], and Steffen Neumann[d]

[a] European Molecular Biology Laboratory, European Bioinformatics Institute, Cambridge, United Kingdom
[b] Distributed Computing Group, CRS4, Italy;
[c] CEA, Laboratory for Data Analysis and Systems Intelligence, MetaboHUB, France
[d] Leibniz Institute of Plant Biochemistry, Dept. of Stress and Developmental Biology, Halle, Germany.
[e] Department of Pharmaceutical Biosciences and Science for Life Laboratory, Uppsala University, Sweden
[f] Melbourne Bioinformatics & EMBL Australia Bioinformatics Resource, University of Melbourne, Australia
[g] Johns Hopkins University, Department of Biology, USA;
[h] Department of Engineering Science, University of Oxford, United Kingdom;
[i] School of Biosciences, University of Birmingham, Edgbaston, Birmingham, B15 2TT, United Kingdom;
[j] University of Freiburg, Department of Computer Science, Georges-Köhler-Allee 106, 79110, Freiburg, Germany
[k] International Agency for Research on Cancer, Lyon CEDEX 08, France
[l] Institut Curie, PSL Research University, Paris, France

Received on XXXXX; revised on XXXXX; accepted on XXXXX
Associate Editor: XXXXXXX

Summary: Making reproducible, auditable and scalable data-processing analysis workflows is an important challenge in the field of bioinformatics. Recently, software containers and cloud computing introduced a novel solution to address these challenges. They simplify software installation, management and reproducibility by packaging tools and their dependencies. In this work we implemented a cloud provider agnostic and scalable container orchestration setup for the popular Galaxy workflow environment. This solution enables Galaxy to run on and offload jobs to most cloud providers (e.g. Amazon Web Services, Google Cloud or OpenStack, among others) through the Kubernetes container orchestrator.

Availability: All code has been contributed to the Galaxy Project and is available (since Galaxy 17.05) at https://github.com/galaxyproject/ in the *galaxy* and *galaxy-kubernetes* repositories. https://public.phenomenal-h2020.eu/ is an example deployment.

Suppl. Information: Supplementary Files are available online.

Contact: pmoreno@ebi.ac.uk, European Molecular Biology Laboratory, EMBL-EBI, Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, UK, Tel: +44-1223-494267, Fax: +44-1223-484696.

1 INTRODUCTION

The ever growing data volumes and data production rates in bioinformatics (Perez-Riverol et al. 2017) require the support of scalable and performant data processing architectures. Cloud computing has gained momentum, providing affordable, large-scale computing resources for high-throughput data processing. Bioinformatics has recently seen the adoption of this provisioning paradigm (Agapito *et al.*, 2017; Langmead and Nellore, 2018), through private clouds (e.g. local OpenStack (Sefraoui *et al.*, 2012)) and public cloud providers. However, most of these solutions are cloud specific, making difficult to reuse in more than one cloud.

Software containers have been recently adopted in bioinformatics–and also by the computing industry at large–as a way to distribute and manage software. Additionally, the development of container orchestrators enables the coordinated execution of a set of containers to host long-running services (e.g., database, web server and application server working together in separate containers) and batch jobs, acting as a clusters for containers. Kubernetes (kubernetes.io), the current *de-facto* container orchestrator solution, executes container-based work-loads in scalable infrastructure on many different cloud providers.

The Galaxy workflow environment (Afgan *et al.*, 2018) allows researchers to create, run and distribute bioinformatics workflows in a reproducible manner. Recent advancements in Galaxy simplified tool dependency resolutions through Bioconda (Grüning, Dale, *et al.*, 2018) and Biocontainers (da Veiga Leprevost *et al.*, 2017). These have paved the way for the use of container orchestration systems in the cloud with Galaxy.

Deploying software on cloud infrastructure normally requires specialised expertise that is foreign to most bioinformaticians. Moreover, each cloud provider implements different mechanisms, further complicating the matter. In the Galaxy context, CloudMan (Afgan *et al.*, 2012) was the first solution available to deploy Galaxy on Amazon Web Services (AWS), but support for other clouds requires extensive development efforts.

Here we introduce Kubernetes-based container orchestration and provisioning for Galaxy. Kubernetes acts as an abstraction layer between Galaxy and the different cloud providers, allowing Galaxy to run on every cloud provider that supports Kubernetes (>10 cloud providers currently). The solution described here combines ease of deployment, replicability of deployment, scalable computational performance, support for multiple cloud providers and use of containers to offload tools for distributed processing.

2 GALAXY-KUBERNETES ARCHITECHTURE

The Galaxy-Kubernetes setup is divided in three main concepts: (i) the Kubernetes Runner for Galaxy; (ii) the Galaxy container image with Kubernetes support; and (iii) packaging to automate the deployment of Galaxy in Kubernetes (Galaxy Helm chart). The Kubernetes Runner translates Galaxy jobs into Kubernetes jobs, specifying the container that needs to be executed for the Galaxy tools, and wrapping the commands that the job needs to execute for running them inside the container, once it is deployed to the Kubernetes cluster. The runner deals with the shared file system between Galaxy and the container running the tool (a requirement for this integration) (**Figure 1**). In addition, the runner handles user permissions, tool CPU/RAM requirements and the general management of the Kubernetes job lifecycle for Galaxy. The runner sends the job to Kubernetes and then enquires about the state of the job. This runner, after some minimal configuration of the appropriate plugin field and tools desired in the Galaxy job configuration file, is able to operate within any Kubernetes cluster that shares a filesystem with Galaxy.

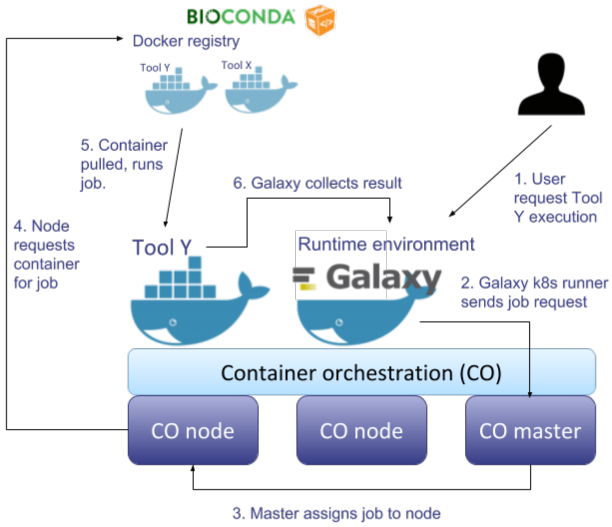
BIOCONDA
Docker registry
Tool Y
Tool X
5. Container pulled, runs job.
6. Galaxy collects result
1. User request Tool Y execution
4. Node requests container for job
Tool Y
Runtime environment
Galaxy
2. Galaxy k8s runner sends job request
Container orchestration (CO)
CO node
CO node
CO master
3. Master assigns job to node

Figure 1: Galaxy running within Kubernetes as a container, and offloading jobs through containers to other nodes in the same Kubernetes cluster.

At this point, the user would still need to install Galaxy manually and plug it into a Kubernetes cluster. To make the integration process easier for developers and bioinformaticians, we improved the *docker-galaxy-stable* community Docker images (Grüning, Chilton, *et al*., 2018), enabling the deployment of Galaxy through these composable containers on Kubernetes. This includes automated routines for runtime setup (using the Galaxy API (Sloggett *et al*., 2013)) of users and workflows, installation of needed packages and additional testing of the container images, among others. Reusing the Galaxy community images reduces the burden of maintainability in the long run and improved the setup with many of the features that the community images have (e.g. web scalability, *sftp* access, etc). This standardisation permits the deployment of Galaxy instances that are complete replicates of each other on different cloud providers, maximizing interoperability and reproducibility.

Deployment of cloud-based solutions is a complex task that demands knowledge of both architecture (e.g. AWS, Google Cloud) and bioinformatics tools. We packaged the Galaxy setup for Kubernetes through Helm charts (helm.sh), which simplifies deployment on any Kubernetes cluster. Currently, the Galaxy Helm chart gives the administrator complete control of most of the configuration variables of Galaxy. This permits setting up different deployments (setups for different clouds, different Galaxy flavours/versions, testing environment, *etc*) as configuration files that can be version controlled (see examples, in the supplementary materials).

3 DISCUSSION AND FUTURE DIRECTIONS

We have implemented an approach to exploit encapsulation, packaging and container orchestration for Galaxy workflows, that simplifies cloud portability. This has been used extensively in the context of the PhenoMeNal H2020 Project (Peters *et al*., 2018), in production at https://public.phenomenal-h2020.eu/ for the past three years (handling tens of thousands of jobs in the public instance). However, the set-up and methods are generic and can be easily applied to other omics, as done recently for instance on Single Cell RNA-Seq. Documentation linked on the supplementary materials shows how to use it.

ACKNOWLEDGEMENTS

PM, LP, PRM, SH, AL, DS, CR, DJ, PRS, RW, RMS, NK, OS, and SN were supported by the European Commission's Horizon 2020 program grant 65424 "PhenoMeNal", and acknowledge the leadership of Christoph Steinbeck;

CONTRIBUTIONS

Design and envisioned Galaxy-Kubernetes integration: PM; Code contributions Galaxy Kubernetes Runner: PM, LP, PR, MVB; Write Galaxy Helm Chart: PM, LP, NG, EA, SH, AL; Contribute to docker-galaxy stable: BG, PM, LP, SN; Manuscript writing: PM, LP, PRS; Galaxy Kubernetes testing: PM, LP, PR, CR, DS, DJ, PRS, RW, RM, YP, SN; Integration Kubenow: AL; Work package lead: PRS, OS, SN; Galaxy expertise guidance: BG; Project Manager Organize hackathons.: NK; Biocontainer interface: YP; Galaxy-Kubernetes Single Cell: PM, IP

REFERENCES

Afgan,E. *et al*. (2012) CloudMan as a platform for tool, data, and analysis distribution. *BMC Bioinformatics*, **13**, 315.

Afgan,E. *et al*. (2018) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res*., **46**, W537–W544.

Agapito,G. *et al*. (2017) Parallel and Cloud-Based Analysis of Omics Data: Modelling and Simulation in Medicine. In, *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*., pp. 519–526.

Grüning,B., Chilton,J., *et al*. (2018) bgruening/docker-galaxy-stable: Galaxy Docker Image 18.05.

Grüning,B., Dale,R., *et al*. (2018) Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat. Methods*, **15**, 475–476.

Langmead,B. and Nellore,A. (2018) Cloud computing for genomic data analysis and collaboration. *Nat. Rev. Genet*., **19**, 208–219.

Peters,K. *et al*. (2018) PhenoMeNal: Processing and analysis of Metabolomics data in the Cloud. *bioRxiv*, 409151.

Sefraoui,O. *et al*. (2012) OpenStack: toward an open-source solution for cloud computing. *Int. J. Comput. Appl. Technol.*, **55**, 38–42.

Sloggett,C. *et al*. (2013) BioBlend: automating pipeline analyses within Galaxy and CloudMan. *Bioinformatics*, **29**, 1685–1686.

da Veiga Leprevost,F. *et al*. (2017) BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics*, **33**, 2580–2582.