From laptop to super-computer: standardizing installation and management of Galaxy

Nuwan Goonasekera\textsuperscript{1}, Enis Afgan\textsuperscript{2}, Vahid Jalili\textsuperscript{3}

\textsuperscript{1}University of Melbourne, Melbourne, Australia, ngoonasekera@unimelb.edu.au
\textsuperscript{2}Johns Hopkins University, Baltimore, MD, USA
\textsuperscript{3}Oregon Health and Science University, Portland, OR, USA

Project Website: https://github.com/galaxyproject/cloudman/tree/v2.0
Source Code: https://github.com/galaxyproject/cloudman/tree/v2.0
License: MIT

The large number of public Galaxy servers (over 100) and many more private servers, is a testament to Galaxy’s broad acceptance and sheer scale. Galaxy solves a complex problem and requires the maintenance of a complex ecosystem of software, including Galaxy itself, a robust database, a job manager, an ftp server and reference data, among other things. In addition, such servers also need user authentication and access control, monitoring and management, regular updates, and a host of other common tasks—often laborious in nature—and repetitive in effort across the community.

Apart from this duplication of effort, there is also a proliferation of different, often out-of-date versions of Galaxy, as administrators are reluctant to upgrade their working installations to avoid down-time and risk. Having a dedicated system administrator to manage an installation is also beyond the budget of smaller labs and teams, leading to poorly maintained Galaxy instances.

This suggests that a standardized, easy-to-use method of installing a production-grade Galaxy instance, and a means of performing common maintenance and configuration tasks on it, is a necessity. The community has made many efforts in this direction, including Ansible playbooks, Docker containers, virtual machines, CloudMan, and a host of similar efforts which address various aspects of the problem, but do not yet consist of an integrated, comprehensive solution.

In this talk, we highlight our efforts to leverage many of these existing solutions into an integrated system that makes the installation, life-cycle management and scaling of Galaxy uniform across a variety of platforms (e.g., workstations, clusters, clouds). We accomplish this by leveraging existing Ansible playbooks and Docker containers and integrate them with Kubernetes Helm charts, CloudLaunch, and CloudMan. This will allow changes with a few clicks through a GUI, as well as via a command line client or an API. Gradually, this Galaxy management system will lead to a mostly “hands-off” approach to running Galaxy, with support for rolling updates and zero-downtime configuration changes.