SAMCloud: Simplifying research driven cloud deployments for education and more

Geoffrey Squire[1], Louis Moresi[2], Romain Beucher[3], Julian Giordani[3], Stuart Woodman[4], Carsten Friedrich[1], Rob Woodcock[5]

[1]CSIRO Data61, Canberra, Australia, geoffrey.squire@data61.csiro.au, carsten.friedrich@data61.csiro.au
[2]Australian National University, Canberra, Australia, louis.moresi@unimelb.edu.au
[3]University of Melbourne, Parkville, Australia, romain.beucher@unimelb.edu.au, julian.giordani@unimelb.edu.au
[4]CSIRO Data61, Melbourne, Australia, stuart.woodman@data61.csiro.au
[5]CSIRO Mineral Resources, Perth, Australia, robert.woodcock@csiro.au

INTRODUCTION

There is continued high demand for geoscience analytics research and the software and infrastructure to support it. Researchers and organisations creating the algorithms and products share a common need to foster the publication, sharing, use and citation of their research and outcomes. A collaboration between the AuScope GRID team at CSIRO and the Underworld team at the University of Melbourne resulted in a prototype system that can simplify and automate the configuration and deployment of research software services and applications in the cloud. The scenario chosen as the focus for the work was the deployment of a service that provisions individual resources for the participants in a small group workshop or tutorial. Initially using the documentation and teaching material developed for the Underworld modelling software, the system provides a simple web interface where researchers can select the software "bundle" to be deployed, configure it according to their needs, select the compute and storage resources required, specify the lifetime for the deployment and manage it once it has been deployed.

REQUIREMENTS

For the workshop scenario the research team came up with the following requirements to drive the initial development.

1. Interface and software environment easily customizable by the instructor
2. Identical software stack for every workshop participant
3. Environment that can be taken away and reused by the student after the course
4. Robust authentication and isolation for users
5. Scalability from a few students in a class to thousands of people remotely
6. Cross-platform and cross-device access by users
7. Straightforward for the instructor to set up a workshop

Many of the requirements could be satisfied by choosing appropriate existing software and systems to use as components in the samcloud stack, so most of the work in this project was around developing new systems to manage the stack and simplify usage for researchers looking to deploy and publish their software and workshops.

THE SOLUTION

The Underworld[1] code was developed for modelling plate-scale fluid mechanics and studying problems in lithosphere dynamics. Though specialized for this task, underworld has a straightforward python user interface that allows it to run within the environment of Jupyter[2] notebooks on a laptop (at modest resolution). The python interface was developed for adaptability in addressing new research problems, but also lends itself to integration into a python-driven learning environment. To manage the heavy demands of installing and running underworld in a teaching laboratory the

[1] http://www.underworldcode.org/
[2] http://jupyter.org

Underworld team developed a workflow that virtualizes the underworld environment as a Docker[3] container, providing the customizable and replicable software environment for workshop participants.

This combination satisfies the first three requirements above. JupyterHub[4] was used to manage users and their individual notebook instances. This provides the authentication and authorization layer for the workshop or tutorial, and makes the process of managing isolated environments transparent to the end user. Each user gets their own jupyter notebook instance and filesystem space to store their changes, results and any other persistent data. Authentication of the workshop participants can be done using an OAuth provider such as Google or GitHub, or for Australian researchers or students the Australian Access Federation (AAF)[5] can be used.

The team at CSIRO developed the system that configures, provisions and manages the workshop environments, combining the technologies described above with the low level infrastructure required to host the workshop in the cloud. Running the software in the cloud, and exposing it via a web interface, enables both scalability and cross-platform remote access. Workshop services and the participants' individual Jupyter instances are deployed into a Kubernetes[6] cluster running in the Amazon cloud[7], and a public URL is provisioned for each deployment. This approach allows for a high level of scalability in response to demand, while minimising resource use under low load and keeping it within the configured resource budget for each workshop.

Researchers running a workshop can select the software environment they wish to deploy (e.g. a JupyterHub service), configure it for their use (e.g. supply a Docker image with the appropriate Jupyter environment) and specify the compute and storage requirements per workshop participant. They can monitor their deployments, and update or delete them as required, including pushing out new versions of their software, documentation and other data to a running deployment.

THE VISION

While this prototype was used for running workshops based on a Jupyter notebook environment with the Underworld code and associated documentation and tutorials, the system as developed is far more general. It can be used to deploy long-running services or applications into the cloud from a curated catalogue. Each option includes a description of the configuration options and environments so that the system can provide the appropriate interface and automation for the researcher setting up and deploying instances. Ongoing work is exploring how it can be applied to other research software and for different use-cases, including running distributed workloads and providing access to large-scale data.

ACKNOWLEDGMENTS

This work was supported by CSIRO and AuScope/NCRIS open source software, SISS and VGL.

[3] http://docker.io
[4] https://jupyterhub.readthedocs.io
[5] https://aaf.edu.au/
[6] https://kubernetes.io/
[7] https://aws.amazon.com/