Using Containers and Tapis to Structure Portable, Composable and Reproducible Climate Science Workflows

Michael Dodge II
mdodge@hawaii.edu
University of Hawaii at Manoa
Honolulu, Hawaii, USA

Sean Cleveland
University of Hawaii - System
USA
seanbc@hawaii.edu

Gwen A. Jacobs
University of Hawaii - System
Honolulu, HI, USA
gwenjh@hawaii.edu

ABSTRACT

Provenance and Reproducibility have been growing needs in scientific computing workflows. This project seeks to split the traditionally monolithic code-base of a climate data computing workflow into small, functional, and semi-independent containers. Each container image is built from public code repositories, and allows a researcher to determine the exact process that was executed for both technical and scientific validation. These containers are composed into their workflows using the Tapis API's Actor-Based Container (Abaco) system, which can be hosted on a variety of computing infrastructures. They may also be run as standalone containers on computers or virtual machines with Docker installed.

CCS CONCEPTS

• **Information systems** → **Computing platforms**; • **Computer systems organization** → *Cloud computing*; • **Applied computing** → *Environmental sciences.*

KEYWORDS

Cloud Computing, High Performance Computing, Cyberinfrastructure Tapis

ACM Reference Format:
Michael Dodge II, Sean Cleveland, and Gwen A. Jacobs. 2022. Using Containers and Tapis to Structure Portable, Composable and Reproducible Climate Science Workflows. In *Practice and Experience in Advanced Research Computing (PEARC '22), July 10–14, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3491418.3535126

1 PORTABILITY

As needed, the containers may be run on virtually any system with Docker, an internet connection, and sufficient specs to run the contained software. They are designed to be executed on any Tapis[2] v3 tenant with Abaco[1] capabilities, and equipped with libraries for communicating with both v3 and v2 systems. Layers of the container images are distributed via Docker Hub. The project includes tooling to allow for the implementation of other workflows, using modular python scripts to handle the majority of operations. These operational scripts are designed to be executed with arguments from a researcher-defined shell script, keeping user modifications as narrow as possible.

2 COMPOSABILITY

Each science workflow is built in layers, with scripts built separately from their respective dependencies. Code libraries for the workflows are pulled into base layers, then the workflow scripts themselves are downloaded and placed into the ready-to-run layer. The scripts and their download locations are read from a simple JSON list, allowing researchers to quickly and easily add or remove scripts from the builds. The thin final layer allows for prototyping in seconds. An additional advantage of layering is the minimization of final image size between different workflows. For example, Workflow A may not need the gigabyte of libraries that Workflow B uses, and thus those libraries can be kept exclusive to Workflow B, saving on bandwidth and storage costs. When running in the form of an Actor on the Abaco system, the containers may be arranged to launch other containers in series. This can be achieved by adding to the first container's launch parameters or by using the native "link" feature in Abaco, depending on the use case. An automated container build system was implemented to establish continuous integration for the climate science workflows. This build system uses webhooks to connect GitHub to the Tapis APIs, leveraging on-demand resources to reconstruct containers upon code being committed to the relevant repositories. Developers may provide "hints" to the APIs stating the cutoff time for execution, allowing builds to be optimally scheduled in a shared system.

3 REPRODUCIBILITY

Each successful build results in a Docker image, which can be saved and executed to verify results. Builds are tagged with the hashes of the repositories that they are built upon, allowing users to determine what code was used without having to launch the container. Jupyter notebooks are used to orchestrate workflows, with tooling provided for reviewing past workflows and re-running them as needed.

4 USE CASE

Two climate science workflows, one involving rainfall and one involving air temperature, make up the initial use case of this project. Both workflows operate in stages involving data acquisition and aggregation (downloading and parsing), followed by quality control and gap-filling, and finally a map-generating stage. The climate science scripts were developed in a VM with heavy use of environment variables for portability, then built into their respective top-level container layers. Data acquisition is handled by a single container, as both workflows use the same daily data set. Workflows

PEARC '22, July 10–14, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9161-0/22/07.
https://doi.org/10.1145/3491418.3535126

then diverge at aggregation, with the rainfall containers running a series of R scripts, and air temperature containers running a series of Python scripts. The data acquisition and workflow steps are all orchestrated in CRON, first in a development VM for testing, then onto Abaco for production usage.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation grants - Tapis Framework:[1931439 and 1931575], Collaborative Research: SS2-SSI: The Agave Platform: An Open Science-As-A-Service Cloud Platform for Reproducible Science NSF OAC #145041. #1931439 and #1931575, and RII Track 1: ʻIke Wai Securing Hawaiʻi's Water Future NSF OIA #1557349.

REFERENCES

[1] Joe Stubbs et al. 2018. Rapid Development of Scalable, Distributed Computation with Abaco. Science Gateways Community Institute, 10th International Workshop on Science Gateways.

[2] Joe Stubbs, Richard Cardone, Mike Packard, Anagha Jamthe, Smruti Padhy, Steve Terry, Julia Looney, Joseph Meiring, Steve Black, Maytal Dahan, Sean Cleveland, and Gwen Jacobs. 2020. Tapis: An API Platform for Reproducible, Distributed Computational Research. (2020). submitted.