Developing & Maintaining Open Source Scientific Cloud Computing in AWS with HelioCloud

Nicholas Lenzi

Software Engineer

Johns Hopkins University Applied Physics Laboratory

Nicholas Lenzi, Peter Shumate, Alex Antunes, Jon Vandegriff, Chris Jeschke, Brian Thomas, Sarah Rourke, Jeffery Bradford

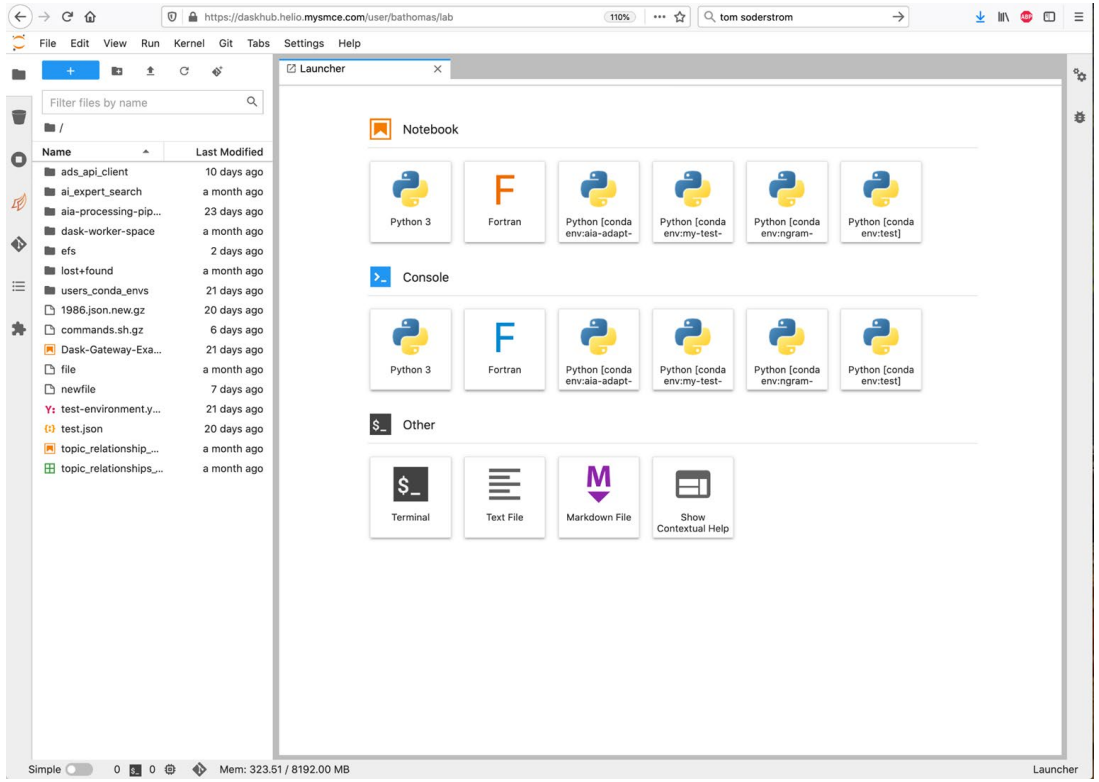
Outline

- HelioCloud Platform
- Engineering Challenges
- Continuous Integration/Continuous Delivery
- Questions

HelioCloud platform: Easy Cloud Onramp for Science

Allows for both Jupyter notebook and terminal-based work

- **Scientific computing focus**
  - Python, IDL, Fortran, C/C++ & libraries
  - AI/ML & GPU tooling
- **Bursting**
  - Create programs which stand up powerful virtual machines (GPU, loads of RAM, etc)
- **Default software install (“Just works”)**
  - Conda (Python) environment for Heliophysics / ML
  - Other research packages
- **Tutorials**
  - How to do a number of basic things
- **Access to large heliophysics datasets**
  - ~8 Pb by 2025
  - Mission, 3rd Party/User Published, “ML Ready”, EPO/Hackathon/Citizen Sci. Datasets
- **Has 125+ science users** so far
  - also 80+ simultaneous student users at PyHC Summer School

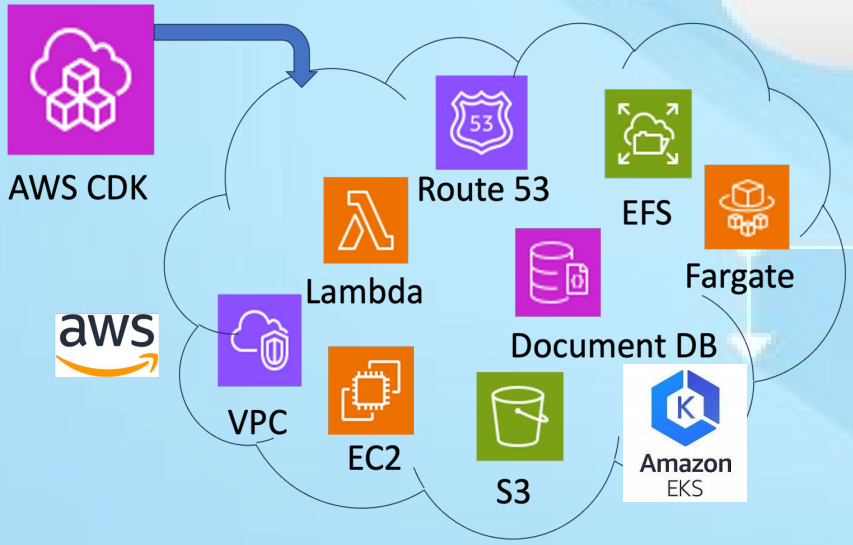


Technology

Architecture

Terraform

Infrastructure as code (IAC) is used to rapidly deploy a fully functional stack atop AWS using one of **Terraform** or **AWS CDK.** This completely automates deployment.



Configurations of perminant installations are stored in git and deployed using the **gitops** model. **Environment specific** configurations are stored in separate **layers** and connected using various **templating tools** allowing **multiple HelioCloud deployments** to be managed within a single repository.

Services

jupyter

dask

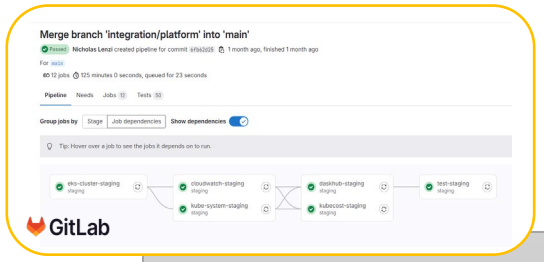
Both platforms prepare a full featured **Jupyter notebook environment** ready for multiple users. **Dask** is included for **parallel computing** - so you can instantly scale up your resources,

User portals let end users spin up their own **customized virtual machines.**

Engineering Challenges

- Software Upgrades of Infrastructure Software - Keeping up with upgrades
  - Kubernetes releases a new minor version every four months, with patch releases monthly.
  - Amazon typically publishes a new AMIs for EKS Managed Node Groups every two weeks; EKS addons once a month
  - Amazon charges more (6 times more) for EKS clusters running older versions of Kubernetes
    - Standard support (approx. $72 per month; $864 per year)
    - Extended support (approx. $432 per month; $5,184 per year)
  - Daskhub typically releases once per year; JupyterHub and Dask are released as a higher frequency
- Making the latest science libraries available to users
  - Interdependency of the various python libraries makes it difficult to upgrade a single library
  - Difficult to tell if existing Science Notebooks “work” with a different runtime
- Multi-institutional
  - A common institution agnostic, base platform
  - To best suit their needs, each institution can choose to tailor the various services

A Multi-repo approach to Configuration Management



HelioCloud Team publishes features with **branching model** with standalone CI/CD pipelines

HelioCloud Platform

GitLab

git

Develop Branch

Feature A

Feature B

...

Kustomize

Jinja

GitLab

On merge to **Develop** branch, **automated pipelines** apply latest changes to **git repo of private institution**

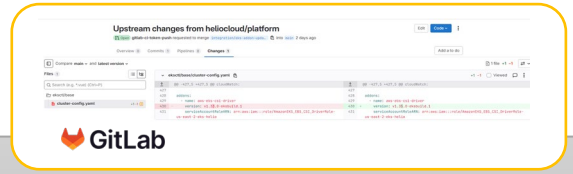
Private HelioCloud Confs

GitLab

git

gitops

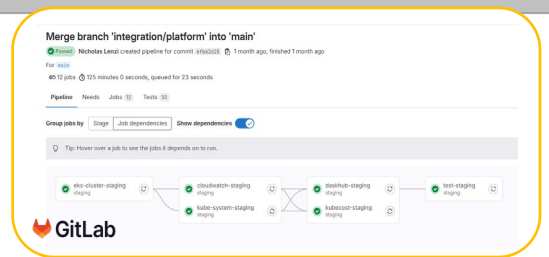
Main Branch



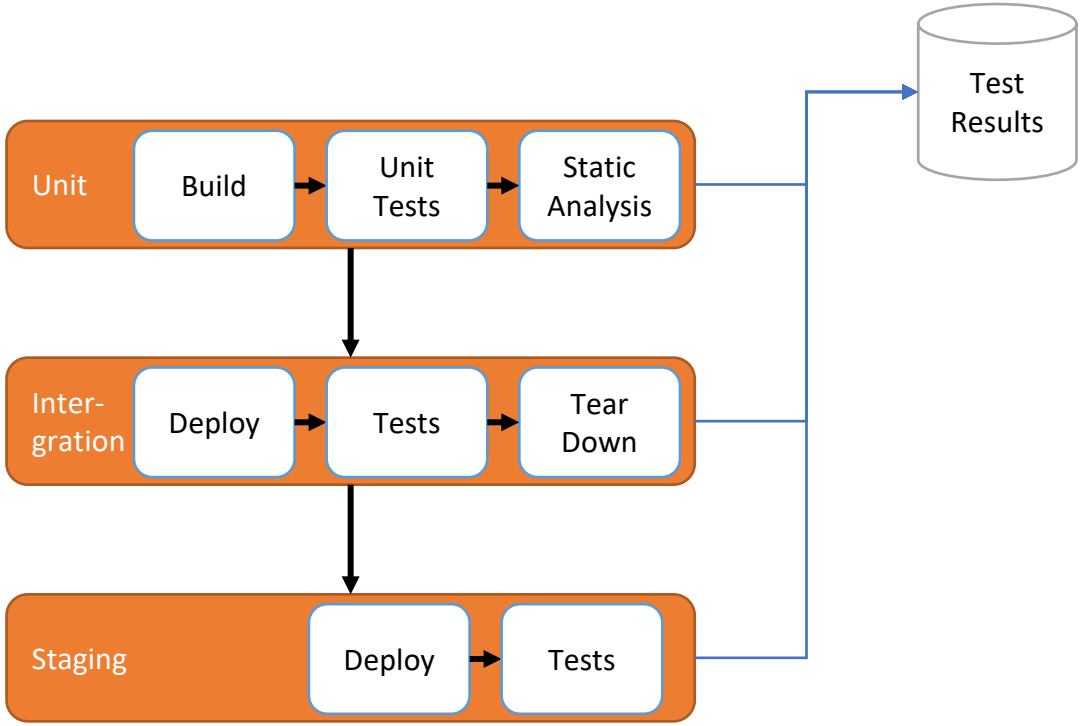
Private institution uses a **Merge Request** process to approve or deny upstream changes

Feature A/Feature B

Private institution apply changes automatically on **merge to main branch**



Continuous Integration Pipeline for the HelioCloud Platform



pytest · Pylint STRICT · Black · Mocking with pytest-mock

aws · aws Cloud Development Kit · Boto 3 · pytest · amazon S3

aws · aws Cloud Development Kit · eksctl · Selenium WebDriver · behave

Questions?

Developing & Maintaining Open Source Scientific Cloud Computing in AWS with HelioCloud

Nicholas Lenzi

Nicholas.Lenzi@jhuapl.edu

HelioCloud Project (https://heliocloud.org)