CATALYST 3: The Future of CERES Earth Radiation Budget Climate Data Production Using Linux Containers

IN21E-0888

Thomas.N.Hillyer@nasa.gov, Science Systems and Applications, Inc.
Jonathan.L.Gleason@nasa.gov, Kathleen.Dejwakh@nasa.gov, NASA Langley Research Center

1. CERES Background

Clouds and the Earth's Radiant Energy System (CERES) is a NASA broad-band radiometer with six flight models flying on the EOS Terra and Aqua missions, the Suomi National Polar-orbiting Partnership (NPP) platform, and the NOAA-20 platform. The CERES science team integrates data collected by the six CERES instruments with nine additional space-based instruments, including the Moderate Resolution Imaging Spectroradiometer (MODIS) and the Visible Infrared Imager Radiometer Suite (VIIRS) instruments, to produce Climate Data Records of the Earth's radiation budget. The CERES production system consists of over 42 Product Generation Executives (PGEs) maintained by twelve subsystem groups. The processing chain fuses space-based measurements with data from eight other unique sources.

Abstract

The CERES Science Team integrates and fuses observations from 6 CERES instruments aboard the Terra, Aqua, S-NPP, and NOAA-20 missions with data from more than 20 other unique data sources. Following the November 2017 launch of CERES Flight Model 6 (FM6) onboard NOAA-20, CERES has now amassed over 80 instrument-years of valuable Earth radiation budget data. The rapidly growing volume of CERES data coupled with the introduction of new data products alongside improvements to existing science algorithms fosters the requirement for faster, more flexible, and scalable data production and orchestration. New virtualized, cloud-centric compute hardware hosted by the NASA Langley Research Center's (LaRC) Atmospheric Sciences Data Center (ASDC) provides an ideal environment for these ever-increasing data production demands for CERES.

This poster discusses updates to the implementation of the CERES Data Management Team's (DMT) CERES AuTomAted job Loading sYSTem (CATALYST), a custom data processing workflow engine for CERES, to use on-demand computing resources to perform automated CERES data production processing in a Linux-based container environment. Linux containers provide CERES the flexibility to build multiple production environments in containers tailored for specific workloads and allow effortless provisioning of resources based on the CERES Science Team's data production requirements.

2. CATALYST Background

CATALYST is a production automation system for CERES which runs at the NASA Langley ASDC. CATALYST began production in 2014 managing a small subset of eight PGEs for CERES Terra/Aqua Clouds and Inversion processing. Today, the system manages all 42 (soon to be 47) CERES production PGEs ranging from Level-0 processing through CERES's monthly gridded products for all six CERES flight models.

CATALYST consists of two major components: the server and the Operator's Console. The CATALYST server executes CERES PGEs in a Univa (formerly Sun) Grid Engine environment according to rules assigned by plug-in modules for each CERES PGE. Execution requests for a time range are provided to CATALYST as Production Requests (PRs) from the CERES Production Request website. CATALYST delivers the processed data to the ASDC for archival and public distribution.

CATALYST provides a cross-platform Java-based Operator's Console for monitoring production activity. This utility provides methods for viewing a CERES job's execution workflow, log files, and processing cluster status information.

3. What are Containers?

A container is a lightweight form of virtualization used to package and execute software. Docker is a popular standard used for containerization. Features of a container image include:

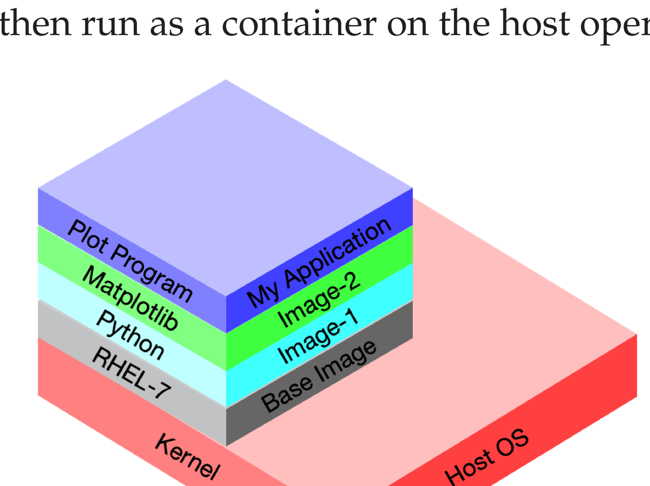
- **Efficiency** - containers share the host machine's Operation System (OS) kernel, therefore do not require an entire OS instance per application
- **Portability** - containers bundle all of an applications dependencies
- **Security** - containers are isolated from one another
- **Scalability** - containers can be started just as quickly either as a single instance on the developer's workstation or hundreds (or thousands) of instances in a production environment

4. Container Image Layers

A Docker container image is usually comprised of several layers. Each container image layer references the image below it to build up to an application. Layers can be reused to create new application stacks leveraging existing components. For example, in the diagram below:

1. the plotting program (My Application) requires the
2. Matplotlib graphics library installed for the
3. Python interpreter that is installed on
4. the Red Hat Enterprise Linux (RHEL-7) base image

This image can then run as a container on the host operating system.



Additional Information

https://ceres.larc.nasa.gov

https://science.larc.nasa.gov

5. Container Orchestration

OpenShift is a Platform as a Service (PaaS) system from Red Hat that provides powerful container cluster management and orchestration. OpenShift leverages **Kubernetes**, a system used for management and scaling of containerized applications, and provides:

- a built-in container registry
- a self-service web-based portal and catalog
- development following a DevOps methodology
- deployment automation with Jenkins automation

6. CERES PGE Containers

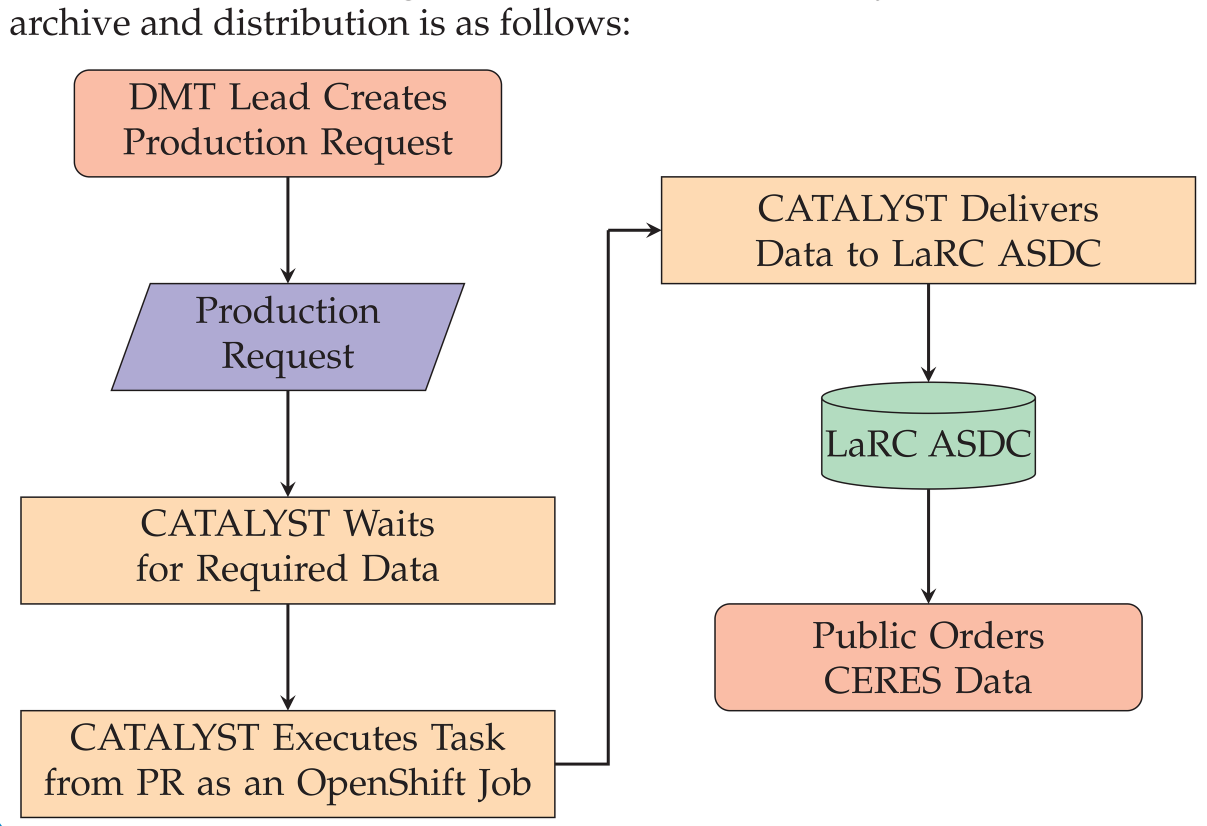
CERES PGE containers can be used in either the development environment or the production environment. Typical CERES PGE container stacks are as follows:

PGE: #1	PGE: #2
CERES Libraries	CERES Libraries
HDF4, HDF5, NetCDF-4, etc.	HDF4, HDF5, NetCDF-4, etc.
compilers: gcc, gfortran, etc.	compilers: gcc, gfortran, etc.
rhel7	rhel7

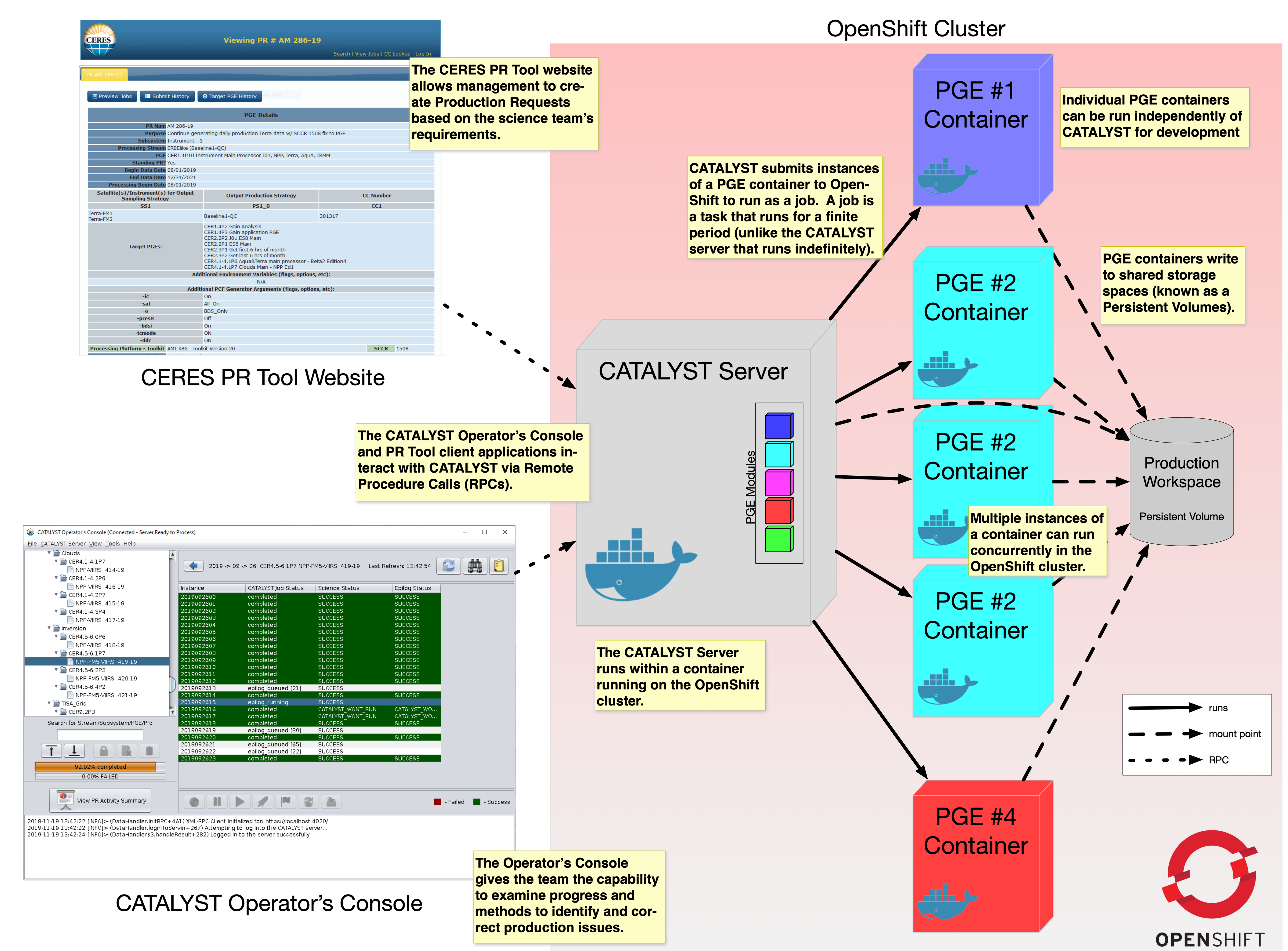
Many layers of the container are shared across the board for CERES PGEs. This limits how much needs to be rebuilt for changes to a PGE, or the creation of an entirely new PGE.

7. Production Request Workflow

The standard workflow for receiving a PR from the CERES Science Team, execution through CATALYST, and delivery to the ASDC for archive and distribution is as follows:



8. CATALYST v3.0 System Design



9. Benefits for CERES Development

- **Exact Replica of the Production Environment** - Software developers can now develop and test in an environment configured exactly like the production environment
- **Easier to Run Large-Scale Tests** - Science developers can easily large scale tests
- **Environment Isolation** - Using the self-service model, developers can more quickly make adjustments to the development environment without impacting other developers

10. Benefits for CERES Production

- **Better Resource Utilization** - Can better utilize available system resources for burst workloads and yield resources for idle periods
- **Easier to Upgrade** - Software/library upgrades can be incrementally applied to CERES PGEs (no longer have to test and validate library updates against all PGEs prior to production)
- **Easier to Snapshot the Entire Environment** - The PGE container is a self-contained unit with all dependencies bundled