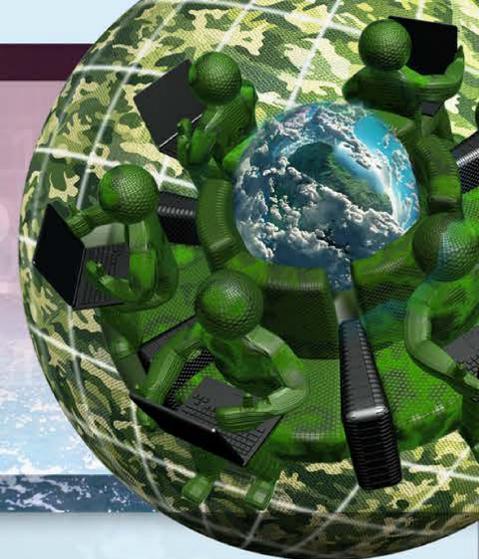


# Cooperation with Google for Deploying ESGF Node on Google Cloud Platform

Serguei Nikonov<sup>1,2</sup>, Hans Vahlenkamp<sup>1,3</sup>, Aparna Radhakrishnan<sup>1,4</sup>, Karan Bhatia<sup>5</sup>, V.Balaji<sup>1,2</sup>, Luca Cinquini<sup>6</sup>, Ryan Abernathy<sup>7</sup>

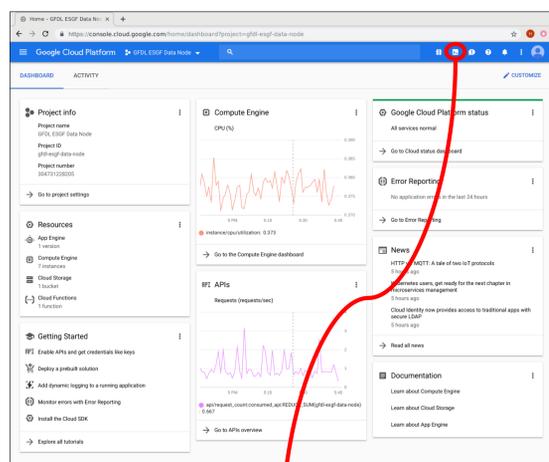
<sup>1</sup>NOAA GFDL, <sup>2</sup>Princeton University, <sup>3</sup>UCAR, <sup>4</sup>Engility, <sup>5</sup>Google, <sup>6</sup>NASA JPL, <sup>7</sup>Columbia University



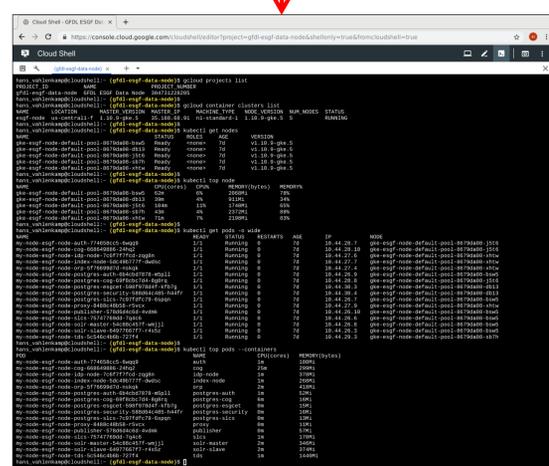
## 1. Introduction

Prototyping ESGF on Google Cloud Platform (GCP) is being conducted under the Distributed Resources for the Earth System Grid Advanced Management (DREAM) project. The DREAM team has worked at converting the current ESGF software stack into a system of interacting Docker containers, which can be deployed and scaled on a generic cluster of distributed nodes.

## 5. GCP Management Interface



Dashboard GUI



Cloud Shell Command-Line

## 2. Goals

The main goal was investigating the feasibility of deploying and running an ESGF node on GCP. This goal included the following stages:

- Preparation of containerized version of Earth System Grid Federation (ESGF) software suite, ESGF/Docker. The ESGF Container Working Team chose Kubernetes Docker Containers among different container technologies.
- Testing and deploying ESGF Kubernetes Containers onto GCP under the auspices of Princeton University.
- Investigating most suitable persistent data storage option. Google Cloud Storage buckets and Cloud Filestore options were setup and tested.
- Publishing representative set of variables on GCP ESGF. CMIP6 data from GFDL was chosen for this purpose. Test publishing was done successfully.

## 3. Published Datasets

The datasets currently published are variables from atmos components of GFDL CM4 model, CMIP6 piControl experiment. Total data volume is not very large (< 100GB), but enough for making a preliminary assessment of performance and robustness. Available access protocols (HTTP, OPeNDAP) were tested.

HTTP download performance was comparable with traditional ESGF nodes as observed by random sampling:

- GCP ESGF Node = 18MB/s
- DKRZ ESGF Node = 35MB/s
- NCI ESGF Node = 20MB/s
- IPSL ESGF Node = 14MB/s

## 4. Lessons Learned

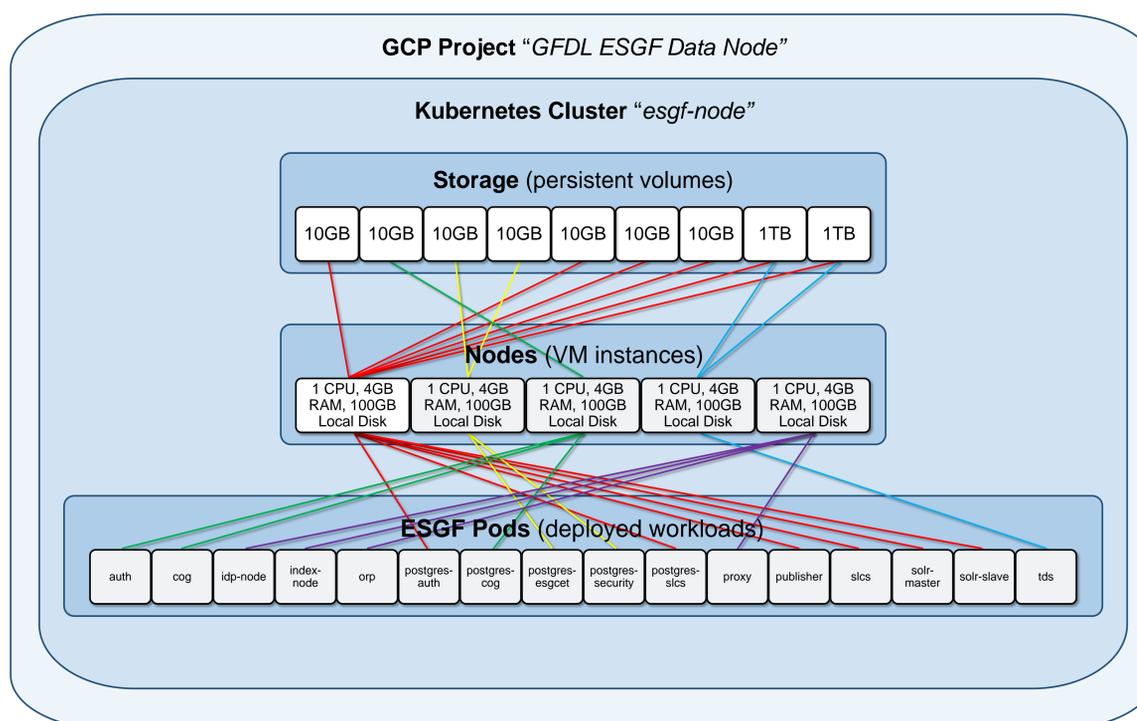
Different GCP data storage options result in different behavior. The Google Cloud Storage buckets option was tested first. A serious problem was observed on multiple occasions where published datasets became unexpectedly inaccessible. Investigation found that the underlying filesystem mounts were actually dropping offline and this might be triggered by higher access loads. There is no solution to this problem currently. However, the Cloud Filestore option was then tested and no problems have been found with it currently.

Newer Tomcat versions apparently have more strict URI encoding rules. This causes failures when accessing published datasets from the THREDDS server via OPeNDAP. One temporary workaround is to use an older Tomcat version, but better solutions are being investigated.

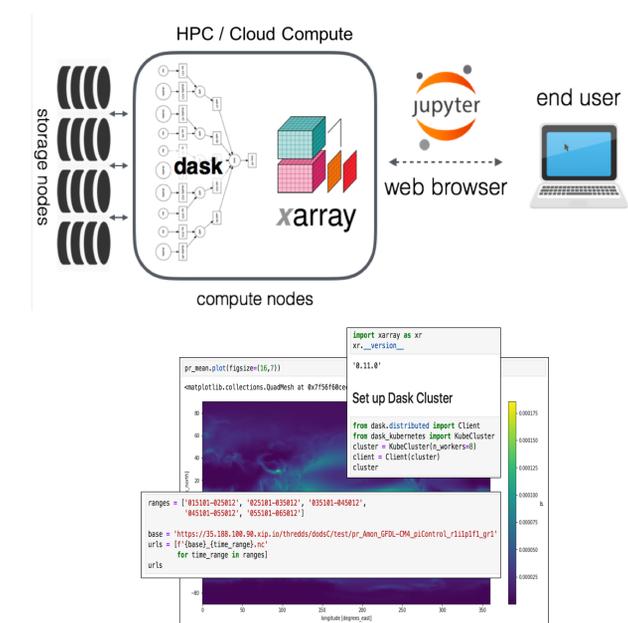
## 7. Future

Continuing experimentation will focus on the practical application of the deployed GCP ESGF Node for data analysis using the Pangeo platform. Most popular algorithms will be implemented via the Jupyter interactivity layer for performance scalability testing.

## 6. GCP ESGF Architecture Example



Pangeo architecture and JupyterHub example:



## 8. References

- ESGF Docker - Google Kubernetes Engine: <https://esgf.github.io/esgf-docker/kubernetes/google/>
- Google Cloud Platform: <https://cloud.google.com/docs/>
- Pangeo: <https://pangeo.io/architecture.html#architecture>, <https://pangeo.pydata.org>

Feedback? Please email us at [oar.gfdl.dpteam@noaa.gov](mailto:oar.gfdl.dpteam@noaa.gov)



Other outstanding Issues to address include:

- GridFTP protocol availability
- Any additional storage options
- Node naming
- Joining ESGF (test?) Federation
- Long-term costs

Results from this experimentation will ultimately help to determine whether ESGF Nodes deployed on GCP can be an alternative to local deployment.