Workshop Report: Container Based Analysis Environments for Research Data Access and Computing

August 14-16, 2017
National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign

Workshop supported by the Data Exploration Lab[1] and the National Data Service[2] (NDS).

[1] https://dxl.ncsa.illinois.edu
[2] http://www.nationaldataservice.org/

Participants

Greg Bauer
National Center for Supercomputing Applications

Max Burnette
National Center for Supercomputing Applications

Matias Carrasco Kind
National Center for Supercomputing Applications

John Fonner
Texas Advanced Computing Center

Roland Haas
National Center for Supercomputing Applications

Eliu Huerta Escudero
National Center for Supercomputing Applications

Jai Won Kim
Johns Hopkins University

Kacper Kowalik
National Center for Supercomputing Applications

Gerard Lemson
Johns Hopkins University

Ian McEwen
CyVerse.org

David LeBauer
National Center for Supercomputing Applications

Yan Liu
National Center for Supercomputing Applications

Jian Tao
Texas A&M University

Jeff Terstriep
National Center for Supercomputing Applications

Matt Turk
National Center for Supercomputing Applications

Craig Willis
National Center for Supercomputing Applications

Andrea Zonca
San Diego Supercomputer Center

Executive Summary

There are several new trends toward reducing barriers to accessing research data and research computing resources to accelerate scientific discovery. Traditional scientific gateway services are being combined with remote interactive analysis environments to facilitate further collaboration and sharing between users. Tools such as Jupyter and RStudio are increasingly becoming first-class applications for remote access and analysis of research data [2,4,5,6,7,8] as well as access to high-performance computing (HPC environments) [3,4,5], offering alternatives to conventional terminal-based access or data transfer services. At the same time, container-based technologies are emerging as a solution to a variety of problems in research computing and research data access.

Container frameworks, such as Docker and Singularity, support packaging of research software stacks for distribution that can easily run in a variety of environments: from laptop to cloud and HPC. In HPC systems, container frameworks such as Singularity and Shifter reduce time required for installing and managing complex software stacks and potentially extend the viable lifetime of compute clusters [15, 16]. In cloud computing environments, container technology is increasingly seen as an alternative to traditional hypervisor-based systems [14]. Open-source frameworks such as Docker Swarm and Kubernetes support the deployment, scaling and management of containerized applications, providing many tools to simplify common IT tasks. This has given rise to a number of services enabling access to research data and research computing infrastructure via containerize applications.

The Container Analysis Environments workshop was held August 14-16, 2017, at the National Center for Supercomputing Applications (NCSA) on the University of Illinois campus. The workshop began with a series of presentations from participants describing science drivers and why container technology has been selected to solve research infrastructure problems. All presentations were followed by Q&A. The second and third day focused on discussion topics selected by participants, including tutorials and demonstrations of existing systems.

The workshop emphasized multiple distinct use cases including remote interactive access to research data and computing resources; remote collaboration; packaging, sharing, and preservation of research software; training and education; science gateways applications; as well as IT infrastructure management.

The workshop also emphasized challenges and opportunities including security; trusted and untrusted software; shared storage and data management including permissions; single sign-on between services as well as authorization and group management; and technology solutions for container management and orchestration.

Workshop Context and Objectives

The goal of the workshop was to bring together groups leveraging container-based technologies for research data access and research computing to share information, identify opportunities for interoperability, and address common challenges. The workshop centered on several themes, focusing on the science cases and how technical solutions are addressing the needs of researchers. Themes included:

- Why researchers are using container-based environments (e.g., consistent environments, packaging applications, shifting computation around)
- Interactive vs non-interactive use cases including approaches to workflow management, orchestration, scalability, and moving analysis to HPC environments
- Data/storage focusing on models for managing and exposing data to containers
- Security/permissions/licensing

Workshop Participants

Participants came from a variety of backgrounds and projects including supercomputing centers, gateway development, as well as a number NSF funded projects across multiple domains. A complete list of participants is included in Appendix A.

Program

The first day consisted of participant presentations with Q&A and discussion. Topics were selected based on participant interest for breakout and deep-dive discussions on subsequent days. The complete program is included in Appendix B.

Workshop Presentations and Demonstrations

Presentation Summaries

Each participant was asked to prepare a 20 minute presentation about the use of container-based technologies in their projects as well as any open issues or challenges. The presentations are available on Github[3].

Matias Carrasco Kind presented on containerized solutions for the

[3] https://github.com/nds-org/container-analysis-workshop

Dark Energy Survey (DES) and Large Synoptic Survey Telescope (LSST) projects. The DES telescope produces 1-3 TB of data per night, which is transferred to NCSA. Processing is done on image to produce astronomical catalog data using FermiGrid, the UIUC Campus Cluster, and Blue Waters. The DESLabs system is a collection of containerized tools used to provide access to the DES data to collaborators. This includes custom applications such as the Cutout Service and easyaccess Web Client as well as an instance of JupyterHub. All data access services run via Docker containers in a Kubernetes cluster. The LSST project is still under development, but will produce 15TB of data per night with processing on an LSST cluster at NCSA. They are planning to use containerized scientific pipelines and data access services, including JupyterHub.

Greg Bauer presented on container support at Blue Waters, the largest academic supercomputer. Blue Waters currently supports NERSC's Shifter container framework. Containers allow Blue Waters to extend the life of the Blue Waters OS, since it is frozen at SLES11SP3 and accommodates new workloads. The traditional HPC "module" approach does not provide support for old and complex software stacks. Containers allow users to use any software stack with full control over their environment from laptop to HPC. Shifter supports Docker image files and native MPI with Gemini interconnect access.

Roland Haas and Eliu Heurta presented on the BOSS-LDG project that combines Blue Waters, Open Science Grid, Shifter and the LIGO Data Grid for gravity wave analysis. Detecting gravitational waves is computationally intensive. The LIGO software stack includes components that are not typically provided by HPC centers. In this project, the LIGO stack was modified to support OSG requirements (single thread, < 2GB memory, 1-12hr processing time, multiple restarts, statically linked, output < 10GB, no MPI). For this project, the LIGO Data Grid was connected to Blue Waters via OSG using Shifter.

Ian McEwen presented on the CyVerse project use of container technology. CyVerse consists of a number of components including on-demand VMs for researchers via Atmosphere and Jetstream and the Discovery Environment, the primary workbench and entrypoint for the CyVerse system. The DE is containerized both for IT deployment and for scientific workflows. CyVerse uses an HTCondor cluster to run Dockerized workflows on data. The project is working on supporting private tools (heavily constrained), groups/teams (resource allocation/permissions), bring-your-own compute, interactive jobs (Jupyter/RStudio), and more flexible tools for high-throughput processing.

John Fonner presented on container support at the Texas Advanced Computing Center (TACC). TACC uses container solutions for internal infrastructure (i.e., containerizing web services such as DesignSafe or JupyterHub). Docker is available to Jetstream users. However, due to the kernel requirements and security issues, Docker will never be supported on HPC systems. TACC does support the use of Singularity on all HPC systems and currently hosts >2,400 Singularity images for the biocontainers project. Containers on HPC support apps with many dependencies, facilitate user-extensible environments, and reduce burden on HPC staff to support new modules. However, containers also have a learning curve for technical (and non technical users) and the large image sizes can be problematic for storage. TACC is working to connect to Singularity Hub, support container building as a service, as well as workflow support. Questions include where to store metadata and documentation, support for MPI-enabled containers, and supporting specialized hardware.

Kacper Kowalik presented on the yt.hub and Whole Tale project architectures. Both systems use containers for IT infrastructure management but also to provide access to research data via interactive interfaces such as Jupyter and RStudio. They have developed a novel framework for exposing data from multiple different sources to containers via a Fuse filesystem. The Whole Tale project relies heavily on Docker for the "Tale" concept. A "tale" is a runnable research environment that consists of a Docker image, data, and configuration information

Andrea Zonca presented on container use at SDSC. SDSC Comet currently supports Singularity 2.3.1 including MPI (MVAPICH) and GPU nodes (CUDA 8). Containers are also used by a number of XSEDE gateways including Singularity containers as HPC applications and Docker containers for web-based services. Andrea demonstrated how JupyterHub can be used to provide interactive access to HPC resources.

Gerard Lemson presented on the use of containers in SciServer. SciServer provides interactive access to large research datasets for a number of domains including astronomy, materials science, oceanography, and genomics. SciServer compute provides access to data via Jupyter notebooks. Data is mounted into the containers via Docker volumes. They have added support for batch processing, also via containerized workflows. SciServer supports resource access control. They are looking at extending to support analysis on HPC resources.

Craig Willis presented on the NDS Labs Workbench platform. Labs Workbench is the first product of the National Data Service consortium, originally intended to support the sharing, discovery, and evaluation of research data management, analysis, and visualization tools. Based on Kubernetes, Labs Workbench allows users to deploy Docker-based applications. It has been successfully used to provide interactive access to research data for the TERRA-REF project and to support tutorials, workshops, and classroom activities centered on data science and data curation education.

David LeBauer and Max Burnette presented on the TERRA-REF project. TERRA-REF provides a compute pipeline and interactive access to a wide variety of sensor-born, genomic, and manually collected data. It is an end-to-end use case for container-based approaches to research computing and data access combining cloud compute (OpenStack), HPC (ROGER), specialized software stacks, training/workshops, and a re-runnable pipeline. TERRA-REF uses the Labs Workbench service and hopes to further support publication and provenance tracking.

Yan Liu and Jeff Terstreip presented on container use at the CyberGIS Center for Advanced Digital and Spatial Studies. CyberGIS uses containers for gateway application development, interactive analysis environments, collaborative research and development, and cyberinfrastructure integration. They presented examples of two use cases. The TauDEM project provides access to 700GB elevation dataset via a custom web application and Jupyter notebook. The TopoLens application is a web-application built on containerized microservices. The National Flood Interoperability Experiment (NFIE) uses JupyterHub and Jupyter notebooks for collaborative development of new methodologies. Researchers share integrated data, code and visualization via notebooks. They noted several ongoing issues with deploying VM infrastructure, user authentication and authorization, container orchestration, HPC computation management via qsub and PBS.

Jian Tao presented on Simulocean, a computational platform for coastal modeling applications. Simulocean is a science gateway application built on Django, Celery, and RabbitMQ. Coastal models are implemented as Docker containers and stored in the CRC Dockerhub repository. Via the gateway service, users can upload data to run models on HPC resources, including Jupyter notebooks.

Demonstration Summaries

- SciServer compute
- BioConda/BioContainers
- Whole Tale data management system
- CyVerse Discovery Environment
- Singularity tutorial

Discussion topics

Prior to the workshop, participants were asked to identify key topics that they hoped to discuss during breakout groups. The topics are listed below with the percent of voting participants interested.

Topic	**Votes**
Integration with HPC environments	67%
Interactive analysis environments	67%
Integration with workflow systems	58%
Shared storage across containers	50%
Security/permissions	50%
Archiving and preservation of images	42%

Education/workshop environments	33%
Science gateways	33%
Container orchestration/scalability	33%
Processes for containerizing applications	25%
Supporting licensed software	17%
Managing images	17%
Batch job support	8%
Supporting X windows applications	8%

Participants were asked to define what each of the top 5 topics meant to them:

Integration with HPC environments	Learning about Singularity/Shifter Exposing Singularity containers vs modules Launching jobs (with data) from interactive environments
Interactive analysis environments	Figuring out "gotchas" -- how are people solving problems. Jupyter vs Rstudio vs X environments Profiling load/capping access
Integration with workflow systems	We'd like to move to more complex workflow support: data staging Common Workflow Language supports containers (Docker specifically) ; could support other environments;
Shared storage across containers	Examples of how people are using data volumes as units Performance/reliability/scalability Kacper tell us more about WT Security? Data with HIPPA compliance rules. Can we make these environments safe enough.
Archiving/preservation	Allowing users to compose images How to archive and preserve images

Integration with HPC environments

Subtopics:

- Learning about Singularity/Shifter
- Exposing Singularity containers vs modules
- Launching jobs (with data) from interactive environments

John Fonner from TACC presented the Agave Platform, a "Science-as-a-Service" API platform for high-performance computing (HPC), high-throughput computing (HTC) and big-data resources[4]. Agave encapsulates underlying storage, transfer, and compute resources using a simple REST API. He reviewed the Agave ToGo portal (https://togo.agaveapi.co) and CLI toolkit (https://bitbucket.org/agaveapi/cli). The Agave Platform can be hosted locally (https://bitbucket.org/tacc-cic/agave-flat) or by TACC. Participants noted other similar systems including Apache Airavata and NERSC Newt. Agave is used by the CyVerse project to allow people to add their own clusters or local HPC resources. There are 12+ Agave tenants for CyVerse alone. The Agave model supports copying data into the target system, executing and monitoring jobs, capturing the final results and transferring back. CyVerse uses iRODS to keep data local (Cyverse public dataset is ~50TB, 2PB total).

Andrea Zonca fro SDSC presented a tutorial[5] using demonstrating Singularity on SDSC Comet. Singularity is a container framework developed by the Lawrence Berkeley Lab[6] to package and execute scientific workflows in HPC environments. Few of the workshop participants had direct experience with Singularity and there were a number of questions clarifying the differences between Singularity and the more commonly used Docker. Singularity doesn't support many of the abstractions provided by Docker (network isolation, not designed for long-running services, doesn't have full support for resource constraints, no orchestration, etc). However, Singularity containers do not run as root and do not face the same root escalation issues that make Docker unsuitable for HPC environments. Singularity does require root access to create images. There is a use case for a Singularity registry separate from Dockerhub.

John Fonner walked through the docker2singularity[7] project that can be used to easily convert Docker images to Singularity images. There is currently a problem in the image size estimation. Images must have mount points pre-specified, so they've included common directories from a variety of HPC systems. docker2singularity has been used for over 2400 images for the biocontainers project.

Participants had questions about Singularity, particularly related to MPI and performance. There is anecdotal evidence of performance improvements with Singularity, particularly with IO,

---

[4] https://agaveapi.co/
[5] https://github.com/zonca/singularity-comet
[6] http://singularity.lbl.gov/
[7] https://github.com/TACC/docker2singularity

but this hasn't been thoroughly tested. Python may also perform faster in containers. There's been no analysis with GPU nodes.

Participants also discussed models for running HPC jobs via science gateways. A common model is to use a common gateway or community user. Since data isn't on the execution system, middleware copies data to the system, then submits jobs to scheduler, monitors job outputs, and provides results. Data and containers are copied to the target system, then SSH into the system and run the job. For XSEDE gateways, very large input/output data is transferred via Globus. One issue with the gateway user approach is that the gateways applications must track users internally.

Shared storage across containers

- Examples of how people are using data volumes as units
- Performance/reliability/scalability
- Whole Tale data management design
- Can we make these environments safe enough for HIPPA?

Participants were particularly interested in Kacper Kowalik's work with the yt.hub and Whole Tale projects to expose data from a variety of sources to containerized applications via Fuse. Their system uses Girder for data management to abstract folders and files. Participants asked questions about read/write availability, multi-user environments, and whether standard seek is always available (depends on backing system).

The Cyverse Discovery Environment uses iRODS to stage data to local disk in the HTcondor system for batch processing.

Archiving, management and preservation of images

Many of the projects are using Docker because of the promise of scientific reproducibility and relying on Dockerhub (or SingularityHub) for long term image storage/archiving.

The Whole Tale project uses a private Docker registry. Internally they are using checksums instead of tags. They do not provide long-term storage/preservation because of lack of storage. They are preserving source/Dockerfile. CyVerse is using a private registry and Github repository with Dockerfiles. Community requirements for reproducibility are unclear. SciServer is planning something similar.

The group discussed whether images were the correct format for preservation (yes, since it's just tarballs for Docker). Singularity uses a sparse file format.

The group discussed whether NDS could support a proper registry for scientific images for both Docker and Singularity environments as well as best practices (i.e., checksums v tags). It may be necessary to limit the number of base images. DOIs represent a social contract (practically ~5 years). There may be issues for long-term preservation (Dockerfiles have lots of external factors).

Interactive analysis environments

Several of the groups represented at the workshop had already developed systems used to provide interactive analysis environments via containers. This includes SciServer, Whole Tale, yt.hub, and NDS Labs Workbench. The CyVerse project is planning to add support and was interested in understanding best practices and key challenges.

The first question centered on application support. Jupyter is a dominant platform, but RStudio is still widely used by some communities. The consensus is that Jupyter and Rstudio are first-class applications and systems should be neutral. SciServer has found that R users are happy using Jupyter and don't necessarily need the full RStudio environment. The NDS Labs Workbench supports other interactive development environments and noted that developers particularly are not satisfied with Jupyter for development purposes (although JupyterLab is a move in this direction).

The group also discussed authentication and authorization. SciServer provides single sign-on via OpenStack Keystone. Whole Tale uses Globus Authentication and noted that NCSA security recommended Oauth with JWT. NDS Labs Workbench has a local user database only and uses the Kubernetes ingress controller (aka nginx reverse proxy) for access to running applications, which is similar to the proxy provided by Jupyter hub.

Participants also discussed issues of permissions, particularly which user accounts are run in containers and whether/how they are mapped to users and groups. This led to a discussion of trusted and untrusted containers and how to monitor what is on an image.

Corners/snags:

- Authentication
- User/permissions
- Proxy requirements (path v subdomain)
- Licenses and user agreements
- Monitoring what goes inside (trusted v untrusted)
- Timeouts/culling
- Resource constraints

Container orchestration frameworks

A final discussion was used to clarify the differences between "container orchestration frameworks" such as Docker Swarm and Kuberentes and traditional clustering and scheduling services such as Slurm, HTCondor, TORQUE.

Docker Swarm and Kubernetes are intended to run containerized microservices. They support a variety of IT infrastructure capabilities (e.g., scaling of web-based services, deployments, upgrade paths, etc). They are increasingly used to provide cloud-based clustering and batch compute services (e.g., Kubernetes Job support).

There was a discussion about the differences between Docker Swarm and Kubernetes (compared to vi/emacs debates of old). Docker Swarm has changed over the past year and offers many of the same features as Kubernetes. It is considered to be more user friendly and "just works". Kubernetes has more abstractions, potentially offering more power/flexibility, but requires more expertise. Examples include the concept of Pods which share the same network stack. LSST, DES, and NDS all use Kubernetes. Whole Tale, Brown Dog and DesignSafe use Docker Swarm. CyVerse is evaluating Kubernetes for all services, but will continue to use HTCondor for container execution.

Interoperability/opportunities

As a closing topic, participant discussed key opportunities for interoperability between systems.

- Authentication: shared authentication between systems will be important -- a problem already seen in the XSEDE project. The group discussed Globus Authentication, CILogon and Oauth2/OIDC. Globus uses CILogon and tries to address the problem of multiple identities. However, Globus and Oauth are intended for web-based flows -- command line and API support is less clear. TACC and Comet support XSEDE authentication. XSEDE and gateway communities use X509 and gsi-ssh, but this is going away. CyVerse/Agave use WSO2 and CAS. Gateways use common accounts instead of per-user accounts).
- Data transfer: interaction between systems will also require transfer of data. What is the best way to allow people to push large amounts of data (iRODS, Globus).
- Sharing tools: Can we define standards for sharing tools across systems (both interactive and non interactive)? Containers solve this to some extent, but maybe there are best practices or standards within this community to facilitate trust. Another discussion focused on tracking changes within a Jupyter notebook. Instead of sharing the container image, can we simply incorporate more information into the Notebook.
- Service APIs: Libraries that allow users (not infrastructure providers) to interact with your system (cyverse.py, wholetale.py) and submit jobs or start containers.

The group discussed several possible science cases:

- A student working on Millennium simulations via yt.hub
- TERRA-REF/CyVerse: TERRA-REF already pushes data to CyVerse for genomics and bisque image analysis.
- SciServer launching containers on Whole Tale
- PEcAn - moving to containerization of ecosystem models, database, statistics tools
- HydroShare 2: container-based hydrological models using iRODS (ROGER, RENCI, TACC) with all models in containers.
- National Flood Interoperability (NFIE)

Major takeaways

Participants were asked to describe their biggest "takeway" from the workshop. A few examples include:

- How similar the architectures are with many core similarities. This suggests that we should be able to create interoperability between projects to move users around for the science.
- Whole Tale's GirderFS/FUSE architecture solves a difficult problem.
- Similarities and overlap between systems like SciServer/WholeTale and science gateways. There are lessons to be learned from the XSEDE community.
- Big differences between Singularity and Docker
- Jupyter notebooks are a portable research product. Everyone is using them.
- Lots of components that could serve as design patterns or shared implementations (GirderFS, Agave API).

Use cases, best practices, and challenges

This section summarizes use cases, best practices and challenges identified during the workshop.

Use cases

Remote, interactive access to research data

Services such as CyVerse, SciServer, Whole Tale and the TERRA-REF Analysis Workbench have been developed specifically to enable remote, interactive analysis to often large research datasets. Container technology is central to these services because of both IT management qualities (efficient storage, fast startup, etc) and the ability to easily package and distribute custom software stacks specific to research communities or even particular datasets. These services also offer remote collaboration, allowing users to share results, code, and data all within the same system. These systems are similar to but distinct from traditional science gateways.

Containers in high-performance computing

Research and high-performance computing centers are increasingly adding support for containers via frameworks such as Shifter and Singularity. Containers offer the ability to accommodate new scientific workloads, supporting more complex software stacks that are controlled by the researchers. Containers also offer the potential to extend the life of supercomputing systems, supporting newer versions of core libraries when OS versions are frozen. The rejection of Docker due to security concerns highlights a central difference between HPC and cloud computing environments. Even with containers, scientific workloads still require traditional HPC features such as MPI and workflow management.

Interactive HPC

Research and high-performance computing centers are increasingly adding support for interactive access to compute resources. Examples include the TACC Visualization Portal[8], the Minnesota Supercomputing Institute (MSI) Interactive HPC service[9], and similar services offered by NERSC[10].   NERSC has also added support for launching and monitoring jobs in Jupyter via their SLURM-MAGIC plugin[11]. Similar models can be found for Spark clusters via both Jupyter and Zeppelin. MSI and NERSC have both argued that interactive environments may replace the login/edge node for most users going forward.

HPC APIs

A number of services have emerged providing REST-based access to HPC resources including the Agave API[12], NEWT[13], Apache Airavata[14], and Livy[15] for Spark.   These APIs are intended to enable remote, web-based submission, management and monitoring of applications and workflows on compute clusters.

Science gateways

Science gateways are custom applications or portals developed by and for science and engineering communities to "access shared data, software, computing services, instruments, education materials, and other resources." Science gateways are typically complex full-stack web applications with rich web interfaces, backend databases, and message queues, often

[8] https://vis.tacc.utexas.edu/
[9] https://www.msi.umn.edu/content/interactive-hpc
[10] https://cug.org/proceedings/cug2017_proceedings/includes/files/pap143s2-file1.pdf
[11] https://github.com/NERSC/slurm-magic
[12] https://agaveapi.co/
[13] https://newt.nersc.gov/
[14] https://airavata.apache.org/
[15] https://livy.incubator.apache.org/

leveraging HPC APIs and community accounts for shared access to compute resources. Container-based analysis environments and Jupyter notebooks are increasingly being viewed as "mini science gateways" offering new forms of collaboration.

Training and education

A central benefit of container-based analysis environments is the ability to support tutorials and workshops. Container-based interactive environments offer the ability to rapidly deploy web-based, consistent software environments with access to real data and potential sharing of results between users. Through the use of orchestration frameworks such as Docker Swarm and Kubernetes, environments can be elastically scaled to support short-term bursts in usage.

Packaging, sharing preserving research software

- Opportunities to share tools/processes

IT infrastructure management

A key value of container technology is

- Container technology enables more efficient resource utilization, optimizing resource usage (in cloud scenarios.
- Packaging applications and dependences
- Packing many applications in a single cluster
- Simplifying management
- Software stacks customizable by the researcher/project
- All dependences available, can be run anywhere

Best practices and challenges

Interactive analysis environments

Authentication and authorization

- SSO between services
- Authentication, authorization, group management

Security

- Docker v singularity
- Kernel requirements and security issues

- Trusted v untrusted containers

Data management

- Data access and permissions

Integration with HPC, workflows, etc

Doesn't change the research problem:

- Job submission/HPC integration
- MPI
- Interaction with traditional compute/HPC resources
- Basic needs are still the same (MPI, workflows, etc)

Image management and preservation

- Image management

Orchestration v clusters

- Container orchestration frameworks vs cloud VMs

Appendices

Participant profiles

Projects	Description
SciServer (Jai Won Kim, Gerard Lemson)	SciServer Compute supports interactive and batch access to to multiple large public datasets across several domains (including the Sloan Digital Sky Survey) via containers. They support Rstudio/Jupyter/Matlab interactive environments and have developed a custom job scheduler for containers, each with supporting scripting libraries for SciServer component and data integration. SciServer compute supports SSO with user defined group access to shared storage and access to centralized datasets, some in relational databases See also: ● Dmitry Medvedev, Gerard Lemson, and Mike Rippin. 2016. SciServer Compute: Bringing Analysis Close to the Data. In Proceedings of the 28th International Conference on Scientific and Statistical Database Management (SSDBM '16).
Cyverse (Ian McEwen)	The CyVerse Discovery Environment (DE) uses containers (via Docker) to support customizable, non-interactive, reproducible workflows using data stored in the CyVerse Data Store, based on iRODS. Additionally, the DE supports using Agave to run analyses in HPC environments, which may themselves be containerized. They are actively working on enabling interactive jobs through the same system, as well as a "bring your own compute" system for those with access to their own computational resources, all made more possible, flexible, and reproducible through containerization. See also: ● Devisetty, U. K., Kennedy, K., Sarando, P., Merchant, N., & Lyons, E. (2016). Bringing your tools to CyVerse Discovery Environment using Docker. F1000Research, 5, 1442.
TACC (John Fonner)	TACC has installed Singularity container support on all of its HPC systems and is working with BioContainers to make 2400+ BioConda applications findable and accessible at TACC or any HPC system that supports Singularity. The end result of these efforts is to support all BioConda packages across all Cyverse infrastructure. This already works using Docker on the Cyverse Condor cluster and will also provide solution for other HPC systems using Singularity.

Whole Tale, yt.Hub, RSL, Data Exploration Lab (Matt Turk, Kacper Kowalik)	The yt Hub provides access to very large datasets (both observation and simulation based) via the integration of Girder and Jupyter Notebook/Lab. Entire available data is locally mounted to compute nodes of a Docker Swarm cluster via NFS. However, the physical location of the data is abstracted through a FUSE filesystem, which allows to provide only a subset of data selected by the user inside the container running Jupyter Notebook/Lab. The basic architecture of the yt Hub: Girder + remote environment with data selection, is currently being extended as a part of the Whole Tale project, which provides (among other things) the ability to launch containerized applications over a wide variety of the *remote* datasets (e.g., via DataOne). They are addressing complexity of exposing data to containers via a variety of underlying mechanisms (posix, S3, HTTP, Globus, etc) through a data management framework. In contrast to the yt Hub, data is provided inside the computing environment on demand using a sync mechanism and local cache, rather than being served locally. Containers also play a role in provenance/preservation of scientific workflows and publication process. The Renaissance Labs project will leverage this same approach to provide access to the Renaissance Simulations at SDSC – adding the ability to move analysis to HPC resources and adding a custom UI.
TERRA-REF (David LeBauer, Max Burnette)	The TERRA-REF projects provides access to a large reference dataset for plant biology. They support interactive access to the data via the NDS Labs Workbench (see below), which allows users to deploy customized container environments near the data. They also use containers for the data processing pipeline, which runs on a combination of VMs and the ROGER cluster. See also: • https://terraref.gitbooks.io/terraref-documentation/content/
Blue Waters (Greg Bauer)	The Blue Waters supercomputer now supports containers through NERSC's Shifter. Container support was added to support three use cases (LCH/Atlas, LIGO, LSST/DES). Technical challenges include containerizing applications, handling permissions, storage/IO, and MPI. See also: • Presentation: Interacting with Shifter on Blue Waters

LSST/DES (Matias Carrasco Kind)	The Dark Energy Survey (DES) project uses a Blue Waters allocation (not in containers) to run processes on raw images and generate catalog data. The DES Labs service provides access to the resulting catalog and image data for over 500 collaborators. All services run in containers on Nebula (DESCut, Jupyterhub, Web client) using Kubernetes. Catalog data (1PB) is available via Oracle. Shared storage is provided via NFS/Cinder. The Large Synoptic Survey Telescope (LSST) is planning to use a Kubernetes cluster for production (reliable, high-availability). With many collaborators, containerization of pipelines offers many benefits (isolation, deployment at scale). LSST will similarly offer Jupyterhub to control notebooks. See also: ● Presentation: Data Access for Astronomical Surveys: A brief description and demo of tools used by DES
LIGO/OSG (Roland Haas, Eliu Huerta)	Title: BOSS-LDG: A Novel Computational Framework that Brings Together Blue Waters, Open Science Grid, Shifter and the LIGO Data Grid to Accelerate Gravitational Wave Discovery Abstract: We present a novel computational framework that connects Blue Waters, the NSF-supported, leadership-class supercomputer operated by NCSA, to the Laser Interferometer Gravitational-Wave Observatory (LIGO) Data Grid via Open Science Grid technology. To enable this computational infrastructure, we configured, for the first time, a LIGO Data Grid Tier-1 Center that can submit heterogeneous LIGO workflows using Open Science Grid facilities. In order to enable a seamless connection between the LIGO Data Grid and Blue Waters via Open Science Grid, we utilize Shifter to containerize LIGO's workflow software. This work represents the first time Open Science Grid, Shifter, and Blue Waters are unified to tackle a scientific problem and, in particular, it is the first time a framework of this nature is used in the context of large scale gravitational wave data analysis. This new framework is designed to run the most computationally demanding gravitational wave search workflows on Blue Waters and accelerate discovery in the emergent field of gravitational wave astrophysics. We discuss the implications of this novel framework for a wider ecosystem of Higher Performance Computing users.

NDS (Craig Willis, Mike Lambert)	Presentation: NDS Labs Workbench and DataDNS Description: Brief presentation of the Labs Workbench service and how it's being used to support in-place analysis of data, development, and education/training environments. We'll also introduce the DataDNS initiative. The NDS Labs Workbench is a generic platform for launching containerized environments near remote datasets, leveraging Kubernetes. Labs Workbench is deployed on OpenStack as a Kubernetes cluster with GlusterFS for a shared user filesystem across containers (e.g., home directory). Workbench is used by the TERRA-REF project and increasingly for training/education environments (hackathons, workshops, bootcamps, etc). The DataDNS project is an emerging vision for supporting access to remote computational environments. Workbench is a single optional component of the DataDNS framework. See also: ● Craig Willis, Mike Lambert, Kenton McHenry, and Christine Kirkpatrick. 2017. Container-based Analysis Environments for Low-Barrier Access to Research Data. In Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact (PEARC17)
CyberGIS (Yan Liu, Jeff Terstriep)	The CyberGIS project recently developed CyberGIS-Jupyter to integrate cloud-based Jupyter notebooks with HPC resources. The project adopts Jupyter notebooks instead of web GIS as the front-end interface for both developers and users. Advanced GIS capabilities are provided in a pre-configured containerized environment. The system also supports on-demand provisioning to deploy multiple instances of gateway applications. See also: ● Dandong Yin, Yan Liu, Anand Padmanabhan, Jeff Terstriep, Johnathan Rush, and Shaowen Wang. 2017. A CyberGIS-Jupyter Framework for Geospatial Analytics at Scale. In Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact (PEARC17)
SDSC (Andrea Zonca)	Presentation: Jupyterhub + Singularity on HPC Description of an experimental deployment of Jupyterhub with Globus Authentication on SDSC Cloud Openstack with Notebooks spawning on Comet computing nodes with Singularity. Users should be able to choose container or bring their own. Use cases of Jupyterhub in HPC for teaching, research and Science Gateways. Also, Jupyterhub with Docker Swarm on SDSC Cloud Openstack for interactive analysis to support HPC jobs on Comet. See also: ● Blog posts: Deploy Jupyterhub on a Supercomputer with SSH Authentication and Jupyterhub deployment on multiple nodes with Docker Swarm

Coastal Resilience Collaboratory (Jian Tao)	CRC is developing an integrated, coupled modeling framework for the coastal modeling community to facilitate the deployment of complex models on cloud and cloud-like architectures with negligible performance overhead. The Coastal Model Repository (CMR) targets cloud and cloud-like architectures to enable quick deployment of coastal models and their working environments. CMR serves as a community repository for precompiled open source models that are widely used by coastal researchers. CMR distribute containerized coastal models via Docker hub.

Program

Monday August 14th

The first day we will focus on participant presentations with significant Q&A and discussion time. Topics that emerge from the presentations will be considered for breakout/deep-dive discussions on Tuesday and Wednesday.

Time	Description	
9:00 - 9:30	Introduction/welcome Logistics	Data Exploration Lab (Turk) NDS Welcome (Willis)
9:30 - 11:00	Presentations Discussion	LSST/DES (Kind) Blue Waters (Bauer) LIGO (Haas, Huerta)
11:00 - 11:15	Break	
11:15 - 12:30	Presentations Discussion	Cyverse (McEwen) Cyverse/TACC (Fonner) yt.hub, Whole Tale (Turk, Kowalik)
12:30 - 1:30	Lunch	
1:30 - 3:00	Presentations Discussion	SDSC (Zonca) SciServer (Lemson) NDS Labs Workbench(Willis)
3:00 - 3:15	Break	
3:15 - 5:00	Presentations Discussion	TERRA-REF (LeBauer, Burnette) CyberGIS (Liu, Terstriep) Simulocean/Coastal Modeling Collaboratory(Tao)
5:00 - 5:30	Discussion/planning	

Tuesday August 15th

Tentative: Based on discussions from Monday, we will organize breakout/deep-dive discussions for topics of interest (feel free to suggest). Additional presentations may be scheduled, depending on participation:

Topics from discussion:

- Integration with HPC environments
- Interactive anlaysis environments
- Integration with workflow systems (not enough expertise to discuss)
- Shared storage across containers
- Managing images/archiving preservation

Time	Description	
9:00 - 9:30	Summary/takeaways from Monday?	
9:30 - 12:00	Topics: Singularity tutorials Integration with HPC environments	Break ~10:45
12:00 - 1:00	Lunch	
1:00 - 3:00	Topics: Demos (Agave, WT, SciServer) Shared storage across containers	
3:00 - 3:15	Break	
3:15 - 5:00	Managing images/archiving presentation Interactive analysis environments	

Wednesday August 16th

Tentative: We will use the morning to conclude any break-out work and to consolidate group output (presentations, documents, etc) for the workshop report.

Time	Description	

9:00 - 9:30	Discussion/planning	
9:30 - 11:30	Breakout groups	
11:30 - 12:00	Closing	

References

1. Devisetty, U. K., Kennedy, K., Sarando, P., Merchant, N., & Lyons, E. (2016). Bringing your tools to CyVerse Discovery Environment using Docker. F1000Research, 5, 1442
2. Medvedev, D., Lemson, G, and Rippin, M. 2016. SciServer Compute: Bringing Analysis Close to the Data. In Proceedings of the 28th International Conference on Scientific and Statistical Database Management (SSDBM '16).
3. Milligan, M., 2017. Interactive HPC Gateways with Jupyter and JupyterHub. In Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact (PEARC17)
4. Piparo, D., Tejedor, E., Mato, P, Mascetti,L., Moscicki, J, and Lamanna, M. SWAN: A service for interactive analysis in the cloud.
5. Thomas, R., Canon, S., Cholia, S., Gerhardt, L., Racah, E., Toward Interactive Supercomputing at NERSC with Jupyter
6. Willis, C., Lambert, M., McHenry, K., Kirkpatrick, C. 2017. Container-based Analysis Environments for Low-Barrier Access to Research Data. In Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact (PEARC17)
7. Yin, D, Liu Y, Padmanabhan A, Terstriep, J, Rush, J., and Wang, S.. 2017. A CyberGIS-Jupyter Framework for Geospatial Analytics at Scale. In Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact (PEARC17)
8. Yu, W., Carrasco Kind, M., Brunner, R.J.. (2017) Vizic: A Jupyter-based interactive visualization tool for astronomical catalogs, In Astronomy and Computing, Volume 20, 2017 https://doi.org/10.1016/j.ascom.2017.06.004.
9. Boettiger, C. (2015). An Introduction to Docker for Reproducible Research. *SIGOPS Oper. Syst. Rev.*, *49*(1), 71–79. http://doi.org/10.1145/2723872.2723882
10. Merkel, D. 2014. Docker: lightweight Linux containers for consistent development and deployment. Linux J. 2014/
11. W. Felter, A. Ferreira, R. Rajamony and J. Rubio, "An updated performance comparison of virtual machines and Linux containers," *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Philadelphia, PA, 2015, pp. 171-172. doi: 10.1109/ISPASS.2015.7095802
12. Bernstein, D., "Containers and Cloud: From LXC to Docker to Kubernetes," in *IEEE Cloud Computing*, vol. 1, no. 3, pp. 81-84, Sept. 2014. doi: 10.1109/MCC.2014.51
13. Gruning, B.A., Rashe, E., Rebolledo-Jaramillo, B., Eberhard, C., Houwaart, T., Chilton, J., Coraor, N., Backofen, R., Taylor, J., Nekrutenko, A. (2017) Jupyter and Galaxy: Easing entry barriers into complex data analyses for biomedical researchers. PLoS Computational Biology
14. R. Dua, A. R. Raja and D. Kakadia, "Virtualization vs Containerization to Support PaaS," *2014 IEEE International Conference on Cloud Engineering*, Boston, MA, 2014, pp. 610-614. doi: 10.1109/IC2E.2014.41

15. Gregory M. Kurtzer, G.M., Sochat, V., Bauer, M. W. (2017). Singularity: Scientific containers for mobility of compute. PLoS One.
16. J. Hale, L. Li, C. Richardson and G. Wells, "Containers for portable, productive and performant scientific computing," in *Computing in Science & Engineering*, vol. PP, no. 99, pp. 1-1. doi: 10.1109/MCSE.2017.2421459