Supporting Hydra at SNL

Gary Lawson

Kristian Beckwith

Todd Heer

May 24th

2022 Tri-lab Advanced Simulation & Computing Sustainable Scientific Software Conference

Albuquerque, NM, USA
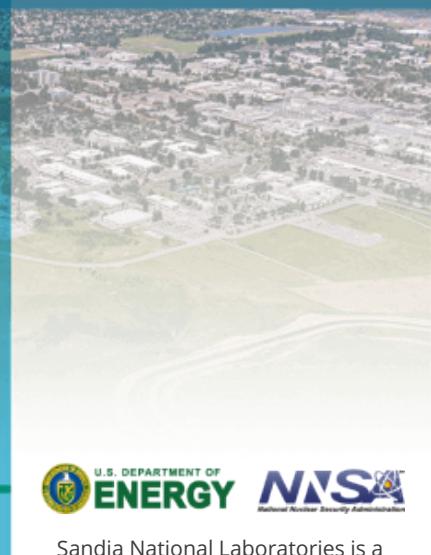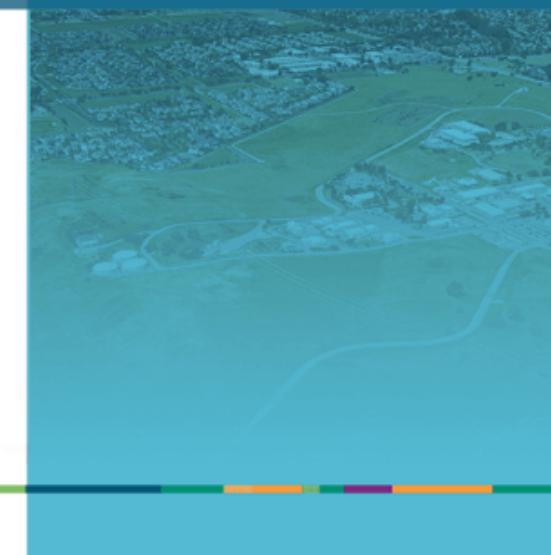
Introduction – What is ASC DevOps Core @ Sandia?

Product Owner:

- Scott Warnock

Scrum Master:

- Etone Mbome

Development Team:

- Jon Grzybowski

- Gary Lawson

- Chris Sullivan

- Paul Wolfenbarger

What do we do?

Code Scan

Code Vulnerability Scanning Tool

Common Development Environment

Consistent Multi-platform Software Stack

Dockerized Services

Jenkins
CDash

General Support

Hydra
VisIt

Hydra Support – what's the ask?

1684 Manager:
- Kristian Beckwith

Sandia National Laboratories

Hydra Project Lead:
- Marty Marinak

Hydra Developer:
- Joseph Koning

Lawrence Livermore National Laboratory

ASC DevOps Core Developer:
- Gary Lawson

Sandia National Laboratories

Support Request:

- Provide a software stack to support Hydra with Spack
  - Intel Compiler
  - MVAPICH2 MPI
  - General and LLNL-specific Third-Party Libraries (TPL's)

- Create and perform tests to verify/validate stack

- Deploy stack to metagroup-controlled project space
  - Optional Lmod module
  - Spack view

- Automate the end-to-end pipeline
  - LLNL to SNL Gitlab Repository Synchronization
  - Hydra TPL Stack Build, Test, & Deployment

Why should anyone care about this effort?

Sandia National Labs analyzes scenarios involving high energy density physics

Hydra is used for simulating high energy density physics problems

- Magnetic Fields
- Burn Products
- Radiation transfer

- Atomic physics
- Hydrodynamics
- Laser propagation

Providing this software for analysts at SNL helps both SNL and LLNL

- Expands expertise on the software

- Allows for collaborating for development and analysis efforts

- Enables coupling to SNL codes for multi-physics simulations

Because Hydra helps analysts to understand a scenario with more confidence

What is the issue?

SNL Programs desire to collaborate with LLNL to use Hydra on
ASC HPC resources for analysis and development

Hydra requires a software stack that is not otherwise supported by SNL

- Stack utilizes several LLNL software packages that are only available if provided by LLNL

- Stack utilizes an implementation of MPI that is not fully supported on SNL HPC platforms

- Stack must be validated and verified without access to Hydra source code

There is difficulty in enabling remote analysis and develop of SNL programs
in cross-site HPC resource environments

- Approval and access control may be required

- Understanding the usage of the HPC environment is required to run efficient simulations

Challenges with providing the Software Stack

1. Providing and verifying MPI on SNL HPC computational resources

   - OpenMPI
     - Supported by SNL admins on all HPC clusters
     - Used by the Hydra team, but found to introduce bugs at runtime

   - MVAPICH2
     - Exists on certain HPC clusters, but not supported by SNL admins
     - Built with Spack and verified against system installation for correctness
       - MiniAMR benchmark selected from Developer Tools Confidence Suite
       - Several problems tested for consistency between builds
   - Built and maintained in the TPL stack for Hydra

Challenges with providing the Software Stack

2. Verifying TPL stack for Hydra

- **Picked two representative packages in stack**
  - Hypre – Scalable linear solver and multigrid methods
  - MFEM – Lightweight & scalable library for finite element methods

- **Verified with E4S validation test suite**
  - Utilize the same stack as Hydra is to be built upon
  - Ensure important features are available and run correctly
    - Rule out features that may fail but have no impact on the final software product
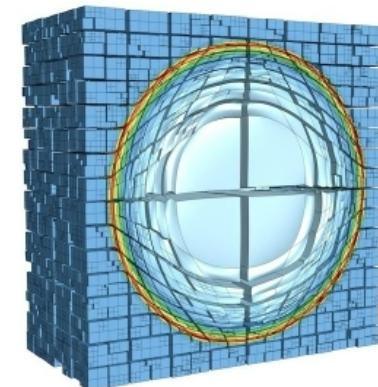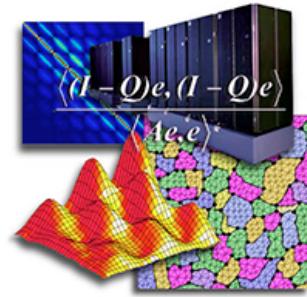
Hydra Workflow

1. Synchronize SNL ihpccluster Gitlab with LLNL RZ Gitlab
   - Hydra Spack Package.py repository

2. Synchronize TPL software & versions
   - Local Spack mirror
   - Spack environment files

3. Build for CTS1 and TLCC2 class platforms
   - SNL specific configurations
   - Impacts MPI and packages requiring MPI
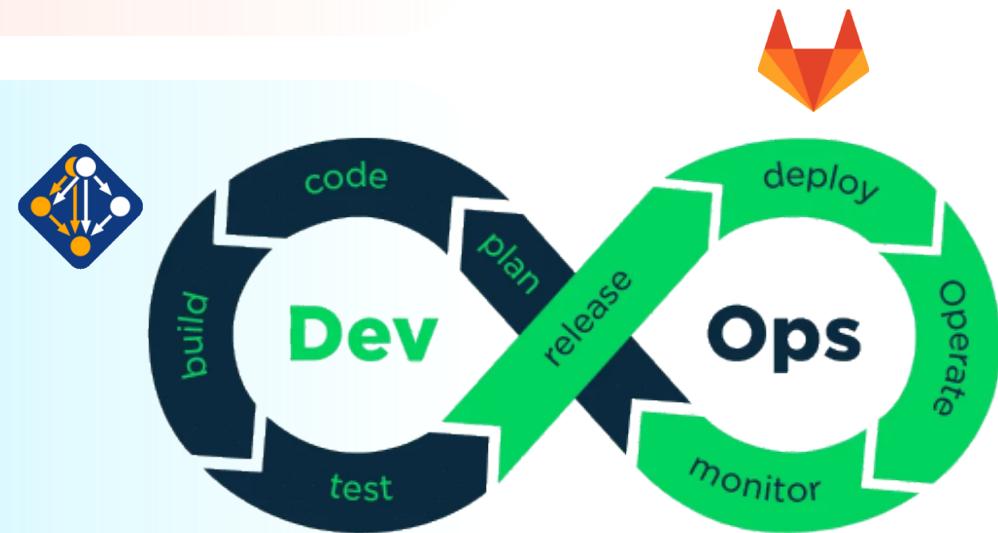
4. Test software deployment
   - MVAPICH2
   - Hypre/MFEM

Limitations & Challenges

Synchronization is manual

- Requires two-factor authentication to synchronize repositories

- Requires email collaboration to synchronize software stack

Python Support

- Not all Python modules are available in Spack

- Utilizing Miniconda installation to manage Python and modules

E4S Validation Tests require tweaks

- Intel Math Kernel Library (MKL) added to makefile for Hypre build

- SNL HPC systems utilize Slurm and the srun command

Tri-Lab Remote Computing Enablement (RCE) Program

RCE is integral to simplifying three HPC centers for a single NNSA userbase

RCE – Auth

Leverage tri-lab trust models for fewer auth hops

RCE - CI

Cross-site CI for leveraging remote compute resources

RCE – network

Improved bandwidth networks between sites

RCE - storage

Data transfer systems and process equity among the tri-labs

Cross-site Automation & Execution

RCE enables improved utilization of tri-lab HPC resources while maintaining system security and integrity

Trusted Authentication

- Synchronizing Gitlab repositories

- Cross-site CI & execution

Jacamar Runners

- User-level CI for improved security

- Automated build & test locally or cross-site

Cross-site Runners: Coming Together to Enable CI across institutional boundaries
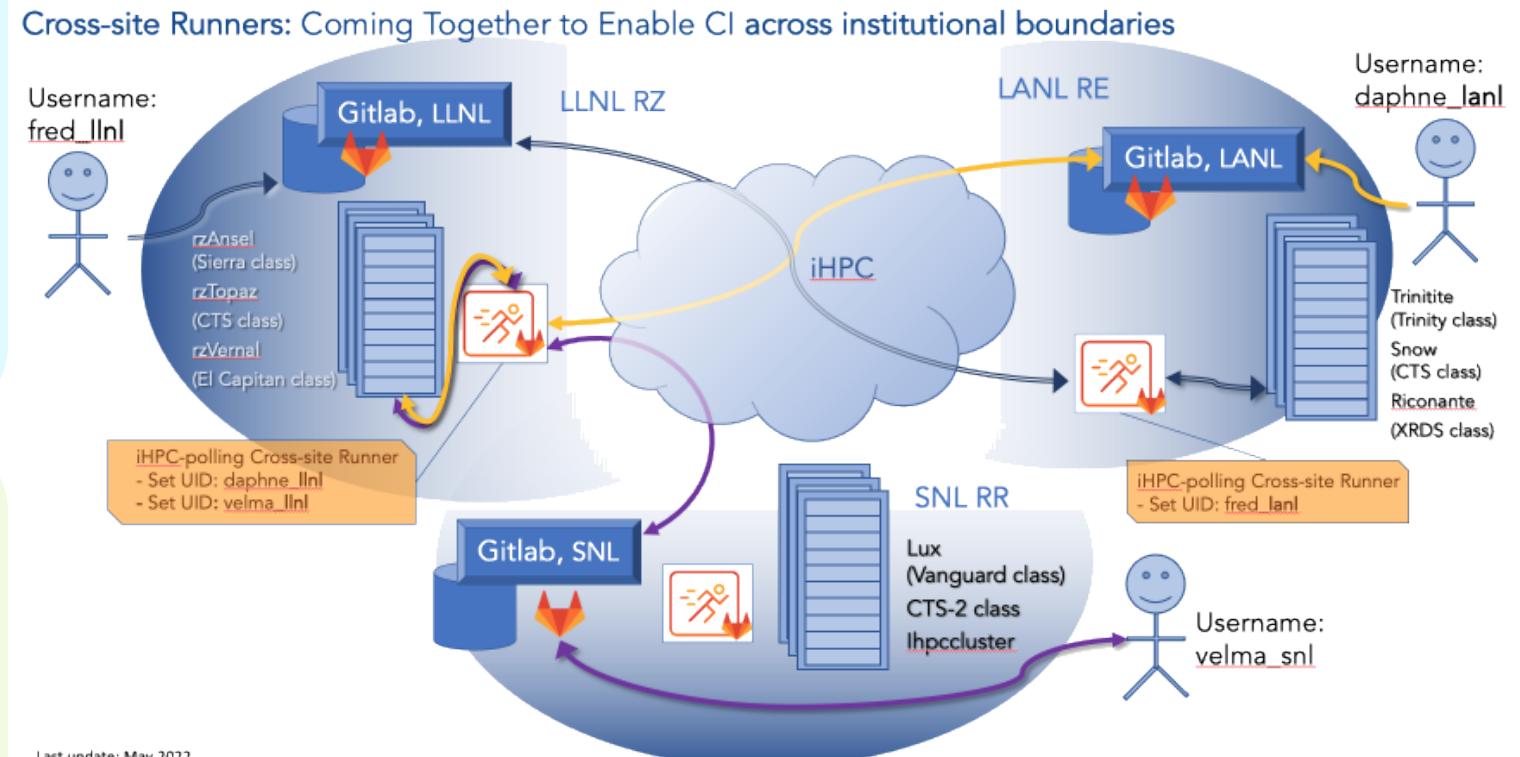
Conclusion – What is the impact of this effort?

Sandia National Labs analyzes scenarios involving high energy density physics

Providing Hydra for analysts at SNL helps both SNL and LLNL

- Reliable and consistent development and runtime environment

- Cross-site collaboration between LLNL & SNL

- Automated synchronization, build, test, and deployment

Hydra helps analysts to understand a scenario with more confidence

Points-of-Contact

Kris Beckwith   kbeckwi@sandia.gov

Scott Warnock   sawarno@sandia.gov

Gary Lawson   glawson@sandia.gov

Todd Heer   heer2@llnl.gov