



Trilinos Testing: Current Infrastructure and Future Direction

James Willenbring

ASC S3C

May 24, 2022

SAND2022-XXXX PE

Outline



- Brief History of Trilinos Testing
- Current Testing
 - Infrastructure
 - Pull request (PR) testing
 - Branch promotional testing
 - Periodic testing
- Where We May be Headed



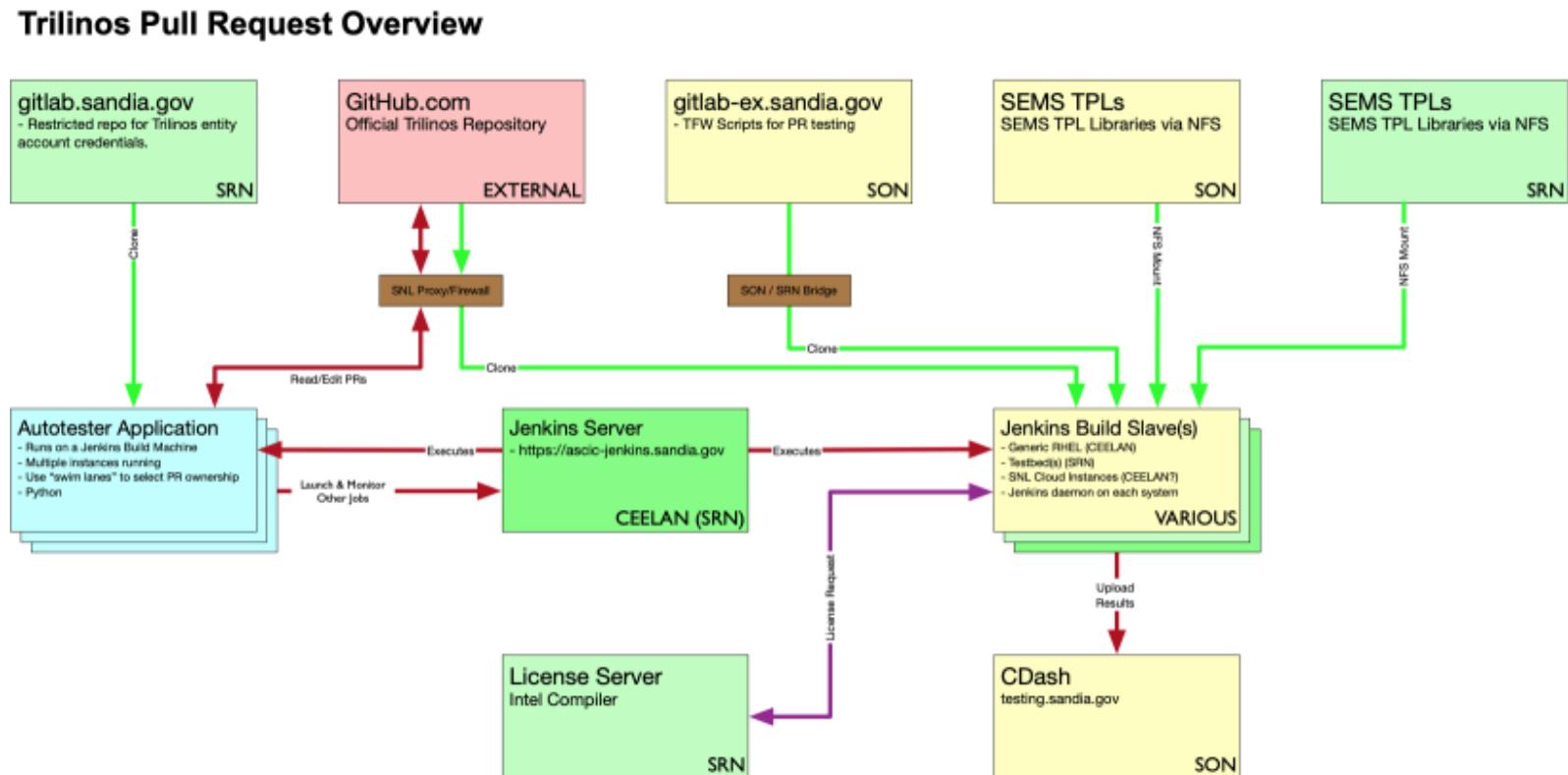
Brief History of Trilinos Testing



- 2002: Package tests run manually. Build system is make-based.
- 2003: First Trilinos test harness developed. One build attempt. Cron tab-driven. Results via email.
- 2006: Test Harness adds ability to report more than one build error. Tests driven via autotools build
- 2009: Checkin test script added to support more stable commits. Tests use CTest and CDash
- 2013: Jenkins (initially Hudson) used in place of cron for driving tests
- 2017: Pull Request testing added using the “autotester”. Emphasis shifted away from nightly testing
- 2018: Promotional branch testing added
- 2020: Trilinos tests added to E4S Validation Test Suite
- 2020-2022: Gen-config developed as a standard way to configure and build Trilinos

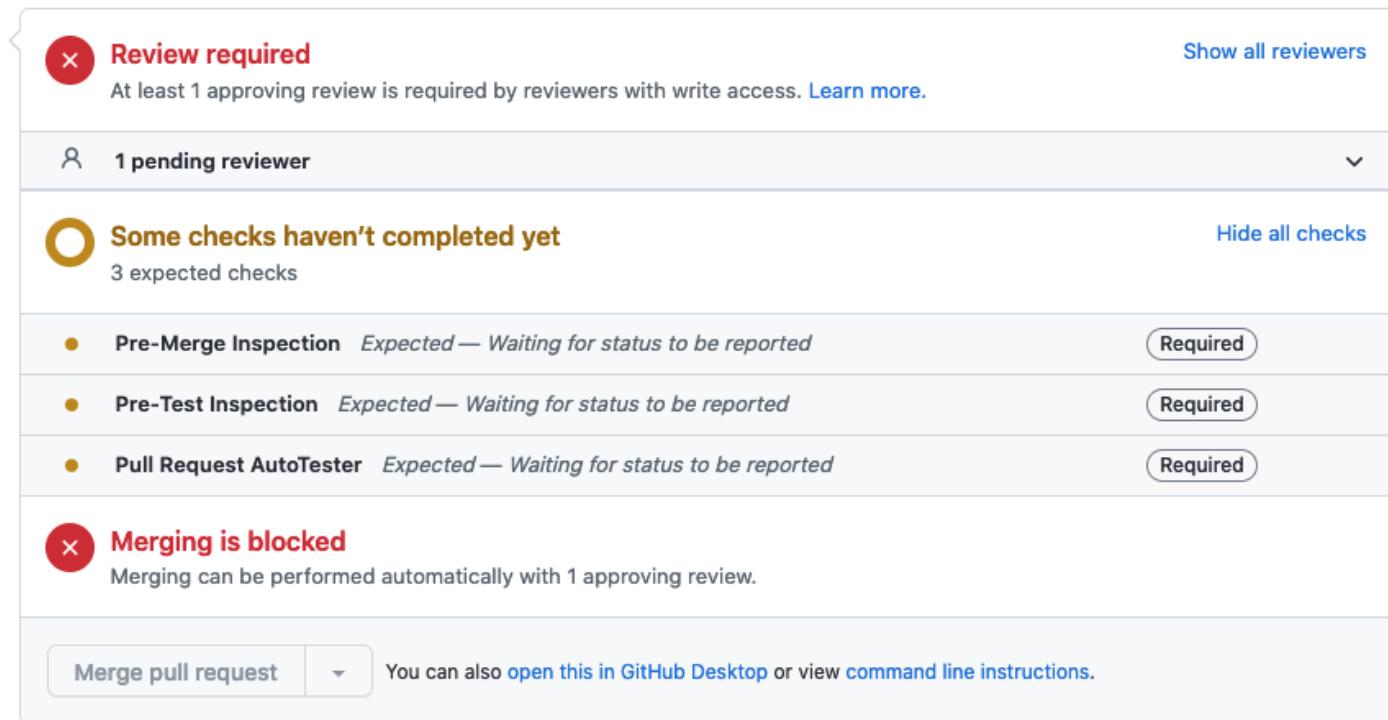
Current Trilinos Testing Infrastructure

- The “AutoTester”
- Jenkins
- CTest/TriBITS
- CDash
- Gen-config
- GitHub
- GitLab



Current Trilinos Testing

- Pull Request testing
 - A set of testing configurations that must pass before changes are made to the develop branch
 - PR test configurations
 - CUDA 10.2.2 (UVM off, temporarily unavailable)
 - CUDA 10.1.243
 - gcc-8.3.0
 - gcc-7.2.0-serial
 - gcc-7.2.0-debug
 - Intel-19.0.5
 - Clang-10.0.0
 - Python-3



Current Trilinos Testing

- Develop to Master promotional testing
 - A set of testing configurations that must pass before changes are promoted from the develop to the master branch
 - PRs are generated automatically to be tested
 - Run nightly. Includes all PR configurations (except Python-3) plus:
 - gcc-7.2.0
 - Intel-17.0.1
 - Clang-7.0.1
 - Clang-9.0.0
 - CUDA 10.1.243 rdc (temporarily unavailable)

Trilinos Master Merge PR Generator: Auto PR created to promote from master_merge_20220513_005814 branch to master #10521

- Periodic testing
 - “Nightly testing” that isn’t always run nightly
 - Used to monitor non-PR/branch promotion builds
 - May include clean and full variants
 - Some builds focused on a specific customer
- Other types of Trilinos tests
 - Experimental
 - Package-owned
 - Customer-owned

<https://testing.sandia.gov/cdash/index.php?project=Trilinos>



- Impact of PR testing
 - **Significant improvement in stability of the code**
 - Much easier to communicate what is and is not working
 - Develop branch is better tested than early releases of Trilinos
 - Prevents “broken window” effect
 - **Lower pre-push testing burden for developers**
 - Much better environment for new contributors
 - Problems with builds are generally easier to reproduce
 - More noticeable impact on development when key infrastructure is down

9 Trilinos Testing Next Steps

- Goals: Improve stability, flexibility, and maintainability
 - Allow single builds to be rerun
 - Fewer points of failure (trim down Will's horrible diagram)
 - Less custom, more COTS
 - CI software has improved greatly in the past 5 years
- What it might look like
 - PR testing outside of Sandia
 - Promotional branch testing inside Sandia
 - Use of containers
 - Add one or more Spack-based builds