Continuous Integration

David E. Bernholdt
Oak Ridge National Laboratory

Jared O'Neal
Argonne National Laboratory

Mark C. Miller, Paul Bryant and the ECP CD/CI Team

Better Scientific Software Tutorial
RF SciDAC 2020 Workshop

License, Citation and Acknowledgements

License and Citation

- **The requested citation the overall tutorial is: David E. Bernholdt, Better Scientific Software tutorial, in RF SciDAC 2020 Workshop, Knoxville, Tennessee. DOI: 10.6084/m9.figshare.11918397**
- Individual modules may be cited as *Speaker, Module Title*, in Better Scientific Software Tutorial…

Acknowledgements

- Additional contributors to this this tutorial include: Anshu Dubey, Mike Heroux, Alicia Klinvex, Jared O'Neal, and Katherine Riley, James M. Willenbring
- This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.
- This work was performed in part at the Argonne National Laboratory, which is managed managed by UChicago Argonne, LLC for the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.
- This work was performed in part at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.
- This work was performed in part at Sandia National Laboratories. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND NO SAND2017-5474 PE

The Short & Sweet of Continuous Integration

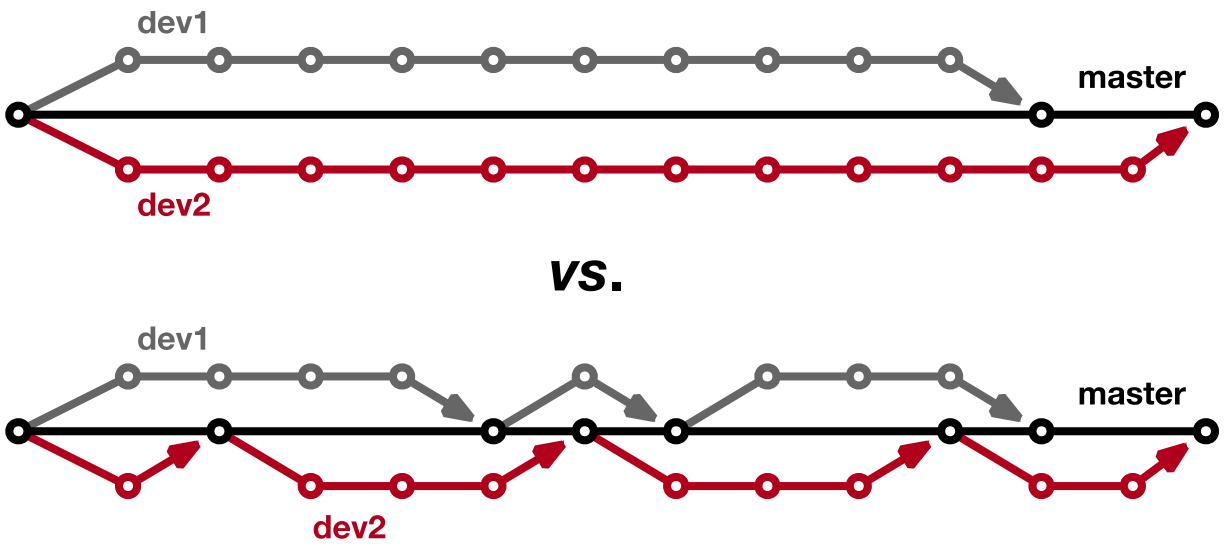
A master branch that always works

- DVCS workflow isolate master from integration environment
- Extend workflow to address difficulties of integrating
  - Minimize likelihood of merge conflict
  - Detect bugs immediately
  - Make debugging process quick and easy

Work Decomposition

Commit and integrate often

- Limit divergence between feature and master branches
- Decreased probability of conflict
- Conflict resolution is simpler and less risky



Error Detection

Test at integration to identify failures immediately

- Control quality of code
- Isolate failure to few commits
- No context switching for programmer

We want a system that

- triggers automated builds/tests on target environments when code changes and
- ideally tests on proposed merge product without finalizing merge.

Test Servers

Servers that

- automate the execution of a test suite or a subset of a test suite,
- allow for running tests on different environments,
- host an interface for viewing results, and
- allows for configuring when the tests are run.

Examples

- CTest/CDash
- Jenkins
- Travis CI and GitLab CI

Cloud-based Test Servers

- Linked to VCS hosts
  - GitHub & Travis CI
  - GitLab CI
  - BitBucket Pipelines
- Automated builds/tests triggered *via* pushes and pull requests
- Builds/tests can be run on cloud systems
- Test results are reported in repository's web interface
- Can trigger code coverage analysis & documentation build

Continuous integration (CI) Summary

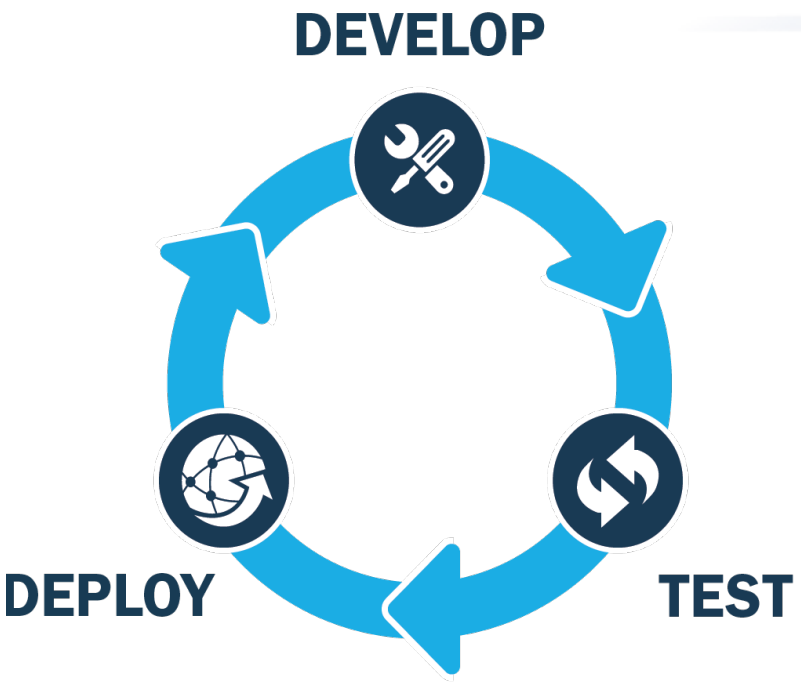
- Has existed for some time and interest is growing
- HPC community working to adapt CI for HPC machines
- Setup, maintenance, and monitoring required
- Prerequisites
  - A reasonably-automated build system
  - An automated test system with significant test coverage & useful feedback
  - Builds/tests must finish in reasonable about of time
  - Ability to bundle subset of tests

ECP Project: WBS 2.4.4
Software Deployment at Facilities - Software Integration

The Software Integration effort was established to bridge the ECP ST software development effort with the Exascale hardware and software environments deployed at the Facilities.

- **Continuous Integration (CI)** - Provide the ability to continuously test AD/ST software on facility hardware resources with software environments established at the Facility.

  Key for software development teams targeting systems being deployed agile feedback loop is key for development



Based on material provided by Mark C. Miller, Paul Bryant, and the ECP CD/CI Team

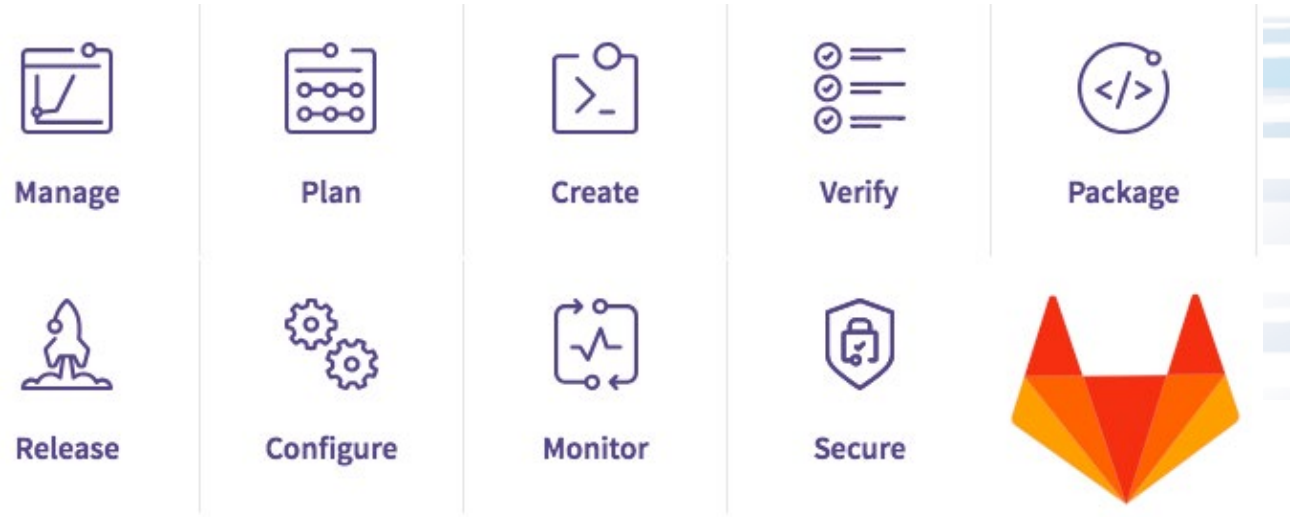
CI Solution for Large-Scale HPC Facilities

Need: HPC centers need new security features within Continuous Integration systems to serve thousands of users with unique hardware and security requirements.

Response: Using GitLab as a basis, extend security, permissions, and auditing features while improving intra-Lab and inter-Lab accessibility to CI pipelines.

GitLab

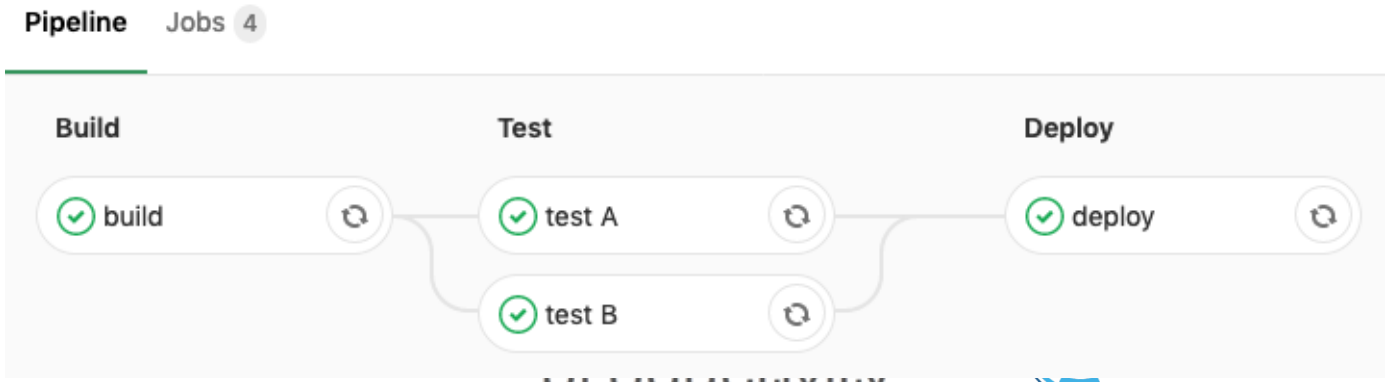
- GitLab is a single platform for the entire DevOps lifecycle
  - Plan, create, verify, and release
- Core functionality (free)
  - Version control, collaboration, CI/CD, and documentation tools
- Open source and capable of self management
  - MIT License
- Additional tiers of functionality are only available via license
  - https://about.gitlab.com/pricing/self-managed/feature-comparison/



Source: https://docs.gitlab.com/ee/

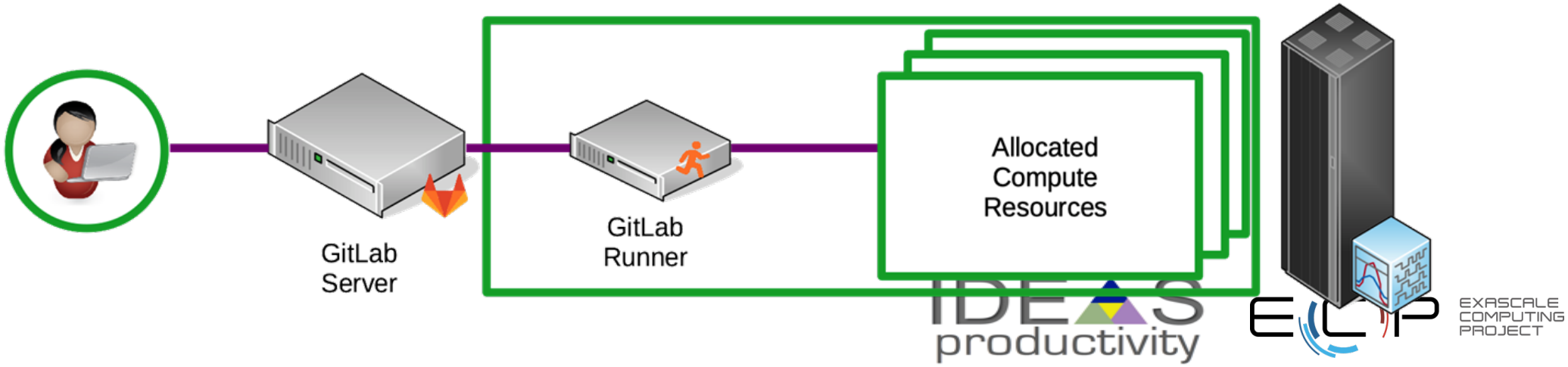
GitLab CI/CD

- GitLab offers robust support for CI/CD (Continues Delivery)
- Primarily controlled via a '.gitlab-ci.yml' file defined in your project repository
  - Defines the jobs that will be run, their structure, and what will trigger their potential execution
  - https://docs.gitlab.com/ee/ci/yaml/
- Terms:
  - Jobs: Core elements of CI, where commands are defined for execution
  - Stages: Defined order for job execution within a pipeline
  - Pipelines: Collection of jobs
- Pipelines can be triggered through a number of mechanisms, including: commits, manually, api, scheduling, merge requests, upstream projects, and more...



GitLab Runner

- “GitLab Runner is the open source project that is used to run your CI/CD jobs and send the results back to GitLab”
- Using GitLab CI/CD requires management of a runner:
  - Application supports the execution and reporting of CI jobs on target systems
  - Installed on target test resources and registered to a GitLab instance
  - Responsible for the execution of scripts whose creation is dictated by both the user’s CI file as well as administrative configuration



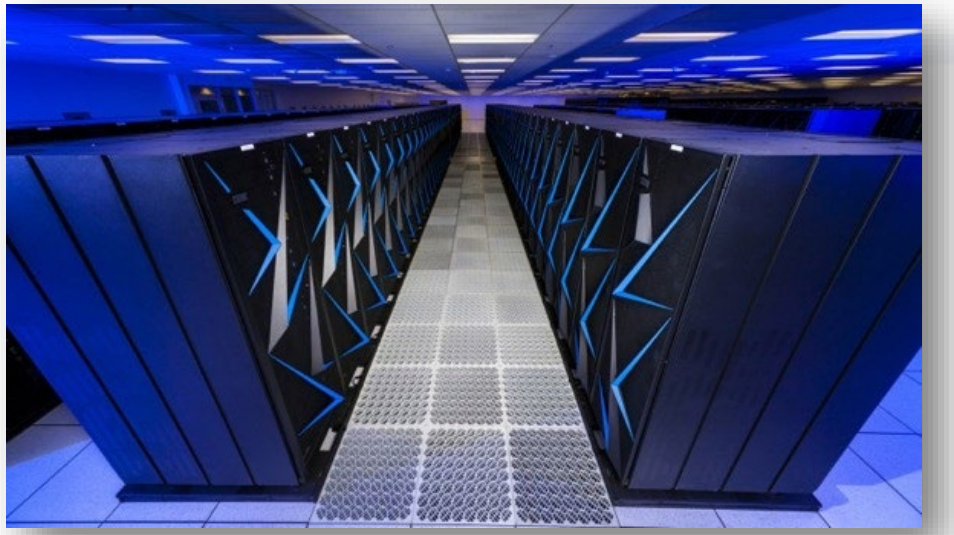
Our Vision

Delivering a secure, easy-to-use CI solution to support Software

Product testing against E6 HPC environments.

Security Challenges and Use Cases

- All GitLab CI jobs run as the single gitlab-runner user
- Achieving any level of isolation between CI jobs requires either:
  - 1) Targeting specific runners such as Docker or Virtual machines
  - 2) Having each user/team manage their own runner
  - Neither of these are ideal
- Use cases we aim to support
  - Runner data isolation
  - Account specific resources
  - Sensitive data sets
  - Administrator management



IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

Enhanced GitLab Functionality

- Developing and supporting enhancements to GitLab applications to ensure targeted HPC testing resources can be available to code teams
  - Integrating facility feedback to improve technical aspects and support logistical challenges.
- Runner capable of setuid, enforcing local permissions
  - Jobs are executed under a users local account
- Batch executor added to interface with Cobalt, LSF, and Slurm
  - A runner side configuration that dictates how a CI job will be executed.
- Federated cross site continuous integration
  - Previously auth endpoints only used at login, now they have been expanded and integrated into the CI process as well.
- Existing GitLab functionality will continue to be supported
  - For instance continued upstream development efforts and improved to GitLab's core CI/CD capabilities can be used for

What is the GitLab Runner Doing?

- The GitLab runner polls the server to identify available CI jobs
  - Will only begin executing a job once it has been scheduled by the server
- Job local to runner result in generated Bash scripts based upon the contents of the *.gitlab-ci.yml*
  - Works similar to upstream GitLab (https://docs.gitlab.com/runner/shells/index.html)
  - Scripts are executed by piping them into a non-interactive Bash login shell.
- Every step in a CI job is executed by a valid local user account
  - This is a key change with setuid enabled runner
- Each user is provided with a clean login shell
  - Ensures runner environment does not compromise subsequent jobs
  - Users experience an environment similar to what they would see on a login node
  - Depending on <u>your</u> environment this may affect the results of your CI

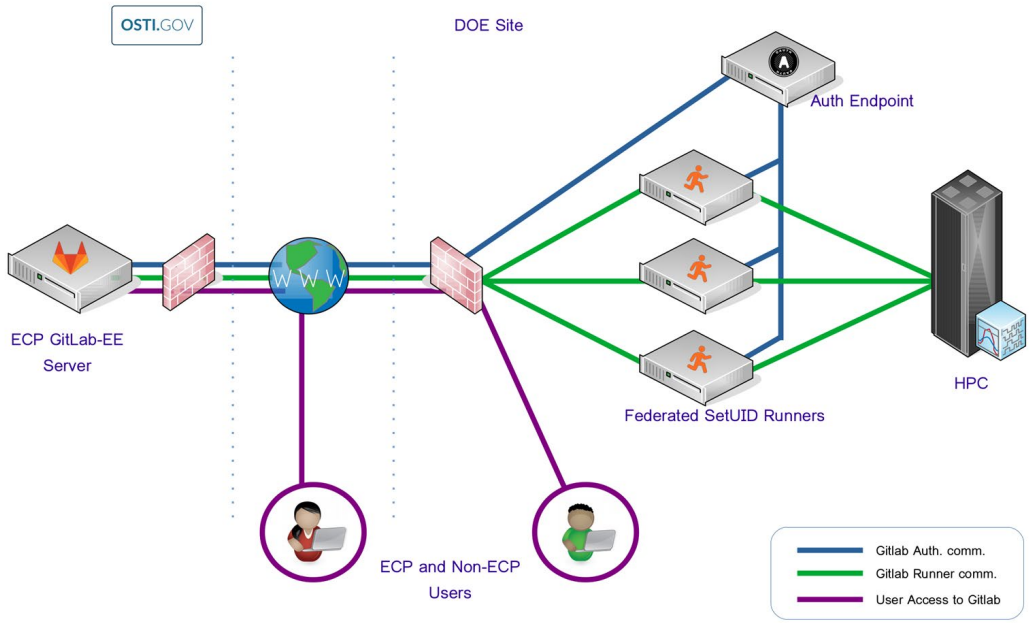
Federation

- Goal of allowing cross site CI in a secure manner while still empowering site admins with the tools to ensure policy is enforced.
  - Site Identify Providers are used and also linked to specific runners
  - Valid site credentials must be present before a job could be run at any specific site.
- Enhanced identity model that establishes a connection between a site runner, auth provider, and the user's session.
- Gitlab Omniauth as a first class citizen:
  - "...Users can choose to sign in using any of the configured mechanisms."
  - I.e. LDAP and standard Gitlab login can still function, but many providers can be added
- Ongoing communication directly with GitLab to upstream enhancements.

Deployment Models

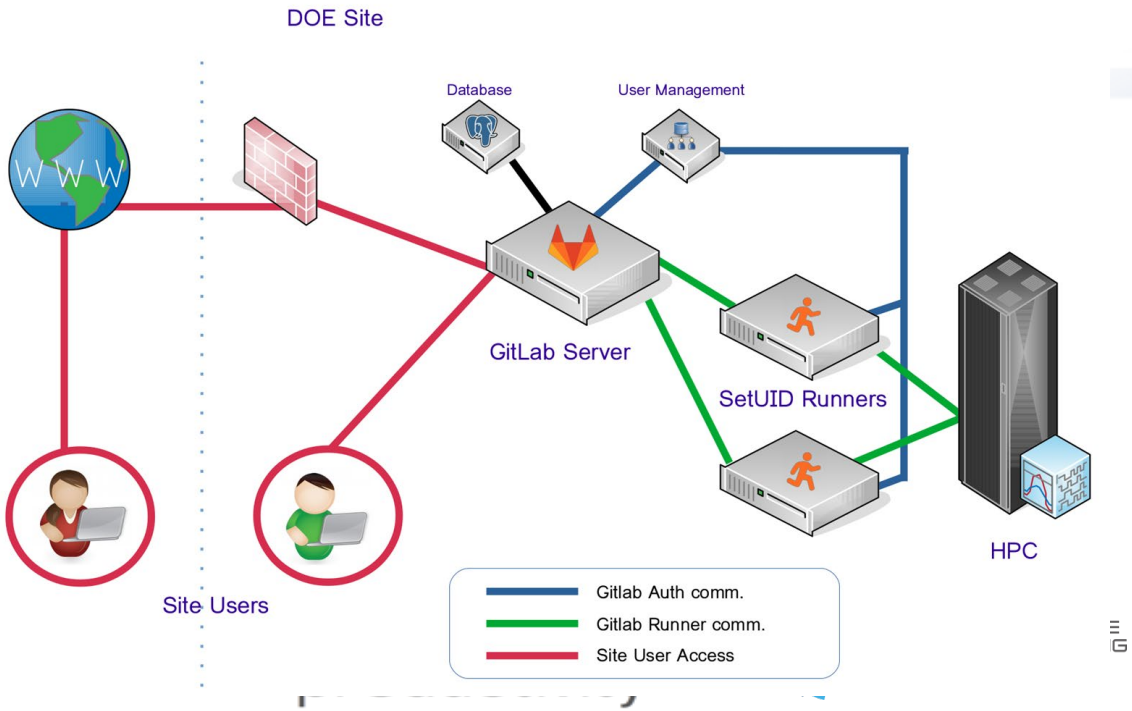
Federated - Centralized (OSTI) Sever

- Single, centrally managed Gitlab Server instance provided and administered by the OSTI team.
- Federated runners at sites
- Multi-site run capabilities



Site - Local Site Server

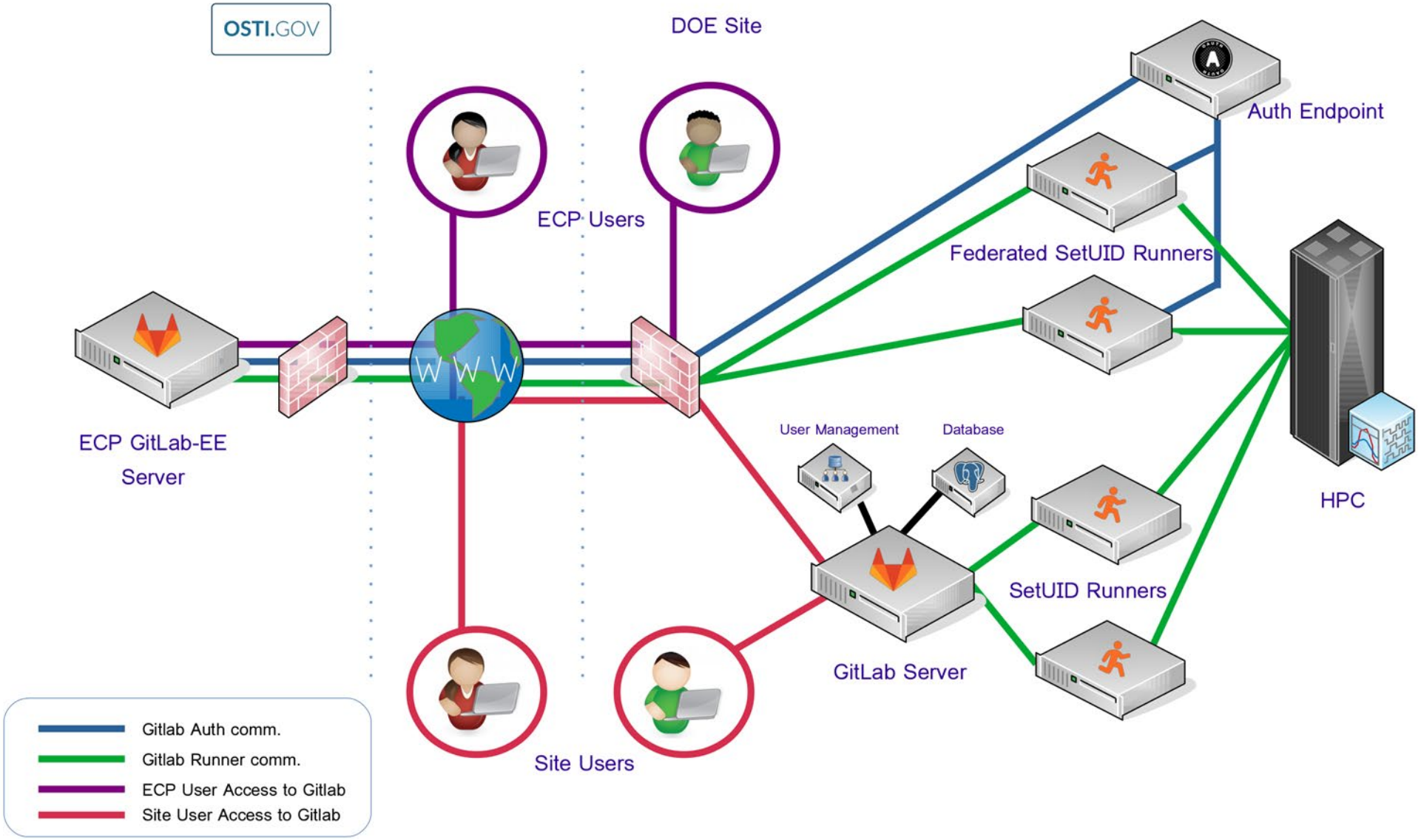
- Separate and isolated Gitlab Server provided and administered by the site's Operations team.
- Local systems/resources capacity



Deployment Models - Contd.

Federated with Site Local - Hybrid

- Most likely to be the typical deployment model for DOE facilities
- Federated capabilities
- Control over Site deployment



CI Testing Tier - HPC Resources

	Tier 0	• What AD/ST projects do now • Existing CI	• May include GitHub/Travis, existing internal systems, cron jobs, etc...
CI Cross-Site Targeting	Tier 1	• Base ECP CI - Build and Run resources • Possible 2 build and 2 run nodes • Build and Smoke tests • Run multiple builds on resource • Unit / Integration tests • Cross-site CI target	• What is ratio of build to test resources? • Work with AD and ST teams to support their needs • Possible to allocate from other HPC resources with separate scheduling policy
CI Cross-Site Facilitating	Tier 2	• Facility test resource (~10 + nodes) • In security enclave – site dependent • Larger scale tests • Facility approval for projects	• Facility managed and may want to approve projects • Possible production security constraints
	Tier 3	• Production machines • Need allocation • Production job rules • Scale tests	• Facility managed and may want to approve projects • Production security constraints

CI Testing Tier - HPC Resources

	Tier 0	• What AD/ST projects do now • Existing CI • Regression tests (no CI)	• May include GitHub/ Travis - internet • cron job based regression on misc. hardware
CI Cross-Site Targeting	Tier 1	• Base ECP CI - Build and Run resources • Possible 2 build and 2 run nodes • Build and Smoke tests • Run multiple builds on resource • Unit / Integration tests • Cross-site CI target	• What is ratio of build to test resources? • Work with AD and ST teams to support their needs • Possible to allocate from other HPC resources with separate scheduling policy
CI Cross-Site Facilitating	Tier 2	• Facility test resource (~10 + nodes) • In security enclave – site dependent • Larger scale tests • Facility approval for projects	• Facility managed and may want to approve projects • Possible production security constraints
	Tier 3	• Production machines • Need allocation • Production job rules • Scale tests	• Facility managed and may want to approve projects • Production security constraints

CI Readiness

Site readiness for OSTI hosted CI with federated runners

	Local CI	MOU	Auth Endpoint	Federated(OSTI) CI
ORNL	Available	In-process	In-process: 2/2020	
NERSC	Available	In-process	Complete	In-process
ANL	Available	In-process	In-process: 2/2020	
LANL	Available	In-process	In-process	
LLNL	Available	In-process	In-process	
SNL	Available	In-process	In-process	
NMC		In-process	In-process: 1/2020	Complete

What’s available and Where?

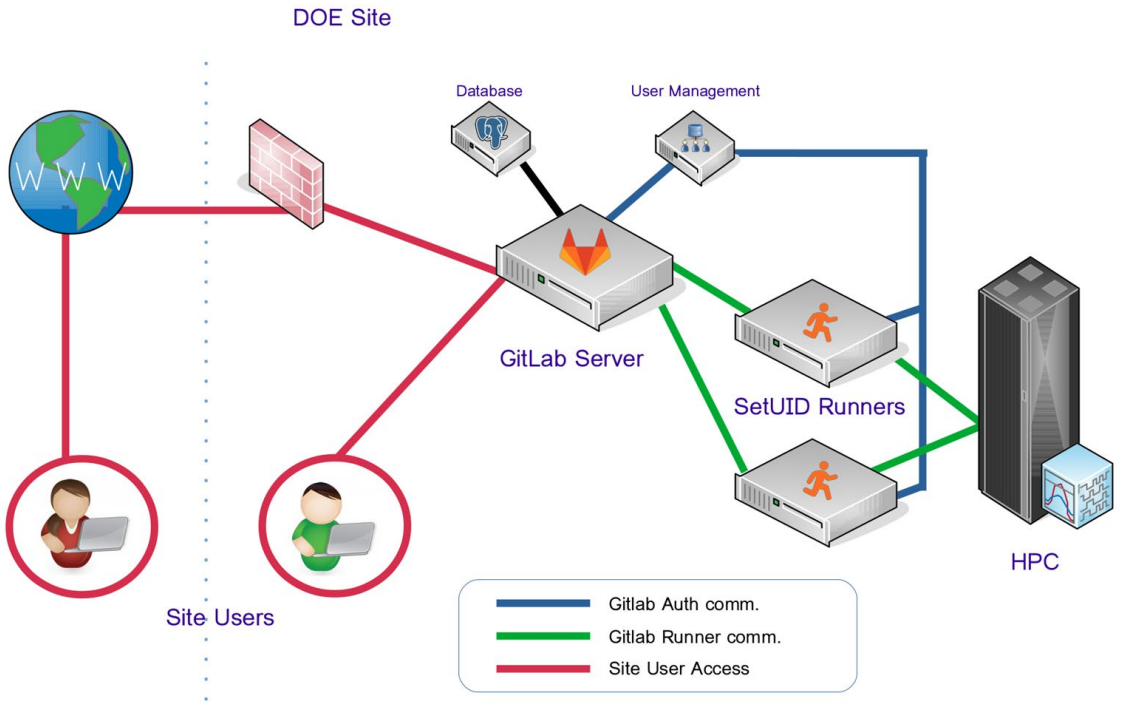
ANL
- Local Gitlab CE Server with ECP enhanced runners
- Integration with Iota(test) and Theta HPC systems

ORNL
- Local Gitlab EE Server with ECP enhanced runners
- Integration with Ascent(test) HPC system

NERSC
- Local Gitlab EE Server with ECP enhanced runners
- Integration with Cori HPC system

Other
- Local Gitlab CE Server with ECP enhanced runners
- Integration with P9, x86, and ARM systems

Local Preview Environments



Documentation

- Need to address ongoing requests for organized documentation
- https://ecp-ci.gitlab.io
  - Using GitLab Pages
  - Ongoing effort to provide value up to date information
- Goal is not to replace upstream documentation but highlight HPC focused enhancements and best-practices.
- Site specific information organized using Confluence
  - https://confluence.exascaleproject.org/display/HISD/Getting+Started+with+CI

Welcome to ECP CI Documentation

The DOE Exascale Computing Project's (ECP) mission:

Develop exascale-ready applications and solutions that address currently intractable problems of strategic importance and national interest.

Create and deploy an expanded and vertically integrated software stack on DOE HPC pre-exascale and exascale systems.

Deliver US HPC vendor technology advances and deploy ECP products to DOE HPC pre-exascale and exascale systems.

Deliver exascale simulation and data science innovations and solutions to national problems that enhance US economic competitiveness, improve our quality of life, and strengthen our national security.

The ECP Software Ecosystem is critical to reach exascale computing capabilities. A Continuous Integration (CI) testing capability was identified as a critical need to allow application and software technology projects to test across HPC architectures and software environments.

Contents

CI Integration and AD/ST Project Teams

- Effort includes the on-boarding of ECP AD/ST and E6 software projects onto the ECP/E6 CI infrastructure.
  - Derives capability from the CI Optimization effort and the CI Test Resources effort. It is focused on supporting software teams and helping them port and utilize the ECP CI infrastructure.
- Goals:
  - Helping individual teams on-board and teams
  - Developing documentation and training
  - Support development of tools to support CI teams
  - Identify limitations of CI infrastructure / resources and areas of potential growth
- Ready to support more projects at sites in a preview environment
  - We want to help teams to get running locally at the sites
  - Assist in transition to the OSTI model down the road, if desired

Agenda

Time	Module	Topic	Speaker
1:00pm-1:05pm	00	Introduction	David E. Bernholdt, ORNL
1:05pm-1:30pm	01	Overview of Best Practices in HPC Software Development	David E. Bernholdt, ORNL
1:30pm-2:00pm	02	Agile Methodologies and Useful GitHub Tools	David E. Bernholdt, ORNL
2:00pm-2:30pm	03	Improving Reproducibility through Better Software Practices	David E. Bernholdt, ORNL
2:30pm-2:45pm		Q&A	All
2:45pm-3:30pm		*Break*	
3:30pm-4:15pm	04	Software Design and Testing	David E. Bernholdt, ORNL
4:14pm-4:45pm	05	Continuous Integration	David E. Bernholdt, ORNL
4:45pm-5:00pm		Q&A	All

BACKUP:
CI HELLO WORLD

Simplest CI example

https://github.com/jrdoneal/CI_HelloWorld
https://travis-ci.org/jrdoneal/CI_HelloWorld

CI example w/ multiple platforms and specific compiler versions

https://github.com/jrdoneal/CI_Multiplatform

Code coverage, testing and CI tutorial (C++)

https://github.com/amklinv/morpheus

Code coverage, testing, and CI example (Fortran, C++)

https://github.com/jrdoneal/infrastructure

GitHub Repository Page

https://github.com/jrdoneal/CI_HelloWorld

jrdoneal / CI_HelloWorld

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

No description, website, or topics provided.

Edit

Manage topics

5 commits | 1 branch | 0 releases | 0 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

Developer D. Develop This change should lead to a correct build environment for the purpos... ··· Latest commit 93a75c4 2 days ago

.travis.yml	This change should lead to a correct build environment for the purpos...	2 days ago
README.md	Add README file to explain the intent and eventual content of this tu...	2 days ago
hello_world.sh	Add the script that tests that the build environment is correctly con...	2 days ago

Travis CI Configuration File

.travis.yml

```yaml
env:
- TRAVIS_CI_ENV="Hello, World"

#before_install:
#- Put commands here to prepare for executing builds/installs
#- Examples would be using apt-get to install dependencies not
#  included in the Travis CI build environment by default.

#install:
#- Put build commands here
#- In each phase, you can execute multiple commands
#- Travis CI stops if any single command fails in this phase

before_script:
- echo $TRAVIS_CI_ENV

script:
- $TRAVIS_BUILD_DIR/hello_world.sh
#- Travis CI will run each command in this phase even if a previous command
#  terminated in failure

after_success:
- echo "You should see that Hello, World was printed by before_script"

after_failure:
- echo "Hello, World should not have been printed by before_script"
```

The Script Phase

hello_world.sh

```
#!/bin/bash

if   [ -z "${TRAVIS_CI_ENV}" ]; then
  echo "Please set the TRAVIS_CI_ENV environment variable"
  exit 1
elif [ "${TRAVIS_CI_ENV}" != "Hello, World" ]; then
  echo "TRAVIS_CI_ENV value is ill-suited for this tutorial"
  exit 2
fi
```

Connecting GitHub & Travis CI

MY ACCOUNT

jrdoneal

Sync account

ORGANIZATIONS

You are not currently a member of any organization.

MISSING AN ORGANIZATION?

Review and add your authorized organizations.

jrdoneal

@jrdoneal

Repositories Settings

We're only showing your public repositories. You can find your private projects on travis-ci.com.

Legacy Services Integration

Filter repositories

CI_HelloWorld Settings

CI_Multiplatform Settings

infrastructure Settings

Repository in Travis CI

https://travis-ci.org/jrdoneal/CI_HelloWorld

jrdoneal / CI_HelloWorld

Current Branches Build History Pull Requests More options

master This change should lead to a correct build environment for the pu

#3 passed

Restart build

tutorial. Travis CI builds should now be successful.

Ran for 18 sec

a day ago

Commit 93a75c4

Compare ff52718..93a75c4

Branch master

jrdoneal

Ruby

TRAVIS_CI_ENV="Hello, World"

Commit History

jrdoneal / CI_HelloWorld

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights

Branch: master

Commits on Nov 3, 2018

This change should lead to a correct build environment for the purpos...
Developer D. Develop committed 2 days ago ✓

Update Travis CI configuration file so that it is a step closer to se...
Developer D. Develop committed 2 days ago ✗

.travis.yml added →

Add Travis CI configuration file. With the present content, the build
Developer D. Develop committed 2 days ago ✗

Add the script that tests that the build environment is correctly con...
Developer D. Develop committed 2 days ago

Add README file to explain the intent and eventual content of this tu...
Developer D. Develop committed 2 days ago

Travis CI Build History

Add Travis CI configuration file. With the present content, the build ...

Developer D. Develop committed 2 days ago ✕

```
1   Worker information                                                                   worker_info
6   Build system information                                                             system_info
413
414
415 Setting APT mirror in /etc/apt/sources.list: http://us-east-1.ec2.archive.ubuntu.com/ubuntu/
416
417 $ git clone --depth=50 --branch=master https://github.com/jrdoneal/CI_HelloWorld.git jrdoneal/CI_HelloWorld   git.checkout   0.54s
427 $ rvm use default                                                                    rvm   5.27s
434 $ ruby --version                                                                     ruby.versions
442 No Gemfile found, skipping bundle install
443 $ echo $TRAVIS_CI_ENV                                                                before_script   0.00s
444
445
446 $ $TRAVIS_BUILD_DIR/hello_world.sh                                                   0.00s
447 Please set the TRAVIS_CI_ENV environment variable
448
449
450 The command "$TRAVIS_BUILD_DIR/hello_world.sh" exited with 1.
451 $ echo "Hello, World should not have been printed by before_script"                 after_failure   0.00s
454
455 Done. Your build exited with 1.
                                                                                         Top
```

Travis CI Build History

Update Travis CI configuration file so that it is a step closer to se...

Developer D. Develop committed 2 days ago ✕

```
1    Worker information                                                                   worker_info
6    Build system information                                                             system_info
413
414
415  Setting APT mirror in /etc/apt/sources.list: http://us-east-1.ec2.archive.ubuntu.com/ubuntu/
416
417  $ git clone --depth=50 --branch=master https://github.com/jrdoneal/CI_HelloWorld.git jrdoneal/CI_HelloWorld   git.checkout   0.52s
427
428  Setting environment variables from .travis.yml
429  $ export TRAVIS_CI_ENV="This content will result in failure"
430
431  $ rvm use default                                                                    rvm   4.53s
438  $ ruby --version                                                                     ruby.versions
446  No Gemfile found, skipping bundle install
447  $ echo $TRAVIS_CI_ENV                                                                before_script   0.00s
448  This content will result in failure
449
450  $ $TRAVIS_BUILD_DIR/hello_world.sh                                                   0.00s
451  TRAVIS_CI_ENV value is ill-suited for this tutorial
452
453
454  The command "$TRAVIS_BUILD_DIR/hello_world.sh" exited with 2.
455  $ echo "Hello, World should not have been printed by before_script"                 after_failure   0.00s
458
459  Done. Your build exited with 1.
```

Travis CI Build History

This change should lead to a correct build environment for the purpos... ···

Developer D. Develop committed 2 days ago ✓

```
1   Worker information                                                                    worker_info
6   Build system information                                                              system_info
413
414
415 Setting APT mirror in /etc/apt/sources.list: http://us-east-1.ec2.archive.ubuntu.com/ubuntu/
416
417 $ git clone --depth=50 --branch=master https://github.com/jrdoneal/CI_HelloWorld.git jrdoneal/CI_HelloWorld   git.checkout   0.53s
427
428 Setting environment variables from .travis.yml
429 $ export TRAVIS_CI_ENV="Hello, World"
430
431 $ rvm use default                                                                     rvm   4.69s
438 $ ruby --version                                                                      ruby.versions
446 No Gemfile found, skipping bundle install
447 $ echo $TRAVIS_CI_ENV                                                                 before_script   0.00s
448 Hello, World
449
450 $ $TRAVIS_BUILD_DIR/hello_world.sh                                                    0.00s
451
452
453 The command "$TRAVIS_BUILD_DIR/hello_world.sh" exited with 0.
454 $ echo "You should see that Hello, World was printed by before_script"               after_success   0.00s
457
458 Done. Your build exited with 0.
```

! →