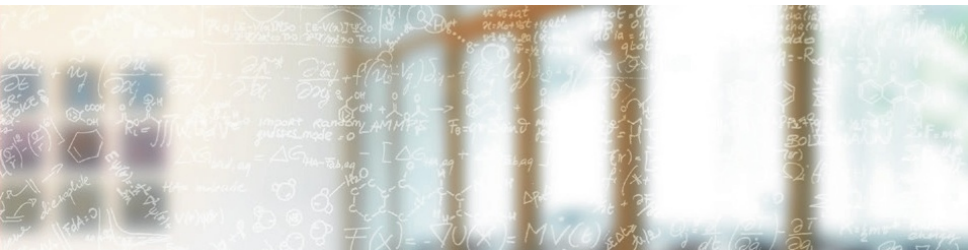




CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Continuous Integration & Regression Testing

CSCS User Lab Day - Meet the Swiss National Supercomputing Centre

Guilherme Peretti-Pezzi, Theofilos Manitaras, Vasileios Karakasis

September 11, 2018

Continuous Integration



- Introducing the CSCS CI service
- Using the CSCS CI service
- Polling a remote repository
- Github pull request builder (ghprb) plugin



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Introducing the CSCS CI Service

Why should you use the CI service?

- The process of quickly integrating new features into software is called continuous integration (CI).
- According to modern software development practices, various tools which automate CI are used, e.g. Jenkins, Travis CI, CircleCI.
- By embracing CI, bugs introduced by code changes are caught before they are actually merged to the master branch.
- CSCS has recently started offering a CI service based on Jenkins to its users, as announced at the PASC 2018 conference.
- It is crucial that HPC applications are tested on the actual system they are going to be executed.

Jenkins



Jenkins

- **Jenkins** is an open source automation server which helps to automate the non-human part of the software development process, with continuous integration and facilitating technical aspects of continuous delivery.
- It offers a large number of plugins which enhance its capabilities.
- Jenkins slaves are configured so that the builds take place on the compute nodes of Piz Daint and therefore software is tested on the actual hardware/software.

Gaining access to the CSCS CI service

In order to be granted access to the CSCS CI service, a principal investigator (PI) responsible for a project including software development has to open a ticket at help@cscs.ch and make the request.

From the CSCS Jenkins instance side, the following apply:

- Each project is assigned a Jenkins folder with the same name on the Jenkins instance.
- The Jenkins jobs related to the project have to be created in the above folder.
- Credentials can be added to be used with version control systems, etc.
- Each project is assigned a Jenkins node(slave) to run the corresponding Jenkins jobs.
- A Jenkins user which is going to be used by the Jenkins node to access Piz Daint also added.



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Using the CSCS CI service

Logging to the Jenkins Web interface (1/2)

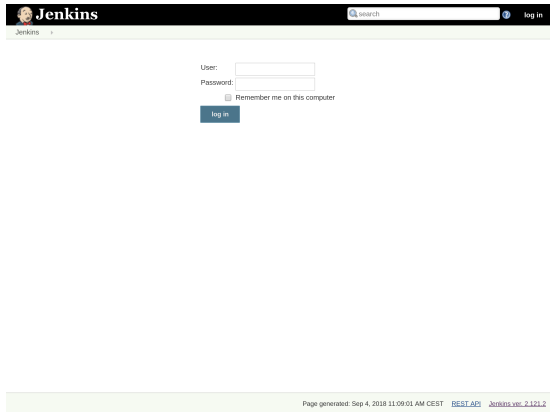
- The Jenkins web interface provided by CSCS is not available in public. In order to be able to access it, a local port forwarding must be performed with ssh. Thus the user has to forward a local port to **lisone.cscs.ch:443** via **ela.cscs.ch**.
- For Linux/Mac users this can be performed from the shell with the following command:

```
$ ssh -L 7000:lisone.cscs.ch:443 ela.cscs.ch
```

This way, the user can access the web interface from a web browser visiting <https://ci.cscs.ch:7000/>. Note that the number of the local port (here 7000) is chosen by the user.

Logging to the Jenkins Web interface (2/2)

Use your CSCS account credentials to login.



The screenshot shows the Jenkins web interface. At the top, there is a black header bar with the Jenkins logo (a person icon) and the word "Jenkins" on the left. On the right side of the header, there is a search bar with the placeholder text "search", a help icon, and a "log in" link. Below the header, the main content area is white. It features a "User:" label followed by a text input field. Below that is a "Password:" label followed by another text input field. Under the password field, there is a checkbox labeled "Remember me on this computer". At the bottom of the login section, there is a blue button with the text "log in". At the very bottom of the page, there is a green footer bar containing the text "Page generated: Sep 4, 2018 11:09:01 AM CEST", a link to the "REST API", and the version "Jenkins ver. 2.121.2".

Dedicated folder and Jenkins slave

Every project has access to a dedicated folder and a corresponding Jenkins slave

The screenshot shows the Jenkins web interface. On the left is a sidebar with navigation links. The main area displays a table of jobs. The first job is 's299', which is in a 'Building' state (indicated by a sun icon). Below the table, there are two panels: 'Build Queue' and 'Build Executor Status'.

S	W	Name	Last Success	Last Failure	Last Duration	Fav
		s299	N/A	N/A	N/A	
		s299	N/A	N/A	N/A	
		s299	N/A	N/A	N/A	
		s299	N/A	N/A	N/A	

Build Queue
No builds in the queue.

Build Executor Status

- master
 - 1 Idle
 - 2 Idle
- s299_daintvm1 (offline)
 - 1 Offline
 - 2 Offline
 - 3 Offline
 - 4 Offline

Creating a new Jenkins project

Enter an item name

» This field cannot be empty, please enter a valid name



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



External Job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Bitbucket Team/Project

Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



GitHub Organization

Scans a GitHub organization (or user account) for all repositories matching some defined markers.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

Best practices

- Adopt the pipeline-as-code modern approach of Jenkins 2 and include the **Jenkinsfile** in your git remote.
- The actual build jobs have to be submitted via sbatch in the job queue. Use of srun is not allowed.
- The `--wait` option should be used when submitting sbatch jobs, else sbatch returns immediately after job submission. Using:

```
$ sbatch --wait <batch_script>
```

forces sbatch to wait for the submitted job to complete before returning.

- For 1-node jobs, it is good practice to use the **cscsci** partition which offers higher priority and is suitable for ci jobs.
- Copy the build output/errors on **SCRATCH** to have access to it.
- Make use of artifacts to store the output/error files.



CSCS

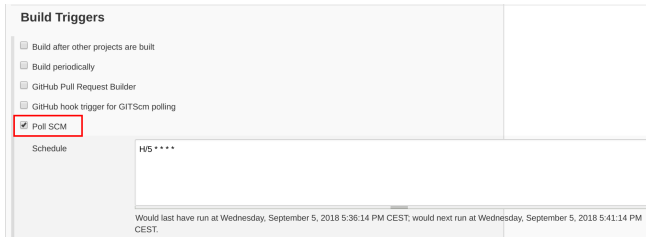
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Polling a remote repository

Polling a remote repository

- Since the CSCS Jenkins instance is not accessible from the public web, Jenkins has to poll the source control repository to be made aware of any changes made to the specified remote and branch and start a job.
- The above option is enabled by using the **Poll SCM** option under Build Triggers:



Build Triggers

- ☐ Build after other projects are built
- ☐ Build periodically
- ☐ GitHub Pull Request Builder
- ☐ GitHub hook trigger for GITScm polling
- ☒ Poll SCM

Schedule: H/5 * * * *

Would last have run at Wednesday, September 5, 2018 5:36:14 PM CEST; would next run at Wednesday, September 5, 2018 5:41:14 PM CEST.



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Github pull request builder (ghprb) plugin

Enabling the ghprb (1/2)

1. Invite the **jenkins-cscs** Github user which belongs to the CSCS UES group and is used by the CSCS jenkins instance. The above user has to be invited with Read & Write privileges.
2. Enable ghprb in your project:

☒ GitHub project

Project url

☐ Pipeline speed/durability override

☐ Preserve stashes from completed builds

☐ Restrict build execution causes

☐ This project is parameterized

☐ Throttle builds

Build Triggers

☐ Build after other projects are built

☐ Build periodically

☒ GitHub Pull Request Builder

GitHub API credentials

Advanced...

Enabling the ghprb (2/2)

3. Set the advanced settings according to your needs:

Use github hooks for build triggering	<input type="checkbox"/>
Trigger phrase	<code>.*test\W+this\W+please.*</code>
Only use trigger phrase for build triggering	<input type="checkbox"/>
Close failed pull request automatically?	<input type="checkbox"/>
Skip build phrase	<code>.*\[skip\W+ci\].*</code>
Display build errors on downstream builds?	<input type="checkbox"/>
Crontab line	<code>*/*1 * * * *</code>

A build is triggered from a new pull request


The screenshot shows a GitHub Pull Request (PR) interface. At the top, there's a navigation bar with tabs for Code, Issues, Pull requests (selected), Projects, Wiki, Insights, and Settings. Below this, the PR title is 'Add hostname command #4'. A green 'Open' button is next to the title. Below the title, it says 'teojgo wants to merge 1 commit into master from feature/add_hostname'. There are tabs for Conversation (0), Commits (1), Checks (0), and Files changed (1). The main content area shows a comment from 'teojgo' 6 days ago with the text 'No description provided.' Below the comment, there's a commit 'Add hostname command' by 'teojgo' 6 days ago. At the bottom, there's a section for checks. It shows 'All checks have passed' with 1 successful check. Below this, there's a check for 'CSCS CI' which is also successful. At the bottom, there's a message 'This branch has no conflicts with the base branch'.

<> Code ① Issues 0 **🔗 Pull requests 1** 📁 Projects 0 📖 Wiki 📊 Insights ⚙️ Settings


Add hostname command #4


Open teojgo wants to merge 1 commit into `master` from `feature/add_hostname`

💬 Conversation 0 ➡️ Commits 1 🛠️ Checks 0 📄 Files changed 1


 teojgo commented 6 days ago Member + 👤 ...


No description provided.

🔗  Add hostname command ✓ 22ffbd5

👤  teojgo self-assigned this 6 days ago

Add more commits by pushing to the **feature/add_hostname** branch on **eth-cscs/UserLabDay**.

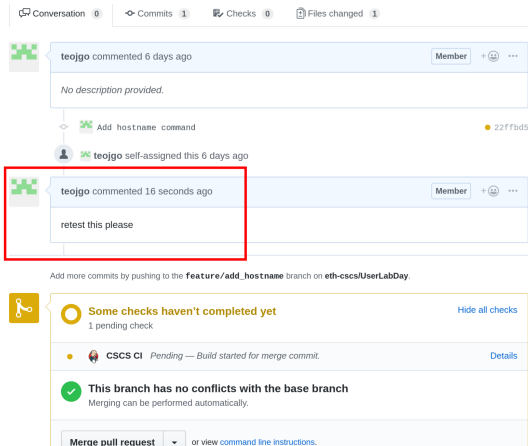
 ✓ **All checks have passed** Hide all checks
1 successful check

✓  **CSCS CI** Details

✓ **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Retriggering a build

To retrigger a build for an already submitted pull request, an admin or whitelisted user has to make a comment matching a predefined pattern. In this case "retest this please".



The screenshot displays a GitHub pull request interface. At the top, navigation tabs show 'Conversation' (0), 'Commits' (1), 'Checks' (0), and 'Files changed' (1). The main content area shows a comment by user 'teojgo' from 6 days ago, with the text 'No description provided.' Below this, a commit 'Add hostname command' by 'teojgo' is shown. A second comment by 'teojgo' from 16 seconds ago, with the text 'retest this please', is highlighted with a red rectangular box. Below the comments, a status bar indicates 'Add more commits by pushing to the feature/add_hostname branch on eth-cscs/UserLabDay.' The bottom section shows build status: 'Some checks haven't completed yet' (1 pending check) and 'This branch has no conflicts with the base branch'. A 'Merge pull request' button is visible at the bottom.

Conversation 0 Commits 1 Checks 0 Files changed 1

teojgo commented 6 days ago

No description provided.

Add hostname command

teojgo self-assigned this 6 days ago

teojgo commented 16 seconds ago

retest this please

Add more commits by pushing to the `feature/add_hostname` branch on `eth-cscs/UserLabDay`.

Some checks haven't completed yet
1 pending check

CSCS CI Pending — Build started for merge commit.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request or view command line instructions.

Pull request of a non-whitelisted user

When a user who is not whitelisted in ghprb submits a pull request, an admin verification is needed to trigger the build.

Search or jump to... Pull requests Issues Markets

eth-cscs / UserLabDay

< Code Issues 0 Pull requests 2 Projects 0 Wiki

Test PR #5

Open rsarm wants to merge 1 commit into master from test-branch

Conversation 0 Commits 1 Checks 0 Files changed 1

rsarm commented 33 seconds ago

No description provided.

Test commit

jenkins-cscs commented just now

Can one of the admins verify this patch?

Add more commits by pushing to the test-branch branch on eth-cscs/UserLabDay.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request or view command line instructions.

rsarm commented 5 minutes ago

No description provided.

Test commit

jenkins-cscs commented 4 minutes ago

Can one of the admins verify this patch?

teoigo commented 37 seconds ago

test this please

Add more commits by pushing to the test-branch branch on eth-cscs/UserLabDay.

Some checks haven't completed yet

1 pending check

- CSCS CI Pending — Build triggered for merge commit.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request or view command line instructions.

Write Preview AA B i “ < >

Useful links

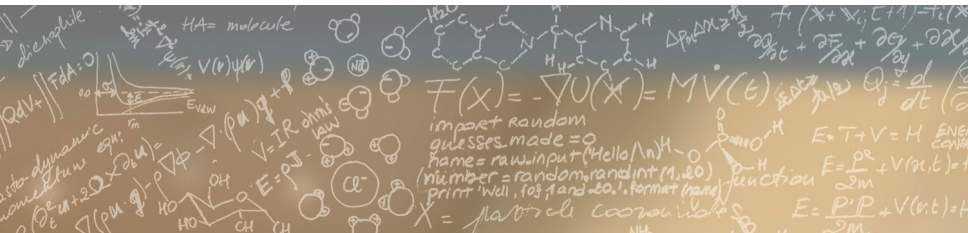
- [CSCS CI documentation](#)
- [User lab day GitHub repository](#)



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for your attention.