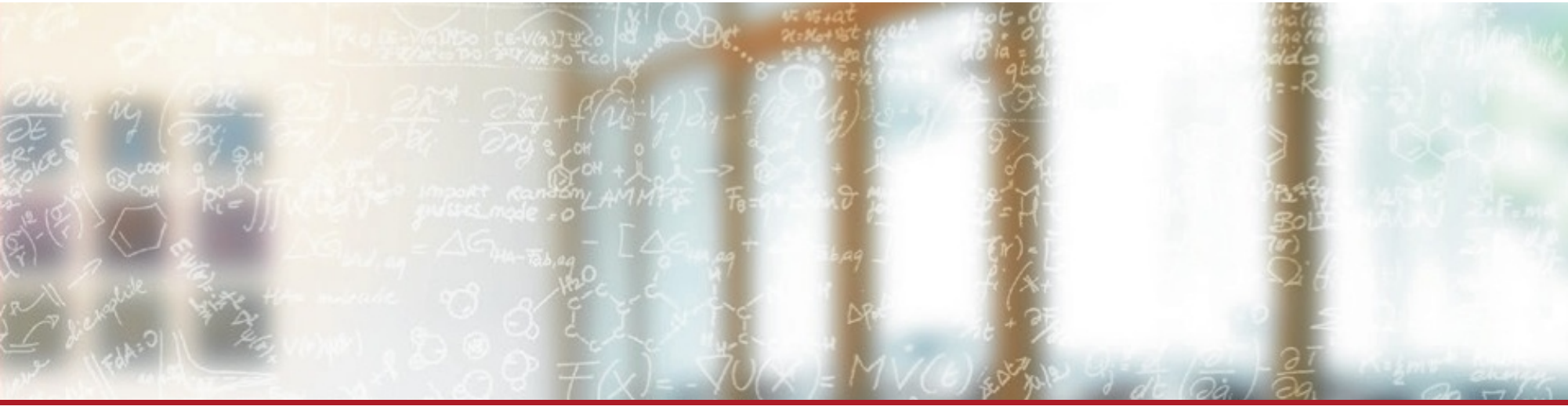
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



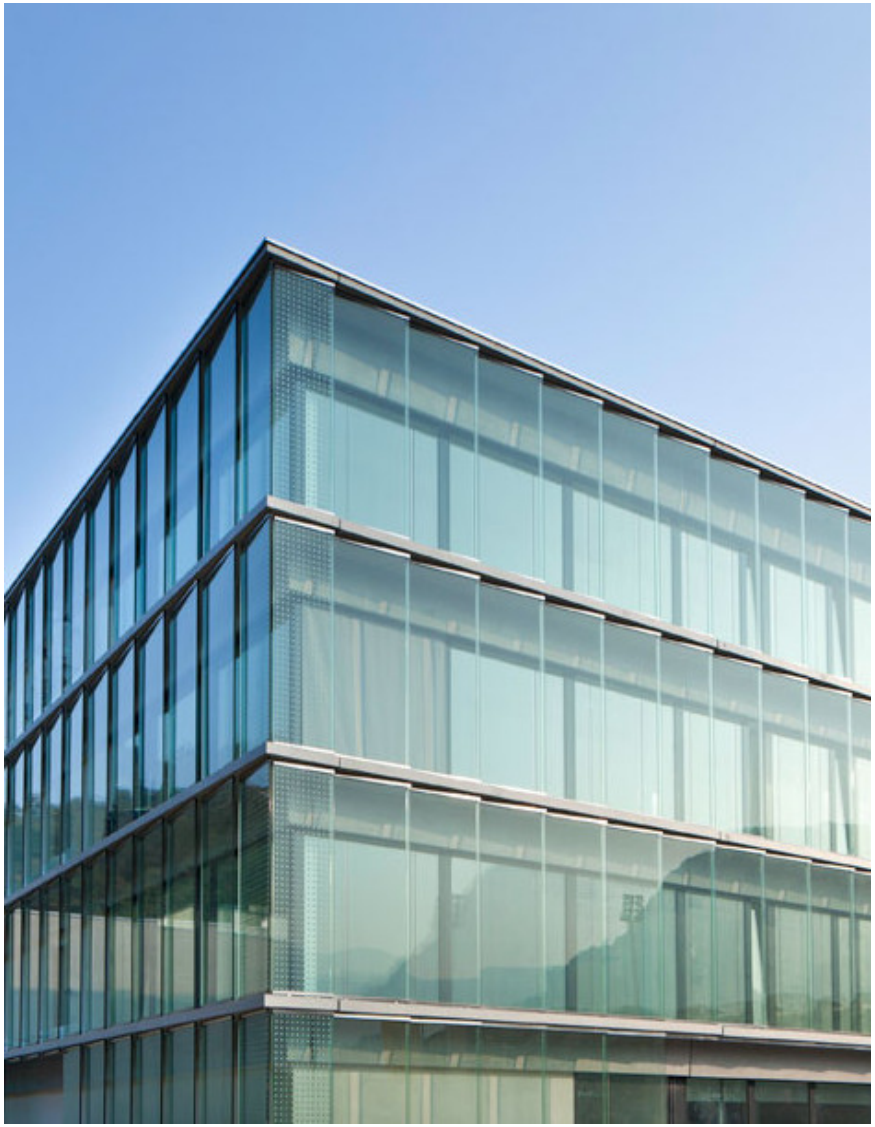
Continuous Integration & Regression Testing

CSCS User Lab Day - Meet the Swiss National Supercomputing Centre

Theofilos Manitaras, Guilherme Peretti-Pezzi, 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 annouced 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 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 CSCI 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 accessible from the public web. 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
```

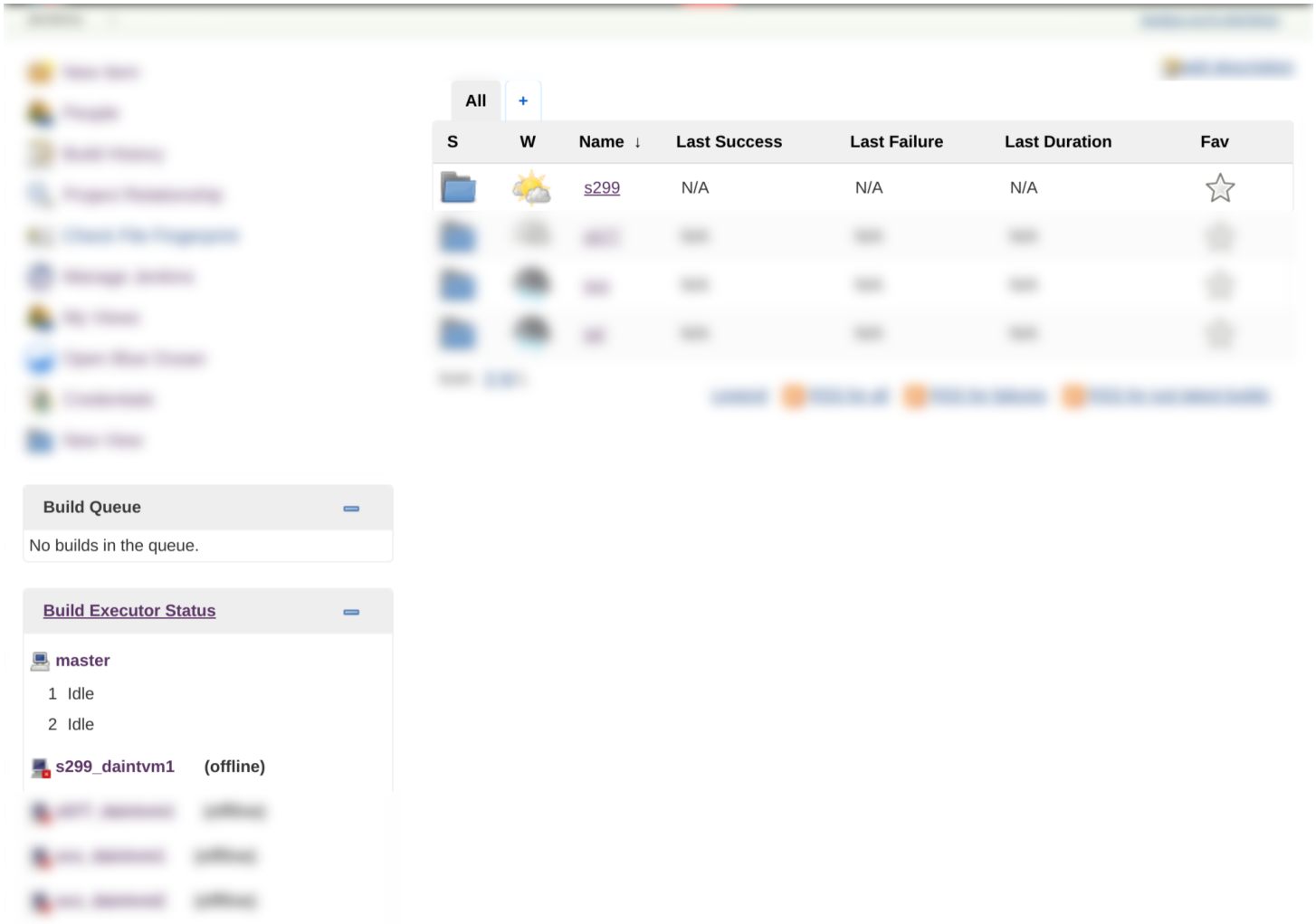
  This way, the user can access the web interface from a web browser visiting `https://ci.cscs.ch:7000/`. Note that the number of the local port (here 7000) is chosen by the user.

Logging to the Jenkins Web interface (2/2)

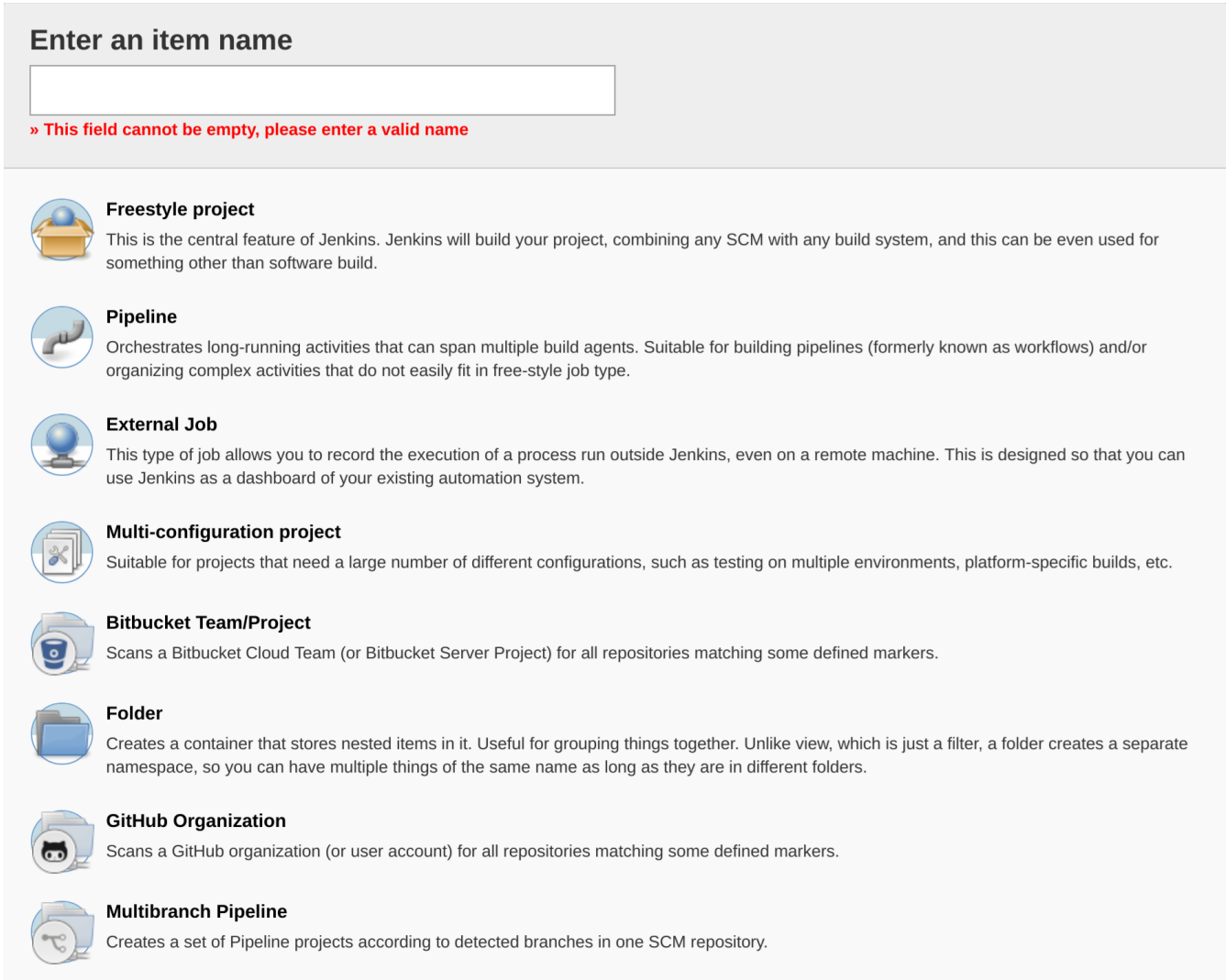
Use your CSCS account credentials to login.

Dedicated folder and Jenkins slave

Every project has access to a dedicated folder and a corresponding Jenkins slave which is a virtual machine and not a login node of Piz Daint.



Creating a new Jenkins project



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:

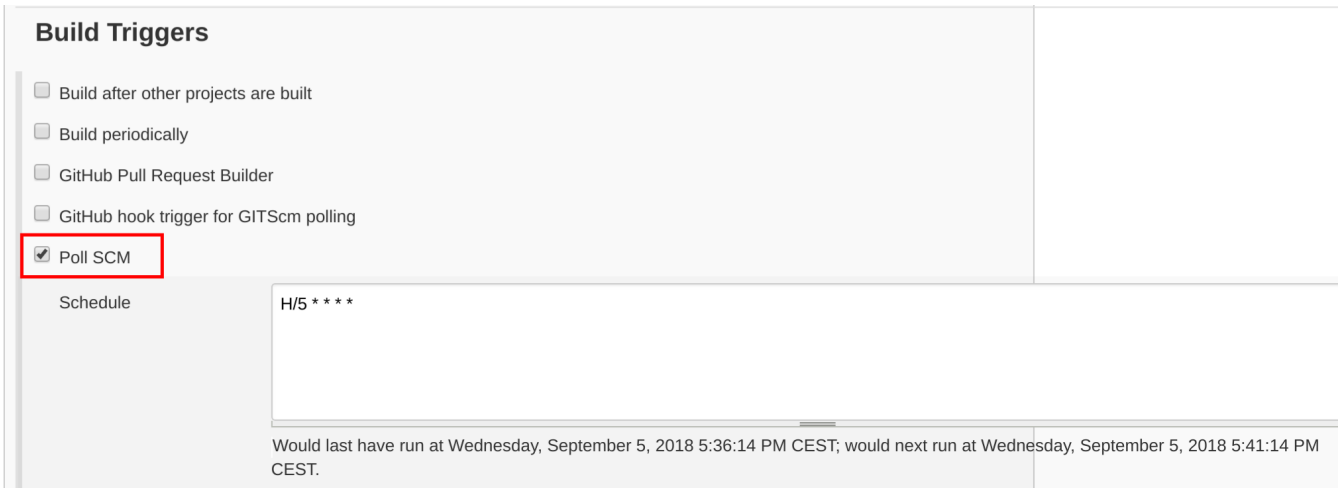
```
$ sbatch --wait <batch_script>
```

  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.

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:



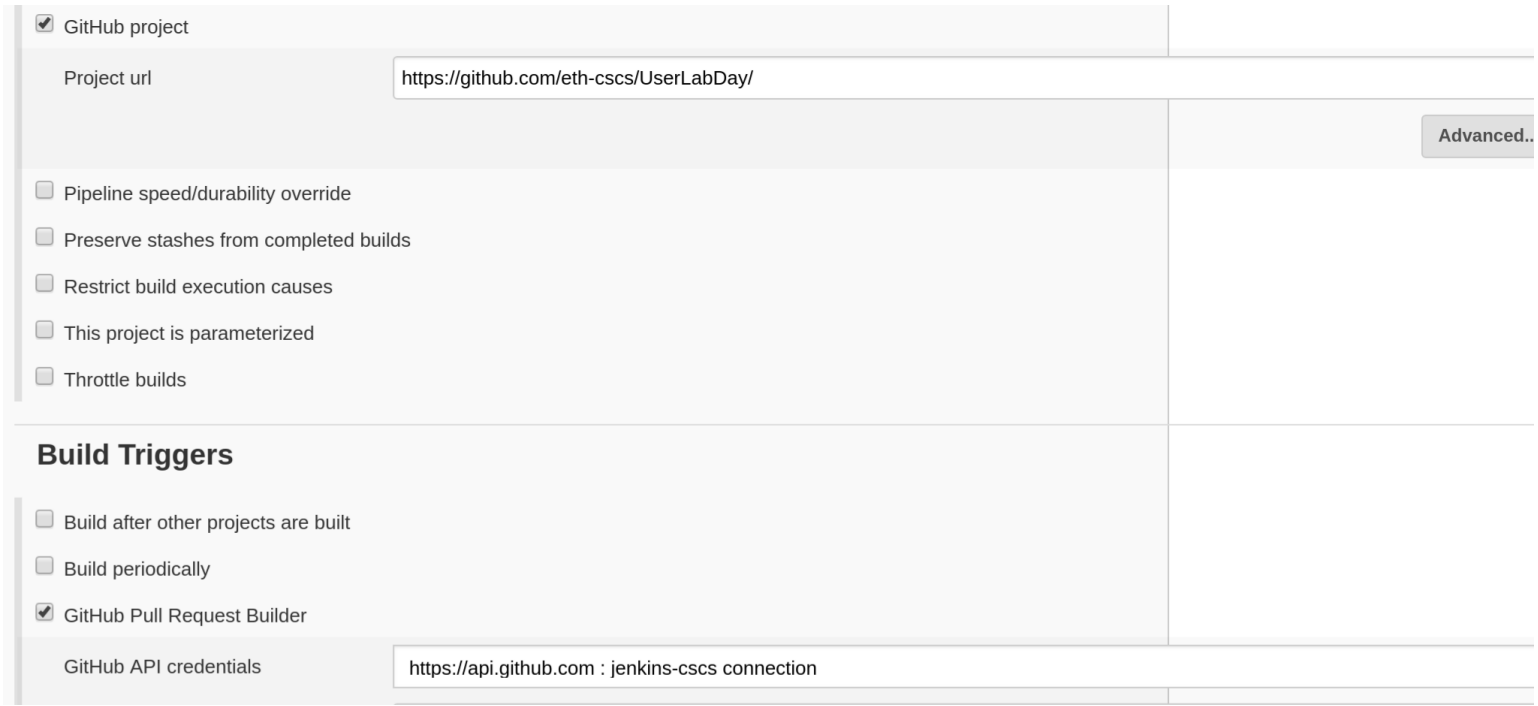
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Github pull request builder (ghprb) plugin

Enabling the ghprb (1/2)

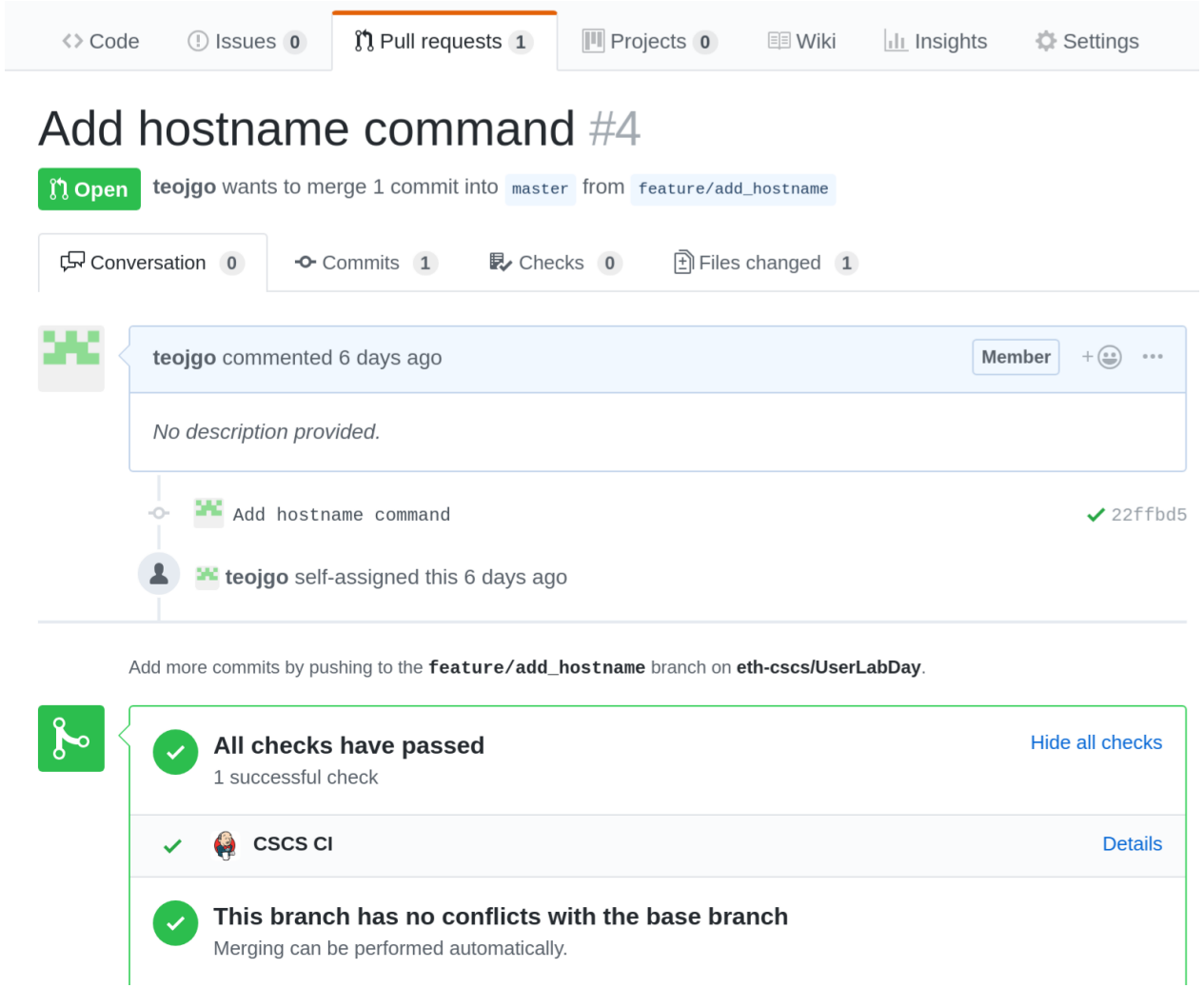
1. Invite the **jenkins-cscs** Github user which belongs to the CSCS UES group and is used by the CSCS jenkins instance. The above user has to be invited with Read & Write privileges.
2. Enable ghprb in your project:



Enabling the ghprb (2/2)

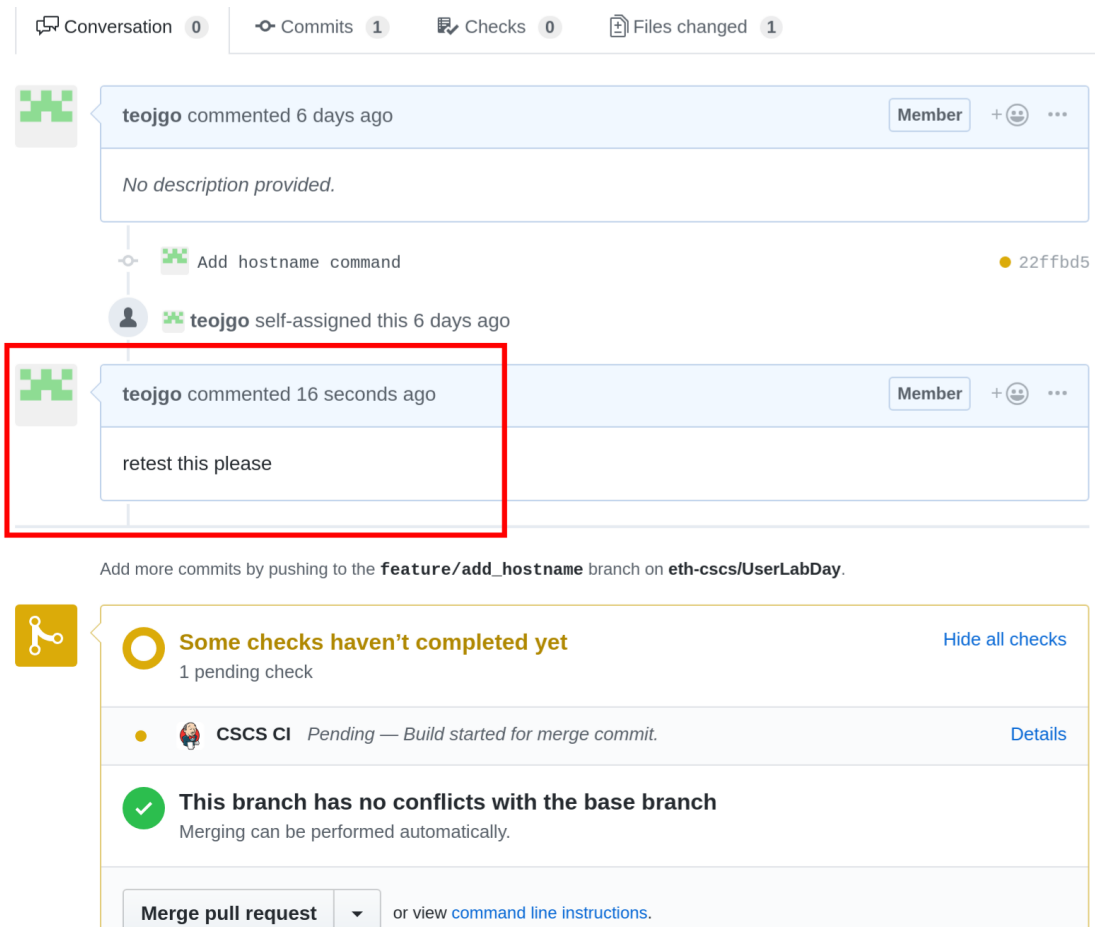
3. Set the advanced settings according to your needs:

A build is triggered from a new pull request



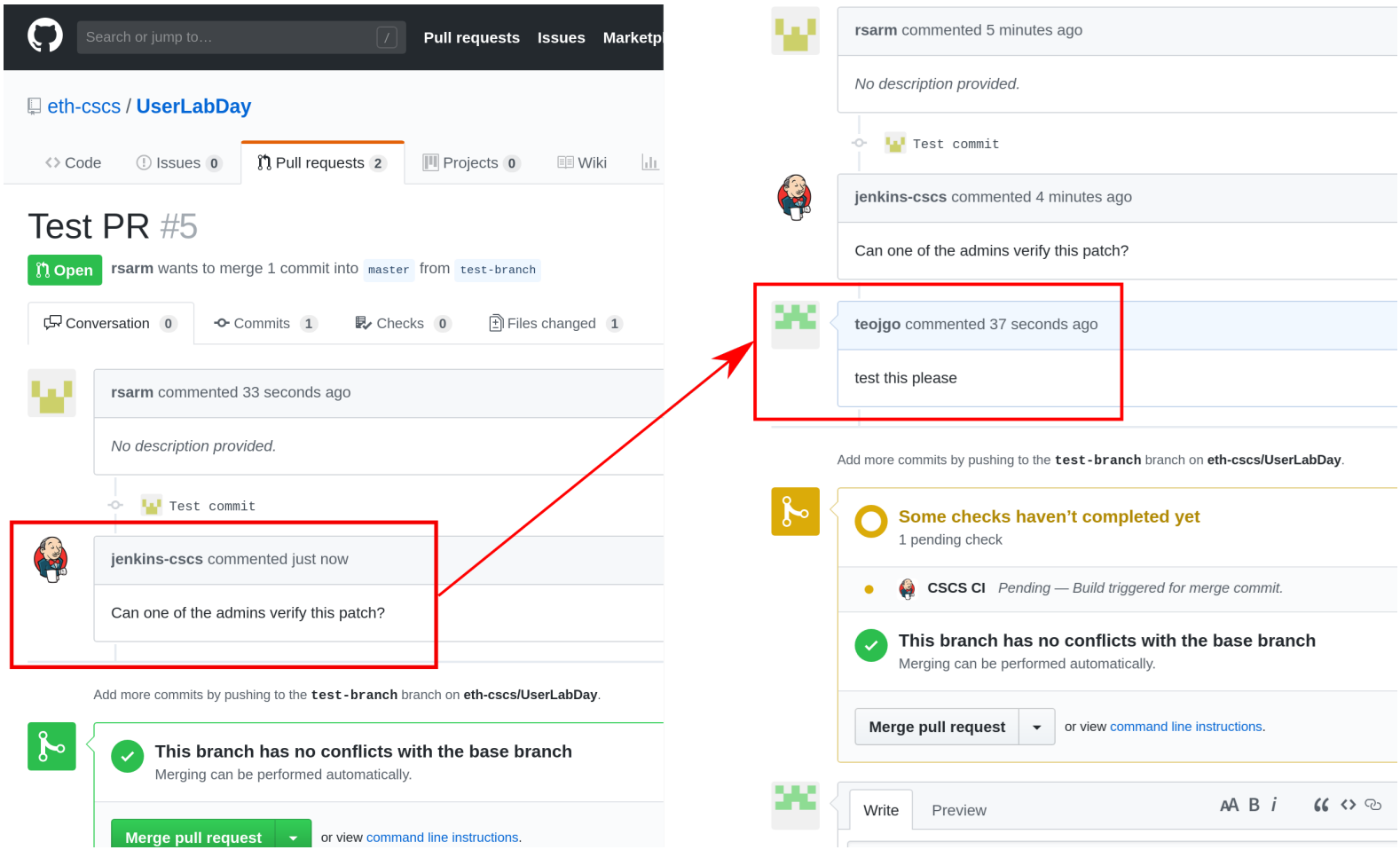
Retriggering a build

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”.



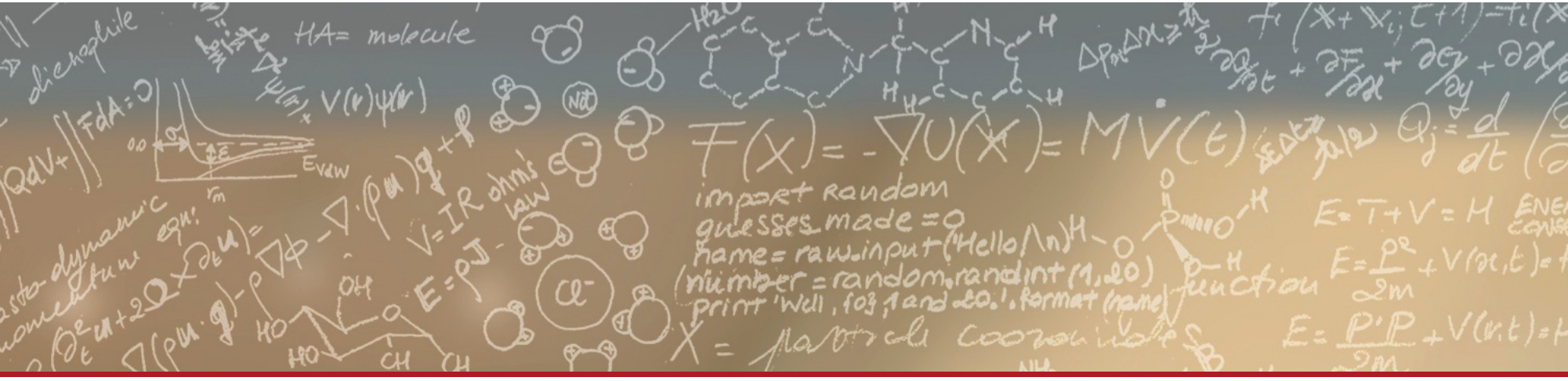
Pull request of a non-whitelisted user

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



Useful links

- CSCS CI documentation
- User lab day GitHub repository



Thank you for your attention.