8th Annual Earth System Grid Federation

ESGF-test-suite
ESGF-jenkins

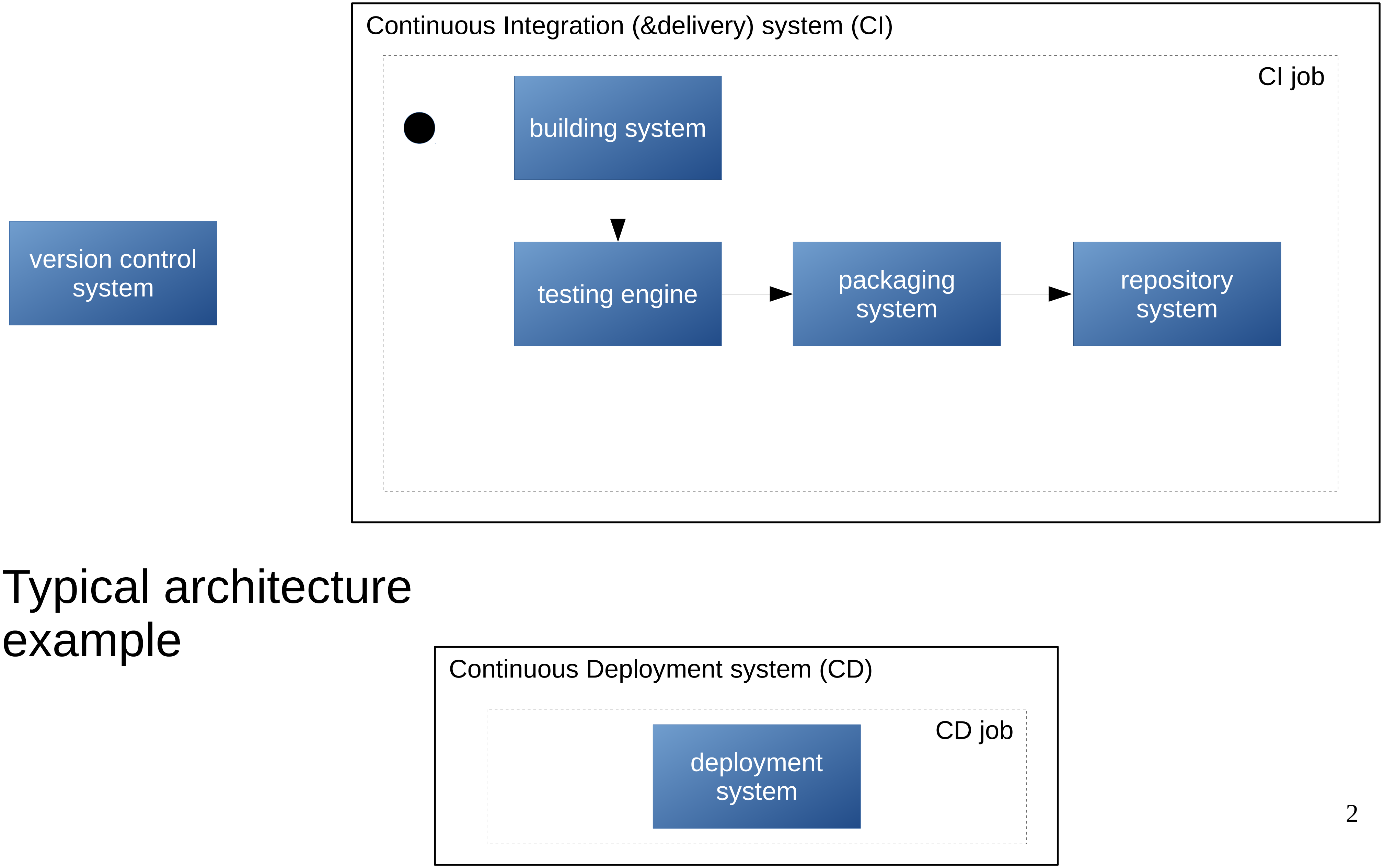
Sébastien
Gardoll
IPSL, software engineer
sgardoll@ipsl.fr

Dec 6th, 2018

Sciences de l'environnement
Institut Pierre Simon Laplace
is-enes
INFRASTRUCTURE FOR THE EUROPEAN NETWORK FOR EARTH SYSTEM MODELLING

Continuous Integration (CI)

Goal: automation of software production.

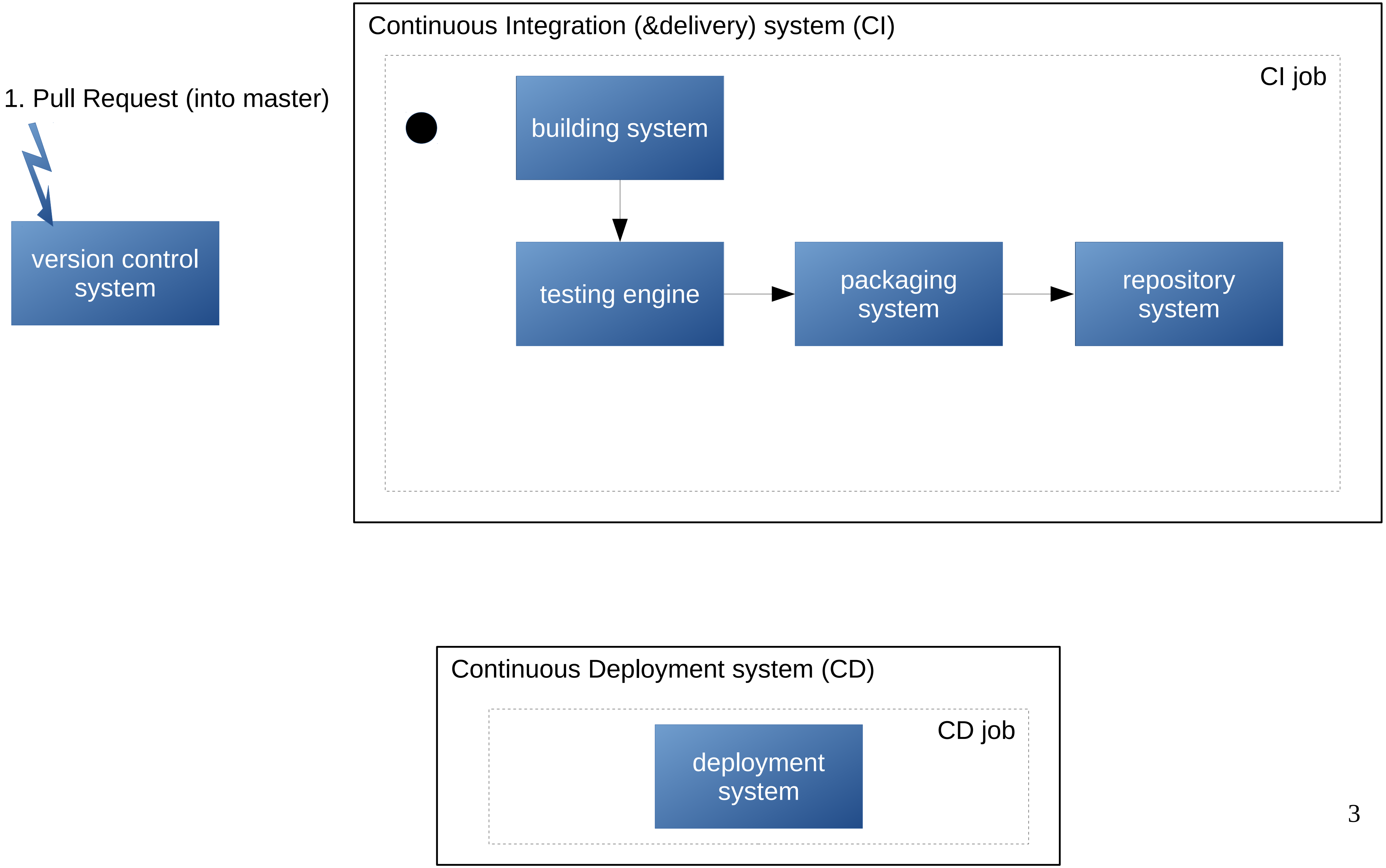


Typical architecture example

1/4

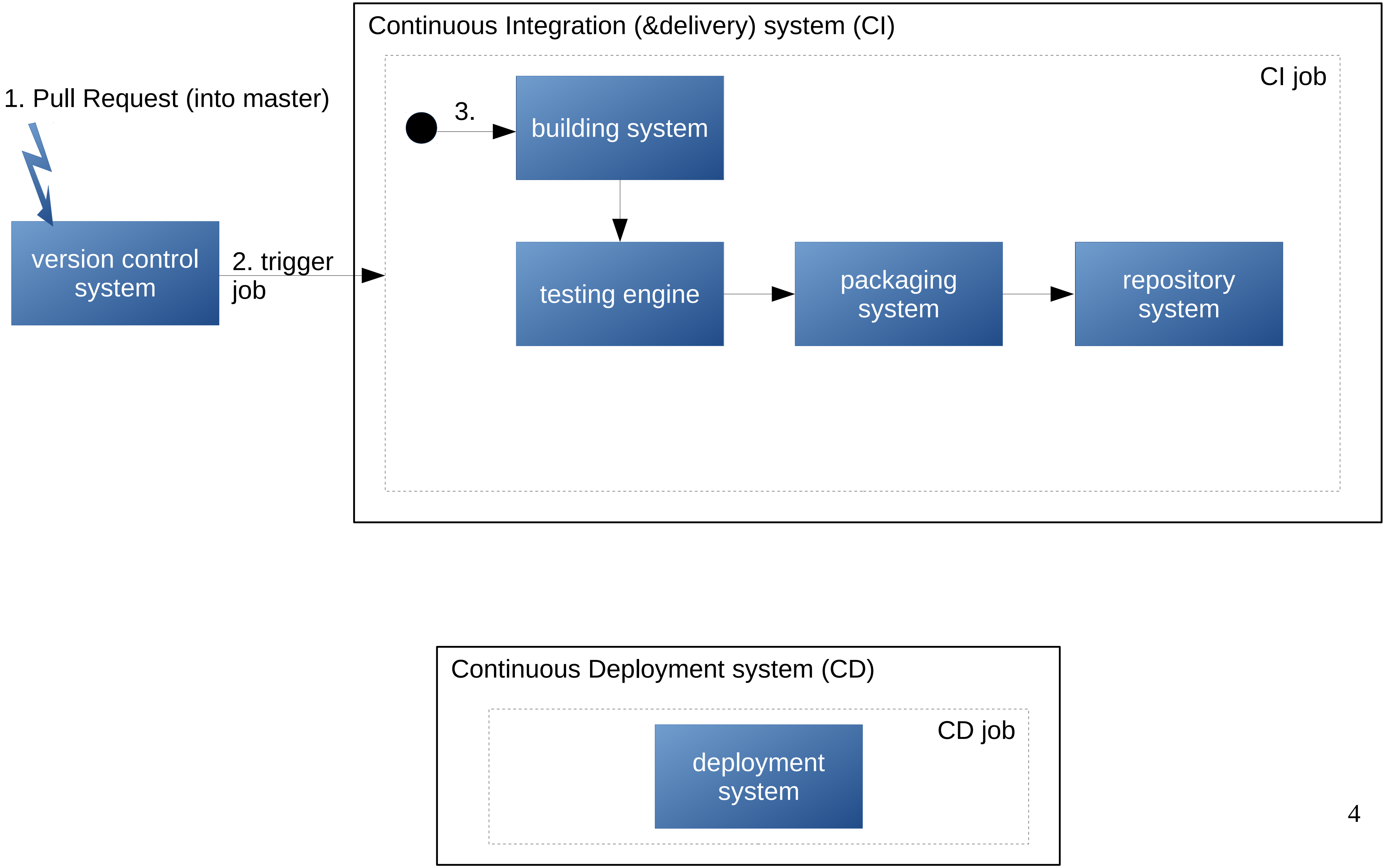
Continuous Integration (CI)

Typical case of CI workflow: management of Pull Requests.



Continuous Integration (CI)

PR pending until the successful completion of the CI job.



1. Pull Request (into master)

version control system

2. trigger job

Continuous Integration (&delivery) system (CI)

CI job

3.

building system

testing engine

packaging system

repository system

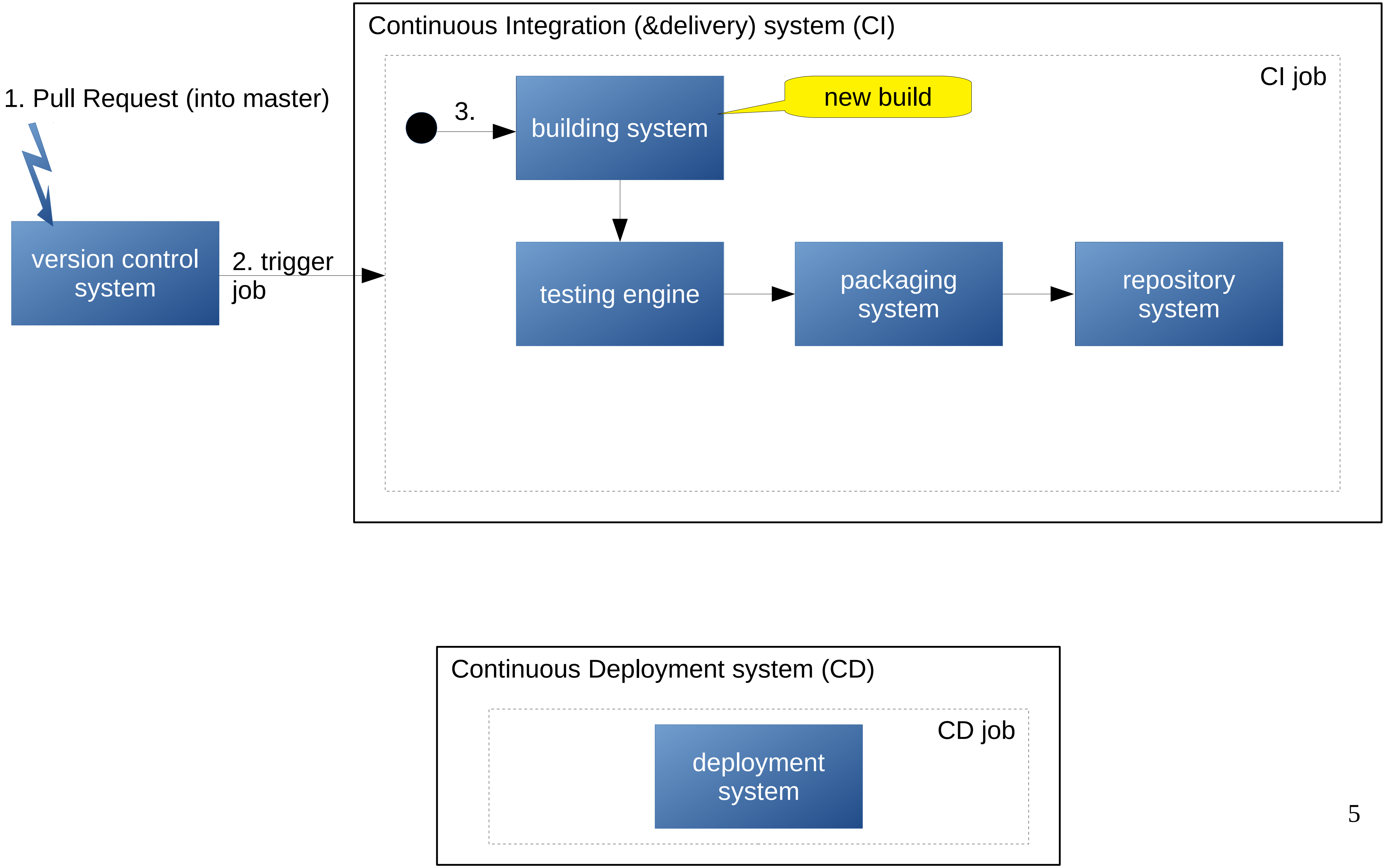
Continuous Deployment system (CD)

CD job

deployment system

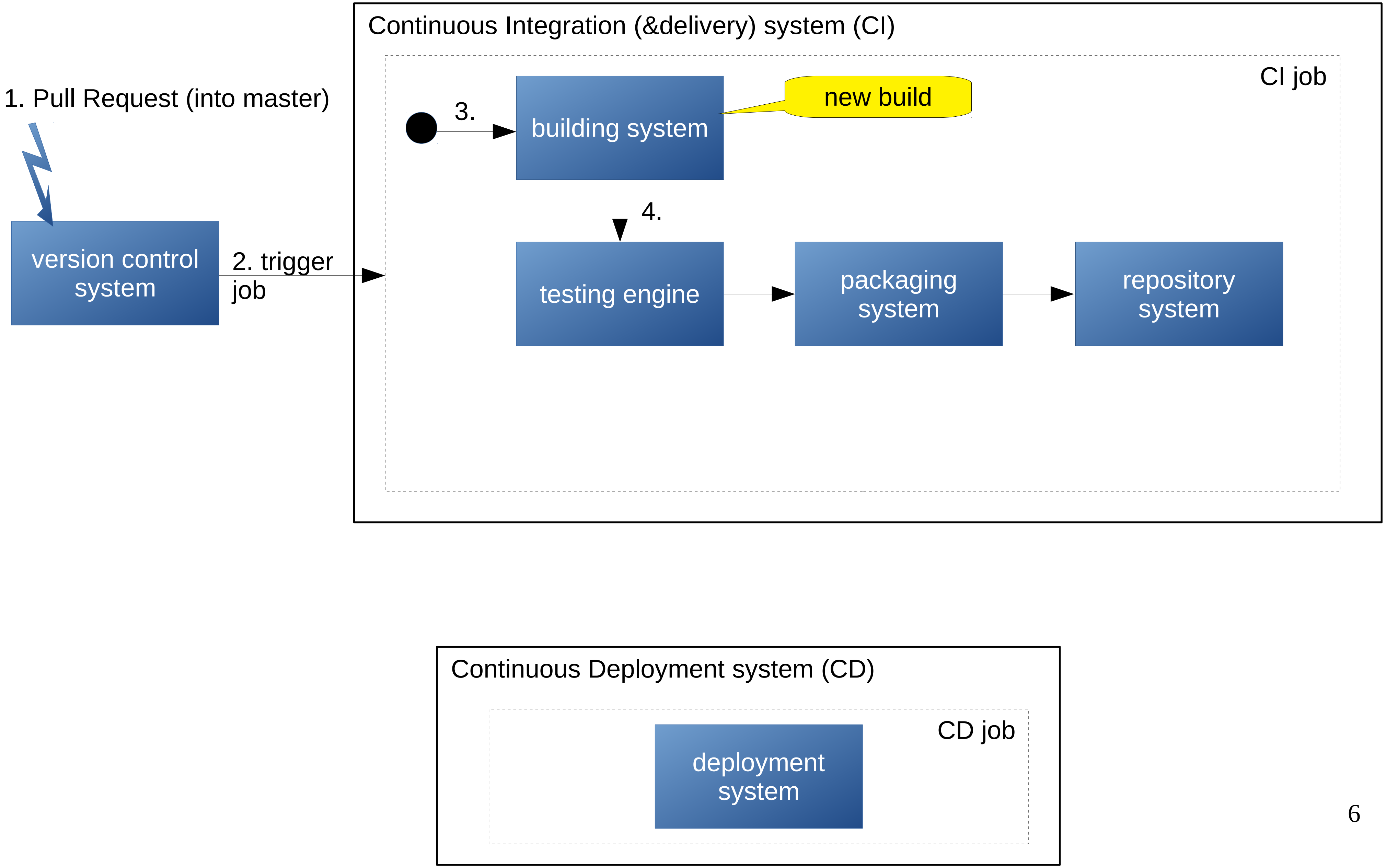
Continuous Integration (CI)

The application is built including the modifications introduced by the PR.



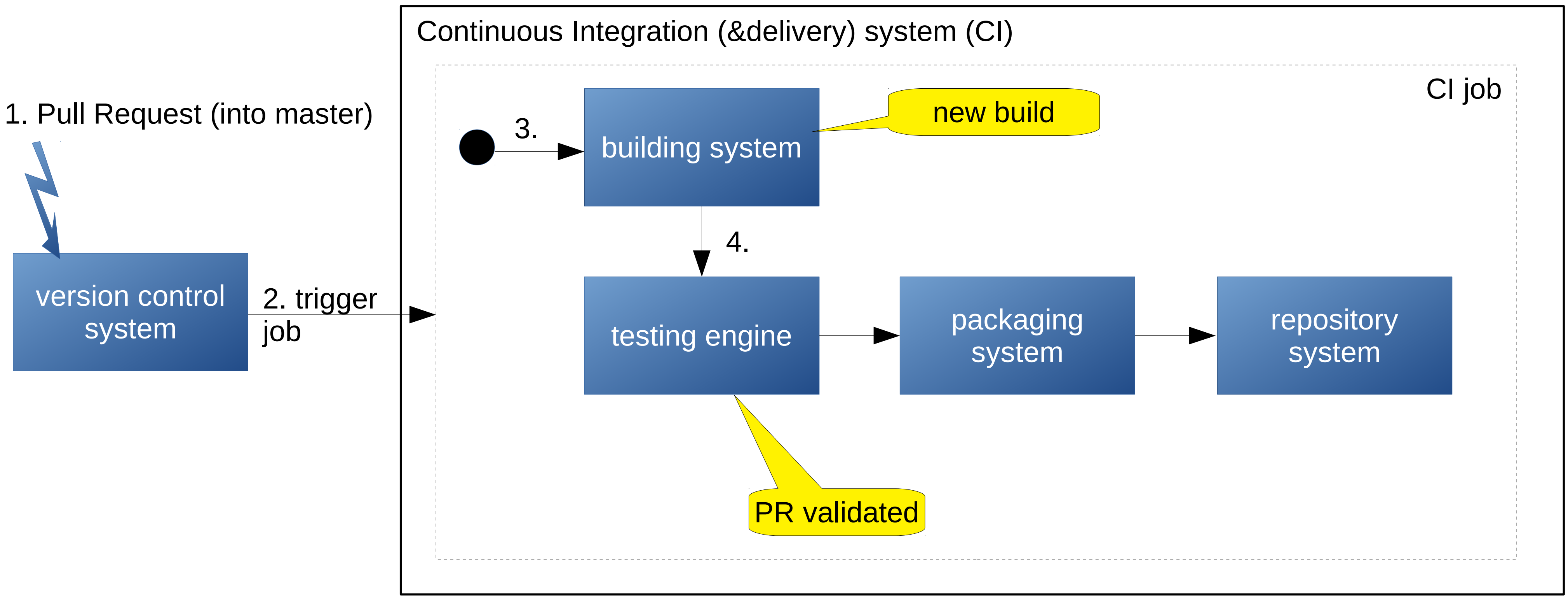
Continuous Integration (CI)

Integration tests are run...



Continuous Integration (CI)

and the PR is validated only if the tests are passed.



1. Pull Request (into master)

version control system

2. trigger job

Continuous Integration (&delivery) system (CI)

CI job

3.

building system

new build

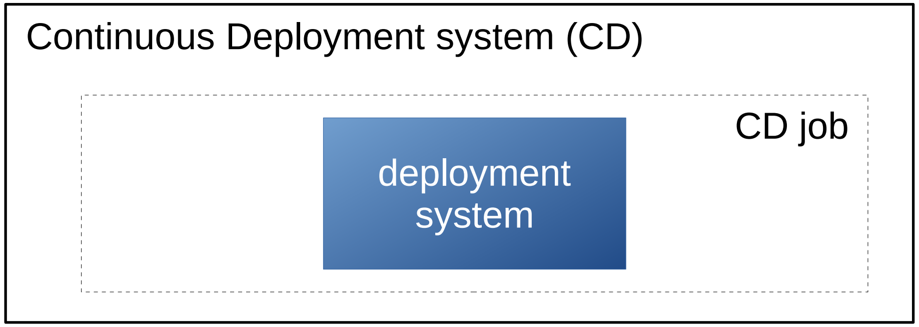
4.

testing engine

packaging system

repository system

PR validated



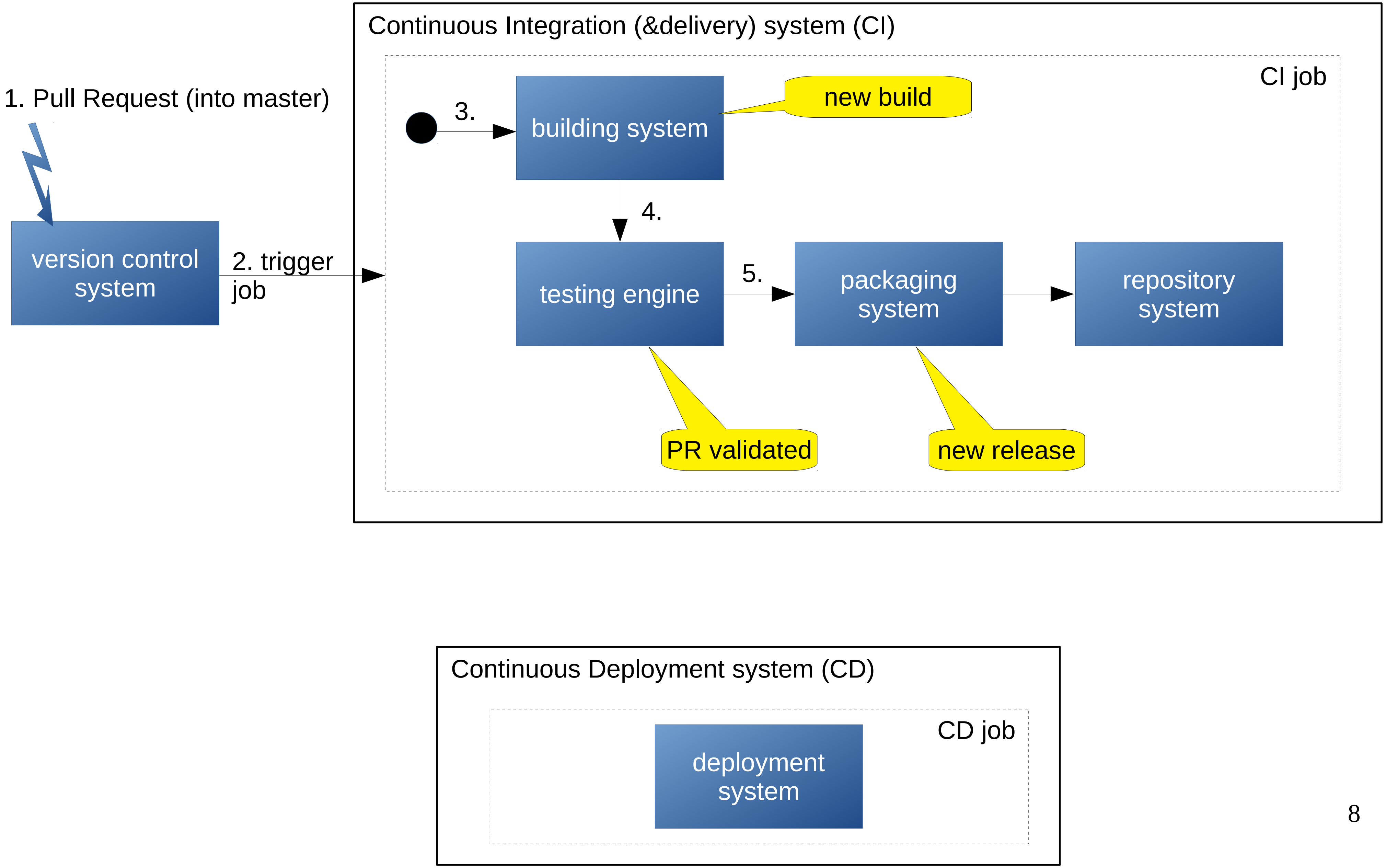
Continuous Deployment system (CD)

CD job

deployment system

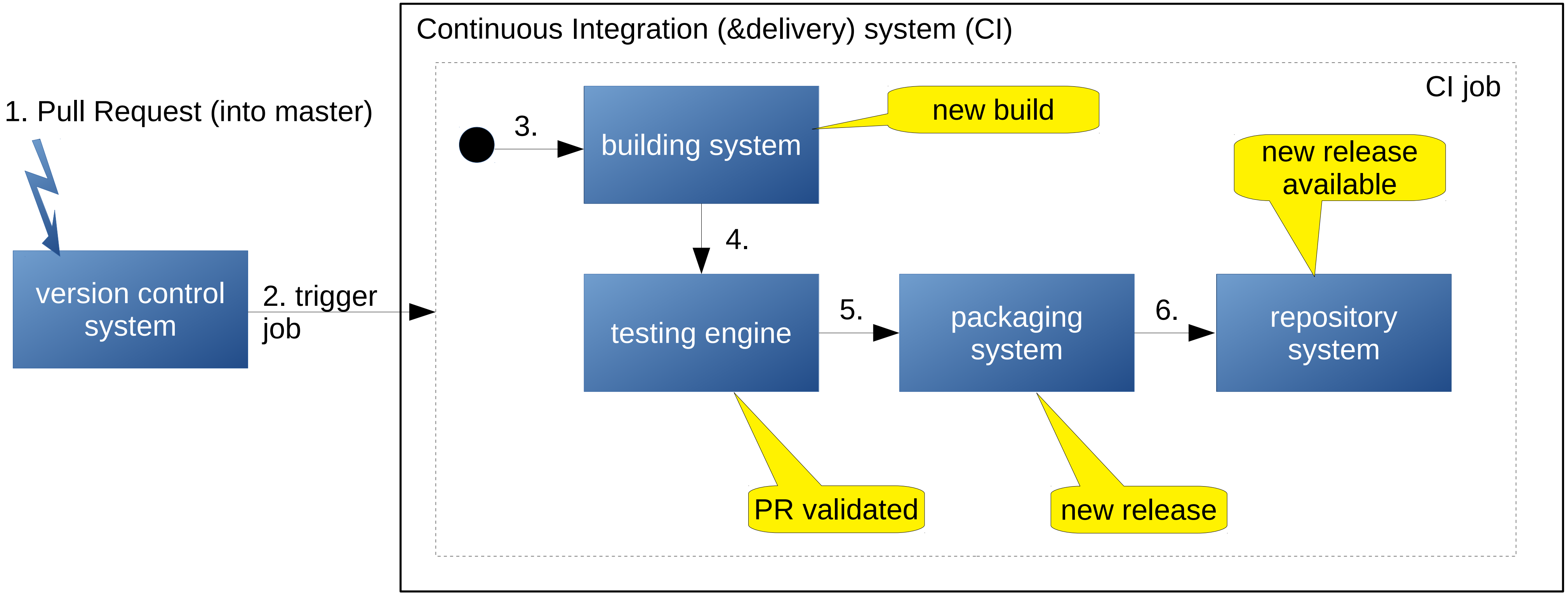
Continuous Integration (CI)

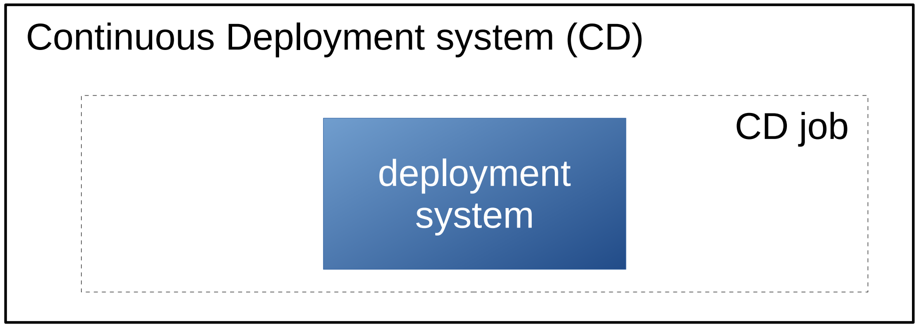
Then application is packaged...



Continuous Integration (CI)

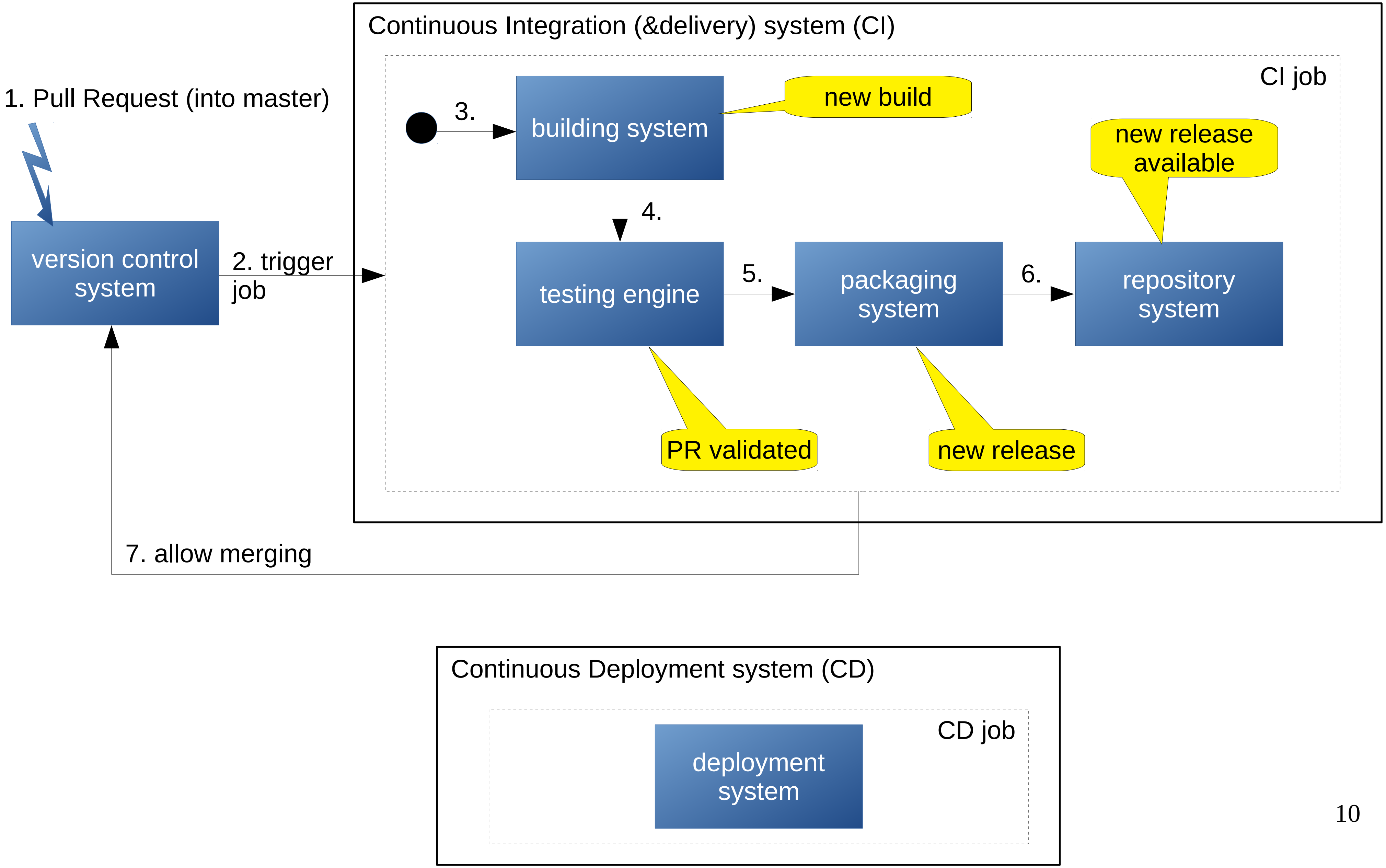
and uploaded so as to be available for every person concerned.





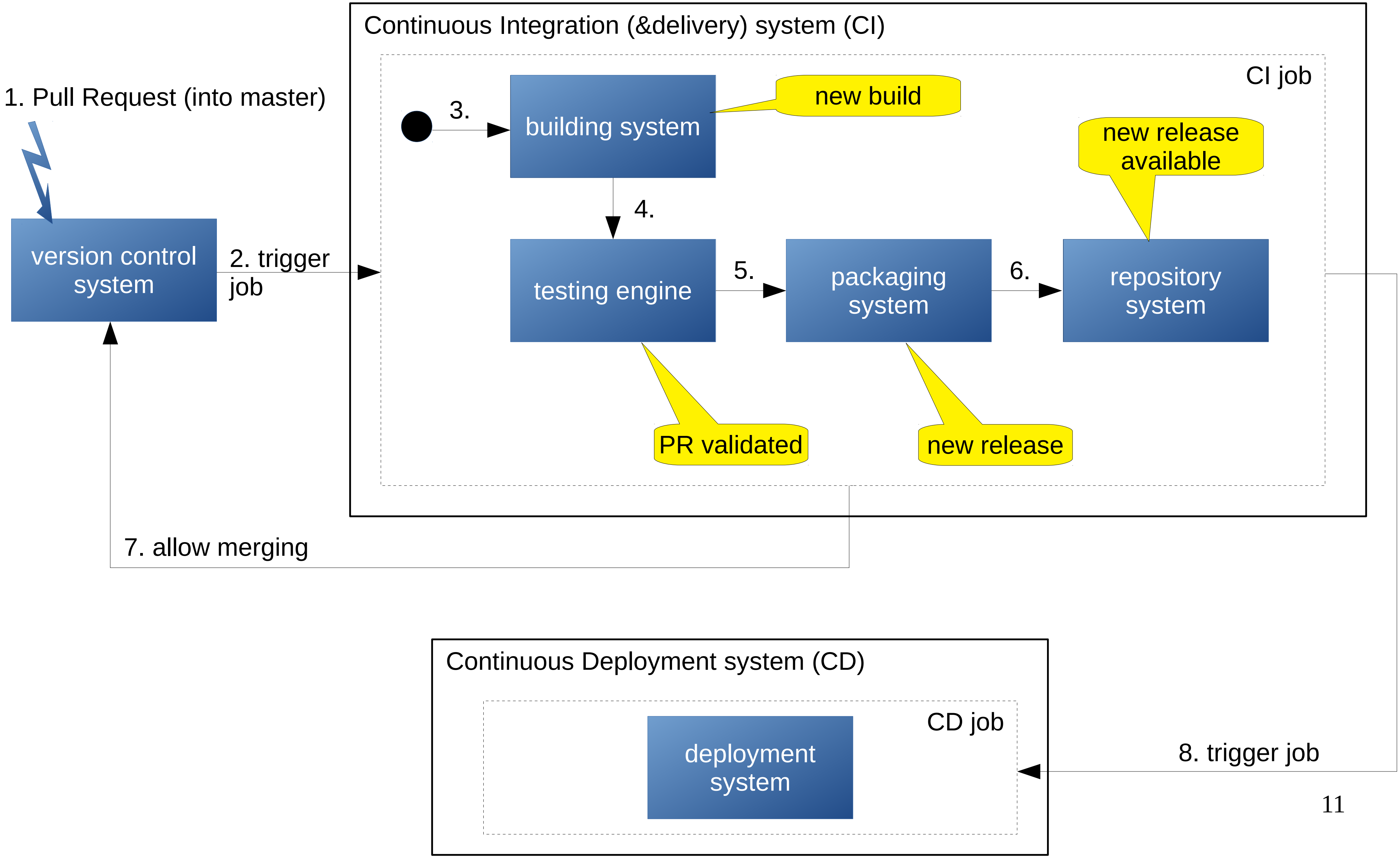
Continuous Integration (CI)

A successful CI job allows the PR to be merged into the code base.



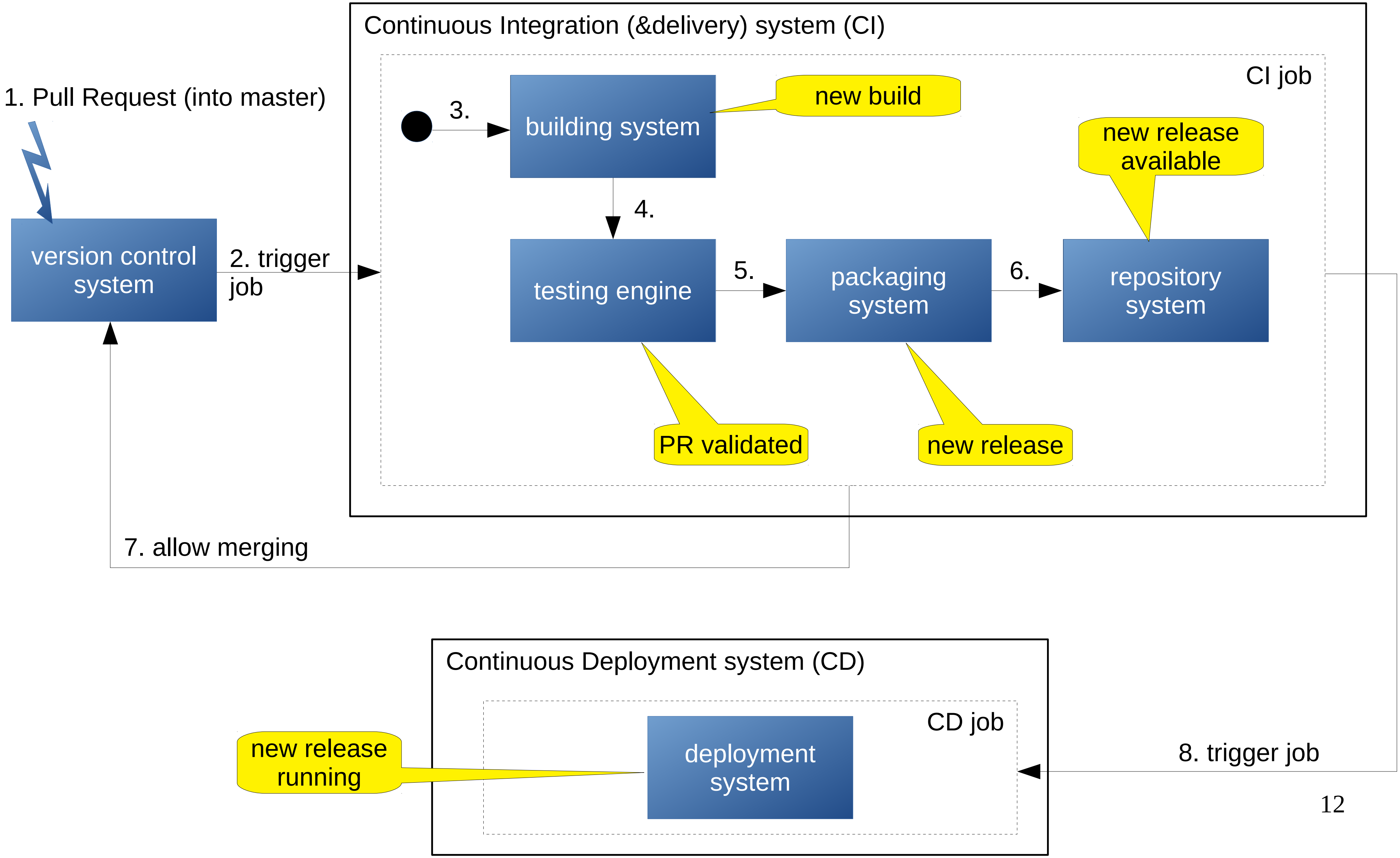
Continuous Integration (CI)

CI job can optionally trigger a Continuous Deployment job...



Continuous Integration (CI)

so as to deploy the new version of the application.



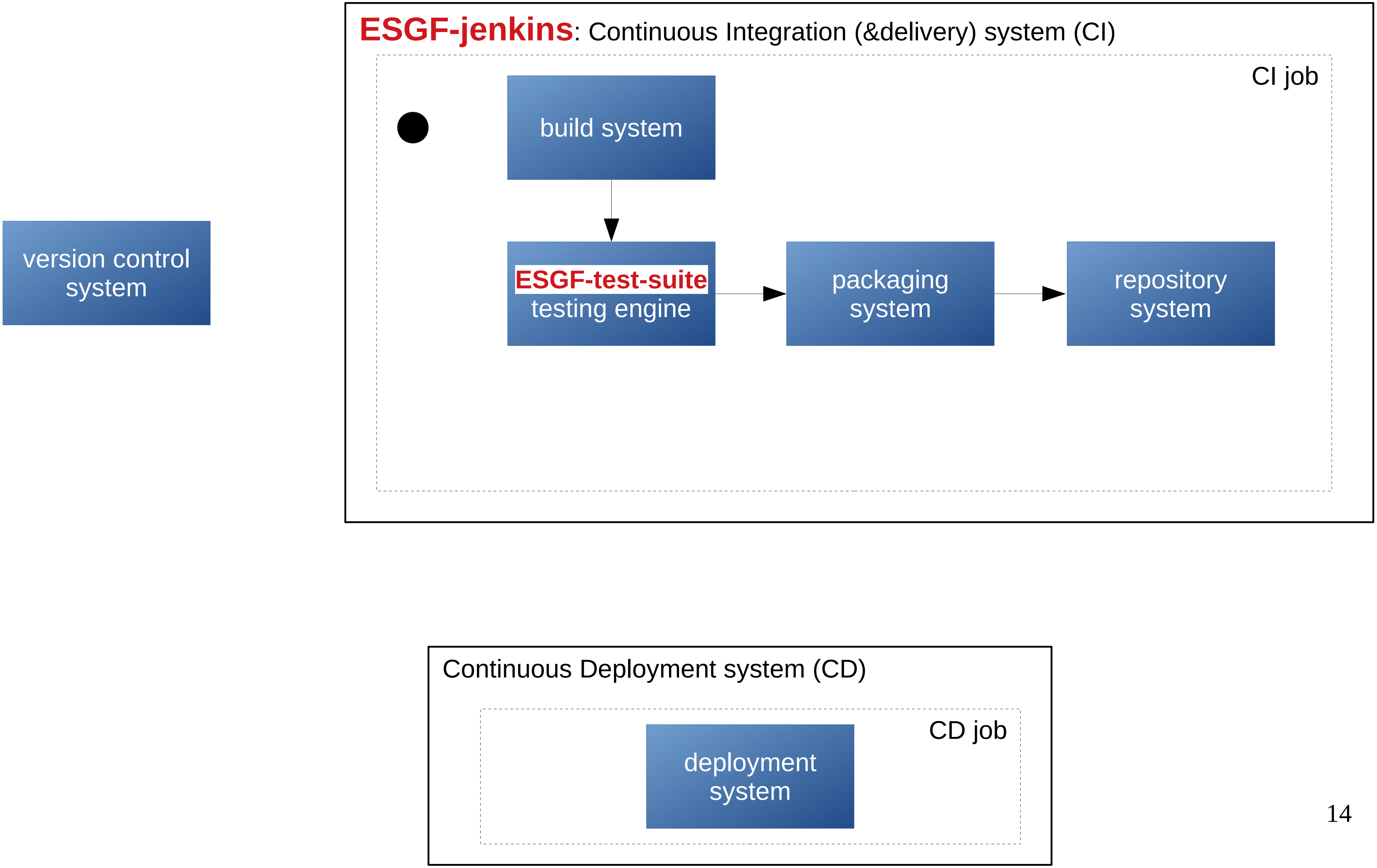
Continuous Integration (CI)

The benefits of the CI (after some reasonable efforts):

- Prevent software regression: prevent faulty code from getting merged into the main repo.
- Less error prone: human out of the loop but everything is still manually operable.
- Improve reactive security: increase the frequency of releasing, from a CVE to a new release.
- Etc.

Continuous Integration (CI)

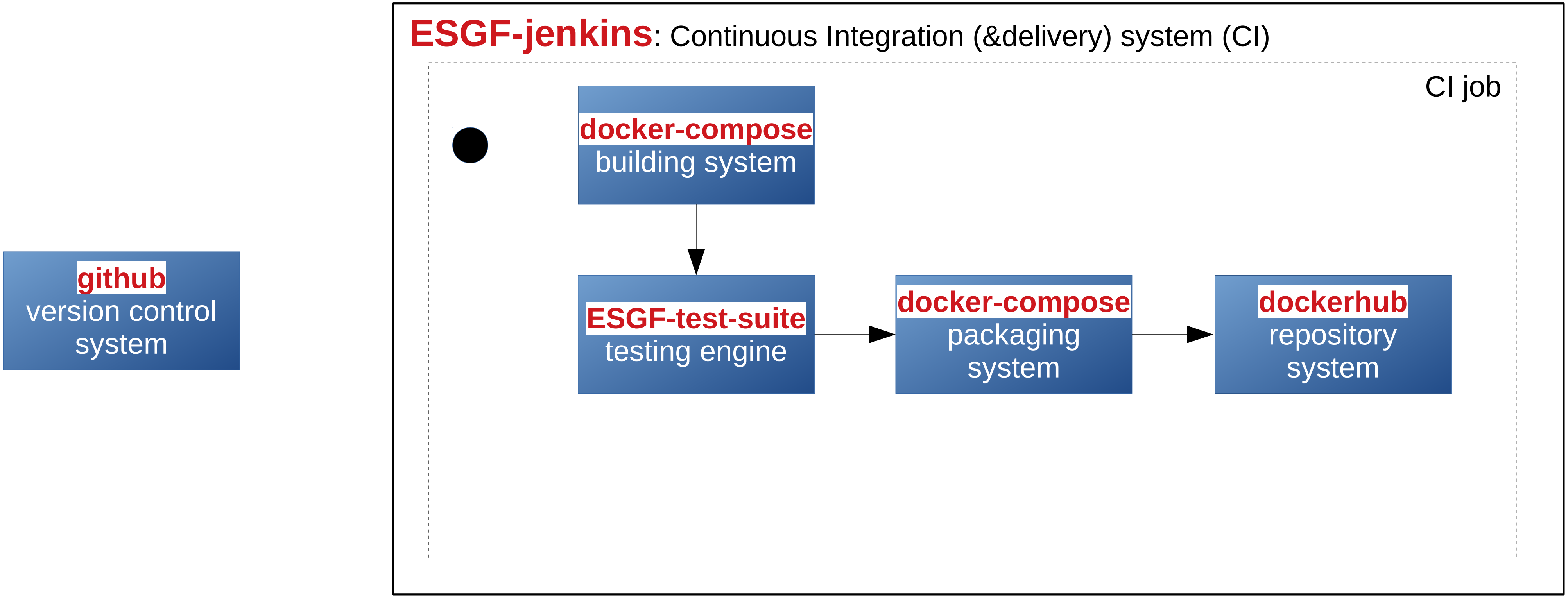
ESGF-test-suite & ESGF-jenkins role:


ESGF-jenkins: Continuous Integration (&delivery) system (CI)
CI job
build system
version control system
ESGF-test-suite testing engine
packaging system
repository system
Continuous Deployment system (CD)
CD job
deployment system

4/4

Continuous Integration (CI)

A CI example involving esgf-docker:



github version control system

ESGF-jenkins: Continuous Integration (&delivery) system (CI)

CI job

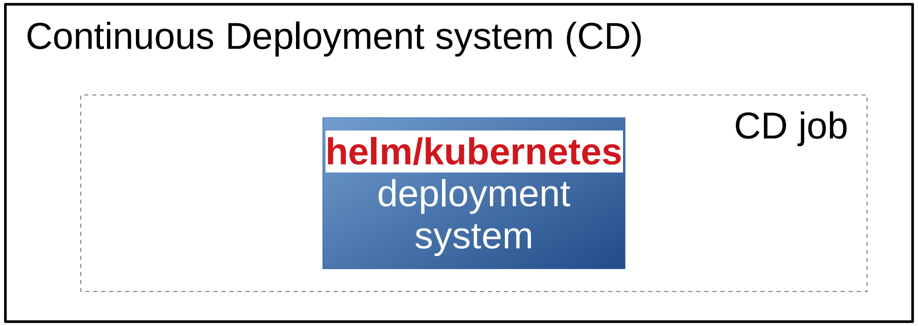
●

docker-compose building system

ESGF-test-suite testing engine

docker-compose packaging system

dockerhub repository system



Continuous Deployment system (CD)

CD job

helm/kubernetes deployment system

Notes:

- ESGF-test-suite is not specific to Continuous Integration and ESGF-docker (e.g. can check your production ESGF nodes).

- ESGF-jenkins is not specific to ESGF-docker: it welcomes any ESGF projects.

1/2

ESGF-test-suite

Goals:

- Gather the integration tests of the ESGF stack.
- Developers of the ESGF stack are invited to implement/port their integration tests into ESGF-test-suite.

Architecture: a well organized set of Python 2.7 scripts based on Nose2 (also based on Unittest). Git repository: https://github.com/ESGF/esgf-test-suite . Author: Nicolas Carenton.

For the time being, only fundamental integration tests has been implemented:

- Pinging web pages (cog, thredds, stats, search, orp, etc.).
- Cog/django login.
- Http and gridftp downloading.

Features provided by esgf-test-suite:

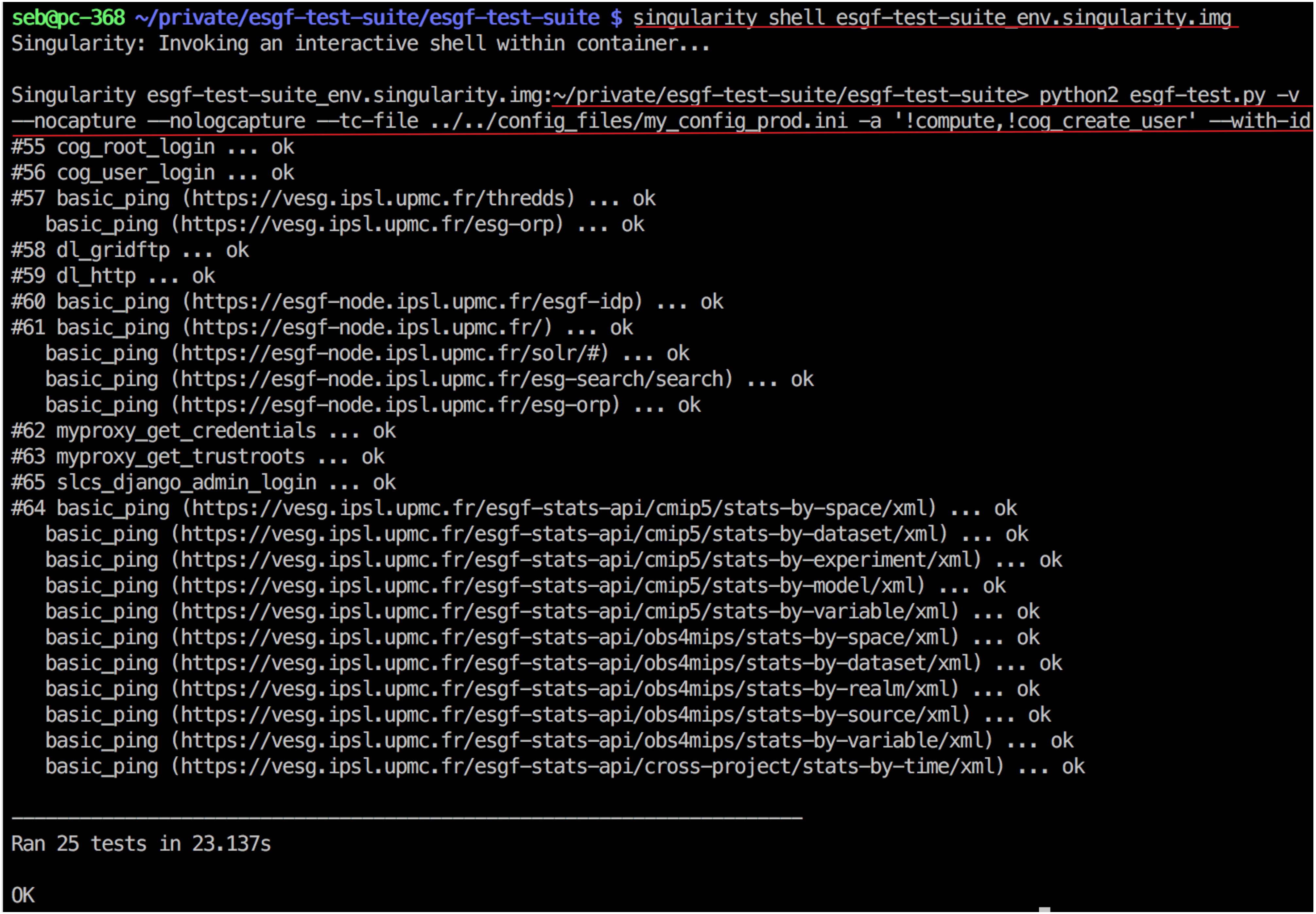
- Myproxy authentication (credential) routines.
- High levels web crawling routines based on Selenium API, using Firefox.
- High levels Thredds catalog routines.
- Nice reporting (human readable, JSON, XML, etc.).
- Selection of tests thanks to Nose.

ESGF-test-suite is standalone application with its dependencies wrapped into a singularity image (container system):

- Install singularity, once.
- Run a interactive shell within the image of the dependencies of ESGF-test-suite.
- Run ESGF-test-suite.

2/3

ESGF-test-suite

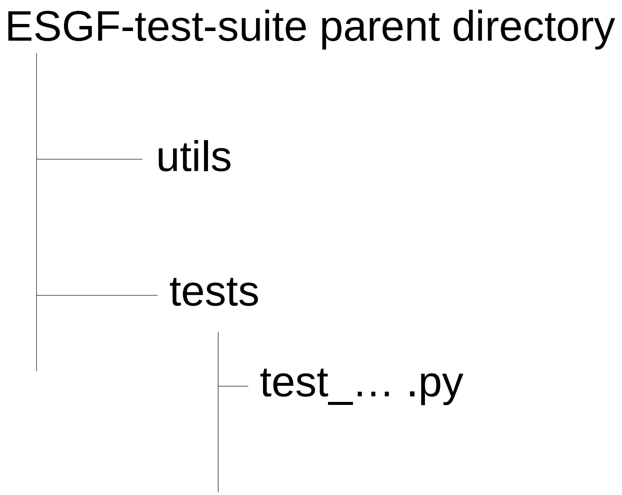


```
seb@pc-368 ~/private/esgf-test-suite/esgf-test-suite $ singularity shell esgf-test-suite_env.singularity.img
Singularity: Invoking an interactive shell within container...

Singularity esgf-test-suite_env.singularity.img:~/private/esgf-test-suite/esgf-test-suite> python2 esgf-test.py -v
--nocapture --nologcapture --tc-file ../../config_files/my_config_prod.ini -a '!compute,!cog_create_user' --with-id
#55 cog_root_login ... ok
#56 cog_user_login ... ok
#57 basic_ping (https://vesg.ipsl.upmc.fr/thredds) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esg-orp) ... ok
#58 dl_gridftp ... ok
#59 dl_http ... ok
#60 basic_ping (https://esgf-node.ipsl.upmc.fr/esgf-idp) ... ok
#61 basic_ping (https://esgf-node.ipsl.upmc.fr/) ... ok
   basic_ping (https://esgf-node.ipsl.upmc.fr/solr/#) ... ok
   basic_ping (https://esgf-node.ipsl.upmc.fr/esg-search/search) ... ok
   basic_ping (https://esgf-node.ipsl.upmc.fr/esg-orp) ... ok
#62 myproxy_get_credentials ... ok
#63 myproxy_get_trustroots ... ok
#65 slcs_django_admin_login ... ok
#64 basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/cmip5/stats-by-space/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/cmip5/stats-by-dataset/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/cmip5/stats-by-experiment/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/cmip5/stats-by-model/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/cmip5/stats-by-variable/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/obs4mips/stats-by-space/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/obs4mips/stats-by-dataset/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/obs4mips/stats-by-realm/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/obs4mips/stats-by-source/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/obs4mips/stats-by-variable/xml) ... ok
   basic_ping (https://vesg.ipsl.upmc.fr/esgf-stats-api/cross-project/stats-by-time/xml) ... ok

----------------------------------------------------------------------
Ran 25 tests in 23.137s

OK
```

ESGF-test-suite

Adding new integration tests:

1. Create a new file prefixed “test_” in the tests directory

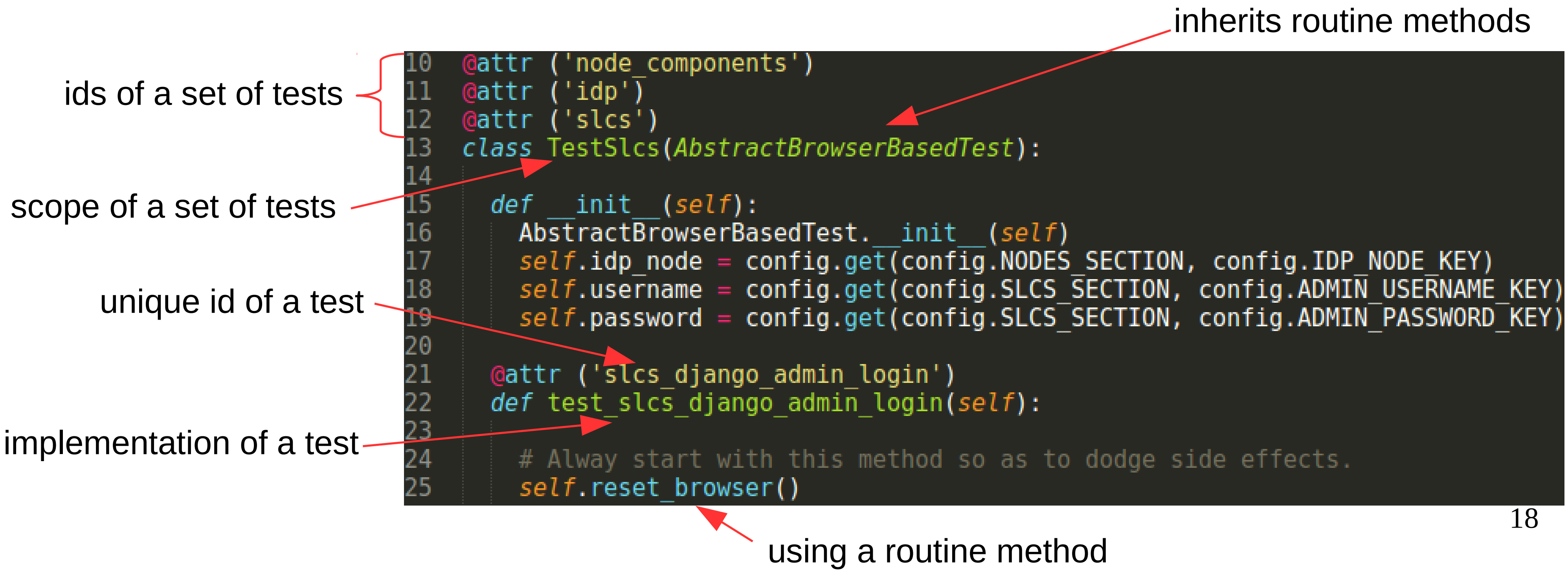

ESGF-test-suite parent directory

utils

tests

test_… .py

2. Implement a Python class (set of tests) and some methods (tests) with annotations (ids):



```
@attr ('node_components')
@attr ('idp')
@attr ('slcs')
class TestSlcs(AbstractBrowserBasedTest):

    def __init__(self):
        AbstractBrowserBasedTest.__init__(self)
        self.idp_node = config.get(config.NODES_SECTION, config.IDP_NODE_KEY)
        self.username = config.get(config.SLCS_SECTION, config.ADMIN_USERNAME_KEY)
        self.password = config.get(config.SLCS_SECTION, config.ADMIN_PASSWORD_KEY)

    @attr ('slcs_django_admin_login')
    def test_slcs_django_admin_login(self):

        # Alway start with this method so as to dodge side effects.
        self.reset_browser()
```

inherits routine methods

ids of a set of tests

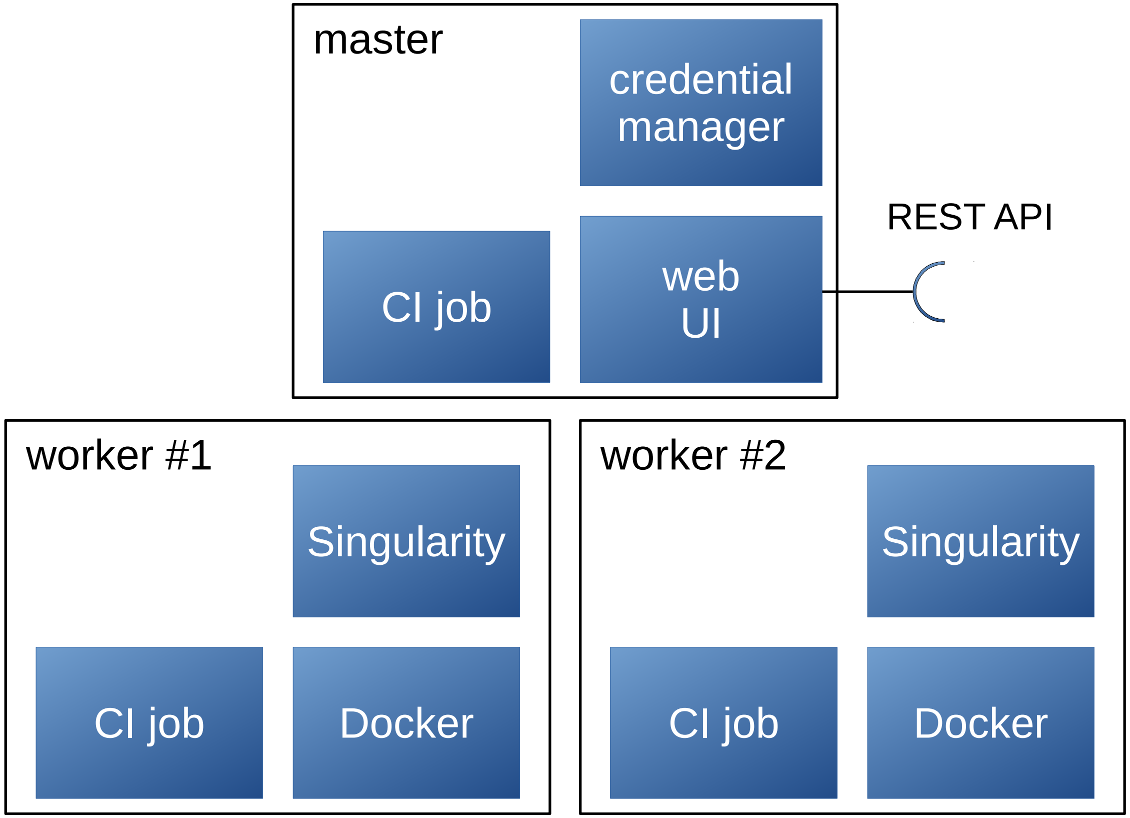
scope of a set of tests

unique id of a test

implementation of a test

using a routine method

ESGF-jenkins



ESGF-jenkins, a cluster of 3 Jenkins nodes:

- Scalable (by adding new worker nodes).
- Job failover (across the nodes of a cluster).
- Extensible (plugin system).
- Nice web interface (https://esgf-build.ipsl.upmc.fr/jenkins/).
- Point&Click configuration of Job.
- CI Job description language (Jenkins pipeline script named Jenkinsfile).
- GUI Job pipeline builder.
- Secure credential manager.
- Provides REST API.

Goals:

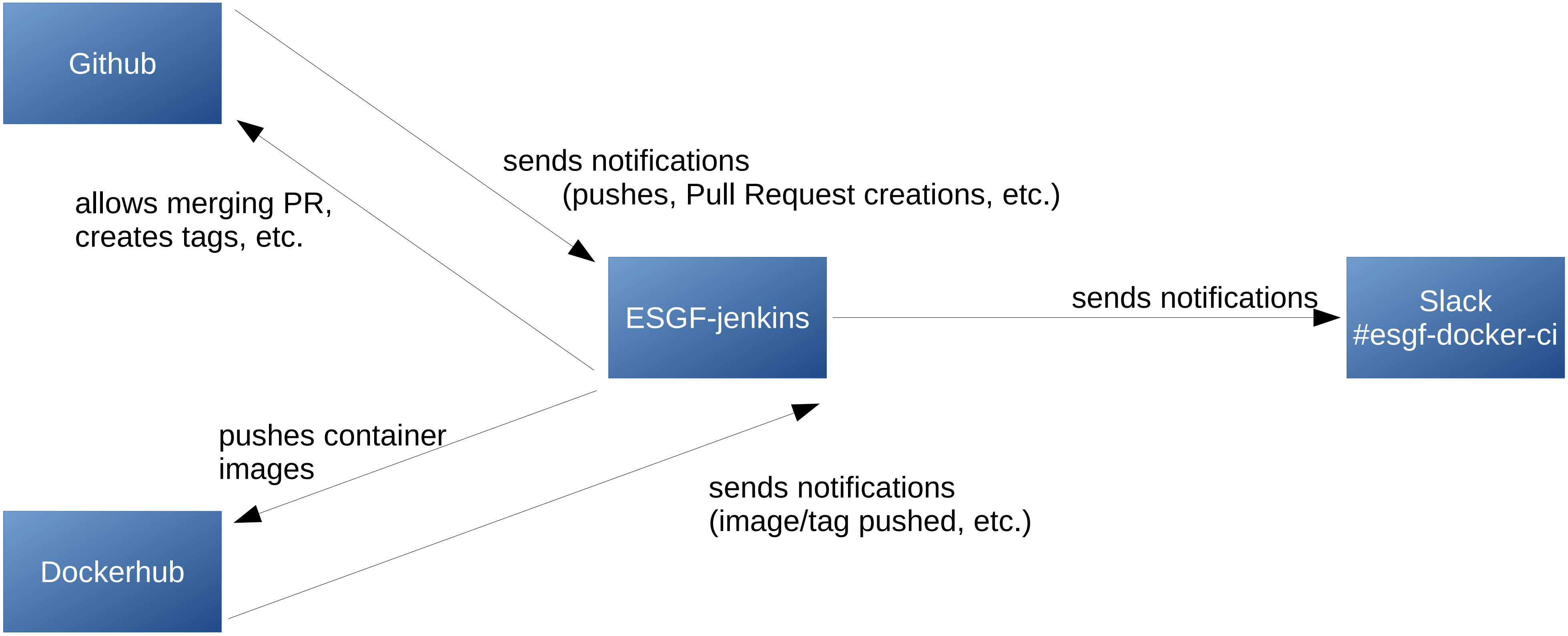
- Run CI jobs (provide computation resources).
- Ease the interactions between CI jobs and other applications (Version Control System, Continuous Deployment systems, etc.).

Working capacities:

- Every node can run 2 jobs at the same time.
- Worker nodes can run 1 job using Docker and Singularity at the same time.
- Provided job subsystems (Docker, Singularity, Git, Maven, Ant, Gradle, Bash, etc.).

ESGF-jenkins

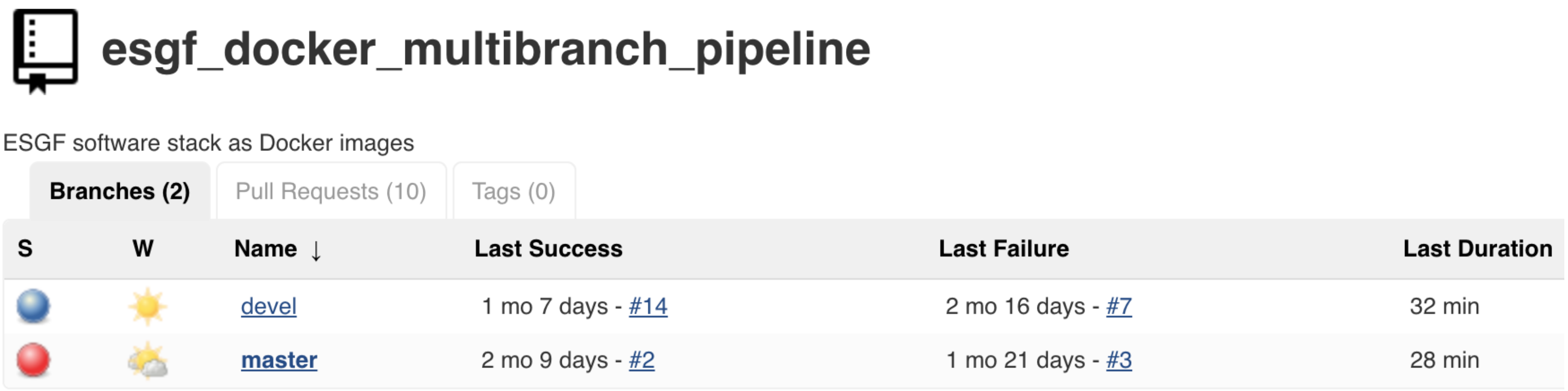
Examples of interactions



A complete and commented job script is available at:
https://github.com/ESGF/esgf-docker/blob/master/Jenkinsfile

ESGF-jenkins

Job status (failed, successful, presented like a weather forecast):

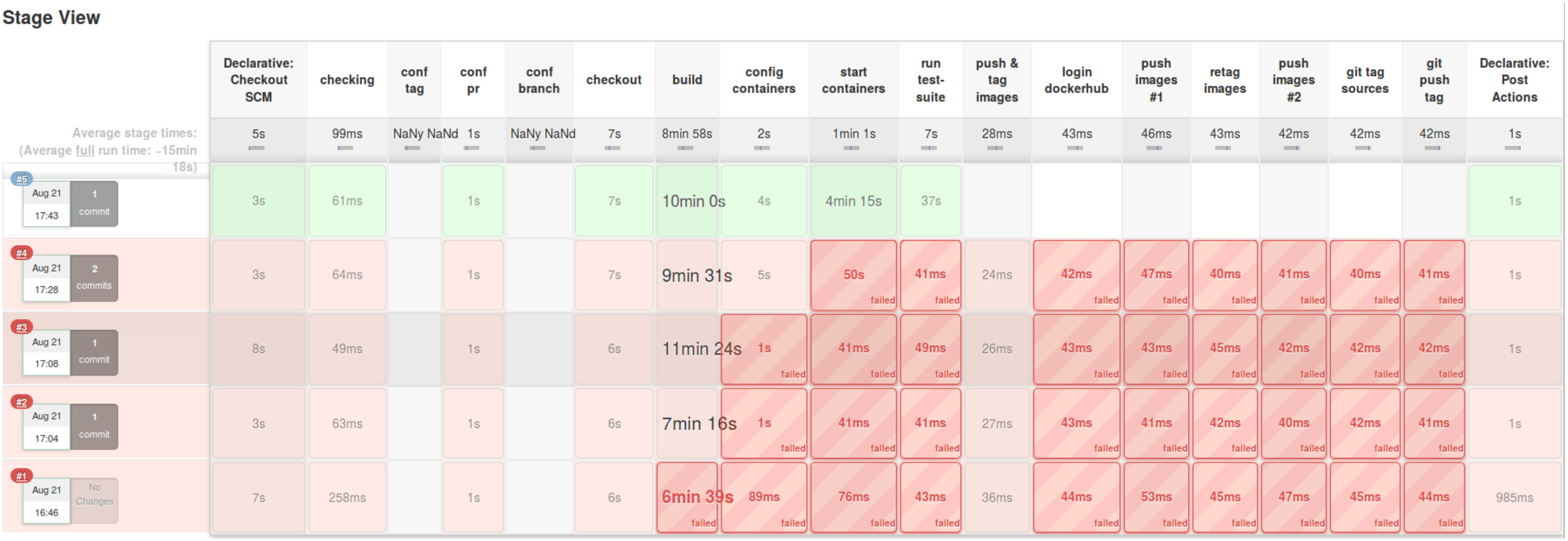


esgf_docker_multibranch_pipeline

ESGF software stack as Docker images

S	W	Name ↓	Last Success	Last Failure	Last Duration
🔵	☀	devel	1 mo 7 days - #14	2 mo 16 days - #7	32 min
🔴	🌤	master	2 mo 9 days - #2	1 mo 21 days - #3	28 min

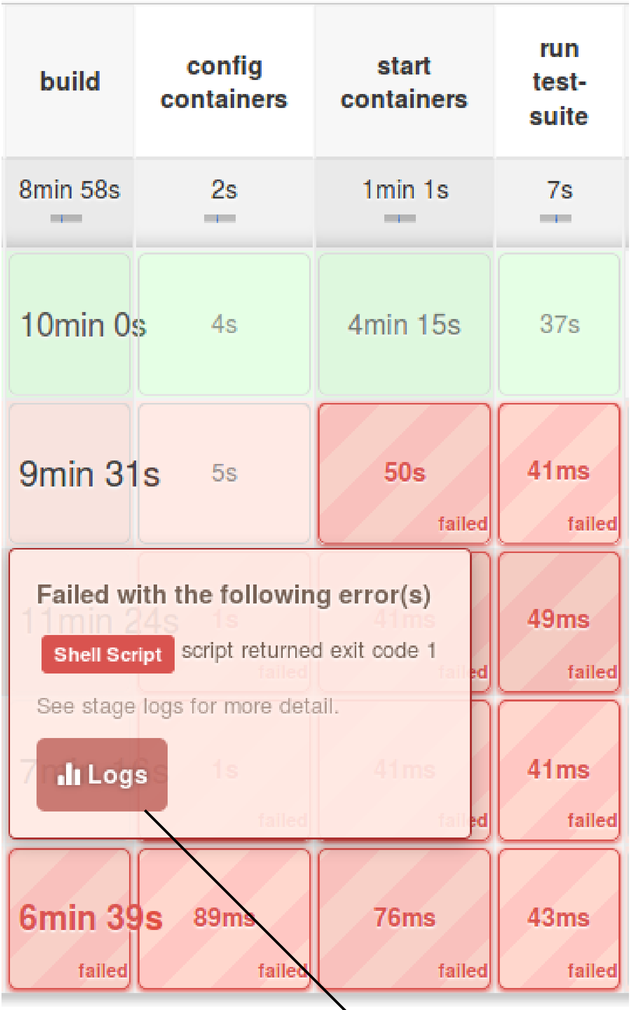
Job stage view:



Stage View

	Declarative: Checkout SCM	checking	conf tag	conf pr	conf branch	checkout	build	config containers	start containers	run test-suite	push & tag images	login dockerhub	push images #1	retag images	push images #2	git tag sources	git push tag	Declarative: Post Actions
Average stage times: (Average full run time: ~15min 18s)	5s	99ms	NaNy NaNd	1s	NaNy NaNd	7s	8min 58s	2s	1min 1s	7s	28ms	43ms	46ms	43ms	42ms	42ms	42ms	1s
#5 Aug 21 17:43 1 commit	3s	61ms		1s		7s	10min 0s	4s	4min 15s	37s								1s
#4 Aug 21 17:28 2 commits	3s	64ms		1s		7s	9min 31s	5s	50s failed	41ms failed	24ms	42ms failed	47ms failed	40ms failed	41ms failed	40ms failed	41ms failed	1s
#3 Aug 21 17:08 1 commit	8s	49ms		1s		6s	11min 24s	1s failed	41ms failed	49ms failed	26ms	43ms failed	43ms failed	45ms failed	42ms failed	42ms failed	42ms failed	1s
#2 Aug 21 17:04 1 commit	3s	63ms		1s		6s	7min 16s	1s failed	41ms failed	41ms failed	27ms	43ms failed	41ms failed	42ms failed	40ms failed	42ms failed	41ms failed	1s
#1 Aug 21 16:46 No Changes	7s	258ms		1s		6s	6min 39s failed	89ms failed	76ms failed	43ms failed	36ms	44ms failed	53ms failed	45ms failed	47ms failed	45ms failed	44ms failed	985ms

Exploring the log of a CI job:



build	config containers	start containers	run test-suite
8min 58s	2s	1min 1s	7s
10min 0s	4s	4min 15s	37s
9min 31s	5s	50s failed	41ms failed
11min 24s	1s failed	41ms failed	49ms failed
7min 16s	1s failed	41ms failed	41ms failed
6min 39s failed	89ms failed	76ms failed	43ms failed

Failed with the following error(s)

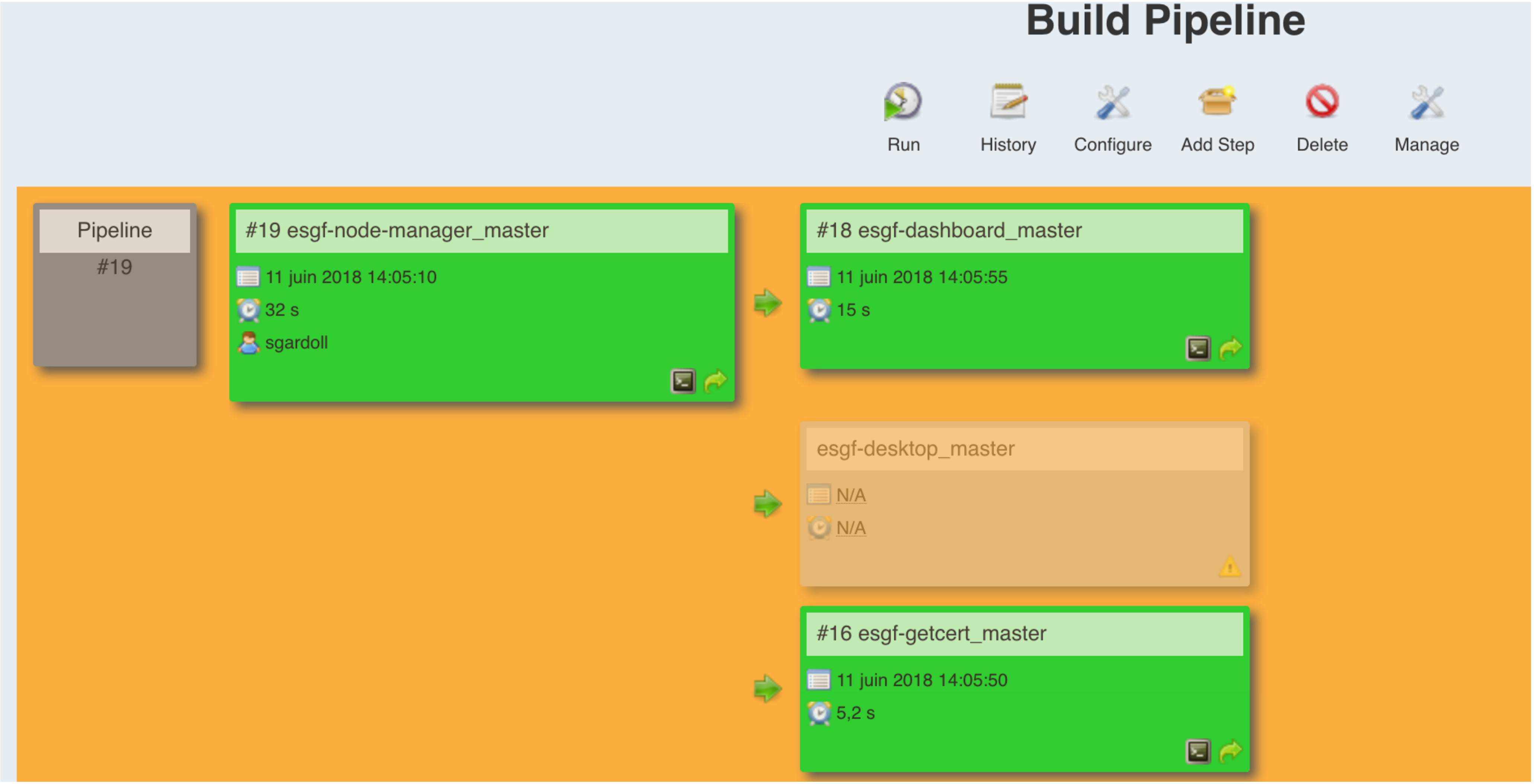
Shell Script script returned exit code 1

See stage logs for more detail.

```
15:55:22 warning: no previously-included files matching '*.py[co]' found anywhere in distribution
15:55:22 no previously-included directories found matching 'docs/_build'
15:55:22   File "build/bdist.linux-x86_64/egg/jinja2/asyncfilters.py", line 7
15:55:22     async def auto_to_seq(value):
15:55:22             ^
15:55:22 SyntaxError: invalid syntax
```

ESGF-jenkins

Builder of pipelines of jobs:



Build Pipeline

Pipeline
#19

#19 esgf-node-manager_master
11 juin 2018 14:05:10
32 s
sgardoll

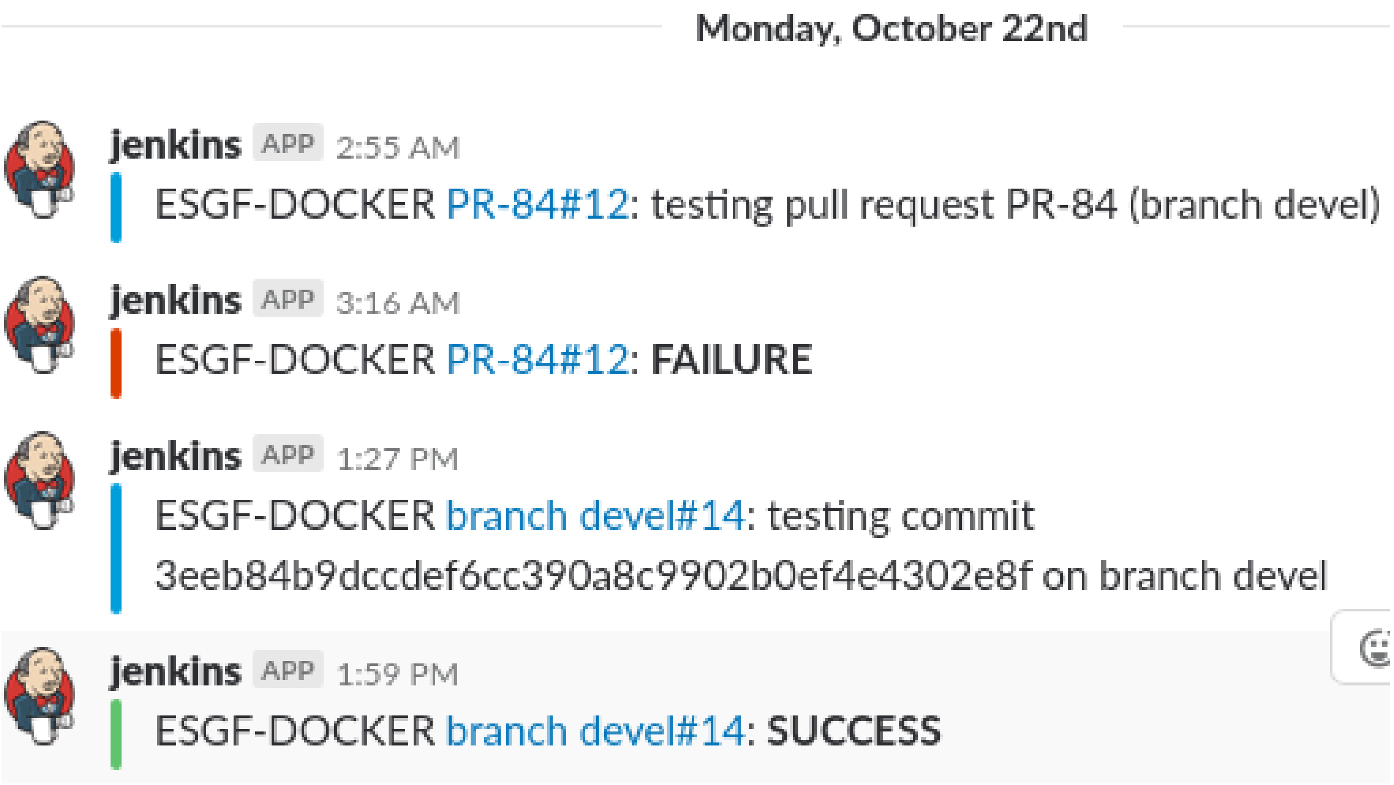
#18 esgf-dashboard_master
11 juin 2018 14:05:55
15 s

esgf-desktop_master
N/A
N/A

#16 esgf-getcert_master
11 juin 2018 14:05:50
5,2 s

ESGF-jenkins

Slack notifications (a PR successively failed then successfully tested):



Monday, October 22nd

jenkins APP 2:55 AM
ESGF-DOCKER PR-84#12: testing pull request PR-84 (branch devel)

jenkins APP 3:16 AM
ESGF-DOCKER PR-84#12: **FAILURE**

jenkins APP 1:27 PM
ESGF-DOCKER branch devel#14: testing commit 3eeb84b9dccdef6cc390a8c9902b0ef4e4302e8f on branch devel

jenkins APP 1:59 PM
ESGF-DOCKER branch devel#14: **SUCCESS**

Words in blue are hyperlinks that point to the jobs.

Want an ESGF-jenkins account ?

Need help to implement integration tests for ESGF-test-suite ?

send me an email: sgardoll@ipsl.fr

Thank you for your attention