A gateway between GitLab CI and LHCbDIRAC

C.Burr, B.Couturier on behalf of the LHCb Collaboration

24th International Conference on Computing in High Energy and Nuclear Physics
4-8 November 2019, Adelaide, Australia

c.b@cern.ch ben.couturier@cern.ch

The problem

Why implement a gateway between GitLab CI and LHCbDIRAC ?
The LHCb experiment uses GitLab[1] to manage its physics software lifecycle. While standard GitLab CI (Continuous Integration) runners are appropriate to run unit tests or small test jobs, data analysis production jobs validation is CPU intensive and exceeds the capacities of standard runners with test runtimes varying from a few minutes to a tens of hours.
⇒ **Hence the idea to run GitLab CI jobs on the WLCG[2] (using LHCb's grid middleware, LHCbDIRAC).**

LHCbDIRAC is developed in Python and based on the DIRAC[3] Interware. Interfacing them meant:

- **Using the GitLab REST[4] API to interact with the GitLab server** i.e.
  - Get the next CI job to run for a given project, including all parameters known to GitLab
  - Report on the status of the job (must run frequently else GitLab may consider the job as stuck)
  - Terminate the job and push artefacts back to the server
- **Managing the CI jobs in a reliable and scalable manner**
  - A **production grade task queue system** is appropriate way to solve this issue

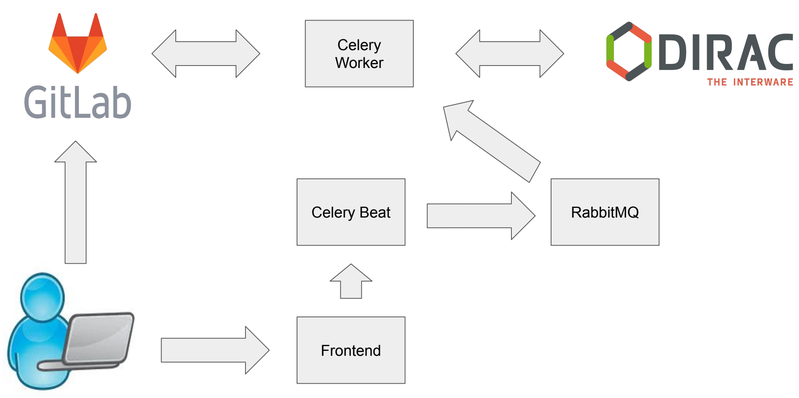
The LHCb-ci-runner system

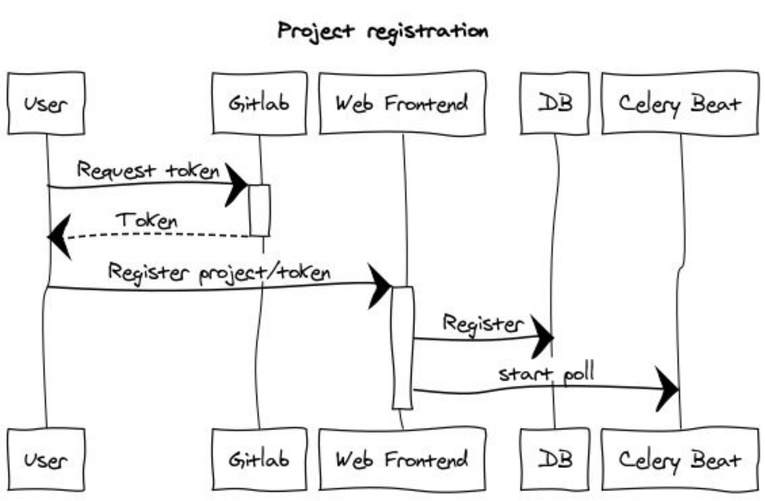
System design

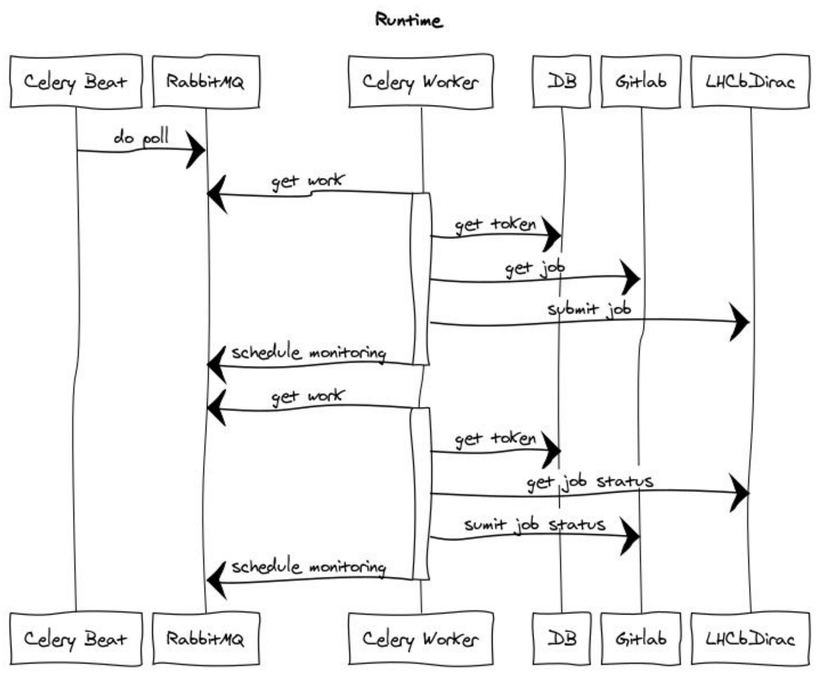
- The **lhcb-ci-runner** standalone package was developed to interface with the GitLab REST API to ease those operations
- Use of the **Celery[5] distributed task queue** system.
  Reliable and scalable way to start and manage jobs.
- Celery **internally** uses the **RabbitMQ[6] messaging system**.
- A **Web Frontend** using **Flask[7]** allows users to manage the projects and tokens.
- Runner code must be implemented as a **python package deployed on PyPI**
- **Plugins** can be added dynamically using Python entry points

Deployment

- Deployed on CERN's instance of the Redhat **OpenShift[8]** Container platform to ensure the scalability of the system.
- To avoid single points of failure, a **RabbitMQ** cluster could be created for higher reliability.
- **Flask Web Frontend** as a container in the same openshift project.
- Data kept on a **volume** in Openshift or a CERN DBonDemand **MySQL instance**.







https://www.websequencediagrams.com/

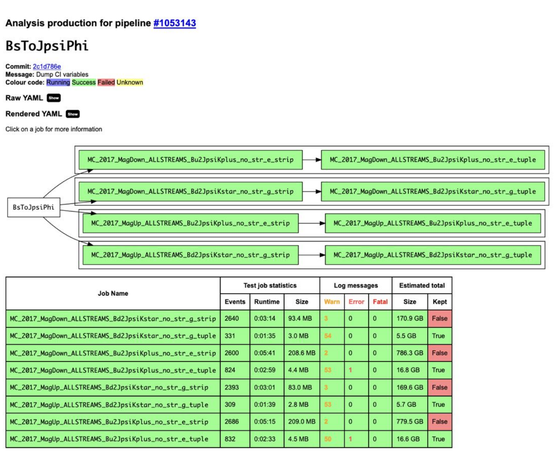
Security considerations

- The GitLab project manager needs to request a token from GitLab to pass it to the system using the web frontend
- GitLab is trusted by the gateway to provide the appropriate credentials when connecting to LHCbDIRAC (trust between the GitLab instance and LHCbDIRAC is therefore essential)
- Authenticated CERN GitLab users are mapped to Grid credentials using the VOMS API

Current prototype and applications

Analysis productions

- The current prototype allows testing running test productions on the grid using LHCbDIRAC
- Allows a dynamic number of jobs to be launched, one per dataset processed (not possible with standard GitLab runners)
- Status summary reported to GitLab CI, additional logs and output for each production can be found in a dedicated web app



Analysis production for pipeline #1053143

BsToJpsiPhi

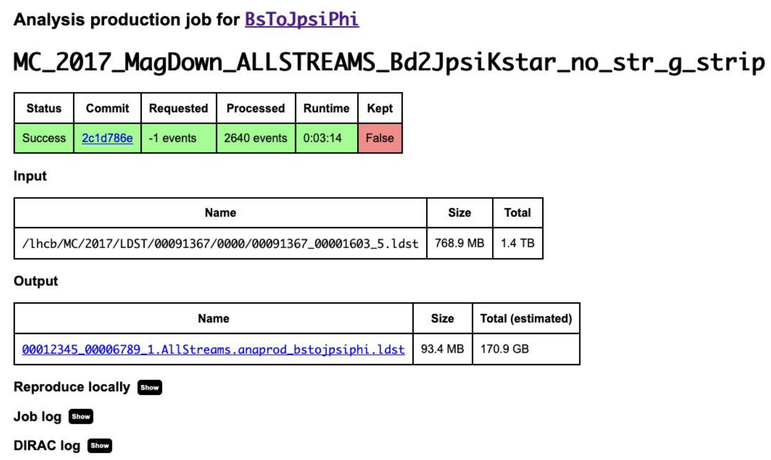
Commit: 2c1d786e
Message: Dump CI variables
Colour code: Running Success Failed Unknown

Raw YAML Show

Rendered YAML Show

Click on a job for more information

Job Name	Test job statistics			Log messages			Estimated total	
	Events	Runtime	Size	Warn	Error	Fatal	Size	Kept
MC_2017_MagDown_ALLSTREAMS_Bd2JpsiKstar_no_str_g_strip	2640	0:03:14	93.4 MB	5	0	0	170.9 GB	False
MC_2017_MagDown_ALLSTREAMS_Bd2JpsiKstar_no_str_g_tuple	331	0:01:35	3.0 MB	54	0	0	5.5 GB	True
MC_2017_MagDown_ALLSTREAMS_Bu2JpsiKplus_no_str_e_strip	2600	0:05:41	208.6 MB	1	0	0	786.3 GB	False
MC_2017_MagDown_ALLSTREAMS_Bu2JpsiKplus_no_str_e_tuple	824	0:02:59	4.4 MB	53	1	0	16.8 GB	True
MC_2017_MagUp_ALLSTREAMS_Bd2JpsiKstar_no_str_g_strip	2393	0:03:01	83.0 MB	5	0	0	169.6 GB	False
MC_2017_MagUp_ALLSTREAMS_Bd2JpsiKstar_no_str_g_tuple	309	0:01:39	2.8 MB	53	0	0	5.7 GB	True
MC_2017_MagUp_ALLSTREAMS_Bu2JpsiKplus_no_str_e_strip	2686	0:05:15	209.0 MB	1	0	0	779.5 GB	False
MC_2017_MagUp_ALLSTREAMS_Bu2JpsiKplus_no_str_e_tuple	832	0:02:33	4.5 MB	53	1	0	16.6 GB	True



Analysis production job for BsToJpsiPhi

MC_2017_MagDown_ALLSTREAMS_Bd2JpsiKstar_no_str_g_strip

Status	Commit	Requested	Processed	Runtime	Kept
Success	2c1d786e	-1 events	2640 events	0:03:14	False

Input

Name	Size	Total
/lhcb/MC/2017/LDST/00091367/0000/00091367_00001603_5.ldst	768.9 MB	1.4 TB

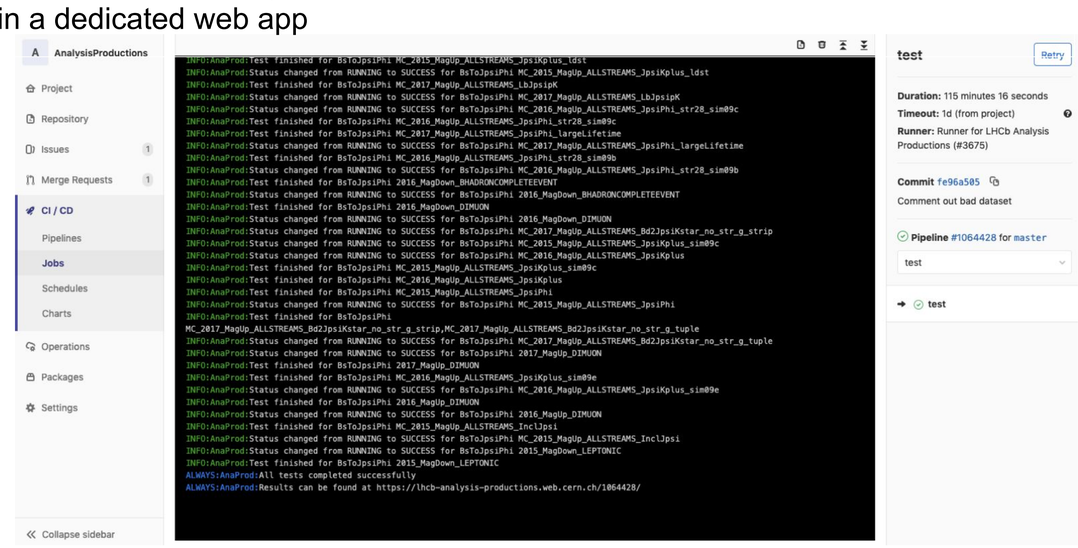
Output

Name	Size	Total (estimated)
00012345_00006789_1.AllStreams.anaprod_bstojpsiphi.ldst	93.4 MB	170.9 GB

Reproduce locally Show

Job log Show

DIRAC log Show



PyPI deployment

- Many PyPI packages created for LHCb core software
- Provides centralised storage of PyPI credentials
- Removes overhead to allow for faster deployment

Documentation deployment

- The documentation can be prepared using Sphinx in the GitLab CI job
- Deployed to S3 using the appropriate credentials

CVMFS deployment

- Packages to deploy can be prepared in the CI job
- A specific PyPI package will be used by the runner to send a message to the installation queue

Current prototype and applications

A prototype has been developed that shows that it is possible to develop a reliable gateway between GitLab CI and LHCbDIRAC. This allows for a more scalable and extensible allocation of resources for the CI jobs. The components developed are however generic and not restricted to running jobs on LHCbDIRAC: they can be used to run CI jobs on any resource available, independently of the LHCb software stack.

1. GitLab https://about.gitlab.com/
2. WLCG http://wlcg.web.cern.ch/
3. DIRAC http://diracgrid.org/
4. Fielding, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.
6. Celery Distributed task queue http://www.celeryproject.org
7. RabbitMQ messaging broker https://www.rabbitmq.com
8. Flask micro web framework https://palletsprojects.com/p/flask/
9. Redhat Openshift https://www.openshift.com