

2014-11-14 Qserv Integration testbed @ CC-IN2P3

Date

14 Nov 2014

Attendees

- [Fabio Hernandez](#) (CC-IN2P3, chair and secretary)
- [Fabrice Jammes](#) (Qserv team)
- Yvan Calas (CC-IN2P3)

Context

CC-IN2P3 is preparing the deployment of an integration testbed for LSST's Qserv. During the first phase, the testbed will be composed of 50 interconnected homogeneous storage nodes adequately configured and devoted to this service.

Since the hardware is not yet delivered, CC-IN2P3 has provided to the Qserv development team a set of 3 nodes for the purpose of building the software and its dependencies in an environment as similar as possible as the one of the machines in the future integration testbed. These 3 machines are reachable from outside the local network, are homogeneous in terms of hardware and software and all have access to a shared file system.

The Qserv team has already been using these machines for building a release and the dependencies are identified and documented.

Goal

The goal of this meeting was to explore possible ways for CC-IN2P3 to provide to the Qserv team a working environment allowing for fast iterations of building, deployment and testing the software in the integration testbed, while exploiting the tools CC-IN2P3 currently uses for automated software deployment so to keep a consistent and controlled testing environment. Fast test cycles at scale are considered very important because of the intrinsic difficulties of automatically testing distributed systems.

Discussion

The following points were discussed and agreed:

- CC-IN2P3 team does not want to rely on the presence of a shared file system on the machines composing the integration testbed for the purposes of software deployment. Qserv is a shared-nothing database that does not require such presence at execution time either.
- CC-IN2P3 team intends to use a mechanism that allows for automatic deployment of the software (Qserv and its dependencies) to the integration testbed both on a regular-basis and on demand. That mechanism must allow CC-IN2P3 team to guarantee that a given version of the software is installed on all or a selected group of machines in the testbed in a controlled and automated way.
RPM packaging is the preferred way for reaching the goals of automated deployment and dependency management and it integrates well to the Puppet-based deployment tools already in use at CC-IN2P3.
- Qserv is currently packaged using EUPS, a source-oriented version manager system. This implies that Qserv and all its dependencies need to be built from sources. No ready-for-deployment binary package is currently produced with this method by the Qserv team. This is a constraint since it seems neither practical nor necessary to build the Qserv software and its dependencies on every single machine in the testbed.
- Yvan Calas volunteered to explore how to package a binary distribution of Qserv and its dependencies, which would allow for automatic and fast deployment of the software. The initial goal is to produce RPMs that could be easily integrated to the Puppet system for fast deployment.
If binary packaging the code using RPM proves to be impractical or impossible at this stage, alternative mechanisms will be explored, such as versioned archives of binary files or replication of the binaries generated by EUPS at build time to the machines in the integration testbed. EUPS provides ways to select the desired version of a particular package (of Qserv itself or of its dependencies) so having all the set of installed packages on the execution nodes is necessary for keeping that flexibility.
The goal of setting up a controlled, automatic mechanism for deployment on the testbed must be satisfied by the selected mechanism.
- Since Qserv is in active development and will require frequent modifications, it was agreed to separate the packages of Qserv itself from the packages of its dependencies. This fine granularity should allow for fast deployment of modified versions of Qserv only or of selected dependencies, while keeping the same installed versions of anything else.
- A convenient versioning scheme will be implemented to the binary packaging of Qserv so that we can track which version is installed on which machine and automatically upgrade or downgrade all or part of the testbed.
- Currently, the 3 machines in the build cluster do not have direct network access to the Internet, in particular, to the official LSST software repository [<https://sw.lsstcorp.org/eupspkg/>]. This is inconvenient and CC-IN2P3 will take the necessary actions to improve the current situation. It is understood that the machines in the integration testbed neither require Internet access nor need to be reachable from outside CC-IN2P3 network.
- It was agreed that Yvan Calas use one of the 3 machines in the build cluster (e.g. ccqserv003) to explore the ways to produce a binary distribution.
- Interactive connection (via SSH) to all the machines in the testbed will be possible to the Qserv team through a gateway machine reachable from outside CC-IN2P3 network. This interactive access using unprivileged user credentials will help the developers to perform, on a per-node basis, typical actions of the software development cycle, such as start or stop Qserv services, modify its configuration, retrieve its log files, diagnose and debug the software. The Qserv software and its dependencies will be installed with the appropriate file permissions so that those actions can be performed using an unprivileged user account.
- CC-IN2P3 will provide a documented mechanism for the Qserv team to trigger the deployment of every new binary package, on demand or automatically, to all or a selected group of machines in the testbed. The reference binary packages will be produced from the software prepared in the build cluster. The automated deployment must include not only the software itself but also its associated configuration files.

Additional considerations

The item below was not discussed during the meeting but is recorded here for future discussion.

It is necessary to populate the Qserv database with large amounts of input data, prior to the execution and tests of the system. Similar operations were performed in the past at CC-IN2P3 using a shared file system.