

# Database Meeting 2022-01-12

## Date

12 Jan 2022

## Attendees

- Igor Gaponenko Fritz Mueller Unknown User (npease) Fabrice Jammes Andy Salnikov John Gates Joanne Bogart

## Goals

- Discussion what we know so far, complications and requirements for Qaserv operation modes and schema upgrade

## Discussion items

Time	Item	Who	Notes'
	Project news	<a href="#">Fritz Mueller</a>	<ul style="list-style-type: none"><li>"OGA rack" will be set up at USDF which will affect <a href="#">Andy Salnikov</a>'s work on APDB. Andy would need to work with Yee (SLAC) on the subject.</li><li>Next week there will be 2-days architectural meeting to discuss requirements and milestones for USDF and other data facilities: <a href="#">2nd Data Facilities Planning Workshop - 2022-01-19/20</a></li></ul>
	Progress on topics discussed at the previous meeting <a href="#">Database Meeting 2022-01-05</a>		<p><a href="#">John Gates</a> On Updates on Qserv lockups. Some progress on improving the code. Need more testing.</p> <p><a href="#">Unknown User (npease)</a> Implementing support for extended command-line attributes (in place of "-- ") seems to be complicated in the "entry points" due to restrictions of the Python command-line parser module. There is a problem with packaging multivalued arguments into a single parameter. Looking for alternatives: JSON or basic key-value pairs in the Docker-style "-e key=val".</p> <p><a href="#">Unknown User (npease)</a> Making progress on other aspects of the "entry points". There will be a PR.</p> <p><a href="#">Fabrice Jammes</a> was wondering if the refined configuration model of the "lite" container and "entry points" would be compatible with <b>qserv-operator</b>. He'd be interested in looking at the details of the effort as earlier as possible to see if there would be any complications.</p>

<p>The main topic: Qserv operation modes and schema upgrade</p>	<p>Igor Gaponenko Fritz Mueller Fabrice Jammes</p>	<p>⚠ This discussion was meant to provide Fritz with requirements and use cases for his work on the proposal.</p> <ul style="list-style-type: none"> <li>Fritz Mueller It's been recognized that Qserv deployments will need to support different modality modes to allow updates. At least 2 operation modes are known so far:             <ul style="list-style-type: none"> <li>The maintenance mode in which only the database services are being run. Schema and MySQL/MariaDB version upgrades will be made in this mode.</li> <li>The full operation node, in which Qserv is fully available for users to submit queries.</li> </ul> </li> </ul> <p>These modes will need to be implemented in the Qserv operator (not clear yet how). Kubernetes provides mechanisms for tracking the status of containers, pods, and services. However, these may not be sufficient. Qserv may need an internal monitoring system to figure a definitive state. The existing R-I system could be extended.</p> <ul style="list-style-type: none"> <li>Igor Gaponenko We may need the 3-rd mode: installing Qserv.</li> </ul> <p>This was followed by an extensive discussion on Kubernetes operator and its use in Qserv.</p> <p>Fabrice Jammes The <b>operator</b> doesn't preclude manual operations (schema upgrades) in the operator-based Qserv. This allows gaining experience and incorporating it into the operator. The manual upgrades are presently used in IDF and IN2P3.</p> <p>Discussed rolling upgrades in Qserv and the usefulness (feasibility) of this upgrade node. Two different scenarios were mentioned in this context:</p> <ul style="list-style-type: none"> <li>fixing software bugs in Qserv code (no schema changes)</li> <li>and schema/database upgrades.</li> </ul> <p>The former seems to be a good use case for the rolling upgrades. Though, there will be complications in the case when incompatible changes in the <b>czar - workers</b> protocol (Protobuf) were made. Then the rolling upgrades won't be possible, and Qserv would need to be brought down into the full maintenance mode.</p> <p>Upgrading Qserv instances serving large-scale data sets could pose a problem for the rolling upgrades as well due to the extended latency of the process (MySQL version upgrade may take many hours or days to upgrade PB-scale instances).</p> <p>A related use case (complications or simplifications?) for multiple DR (Data Releases). Each DR is an independent Qserv. There will be at least 2, and potentially many such DR instances online. There are open questions here. For example, do we need to upgrade all DR instances or just the latest one?</p> <p>Then there was a discussion on the technical aspects of implementing schema upgrades in Kubernetes. Options here:</p> <ul style="list-style-type: none"> <li>The interactive upgrade by a human operator (mentioned by Fabrice Jammes). This will require authorizing special "external" hosts and users (operators) to do the upgrade. One problem with this model is that upgrades could take quite a bit of time making the interactive mode susceptible to various risks (losing connection to the services, etc.).</li> <li>An alternative option would be to launch Kubernetes Jobs for upgrading schemas. The benefits of the jobs are that they would live log files, and it would be possible to track their progress. It should be possible to lunch jobs in the existing Kubernetes cluster.</li> </ul> <p>Igor Gaponenko schema tools need to be refined to reduce code duplication and improve the design. Also, the current model doesn't separate database <i>initialization</i> (including creating user accounts, granting privileges, and creating databases) versus schema upgrades <i>per se</i>. This issue needs to be addressed in Fritz Mueller 's proposal.</p>
---	--	--

## Action items

- ☐ Igor Gaponenko will test Qserv lockups in the **main** branch using the **large** instance of Qserv at NCSA.
- ☐ Unknown User (npease) will work with Fabrice Jammes on the configuration aspects of the "entry points"
- ☐ Fritz Mueller will prepare a proposal to present a comprehensive view of the Qserv operation modes and schema upgrade scenarios, and possibly for other management operations. The proposal will address both the Kubernetes-based and Docker-compose-based deployment scenarios.