

Table of Contents

LbSoftConfDb2.....	1
Architecture.....	1
Service Maintenance and:.....	1
FAQ:.....	2
server code update procedure:.....	2
to database migration procedure:.....	3
lbsoftdb.cern.ch new server VM initialization procedure:.....	3

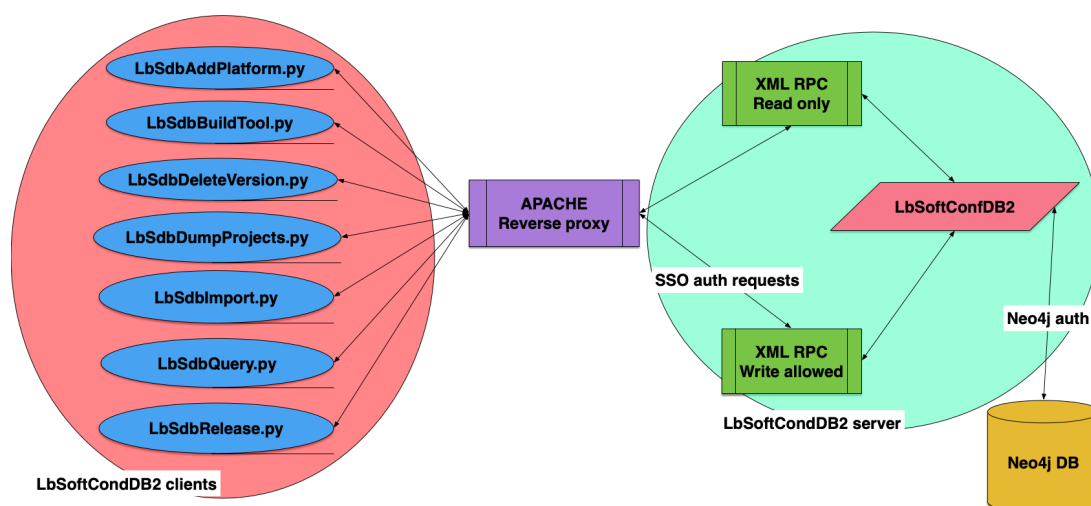
LbSoftConfDb2

Architecture

LbSoftConfDb2 is web service that provides a Neo4j database used to store the dependencies of the LHCb software projects.

The project is composed of 2 packages:

- a client side package that contains all the clients. The clients connect to the server side application using XML RPC via HTTP requests. The communication is done for clients that only query the database using non-auth requests, while the database modifications are done using SSO auth requests.
- a server side package. It provides 2 XML RPC servers: one for non-auth requests and the other one for auth-requests. The SSO auth is managed by a reverse Apache proxy. The data exchange with the Neo4j server is executed only the main component of the application.



Service Maintenance and:

Code location for libraries involved in the project:

1. LbSoftConfDb2 server: <https://gitlab.cern.ch/lhcb-core/LbSoftConfDb2Server>
2. LbSoftConfDb2 clients: <https://gitlab.cern.ch/lhcb-core/LbSoftConfDb2Clients>

The LbSoftConfDb2 server is located at lbsoftdb.cern.ch. The entry points for the communications between from the clients and server is done using 3 type of interfaces:

1. A read only interface (RO) that performs all the queries on the neo4j server without any authentication that can be access using XMLRPC requests on <https://lbsoftdb.cern.ch/read/>
2. A read write interface (RW) that performs all the inserts, updates, deletes and queries on the neo4j server with SSO authentication using the XMLRPC requests on <https://lbsoftdb.cern.ch/write/>. The SSO cookie needs to be included in the header of the requests.
3. A REST interface that is used by the nightlies to query the database:
 1. <https://lbsoftdb.cern.ch/cgi-bin/lb-projects/> - lists the projects in the databases
 2. <https://lbsoftdb.cern.ch/cgi-bin/getreleases> - lists the stacks to be built by the nightlies

For the RO and RW backends are managed by the LbSoftConfDb2 server package and the REST interface is managed by independent CGI-BIN python scripts. All the interfaces are behind an apache server as front-end. The communication with the back-end is done via reverse proxy. The SSO authentication is done by Apache shibboleth.

Demons and files locations on the SoftConfDb2 server:

1. The virtual env that contains all the python packages: `/build/env`
2. The RO interface demon:
`/etc/systemd/system/multi-user.target.wants/lbSdbServerReadOnly.service`. To start the demon use `systemctl start lbSdbServerReadOnly`. It calls the script in `/build/startLbSoftConfDb2ReadOnlyServer.sh` which activates the virtualenv and start the RO XMLRPC server on `127.0.0.1:8080`
3. The RW interface demon
`/etc/systemd/system/multi-user.target.wants/lbSdbServerWrite.service`. To start the demon use `systemctl start lbSdbServerWrite`. It calls the script in `/build/startLbSoftConfDb2WriteServer.sh` which activates the virtualenv and start the RW XMLRPC server on `127.0.0.1:8090`

FAQ:

1. How to manage Neo4j?
 - ◆ To check the Neo4j logs use: `systemctl status neo4j` for last 10 logs entries or `journalctl -r -u neo4j.service` for full logs
 - ◆ If you need to restart the demone use: `systemctl restart neo4j`
2. How to check the front-end logs?
 - ◆ `systemctl status httpd` for last 10 entries
 - ◆ `journalctl -r -u httpd.service` for full logs
3. How to check the RO server logs?
 - ◆ `systemctl status lbSdbServerReadOnly` for last 10 entries
 - ◆ `journalctl -r -u lbSdbServerReadOnly.service` for full logs
4. How to check the RW logs?
 - ◆ `systemctl status lbSdbServerWrite` for last 10 entries
 - ◆ `journalctl -r -u lbSdbServerWrite.service` for full logs
5. How to restart the front-end Apache?
 - ◆ `systemctl restart httpd`
6. How to restart the RO server?
 - ◆ `systemctl restart lbSdbServerReadOnly`
7. How to restart the RW?
 - ◆ `systemctl restart lbSdbServerWrite`

This section will be updated over time with procedures in order to facilitate the DevOps.

server code update procedure:

1. Activate the newly created virtualenv: `source ./migrationEnv/bin/activate`
2. Install the migration library: `pip install --extra-index-url https://cern.ch/lhcb-pypi/simple/ --trusted-host cern.ch lbsoftconfdb2server --upgrade`
3. Restart the lbSdbServerReadOnly demon: `systemctl restart lbSdbServerReadOnly`
4. Verify the restart was ok: `systemctl status lbSdbServerReadOnly`

...
5. Restart the lbSdbServerWrite demon: `systemctl restart lbSdbServerWrite`

6. Verify the restart was ok: `systemctl status lbSdbServerWrite`

```
...
Listening for '127.0.0.1' on port '8090'...
```

7. Restart Apache demon: `systemctl restart httpd`

to database migration procedure:

In order to migrate the LbSoftDb 1 database to the LbSoftDb2, you need to connect to any lxplus nodes and:

1. Create a new virtualenv: `virtualenv ./migrationEnv`
2. Activate the newly created virtualenv: `source ./migrationEnv/bin/activate`
3. Install the migration library: `pip install --extra-index-url https://cern.ch/lhcb-pypi/simple/ --trusted-host cern.ch lbsoftconfdbmigration`
4. Execute the migration: `lb-sdb-migration` **Attention: The new database is dropped before the migration starts.**

lbsoftdb.cern.ch new server VM initialization procedure:

LbSoftConfDb2 server installation steps (as root):

1. Start neo4j demon: `systemctl start neo4j`
2. Install pip: `easy_install pip`
3. Install virtual env: `pip install virtualenv`
4. Create a virtual environment in /build/: `virtualenv /build/env/`
5. Activate the newly created virtualenv: `source /build/env/bin/activate`
6. Install LbSoftConfDb 2 server : `pip install --extra-index-url https://cern.ch/lhcb-pypi/simple/ --trusted-host cern.ch lbsoftconfdb2server`
7. Install LbSoftConfDb 2 clients : `pip install --extra-index-url https://cern.ch/lhcb-pypi/simple/ --trusted-host cern.ch lbsoftconfdb2clients`
8. Create the RO demon start script:

```
cat <<EOT >> /build/startLBSoftConfDb2ReadOnlyServer.sh
#!/bin/bash

DIRECTORY=/build/
export HOME=/root/
source $DIRECTORY/env/bin/activate

lb-sdb-server --listen=127.0.0.1:8080
EOT
```

9. Create the RW demon start script:

```
cat <<EOT >> /build/startLBSoftConfDb2WriteServer.sh
#!/bin/bash

DIRECTORY=/build/
export HOME=/root/
source $DIRECTORY/env/bin/activate

lb-sdb-server --writeAllowed --listen=127.0.0.1:8090
EOT
```

10. Login with neo4j/neo4j to `http://lbsoftdb.cern.ch:7474` and **change the neo4j password**
11. Create credentials in /root/private: `mkdir -p /root/private & echo 'neo4j/new password' > /root/private/neo4j.txt`
12. Restart the lbSdbServerReadOnly demon: `systemctl restart lbSdbServerReadOnly`
13. Verify the restart was ok: `systemctl status lbSdbServerReadOnly`

```
...  
Listening for '127.0.0.1' on port '8080'...
```

14. Restart the lbSdbServerWrite demon: `systemctl restart lbSdbServerWrite`

15. Verify the restart was ok: `systemctl status lbSdbServerWrite`

```
...  
Listening for '127.0.0.1' on port '8090'...
```

16. Create the cgi script for release stacks:

```
cat <<EOT >> /var/www/cgi-bin/getreleases  
#!/bin/sh -x  
  
echo "Content-type: text/json"  
echo ""  
export USER=apache  
  
. /build/env/bin/activate  
  
lb-sdb-query --json listReleaseStacks  
EOT
```

17. Create the cgi script for projects listing:

```
cat <<EOT >> /var/www/cgi-bin/lb-projects  
#!/bin/sh -x  
  
echo "Content-type: text/json"  
echo ""  
export USER=apache  
  
. /build/env/bin/activate  
  
lb-sdb-dumpprojects -d  
EOT
```

18. Assign execution permissions to the cgi-bin: `chmod a+x /var/www/cgi-bin/*`

19. Verify that the apache proxy is working. Go in your browser to <https://lbsoftdb.cern.ch/read/>. If you get Service Unavailable means that the proxy is not working. The solution is to disable the SELinux: `/usr/sbin/setsebool -P httpd_can_network_connect 1`. If you get a Error response. Error code 501. means that the service is working.

20. **Congrats!** You've made it 😊

This topic: LHCb > LbSoftConfDb2

Topic revision: r4 - 2019-03-19 - StefanGabrielChitic



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback