(/)

PUBLIC | AUTOMATED BUILD

tenforce (/u/tenforce/)/virtuoso (/r/tenforce/virtuoso/)

Last pushed: 2 months ago

Repo Info (/r/tenforce/virtuoso/)

Short Description

Docker for hosting Virtuoso.

Full Description

Virtuoso docker

Docker for hosting Virtuoso.

The Virtuoso is built from a specific commit SHA in https://github.com/openlink/virtuoso-opensource (https://github.com/openlink/virtuoso-opensource).

The Docker image tags include the Virtuoso version installed in the container. The following versions are currently available (https://hub.docker.com/r/tenforce/virtuoso/tags/):

- 1.3.2-virtuoso7.2.5.1 (or virtuoso7.2.5 for latest)
- 1.3.2-virtuoso7.2.4 (or virtuoso7.2.4 for latest)
- 1.3.2-virtuoso7.2.2 (or virtuoso7.2.2 for latest)
- 1.3.2-virtuoso7.2.1 (or virtuoso7.2.1 for latest)
- 1.3.2-virtuoso7.2.0 (or virtuoso7.2.0 for latest)

Running your Virtuoso

```
docker run --name my-virtuoso \
    -p 8890:8890 -p 1111:1111 \
    -e DBA_PASSWORD=myDbaPassword \
    -e SPARQL_UPDATE=true \
    -e DEFAULT_GRAPH=http://www.example.com/my-graph \
    -v /my/path/to/the/virtuoso/db:/data \
    -d tenforce/virtuoso
```

The Virtuoso database folder is mounted in `/data` .

The Docker image exposes port 8890 and 1111.

Docker compose

The image can also be configured and used via docker-compose.

```
db:
  image: tenforce/virtuoso:1.3.1-virtuoso7.2.2
  environment:
    SPARQL_UPDATE: "true"
    DEFAULT_GRAPH: "http://www.example.com/my-graph"
  volumes:
    - ./data/virtuoso:/data
  ports:
    - "8890:8890"
```

Configuration

dba password

The `dba` password can be set at container start up via the `DBA_PASSWORD` environment variable. If not set, the default `dba` password will be used.

SPARQL update permission

The `SPARQL_UPDATE` permission on the SPARQL endpoint can be granted by setting the `SPARQL_UPDATE` environment variable to `true` .

.ini configuration

All properties defined in `virtuoso.ini` can be configured via the environment variables. The environment variable should be prefixed with `VIRT_` and have a format like `VIRT_$SECTION_$KEY` . `$SECTION` and `$KEY` are case sensitive. They should be CamelCased as in `virtuoso.ini` . E.g. property `ErrorLogFile` in the `Database` section should be configured as `VIRT_Database_ErrorLogFile=error.log` .

Dumping your Virtuoso data as quads

Enter the Virtuoso docker, open ISQL and execute the `dump_nquads` procedure. The dump will be available in `/my/path/to/the/virtuoso/db/dumps` .

```
 docker exec -it my-virtuoso bash
isql-v -U dba -P $DBA_PASSWORD
SQL> dump_nquads ('dumps', 1, 10000000, 1);
```

For more information, see http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtRDFDumpNQuad (http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtRDFDumpNQuad)

Loading quads in Virtuoso

Manually

Make the quad `.nq` files available in `/my/path/to/the/virtuoso/db/dumps` . The quad files might be compressed. Enter the Virtuoso docker, open ISQL, register and run the load.

```
 docker exec -it my-virtuoso bash
isql-v -U dba -P $DBA_PASSWORD
SQL> ld_dir('dumps', '*.nq', 'http://foo.bar');
SQL> rdf_loader_run();
```

Validate the `ll_state` of the load. If `ll_state` is 2, the load completed.

```
select * from DB.DBA.load_list;
```

For more information, see http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtBulkRDFLoader (http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtBulkRDFLoader)

Automatically

By default, any data that is put in the `toLoad` directory in the Virtuoso database folder (`/my/path/to/the/virtuoso/db/toLoad`) is automatically loaded into Virtuoso on the first startup of the Docker container. The default graph is set by the DEFAULT_GRAPH environment variable, which defaults to `http://localhost:8890/DAV` .

Creating a backup

A virtuoso backup can be created by executing the appropriate commands via the ISQL interface.

```
 docker exec -i virtuoso_container mkdir -p backups
docker exec -i virtuoso_container isql-v <<EOF
    exec('checkpoint');
        backup_context_clear();
        backup_online('backup_',30000,0,vector('backups'));
        exit;
```

Restoring a backup

To restore a backup, stop the running container and restore the database using a new container.

```
atabase:/data tenforce/virtuoso virtuoso-t +restore-backup backups/backup_ +
```

The new container will exit once the backup has been restored, you can then restart the original db container.

Contributing

Contributions to this repository are welcome, please create a pull request on the master branch.

New features will be tested on tenforce/virtuoso:latest first. Once the image is verified, version branches will be rebased on master.

Docker Pull Command

```
docker pull tenforce/virtuoso
```

Owner

tenforce

Source Repository

tenforce/docker-virtuoso (https://github.com/tenforce/docker-virtuoso)