

mod_cdr_mongodb

About

This module stores CDRs (Call Detail Record) directly to a MongoDB database.

Install & Configure

To compile the module in FreeSWITCH, uncomment the module in modules.conf and add the module to *freeswitch/conf/autoload_configs/modules.conf.xml* for autoload.

Uncomment in modules.conf:

```
event_handlers/mod_cdr_mongodb
```

Add in *freeswitch/conf/autoload_configs/modules.conf.xml*:

```
<load module="mod_cdr_mongodb" />
```

Schema Definition

Compared to other modules, such as [mod_cdr_pg_csv](#) or [mod_cdr_sqlite](#), there is no need with MongoDB to define a schema due to the nature of NoSQL

MongoDB Server Configuration

Recent versions of MongoDB (on 64-bit platforms) preallocate by default a 3GB journal. If you simply want to test this module in a non-production environment, or if you have low call volume, you can disable this journal preallocation. To do so, make sure that the journal config option is commented out, and instead add a nojournal option:

```
# Enable journaling, http://www.mongodb.org/display/DOCS/Journaling
#journal=true
nojournal=true
```

This is equivalent to launching the MongoDB server with the `--nojournal` command line option. Once the server is running with journaling disabled, you can safely delete the journal files (usually in `/var/lib/mongodb/journal`).

Module Configuration

The default configuration file can be found in *freeswitch/conf/autoload_configs/cdr_mongodb.conf.xml*.

Default values are shown.

```

<configuration name="cdr_mongodb.conf" description="MongoDB CDR logger">
  <settings>
    <!-- Specify MongoDB server in hostname[:port] format. Specify multiple
    comma-delimited hosts for a replica set -->
    <param name="host" value="127.0.0.1"/>
    <param name="port" value="27017"/>

    <!-- Optionally override default replica set name -->
    <!-- <param name="replica_set_name" value="cdr_mongodb"/> -->

    <!-- Namespace format is database.collection -->
    <param name="namespace" value="test.cdr"/>

    <!-- Set username and password to enable MongoDB authentication -->
    <!-- <param name="username" value="freeswitch"/> -->
    <!-- <param name="password" value="secret"/> -->

    <!-- If true, create CDR for B-leg of call (default: true) -->
    <param name="log-b-leg" value="false"/>
  </settings>
</configuration>

```

Authentication

If you have enabled authentication on your MongoDB server, simply uncomment and configure the username and password params.

Replica Sets

Replica sets are the preferred method of replication in MongoDB, superseding the older master/slave style replication. Replica sets support up to 12 member servers, and the configuration is largely automatic. To enable replica set support in the module, simply specify more than one server in the host param. Members can run on different ports if desired.

For example, the following will connect to a 3-member replica set, comprised of localhost and two additional servers. The third server is running on a non-standard port:

```

<param name="host" value="127.0.0.1,192.0.2.100,192.0.2.200:37017"/>
<param name="replica_set_name" value="my_replica"/>

```

Remember to configure the replica_set_name to match what your MongoDB replica set is configured as.

Unix Domain Sockets

If the MongoDB server is running on the same host as your FreeSWITCH server, you can configure the module to connect via a Unix domain socket. This may provide a small performance boost over connecting via TCP. Specify the path to the Unix domain socket in the host param, and specify a port number less than zero, e.g.:

```

<param name="host" value="/tmp/mongodb-27017.sock"/>
<param name="port" value="-1"/>

```