

# Distributed SQL Engine

- [Running the Thrift JDBC/ODBC server](#)
- [Running the Spark SQL CLI](#)

» Spark SQL can also act as a distributed query engine using its JDBC/ODBC or command-line interface. In this mode, end-users or applications can interact with Spark SQL directly to run SQL queries, without the need to write any code.

## Running the Thrift JDBC/ODBC server

The Thrift JDBC/ODBC server implemented here corresponds to the [HiveServer2](#) in built-in Hive. You can test the JDBC server with the beeline script that comes with either Spark or compatible Hive.

To start the JDBC/ODBC server, run the following in the Spark directory:

```
./sbin/start-thriftserver.sh
```

This script accepts all `bin/spark-submit` command line options, plus a `--hiveconf` option to specify Hive properties. You may run `./sbin/start-thriftserver.sh --help` for a complete list of all available options. By default, the server listens on `localhost:10000`. You may override this behaviour via either environment variables, i.e.:

```
export HIVE_SERVER2_THRIFT_PORT=<listening-port>
export HIVE_SERVER2_THRIFT_BIND_HOST=<listening-host>
./sbin/start-thriftserver.sh \
  --master <master-uri> \
  ...
```

or system properties:

```
./sbin/start-thriftserver.sh \
  --hiveconf hive.server2.thrift.port=<listening-port> \
  --hiveconf hive.server2.thrift.bind.host=<listening-host> \
  --master <master-uri>
...
```

Now you can use beeline to test the Thrift JDBC/ODBC server:

```
./bin/beeline
```

Connect to the JDBC/ODBC server in beeline with:

```
beeline> !connect jdbc:hive2://localhost:10000
```

Beeline will ask you for a username and password. In non-secure mode, simply enter the username on your machine and a blank password. For secure mode, please follow the instructions given in the [beeline documentation](#).

Configuration of Hive is done by placing your `hive-site.xml`, `core-site.xml` and `hdfs-site.xml` files in `conf/`.

You may also use the beeline script that comes with Hive.

Thrift JDBC server also supports sending thrift RPC messages over HTTP transport. Use the following setting to enable HTTP mode as system property or in `hive-site.xml` file in `conf/`:

```
hive.server2.transport.mode - Set this to value: http
hive.server2.thrift.http.port - HTTP port number to listen on; default is 10001
hive.server2.http.endpoint - HTTP endpoint; default is cliservice
```

To test, use beeline to connect to the JDBC/ODBC server in http mode with:

```
beeline> !connect jdbc:hive2://<host>:<port>/<database>?hive.server2.transport.mode=http;hive.server2.thrift.http.path=
<http_endpoint>
```

If you closed a session and do CTAS, you must set `fs.%s.impl.disable.cache` to `true` in `hive-site.xml`. See more details in [\[SPARK-21067\]](#).

## Running the Spark SQL CLI

To use the Spark SQL command line interface (CLI) from the shell:

```
./bin/spark-sql
```

For details, please refer to [Spark SQL CLI](#)

[Getting Started](#)

[Data Sources](#)

[Performance Tuning](#)

»

[Distributed SQL Engine](#)

- [Running the Thrift JDBC/ODBC server](#)
- [Running the Spark SQL CLI](#)

[PySpark Usage Guide for](#)

[Pandas with Apache Arrow](#)

[Migration Guide](#)

[SQL Reference](#)