Redis SQL

Julien Ruaux <julien@redis.com>

Version 0.2.6

Table of Contents

Chapter 1. Introduction

Redis SQL is a Trino connector which allows access to RediSearch data from Trino.

This guide provides documentation and usage information across the following topics:

- Installation
- Configuration
- Clients
- Walkthrough

Chapter 2. Installation

2.1. Trino

Trino installation instructions are available at https://trino.io/docs/current/installation.html.

2.2. RediSearch connector

Download latest release and unzip without any directory structure under `<trino>/plugin/redisearch`.

Create a RediSearch connector configuration file and change/add properties as needed.

Chapter 3. Configuration

To configure the RediSearch connector, create a catalog properties file and change/add properties as needed.

etc/catalog/redisearch.properties

```
connector.name=redisearch
redisearch.uri=redis://localhost:6379
```

Table 1. Connector properties

Property name	Description	Default
`redisearch.default-schema-name`	The schema that contains all tables defined without a qualifying schema name.	`default`
`redisearch.case-insensitive-names`	Match index names case insensitively.	`false`
`redisearch.default-limit`	Max number of documents returned by FT.SEARCH and FT.AGGREGATE when no limit is present in the SQL query.	`10000`
`redisearch.cursor-count`	Number of rows read during each aggregation cursor fetch.	`1000`

Table 2. Redis connection properties

Property name	Description	Default
`redisearch.uri`	A Redis connection string. Redis URI syntax.	
`redisearch.username`	Redis connection username.	
`redisearch.password`	Redis connection password.	
`redisearch.cluster`	Connect to a Redis Cluster.	`false`

The RediSearch connector provides additional security options to support Redis servers with TLS mode.

Table 3. TLS properties

Property name	Description	Default
`redisearch.insecure`	Allow insecure connections (e.g. invalid certificates) when using SSL.	`false`
`redisearch.cacert-path`	X.509 CA certificate file to verify with.	

Property name	Description	Default
`redisearch.key-path`	PKCS#8 private key file to authenticate with (PEM format).	
`redisearch.key-password`	Password of the private key file, or null if it's not password-protected.	
`redisearch.cert-path`	X.509 certificate chain file to authenticate with (PEM format).	

Chapter 4. Clients

4.1. JDBC Driver

The Trino JDBC driver allows users to access Trino from Java-based applications, and other non-Java applications running in a JVM.

Refer to the Trino documentation for setup instructions.

The following is an example of a JDBC URL used to create a connection to Redis SQL:

```
jdbc:trino://example.net:8080/redisearch/default
```

4.2. Tableau

Refer to the Tableau documentation for setup instructions.

4.3. Trino CLI

Refer to the Trino CLI documentation for setup instructions.

Chapter 5. Build

Run these commands to build the Trino connector for RediSearch from source (requires Java 17+):

```
git clone https://github.com/redis-field-engineering/redis-sql.git
cd Redis SQL
./mvnw clean package -DskipTests
```

Chapter 6. Complete Walkthrough

Follow these step-by-step instructions to deploy a single-node Trino server on Ubuntu.

Trino requires a 64-bit version of Java 17. It is recommended to use Azul Zulu as the JDK.

Install Java

```
$ java -version
openjdk version "17.0.4.1" 2022-08-12 LTS
OpenJDK Runtime Environment Zulu17.36+17-CA (build 17.0.4.1+1-LTS)
OpenJDK 64-Bit Server VM Zulu17.36+17-CA (build 17.0.4.1+1-LTS, mixed mode, sharing)
```

Download the Trino server tarball and unpack it.

Install Trino

```
wget https://repo1.maven.org/maven2/io/trino/trino-server/{trino-version}/trino-
server-{trino-version}.tar.gz
mkdir {trino-dir}
tar xzvf trino-server-{trino-version}.tar.gz --directory {trino-dir} --strip
-components 1
```

Trino needs a data directory for storing logs, etc. It is recommended to create a data directory outside of the installation directory, which allows it to be easily preserved when upgrading Trino.

Create a data directory

```
mkdir -p {trino-datadir}
```

Create an `etc` directory inside the installation directory to hold configuration files.

Create etc directory

```
mkdir {trino-dir}/etc
```

Create a node properties file.

etc/node.properties

```
node.environment=production
node.id=ffffffff-ffff-ffff-ffff-ffffffffffff
node.data-dir={trino-datadir}
```

Create a JVM config file.

etc/jvm.config

```
-server
-Xmx16G
-XX:InitialRAMPercentage=80
-XX:MaxRAMPercentage=80
-XX:G1HeapRegionSize=32M
-XX:+ExplicitGCInvokesConcurrent
-XX:+ExitOnOutOfMemoryError
-XX:+HeapDumpOnOutOfMemoryError
-XX:-OmitStackTraceInFastThrow
-XX:ReservedCodeCacheSize=512M
-XX:PerMethodRecompilationCutoff=10000
-XX:PerBytecodeRecompilationCutoff=10000
-Djdk.attach.allowAttachSelf=true
-Djdk.nio.maxCachedBufferSize=2000000
-XX:+UnlockDiagnosticVMOptions
-XX:+UseAESCTRIntrinsics
```

Create a config properties file.

etc/config.properties

```
coordinator=true
node-scheduler.include-coordinator=true
http-server.http.port=8080
discovery.uri=http://localhost:8080
```

Create a logging configuration file.

etc/log.properties

```
io.trino=INFO
```

Download latest release and unzip without any directory structure under `plugin/redisearch`.

Install RediSearch plugin

```
wget https://github.com/redis-field-engineering/redis-sql/releases/download/v{trino-
version}/{artifact-id}-{trino-version}.zip
unzip -j {artifact-id}-0.2.6.zip -d {trino-dir}/plugin/redisearch
```

Create a RediSearch connector configuration file.

etc/catalog/redisearch.properties

```
connector.name=redisearch
redisearch.uri=redis://localhost:6379
```

Change and/or add properties as needed.

Start the Trino server.

Run Trino server

```
{trino-dir}/bin/launcher run
```

Download trino-cli-{trino-version}-executable.jar, rename it to `trino`, make it executable with `chmod +x`, and run it to show the version of the CLI.

Install Trino CLI

```
wget https://repo1.maven.org/maven2/io/trino/trino-cli/{trino-version}/trino-cli-
{trino-version}-executable.jar
mv trino-cli-{trino-version}-executable.jar trino
chmod +x trino
```

Connect to Trino using the CLI and run a SQL query.

Run Trino CLI

```
./trino --catalog redisearch --schema default
trino:default> select * from mySearchIndex;
```