

Redis SQL Trino

Version 0.3.0

Table of Contents

Introduction.....	1
Installation.....	2
Configuration.....	3
Clients	5
Build.....	6
Complete Walkthrough.....	7

Introduction

Redis SQL Trino 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](#)

Installation

Trino

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

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.

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

Table 2. Redis connection properties

Property name	Description	Default
<code>redisearch.uri</code>	A Redis connection string. Redis URI syntax .	
<code>redisearch.username</code>	Redis connection username.	
<code>redisearch.password</code>	Redis connection password.	
<code>redisearch.cluster</code>	Connect to a Redis Cluster.	<code>false</code>
<code>redisearch.resp2</code>	Force Redis protocol version to RESP2.	<code>false</code>

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

Table 3. TLS properties

Property name	Description	Default
<code>redisearch.insecure</code>	Allow insecure connections (e.g. invalid certificates) when using SSL.	<code>false</code>

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

Clients

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
```

Tableau

Refer to the [Tableau documentation](#) for setup instructions.

Trino CLI

Refer to the [Trino CLI documentation](#) for setup instructions.

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-trino.git
cd Redis SQL Trino
./mvnw clean package -DskipTests
```


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/403/trino-server-403.tar.gz
mkdir /usr/lib/trino
tar xzvf trino-server-403.tar.gz --directory /usr/lib/trino --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 /var/trino
```

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

Create etc directory

```
mkdir /usr/lib/trino/etc
```

Create a node properties file.

etc/node.properties

```
node.environment=production
node.id=ffffffff-ffff-ffff-ffff-ffffffffffffff
node.data-dir=/var/trino
```

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:+UseAESCTRIinsics
```

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-
trino/releases/download/v403/redis-sql-trino-403.zip
unzip -j redis-sql-trino-0.3.0.zip -d /usr/lib/trino/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

```
/usr/lib/trino/bin/launcher run
```

Download [trino-cli-403-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/403/trino-cli-403-  
executable.jar  
mv trino-cli-403-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;
```