

Installing the Server

Table of Contents

- [Installing the Database](#)
- [Installing the Web Server](#)
- [Next Steps](#)
- [Troubleshooting/FAQ](#)

Installing the Database

Several database engines are supported. Be sure to follow the requirements listed for your database, they are real requirements not recommendations.

Create an empty schema and a sonarqube user. Grant this sonarqube user permissions to create, update and delete objects for this schema.

Microsoft SQL Server

⚠ Note that collation **MUST** be case-sensitive (CS) and accent-sensitive (AS).

⚠ Note that READ_COMMITTED_SNAPSHOT **MUST** be set on the SonarQube database.

MS SQL database's shared lock strategy may impact SonarQube runtime. Making sure that "*is_read_committed_snapshot_on*" is set to *true* prevents SonarQube facing potential deadlocks under heavy loads.

Example of query to check "is_read_committed_snapshot_on"

```
SELECT is_read_committed_snapshot_on FROM sys.databases WHERE name='YourSonarQubeDatabase';
```

Example of query to update "is_read_committed_snapshot_on":

```
ALTER DATABASE YourSonarQubeDatabase SET READ_COMMITTED_SNAPSHOT ON WITH ROLLBACK IMMEDIATE;
```

If you want to use integrated security, you have to download the Microsoft SQL JDBC Driver 6.2 package from <https://www.microsoft.com/en-us/download/details.aspx?id=55539> and copy sqljdbc_auth.dll to any folder in your path. You have to copy the 32 bit or 64 bit version of the dll depending upon the architecture of your server machine. If you are running the SonarQube as a windows service and want to use Integrated security, please make sure the windows account under which the service is running has permission to connect your SQL Server. The account should have db_owner database role membership. Otherwise, if you are running the SonarQube server from a command prompt and want to use Integrated security, the user under which the command prompt is running should have db_owner database role membership. Also ensure that sonar.jdbc.username or sonar.jdbc.password properties are commented, otherwise SonarQube will use SQL Authentication.

```
sonar.jdbc.url=jdbc:sqlserver://localhost;databaseName=sonar;integratedSecurity=true
```

If you want to use SQL Authentication, use the following connection string. Also ensure that sonar.jdbc.username and sonar.jdbc.password are set appropriately.

```
sonar.jdbc.url=jdbc:sqlserver://localhost;databaseName=sonar
sonar.jdbc.username=sonarqube
sonar.jdbc.password=mypassword
```

Oracle

When having two SonarQube schemas on the same Oracle instance, especially if they are of two different versions, SonarQube gets confused and picks the first it finds. To fix this issue:

- Either privileges associated to the SonarQube Oracle users should be decreased
- Or a trigger should be defined on Oracle side to automatically alter the SonarQube Oracle user session when establishing a new connection:

```
ALTER SESSION SET current_schema="MY_SONARQUBE_SCHEMA"
```



Driver Version

Oracle driver versions 12.1.0.1 and 12.1.0.2 have major bugs, and are not recommended for use with the SonarQube ecosystem ([see more details](#)).

PostgreSQL

If you want to use a custom schema and not the default "public" one, the PostgreSQL `search_path` property must be set:

```
ALTER USER mySonarUser SET search_path to mySonarQubeSchema
```

MySQL (not recommended)



Data Center Edition

MySQL is **not supported** for Data Center Edition

There are two well-known engines that can be used in MySQL: MyISAM and InnoDB. MyISAM is the oldest of the two engines and is being progressively replaced by InnoDB. InnoDB is clearly faster and scales better with SonarQube as the number of projects under quality control increases. If you were an early adopter of SonarQube, you probably have a series of table that are still using MyISAM. To improve performances, you should change the engine for all tables to InnoDB.

Once all SonarQube tables are using the InnoDB engine, the first thing to do is allocate a maximum amount of RAM to your MySQL instance with the `innodb_buffer_pool_size` parameter and give at least 15Mb to the `query_cache_size` parameter. Read this article about [InnoDB Performance Optimization Basics](#) for more information.

Installing the Web Server

First, check the [requirements](#).

Then download and unzip the [distribution](#) (do not unzip into a directory starting with a digit).

`<install_directory>` (below) refers to the path to the directory where the SonarQube distribution has been unzipped.

Setting the access to the Database

Edit `<install_directory>/conf/sonar.properties` to configure the database settings. Templates are available for every supported database. Just uncomment and configure the template you need and comment out the lines dedicated to H2:

Example for PostgreSQL

```
sonar.jdbc.username=sonarqube
sonar.jdbc.password=mypassword
sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube
```

Adding the JDBC Driver

Drivers for the supported databases (except Oracle) are already provided. Do not replace the provided drivers; they are the only ones supported.

For Oracle, copy the JDBC driver into `<install_directory>/extensions/jdbc-driver/oracle`.

Configuring the Elasticsearch storage path

By default, Elasticsearch data is stored in `<install_directory>/data`, but this is not recommended for **production instances**. Instead, you should store this data elsewhere, ideally in a dedicated volume with fast I/O. Beyond maintaining acceptable performance, doing so will also ease the upgrade of SonarQube.

Edit `<install_directory>/conf/sonar.properties` to configure the following settings:

sonar.properties

```
sonar.path.data=/var/sonarqube/data
sonar.path.temp=/var/sonarqube/temp
```

The user used to launch SonarQube **must have read and write access to those directories**.

Starting the Web Server

The default port is "9000" and the context path is "/". These values can be changed in `<install_directory>/conf/sonar.properties`.

sonar.properties

```
sonar.web.host=192.0.0.1
sonar.web.port=80
sonar.web.context=/sonar
```

Execute the following script to start the server:

- On Linux/Mac OS: `bin/<YOUR OS>/sonar.sh start`
- On Windows: `bin/windows-x86-XX/StartSonar.bat`

You can now browse SonarQube at `http://localhost:9000` (the default [System administrator](#) credentials are `admin/admin`).

Tuning the Web Server

By default, SonarQube is configured to run on any computer with a simple Java JRE.

For better performance, the first thing to do when installing a production instance is to use a Java JDK and activate the server mode by uncommenting /setting the following line in `<install_directory>/conf/sonar.properties`.

```
sonar.web.javaOpts=-server
```

To change the Java JVM used by SonarQube, simply edit `<install_directory>/conf/wrapper.conf` and update the following line:

```
wrapper.java.command=/path/to/my/jdk/bin/java
```

Advanced Installation Features

- Running SonarQube as a Service on [Windows](#) or [Linux](#)
- Running SonarQube [behind a Proxy](#)

Next Steps

Once your server is installed and running, you may also want to [Install Plugins](#). Then you're ready to begin [Analyzing Source Code](#).

Troubleshooting/FAQ

Grant more memory to the web server / compute engine / elastic search

To grant more memory to a server-side process, uncomment and edit the relevant `javaOpts` property in `SONARQUBE_HOME/conf/sonar.properties`, specifically:

- `sonar.web.javaOpts` (minimum values: `-server -Xmx768m`)
- `sonar.ce.javaOpts`
- `sonar.search.javaOpts`

Cannot connect to MySQL database

By default, remote access to MySQL database server is disabled for security reasons. If you want to remotely access the database server, you need to follow this [quick guide](#).

Failed to start on Windows Vista

SonarQube seems unable to start when installed under the *Program Files* directory on Windows Vista. It should therefore not be installed there.

Failed to start SonarQube with Oracle due to bad USERS table structure

When another(s) USERS table exists in the Oracle DB, if the sonarqube user has read access on this other USERS table, the SonarQube web server can't start and an exception like the following one is thrown:

```
ActiveRecord::ActiveRecordError: ORA-00904: "TOTO": invalid identifier
: INSERT INTO users (login, name, email, crypted_password, salt,
created_at, updated_at, remember_token, remember_token_expires_at, toto, id)
VALUES('admin', 'Administrator', '', 'bba4c8a0f808f9798cf8b1c153a4bb4f9178cf59',
'2519754f77ea67e5d7211cd1414698f465aacebb',
TIMESTAMP'2011-06-24 22:09:14', TIMESTAMP'2011-06-24 22:09:14', null, null, null, ?)
ActiveRecord::ActiveRecordError: ORA-00904: "TOTO": invalid identifier

: INSERT INTO users (login, name, email, crypted_password, salt,
created_at, updated_at, remember_token, remember_token_expires_at, toto, id)
VALUES('admin', 'Administrator', '', 'bba4c8a0f808f9798cf8b1c153a4bb4f9178cf59',
'2519754f77ea67e5d7211cd1414698f465aacebb', TIMESTAMP'2011-06-24 22:09:14', TIMESTAMP'2011-06-24 22:09:14',
null, null, null, ?)
```

To fix this issue, the rights of the sonarqube Oracle user must be decreased to remove read access on the other USERS table(s).

Failed to connect to the Update Center via proxy

Double check that settings for proxy in file *SONARQUBE_HOME/conf/sonar.properties* are correctly set.

Note that if your proxy username contains "\" (backslash), then it should be escaped - for example username "domain\\user" in file should look like:

```
http.proxyUser=domain\\user
```

For some proxies, exception "java.net.ProtocolException: Server redirected too many times" might mean an incorrect username or password has been configured.