



# QuartzDesk Web Application Installation and Upgrade Guide for WildFly AS 10.x – 18.x

QuartzDesk Version: 4.x

March 3, 2020



## Table of Contents

<b>1.</b>	<b>PURPOSE .....</b>	<b>3</b>
<b>2.</b>	<b>DEFINITIONS .....</b>	<b>4</b>
<b>3.</b>	<b>REQUIREMENTS.....</b>	<b>5</b>
3.1	SOFTWARE REQUIREMENTS .....	5
3.1.1	<i>Browser .....</i>	<i>5</i>
3.1.2	<i>Operating System .....</i>	<i>5</i>
3.1.3	<i>JVM .....</i>	<i>5</i>
3.1.4	<i>Application Server.....</i>	<i>5</i>
3.1.5	<i>Database .....</i>	<i>5</i>
3.1.6	<i>Database JDBC Driver .....</i>	<i>6</i>
3.1.7	<i>QuartzDesk Web Application Archive.....</i>	<i>6</i>
3.2	HARDWARE REQUIREMENTS.....	6
<b>4.</b>	<b>INSTALLATION.....</b>	<b>7</b>
4.1	DATABASE.....	7
4.2	JDBC DRIVER .....	7
4.3	JDBC DATASOURCE.....	7
4.3.1	<i>DB2 .....</i>	<i>8</i>
4.3.2	<i>H2.....</i>	<i>13</i>
4.3.3	<i>Microsoft SQL Server.....</i>	<i>19</i>
4.3.4	<i>MySQL.....</i>	<i>25</i>
4.3.5	<i>Oracle.....</i>	<i>31</i>
4.3.6	<i>PostgreSQL.....</i>	<i>34</i>
4.4	APPLICATION WORK DIRECTORY .....	39
4.5	APPLICATION CONFIGURATION.....	40
4.6	DEPLOY APPLICATION.....	41
4.7	START APPLICATION .....	44
<b>5.</b>	<b>UPGRADING .....</b>	<b>47</b>
5.1	STOP EXISTING APPLICATION .....	47
5.2	BACKUP.....	48
5.3	REMOVE EXISTING APPLICATION .....	48
5.4	DEPLOY NEW APPLICATION.....	48
5.5	START NEW APPLICATION .....	49
<b>6.</b>	<b>QUARTZDESK 2.X TO 3.X MIGRATION NOTES .....</b>	<b>50</b>
6.1	MINIMUM REQUIRED JAVA VERSION.....	50
6.2	RENAME CONFIGURATION FILE .....	50
6.3	RENAME LOG FILES .....	50
6.4	ACCESS TO MONITORING URLS (REST API) .....	51
6.5	ACCESS TO JAX-WS ENDPOINTS .....	52
6.6	REMOVE UNUSED SECURITY DOMAIN.....	53
<b>7.</b>	<b>QUARTZDESK 3.X TO 4.X MIGRATION NOTES .....</b>	<b>54</b>
<b>8.</b>	<b>CLUSTER DEPLOYMENT NOTES.....</b>	<b>55</b>
8.1	HTTP SESSION REPLICATION AND AFFINITY .....	55
8.2	SHARED WORK DIRECTORY.....	55
8.3	LOGGING CONFIGURATION .....	55
8.3.1	<i>Using Shared Log Files.....</i>	<i>56</i>
8.3.2	<i>Using Separate Log Files.....</i>	<i>57</i>
8.4	INTERNAL QUARTZ SCHEDULER .....	58

## 1. Purpose

This document describes the installation and upgrade process for QuartzDesk Web Application 4.x on WildFly Application Server 10.x–18.x running in the **standalone** mode.

The installation and upgrade process in the **domain** mode is similar and is not described herein.

If you experience any problems installing or upgrading QuartzDesk Web Application, please let us know at [support@quartzdesk.com](mailto:support@quartzdesk.com).



## 2. Definitions

The following table lists all acronyms and shortcuts used throughout this document.

Acronym / Shortcut	Definition
AS	Application Server.
EAR	Enterprise Application Archive. A file with <code>.ear</code> extension.
JAR	Java Application Archive. A file with <code>.jar</code> extension.
JVM	Java Virtual Machine.
WFAC	WildFly Administrative Console.
WFAS	WildFly Application Server.
WAR	Web Application Archive. A file with <code>.war</code> extension.

The following table lists all locations and properties used throughout this document.

Location / Property	Example	Description
DB_HOST	localhost	QuartzDesk Web Application database server host.
DB_PORT	5432	QuartzDesk Web Application database server port.
DB_NAME	quartzdesk	QuartzDesk Web Application database name.
DB_SCHEMA	quartzdesk	QuartzDesk Web Application database schema.
DB_USER	quartzdesk	QuartzDesk Web Application database user.
DB_PASSWORD	quartzdesk	QuartzDesk Web Application database user password.
WFAS_INSTALL_ROOT	<code>/usr/local/wildfly</code>	WildFly Application Server installation directory.
WFAS_CONFIG	standalone	WildFly Application Server configuration.
WFAS_HTTP_HOST	localhost	WildFly HTTP listener host.
WFAS_HTTP_PORT	8080	WildFly HTTP listener port.
WORK_DIR	<code>/var/quartzdesk-web.work</code>	QuartzDesk Web Application work directory.

## 3. Requirements

### 3.1 Software Requirements

#### 3.1.1 Browser

The QuartzDesk Web Application's GUI requires a modern JavaScript-enabled browser. Please make sure JavaScript is enabled and not blocked by third party anti-virus/anti-malware software.

QuartzDesk Web Application has been tested with the following browser versions. These are also the minimum browsers versions required.

Browser	Minimum Version
Chrome	64
Firefox	45
Internet Explorer	8
Microsoft Edge	12
Opera	43
Safari	10

#### 3.1.2 Operating System

Windows 7, Windows 8, Windows 10.

Linux (any distribution) with kernel 2.6.x and above.

Solaris 11.x and above.

#### 3.1.3 JVM

Oracle JDK 8–13.

IBM JDK 8.

OpenJDK 8–13.

#### 3.1.4 Application Server

WildFly Application Server 10.x.

WildFly Application Server 11.x.

WildFly Application Server 12.x.

WildFly Application Server 13.x.

WildFly Application Server 14.x.

WildFly Application Server 15.x.

WildFly Application Server 16.x.

WildFly Application Server 17.x.

WildFly Application Server 18.x.

#### 3.1.5 Database

Database	Minimum Version
DB2	10.1
H2	1.3.174
Microsoft SQL Server	2008 R2 SP1

MySQL	5.6.4
Oracle	10.2 (10g R2)
PostgreSQL	8.1

### 3.1.6 Database JDBC Driver

Database	JDBC Driver
DB2	IBM DB2 JDBC 4.0 driver available at <a href="http://www-01.ibm.com/support/docview.wss?uid=swg21363866">http://www-01.ibm.com/support/docview.wss?uid=swg21363866</a> .
H2	Database engine including the JDBC driver is available at <a href="http://www.h2database.com">http://www.h2database.com</a> .
Microsoft SQL Server	Microsoft JDBC driver 4.0 for SQL Server available at <a href="http://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx">http://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx</a> .  We strongly advise against using the alternative JTDS JDBC driver because it does not support the datetime2 data type at this time. As a result, all datetime values written by QuartzDesk Web Application would end up rounded up, or down. For datetime data type rounding details, please refer to <a href="http://msdn.microsoft.com/en-us/library/ms187819.aspx">http://msdn.microsoft.com/en-us/library/ms187819.aspx</a> .
MySQL	Connector/J JDBC driver available at <a href="http://dev.mysql.com/downloads/connector/j/">http://dev.mysql.com/downloads/connector/j/</a> .
Oracle	Oracle JDBC driver available at <a href="http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html">http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html</a> .  For a comprehensive overview of JDBC driver versions vs. supported database versions, please refer to <a href="http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-faq-090281.html#01_02">http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-faq-090281.html#01_02</a> .
PostgreSQL	JDBC4 PostgreSQL driver available at <a href="http://jdbc.postgresql.org/">http://jdbc.postgresql.org/</a> .

### 3.1.7 QuartzDesk Web Application Archive

To install QuartzDesk Web Application, you need to obtain the quartzdesk-web-x.y.z.war file. The latest version can be downloaded at [www.quartzdesk.com](http://www.quartzdesk.com) (click Downloads → Latest Release → View files → quartzdesk-web-x.y.z.war).

## 3.2 Hardware Requirements

QuartzDesk Web Application runs on any physical or virtualized hardware that supports the above software requirements.

## 4. Installation

This chapter describes the standard QuartzDesk Web Application installation. If you are only evaluating, you can run QuartzDesk Web Application in the **one-step installation mode** to dramatically reduce the number of required installation steps. For details, please see our [FAQs](#) and search for “one-step installation”.

### 4.1 Database

Create a new database user named `quartzdesk` (`DB_USER`) with an arbitrary password (`DB_PASSWORD`).

Create a new QuartzDesk Web Application database named `quartzdesk1` (`DB_NAME`) owned by `DB_USER`.

In the `quartzdesk` database create a new schema named `quartzdesk` (`DB_SCHEMA`). The schema must be owned by `DB_USER`. Make the created `DB_SCHEMA` the default schema of `DB_USER` and/or add the schema to the `DB_USER`'s schema search path.

Please contact your DBA, or refer to the database engine documentation for instructions on how to complete the above database-specific tasks.



Please note that you do not have to create any database objects (tables, keys, indices etc.) in the `quartzdesk` database / schema. These objects will be automatically created by QuartzDesk Web Application during its first start.

### 4.2 JDBC Driver

Download and install the JDBC driver for the created database. For a list of supported JDBC drivers please refer to chapter 3.1.6.

Copy the JDBC driver JAR file(s) to `WFAS_INSTALL_ROOT/WFAS_CONFIG/deployments` directory and restart the application server.



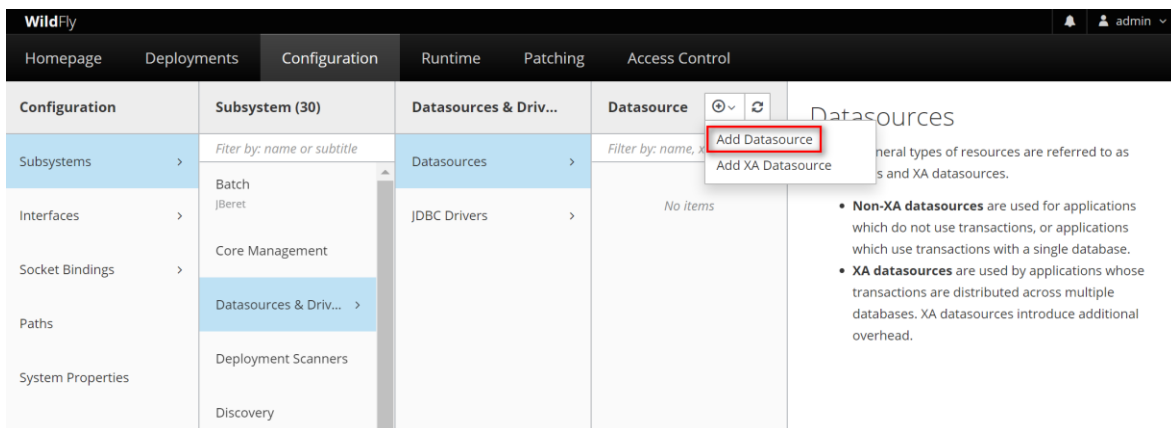
To install the H2 JDBC driver, do not copy it to `WFAS_INSTALL_ROOT/WFAS_CONFIG/deployments` directory because the H2 driver is already installed as a module (`JAS_INSTALL_ROOT/modules/system/layers/base/com/h2database/h2`). To update the existing driver module JAR, copy the new H2 driver JAR to the module's main directory and edit the module descriptor file `module.xml` to update the JAR name.

### 4.3 JDBC Datasource

In WFAc select Configuration → Subsystems → Datasources & Drivers → Datasources and click the Add button in the Datasource column and select the Add Datasource menu option to create a new JDBC datasource.

---

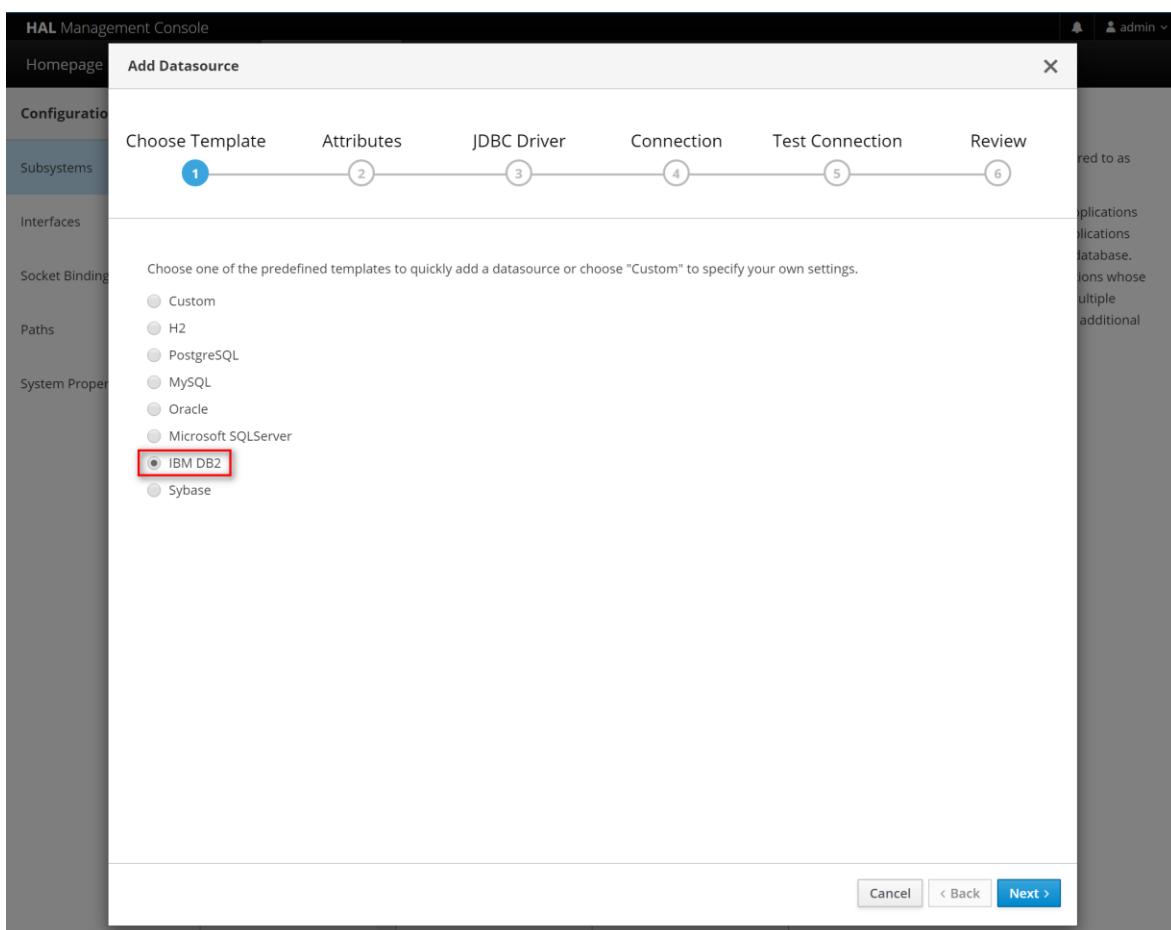
<sup>1</sup> If you use DB2, the database name length is restricted to the maximum of 8 characters. Please adjust the database name accordingly (e.g. `qdesk`).



The next steps depend on the QuartzDesk Web Application database type and are described in the following sub-chapters.

### 4.3.1 DB2

In the Create Datasource dialog, select the IBM DB2 option.



Click Next.

In Step 2, enter the following datasource attributes:

Name: QuartzDeskDS

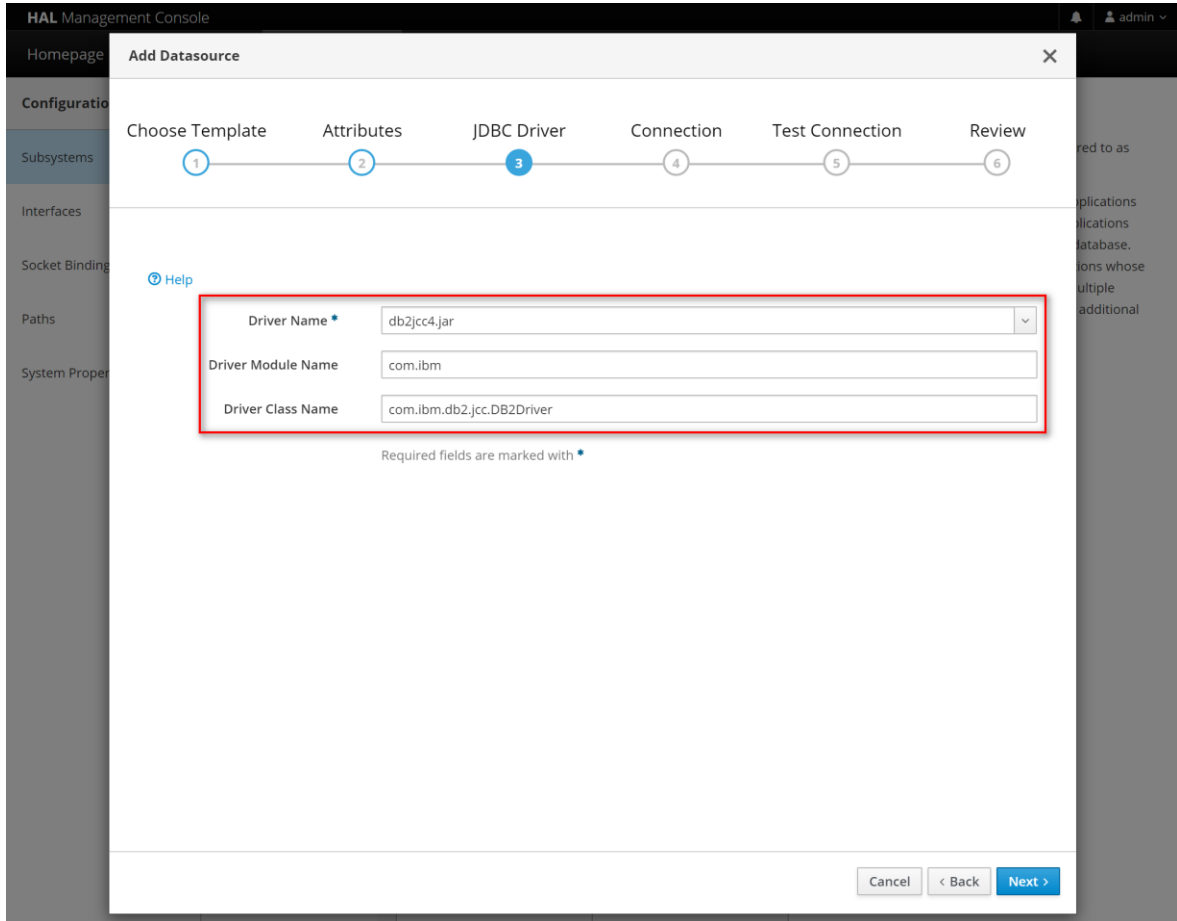
JNDI Name: java:/jdbc/QuartzDeskDS



In Step 3, select the installed DB2 JDBC driver JAR in the Driver Name field and enter the following values:

Driver Module Name: com.ibm

Driver Class Name: com.ibm.db2.jcc.DB2Driver



Click Next.

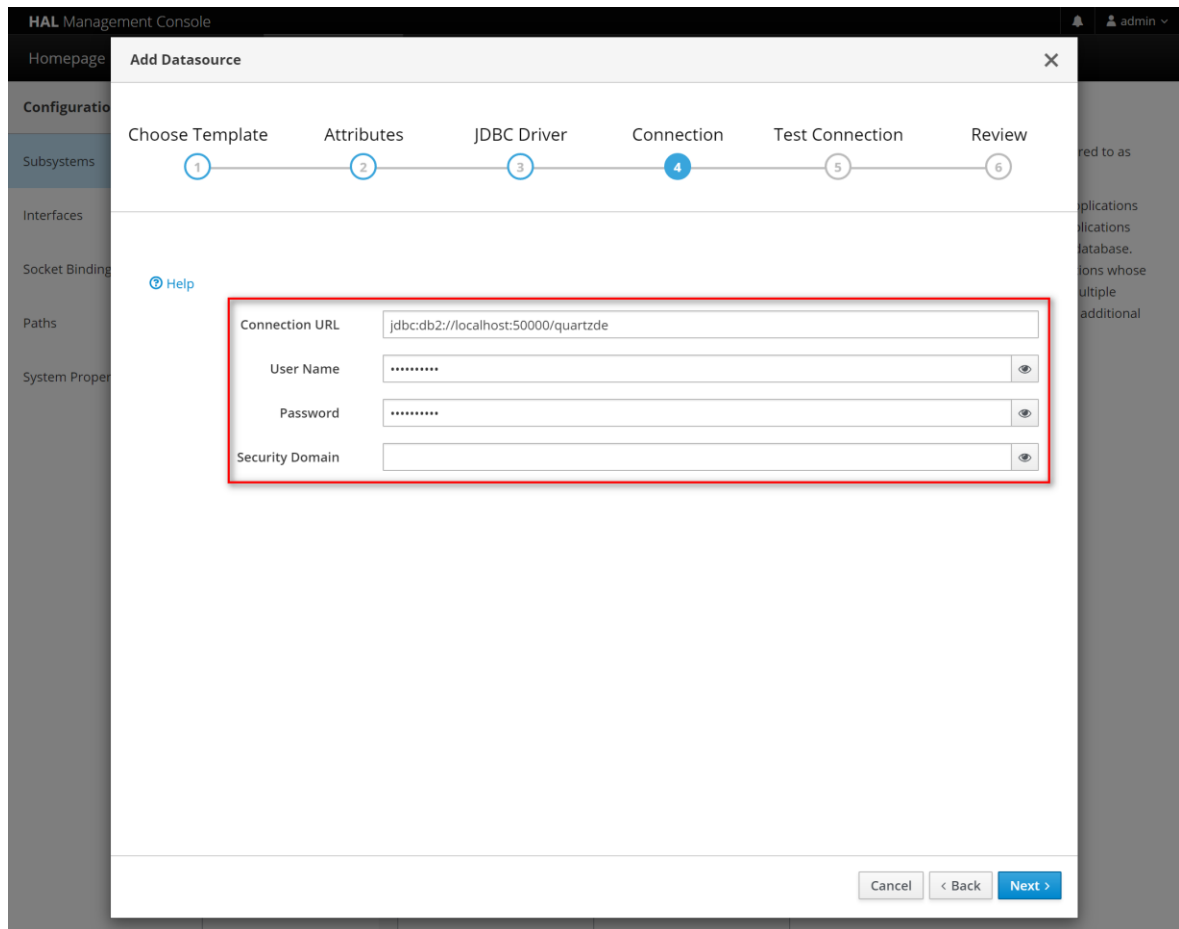
In Step 4, enter the following values:

Connection URL: jdbc:db2://DB\_HOST:DB\_PORT/DB\_NAME

Username: DB\_USER

Password: DB\_PASSWORD

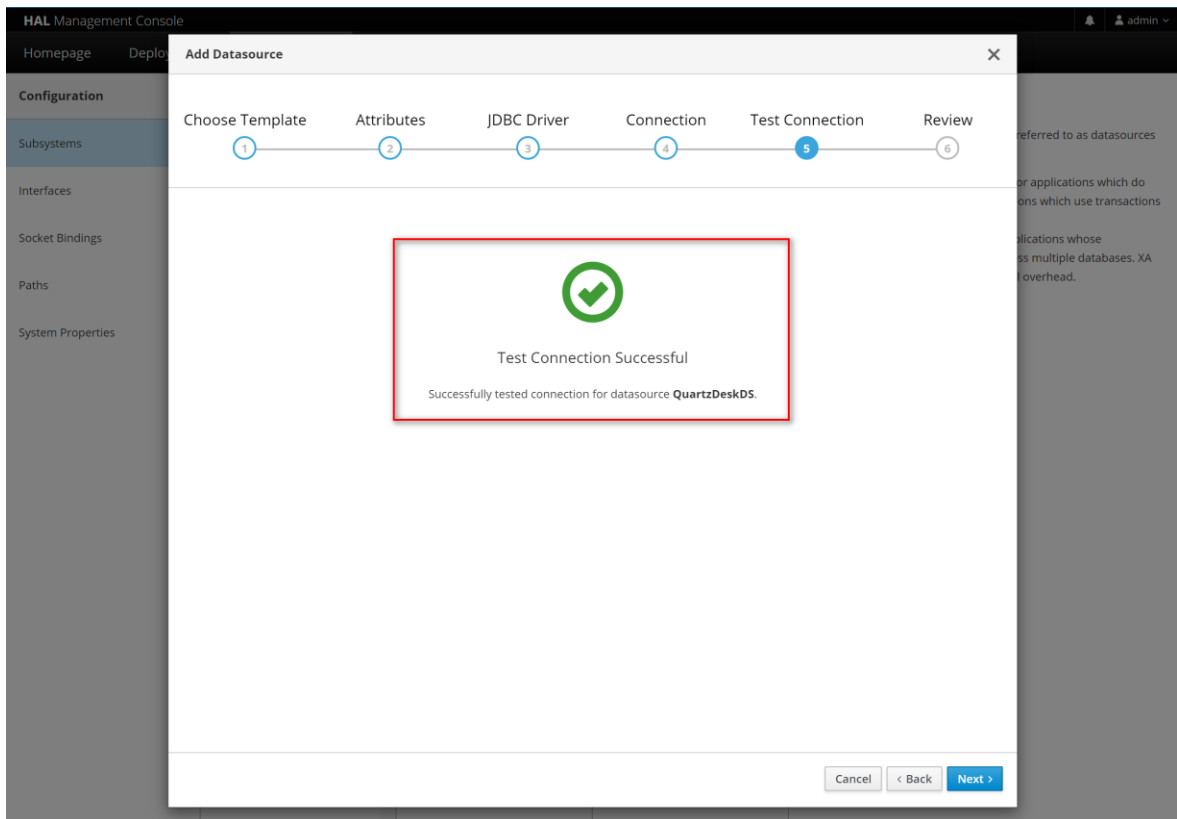
Security Domain: leave empty



Click Next.

In Step 5, test the datasource by clicking on the Test Connection button.

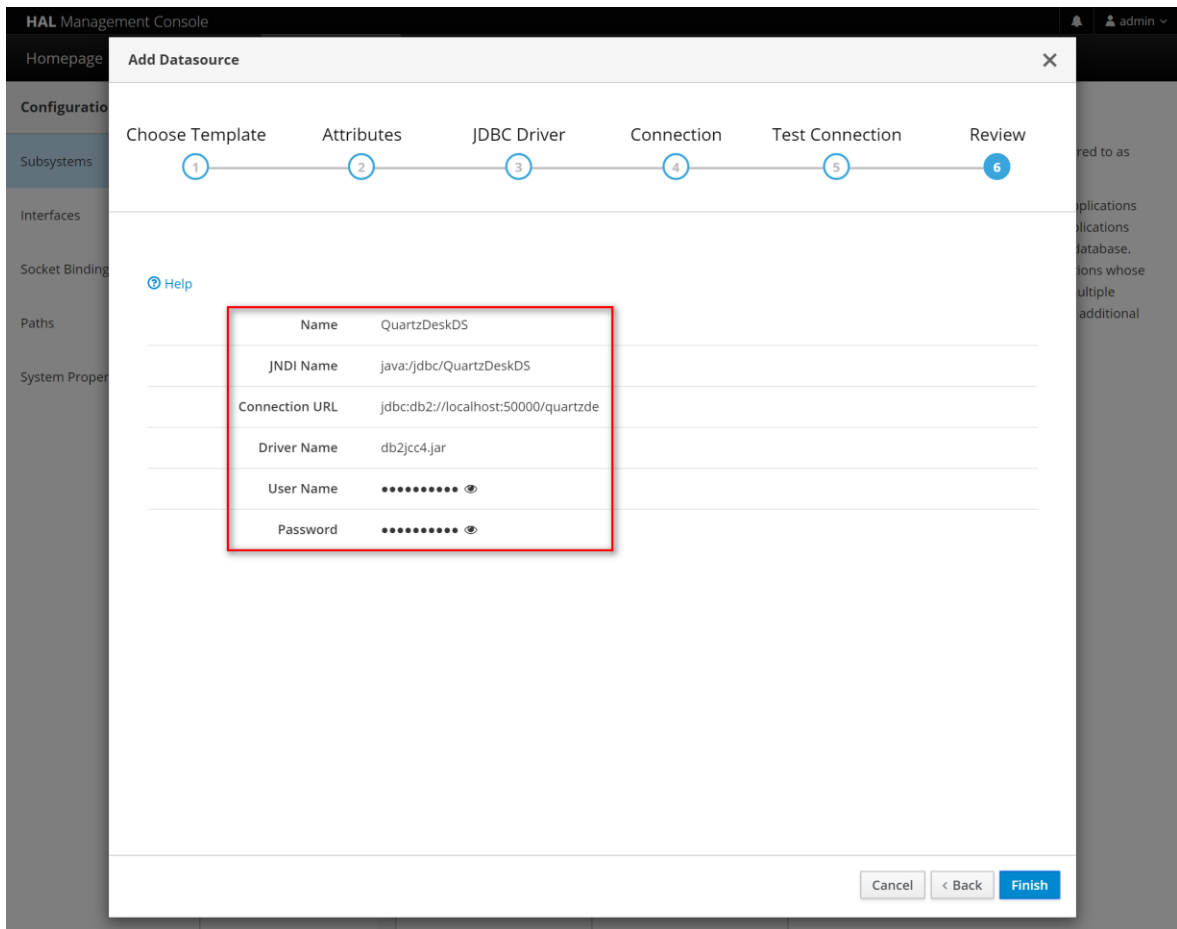




Click Next.

In Step 6, review the datasource parameters.





Click Finish.

Select the registered QuartzDeskDS datasource, click View and modify the datasource configuration under the following tabs:

### Connection

Add the following property:

clientApplicationInformation=QuartzDesk Web Application

Click Save.

### Pool

Click the Edit button and enter the following values:

Initial Pool Size: 2

Min Pool Size: 2

Max Pool Size: 10

Pool Prefill: On

Click Save.

### Validation

Click the Edit button and enter the following values:

Valid Connection Checker Class Name:

org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker

Stale Connection Checker Class Name:

org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionChecker

Exception Sorter Class Name:

org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter

Click Save.

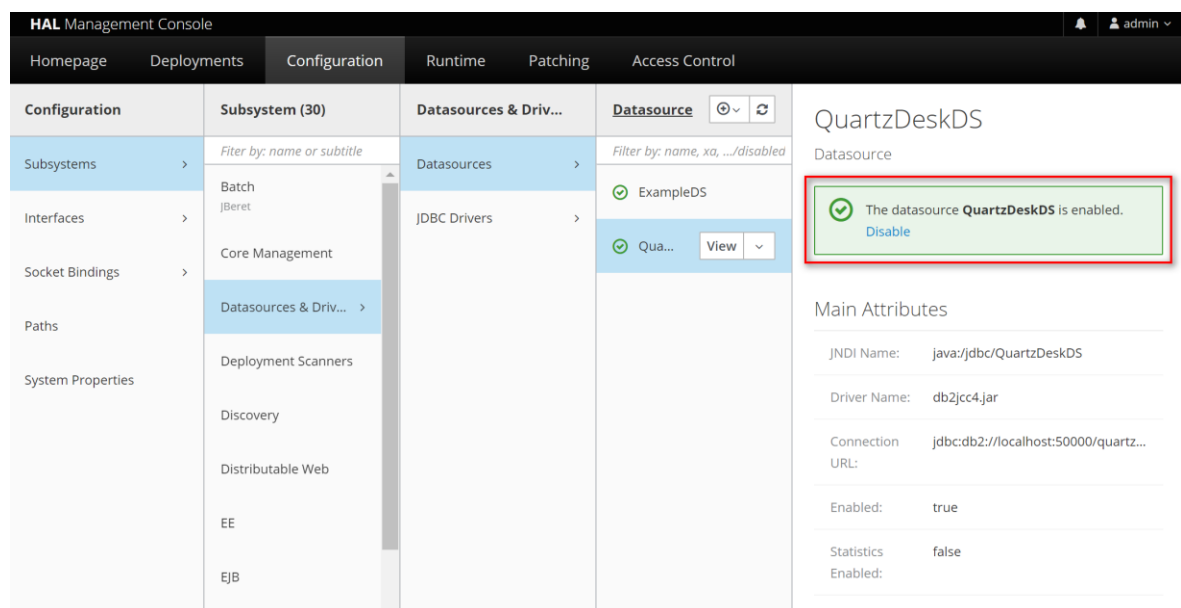
### Statements / Tracking

Click the Edit button and enter the following values:

Prepared Statements Cache Size: 100

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.



The screenshot shows the HAL Management Console interface. The top navigation bar includes 'Homepage', 'Deployments', 'Configuration', 'Runtime', 'Patching', and 'Access Control'. The 'Configuration' tab is active, and the 'Datasources & Drivers' section is expanded. The 'QuartzDeskDS' datasource is selected, and its configuration is displayed on the right. A green notification box with a checkmark icon and the text 'The datasource QuartzDeskDS is enabled. Disable' is visible, indicating that the datasource is successfully enabled. The main attributes for the datasource are listed below the notification:

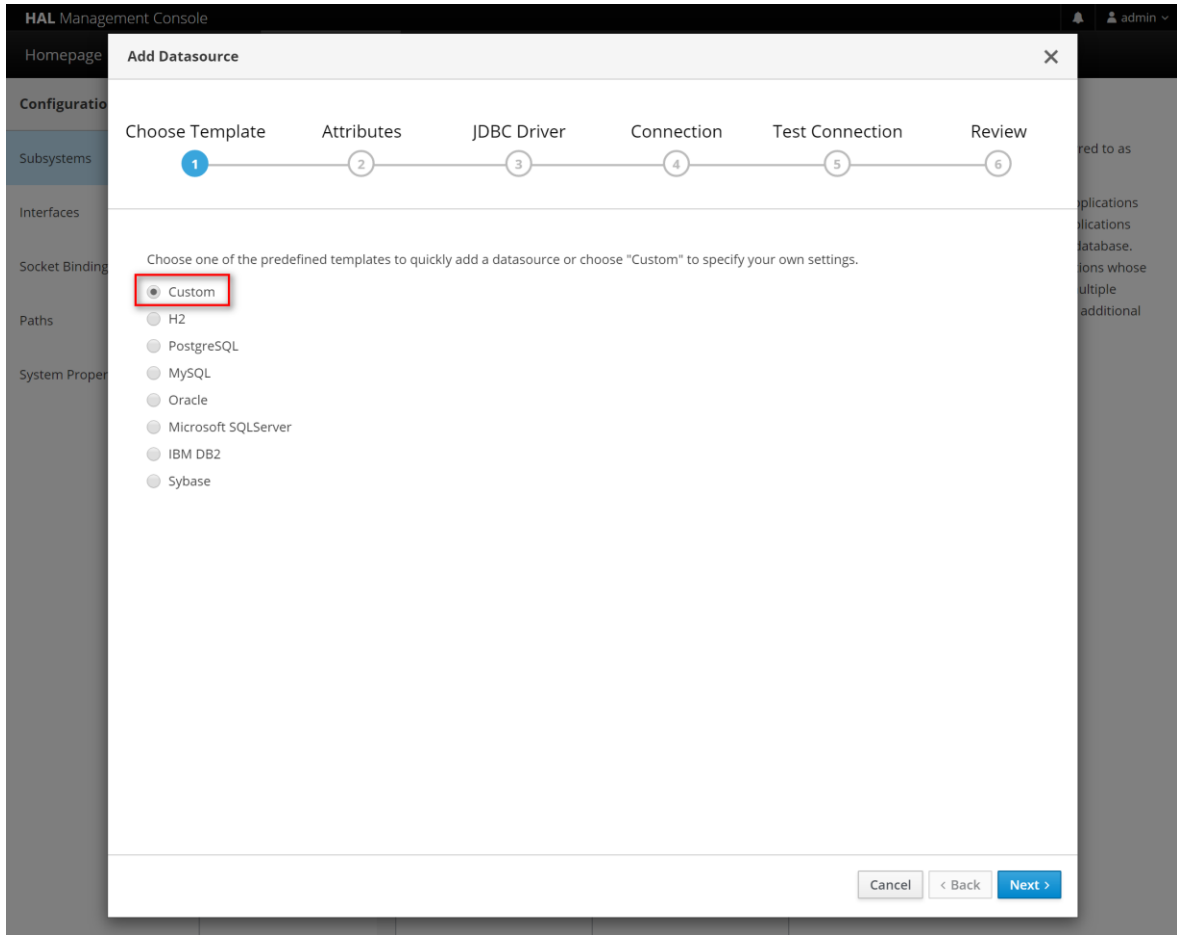
Main Attributes	
JNDI Name:	java:/jdbc/QuartzDeskDS
Driver Name:	db2jcc4.jar
Connection URL:	jdbc:db2://localhost:50000/quartz...
Enabled:	true
Statistics Enabled:	false

### 4.3.2 H2



H2 is a light-weight Java database with limited fault tolerance and recovery functionality. We recommend using H2 for evaluation and experimental purposes only.

In the Create Datasource dialog, select the Custom option.



Click Next.

In Step 2, enter the following datasource attributes:

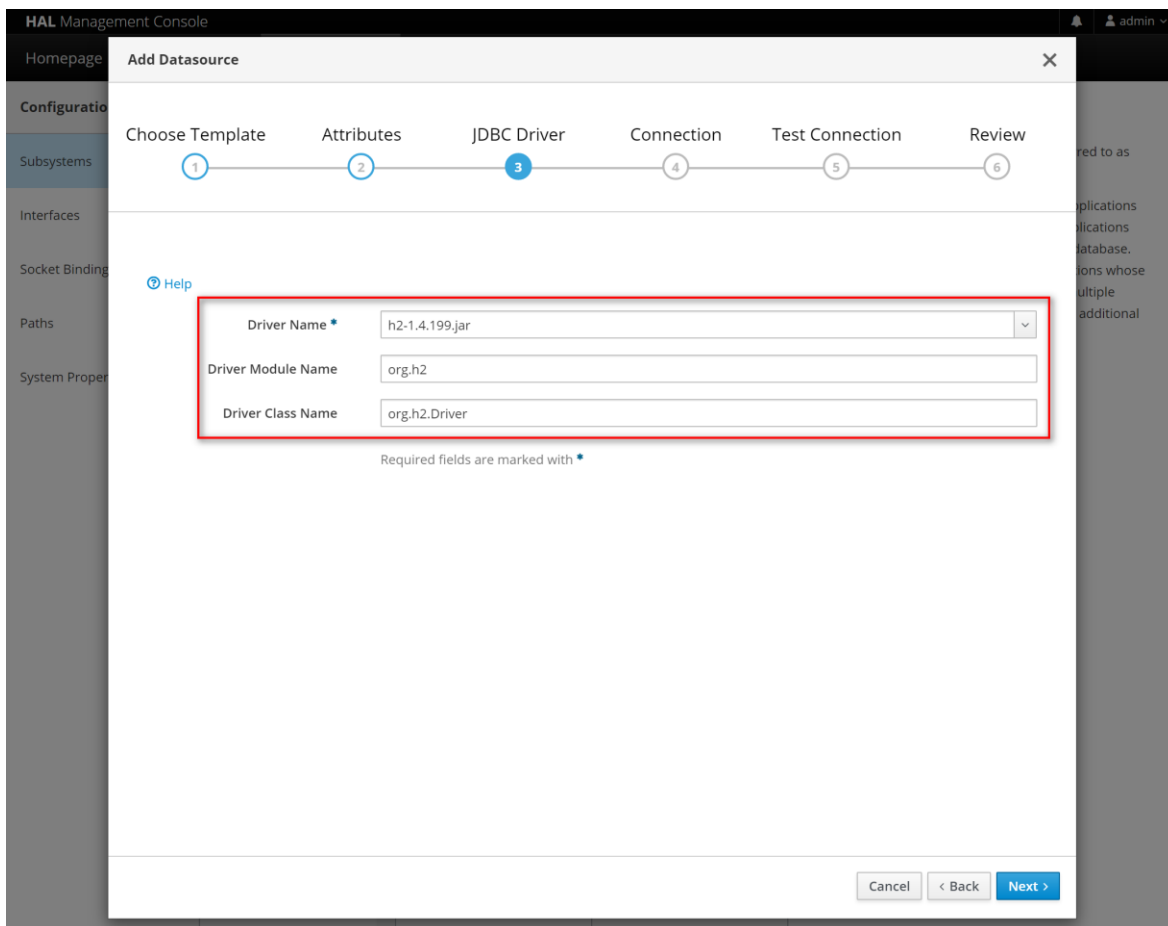
Name: QuartzDeskDS

JNDI Name: java:/jdbc/QuartzDeskDS

In Step 3, select the installed H2 JDBC driver JAR in the Driver Name field and enter the following values:

Driver Module Name: org.h2

Driver Class Name: org.h2.Driver



Click Next.

In Step 4, enter the following values:

Connection URL: jdbc:h2:file:<H2\_DB\_FILE\_PATH>

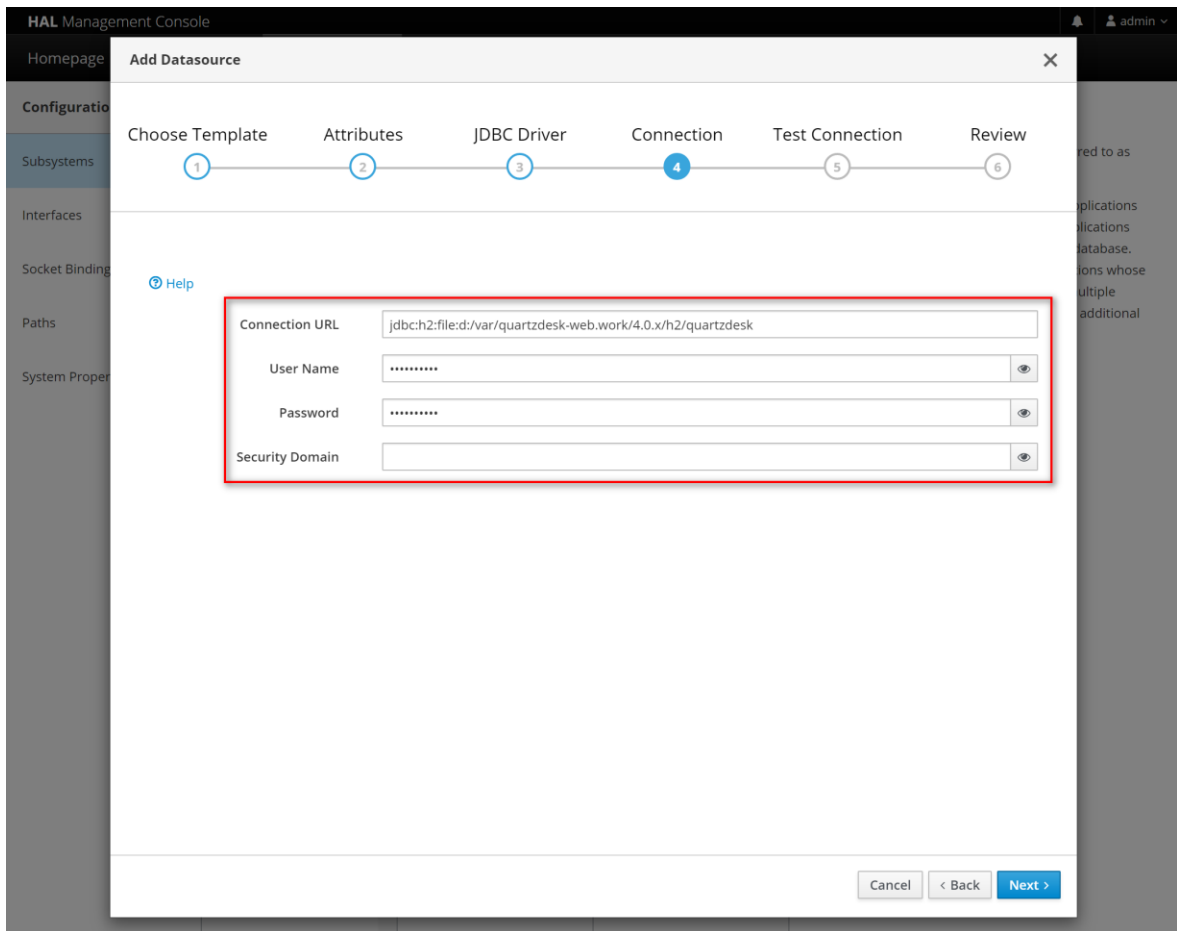
Username: DB\_USER

Password: DB\_PASSWORD

Security Domain: leave empty

Please note that H2 can be configured to run in various operating modes by adjusting the Connection URL value. For details, please refer to the H2 documentation at

[http://www.h2database.com/html/features.html#database\\_url](http://www.h2database.com/html/features.html#database_url).

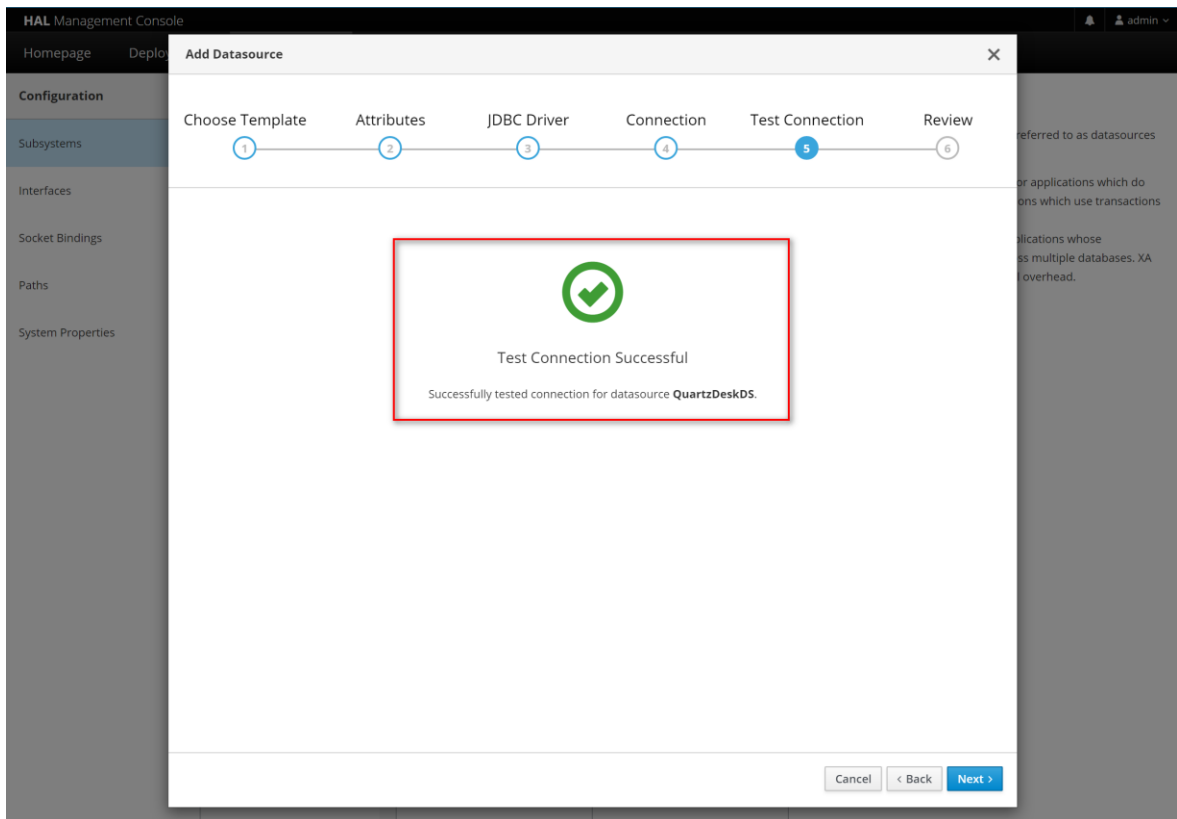


Click Next.

In Step 5, test the datasource by clicking on the Test Connection button.



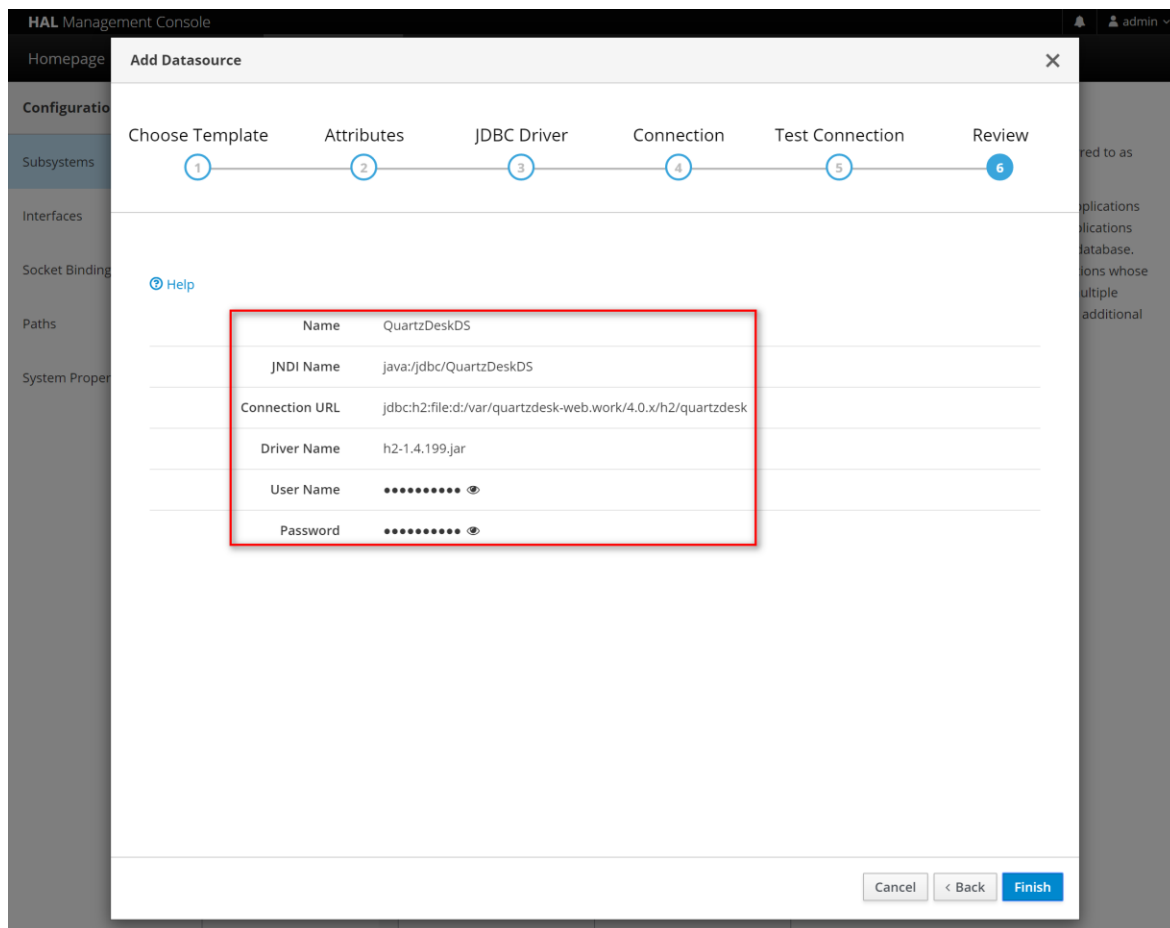




Click Next.

In Step 6, review the datasource parameters.





Click Finish.

Select the registered QuartzDeskDS datasource, click View and modify the datasource configuration under the following tabs:

### Connection

Add the following property:

applicationName=QuartzDesk Web Application

Click Save.

### Pool

Click the Edit button and enter the following values:

Initial Pool Size: 2

Min Pool Size: 2

Max Pool Size: 10

Pool Prefill: On

Click Save.

### Validation

Click the Edit button and enter the following values:

Valid Connection Checker Class Name:  
`org.jboss.jca.adapters.jdbc.extensions.novendor.JDBC4ValidConnectionChecker`

Stale Connection Checker Class Name:  
`org.jboss.jca.adapters.jdbc.extensions.novendor.NullStaleConnectionChecker`

Exception Sorter Class Name:  
`org.jboss.jca.adapters.jdbc.extensions.novendor.NullExceptionSorter`

Click Save.

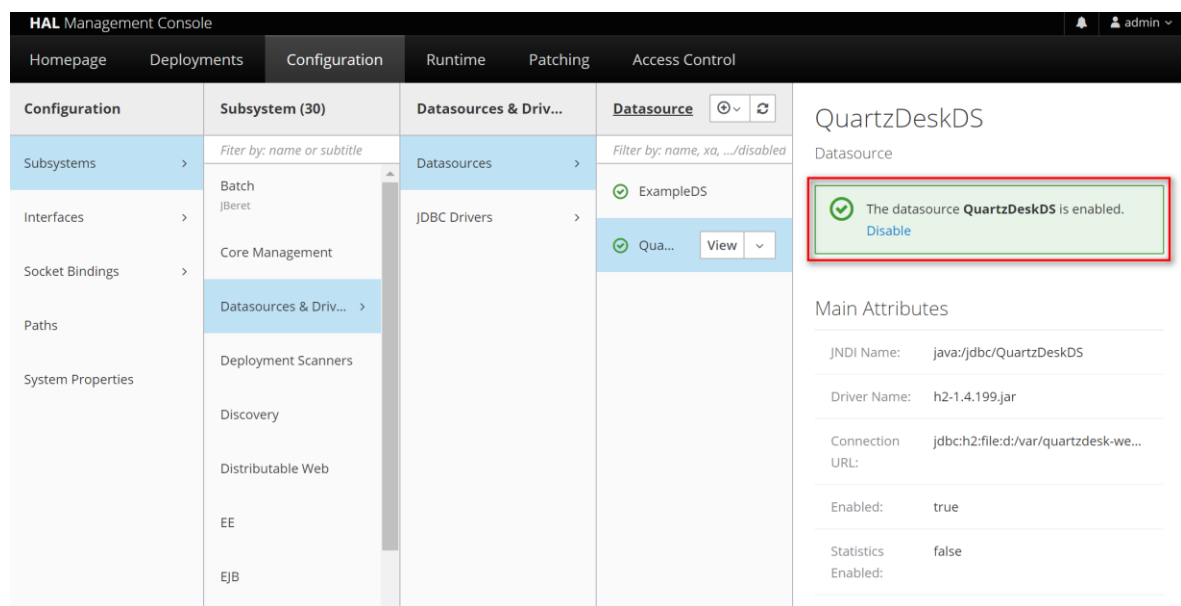
### Statements / Tracking

Click the Edit button and enter the following values:

Prepared Statements Cache Size: 100

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.

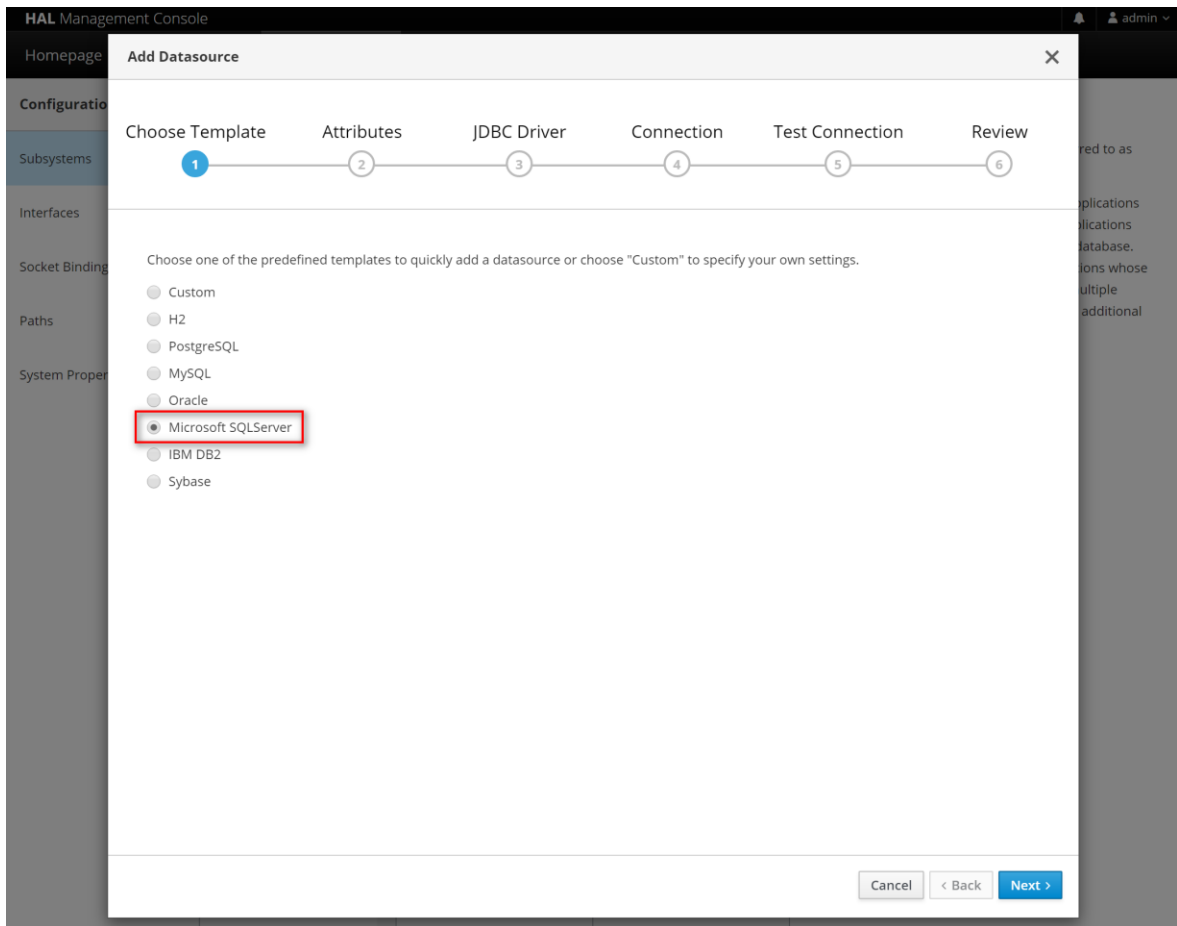


The screenshot shows the HAL Management Console interface. The top navigation bar includes 'Homepage', 'Deployments', 'Configuration', 'Runtime', 'Patching', and 'Access Control'. The 'Configuration' tab is active, showing a tree view of subsystems on the left. The 'Databases & Drivers' section is expanded, showing 'Databases' and 'JDBC Drivers'. The 'Databases' section is further expanded, showing 'ExampleDS' and 'QuartzDeskDS'. The 'QuartzDeskDS' entry is selected, and its configuration details are shown on the right. A red box highlights a green notification that says 'The datasource QuartzDeskDS is enabled. Disable'. Below the notification, the 'Main Attributes' section is visible, showing the following configuration:

Attribute	Value
JNDI Name	java:/jdbc/QuartzDeskDS
Driver Name	h2-1.4.199.jar
Connection URL	jdbc:h2:file:d:/var/quartzdesk-we...
Enabled	true
Statistics Enabled	false

### 4.3.3 Microsoft SQL Server

In the Create Datasource dialog, select the Microsoft SQLServer option.



Click Next.

In Step 2, enter the following datasource attributes:

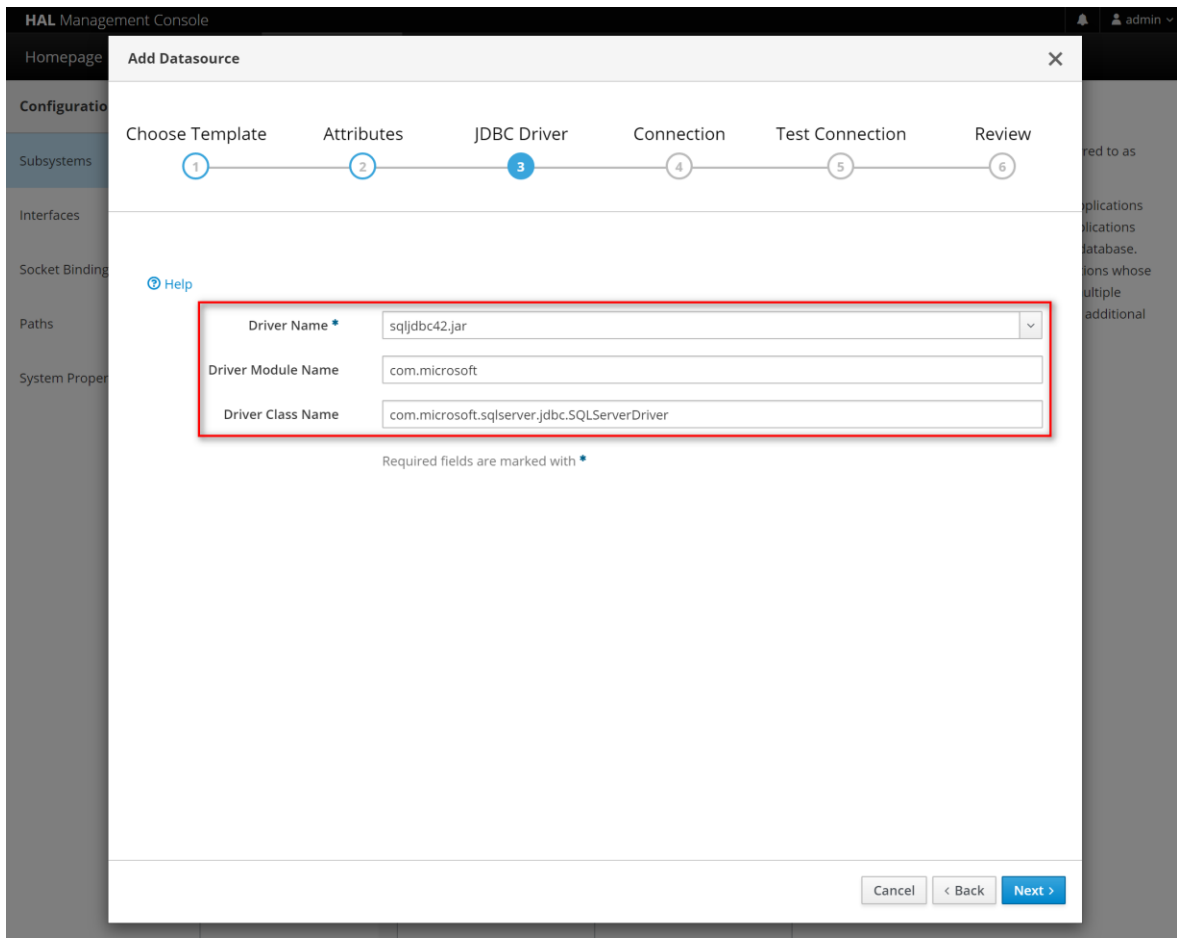
Name: QuartzDeskDS

JNDI Name: java:/jdbc/QuartzDeskDS

In Step 3, select the installed Microsoft SQL Server JDBC driver JAR in the Driver Name field and enter the following values:

Driver Module Name: com.microsoft

Driver Class Name: com.microsoft.sqlserver.jdbc.SQLServerDriver



Click Next.

In Step 4, enter the following values:

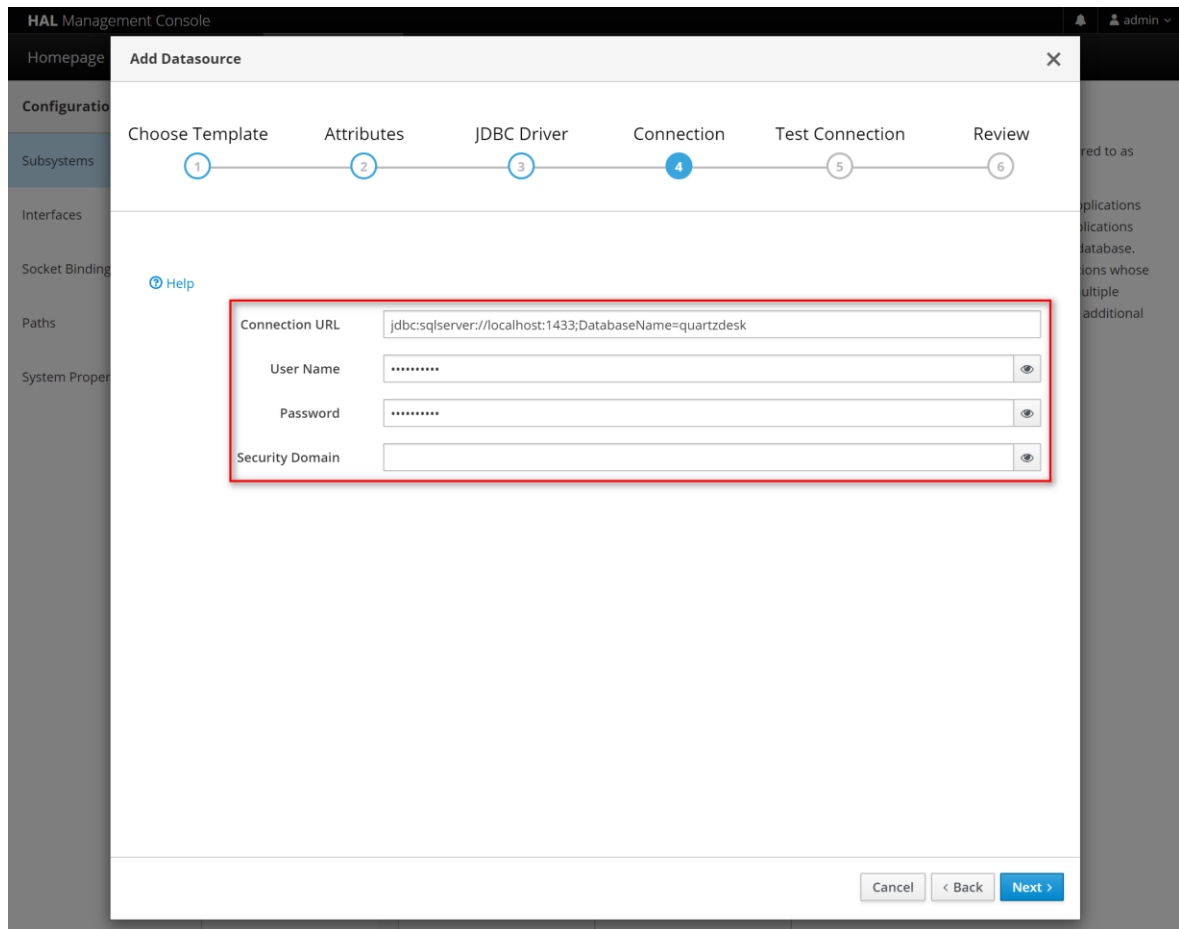
Connection URL: jdbc:sqlserver://DB\_HOST:DB\_PORT;DatabaseName=DB\_NAME

Username: DB\_USER

Password: DB\_PASSWORD

Security Domain: leave empty

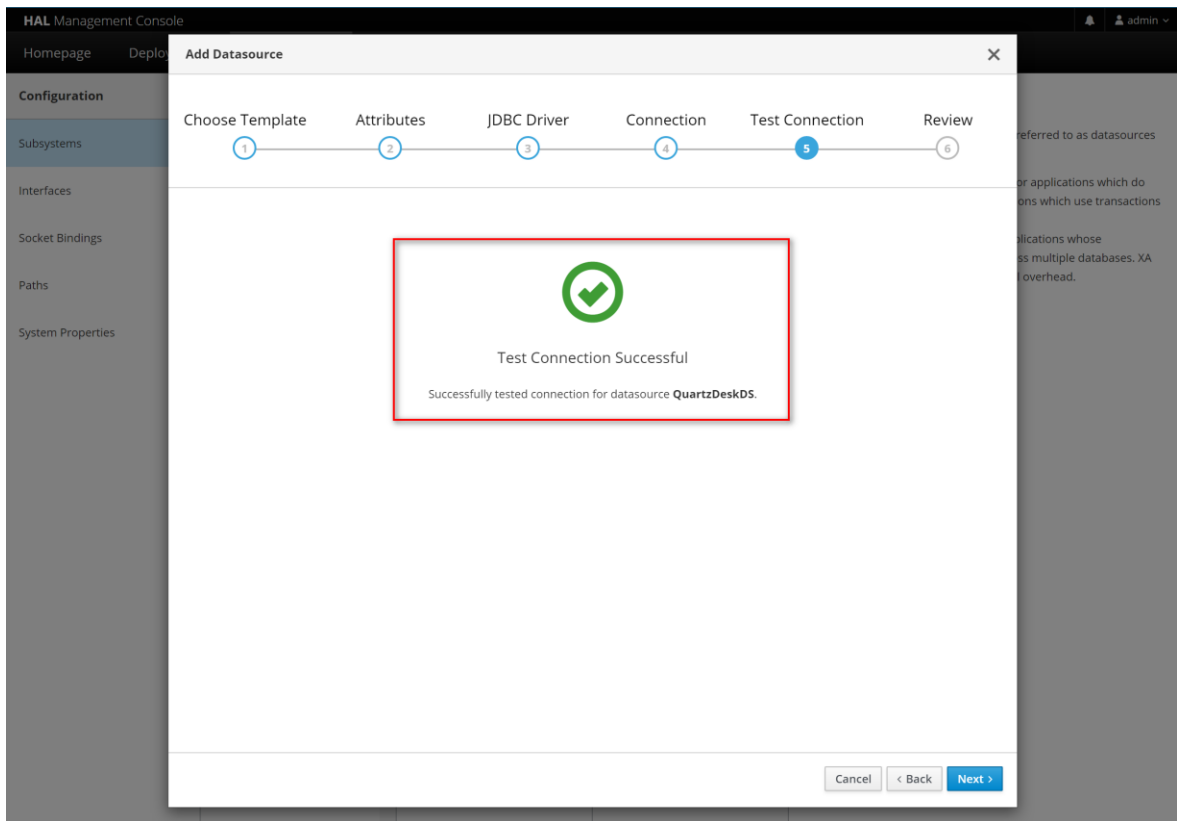




Click Next.

In Step 5, test the datasource by clicking on the Test Connection button.

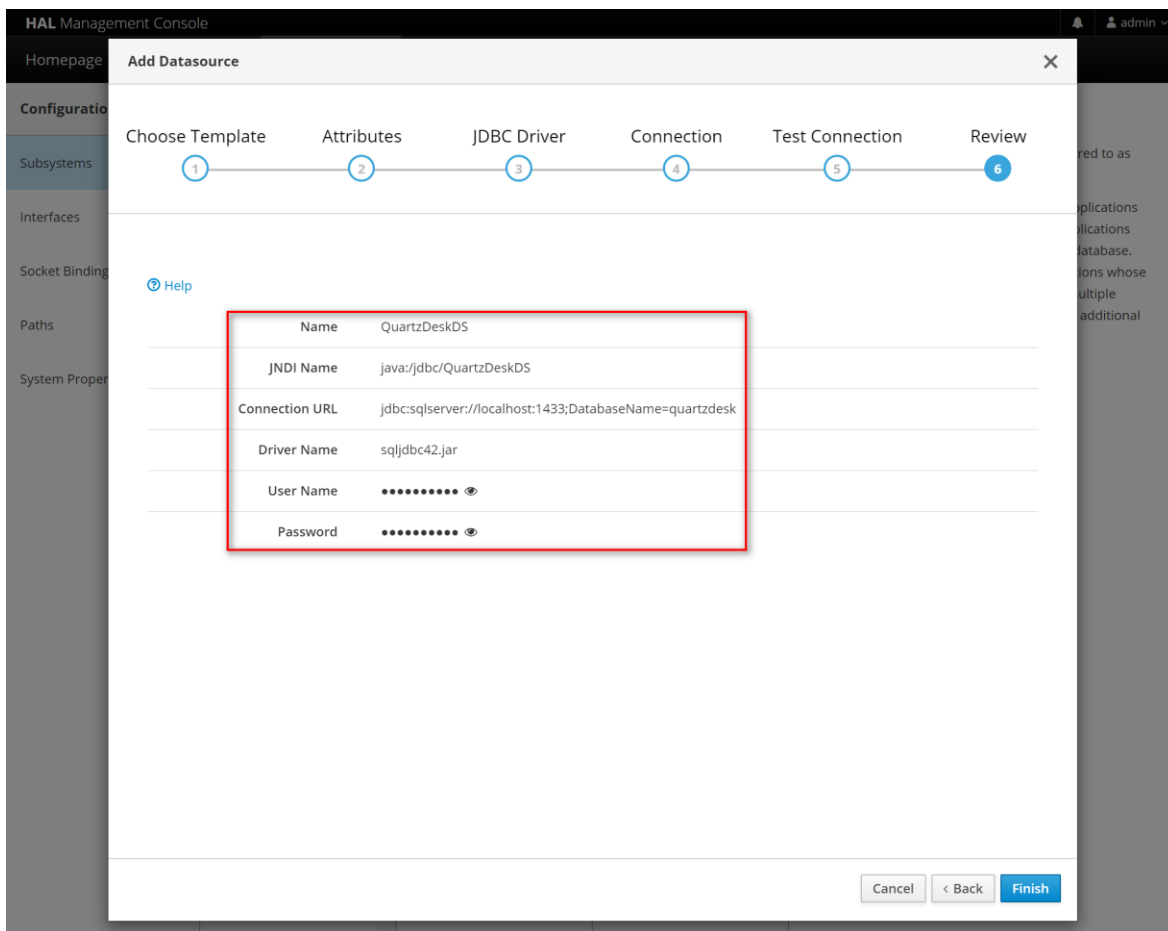




Click Next.

In Step 6, review the datasource parameters.





Click Finish.

Select the registered QuartzDeskDS datasource, click View and modify the datasource configuration under the following tabs:

### Connection

Add the following property:

applicationName=QuartzDesk Web Application

Click Save.

### Pool

Click the Edit button and enter the following values:

Initial Pool Size: 2

Min Pool Size: 2

Max Pool Size: 10

Pool Prefill: On

Click Save.

### Validation

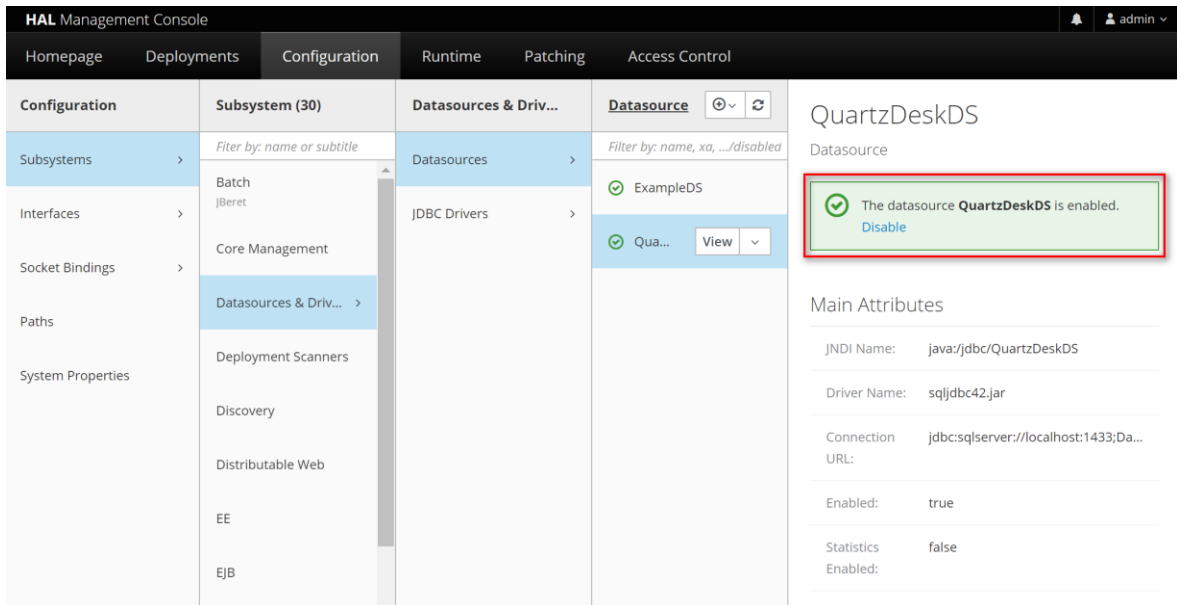
Click the Edit button and enter the following values:



Valid Connection Checker Class Name:  
`org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker`

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.



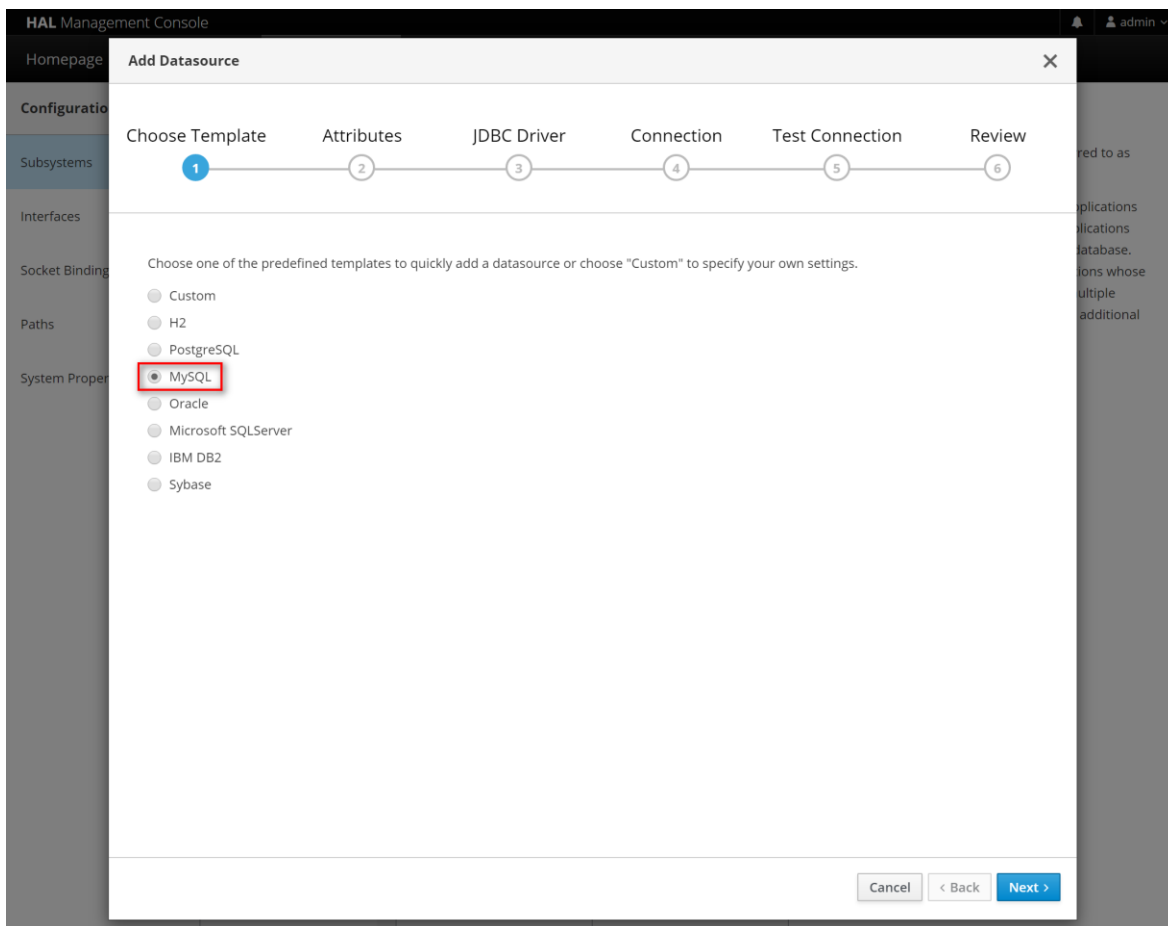
The screenshot shows the HAL Management Console interface. The 'Configuration' tab is active, and the 'Datasources & Drivers' section is selected. The 'QuartzDeskDS' datasource is highlighted, and a red box highlights a green status message: 'The datasource QuartzDeskDS is enabled. Disable'. The 'Main Attributes' section shows the following configuration:

Attribute	Value
JNDI Name:	java:/jdbc/QuartzDeskDS
Driver Name:	sqljdbc42.jar
Connection URL:	jdbc:sqlserver://localhost:1433;Da...
Enabled:	true
Statistics Enabled:	false

#### 4.3.4 MySQL

In the Create Datasource dialog, select the MySQL option.





Click Next.

In Step 2, enter the following datasource attributes:

Name: QuartzDeskDS

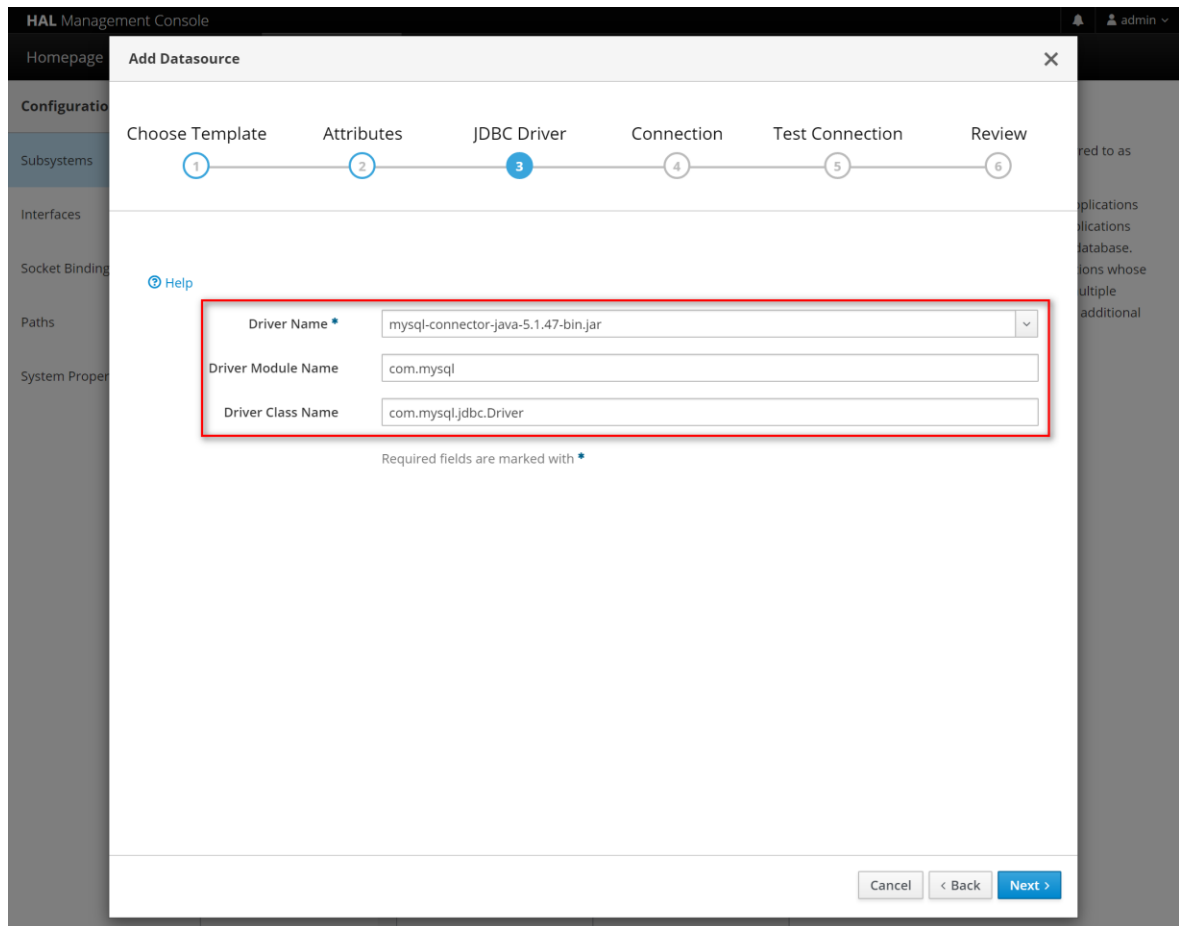
JNDI Name: java:/jdbc/QuartzDeskDS

In Step 3, select the installed MySQL JDBC driver JAR in the Driver Name field and enter the following values:

Driver Module Name: org.mysql

Driver Class Name: org.mysql.jdbc.Driver





Click Next.

In Step 4, enter the following values:

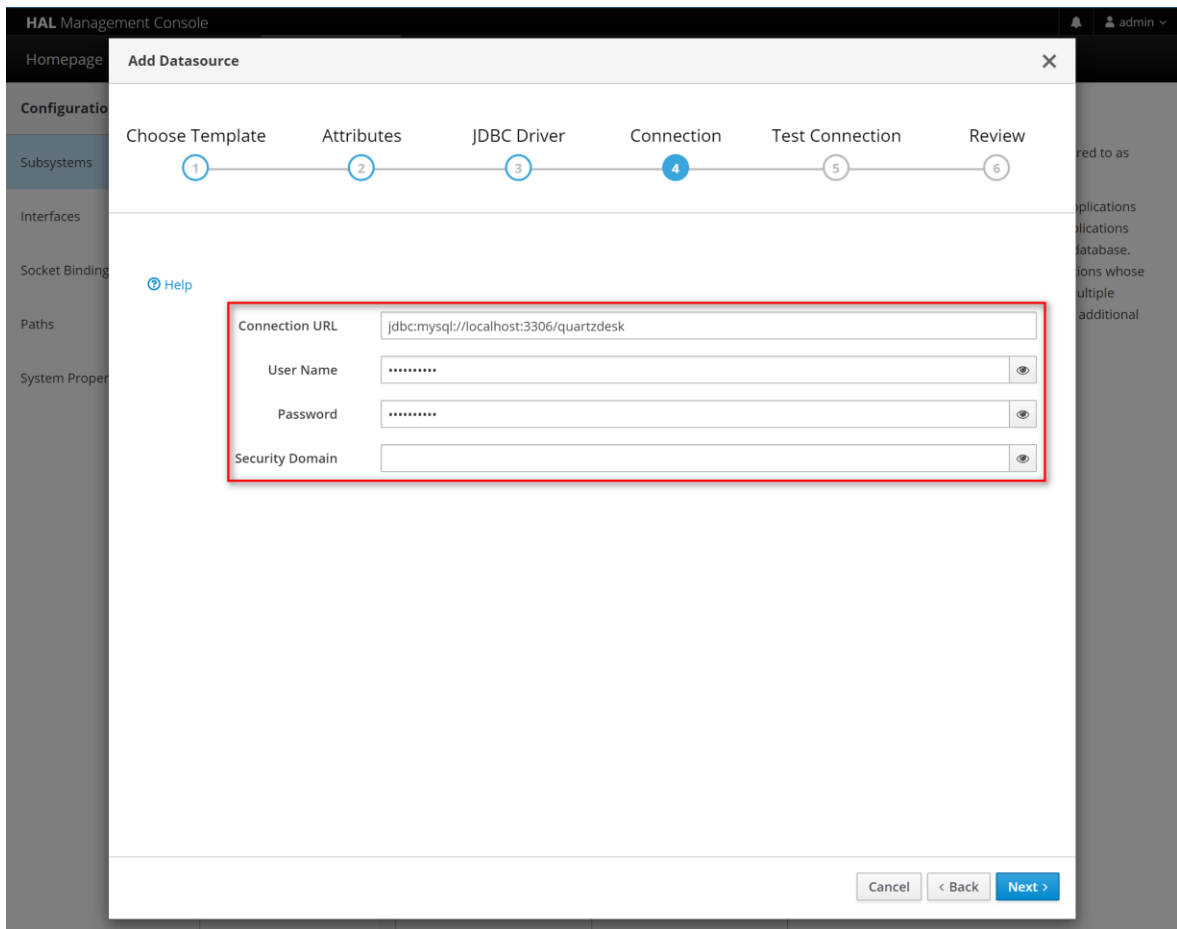
Connection URL: jdbc:mysql://DB\_HOST:DB\_PORT/DB\_NAME

Username: DB\_USER

Password: DB\_PASSWORD

Security Domain: leave empty

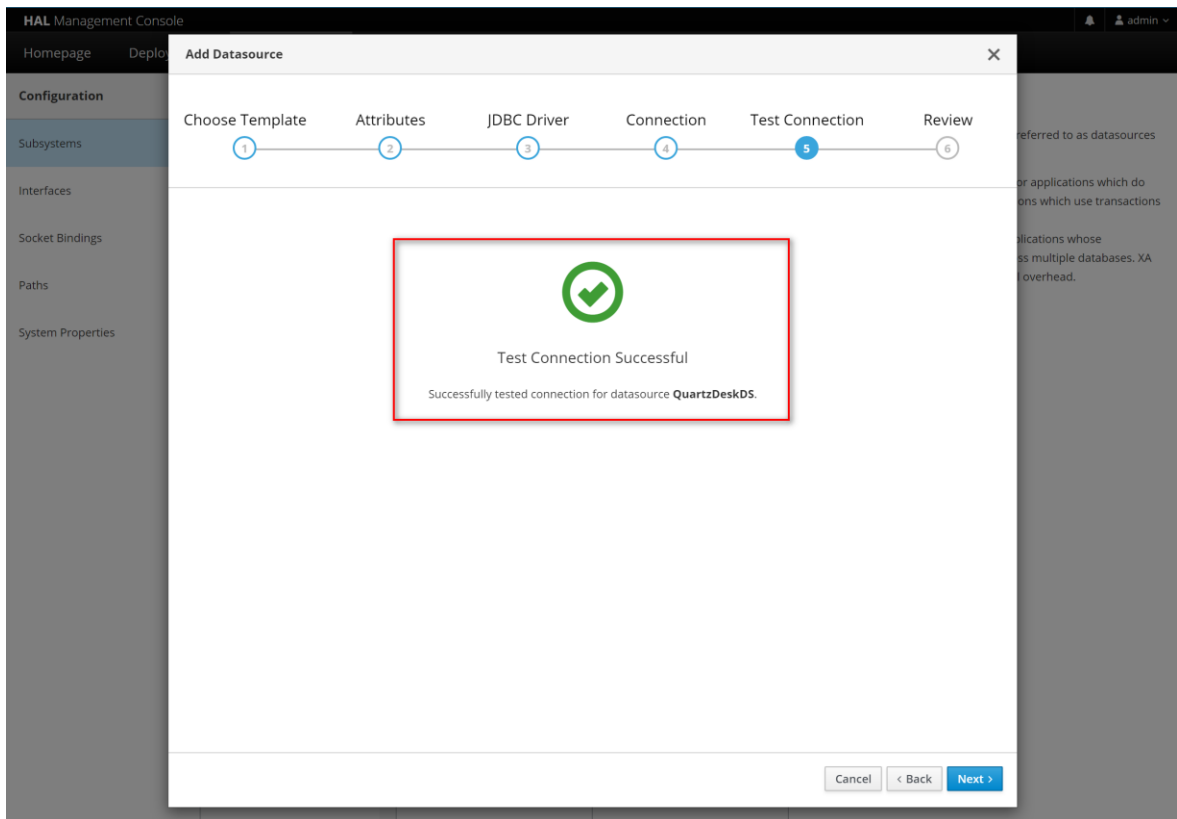




Click Next.

In Step 5, test the datasource by clicking on the Test Connection button.

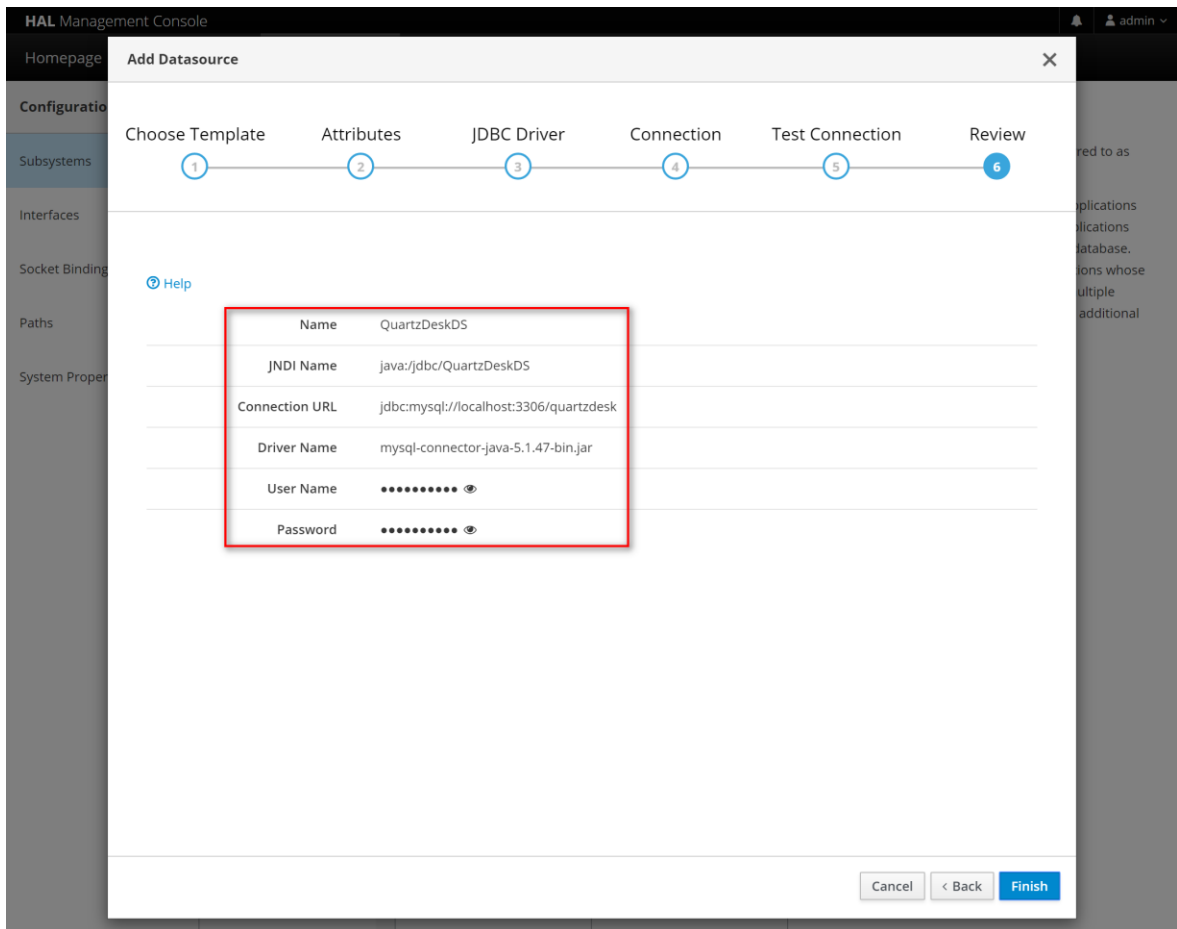




Click Next.

In Step 6, review the datasource parameters.





Click Finish.

Select the registered QuartzDeskDS datasource, click View and modify the datasource configuration under the following tabs:

### Pool

Click the Edit button and enter the following values:

Initial Pool Size: 2

Min Pool Size: 2

Max Pool Size: 10

Pool Prefill: On

Click Save.

### Validation

Click the Edit button and enter the following values:

Valid Connection Checker Class Name:

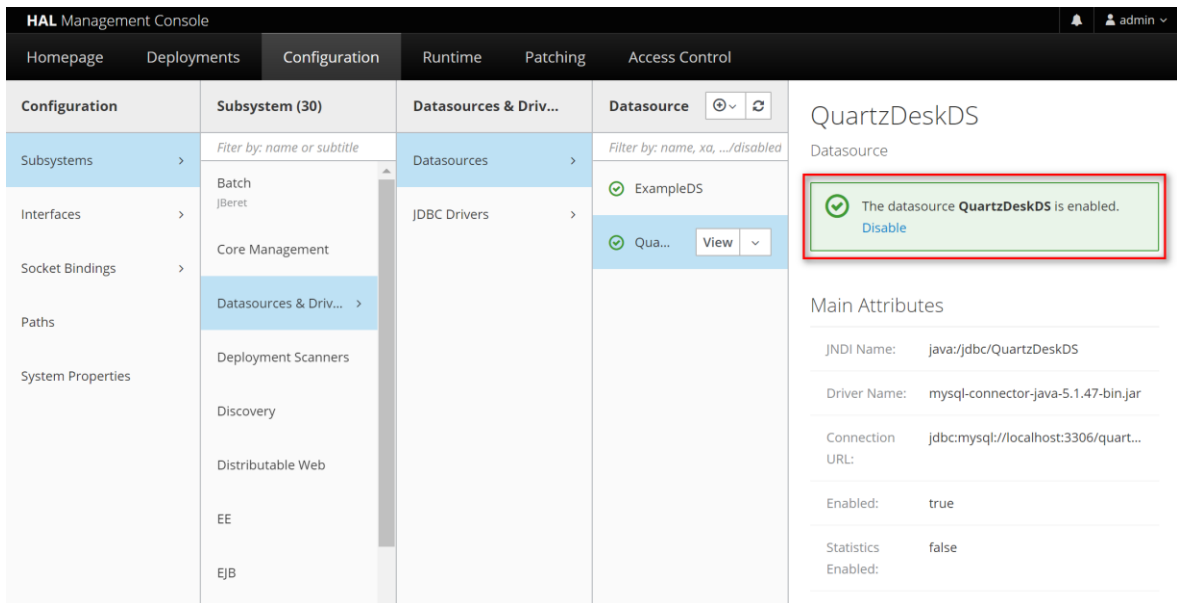
`org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker`

Exception Sorter Class Name:

`org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter`

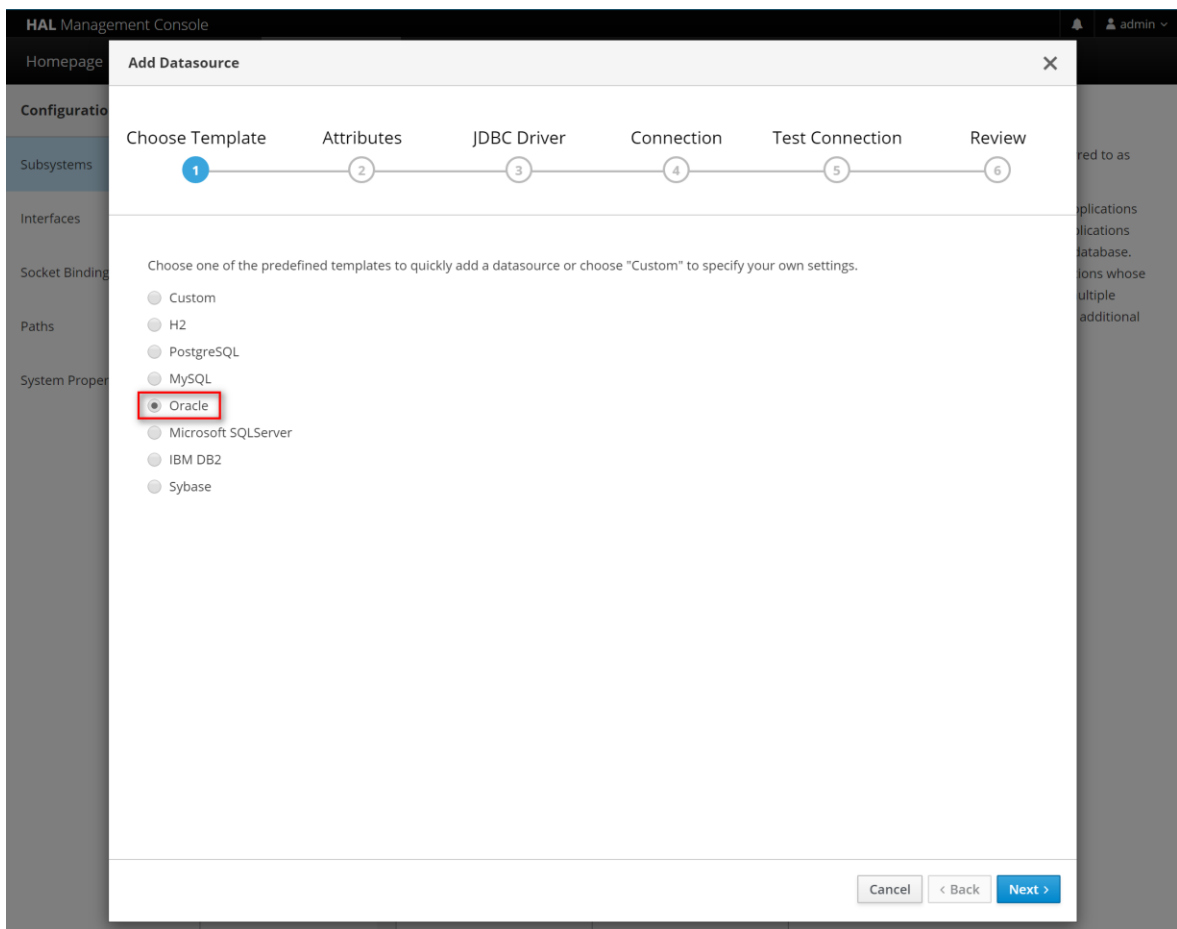
Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.



### 4.3.5 Oracle

In the Create Datasource dialog, select the Oracle option.



Click Next.

In Step 2, enter the following datasource attributes:

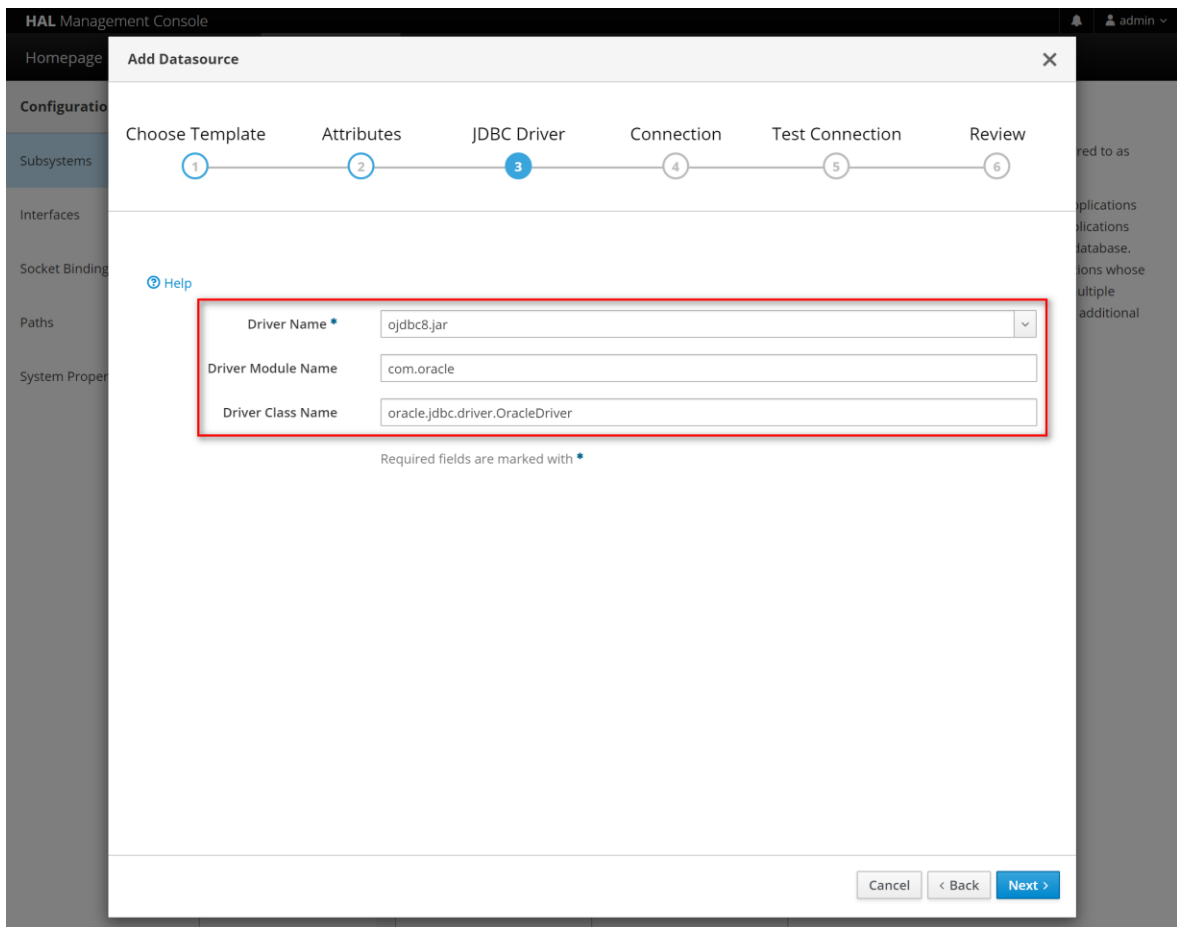
Name: QuartzDeskDS

JNDI Name: java:/jdbc/QuartzDeskDS

In Step 3, select the installed Oracle JDBC driver JAR in the Driver Name field and enter the following values:

Driver Module Name: com.oracle

Driver Class Name: oracle.jdbc.driver.OracleDriver



Click Next.

In Step 4, enter the following values:

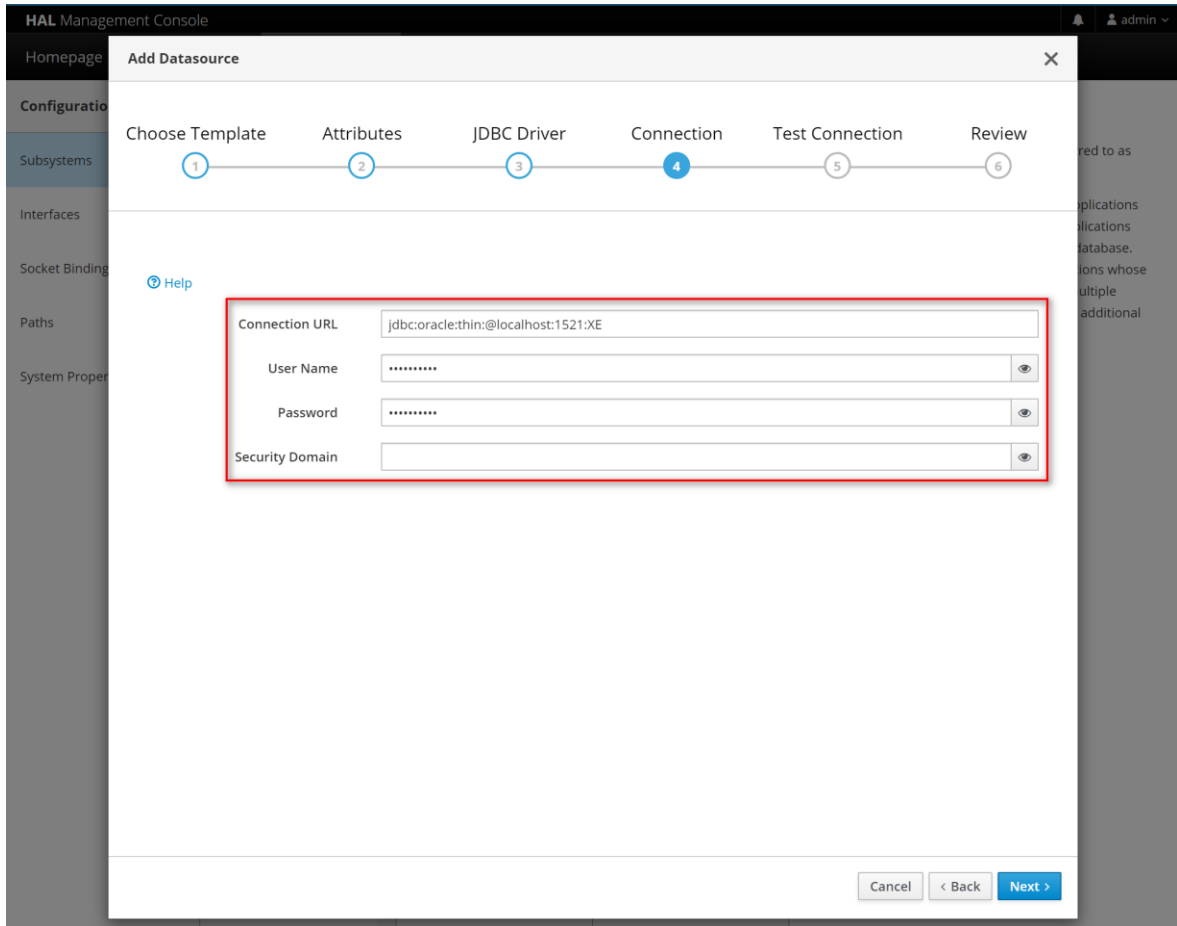
Connection URL: jdbc:oracle:thin:@DB\_HOST:DB\_PORT:ORACLE\_SERVICE\_NAME

Username: DB\_USER

Password: DB\_PASSWORD

Security Domain: leave empty





Click Next.

Select the registered QuartzDeskDS datasource and modify the datasource configuration under the following tabs:

### Pool

Click the Edit button and enter the following values:

Min Pool Size: 2  
Initial Pool Size: 2  
Max Pool Size: 10  
Pool Prefill: On

Click Save.

### Validation

Click the Edit button and enter the following values:

Valid Connection Checker Class Name:

`org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker`

Stale Connection Checker Class Name:

`org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker`

Exception Sorter Class Name:

`org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter`

Click Save.

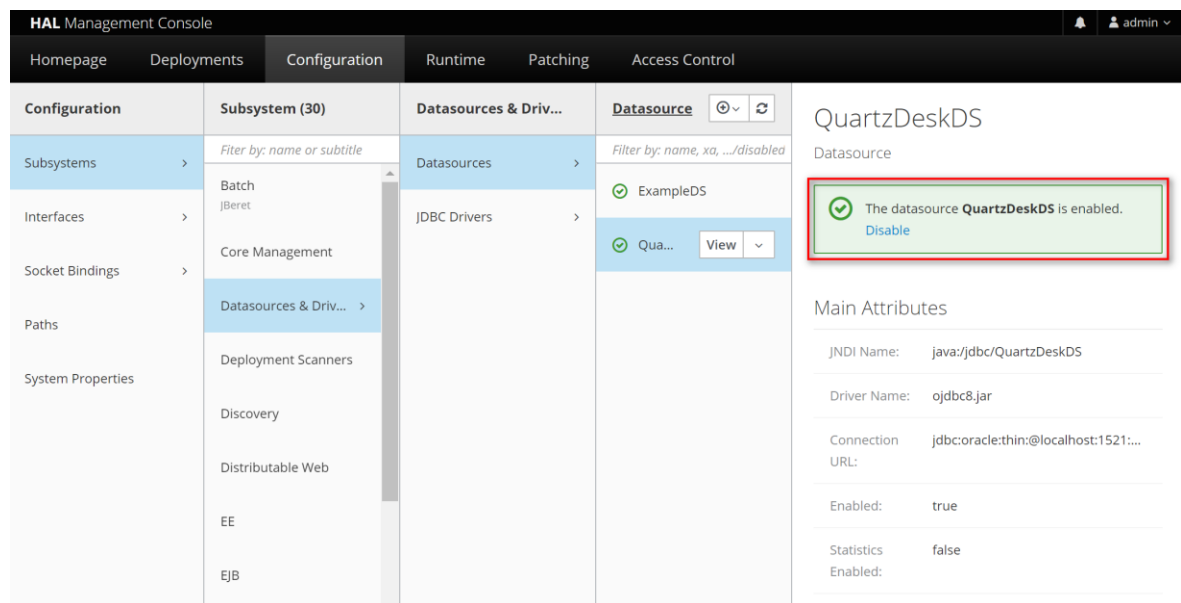
### Statements

Click the Edit button and enter the following values:

Statement Cache Size: 100

Click Save.

Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.



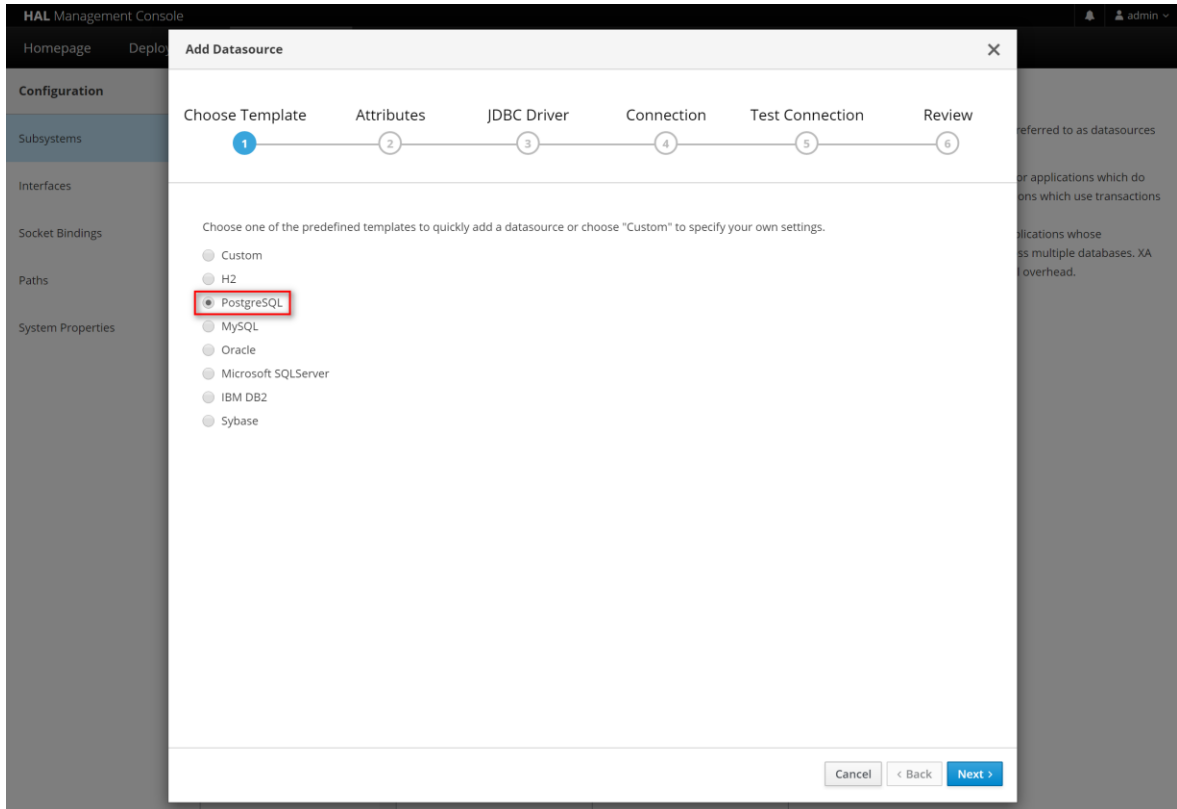
The screenshot shows the HAL Management Console interface. The 'Configuration' tab is active, and the 'Databases & Drivers' section is expanded. The 'QuartzDeskDS' datasource is selected, and its configuration page is displayed. A green checkmark icon and the text 'The datasource QuartzDeskDS is enabled.' are highlighted with a red box. Below this, the 'Main Attributes' section shows the following values:

Attribute	Value
JNDI Name:	java:/jdbc/QuartzDeskDS
Driver Name:	ojdbc8.jar
Connection URL:	jdbc:oracle:thin:@localhost:1521:...
Enabled:	true
Statistics Enabled:	false

### 4.3.6 PostgreSQL

In the Create Datasource dialog, select the PostgreSQL option.





Click Next.

In Step 2, enter the following datasource attributes:

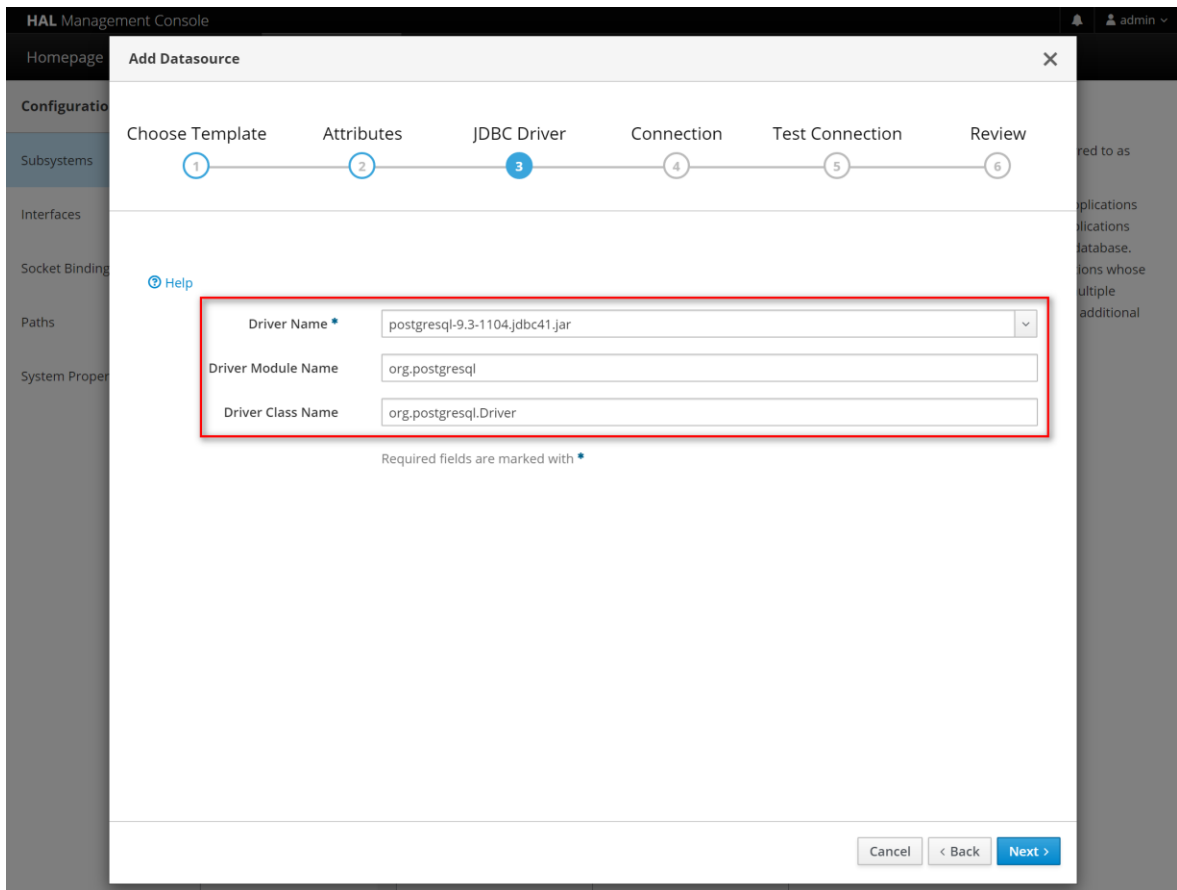
Name: QuartzDeskDS

JNDI Name: java:/jdbc/QuartzDeskDS

In Step 3, select the installed PostgreSQL JDBC driver JAR in the Driver Name field and enter the following values:

Driver Module Name: org.postgresql

Driver Class Name: org.postgresql.Driver



Click Next.

In Step 4, enter the following values:

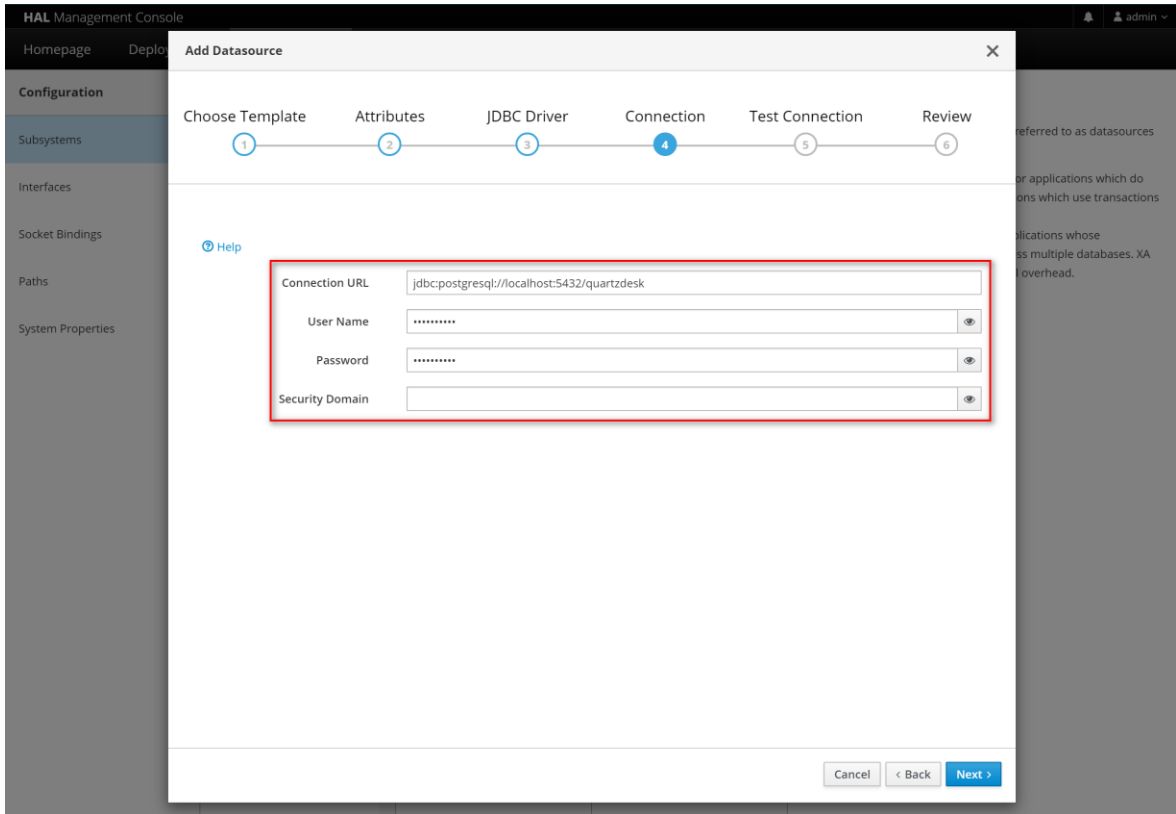
Connection URL: jdbc:postgresql://DB\_HOST:DB\_PORT/DB\_NAME

User Name: DB\_USER

Password: DB\_PASSWORD

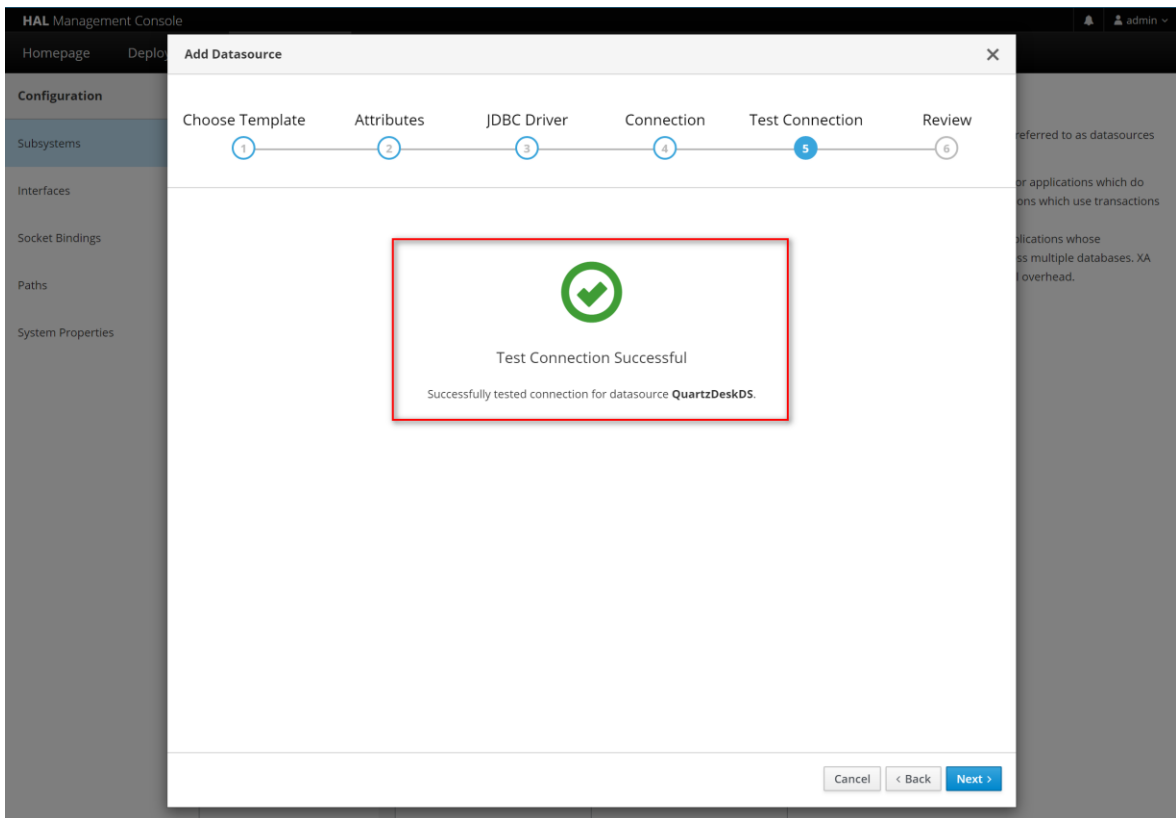
Security Domain: leave empty





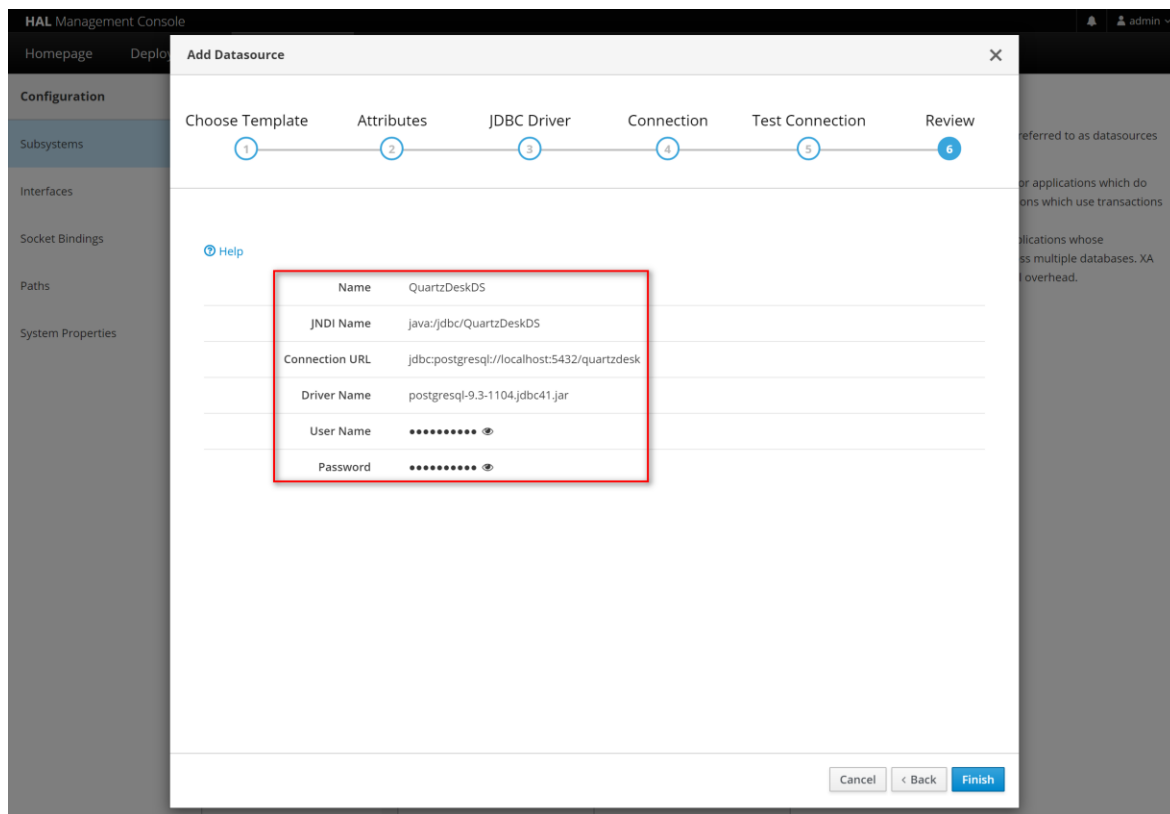
Click Next.

In Step 5, test the datasource by clicking on the Test Connection button.



Click Next.

In Step 6, review the datasource parameters.



Click Finish.

Select the registered QuartzDeskDS datasource, click View and modify the datasource configuration under the following tabs:

### Connection

Add the following property:

applicationName=QuartzDesk Web Application

Click Save.

### Pool

Click the Edit button and enter the following values:

Initial Pool Size: 2

Min Pool Size: 2

Max Pool Size: 10

Pool Prefill: On

Click Save.

### Validation

Click the Edit button and enter the following values:

Valid Connection Checker Class Name:

`org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker`

Exception Sorter Class Name:

`org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter`

Click Save.

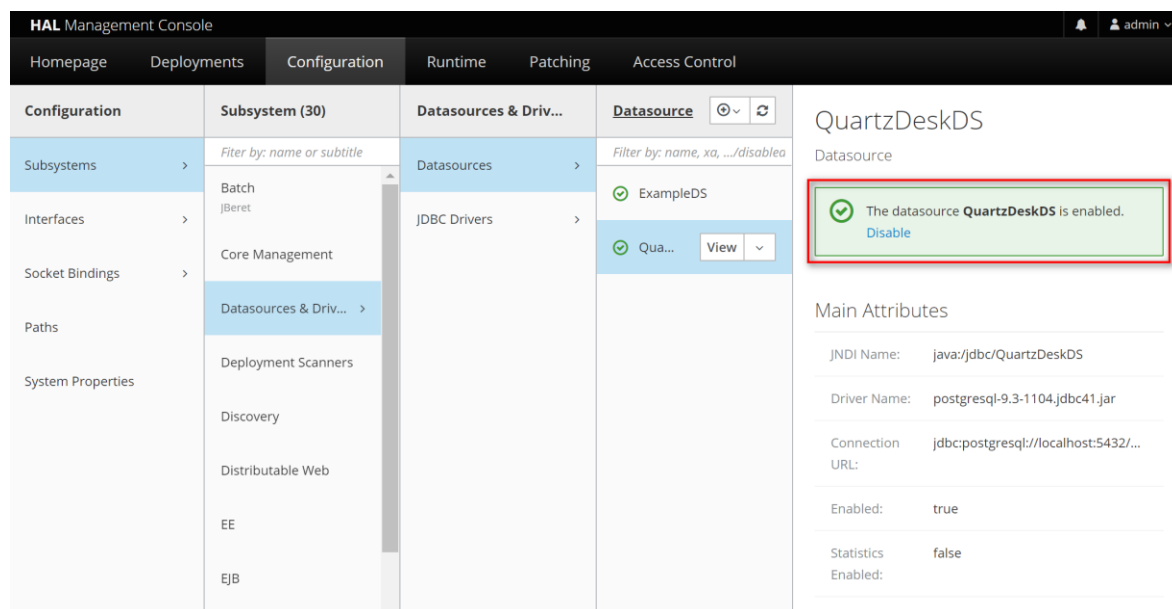
### Statements / Tracking

Click the Edit button and enter the following values:

Prepared Statements Cache Size: 100

Click Save.


Select the QuartzDeskDS datasource in the list of available datasources and make sure it is enabled.




## 4.4 Application Work Directory

Create a QuartzDesk Web Application work directory (`WORK_DIR`) anywhere on the local file system. The directory must be readable and writable by the user the WFAS process runs under.

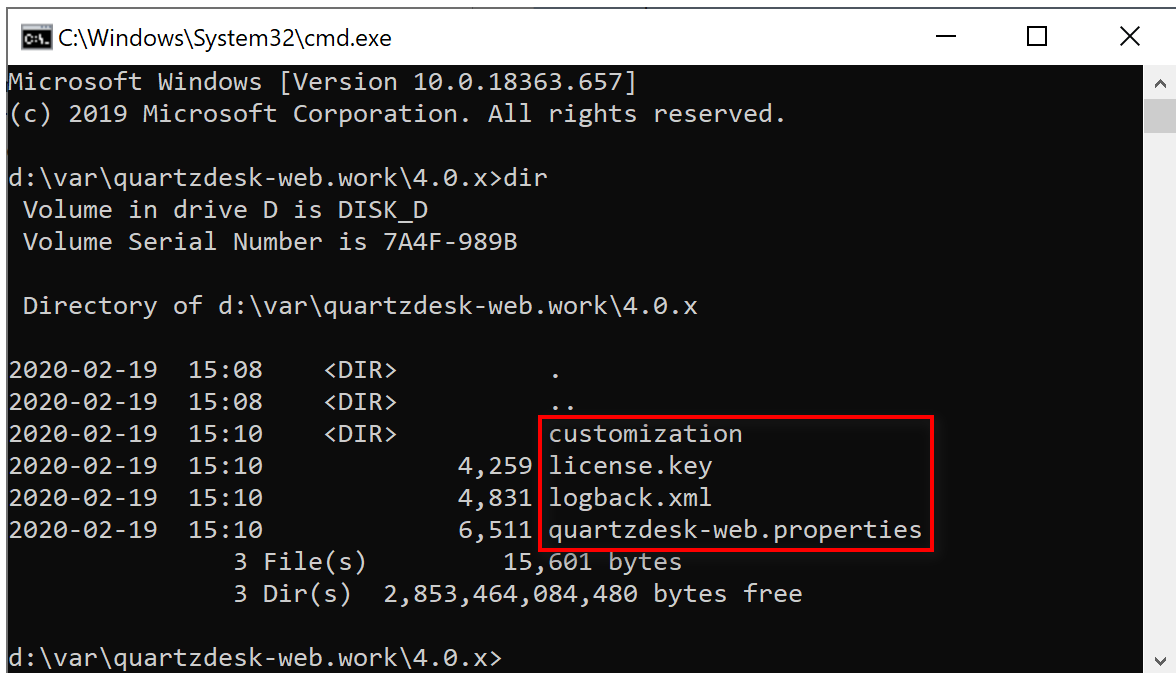
Copy your QuartzDesk license key file (`license.key`) to `WORK_DIR`.

 You can obtain a free 30-day trial license key at [www.quartzdesk.com](http://www.quartzdesk.com) (go to Try / Purchase > Get Trial License Key).

Open the QuartzDesk Web Application archive (`quartzdesk-web-x.y.z.war`) and copy all files from the `extras/work` directory into `WORK_DIR`.

 If you cannot open the WAR file directly, rename it to `*.zip`. Do not forget to rename the file back to `*.war` once you have extracted the required files.

In the following figure you can see an example of a QuartzDesk Web Application work directory correctly set up on a Microsoft Windows machine.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.657]
(c) 2019 Microsoft Corporation. All rights reserved.

d:\var\quartzdesk-web.work\4.0.x>dir
Volume in drive D is DISK_D
Volume Serial Number is 7A4F-989B

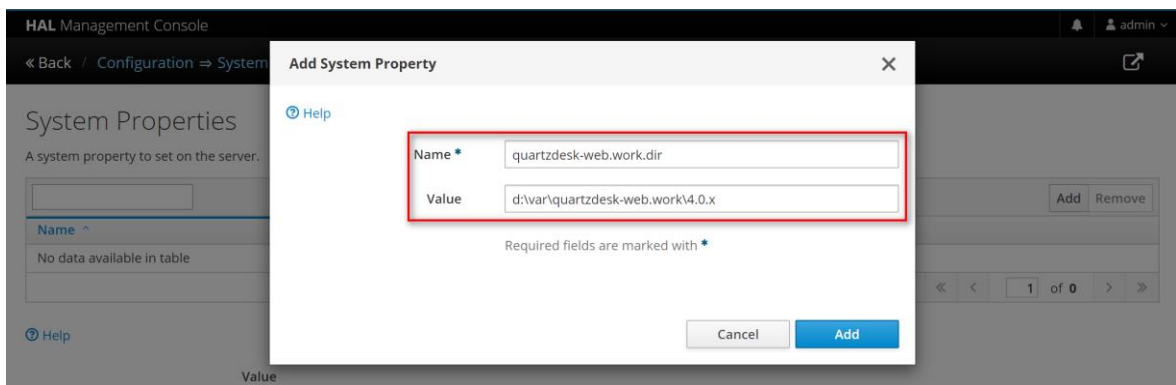
Directory of d:\var\quartzdesk-web.work\4.0.x

2020-02-19  15:08    <DIR>          .
2020-02-19  15:08    <DIR>          ..
2020-02-19  15:10    <DIR>          customization
2020-02-19  15:10             4,259 license.key
2020-02-19  15:10             4,831 logback.xml
2020-02-19  15:10             6,511 quartzdesk-web.properties
                3 File(s)      15,601 bytes
                3 Dir(s)  2,853,464,084,480 bytes free

d:\var\quartzdesk-web.work\4.0.x>
```

In WFAC go to Configuration → System Properties → View. Click the Add button to add a new boot-time system property:

Name: quartzdesk-web.work.dir  
Value: WORK\_DIR



Click Add.

Restart WFAS for the changes to take effect.

## 4.5 Application Configuration

Open the QuartzDesk Web Application configuration file `WORK_DIR/quartzdesk-web.properties`.

Based on the type and version of the database created in 4.1, change the value of the `db.profile` configuration property according to the following table.



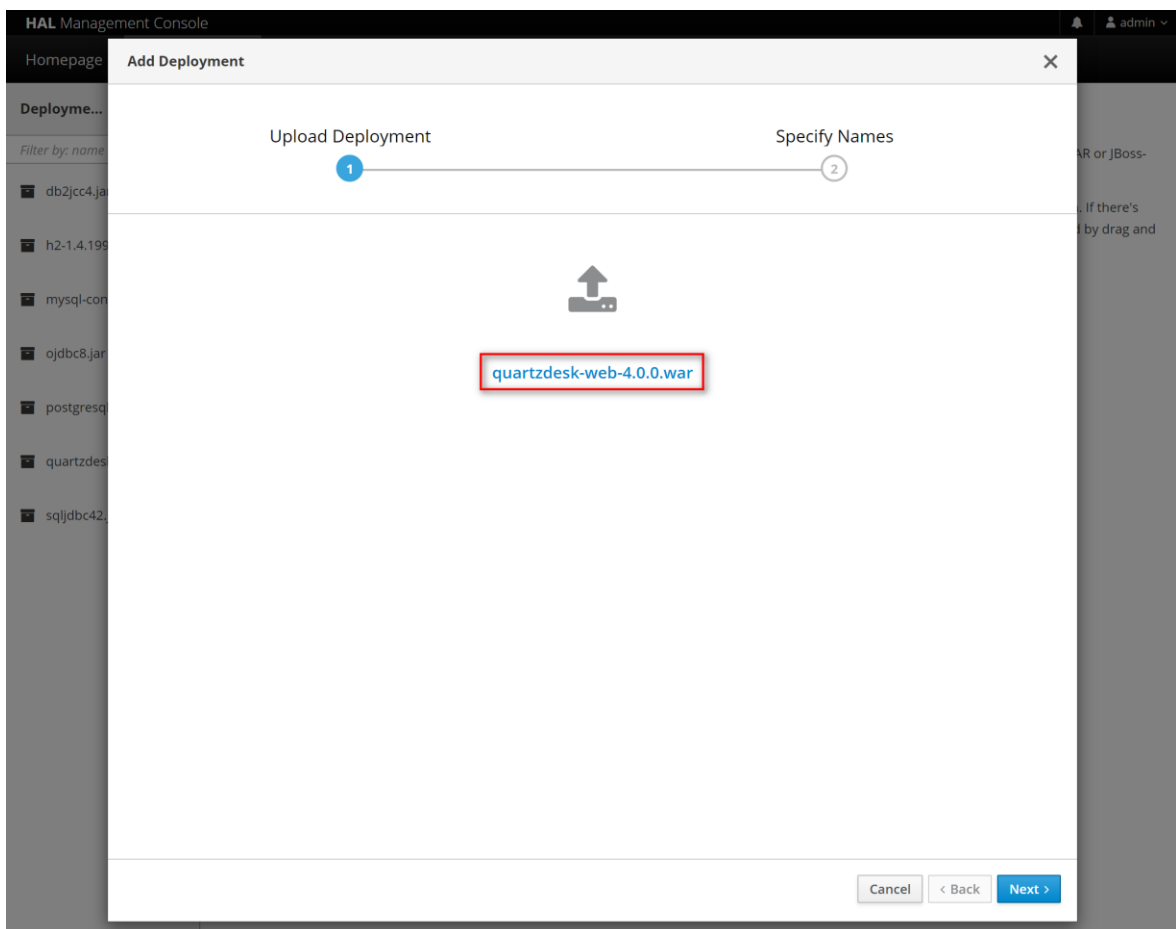
Database	Database Version	db.profile Value
DB2	>= 10.0	db2
H2	>= 1.3.170	h2
Microsoft SQL Server	>= 2008	mssql
MySQL (MyISAM)	>= 5.6	mysql
MySQL (InnoDB)	>= 5.6	mysql_innodb
Oracle	== 8i	oracle8
Oracle	>= 9i	oracle9
PostgreSQL	== 8.1	postgres81
PostgreSQL	>= 8.2	postgres82

Optionally, you can adjust the QuartzDesk Web Application logging parameters by editing the `WORK_DIR/logback.xml` configuration file. The default sample `logback.xml` configuration file makes QuartzDesk Web Application log under the `WORK_DIR/logs` directory that is automatically created when the web application starts. Please refer to the [Logback Manual](#) for Logback configuration details.

## 4.6 Deploy Application

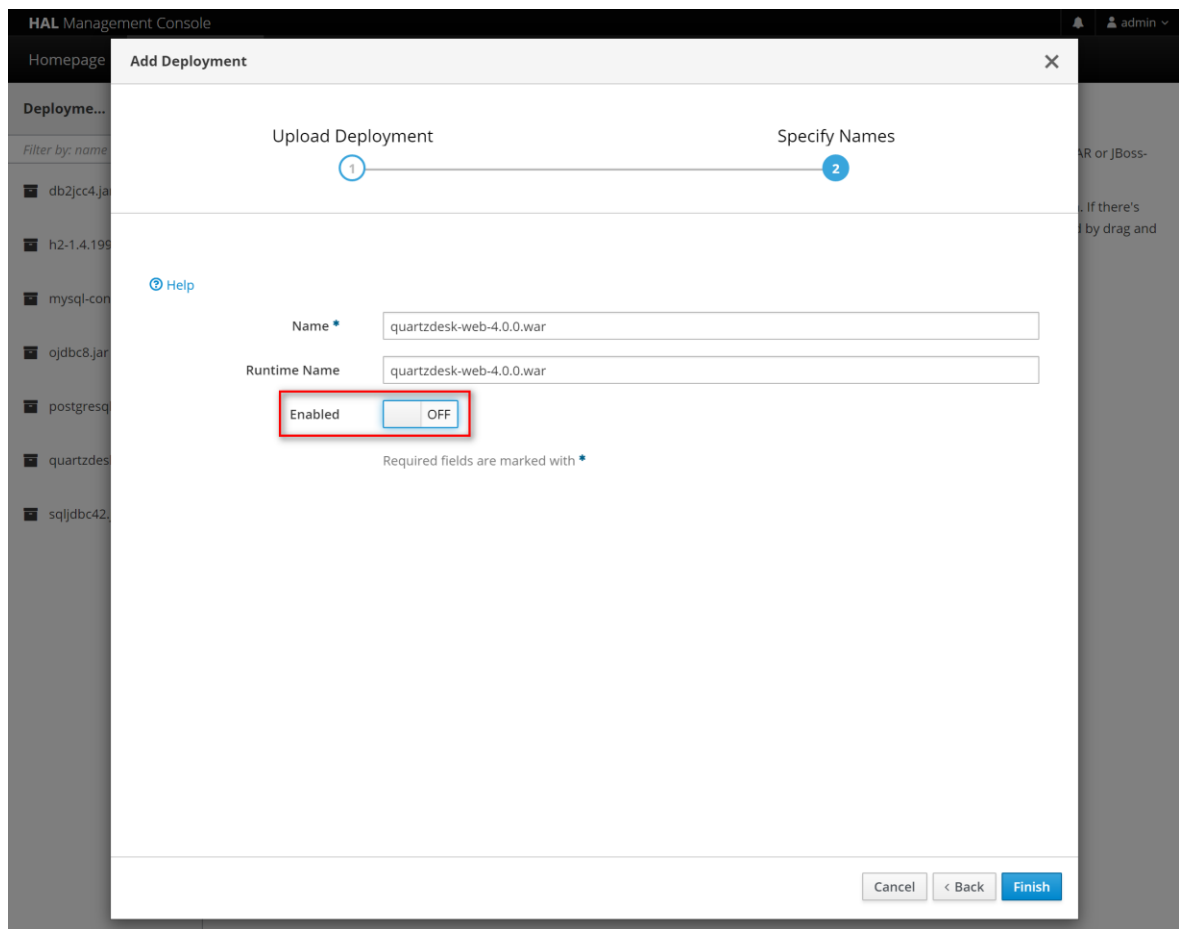
In WFAC go to Deployments, click the Add button and select Upload Deployment.

In Step 1, select the QuartzDesk Web Application WAR file to deploy.



Click Next.

In Step 2, set Enabled to OFF.



HAL Management Console

Homepage Add Deployment X

Deployme...

Filter by: name

db2jcc4.jar

h2-1.4.199

mysql-con

Help

ojdbc8.jar

postgresq

quartzdes

sqljdbc42.

Upload Deployment Specify Names

1 2

Name \* quartzdesk-web-4.0.0.war

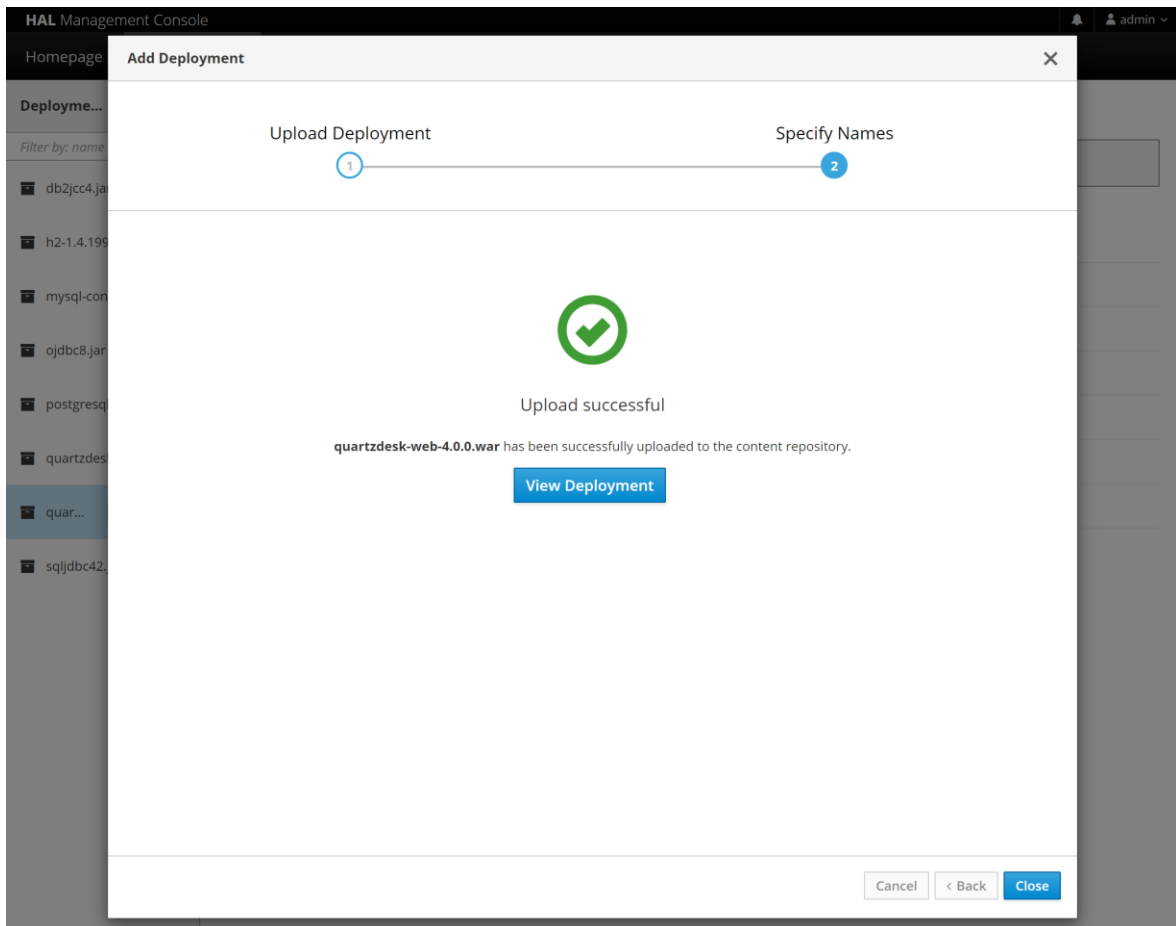
Runtime Name quartzdesk-web-4.0.0.war

Enabled OFF

Required fields are marked with \*

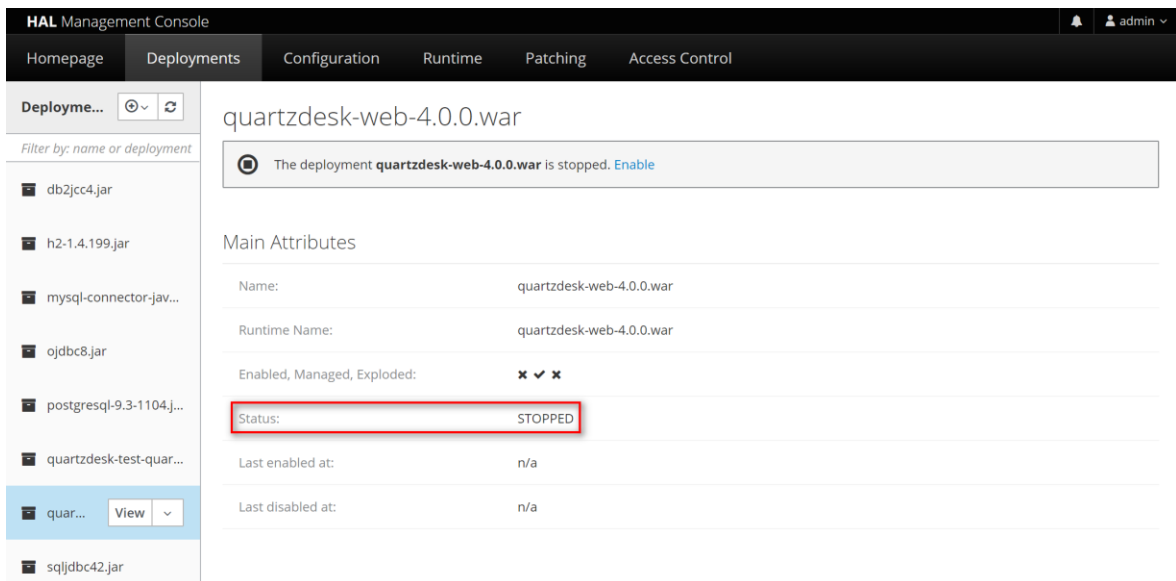
Cancel < Back Finish

Click Finish and wait for the deployment to complete.



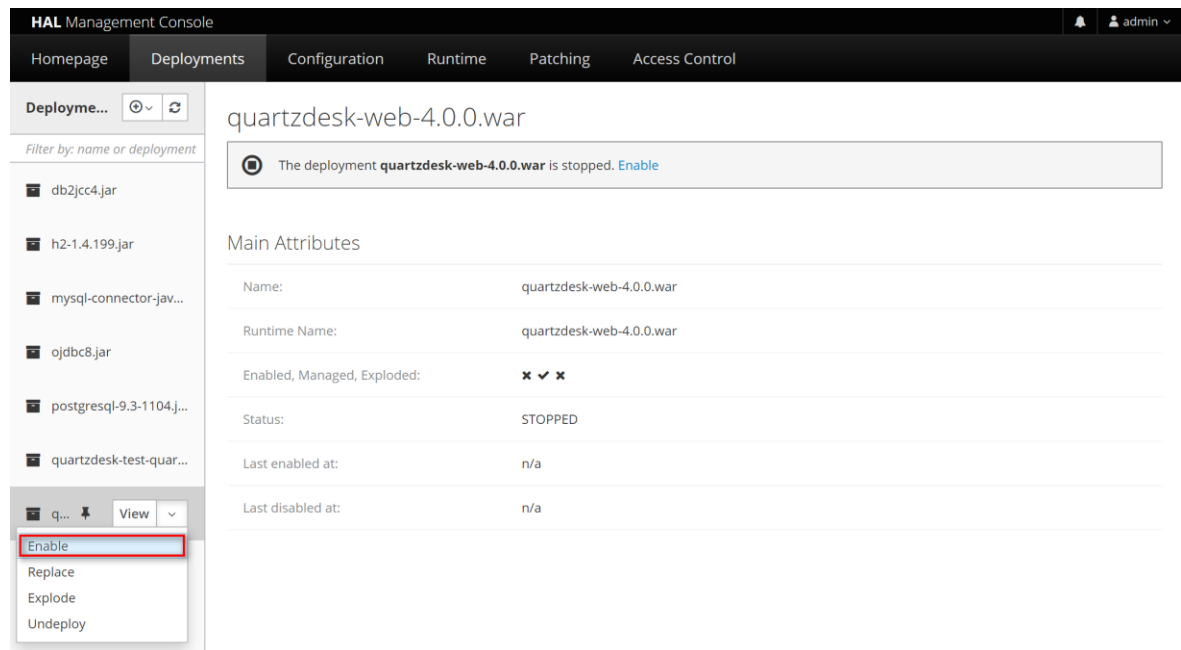
Click Close.

Deployed quartzdesk-web-x.y.z.war file should appear in the list of deployments. Its status should be STOPPED.



## 4.7 Start Application

In WFAC go to Deployments, select QuartzDesk Web Application, open the local menu next to it and select Enable to start the application.



The application should be starting now.

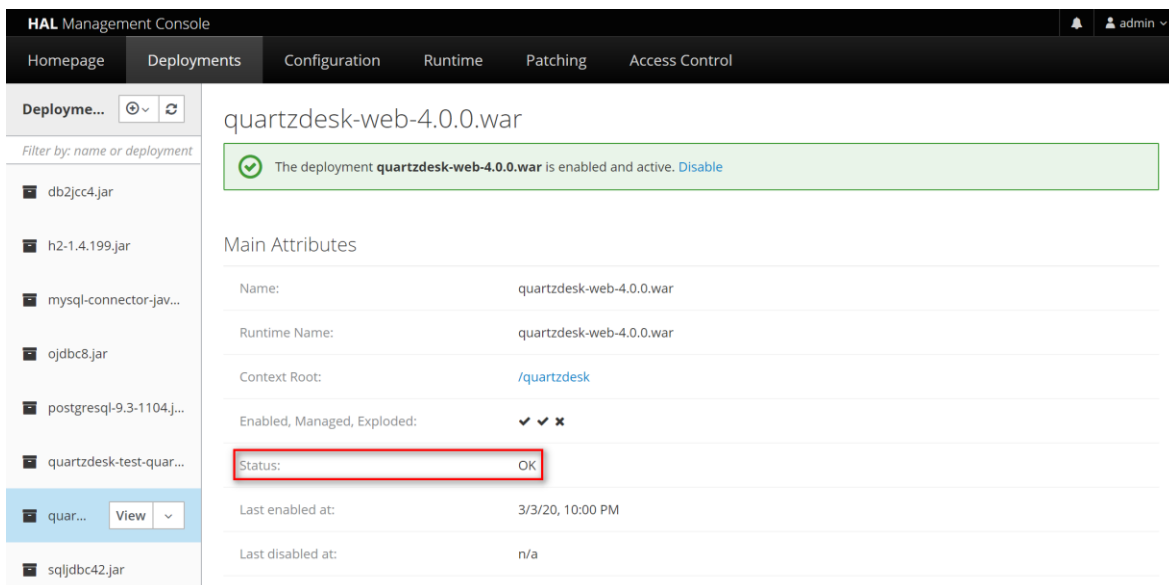
Monitor the WFAS server.log log file under  
WFAS\_INSTALL\_ROOT/WFAS\_CONFIG/logs for errors and wait for the application start  
operation to complete.

You can safely ignore the following warning messages:

```
02:07:20,714 WARN [org.jboss.as.server.deployment] (MSC service  
thread 1-8) WFLYSRV0059: Class Path entry foo.jar in  
/d:/Java/wildfly-18.0.1.Final/bin/content/quartzdesk-web-  
x.y.z.war/WEB-INF/lib/bar.jar does not point to a valid jar for a  
Class-Path reference.
```

Upon successful starting, the QuartzDesk Web Application's status changes to OK.





The screenshot shows the HAL Management Console interface. The 'Deployments' tab is active, displaying a list of deployed applications on the left and details for 'quartzdesk-web-4.0.0.war' on the right. A green notification bar at the top indicates that the deployment is enabled and active. The 'Main Attributes' section shows the following details:

Name:	quartzdesk-web-4.0.0.war
Runtime Name:	quartzdesk-web-4.0.0.war
Context Root:	/quartzdesk
Enabled, Managed, Exploded:	✓ ✓ ✗
Status:	OK
Last enabled at:	3/3/20, 10:00 PM
Last disabled at:	n/a

Check the QuartzDesk Web Application logs (by default located in the `WORK_DIR/logs` directory) for errors.

If there are no errors, point your browser to [http://WFAS\\_HTTP\\_HOST:WFAS\\_HTTP\\_PORT/quartzdesk/](http://WFAS_HTTP_HOST:WFAS_HTTP_PORT/quartzdesk/) and verify that the QuartzDesk Web Application's GUI is accessible.

Check the version number of the deployed QuartzDesk Web Application.



To log in, use the default administrator login credentials:

Username: admin  
Password: admin123

Once logged in, you can go to Settings > Users to manage users with access to the QuartzDesk Web Application's GUI. Users can be assigned different access permissions based on their intended roles.

In Settings > Groups, you can manage groups and assign access permissions to these groups. A group can contain users (members) who inherit access permissions of the group. A user can be a member of any number of groups.

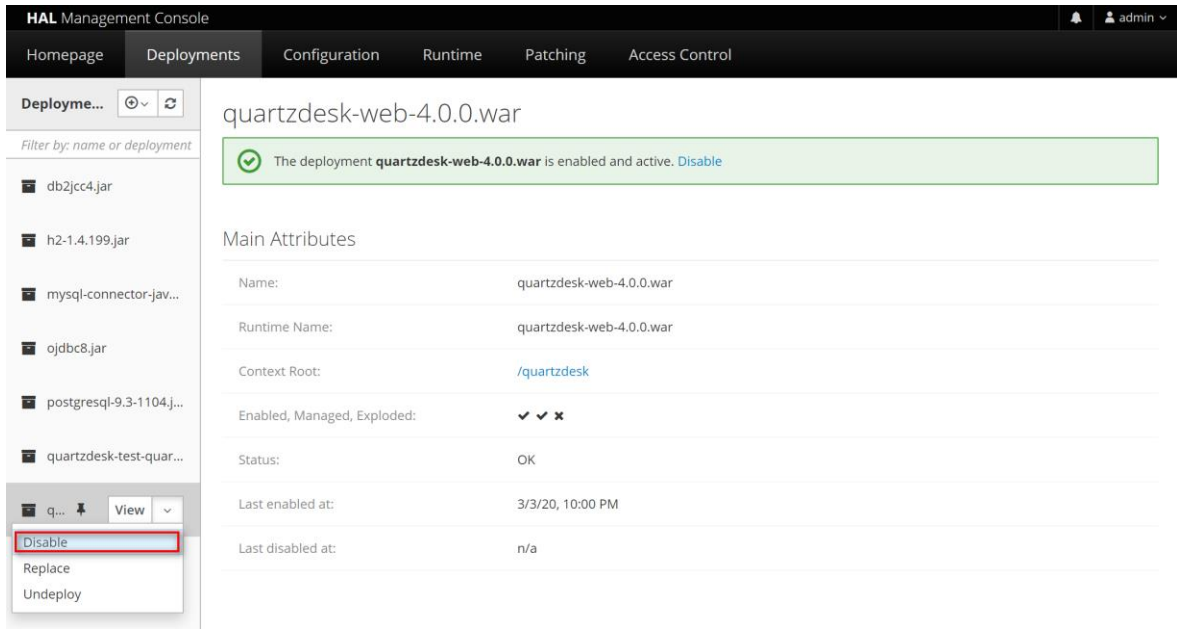
Effective access permissions of a user are permissions associated directly with the user plus access permissions of all groups the user is a member of.



## 5. Upgrading

### 5.1 Stop Existing Application

In WFAC go to Deployments, select QuartzDesk Web Application, open the local menu next to it and select Disable to stop the application.

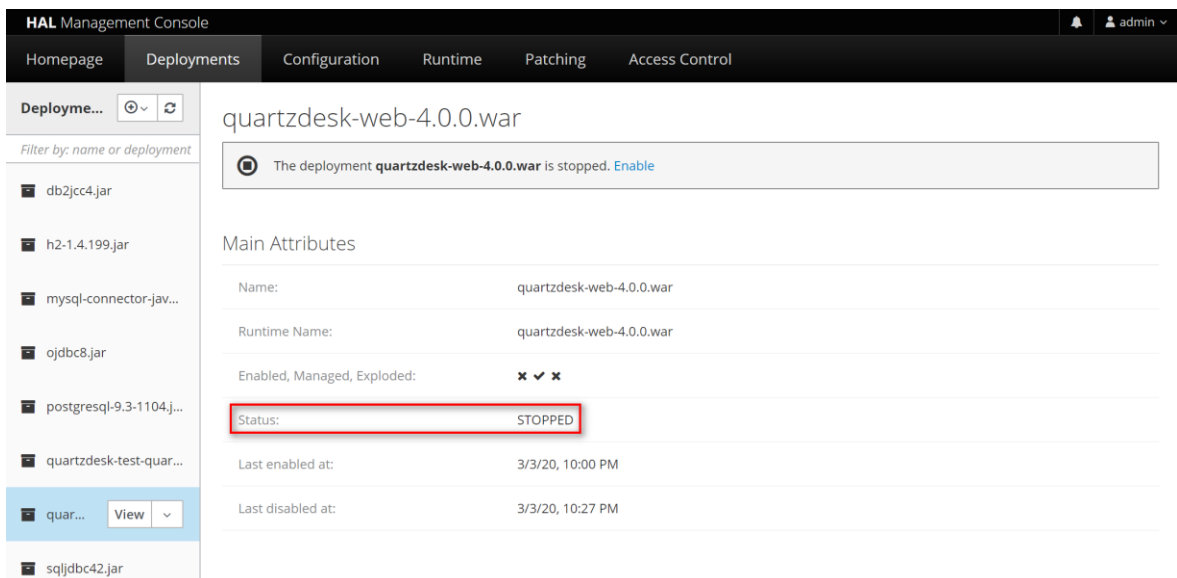


The screenshot shows the HAL Management Console interface. The 'Deployments' tab is selected. A list of deployments is shown on the left, with 'quartzdesk-web-4.0.0.war' selected. The main area displays the details for this deployment, including its name, runtime name, context root, and status. The status is 'OK'. A context menu is open over the deployment, with 'Disable' highlighted. A green notification bar at the top indicates that the deployment is enabled and active, with a 'Disable' link.

The application should be stopping now.

Monitor the WFAS server.log log file under `WFAS_INSTALL_ROOT/WFAS_CONFIG/logs` for errors and wait for the application stop operation to complete.

Upon successful stopping, the QuartzDesk Web Application's status changes to STOPPED.



The screenshot shows the HAL Management Console interface. The 'Deployments' tab is selected. A list of deployments is shown on the left, with 'quartzdesk-web-4.0.0.war' selected. The main area displays the details for this deployment, including its name, runtime name, context root, and status. The status is 'STOPPED'. A red box highlights the 'Status' field. A grey notification bar at the top indicates that the deployment is stopped, with an 'Enable' link.

## 5.2 Backup

Backup the QuartzDesk Web Application database. We recommend performing a **full database backup**.

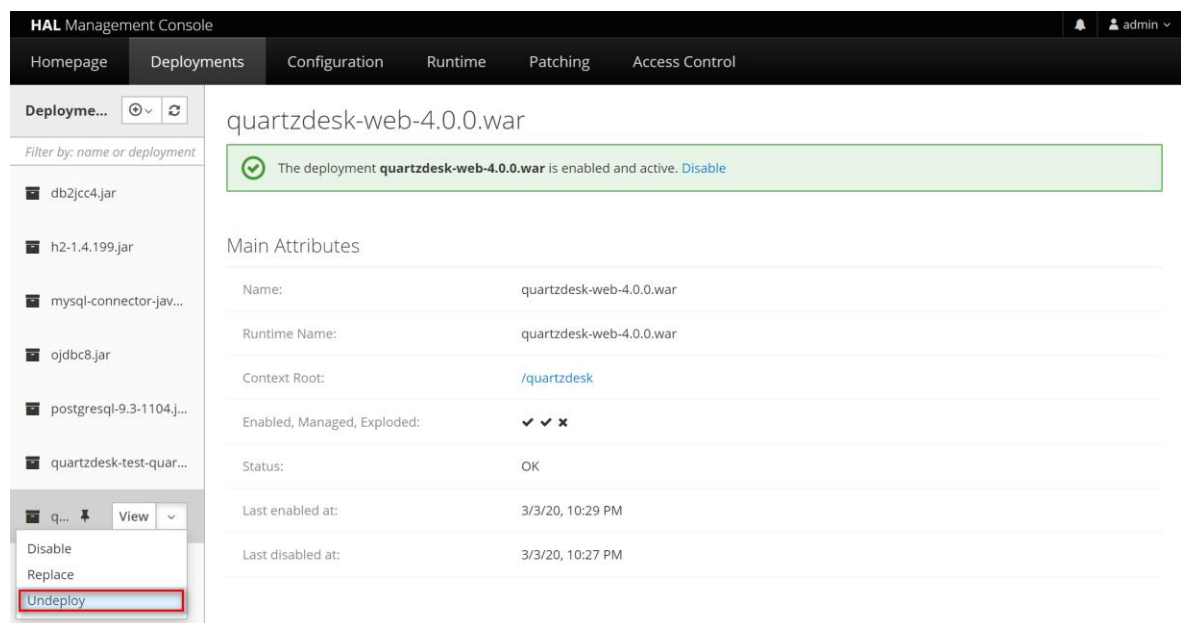
Backup the contents of the QuartzDesk Web Application work directory.

Make sure you have the WAR file of the existing QuartzDesk Web Application.

Store the backup files in a safe place so you can restore the existing QuartzDesk Web Application version if the need arises.

## 5.3 Remove Existing Application

In WFAC go to Deployments, select the deployed QuartzDesk Web Application, open the local menu next to it and select Undeploy to remove the application.



The screenshot shows the HAL Management Console interface. The 'Deployments' tab is active, displaying a list of deployments. The deployment 'quartzdesk-web-4.0.0.war' is selected, and its context menu is open, with 'Undeploy' highlighted. The main attributes for the deployment are shown below the list:

Main Attributes	
Name:	quartzdesk-web-4.0.0.war
Runtime Name:	quartzdesk-web-4.0.0.war
Context Root:	/quartzdesk
Enabled, Managed, Exploded:	✓ ✓ ✗
Status:	OK
Last enabled at:	3/3/20, 10:29 PM
Last disabled at:	3/3/20, 10:27 PM

Upon successful removal, QuartzDesk Web Application disappears from the Deployment list.

## 5.4 Deploy New Application

Deploy the new version of the QuartzDesk Web Application by following the deployment steps outlined in 4.6.



Some WFAS releases contain a bug that prevents the application server from invalidating and recompiling the JSP cache of redeployed web applications. Therefore, we recommend that you to stop the WFAS instance and manually purge the QuartzDesk Web Application JSP cache located at `WFAS_INSTALL_ROOT/WFAS_CONFIG/tmp/quartzdesk-web-x.y.z` Once the cache has been purged, start the WFAS instance.



## 5.5 Start New Application

Start the new version of QuartzDesk Web Application by following the steps outlined in 4.7.



## 6. QuartzDesk 2.x to 3.x Migration Notes

To upgrade QuartzDesk Web Application 2.x to 3.x, follow the upgrade steps outlined in 5.

Before deploying the new QuartzDesk Web Application WAR file (`quartzdesk-web-x.y.z.war`), as outlined in 5.4, make sure you have implemented changes described in this chapter.

### 6.1 Minimum Required Java Version

QuartzDesk Web Application 3.x requires Java 7 or higher. Make sure WFAS is configured to use Java 7 or higher.

### 6.2 Rename Configuration File

The name of the QuartzDesk Web Application 3.x configuration file has changed from `quartzdesk.properties` to `quartzdesk-web.properties`.

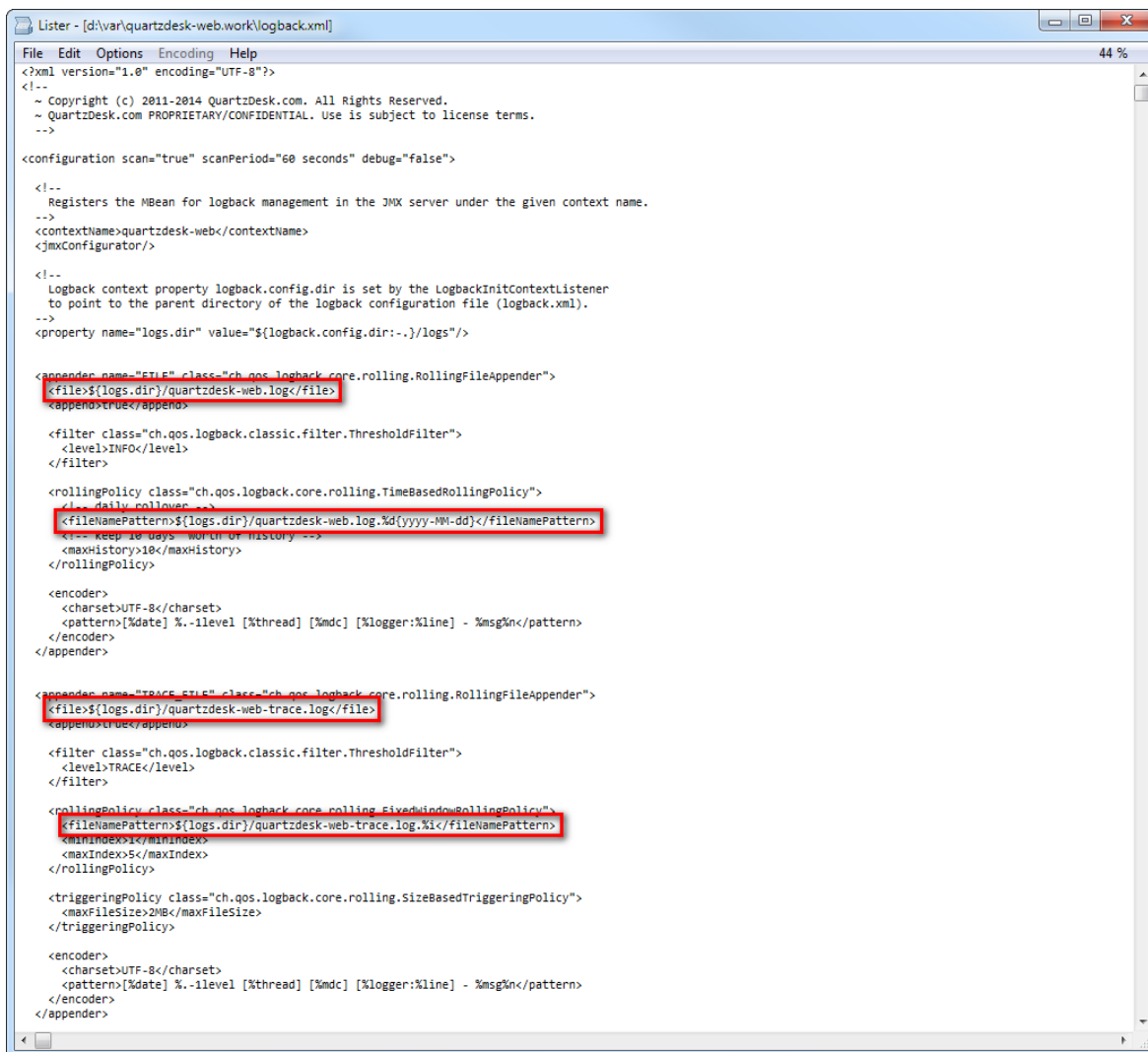
Rename the existing configuration file `quartzdesk.properties` located in the QuartzDesk Web Application work directory.

### 6.3 Rename Log Files

The names of QuartzDesk Web Application 3.x log files have changed.

Original Log File Name (2.x)	New Log File Name (3.x)
<code>quartzdesk.log</code>	<code>quartzdesk-web.log</code>
<code>quartzdesk-trace.log</code>	<code>quartzdesk-web-trace.log</code>

To use these new log file names, edit the QuartzDesk Web Application logging configuration file (`WORK_DIR/logback.xml`) and change the following lines:



```
File Edit Options Encoding Help
<?xml version="1.0" encoding="UTF-8"?>
<!--
~ Copyright (c) 2011-2014 QuartzDesk.com. All Rights Reserved.
~ QuartzDesk.com PROPRIETARY/CONFIDENTIAL. Use is subject to license terms.
-->

<configuration scan="true" scanPeriod="60 seconds" debug="false">

  <!--
  Registers the MBean for logback management in the JMX server under the given context name.
  -->
  <contextName>quartzdesk-web</contextName>
  <JMXConfigurator/>

  <!--
  Logback context property logback.config.dir is set by the LogbackInitContextListener
  to point to the parent directory of the logback configuration file (logback.xml).
  -->
  <property name="logs.dir" value="${logback.config.dir:-./logs}/>

  <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${logs.dir}/quartzdesk-web.log</file>
    <append>true</append>

    <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
      <level>INFO</level>
    </filter>

    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <!-- daily rollover -->
      <fileNamePattern>${logs.dir}/quartzdesk-web.log.%d{yyyy-MM-dd}</fileNamePattern>
      <!-- keep 10 days' worth of history -->
      <maxHistory>10</maxHistory>
    </rollingPolicy>

    <encoder>
      <charset>UTF-8</charset>
      <pattern>[%date] %.-1level [%thread] [%mdc] [%logger:%line] - %msg%n</pattern>
    </encoder>
  </appender>

  <appender name="TRACE FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${logs.dir}/quartzdesk-web-trace.log</file>
    <append>true</append>

    <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
      <level>TRACE</level>
    </filter>

    <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
      <fileNamePattern>${logs.dir}/quartzdesk-web-trace.log.%i</fileNamePattern>
      <minIndex>1</minIndex>
      <maxIndex>5</maxIndex>
    </rollingPolicy>

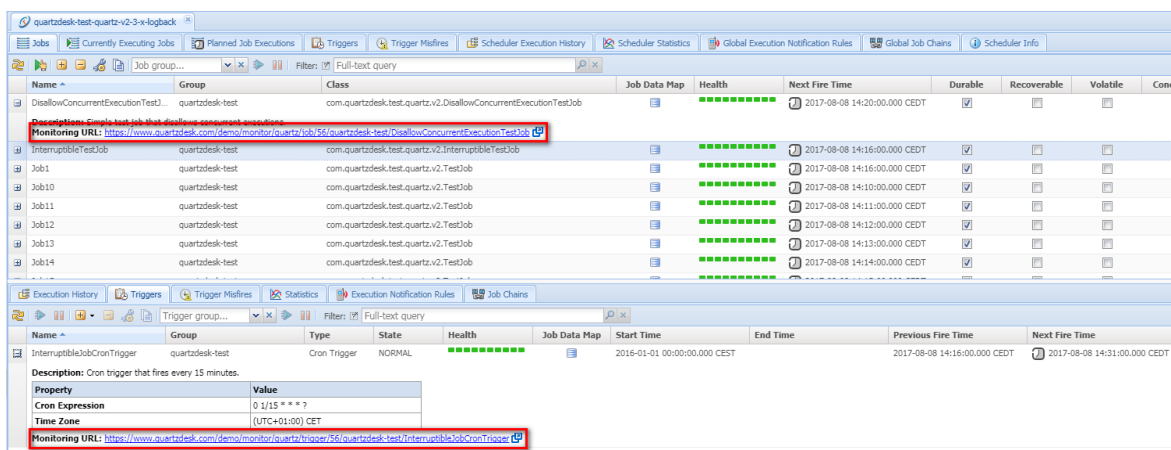
    <triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
      <maxFileSize>2MB</maxFileSize>
    </triggeringPolicy>

    <encoder>
      <charset>UTF-8</charset>
      <pattern>[%date] %.-1level [%thread] [%mdc] [%logger:%line] - %msg%n</pattern>
    </encoder>
  </appender>
```

Alternatively, extract the default `logback.xml` configuration file from the QuartzDesk Web Application 3.x WAR (`quartzdesk-web-x.y.z.war/extras/work/logback.xml`) and copy it to `WORK_DIR`.

## 6.4 Access to Monitoring URLs (REST API)

In QuartzDesk Web Application 2.x, the monitoring REST API URLs could be accessed by users with the QuartzDeskMonitor J2EE security role. In QuartzDesk Web Application 3.x, these monitoring URLs can be accessed by all authenticated users.



We recommend that you create a dedicated user account to access these monitoring URLs. The user account can be created in Settings → Users in the QuartzDesk Web Application’s GUI.



All monitoring URLs in QuartzDesk Web Application 3.x support the HTTP Basic authentication scheme where the user’s authentication credentials are passed in the `Authorization` HTTP header. Please note that the same authentication scheme was used by monitoring URLs in QuartzDesk Web Application 2.x.

## 6.5 Access to JAX-WS Endpoints

In QuartzDesk Web Application 2.x, all JAX-WS web service endpoints could be accessed by users with the QuartzDeskService J2EE security role. In QuartzDesk Web Application 3.x, these web service end points can only be accessed by authenticated users with particular access permissions.

The following table lists all JAX-WS web services and the security permissions that are required to access these web services.

JAX-WS Service	Required Permission
Connection Service	WS_CONNECTION
Security Service	WS_SECURITY
Quartz Service	WS_QUARTZ
Quartz Execution History Service	WS_QUARTZ_EXEC_HISTORY
Quartz Execution Notification Rule Service	WS_QUARTZ_EXEC_NOTIF_RULE
Quartz Job Chain Service	WS_QUARTZ_JOB_CHAIN

We recommend that you create a dedicated user account to access these JAX-WS endpoints. The user account can be created in Settings → Users in the QuartzDesk Web Application’s GUI. Do not forget to assign the user the relevant permission(s).

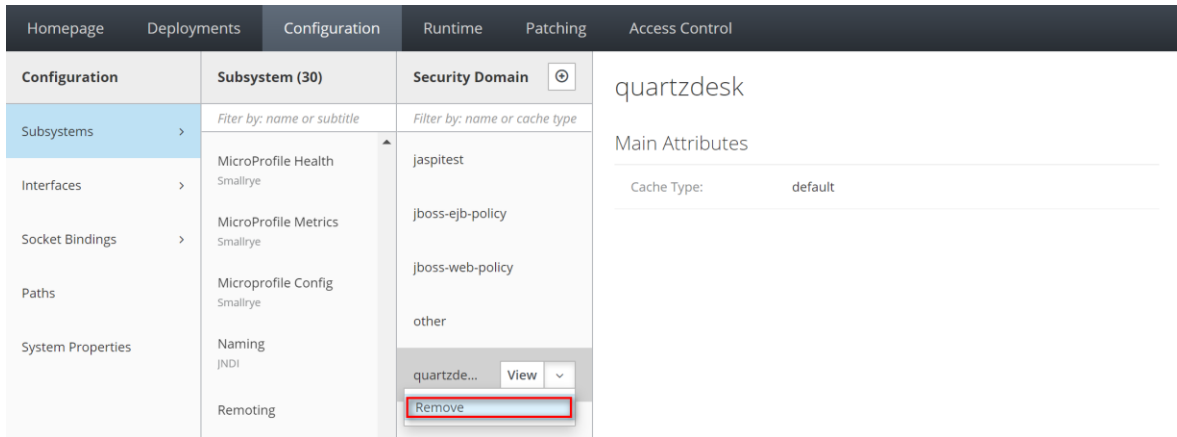


All JAX-WS web service endpoints in QuartzDesk Web Application 3.x support the HTTP Basic authentication scheme where the user’s authentication credentials are passed in the `Authorization` HTTP header. Please note that the same authentication scheme was used by JAX-WS endpoints in QuartzDesk Web Application 2.x.

## 6.6 Remove Unused Security Domain

In WFAC go to Configuration → Subsystems → Security → Security Domain.

Select **quartzdesk** security domain and remove it by clicking on the Remove menu option in the menu next to the security domain name.



Remove the users and roles properties files that were used by the removed security domain:

```
WFAS_INSTALL_ROOT/WFAS_CONFIG/quartzdesk-users.properties
```

```
WFAS_INSTALL_ROOT/WFAS_CONFIG/quartzdesk-roles.properties
```

## 7. QuartzDesk 3.x to 4.x Migration Notes

No configuration changes are required.



## 8. Cluster Deployment Notes

When deploying QuartzDesk Web Application to a WildFly cluster you need to follow the configuration steps described in preceding chapters. In addition to these, there are several extra configuration steps that must be performed for a cluster deployment.

### 8.1 HTTP Session Replication and Affinity

QuartzDesk Web Application makes use of HTTP sessions and to store some short-lived and user-specific data. To achieve high-availability (HA), it is necessary to make the session data available on all application cluster members so that when one cluster member becomes unavailable, the remaining cluster members can take over and handle user requests without the user noticing any service interruption. To make the session data available on all application cluster members, the HTTP session replication process must be enabled on the cluster.



The amount of data stored by QuartzDesk Web Application in an HTTP session is kept at the absolute minimum to reduce the session replication overhead. The total size of data stored in the session does not exceed 1KB.

When configuring session replication, we recommend that you also enable session affinity (sticky-sessions) on the load-balancer so that all user requests are preferably passed to the WildFly instance that handled the first user request that established the session.

Please refer to the WildFly and load-balancer documentation for details on how to configure session replication and session affinity because the actual steps may vary depending on the WildFly cluster topology and configuration.

### 8.2 Shared Work Directory

We recommend that you put the QuartzDesk Web Application work directory, described in 4.4, on a shared drive and make this work directory available to all cluster members. Not only does this make application and configuration upgrading easier, it is actually required by all “Save” (for example, Save Log, Save Chart etc.) actions provided by the QuartzDesk Web Application’s GUI. These actions trigger two subsequent HTTP requests where the first request prepares the data and stores it in the `WORK_DIR/tmp` directory and the second request downloads the data and makes the browser open the Save As dialog.

During a fail-over or if the session affinity is not enabled, it can easily happen that the first request is handled by cluster member A and the second request is handled by cluster member B. If A and B are not configured to use the same `WORK_DIR/tmp` directory, then B will fail to serve the data prepared by A during the preceding request because the data will not be found.

### 8.3 Logging Configuration

If you set up your cluster to use a shared QuartzDesk Web Application work directory, as described in the previous chapter, you will need to edit the QuartzDesk Web Application logging configuration file `WORK_DIR/logback.xml` and decide where QuartzDesk Web Application instances running on individual cluster members should log. There are two options:

- 1) Logging into the same (shared) log files.
- 2) Logging into separate log files.

QuartzDesk Web Application uses two log files – `quartzdesk-web.log` and `quartzdesk-web-trace.log` that are stored in `WORK_DIR/logs` directory. The following chapters discuss these two options.

### 8.3.1 Using Shared Log Files

In order to make individual QuartzDesk Web Application instances log into the same log files, you must enable the prudent mode on both file appenders used in the `WORK_DIR/logback.xml` configuration file:

```
...

<appender name="FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${logs.dir}/quartzdesk-web.log</file>
  <append>true</append>
  <prudent>true</prudent>
  ...
</appender>

<appender name="TRACE_FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${logs.dir}/quartzdesk-web-trace.log</file>
  <append>true</append>
  <prudent>true</prudent>
  ...

<!--
  We must use the TimeBasedRollingPolicy because the
  FixedWindowRollingPolicy is not supported in prudent mode!
-->
<rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
  <!-- daily rollover -->
  <fileNamePattern>${logs.dir}/quartzdesk-web.log.%d{yyyy-MM-
dd}</fileNamePattern>
  <!-- keep 10 days' worth of history -->
  <maxHistory>10</maxHistory>
</rollingPolicy>

<!--
  The SizeBasedTriggeringPolicy removed because it is used only in
  conjunction with the FixedWindowRollingPolicy.
-->

<encoder>
  <charset>UTF-8</charset>
  <pattern>[%date] %.-1level [%thread] [%mdc] [%logger:%line] -
%msg%n</pattern>
</encoder>
</appender>

...
```

For details on the Logback prudent mode, please refer to <http://logback.qos.ch/manual/appenders.html#FileAppender>.





Because prudent mode relies on exclusive file locks to manage concurrent access to the log files and these locks can have negative impact on the QuartzDesk Web Application's performance, we generally discourage using the prudent mode and shared log files.

### 8.3.2 Using Separate Log Files

In order to make individual QuartzDesk Web Application instances log into separate log files, you can use a JVM system property set on all cluster member JVMs. The value of this property must be unique for all cluster members. The property can be referred to from the `WORK_DIR/logback.xml` logging configuration file.

The following examples assume the use of the `cluster.member.instanceId` JVM system property, but any JVM system property name can be used.

There are two common approaches as to where the separate log files produced by individual QuartzDesk Web Application instances are stored:

- 1) Log files created under a common log root directory.

```
...
<appender name="FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${logs.dir}/quartzdesk-web--${cluster.member.instanceId}.log</file>
  <append>true</append>
...
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <!-- daily rollover -->
    <fileNamePattern>${logs.dir}/quartzdesk-web-
-${cluster.member.instanceId}.log.%d{yyyy-MM-dd}</fileNamePattern>
    <!-- keep 10 days' worth of history -->
    <maxHistory>10</maxHistory>
  </rollingPolicy>
...
</appender>

<appender name="TRACE_FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${logs.dir}/quartzdesk-web--${cluster.member.instanceId}-
trace.log</file>
  <append>true</append>
...
  <rollingPolicy
class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
  <fileNamePattern>${logs.dir}/quartzdesk-web-
-${cluster.member.instanceId}-trace.log.%i</fileNamePattern>
  <minIndex>1</minIndex>
  <maxIndex>5</maxIndex>
  </rollingPolicy>
...
</appender>
...
```

2) Log files created in separate (cluster member specific) log root directories.

```

...
<!--
  Logback context property logback.config.dir is set by the
  LogbackInitContextListener to point to the parent directory of the Logback
  configuration file (logback.xml).
-->
<property name="logs.dir" value="${logback.config.dir:-
.}/${cluster.member.instanceId}/logs"/>
...

```

## 8.4 Internal Quartz Scheduler

QuartzDesk Web Application ships with an embedded Quartz scheduler to periodically execute its internal jobs. When deploying QuartzDesk Web Application to a cluster, it is necessary to **assign unique instance IDs to Quartz scheduler instances** running in the clustered QuartzDesk Web Application instances.

For these purposes the QuartzDesk Web Application configuration (`quartzdesk-web.properties` file) provides the `scheduler.org.quartz.scheduler.instanceIdGenerator.class` configuration property. The value of this property must be a fully-qualified class name of a Java class that implements the `org.quartz.spi.InstanceIdGenerator` Quartz API interface. Quartz API provides two out of the box implementations suitable for clustered QuartzDesk Web Application deployments:

Implementation	Description
<code>org.quartz.simpl.HostnameInstanceIdGenerator</code>	<p>This implementation is suitable for QuartzDesk Web Application deployments where individual clustered QuartzDesk Web Application instances run on distinct hosts and each of these hosts is assigned a unique hostname.</p> <p>This is the default implementation used by QuartzDesk Web Application. No configuration changes are necessary to use this instance ID generator.</p>
<code>org.quartz.simpl.SystemPropertyInstanceIdGenerator</code>	<p>This implementation is suitable for QuartzDesk Web Application deployments where some of the clustered QuartzDesk Web Application instances run on the same host.</p> <p>This implementation extracts the Quartz scheduler instance ID from the <code>org.quartz.scheduler.instanceId</code> JVM system property that must be explicitly set.</p> <p>Please refer to the WildFly documentation for details on how to add a new JVM system property.</p>

Please refer to the table above and optionally modify the value of the `scheduler.org.quartz.scheduler.instanceIdGenerator.class` configuration property according to the cluster configuration.

