POLITECNICO
DI MILANO

Installation manual (with Windows Server 2012 R2 OS examples)

MeteoCal

January 17, 2015

Manuel Grillo
Francesco Fermi
Jerry M. Jude

0. Index

Table of Contents of Installation manual v1.0

0. Index 1
1. Introduction 2
2. Getting started 3
   2.1. Make sure JDK 8 is installed 3
   2.2. Download GlassFish 5
   2.3. Install GlassFish 5
   2.4. Start GlassFish 6
   2.5. Install MySQL Community Server 8
   2.6. Run MySQL 14
3. Database configuration 15
   3.1. Create the database schema in MySQL 15
   3.2. Install MySQL Java connector 15
   3.3. Configure GlassFish 16
4. Application deployment 23
   4.1. Deploy the application in GlassFish 23

The opinions expressed in this document are those of the authors, and do not necessarily reflect the opinions of the mebers of the Politecnico di Milano university.

1. Introduction

MeteoCal beta is a web application developed in Java EE 7 for the GlassFish application server and it exploits a MySQL database to store data. If you are using a server operating system, you probably want to whitelist the sites listed in *Appendix A – Secure links for sever Oss*. Please, also remember that the server must be connected to the internet.

2. Getting started

2.1. Make sure JDK 8 is installed

Java EE 7 requires JDK 7 (or above) as it make use of some of the latest SE 7 features. For GlassFish 4.1 and – thus – for MeteoCal beta, we recommend to have JDK 8 Build 20 (or above) installed (full JDK, not just JRE). The Java SE Development Kit 8 can be downloaded here:

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

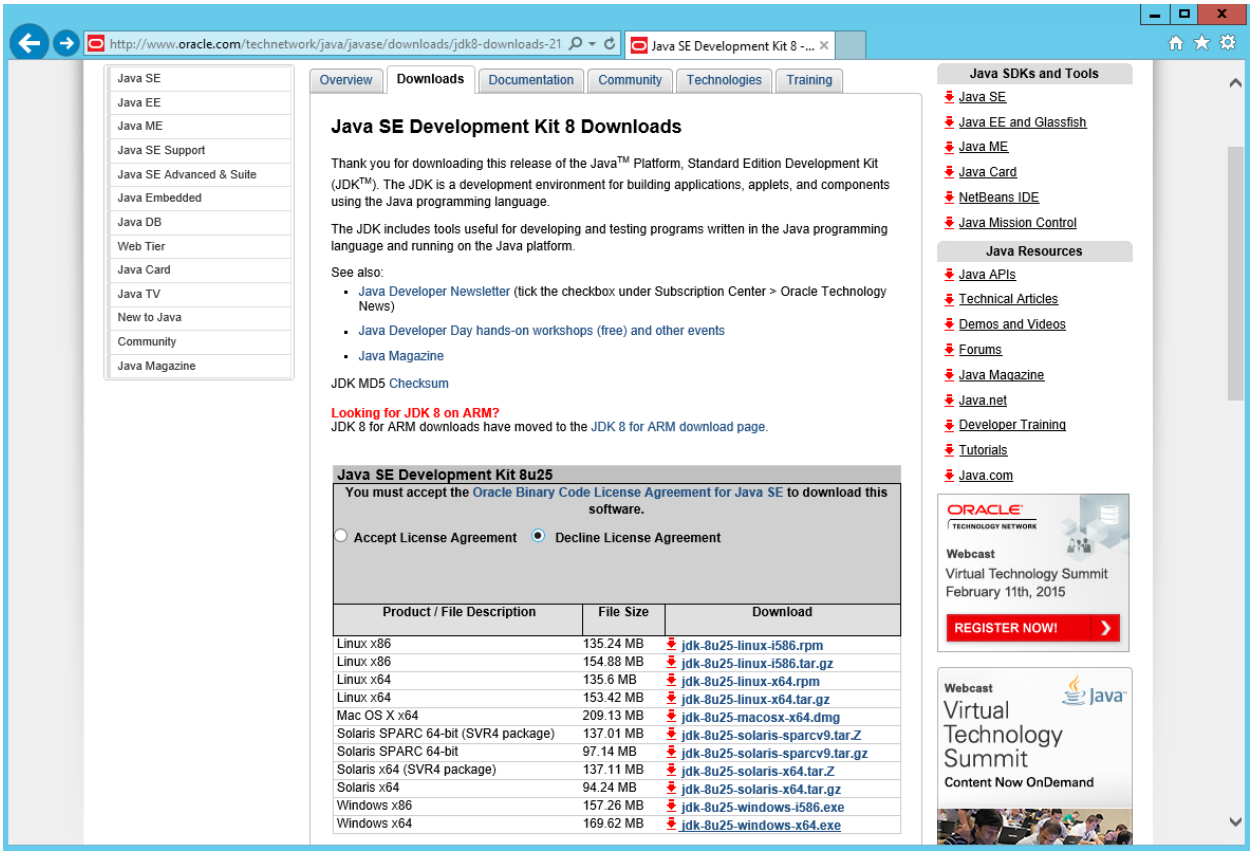


Figure 1 - Java SE Development Kit 8 Downloads.

Problems are likely to arise if you attempt to install on a platform that is not supported or on a system that in some other way does not meet release requirements. Install the correct version of the JDK depending on your system (x86/32-bit or x64/64-bit) and your operating system (Windows, Linux or OS X).

For Mac OS X systems, GlassFish Server uses the JDK that is part of the Macintosh operating system. If necessary, obtain the required JDK version from the Mac OS X Updates site:

http://support.apple.com/downloads/

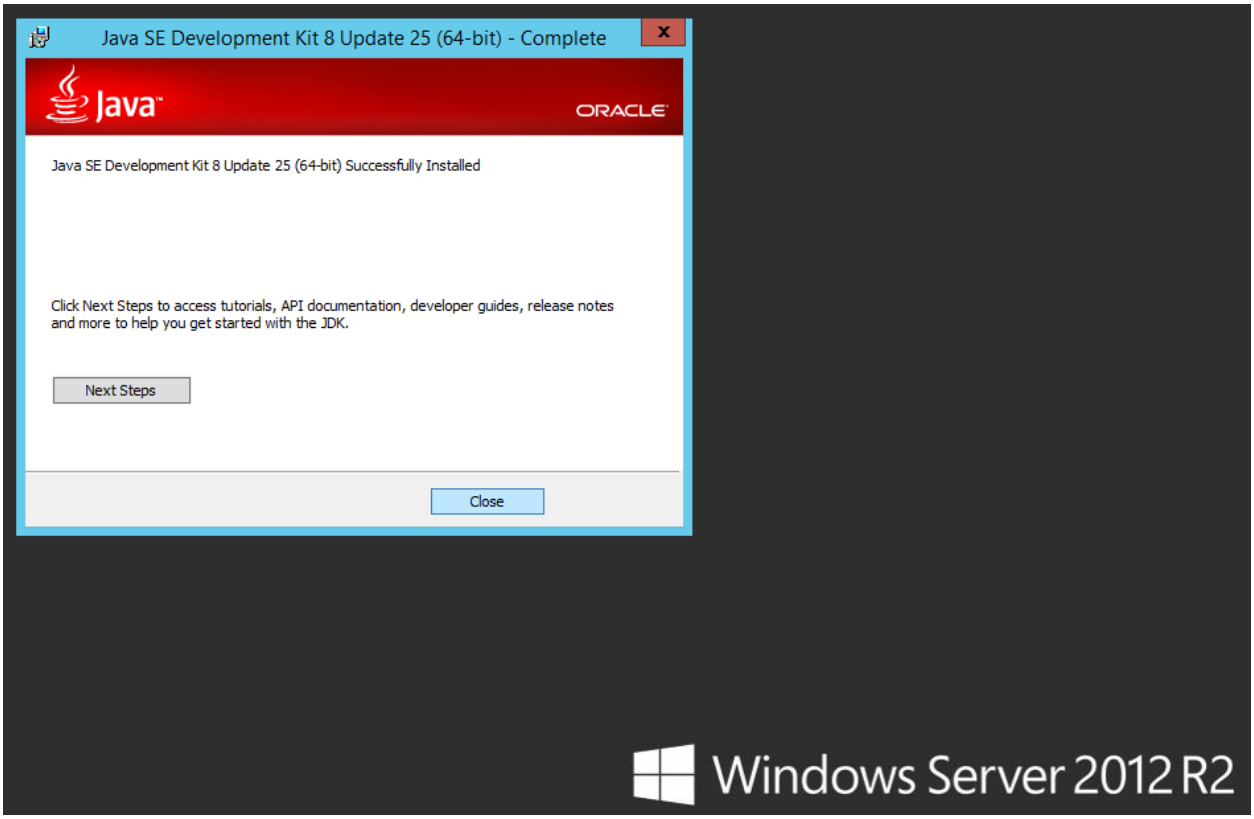


Figure 2 - JDK successfully installed.

GlassFish is also part of the Java EE SDK distributions. The SDK distributions are available from the Java EE downloads page:

http://www.oracle.com/technetwork/java/javaee/downloads/index.html

2.2. Download GlassFish

You can download GlassFish 4.1 Full Platform from the GlassFish Server Open Source Edition 4.1 download page:

https://glassfish.java.net/download.html

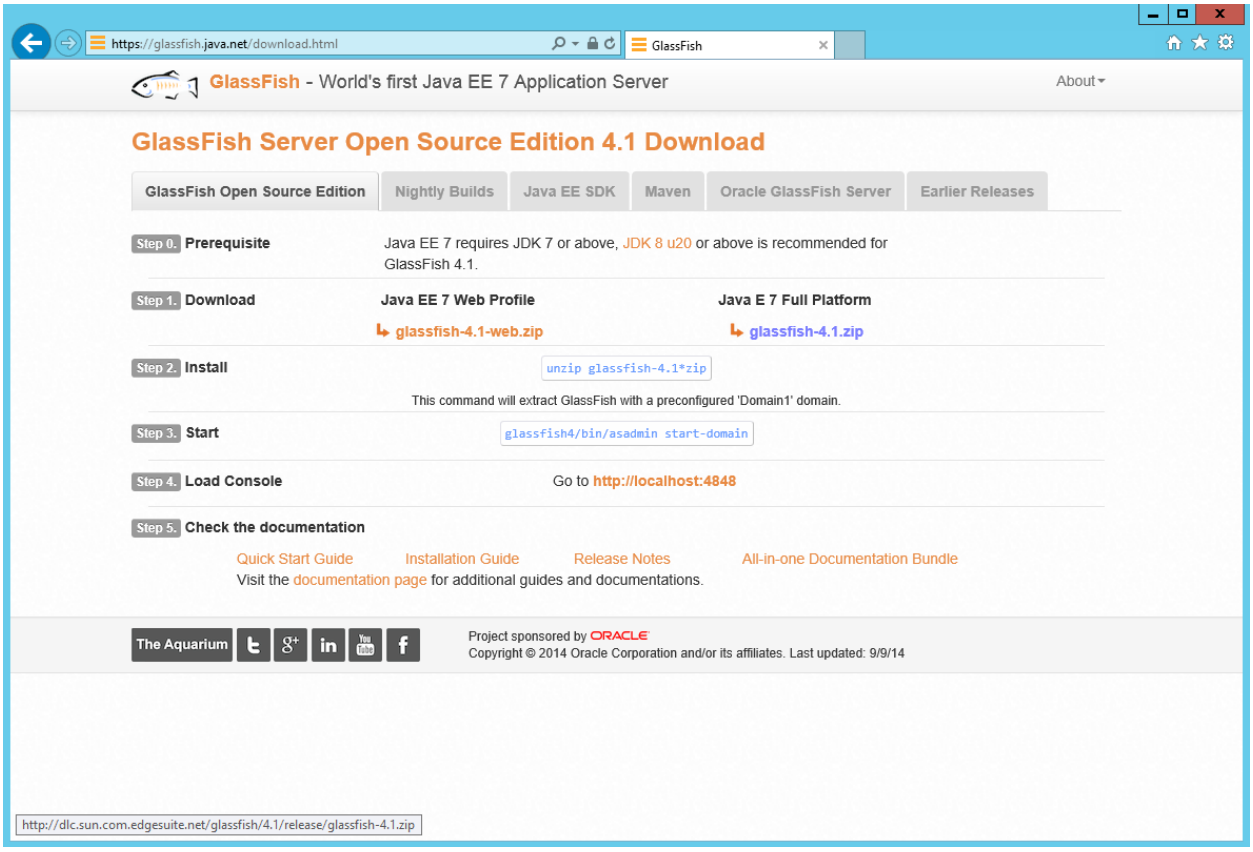


Figure 3 - GlassFish Server Open Source Edition 4.1 Download.

It is compatible with Solaris, Linux, Mac OS and Windows operating systems.

2.3. Install GlassFish

GlassFish Server is installed by unzipping the file in the installation directory of your choice: just extract the ZIP file in the directory where want to have GlassFish installed (e.g. `C:\`).

Since it is not possible to specify any GlassFish Server configuration options during a ZIP file installation, by default, GlassFish is configured with one domain called `domain1` which uses port `8080` for HTTP, port 8181 for HTTPS and port 4848 for administration. Any configuration changes must be made manually after installation (please see *Section 3.3 – Configure GlassFish*).

GlassFish Server is installed into a new `glassfish4` directory under your current directory (e.g. `C:\glassfish4`).

2.4. Start GlassFish

To start GlassFish Server from the command line (e.g. Windows PowerShell), open a terminal window or command prompt and invoke, from the GlassFish directory, the following command:

```
glassfish4/bin/asadmin start-domain –verbose
```

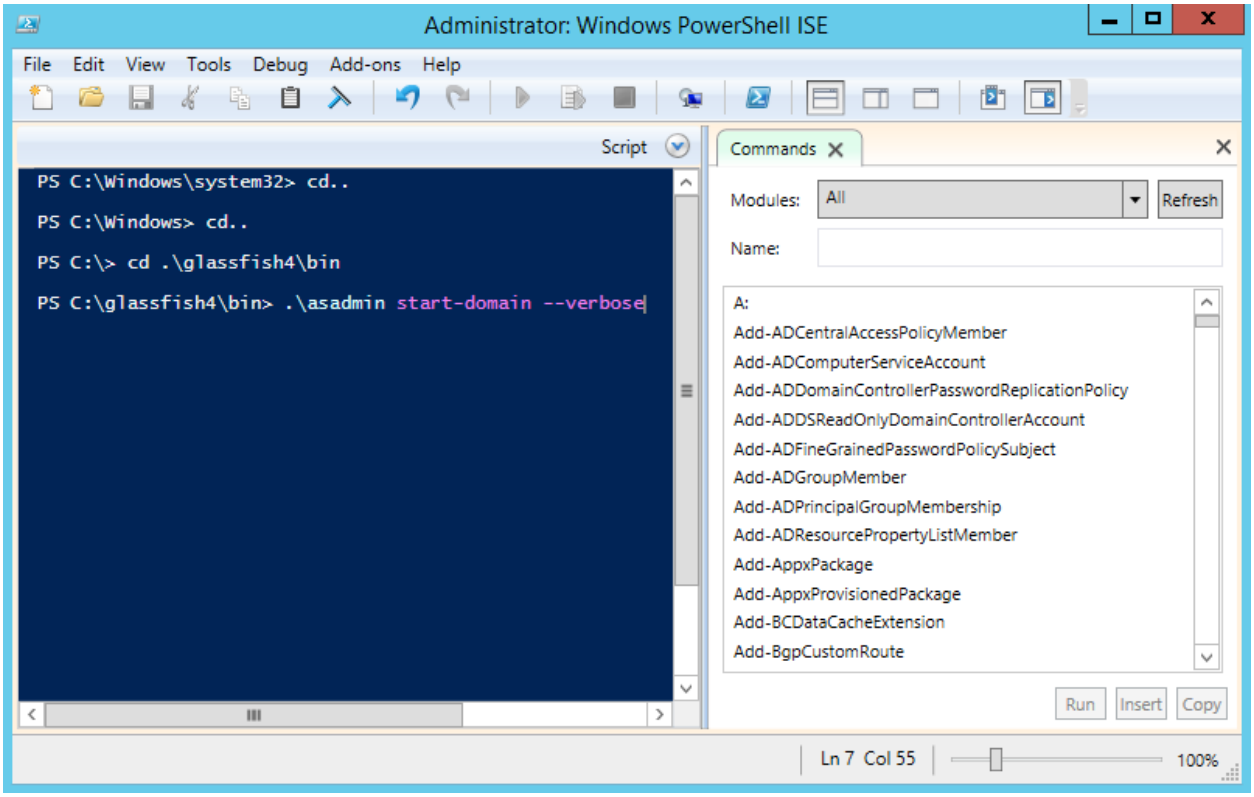


Figure 4 - Command to start the GlassFish server.

Note that, since you only have one domain (i.e. a set of one or more GlassFish Server instances managed by one administration server) configured, there is no need to mention which domain to start.

With no arguments, the `start-domain` command initiates the default domain, which is `domain1`. The `--verbose` flag causes all logging and debugging output to appear on the terminal window or command prompt. The output also goes into the server log, which is located in `domain-dir/logs/server.log`

If the message:

```
Loading application __admingui done
```

is displayed, the server has successfully been started.

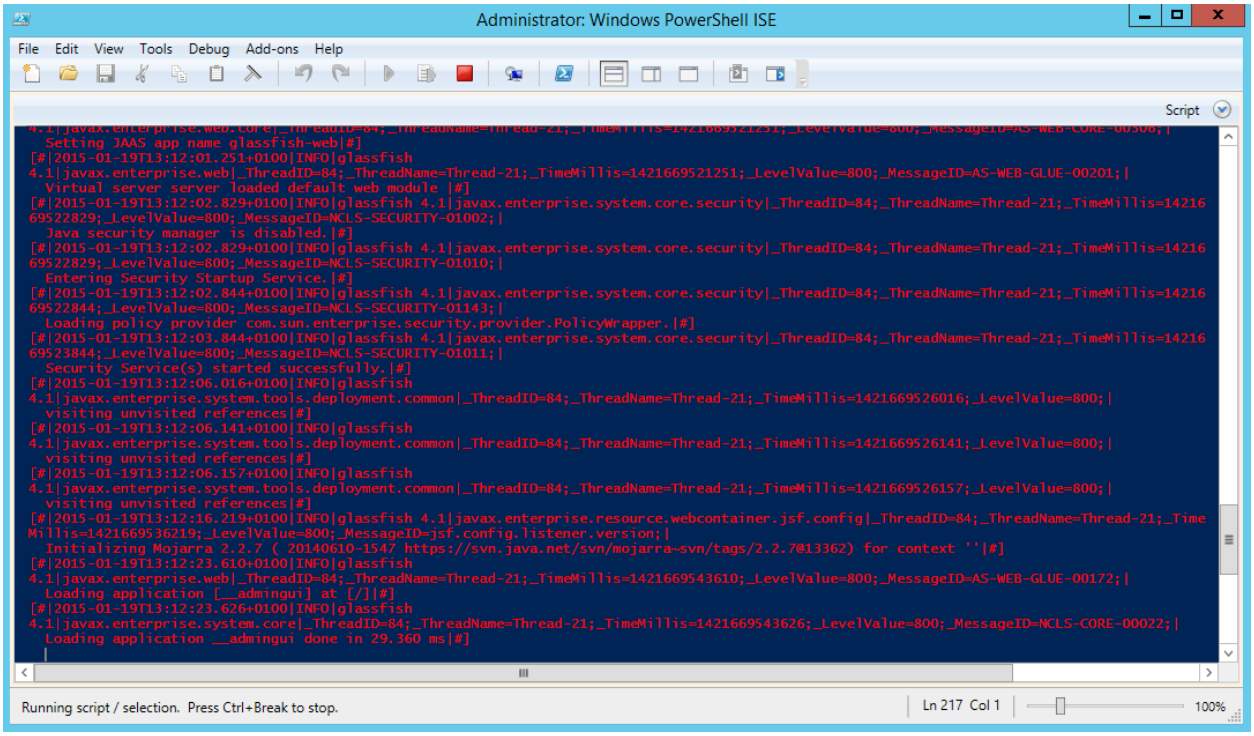


Figure 5 - GlassFish has been successfully loaded.

To stop GlassFish, open a terminal window or command prompt and execute the command:

```
glassfish4/bin/asadmin stop-domain domain1
```

2.5. Install MySQL Community Server

To complete the installation we have to install the database server that MeteoCal beta will use. We use a MySQL database. A free version of the Community Server is available here:

http://dev.mysql.com/downloads/mysql/

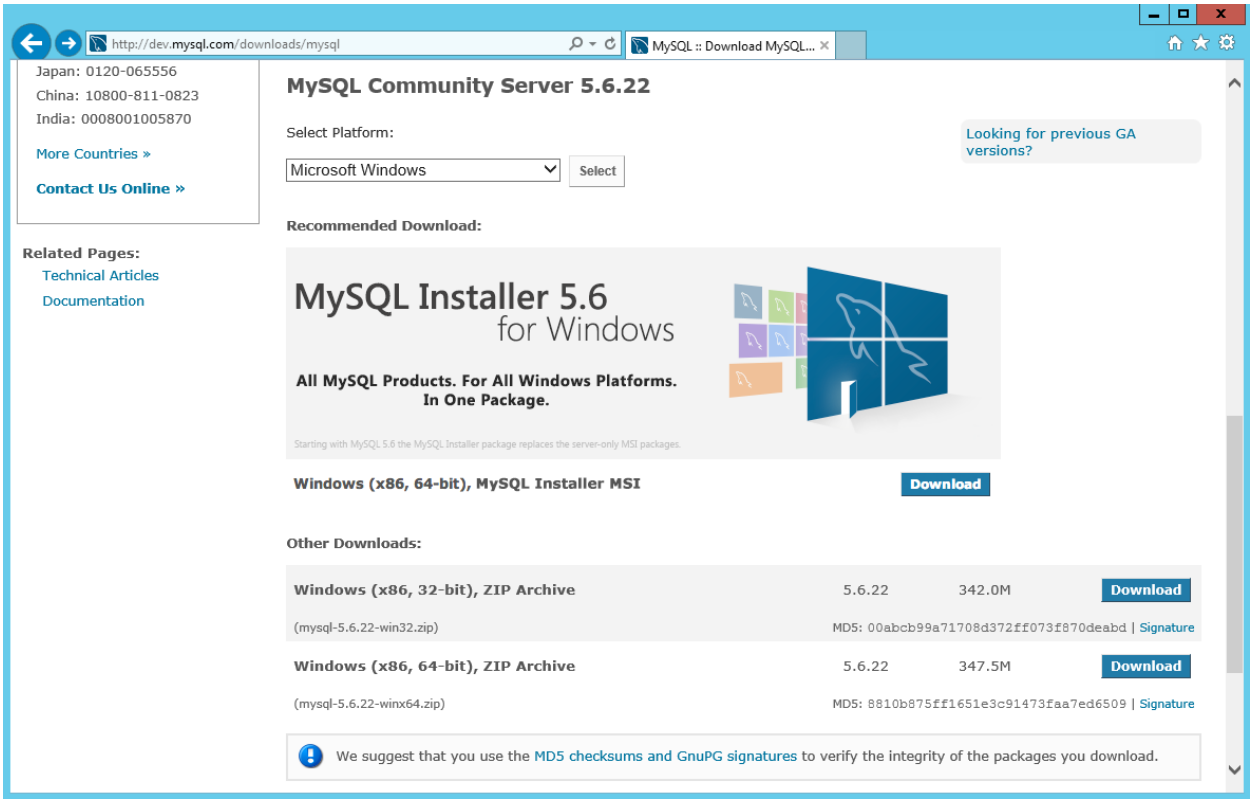


Figure 6 - MySQL Community Server download page.

Then follow the installation wizard, as in the following steps.

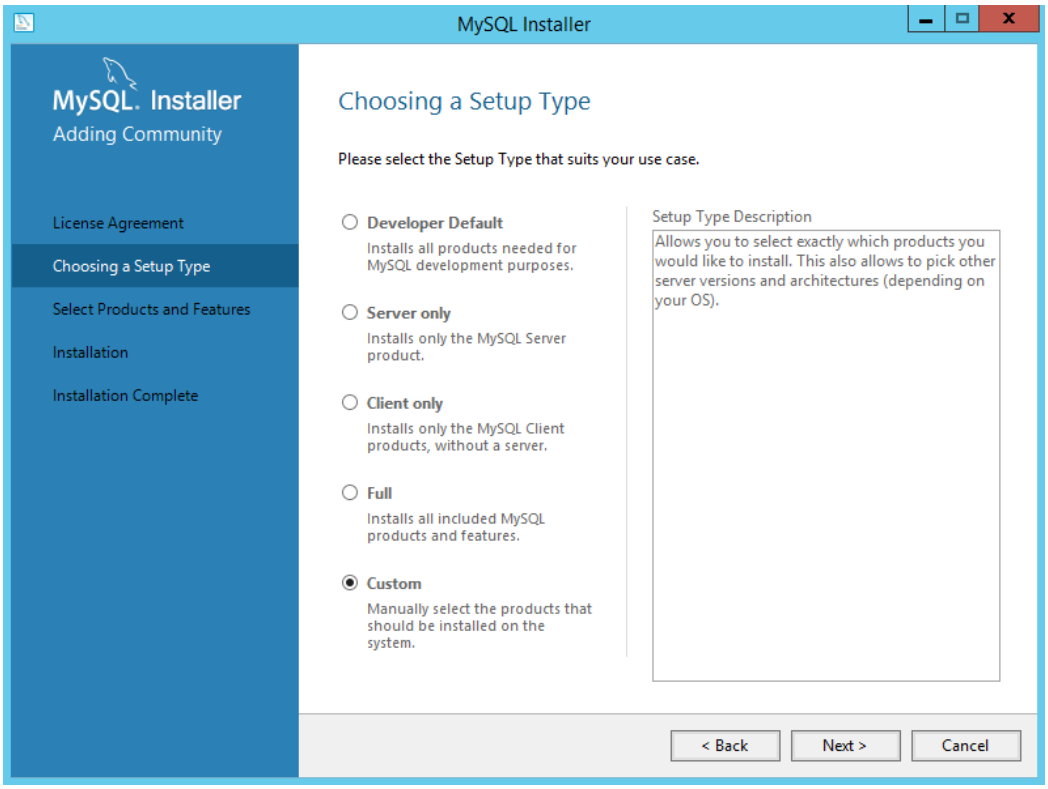


Figure 7 - Select the custom installation and, then click "Next".

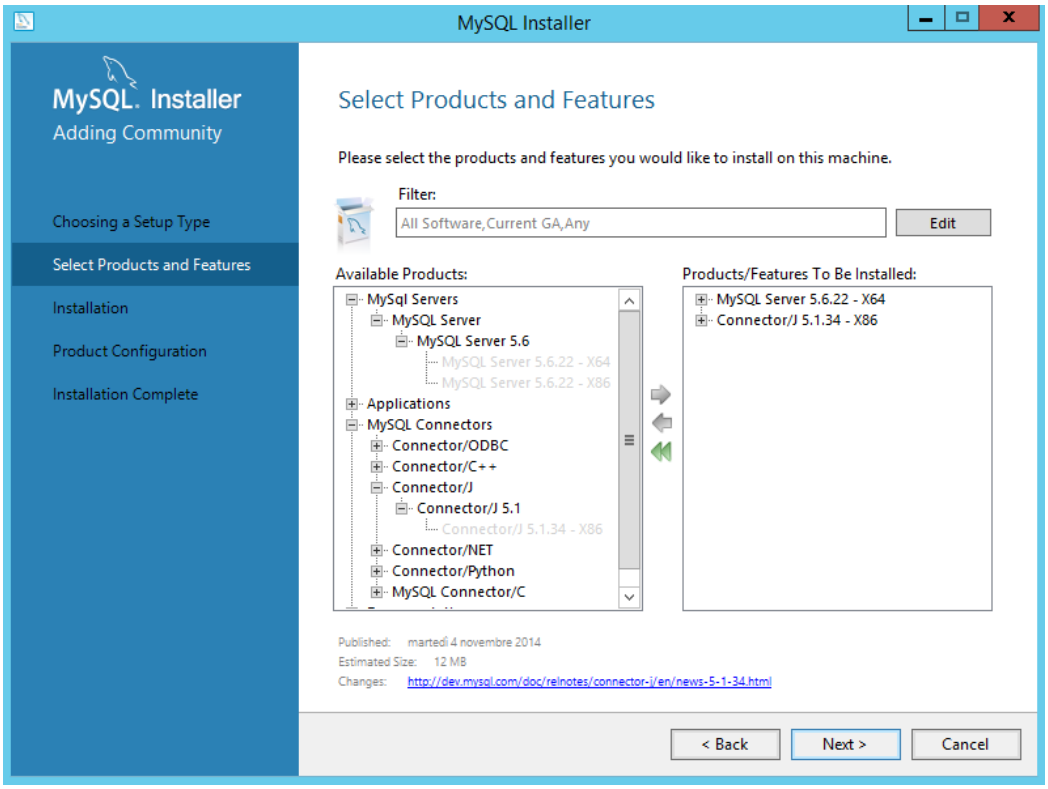


Figure 8 - Select the required MySQL Server and the Connector/J using the arrows in the middle of the window, then click "Next".

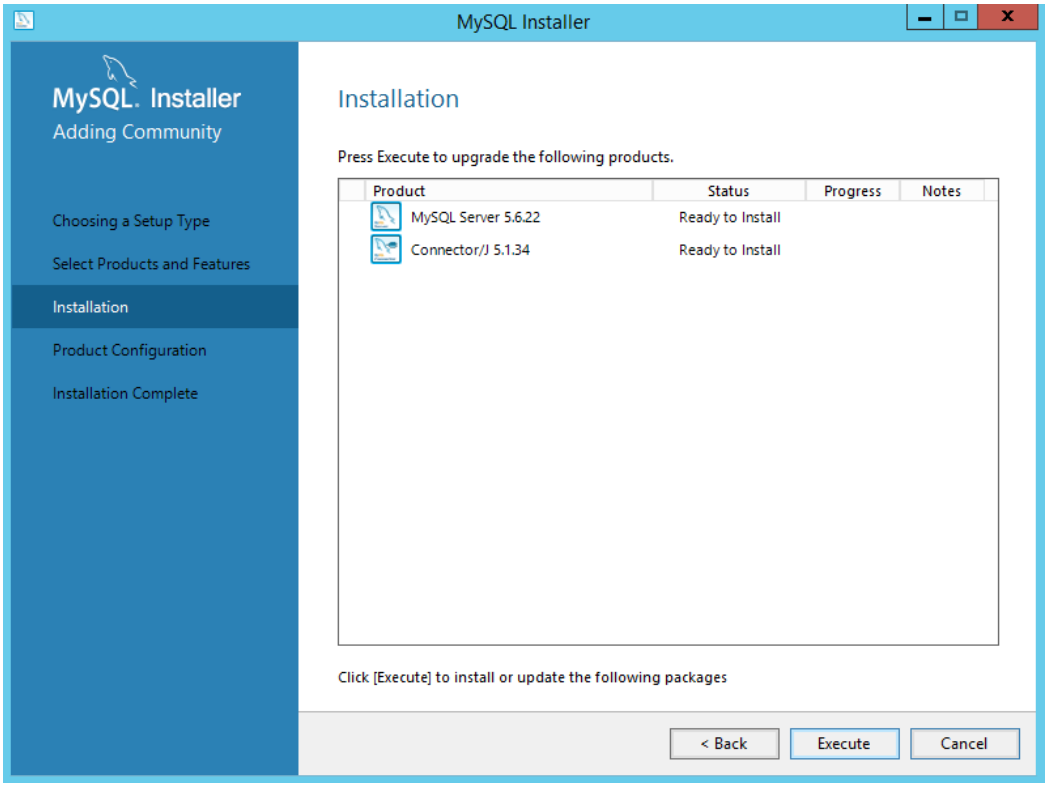


Figure 9 - Confirm clicking on "Execute".

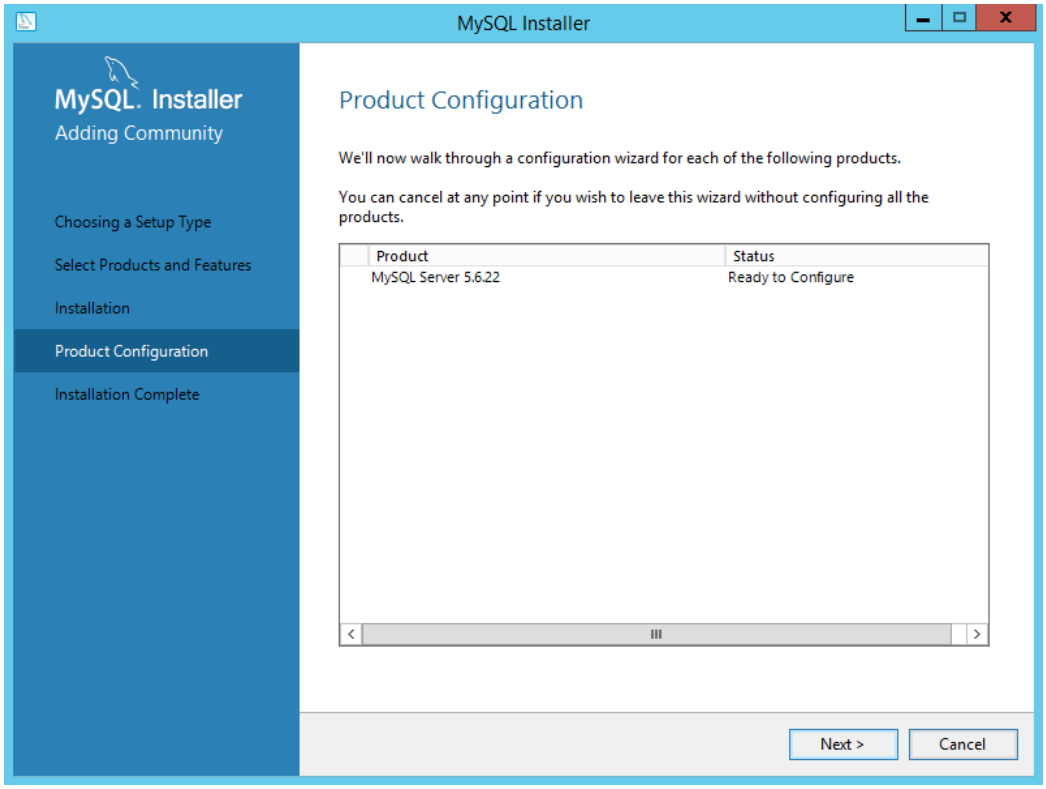


Figure 10 - Confirm clicking on "Next".

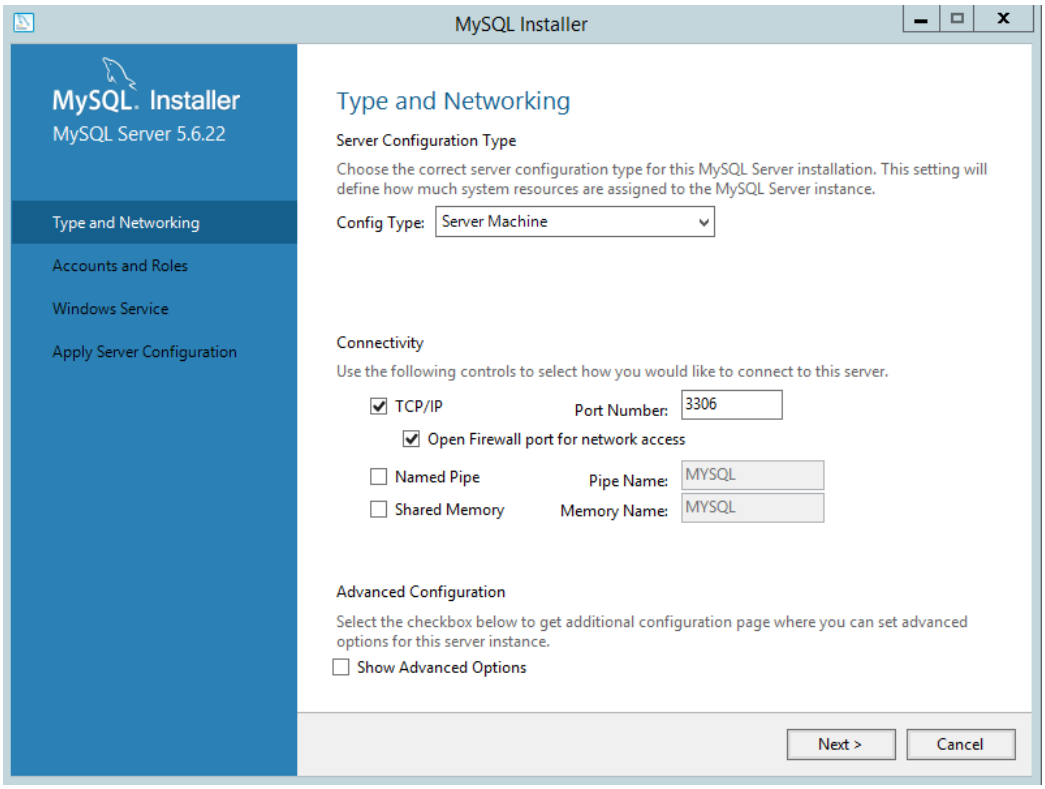


Figure 11 - Confirm clicking on "Next".

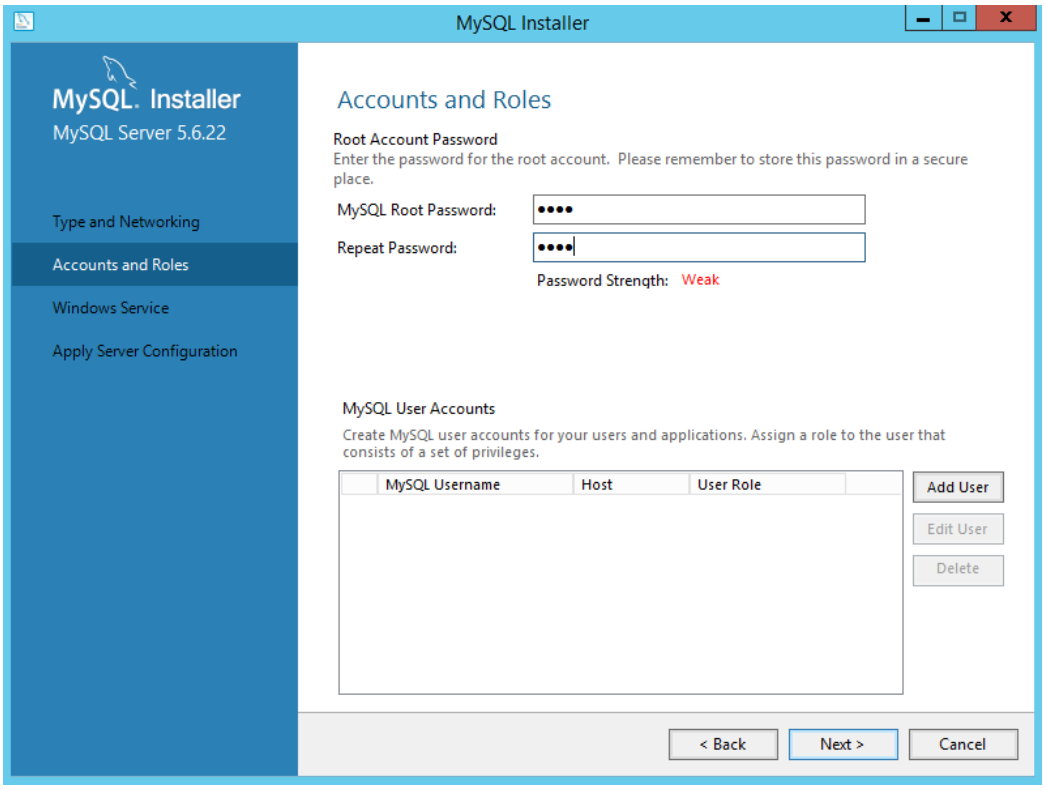


Figure 12 - Insert the password (we have used "root") and then click "Next". You cannot leave this field empty.

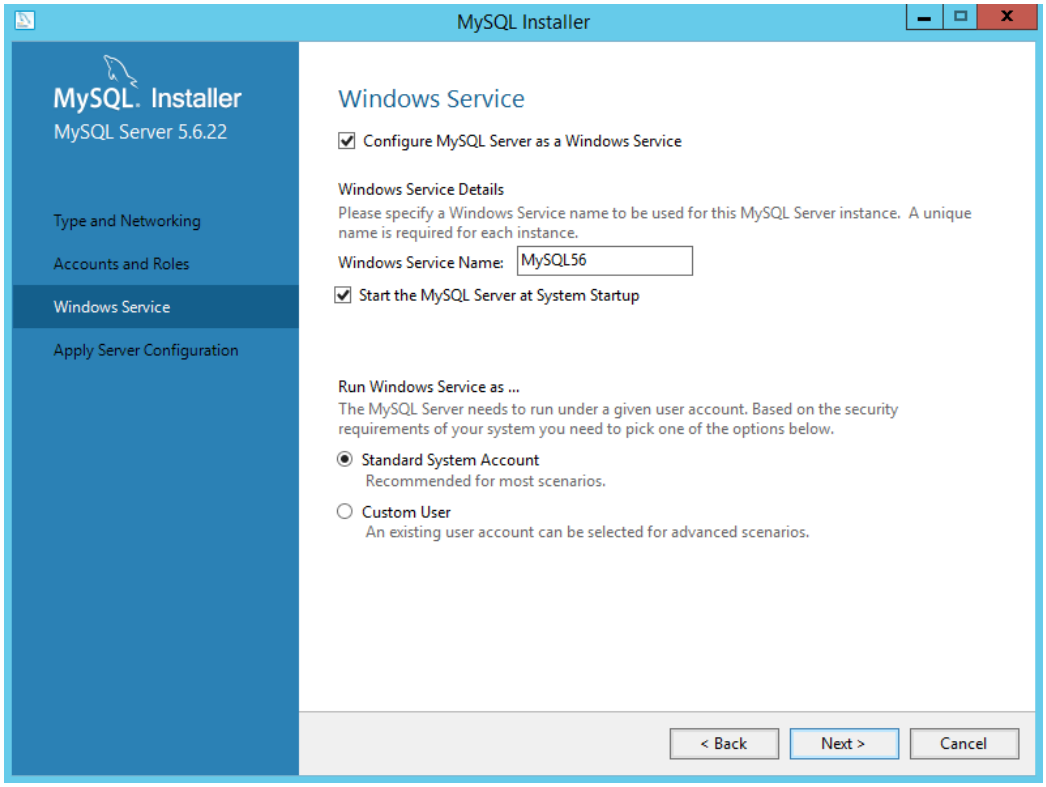


Figure 13 - Click on "Next".

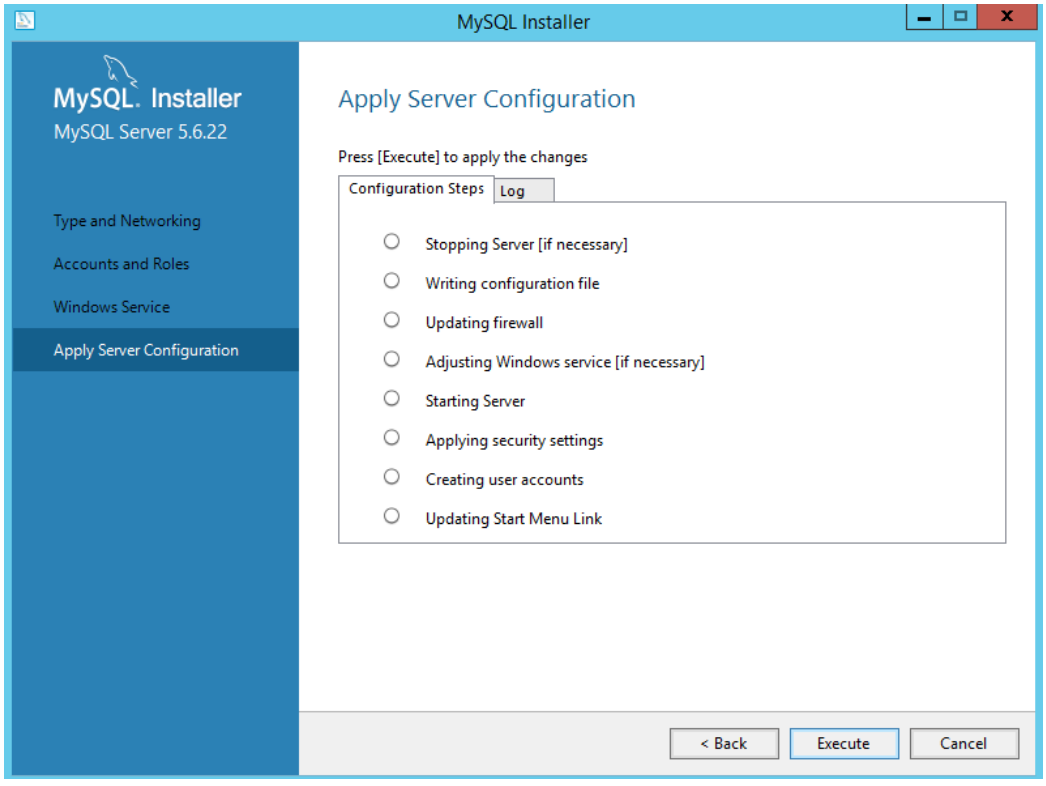


Figure 14 - Click on "Execute".

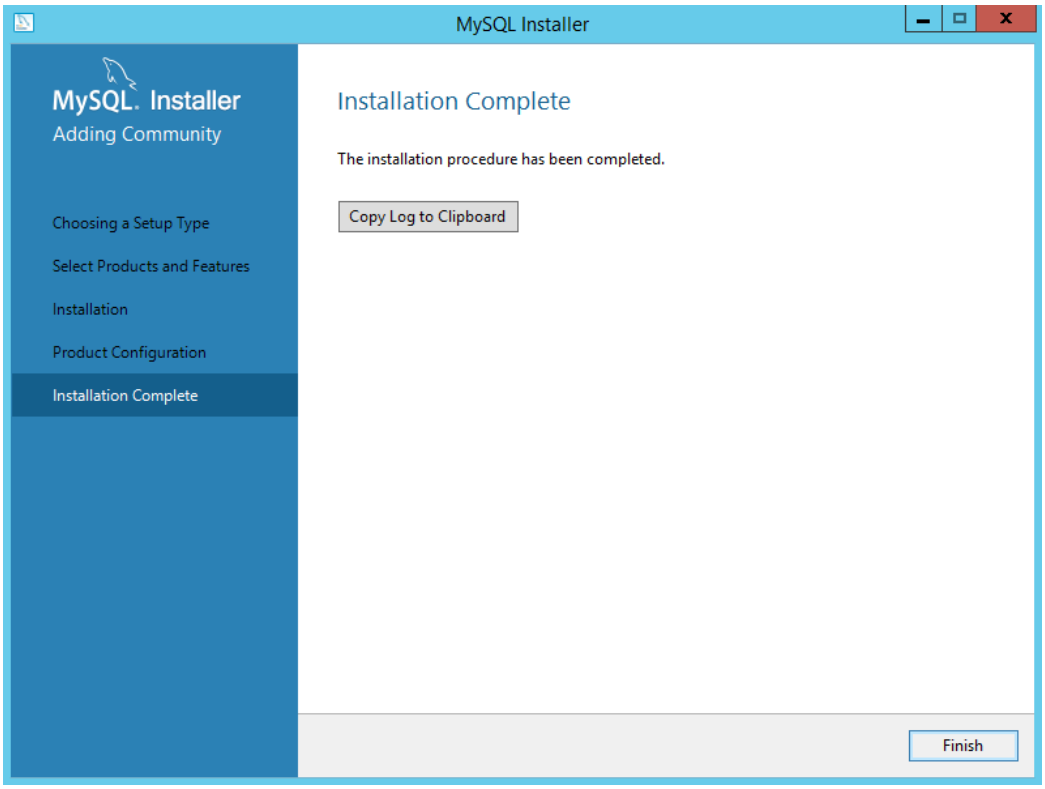
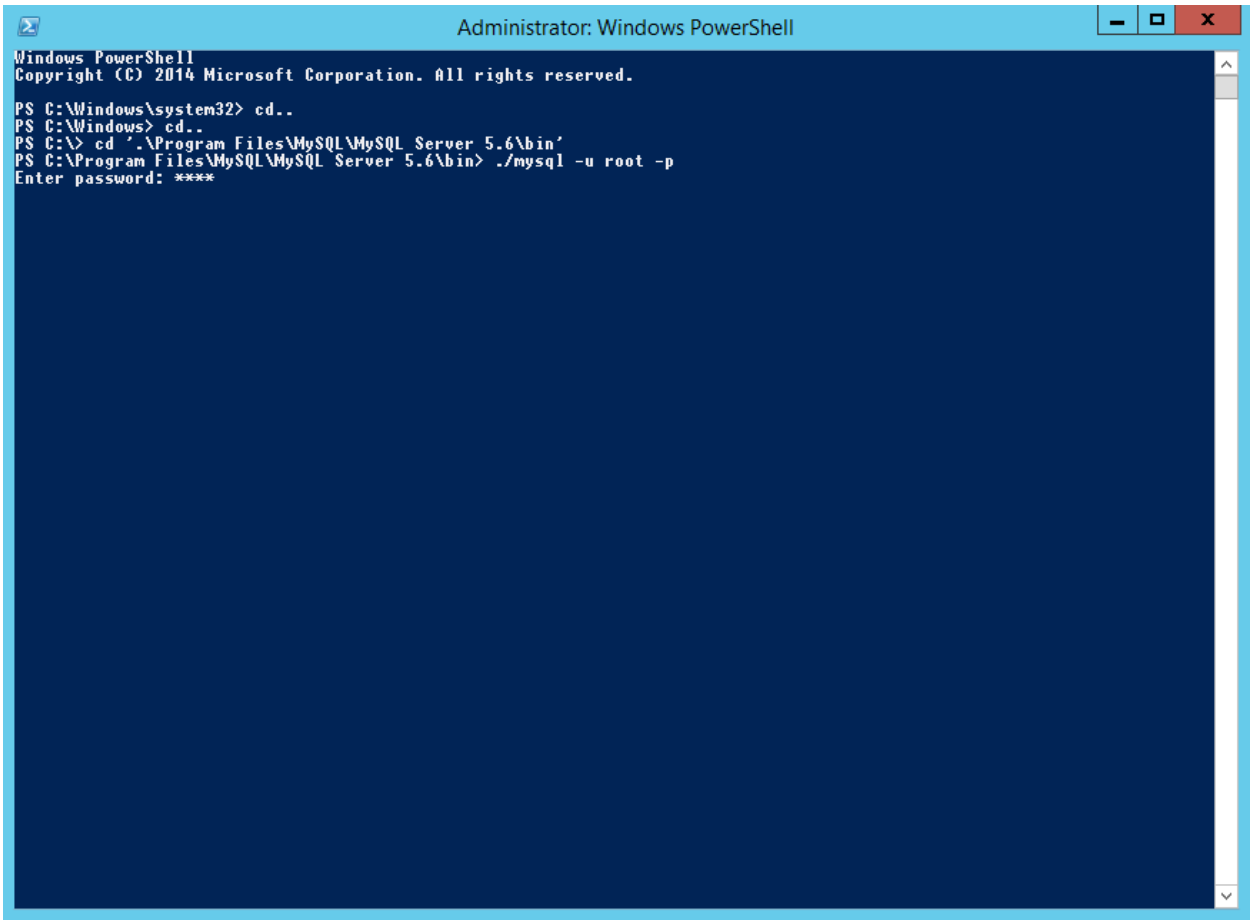


Figure 15 - Click on "Finish".

2.6. Run MySQL

To start MySQL, just move in the installation folder and, then, run `mysql -u root -p`. When requested, provide the password (e.g. `root`) and enter.



3. Database configuration

3.1. Create the database schema in MySQL

Download this file (e.g. in `C:\Users\Administrator\Downloads`):

https://meteocal-fermi-grillo-jerry.googlecode.com/svn/downloads/meteocaldb-dump.sql

Inside the MySql console (opened in *Section 2.6. – Run MySQL*) type this command to load the actual database (i.e. `meteocaldb`):

```
source C:\Users\Administrator\Downloads\meteocaldb-dump.sql
```

3.2. Install MySQL Java connector

Now you have to install the MySQL Java connector (i.e. `Connector/J`) in GlassFish to let the Server communicate with the database. You just have to copy the file (e.g. `mysql-connector-java-5.1.34-bin`) from the MySQL folder (e.g. `C:\Program Files (x86)\MySQL\Connector.J 5.1`) to the GlassFish Server folder:

```
glassfish4/glassfish/bin
```

3.3. Configure GlassFish

Open your browser and open the admin's GlassFish console going to:

http://localhost:4848/

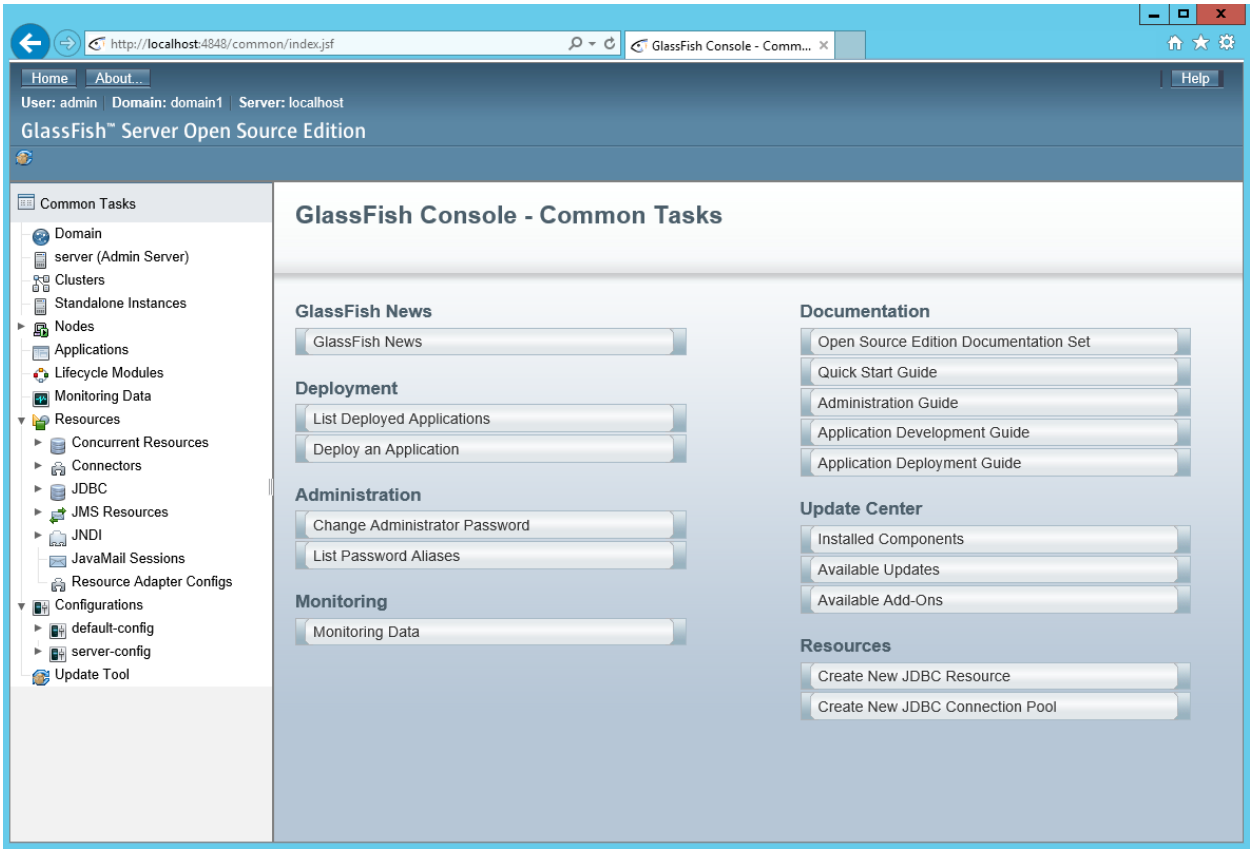


Figure 16 - Glassfish admin's console.

To allow the form-based authentication, create a JDBCRealm following these steps:

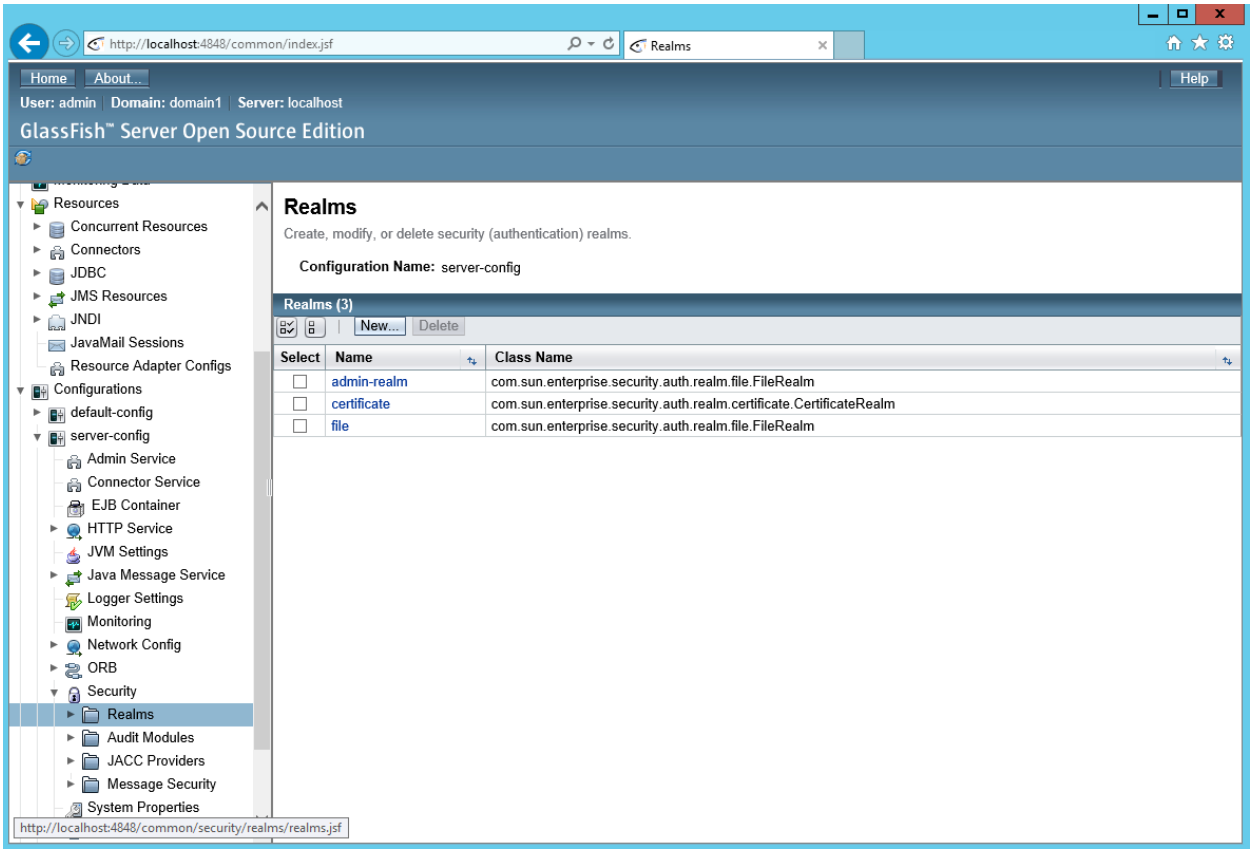


Figure 17 - Configurations > server-config > Realms > New...

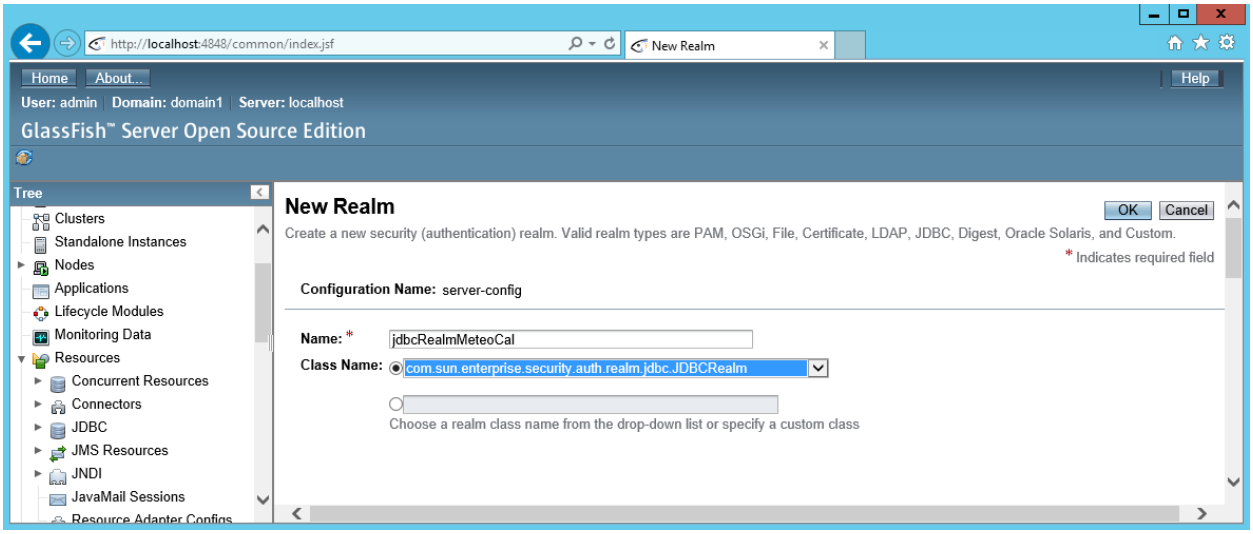


Figure 18 - Name: jdbcRealmMeteoCal > Class Name: com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm > OK

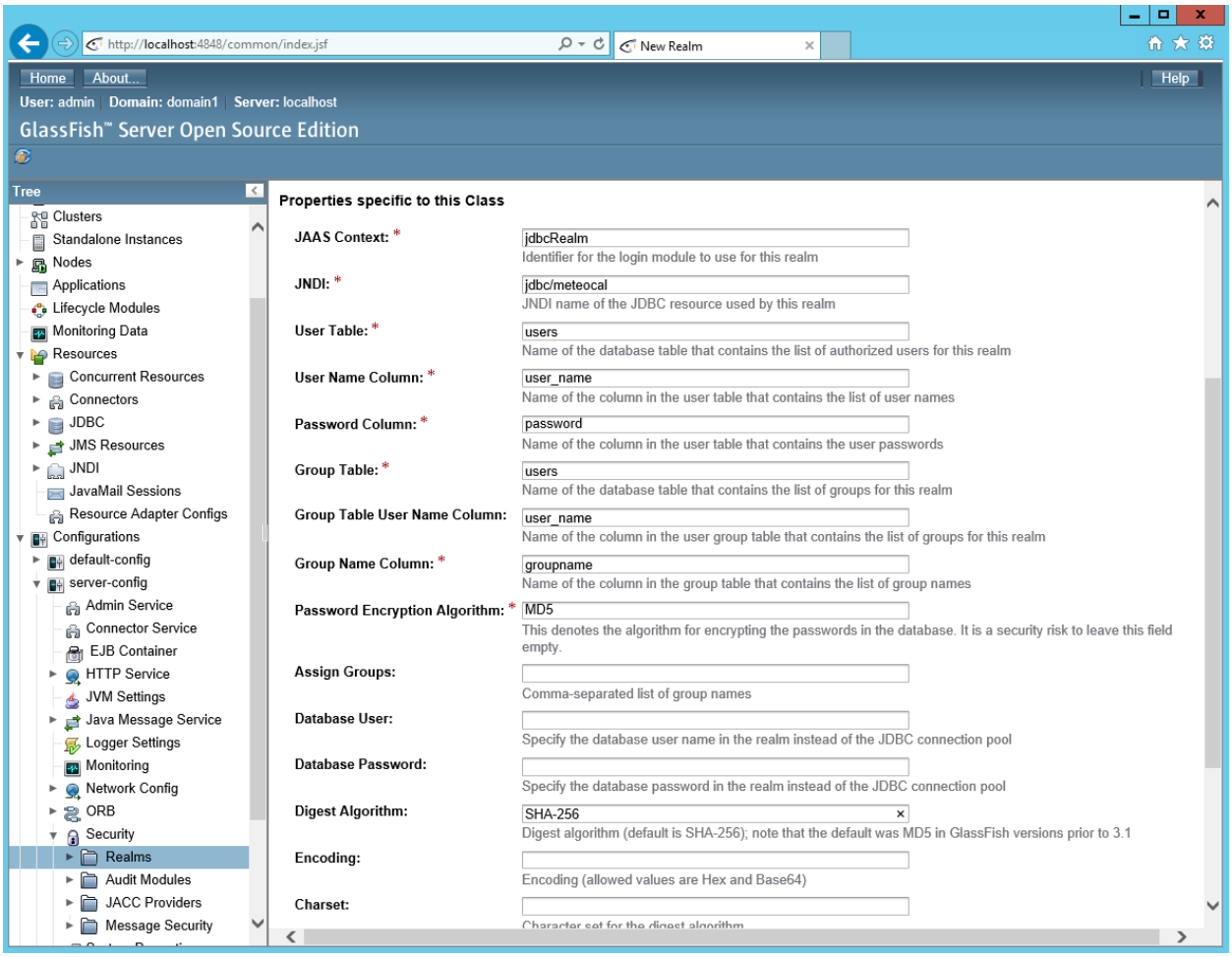


Figure 19 - In the bottom of the page, JAAS Context: jdbcRealm > JNDI: jdbc/meteocal > User Table: users > User Name Column: user_name > Password Column: password > Group Table: users > Group Table User Name Columns: user_name > Group Name Column: groupname > Password Encryption Algorithm: MD5 > Digest Algorithm: SHA-256.

Now re-start the server to configure the JDBC connection pool. To restart the server, in the `Common task` menu, select `server (Admin server)` and then click on the `Stop` button. At the end repeat the starting procedure described in *Section 2.4 – Start GlassFish*.

Continue with the following steps:

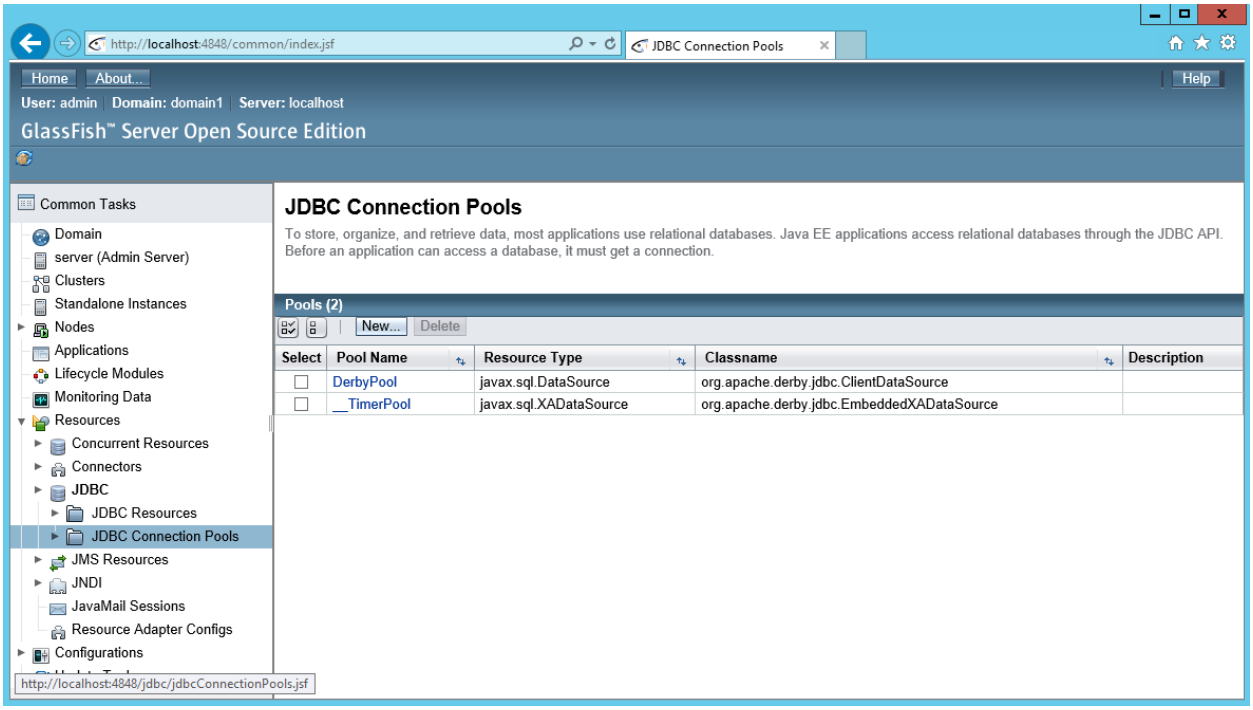


Figure 20 - Resources > JDBC > JDBC Connection Pools > New...

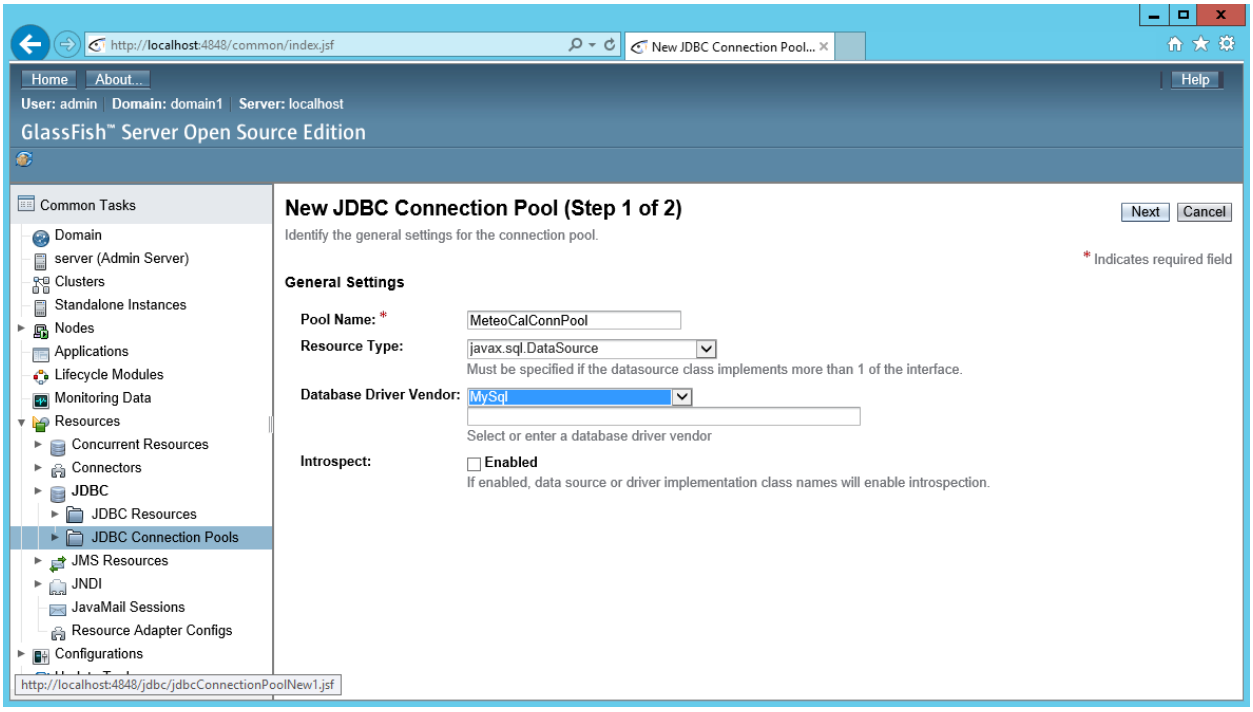


Figure 21 - Pool Name: MeteoCalConnPool > Resource Type: javax.sql.DataSource > Database Driver Vendor: MySql > Next

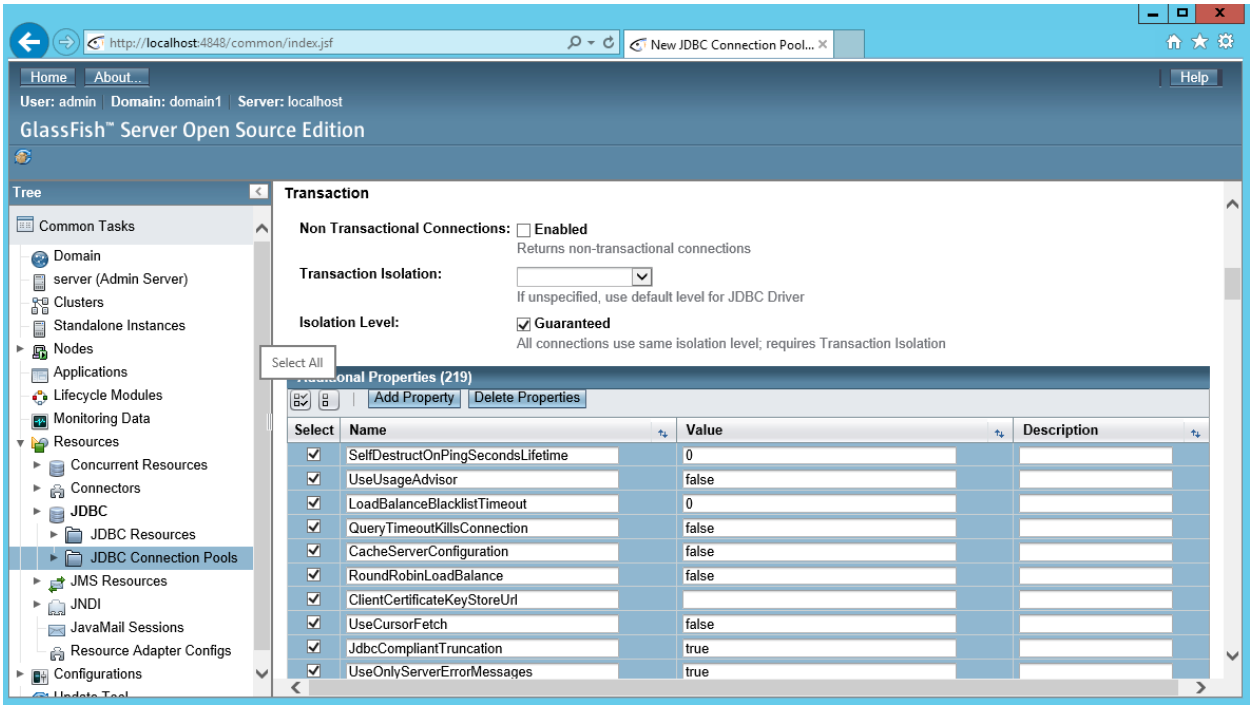


Figure 22 - At the botton of the screen, select all the rows and click on "Delete Properties".

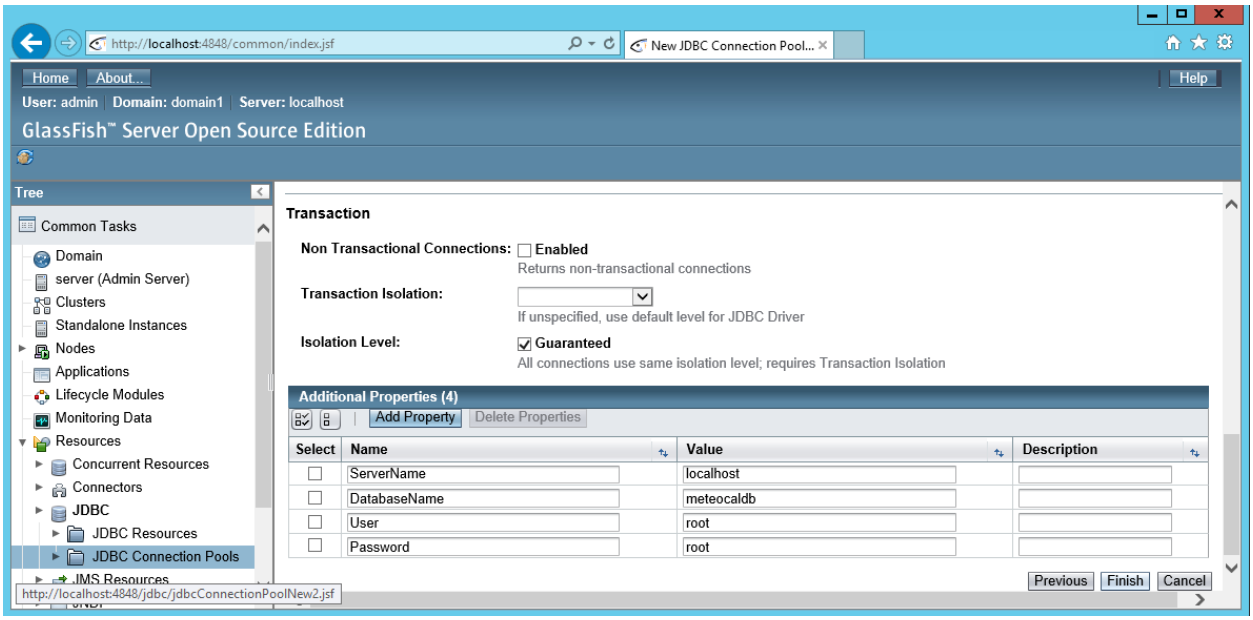


Figure 23 - ServerName: localhost > DatabaseName: meteocaldb > User: root > Password: root > Finish

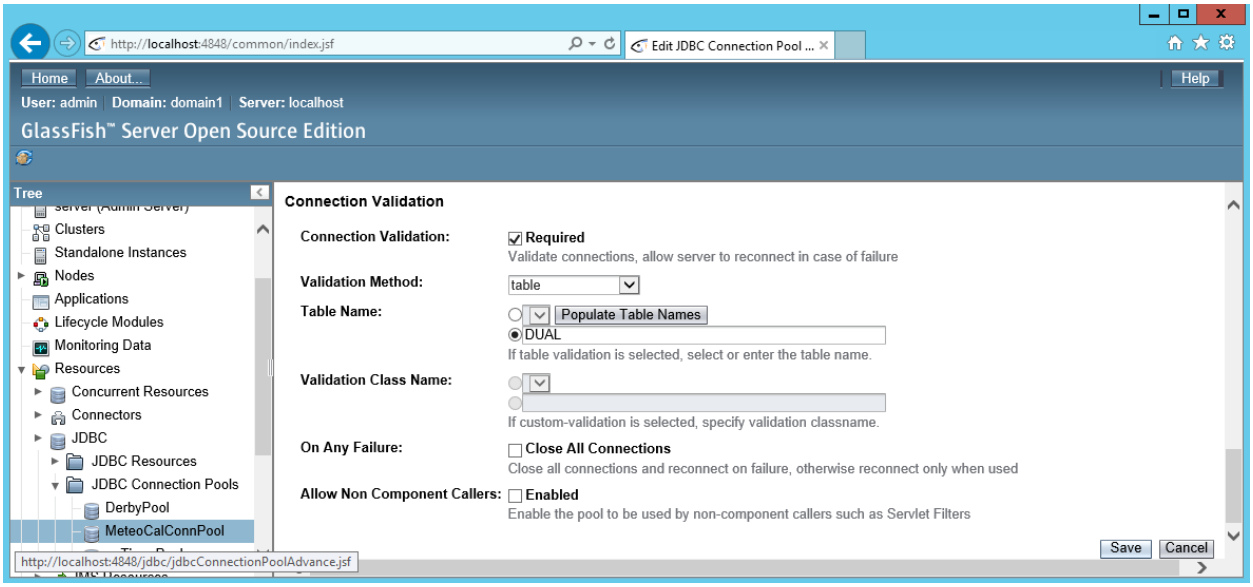


Figure 24 - In the JDBC Connection Pools frame click on the connection pool you just created. Here, you can review and edit information about the connection pool. Because Connector/J does not support optimized validation queries, go to the Advanced tab and, under Connection Validation, configure the following settings: 1) Connection Validation: Required; 2) Validation Method: table; 3) Table Name: DUAL.

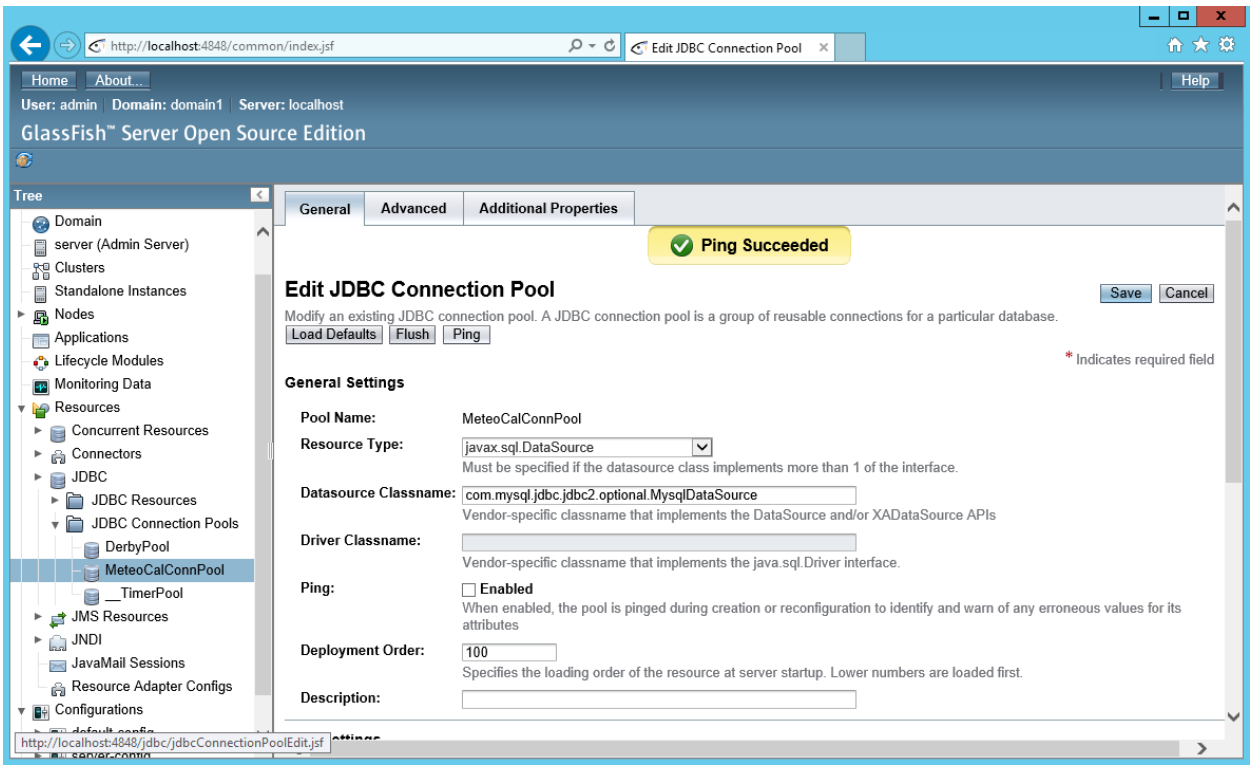


Figure 25 - To test your connection pool click the "Ping" button at the top of the frame. A message will be displayed confirming correct operation (or otherwise). If an error message is received recheck the previous steps and ensure that MySQL Connector/J has been correctly copied into the previously specified location (see Section 3.2 – Install MySQL Java connector).

Now that you have created a connection pool you will also need to create a JDBC Resource (data source) for use by MeteoCal, as follows:

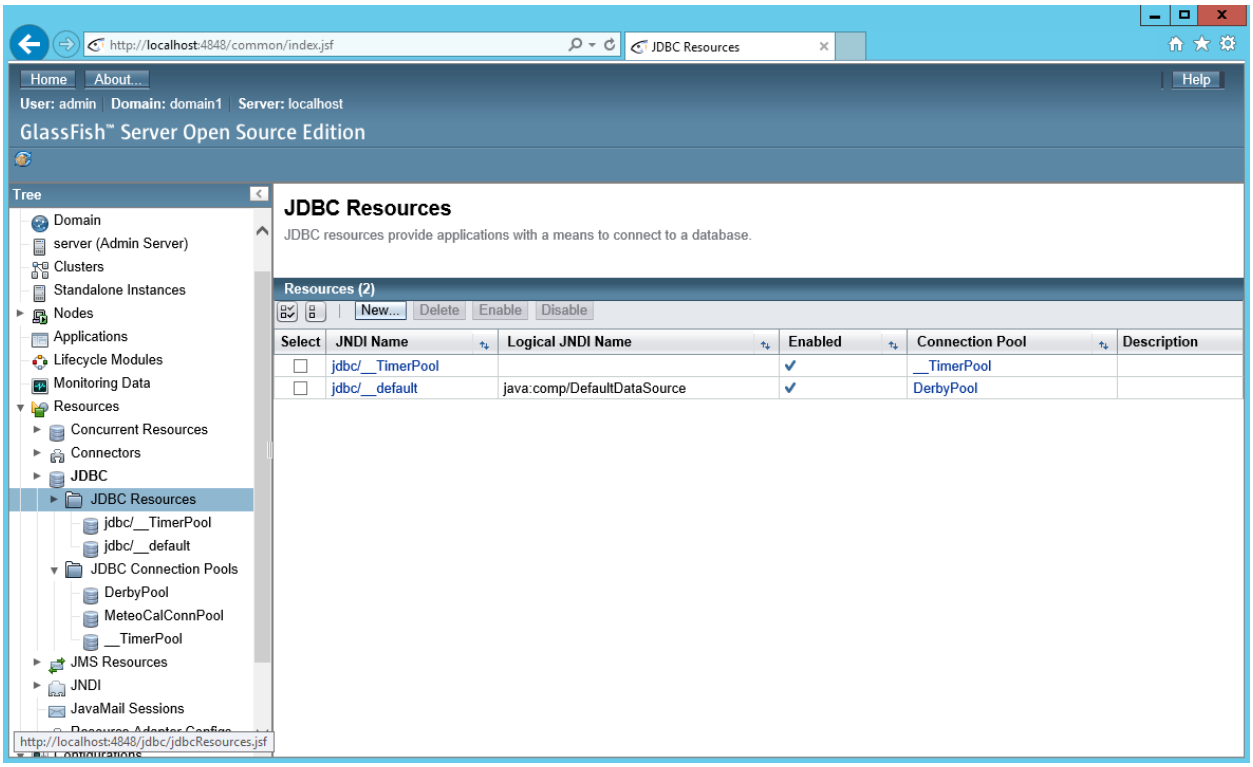


Figure 26 - Resources > JDBC Resources > New...

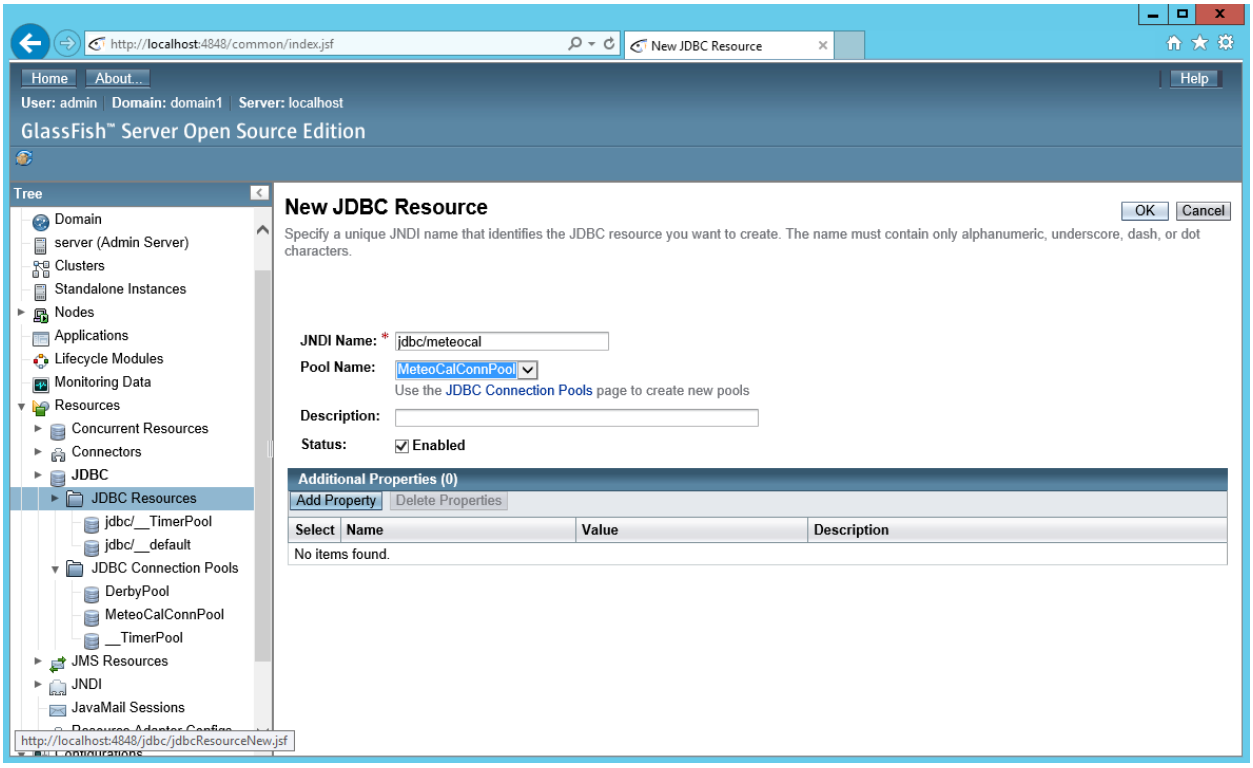


Figure 27- JNDI Name: jdbc/meteocal > Pool Name: MeteoCalConnPool > OK

4. Application deployment

4.1. Deploy the application in GlassFish

To deploy the application, just download the `meteocal.war` file from here:

https://meteocal-fermi-grillo-jerry.googlecode.com/svn/downloads/meteocal.war

then, from the main GlassFish page select `Deploy an application`, browse the previously downloaded file, finally, confirm clicking on the `OK` button in the upper-right corner of the page. The deployed application will be available at the following address:

http://localhost:8080/meteocal

Thank you

Appendix A – Secure links for server OSs

Trusted sites

- www.oracle.com
- download.oracle.com
- edelivery.oracle.com
- dlc.sun.comom.edgesuite.net
- cdn.mysql.com

Direct links

- https://edelivery.oracle.com/otn-pub/java/jdk/8u25-b18/jdk-8u25-windows-x64.exe

- http://dlc.sun.com.edgesuite.net/glassfish/4.1/release/glassfish-4.1.zip

- http://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-community-5.6.22.0.msi

Appendix B – Server settings recap

Configurations

- **server-config** (name)
  - **Security**
    - **Realm**
      - **Name**: `jdbcRealmMeteoCal`
      - **Class name**: `com.sun.enterprise.securit.auth.realm.jdbc.JDBCRealm`
        - **Properties**
          - **JAAS Context:** `jdbcRealm`
          - **JNDI**: `jdbc/meteocal`
          - **User table**: `users`
          - **User name column**: `user_name`
          - **Password column**: `password`
          - **Group table**: `users`
          - **Group table user name column**: `user_name`
          - **Group name column**: `groupname`
          - **Password encryption algorithm**: `MD5`
          - **Digest algorithm**: `SHA-256`

Resources

- **JDBC**
  - **Connection pool**
    - **General**
      - **General settings**
        - **Pool name**: `MeteoCalConnPool`
        - **Resource type**: `javax.sql.DataSource`
        - **Database driver vendor**: `MySql`
        - **Datasource classname**: `com.mysql.jdbc2.optional.MysqlDataSource`
        - **Introspect**: not enabled
        - **Ping**: not enabled
        - **Deployment order**: `100`
      - **Transaction**
        - **Non transactional connections**: not enabled
        - **Transaction isolation**: default level for JDBC Driver
        - **Isolation level**: guaranteed
      - **Pool settings**
        - **Initial and minimum pool**: 8 connections
        - **Maximum pool size**: 32 connections
        - **Pool resize quantity**: 2 connections
        - **Idle timeout**: `300` seconds
        - **Max wait time**: `60000` milliseconds
    - **Additional properties** (Name/Value)
      - `ServerName/localhost`
      - `DatabaseName/meteocaldb`
      - `User/root`
      - `Password/root`

        - **Advanced attributes**
            - **Connection validation**
                - **Connection validation**: required
                - **Validation method**: table
                - **Table name**: DUAL
                - **On any failure**: reconnect only when used
                - **Allow non-component callers**: not enabled
            - **Connection settings**
                - **Validate at most once**: 0 seconds
                - **Connection leak timeout**: 0 seconds
                - **Connection leak reclaim**: not enabled
                - **Statement leak timeout**: 0 seconds
                - **Statement leak reclaim**: not enabled
                - **Creation retry attempts**: 0
                - **Retry interval**: 10 seconds
                - **Lazy association**: not enabled
                - **Lazy connection enlistment**: not enabled
                - **Associate with thread**: not enabled
                - **Match connections**: not enabled
                - **Max connection usage**: 0
            - **Statement timeout**: not enabled
            - **Statement cache size**: 0
            - **Wrap JDBC objects** (enabled)
                - Statement
                - PreparedStatement
                - CallableStatement
                - ResultSet
                - DatabaseMetaData
            - **Pooling** (enabled)
                - Connection pooling for the pool
    - **Resources**
        - **JNDI name**: jdbc/meteocal
        - **Pool name**: MeteoCalConnPool
        - **Status**: enabled

Appendix C – Revision history

Initial release:	v1.0
Current release:	v1.0
Date of the last review:	2015/01/24

V0.90

2014/01/19 – Release candidate 1 (complete)

V0.99

2014/01/24 – Release candidate 2 (added appendix B)

V1.0

2014/01/24 – Initial release