16-6-2015

Mike Rooijackers, Noor van Oekel, Jordi Knol, Maaike Jansen, Tim Hermens

Group E

Software Development Street

PTS6

Table of Contents

Infrastructure 2

Configure Jenkins 4

Glassfish JDBC Configuration 8

Glassfish JMS Configuration 10

Install Database Server 12

How to use Software Development Street 14

# Infrastructure

**Windows Server 2008 – Jenkins**

Windows Server 2008 R2 x64

Username: administrator

Password: admin

IP address: 192.168.24.50

Subnet mask: 255.255.255.0

Default Gateway: 192.168.24.1

DNS server: 192.168.230.92

* Jenkins, v1.604
  + <http://localhost:8080>
  + Sonar plugin v2.1
  + GitHub plugin v1.11
* Sonarqube, v5.0.1
  + <http://localhost:9000/dashboard/index>
* Java JDK 1.8.0\_40

**Windows Server 2008 – Application**

Windows Server 2008 R2 x64

Username: administrator

Password: admin

IP address: 192.168.24.51

Subnet mask: 255.255.255.0

Default Gateway: 192.168.24.1

DNS server: 192.168.230.92

**Windows Server 2008 – Workspaceserver 1**

Windows Server 2008 R2 x64

Username: administrator

Password: admin

IP address: 192.168.24.54

Subnet mask: 255.255.255.0

Default Gateway: 192.168.24.1

DNS server: 192.168.230.92

**Windows Server 2008 – Workspaceserver 2**

Windows Server 2008 R2 x64

Username: administrator

Password: admin

IP address: 192.168.24.55

Subnet mask: 255.255.255.0

Default Gateway: 192.168.24.1

DNS server: 192.168.230.92

**Windows 7 – Client**

Windows 7 Enterprise x64

IP address: 192.168.24.52

Subnet mask: 255.255.255.0

Default Gateway: 192.168.24.1

DNS server: 192.168.230.92

**Ubuntu – Database server**

Ubuntu 14.10 server

Username: administrator

Password: admin

MySql username: root

MySql password: admin

IP address: 192.168.24.53

Subnet mask: 255.255.255.0

Defaul Gateway: 192.168.24.1

DNS Server: 192.168.230.92

On server “Windows Server 2008 – Jenkins” is running Jenkins. Jenkins is connecting to SonarQube and GitHub. Jenkins check every hour for updates.

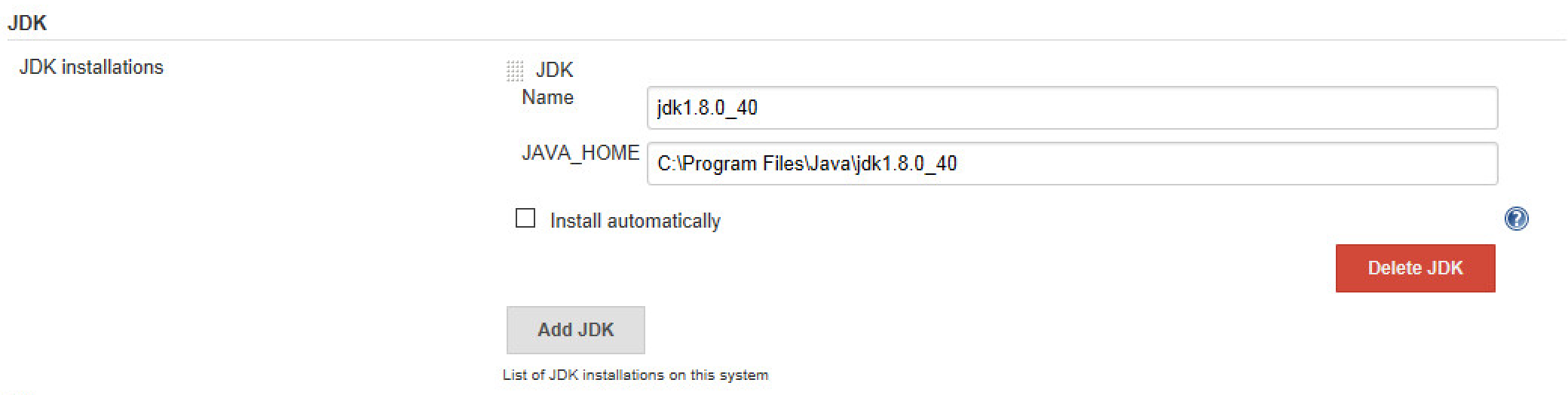
The server “Windows Server 2008 – Application” is for run the application for testing and deployment.

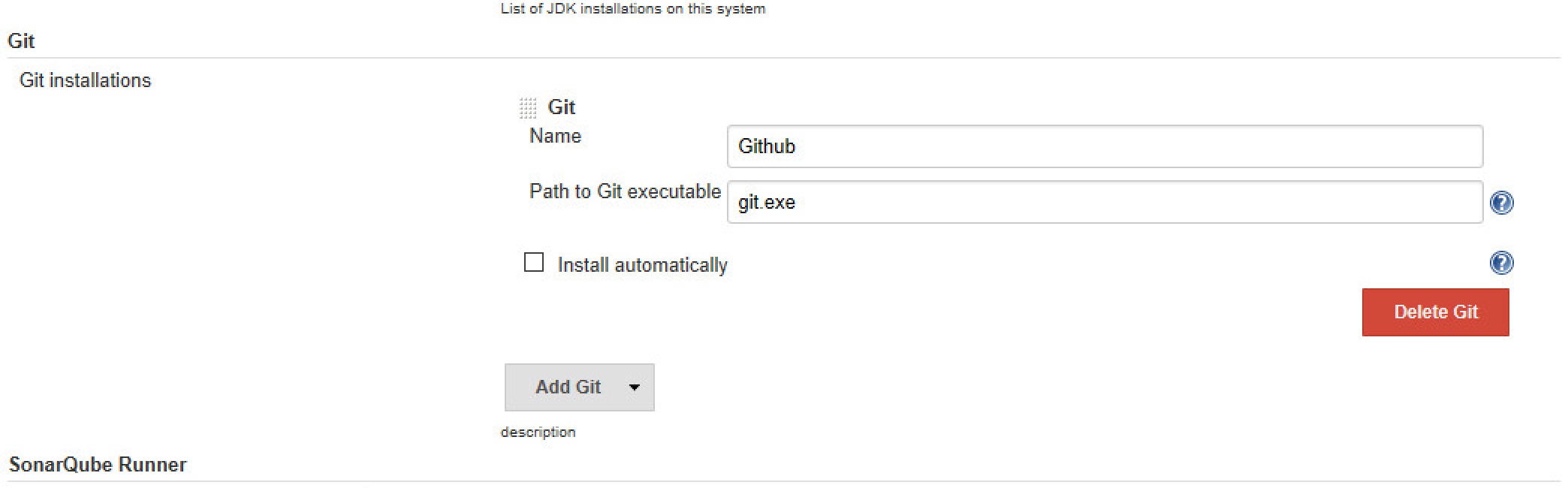
The PC “Windows 7 - Client” is for run the client side application and is for testing and deployment.

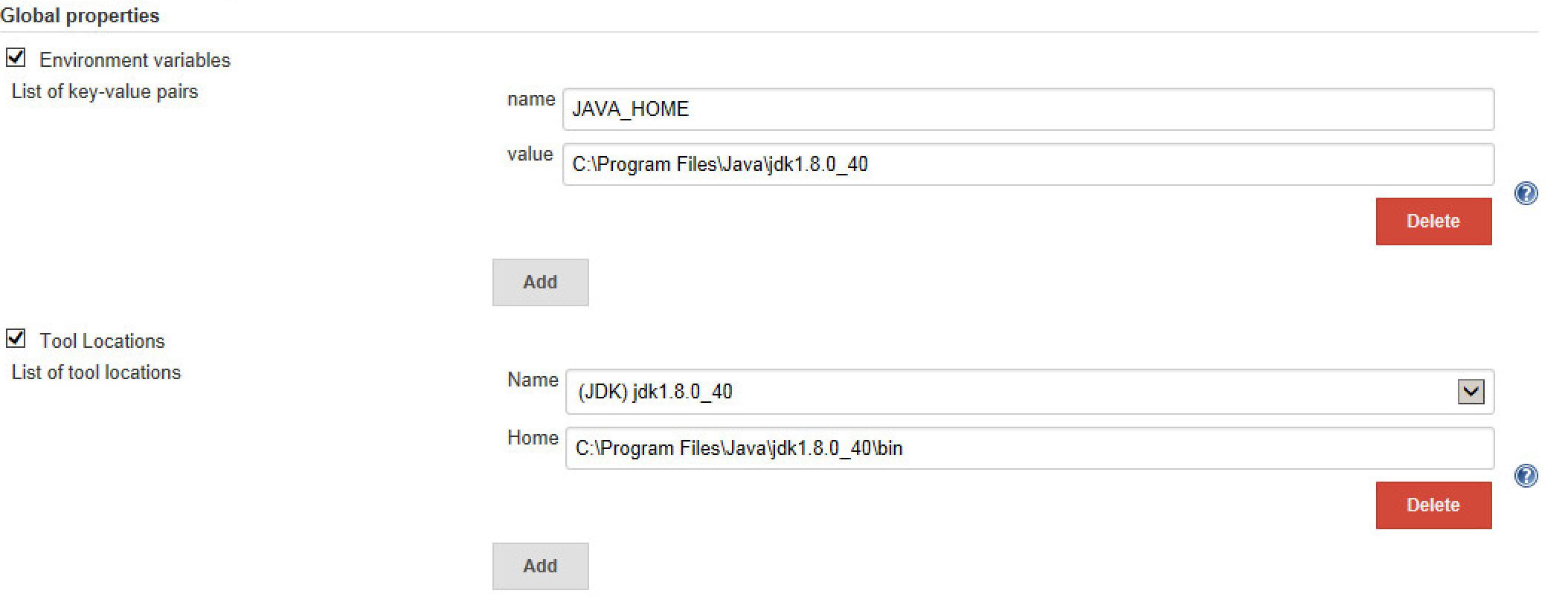
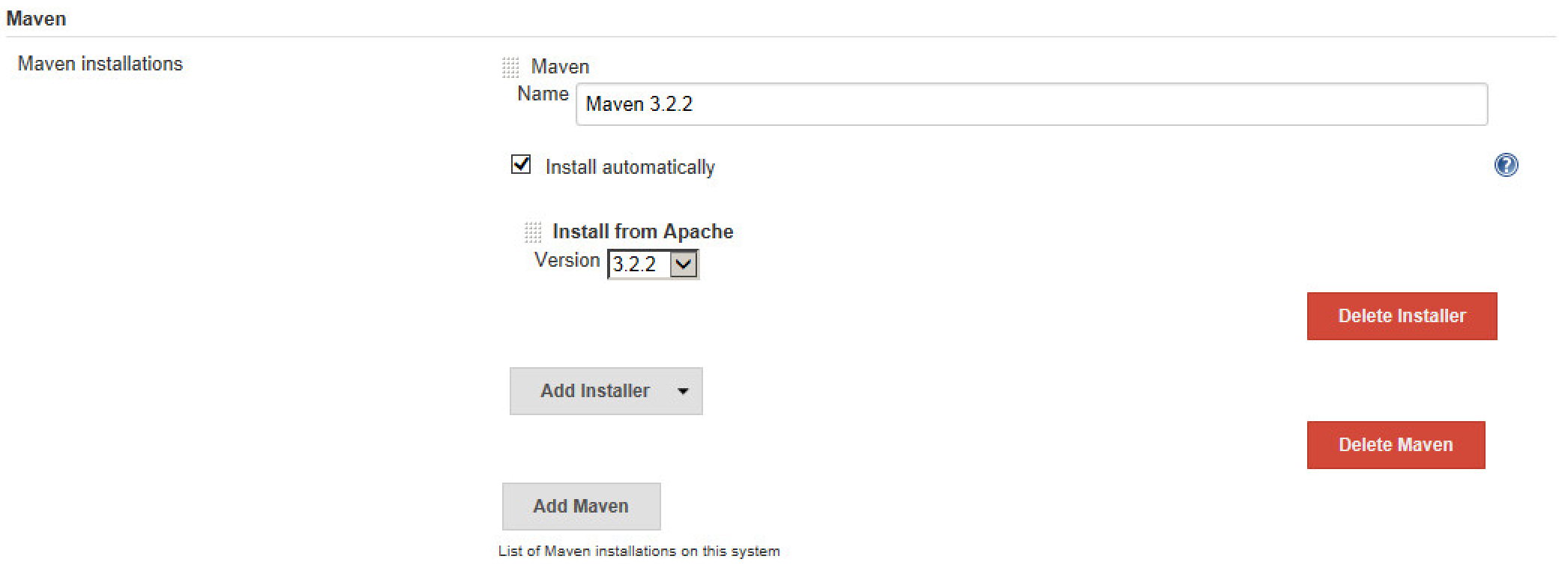
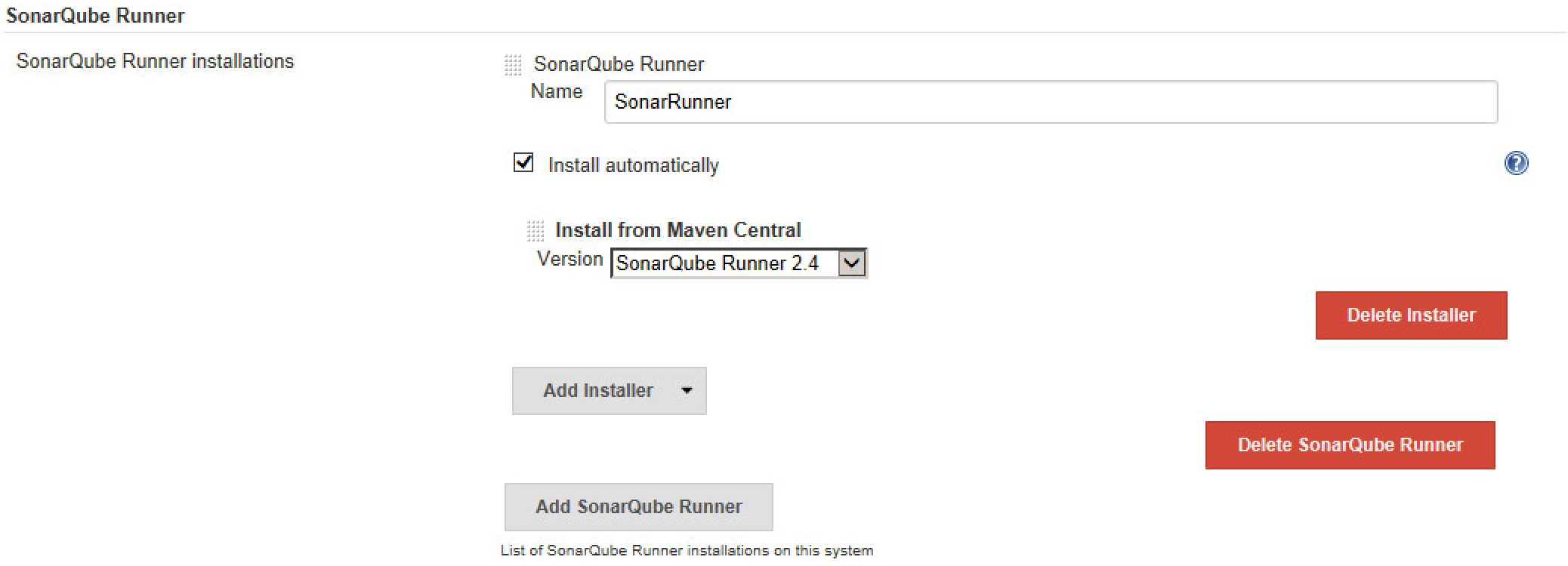
On server *“Ubuntu – Database server”* is running a MySql database.

# Configure Jenkins

1. Go to <http://localhost:8080>
2. Go to *“Manage Jenkins” 🡪* *“Manage Plugin”*
3. Install *“Git Plugin” and “SonarQube Plugin”* and restart Jenkins
4. Go now to *“Manage Jenkins” 🡪 “Configure System”*
5. Set now the follow values:

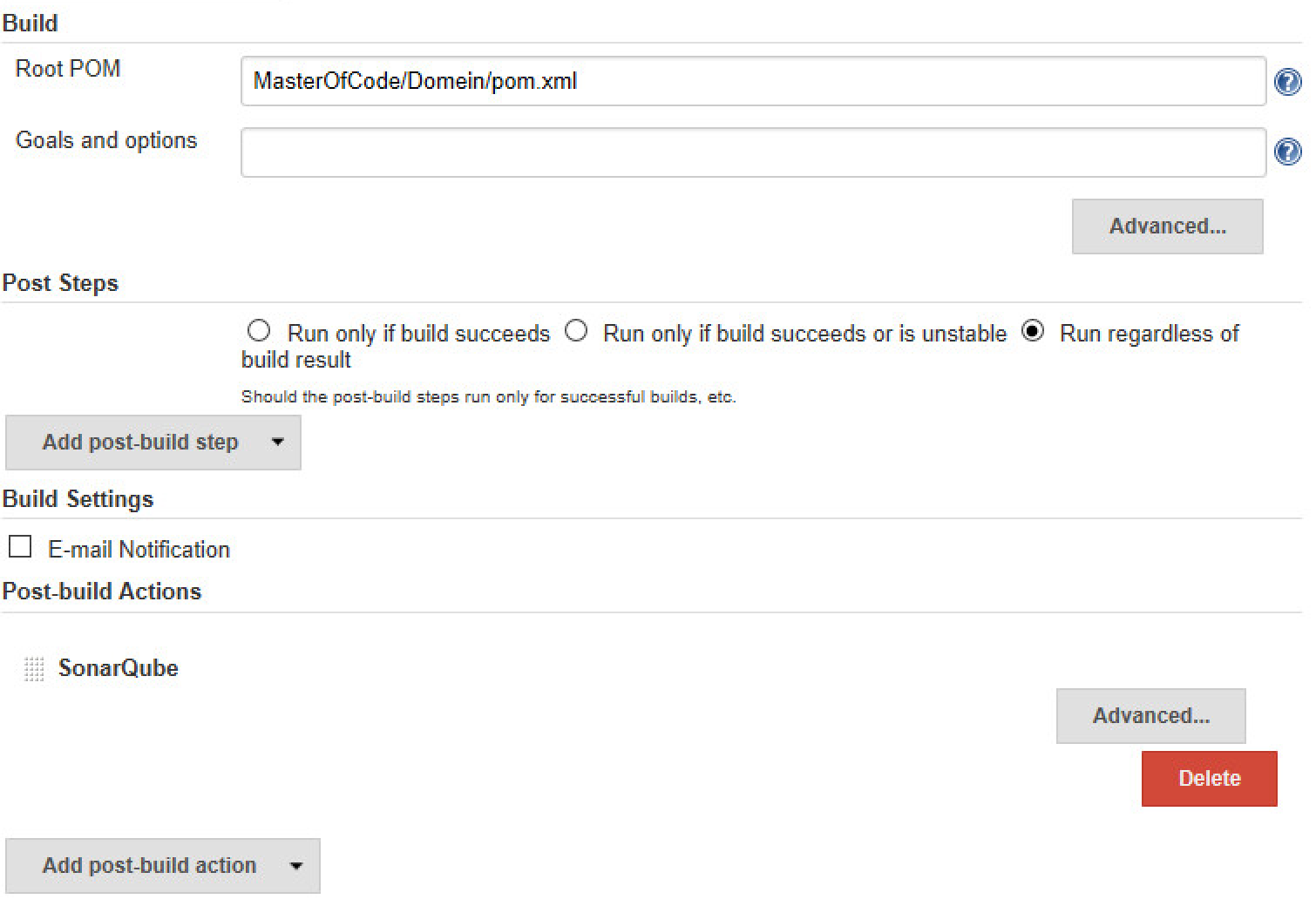
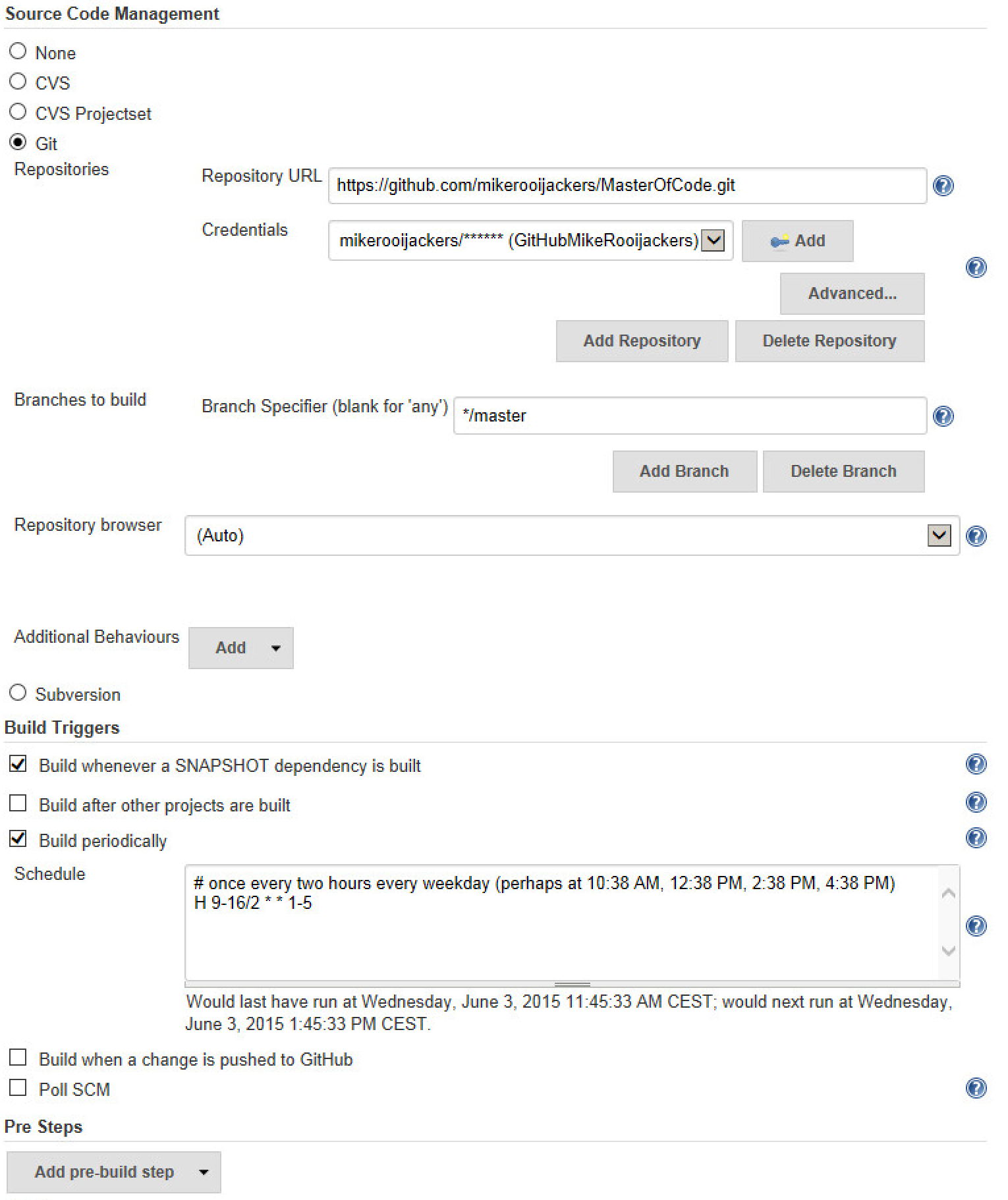






1. Save all the settings.
2. Click on *“New Item”*
3. Fill in *“Item name”*
4. Select *“Maven Project”* and click *“OK”*
5. Set the variables so as on de screenshots.

Build Periodically: The turning every weekday from 9 am to 4 pm and every 2 hours.



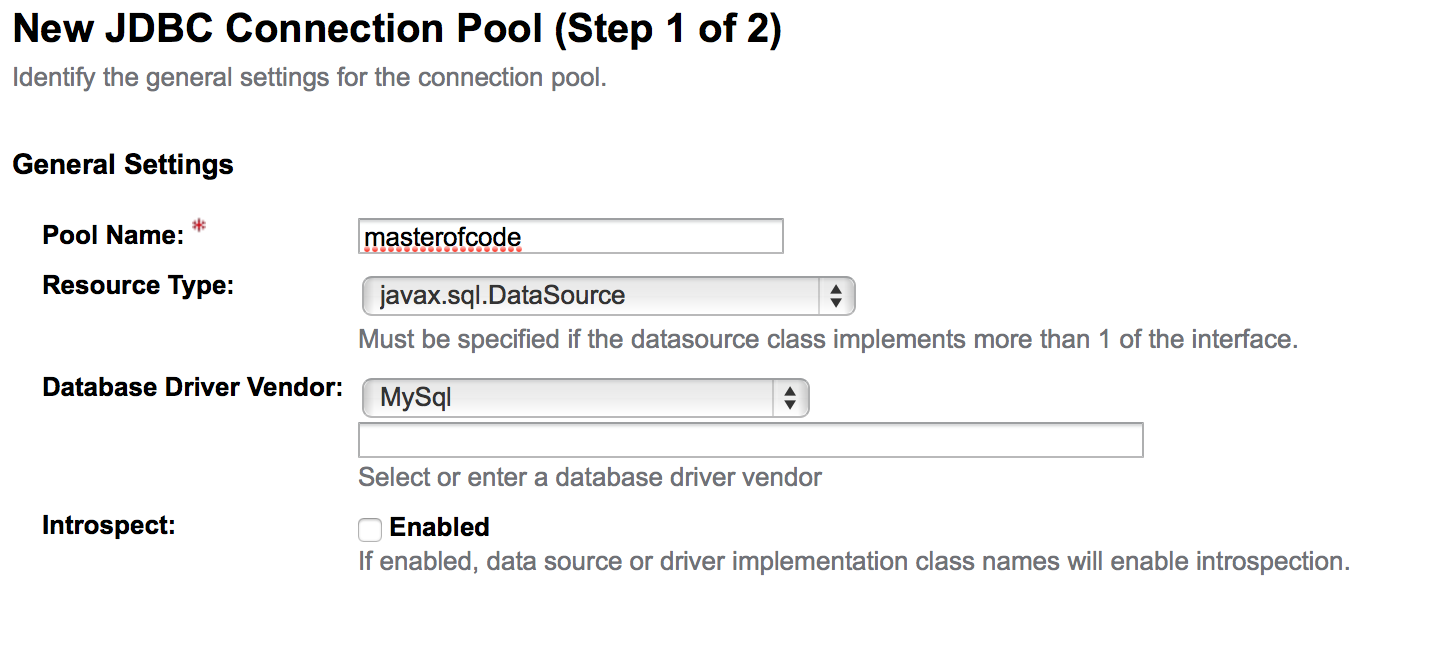
# Glassfish JDBC Configuration

1. Run glassfish
2. Go to localhost:4848
3. Go to *“Resources” 🡪 “ JDBC” 🡪 “JDBC Connection Pools”*
4. Press *“New…”*
5. Enter the following:

Pool Name: masterofcode

Resource Type: javax.sql.Datasource

Database Driver Vendor: MySql



1. Press *“next”*
2. Remove all property’s
3. Enter the following:

URL jdbc:mysql://192.168.24.53/masterofcodedb

PortNumber 3306

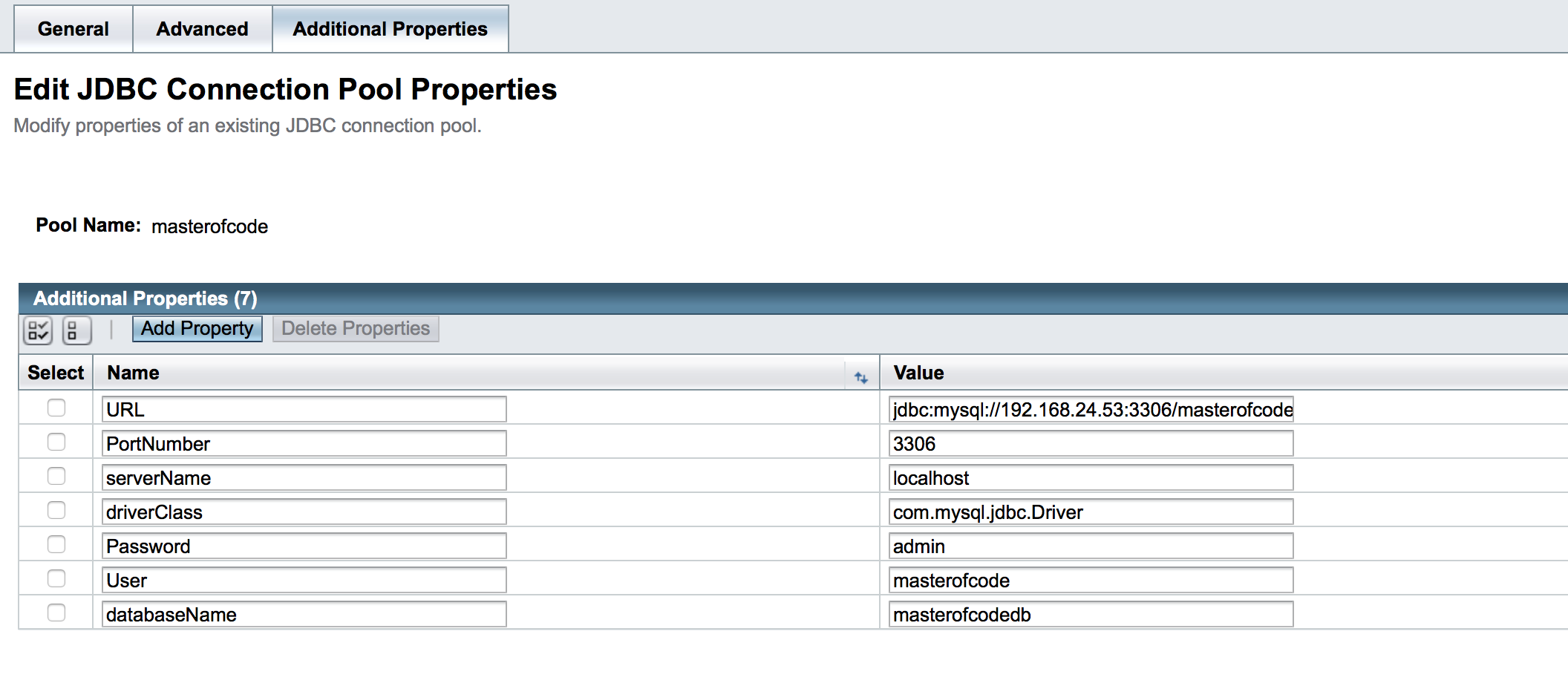
serverName localhost

driverClass com.mysql.jdbc.Driver

Password admin

User masterofcode

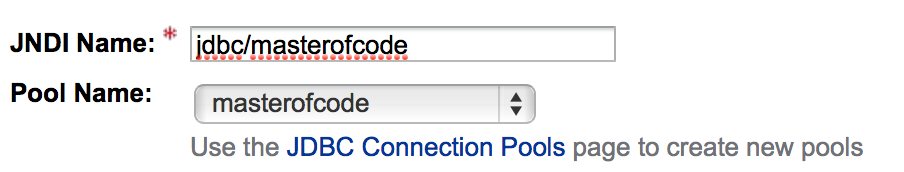
databaseName masterofcodedb



1. Press *“Finish”*
2. Go to *“JDBC Resources”*
3. Press *“New”*
4. Enter the following:

JNDI Name: jdbc/masterofcode

Pool Name: masterofcode



1. Press *“OK”* now is glassfish ready to use

# Glassfish JMS Configuration

1. Run glassfish
2. Go to localhost:4848
3. Go to *“Resources” 🡪 “ JMS Rescources” 🡪 “Connection Factories”*
4. Press *“New…”*
5. Enter the following:

JNDI Name: jms/MasterOfCodeConnectionFactory

Resource Type: javax.jms.ConnectionFactory

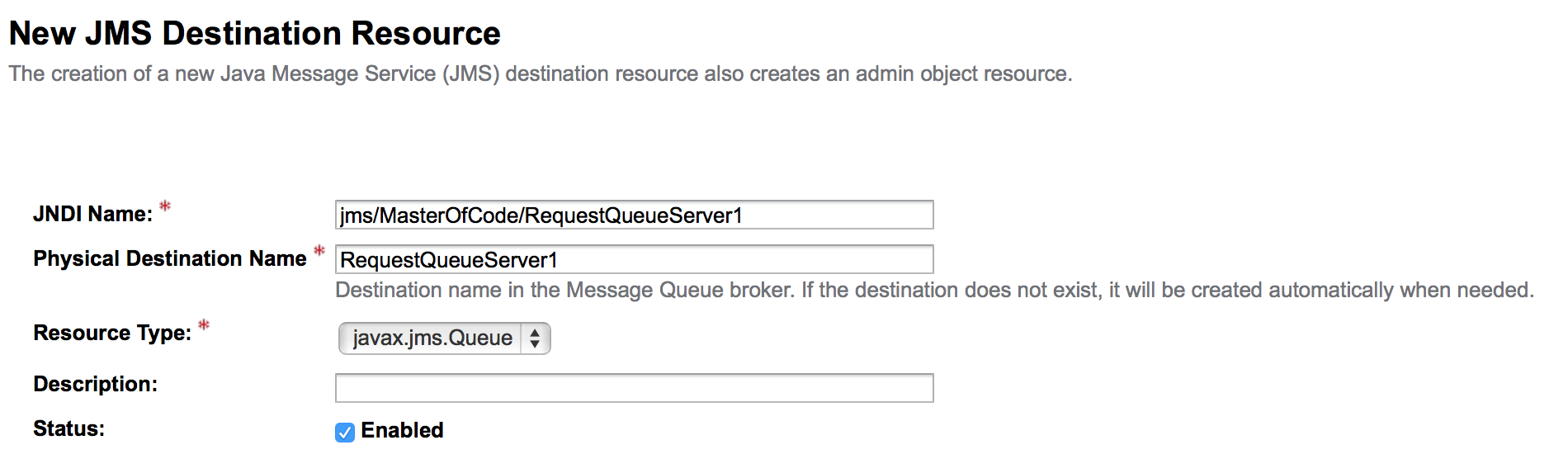


1. Press *“OK”*
2. Go to *“Resources” 🡪 “ JMS Rescources” 🡪 “Destination Resources”*
3. Press *“New…”*
4. Enter the following:

JNDI Name: jms/MasterOfCode/RequestQueueServer1

Physical Destination Name: RequestQueueServer1

Resource Type: javax.jms.Queue

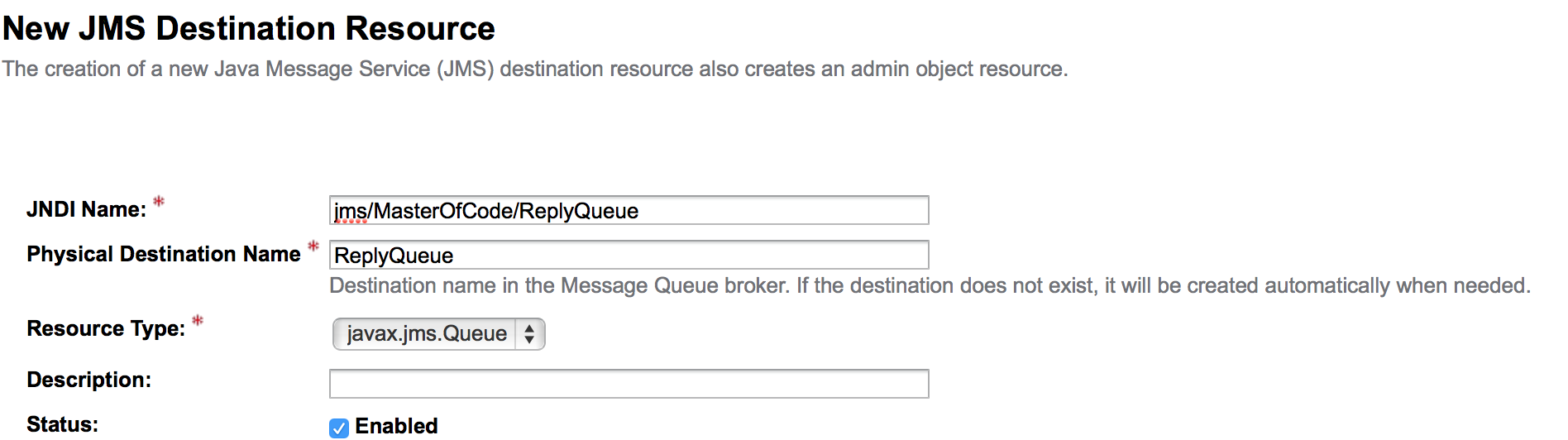


1. Press *“OK”*
2. Press *“New…”*
3. Enter the following:

JNDI Name: jms/MasterOfCode/ReplyQueue

Physical Destination Name: ReplyQueue

Resource Type: javax.jms.Queue

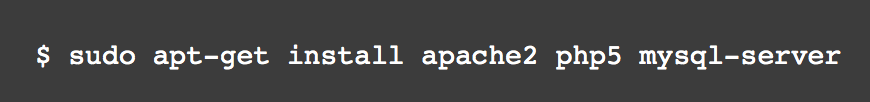


1. Press *“OK”,* JMS is now ready to use

# Install Database Server

1. Install Apache2, PHP and MySQL

We assume you already have installed LAMP setup on your system. If you do not have installed, Use following command to install it. Below command will install Apache2, PHP5 and MySQL server in your ubuntu system.



1. Install phpMyAdmin

After installing LAMP stack, lets install phpMyAdmin using below given command in your Ubuntu system



1. Configure Apache2 for phpMyAdmin

After installation of phpMyAdmin it creates a configuration file for Apache2. Edit Apache2 main configuration file /etc/apache2/apache2.conf and add following line at the end of file.

Include /etc/phpmyadmin/apache2.conf

After updating above entry, restart Apache2 service using following command

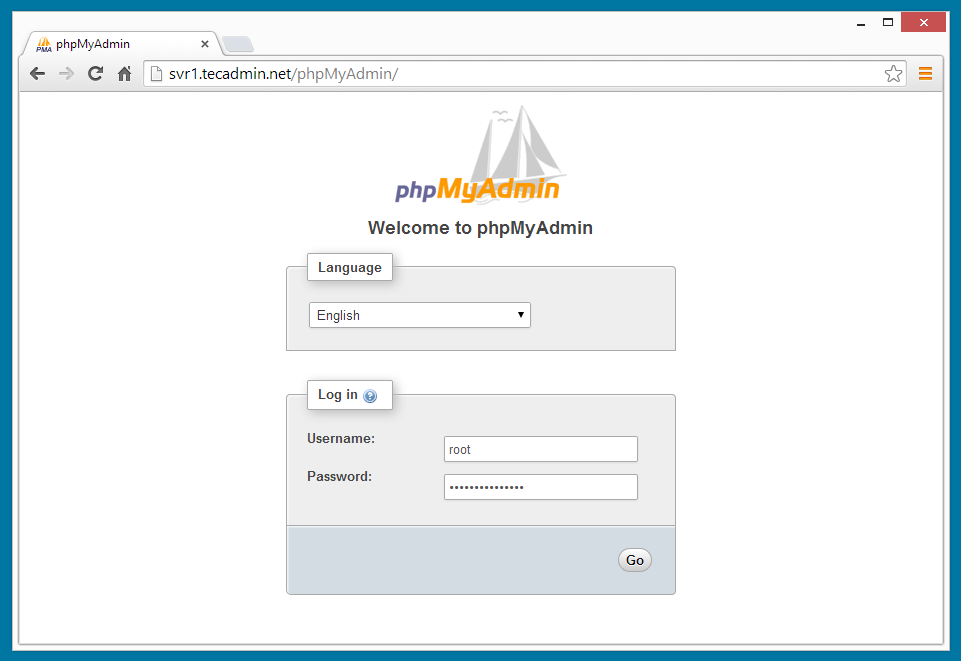


1. Access phpMyAdmin in Browser

You phpMyAdmin installation has been completed successfully. Open you favorite web browser and open below url to access phpMyAdmin

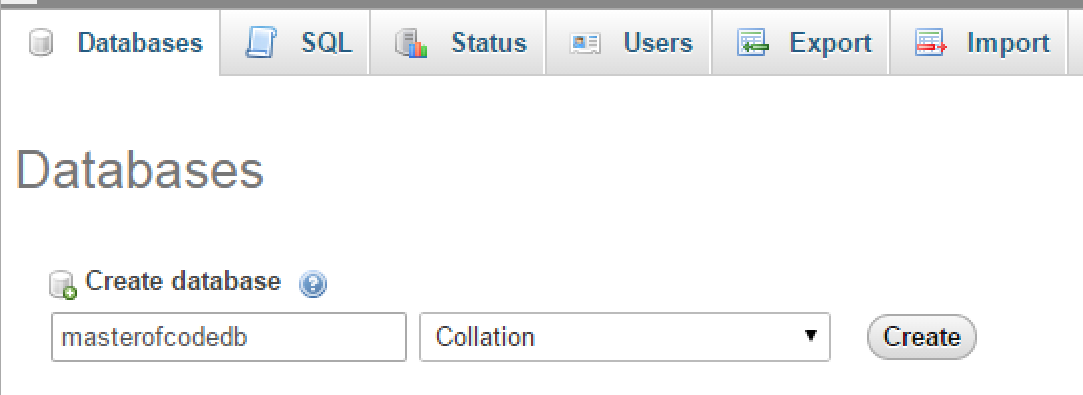
http://ip-adres/phpmyadmin/

If you are accessing from remote system, change localhost to system ip address.



1. Create a new Databases

Database: masterofcodedb



The database is now ready.

# How to use Software Development Street

Clone the github project to your local PC. Create a new remote branch. Then switch to your own branch. Here you create your own piece of code. When complete push you to your remote branch. After this is done gives you all the data on the master branch. The master branch; add together your own branch. If there are conflicts you solve this and push you all for your remote branch. Then you go through the github merge your branch to the master. Now you can delete your remote branch and create a new remote branch to create a new function.

Jenkins checks Monday to Friday from 9 am to 16h every 2 hours if the project continues to operate. If he puts all the data properly to sonar containing improvements and cleanliness of the code.

If it does not go well, he gets a red dot and we need to correct the mistakes.

It was decided to do until Friday for control on Monday between 9 am and 16 pm every 2 hours so that the projects do not overlap with the running of the test run. Also, these times are chosen because we then most work on the project.