

JIRA и MySQL 5.1 под Linux

JIRA после установки оказалась непривычно прожорливой к памяти. Я, конечно, понимаю, что это серьезный продукт и работает он на Java, но поглощение сразу 845MB памяти (в контейнере OpenVZ), показалось мне несколько неприличным. Контейнеру была выделена дополнительная память, а JIRA было решено перевести на MySQL, благо сервер установлен в том же контейнере. Расход памяти это не уменьшит, но повысит стабильность системы. Как установить последнюю версию MySQL на Debian Lenny, я [уже писал](#).

Как [перевести JIRA на MySQL](#) описано на сайте разработчиков. Есть [специфические инструкции для Linux](#), их и возьмем за основу.

Если система уже используется и в ней есть важные данные, их необходимо сохранить через [XML Backup](#). После переключения баз данных с этой копии произведем восстановление. В противном случае у нас будет абсолютно чистая система, ведь все осталось в HSQLDB.

Сначала проведем всю черновую работу и настроим работу с MySQL. Считаем, что сервер уже настроен и принимает соединения на порту 3306 на локальной машине.

Создадим базу данных и пользователя:

```
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 46
Server version: 5.1.40-0.dotdeb.1 (Debian)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database jiradb character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER,INDEX on
jiradb.* TO 'jira'@'localhost' IDENTIFIED BY 'mypassword';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit
Bye
```

Для работы с MySQL из Java понадобится [MySQL Connector/J 5.1](#) (последняя версия на момент написания). Скачаем и установим (все операции относительно корневого каталога JIRA):

```
$ wget http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.10.tar.gz/from/http://mysql.bst.lt/
$ tar -zxvf mysql-connector-java-5.1.10.tar.gz
$ mv mysql-connector-java-5.1.10/mysql-connector-java-5.1.10-bin.jar common/lib
```

Необходимо внести изменения в файл conf/server.xml (резервную копию создадим как server.xml.bak). Изменения примерно такие:

```
$ diff -u conf/server.xml.bak conf/server.xml
--- server.xml.bak      2009-11-26 11:03:01.000000000 +0200
+++ server.xml          2009-11-26 12:02:29.000000000 +0200
@@ -11,14 +11,14 @@
- username="sa"
- password=""
- driverClassName="org.hsqldb.jdbcDriver"
- url="jdbc:hsqldb:${catalina.home}/database/jiradb"
- minEvictableIdleTimeMillis="4000"
- timeBetweenEvictionRunsMillis="5000"
+ username="jira"
+ password="mypassword"
+ driverClassName="com.mysql.jdbc.Driver"
+ url="jdbc:mysql://localhost/jiradb?autoReconnect=true&useUnicode=true&characterEncoding=UTF8"
maxActive="20" />
+ validationQuery="Select 1"
```

Также, необходимо отредактировать atlassian-jira/WEB-INF/classes/entityengine.xml (сохранив копию). Изменения:

```
$ diff -u atlassian-jira/WEB-INF/classes/entityengine.xml.bak atlassian-jira/WEB-INF/classes/entityengine.xml
--- entityengine.xml.bak      2009-11-26 11:22:07.000000000 +0200
+++ entityengine.xml          2009-11-26 12:12:44.000000000 +0200
@@ -97,8 +97,7 @@
- <datasource name="defaultDS" field-type-name="hsqldb"
- schema-name="PUBLIC"
+ <datasource name="defaultDS" field-type-name="mysql"
helper-class="org.ofbiz.core.entity.GenericHelperDAO"
check-on-start="true"
use-foreign-keys="false"
```

После этого можно запускать JIRA:

```
$ bin/startup.sh
```

Если все сделано правильно, система поднимется и сама создаст структуру базы данных. Далее необходимо восстановить данные с XML (<http://localhost:8080/secure/SetupImport!default.jspa>):

http://rat.admin.lv/wp-content/uploads/2009/11/data_restore.png

http://rat.admin.lv/wp-content/uploads/2009/11/restore_complete.png

Все, дальше используется более привычная система MySQL, которая легко интегрируется в отлаженный процесс создания резервных копий данных.

Оригинал статьи: <http://rat.admin.lv/?p=348>