# Table of Contents

jBPM Clustered deployment

Primary purpose of this document is to set up a clustered jBPM environment.

Cluster Architecture

Usually, a BPM system mainly contain 3 components:

1. Design Time Tools - Used to design process, a version control and a process persist tool are necessary, some BPM system use RDBMS to do this, but jBPM use git for version control, maven for persist final process, this also for compatible with recent DevOps, Cloud Solution, git is well demonstrate by Github for version control, maven are great for distributing archive across servers/machines.

2. Runtime Engine - A business process is a abstraction of business logic and involve business people via Human based Task, some time the business logic need roll back or re-orchestrate, that means great number of database interations are necessary.

3. RDBMS - Some BPM system use store procedure to operated Database, but jBPM use Hibernate with native sql

After the overview of the basic architecture of jBPM, below figure is for how to cluster jBPM.
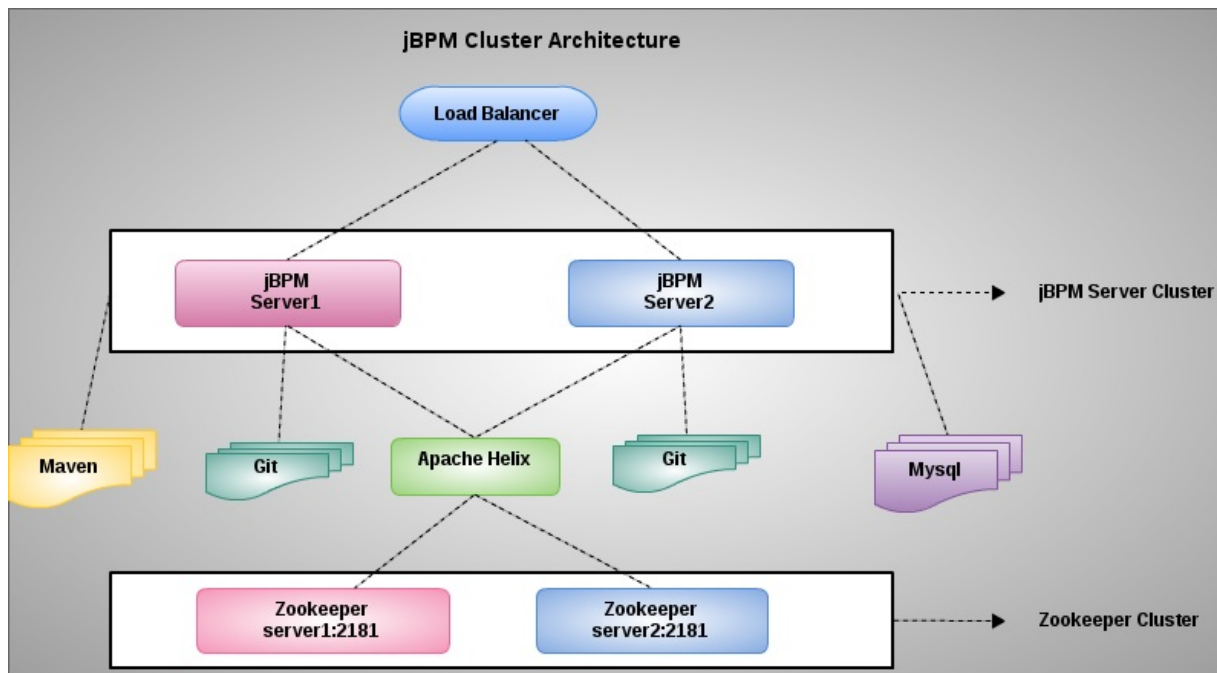
- Server1 and Server2 are two physical linux server

- Both servers point to a shred Mysql database

- Both servers point to a shred Maven repository

- Both servers point to a shred OpenLDAP server

- Apache Zookeeper and Helix used to replicate assets(process, data modules) between 2 git repositories

- Quartz Job Scheduler used to run complexed timer related process

- Apache httpd with mod_proxy_balancer used as a load balancer

Overvew

This section contain how to download and install jBPM 6.5 version.

Table of Contents

Download

Download jBPM 6.5(jbpm-6.5.0.Final-installer-full.zip) from http://jbpm.org/download/download.html

Installaton

Note	This section depend on above Download section.

0 - Unzip jBPM full zip

```
unzip jbpm-6.5.0.Final-installer-full.zip
```

Note	A `jbpm-installer` folder will be generated, the following installation steps depend on this folder.

1 - Install WildFly

- Extract the zip file to install WildFly

```
unzip jbpm-installer/lib/jboss-wildfly-10.0.0.Final.zip
```

- Create placeholder directories

Change into WildFly Home, create 2 directories: `clustering` and `installation` :

```
cd wildfly-10.0.0.Final/
mkdir clustering
mkdir installation
```

Note	the `clustering` and `installation` directory are used across the whole installation, including clustering, test, etc. **clustering** used to keep cluster related software, scripts. **installation** used to keep other cli/shell scripts, 3rd party libaries, etc.

- Copy all cli/shell scripts to WFY_HOME/installation

```
cp *.cli ~/wildfly-10.0.0.Final/installation
cp *.sh ~/wildfly-10.0.0.Final/installation
```

2 - Install jBPM console

Unzip jbpm console war to WildFly deployment folder and add a dodeploy file:

```
unzip jbpm-installer/lib/jbpm-console-6.5.0.Final-wildfly-10.0.0.Final.war -d ./wildfly-10.0.0.Final/standalone/deployments/jbpm-
touch ./wildfly-10.0.0.Final/standalone/deployments/jbpm-console.war.dodeploy
```

3 - Install kie Server

Unzip kie server war to WildFly deployment folder and add a dodeploy file:

```
unzip jbpm-installer/lib/kie-server-6.5.0.Final-wildfly-10.0.0.Final.war -d ./wildfly-10.0.0.Final/standalone/deployments/kie-ser
touch ./wildfly-10.0.0.Final/standalone/deployments/kie-server.war.dodeploy
```

4. Set up user group

Change into WildFly Home, execute add users shell script add-users.sh:

```
./installation/add-users.sh
```

Alternatively, execute below commands under WildFly Home

```
./bin/add-user.sh -a -u admin -p password1! -g admin,analyst,kiemgmt,rest-all,kie-server
./bin/add-user.sh -a -u krisv -p password1! -g admin,analyst,rest-all,kie-server
./bin/add-user.sh -a -u john -p password1! -g analyst,Accounting,PM
./bin/add-user.sh -a -u mary -p password1! -g analyst,HR
./bin/add-user.sh -a -u sales-rep -p password1! -g analyst,sales
./bin/add-user.sh -a -u jack -p password1! -g analyst,IT
./bin/add-user.sh -a -u katy -p password1! -g analyst,HR
./bin/add-user.sh -a -u salaboy -p password1! -g admin,analyst,IT,HR,Accounting,rest-all
./bin/add-user.sh -a -u kieserver -p password1! -g kie-server
```

Note	These users are used for test and use simple plain txt based jaas login module, if use OpenLDAP server, this step is redundant

5. Installation Validation

Change into Wildfly Home, edit `bin/standalone.conf` , make sure WildFly/jBPM server has enough memory

```
-Xms2048m -Xmx2048m -XX:MetaspaceSize=256M -XX:MaxMetaspaceSize=512m
```

add system properties to disable import example data

```
-Dorg.kie.demo=false -Dorg.kie.example=false
```

start WildFly/jBPM via

```
./bin/standalone.sh -b 0.0.0.0 -bmanagement=0.0.0.0 -c standalone-full.xml
```

Once start finished, access http://localhost:8080/jbpm-console will log into jBPM console with `admin / admin` .

Alternative Options

Download/Install Production BPM Suite is a alternative. Refer to Download and Installation BPMS 6 for details.

What's it

jBPM default configured to use an example h2 data source, this not suitable for production. This section including setps of switch to mysql from default h2.

Table of Contents

Install Mysql on Linux and create jbpm database

- Install Mysql on Linux via:

```
yum install mysql
```

- Start Mysql in Linux via:

```
service mysqld start
```

- Log into mysql with root user, create database, user and grant privileges

```
CREATE DATABASE jbpm;
CREATE USER 'jbpm_user'@'%' IDENTIFIED BY 'jbpm_pass';
GRANT ALL PRIVILEGES ON jbpm.* TO 'jbpm_user'@'%';
```

- Log into mysql with jbpm_user import the DDL script to jbpm database

```
mysql -u jbpm_user -p jbpm < ~/work/jbpm/jbpm-installer/db/ddl-scripts/mysql5/mysql5-jbpm-schema.sql
mysql -u jbpm_user -p jbpm < ~/work/jbpm/jbpm-installer/db/ddl-scripts/mysql5/quartz_tables_mysql.sql
```

Set up Mysql Data Source

- Download mysql driver (http://dev.mysql.com/downloads/connector/) to WFY_HOME/installation, assume `mysql-connector-java-5.1.38.jar` be downloaded.
- Add Mysql Driver as a Module

Assume module-add-mysql.cli already be copy to WFY_HOME/installation, make sure WildFly Server is running and execute:

```
./bin/jboss-cli.sh --connect --file=installation/module-add-mysql.cli
```

- Create Mysql Data Source

Assume create-mysql-ds.cli already be copy to WFY_HOME/installation, make sure WildFly Server is running and execute:

```
./bin/jboss-cli.sh --connect --file=installation/create-mysql-ds.cli
```

Note	If above cli execute success, you will find the output like `"result" ⇒ [true]` , if failed, please check the database name, user, password, etc.

Amend Data Source

If want to chenge datasource url, log into CLI console, execute

```
./bin/jboss-cli.sh --connect
/subsystem=datasources/data-source=MysqlDS:write-attribute(name=connection-url,value="jdbc:mysql://191.168.1.101:3306/jbpm")
/subsystem=datasources/data-source=quartzNotManagedDS:write-attribute(name=connection-url,value="jdbc:mysql://191.168.1.101:3306/
```

Switch from h2 to mysql

Make sure WildFly Server is shut dwon, navigate to WildFly Home, Edit `standalone/deployments/jbpm-console.war/WEB-INF/classes/META-INF/persistence.xml` ,

- Locate the <jta-data-source> tag and change it to the JNDI name of your data source, for example:

```
<jta-data-source>java:jboss/datasources/MysqlDS</jta-data-source>
```

- Locate the <properties> tag and change the hibernate.dialect property, for example:

```
<property name="hibernate.dialect" value="org.hibernate.dialect.MySQL5Dialect" />
```

- Locate the hibernate.hbm2ddl.auto, chenge the update to none:

```
<property name="hibernate.hbm2ddl.auto" value="none" />
```

- Assume switch-kie-server.cli already be copy to WFY_HOME/installation, make sure WildFLy Server is running and execute:

```
./bin/jboss-cli.sh --connect --file=installation/switch-kie-server.cli
```

- Restart WildFly Server, to make sure the switch work fine.

Switch to LDAP

This section including how to configure jbpm-console to use LDAP for authentication and authorization of users.

- Change into WildFly Home, execute create-security-domain-ldap.cli cli:

```
./bin/jboss-cli.sh --connect --file=installation/create-security-domain-ldap.cli
```

- Edit `jbpm-console.war/WEB-INF/jboss-web.xml` , change security domain to `jbpm-cluster`

```xml
<jboss-web>
    <security-domain>jbpm-cluster</security-domain>
</jboss-web>
```

Note	`admin,analyst,reviewer` are necessary in LDAP.

Design-Time Clustering

Design-time clustering makes use of Apache Helix and Apache ZooKeeper, to share assets(processes, rules, data model objects) in the Git repository between all cluster nodes. This section including contents of how to set up Design-Time Clustering

What's this?

This section including steps to download and setting a clustering.

Table of Contents

Download

Change into WFY_HOME/clustering directory

- Download `zookeeper-3.3.6.tar.gz` from http://apache.fayea.com/zookeeper/zookeeper-3.3.6/

```
wget http://apache.fayea.com/zookeeper/zookeeper-3.3.6/zookeeper-3.3.6.tar.gz
```

- Download `helix-core-0.6.6-pkg.tar` from http://helix.apache.org/0.6.6-docs/download.cgi

Installation

- Change into WFY_HOME/clustering directory.

Note the `WFY_HOME/clustering` used to keep zookeeper/helix for each servers.

- Extract `zookeeper-3.3.6.tar.gz` to WFY_HOME/clustering

```
tar -xvf zookeeper-3.3.6.tar.gz
```

- Extract `helix-core-0.6.6-pkg.tar` to WFY_HOME/clustering

```
tar -xvf helix-core-0.6.6-pkg.tar -C ~/wildfly-10.0.0.Final/clustering/
```

Configure Apache ZooKeeper

- Change into WFY_HOME/clustering/zookeeper/conf, create a `zoo.cfg` from sample

```
cp zoo_sample.cfg zoo.cfg
```

- Edit `zoo.cfg` , add the following content

```
# Defining ZooKeeper ensemble.
server.1=localhost:2888:3888
server.2=localhost:2889:3889
```

- Assign a node ID to each member that will run ZooKeeper. For example, use 1, 2 for node 1, node 2 respectively. The ZooKeeper node ID is specified in a field called myid under the data directory of ZooKeeper on each node. For example, on node 1, execute:

```
mkdir /tmp/zookeeper
echo "1" > /tmp/zookeeper/myid
```

- Start Zookeeper

Change into zookeeper home execute

```
./bin/zkServer.sh start
```

Configure Apache Helix

- Change into WFY_HOME/clustering/helix-core, create the cluster

```
./bin/helix-admin.sh --zkSvr server.1:2181,server.2:2181 --addCluster jbpm-cluster
```

- Add nodes to the cluster

```
./bin/helix-admin.sh --zkSvr server.1:2181,server.2:2181 --addNode jbpm-cluster node1:12345
./bin/helix-admin.sh --zkSvr server.1:2181,server.2:2181 --addNode jbpm-cluster node2:12346
```

- Add resources to the cluster.

```
./bin/helix-admin.sh --zkSvr server.1:2181,server.2:2181 --addResource jbpm-cluster vfs-repo 1 LeaderStandby AUTO_REBALANCE
```

- Rebalance the cluster with the three nodes.

```
./bin/helix-admin.sh --zkSvr server.1:2181,server.2:2181 --rebalance jbpm-cluster vfs-repo 1
./bin/helix-admin.sh --zkSvr server.1:2181,server.2:2181 --rebalance jbpm-cluster vfs-repo 2
```

- Start the Helix controller in all the nodes in the cluster.

```
./bin/run-helix-controller.sh --zkSvr server.1:2181,server.2:2181 --cluster jbpm-cluster 2>&1 > ./controller.log &
```

Confige WildFly/jBPM Server

Configure individual server nodes with a series if System properties to point to Apache Helix controller.

Confige Server 1

- Create the following directories on Server 1

```
mkdir -p /tmp/jbpm/node1
mkdir -p /tmp/jbpm/quartz
```

- Copy design-time-cluster-server-1.cli to WFY_HOME/installation, make sure WildFLy Server is running and execute:

```
./bin/jboss-cli.sh --connect --file=installation/design-time-cluster-server-1.cli
```

- If want to amend s specific property, log into CLI console, ececute

```
./bin/jboss-cli.sh --connect
/system-property=org.uberfire.cluster.zk:write-attribute(name=value,value="10.66.192.120:2181,10.66.192.121:2181")
```

Confige Server 2

- Create the following directories on Server 2

```
mkdir -p /tmp/jbpm/node2
mkdir -p /tmp/jbpm/quartz
```

- Copy design-time-cluster-server-2.cli to WFY_HOME/installation, make sure WildFLy Server is running and execute:

```
./bin/jboss-cli.sh --connect --file=installation/design-time-cluster-server-2.cli
```

- If want to amend s specific property, log into CLI console, ececute

```
./bin/jboss-cli.sh --connect
/system-property=org.uberfire.cluster.zk:write-attribute(name=value,value="10.66.192.120:2181,10.66.192.121:2181")
```

Run-Time Clustering

The runtime clustering including setup Quartz Enterprise Job Scheduler, kie-server cluster, dashbuilder cluser, etc.

Configure Quartz

Configure Quartz is necessary to make sure jBPM cluster works fine.

- Prepare data source

This step are finished in Set up data sources section, `quartzNotManagedD` should be used in the following configuration.

- Create Quartz tables on `quartzNotManagedD` referred database.

```
cp jbpm-installer/db/ddl-scripts/mysql5/quartz_tables_mysql.sql wildfly-10.0.0.Final/installation/
mysql -u jbpm_user -p jbpm < wildfly-10.0.0.Final/installation/quartz_tables_mysql.sql
```

- Create the Quartz configuration file quartz-definition-mysql.properties

Copy it to `/tmp/jbpm/quartz` which compatible with `org.quartz.properties` property definition in Configure WildFly/jBPM.

Configuring Load Balancer

This section use Apache `httpd` with `mod_proxy_balancer` as a Load Balancer, stick session are enabled.

- httpd install

```
yum -y install httpd
```

- httpd configuration

Copy jbpm-cluster.conf to `/etc/httpd/conf.d`

- httpd start

```
systemctl start httpd.service
```

Once httpd started, http://localhost:8080/jbpm-console/ can use to log into jbpm console.

Configuring Load Balancer

Start and Stop Cluster

Starting a Cluster

Start cluster with the following order:

- Start ZooKeeper servers

```
./bin/zkServer.sh start
```

Note	Two nodes need to start accordingly.

- Start Helix Controller

```
./bin/run-helix-controller.sh --zkSvr server.1:2181,server.2:2181 --cluster jbpm-cluster 2>&1 > ./controller.log &
```

- Start WilFly/jBPM Server

```
./bin/standalone.sh -b 0.0.0.0 -bmanagement=0.0.0.0 -c standalone-full.xml
```

Note	Two servera need to start accordingly.

Stopping a Cluster

To stop your cluster, stop the components in the reversed order from starting it:

- Stop WilFly/jBPM Server

Either use the stop shell scripts, or Ctrl + c.

Note	Two servers need to stop accordingly.

- Stop Helix Controller

```
ps aux|grep HelixControllerMain
kill -15
```

- Stop ZooKeeper server(s).

```
./bin/zkServer.sh stop
```

Note	Two nodes need to stop accordingly.