Monitoring external jobs

Plugin Information
View External Monitor Job Type on the plugin site for more information.

Adds the ability to monitor the result of externally executed jobs.

Jenkins is useful for monitoring the non-interactive execution of processes, such as cron jobs, procmail, inetd-launched processes. Often those tasks are completely unmonitored (which makes it hard for you to notice when things go wrong), or they send e-mails constantly regardless of the success or failure (which results into the same situation as you'll quickly start ignoring them anyway.) Using Jenkins enables you to monitor a large number of such tasks with little overhead.

Setting up a project

Create a new job and choose "Monitor an external job" as the job type.

Monitoring an execution

For Debian / Ubuntu:

```
sudo apt-get install jenkins-external-job-monitor
# obviously you should replace localhost with a FQDN if you want to run jobs from other machines.
export JENKINS_HOME=http://@localhost:8080/
java -jar /usr/share/jenkins/external-job-monitor/java/jenkins-core-*.jar "external-build-job-name" command-to-run
java -jar /usr/share/jenkins/external-job-monitor/java/jenkins-core-*.jar "external-build-test" ls -l
```

Once you set up a project, you can monitor an execution by running a command like this:

```
$ export JENKINS_HOME=http://user:pw@myserver.acme.org/path/to/jenkins/
$ java -jar /path/to/WEB-INF/lib/jenkins-core-*.jar "job name" <program arg1 arg2...>
```

For Windows:

```
> set JENKINS_HOME=http://user:pw@myserver.acme.org/path/to/jenkins/
> java -jar \path\to\WEB-INF\lib\jenkins-core-*.jar "job name" cmd.exe /c <program arg1 arg2...>
```

If your webserver extracts the `jenkins.war` file when it deploys Jenkins then you may use the path directly to the `WEB-INF/lib` directory and all other required jars will be found there. Otherwise you may extract these from the war file:

```
jenkins-core-*.jar
remoting-*.jar
ant-1.7.0.jar
commons-io-1.4jar
commons-lang-2.4.jar
jna-posix-*.jar
xstream-*.jar
```

All are found in the `WEB-INF/lib` path inside the war file. As long as they are all in the same directory, the `java -jar /path/to/jenkins-core-*.jar` command will find the other required jars.

- Note: Older versions of Jenkins (before 1.324) also require `winstone.jar` in order to run this command. This jar is found at the top level directory inside the war file, and must be manually added to the classpath (with `-classpath` or `-cp`) in the java command.

The JENKINS_HOME variable is used to locate the server Jenkins is running, so this must be set. Unless your Jenkins job has build permission for guest users, include the `username:password@` portion of the URL, as seen in the examples above.

- Note: Authentication via username:password in JENKINS_HOME is added in Jenkins 1.324; with previous versions either grant anonymous build permission on the job, or use `curl` to post XML to Jenkins (see below).

You can copy `jenkins-core-*.jar` and the other required jars to other machines if you want to monitor jobs that are run on a different machine.

stdout and stderr of the program will be recorded, and a non-zero exit code will be considered as a failure.

Monitoring cron jobs

To monitor a cron job, simply run the above set up from your cron script. To avoid receiving e-mails from cron daemon, you might want to write something like:

```
JENKINS_HOME=http://myserver.acme.org/path/to/jenkins/
0 * * * *     export JENKINS_HOME=$JENKINS_HOME; java -jar jenkins-core-*.jar "backup" backup.sh 2>&1 > /dev
/null
```

Note that you can also move the cron job itself to Jenkins by using free-style software project, which would also allow you to manually execute the job outside the scheduled executions.

Submit a run programatically

The above command submits the execution and its result by sending XML to HTTP. This means you can submit an execution record from any program, as long as you follow the same XML format.

The format is explained below:

```
<run>
  <log encoding="hexBinary">...hex binary encoded console output...</log>
  <result>... integer indicating the error code. 0 is success and everything else is failure</result>
  <duration>... milliseconds it took to execute this run ...</duration>
  <displayName>... The name to be displayed rather than the build number ...</displayName>
  <description>... Description of the build ...</description>
</run>
```

The `duration` element is optional, since 1.429 `displayName` and `description` can also optionally be appended, these three elements can be in any order, but must appear after log and result in that order. Console output is hexBinary encoded so that you can pass in any control characters that are otherwise disallowed in XML.

The above XML needs to be sent to `http://myhost/jenkins/job/_jobName_/postBuildResult`.

A simple example using curl would be (using no real data; the sequence `4142430A` encodes a console output of ABC plus a trailing line feed symbol):

```
$ curl -X POST -d '<run><log encoding="hexBinary">4142430A</log><result>0</result><duration>2000</duration><
/run>' \
http://user:pass@myhost/jenkins/job/_jobName_/postBuildResult
```

Submit a run per CLI

The easiest option is for the execution can be submitted per CLI/ssh command. The gzipped log file can be transported through pipe:

```
$ cat result.log.gz | ssh jenkins set-external-build-result --display 7d552c4ba_Linux_tb21 --job buildbot --
result 1 -b --duration 42 --log -
```

Sometimes build number is needed, as the calling program might need to put an URL to the submitted build. The CLI command above returns the new build number. The command above can be called programatically from an Java application. You can find a sample for it here.

Changelog

Version 1.7 (Jan 06, 2017)

- Allowing non-administrators to configure or delete jobs, given appropriate permissions.
- Japanese localization update.

Version 1.6 (July 26, 2016)

- JENKINS-36875 Empty screen when creating new jobs
- Follow up of JENKINS-35284

Version 1.5 (July 12, 2016)

- JENKINS-35284 Updated to the new Parent POM and fixed findbugs warnings.
- JENKINS-31162 Item categorization.

Version 1.4 (Nov 18, 2014)

- Updated item label to match new noun-oriented convention.

Version 1.3 (Nov 14, 2014)

- Refined where Build permission was checked to cover also the CLI command.

Version 1.2 (Aug 29, 2013)

- new CLI command (pull #5)
- JENKINS-19377 workaround for log rotation failure
- NPE fix
- translations