Automate Jenkins with the CLI or the REST API

Last updated: 10-Oct-2015

Jenkins is used a lot to automate your build process and make it a real continu
step further and automate the configuration of Jenkins itself.
In this short instruction I will show two ways to do this: the CLI and the REST A
write programs to create, backup, restore, start and view Jenkins jobs.

Download the Try-it-out-yourself code to provision an Ubuntu VM with Jenkins
examples below.
Download

The CLI
The Jenkins CLI is distributed inside the jenkins.war, but you have to download
http://localhost:8080/jenkins Then the CLI can be downloaded like this:

```
wget http://localhost:8080/jenkins/jnlpJars/jenkins-cli.jar
```

Note In the remainder of this document I assume the Jenkins url is http://local
your Jenkins instance.
Example: in the try-it-out-yourself VM the correct command is:
wget http://192.168.33.65:8080/jnlpJars/jenkins-cli.jar
You can always check the correct address of the CLI by typing:
http://localhost:8080/jenkins/cli in your browser.

I will first show a few basic examples.

A list of all CLI functions:
Note: For the first example I also include the command for the try-it-out-yours

```
$ java -jar jenkins-cli.jar -s  http://localhost:8080/jenkins help
$ java -jar jenkins-cli.jar -s  http://192.168.33.65:8080 help
```

Login to the Jenkins system:

```
$ java -jar jenkins-cli.jar -s  http://localhost:8080/jenkins  \
        login --username someuser --password secret
```

A list of all jobs:

```
$ java -jar jenkins-cli.jar -s  http://localhost:8080/jenkins list-jobs
```

Copy an existing job named test to a new job named test2:

```
$ java -jar jenkins-cli.jar -s  http://localhost:8080/jenkins copy-job test t
```

and build that new job:

```
$ java -jar jenkins-cli.jar -s  http://localhost:8080/jenkins build test2
```

View the console output of the last run of this job:

```
$ java -jar jenkins-cli.jar -s  http://localhost:8080/jenkins  console test2
```

More instructions can be found inside your Jenkins. Just type the following in y
http://localhost:8080/jenkins/cli and you'll find instructions on how to use the

If you haven't set up public key authentication you will get a warning every tim
an ssh-key on:
https://help.github.com/articles/generating-ssh-keys/
Then in your browser go to:
http://localhost:8080/jenkins/user/myuserid/configure
Replace myuserid for the userid you use in Jenkins. Copy the ssh-key you gene

Ok with that solved, let's look at a couple more examples.

Backup (save) a job named test2 definition in an XML file named config.xml:

```
$ java -jar jenkins-cli.jar -s  http://localhost:8080/jenkins  get-job test2
```

Restore a saved job to a job named test3 from an XML file named config.xml:

```
$ java -jar jenkins-cli.jar -s  http://localhost:8080/jenkins \
        create-job test3 < config.xml
```

Installing a plugin:

```
$ java -jar jenkins-cli.jar -s  http://localhost:8080/jenkins install-plugin
http://updates.jenkins-ci.org/latest/build-monitor-plugin.hpi  -restart
```

How to use the CLI in a program (bash script)

Now we can use this knowledge to write programs that automatically configur
jobs in xml files. It first dumps the names of all jobs in a text file, and then loop

```
#!/bin/bash

# Sample bash script to backup Jenkins jobs.

java -jar jenkins-cli.jar -s http://localhost:8080/jenkins/ \
      list-jobs > jobs.txt

NUMBER=1
while read p; do
    echo $NUMBER " " $p
    FILENAME="$p.xml"
    java -jar jenkins-cli.jar -s http://localhost:8080/jenkins/
              get-job "$p" > "$FILENAME" ;
    (( NUMBER++ ))
done < jobs.txt

exit 0
```

The Jenkins REST API

It is also possible to program against a REST API of Jenkins (also called Remote
does that mean? You can try it in your browser by entering the following url's.
http://localhost:8080/jenkins/pluginManager/api/xml?depth=1
http://localhost:8080/jenkins/pluginManager/api/json?depth=1
From your terminal you can get the same result by adding a curl command bef

In your browser you can type http://localhost:8080/jenkins/api to get more inf
to identify yourself. This token can be found under: http://localhost:8080/jenk

Sounds difficult? Let's look a some examples.

Write the config file of job test2 and save it in a new config file named configts

```
$ curl "http://localhost:8080/jenkins/job/test2/config.xml" > configtst2.xml
```

Create a new job named test6 from a saved config.xml file:

```
$ curl -X POST -H "Content-Type:application/xml" -d @config.xml \
 http://localhost:8080/jenkins/createItem?name=test6
```

Trigger a build job:

```
$ curl -X POST http://localhost:8080/jenkins/job/test2/build \
--data token=0123456789abcdefghijklmnopqrstuvwxyz
```

Using wget
Data from Jenkins can also be downloaded with wget. For example: You want t
called test2 and we want the output of the 4th build.

```
$  wget "http://localhost:8080/jenkins/job/test2/4/consoleText"
```