



Starting and Accessing Jenkins

Added by [Derek Basch](#), last edited by [MICHAEL LUPO](#) on Sep 03, 2014

Starting Jenkins

The easiest way to execute Jenkins is through the built in Jetty servlet container. You can execute Jenkins like this:

```
$ java -jar jenkins.war
```

Of course, you probably want to send the output of Jenkins to a log file, and if you're on Unix, you probably want to use **nohup**:

```
$ nohup java -jar jenkins.war > $LOGFILE 2>&1
```

Accessing Jenkins

To see Jenkins, simply bring up a web browser and go to URL **http://myServer:8080** where **myServer** is the name of the system running Jenkins.

[Top of page](#)

Command Line Parameters

Jenkins normally starts up using port 8080. However, if you have other web services starting up you might find that this port is already taken. You can specify a different port by using `--httpPort=$HTTP_PORT` where `$HTTP_PORT` is the port you want Jenkins to run on. Other command line parameters include:

Command Line Parameter	Description
<code>--httpPort=\$HTTP_PORT</code>	Runs Jenkins listener on port \$HTTP_PORT using standard <i>http</i> protocol. The default is port 8080. To disable (because you're using <i>https</i>), use port -1.
<code>--httpListenAddress=\$HTTP_HOST</code>	Binds Jenkins to the IP address represented by \$HTTP_HOST. The default is 0.0.0.0 — i.e. listening on all available interfaces. For example, to only listen for requests from localhost, you could use: <code>--httpListenAddress=127.0.0.1</code>
<code>--httpsPort=\$HTTP_PORT</code>	Uses HTTPS protocol on port \$HTTP_PORT
<code>--httpsListenAddress=\$HTTPS_HOST</code>	Binds Jenkins to listen for HTTPS requests on the IP address represented by \$HTTPS_HOST.
<code>--prefix=\$PREFIX</code>	Runs Jenkins to include the \$PREFIX at the end of the URL. For example, to make Jenkins accessible at http://myServer:8080/jenkins , set <code>--prefix=/jenkins</code>
<code>--ajp13Port=\$AJP_PORT</code>	Runs Jenkins listener on port \$AJP_PORT using standard <i>AJP13</i> protocol. The default is port 8009. To disable (because you're using <i>https</i>), use port -1.
<code>--ajp13ListenAddress=\$AJP_HOST</code>	Binds Jenkins to the IP address represented by \$AJP_HOST. The default is 0.0.0.0 — i.e. listening on all available interfaces.
<code>--argumentsRealm.passwd.\$ADMIN_USER</code>	Sets the password for user \$ADMIN_USER. If Jenkins security is turned on, you must log in as the \$ADMIN_USER in order to configure Jenkins or a Jenkins project. NOTE: You must also specify that this user has an <i>admin</i> role. (See next argument below).
<code>--argumentsRealm.roles.\$ADMIN_USER=admin</code>	Sets that \$ADMIN_USER is an administrative user and can configure Jenkins if Jenkins' security is turned on. See Securing Jenkins for more information.
<code>-Xdebug -Xrunjdwp:transport=dt_socket,address=\$DEBUG_PORT,server=y,suspend=n</code>	Sets debugging on and you can access debug on \$DEBUG_PORT.
<code>-logfile=\$LOG_PATH/winstone_date+"%Y%m-%d_%H-%M".log</code>	Logging to desired file
<code>-XX:PermSize=512M -XX:MaxPermSize=2048M -Xmn128M -Xms1024M -Xmx2048M</code>	referring to these options for Oracle Java

Jenkins passes all (or just leading parameters until the first Jenkins-specific parameter?) command line parameters to the Winstone servlet container, so you can get more information by looking at the [Winstone Command Line Parameter Reference](#)

⚠ Be Careful with Command Line Parameters

Jenkins ignores command line parameters it doesn't understand instead of producing an error. Be careful when using command line parameters and make sure you have the correct spelling. For example, the parameter needed for defining the Jenkins administrative user is **--argumentsRealm** and not **--argumentRealm**.

A very simple init script

i The following script is for Ubuntu based systems. The [RedHat Jenkins distribution](#) contains a startup script.

```
#!/bin/sh

DESC="Jenkins CI Server"
NAME=jenkins
PIDFILE=/var/run/$NAME.pid
RUN_AS=jenkins
COMMAND=/usr/bin/java -- -jar /home/jenkins/jenkins.war

d_start() {
    start-stop-daemon --start --quiet --background --make-pidfile --pidfile $PIDFILE --chuid $RUN_AS --exec $COMMAND
}

d_stop() {
    start-stop-daemon --stop --quiet --pidfile $PIDFILE
    if [ -e $PIDFILE ]
    then rm $PIDFILE
    fi
}

case $1 in
    start)
        echo -n "Starting $DESC: $NAME"
        d_start
        echo "."
        ;;
    stop)
        echo -n "Stopping $DESC: $NAME"
        d_stop
        echo "."
        ;;
    restart)
        echo -n "Restarting $DESC: $NAME"
        d_stop
        sleep 1
        d_start
        echo "."
        ;;
    *)
        echo "usage: $NAME {start|stop|restart}"
        exit 1
        ;;
esac

exit 0
```

Using HTTPS with an existing certificate

If you're setting up Jenkins using the built-in Winstone server and want to use an existing certificate for HTTPS:

```
--httpPort=-1 --httpsPort=443 --httpsKeyStore=path/to/keystore --httpsKeyStorePassword=keystorePassword
```

The keystore should be in JKS format (as created by the JDK 'keytool') and the keystore and target key must have the same password. (Placing the keystore arguments after Jenkins-specific parameters does not seem to work; either they are not forwarded to Winstone or Winstone ignores them coming after unknown parameters. So, make sure they are adjacent to the working --httpsPort argument.)

If your keystore contains multiple certificates (e.g. you are using CA signed certificate) Jenkins might end-up using a incorrect one. In this case you can [convert the keystore to PEM](#) and use following command line options:

```
--httpPort=-1 --httpsPort=443 --httpsCertificate=path/to/cert --httpsPrivateKey=path/to/privatekey
```

Passing the Command Line Parameters to an instance on a Mac OSX (Currently is Mavericks 10.9.4) that uses launchctl (rather than using Jenkins.jar to start up)

In this example, we set the Jenkins server to listen for HTTPS on port 8443. Not we do not disable the httpPort by passing in -1. So in this example, your server would answer on both http and https. We also assume that the user has already created the keystore (see the "Using SSL" section from <http://wiki.wocommunity.org/display/documentation/Installing+and+Configuring+Jenkins>)

```
sudo launchctl unload /Library/LaunchDaemons/org.jenkins-ci.plist

sudo defaults write /Library/Preferences/org.jenkins-ci httpsPort 8443
sudo defaults write /Library/Preferences/org.jenkins-ci httpsKeyStore /path/to/your/keystore/file
sudo defaults write /Library/Preferences/org.jenkins-ci httpsKeyStorePassword <keystore password>

sudo launchctl load /Library/LaunchDaemons/org.jenkins-ci.plist
```

[Top of page](#)

Child Pages (1)

Comments (8) [Show Comments](#) | [Add Comment](#)

Powered by a free **Atlassian Confluence Open Source Project License** granted to Jenkins. Evaluate Confluence today.

Printed by Atlassian Confluence 3.4.7, the Enterprise Wiki.