



## Quick and Simple Security

Added by [Kohsuke Kawaguchi](#), last edited by [Martin Rust](#) on Oct 08, 2014

Many users run Jenkins directly from Winstone like this:

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

This page describes steps to quickly setup a single administrative account. If you use Jenkins without logging into the administrative account, Jenkins will be read only and you cannot change Jenkins' configuration or trigger a build. However, you can download builds, see the build results, and examine the logs. If you log into the administrative account, you may adjust Jenkins' configuration and start builds.



For users who execute Jenkins through Tomcat or some other J2EE servlet container, or for users who need more complex security arrangements, please see the [Standard Security Setup](#) directions.

1. Start Jenkins by executing as you normally do, but add two extra parameters to the startup:

```
$ java -jar jenkins.war --argumentsRealm.passwd.user=password --  
argumentsRealm.roles.user=admin
```

Note that **user** should be the name of the administrative user, and **password** should be the password for that user. Also please note that the user is specified twice - once in each argument.

For example, I want to have a user **jenkins** with a password of **swordfish** as my administrative user:

```
$ java -jar jenkins.war --argumentsRealm.passwd.jenkins=swordfish --  
argumentsRealm.roles.jenkins=admin
```

Notice that the word *arguments* is plural in both `--argumentsRealm` command line parameters.

2. Enable the security setting, go to <http://yourhost/jenkins/configureSecurity> and select *enable security*, then choose *Delegate to servlet container* for security realm and *Legacy mode* for authorization strategy. Or, you may click on the **Manage Jenkins** link on the left side of the main *Jenkins* dashboard page to get to the configuration page, then click **Configure Global Security**.
3. To log into *Jenkins* go to <http://yourhost/jenkins/loginEntry>, or click on the **login** link located on the top right hand corner of any Jenkins Dashboard page.
4. This will bring you to the Login page. Enter the user name and password, and click on the **Submit** button. This will take you back to the main Jenkins dashboard. You will now see the **Manage Jenkins** link on the left side of the page.
5. To logout simply click the **logout** link located on the top right hand corner of any *Jenkins* dashboard page.

## Allowing Developers to do Forced Builds with Security On

In this quick and simple set up, security on Jenkins is an all or nothing affair. If you do not turn on security, anyone may create, modify, or even delete Jenkins projects. If security is on, only the administrator may modify Jenkins projects, but developers lose the ability to force rebuilds. Fortunately, there is a way to allow developers to force rebuilds without giving them the ability to modify Jenkins itself.

1. Log into Jenkins, so you can modify its configuration.
2. From the main dashboard page, go to the specific job you want developers to be able to force a build on by clicking on its name.
3. Click on the *Configure* link located on the left hand side of the page.
4. In the **Build Triggers** section, select the *Trigger builds remotely (e.g., from scripts)* checkbox.
5. An **Authentication Token** field box will display. Put in a text string that you can use to help trigger builds. In the example below, the string is *build*.

**Build Triggers**

☐ Build after other projects are built

☒ Trigger builds remotely (e.g., from scripts)

Authentication Token

Use the following URL to trigger build remotely:  
HUDSON\_URL/job/AIM/build?token=TOKEN\_NAME

☒ Poll SCM

Schedule

☐ Build periodically

6. You will need to either create a script, or if you have a webserver, create a webpage that will allow you to trigger the build. The format of the build URL is:

`http://jenkinsHost/job/project/build?token=token`

Where:

- **jenkinsHost** is the URL for your Jenkins Host (example: `jenkins:8080`)
- **project** is the name of your project or job
- **token** is the token string that you entered into the **Authentication Token** field

If you have a Webserver, you can create a webpage with the URL. Here's an example below:

```
<h1>Jenkins Force Build Page</h1>
<ul>
  <li>
    <a href="http://jenkins:8080/job/FOO/build?token=build">Force build of Project FOO on Jenkins</a>
  </li>
</ul>
```

The above code is to force a build on the Jenkins server *jenkins* that uses port 8080 for project FOO. It assumes that the string entered into the **Authentication Token** field was *build*.

### ✓ Using a Shell Script

You can also force a build by specifying the *force build URL* with either the [wget](#) command or the [curl](#) command.

### ⚠ Anonymous read access required

The preceding instructions require that anonymous users have read access to Jenkins. If you disable overall read access for anonymous users, the [Build Token Root Plugin](#) allows use of tokens.

**Comments (2)** [Show Comments](#) | [Add Comment](#)