# Run Jenkins Build From Command Line

Jenkins has support to command line client that allows you to access Jenkins from command line.

**To Trigger Jenkins build from command line some prequiste are there**

1. Jenkins service is running.
2. Enable security option under “Configure Global Security”

Go to jenkins dashboard in Home page ( e.g http://localhost:8080/ ) -> Manage Jenkins  
-> Configure Global Security -> Click on “**Enable security**” checkbox

You can also configure “**Access Control**” and “**Authorization**” option in Global Security page.  
Jenkins allow us to trigger Jenkins build with any specific user, For that we have to pass username and password in command line.



You can check all CLI option in **http://<jenkins server >/cli/** ( e.g http://localhost:8080/cli/ ) page.

To access **CLI** feature in Jenkins download **jenkins-cli.jar** (http://<jenkins server>/jnlpJars/jenkins-cli.jar) and place this jar your directory or location from you are going to run jenkins build command.

Jenkins Command build Option:-

**Syntax :-**

java -jar jenkins-cli.jar -s http://<jenkins server>/ build build-name [-c] [-f] [-p] [-r N] [-s] [-v] [-w]

**Option Descriptions:-**

build-name : Name of the job to build

-c  : Check for SCM changes before starting the build, and if there's no change, exit without doing a build

-f  : Follow the build progress. Like -s only interrupts are not passed through to the build.

-p  : Specify the build parameters in the key=value format.

-s  : Wait until the completion/abortion of the command. Interrupts are passed through to the build.

-v  : Prints out the console output of the build. Use with -s

-w  : Wait until the start of the command

e.g.

**java -jar jenkins-cli.jar -s http://localhost:8080/ build 'my-project-build' --username roop --password roop**

**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 test2

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

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 > config.xml

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