

Jenkins Tutorial -DevOps Tool

Jenkins Tutorial

1. Introduction

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks such as building, testing, and deploying software. Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with the Java Runtime Environment installed. Jenkins is simply the old **Hudson** with a new name.

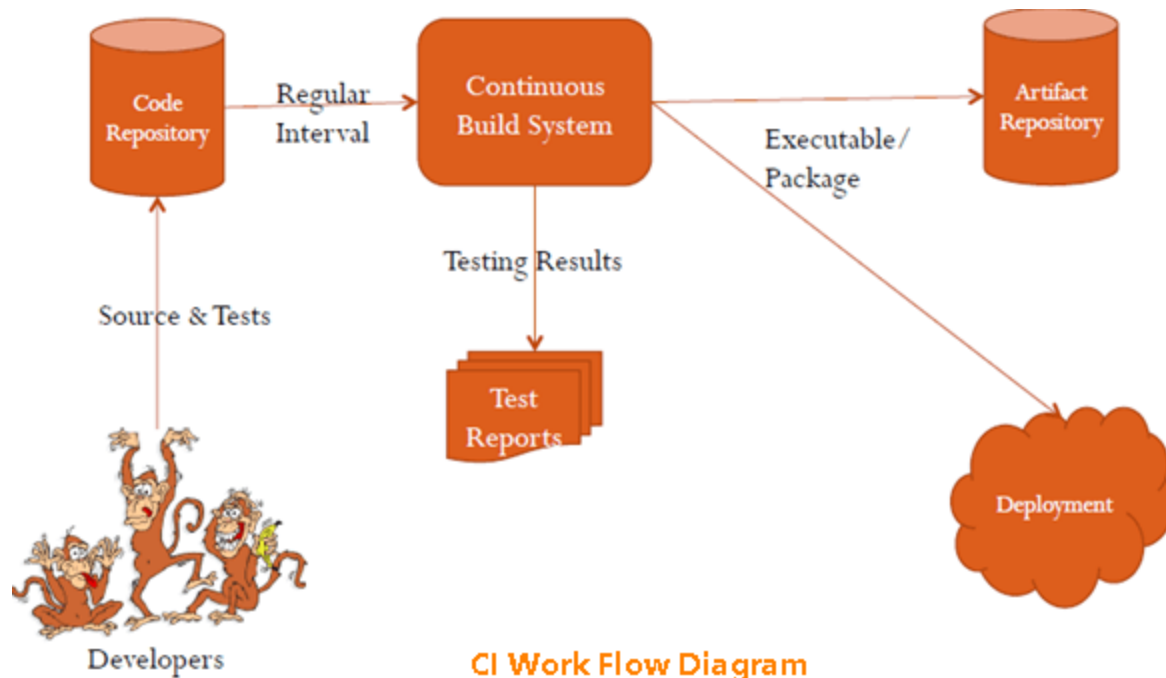
Jenkins is a software that allows **continuous integration**. Jenkins will be installed on a server where the central build will take place. The following flowchart demonstrates a very simple workflow of how Jenkins works.

1.1 Continuous Integration Workflow

Martin Fowler said “Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible”

At a regular frequency (ideally at every commit), the system is:

1. **Integrated** – All changes up until that point are combined into the project
2. **Built** – The code is compiled into an executable or package
3. **Tested** – Automated test suites are run
4. **Archived** – Versioned and stored so it can be distributed as is, if desired
5. **Deployed** – Loaded onto a system where the developers can interact with it



Continuous Integration Tools

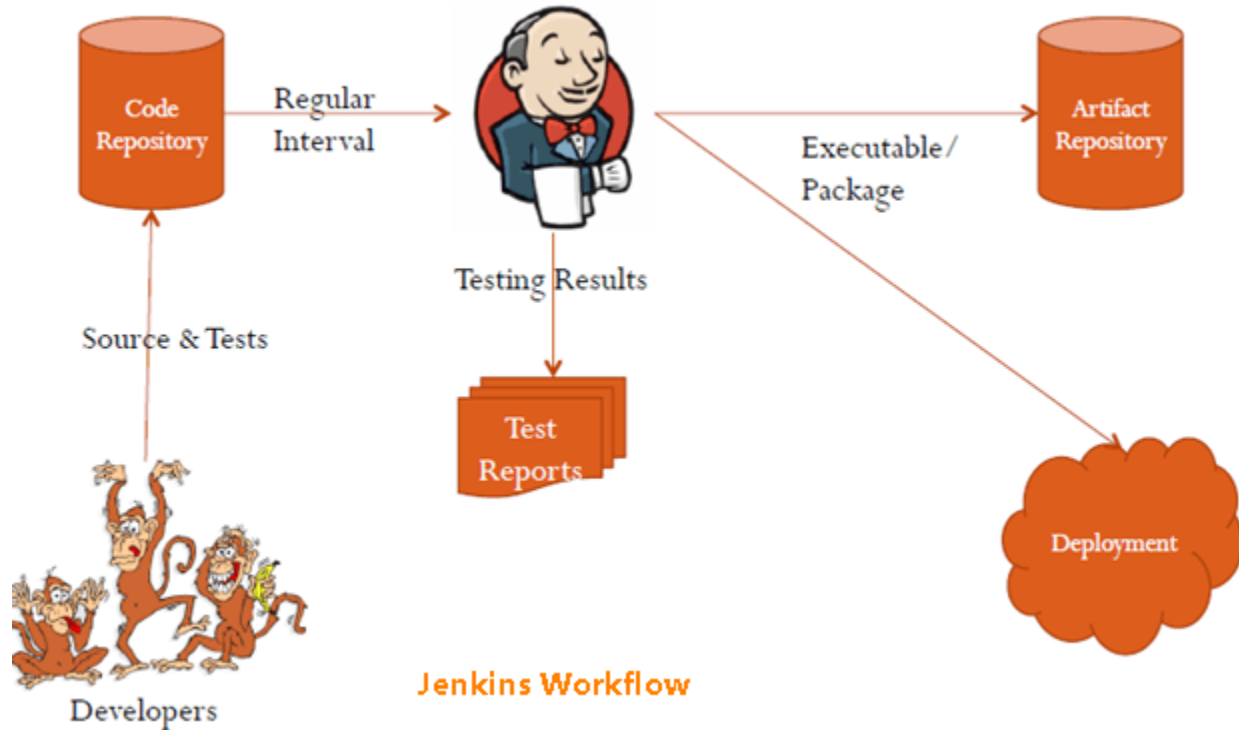
- Code Repositories : **SVN, Mercurial, Git**
- Continuous Build Systems : **Jenkins, Bamboo, Cruise Control**
- Test Frameworks : **JUnit, Cucumber, CppUnit**
- Artifact Repositories : **Nexus, Artifactory, Archiva**

1.2 Jenkins Tool Workflow

Jenkins is a **Java based Continuous Build System** Branched from Hudson, **Runs in servlet container (Glassfish, Tomcat)**. It is supported by over 400 plugins like SCM, Testing, Notifications, Reporting, Artifact Saving, Triggers, and External Integration etc.

In 2005 – Hudson was first release by Kohsuke Kawaguchi of Sun Microsystems. **2010** – **Oracle** bought Sun Microsystems Due to a naming dispute, **Hudson was renamed to Jenkins**

Oracle continued development of Hudson (as a branch of the original)



Key Features of Jenkins

- It will Generate test reports
- Jenkins can Integrate with many different Version Control Systems
- Jenkins will Deploys directly to production or test environments and many more

2. Jenkins Installation

Requirements

- Java 7 or Java 8 must be installed.
- Requires minimum RAM of 512MB

2.1 Installing Jenkins in Windows

1. **Download Jenkins.war**. (if it is downloaded in .zip format rename/change extension to .war)



The screenshot shows a Windows Command Prompt window. The title bar reads "Command Prompt". The command prompt shows the command `java -version` has been entered and executed. The output is:


```
java version "1.8.0_101"
Java(TM) SE Runtime Environment
(c) 2015 Oracle Corporation and/or its affiliates. All rights reserved.
Oracle and Java are registered trademarks of Oracle Corporation and/or its
affiliates. Other names and brands may be trademarks of their respective
owners.
```

```
C:\DevOps>java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.111-b14, mixed mode)
```

```
C:\Program Files\Java\jre6\bin>
```

```
C:\DevOps>java -jar jenkins.war
Running from: C:\DevOps\jenkins.war
webroot: $user.home/.jenkins
+[[33mMar 07, 2017 5:39:19 PM Main deleteWinstoneTempContents
=====
=====
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
59caa78abb75463ca14a8a9df718a908
This may also be found at: C:\Users\k          s\jenkins\secrets\initialAdminPassword
=====
=====
```

Now browse to <http://localhost:8080> and enter the generated password to continue the installation

5.Next it will takes you to “Create First Admin User”, provide details & finish

Getting Started

Create First Admin User

Username:

admin

Password:

.....

Confirm password:

.....

Full name:

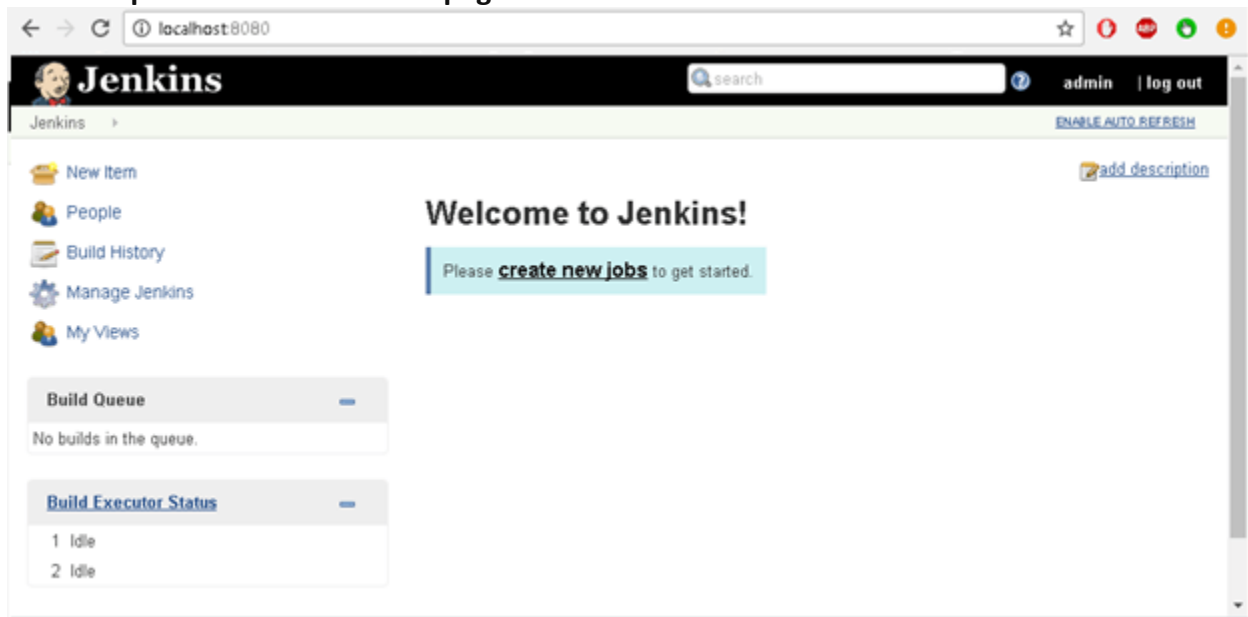
admin

Jenkins 2.32.8

Continue as admin

Save and Finish

6.It will opens the Jenkins Home page as below



We can also install Jenkins using Apache Tomcat also. Just download & Start the tomcat. Upload the Jenkins.war in tomcat from admin panel. You can access by using <http://localhost:8080/jenkins>

2.2 How to Change Jenkins Port number

Some times 8080 is busy with some other services. In that case we can change port to some other number by using following steps

1. Press Ctrl+C on Jenkins command line to Stop the Service

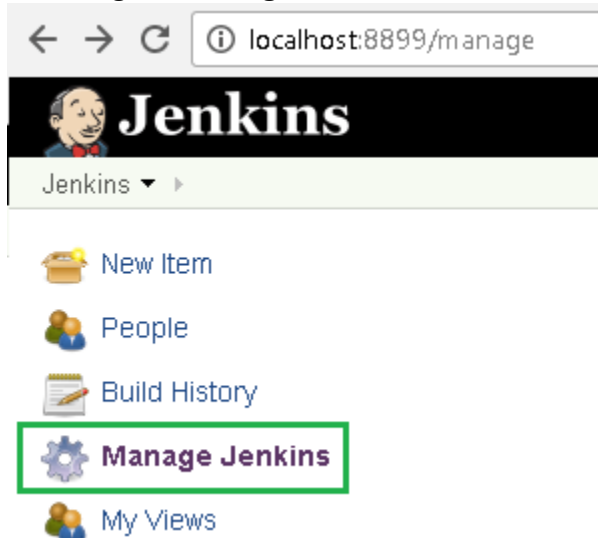
2. If it is Jenkins.war Installation, Start Jenkins from cmd line using :

```
java -jar jenkins.war --httpPort=8081
```

3. If it is tomcat Installation, open **xml** & change **"--httpPort=8080"** with new port number

3. Jenkins Configuration

We can configure Jenkins jobs based up on our requirement. For doing any configuration we have to go to **Manage Jenkins** on the left menu of the Dashboards.



It contains following modules for configuration

Manage Jenkins

	Configure System Configure global settings and paths.		Jenkins CLI Access/manage Jenkins from your shell, or from your script.
	Configure Global Security Secure Jenkins; define who is allowed to access/use the system.		Script Console Executes arbitrary script for administration/trouble-shooting/diagnostics.
	Global Tool Configuration Configure tools, their locations and automatic installers.		Manage Nodes Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
	Reload Configuration from Disk Discard all the loaded data in memory and reload everything from file system.		About Jenkins See the version and license information.
	Manage Plugins Add, remove, disable or enable plugins that can extend the functionality of Jenkins.		Manage Old Data Scrub configuration files to remove remnants from old plugins and earlier versions.
	System Information Displays various environmental information to assist trouble-shooting.		Install as Windows Service Installs Jenkins as a Windows service to this system, so that Jenkins starts automatically.
	System Log System log captures output from <code>java.util.logging</code> output related to Jenkins.		Manage Users Create/delete/modify users that can log in to this Jenkins
	Load Statistics Check your resource utilization and see if you need more computers for your builds.		Prepare for Shutdown Stops executing new builds, so that the system can be eventually shut down safely.

We have to use the above configuration as per our requirement. As of now we don't all of them. we need some basic configuration required for working with Jenkins.

For that we need to Install and Configure below Tools/ Softwares

- [Install Java](#)
- [Install Git](#) (just download & install as normal software)
- [Install Maven](#)
- [Install Ant](#)

3.1 Configure System

Here we can manage paths to the various tools to use in builds, such as the JDKs, the versions of Ant and Maven, as well as security options, email servers, and other system-wide configuration details. When plugins are installed. Jenkins will add the required configuration fields dynamically after the plugins are installed.

For Configure System we have to navigate to **Manage Jenkins → Select Configure System**

Global properties

☒ Environment variables

List of variables

Name

Value

Delete

Add

☒ Tool Locations

List of tool locations

Name

Home

Delete

Add

3.2 Configure Global Security

It is used for Securing Jenkins & define who is allowed to access/use the system.

The **Configure Global Security** page has two sections in which you:

- Set the security realm to determine who is allowed access
- Set the authorization to determine what each user is allowed to do

-

Jenkins' Own User Database

This is the simplest authentication scheme—Jenkins maintains its own independent user database. People can sign up for their own accounts, and you as the administrator decide who can do what in Jenkins.

1. Go to the Jenkins **dashboard**, usually <http://server:8080> or <http://server/jenkins:8080>, where server is the host on which Jenkins is running
2. Select **Manage Jenkins**, then **Configure Global Security**
3. Click **Enable Security** The page will expand to offer a choice of access control.
4. Select **Jenkins' own user database & check mark next to Allow users to sign up**
5. Continue with Authorization, below. In particular, do not forget to press the Save button at the bottom of the page.



Configure Global Security

☒ Enable security

TCP port for JNLP agents ☐ Fixed : ☒ Random ☐ Disable

Agent protocols...

Disable remember me ☐

Access Control

Security Realm

☐ Delegate to servlet container

☒ Jenkins' own user database

☒ Allow users to sign up

Authorization

☐ Anyone can do anything

☐ Legacy mode

☒ Logged-in users can do anything

☐ Allow anonymous read access

Save

Apply

3.3 Global Tool Configuration

The Global Tool Configuration lets you define variables that can be managed centrally but used in all of your build jobs. You can add as many properties as you want here, and use them in your build jobs. Jenkins will make them available within your build job environment, so you can freely use them within your Ant and Maven build scripts. Note that you shouldn't put periods (".") in the property names, as they won't be processed correctly.

3.3.1 Java Configuration in Jenkins

1. **Navigate to Manage Jenkins → Global Tool Configuration → JDK Installations → Add JDK**

2. **Set JDK Name.** For Ex: JDK_1.8.0_111 (uncheck Install automatically)

3. **Set JAVA_HOME directory path.** For Ex: C:\Program Files\Java\jdk 1.8.0_111

JDK

JDK installations

JDK	
Name	JDK_1.8.0_111
JAVA_HOME	C:\Program Files\Java\jdk 1.8.0_111
<input type="checkbox"/> Install automatically	Delete JDK

[Save](#) [Apply](#)

4. **Apply & Save.**

3.3.2 Apache Maven Configuration in Jenkins

1. **Navigate to Manage Jenkins → Global Tool Configuration → Maven Installations → Add Maven**

2. **Set Maven Name.** For Ex: apache-maven-3.3.9 (uncheck Install automatically)

3. **Set MAVEN_HOME directory path.** For Ex: C:\apache-maven-3.3.9

Maven

Maven installations

Maven	
Name	apache-maven-3.3.9
MAVEN_HOME	C:\apache-maven-3.3.9
Add Maven	

[Save](#) [Apply](#)

4. **Apply & Save.**

3.4 Reload Configuration from Disk

This Option will Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk. Be careful using this, it won't display any page directly deletes the data when you click on OK button on alert message!!

3.5 Manage Plugins

Here we can Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

The screenshot shows the Jenkins 'Manage Plugins' interface with the 'Available' tab selected. The table lists plugins under the '.NET Development' category. Each row includes a checkbox, the plugin name with a link, a description, and the version number. At the bottom, there are three buttons: 'Install without restart', 'Download now and install after restart', and 'Check now'. A status message indicates 'Update information obtained: 1 min 20 sec ago'.

Install	Name	Version
.NET Development		
<input type="checkbox"/>	CCM Plug-in This plug-in generates the trend report for CCM, an open source static code analysis program.	3.1
<input type="checkbox"/>	FxCop Runner plugin	1.1
<input type="checkbox"/>	MSBuild Plugin	1.27
<input type="checkbox"/>	MSTest plugin Generates test reports for MSTest.	0.19
<input type="checkbox"/>	MSTestRunner plugin	1.3.0
<input type="checkbox"/>	NAnt Plugin	1.4.3
<input type="checkbox"/>	NCover plugin	0.3

Install without restart Download now and install after restart Update information obtained: 1 min 20 sec ago Check now

We can install the plugins directly from Available plugins tab, and uninstall when ever you want

How to add Jenkins Plugins manually

Some time proxy settings won't allow us to contact Jenkins CI server directly. In those cases we can install Jenkins plugins by downloading them manually. Follow below Steps to do so

- 1.Go to <https://plugins.jenkins.io/>
- 2.Search the plugin you want & click on plugin page
- 3.on the right side of the page, click on Archives, select latest ,it will download plugin in **.hpi** format

4. Now go to **Manage Plugins** → **Advanced** tab come down & **upload plugin**

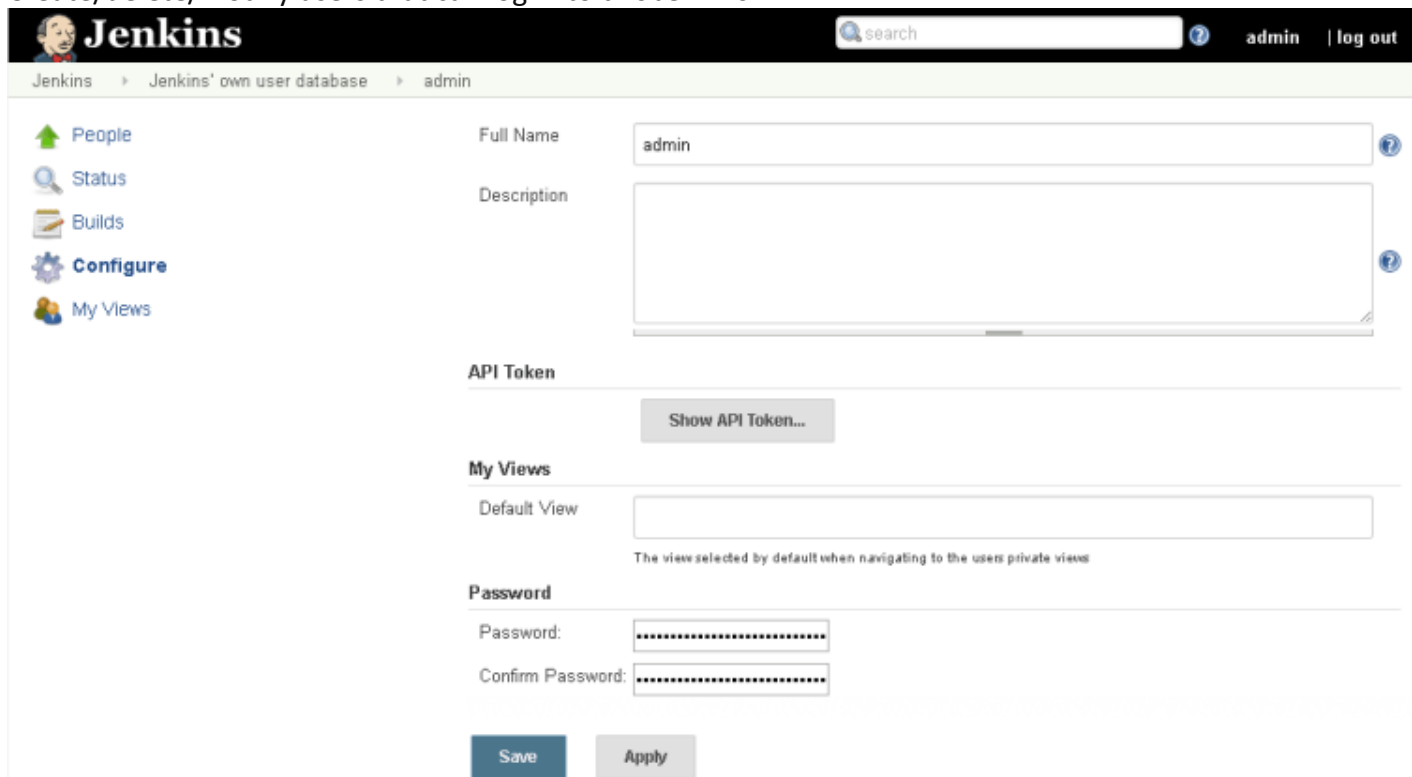
Upload Plugin

You can upload a .hpi file to install a plugin from outside the central plugin repository.

File: ant.hpi

3.6 Manage Users

Create/delete/modify users that can log in to this Jenkins



The screenshot shows the Jenkins web interface for managing users. The top navigation bar includes the Jenkins logo, a search bar, and the user 'admin' with a 'log out' link. The breadcrumb trail is 'Jenkins > Jenkins' own user database > admin'. On the left sidebar, there are links for 'People', 'Status', 'Builds', 'Configure' (highlighted), and 'My Views'. The main content area is titled 'admin' and contains several sections: 'Full Name' with a text input field containing 'admin'; 'Description' with a large text area; 'API Token' with a 'Show API Token...' button; 'My Views' with a 'Default View' dropdown menu and a note 'The view selected by default when navigating to the users private views'; and 'Password' with 'Password:' and 'Confirm Password:' fields, both masked with dots. At the bottom are 'Save' and 'Apply' buttons.

Jenkins search admin | log out

Jenkins > Jenkins' own user database > admin

People

Status

Builds

Configure

My Views

Full Name admin

Description

API Token

Show API Token...

My Views

Default View

The view selected by default when navigating to the users private views

Password

Password:

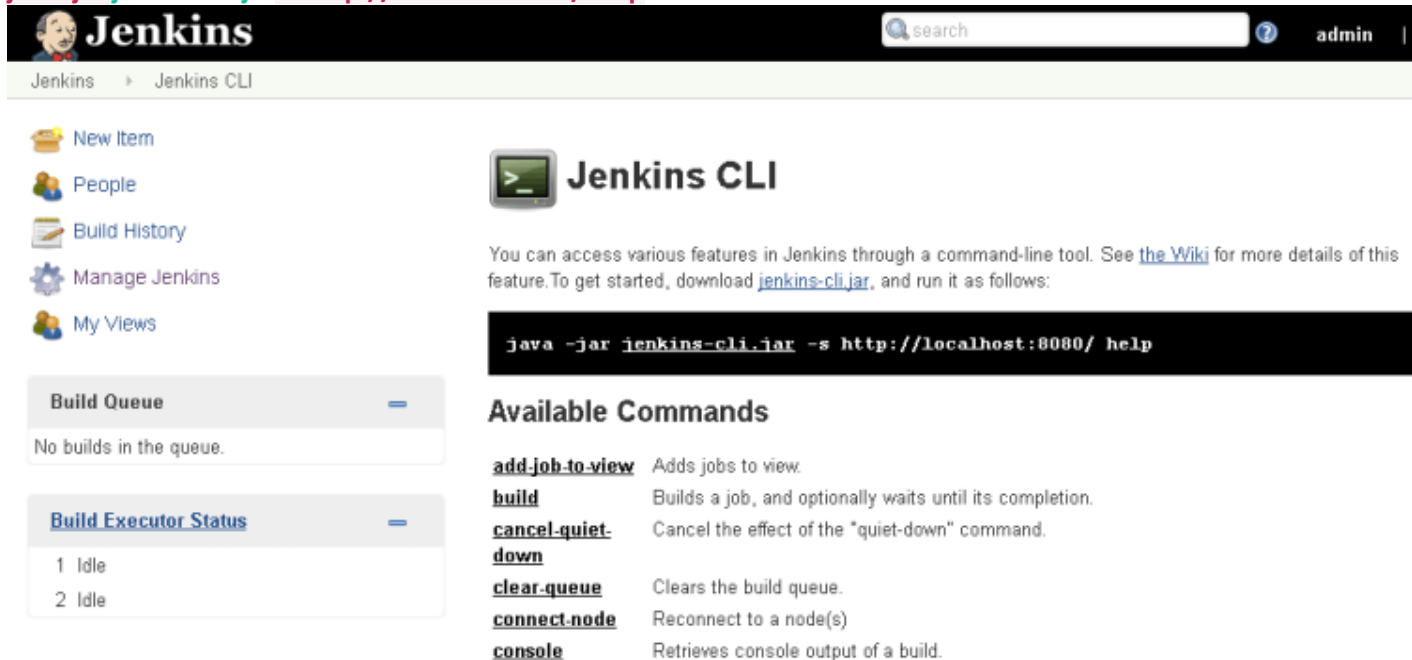
Confirm Password:

Save Apply

3.7 Jenkins CLI

Access/manage Jenkins from your shell, or from your script. You can access various features in Jenkins through a command-line tool. See [the Wiki](#) for more details of this feature. To get started, download [jenkins-cli.jar](#), and run it as follows:

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



The screenshot shows the Jenkins web interface. At the top is the Jenkins logo and a search bar. Below the logo is a navigation bar with links to Jenkins and Jenkins CLI. On the left sidebar, there are links for New Item, People, Build History, Manage Jenkins, and My Views. The main content area is titled "Jenkins CLI" and contains a terminal icon. Below the title, there is a text block explaining that various features can be accessed via a command-line tool, with a link to the Wiki. A code block shows the command: `java -jar jenkins-cli.jar -s http://localhost:8080/ help`. Below this, there is a section titled "Available Commands" with a list of commands and their descriptions:

Command	Description
<code>add-job-to-view</code>	Adds jobs to view.
<code>build</code>	Builds a job, and optionally waits until its completion.
<code>cancel-quiet-down</code>	Cancel the effect of the "quiet-down" command.
<code>clear-queue</code>	Clears the build queue.
<code>connect-node</code>	Reconnect to a node(s)
<code>console</code>	Retrieves console output of a build.

3.8 Script Console

Executes arbitrary script for administration/troubleshooting/diagnostics.



The screenshot shows the Jenkins web interface. On the left is a sidebar with links: People, Build History, Manage Jenkins, and My Views. Below these are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing two idle executors). The main area is titled 'Script Console' with a notepad icon. It contains instructions on how to use Groovy scripts, an example code snippet `println(Jenkins.instance.pluginManager.plugins)`, and a note about pre-imported classes. At the bottom of the main area is a large text input field with a cursor at the first line. The footer shows 'localhost:8080' and a 'Run' button.

3.9 Install as Windows Service

Installs Jenkins as a Windows service to this system, so that Jenkins starts automatically when the machine boots.



Install as Windows Service


Installing Jenkins as a Windows service allows you to start Jenkins as soon as the machine starts, and regardless of who is interactively using Jenkins.

Installation Directory

Install

3.10 Manage Nodes

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Windows Server 2012 R2 (amd64)	In sync	1.46 GB	16.83 GB	1.46 GB	0ms
Data obtained		5 min 1 sec	5 min 1 sec	5 min 1 sec	5 min 1 sec	5 min 1 sec	5 min 1 s

Refresh status

And also we have some other things in manage Jenkins, they are

- **Manage Old Data:** remove remnants from old plugins and earlier versions.
- **System Information:** Displays various environmental information to assist trouble-shooting.
- **System Log:** system log captures output from java.util.logging output related to Jenkins.
- **Load Statistics :** Check your resource utilization
- **Prepare for Shutdown:** Stops executing new builds, so that the system can be shut down safely.

3.11 Jenkins Pipeline

Jenkins Pipeline is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins

Typically, this “Pipeline as Code” would be written to a **Jenkinsfile** and checked into a project’s source control repository, for example:

Jenkinsfile (Declarative Pipeline)

```
pipeline {
  agent any

  stages {
    stage('Build') {
      steps {
        echo 'Building..'
      }
    }
    stage('Test') {
      steps {
        echo 'Testing..'
      }
    }
  }
}
```

```
}
stage('Deploy') {
  steps {
    echo 'Deploying....'
  }
}
}
```