

# Jenkins Essentials



# Learning Objectives

- ☛ Overview of Jenkins
- ☛ Installation & Configuration
- ☛ Creating a Sample Job
- ☛ Jenkins Plugins
- ☛ The Big Picture

About  
Jenkins



# Jenkins

# Open Source



- ☞ You can view source
- ☞ License is free
- ☞ Good Support
- ☞ Frequent Updates

# Web Based JAVA Tool

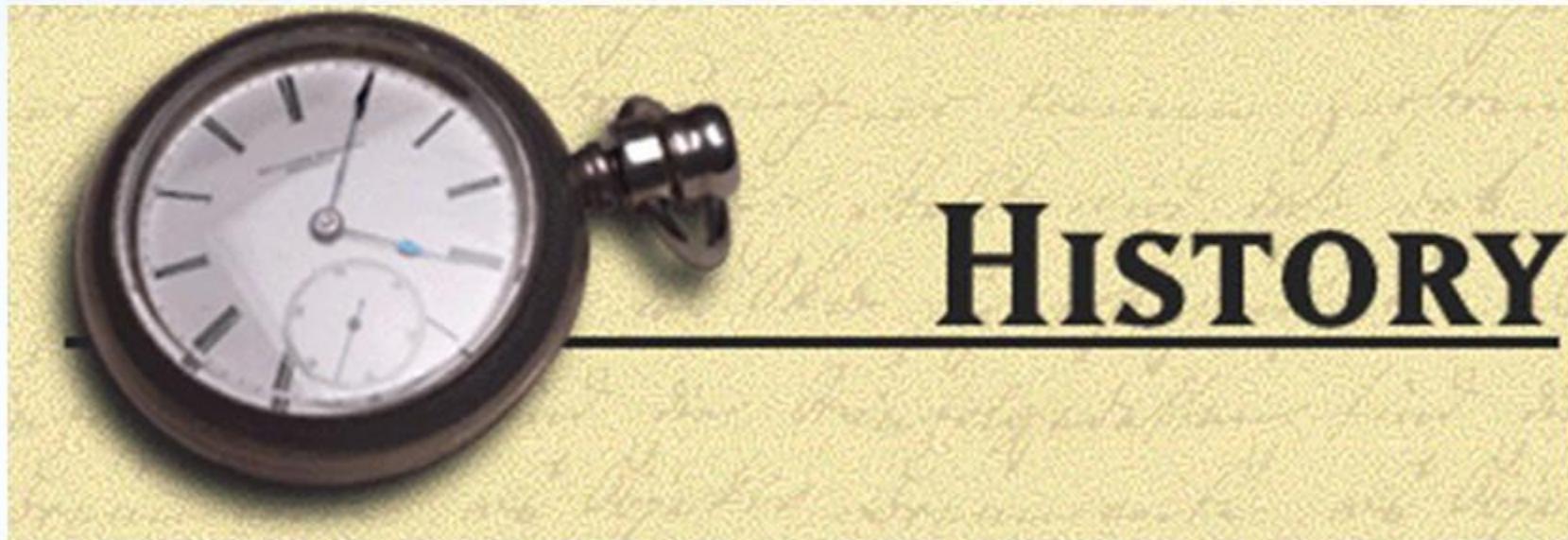


- ☞ Built in JAVA
- ☞ Works well with other languages
- ☞ Organized as JAVA Deployable package
- ☞ Built in web server
- ☞ Very Easy to get it up and running

# Highly Extensible

- ☞ Variety of Plugins Available
- ☞ Flexibility to create custom plugins





**HISTORY**

# History of Jenkins

**Kohsuke  
Kawaguchi**

Started  
around 2007  
as Hudson

**ORACLE  
Bought SUN**

2009-2010  
disturbance  
in the ranks



Rapid rise in  
popularity in  
2008

**Replaced  
Cruise Control**

2011 The  
great  
division

**HUDSON and JENKINS  
Separated**

# The Jenkins / Hudson Controversy



# The Jenkins / Hudson Controversy

**Hudson**

Oracale/Eclipse

**Jenkins**

Kohsuke  
Kawaguchi

# The Problem

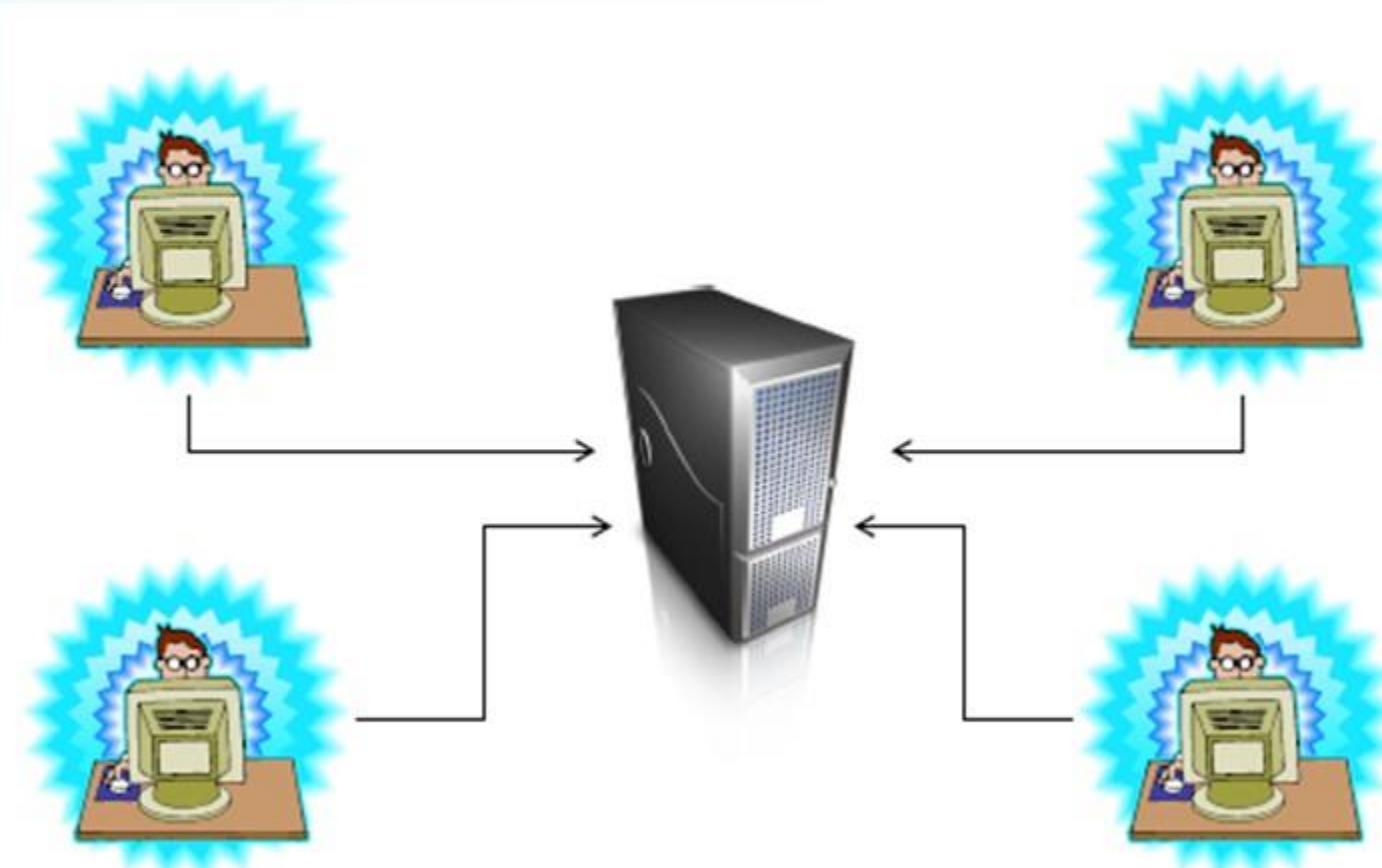


Integration is Painful

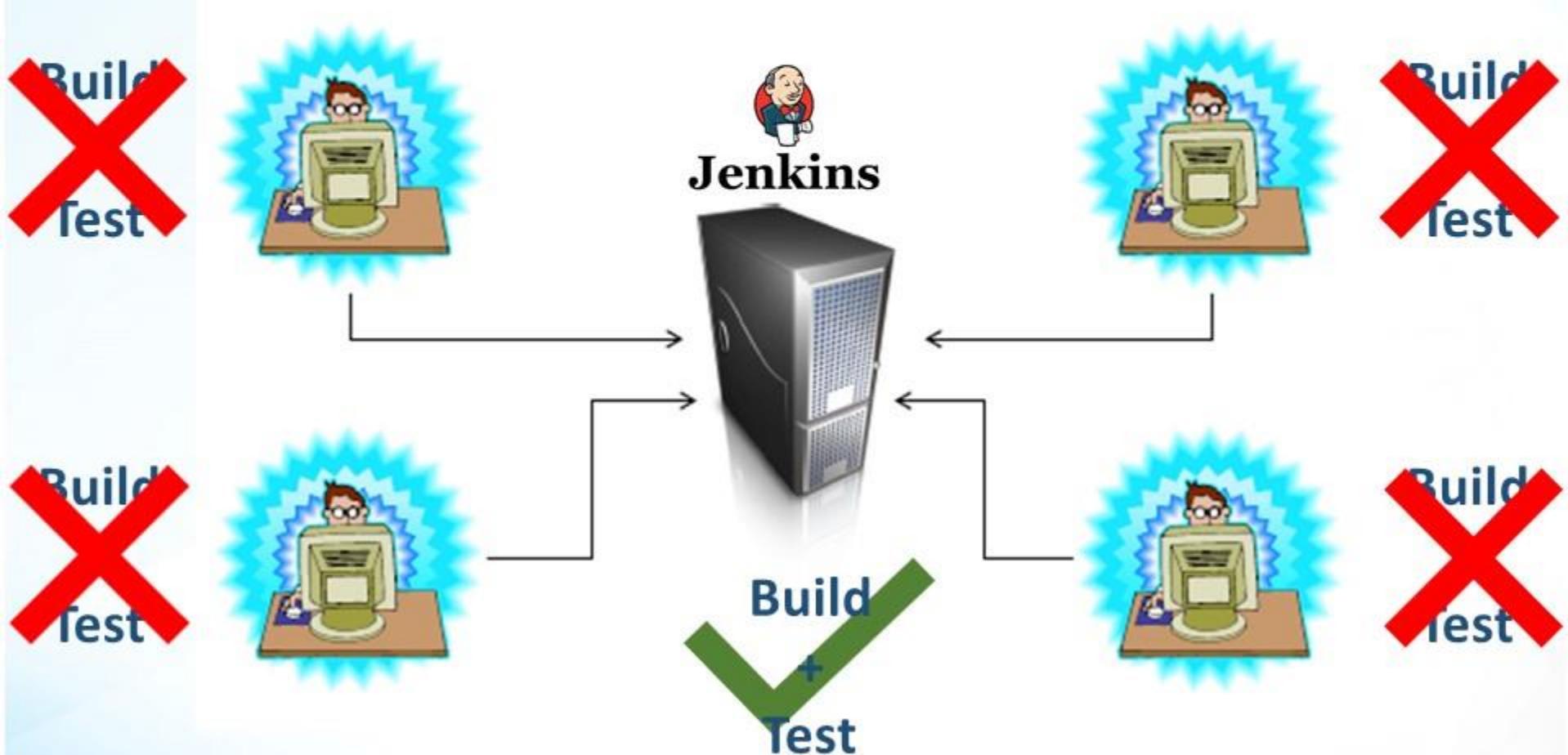
# The Problem



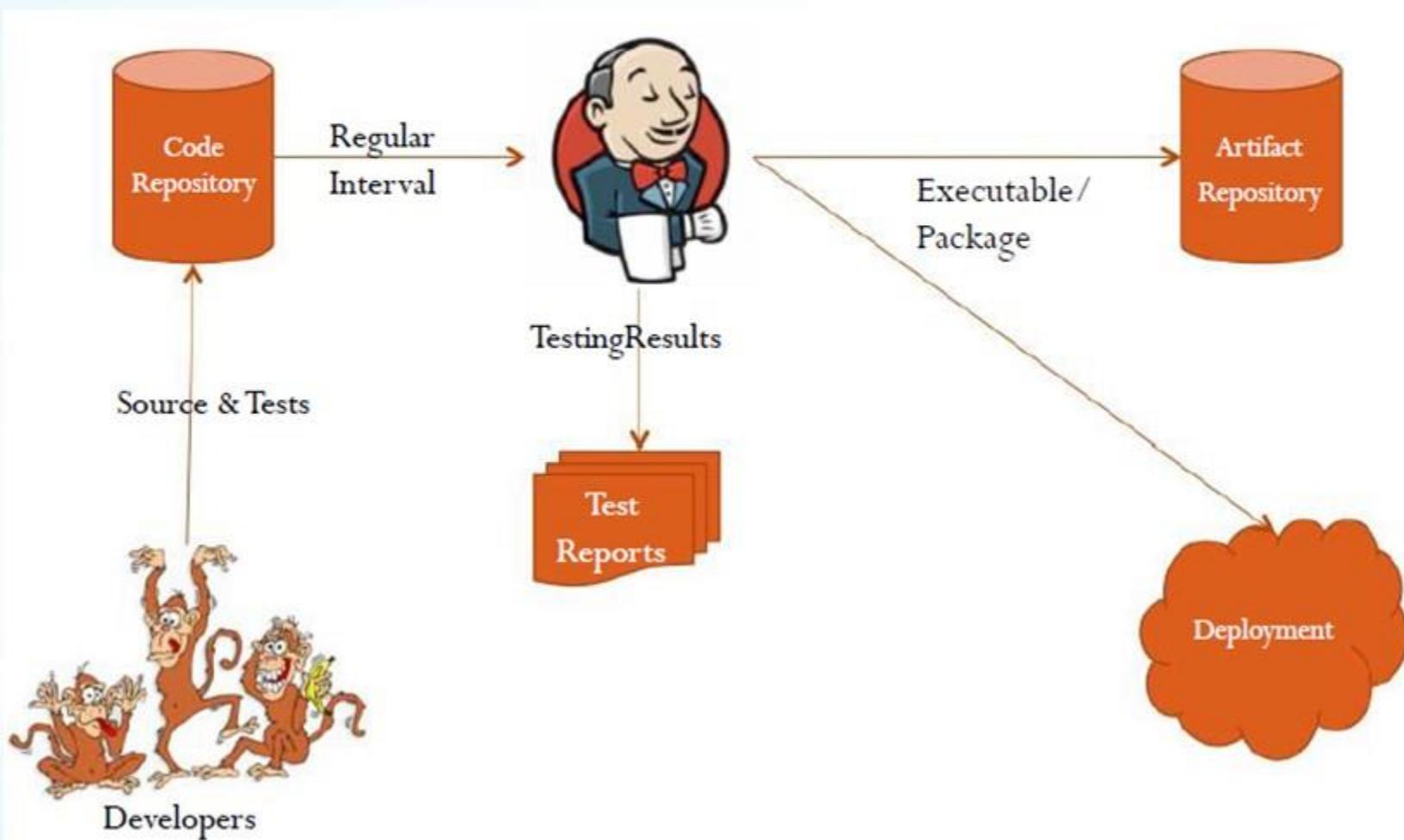
# What is Continuous Integration? (CI)



# Where Jenkins Fits In



# Where Jenkins Fits In



# Installation & Configuration

# Prerequisites

- ☞ Java 1.5 or higher
- ☞ Java in your path (optional, but preferred)
- ☞ Web server (only if you want to host outside of Jenkins)



## ☞ Install JAVA

- ☞ `sudo add-apt-repository ppa:webupd8team/java`
- ☞ `sudo apt-get update`
- ☞ `sudo apt-get install oracle-java8-installer`

## ☞ Set Environment Variable

- ☞ `sudo update-alternatives --config java`
- ☞ `sudo vim /etc/environment`
- ☞ `JAVA_HOME="/usr/lib/jvm/java-8-oracle"`
- ☞ `echo $JAVA_HOME`

# Getting Jenkins

- ☞ Open the Jenkins Website <https://jenkins.io/>
- ☞ Got to Download
  - ☞ Weekly Release - A new release is produced weekly to deliver bug fixes and features to users and plugin developers.
  - ☞ **Long Term Support Release** – Stable release b/w every 12 weekly release.
- ☞ Download the latest stable release.

## Starting Jenkins

- ☞ Open Command Line
- ☞ Go to the folder where you have kept the war file.
- ☞ Type the command “java –jar Jenkins.war”

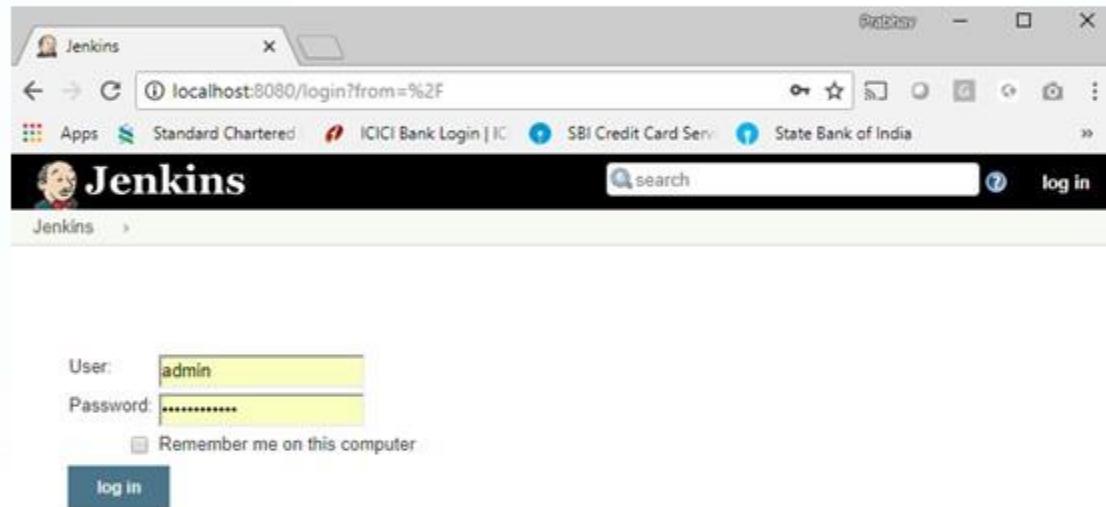
**Note:** You must have the JAVA Path set in the environment variables.



# Jenkins

# Testing Jenkins

- ☞ The default port number for Jenkins is “8080”
- ☞ Open the web server <http://localhost:8080>



**Hit **Ctrl+C** to stop this Jenkins.**

# Common Command Line Options

☞ Run Jenkins on a different port number.

☞ `java -jar Jenkins.war --httpPort=8081`

☞ Run Jenkins on HTTPS

☞ `java -jar Jenkins.war --httpPort=-1 --httpsPort=443`

☞ This is commonly used when you want to secure Jenkins

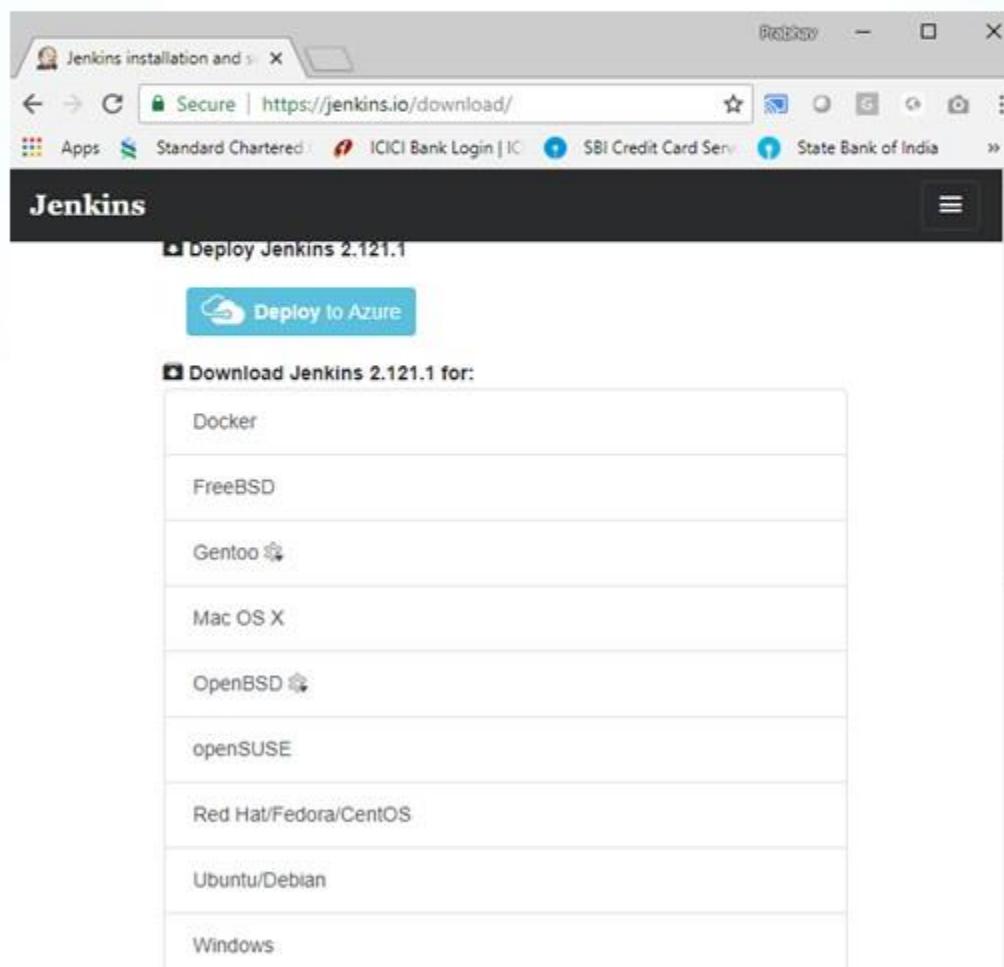
☞ Open the web server and see Jenkins running.

More command line options can be found here

<https://wiki.jenkins.io/display/JENKINS/Starting+and+Accessing+Jenkins>

# Getting the Service Installer

- ☞ Open the Jenkins Website  
<https://jenkins.io/>
- ☞ Got to Download
- ☞ See the various available service packages.
- ☞ Select and download the package based on the OS



# Installing as a Service - Ubuntu

- ☞ Open the terminal
- ☞ Get Jenkins
  - ☞ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
- ☞ Add Library in /etc/apt/sources.list
  - ☞ deb https://pkg.jenkins.io/debian-stable binary/
- ☞ Update
  - ☞ sudo apt-get update
- ☞ Install
  - ☞ sudo apt-get install jenkins

# Installing as a Service - Windows

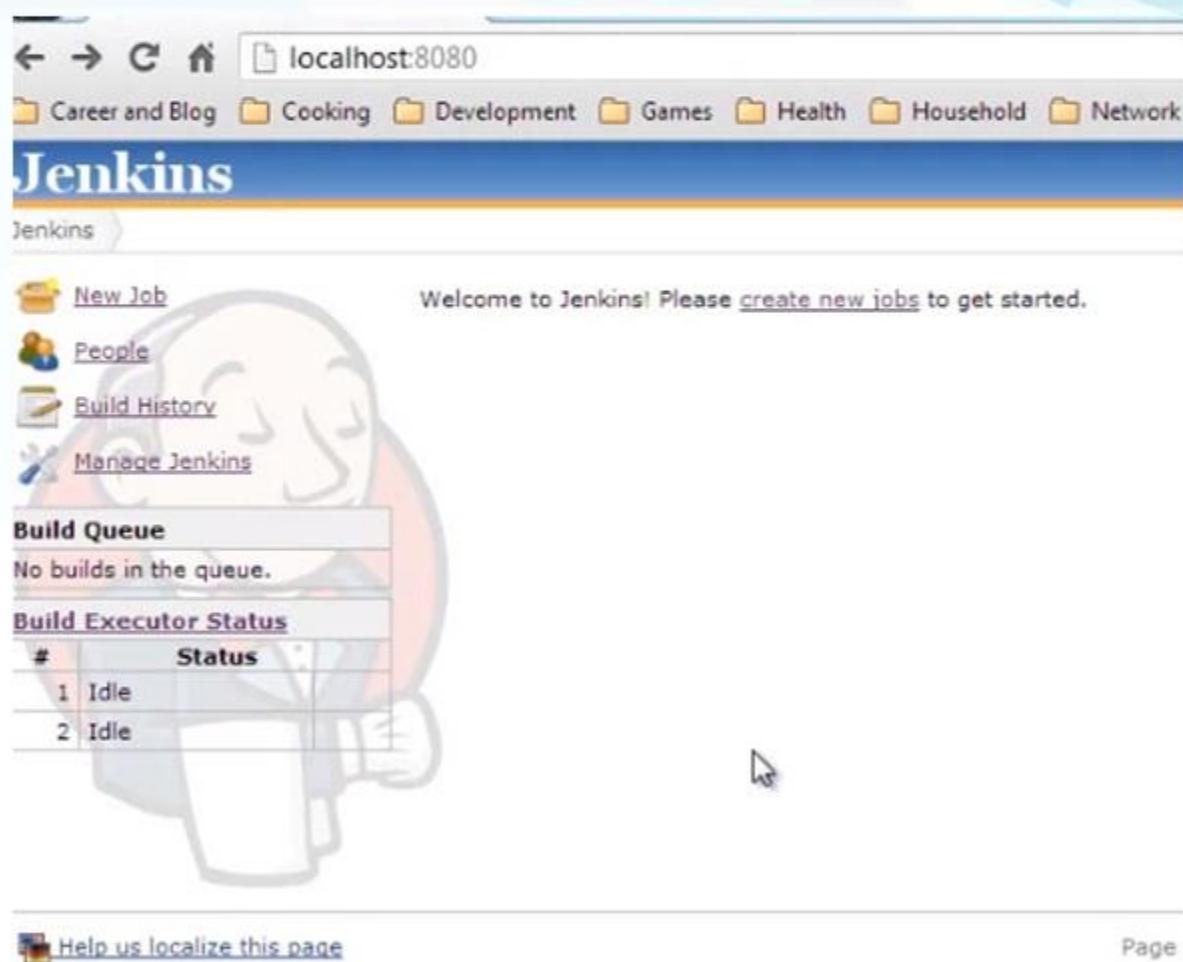
- ☛ Open the Zip file
- ☛ Run Jenkins.msi file  
(windows installer Package)
- ☛ Install the file
- ☛ We can check “Jenkins running as a Service”
- ☛ Restart System

Services (Local)

Jenkins	Name	Description	Status	Startup Type	Log On As
<a href="#">Stop the service</a>	Google Update Service (gupdate)	Keeps your ...	Automatic (D...)	Local Syst...	
<a href="#">Restart the service</a>	Google Update Service (gupdatem)	Keeps your ...	Manual	Local Syst...	
	Group Policy Client	The service ...	Started	Automatic	Local Syst...
	Health Key and Certificate Manage...	Provides X.5...	Manual	Local Syst...	
	HomeGroup Listener	Makes local...	Started	Manual	Local Syst...
	HomeGroup Provider	Performs ne...	Started	Manual	Local Service
	Human Interface Device Access	Enables gen...	Started	Manual	Local Syst...
	IKE and AuthIP IPsec Keying Modules	The IKEEXT ...	Started	Automatic	Local Syst...
	Intel(R) PROSet Monitoring Service	The Intel(R) ...	Started	Automatic	Local Syst...
	Intel(R) Rapid Storage Technology	Provides sto...	Started	Automatic (D...	Local Syst...
	Interactive Services Detection	Enables use...	Manual	Local Syst...	
	Internet Connection Sharing (ICS)	Provides ne...	Disabled	Local Syst...	
	IP Helper	Provides tu...	Started	Automatic	Local Syst...
	iPod Service	iPod hardw...	Started	Manual	Local Syst...
	IPsec Policy Agent	Internet Pro...	Started	Manual	Network S...
	Jenkins	Jenkins Con...	Started	Automatic	Local Syst...
	KtmRm for Distributed Transaction ...	Coordinates...	Manual	Network S...	
	Link-Layer Topology Discovery Map...	Creates a N...	Manual	Local Service	
	Logitech Bluetooth Service		Manual	Local Syst...	

# Alternate Method

- ☞ Go to manage Jenkins
- ☞ Select Install as windows service
- ☞ Select directory
- ☞ Click Install
- ☞ Restart System



# Nodes

- ☞ Jenkins server is called as Master and its various nodes are called as Slaves
- ☞ Ability to function as distributes build server.
- ☞ Make Jenkins more powerful by distributing builds across multiple machines. Provides ability to act like a cluster
- ☞ Important to think about nodes when you are designing.
- ☞ Makes build faster
- ☞ Ability to run a build on different platforms

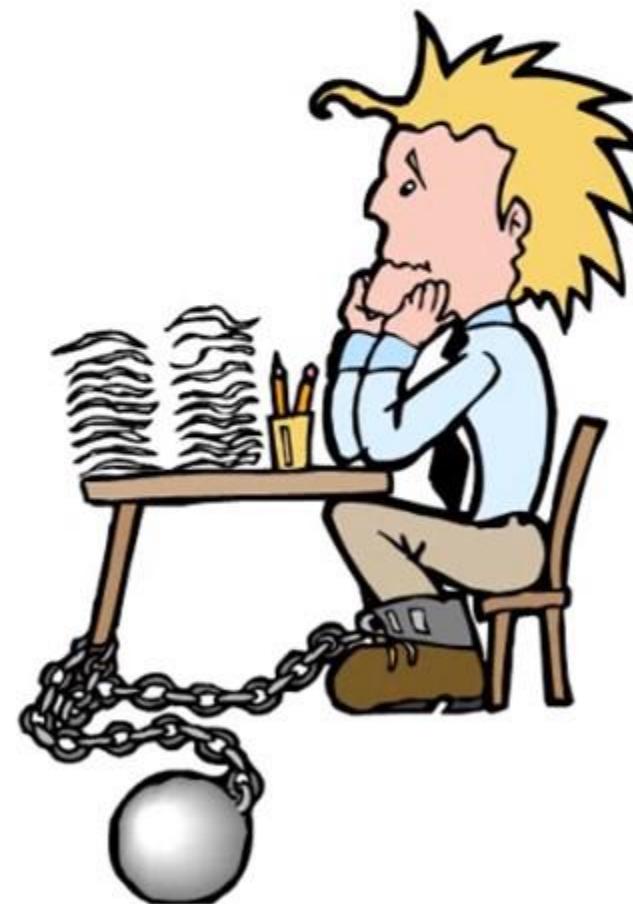
# Nodes



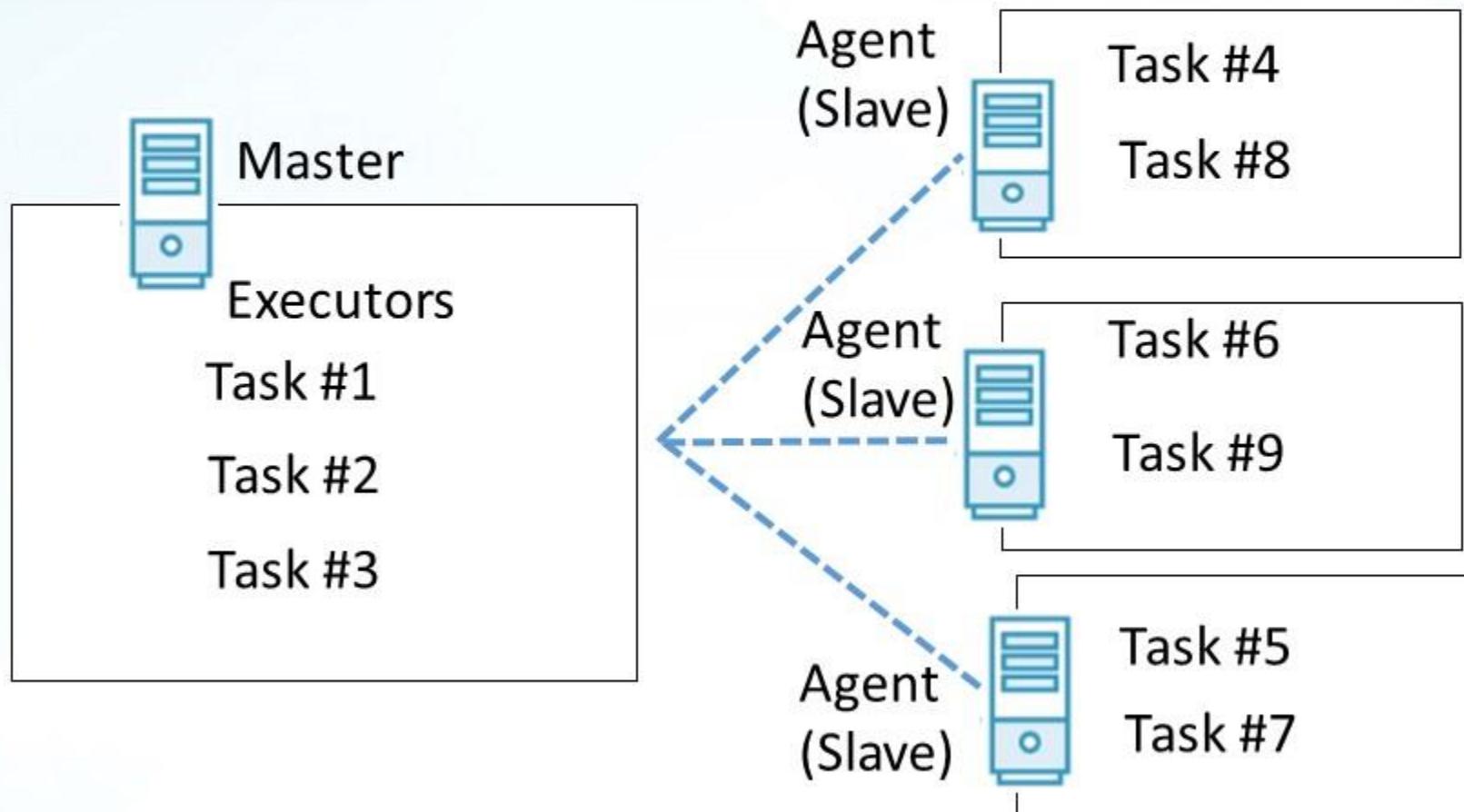
# Nodes



Run my automated tests, boy!



# The Master Agent Model



# Configuring a Node

Nodes [Jenkins]

localhost:8080/computer/?

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout Other bookmarks

search

ENABLE AUTO REFRESH

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	master	Windows 7 (x86)	In sync	50GB	49291MB	50GB	8ms

Back to Dashboard

Manage Jenkins

New Node

Configure

Build Queue

No builds in the queue.

Build Executor Status

#	Status
1	Idle
2	Idle

- ☞ Go to `localhost:8080`
- ☞ Go to `Manage Jenkins`
- ☞ Go to `Manage nodes`

# Configuring a Node

 Jenkins

[Jenkins](#) > [Nodes](#) >

[Back to Dashboard](#)

[Manage Jenkins](#)

[New Node](#)

[Configure](#)

Node name

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Build Queue =

No builds in the queue.

OK

Build Executor Status =

1 Idle

2 Idle

- ☞ Go to New nodes
- ☞ Give node name
- ☞ Select the node type
- ☞ Click Ok

## Configuring a Node

Screenshot of a web browser showing the Jenkins 'nodes' configuration page. The URL is `localhost:8080/computer/createItem`.

**Left Sidebar:**

- [Back to Dashboard](#)
- [Manage Jenkins](#)
- [New Node](#) (highlighted)
- [Configure](#)

**Build Queue:**  
No builds in the queue.

**Build Executor Status:**

#	Status
1	Idle
2	Idle

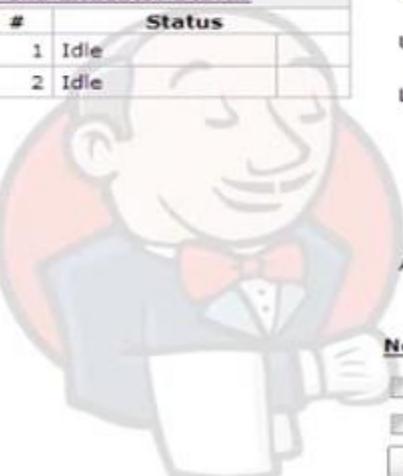
**Node Configuration Form:**

Name	<input type="text" value="Test Node"/>	<a href="#">?</a>
Description	<input type="text"/>	<a href="#">?</a>
# of executors	<input type="text" value="1"/> <span>Not a number</span>	<a href="#">?</a>
Remote FS root	<input type="text"/> <span>Remote directory is mandatory</span>	<a href="#">?</a>
Labels	<input type="text"/>	<a href="#">?</a>
Usage	<input type="text" value="Utilize this slave as much as possible"/>	<a href="#">?</a>
Launch method	<input type="text" value="Launch slave agents on Unix machines via SSH"/>	<a href="#">?</a>
Host	<input type="text"/>	<a href="#">Advanced...</a>
Availability	<input type="text" value="Keep this slave on-line as much as possible"/>	<a href="#">?</a>

**Node Properties:**

- Environment variables
- Tool Locations

[Save](#)



# Configuring a Node



Nodes [Jenkins] x localhost:8080/computer/ Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout Other bookmarks

## Jenkins

Back to Dashboard Manage Jenkins New Node Configure

Build Queue No builds in the queue.

Build Executor Status

#	Master
1	Idle
2	Idle
Test Node	(offline)

ENABLE AUTO REFRESH

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	master	Windows 7 (x86)	In sync	50GB	49291MB	50GB	8ms
2	Test Node		N/A	N/A	N/A	N/A	N/A

Help us localize this page Page generated: Jan 30, 2013 9:36:10 PM REST API Jenkins ver. 1.500

## Setting up a Node

Upgraded Machine - VMware Player (Non-commercial use only)

Player | Test Node [Jenkins] x

← → C H [mario:8080/computer/Test%20Node/](http://mario:8080/computer/Test%20Node/)

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout

### Jenkins

Jenkins > nodes > Test Node

 [Back to List](#)

[Status](#)

[Delete Slave](#)

[Configure](#)

[Build History](#)

[Load Statistics](#)

[Script Console](#)

[Log](#)

[System Information](#)

**Slave Test Node**

Connect slave to Jenkins one of these ways:

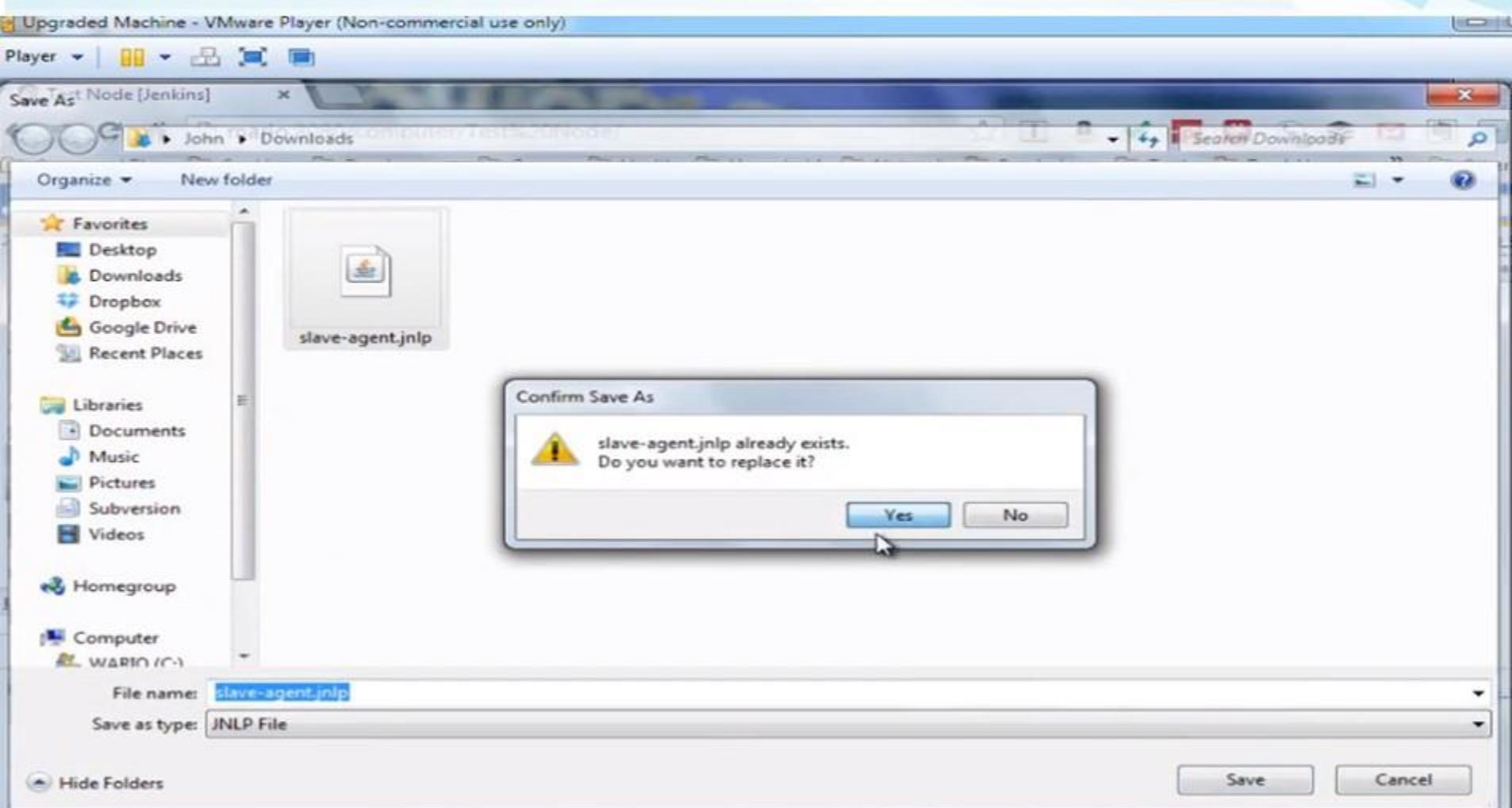
- [Launch](#) Launch agent from browser on slave
- Run from slave command line:  
`javaws http://mario:8080/computer/Test%20Node/slave-agent.jnlp`
- Or if the slave is headless:  
`java -jar slave.jar -jnlpUrl http://mario:8080/computer/Test%20Node/slave-agent.jnlp`

Created by anonymous user

#### Projects tied to Test Node

#	Status
	None

# Setting up a Node



## Setting up a Node

Player |   Test Node [Jenkins] x mario:8080/computer/Test%20Node/ Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbo... Jenkins search

Back to List Status Delete Slave Configure Build History Load Statistics Script Console Log System Information Build Executor Status Status None

Help us localize this page

Warning - Security

**The application's digital signature cannot be verified. Do you want to run the application?**

Name: Slave Agent for Test Node  
Publisher: CloudBees, Inc.

Always trust content from this publisher.

Run Cancel

Part of the application is missing a digital signature. Only run if you trust the origin of the application. More Information...

Page generated: Jan 30, 2013 9:38:43 PM

# Setting up a Node

Upgraded Machine - VMware Player (Non-commercial use only)

Player | Test Node [Jenkins] | mario:8080/computer/Test%20Node/ | Career and Blog | Cooking | D | Technology | Tools | TrackAbout | Other bookmarks

ENABLE AUTO REFRESH

Mark this node temporarily offline

## Jenkins

Jenkins nodes Test Node

Back to List

[Status](#)

[Delete Slave](#)

[Configure](#)

[Build History](#)

[Load Statistics](#)

[Script Console](#)

[Log](#)

[System Information](#)

Projects tied to Test Node

#	Status
1	Idle

Help us localize this page

Page generated: Jan 30, 2013 9:38:43 PM REST API Jenkins ver. 1.500

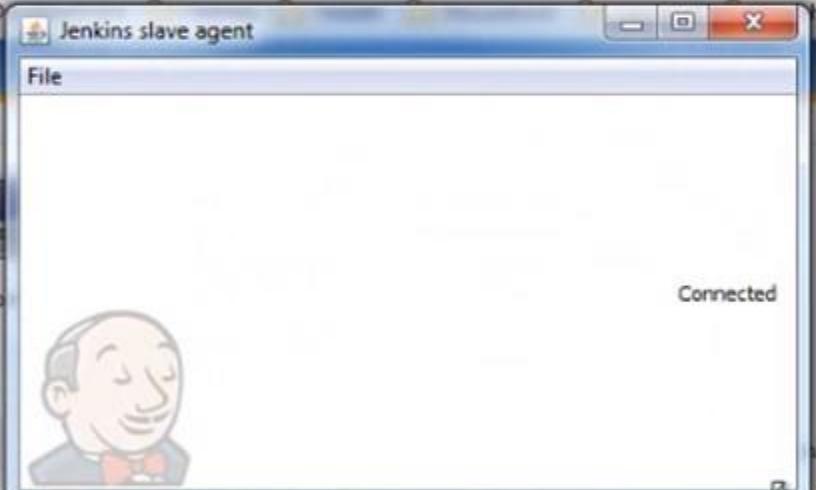
**Jenkins slave agent**

File

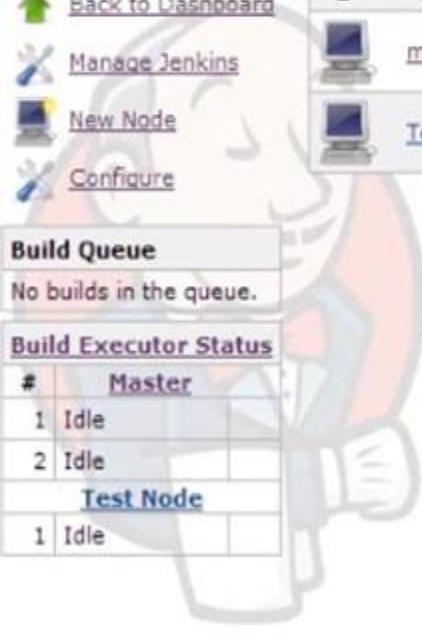
Connected

java -jar slave.jar -jnlpUrl http://mario:8080/computer/Test%20Node/slave-agent.jnlp

Created by anonymous user



# Setting up a Node



Nodes [Jenkins] x localhost:8080/computer/ Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout Other bookmarks

## Jenkins

nodes

ENABLE AUTO REFRESH

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	<a href="#">master</a>	Windows 7 (x86)	In sync	50GB	48619MB	50GB	4ms
	<a href="#">Test Node</a>	Windows 7 (x86)	In sync	29GB	12802MB	29GB	7ms

[Back to Dashboard](#)

[Manage Jenkins](#)

[New Node](#)

[Configure](#)

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Master
1	Idle
2	Idle
#	Test Node
1	Idle

[Help us localize this page](#)

Page generated: Jan 30, 2013 9:40:24 PM REST API Jenkins ver. 1.500

## Setting up a Node

A screenshot of a Jenkins slave agent window displayed in a web browser. The window title is 'Jenkins slave agent'. The content area shows a cartoon character of a man in a tuxedo, a status message 'Connected', and a command line instruction: 'java -jar slave.jar -jnlpUrl http://mario:8080/computer/Test%20Node/slave-agent.jnlp'. Below this, it says 'Created by anonymous user'. At the bottom, there is a section titled 'Projects tied to Test Node' with the subtext 'None'. The background shows the Jenkins master interface with a sidebar of options like 'Back to List', 'Status', 'Delete Slave', 'Configure', etc. The browser's address bar shows the URL 'mario:8080/computer/Test%20Node/'. The top of the image shows the VMware Player interface.

# Setting up a Node

Test Node [Jenkins] x

mario:8080/computer/Test%20Node/

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout Other bookmarks

search

ENABLE AUTO REFRESH

Mark this node temporarily offline

## Jenkins

Jenkins nodes Test Node

Back to List Status Delete Slave Configure Build History Load Statistics Script Console Log System Information

Slave

Connect slave to Jenkins

- Launch
- Run from slave
- Or if the slave has Java installed

Connected

Created by anonymous user agent.jnlp

Projects tied to Test Node

#	Status
1	Idle

Help us localize this page

Page generated: Jan 30, 2013 9:38:43 PM REST API Jenkins ver. 1.500

A screenshot of a Jenkins master node interface. The main page shows a 'Test Node' configuration. A modal window titled 'Jenkins slave agent' is open, showing a 'File' menu with an 'Install as a service' option highlighted. The Jenkins logo is visible in the background.

## UI Tour

localhost:8080

Career and Blog Cooking Development Games Health Household Network Psychology Tools

# Jenkins

New Job

People

Build History

Manage Jenkins

Welcome to Jenkins! Please [create new jobs](#) to get started.

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Master
1	Idle
2	Idle
<a href="#">Test Node (offline)</a>	

## UI Tour – New Job

New Job [Jenkins] x

localhost:8080/view/All/newJob

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout Other bookmarks

search

Jenkins All

New Job

People

Build History

Manage Jenkins

Build Queue

No builds in the queue.

Build Executor Status

#	Master
1	Idle
2	Idle

Test Node (offline)

Job name

- Build a free-style software project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Build a maven2/3 project**  
Build a maven2 project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Build multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Monitor an external job**  
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).

## UI Tour - People

People - [Jenkins] x

localhost:8080/asyncPeople/

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout Other bookmarks

**Jenkins** search ENABLE AUTO REFRESH

**People**

User Id	Name	Last Active	On
anonymous	anonymous	N/A	

Icon: S M L

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Master
1	Idle
2	Idle
Test Node (offline)	

## UI Tour – Build History

All [Jenkins] × localhost:8080/view/All/builds

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout Other bookmarks

search

ENABLE AUTO REFRESH

**Jenkins**

New Job

People

**Build History**

Manage Jenkins

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Master
1	Idle
2	Idle
Test Node (offline)	

Timeline  SINGLE

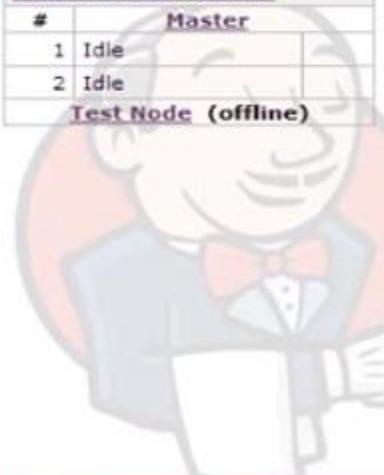
### Build History of Jenkins

Jan 29	Jan 30	Jan 31	Feb 1
20hr	21hr	22hr	23hr

Export as plain XML

Build	Time Since	Status
Icon: 		

Legend  RSS for all  RSS for failures  RSS for just latest builds



## UI Tour – Manage Jenkins

Manage Jenkins [Jenkins] x localhost:8080/manage

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout > C

**Jenkins** search ENABLE AUTO

New Job People Build History Manage Jenkins

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Master
1	Idle
2	Idle
<a href="#">Test Node (offline)</a>	

**Manage Jenkins**

- [Configure System](#)  
Configure global settings and paths.
- [Configure Global Security](#)  
Secure Jenkins; define who is allowed to access/use the system.
- [Reload Configuration from Disk](#)  
Discard all the loaded data in memory and reload everything from file system. Useful when you modified configuration directly on disk.
- [Manage Plugins](#)  
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. (**updates available**)
- [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 and see if you need more computers for your builds.
- [Jenkins CLI](#)  
Access/manage Jenkins from your shell, or from your script.
- [Script Console](#)  
Executes arbitrary script for administration/trouble-shooting/diagnostics.
- [Manage Nodes](#)  
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- [About Jenkins](#)  
See the version and license information.



## UI Tour – Configure System

Configure System [Jenkins] × localhost:8080/configure

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout

Simplify your life: Use LastPass to autofill in this site's login info! AutoFill

People Build History Manage Jenkins

**System Message**

# of executors: 2

Labels:

Usage: Utilize this slave as much as possible

Quiet period: 5

SCM checkout retry count: 0

Restrict project naming

**Global properties**

Environment variables

Tool Locations

**JDK**

JDK installations: Add JDK List of JDK installations on this system

**Ant**

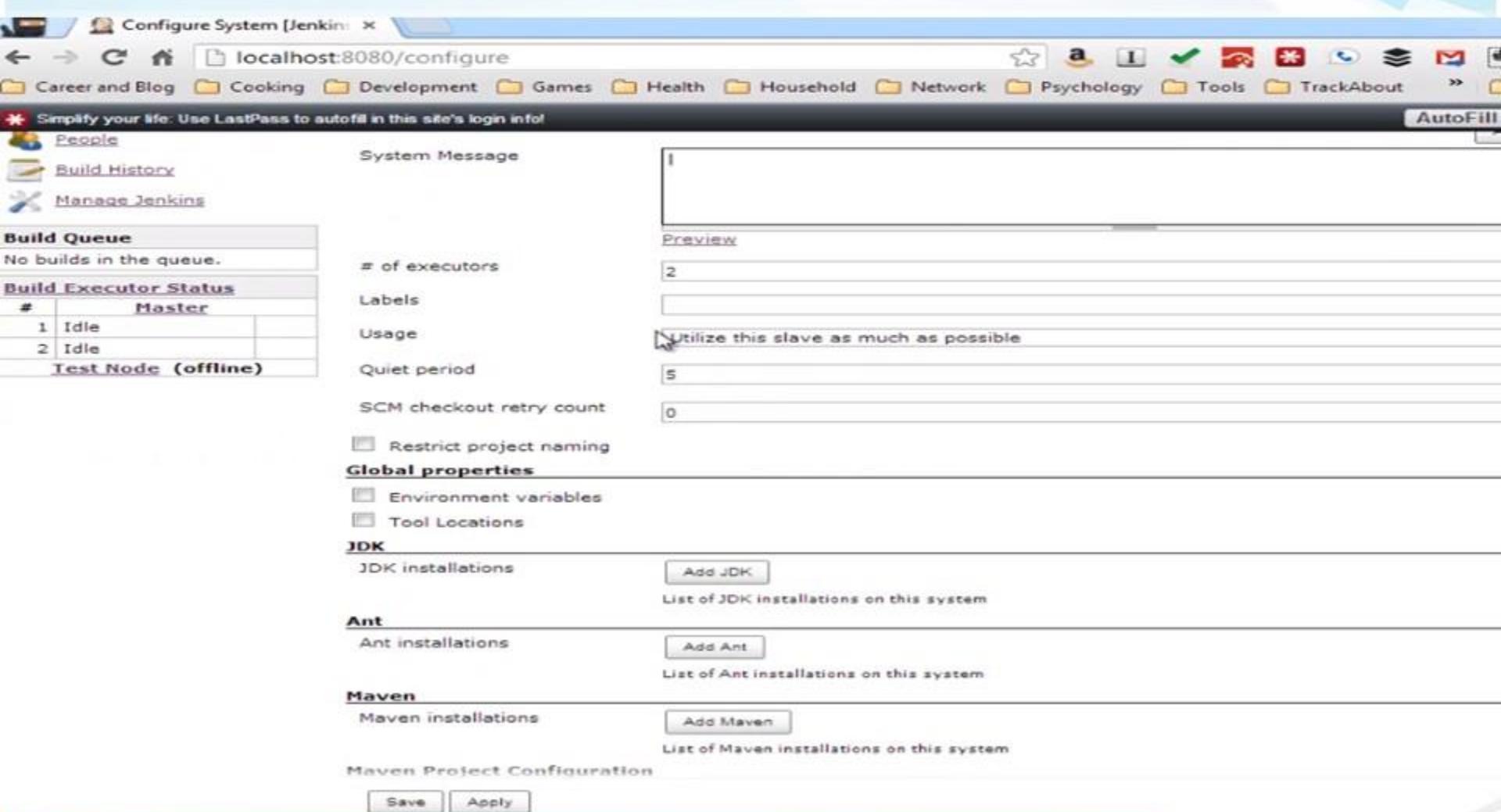
Ant installations: Add Ant List of Ant installations on this system

**Maven**

Maven installations: Add Maven List of Maven installations on this system

Maven Project Configuration

Save Apply



# UI Tour – Configure Global Security

Configure Global Security x localhost:8080/configureSecurity/?

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout AutoFill

Simplify your life: Use LastPass to autofill in this site's login info!

Jenkins Configure Global Security

## Configure Global Security

Enable security

TCP port for JNLP slave agents  Fixed :   Random  Disable

Markup Formatter

Treat the text as HTML and use it as is without any translation

Disable syntax highlighting

**Access Control**

**Security Realm**

Delegate to servlet container

Jenkins's own user database

LDAP

**Authorization**

Anyone can do anything

Legacy mode

Logged-in users can do anything

Matrix-based security

Project-based Matrix Authorization Strategy

Prevent Cross Site Request Forgery exploits



## UI Tour

localhost:8080/manage

ENABLE AUTO REFRESH

**Build History**

**Manage Jenkins**

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Master
1	Idle
2	Idle
<a href="#">Test Node (offline)</a>	

**Configure System**  
Configure global settings and paths.

**Configure Global Security**  
Secure Jenkins; define who is allowed to access/use the system.

**Reload Configuration from Disk**  
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

**Manage Plugins**  
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. (**updates available**)

**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 and see if you need more computers for your builds.

**Jenkins CLI**  
Access/manage Jenkins from your shell, or from your script.

**Script Console**  
Executes arbitrary script for administration/trouble-shooting/diagnostics.

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



# UI Tour – System Properties

localhost:8080/systemInfo

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout

**Jenkins** search ENABLE AUTOMATION

New Job People Build History Manage Jenkins

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Master
1	Idle
2	Idle

**Test Node (offline)**

## System Properties

Name	Value
awt.toolkit	sun.awt.windows.WToolkit
executable-war	C:\WIP\jenkins\workspace\jenkins.war
file.encoding	Cp1252
file.encoding.pkg	sun.io
file.separator	\
hudson.diyChunking	true
java.awt.graphicsenv	sun.awt.Win32GraphicsEnvironment
java.awt.headless	true
java.awt.printerjob	sun.awt.windows.WPrinterJob
java.class.path	jenkins.war
java.class.version	51.0
java.endorsed.dirs	C:\Program Files (x86)\Java\jre7\lib\endorsed
java.ext.dirs	C:\Program Files (x86)\Java\jre7\lib\ext;C:\Windows\Sun\Java\lib\ext
java.home	C:\Program Files (x86)\Java\jre7
java.io.tmpdir	C:\Users\John\AppData\Local\Temp\
java.library.path	C:\Program Files (x86)\Java\jre7\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows
java.runtime.name	Java(TM) SE Runtime Environment
java.runtime.version	1.7.0_11-b21
java.specification.name	Java Platform API Specification
java.specification.vendor	Oracle Corporation
java.specification.version	1.7
java.vendor	Oracle Corporation
java.vendor.url	http://java.oracle.com/
java.vendor.url.bug	http://bugreport.sun.com/bugreport/
java.version	1.7.0_11

# UI Tour – System Log

localhost:8080/log/all

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout

search

ENABLE AUTOMATION

[Back to Dashboard](#)

[Manage Jenkins](#)

[Logger List](#)

[All Logs](#)

[New Log Recorder](#)

[Log Levels](#)

## Jenkins Log

```
Jan 30, 2013 9:41:54 PM jenkins.slaves.JnlpSlaveAgentProtocol$Handler$1 onClosed
WARNING: Channel reader thread: Test Node for + Test Node terminated
java.net.SocketException: Connection reset
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at java.io.BufferedInputStream.fill(Unknown Source)
    at java.io.BufferedInputStream.read(Unknown Source)
    at java.io.ObjectInputStream$PeekInputStream.peek(Unknown Source)
    at java.io.ObjectInputStream$BlockDataInputStream.peek(Unknown Source)
    at java.io.ObjectInputStream$BlockDataInputStream.peekByte(Unknown Source)
    at java.io.ObjectInputStream.readObject0(Unknown Source)
    at java.io.ObjectInputStream.readObject(Unknown Source)
    at hudson.remoting.Command.readFrom(Command.java:92)
    at hudson.remoting.ClassicCommandTransport.read(ClassicCommandTransport.java:59)
    at hudson.remoting.SynchronousCommandTransport$ReaderThread.run(SynchronousCommandTransport.java:100)

Jan 30, 2013 9:41:54 PM hudson.remoting.SynchronousCommandTransport$ReaderThread run
SEVERE: I/O error in channel Test Node
java.net.SocketException: Connection reset
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at java.io.BufferedInputStream.fill(Unknown Source)
    at java.io.BufferedInputStream.read(Unknown Source)
    at java.io.ObjectInputStream$PeekInputStream.peek(Unknown Source)
    at java.io.ObjectInputStream$BlockDataInputStream.peek(Unknown Source)
    at java.io.ObjectInputStream$BlockDataInputStream.peekByte(Unknown Source)
    at java.io.ObjectInputStream.readObject0(Unknown Source)
    at java.io.ObjectInputStream.readObject(Unknown Source)
    at hudson.remoting.Command.readFrom(Command.java:92)
    at hudson.remoting.ClassicCommandTransport.read(ClassicCommandTransport.java:59)
    at hudson.remoting.SynchronousCommandTransport$ReaderThread.run(SynchronousCommandTransport.java:100)

Jan 30, 2013 9:40:05 PM hudson.TcpSlaveAgentListener$ConnectionHandler run
INFO: Accepted connection #1 from /192.168.1.6:50918
Jan 30, 2013 9:24:33 PM hudson.WebAppMain$2 run
INFO: Jenkins is fully up and running

```

## UI Tour – Load Statistics

Career and Blog   Cooking   Development   Games   Health   Household   Network   Psychology   Tools   TrackAbout   >   C

### Jenkins

search   [ENABLE AUTO](#)

[New Job](#)  
[People](#)  
[Build History](#)  
[Manage Jenkins](#)

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Master
1	Idle
2	Idle
<a href="#">Test Node (offline)</a>	

**Load statistics: Jenkins**

Timespan: [Short](#) [Medium](#) [Long](#)

Date	Total executors	Busy executors	Queue length
2124	0	0	0
2125	0.2	0	0
2126	0.4	0	0
2127	0.6	0	0
2128	0.8	0	0
2129	1.0	0	0
2130	1.2	0	0
2131	1.4	0	0
2132	1.6	0	0
2133	1.8	0	0
2134	2.0	0	0
2135	2.2	0	0
2136	2.4	0	0
2137	2.6	0	0
2138	2.8	0	1
2139	2.8	0	1
2140	2.9	0	0
2141	2.9	0	0
2142	2.9	0	0
2143	2.9	0	0
2144	2.9	0	0
2145	2.9	0	0
2146	2.9	0	0

Load statistics keep track of three key metrics of resource utilization:

**Total number of executors**  
For a computer, this is the number of executors that the computer has. For a label, this is the sum of all executors across computers in this label. For the entire Jenkins, this is the sum of all executors across all computers in this Jenkins instance. Other than configuration changes, this value can also change when slaves go offline.

**Number of busy executors**  
This line tracks the number of executors (among the executors counted above) that are carrying out builds. The ratio of the total number of executors gives you the resource utilization. If all your executors are busy for a prolonged period, consider adding more computers to your Jenkins cluster.

**Queue length**  
This is the number of jobs that are in the build queue, waiting for an available executor (of this computer, of this label, Jenkins, respectively). This doesn't include jobs that are in the quiet period, nor does it include jobs that are in the

# UI Tour – Jenkins CLI

Career and Blog   Cooking   Development   Games   Health   Household   Network   Psychology   Tools   TrackAbout   >   C

## Jenkins

search

New Job   People   Build History   Manage Jenkins

Build Queue  
No builds in the queue.

Build Executor Status

#	Master
1	Idle
2	Idle
Test Node (offline)	

### Jenkins CLI

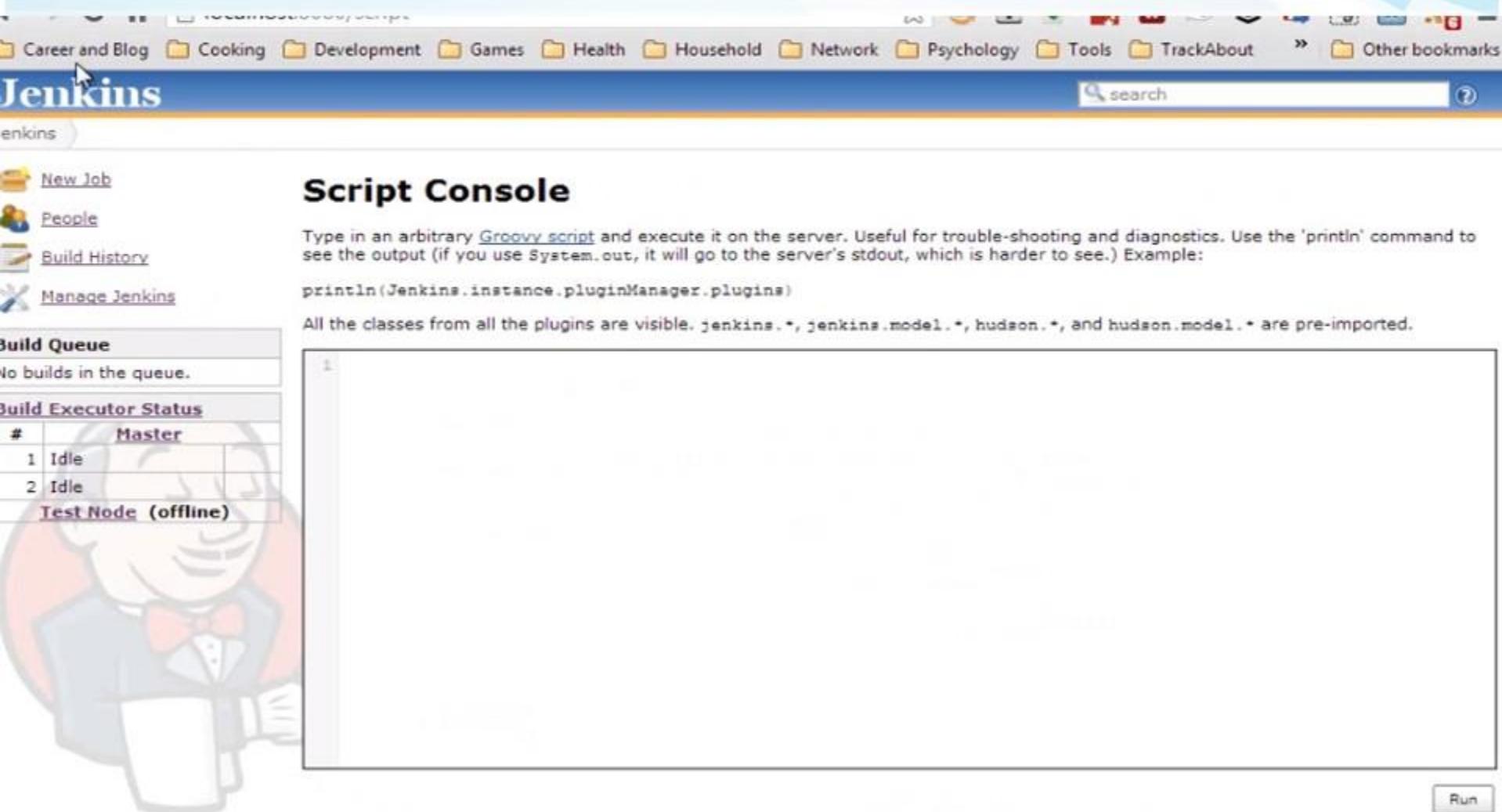
You can access various features in Jenkins through a command-line tool. See the [Wiki](#) for more details of this feature. To get started, download [jenkins-cli.jar](#), and run it as follows:

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

#### Available Commands

- **build**: Builds a job, and optionally waits until its completion.
- **cancel-quiet-down**: Cancel the effect of the "quiet-down" command.
- **clear-queue**: Clears the build queue
- **connect-node**: Reconnect to a node
- **console**: Retrieves console output of a build
- **copy-job**: Copies a job.
- **create-job**: Creates a new job by reading stdin as a configuration XML file.
- **delete-builds**: Deletes build record(s).
- **delete-job**: Deletes a job
- **delete-node**: Deletes a node
- **disable-job**: Disables a job
- **disconnect-node**: Disconnects from a node
- **enable-job**: Enables a job
- **get-job**: Dumps the job definition XML to stdout
- **groovy**: Executes the specified Groovy script.
- **groovysh**: Runs an interactive groovy shell.
- **help**: Lists all the available commands.
- **install-plugin**: Installs a plugin either from a file, an URL, or from update center.
- **install-tool**: Performs automatic tool installation, and print its location to stdout. Can be only called from inside a build.
- **keep-build**: Mark the build to keep the build forever.
- **list-changes**: Dumps the changelog for the specified build(s).
- **list-jobs**: Lists all jobs in a specific view or item group.
- **list-plugins**: Outputs a list of installed plugins.
- **login**: Saves the current credential to allow future commands to run without explicit credential information.
- **logout**: Deletes the credential stored with the login command.
- **mail**: Reads stdin and sends that out as an e-mail.
- **offline-node**: Stop using a node for performing builds temporarily, until the next "online-node" command.
- **online-node**: Resume using a node for performing builds, to cancel out the earlier "offline-node" command.
- **quiet-down**: Quiet down Jenkins, in preparation for a restart. Don't start any builds.
- **reload-configuration**: Discard all the loaded data in memory and reload everything from file system. Useful when...

# UI Tour – Script Console



Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout » Other bookmarks

## Script Console

Type in an arbitrary [Groovy script](#) and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

```
1
```

Run

# UI Tour – Script Console

## Get-current-queue-state

```
/** BEGIN META {
  "name" : "Get Queue State",
  "comment" : "Get current Queue/Executors state. Useful to troubleshoot queue and check that
tasks are correctly handled",
  "parameters" : [ ],
  "core": "1.642",
  "authors" : [
    { name : "Allan Burdajewicz" }
  ]
} END META**/

def all = [ Jenkins.instance ]
all.addAll(Jenkins.instance.nodes)
all.each {
  def c = it.toComputer()
  println "[${it.nodeName} - ${it.numExecutors} - ${it.assignedLabels} - ${it.acceptingTasks} -
${it.nodeProperties} - ${c.offline}"
}
println "----"
Jenkins.instance.queue.items.each {
  println "${it.id} ${it.blocked} ${it.buildable} ${it.stuck} ${it.assignedLabel} ${it.causes}"
}
```

# UI Tour – About Jenkins

## Jenkins

 search[Jenkins](#) > [About Jenkins](#)

### About Jenkins 1.500

[Jenkins](#) is a community-developed open-source continuous integration server.

Jenkins depends on the following 3rd party libraries.

Name	Maven ID	
<a href="#">Jenkins.war</a>	org.jenkins-ci.main:jenkins-war:1.500	<a href="#">The MIT license</a>
<a href="#">Commons BeanUtils</a>	commons-beanutils:commons-beanutils:1.8.3	<a href="#">The Apache Software License</a>
<a href="#">Apache Ant Launcher</a>	org.apache.ant:ant-launcher:1.8.3	<a href="#">The Apache Software License</a>
<a href="#">Bouncy Castle Provider</a>	org.bouncycastle:bcprov-jdk15on:1.47	<a href="#">Bouncy Castle Software License</a>
<a href="#">oro</a>	oro:oro:2.0.8	<a href="#">The Apache Software License</a>
<a href="#">Jenkins CLI</a>	org.jenkins-ci.main:cli:1.500	<a href="#">The MIT license</a>
<a href="#">StAX API</a>	stax:stax-api:1.0.1	<a href="#">The Apache Software License</a>
<a href="#">jinterop-proxy</a>	org.kohsuke.jinterop:jinterop-proxy:1.1	<a href="#">MIT license</a>
<a href="#">Stapler</a>	org.kohsuke.stapler:stapler:1.199	<a href="#">2-clause BSD License</a>
<a href="#">SSH CLI client authenticator</a>	org.jenkins-ci.modules:ssh-cli-auth:1.2	<a href="#">MIT License</a>
<a href="#">memory-monitor</a>	org.jenkins-ci:memory-monitor:1.7	<a href="#">MIT License</a>
<a href="#">commons-jelly-tags-xml</a>	commons-jelly:commons-jelly-tags-xml:1.1	<a href="#">The Apache Software License</a>
<a href="#">CodeMirror library from <a href="http://codemirror.net/">http://codemirror.net/</a></a>	org.kohsuke.stapler:stapler-adjunct-codemirror:1.1	<a href="#">MIT License</a>
<a href="#">Spring Framework: DAO</a>	org.springframework:spring-dao:1.2.9	<a href="#">The Apache Software License</a>
<a href="#">JLine</a>	jline:jline:0.9.94	<a href="#">BSD</a>
<a href="#">Java binding for libpam.so</a>	org.kohsuke:libpam4j:1.6	<a href="#">The MIT license</a>
<a href="#">Spring Framework: Context</a>	org.springframework:spring-context:2.5	<a href="#">The Apache Software License</a>
<a href="#">HttpClient</a>	commons-httpclient:commons-httpclient:3.1	<a href="#">Apache License 2.0</a>
<a href="#">Woodstox</a>	org.codehaus.woodstox:wstx-asl:3.2.9	<a href="#">The Apache Software License</a>
<a href="#">bcrypt</a>	org.mindrot:bcrypt:0.3m	<a href="#">ISC/BSD License</a>
<a href="#">Apache Ant Core</a>	org.apache.ant:ant:1.8.3	<a href="#">The Apache Software License</a>
<a href="#">libzfs-java</a>	org.jvnet.libzfs:libzfs:0.5	<a href="#">COMMON DEVELOPMENT ENVIRONMENT</a>
<a href="#">Gantmed SSH2 for Java</a>	org.jenkins-ci:trilead-ssh2:build214-jenkins-1	<a href="#">BSD style license</a>
<a href="#">Stapler Jelly module</a>	org.kohsuke.stapler:stapler-jelly:1.199	<a href="#">2-clause BSD License</a>

# UI Tour

Jenkins

Build Executor Status		
#	Master	ENABLING
1	Idle	
2	Idle	

**Test Node (offline)**



[Discard All Data](#)

Discard all the loaded data in memory and reload everything from file system. Useful when you modify the configuration.



[Manage Plugins](#)

Add, remove, disable or enable plugins that can extend the functionality of Jenkins. **(updates available)**



[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 and see if you need more computers for your builds.



[Jenkins CLI](#)

Access/manage Jenkins from your shell, or from your script.



[Script Console](#)

Executes arbitrary script for administration/trouble-shooting/diagnostics.



[Manage Nodes](#)

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



[About Jenkins](#)

See the version and license information.



[Manage Old Data](#)

Scrub configuration files to remove remnants from old plugins and earlier versions.



[Install as Windows Service](#)

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



[Prepare for Shutdown](#)

Stop executing new builds, so that the system can be eventually shut down safely.

# UI Tour – Build Queue

## Jenkins

Jenkins

 [New Job](#)

 [People](#)

 [Build History](#)

 [Manage Jenkins](#)

 [Build Queue](#)

No builds in the queue.

 [Build Executor Status](#)

#	Master
1	Idle
2	Idle

**Test Node (offline)**

## Manage Jenkins



### [Configure System](#)

Configure global settings and paths.



### [Configure Global Security](#)

Secure Jenkins; define who is allowed to access/use the system.



### [Reload Configuration from Disk](#)

Discard all the loaded data in memory and reload everything from file system directly on disk.



### [Manage Plugins](#)

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



### [System Information](#)

Displays various environmental information to assist trouble-shooting.



### [System Log](#)

# UI Tour – Home Page Auto Refresh

## Jenkins

Jenkins

search

ENABLE AUTO REFRESH

 [add desc...](#)

 [New Job](#)

Welcome to Jenkins! Please [create new jobs](#) to get started.

 [People](#)

 [Build History](#)

 [Manage Jenkins](#)

### Build Queue

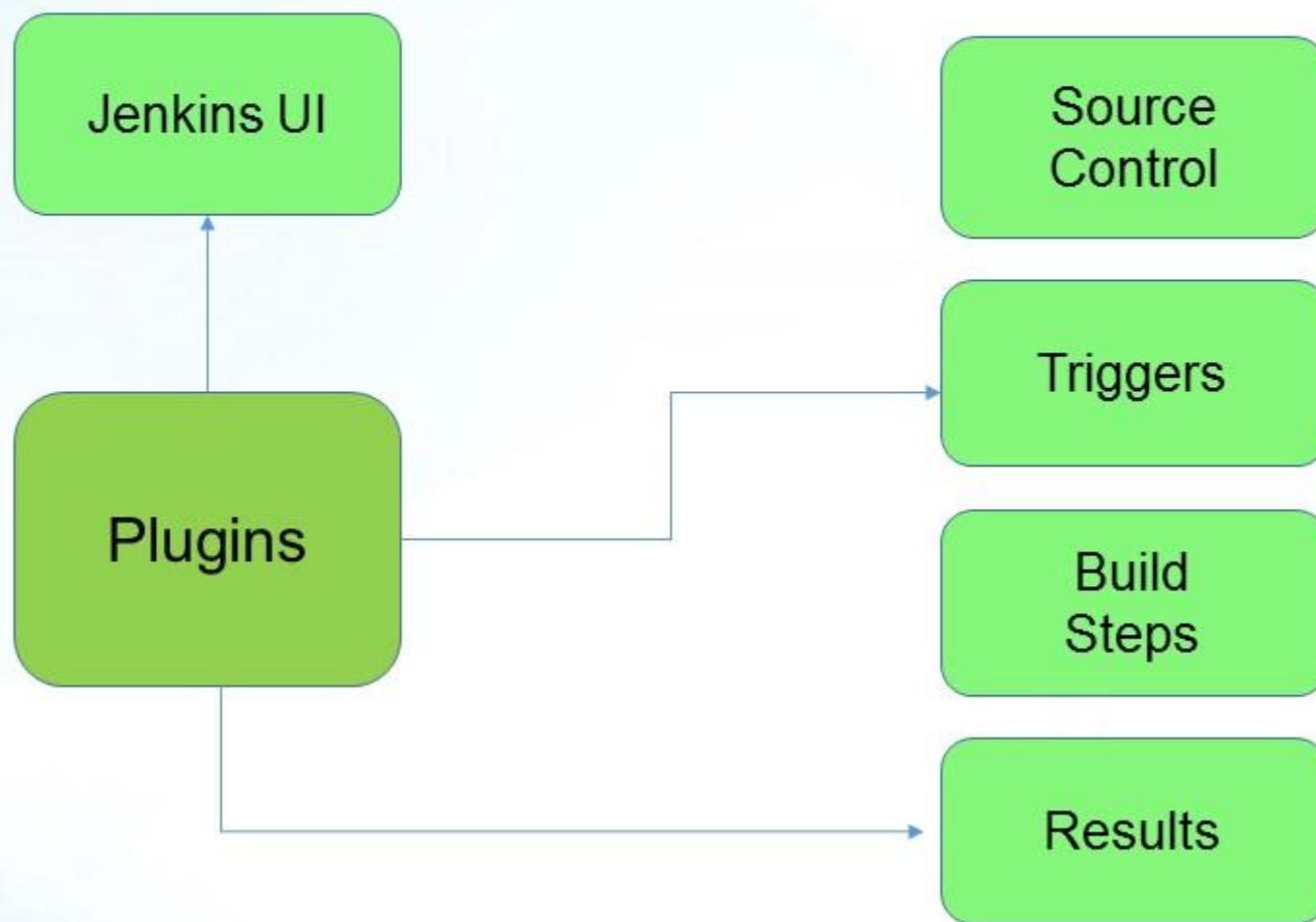
No builds in the queue.

### Build Executor Status

#	Master
1	Idle
2	Idle
	<a href="#">Test Node (offline)</a>

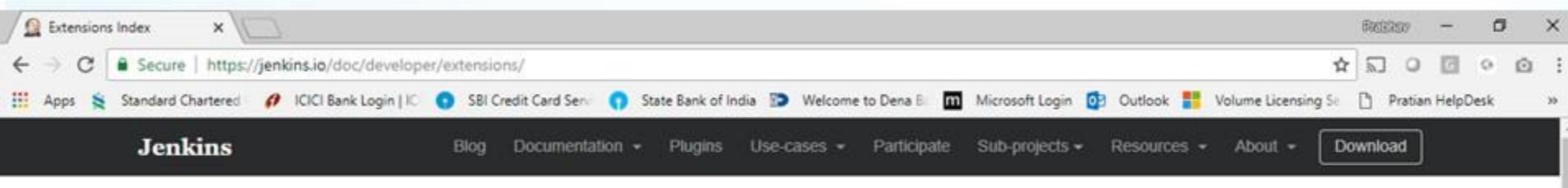
# Jenkins Plugins

# Plugin Architecture



# Extension Points

<https://jenkins.io/doc/developer/extensions/>



Extensions Index

Secure | <https://jenkins.io/doc/developer/extensions/>

Apps Standard Chartered ICICI Bank Login | IC SBI Credit Card Serv State Bank of India Welcome to Dena B Microsoft Login Outlook Volume Licensing Se Pratian HelpDesk

Jenkins

Blog Documentation Plugins Use-cases Participate Sub-projects Resources About Download

## Extensions Index

Jenkins defines extension points, which are interfaces or abstract classes that model an aspect of its behavior. Those interfaces define contracts of what need to be implemented, and Jenkins allows plugins to contribute those implementations. In general, all you need to do to register an implementation is to mark it with `@Extension`. Creating a new extension point is quite easy too; see [defining a new extension point](#) for details.

This index has been generated automatically. Javadoc excerpts are taken from core and plugin source code, and may have been improperly converted to HTML, so some may appear broken.

- Extension points defined in Jenkins Core
- Extension points defined in All changes Plugin
- Extension points defined in Android Signing Plugin
- Extension points defined in Apica Loadtest Plugin
- Extension points defined in Artifact Manager on S3 Plugin
- Extension points defined in Attention Plugin
- Extension points defined in Audit to Database Plugin
- Extension points defined in Audit Trail Plugin
- Extension points defined in Authentication Tokens API Plugin
- Extension points defined in Authorize Project Plugin
- Extension points defined in Autocomplete Parameter Plugin
- Extension points defined in Bitbucket Branch Source Plugin
- Extension points defined in Bitbucket Pipeline for Blue Ocean Plugin
- Extension points defined in Block Queued Job Plugin
- Extension points defined in Blue Ocean Pipeline Editor Plugin
- Extension points defined in Branch API Plugin
- Extension points defined in Build Alias Setter Plugin

# Plugin Wiki

Plugins - Jenkins - Jenkins

<https://wiki.jenkins-ci.org/display/JENKINS/Plugins>

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout Other bookmarks

Dashboard > Jenkins > Plugins

Browse Search

 **Plugins**  Edit  Add  Tools

1 Added by [Kohsuke Kawaguchi](#), last edited by [Kohsuke Kawaguchi](#) on Jan 30, 2013 [\(view change\)](#)

**Jenkins**

- Home
- Mailing lists
- Source code
- Bugtracker
- Security Advisories
- Donation
- Commercial Support
- Wiki Site Map

**Documents**

- Meet Jenkins
- Use Jenkins
- Extend Jenkins
- Plugins
- Servlet Container Notes

1 [How to install plugins](#)  
1.1 [Using the interface](#)  
1.2 [By hand](#)  
2 [Getting notified of plugin releases](#)  
3 [Developers](#)  
4 [Plugins by topic](#)  
4.1 [Source code management](#)  
4.2 [Build triggers](#)  
4.3 [Build tools](#)   
4.4 [Build wrappers](#)  
4.5 [Build notifiers](#)  
4.6 [Slave launchers and controllers](#)  
4.7 [Build reports](#)  
4.8 [Artifact uploaders](#)  
4.9 [Other post-build actions](#)  
4.10 [External site/tool integrations](#)  
4.11 [UI plugins](#)  
4.12 [List View column plugins](#)  
4.13 [Page decorators](#)  
4.14 [Authentication and user management](#)  
4.15 [Cluster management and distributed build](#)  
4.16 [CLI extensions](#)  
4.17 [Maven](#)  
4.18 [Parameters](#)  
4.19 [iOS development](#)  
4.20 [.NET development](#)

**Click on any of the plugin to see details.**

# Getting Plugins

## Jenkins

Jenkins Plugin Manager

 [Back to Dashboard](#)

 [Manage Jenkins](#)

 [search](#)



Updates	Available	Installed	Advanced
Install	Name	Version	Installed
<input type="checkbox"/>	<a href="#">LDAP Plugin</a> Security realm based on LDAP authentication.	1.2	1.1
<input type="checkbox"/>	<a href="#">CVS Plugin</a> This bundled plugin integrates Jenkins with CVS version control system.	2.7	1.6
<input type="checkbox"/>	<a href="#">Subversion Plugin</a> This plugin adds the Subversion support (via SVNKit) to Jenkins.	1.45	1.39
<input type="checkbox"/>	<a href="#">Translation Assistance Plugin</a> This plugin adds an additional dialog box in every page, which enables people to contribute localizations for the messages they are seeing in the current page.	1.10	1.8



[Install without restart](#)

[Download now and install after restart](#)

# Getting Plugins

Updates	Available	Installed	Advanced
Install	Name	Vers	
	Artifact Uploaders		
	Authentication and User Management		
	Build Notifiers		
	Build Reports		
	Build Tools		
	Build Triggers		
	Build Wrappers		
	Cluster Management and Distributed Build		
	Command Line Interface		
	External Site/Tool Integrations		
	List view columns		
	Maven		
	Misc (android)		
<input type="checkbox"/>	<a href="#">Android Emulator Plugin</a>	2.8	
	Lets you automatically generate, launch and interact with an <a href="#">Android</a> emulator during a build, with the emulator logs being captured as artifacts.		
<input type="checkbox"/>	<a href="#">Android Lint Plugin</a>	2.0.2	
	Parses output from the Android lint tool and displays the results for analysis.		
<input type="checkbox"/>	<a href="#">Appaloosa Plugin</a>	1.3.1	
	Publish your mobile applications (Android, iOS, ...) to the <a href="#">appaloosa-store.com</a> platform.		
<input type="checkbox"/>	<a href="#">External Resource Dispatcher</a>	1.0b	
	This plugin adds support for external resources in Jenkins. An external resource is something external attached to a Jenkins slave and can be locked by a build, which thus gets exclusive access to it, then		

# Upload Plugin



Update Center [Jenkins] × [localhost:8080/pluginManager/advanced](http://localhost:8080/pluginManager/advanced)

User name

Password

No Proxy Host

### Upload Plugin

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

File:  No file chosen

### Update Site

URL

# Useful Plugins Overview



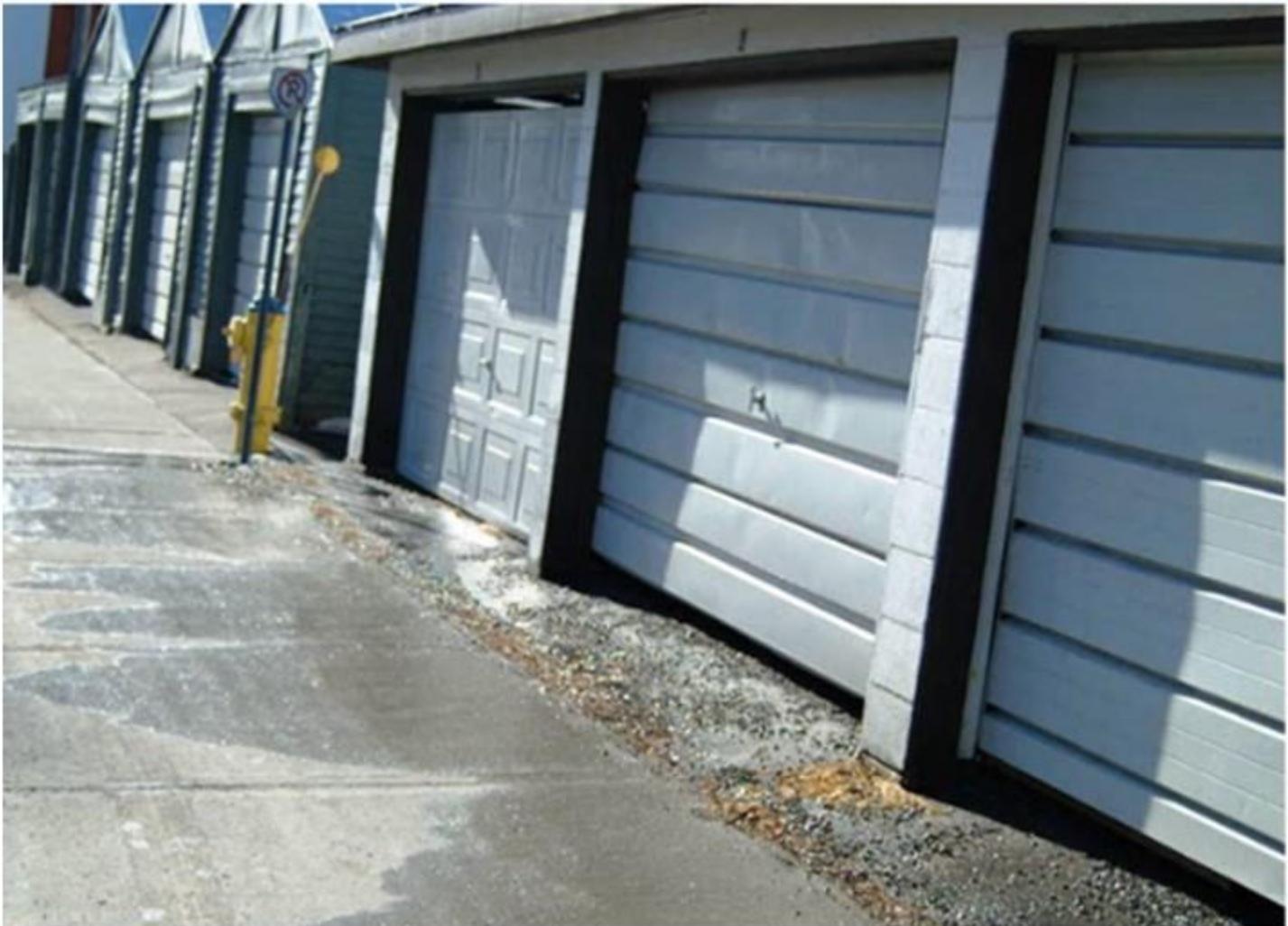
# Source Code Plugins

☞ Git

☞ Mecurial

☞ BitBucket

☞ TFS



# Triggers Plugins

☞ Github Pull Request

☞ Join

☞ Locks & Latches



# Build Tool Plugins

- ☛ Copy artifacts
- ☛ Fitness
- ☛ MSBuild
- ☛ MSTest Runner
- ☛ Nant
- ☛ PowerShell
- ☛ Promoted Builds



# Wrapper Plugins

- ☞ Selenium RC
- ☞ VMWare
- ☞ VirtualBox



## Notifier Plugins

- ☞ Hipchat
- ☞ IRC
- ☞ Twitter
- ☞ Jabber



Calling to let you know, you  
broke the build!

# Reporting Plugins

- ☞ Cobertura – Java Code Coverage
- ☞ Findbugs – Java Static Analysis
- ☞ MSTest - MSTest reporting
- ☞ Ncover - .Net code coverage
- ☞ PMD – Static analysis Java code

REPORT CARD					
GRADING PERIOD	1	2	3	4	
READING	A				
WRITTEN COMMUNICATION	A				
MATHEMATICS	C				
SCIENCE/HEALTH	B				
SOCIAL STUDIES	B				
ART	A				
MUSIC	A				
PHYSICAL EDUCATION	C				
Grade Average	B				
Attendance:	Present	48			
	Absent	0			
	Tardy	1			
A = Excellent • B = Good • C = Satisfactory • N = Needs Improvement U = Unsatisfactory • I = Insufficient / Incomplete					
Student: _____ Grade: _____ Year: _____					

## Artifacts / UI

- ☞ Artifact deployer
- ☞ Deploy
- ☞ Chunk Norris
- ☞ Emotional Jenkins
- ☞ Claim
- ☞ CI Game



# Installing a Plugin

chuck norris

2 of 3



<input type="checkbox"/>	<a href="#">ViewVC Plugin</a> This plugin integrates <a href="#">ViewVC</a> browser interface for CVS and Subversion with Hudson.	1.5
<b>Uncategorized</b>		
<input type="checkbox"/>	<a href="#">Selection Tasks Plugin</a> This plugin adds new variable. You may select several tasks (for example, Selenium tests) for run.	1.0
<input type="checkbox"/>	<a href="#">SeleniumRC Plugin</a> This plugin allows you to create Selenium server instance for each project build.	1.0
<b>User Interface</b>		
<input type="checkbox"/>	<a href="#">All Changes Plugin</a> Shows all changes which influenced the builds of a project.	1.3
<input type="checkbox"/>	<a href="#">BruceSchneier Plugin</a> Displays a picture of Bruce Schneier (instead of Jenkins the butler) and a random Bruce Schneier fact on each build page.	0.1
<input type="checkbox"/>	<a href="#">Build Trigger Badge Plugin</a> This plugin displays icon(s) representing the cause(s) of a build directly in the build history. It lets you quickly know which cause triggered a build.	0.9
<input checked="" type="checkbox"/>	<a href="#">ChuckNorris Plugin</a> Displays a picture of <a href="#">Chuck Norris</a> (instead of Jenkins the butler) and a random <a href="#">Chuck Norris</a> 'The Programmer' fact on each build page.	0.5
<input type="checkbox"/>	<a href="#">Collapsing Console Sections Plugin</a> This plugin allows the creation of sections in build consoles.	1.3
<input type="checkbox"/>	<a href="#">Compact Columns</a> More compact columns for showing last success and failure. Easier to understand, and takes less room in your view.	1.9

# Install Plugins

## Jenkins

Jenkins > Update center

 [Back to Dashboard](#)

 [Manage Jenkins](#)

 [Manage Plugins](#)

## Installing Plugins/Upgrades

### Preparation

- Checking internet connectivity

ChuckNorris Plugin



Pending

 [Go back to the top page](#)

(you can start using the installed plugins right away)



 [Restart Jenkins when installation is complete and no jobs are running](#)

# Install Plugins

Updates	Available	Installed	Advanced				
Enabled		Name	Version	Previously installed version	Pinned	Uninstall	
<input checked="" type="checkbox"/>		<a href="#">Jenkins Mailer Plugin</a>	<u>1.4</u>				
<input checked="" type="checkbox"/>		<a href="#">External Monitor Job Type Plugin</a>	<u>1.1</u>				
<input checked="" type="checkbox"/>		<a href="#">LDAP Plugin</a>	<u>1.1</u>				
<input checked="" type="checkbox"/>		<a href="#">pam-auth</a>	<u>1.0</u>				
<input checked="" type="checkbox"/>		<a href="#">ant</a>	<u>1.1</u>				
<input checked="" type="checkbox"/>		<a href="#">ChuckNorris Plugin</a>  ChuckNorris plugin displays a picture of Chuck Norris (instead of Jenkins the butler) and a random Chuck Norris 'The Programmer' fact on each build page.	<u>0.5</u>			<a href="#">Uninstall</a>	
<input checked="" type="checkbox"/>		<a href="#">javadoc</a>	<u>1.0</u>				
<input checked="" type="checkbox"/>		<a href="#">CVS Plugin</a>		<u>1.6</u>			

# Installing a Plugin

## Jenkins

The screenshot shows the Jenkins interface for the 'Hello World' project. The top navigation bar includes 'Jenkins', 'Hello World', '#8', and 'ENABLE AUTO REFRESH'. The main content area displays 'Build #8 (Feb 3, 2013 7:34:27 PM)' with a status message 'Started 5.9 sec ago Took 1.8 sec'. To the right is a 'add description' link. On the left, a sidebar lists project actions: 'Back', 'Changes', 'Workspace', 'Build Now', 'Delete Project', 'Configure' (which is highlighted with a yellow background and a cursor icon), 'Subversion Polling Log', 'Test Result', and 'Previous Build'. The 'Configure' link is described as 'Revision: 7 No changes.' Below the sidebar, it says 'Started by anonymous user' and 'Test Result (no failures)'.

Build #8 (Feb 3, 2013 7:34:27 PM)

Started 5.9 sec ago  
Took 1.8 sec

add description

Changes

Workspace

Build Now

Delete Project

Configure

Subversion Polling Log

Test Result

Previous Build

Revision: 7  
No changes.

Started by anonymous user

Test Result (no failures)

# Installing a Plugins

localhost:8080/job/Hello%20World/configure

Cooking Development Games Health Household Network Psychology Tools TrackAbout >

World configuration

Run unit tests with MSTest

MSTest Version Default

Test Files HelloWorld/HelloWorldTests/bin/debug/HelloWorldTests.dll

Test Categories

Result File Name Results trx

Command Line Arguments

Add build step ▾

**Post-build Actions**

Activate Chuck Norris

Publish MSTest test result report

Test report TRX file Results trx

Basedir of the path is the workspace root.

Add post-build action ▾

Save Apply



## Installing a Plugins

### Jenkins

Jenkins > Hello World

[ENABLE AUTO REFRESH](#)

search



[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Subversion Polling Log](#)

#### [Build History](#) [\(trend\)](#)

#8 [Feb 3, 2013 7:34:27 PM](#)

#7 [Feb 3, 2013 12:07:47 PM](#)

#6 [Feb 3, 2013 9:39:47 AM](#)

#5 [Feb 3, 2013 9:37:00 AM](#)

#4 [Feb 3, 2013 9:29:56 AM](#)

#3 [Feb 3, 2013 9:26:47 AM](#)

## Project Hello World



Chuck Norris doesn't program with a keyboard. He stares the computer down until it does what he wants.



[Workspace](#)



[Recent Changes](#)



[Latest Test Result](#) (no failures)

[add description](#)

[Disable Project](#)

### Permalinks

- [Last build \(#8\), 1 min 33 sec ago](#)
- [Last stable build \(#8\), 1 min 33 sec ago](#)
- [Last successful build \(#8\), 1 min 33 sec ago](#)
- [Last failed build \(#6\), 9 hr 56 min ago](#)
- [Last unsuccessful build \(#6\), 9 hr 56 min ago](#)

# Plugin Configuration

List of JDK installations on this system

## MSBuild

MSBuild installations

MSBuild

Name

Default

Path to MSBuild

C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe

Default parameters

Install automatically

Delete MSBuild

[Add MSBuild](#)

List of MSBuild installations on this system

## Ant

Ant installations

[Add Ant](#)

List of Ant installations on this system

## Maven

Maven installations

[Add Maven](#)

List of Maven installations on this system

## MSTest

MSTest installations...

## Maven Project Configuration

[Save](#)

[Apply](#)

# Plugin Configuration

**Build****Build a Visual Studio project or solution using MSBuild**

MSBuild Version

Default

MSBuild Build File

(Default)  
Default

Command Line Arguments

Advanced...Delete**Run unit tests with MSTest**

MsTest Version

Default

Test Files

HelloWorld/HelloWorldTests/bin/debug/HelloWorldTests.dll

Test Categories

Result File Name

Results.trx

Command Line Arguments

Delete**Add build step ▾**

# Security Overview

☞ Go to Manage Jenkins

☞ Go to Global Security

☞ Check Enable Security



# Security Overview

## Configure Global Security

Enable security

TCP port for JNLP slave agents  Fixed :   Random  Disable

Markup Formatter

Raw HTML

Treat the text as HTML and use it as is without any translation

Disable syntax highlighting

Access Control

### Security Realm

Delegate to servlet container

Jenkins's own user database

Allow users to sign up

LDAP

### Authorization

Anyone can do anything

Legacy mode

Logged-in users can do anything

Matrix-based security

Project-based Matrix Authorization Strategy



User/group	Overall	Slave
Anonymous	Administer Read Run Scripts Upload Plugins Configure Update Center Configure Delete Create Disconnect Connect	
User/group to add:		<input type="button" value="Add"/>

User/group to add:

## Security Overview

jenkins > Hello World > configuration

Back to Dashboard | Status | Changes | Workspace | Build Now | Delete Project | Configure | Subversion Polling Log

**Build History (trend)**

- #8 Feb 3, 2013 7:34:27 PM
- #7 Feb 3, 2013 12:07:47 PM
- #6 Feb 3, 2013 9:39:47 AM
- #5 Feb 3, 2013 9:37:00 AM
- #4 Feb 3, 2013 9:29:56 AM
- #3 Feb 3, 2013 9:26:47 AM
- #2 Feb 3, 2013 9:24:24 AM
- #1 Feb 3, 2013 9:18:34 AM

RSS for all | RSS for failures

**Project name** Hello World

**Description**

[Preview](#)

Discard Old Builds

Enable project-based security

User/group	Job	Run	SCM							
Anonymous	Delete	Configure	Read	Discover	Build	Workspace	Cancel	Delete	Update	Tag

User/group to add:

This build is parameterized

Disable Build (No new builds will be executed until the project is re-enabled.)

Execute concurrent builds if necessary

**Advanced Project Options**

**Source Code Management**

CVS

None

Subversion

Modules

Repository URL

Local module directory (optional)

## Exercise

- Create a Jenkins Job.
- Restart Jenkins Server.

# Exercise

- **LDAP User**

- Integrate Jenkins with LDAP.
- Create two users in LDAP (e.g. adminJ, readJ)
- Give administrator access to **adminJ** and read only access to **readJ**.

- **Jenkins Security**

- Manage Jenkins
- Configure Global Security
- Under Authorization select “Matrix-based security”
- Click Add User/Group.
- Provide User or group name as **Pratian**.
- Check Administer permission for **Pratian**
- Uncheck Administer permission for **adminJ**.

Did you just lock out yourself  
in Jenkins.

How will you fix it?

- **Switch back to Jenkins User Authentication**

- Create a Jenkins user.
- Give read only access to that user.

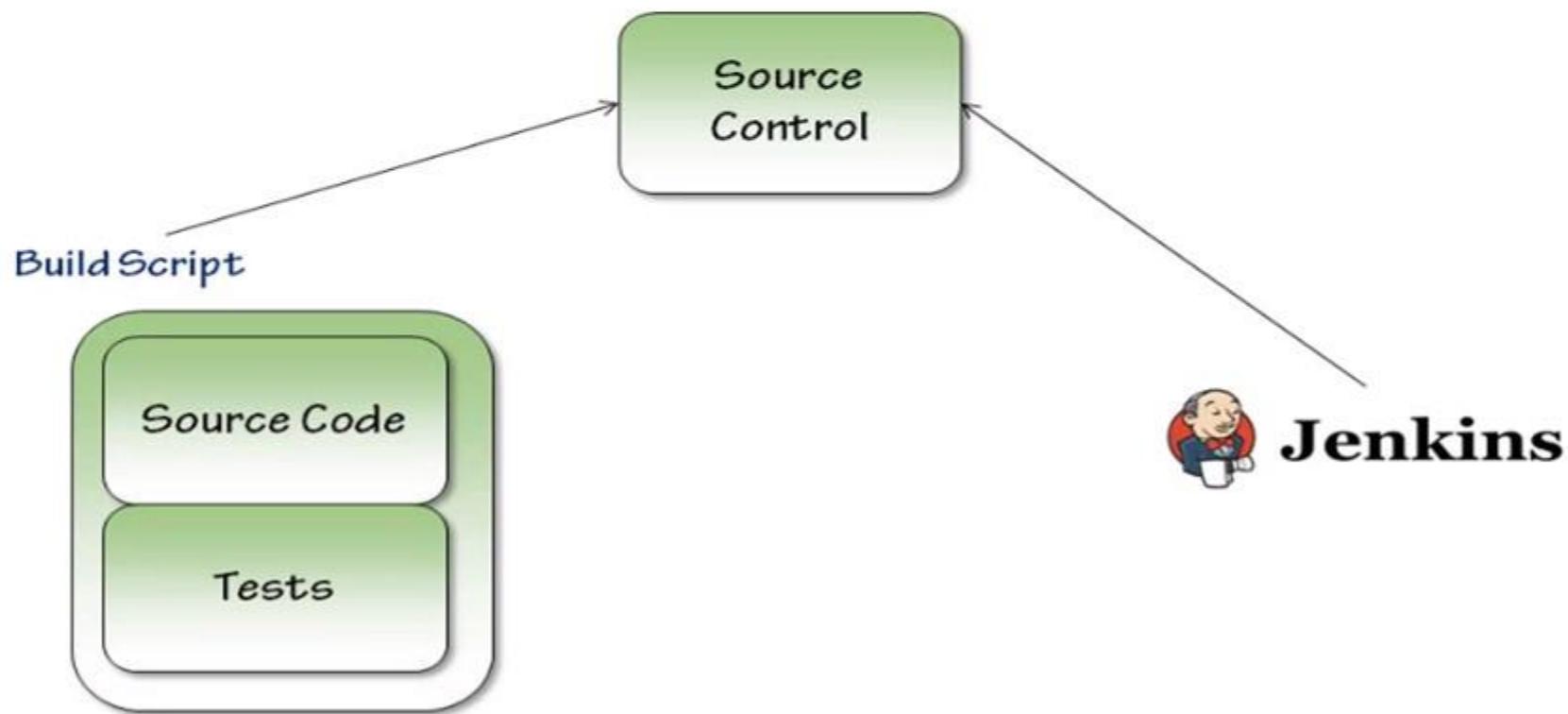
## Exercise

- Change the Jenkins status balls from “BLUE” to “GREEN” for successful builds.
- Integrate MSBuild with Jenkins.

# Creating a Simple Job

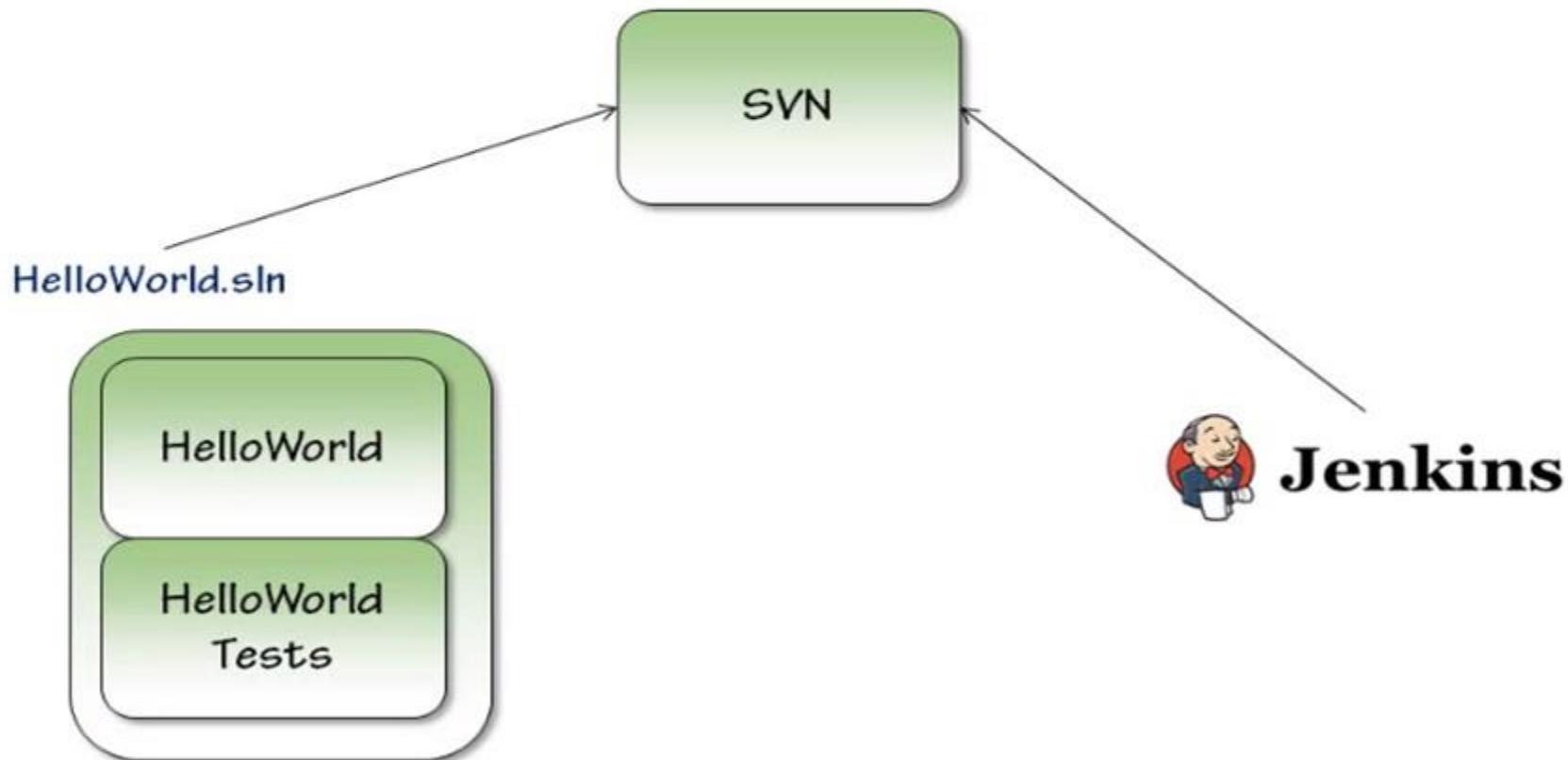
# Basic Project

## Our Project

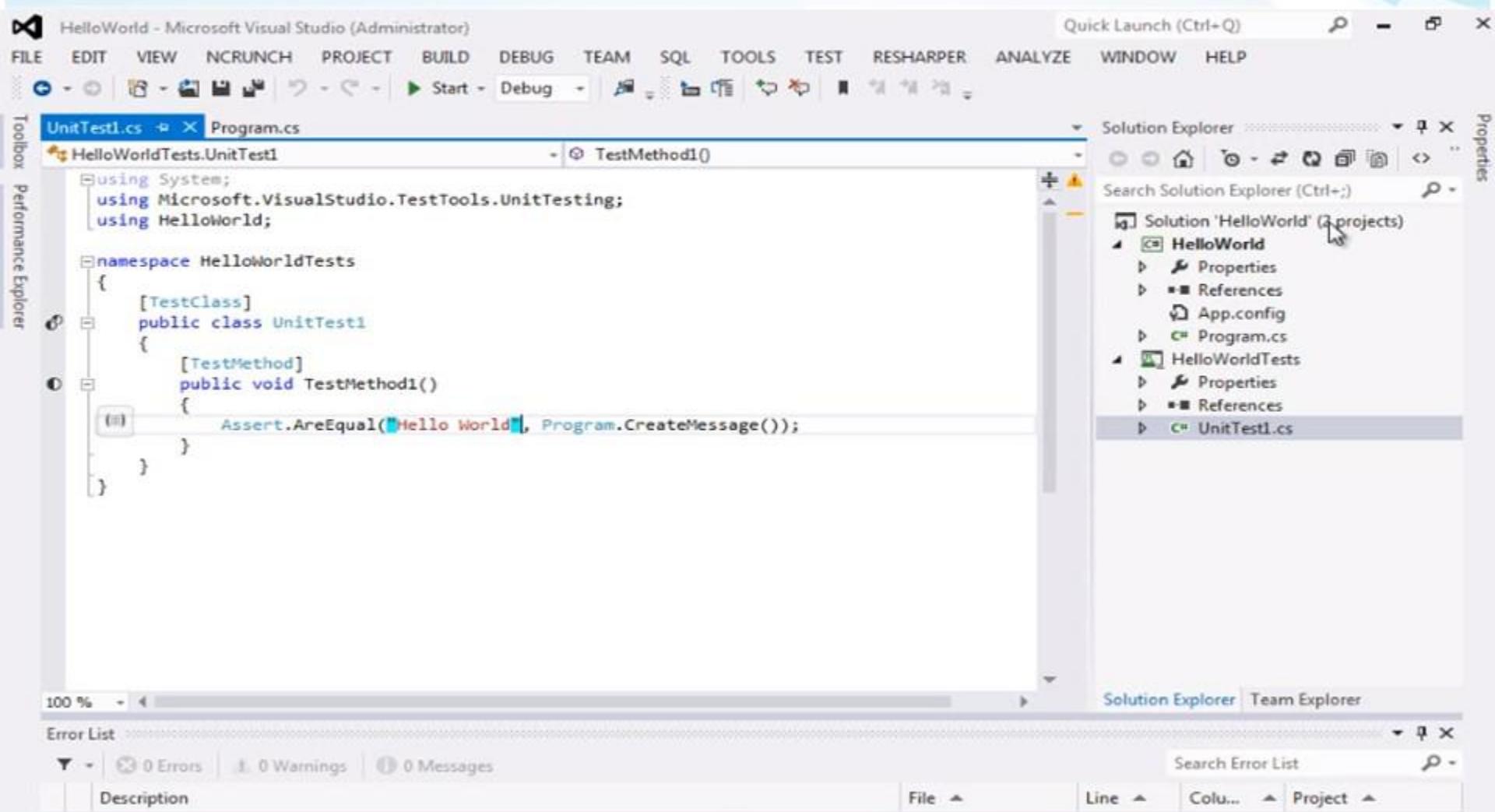


# Our Project

## Our Project



# Looking at the Code



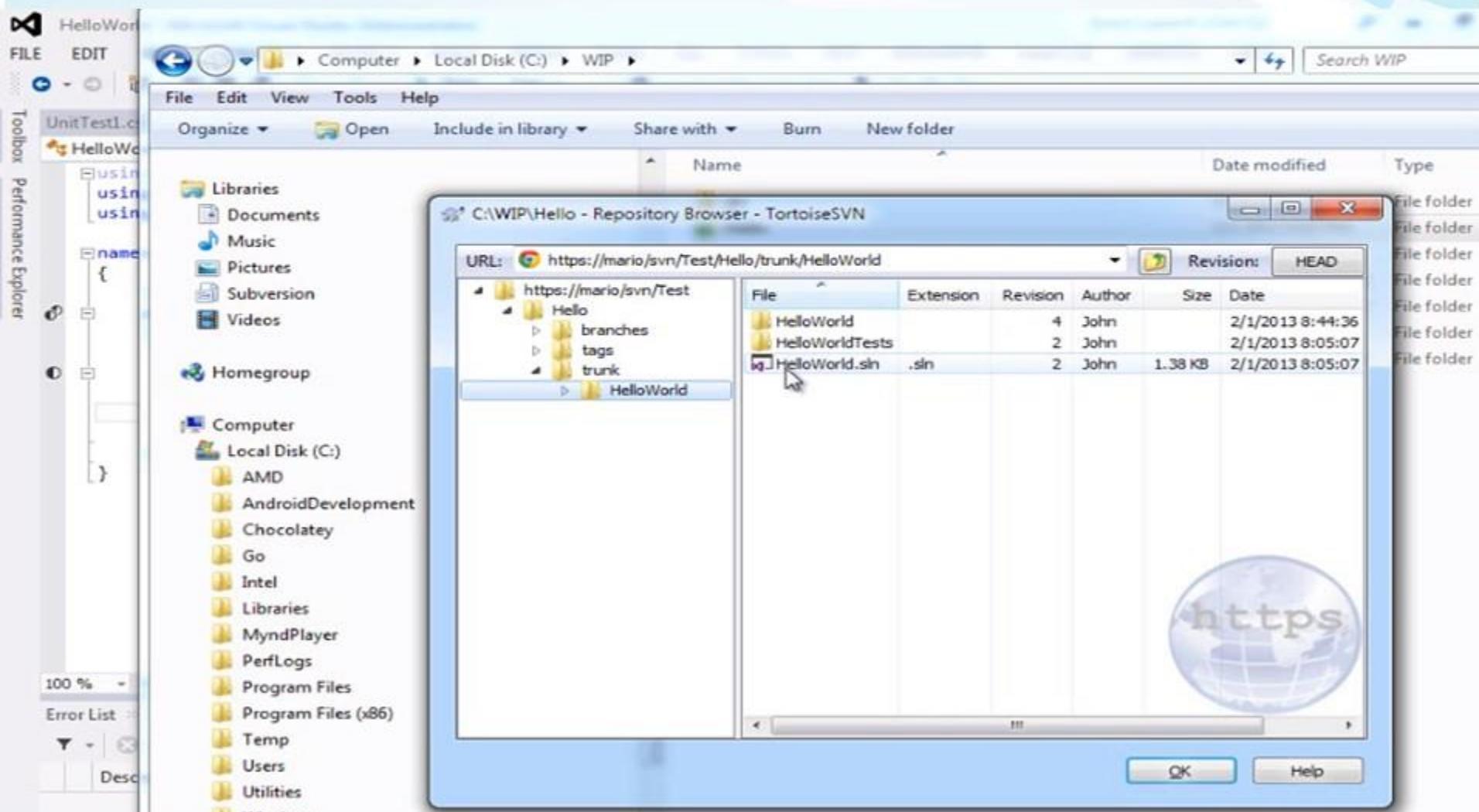
The screenshot shows the Microsoft Visual Studio interface with the following details:

- WindowTitle:** HelloWorld - Microsoft Visual Studio (Administrator)
- MenuBar:** FILE EDIT VIEW NCRUNCH PROJECT BUILD DEBUG TEAM SQL TOOLS TEST RESHARPER ANALYZE WINDOW HELP
- Toolbars:** Standard toolbar with icons for Save, Undo, Redo, Cut, Copy, Paste, Find, and others.
- Code Editor:** Displays `UnitTest1.cs` with the following code:

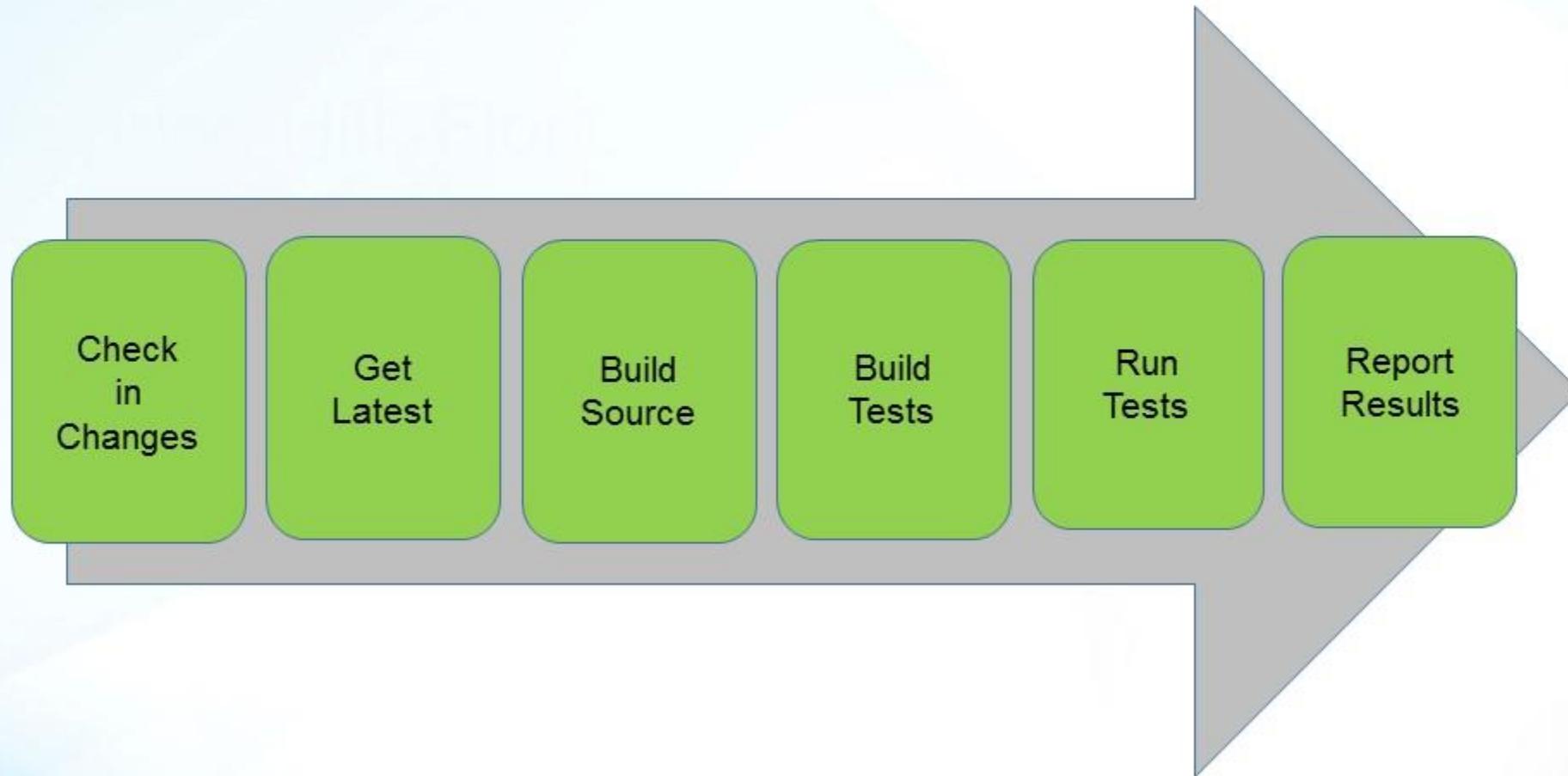
```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using HelloWorld;

namespace HelloWorldTests
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestMethod1()
        {
            Assert.AreEqual("Hello World", Program.CreateMessage());
        }
    }
}
```
- Solution Explorer:** Shows the project structure:
  - Solution 'HelloWorld' (2 projects)
    - >HelloWorld
      - Properties
      - References
      - App.config
      - Program.cs
    - >HelloWorldTests
      - Properties
      - References
      - UnitTest1.cs
  - Properties Explorer:** Shows the properties for the selected project and file.
  - Task List:** Shows the tasks available in the current file.
  - Toolbox:** Shows various development tools and components.
  - Performance Explorer:** Shows performance-related data.
  - Toolbox:** Shows various development tools and components.
  - Code Editor:** Shows the code for `UnitTest1.cs` with the test method `TestMethod1` selected.
  - Solution Explorer:** Shows the project structure with the `UnitTest1.cs` file highlighted.
  - Error List:** Shows 0 Errors, 0 Warnings, and 0 Messages.
  - Search:** Shows the search bar for the error list.

# Looking at the Code



# Moving to Jenkins

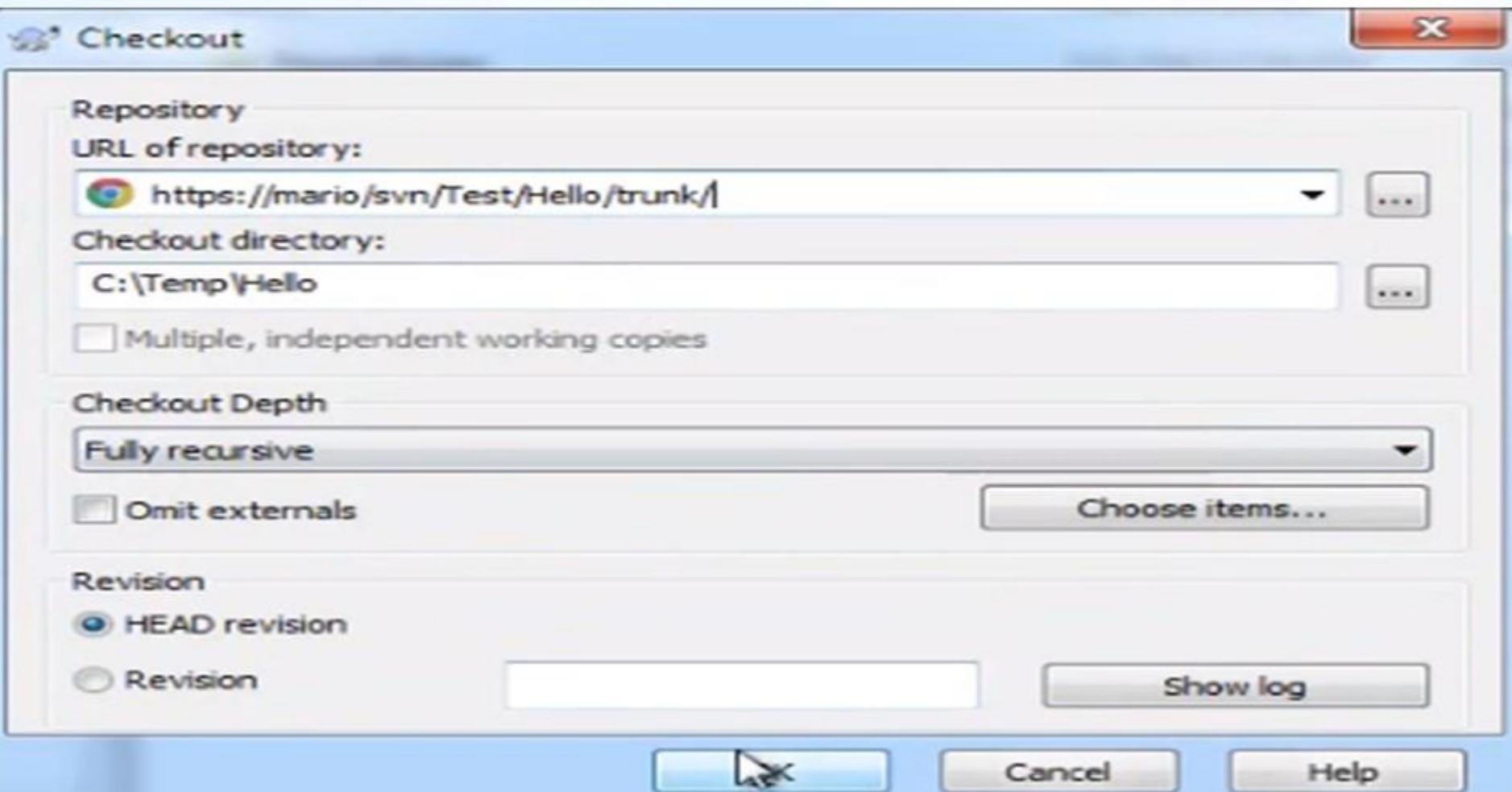


# Moving to Jenkins

1. Diagram manual process
2. Follow manual steps
3. Get most basic automation working
4. Add each step
5. Never leave the build in a failed state



# Manually Building



# Manually Building

```
Administrator: Developer Command Prompt for VS2012
```

```
Directory of C:\Temp\Hello
```

```
02/03/2013  09:01 AM  <DIR> .
02/03/2013  09:01 AM  <DIR> ..
02/03/2013  09:01 AM  <DIR>     HelloWorld
              0 File(s)           0 bytes
              3 Dir(s)  52,596,002,816 bytes free
```

```
C:\Temp\Hello>cd HelloWorld
```

```
C:\Temp\Hello\HelloWorld>dir
```

```
Volume in drive C has no label.
Volume Serial Number is C4A3-536A
```

```
Directory of C:\Temp\Hello\HelloWorld
```

```
02/03/2013  09:01 AM  <DIR> .
02/03/2013  09:01 AM  <DIR> ..
02/03/2013  09:01 AM  <DIR>     HelloWorld
02/03/2013  09:01 AM  1,419 <DIR>     HelloWorld.sln
02/03/2013  09:01 AM  <DIR>     HelloWorldTests
              1 File(s)           1,419 bytes
              4 Dir(s)  52,594,954,240 bytes free
```

```
C:\Temp\Hello\HelloWorld>msbuild HelloWorld.sln
```

# Manually Building

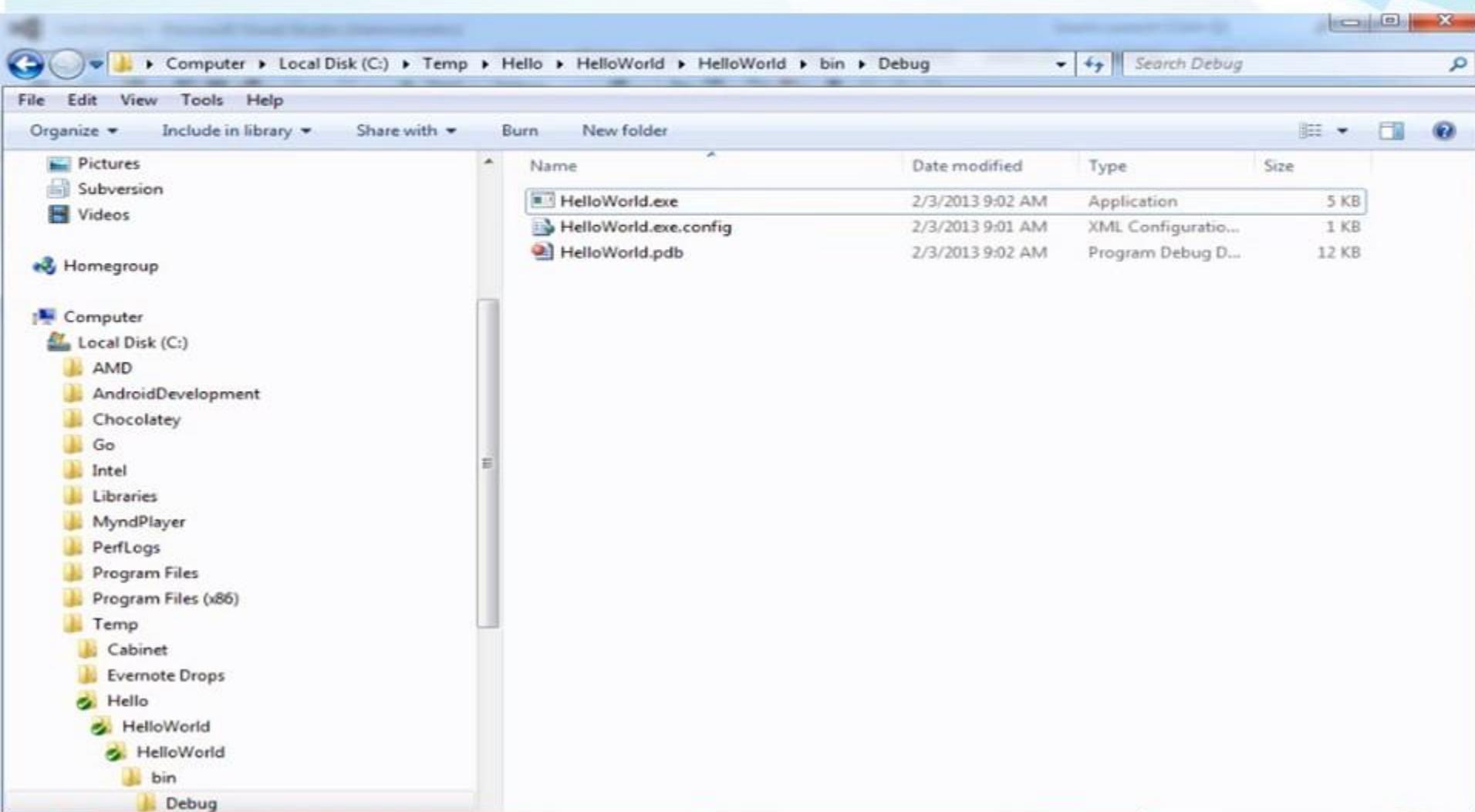
```
Administrator: Developer Command Prompt for VS2012
ts.dll".
HelloWorldTests -> C:\Temp\Hello\HelloWorld\HelloWorldTests\bin\Debug\HelloWorldTests.dll
Copying file from "obj\Debug\HelloWorldTests.pdb" to "bin\Debug\HelloWorldTests.pdb".
Done Building Project "C:\Temp\Hello\HelloWorld\HelloWorldTests\HelloWorldTests.csproj" (default targets).

Done Building Project "C:\Temp\Hello\HelloWorld\HelloWorld.sln" (default targets).

Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:00.31
C:\Temp\Hello\HelloWorld>_
```

# Manually Building



# Manually Building

```
Administrator: Developer Command Prompt for VS2012 - mstest /testcontainer:HelloWorldTests\...
HelloWorldTests -> C:\Temp\Hello\HelloWorld\HelloWorldTests\bin\Debug\HelloWorldTests.dll
Copying file from "obj\Debug\HelloWorldTests.pdb" to "bin\Debug\HelloWorldTests.pdb".
Done Building Project "C:\Temp\Hello\HelloWorld\HelloWorldTests\HelloWorldTests.csproj" (default targets).

Done Building Project "C:\Temp\Hello\HelloWorld\HelloWorld.sln" (default targets).

Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:00.31

C:\Temp\Hello\HelloWorld>mstest /testcontainer:HelloWorldTests\bin\debug\HelloWorldTests.dll
Microsoft (R) Test Execution Command Line Tool Version 11.0.50727.1
Copyright (c) Microsoft Corporation. All rights reserved.

Loading HelloWorldTests\bin\debug\HelloWorldTests.dll...
Starting execution...
```

# Manually Building

```
Administrator: Developer Command Prompt for VS2012
```

```
C:\Temp\Hello\HelloWorld>mstest /testcontainer:HelloWorldTests\bin\debug\HelloWorldTests.dll
Microsoft (R) Test Execution Command Line Tool Version 11.0.50727.1
Copyright (c) Microsoft Corporation. All rights reserved.

Loading HelloWorldTests\bin\debug\HelloWorldTests.dll...
Starting execution...

Results                               Top Level Tests
-----                               -----
Passed                               HelloWorldTests.UnitTest1.TestMethod1
1/1 test(s) Passed

Summary
-----
Test Run Completed.
  Passed  1
-----
  Total  1
Results file:  C:\Temp\Hello\HelloWorld\TestResults\John_MARIO 2013-02-03 09_04_04.trx
Test Settings: Default Test Settings

C:\Temp\Hello\HelloWorld>_
```

# Manually Building

Solution1 - Microsoft Visual Studio (Administrator)

File Edit View Project Debug Team Data Tools Test ReSharper Analyze Window Help

Windows Phone Emulator - 51

Test Results

John@MARIO 2013-02-03 09:04:0 | Run | Debug | Group By: [None]

Test run completed Results: 1/1 passed; Item(s) checked: 0

Result	Test Name	Project	Error Message
Passed	TestMethod1		

## Build Triggers



After other projects



Poll source control



Periodically

# Creating a job

## Jenkins

Jenkins &gt; All

 search [New Job](#)

Job name

Hell

 [People](#) [Build History](#) [Manage Jenkins](#)

### Build Queue

No builds in the queue.

### Build Executor Status

#	Status
1	Idle
2	Idle

### **Build a free-style software project**

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



### **Build a maven2/3 project**

Build a maven2 project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

### **Build multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

### **Monitor an external job**

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).

### **Copy existing Job**

Copy from

 OK

# Creating a job

Jenkins > Help > configuration

 [Back to Dashboard](#)

 [Status](#)

 [Changes](#)

 [Workspace](#)

 [Build Now](#)

 [Delete Project](#)

 [Configure](#)

**Build History** (trend)

 [RSS for all](#)  [RSS for failures](#)

Project name

Description

[Preview](#)

- Discard Old Builds
- This build is parameterized
- Disable Build (No new builds will be executed until the project is re-enabled.)
- Execute concurrent builds if necessary

[Advanced Project Options](#)

[Advanced...](#)

**Source Code Management**

- CVS
- None
- Subversion

[Build Triggers](#)

# Creating a job

## Source Code Management

- CVS
- None
- Subversion

**Modules****Repository URL**

<https://mario/svn/Test>Hello/trunk>

**Local module directory (optional)**

.



[Add more locations...](#)

**Check-out Strategy**

[Use 'svn update' as much as possible](#)



Use 'svn update' whenever possible, making the build faster. But this causes the artifacts from the previous build to remain when a new build starts.

**Repository browser**

(Auto)



[Advanced...](#)

# Creating a job

**Build Triggers**

- Build after other projects are built
- Build periodically
- Poll SCM

**Schedule**

This field follows the syntax of cron (with minor differences). Specifically, each line consists of 5 fields separated by TAB whitespace:

MINUTE HOUR DOM MONTH DOW

MINUTE Minutes within the hour (0-59)

HOUR The hour of the day (0-23)

DOM The day of the month (1-31)

MONTH The month (1-12)

DOW The day of the week (0-7) where 0 and 7 are Sunday.

To specify multiple values for one field, the following operators are available. In the order of precedence,

- '\*' can be used to specify all valid values.
- 'M-N' can be used to specify a range, such as "1-5"
- 'M-N/X' or '\*/X' can be used to specify skips of X's value through the range, such as "\*/15" in the MINUTE field "0,15,30,45" and "1-6/2" for "1,3,5"
- 'A,B,...,Z' can be used to specify multiple values, such as "0,30" or "1,3,5"

To allow periodically scheduled tasks to produce even load on the system, the 'H' token can be used. For example, people often use '0 0 \* \* \*' for a daily job, but this ends up causing a large spike in midnight. In contrast, doing 'H H \* \* \*' would still execute a job once a day, but the actual time of the day this gets executed will be spread over by Jenkins.

The 'H' token can be used with a range. For example, 'H H(0-7) \* \* \*' means some time between midnight to 7:59a

The 'H' token can be thought of as a random value over a range, but it actually is a hash of the job name, not a random function, so that the value remains stable for any given project.

Empty lines and lines that start with '#' will be ignored as comments.

## Creating a job

Jenkins

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Subversion Polling Log

Build History (trend)

RSS for all RSS for failures

Help us localize this page

Project Hello World

Workspace

Recent Changes

Permalinks

search

ENABLE AUTO REFRESH

Add description

Disable Project

Page generated: Feb 3, 2013 9:18:23 AM REST API Jenkins ver. 1.500

Jenkins

New Job

People

Build History

Manage Jenkins

Build Queue

No builds in the queue.

Build Executor Status

#	Status
1	Building Hello World #1
2	Idle

All +

S	W	Name	Last Success	Last Failure	Last Duration
		Hello World	N/A	N/A	N/A
		Test Project	1 day 12 hr (#20)	1 day 12 hr (#19)	1.8 sec

Icon: S M L

Legend RSS for all RSS for failures RSS for just latest builds

# Creating a job

## Jenkins

 search

Jenkins &gt; Hello World &gt; #1

 [Back to Project](#) [Status](#) [Changes](#) [Console Output](#) [View as plain text](#) [Edit Build Information](#) [Delete Build](#) [Tag this build](#)

## Console Output

```
Started by user anonymous
Building in workspace C:\Program Files (x86)\Jenkins\jobs\Hello World\workspace
Checking out a fresh workspace because there's no workspace at C:\Program Files (x86)\Jenkins\jobs\Hello
World\workspace
Cleaning local Directory .
Checking out https://mario/svn/Test>Hello/trunk
A     HelloWorld
A     HelloWorld\HelloWorld.sln
A     HelloWorld\HelloWorld
A     HelloWorld\HelloWorld\App.config
A     HelloWorld\HelloWorld\HelloWorld.csproj
A     HelloWorld\HelloWorld\Program.cs
A     HelloWorld\HelloWorld\Properties
A     HelloWorld\HelloWorld\Properties\AssemblyInfo.cs
A     HelloWorld\HelloWorldTests
A     HelloWorld\HelloWorldTests\UnitTest1.cs
A     HelloWorld\HelloWorldTests\Properties
A     HelloWorld\HelloWorldTests\Properties\AssemblyInfo.cs
A     HelloWorld\HelloWorldTests\HelloWorldTests.csproj
At revision 4
Finished: SUCCESS
```

# Creating a job

## Jenkins

 search[ENABLE AUTO REFRESH](#) [Add description](#)

- [New Job](#)
- [People](#)
- [Build History](#)
- [Manage Jenkins](#)

### Build Queue

No builds in the queue.

### Build Executor Status

#	Status
1	Idle
2	Idle

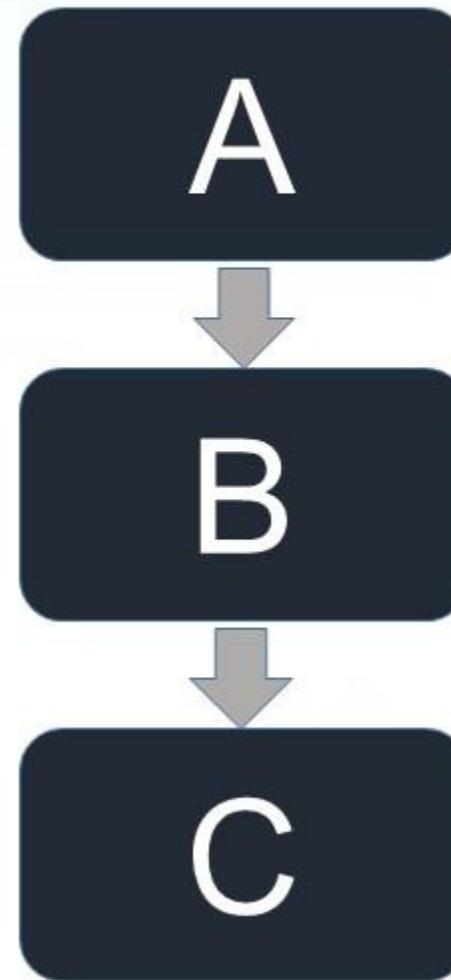
[All](#) [+](#)

S	W	Name	Last Success	Last Failure	Last Duration
		<a href="#">Hello World</a>	23 sec (#1)	N/A	0.54 sec
		<a href="#">Test Project</a>	1 day 12 hr (#20)	1 day 12 hr (#19)	1.8 sec

Icon:

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

## Build Steps



# First Build Step

- ☞ Click Add Build Step
- ☞ Select Build Type
- ☞ Select MSBuild Version
- ☞ Add MSBuild build file (Solution File )
- ☞ Click Save

**Build Triggers**

Build after other projects are built  
 Build periodically  
 Poll SCM

Schedule

\*\*\*\*\*

**Do you really mean "every minute" when you say "\*\*\*\*\*"? Perhaps you**

**Ignore post-commit hooks**

**Build**

**Build a Visual Studio project or solution using MSBuild**

MSBuild Version: Default

MSBuild Build File: **HelloWorld\HelloWorld.sln**

Command Line Arguments:

**Post-build Actions**

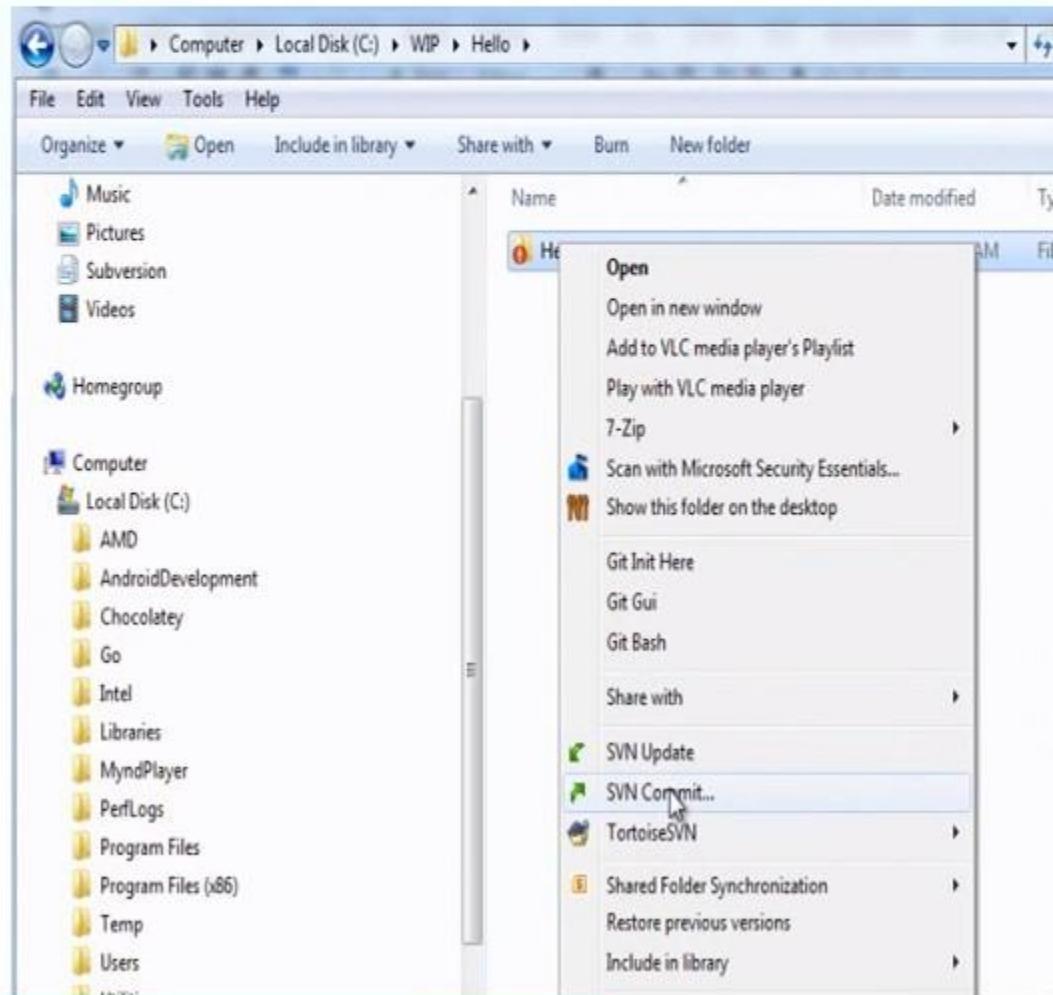
Add post-build action ▾

**Add build step ▾**

Save Apply

# Testing the Trigger

- ☛ Make some change in Code
- ☛ Enable Auto refresh in Jenkins
- ☛ Commit this change in SVN



# Add Tests

- ☞ Add Build Step
- ☞ Select Run Unit Test with MSTests
- ☞ Select MSTest Version
- ☞ Add Location of Test File
- ☞ Add Result File Name
- ☞ Click Save

## Build

### Build a Visual Studio project or solution using MSBuild

MSBuild Version

Default

MSBuild Build File

HelloWorld\HelloWorld.sln

Command Line Arguments

[Advanced...](#)[Delete](#)

### Run unit tests with MSTest

MSTest Version

Default

Test Files

HelloWorld/HelloWorldTests/bin/debug/HelloWorldTests.dll

Test Categories

[?](#)

Result File Name

Results.trx

[?](#)

Command Line Arguments

[?](#)[?](#)[Add build step ▾](#)

## Workspaces

### Jenkins

- [New Job](#)
- [People](#)
- [Build History](#)
- [Manage Jenkins](#)

#### Build Queue

No builds in the queue.

#### Build Executor Status

#	Status
1	Idle
2	Idle

### Jenkins

Jenkins > Hello World

- [Back to Dashboard](#)
  - [Status](#)
  - [Changes](#)
  - [Workspace](#)
  - [Wipe Out Workspace](#)
  - [Build Now](#)
  - [Delete Project](#)
  - [Configure](#)
  - [Subversion Polling Log](#)
- Build History (trend)**
- #4 Feb 3, 2013 9:29:56 AM
  - #3 Feb 3, 2013 9:26:47 AM



## Post Build Actions



Suzie's parents were so proud of her Successful Building of her enterprise application

# Reporting Test Results

## Jenkins

Jenkins > Hello World

search

ENABLE AUTO REFRESH

 [Back to Dashboard](#)

 [Status](#)

 [Changes](#)

 [Workspace](#)

 [Build Now](#)

 [Delete Project](#)

 [Configure](#)

 [Subversion Polling Log](#)

 [Build History](#) [\(trend\)](#)

 #4 [Feb 3, 2013 9:29:56 AM](#)

 #3 [Feb 3, 2013 9:26:47 AM](#)

 #2 [Feb 3, 2013 9:24:24 AM](#)

 #1 [Feb 3, 2013 9:18:34 AM](#)

 [RSS for all](#)  [RSS for failures](#)

## Project Hello World

 [Add description](#)

[Disable Project](#)



[Workspace](#)



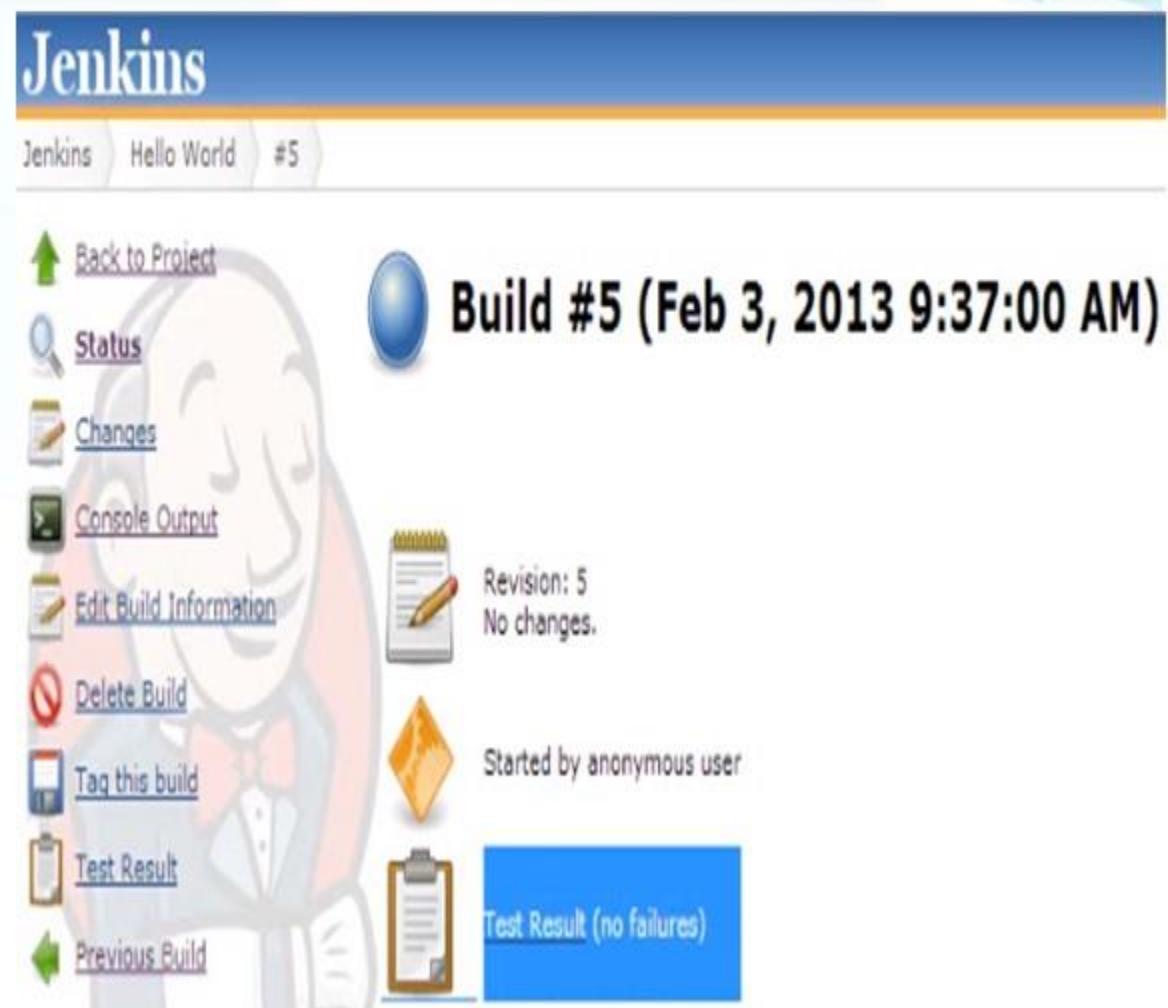
[Recent Changes](#)

### Permalinks

- [Last build \(#4\), 4 min 46 sec ago](#)
- [Last stable build \(#4\), 4 min 46 sec ago](#)
- [Last successful build \(#4\), 4 min 46 sec ago](#)

# Reporting Test Results

- ☞ Go to Project
- ☞ Select Configuration
- ☞ Click Add post-build action
- ☞ Select Publish MSTest test result report
- ☞ Add Test report file name
- ☞ Click Save



# Failing the Build

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** HelloWorld - Microsoft Visual Studio (Administrator)
- Menu Bar:** FILE EDIT VIEW INCRUNCH PROJECT BUILD DEBUG TEAM SQL TOOLS TEST RESHARPER ANALYZE WINDOW HELP
- Toolbar:** Standard Visual Studio toolbar with icons for file operations, search, and other tools.
- Code Editor:** The `Program.cs*` file is open, showing the following C# code:

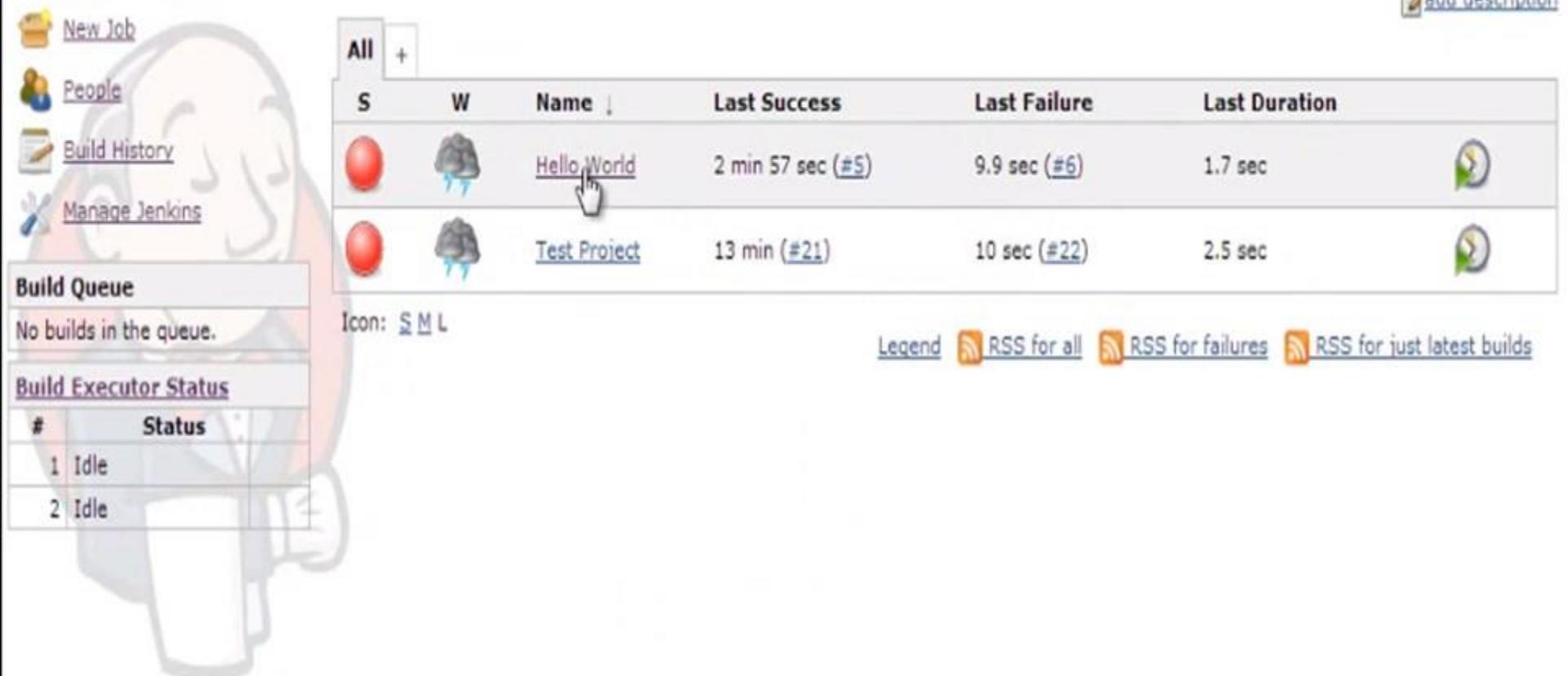
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HelloWorld
{
    public class Program
    {
        private static void Main(string[] args)
        {
            Console.WriteLine(CreateMessage());
        }

        public static string CreateMessage()
        {
            return "Hello World!";
        }
    }
}
```
- Solution Explorer:** Shows the project structure:
  - Solution 'HelloWorld' (2 projects)
    - HelloWorld**
      - Properties
      - References
      - App.config
      - Program.cs** (selected)
    - HelloWorldTests**
      - Properties
      - References
      - UnitTest1.cs
  - Search Solution Explorer (Ctrl+Shift+F)
- Toolbox:** Standard Visual Studio toolbox.
- Performance Explorer:** Standard Visual Studio performance explorer.

## Failing the Build

### Jenkins

[ENABLE AUTO REFRESH](#)[add description](#)

New Job

People

Build History

Manage Jenkins

Build Queue

No builds in the queue.

Build Executor Status

#	Status
1	Idle
2	Idle

All [+](#)

S	W	Name	Last Success	Last Failure	Last Duration
		<a href="#">Hello World</a>	2 min 57 sec (#5)	9.9 sec (#6)	1.7 sec
		<a href="#">Test Project</a>	13 min (#21)	10 sec (#22)	2.5 sec

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

# Failing the Build

## Jenkins

Jenkins > Hello World

[ENABLE AUTO REFRESH](#)



[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Subversion Polling Log](#)

[Build History](#) [\(trend\)](#)

#6 [Feb 3, 2013 9:39:47 AM](#)

#5 [Feb 3, 2013 9:37:00 AM](#)

#4 [Feb 3, 2013 9:29:56 AM](#)

#3 [Feb 3, 2013 9:26:47 AM](#)

#2 [Feb 3, 2013 9:24:24 AM](#)

#1 [Feb 3, 2013 9:18:34 AM](#)

## Project Hello World



[Workspace](#)



[Recent Changes](#)



[Latest Test Result \(1 failure / +1\)](#)



## Permalinks

- [Last build \(#6\), 13 sec ago](#)
- [Last stable build \(#5\), 3 min 1 sec ago](#)
- [Last successful build \(#5\), 3 min 1 sec ago](#)
- [Last failed build \(#6\), 13 sec ago](#)
- [Last unsuccessful build \(#6\), 13 sec ago](#)

# Failing the Build

## Jenkins

Jenkins > Hello World > #6 > Test Results

[ENABLE AUTO REFRESH](#)

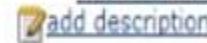


## Test Result

1 failures (+1)

1 tests (±0)

Took 11 ms.



## All Failed Tests

Test Name	Duration	Age
>>> <a href="#">HelloWorldTests.UnitTest1.TestMethod1</a>	11 ms	1

## All Tests

Package	Duration	Fail	(diff)	Skip	(diff)	Total	(diff)
<a href="#">HelloWorldTests</a>	11 ms	1	+1	0	0	1	1

# Failing the Build

Jenkins

search

Jenkins > Hello World #6 Test Results > HelloWorldTests > UnitTest1 > TestMethod1

ENABLE AUTO REFRESH

 [Back to Project](#) [Status](#) [Changes](#) [Console Output](#) [Edit Build Information](#) [History](#) [Polling Log](#) [Tag this build](#) [Test Result](#) [Previous Build](#)

**Regression**

**HelloWorldTests.UnitTest1.TestMethod1** (from MSTestSuite)

Failing for the past 1 build (Since #6) Took 11 ms. [add description](#)

**Stacktrace**

MESSAGE:  
Assert.AreEqual failed. Expected:<Hello World>. Actual:<Hello World!>.  
\*\*\*\*\*  
STACK TRACE:  
at HelloWorldTests.UnitTest1.TestMethod1() in c:\Program Files (x86)\Jenkins\jobs\HelloWorld\workspace\HelloWorld\HelloWorldTests\UnitTest1.cs:line 13

[Help us localize this page](#)

Page generated: Feb 3, 2013 9:40:18 AM REST API Jenkins ver. 1.500

# Fixing the Build

## Jenkins

 search[ENABLE AUTO REFRESH](#) [Add description](#)

 [New Job](#)

 [People](#)

 [Build History](#)

 [Manage Jenkins](#)

**Build Queue**  
No builds in the queue.

**Build Executor Status**

#	Status
1	Idle
2	Idle

All 

S	W	Name	Last Success	Last Failure	Last Duration
		<a href="#">Hello World</a>	9 sec (#7)	2 hr 28 min (#6)	2.4 sec
		<a href="#">Test Project</a>	9 sec (#23)	2 hr 28 min (#22)	2.4 sec

Icon:   

Legend  [RSS for all](#)  [RSS for failures](#)  [RSS for just latest builds](#)

## Fixing the Build

localhost:8080/job/Hello%20World/

Career and Blog Cooking Development Games Health Household Network Psychology Tools TrackAbout Other bookmarks

### Jenkins

Back to Dashboard Status Changes Workspace Build Now Delete Project Configure Subversion Polling Log

### Project Hello World

Workspace Recent Changes Latest Test Result (no failures)

**Test Result Trend**



Build History (trend)

- #7 Feb 3, 2013 12:07:47 PM
- #6 Feb 3, 2013 9:39:47 AM
- #5 Feb 3, 2013 9:37:00 AM
- #4 Feb 3, 2013 9:29:56 AM
- #3 Feb 3, 2013 9:26:47 AM
- #2 Feb 3, 2013 9:24:24 AM

RSS for all RSS for failures

Help us localize this page

Page generated: Feb 3, 2013 12:12:39 PM BEST API Jenkins ver. 1.500

## Exercise

- Run a Jenkins Job in Ubuntu Agent and print Build Number.
- Run a Jenkins Job in Windows Agent and print Build Number.

## Exercise

- Configure a single Jenkins job to make the release from multiple branches (e.g. trunk or feature branches)

## Exercise

- Checkout a Project from SVN.
- Checkout a Project from Github.
- Checkout a Project from GitLab.

## Exercise

- Integrate JAVA with Jenkins.
- Integrate Maven with Jenkins.
- Enable Jenkins to send email notifications.
- Build a Maven Project in Jenkins.
- Build a .NET Project in Jenkins.

## Exercise

- Integrate Jenkins & SonarQube.
- Perform SonarQube analysis for a Java project using Jenkins as CI Tool.
- Perform SonarQube analysis for a .NET project using Jenkins as CI Tool.
- Run recurring SonarQube analysis for any of the above project every mid-night.
- Save the reports of recurring SonarQube Analysis by associating it with Jenkins Build number.
- Make it optional for user to run SonarQube Analysis or not.

## Exercise

- As a DevOps Engineer I did setup a lot of jobs in Jenkins. Accidentally I lost all the jobs. It took me 5 Days to recreate the jobs. What solution do you propose so that it does not take so much time if it happens again?
- As a user I want the build to be triggered at a specific schedule. The schedule of the build is random. What solution do you propose to my problem?
- Looking at the console output it becomes difficult to differentiate between error and normal message. What solution do you suggest?
- How do you create a new job which is an exact copy of existing job?

## Exercise

- On completion of 1<sup>st</sup> job trigger 2nd job.
- Make it optional for user if he wants to trigger 2<sup>nd</sup> job or no.
- Print the Build number of 1<sup>st</sup> job in 2<sup>nd</sup> job

# Thank You

# Exercise

- ☞ Project: HappyTrip
- ☞ Source Control: GIT
- ☞ Application Server: Tomcat
  
- ☞ Problem Statement:
  - ☞ Step 1: Get the HappyTrip Project from GIT
  - ☞ Step 2: Build the HappyTrip Project.
  - ☞ Step 3: Poll the SCM and build the project every minute.
  - ☞ Step 4: Send an email notification at the end of the build
  - ☞ Step 5: Deploy the HappyTrip Project to Tomcat Server.