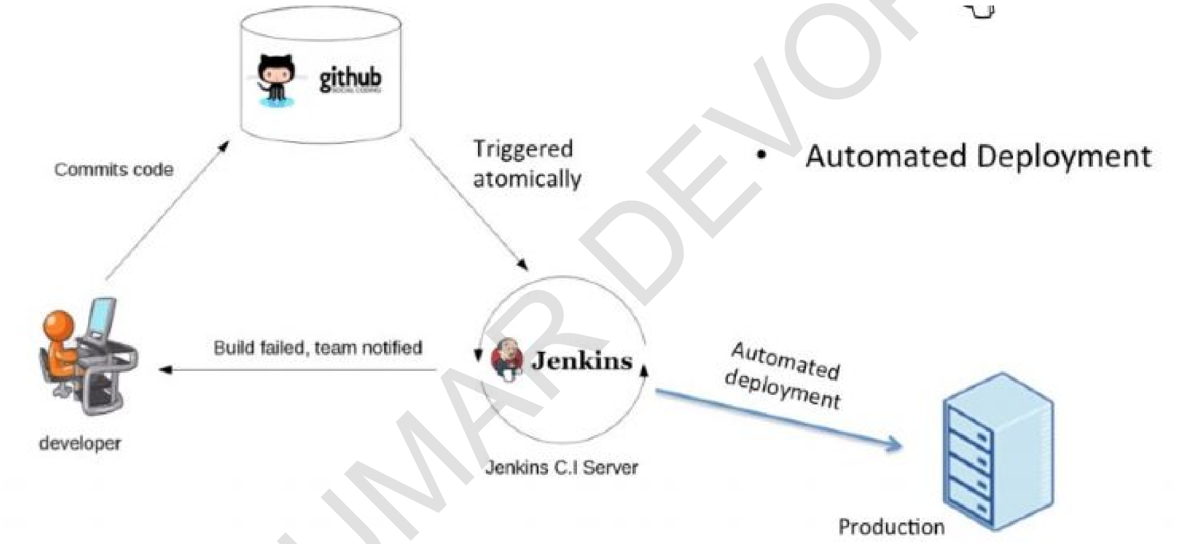
JENKINS

Continuous Integration(CI):

- Continuous integration (CI) is a software engineering practice where, the fresh changes are immediately tested, validated and reported , then they are integrated to main branch
- In traditional software development process integration is generally done at the end of day when every developer has completed their work as a result of this integration took lots of time, however CI is done as soon as code is developed.

Jenkins workflow:



Ways of CI:

- Maintain a code repository.
- Automate your build.
- Daily commits to the baseline by everyone on the team.
- Every commit (to the baseline) should be build.

Benefits of CI:

- Faster product release
- Reduce repetitive manual processes
- Enable better project visibility
- Reduction in bugs count
- Good quality code

Why only jenkins:

- Easy to use
- Great extensibility
- Support different version control systems
- Build notifiers
- UI customization

Java installation and configuration:

Step 1: Download JDK

- Goto Java SE download site @ http://www.oracle.com/technetwork/java/javase/downloads/index.html.
- Download the JDK 1.8 Version

Step 2 : Install JDK on Computer

Step 3: set Path, Include JDK's "bin" Directory in the PATH

You need to include JDK's "bin" in the PATH to run the JDK programs.

To edit the PATH environment variable in Windows :

1. Launch "Control Panel" ⇒ (Optional) "System and Security" ⇒ "System" ⇒ Click "Advanced system settings" on the left pane.
2. Switch to "Advanced" tab ⇒ Click "Environment Variables" button.
3. Under "System Variables"
4. Add PATH and JAVA_HOME information

Step 4: Verify the JDK Installation

Launch a CMD via one of the following means:

1. Click "Search" button ⇒ Type "cmd" ⇒ Choose "Command Prompt", or
2. Right-click the "Start" button ⇒ run... ⇒ enter "cmd", or
3. Click "Start" button ⇒ Windows System ⇒ Command Prompt
4. Use java --version to check java version

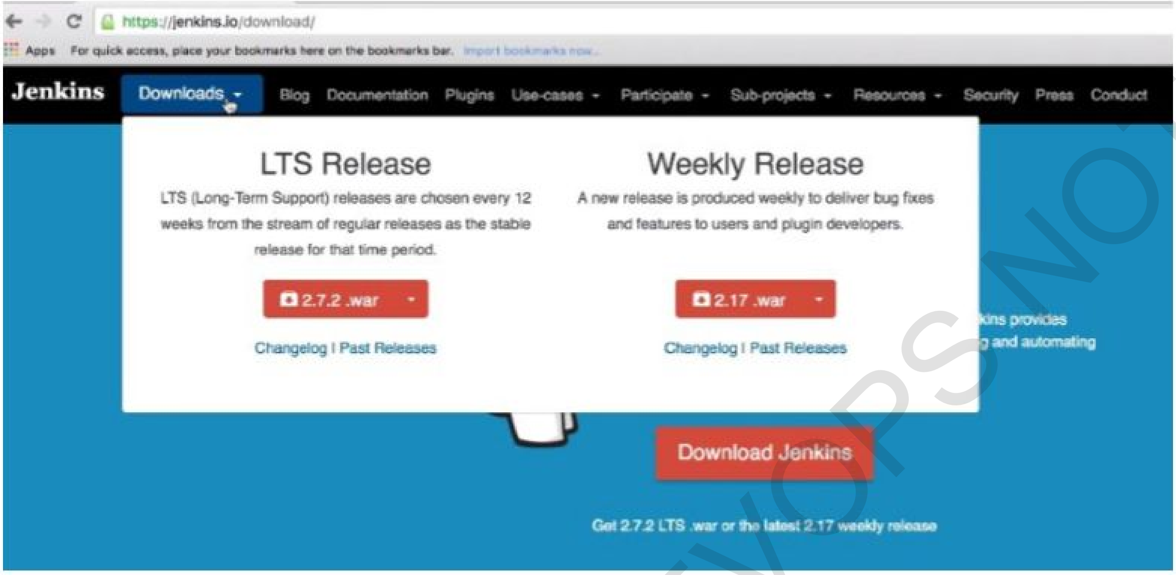
Jenkins installation and configuration:

Step 1 - Open Official Page of Jenkins, https://jenkins.io

Step 2 - Click on downloads

Step 3 - Make sure to download, Long-Term Support version of Jenkins

● Download the LTS version


https://jenkins.io/download/
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Jenkins
Downloads
Blog
Documentation
Plugins
Use-cases
Participate
Sub-projects
Resources
Security
Press
Conduct
2.7.2 .war
2.17 .war
Changelog | Past Releases
Changelog | Past Releases
Download Jenkins

LTS Release

LTS (Long-Term Support) releases are chosen every 12 weeks from the stream of regular releases as the stable release for that time period.

Weekly Release

A new release is produced weekly to deliver bug fixes and features to users and plugin developers.

kins provides

g and automating

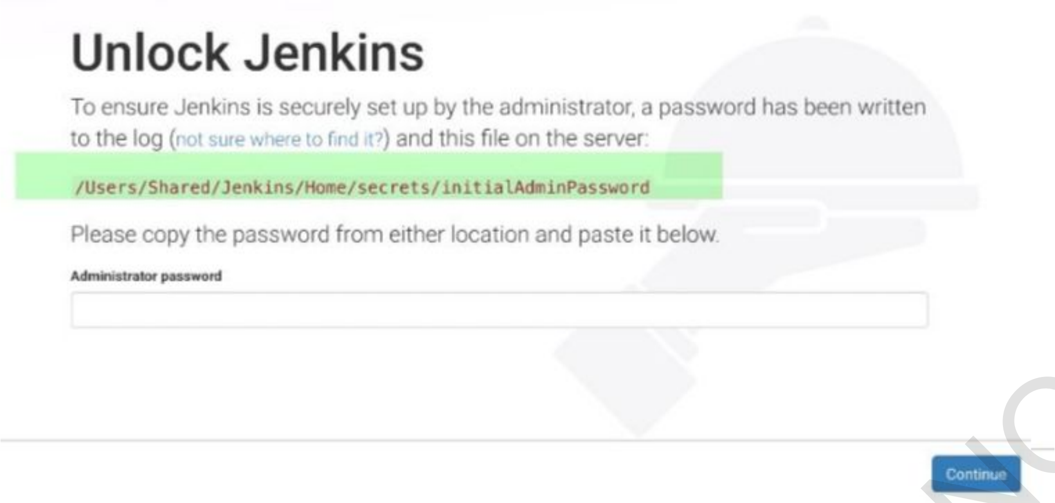
Get 2.7.2 LTS .war or the latest 2.17 weekly release

Step 4 - Preferable to download " Generic Java package (.war)"

Note : WAR file:

The Web application ARchive (WAR) file version of Jenkins can be installed on any operating system or platform that supports Java. To download and run the WAR file version of Jenkins:

1. Download the latest stable Jenkins WAR file to an appropriate directory on your machine.
2. Open up a terminal/command prompt window to the download directory.
3. Run the command java -jar jenkins.war.
4. Browse to http://localhost:8080 and wait until the Unlock Jenkins page appears.
5. Continue on with the Post-installation setup wizard below.



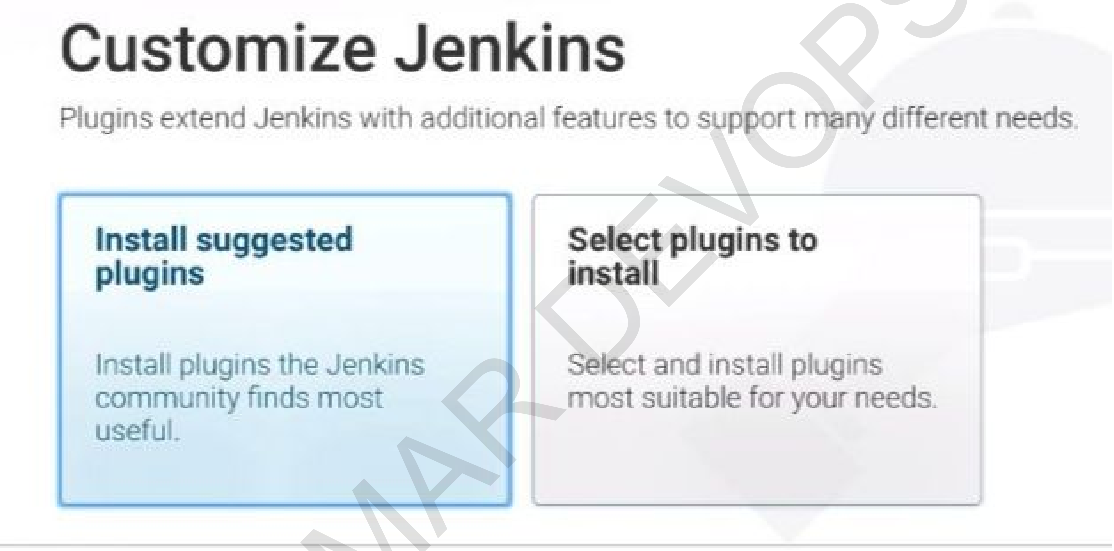
Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
/Users/Shared/Jenkins/Home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password



Customize Jenkins

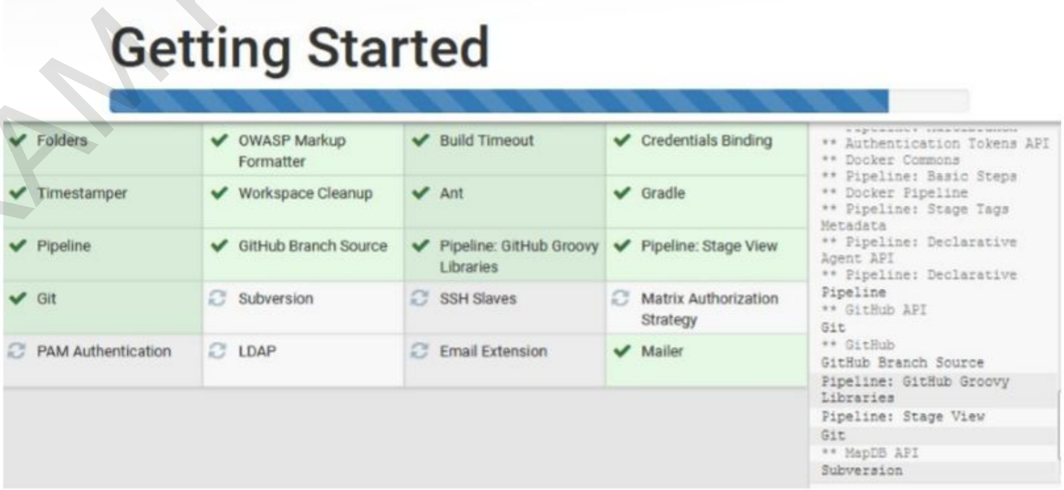
Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

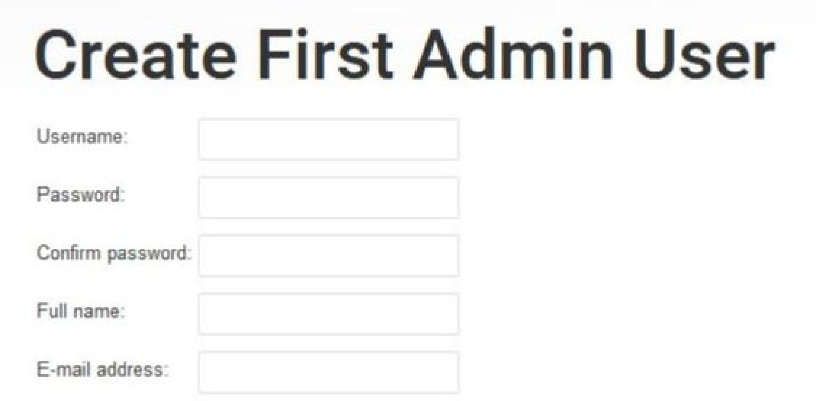
Select and install plugins most suitable for your needs.



Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle
✓ Pipeline	✓ GitHub Branch Source	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View
✓ Git	⟳ Subversion	⟳ SSH Slaves	⟳ Matrix Authorization Strategy
⟳ PAM Authentication	⟳ LDAP	⟳ Email Extension	✓ Mailer

```
** Authentication Tokens API
** Docker Commons
** Pipeline: Basic Steps
** Docker Pipeline
** Pipeline: Stage Tags
Metadata
** Pipeline: Declarative
Agent API
** Pipeline: Declarative
Pipeline
** GitHub API
Git
** GitHub
GitHub Branch Source
Pipeline: GitHub Groovy
Libraries
Pipeline: Stage View
Git
** MapDB API
Subversion
```



Create First Admin User

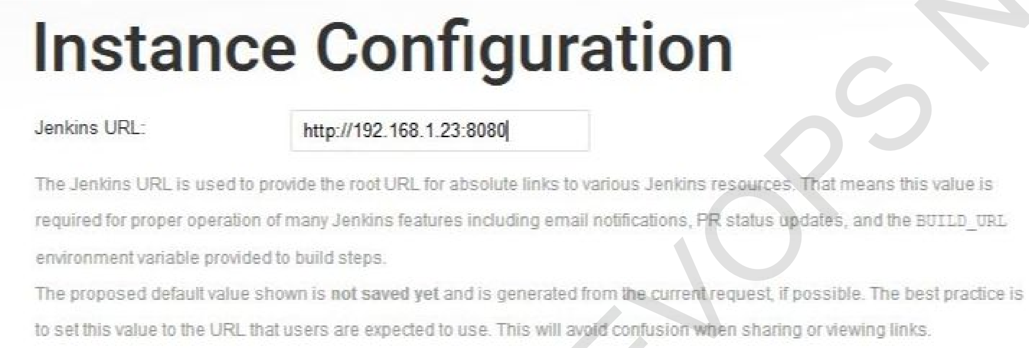
Username:

Password:

Confirm password:

Full name:

E-mail address:

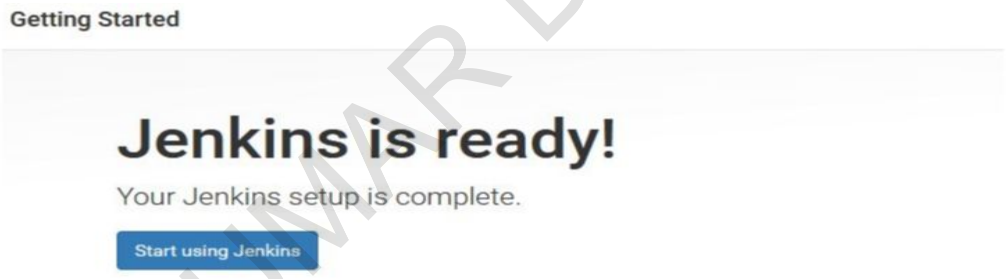


Instance Configuration

Jenkins URL: http://192.168.1.23:8080

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

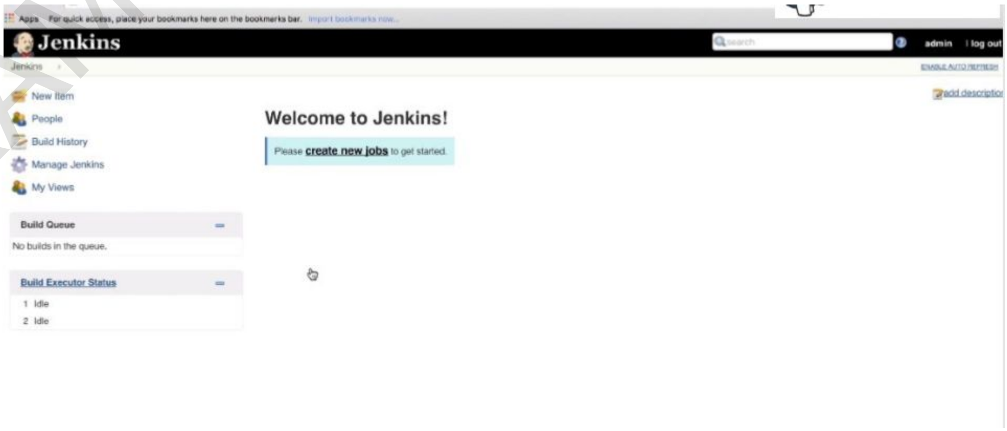
The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.



Getting Started

Jenkins is ready!

Your Jenkins setup is complete.



Welcome to Jenkins!

Please **create new jobs** to get started.

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Free style project:

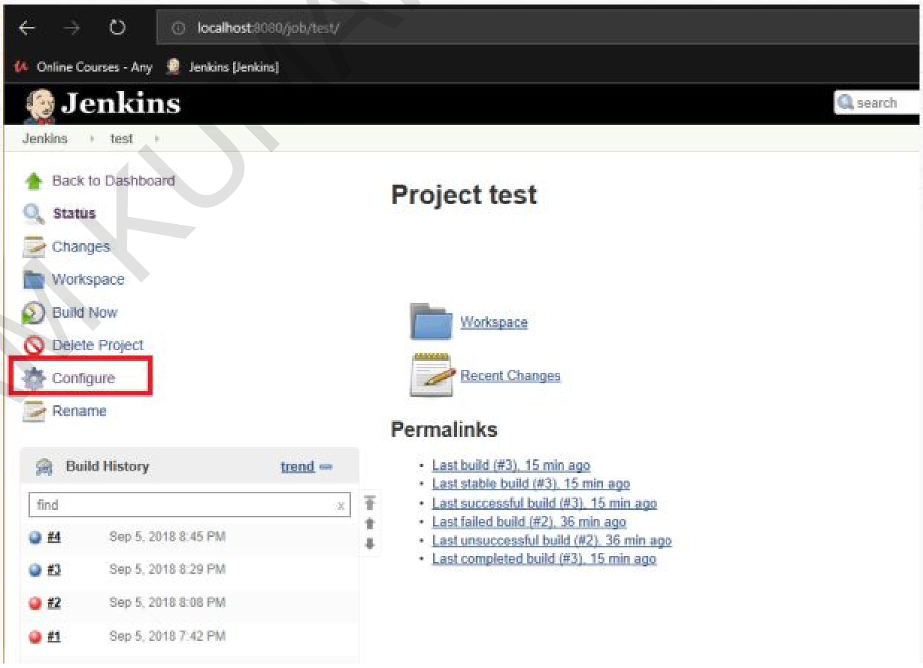
- A freestyle project in Jenkins is a project that spans multiple operations. It can be a build, a script run, or even a pipeline.
- A freestyle project is a typical build job or task. This could be as simple as running tests, building or packaging an application, sending a report, or even running some commands.

Jenkins plugins:

- Plugins are the primary means of enhancing the functionality of a Jenkins environment to suit organization- or user-specific needs.
- The 1,400 plugins encompass source code management, administration, platforms, UI/UX, building management, and much more.
- There are over a thousand different plugins which can be installed on a Jenkins master and to integrate various build tools, cloud providers, analysis tools, and much more.
- Plugins can be automatically downloaded, with their dependencies, from the Update Center.

Source code polling(GIT):

- Under project click Configure

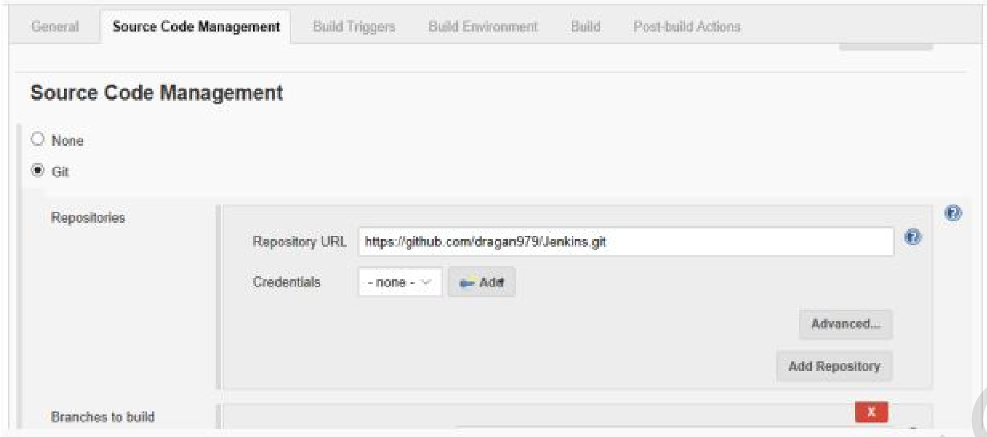


#4 Sep 5, 2018 8:45 PM
#3 Sep 5, 2018 8:29 PM
#2 Sep 5, 2018 8:08 PM
#1 Sep 5, 2018 7:42 PM

Permalinks

- Last build (#3), 15 min ago
- Last stable build (#3), 15 min ago
- Last successful build (#3), 15 min ago
- Last failed build (#2), 36 min ago
- Last unsuccessful build (#2), 36 min ago
- Last completed build (#3), 15 min ago

- Under Source Code Management click Git-paste Git Repo URL



Source Code Management

○ None

● Git

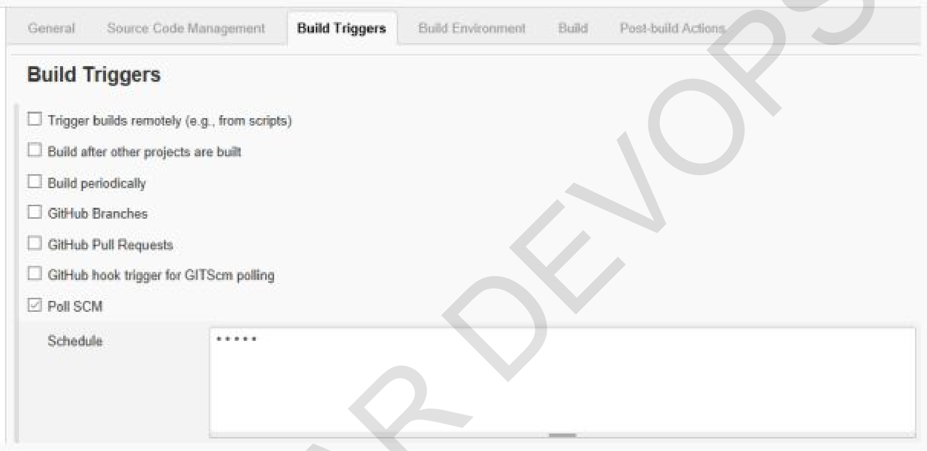
Repositories

Repository URL https://github.com/dragan979/Jenkins.git

Credentials - none -

Branches to build

- Under Build Triggers click Poll SCM and specify time interval for checking GitHub repository for changes



Build Triggers

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

□ Build after other projects are built

□ Build periodically

□ GitHub Branches

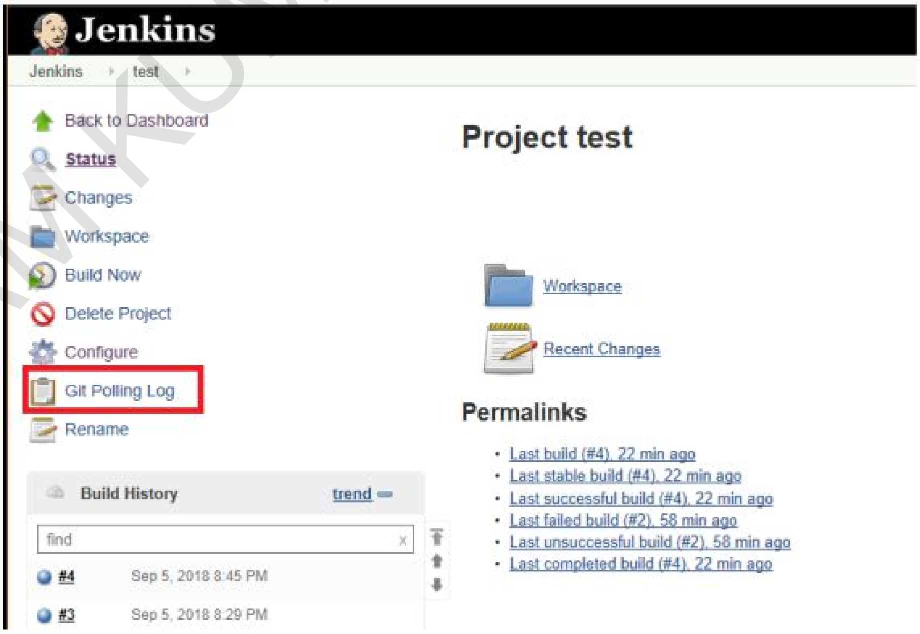
□ GitHub Pull Requests

□ GitHub hook trigger for GITScm polling

☑ Poll SCM

Schedule * * * * *

- Click on Git Polling log to see changes



Project test

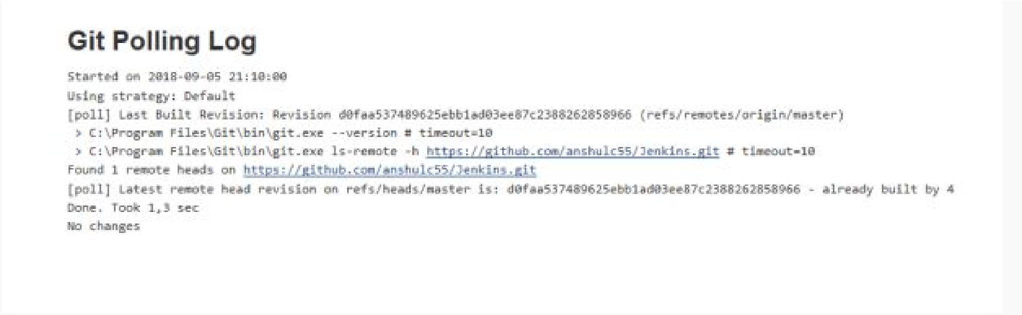
Permalinks

- Last build (#4), 22 min ago
- Last stable build (#4), 22 min ago
- Last successful build (#4), 22 min ago
- Last failed build (#2), 58 min ago
- Last unsuccessful build (#2), 58 min ago
- Last completed build (#4), 22 min ago

#4 Sep 5, 2018 8:45 PM

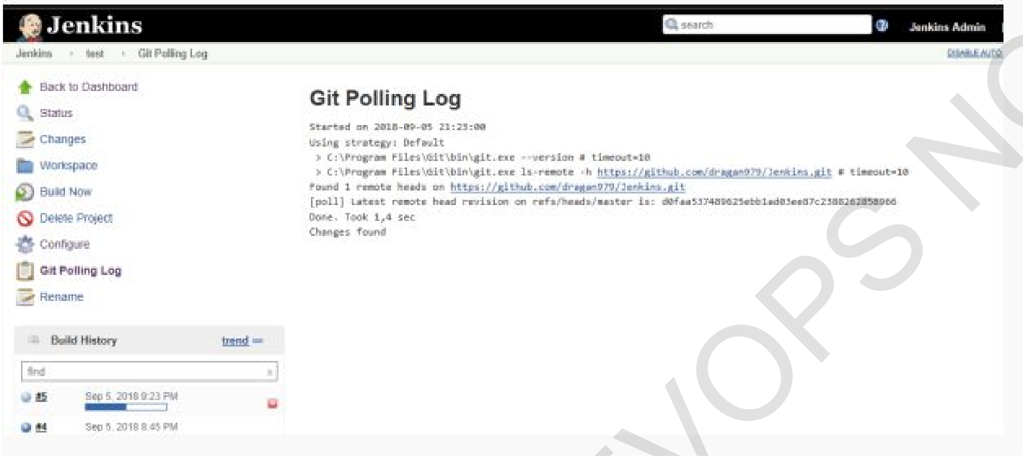
#3 Sep 5, 2018 8:29 PM

- In case i changed something in code, project build will start/



Git Polling Log

```
Started on 2018-09-05 21:10:00
Using strategy: Default
[poll] Last Built Revision: Revision d0faa537489625ebb1ad03ee87c2388262858966 (refs/remotes/origin/master)
 > C:\Program Files\Git\bin\git.exe --version # timeout=10
 > C:\Program Files\Git\bin\git.exe ls-remote -h https://github.com/anshulc55/Jenkins.git # timeout=10
Found 1 remote heads on https://github.com/anshulc55/Jenkins.git
[poll] Latest remote head revision on refs/heads/master is: d0faa537489625ebb1ad03ee87c2388262858966 - already built by 4
Done. Took 1,3 sec
No changes
```



Git Polling Log

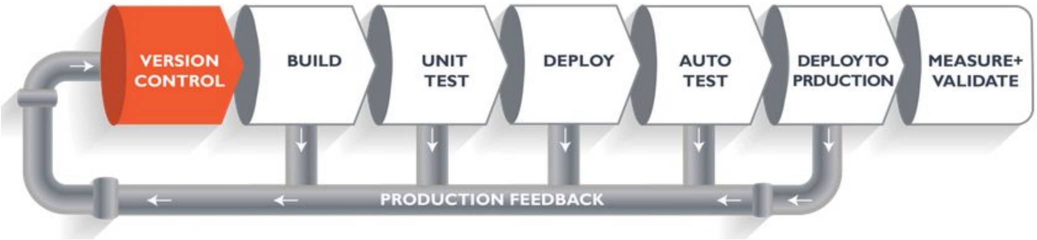
```
Started on 2018-09-05 21:23:00
Using strategy: Default
 > C:\Program Files\Git\bin\git.exe --version # timeout=10
 > C:\Program Files\Git\bin\git.exe ls-remote -h https://github.com/dragan979/Jenkins.git # timeout=10
Found 1 remote heads on https://github.com/dragan979/Jenkins.git
[poll] Latest remote head revision on refs/heads/master is: d0faa537489625ebb1ad03ee87c2388262858966
Done. Took 1,4 sec
Changes found
```

Upstream and Downstream projects:

- An upstream job is a configured project that triggers a project as of its execution.
- A downstream job is a configured project that is triggered as part of a execution of a pipeline.
- We can configure the actual job not to be triggered if the upstream job is failed.
- In the same way, we can configure the downstream job not to be triggered if the actual job is failed.
- Upstream and downstream jobs help you to configure the sequence of execution for different operations and hence you can orchestrate the flow of execution.
- We can configure one or more projects as downstream jobs in Jenkins.

CI-CD pipeline:

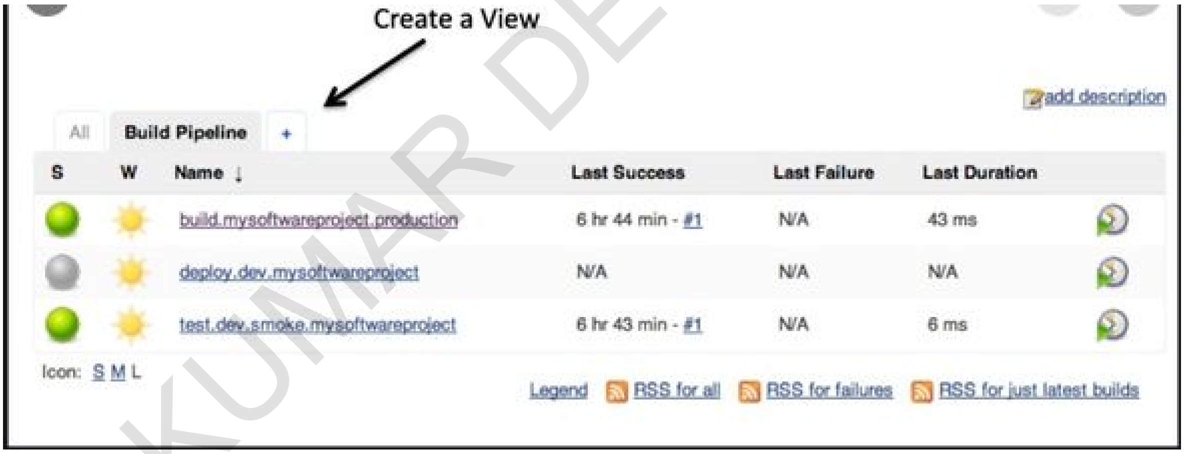
- It bridges the gap between development and operations teams by automating the building, testing, and deployment of applications.



- The above pipeline is a logical demonstration of how software will move along the various stages in this lifecycle before it is delivered to the customer or before it is live in production.
- A CI/CD pipeline helps you automate steps in your software delivery process, such as initiating code builds, running automated tests, and deploying to a staging or production environment.
- Automated pipelines remove manual errors, provide standardized development feedback loops and enable fast product iterations.

Jenkins views:

- Views in Jenkins allow us to organize jobs and content into tabbed categories, which are displayed on the main dashboard.



Create a View

add description

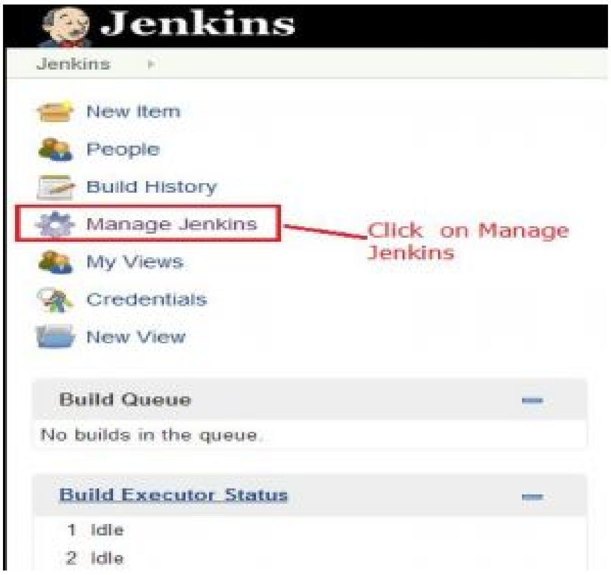
S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	build.mysoftwareproject.production	6 hr 44 min - #1	N/A	43 ms
●	☀	deploy.dev.mysoftwareproject	N/A	N/A	N/A
●	☀	test.dev.smoke.mysoftwareproject	6 hr 43 min - #1	N/A	6 ms

- Clicking on the + tab, jenkins will navigate us to the basic view configuration page. From this page we will need to specify a name and view type for our new view.
- As a Jenkins instance expands, it is logical to create associated views for appropriate groups and categories.
- The default view type available in Jenkins is the List view.

User management:

User creation in jenkins:

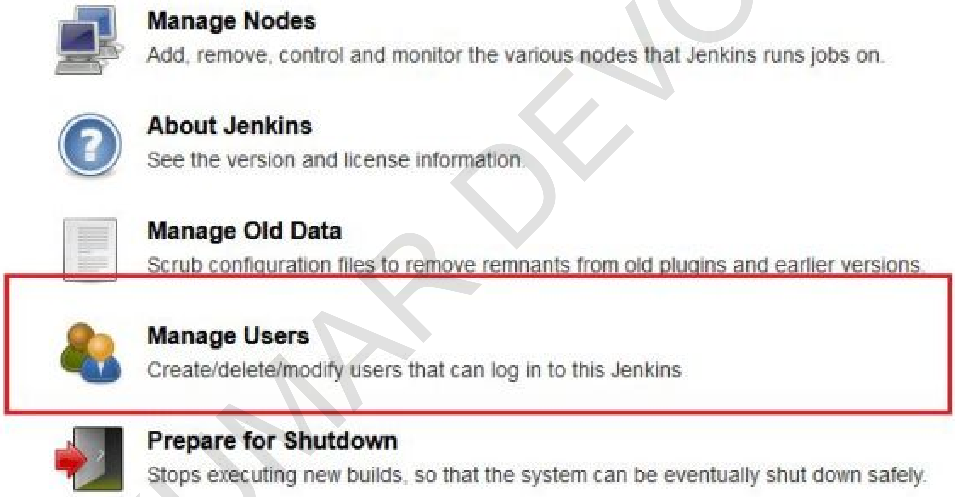
Step 1 - Click on Manage Jenkins in Dashboard



Click on Manage Jenkins

No builds in the queue.

Step 2 - select Manage Users


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.

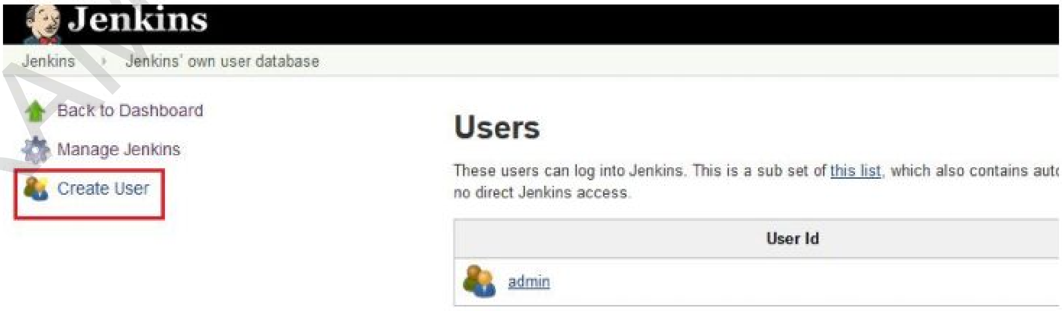
Manage Users

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

Prepare for Shutdown

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

Step 3 - select Create User

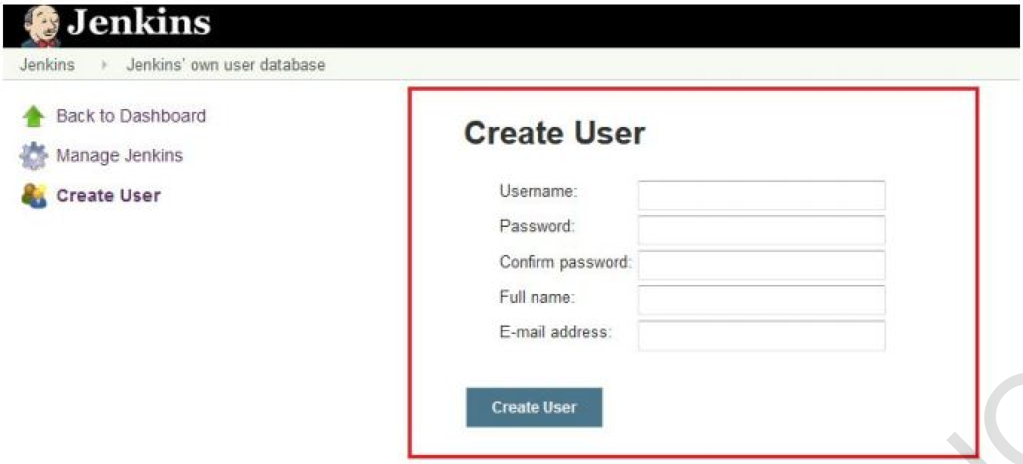


Users

These users can log into Jenkins. This is a sub set of this list, which also contains aut
no direct Jenkins access.

User Id
admin

Step 4 - Now, fill the User Details and then click create user



Create User

Username:

Password:

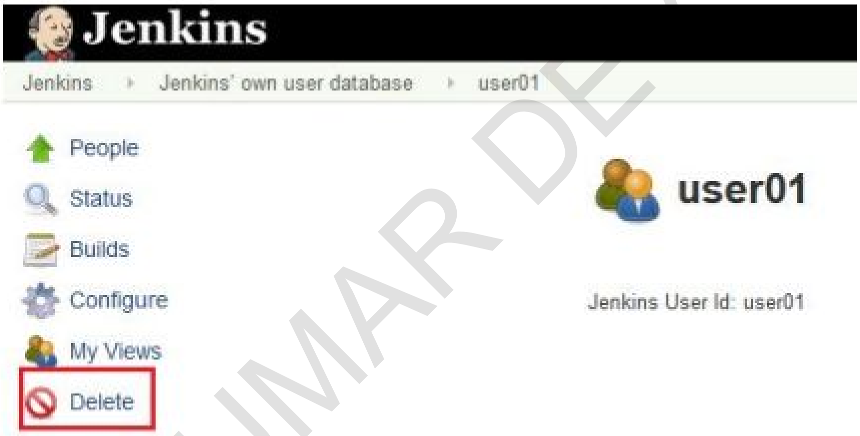
Confirm password:

Full name:

E-mail address:

Q. How to delete an user account?

- Go to manage users
- Open user details
- Then from dashboard click delete



user01

Jenkins User Id: user01

Jenkins slave :

- if we setup builds to happen in server-client environments, then it is called as Master-Slave setup.
- It helps -
- ➢ load on server is reduced
- ➢ platform testing
- ➢ reducing cost

Requirements for setup of Jenkins slave -

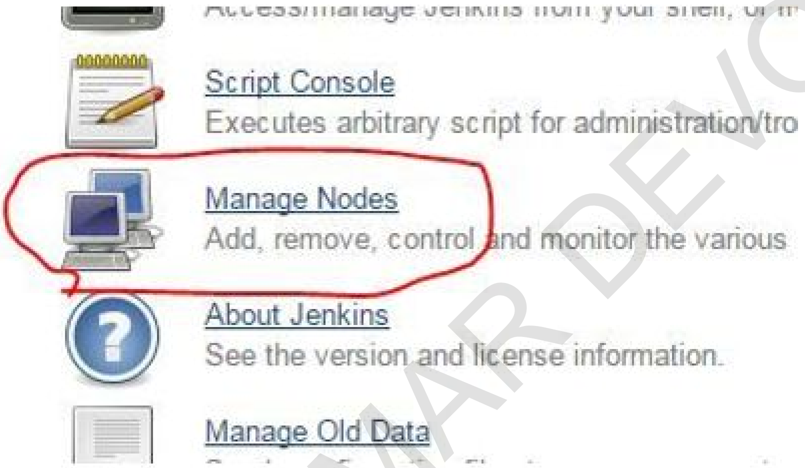
- Need to have one Jenkins master
- Need to have a client/node pc, referred as Jenkins slave

- Jenkins master can be windows, Linux, or mac,
- Jenkins slave can be windows, Linux, mac
- Jenkins Master and slave should be on same network
- we do not require any Jenkins installation on Jenkins slave
- On Jenkins slave, need to have a user id and password that has administrative privileges.
- Jenkins Master these are plugin need to be installed

1. if Jenkins slave is windows, install java web start plugin
2. if Jenkins slave is Linux or mac install ssh slave agent plugin

Setup Relationship between Master and Slave (Windows)

- Note -- Make sure that Manage Jenkins -> Manage plugins and install 'SSH Slaves Plugin' or SSH Agent Plugin
- Go to Manage Jenkins -> Manage node



Access/manage Jenkins from your shell, or if

Script Console
Executes arbitrary script for administration/tro

Manage Nodes
Add, remove, control and monitor the various

About Jenkins
See the version and license information.

Manage Old Data

- Select the new node from the left pane, then enter slave machine's IP address and select 'Permanent Agent' and click okay.
- Now, proceed with filling further details of node
- In the 'remote root directory', add the path for a directory dedicated to be used by agent which should be /home/jenkins

example 1: if Slave OS is linux, and if user id is : root , then its home directory will be /root

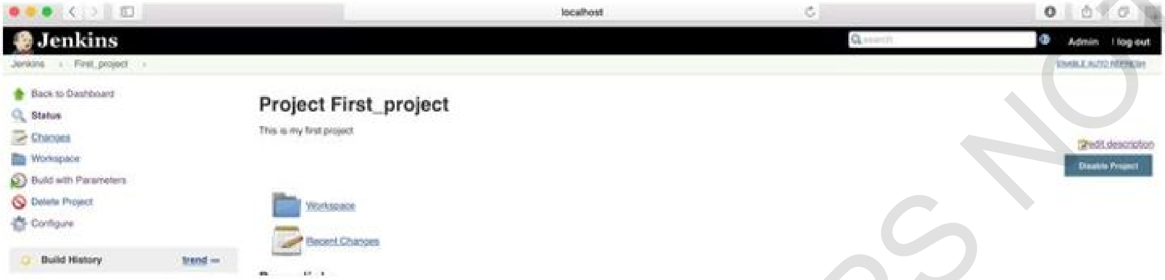
Example 2 : if Slave OS is linux, and if user id is : india , then its home directory will be /home

- In the launch method, select 'launch slave agents via ssh' and add the slave machine's ip address and credentials
- ** Lanch method --> if linux --> SSH option and if windows--> java webstart

Configure Project to build on slave

Step 1 -

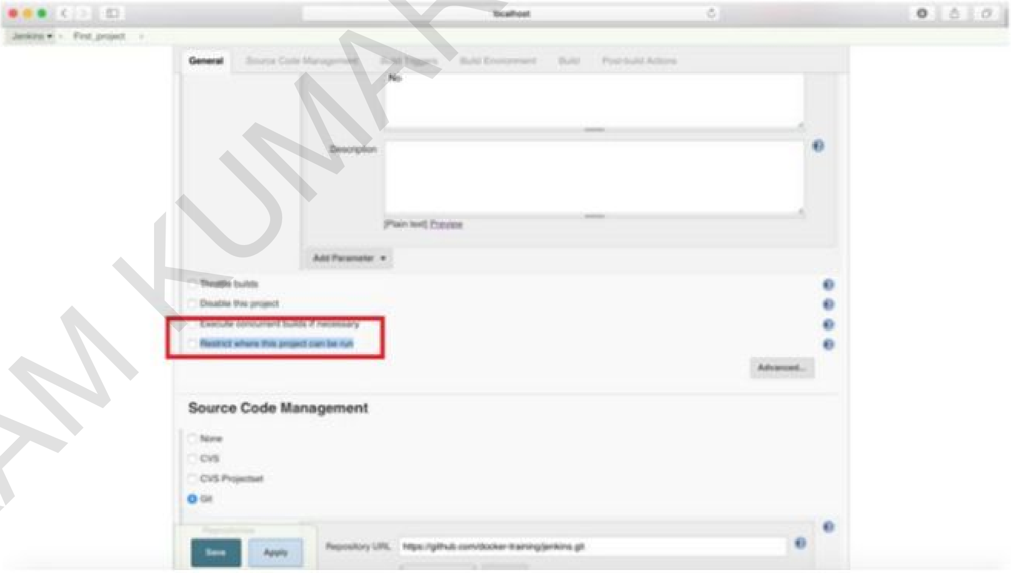
Open the **Project** and click on **Configure**



Find the check-box in General section "**Restrict where this project can be run**" and click on it

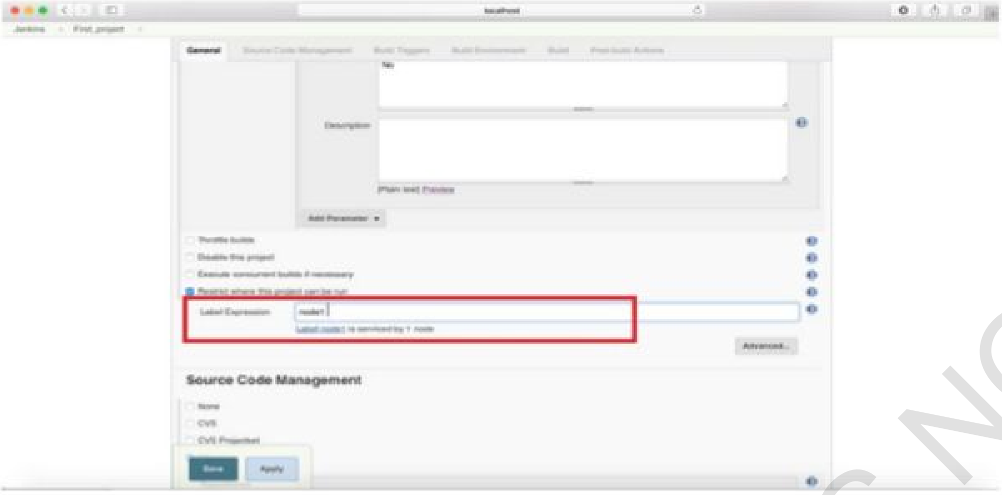
Step 2 -

Find the check-box in General section "**Restrict where this project can be run**" and click on it



Step 3 -

Just type the **Label** name of your slave



Label Expression: node1

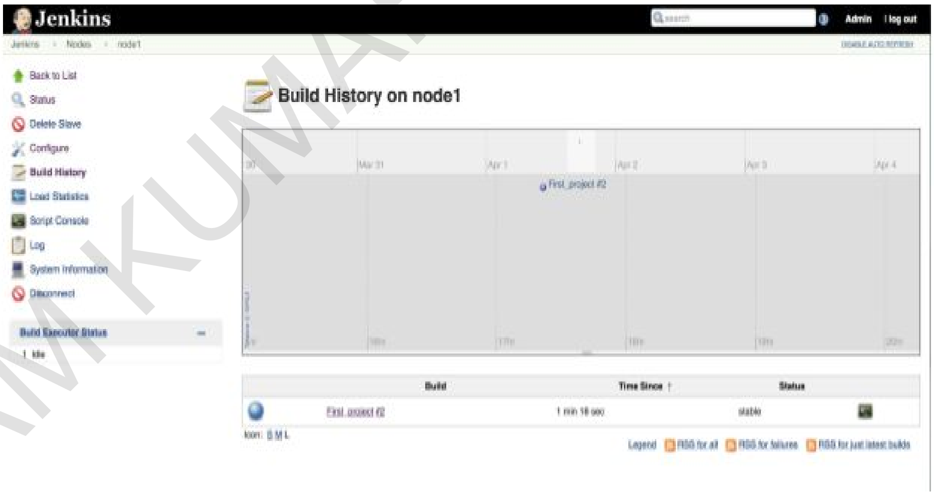
Label node1 is serviced by 1 node

And then **Save** the changes.

- You can also define more than one labels (i.e. more than one node to the project) which will allow you to run the build in parallel and Jenkins will choose the faster slave for your build.

Step 4 - Now, execute the build from Manage Jenkins > Manage Nodes > Select the node > select project to build

- You can also check the project build statistics



Build History on node1

	Build	Time Since ↑	Status	
●	First_project #2	1 min 18 sec	stable	

Icon: S M L

<u>Tomcat web server:</u>

Step 1 : Download Tomcat ZIP file from official site
Step 2 : Unzip in drive c
Step 3 : Go into the unzipped folder of tomcat, then go into the webapps folder and copy jenkins.war file into it.

Step 4 : Open Command prompt and navigate to tomcat’s bin folder. And then command, # startup.bat
Step 5: then open any browser, open jenkins using link
http://localhost:8080/jenkins