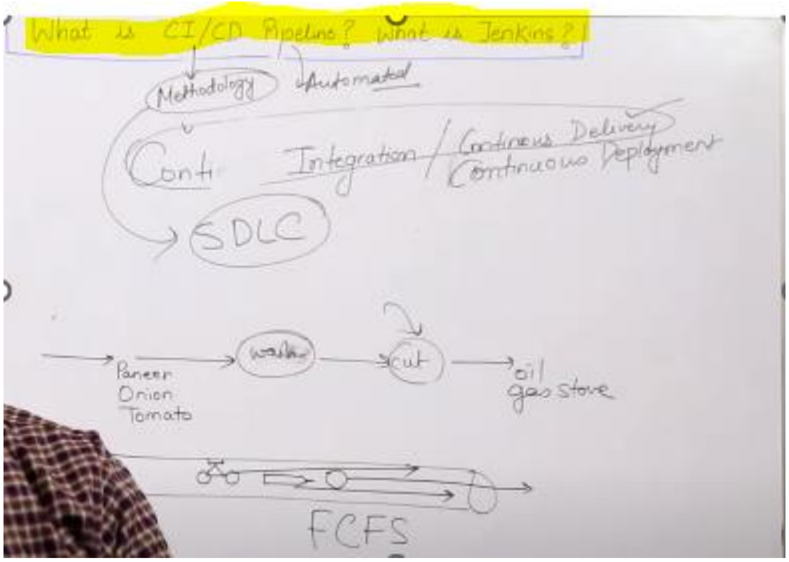
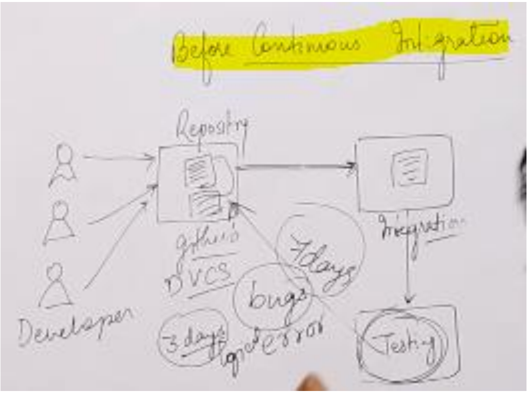
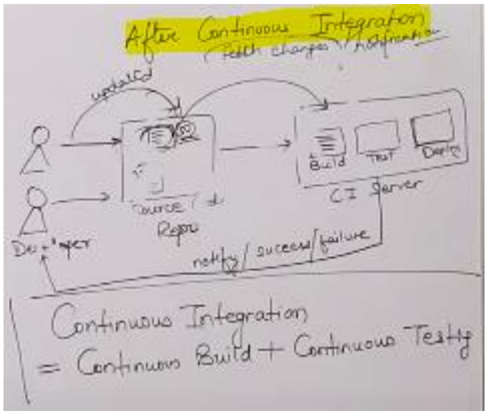
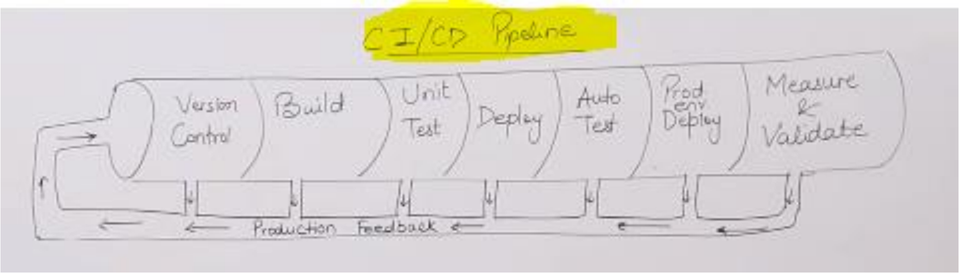
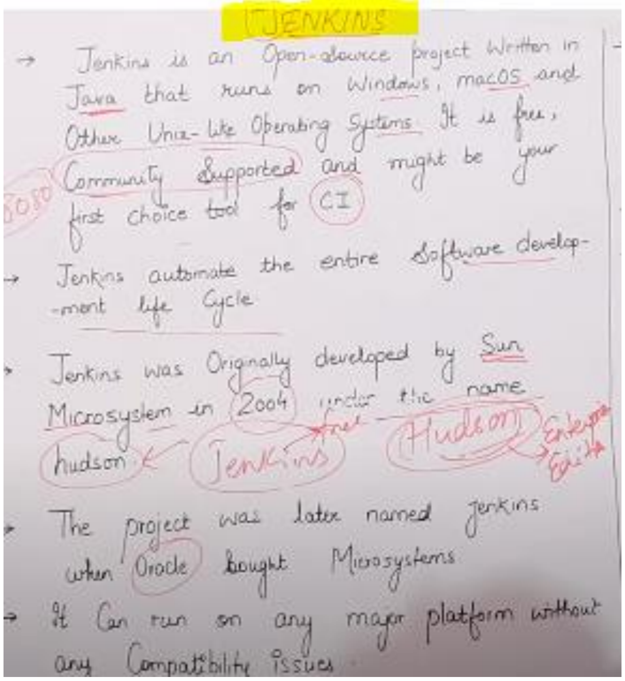
What is Jenkins and how it works | CI/CD Pipeline using Jenkins











JENKINS

→ Jenkins is an Open-source project Written in Java that runs on Windows, macOS and Other Unix-like Operating Systems. It is free, Community Supported and might be your first choice tool for CI

8080

→ Jenkins automate the entire Software development life Cycle

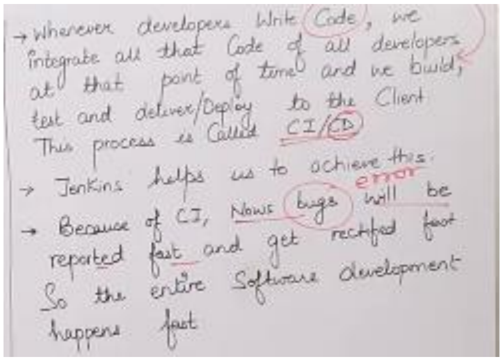
→ Jenkins was Originally developed by Sun Microsystem in 2004 under the name hudson

Jenkins — free

Hudson → Enterprise Edition

→ The project was later named Jenkins when Oracle bought Microsystems

→ It Can run on any major platform without any Compatibility issues



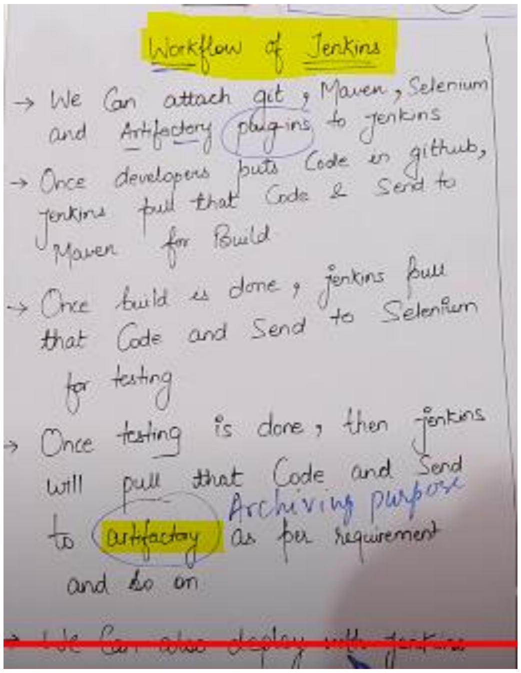
→ Whenever developers Write Code, we integrate all that Code of all developers at that point of time and we build, test and deliver/Deploy to the Client

This process is Called CI/CD

→ Jenkins helps us to achieve this.

→ Because of CI, Now bugs error will be reported fast and get rectified fast

So the entire Software development happens fast



Workflow of Jenkins

→ We Can attach git, Maven, Selenium and Artifactory plug-ins to Jenkins

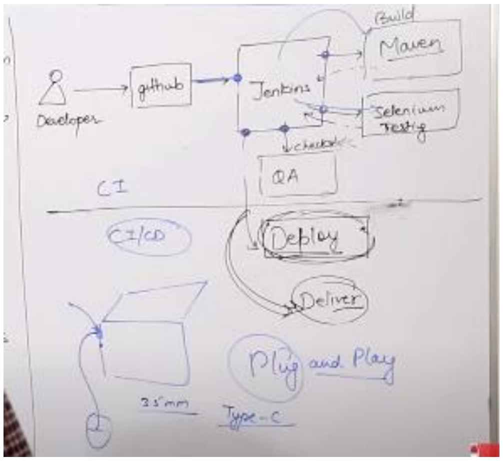
→ Once developers puts Code in github, Jenkins pull that Code & Send to Maven for Build

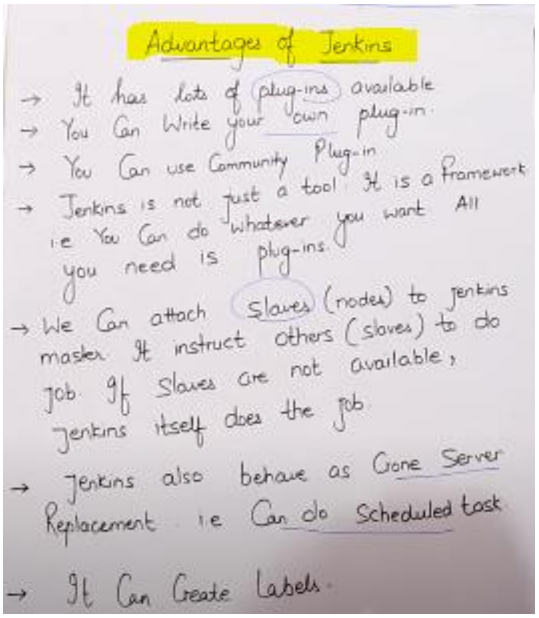
→ Once build is done, jenkins pull that Code and Send to Selenium for testing

→ Once testing is done, then jenkins will pull that Code and Send to artifactory as per requirement and so on

Archiving purpose

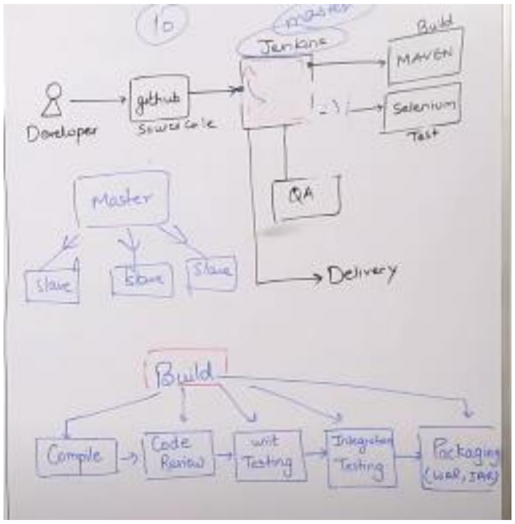
~~→ We Can also deploy with Jenkins~~





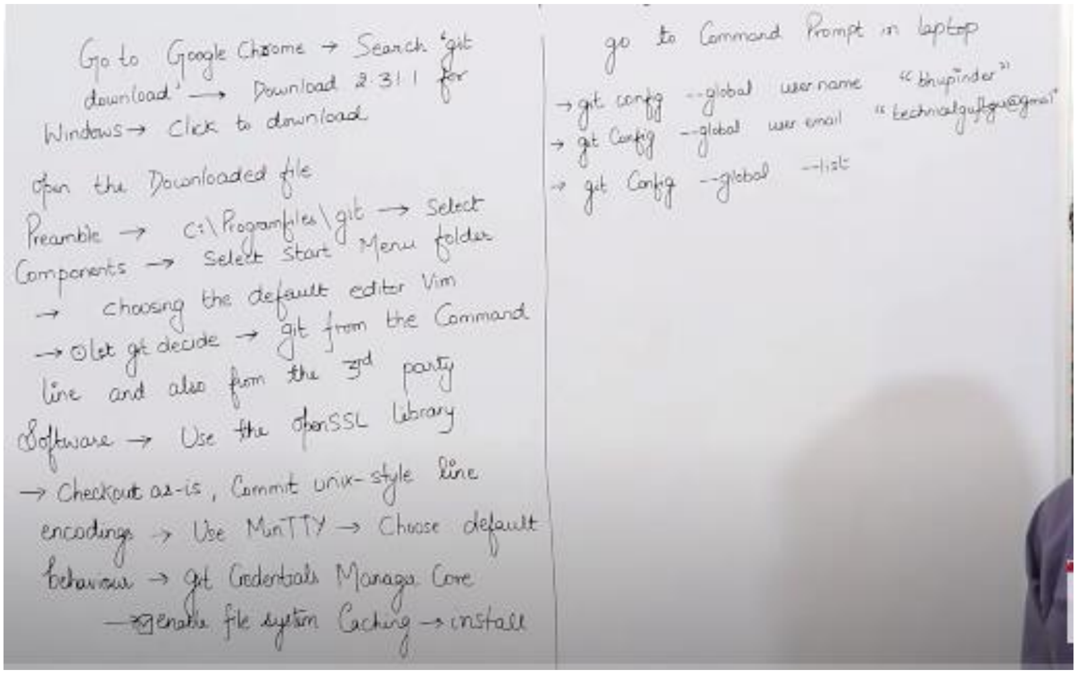
Advantages of Jenkins

→ It has lots of plug-ins available
→ You Can Write your own plug-in
→ You Can use Community Plug-in
→ Jenkins is not just a tool. It is a framework i.e You Can do whatever you want. All you need is plug-ins.
→ We Can attach Slaves (nodes) to jenkins master. It instruct others (slaves) to do job. If Slaves are not available, Jenkins itself does the job.
→ Jenkins also behave as Crone Server Replacement. i.e Can do Scheduled task
→ It Can Create Labels.



CI/CD Pipeline | What is Jenkins and Maven

GIT SETUP AND INSTALLATION (WINDOWS)

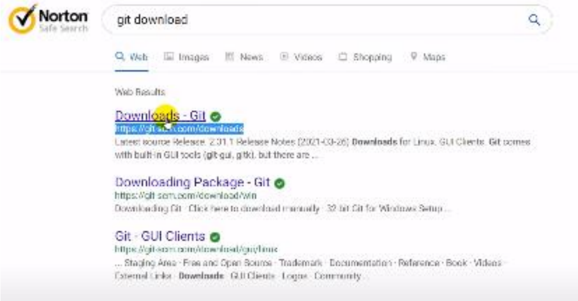


Go to Google Chrome → Search 'git download' → Download 2.31.1 for Windows → Click to download

Open the Downloaded file
Preamble → C:\Programfiles\git → Select Components → Select Start Menu folder → choosing the default editor Vim → ○ Let git decide → git from the Command line and also from the 3rd party Software → Use the OpenSSL library → Checkout as-is, Commit unix-style line encodings → Use MinTTY → Choose default behaviour → git Credentials Manager Core → ☑ enable file system Caching → install

go to Command Prompt in laptop

→ git config --global user.name "bhupinder"
→ git Config --global user.email "technicalguftgu@gmai
→ git Config --global --list

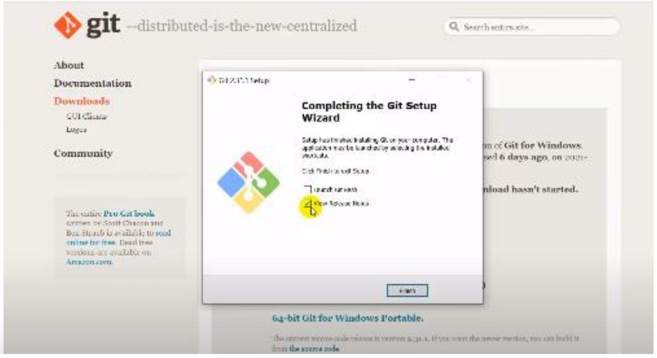


Web Results

Downloads - Git
https://git-scm.com/downloads
Latest source Release 2.31.1 Release Notes (2021-03-26) **Downloads** for Linux. GUI Clients. **Git** comes with built-in GUI tools (git-gui, gitk), but there are ...

Downloading Package - Git
https://git-scm.com/download/win
Downloading Git · Click here to download manually · 32-bit Git for Windows Setup ...

Git - GUI Clients
https://git-scm.com/download/gui/linux
... Staging Area · Free and Open Source · Trademark · Documentation · Reference · Book · Videos · External Links · **Downloads** · GUI Clients · Logos · Community



Git 2.31.1 Setup

Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

□ Launch Git Bash
☑ View Release Notes

The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

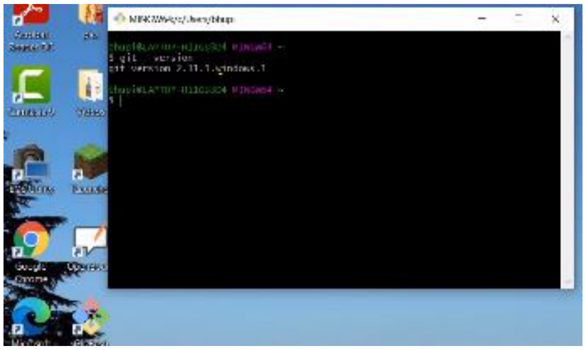
...n of Git for Windows ...sed 6 days ago, on 2021-

...nload hasn't started.

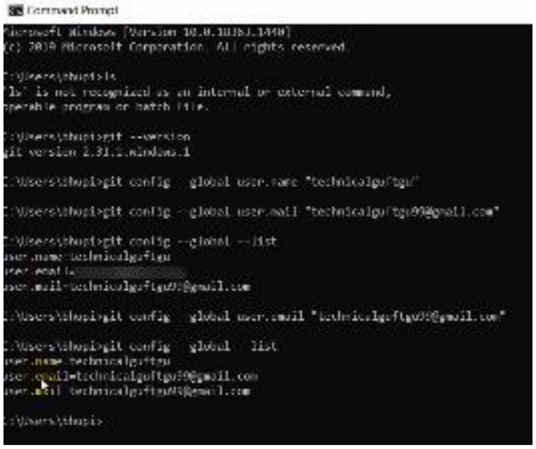
64-bit Git for Windows Portable.

The current source code release is version 2.31.1. If you want the newer version, you can build it from the source code.

GIT BASH (FOR WINDOWS)

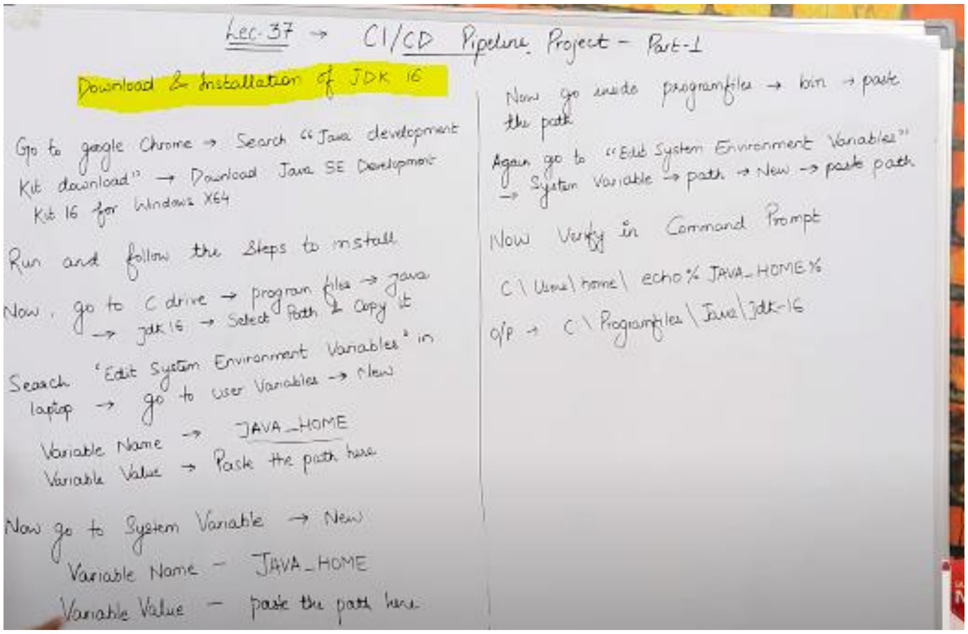

MINGW64:/c/Users/bhupi

```
bhupi@LAPTOP-H1I0S3O4 MINGW64 ~
$ git --version
git version 2.31.1.windows.1

bhupi@LAPTOP-H1I0S3O4 MINGW64 ~
$ |
```


Command Prompt

```
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\bhupi>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\bhupi>git --version
git version 2.31.1.windows.1

C:\Users\bhupi>git config --global user.name "technicalguftgu"

C:\Users\bhupi>git config --global user.mail "technicalguftgu99@gmail.com"

C:\Users\bhupi>git config --global --list
user.name=technicalguftgu
user.email=[redacted]
user.mail=technicalguftgu99@gmail.com

C:\Users\bhupi>git config --global user.email "technicalguftgu99@gmail.com"

C:\Users\bhupi>git config --global --list
user.name=technicalguftgu
user.email=technicalguftgu99@gmail.com
user.mail=technicalguftgu99@gmail.com

C:\Users\bhupi>
```

DOWNLOAD AND INSTALLATION OF JAVA :


Lec-37 → CI/CD Pipeline Project – Part-1

Download & Installation of JDK 16

Go to google Chrome → Search "Java development Kit download" → Download Java SE Development Kit 16 for Windows X64

Run and follow the Steps to install

Now, go to C drive → program files → java → jdk 16 → Select Path & Copy it

Search 'Edit System Environment Variables' in laptop → go to user Variables → New

Variable Name → JAVA_HOME
Variable Value → Paste the path here

Now go to System Variable → New
Variable Name – JAVA_HOME
Variable Value – paste the path here

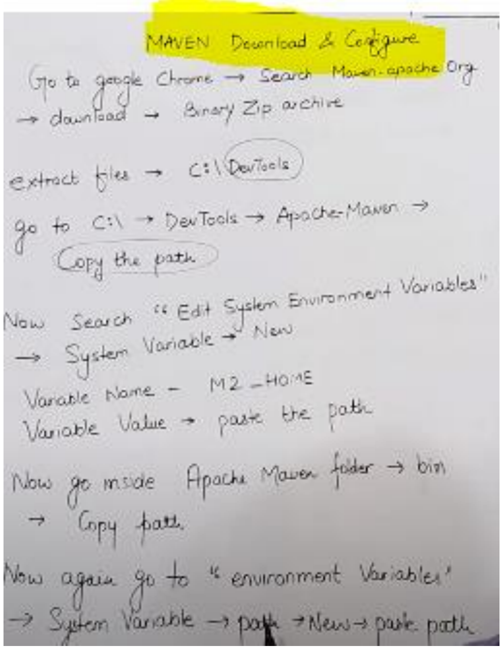
Now go inside programfiles → bin → paste the path

Again go to "Edit System Environment Variables" → System Variable → path → New → paste path

Now Verify in Command Prompt

C:\Users\home\ echo %JAVA_HOME%

o/p → C:\Programfiles\Java\Jdk-16



MAVEN Download & Configure

Go to google Chrome → Search Maven-apache Org → download → Binary Zip archive

extract files → C:\ DevTools

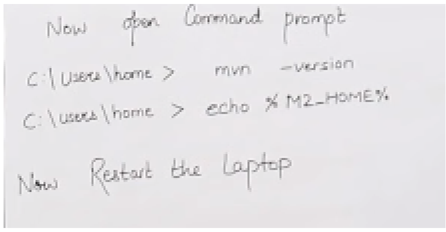
go to C:\ → DevTools → Apache-Maven → Copy the path

Now Search "Edit System Environment Variables" → System Variable → New

Variable Name - M2_HOME
Variable Value → paste the path

Now go inside Apache Maven folder → bin → Copy path

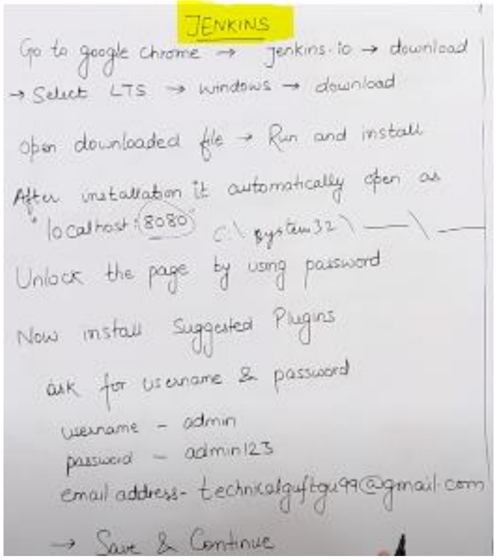
Now again go to "environment Variables" → System Variable → path → New → paste path



Now open Command prompt

```
C:\Users\home > mvn --version
C:\users\home > echo %M2_HOME%
```

Now Restart the laptop



JENKINS

Go to google chrome → jenkins.io → download → Select LTS → windows → download

open downloaded file → Run and install

After installation it automatically open as "localhost:8080" C:\ system32 \ —— \ ——

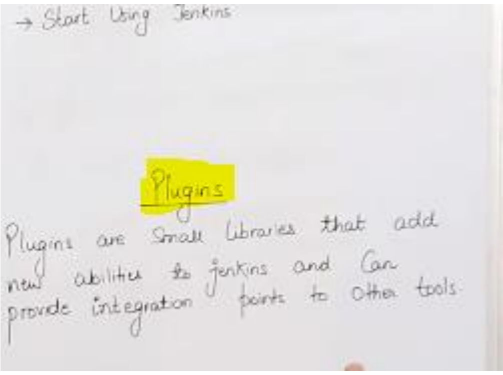
Unlock the page by using password

Now install Suggested Plugins

ask for username & password

username - admin
password - admin123
email address- technicalguftgu99@gmail.com

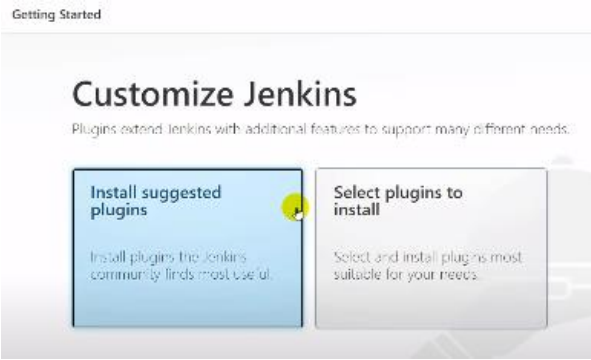
→ Save & Continue



→ Start Using Jenkins

Plugins

Plugins are Small Libraries that add new abilities to Jenkins and Can provide integration points to other tools.

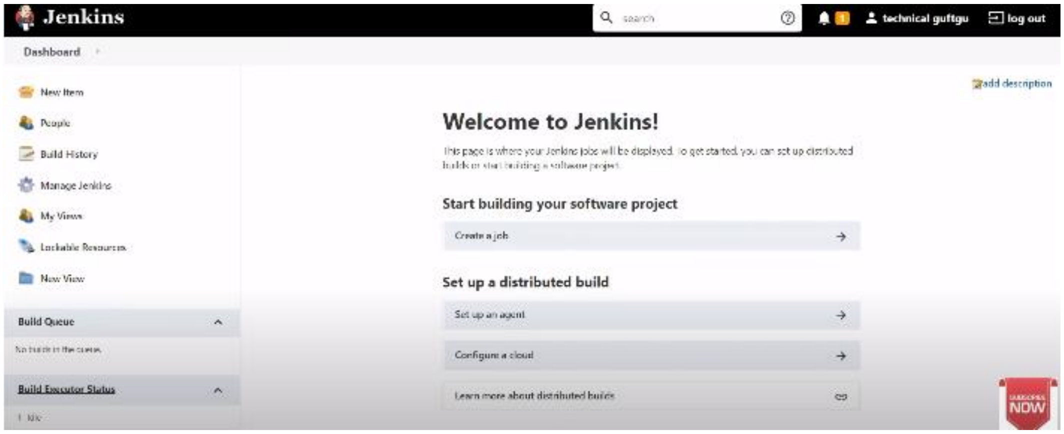


Getting Started

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.



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

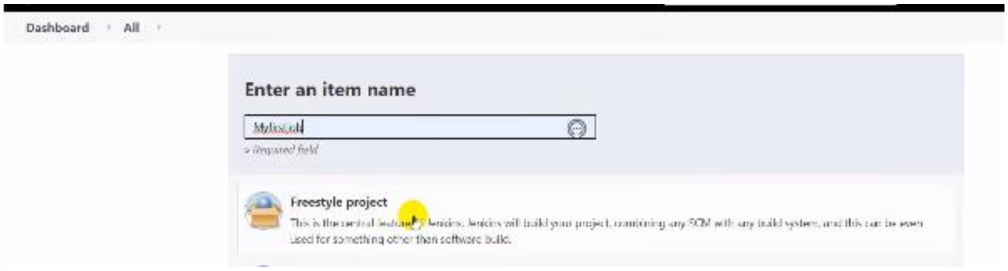
Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds



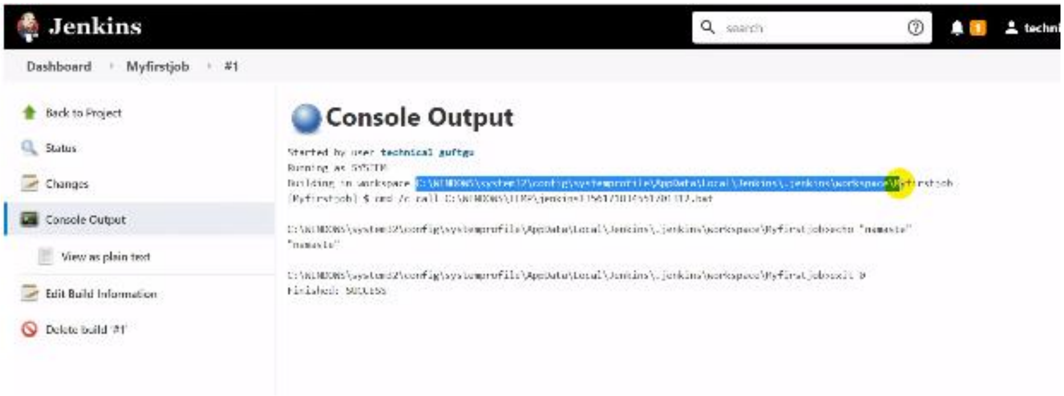
Enter an item name

Myfirstjob

» Required field

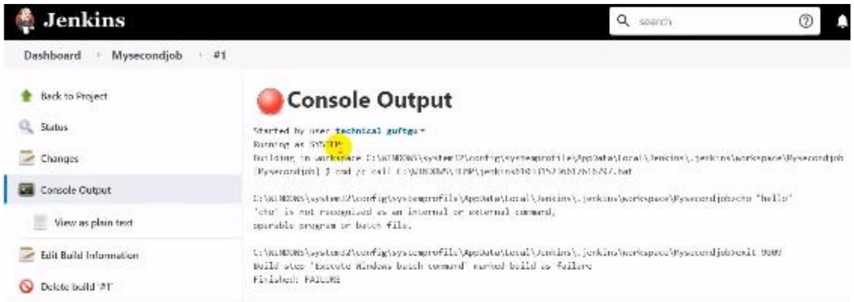
Freestyle project

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.



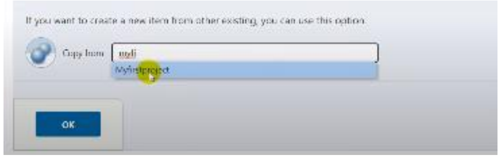
Console Output

```
Started by user technical guftgu
Running as SYSTEM
Building in workspace C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Myfirstjob
[Myfirstjob] $ cmd /c call C:\WINDOWS\TEMP\jenkins[illegible].bat

C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Myfirst_job>echo "namaste"
"namaste"

C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Myfirst_job>exit 0
Finished: SUCCESS
```



Console Output

```
Started by user technical guftgu
Running as SYSTEM
Building in workspace C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Mysecondjob
[Mysecondjob] $ cmd /c call C:\WINDOWS\TEMP\jenkins[illegible].bat

C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Mysecondjob>ehco "hello"
'eho' is not recognized as an internal or external command,
operable program or batch file.

C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Mysecondjob>exit 9009
Build step 'Execute Windows batch command' marked build as failure
Finished: FAILURE
```

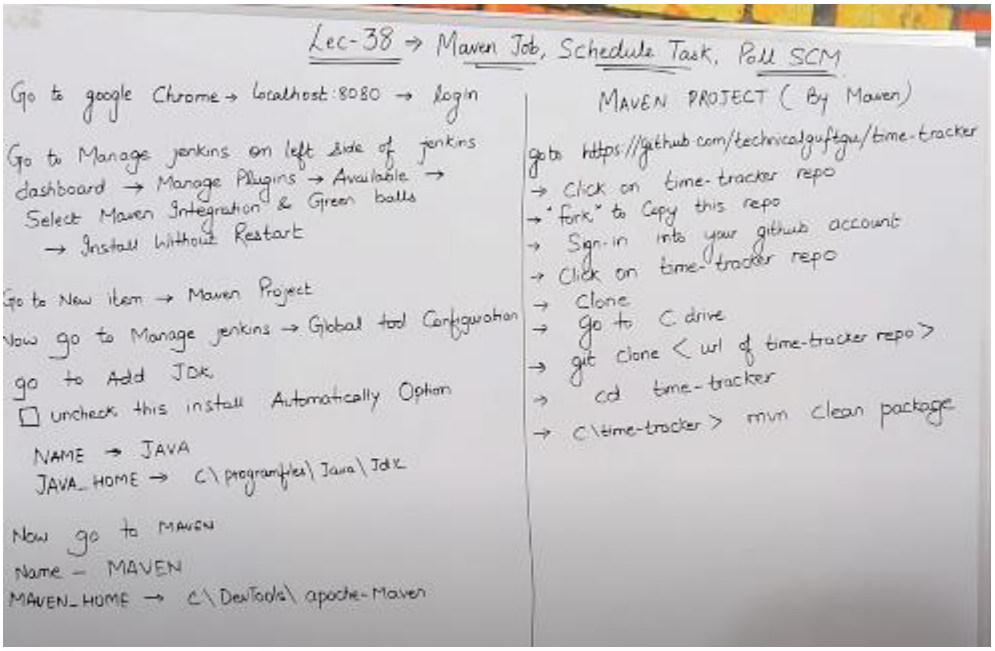

If you want to create a new item from other existing, you can use this option.

Copy from myfi

Myfirstproject

OK

Maven job, schedule task and Source code Polling| Jenkins and Maven Basic | DevOps



Lec-38 → Maven Job, Schedule Task, Poll SCM

Go to google Chrome → localhost:8080 → login

Go to Manage jenkins on left side of jenkins dashboard → Manage Plugins → Available → Select Maven Integration & Green balls → Install Without Restart

Go to New item → Maven Project

Now go to Manage jenkins → Global tool Configuration

go to Add JDK

□ uncheck this install Automatically Option

NAME → JAVA
JAVA_HOME → C:\programfiles\Java\Jdk

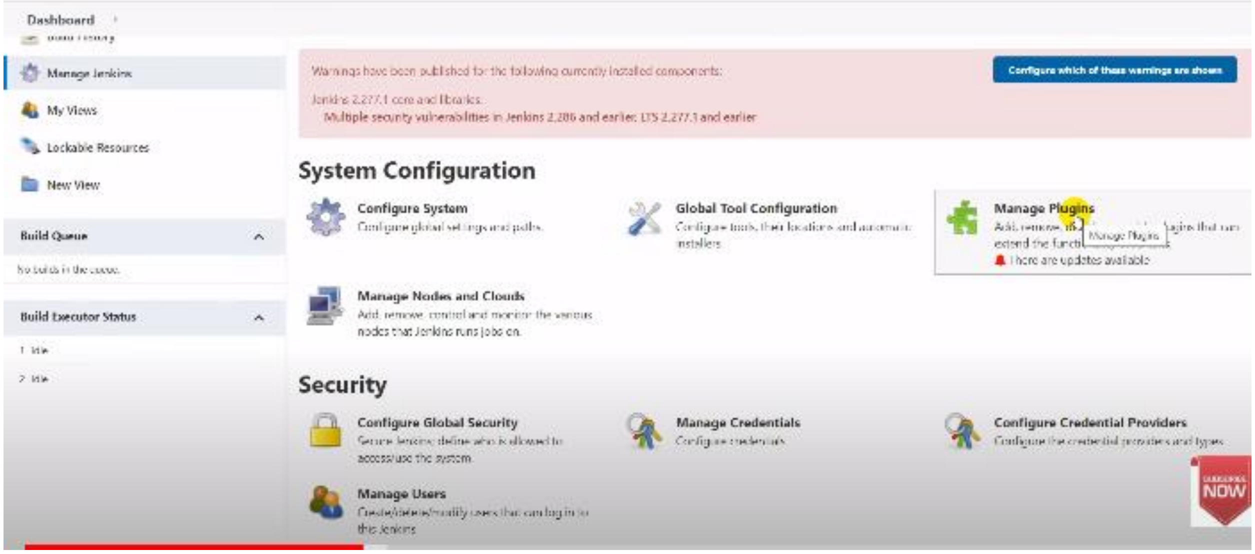
Now go to MAVEN

Name – MAVEN
MAVEN_HOME → C:\DevTools\apache-Maven

MAVEN PROJECT (By Maven)

goto https://github.com/technicalguftgu/time-tracker

→ Click on time-tracker repo
→ "fork" to Copy this repo
→ Sign-in into your github account
→ Click on time-tracker repo
→ Clone
→ go to C drive
→ git clone < url of time-tracker repo >
→ cd time-tracker
→ C:\time-tracker > mvn clean package



No builds in the queue.

1 Idle

2 Idle

Warnings have been published for the following currently installed components:

Jenkins 2.277.1 core and libraries:
Multiple security vulnerabilities in Jenkins 2.286 and earlier, LTS 2.277.3 and earlier

System Configuration

Configure System
Configure global settings and paths.

Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
There are updates available

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

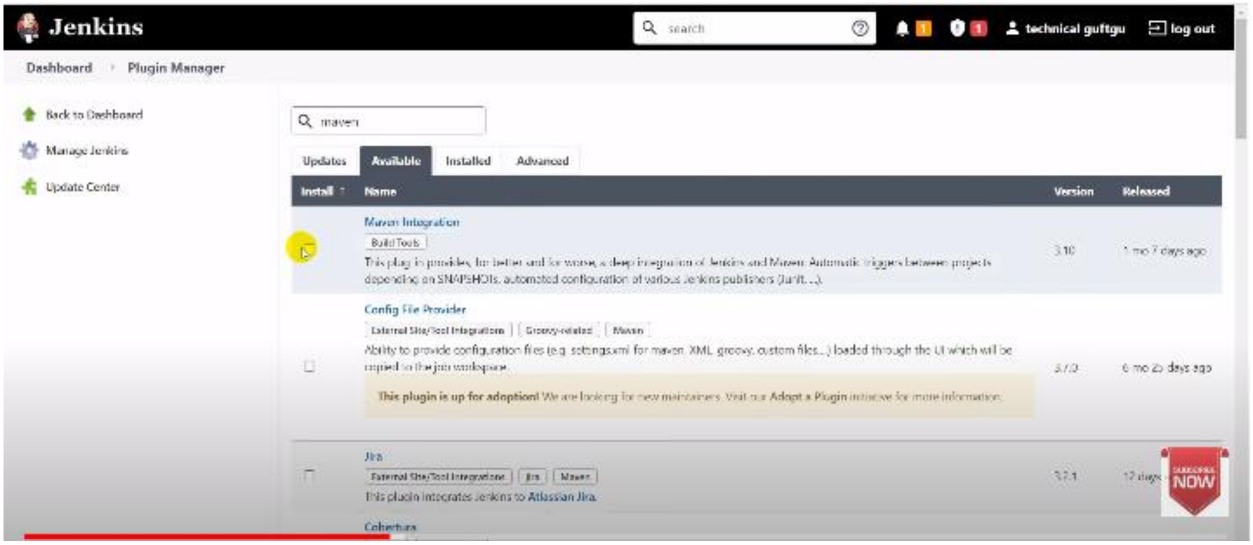
Security

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

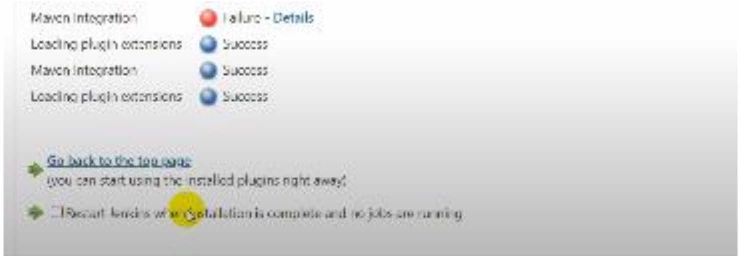
Manage Credentials
Configure credentials

Configure Credential Providers
Configure the credential providers and types

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



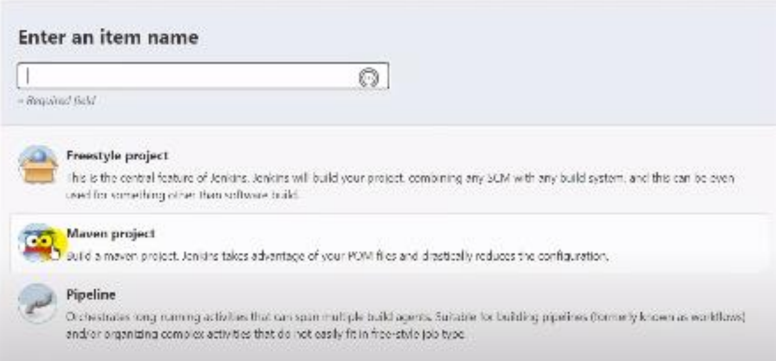
Install ↑	Name	Version	Released
□	Maven Integration Build Tools This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...).	3.10	1 mo 7 days ago
□	Config File Provider External Site/Tool Integrations | Groovy-related | Maven Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace. **This plugin is up for adoption!** We are looking for new maintainers. Visit our **Adopt a Plugin** initiative for more information.	3.7.0	6 mo 25 days ago
□	Jira External Site/Tool Integrations | Jira | Maven This plugin integrates Jenkins to Atlassian Jira.	3.7.1	12 days
	Cobertura		



Maven Integration ● Failure - Details
Loading plugin extensions ● Success
Maven Integration ● Success
Loading plugin extensions ● Success

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



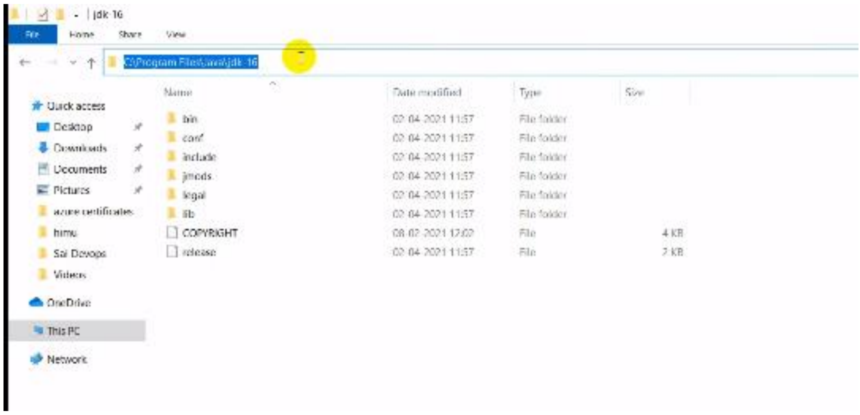
Enter an item name

» Required field

Freestyle 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.

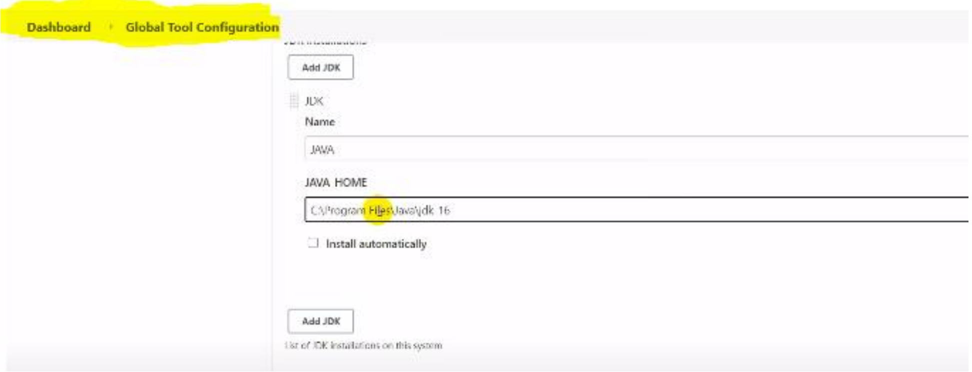
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



C:\Program Files\Java\jdk-16

Name	Date modified	Type	Size
bin	02-04-2021 11:57	File folder	
conf	02-04-2021 11:57	File folder	
include	02-04-2021 11:57	File folder	
jmods	02-04-2021 11:57	File folder	
legal	02-04-2021 11:57	File folder	
lib	02-04-2021 11:57	File folder	
COPYRIGHT	08-02-2021 12:02	File	4 KB
release	02-04-2021 11:57	File	2 KB

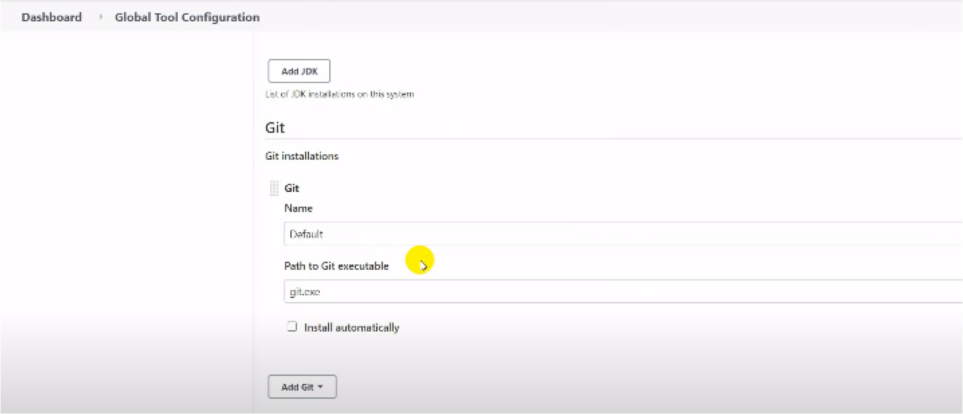


Name: JAVA

JAVA_HOME: C:\Program Files\Java\jdk-16

☐ Install automatically

List of JDK installations on this system



List of JDK installations on this system

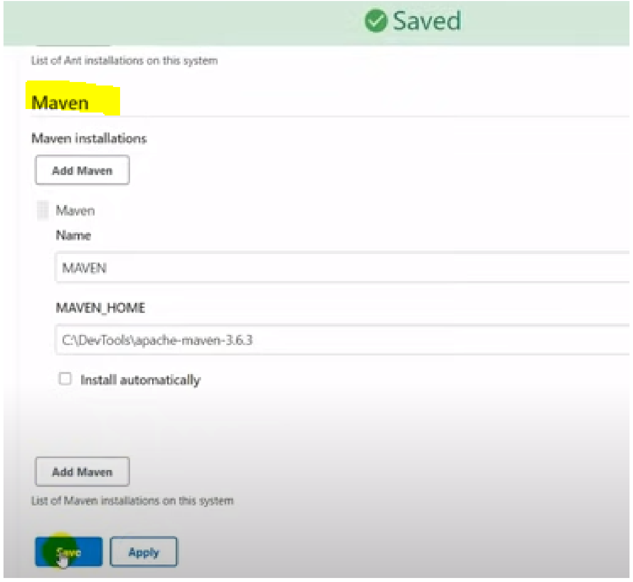
Git

Git installations

Name: Default

Path to Git executable: git.exe

☐ Install automatically



List of Ant installations on this system

Maven

Maven installations

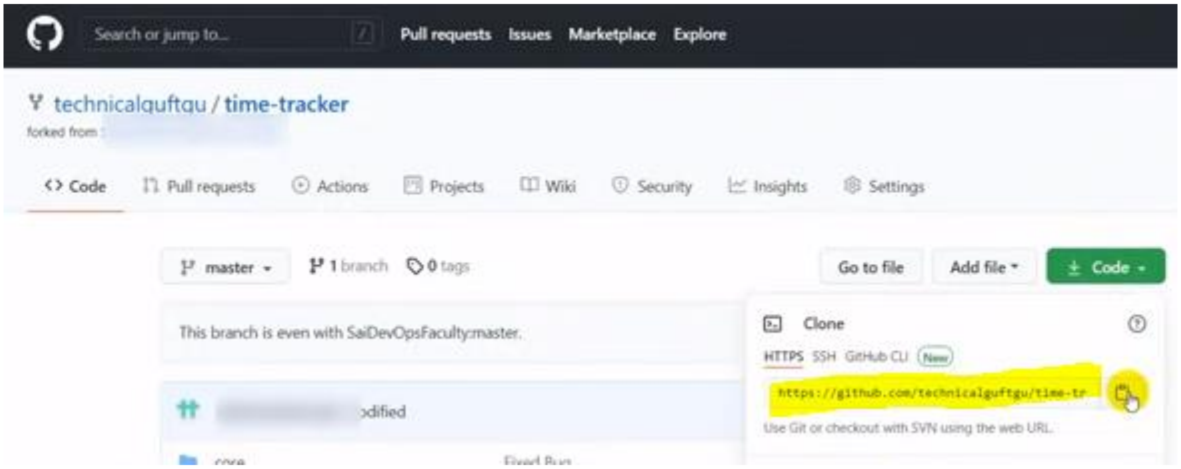
Maven

Name: MAVEN

MAVEN_HOME: C:\DevTools\apache-maven-3.6.3

□ Install automatically

List of Maven installations on this system



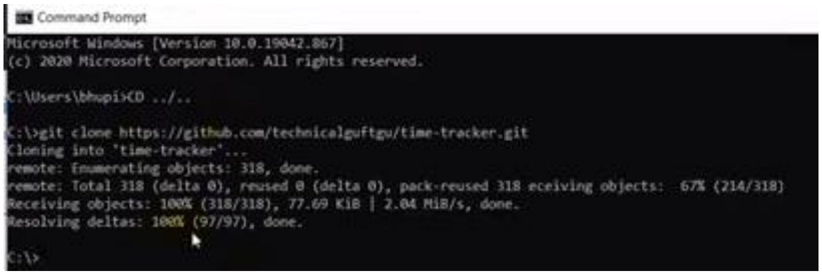
This branch is even with SaiDevOpsFaculty:master.

[redacted] odified

core Fixed Bug

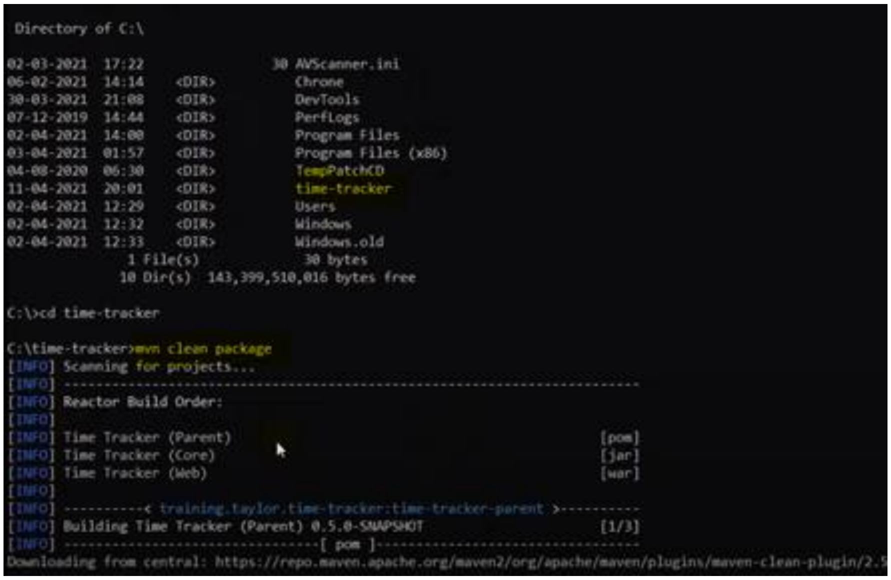
https://github.com/technicalguftgu/time-tr

Use Git or checkout with SVN using the web URL.

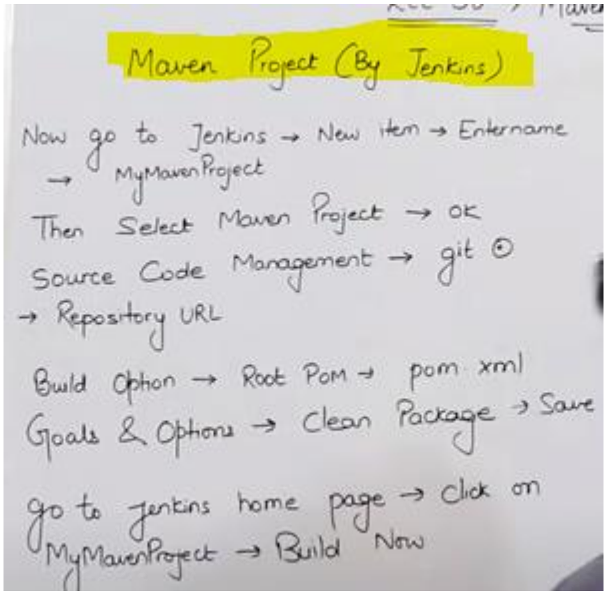


```
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\bhupi>CD ../..

C:\>git clone https://github.com/technicalguftgu/time-tracker.git
Cloning into 'time-tracker'...
remote: Enumerating objects: 318, done.
remote: Total 318 (delta 0), reused 0 (delta 0), pack-reused 318 eceiving objects:  67% (214/318)
Receiving objects: 100% (318/318), 77.69 KiB | 2.04 MiB/s, done.
Resolving deltas: 100% (97/97), done.

C:\>
```

Build (code is convert into machine level language)



```
 Directory of C:\

02-03-2021  17:22                30 AVScanner.ini
06-02-2021  14:14    <DIR>          Chrome
30-03-2021  21:08    <DIR>          DevTools
07-12-2019  14:44    <DIR>          PerfLogs
02-04-2021  14:00    <DIR>          Program Files
03-04-2021  01:57    <DIR>          Program Files (x86)
04-08-2020  06:30    <DIR>          TempPatchCD
11-04-2021  20:01    <DIR>          time-tracker
02-04-2021  12:29    <DIR>          Users
02-04-2021  12:32    <DIR>          Windows
02-04-2021  12:33    <DIR>          Windows.old
               1 File(s)             30 bytes
              10 Dir(s)  143,399,510,016 bytes free

C:\>cd time-tracker

C:\time-tracker>mvn clean package
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Build Order:
[INFO]
[INFO] Time Tracker (Parent)                                              [pom]
[INFO] Time Tracker (Core)                                                [jar]
[INFO] Time Tracker (Web)                                                 [war]
[INFO]
[INFO] ----------< training.taylor.time-tracker:time-tracker-parent >----------
[INFO] Building Time Tracker (Parent) 0.5.0-SNAPSHOT                      [1/3]
[INFO] --------------------------------[ pom ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5
```



Maven Project (By Jenkins)

Now go to Jenkins → New item → Entername
→ MyMavenProject

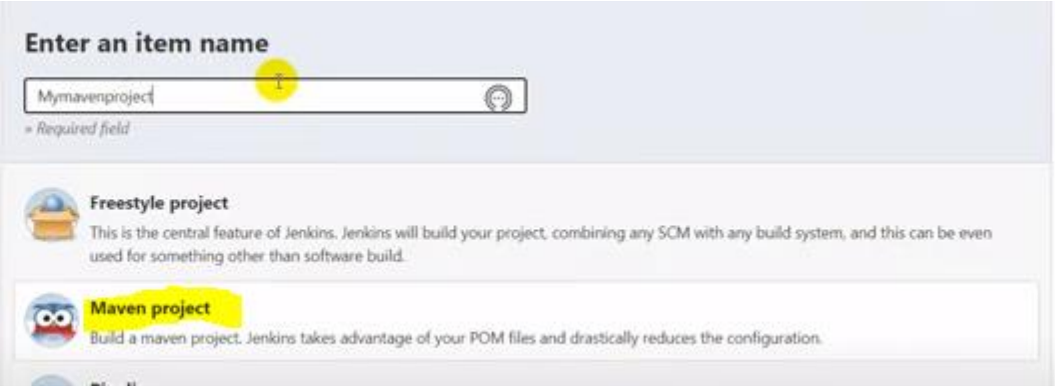
Then Select Maven Project → OK

Source Code Management → git ⊙
→ Repository URL

Build Option → Root Pom → pom xml

Goals & Options → Clean Package → Save

go to Jenkins home page → Click on
MyMavenProject → Build Now



Enter an item name

Mymavenproject

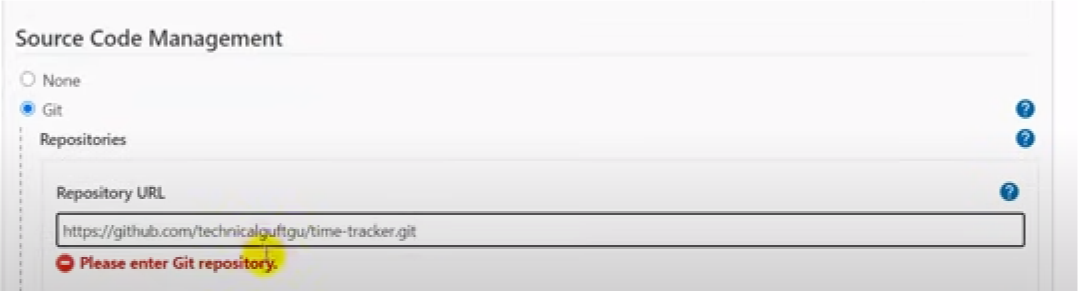
» Required field

Freestyle 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.

Maven project

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



Source Code Management

○ None

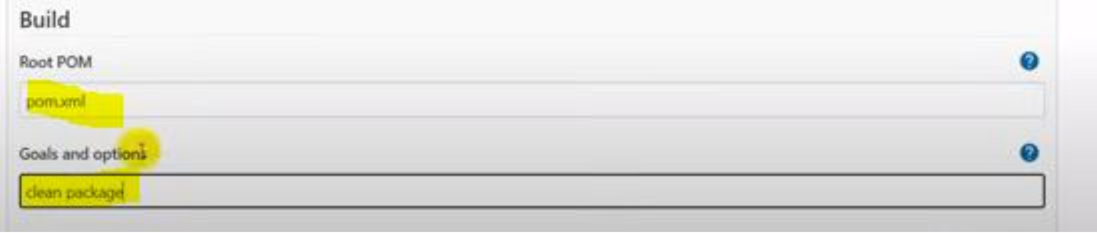
● Git

Repositories

Repository URL

https://github.com/technicalguftgu/time-tracker.git

Please enter Git repository.



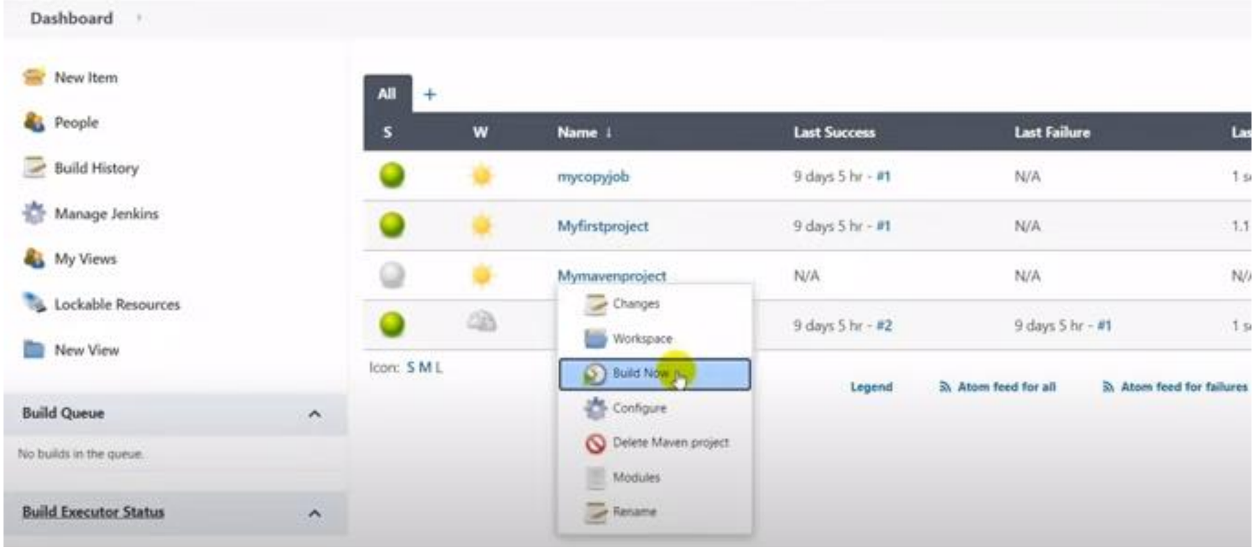
Build

Root POM

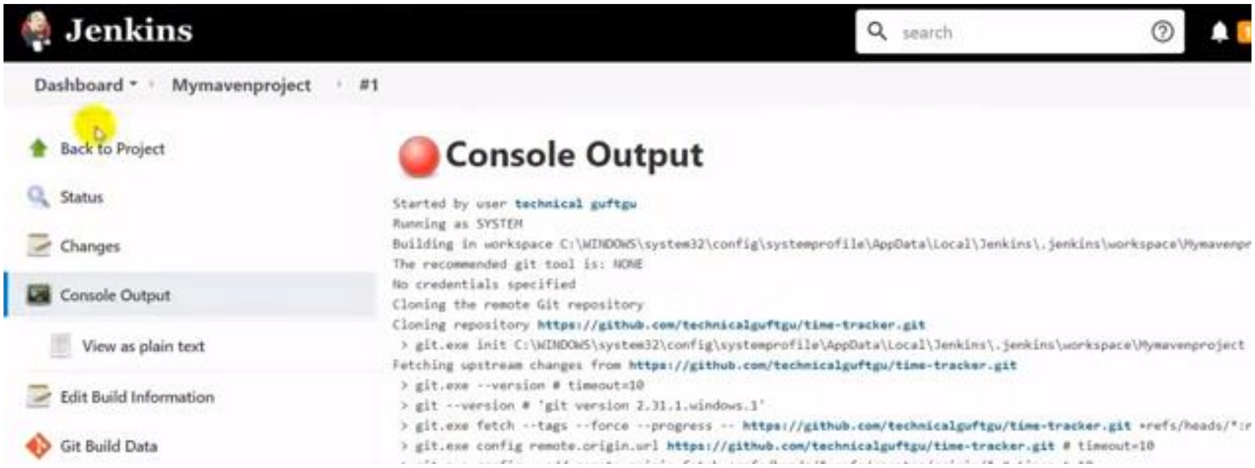
pom.xml

Goals and options

clean package

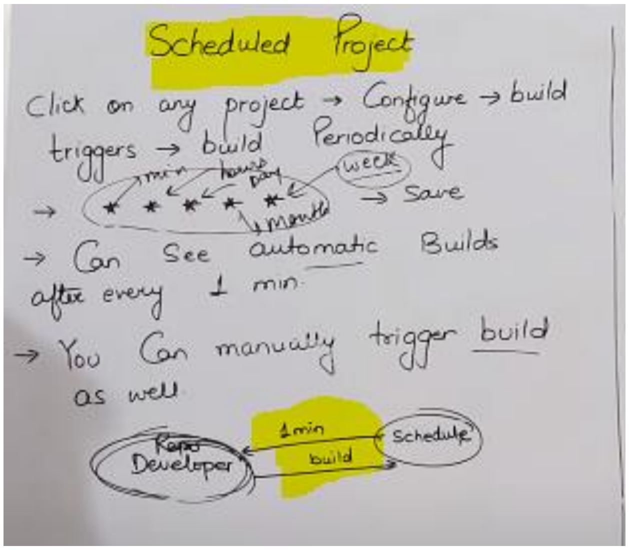


S	W	Name ↓	Last Success	Last Failure	Las
●	☀	mycopyjob	9 days 5 hr - #1	N/A	1 s
●	☀	Myfirstproject	9 days 5 hr - #1	N/A	1.1
○	☀	Mymavenproject	N/A	N/A	N/
●	☁		9 days 5 hr - #2	9 days 5 hr - #1	1 s



Console Output

```
Started by user technical guftgu
Running as SYSTEM
Building in workspace C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Mymavenpr
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/technicalguftgu/time-tracker.git
 > git.exe init C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Mymavenproject
Fetching upstream changes from https://github.com/technicalguftgu/time-tracker.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.31.1.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/technicalguftgu/time-tracker.git +refs/heads/*:r
 > git.exe config remote.origin.url https://github.com/technicalguftgu/time-tracker.git # timeout=10
```



Scheduled Project

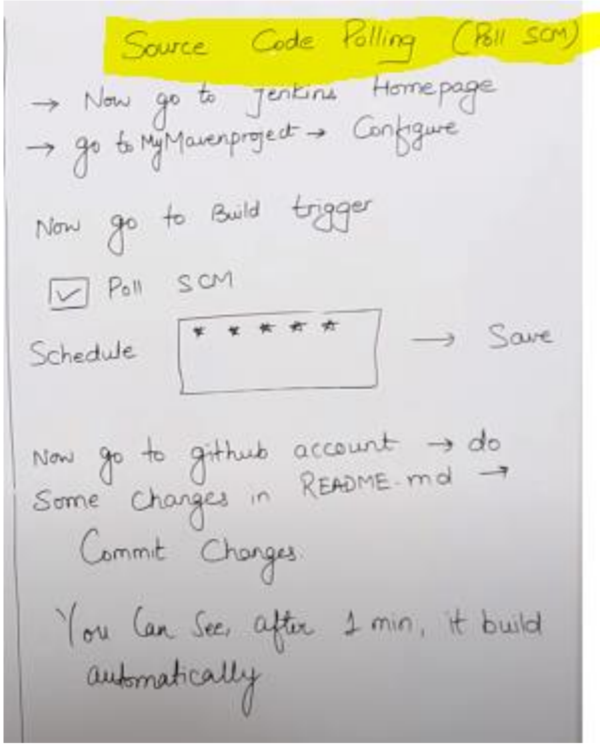
Click on any project → Configure → build triggers → build Periodically

→ * * * * * → Save

→ Can See automatic Builds after every 1 min.

→ You Can manually trigger build as well

min
hours
day
week
month
~~Repo~~
Developer
1 min
build
Schedule



Source Code Polling (Poll SCM)

→ Now go to Jenkins Homepage

→ go to MyMavenproject → Configure

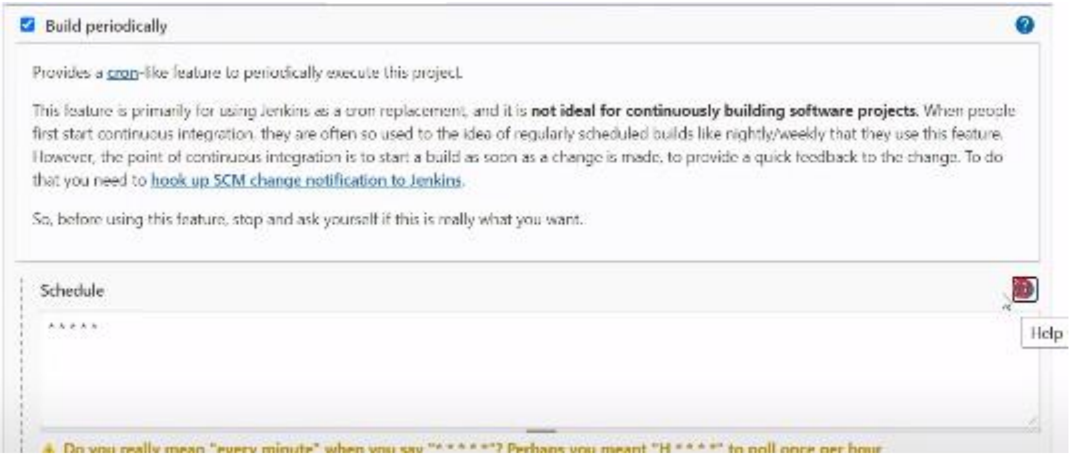
Now go to Build trigger

☑ Poll SCM

Schedule * * * * * → Save

Now go to github account → do Some changes in README.md → Commit Changes.

You Can See, after 1 min, it build automatically



☑ Build periodically

Provides a cron-like feature to periodically execute this project.

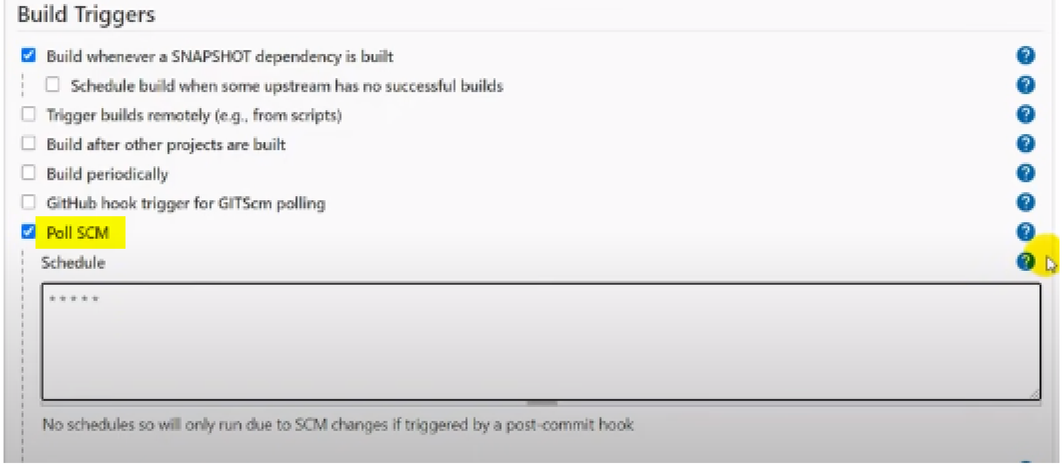
This feature is primarily for using Jenkins as a cron replacement, and it is **not ideal for continuously building software projects.** When people first start continuous integration, they are often so used to the idea of regularly scheduled builds like nightly/weekly that they use this feature. However, the point of continuous integration is to start a build as soon as a change is made, to provide a quick feedback to the change. To do that you need to hook up SCM change notification to Jenkins.

So, before using this feature, stop and ask yourself if this is really what you want.

Schedule

```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour



Build Triggers

☑ Build whenever a SNAPSHOT dependency is built

☐ Schedule build when some upstream has no successful builds

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

☐ Build after other projects are built

☐ Build periodically

☐ GitHub hook trigger for GITScm polling

☑ Poll SCM

Schedule

```
* * * * *
```

No schedules so will only run due to SCM changes if triggered by a post-commit hook