

# Continuous Integration (CI)

Maven

Jenkins

Git

GitHub

# CI Tools

- Maven
- Jenkins
- Git
- GitHub



# Maven

# Run test cases using Maven pom.xml

- You need to add 2 plug-ins to pom.xml File
  - maven-compiler-plugin
  - maven-surefire-plugin
- Right click on pom.xml → Run as Maven test
- **Issue:**
- **Error:** [ERROR] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?
- **Solution:** Eclipse Window → Preferences → Java → Installed JRE's → and check your installed JREs.
- You should have an entry with a JDK there.
- Select the Execution Environment as "*C:\Program Files\Java\jdk1.8.0\_151*" → Click OK

# Run test cases through Maven CLI

- **Install Maven on Windows OS**
  - Download Maven for Windows
  - <http://redrockdigimark.com/apachemirror/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.zip>
- **Add maven path to Windows environment variable.**
  - Right click on MyPC → Advanced System settings → Environment Variables
- **Check version of Maven installed on Windows.**
  - c: mvn -version
- **Issue:**
- **Error:** [ERROR] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?
- **Solution:** REFERE LINK: <http://roufid.com/no-compiler-is-provided-in-this-environment/>
- Run Test cases through command prompt.
  - cd C:\Users\admin\eclipse-workspace\InetBanking
  - mvn clean install

# Run automation through run.bat

- Create **run.bat** file which contains below entries

```
cd C:\Users\admin\eclipse-workspace\InetBanking  
mvn clean install
```

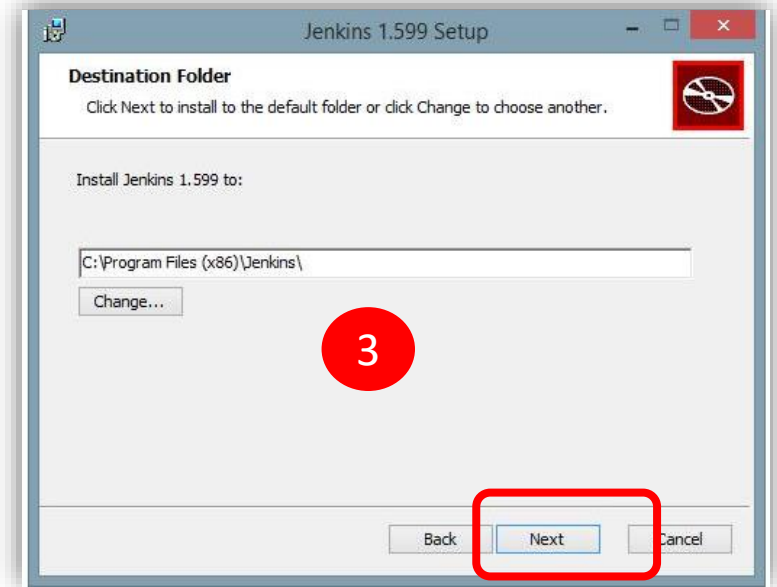
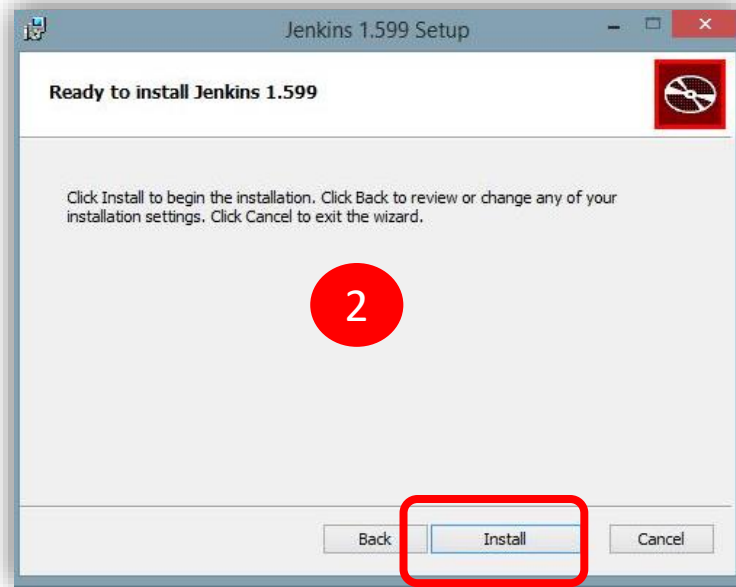
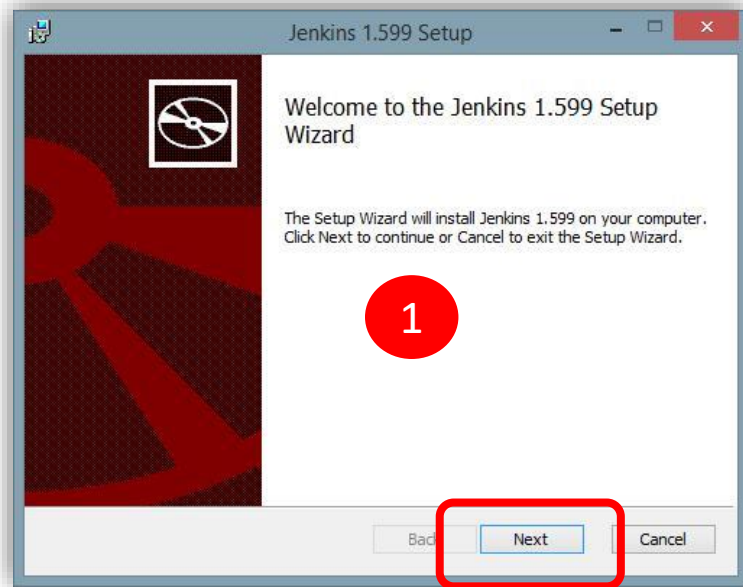


**Jenkins**

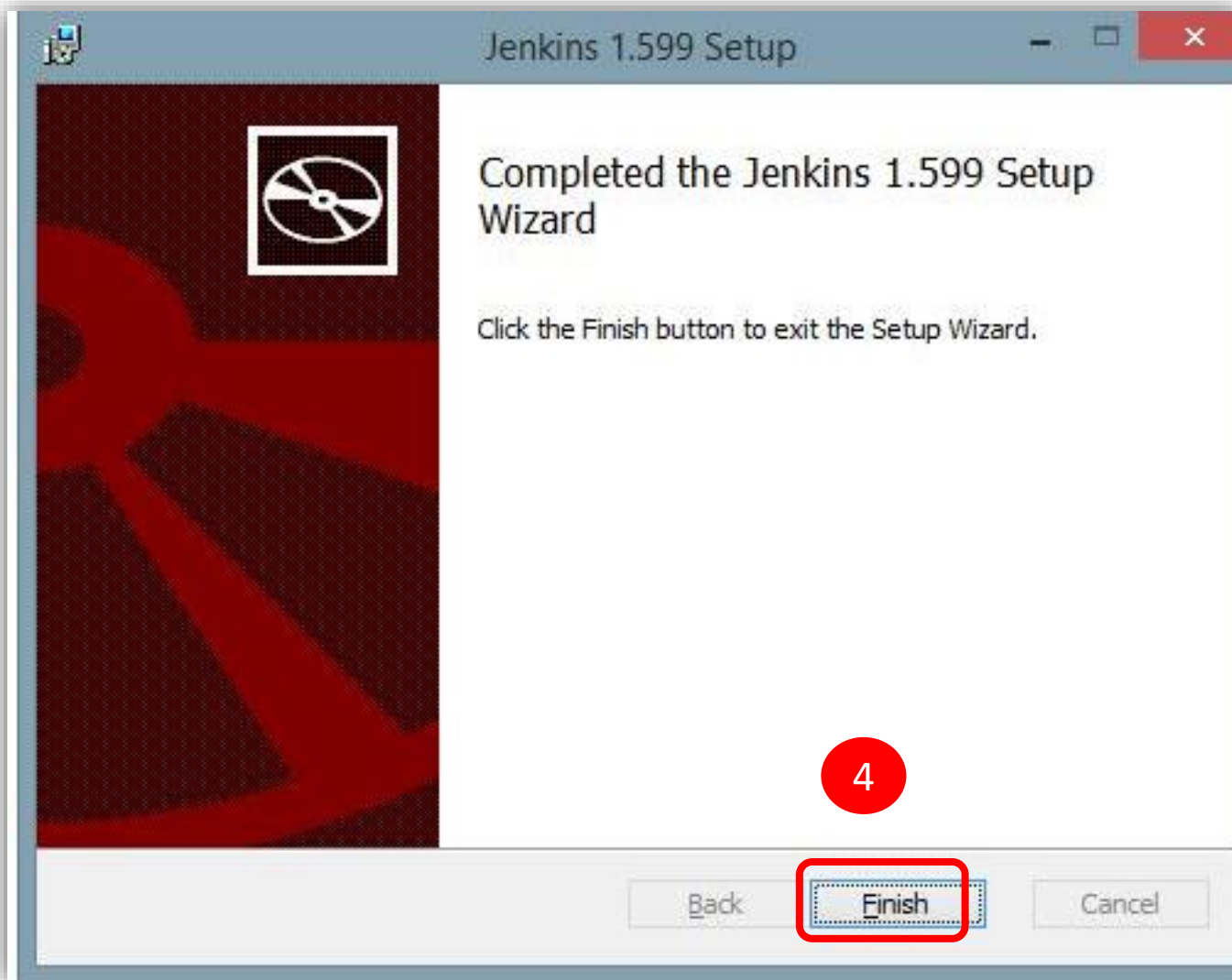
# Jenkins installation & Configuration

# Jenkins Step By step Installation

Download Link: <https://jenkins.io/download/>







# Jenkins Configuration

URL to open Jenkins: <http://localhost:8080>

Getting Started

## 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:

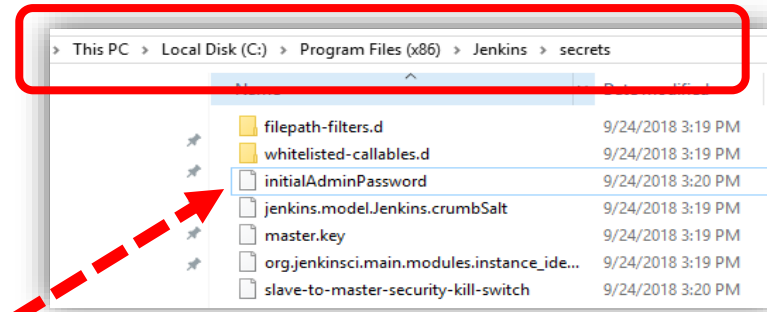
`C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword`

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

Administrator password

1

Continue



Getting Started

## 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:

`C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword`

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

Administrator password

2

# Jenkins Configuration

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.

**3**

**Select plugins to install**  
  
Select and install plugins most suitable for your needs.

Getting Started

## Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	Folders <ul style="list-style-type: none"><li>** JDK Tool</li><li>** Script Security</li><li>** Command Agent Launcher</li><li>** bouncycastle API</li><li>** Struts</li><li>** Pipeline: Step API</li><li>** SCM API</li><li>** Pipeline: API</li><li>** JUnit</li></ul>
✓ Timestampers	Workspace Cleanup	Ant	Gradle	OWASP Markup Formatter <ul style="list-style-type: none"><li>** Token Macro</li></ul>
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	Build Timeout <ul style="list-style-type: none"><li>** Credentials</li><li>** SSH Credentials</li><li>** Plain Credentials</li></ul>
Git	Subversion	SSH Slaves	Matrix Authorization Strategy	Credentials Binding
PAM Authentication	LDAP	Email Extension	Mailer	Timestampers <ul style="list-style-type: none"><li>** Pipeline: Supporting APIs</li><li>** Durable Task</li><li>** Pipeline: Nodes and Processes</li><li>** Matrix Project</li><li>** Resource Disposer</li></ul>
				Workspace Cleanup

# Jenkins Configuration

Getting Started

## Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

4

Jenkins 2.138.1

Continue as admin

Save and Continue

Getting Started

## Instance Configuration

Jenkins URL:

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.

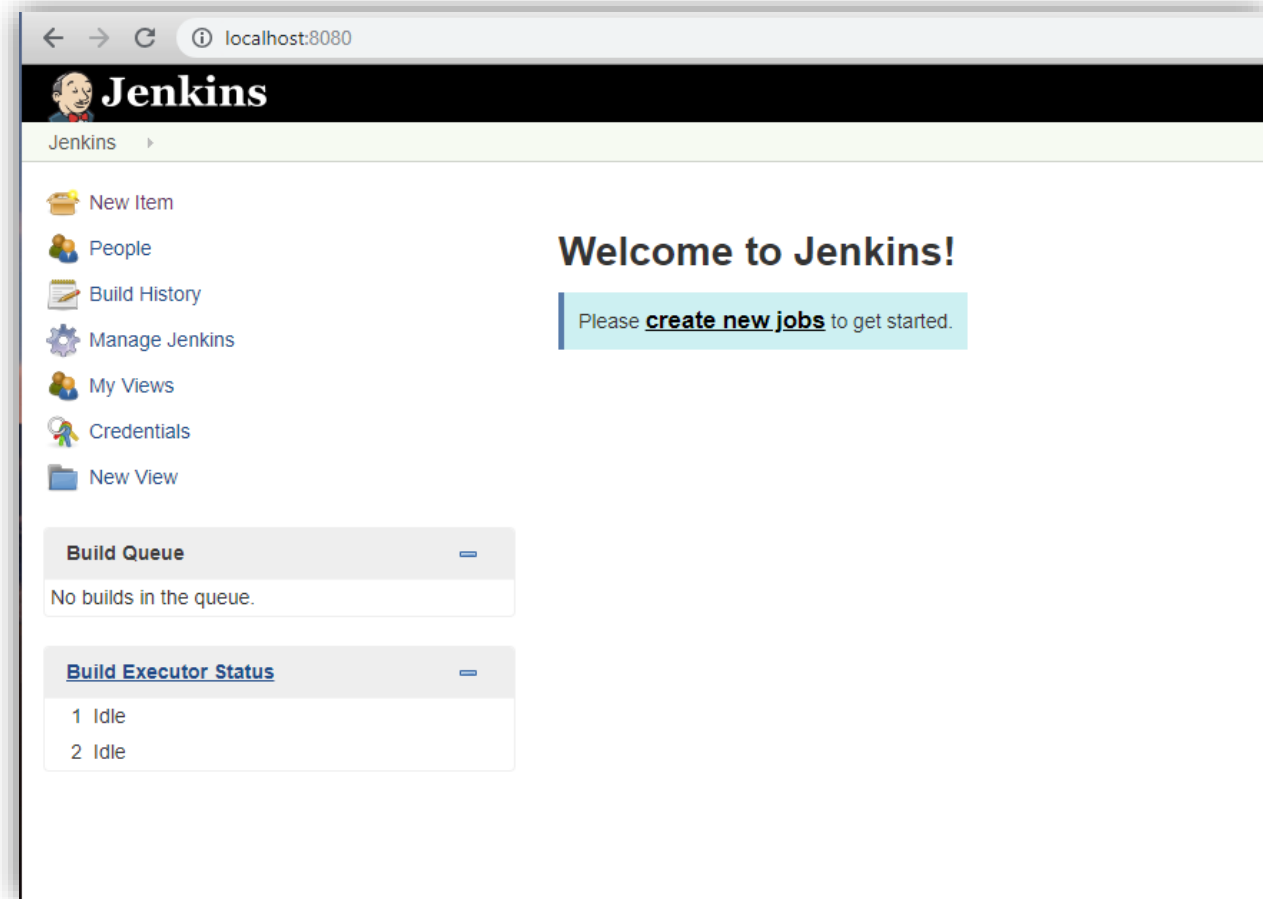
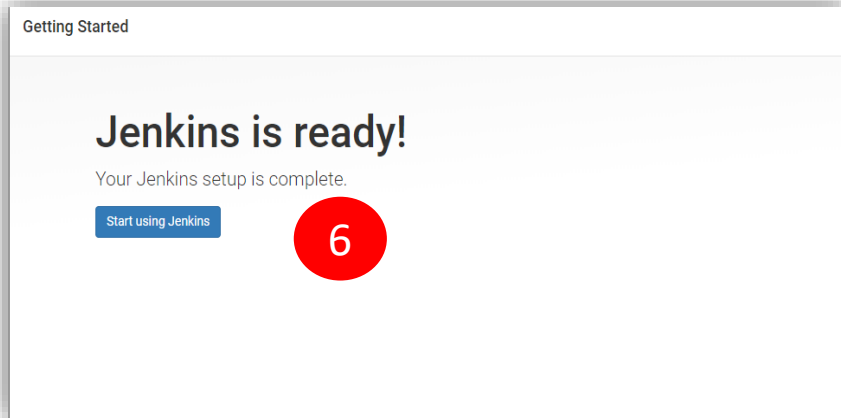
5

Jenkins 2.138.1

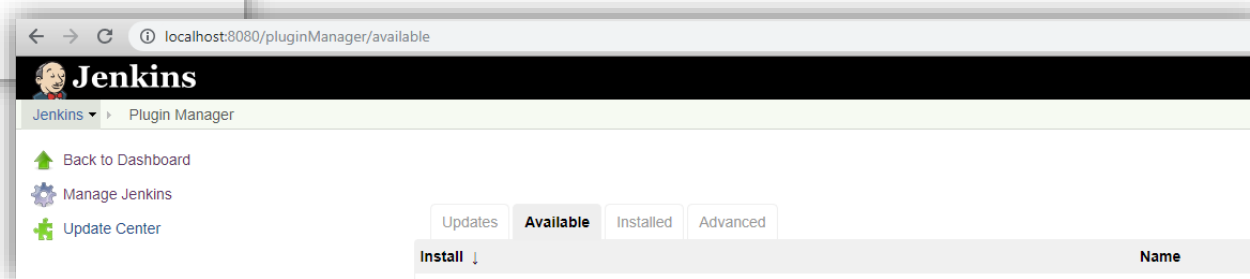
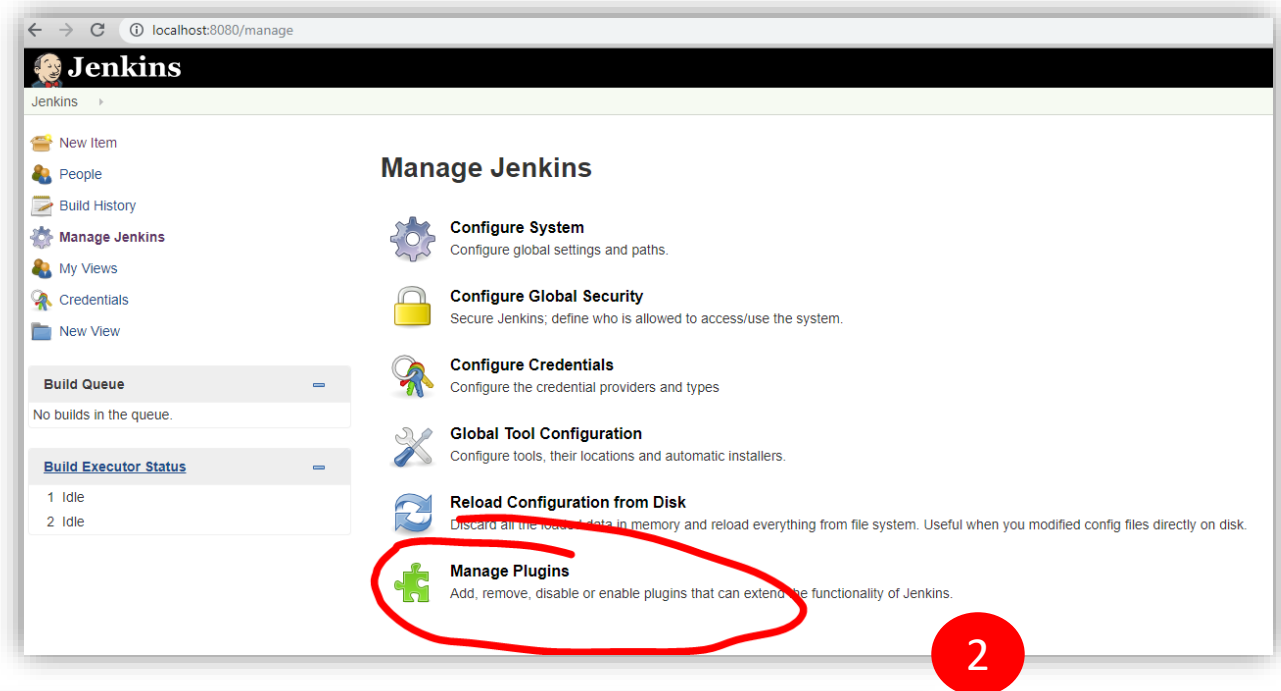
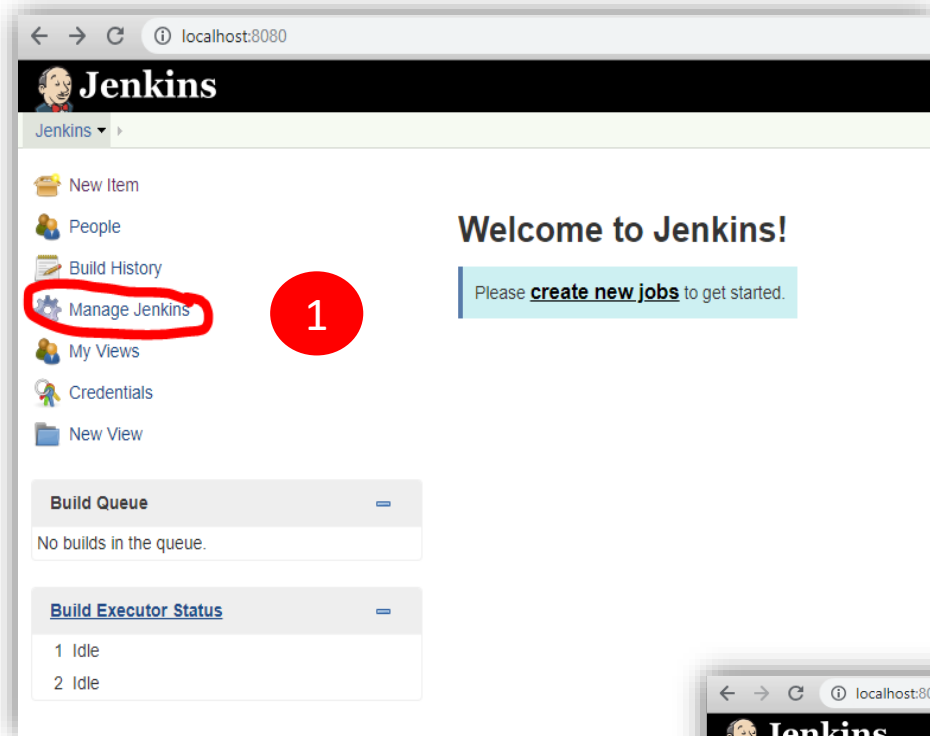
Not now

Save and Finish

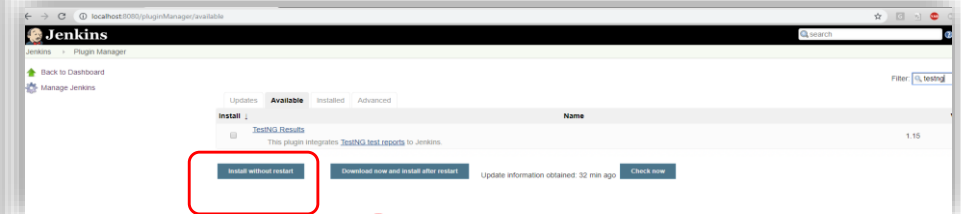
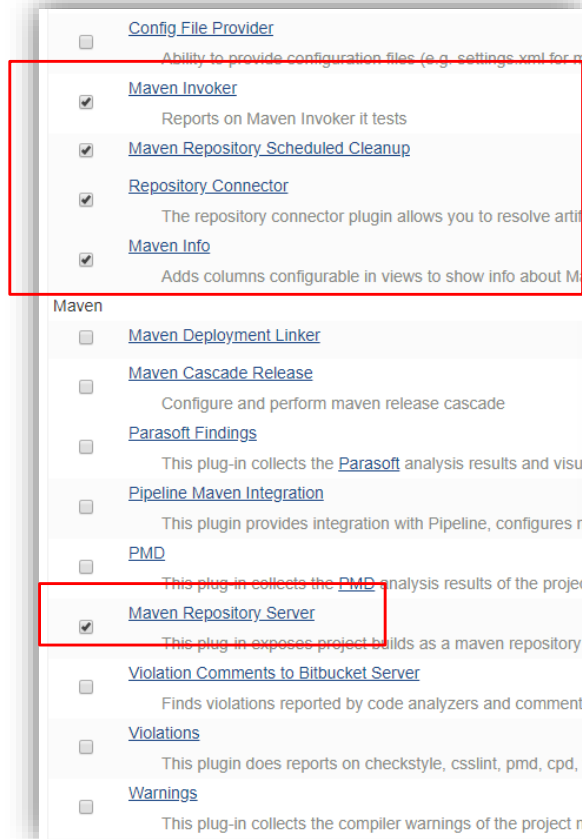
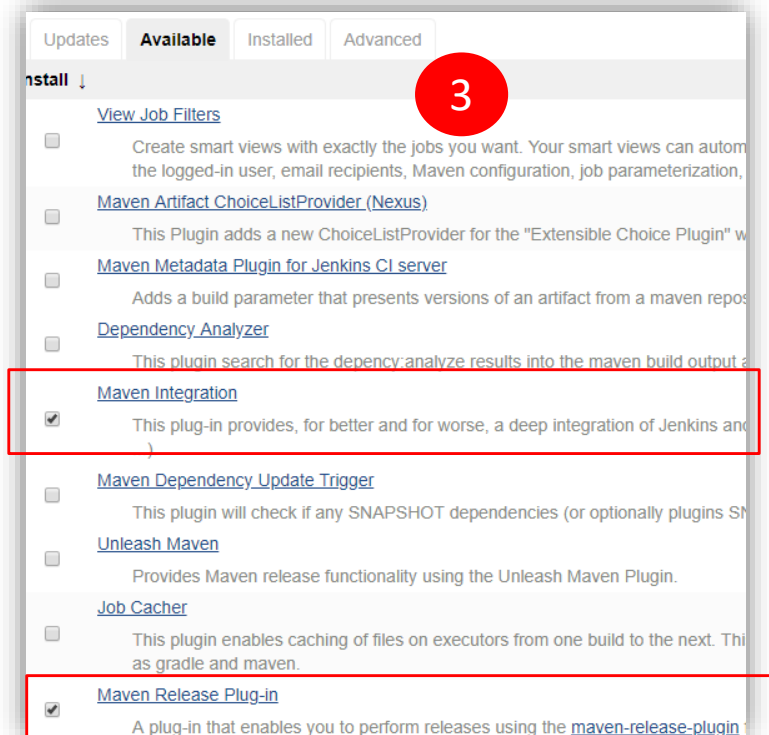
# Jenkins Configuration



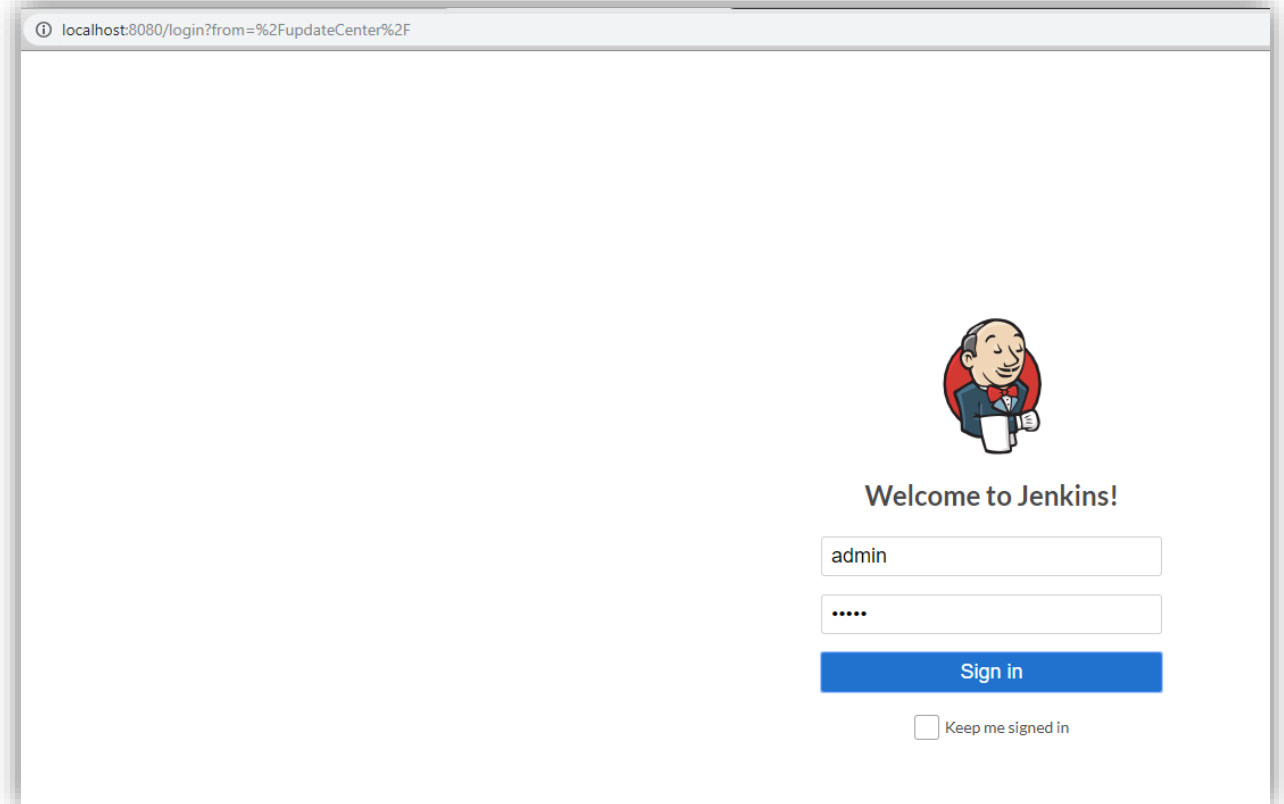
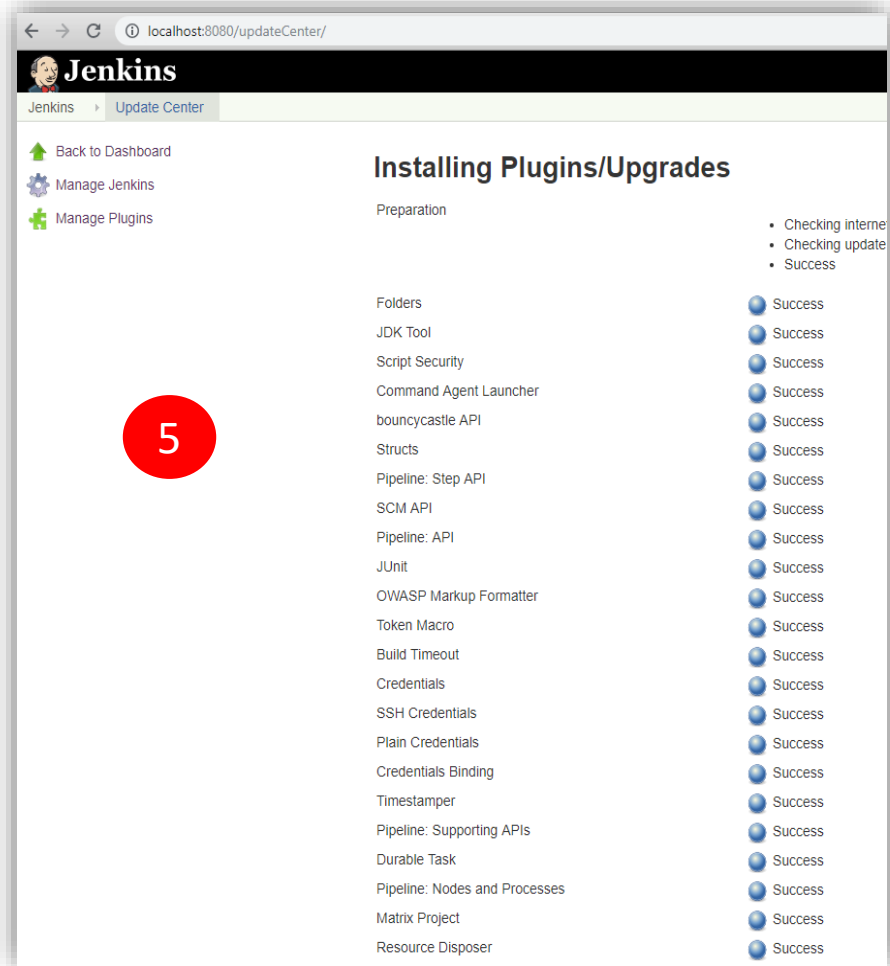
# Install Maven Plugins in Jenkins



# Maven Plugins for Jenkins



# Maven Plugins for Jenkins

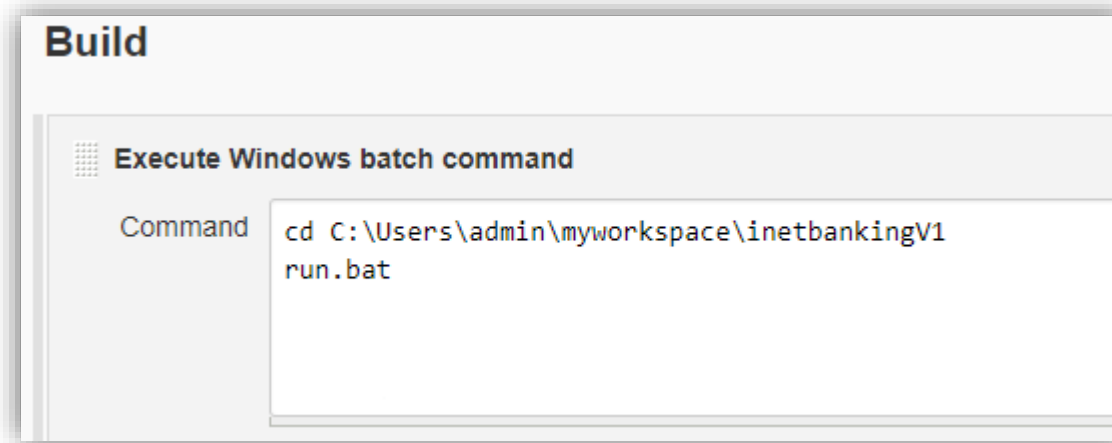




# Run Automation through Jenkins

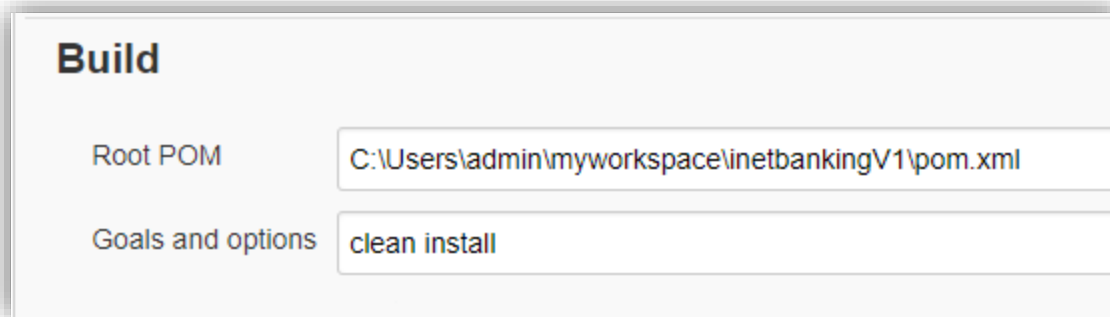
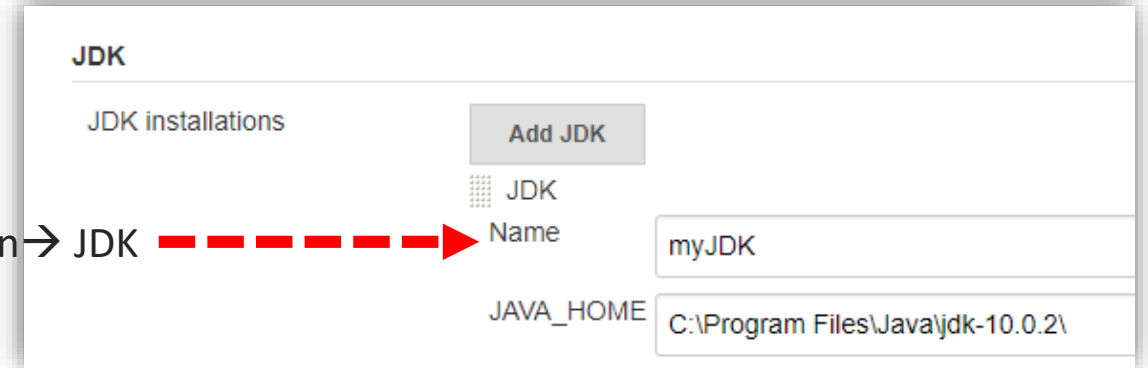
# Run Automation using Jenkins Free Style Project

- Jenkins URL: <http://localhost:8080/>
- **Steps:**
- New item → provide name of the project → Free style project → OK.
- Go to build section → Execute windows batch command → Specify path of run.bat file → Save.
- Go to Dash board → you can see new item(project) is created.



# Run Automation using Jenkins Maven Project

- Jenkins URL: <http://localhost:8080/>
- **Pre-requisite:** JAVA\_HOME Configuration in Jenkins
  - Dashboard → Manage Jenkins → Global Tool Configuration → JDK
- **Steps:**
  - New item → provide name of the project → Maven project → OK.
  - Go to build section → Specify path of pom.xml and Goals as shown in picture
  - Go to Dash board → you can see new item(project) is created.

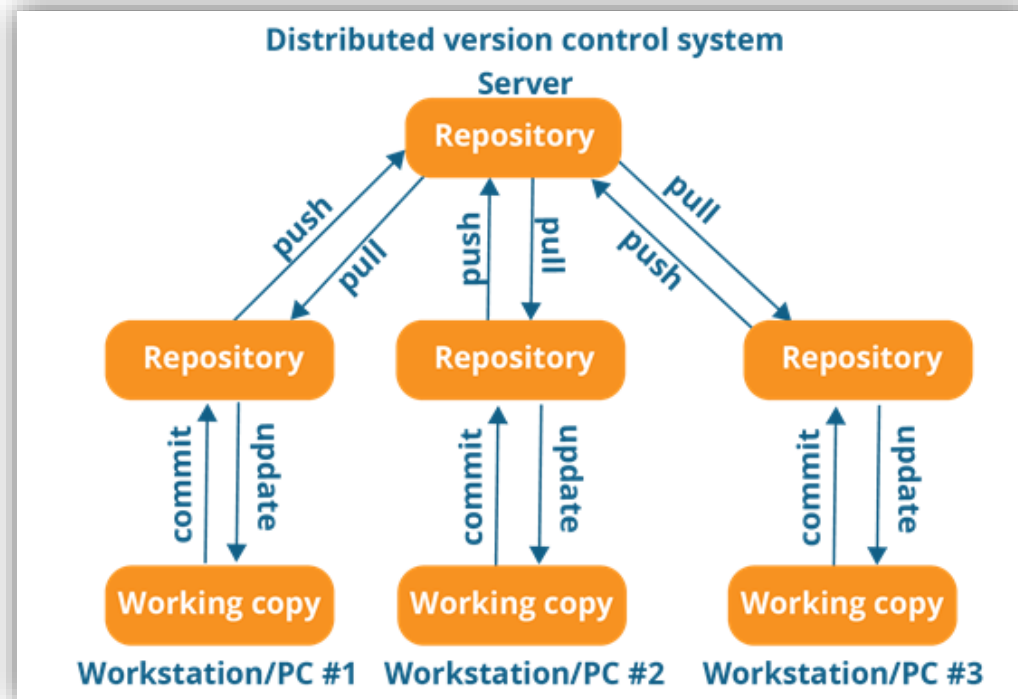
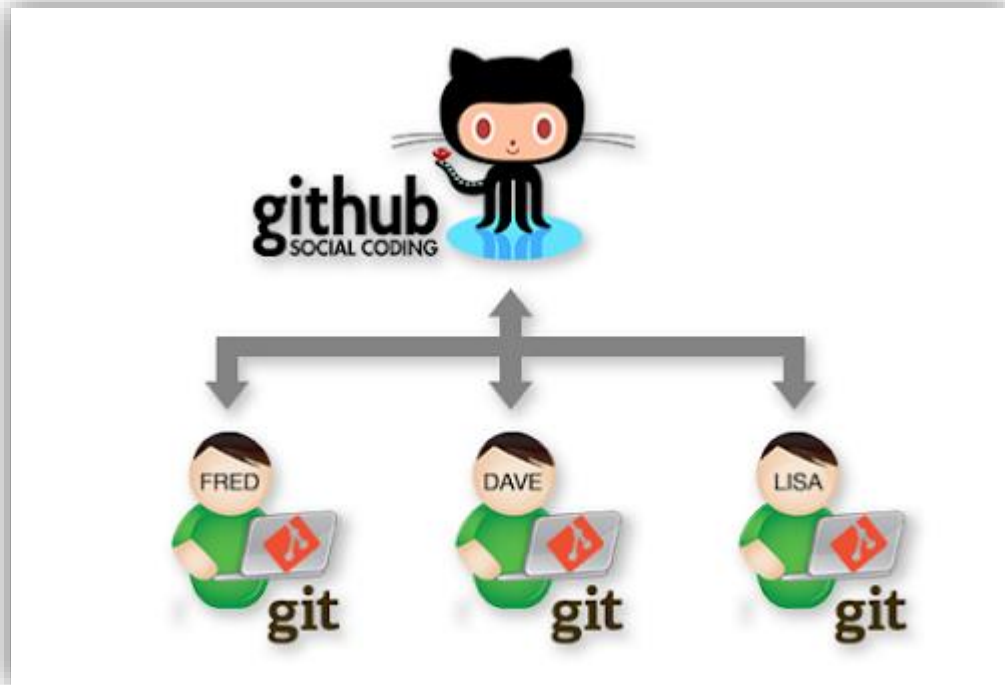




# Git & GitHub

# Git & GitHub

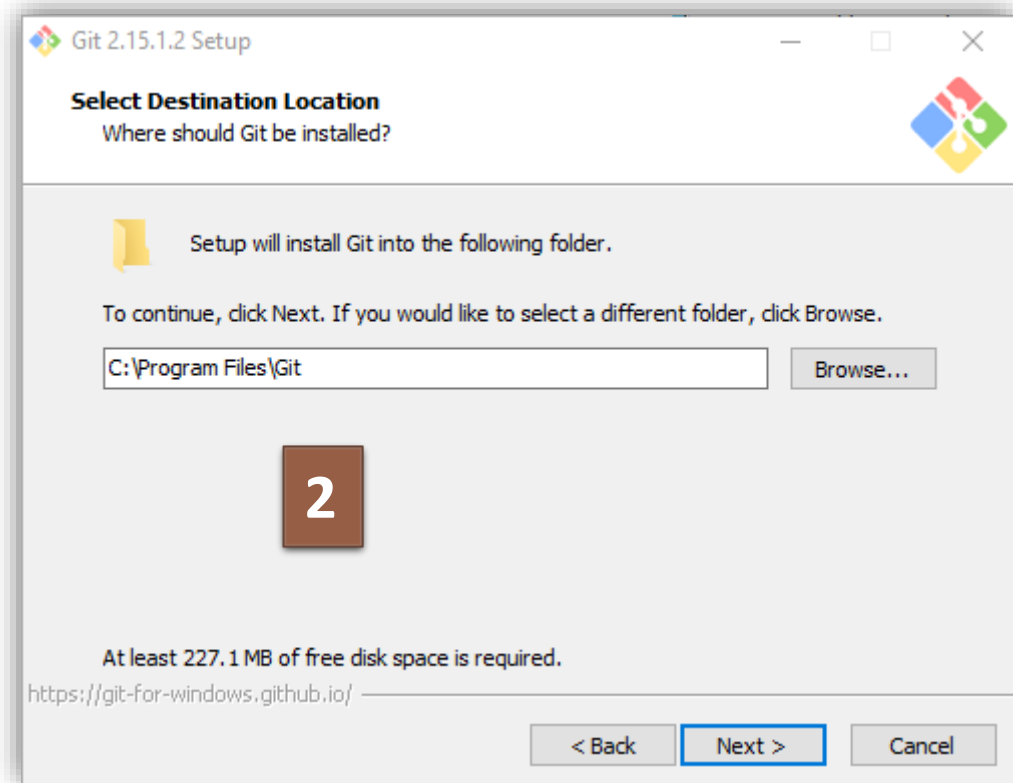
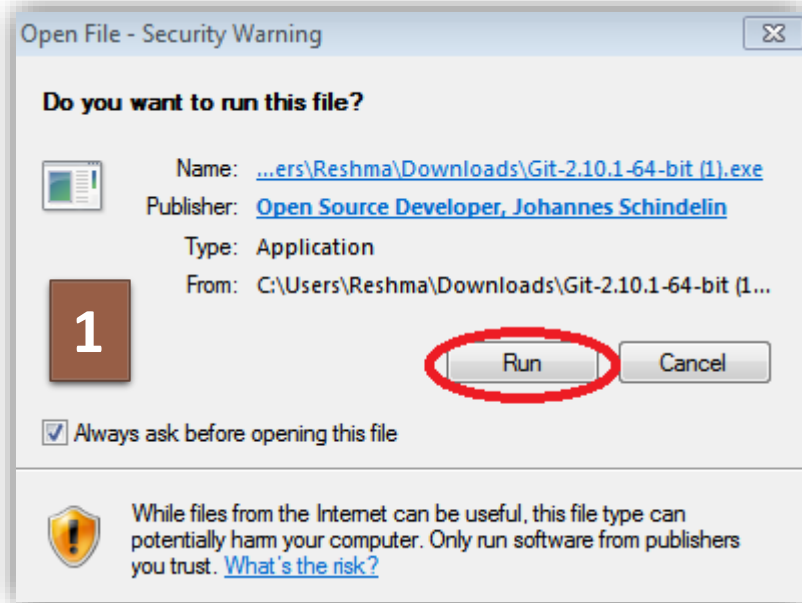
- **Git** is a revision control system used to track changes in computer files. It's a tool to manage your code & file history while coordinating work remotely on those files with others. **GitHub** is a hosting service for **git** repositories. **Git** is the tool, while **GitHub** is the service to use **git**.



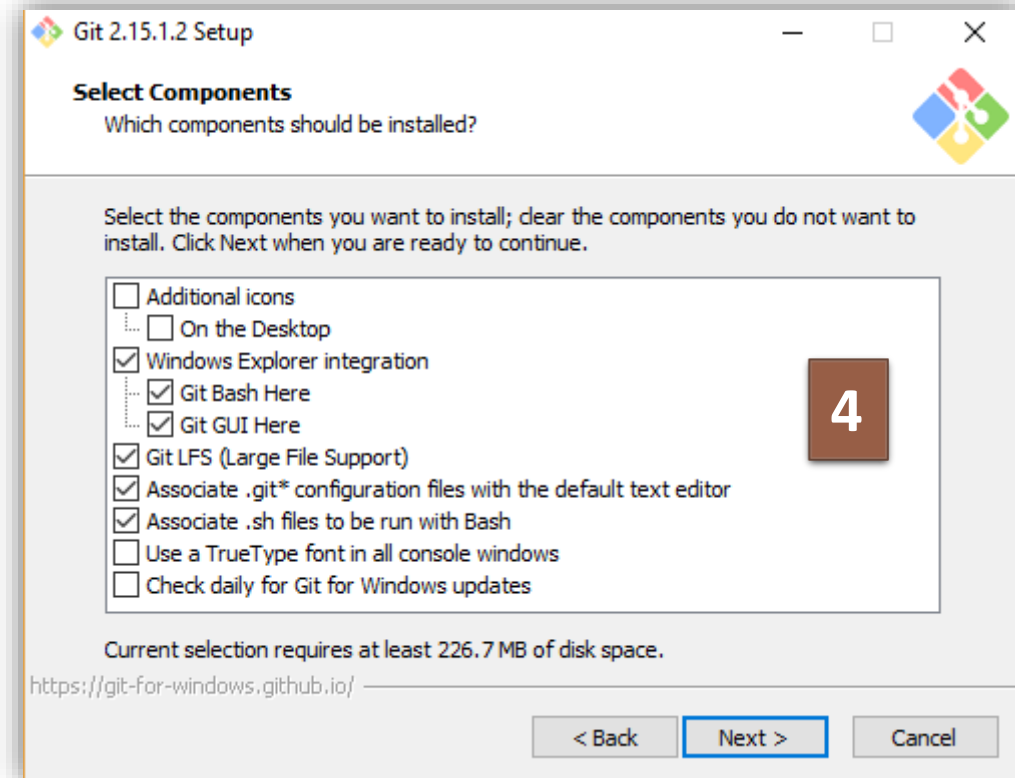
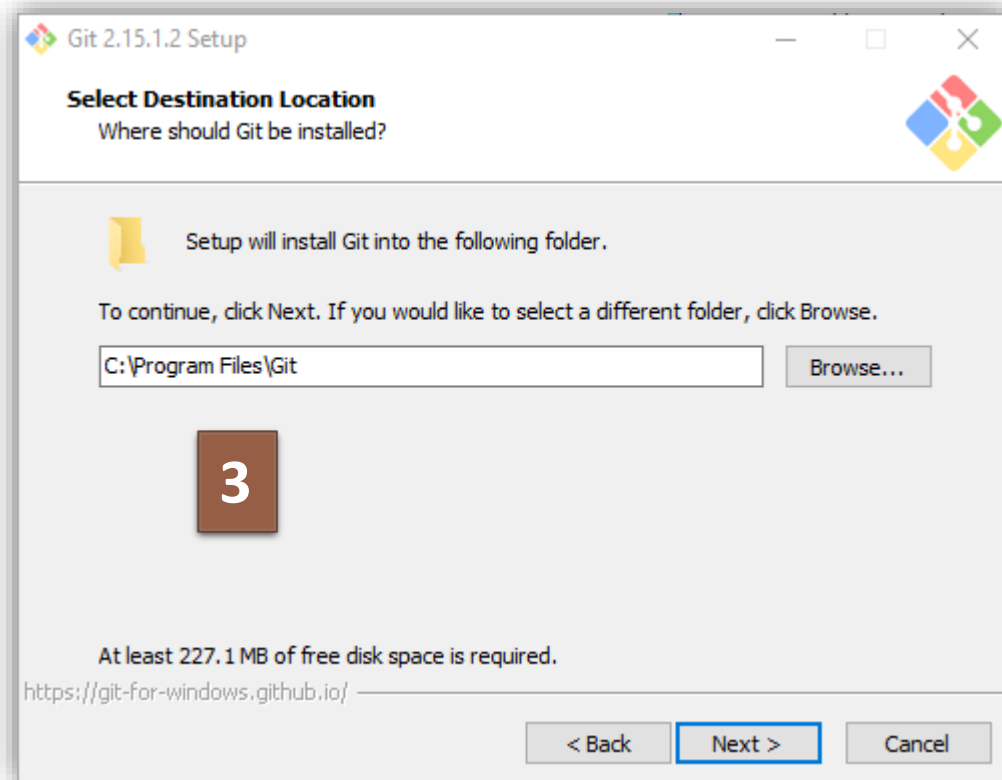
# GIT Installation on Windows

- To download the latest version of Git, click on the link below:
- <https://git-scm.com/download/win/>

# GIT Installation on Windows

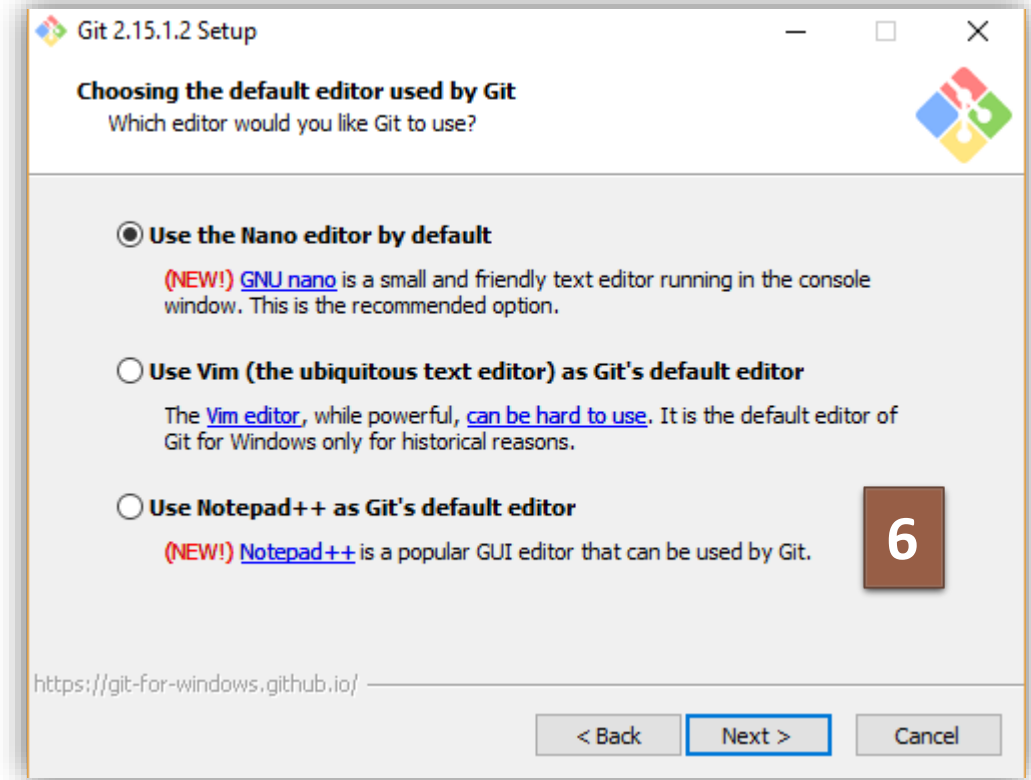
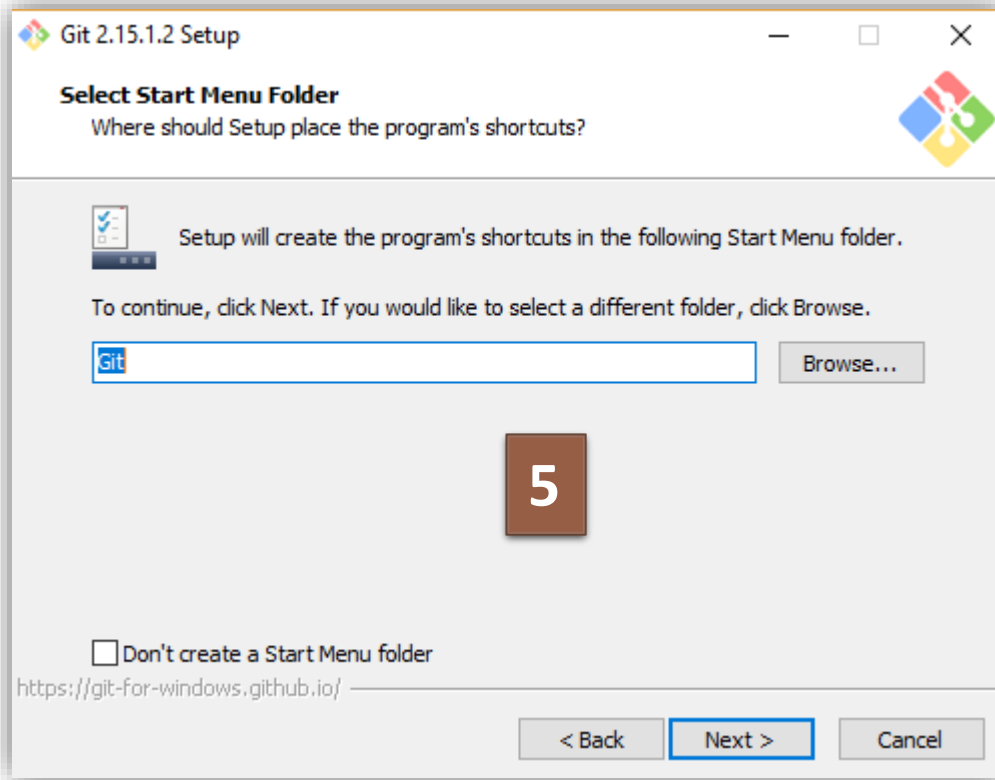


# GIT Installation on Windows

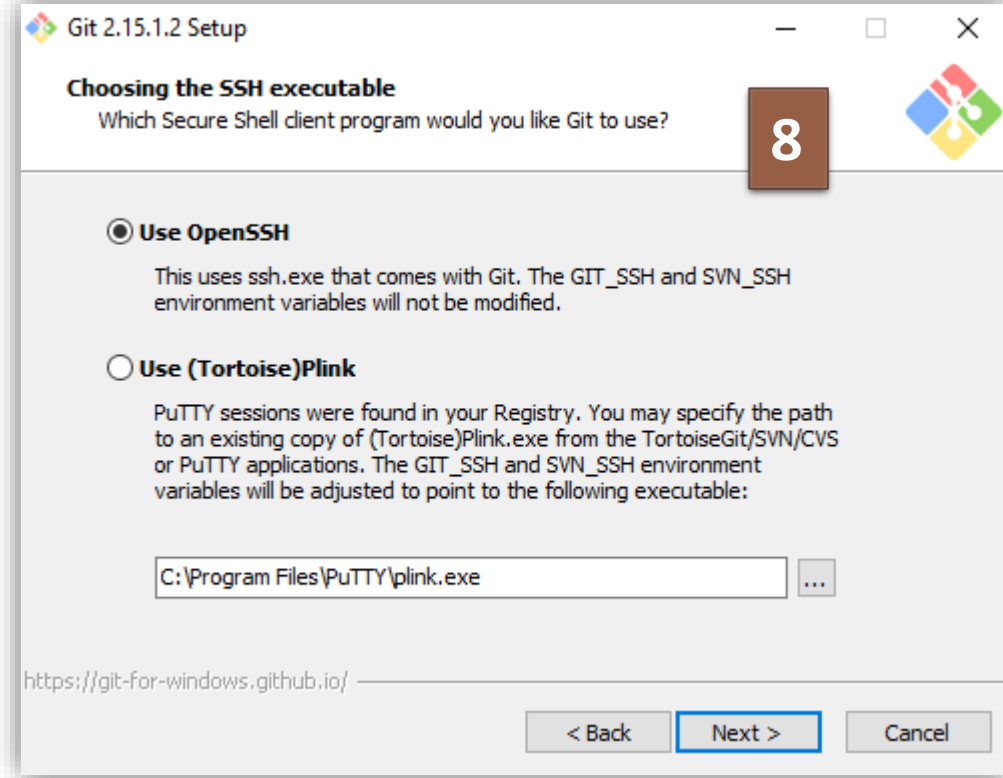
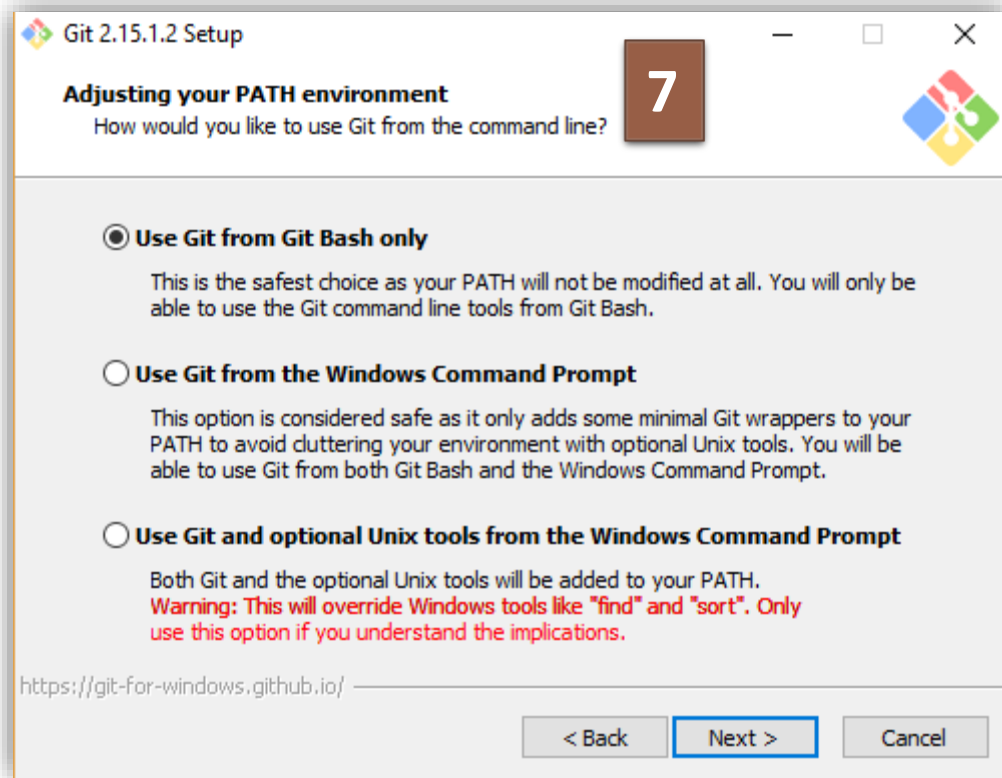




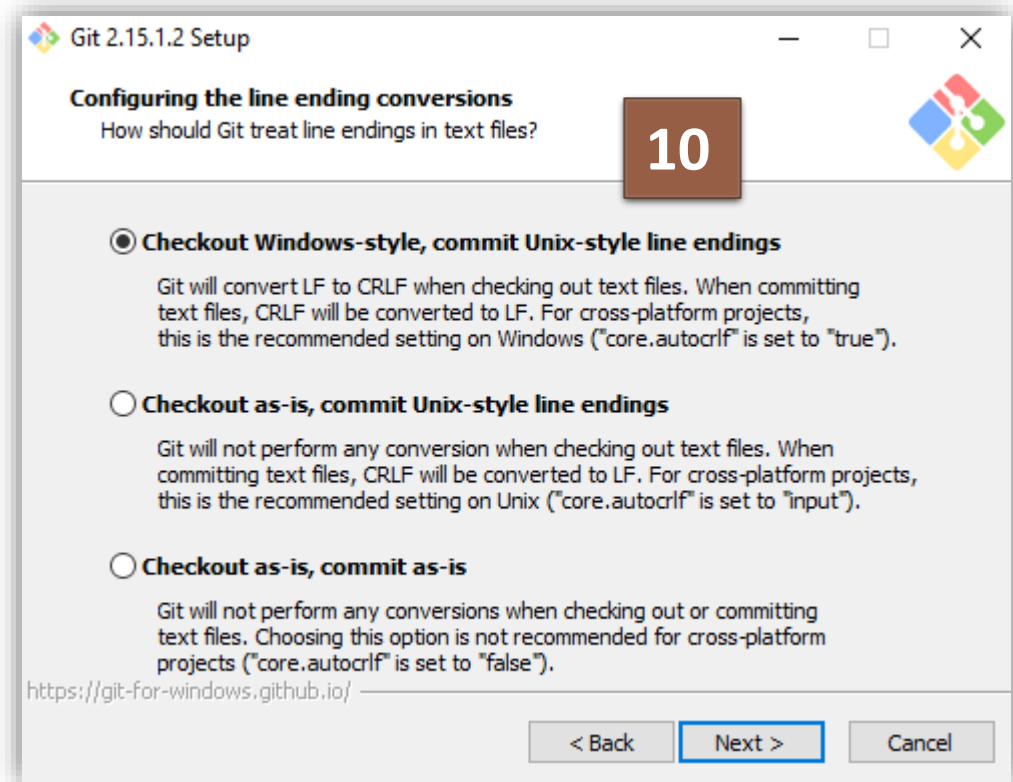
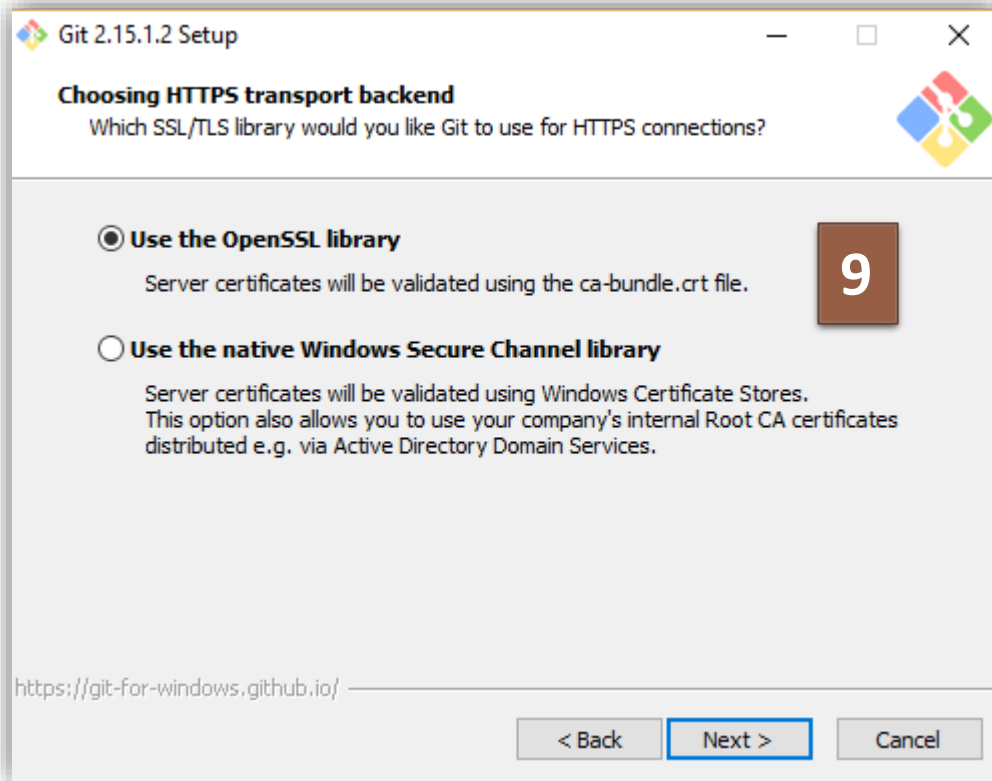
# GIT Installation on Windows



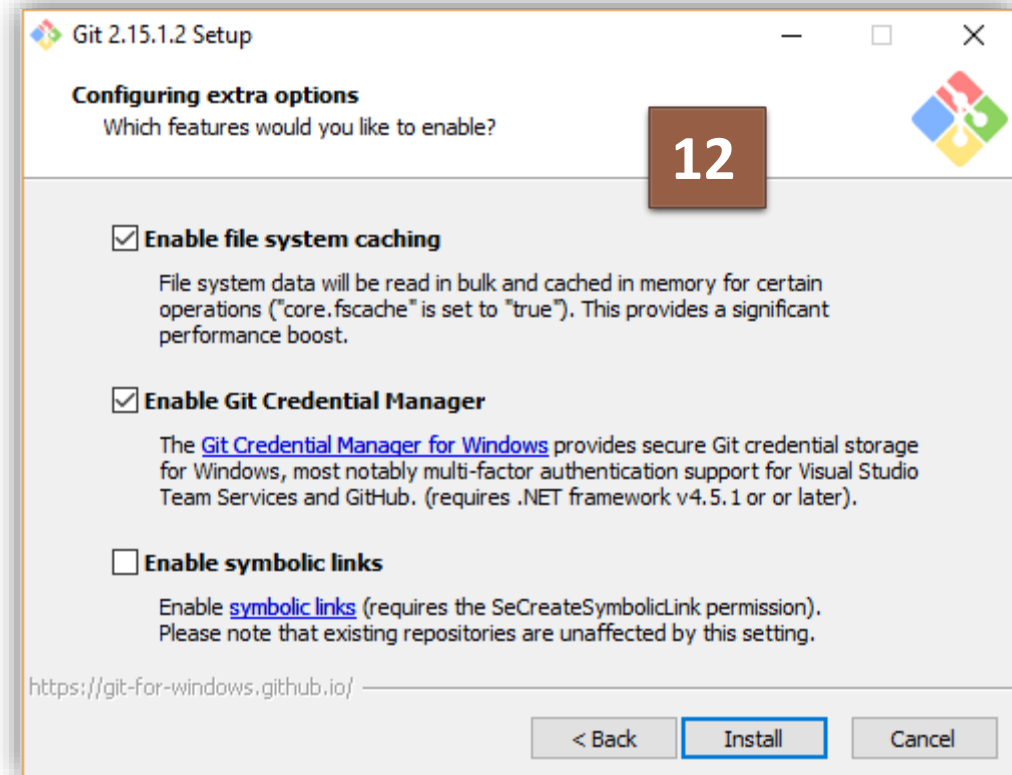
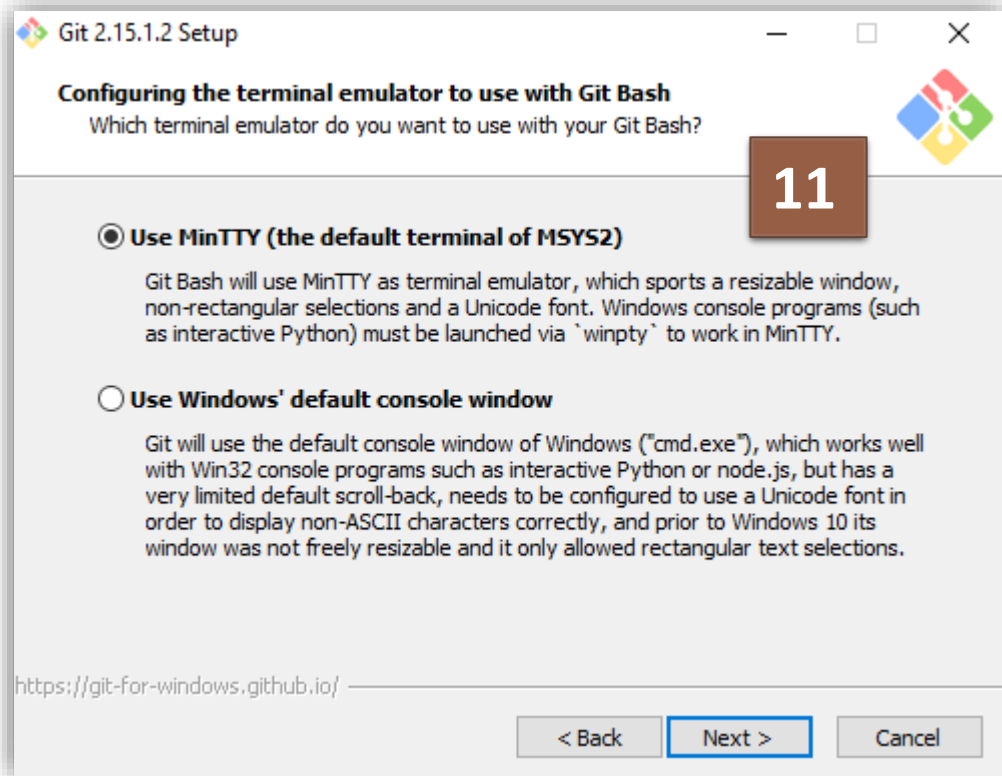
# GIT Installation on Windows



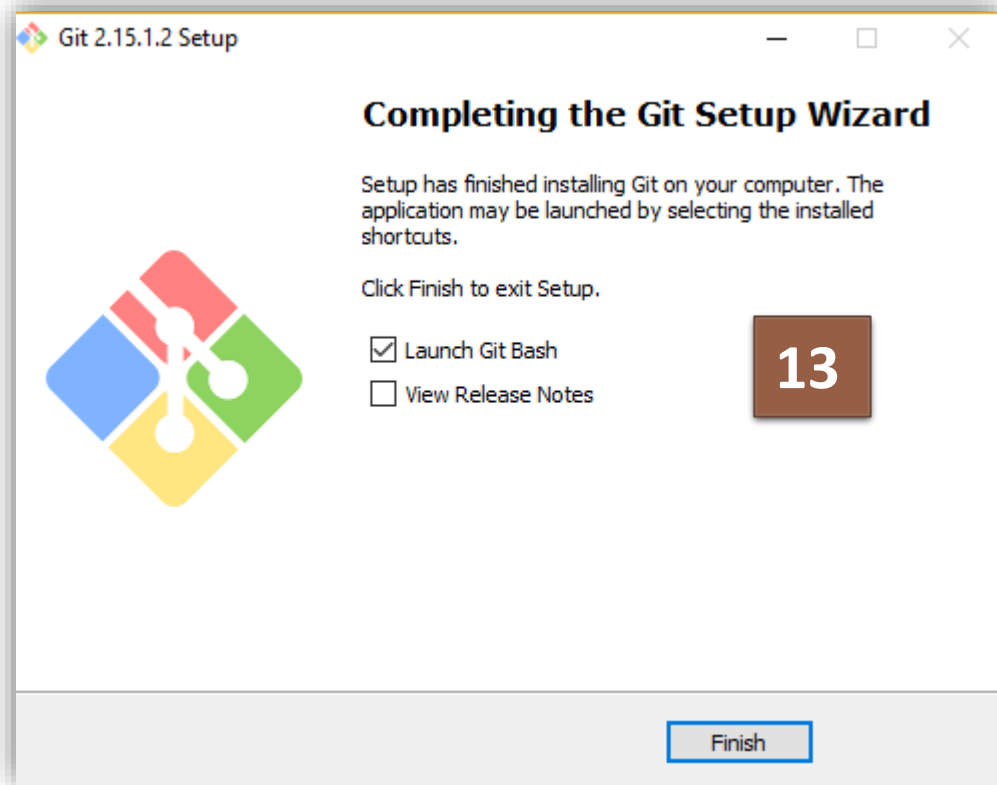
# GIT Installation on Windows



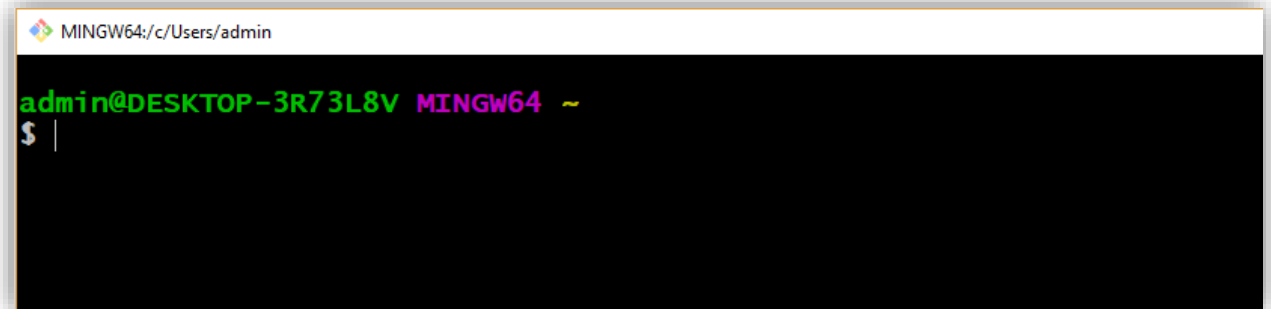
# GIT Installation on Windows



# GIT Installation on Windows



This will launch Git Bash on your screen which looks like the snapshot below:



# GITHUB Account

Owner: pavanoltraining / Repository name: git-githubdemo ✓

Great repository names are short and memorable. Need inspiration? How about [literate-tribble](#).

Description (optional)

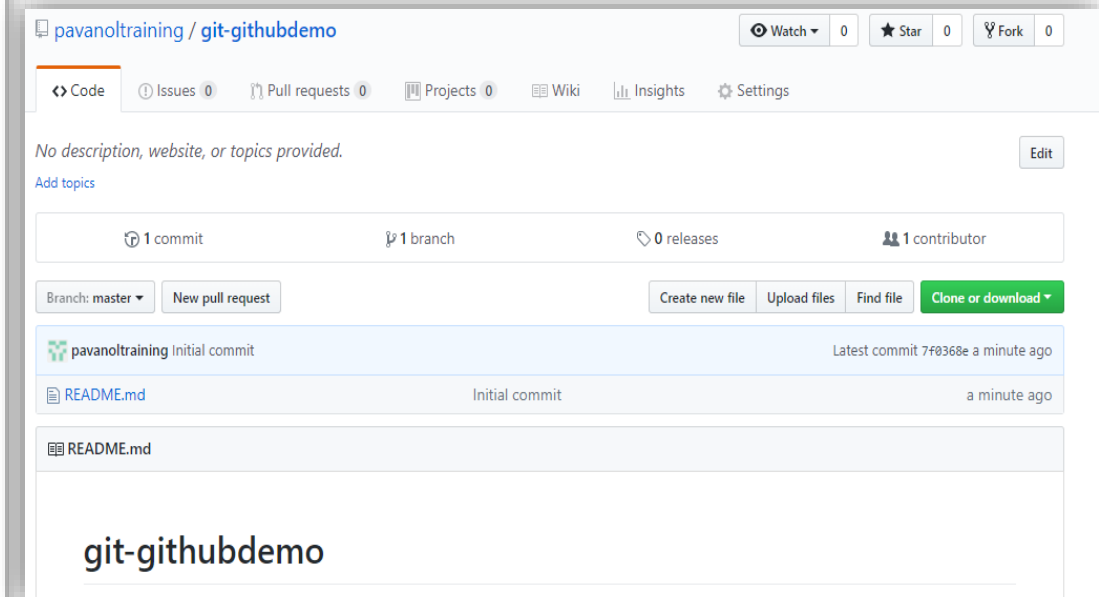
☒ **Public**  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

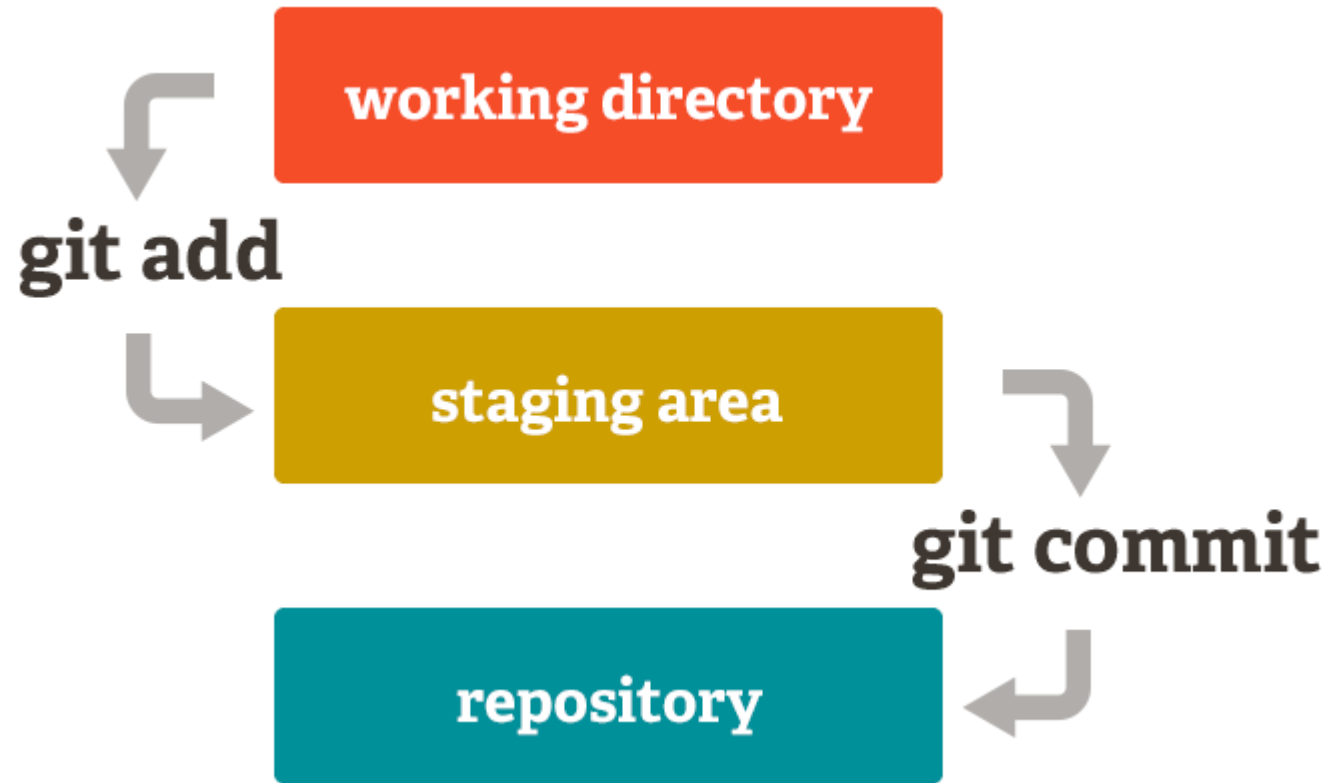
Add .gitignore: None | Add a license: None ⓘ

[Create repository](#)



# Workflow







# Git commands

Command	Usage
\$ git init	This command is used to start a new repository
\$ git remote add origin "URL of git hub repository"	This command is used to add a "remote" repository URL <url> which is referred in other git commands (such as pull or push) with the provided name
\$ git status	This command lists all the files that have to be committed.
\$ git add -A	This command add all the files to the staging area.
\$ git config --global user.name "Your Name" \$ git config --global user.email "Your email ID"	This command sets the author name and email address respectively to be used with your commits.
\$ git commit -m "This is my first commit!"	This command commits any files you've added with the git add command and also commits any files you've changed since then.
\$ git push -u origin master	This command sends the committed changes of master branch to your remote repository.

# Pulling files from Github to git repository

- `$ git pull origin master`

# Step by Step Execution

- **Setup**

- \$ git init

- \$ git remote add origin "https://github.com/pavanoltraining/inetbankingV1.git"

- \$ git status

- \$ git add -A

- \$ git config --global user.name "pavan"

- \$ git config --global user.email "pavanoltraining@gmail.com"

- \$ git commit -m "This is my first commit!"

- **Round2:**

- \$ git status

- \$ git add -A

- \$ git commit -m "This is my first commit!"

- **Pushing the files from git to Git Hub**

- \$ git push -u origin master

- **Pulling files from Github to git repository**

- \$ git pull origin master

# Run Github project from Jenkins

**Source Code Management**

☐ None  
☒ Git

**Repositories**

Repository URL

Credentials  [Add](#)

[Advanced...](#)

[Add Repository](#)

**Branches to build**

Branch Specifier (blank for 'any')

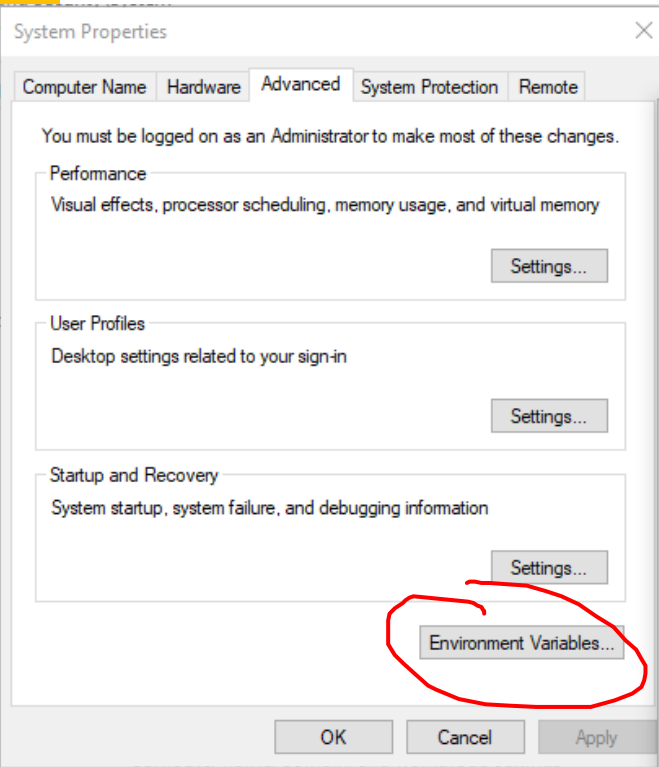
[Add Branch](#)

**Repository browser**

**Build**

Root POM

Goals and options



Computer name: DESKTOP-3R73L8V  
Full computer name: DESKTOP-3R73L8V  
Computer description:  
Workgroup: WORKGROUP

Windows activation  
Windows is activated [Read the Microsoft Software License Terms](#)  
Product ID: 00330-80000-00000-AA691

