Continuous Integration (CI)

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\ProjectFolder
  - mvn clean install

Run automation through run.bat

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

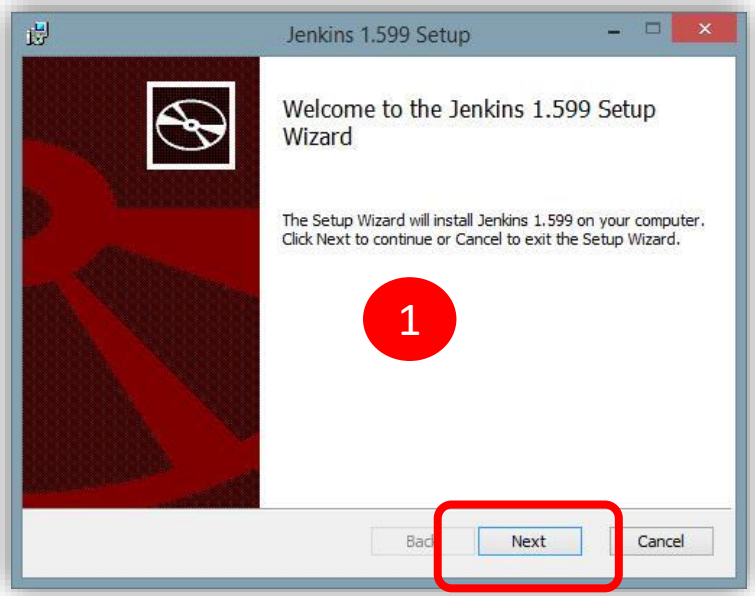
  cd C:\Users\admin\eclipse-workspace\ ProjectFolder

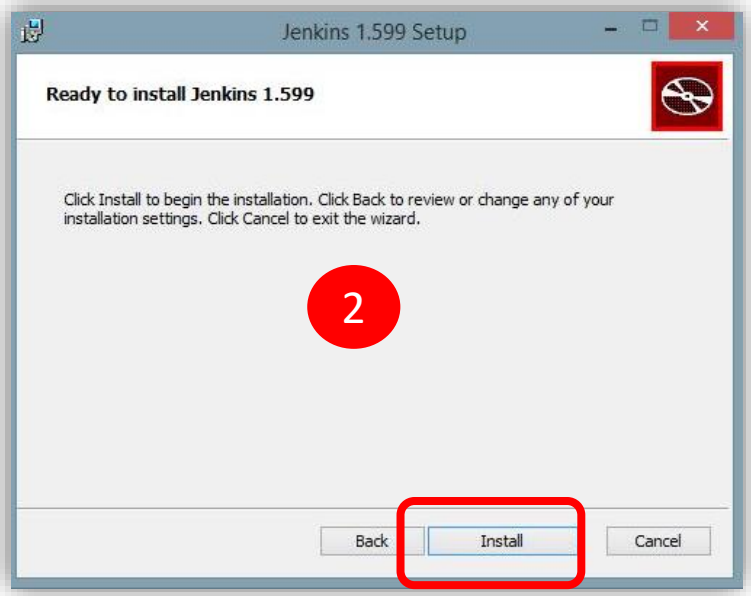
  mvn clean install

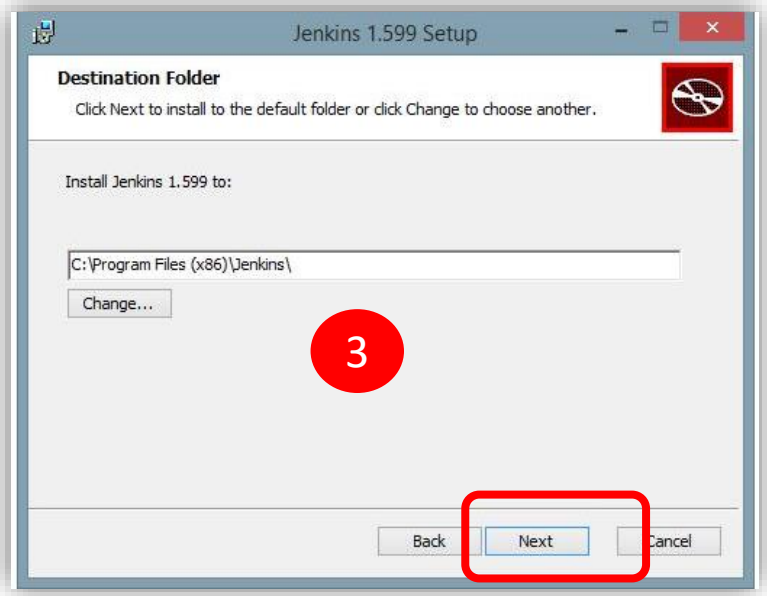
Jenkins installation & Configuration

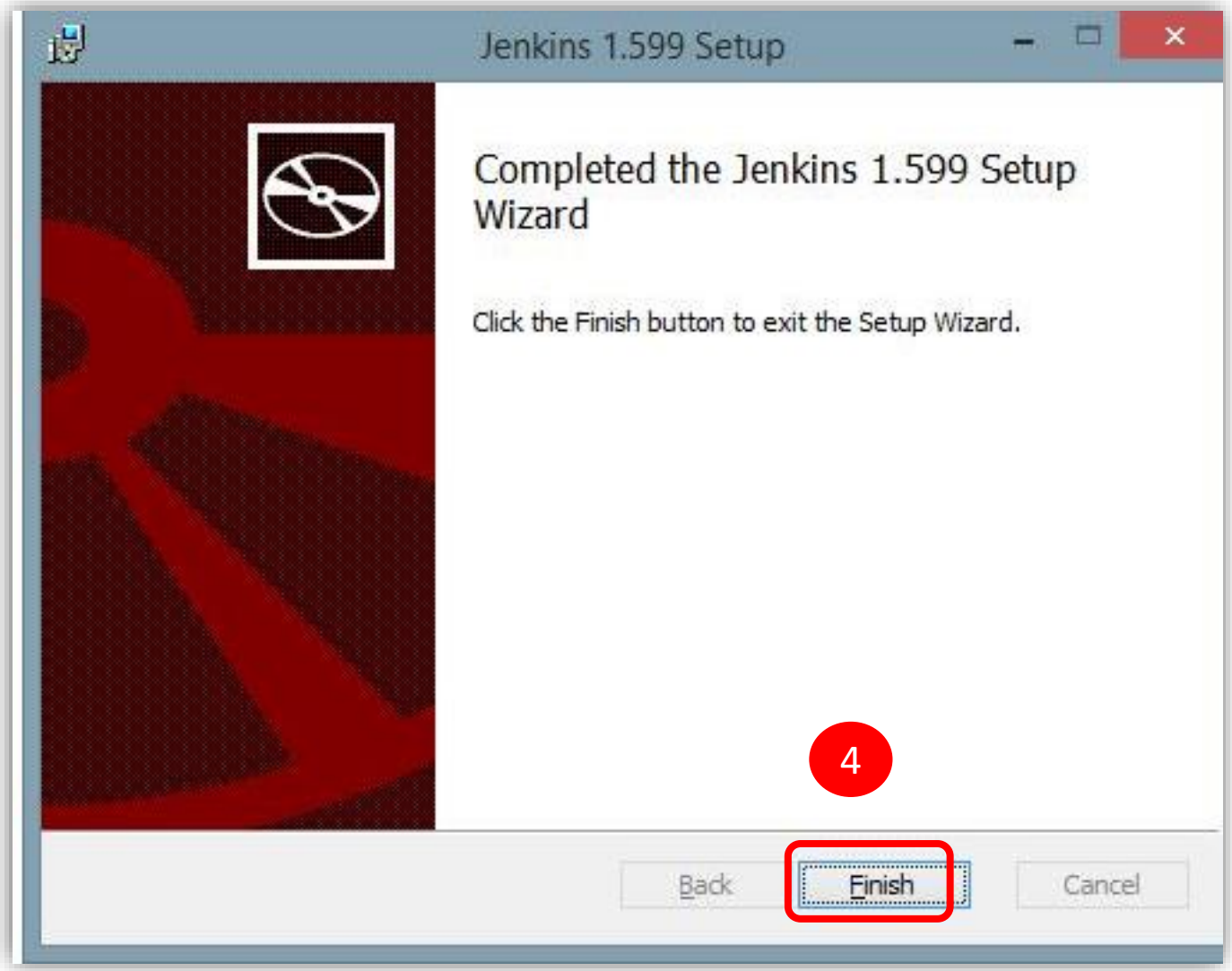
Jenkins Step By step Installation

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



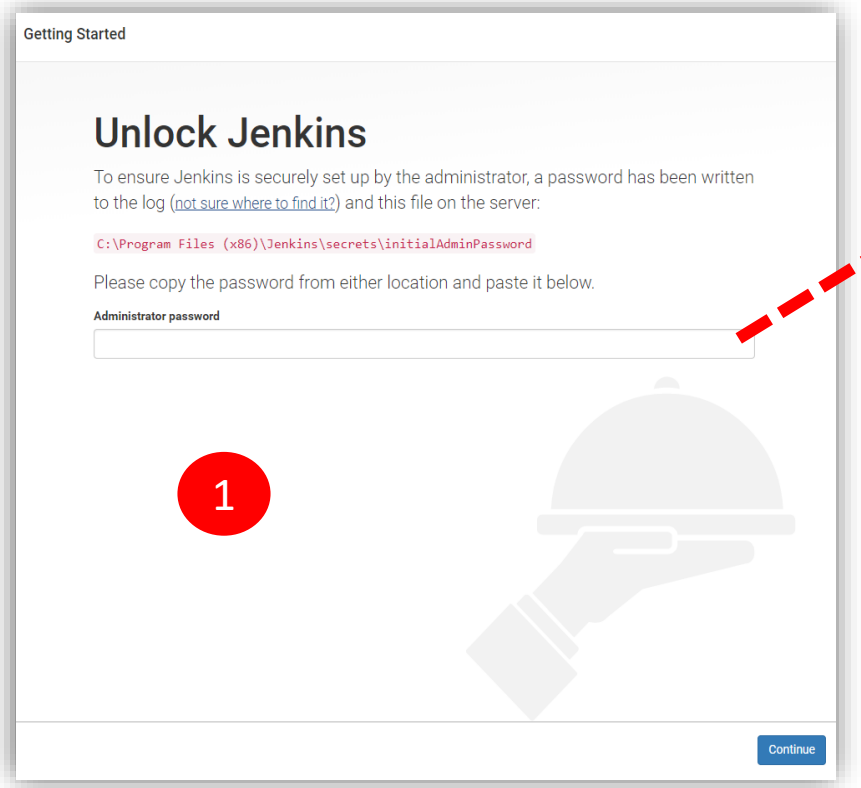


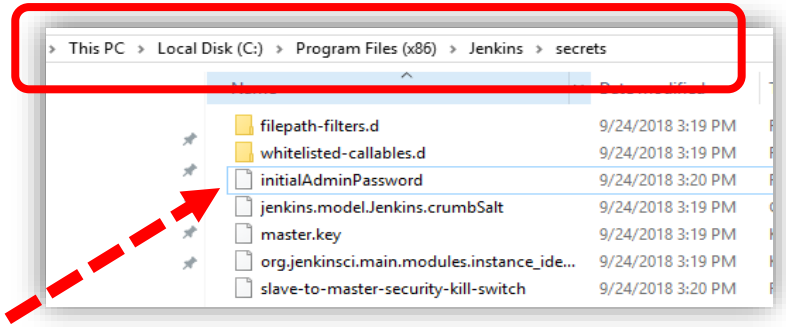


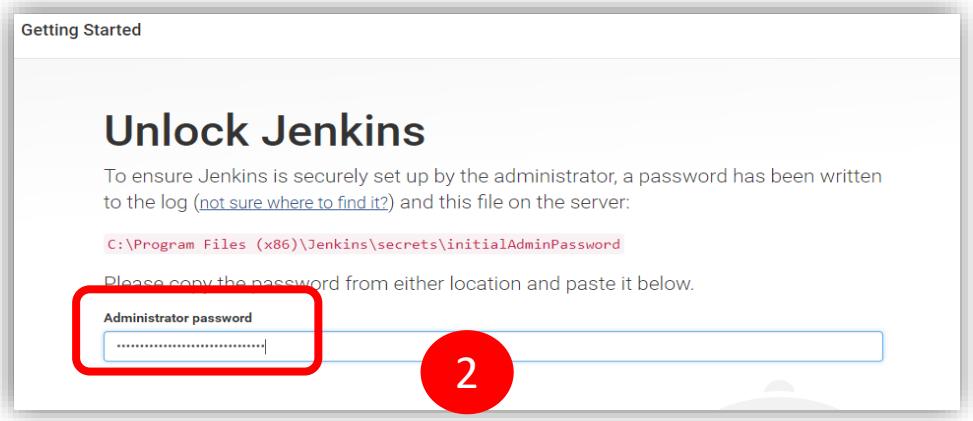

Jenkins 1.599 Setup
Completed the Jenkins 1.599 Setup Wizard
Click the Finish button to exit the Setup Wizard.
4
Back
Finish
Cancel

Jenkins Configuration

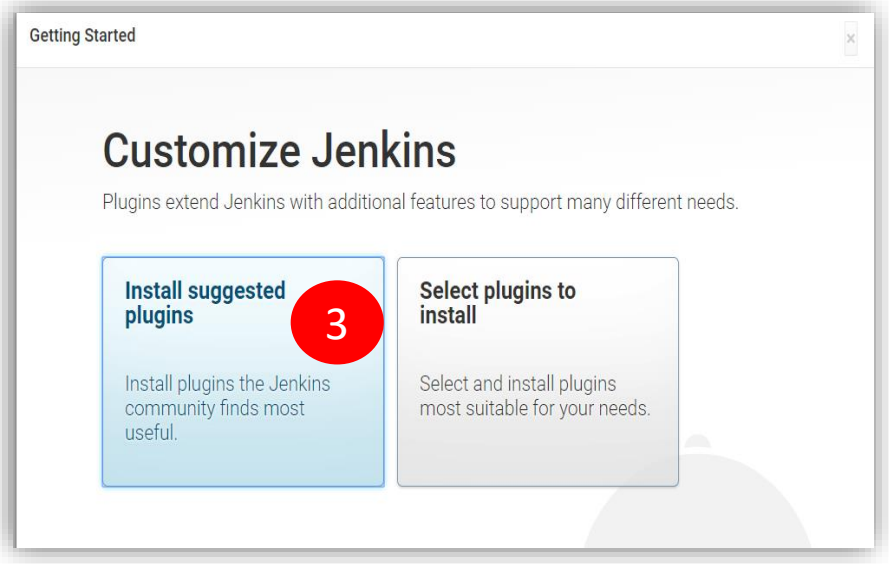
URL to open Jenkins: http://localhost:8080

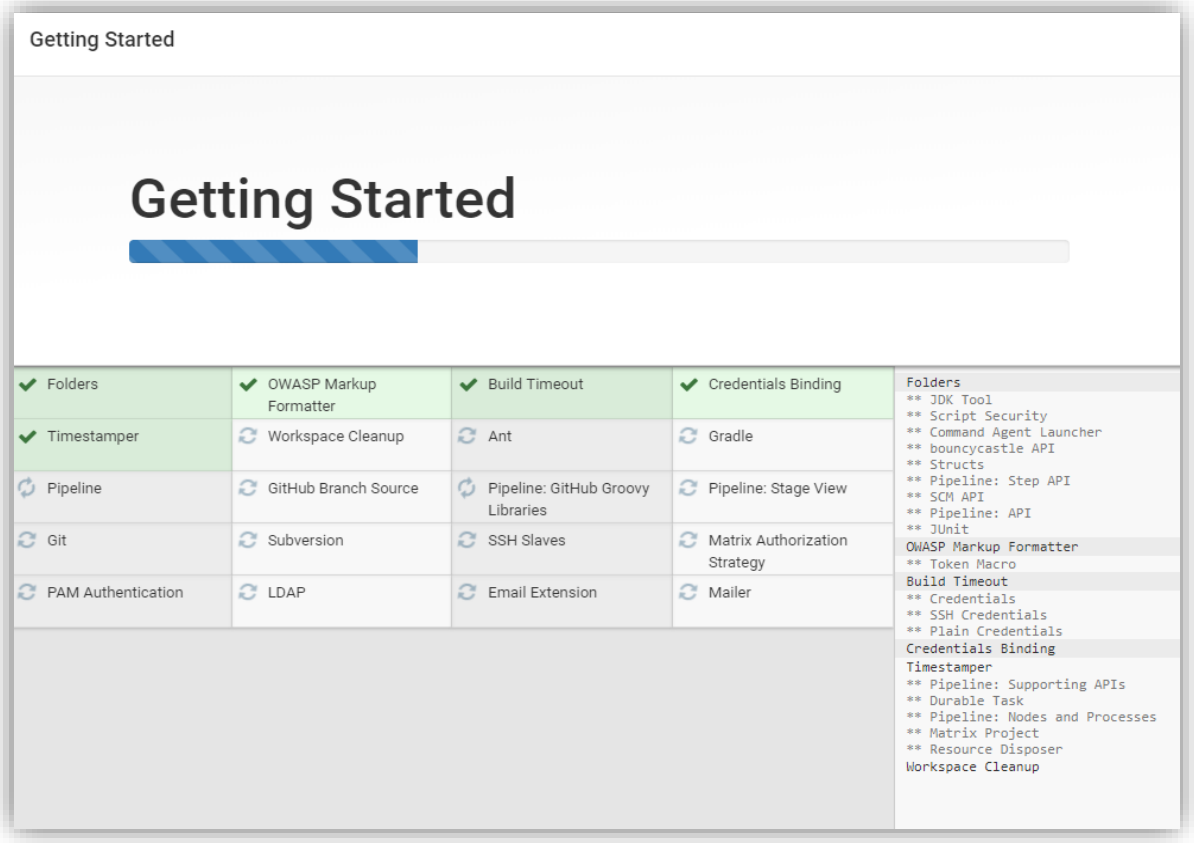




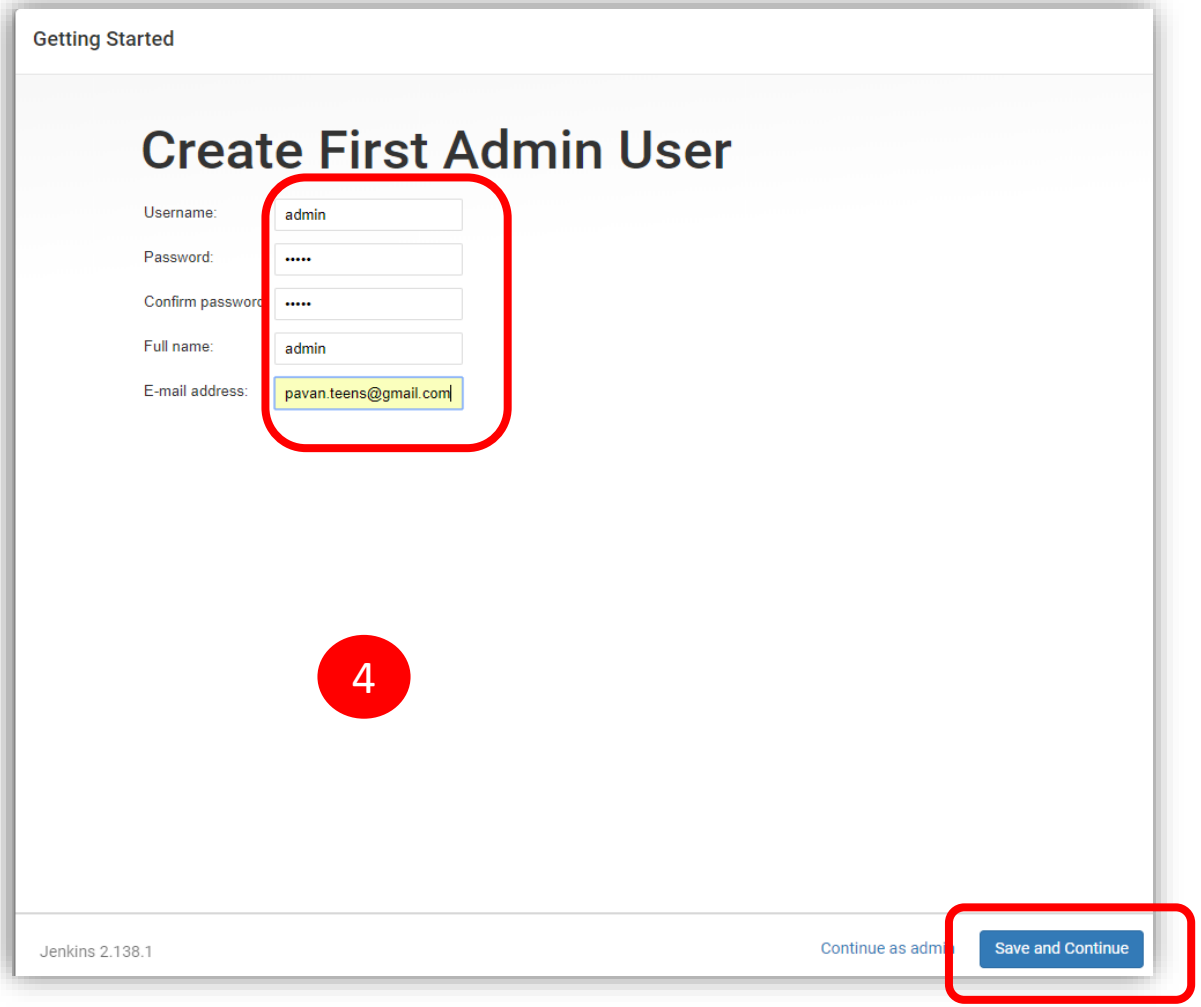


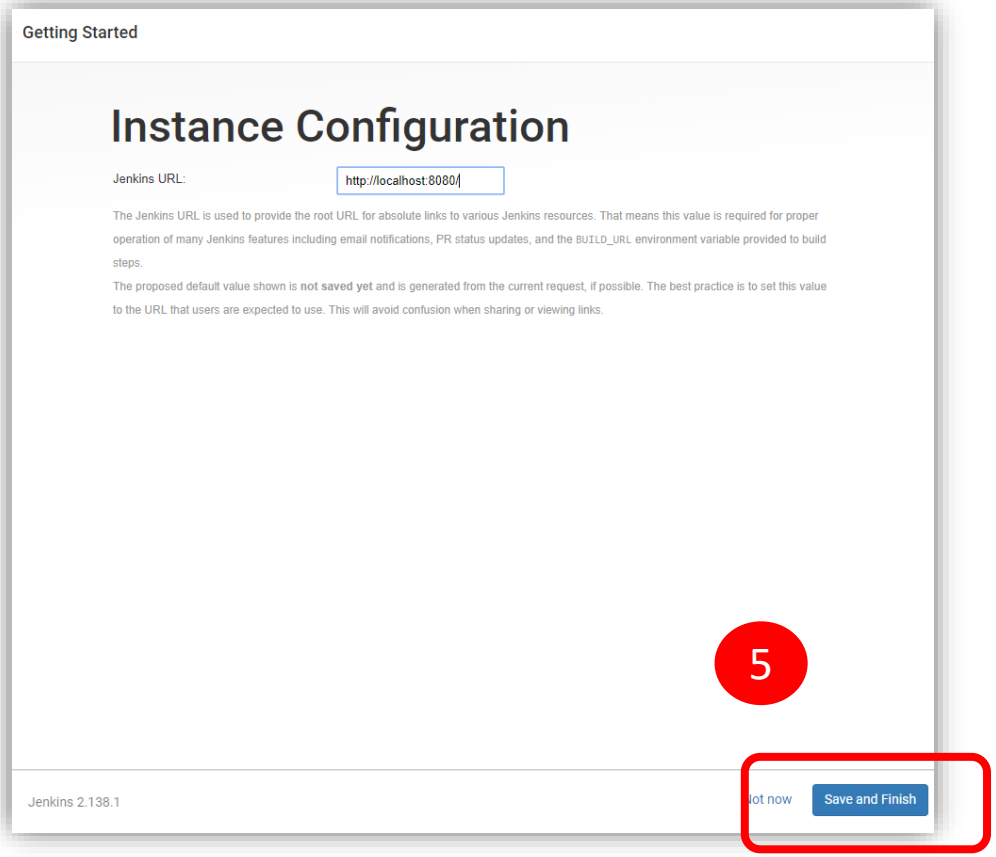
Jenkins Configuration



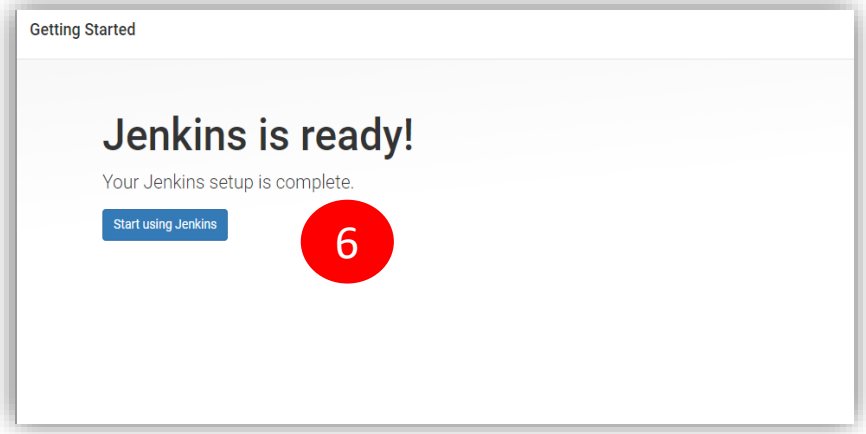


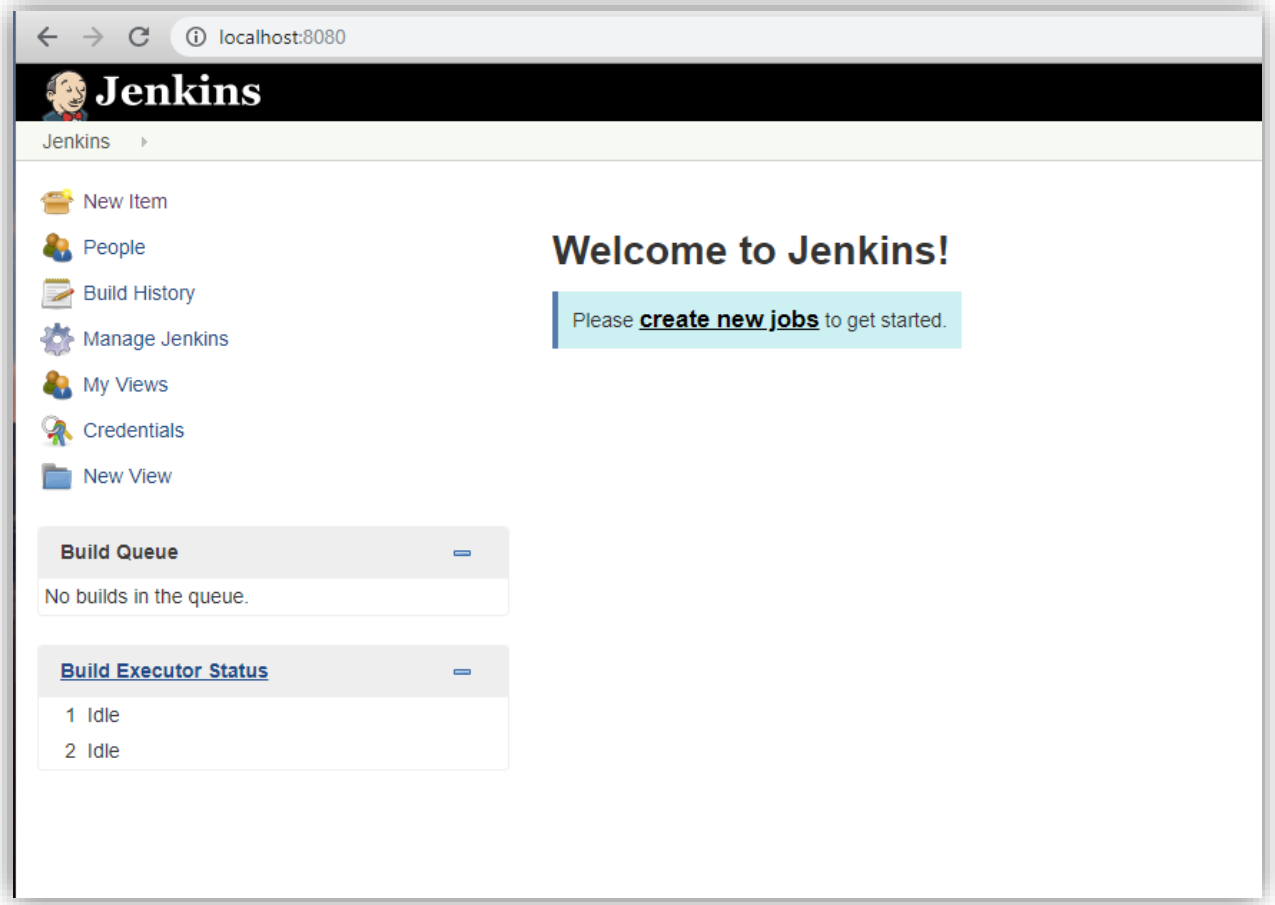
Jenkins Configuration



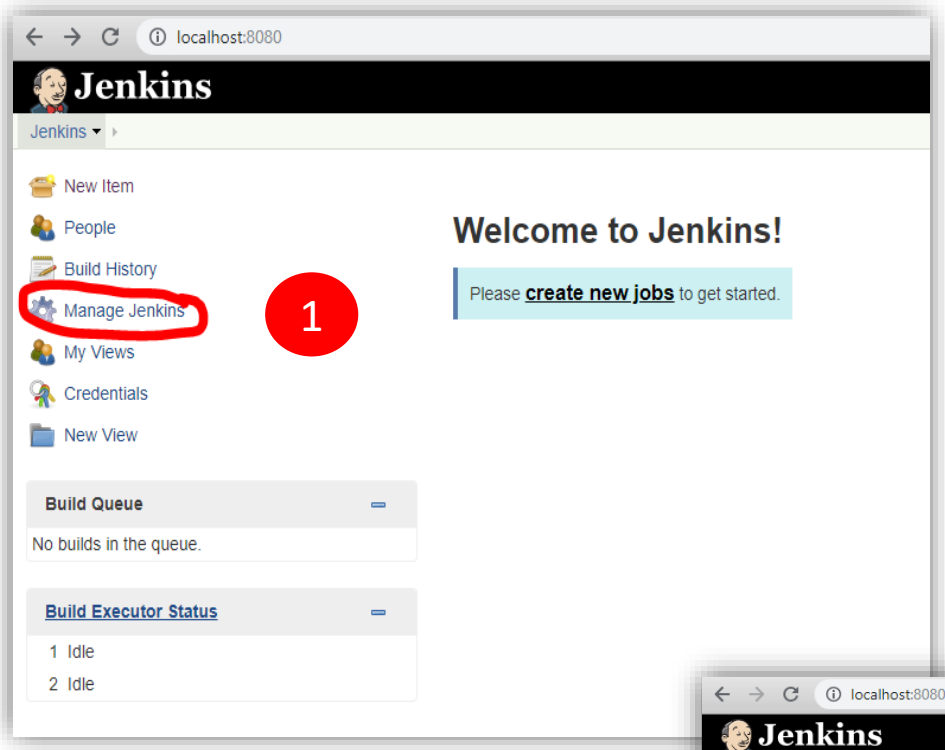


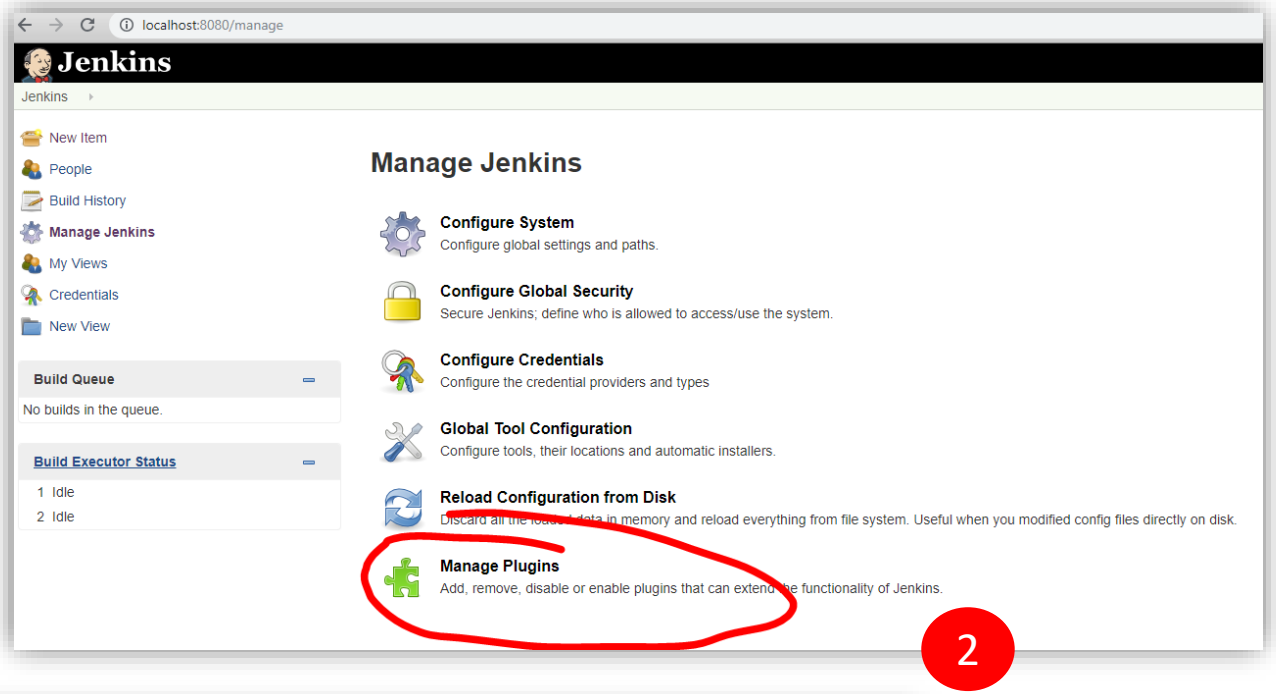
Jenkins Configuration

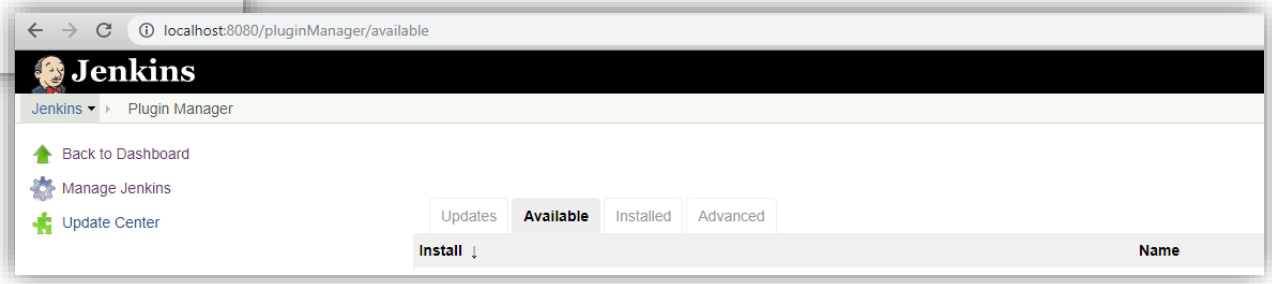




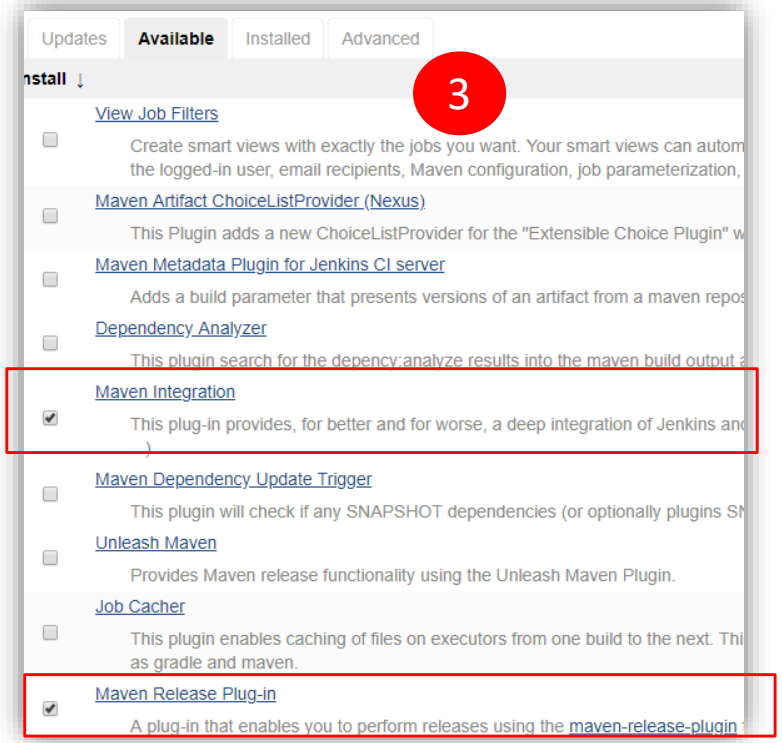
Install Maven Plugins in Jenkins

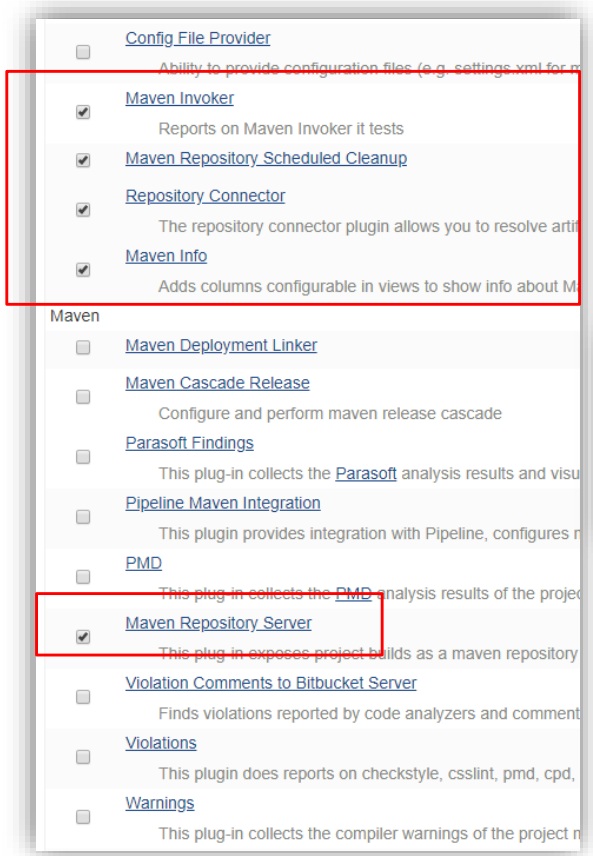


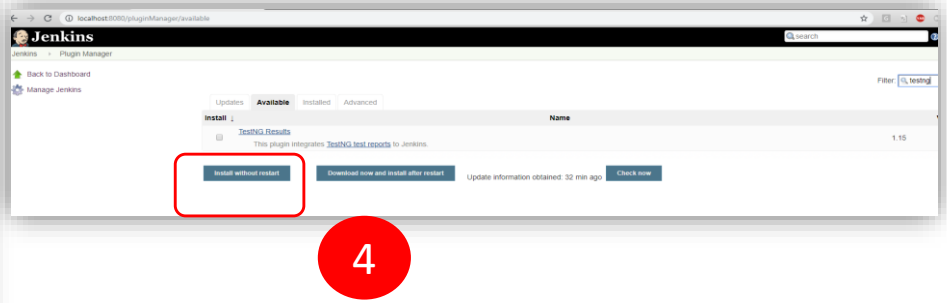




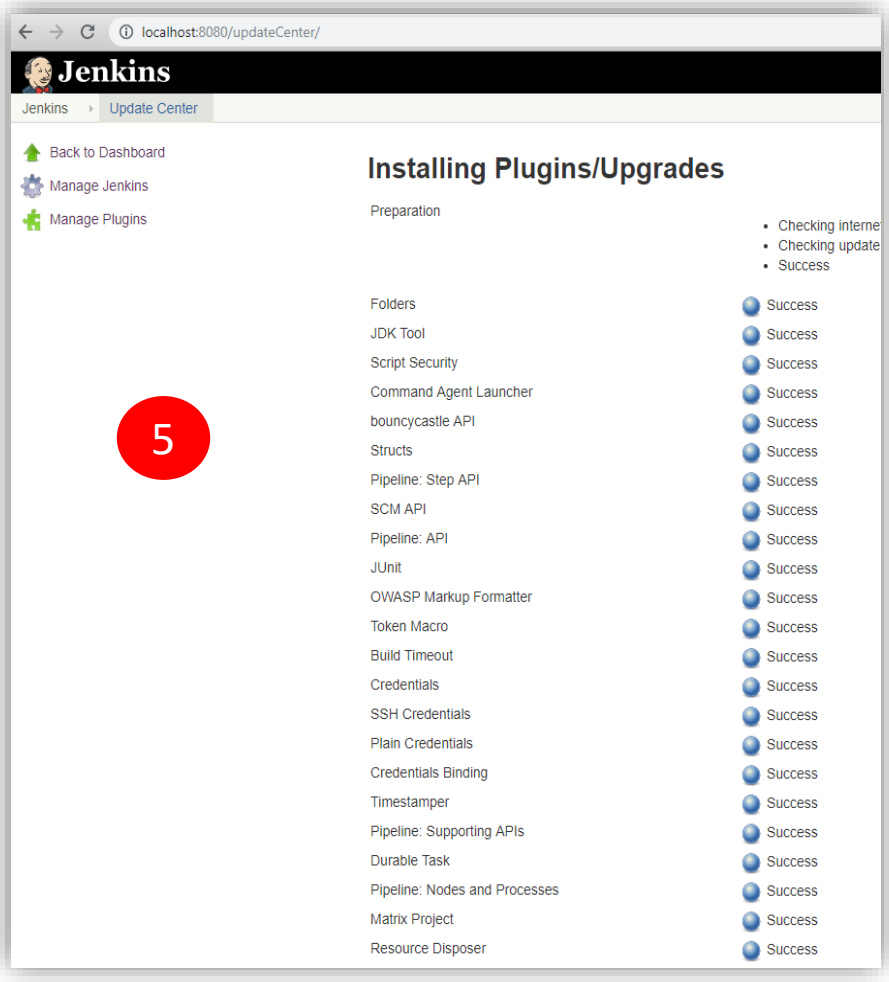
Maven Plugins for Jenkins

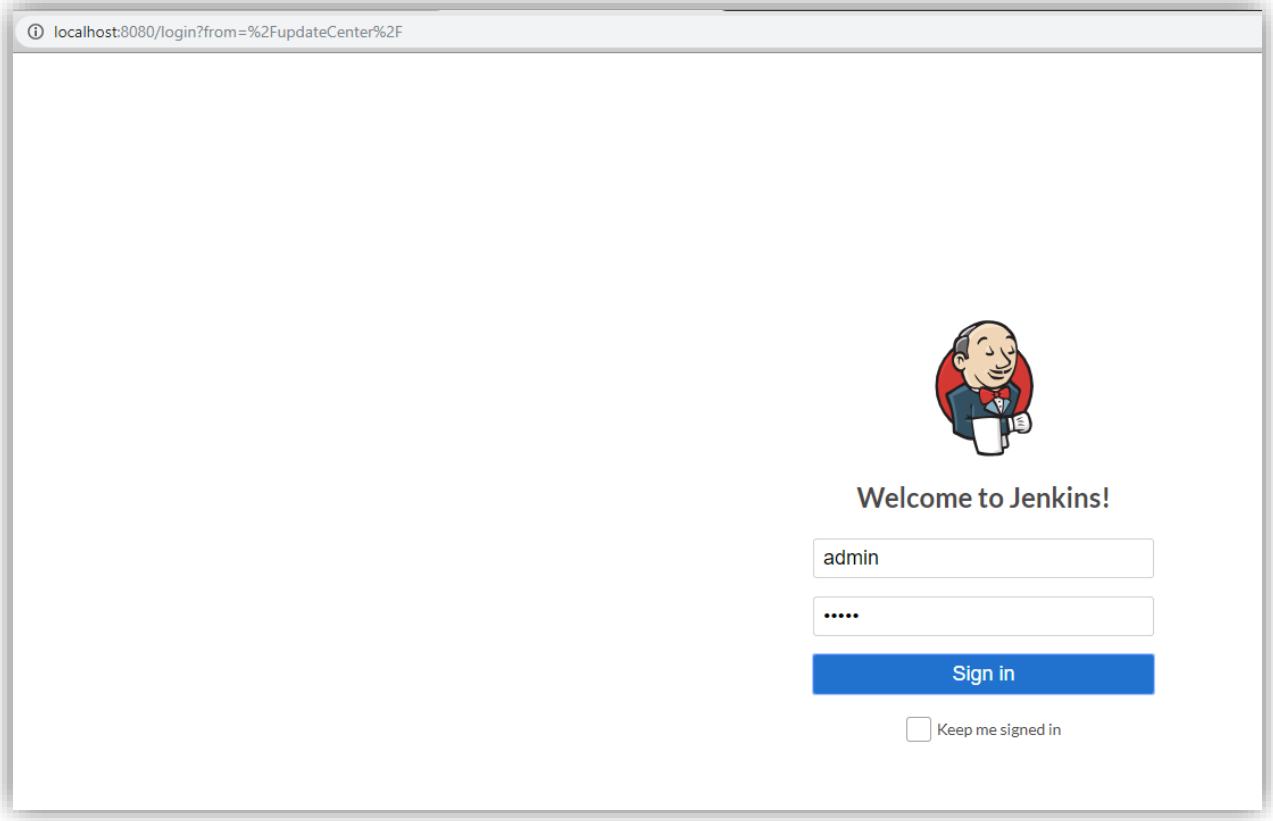






Maven Plugins for Jenkins

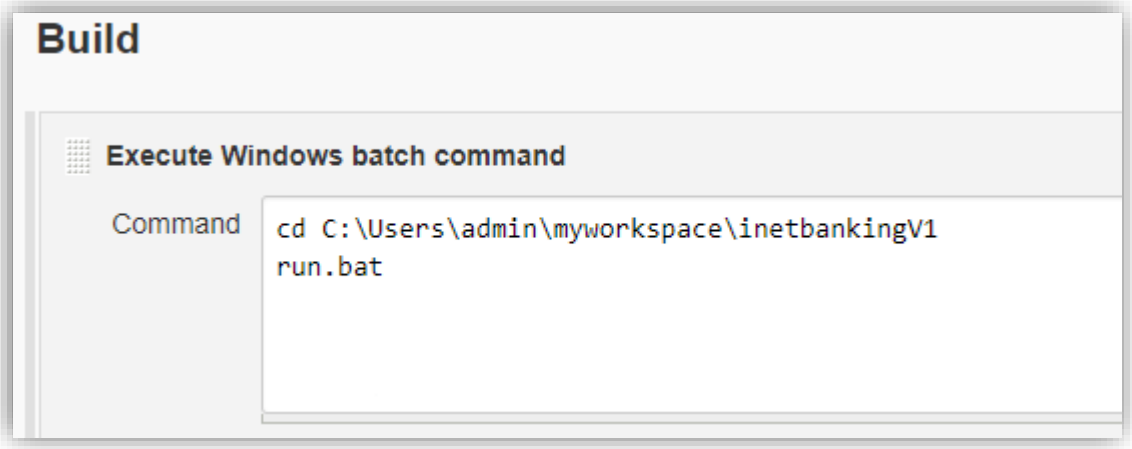




Run Automation through Jenkins

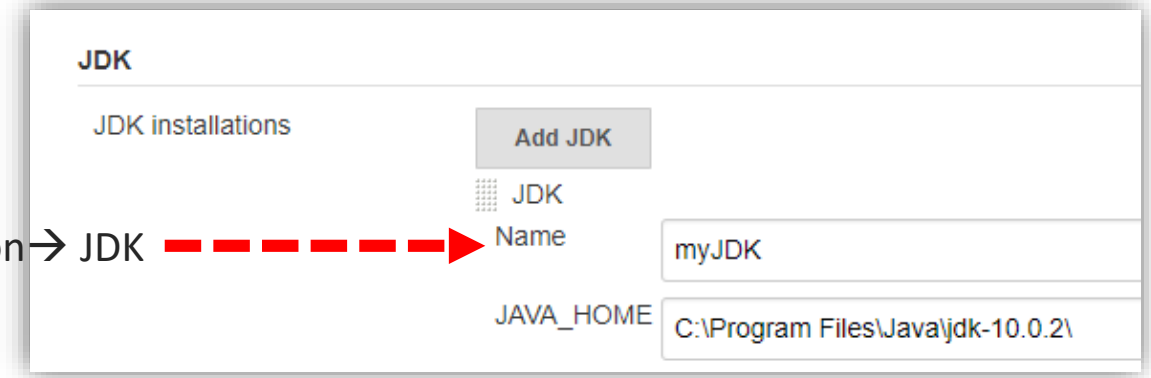
Run Automation using Jenkins Free Style Project

- 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.

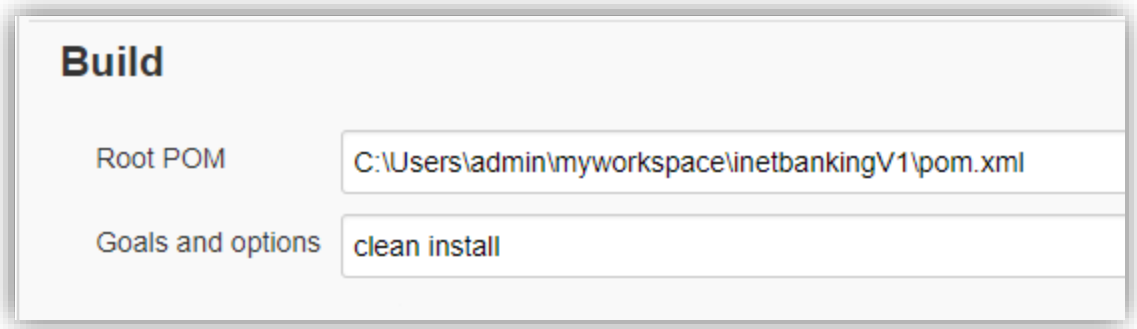


Run Automation using Jenkins Maven Project

- Jenkins URL: http://localhost:8080/
- **Pre-requisite:** JAVA_HOME Configuration in Jenkins
  - Dashboard→ Manage Jenkins→ Global Tool Configuration→ JDK



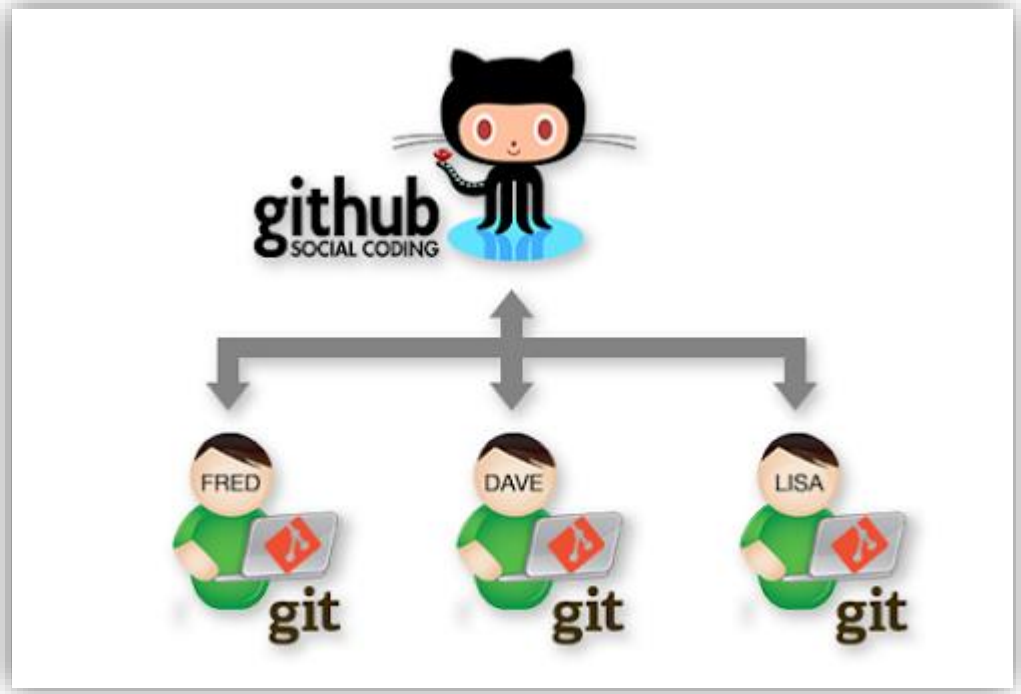
- **Steps:**
- New item → provide name of the project → Maven project →OK.
- Go to build section → Specify path of pom.xml and Goals as shown in picture
- Go to Dash board → you can see new item(project) is created.

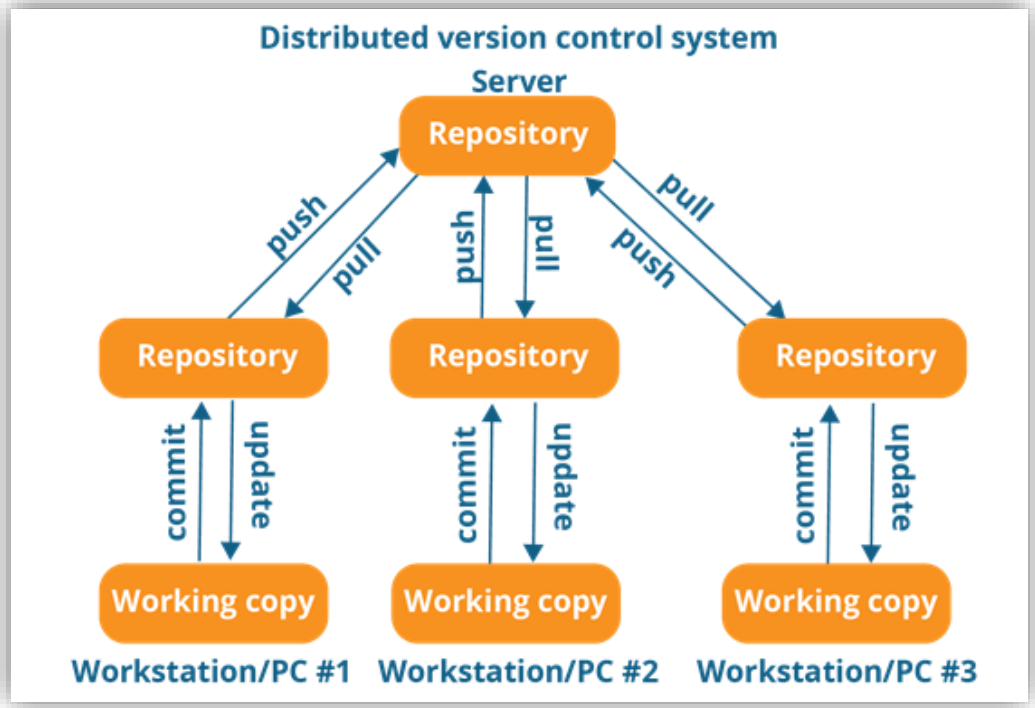


Git & GitHub

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**.

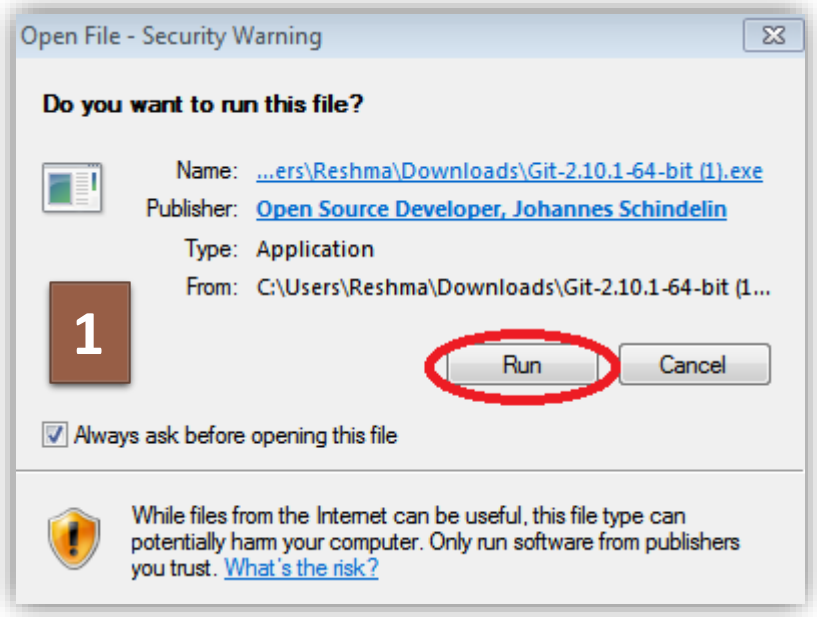


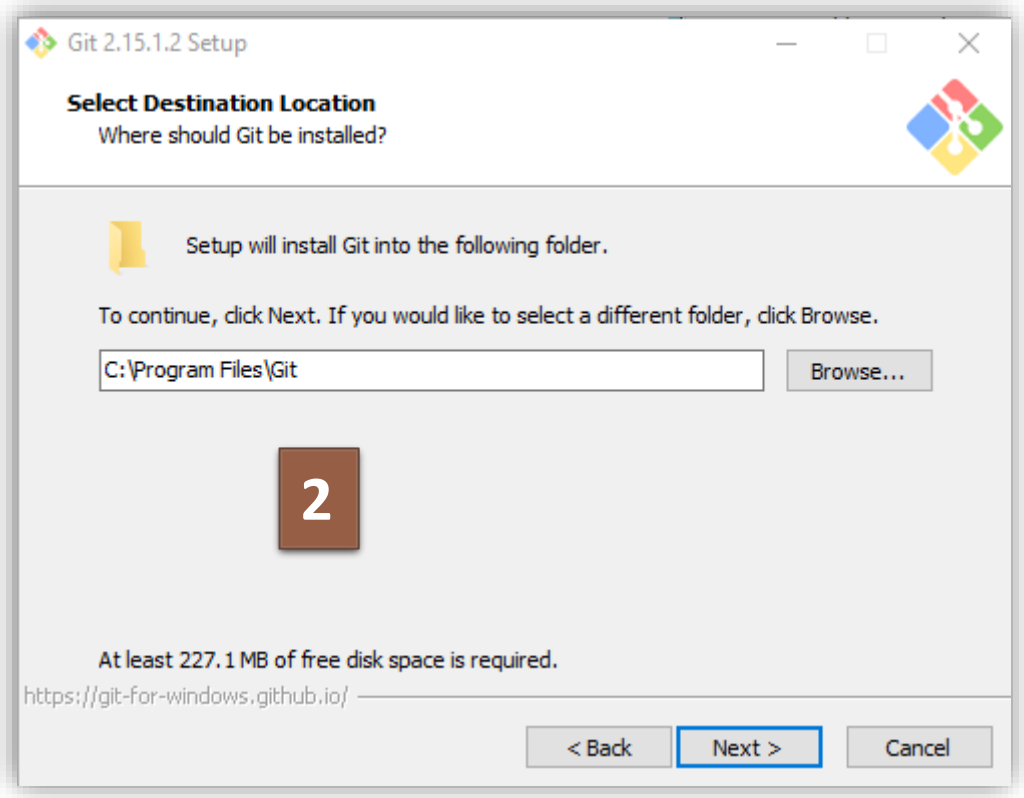


GIT Installation on Windows

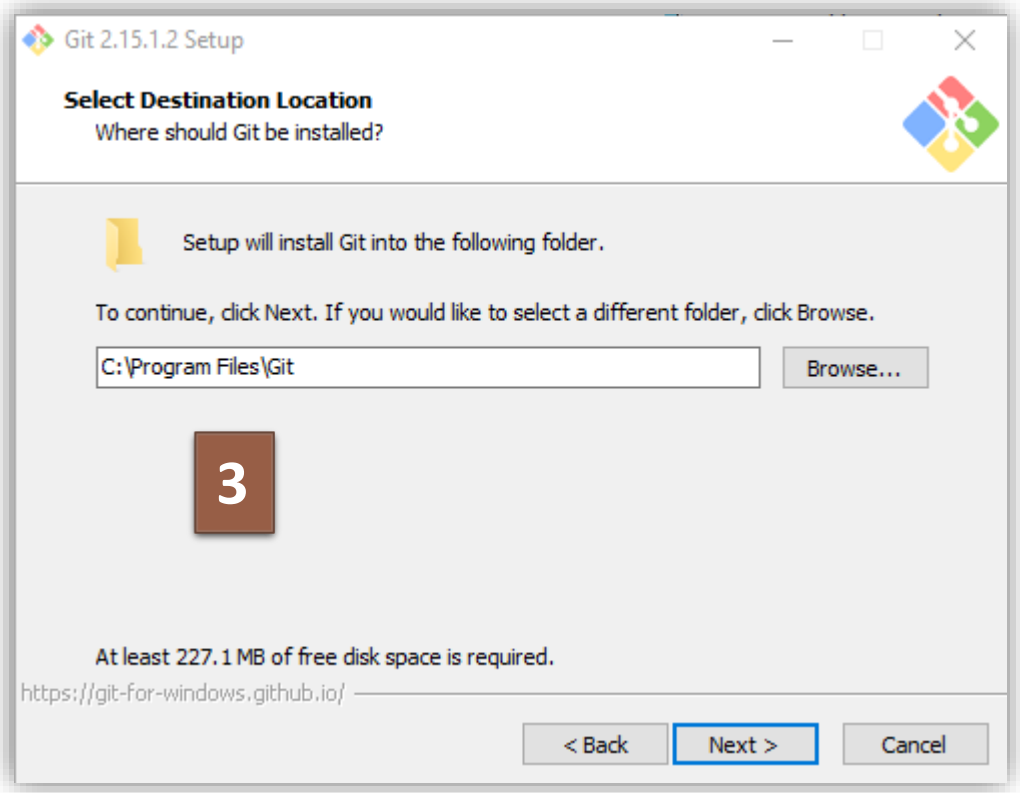
- To download the latest version of Git, click on the link below:
- https://git-scm.com/download/win/

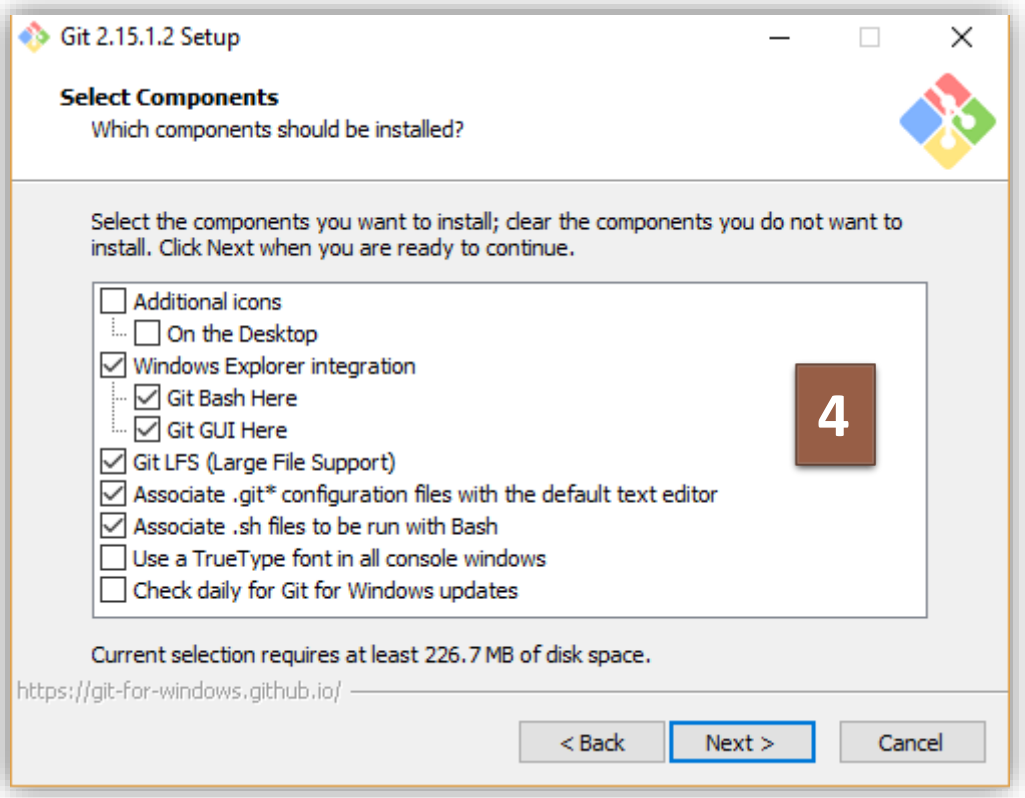
GIT Installation on Windows



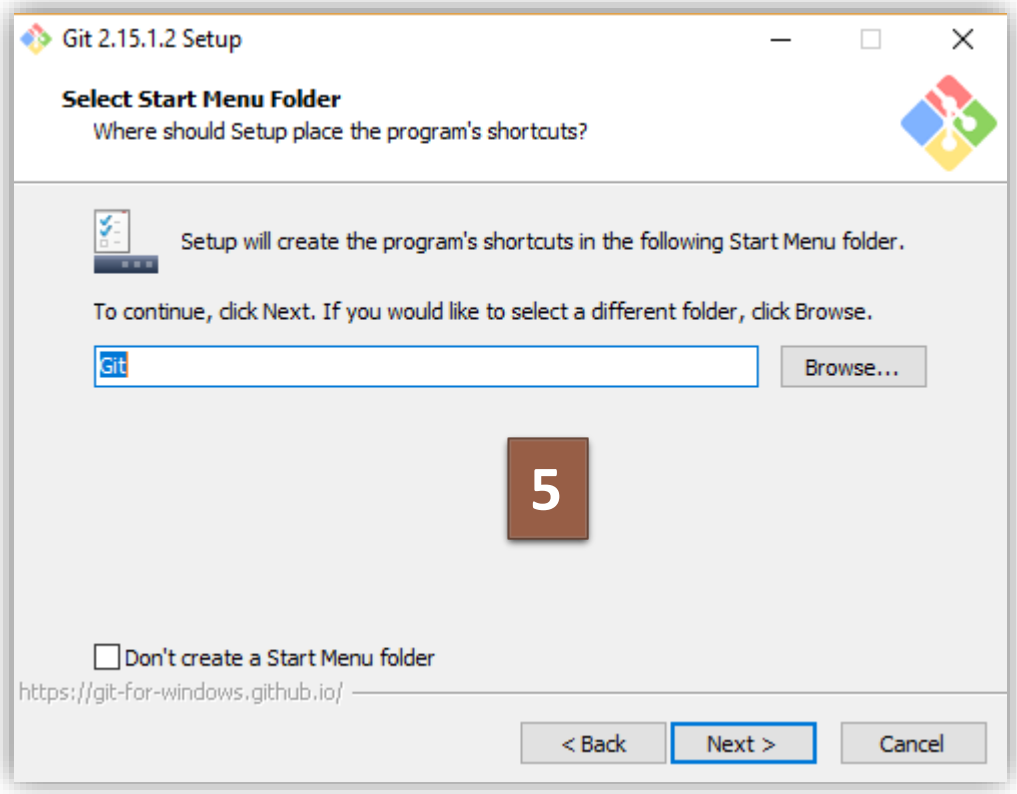


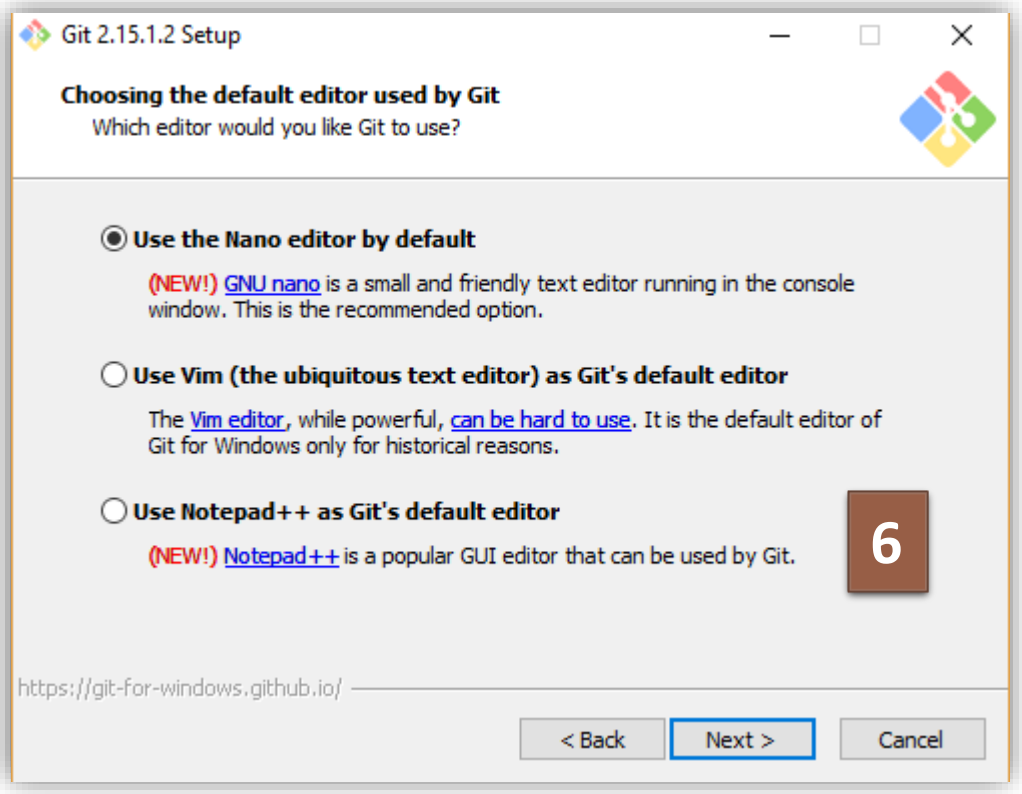
GIT Installation on Windows



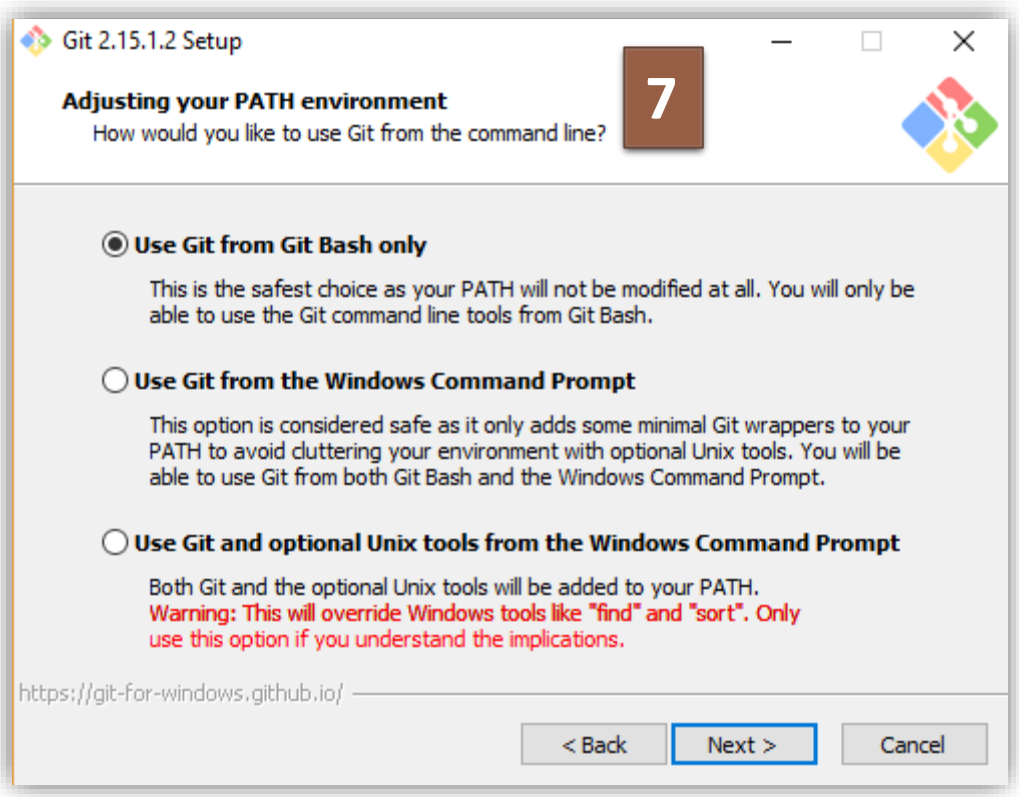


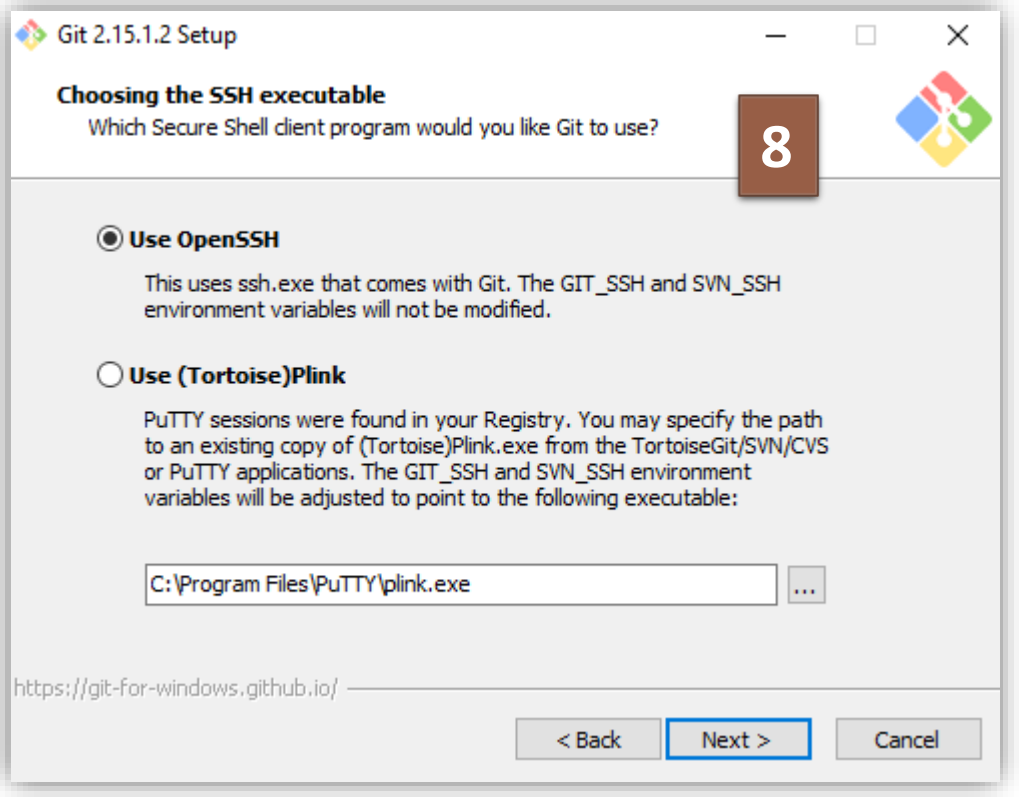
GIT Installation on Windows



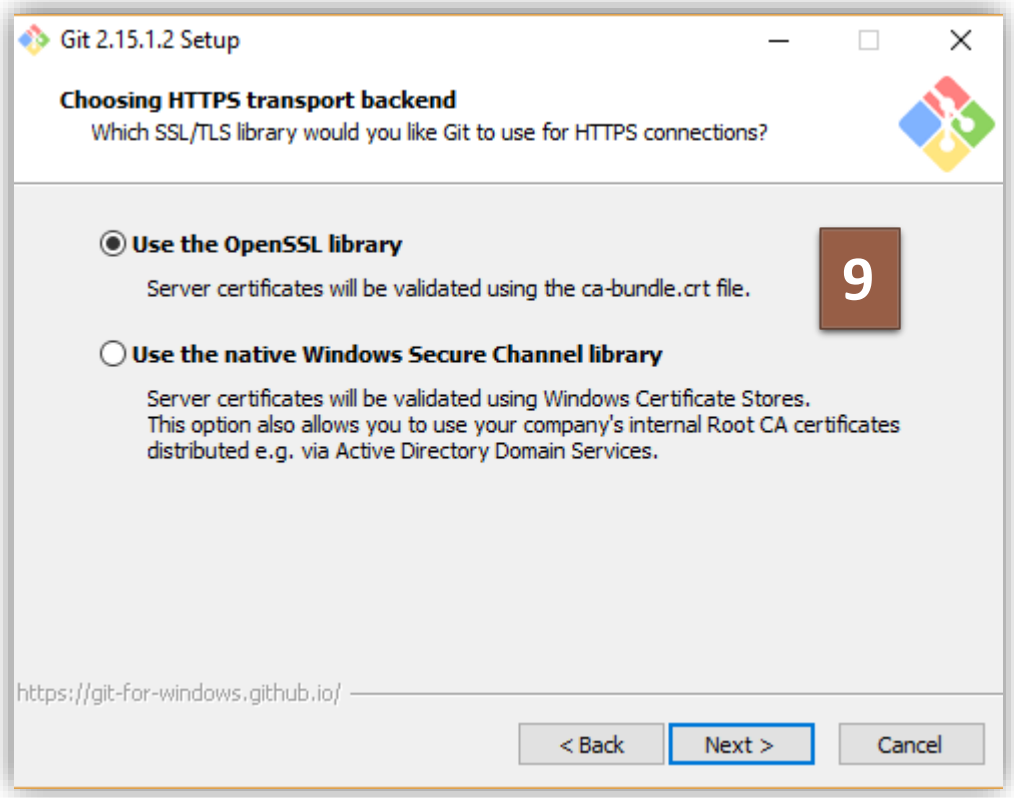


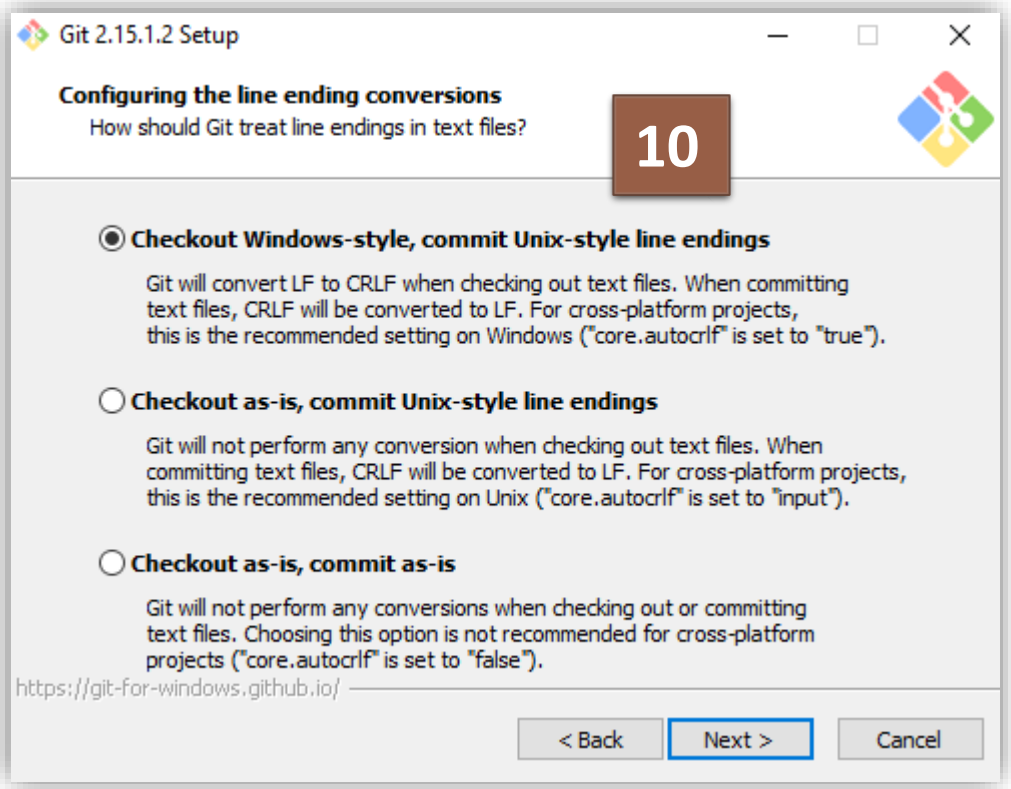
GIT Installation on Windows



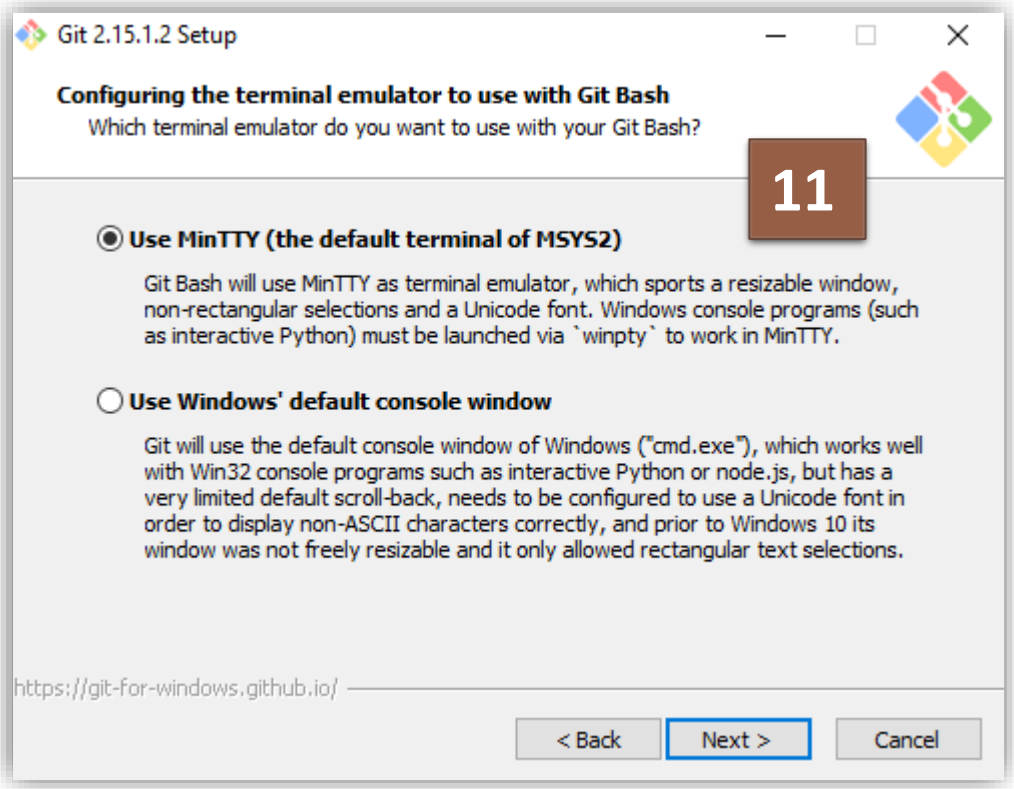


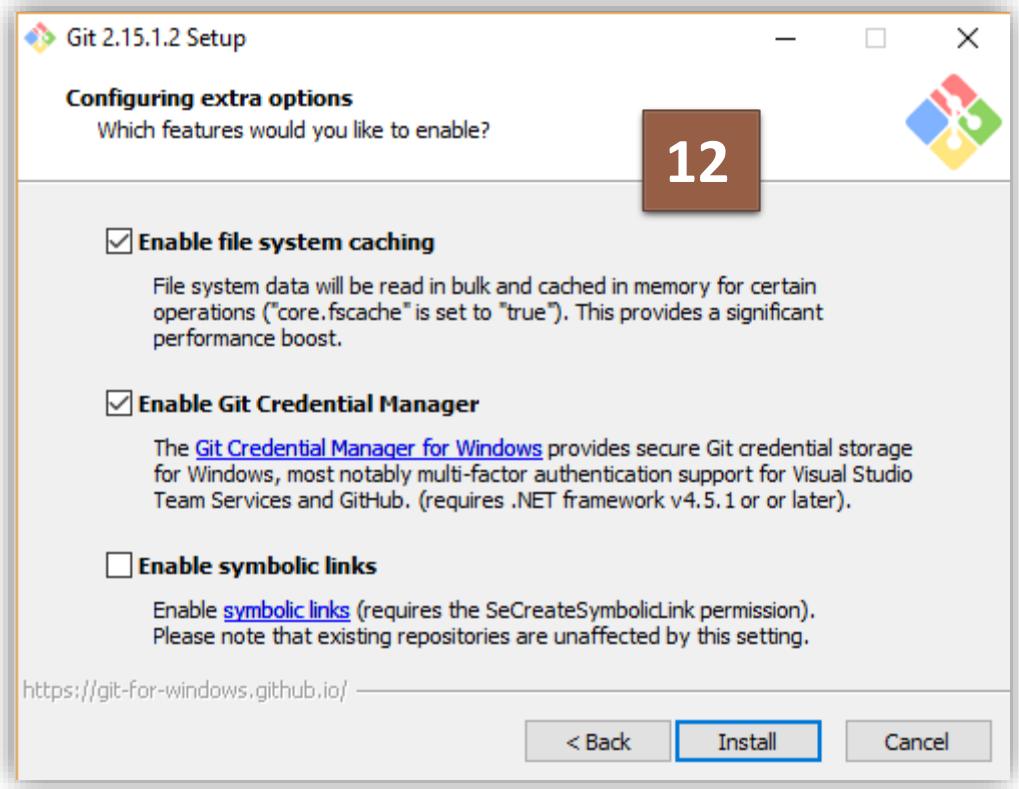
GIT Installation on Windows



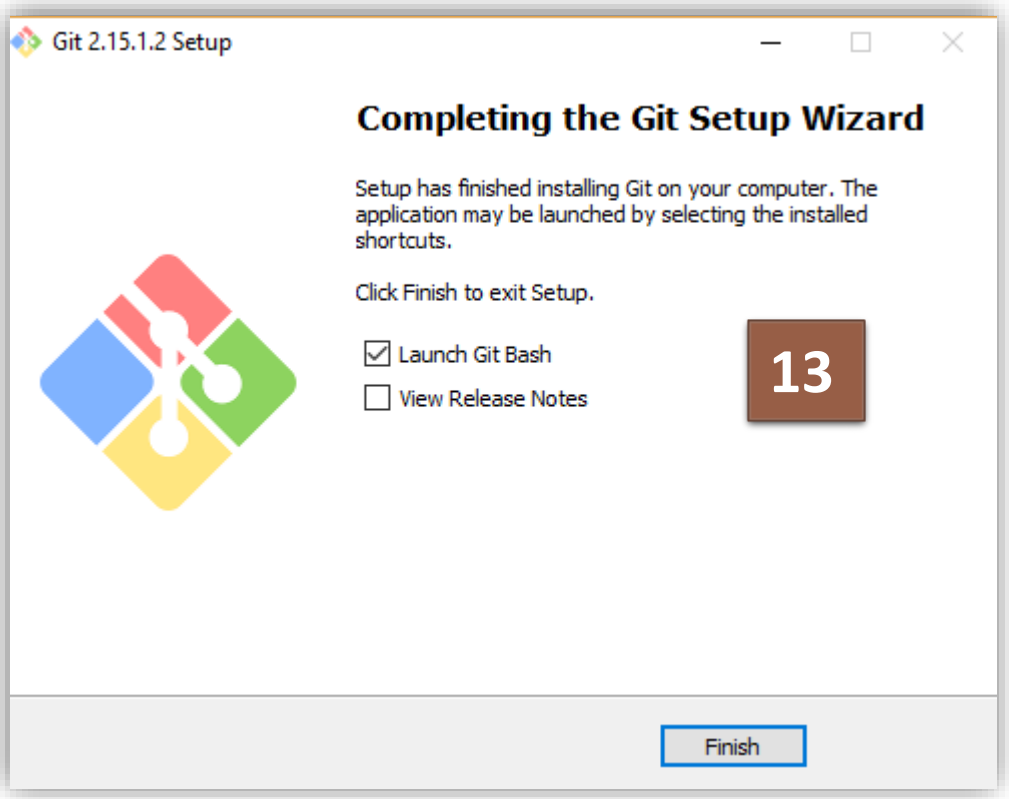


GIT Installation on Windows

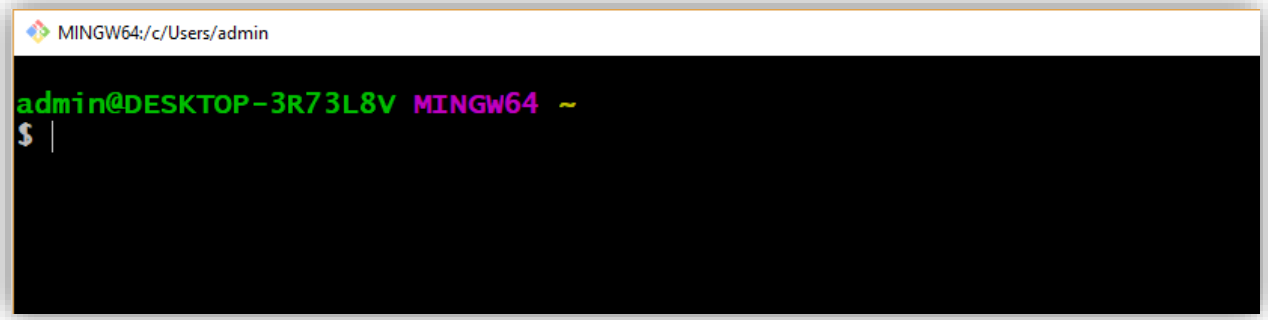




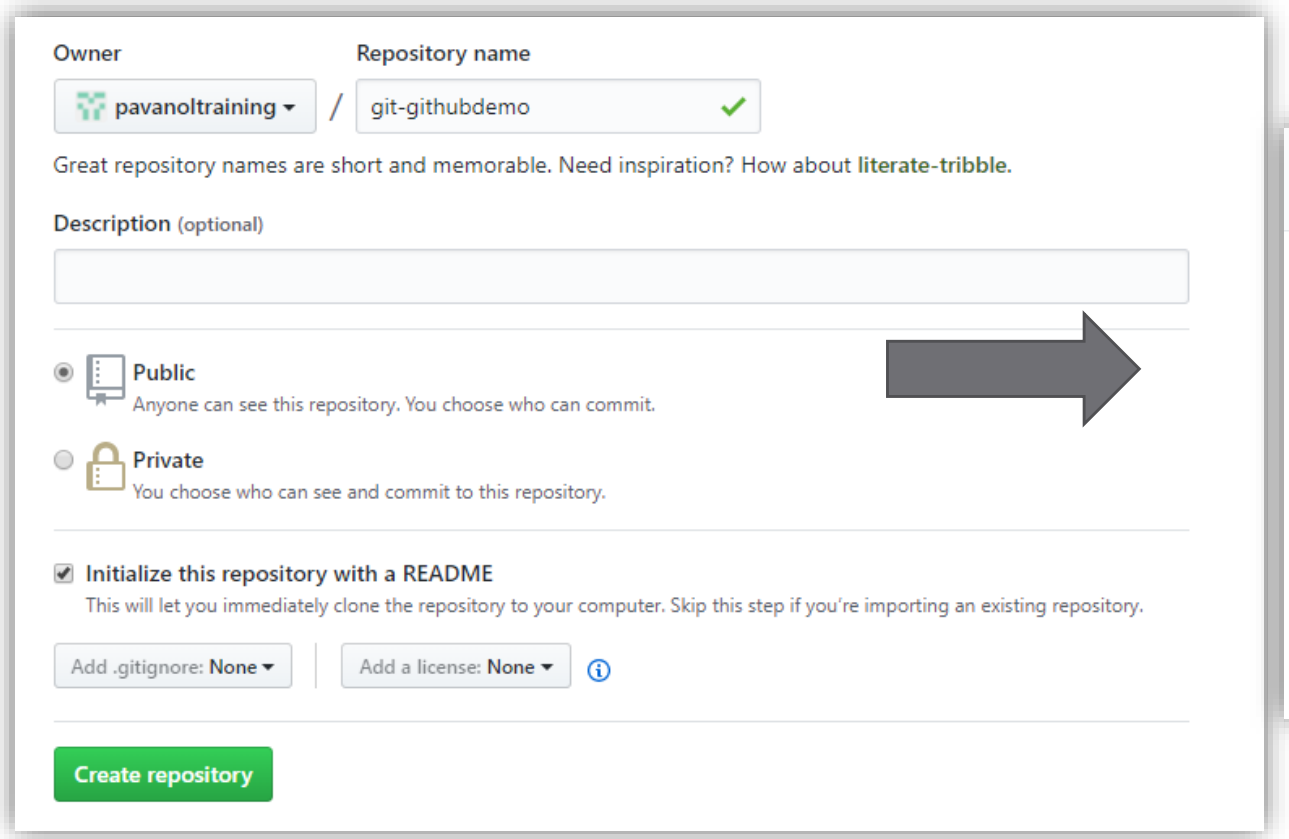
GIT Installation on Windows

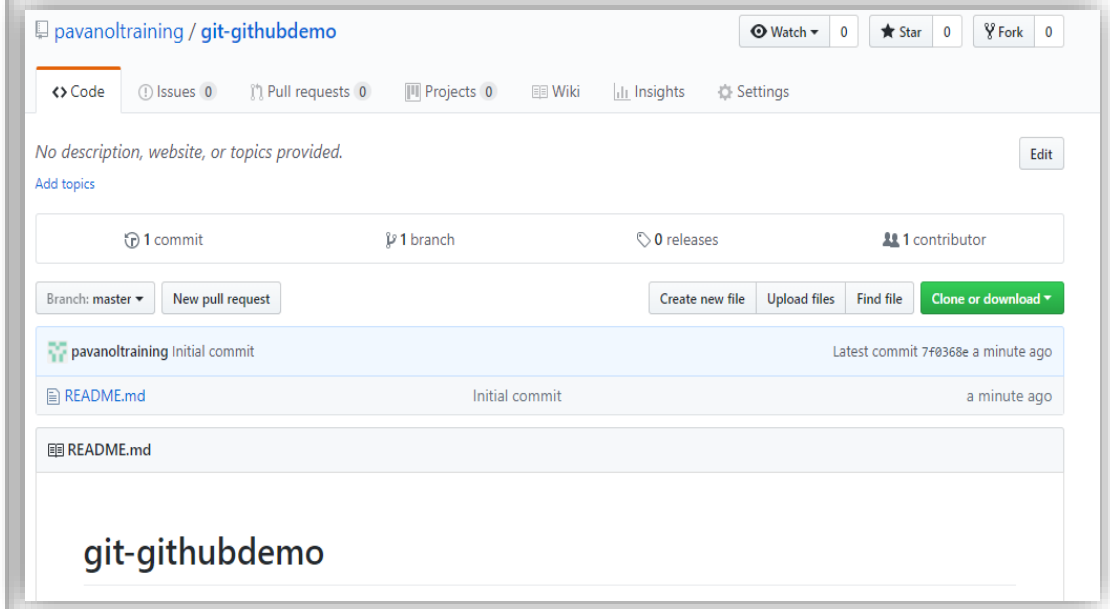


This will launch Git Bash on your screen which looks like the snapshot below:

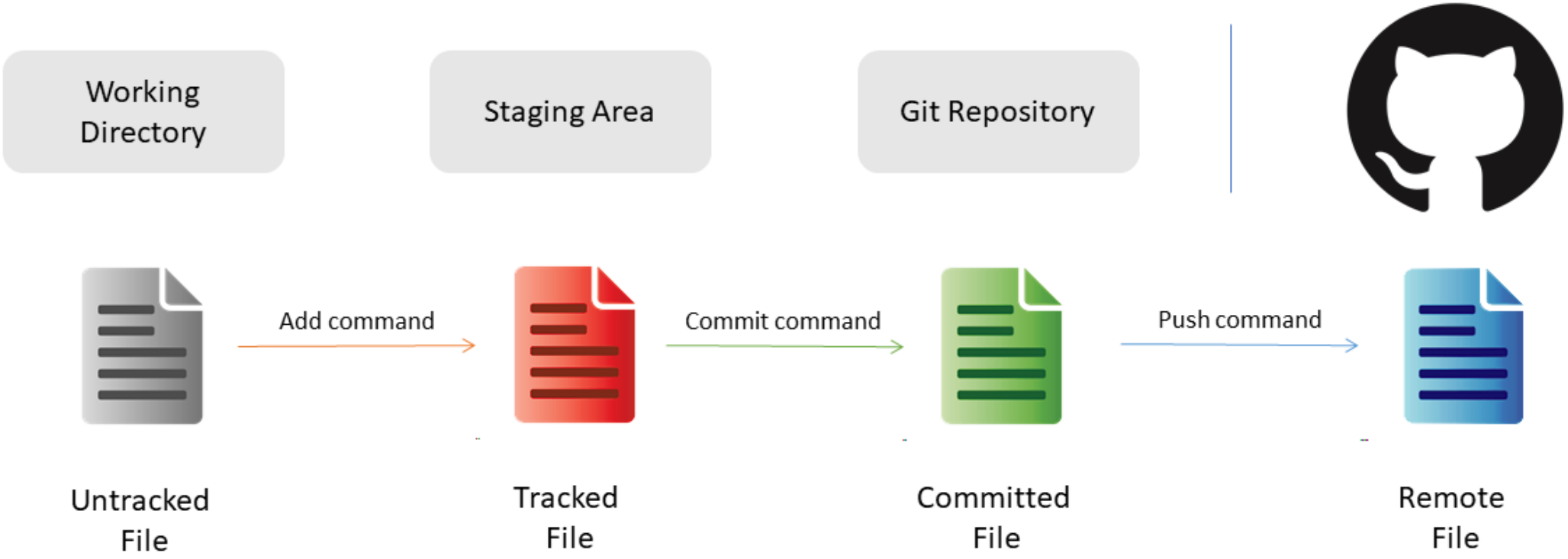


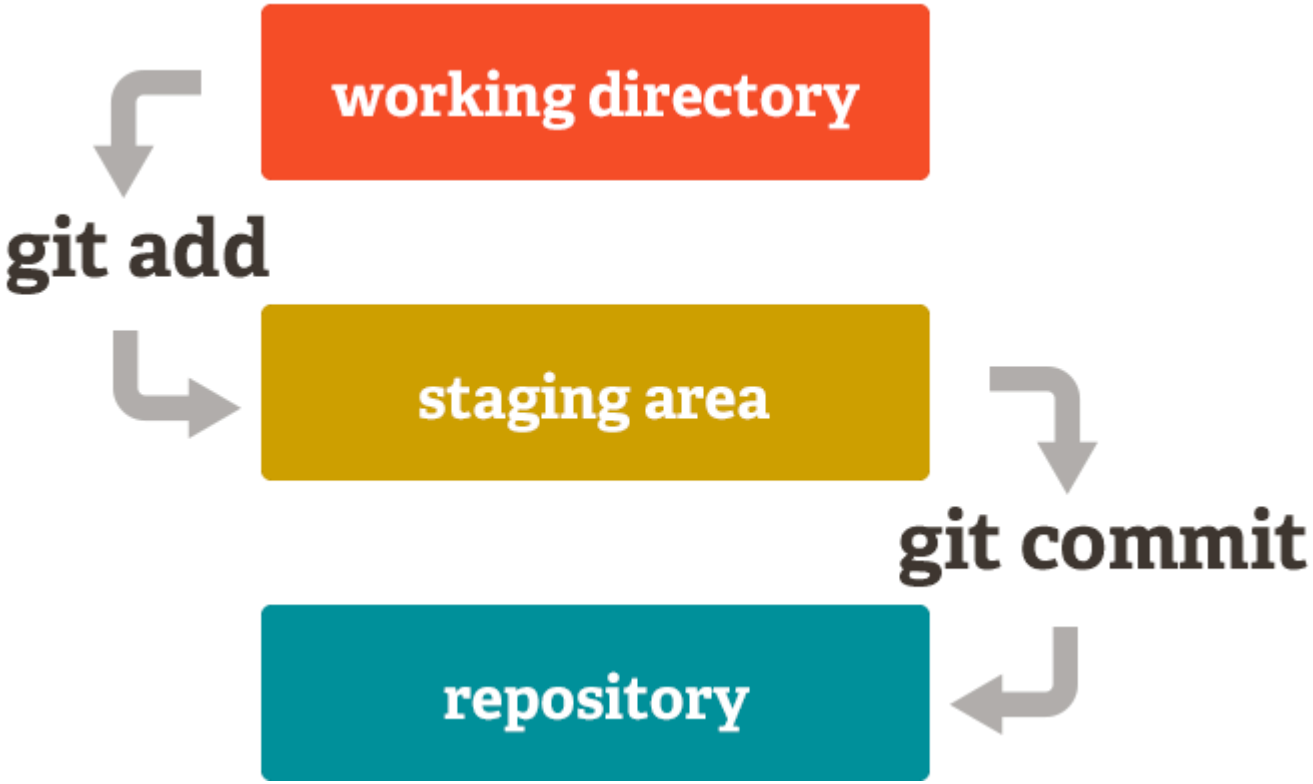
GITHUB Account





Workflow


Working Directory
Staging Area
Git Repository
Add command
Commit command
Push command
Untracked File
Tracked File
Committed File
Remote File


working directory
git add
staging area
git commit
repository

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/project.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