

# Tutorial - Jenkins

## Working with Integration Tool: Jenkins

This tutorial is going to cover the following topics:

- Jenkins core concepts
- Jenkins installation
  - Installing Jenkins using Docker
  - Using a Virtual Machine for Jenkins
  - Not using Virtual Machine? Fresh Install Jenkins
- After installed Jenkins: Jenkins setup
- Accessing Jenkins
- Ngrok installation
- Integrate Jenkins with GitHub using web-hooks
- Continuous integration using Jenkins
- **Exercises (At the end of this tutorial)**

<https://powcoder.com>

Assignment Project Exam Help

Assignment Project Exam Help  
Add WeChat powcoder

<https://powcoder.com>

Add WeChat powcoder

## Jenkins Core Concepts

What is Jenkins? Jenkins is an automated server which helps in the software development process. For example, it can be used to trigger code builds when there is changes made to the source code repository.

Below find the steps for installing and configuring Jenkins. The installation can be difficult, so pay attention to the steps carefully while configuring Jenkins.

## Installing Jenkins

## Installing Jenkins using Docker

One of the easiest ways of setting up Jenkins is through Docker. We will cover docker next week, however for now, know that docker is a software that allows delivering software in containers.

You must install docker first. Follow [this tutorial](https://docs.docker.com/get-docker/) [\(https://docs.docker.com/get-docker/\)](https://docs.docker.com/get-docker/) to install docker.

Once docker is installed, run the following command:

```
docker run -d -v jenkins_home:/var/jenkins_home -p 8080:8080 -p 50000:50000 --name jenkins jenkins/jenkins:lts
```

To get the unlock password as described in the **After installed Jenkins: Jenkins setup** section, run:

```
docker exec -it jenkins /bin/bash
jenkins@c537b805f333:/$ cat /var/jenkins_home/secrets/initialAdminPassword
```

*And keep note of the password.*

To stop the containers whenever needed, run:

```
docker stop jenkins
```

Visit [localhost:8080](http://localhost:8080) [\(http://localhost:8080\)](http://localhost:8080) for the Jenkins portal.

Jump to **After installed Jenkins: Jenkins setup** after having done these steps.

## Using a Virtual Machine for Jenkins

We have a Virtual Machine ready for you that has Jenkins and ngrok installed respectively. You may download this from [here \(https://canvas.sydney.edu.au/courses/25875/pages/soft2412-virtual-machine?module\\_item\\_id=864405\)](https://canvas.sydney.edu.au/courses/25875/pages/soft2412-virtual-machine?module_item_id=864405). You may be prompted to update the network settings to the wifi interface before the machine loads. The password for the user "soft2412" is "**soft2412**". Kindly note that gradle is not installed in the VM....figure out the workaround to this without installing gradle...?? (Think about wrapper).

After you login to the Ubuntu machine. visit [localhost:8080](http://localhost:8080) [\(http://localhost:8080\)](http://localhost:8080) in firefox, you should see a jenkins login page. Again, the username and password is same as soft2412. After you have done this, jump to **Integrate Jenkins with GitHub** section of the tutorial.

## Not using Virtual Machine? - Fresh Install Jenkins

If the virtual machine runs slow on your machine or you cannot use docker, you may choose to install and run jenkins as a software in your own PC.

For that you will have to install Jenkins in your machine if you do not have it.

## Linux users



For ubuntu/debian users, we will use the apt-get package. For other distributions, use yum/dnf/pacman.

```
sudo apt-get update
sudo apt update
```

Then, install java:

```
sudo apt install openjdk-8-jdk
```

Then, import the public GPG key of jenkins:

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
```

For ubuntu/debian users, run the following command:

```
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
```

Then you can run:

```
sudo apt update
sudo apt install jenkins
```

Then, after jenkins gets successfully installed, run the following commands:

```
sudo systemctl start jenkins
```

You can check the status of jenkins through:

```
sudo systemctl status jenkins
```

Output:

```
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Fri 2019-09-06 18:10:18 AEST; 2min 3s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 4658)
   CGroup: /system.slice/jenkins.service

Sep 06 18:10:17 soft2412-VirtualBox systemd[1]: Starting LSB: Start Jenkins at boot time...
Sep 06 18:10:17 soft2412-VirtualBox jenkins[7260]: Correct java version found
Sep 06 18:10:17 soft2412-VirtualBox jenkins[7260]: * Starting Jenkins Automation Server jenkins
Sep 06 18:10:17 soft2412-VirtualBox su[7308]: Successful su for jenkins by root
```

```
Sep 06 18:10:17 soft2412-VirtualBox su[7308]: + ??? root:jenkins
Sep 06 18:10:17 soft2412-VirtualBox su[7308]: pam_unix(su:session): session opened for user jenkins by (uid=0)
Sep 06 18:10:17 soft2412-VirtualBox su[7308]: pam_unix(su:session): session closed for user jenkins
Sep 06 18:10:18 soft2412-VirtualBox jenkins[7260]: ...done.
Sep 06 18:10:18 soft2412-VirtualBox systemd[1]: Started LSB: Start Jenkins at boot time.
```

Finally. enable jenkins to start on the system boot:

```
sudo systemctl enable jenkins
```

You may find this link helpful: <https://linuxize.com/post/how-to-install-jenkins-on-centos-7/>

Jump to **After installed Jenkins: Jenkins setup.**

<https://powcoder.com>

MacOS users

For MacOSX users, you will need to install jenkins through brew. Brew is a package manager for Mac and Linux.

<https://powcoder.com>

Type brew to check if you have it installed:

```
brew --version
```

If you have brew installed you should see something similar to:

```
Homebrew 2.1.11
Homebrew/homebrew-core (git revision a5bdae; last commit 2019-08-26)
Homebrew/homebrew-cask (git revision 8442f; last commit 2019-08-26)
```

Make sure java is installed.

```
java --version
```

Now, to install jenkins, run:

```
brew install jenkins
```

This will take some time. You should see the following after the installation:

```
...
==> Downloading http://mirrors.jenkins.io/war/2.193/jenkins.war
==> Downloading from http://ftp-nyc.osuosl.org/pub/jenkins/war/2.193/jenkins.war
##### 100.0%
==> jar xvf jenkins.war
```

==> Caveats

Note: When using launchctl the port will be 8080.

To have launchd start jenkins now and restart at login:

```
brew services start jenkins
```

Or, if you don't want/need a background service you can just run:

```
jenkins
```

==> Summary

 /usr/local/Cellar/jenkins/2.193: 7 files, 78MB, built in 5 minutes 1 second

Then, you can start jenkins by running the following command: (or just run "jenkins")

```
brew services start jenkins
```

Output:

```
==> Successfully started `jenkins` (label: homebrew.mxcl.jenkins)
```

You can access [localhost:8080](http://localhost:8080) (<http://localhost:8080>) to visit the jenkins homepage.

Jump to **After installed Jenkins**. Jenkins setup after having done these steps in Mac.

Assignment Project Exam Help

Assignment Project Exam Help  
Add WeChat powcoder

<https://powcoder.com>

Windows Users



Add WeChat powcoder

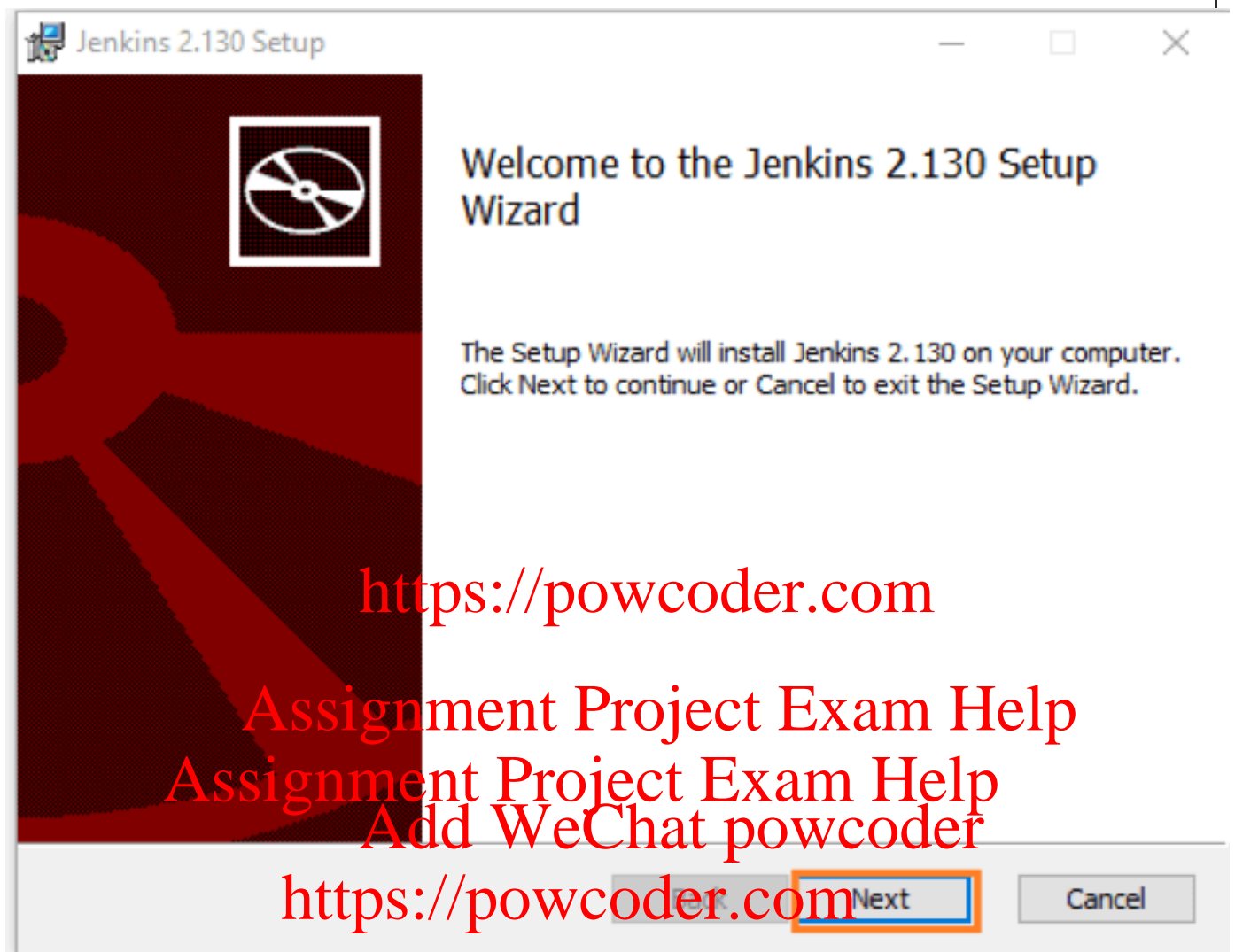
Before installing Jenkins, make sure that you have installed Java (JDK).

```
java -version
```

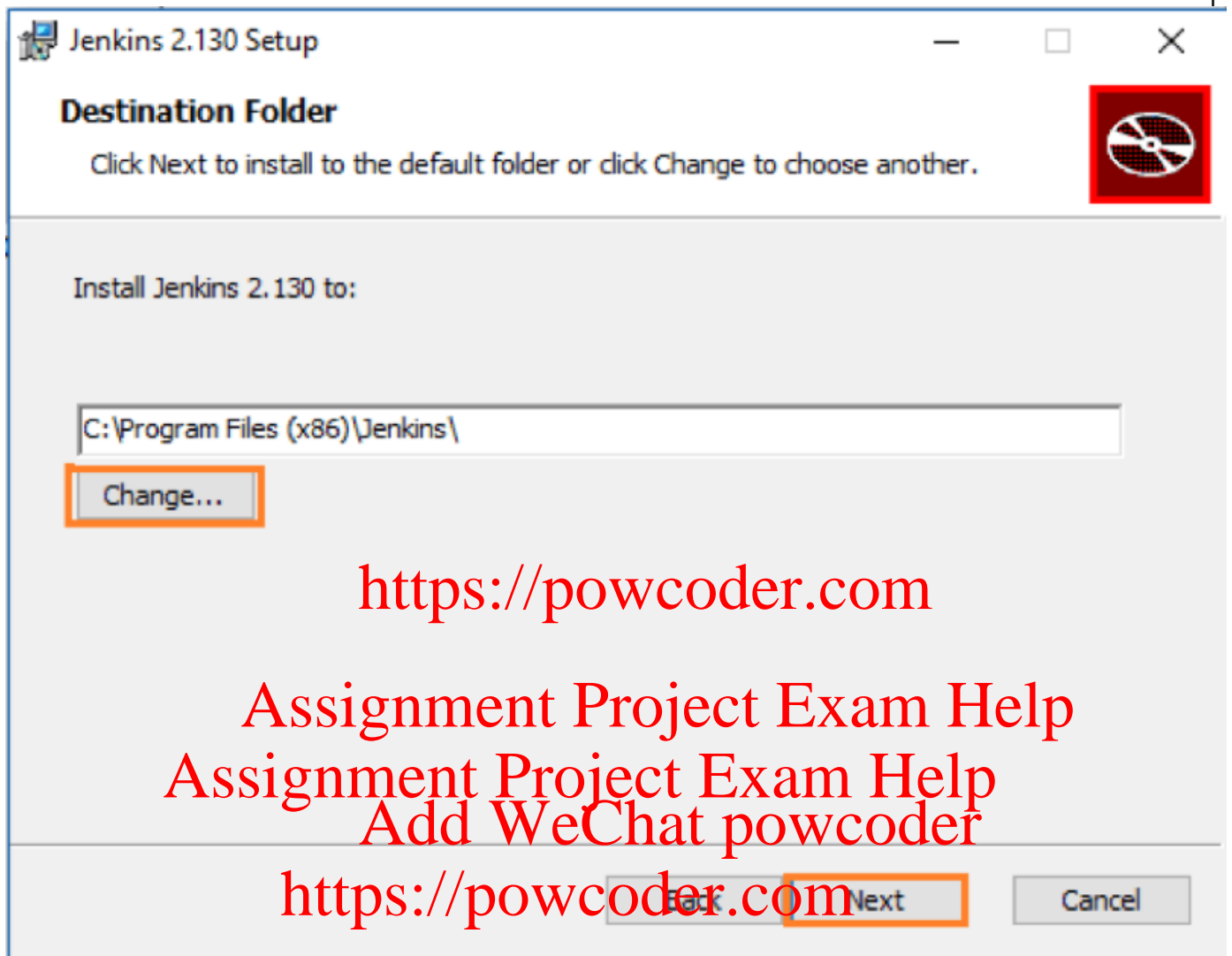
You can download Jenkins installer from this link: <https://jenkins.io/download/>  
(<https://jenkins.io/download/>)

After you select 'windows', your browser will automatically download the installer. Unzip the downloaded file to get "jenkins.exe" ready. These are the step to install Jenkins:

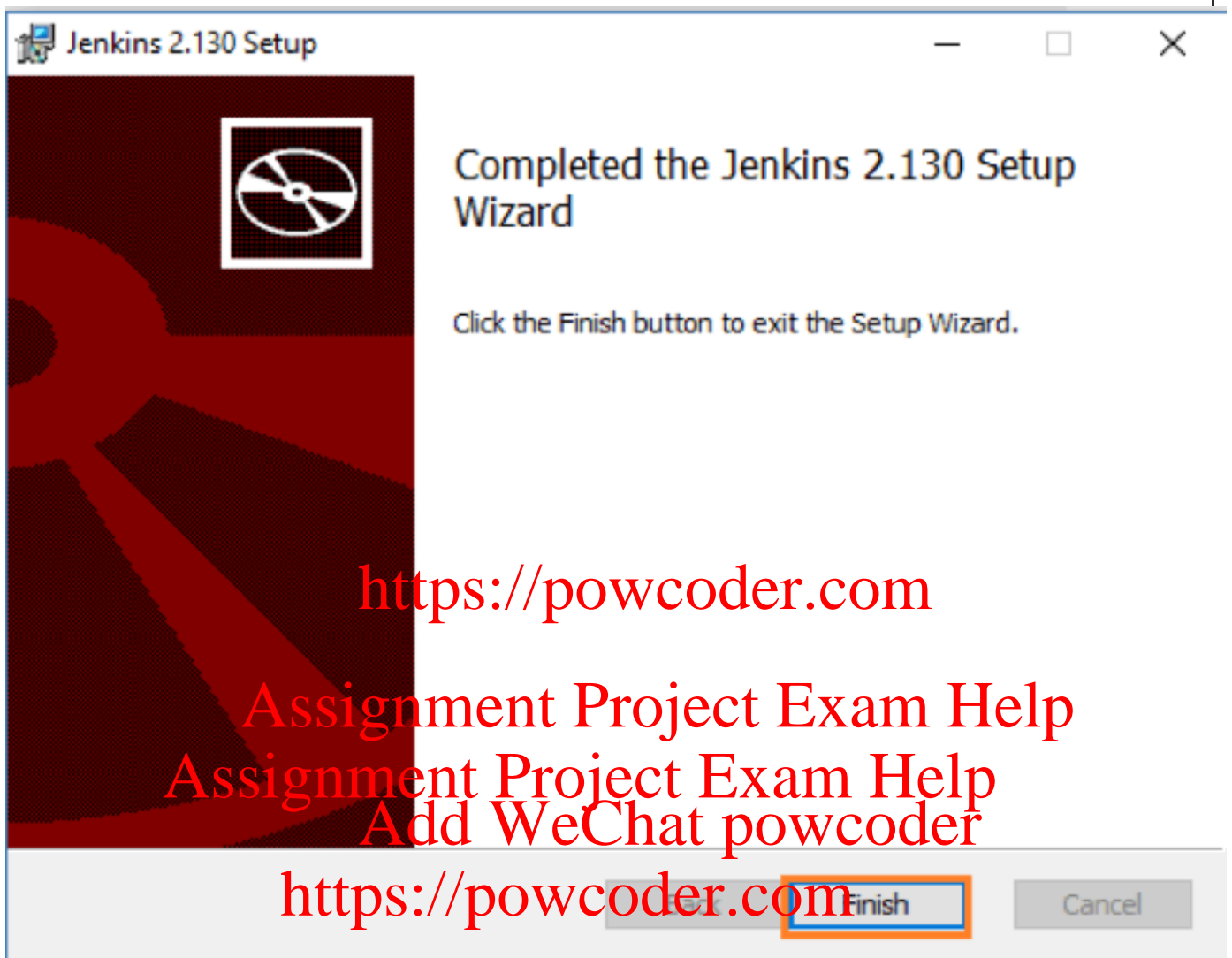
Click "Next" button to start the setup.



Click "Change" if you want to install Jenkins in another folder. Do remember this installation path as we will need the path later. Press "next" to continue the installation.



Click the "Install" button to start the installation and wait until it is done. Afterwards, click "Finish" button after the installation is complete.



You will have to wait for at least 3-5 minutes as the Jenkins service is starting in the background (as a Windows's service). Afterwards, open your browser and navigate to page:

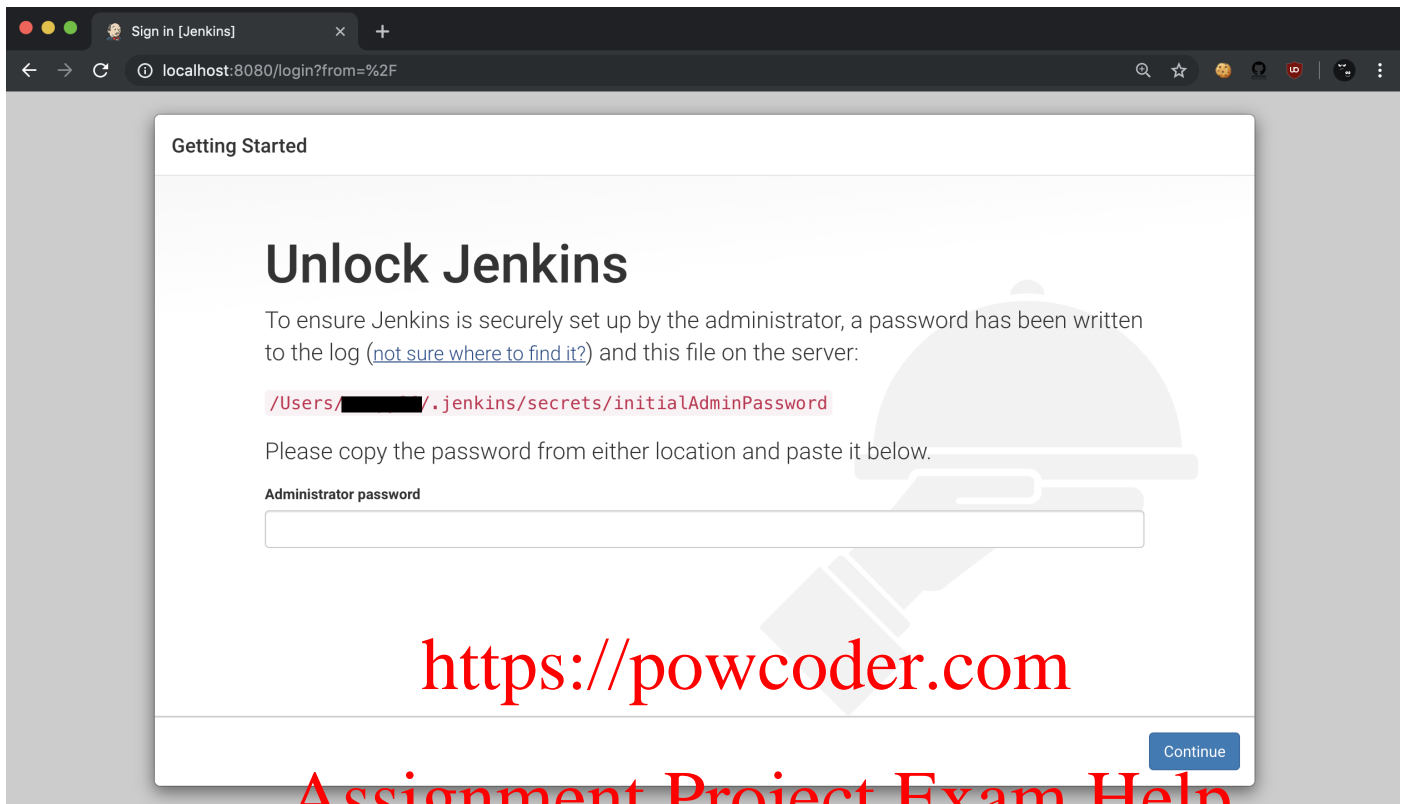
<http://localhost:8080> [\\_ \(http://localhost:8080\) \\_](http://localhost:8080) [\\_ \(http://localhost:8080/\)](http://localhost:8080/)

Jump to **After installed Jenkins: Jenkins setup**

## After installed Jenkins: Jenkins setup

Then visit <http://localhost:8080> [\\_ \(http://localhost:8080\) \\_](http://localhost:8080) to view the homepage of jenkins. You will be prompted to enter a password





<https://powcoder.com>

Assignment Project Exam Help

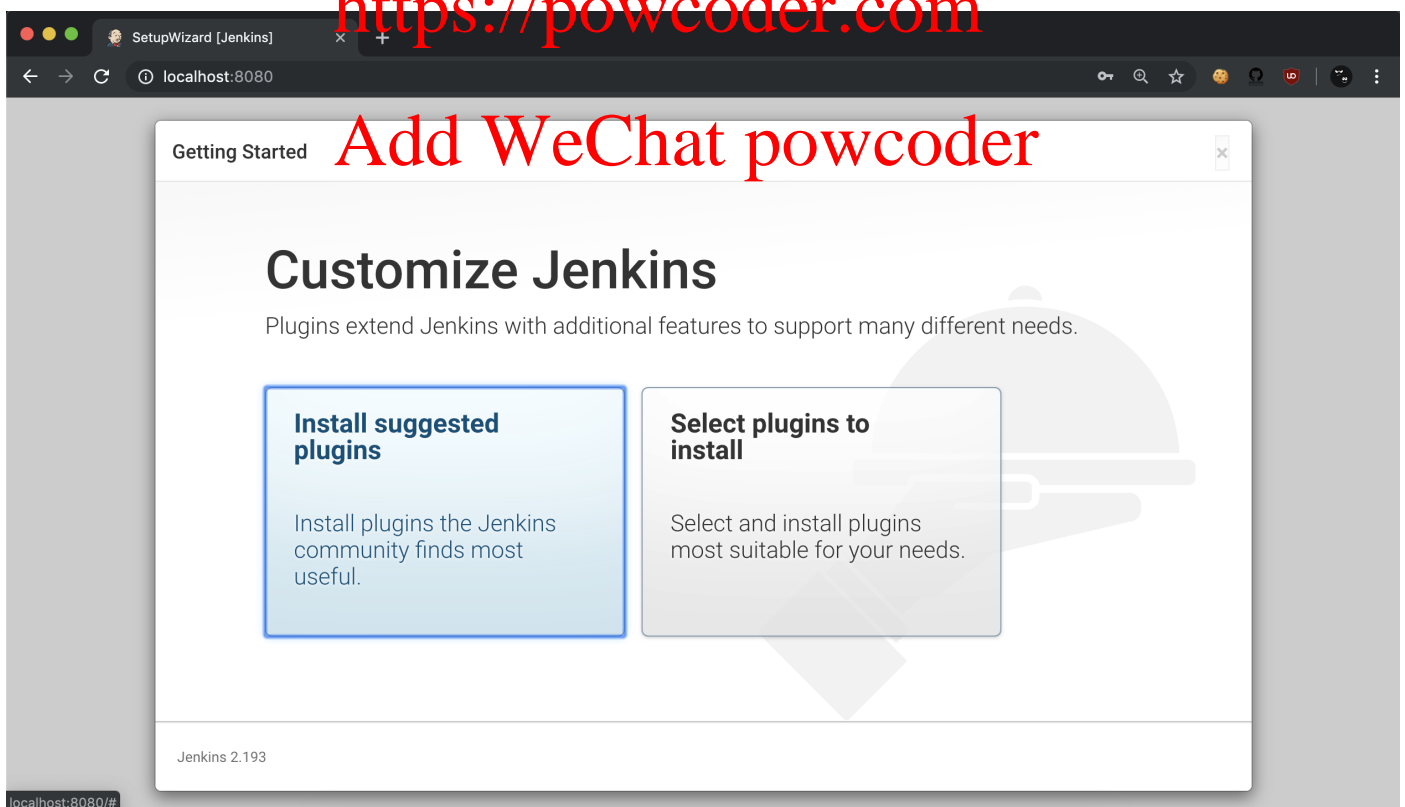
Copy paste the password from the file specified in the above image.

When you do so, you should be redirected here:

Assignment Project Exam Help

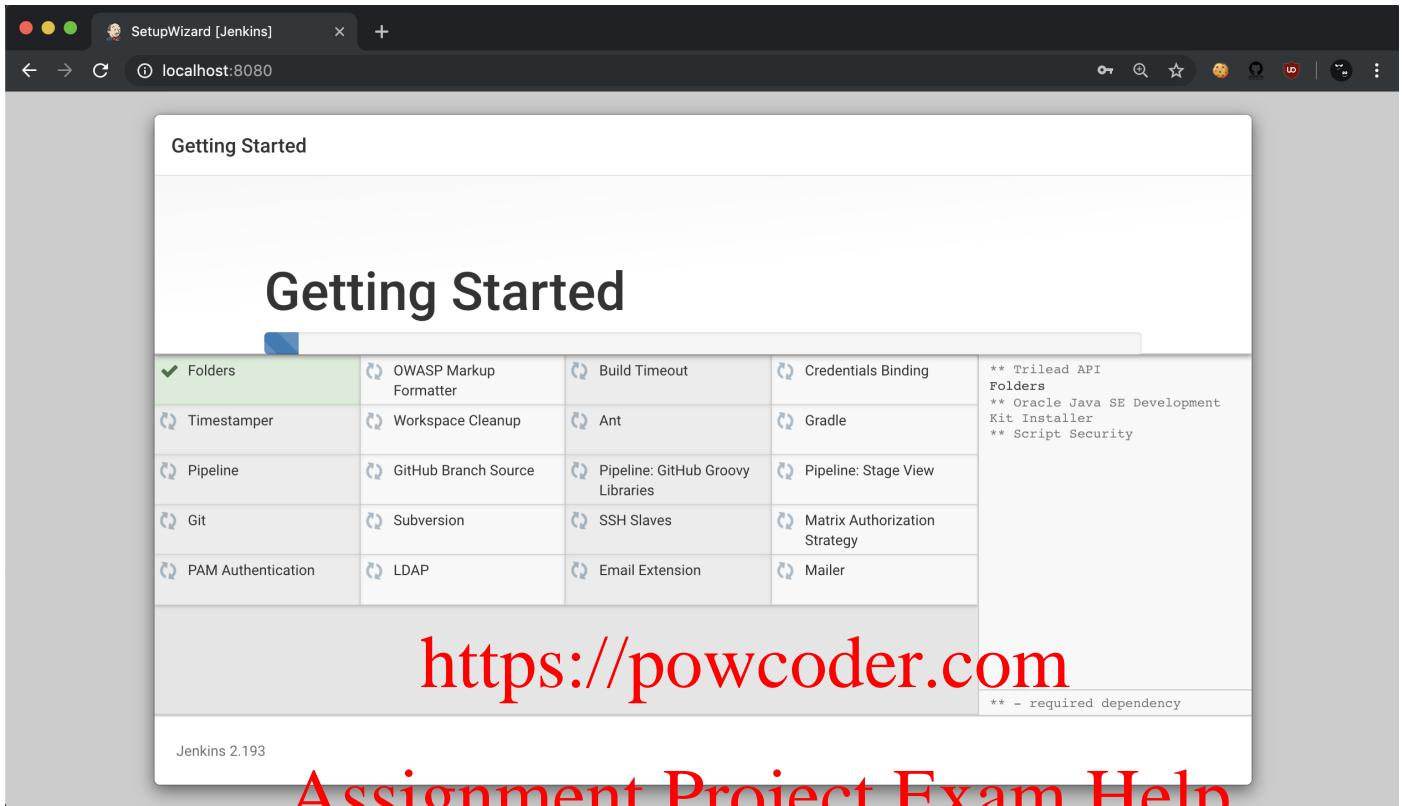
Add WeChat powcoder

<https://powcoder.com>

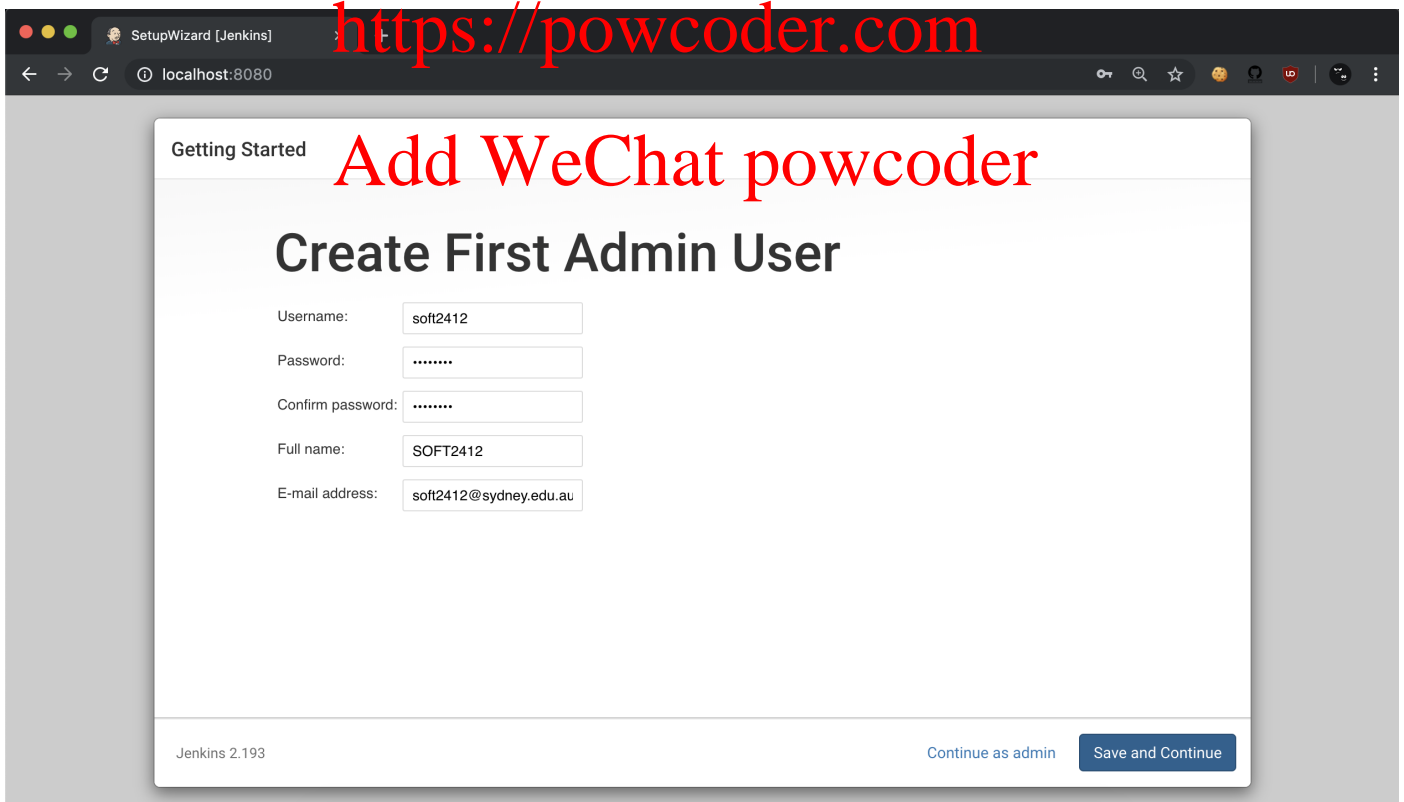


Add WeChat powcoder

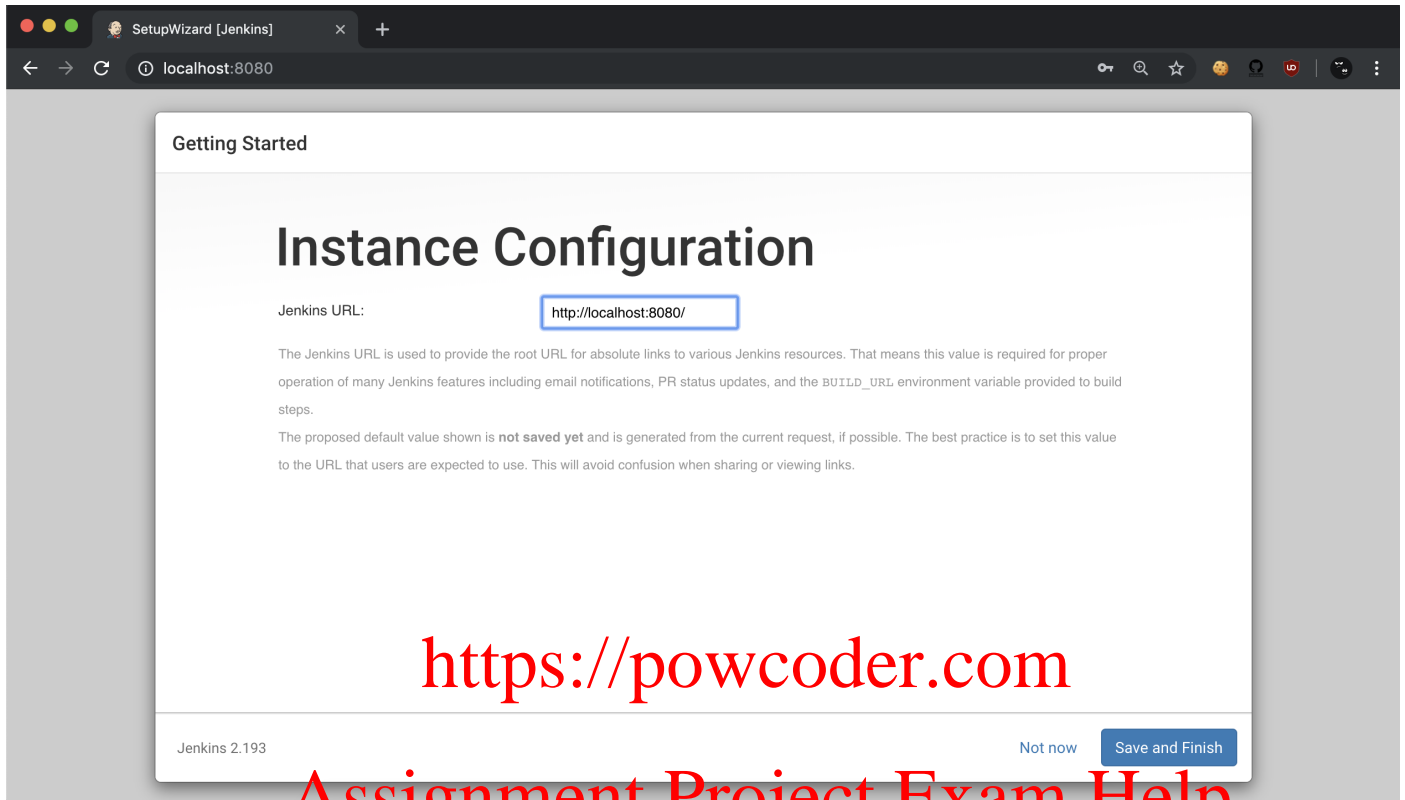
This will install packages. It may take some time



After the installation is completed you will be asked to create a admin user. Fill in the admin credentials and click "Save and continue"



Then you will be asked to set the url. Let it remain default:



<https://powcoder.com>

Assignment Project Exam Help

Click on "Save and Finish" and then click on "Start using Jenkins". You may have to restart Jenkins. You can now start using it.

Add WeChat powcoder

Accessing Jenkins

<https://powcoder.com>

**If you have installed Jenkins in your own machine:** You should visit [localhost:8080](http://localhost:8080) (<http://localhost:8080>).

Add WeChat powcoder

**If you are using the virtual machine:** You can access Jenkins in Firefox in the VM.

## Installing Ngrok

Since Jenkins is running on localhost at port 8080, we need to expose it on the internet in order to allow collaboration. We will use a free software called ngrok that provides this.

You can download ngrok from here: <https://ngrok.com/download> (<https://ngrok.com/download>). Follow the steps there to unzip ngrok. Then, cd to the directory containing the ngrok program and run the following command:

```
ngrok http 8080
```

Output:

```
@inconsreveable                               (Ctrl+C to quit)

Session Status online
Account <Name>
Version 2.3.34
Region United States (us)
```

```
Web Interface http://127.0.0.1:4040
Forwarding http://f9e633c8.ngrok.io -> http://localhost:8080
Forwarding https://f9e633c8.ngrok.io -> http://localhost:8080

Connections ttl opn rt1 rt5 p50 p90
0 0 0.00 0.00 0.00 0.00
```

You should get a public IP address that expires in 8 hours. **You could create an account in ngrok's website to extend that time, however you can obtain a new domain by typing the above command again after the current session expires.**

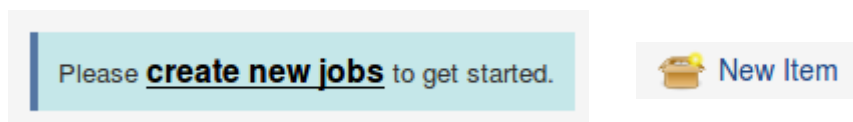
**To create an ngrok account for free sign up [here](https://dashboard.ngrok.com/signup) (<https://dashboard.ngrok.com/signup>) and then follow the instructions in the ngrok dashboard.**

The image here shows the obtained the URL <https://f9e633c8.ngrok.io> (<https://f9e633c8.ngrok.io>). You will obtain a different one. Keep a note of this URL as you will require it later. Make sure you take a note of the "https" one and not the "http" one for obvious reasons.

## Integrate Jenkins with Github using Web hooks

Integrating Jenkins with Github provides developers with an automated checking of new code builds. It helps to determine if new code builds are a success or a failure. This ensures that erroneous codes is identified when committed to the final or master repository where it will be used for eventual product or application development.

To start using Jenkins, we need to create a new job or item. This can be done by clicking on one of the following links shown below.



Fill in the item name as *jenkins-github* and select *Freestyle project*. Click *Ok* to proceed.

New Item [Jenkins] x

Not secure | 192.168.9.1/view/all/newJob

Jenkins

2 search soft2412 | log out

Enter an item name

jenkins-github

» 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.

**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.

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

**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**GitHub Organization**  
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

**Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

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

Copy from: Type to autocomplete

OK

Under General, you may want to fill in the description to provide more detail about the item that has been created. Although this is optional. What is required is to fill in the Source Code Management. Select Git.

jenkins-github Config [Je x]

Not secure | 192.168.9.1/job/jenkins-github/configure

Jenkins » jenkins-github »

**General** Source Code Management Build Triggers Build Environment Build Post-build Actions

Description Jenkins Github Integration

[Plain text] Preview

- ☐ Commit agent's Docker container
- ☐ Define a Docker template
- ☐ Discard old builds
- ☐ GitHub project
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary

Advanced...

**Source Code Management**

- ☐ None
- ☒ Git

Repositories

Repository URL

Repository

Branches to build

Branch Specifier (blank for 'any') \*/\*master

Save Apply

<https://powcoder.com>

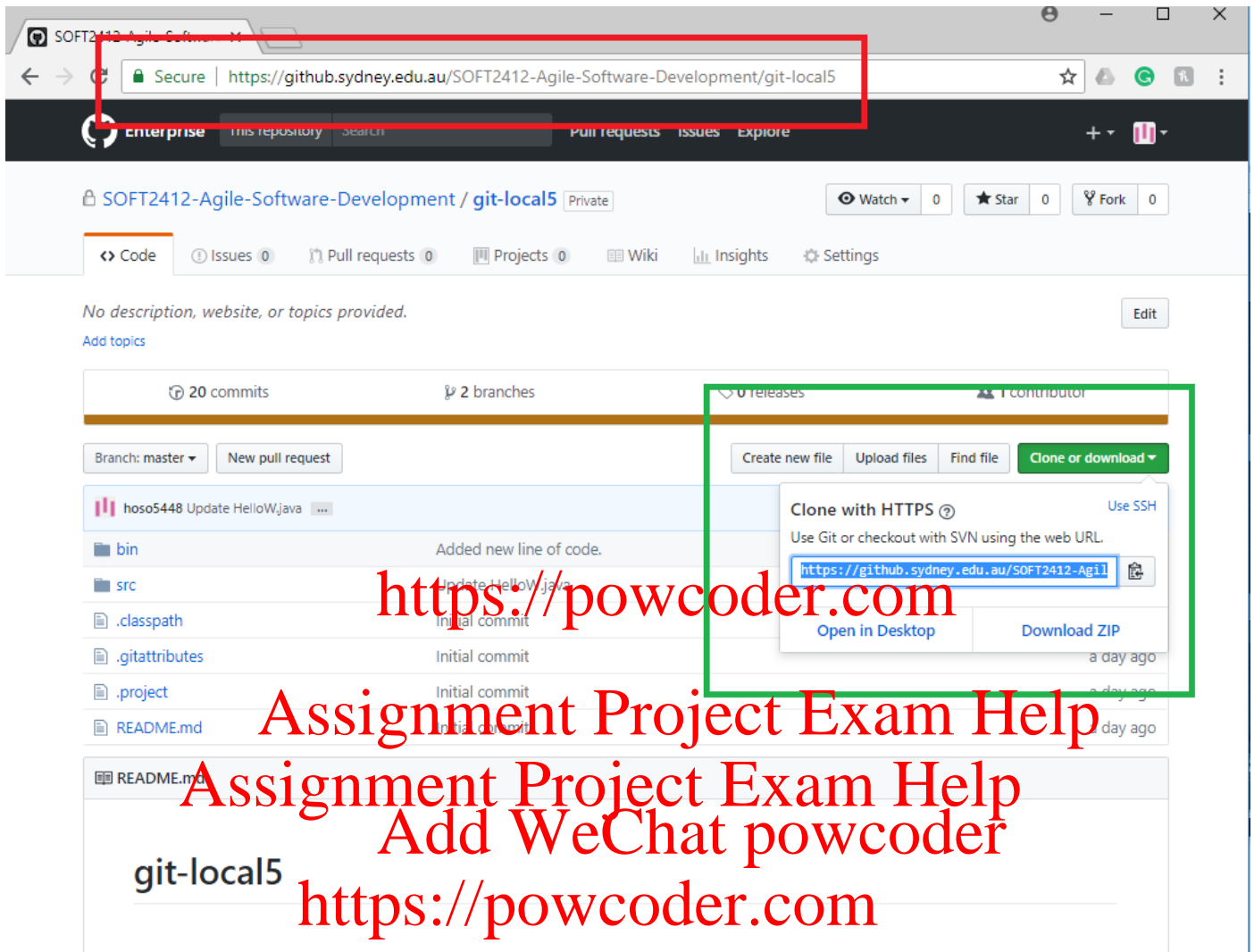
Assignment Project Exam Help

Add WeChat powcoder

<https://powcoder.com>

Add WeChat powcoder

Indicate your repository URL. To get your working git repository URL, you need to switch to github.sydney.edu.au and select the repository that you will work on.

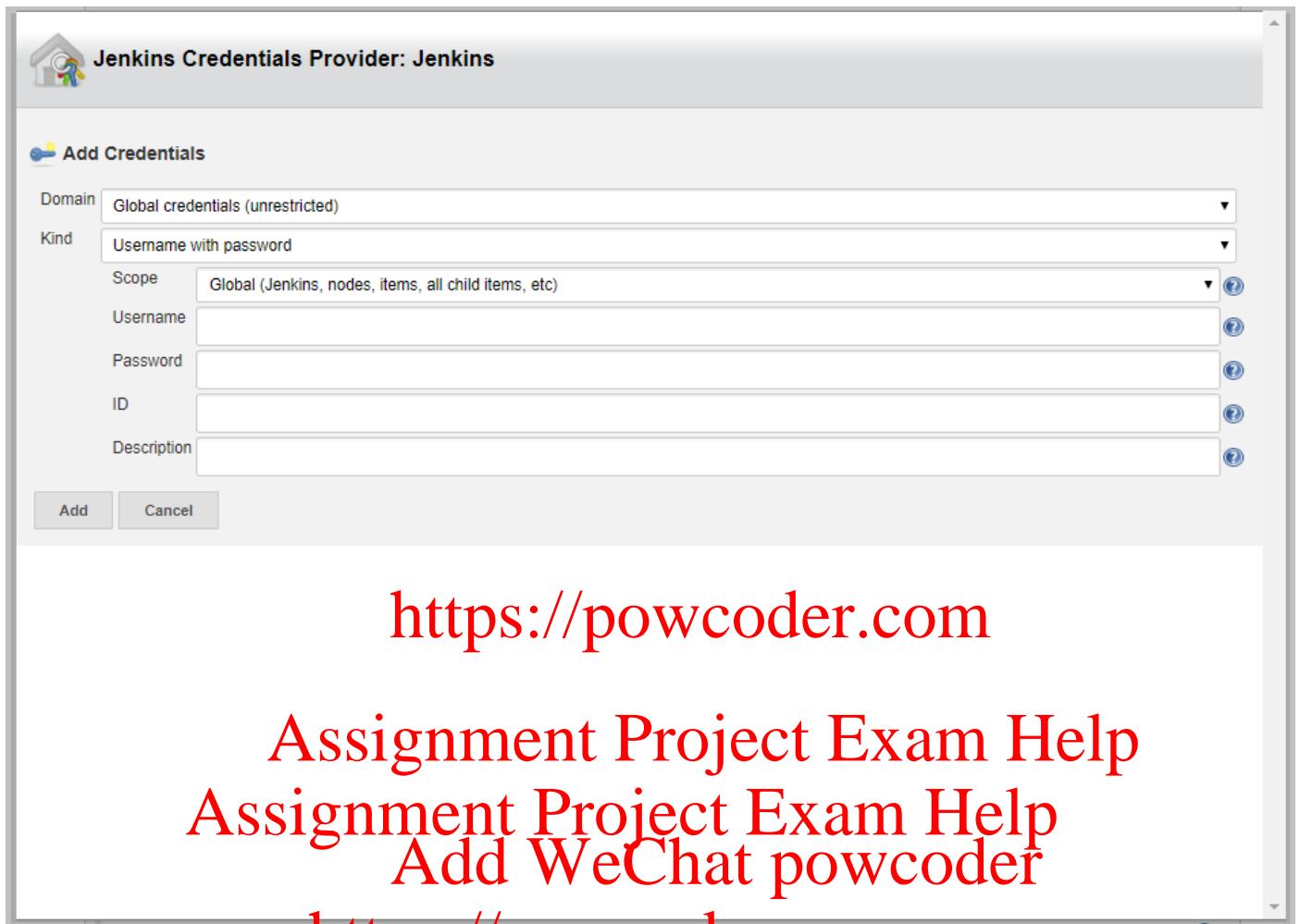


## Add WeChat powcoder

The URL is not what is displayed on the browser (red box). Instead, search for the *Clone or download* button and click on it. Copy the URL provided. This is the URL which references the repository URL (green box). Paste this URL back to the *Jenkins Source Code Management*.

Since we are using github.sydney.edu.au, you need to supply your Unikey credentials to allow Jenkins access to the repository.

Click on *Add* to add new credentials. Select *Domain* as *Global credentials*. Select *Kind* to be *Username with password*. Select *Scope* to be *Global*. Then indicate your Unikey under *Username* and your Unikey password under **Password**. Next click on *Add*.



**Jenkins Credentials Provider: Jenkins**

**Add Credentials**

Domain: Global credentials (unrestricted)

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username:

Password:

ID:

Description:

<https://powcoder.com>

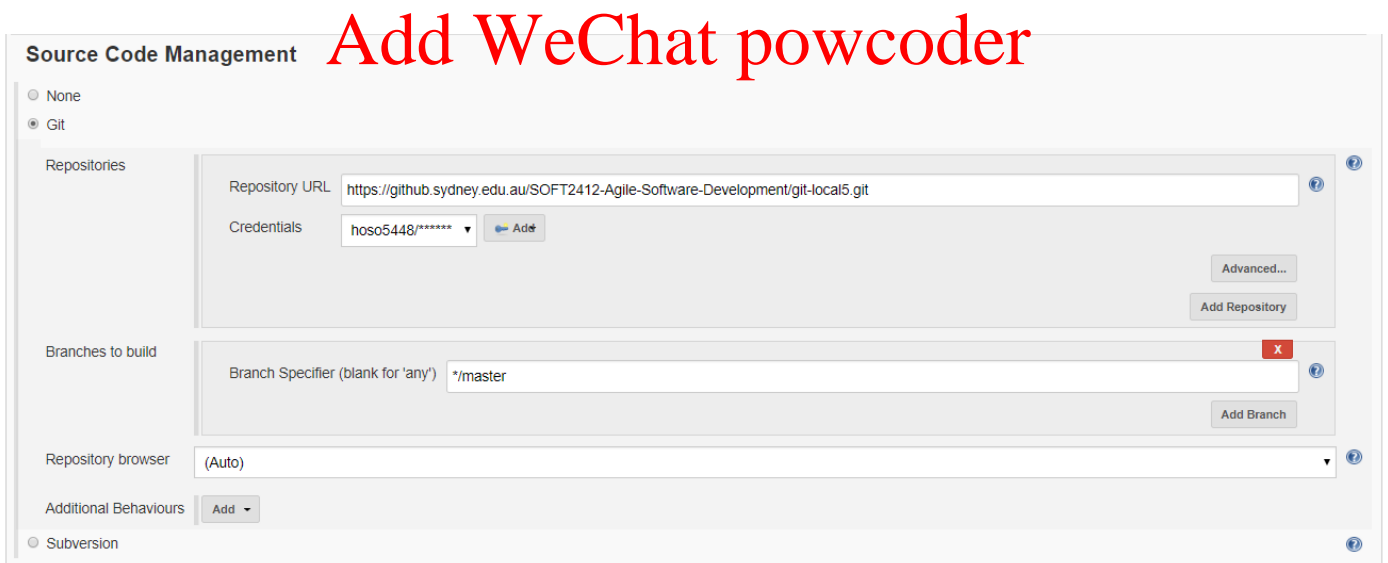
Assignment Project Exam Help

Assignment Project Exam Help

Add WeChat powcoder

<https://powcoder.com>

You should not be getting any more error messages regarding access to the git repository.



**Source Code Management**

☐ None

☒ Git

Repositories

Repository URL:

Credentials:

Branches to build

Branch Specifier (blank for 'any'):

Repository browser:

Additional Behaviours:

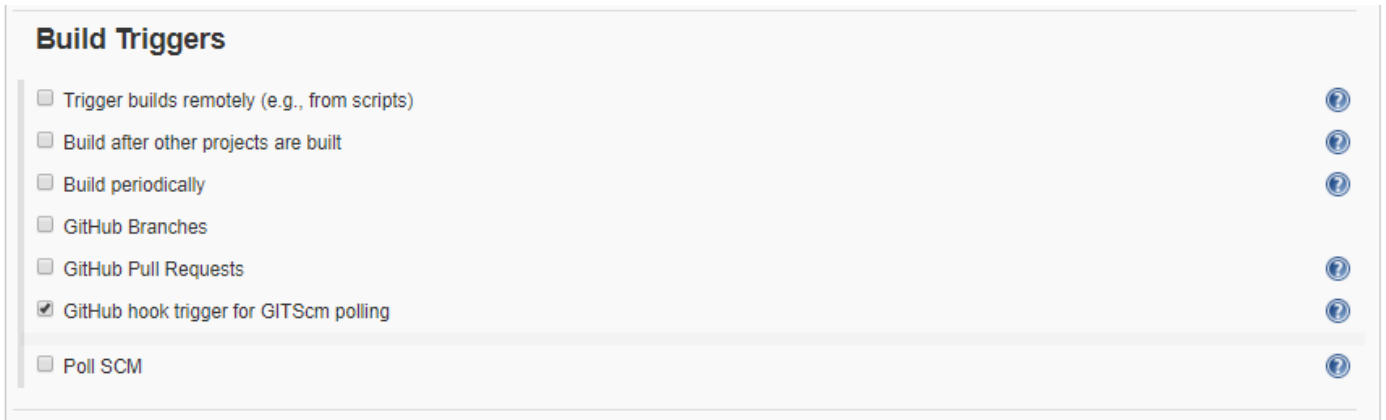
☐ Subversion

<https://powcoder.com>

Add WeChat powcoder

Since we are utilising Jenkins to check for build errors, we select *GitHub hook trigger for GITScm polling*. This feature triggers build in Jenkins when a git event is performed, for example, *push*.





Next click on Save.

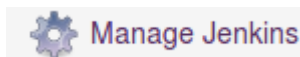
If you return to your dashboard, you should be able to see the github-jenkins item added as shown below.

https://powcoder.com

S	W	Name ↓	Last Success	Last Failure	Last Duration
		github-jenkins	N/A	N/A	N/A

To enable Jenkins to receive triggers from GitHub, you will have to set some plugins in Jenkins.

Click on *Manage Jenkins*.



https://powcoder.com

Click on *Manage Plugins*.



### Manage Plugins

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

There are updates available

Add WeChat powcoder

Click on the *Available* tab.

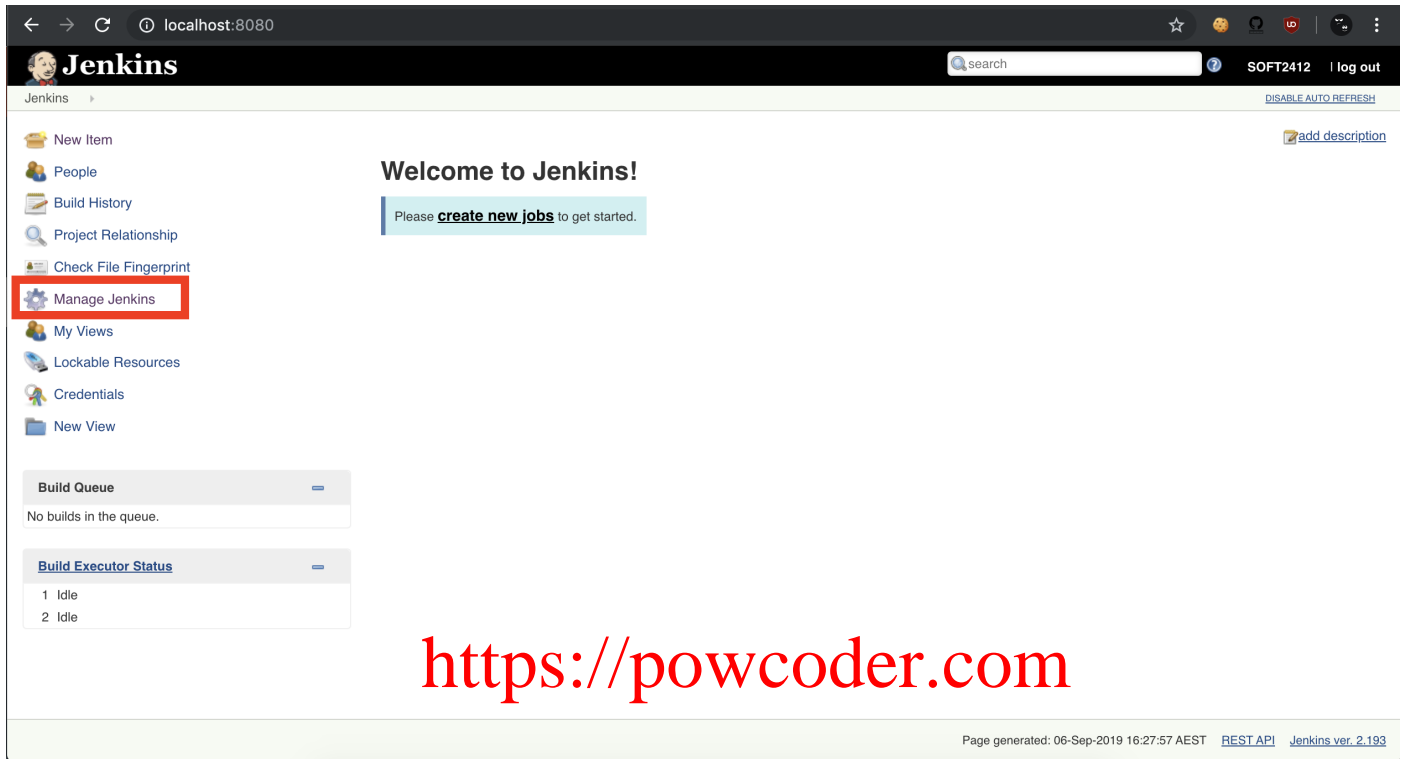


Using the Filter option, key in 'GitHub'. This will display a list of plugins but we are only interested on the *GitHub Integration Plugin*.

Check this plugin and click on *Download now and install after the restart*. Jenkins will install the plugin and restarts the server. You may be required to log in again.

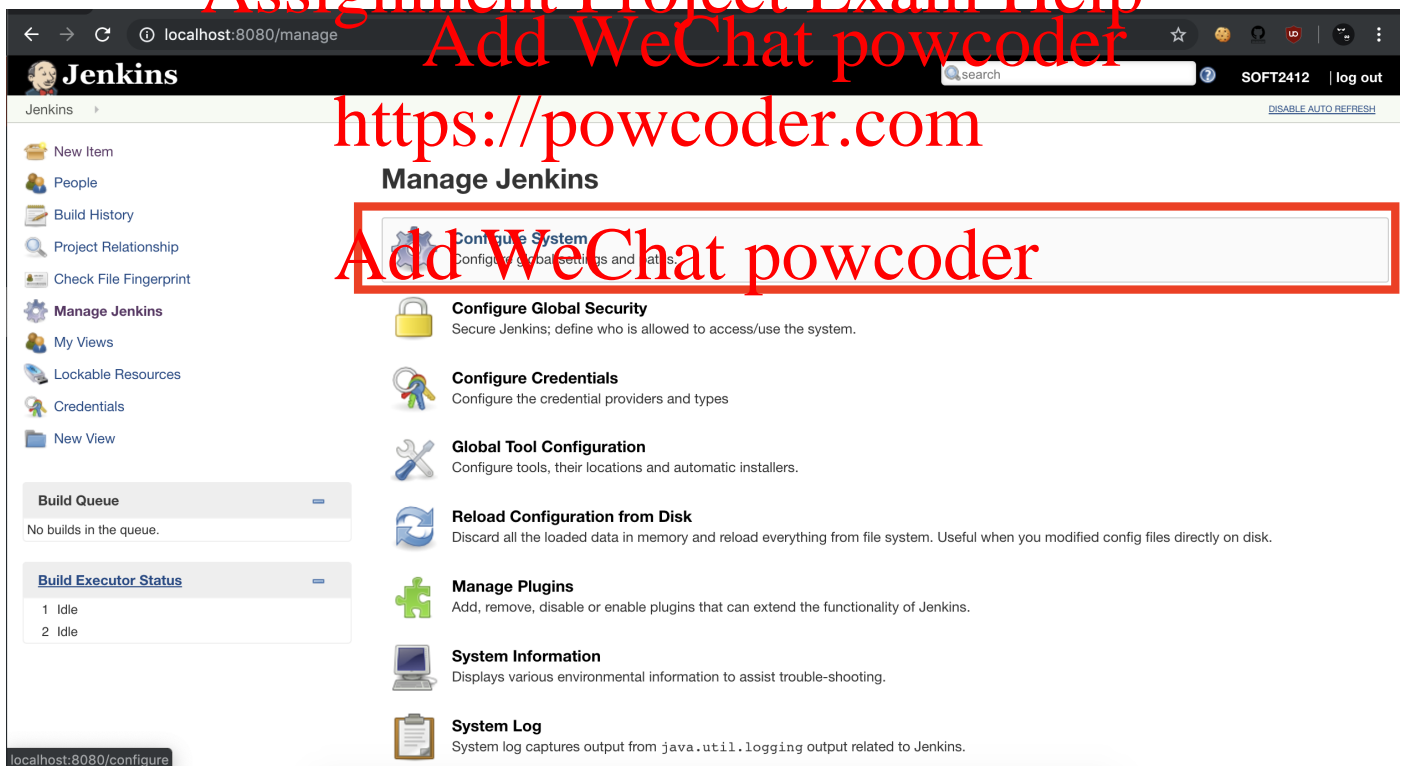
**(THE BELOW IS NOT NEEDED FOR NEW JENKINS VERSIONS, SKIP UNTIL YOU SEE "-----")**

**If you are installing jenkins in your own machines:** Make sure you add the Github enterprise in jenkins. We need to tell jenkins that a github server that we will be pushing code into is the enterprise. Goto manage jenkins in the homepage:



<https://powcoder.com>

Then click on "Configure Jenkins".



<https://powcoder.com>

Add WeChat powcoder

Scroll down to "GitHub enterprise servers" and click "Add"

Administrative monitors...

**Jenkins Location**

Jenkins URL:  ?  
 ⚠ Please set a valid host name, instead of localhost

System Admin e-mail address:  ?

**Lockable Resources Manager**

Lockable Resources:

**GitHub**

GitHub Servers:  ?

**GitHub Enterprise Servers** (highlighted with a red box)

**Pipeline Model Definition**

Docker Label:

Docker registry URL:  ?

Registry credentials:

**Global Pipeline Libraries**

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be trusted, meaning they run without "sandbox" restrictions and may use @Grab.

**Build-timeout Plugin > BuildStep Action**

<https://powcoder.com>

Assignment Project Exam Help

Then add the GitHub Sydney url.

Add WeChat powcoder

GitHub Enterprise Servers

<https://powcoder.com>

Add WeChat powcoder

API endpoint:  ?

Name:

Apply and Save the changes.

(-----)

Next in GitHub

While waiting for Jenkins to restart, you will also need to configure your GitHub settings so that it can work with Jenkins.

Use ngrok ip address that you kept note of from above, return to your working GitHub repository.

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Insights **Settings**

Click on the Settings tab from your working repository.

Options
Collaborators & teams
Branches
<b>Hooks</b>
Integrations & services
Deploy keys
Custom tabs

On the left panel, click on Hooks. This will create a service which notifies Jenkins when certain git events happen. Next, click on Add webhook.

Add webhook

<https://powcoder.com>

You will be presented with a form. With the machine's URL, paste it on the Payload URL. This is the URL which will be triggered by Github after a push event is created to the repository.

Update your noted URL to:

<https://<domain-url>/github-webhook/>. In this case, it would look like:

<https://9387d23f.ngrok.io/github-webhook/> [\(https://soft2412:soft2412@9387d23f.ngrok.io/\)](https://soft2412:soft2412@9387d23f.ngrok.io/).

Add this URL into the “payload URL” input field.

Add WeChat powcoder

Webhooks / Manage webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL \*

<https://94e5c36a.ngrok.io/github-webhook/>

Content type

application/json

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

☒ Enable SSL verification ☐ Disable (not recommended)

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

If the URL is correctly configured, there will be a green tick beside it. If there is no green tick, refresh the page several times or restart your virtual machine to restart Jenkins.

## Webhooks

[Add webhook](#)

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

✓ <https://94e5c36a.ngrok.io/github-webhook/> (push)

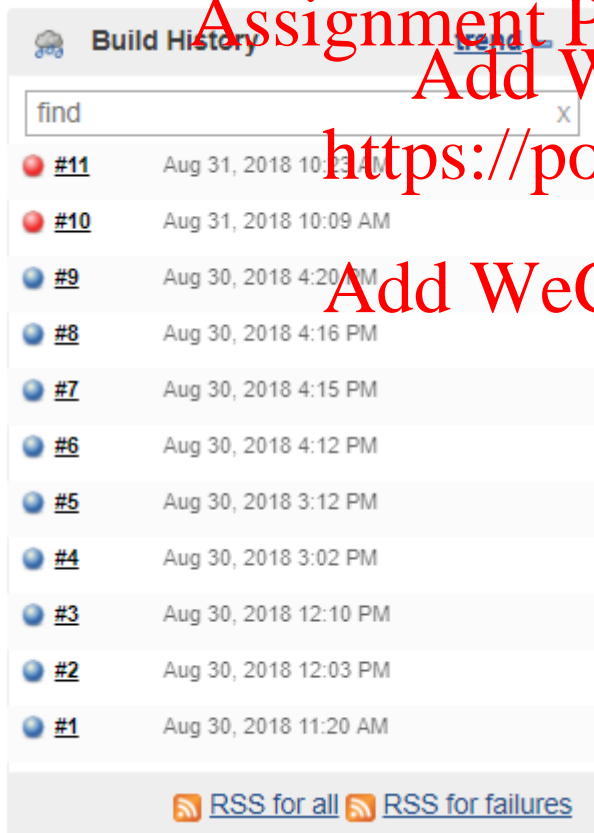
[Edit](#)[Delete](#)

If a green tick is present, it means that the URL is accessible and we can start using GitHub and Jenkins integration.

To test if this integration works, we can make a few code changes on git repository and trigger a webhook to Jenkins.

Return to your own code repository. If you do not have any files in your repository, add a new text file. In that file, add the following comment, 'Sending trigger to Jenkins from GitHub'.

Return to your Jenkins Dashboard and monitor the build history for the newly created Jenkins item. If your build is successful, the circle icon will turn to blue.



Click on the build number to see the build information, like the one shown below.

## Build #9 (Aug 30, 2018 4:20:26 PM)



### Changes

1. Added new line of code. ([detail](#))
2. Commit changes from repo ([detail](#))



[Started by GitHub push by hoso5448](#)



**Revision:** 1a754dbc0525a5f53cdc7469938e3d4896ad8ba8

- refs/remotes/origin/master

<https://powcoder.com>

You can also see the build information from Console Output as shown below.



### Console Output

```
Started by GitHub push by hoso5448
Building in workspace /var/lib/jenkins/workspace/webhook-git-local5
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.sydney.edu.au/SOFT2412-Agile-Software-Development/git-local5.git # timeout=10
Fetching upstream changes from https://github.sydney.edu.au/SOFT2412-Agile-Software-Development/git-local5.git
> git --version # timeout=10
using GIT_ASKPASS to set credentials
> git fetch --tags --progress https://github.sydney.edu.au/SOFT2412-Agile-Software-Development/git-local5.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 1a754dbc0525a5f53cdc7469938e3d4896ad8ba8 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1a754dbc0525a5f53cdc7469938e3d4896ad8ba8
Commit message: "Merge branch 'local1'"
> git rev-list --no-walk dc061df515fcea6f45adc4f5e2b685abc06dfa0f # timeout=10
Finished: SUCCESS
```

## Continuous Integration using Jenkins

### Integrating GitHub -> Jenkins -> Gradle

You can also automate your code build by integrating Gradle to Jenkins after every new code push to GitHub. To enable this, click on *Manage Jenkins* followed by *Global Tool Configuration*.

Next, under *Gradle Installations*, click *Add Gradle*. Input Gradle name as *gradle\_build* and check *Install automatically*. Click *Save*.

Gradle

Gradle Installations

Add Gradle

Gradle

name

**Required**

☒ Install automatically

Install from Gradle.org

Version

Add Installer

Delete Installer

Delete Gradle

Next, you will need to invoke this `gradle_build`. To do that, click on *github-jenkins* from the dashboard. Then click on Configure. This will bring up the settings for the *github-jenkins* item. Click on the Build tab. Click on Add build step and select Invoke Gradle script as shown below.

Build

Add build step

- Add a new template to all docker clouds
- Build / Publish Docker Image
- Execute Windows batch command
- Execute shell
- GitHub PR: set 'pending' status
- Invoke Ant
- Invoke Gradle script**
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit
- Start/Stop Docker Containers

<https://powcoder.com>

Assignment Project Exam Help

Under Invoke Gradle script, check Invoke Gradle. Select the Gradle version name as the name you have created previously in Gradle installations. In this case, we indicated the name as *gradle\_build*. Then click Save.

Invoke Gradle script

☒ Invoke Gradle

Gradle Version

☐ Use Gradle Wrapper

Tasks

Advanced...

Add build step

Add WeChat powcoder

Add the appropriate Gradle tasks i.e. clean build and test in the input field in the above image.

You can also add Post-build Actions into this process by selecting the appropriate post-build actions

## Exercises

In this exercise, you will be tasked with creating a new repository in GitHub and a webhook that will trigger Jenkins and Gradle build. To assist you with this task, a breakdown of the items to do is provided below.

Task 1: (Important to stay on track with the assignment)

- Choose any existing repository you may have in your own github sydney account. Add your team members as collaborators in this repository.
- Push your JUnit test codes from last week's tutorial (the calculator program) to this new repository. The solutions to this must be available now.
- Integrate your GitHub repository with Jenkins and Gradle.
- Create a webhook to trigger Jenkins and Gradle.
- Test that this webhook is trigger correctly.
- Next, each member should include a new function in the code (one example below, find more examples like factorial, modulus, round etc.):

```
public int pow (int a, int b) {  
  
    return Math.pow(a,b);  
  
}
```

<https://powcoder.com>

- Assignment Project Exam Help  
Add WeChat powcoder  
<https://powcoder.com>  
Add WeChat powcoder
- One team member must have jenkins setup, with the webhook added in github.
  - Each member should build and commit this new source code to your repository.
  - The jenkins member setting up jenkins can see and provide a console report
  - Next, each member must create a new test case to test their newly added source code.
  - Each commit and build this new source code to your repository, jenkins must be triggered again.
  - The same member that setup jenkins, can see all the test cases and builds the program through Jenkins and Gradle and can provide a console report.
  - To see changes in the report, one member can try to fail a JUnit test and push the code, simultaneously triggering the webhook. There should be a red dot instead of a blue one in the jenkins build list.

#### Task 2:

- Try and integrate Jacoco with jenkins, such that every successful build will generate a Jacoco report.
- You may find this helpful: <https://dzone.com/articles/jacoco-jenkins-plugin>  
(<https://dzone.com/articles/jacoco-jenkins-plugin>)

#### Extensions:



Export a .jar file (artefact) from jenkins:

<https://wiki.jenkins.io/display/JENKINS/Copy+Artifact+Plugin>

(<https://wiki.jenkins.io/display/JENKINS/Copy+Artifact+Plugin>)

## Additional Resources

- <https://jenkins.io/solutions/github/>

Copyright © The University of Sydney. Unless otherwise indicated, 3rd party material has been reproduced and communicated to you by or on behalf of the University of Sydney in accordance with section 113P of the Copyright Act 1968 (Act). The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act. Do not remove this notice.

Live streamed classes in this unit may be recorded to enable students to review the content. If you have concerns about this, please visit our [student guide \(https://canvas.sydney.edu.au/courses/4901/pages/zoom\)](https://canvas.sydney.edu.au/courses/4901/pages/zoom) and contact the unit coordinator.

<https://powcoder.com>

Assignment Project Exam Help  
Assignment Project Exam Help  
Add WeChat powcoder

<https://powcoder.com>

Add WeChat powcoder