Jenkins

Continuous Integration :-

Importance of Continuous Integration :-

1.**Improves Quality :-**

- Improves the quality by running multiple unit tests and analysing various static code

2.**Increases Productivity :-**

- Automating build of code saves a lot of time, there by increasing productivity

3.**Reduces Risk :-**

- Elimanate the risk of Potential human errors by automating test

Introduction to Jenkins

Features of Jenkins :-

- Easy Installation process
- Provides advance security
- Optimized performance
- Upgrades are easily available
- Light weight containers support
- Distributed team management

What is continuous Integration ?

- It is the process of automating the building and testing of code , each time one of the team member commits changes to version control

CI&CD → Continuous Deployment & also Delivery
Continous Integration

Popular Continuous Integration Tools :-

- Gitlab CI
- Code ship
- Bamboo
- Jenkins
- Team city
- Travis CI

In AWS Cloud perform CI&CD there are various services like code commit , code display , code build , code pipeline , code deploy & code guru

From AZURE Cloud we have AZURE DevOps :-

Azure Boards , Azure Repos , Azure Piplines , Azure testplans , Azure artifacts

What is Jenkins ?

- A Continous Integration Server which manages and control process such as plan , code , build , test , deploy , operate and monitor in DevOps Environment .

Jenkins :-

Plan , Code , Build , Test , Deploy , Operate , Monitor

Why Jenkins is So Popular ?

Jenkins :-

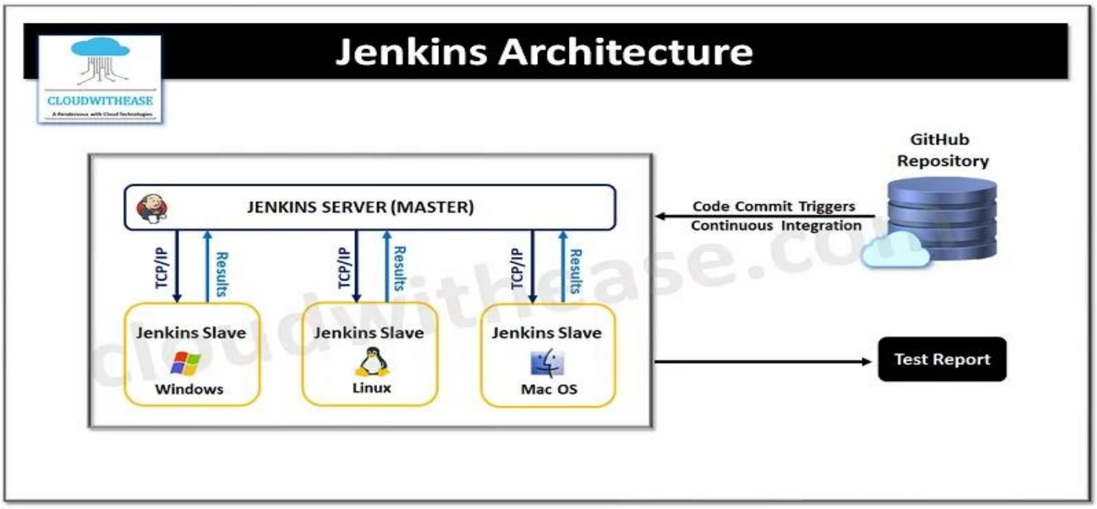
- Open Source
- Good Plugin support
- Good community support
- Fast and Reliable
- Good OS support
- Scripted Builds

Topics :-

- Jenkins Architecture
- Plugin Management in Jenkins
- Jenkins security Management
- Notification in Jenkins
- Jenkins Master slave Architecture
- Jenkins Delivery Pipeline
- Jenkins Declarative Pipeline

Jenkins Architecture :-

Source Control Management :-



Tabs :-

Update :-

- Shows updates to plugins already installed

Available :-

- Shows Plugins that are available for installation

Installed :-

- Displays Plugins Installed that have no updates

Advanced :-

- Lists Configuration of Http Proxy , allows manual upload of plugin and URL of Plugin site

In real time we can’t install the plugins as simple as reason being the might be Challengs with proxy settings as well as vpn in order to avoid this we should configuration , http proxiy in Jankins , Plugin of advance tab

How To Install Jenkins On Windows ?

- **Installing Jenkins**
    - Docker
    - Kubernetes
    - Linux
    - macOS
    - **Windows**
    - Other Systems
    - WAR file

  - Other Servlet Containers
  - Offline Installations
  - Initial Settings

Prerequisites

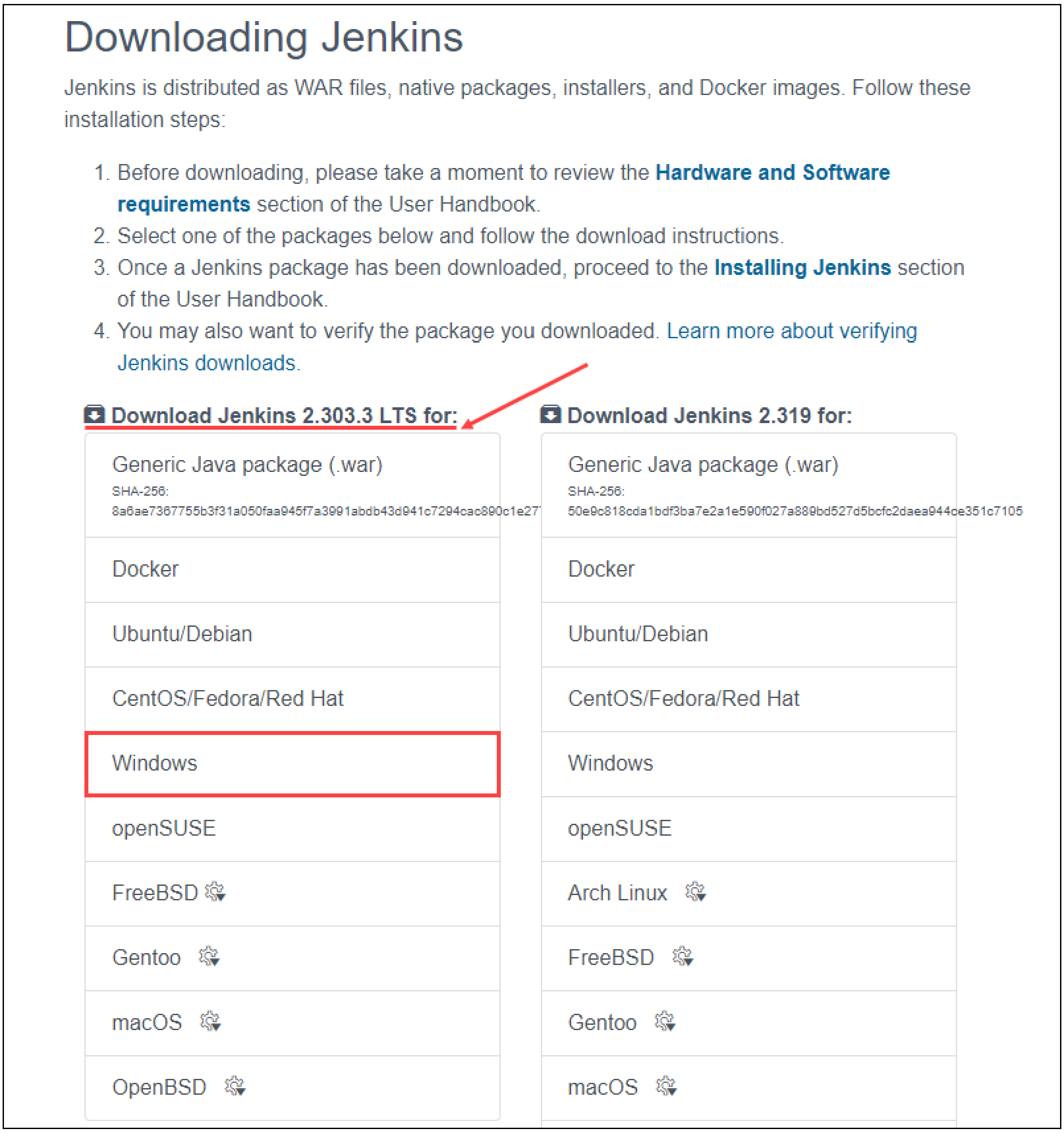
- 256 MB of RAM
- 1 GB of drive space (although 10 GB is a recommended minimum if running Jenkins as a Docker container)
- 4 GB+ of RAM
- 50 GB+ of drive space

Jenkins Tool is developed on Javacode

Prerequisites

- A system running Windows 10
- The latest copy of Java Development Kit or Java Runtime Environment installed
- Access to an account with administrator privileges

Browse to the **official Jenkins download page**. Under the **Downloading Jenkins** section is a list of installers for the long-term support (LTS) version of Jenkins. Click the **Windows** link to begin the download.



Downloading Jenkins

Jenkins is distributed as WAR files, native packages, installers, and Docker images. Follow these installation steps:

1. Before downloading, please take a moment to review the **Hardware and Software requirements** section of the User Handbook.
2. Select one of the packages below and follow the download instructions.
3. Once a Jenkins package has been downloaded, proceed to the **Installing Jenkins** section of the User Handbook.
4. You may also want to verify the package you downloaded. Learn more about verifying Jenkins downloads.

Download Jenkins 2.303.3 LTS for:

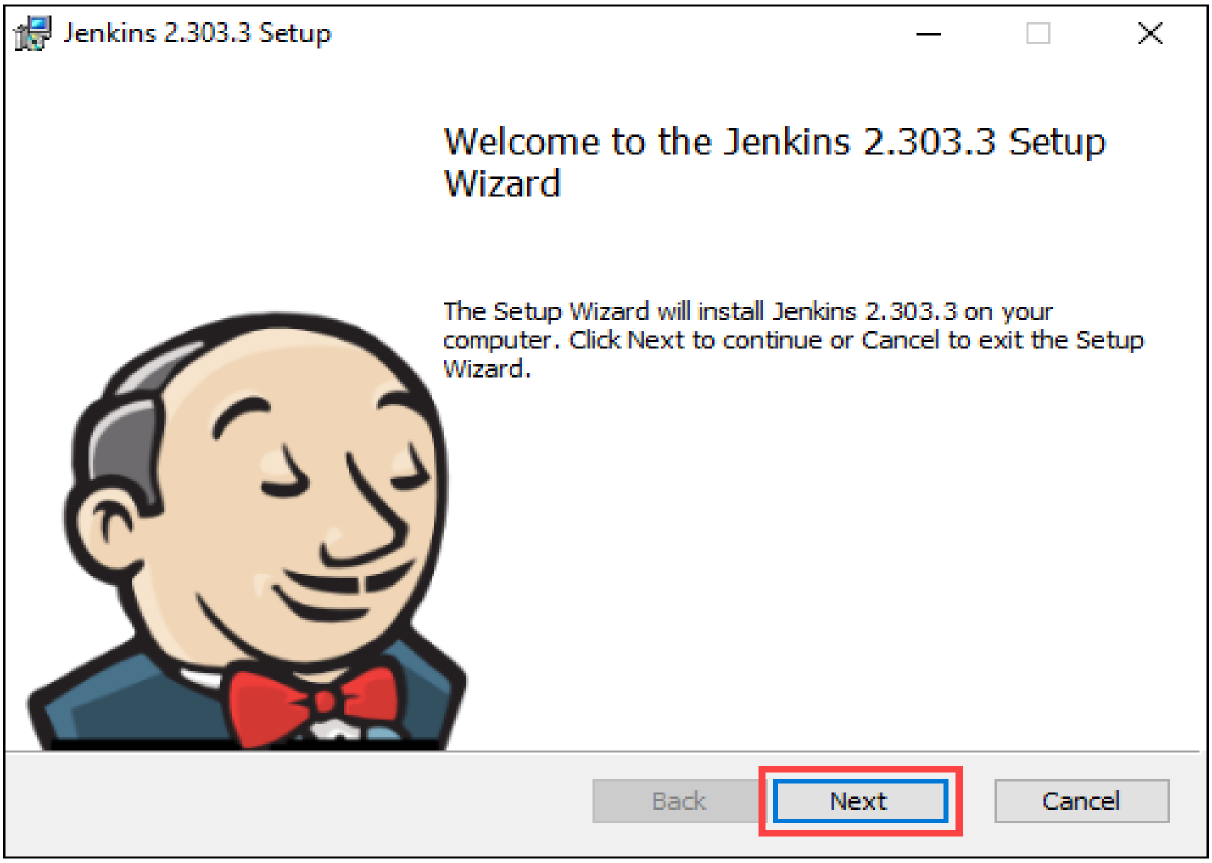
- Generic Java package (.war)
  SHA-256:
  8a6ae7367755b3f31a050faa945f7a3991abdb43d941c7294cac890c1e27
- Docker
- Ubuntu/Debian
- CentOS/Fedora/Red Hat
- Windows
- openSUSE
- FreeBSD
- Gentoo
- macOS
- OpenBSD

Download Jenkins 2.319 for:

- Generic Java package (.war)
  SHA-256:
  50e9c818cda1bdf3ba7e2a1e590f027a889bd527d5bcfc2daea944ce351c7105
- Docker
- Ubuntu/Debian
- CentOS/Fedora/Red Hat
- Windows
- openSUSE
- Arch Linux
- FreeBSD
- Gentoo
- macOS

2. Once the download is complete, run the **jenkins.msi** installation file.

3. The setup wizard starts. Click **Next** to proceed.

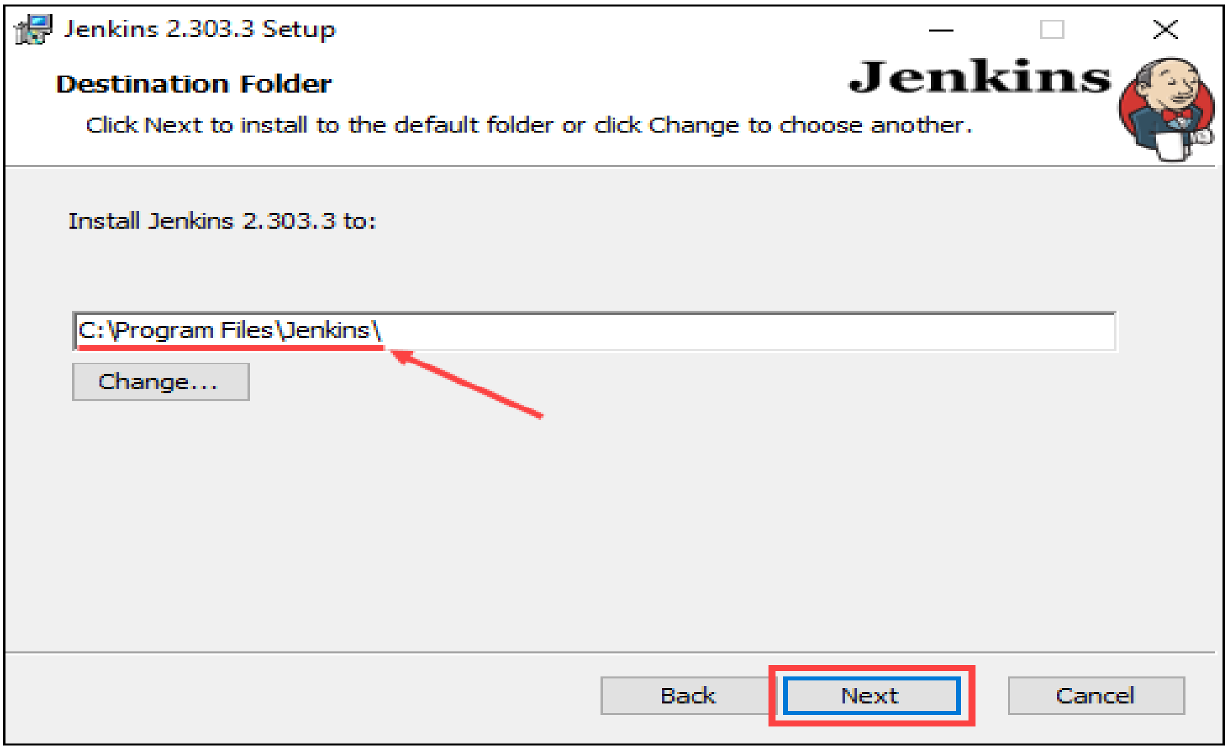


Jenkins 2.303.3 Setup

Welcome to the Jenkins 2.303.3 Setup Wizard

The Setup Wizard will install Jenkins 2.303.3 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

4. Select the install destination folder and click **Next** to continue.



Jenkins 2.303.3 Setup

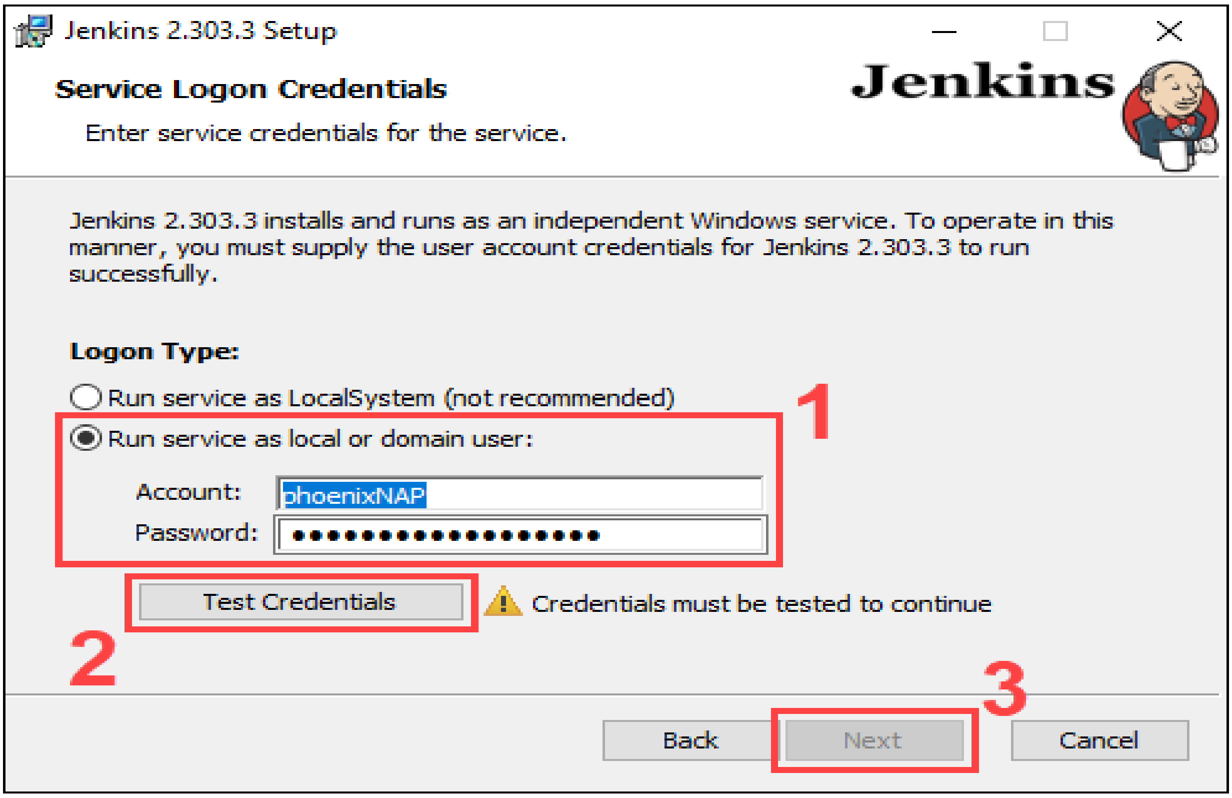
Destination Folder

Click Next to install to the default folder or click Change to choose another.

Install Jenkins 2.303.3 to:

C:\Program Files\Jenkins\

5. Under the **Run service as a local or domain user** option, enter the **domain** username and password for the user account you want to run Jenkins with. Click **Test Credentials** to verify the login data, then click **Next** to proceed.



Jenkins 2.303.3 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.303.3 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.303.3 to run successfully.

Logon Type:

○ Run service as LocalSystem (not recommended)

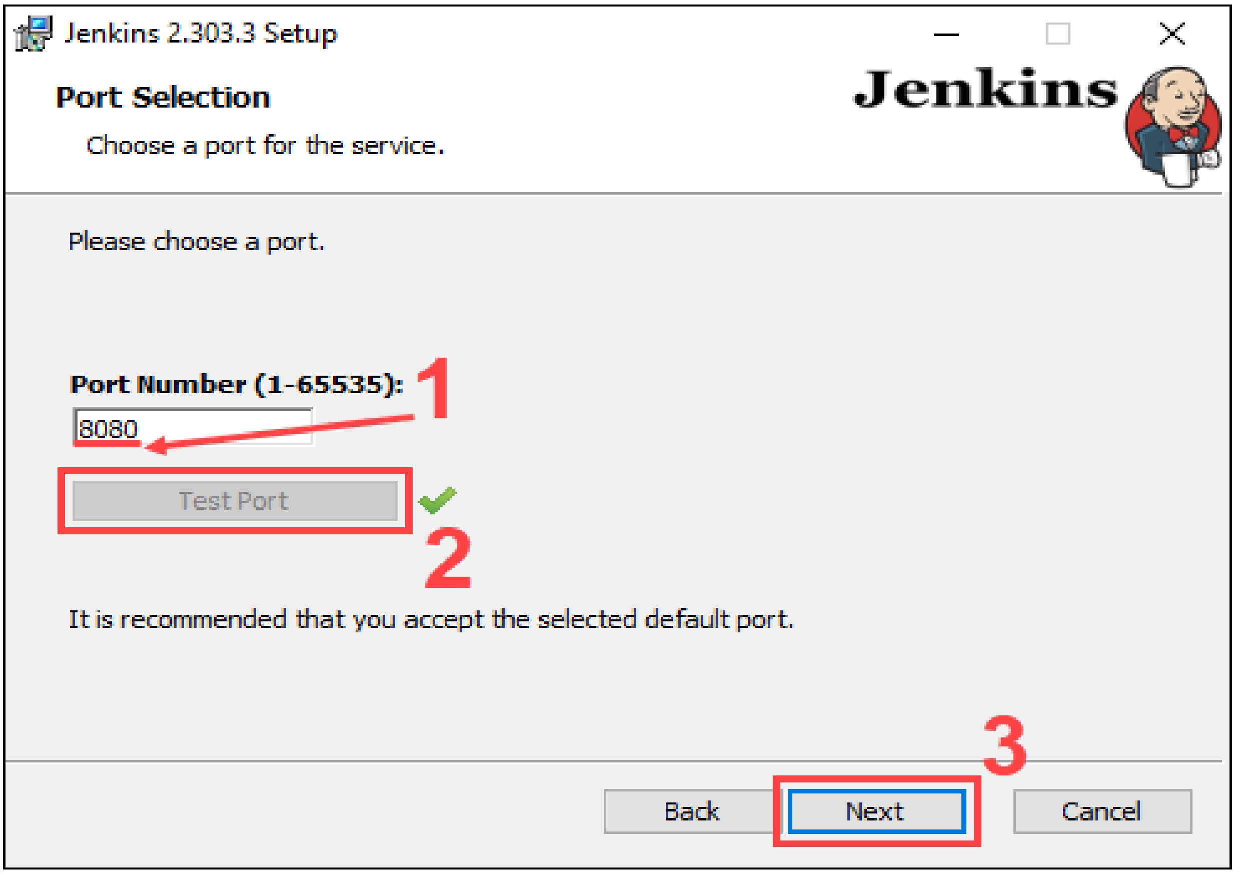
● Run service as local or domain user:

Account: phoenixNAP

Password: ••••••••••••••••••

Credentials must be tested to continue

6. Enter the port number you want Jenkins to run on. Click **Test Port** to check if the selected port is available, then click **Next** to continue.



Jenkins 2.303.3 Setup

Port Selection

Choose a port for the service.

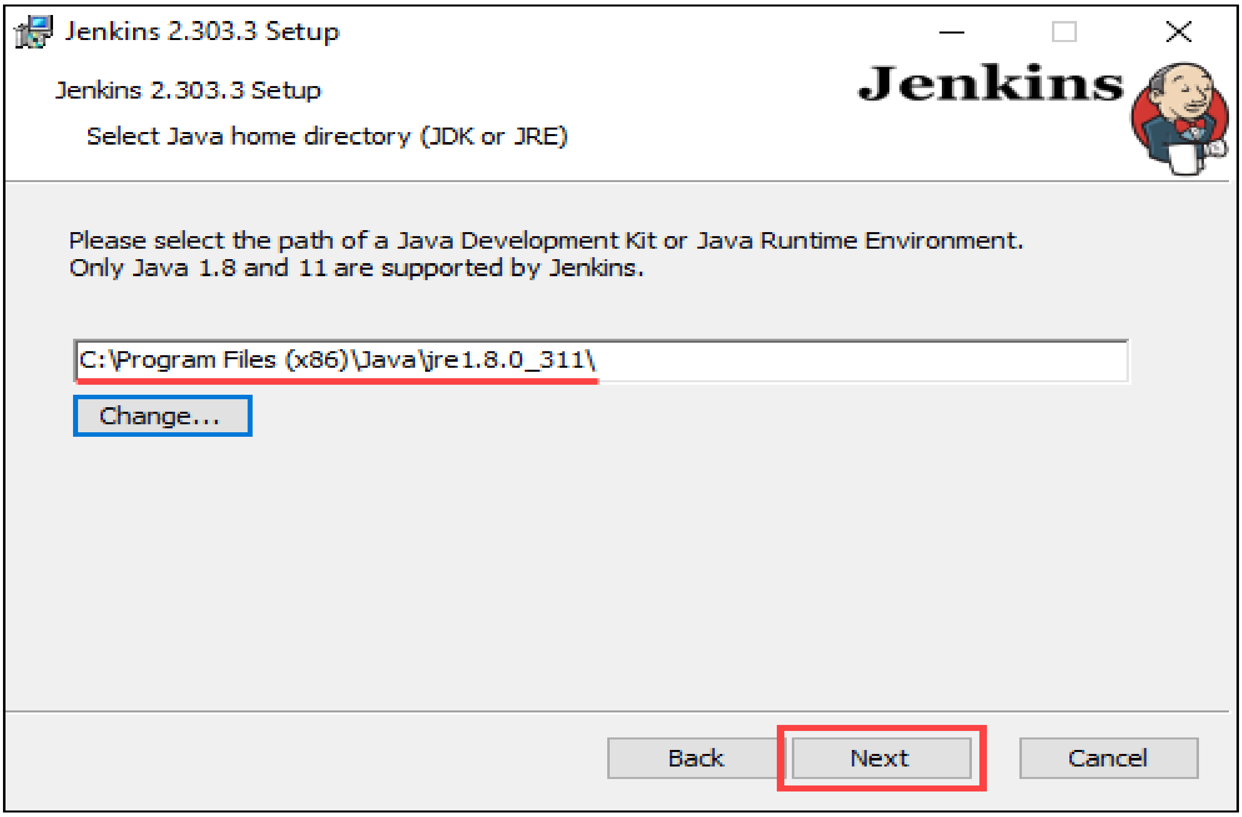
Please choose a port.

Port Number (1-65535): 8080

Test Port ✓

It is recommended that you accept the selected default port.

7. Select the directory where **Java is installed** on your system and click **Next** to proceed.



Jenkins 2.303.3 Setup

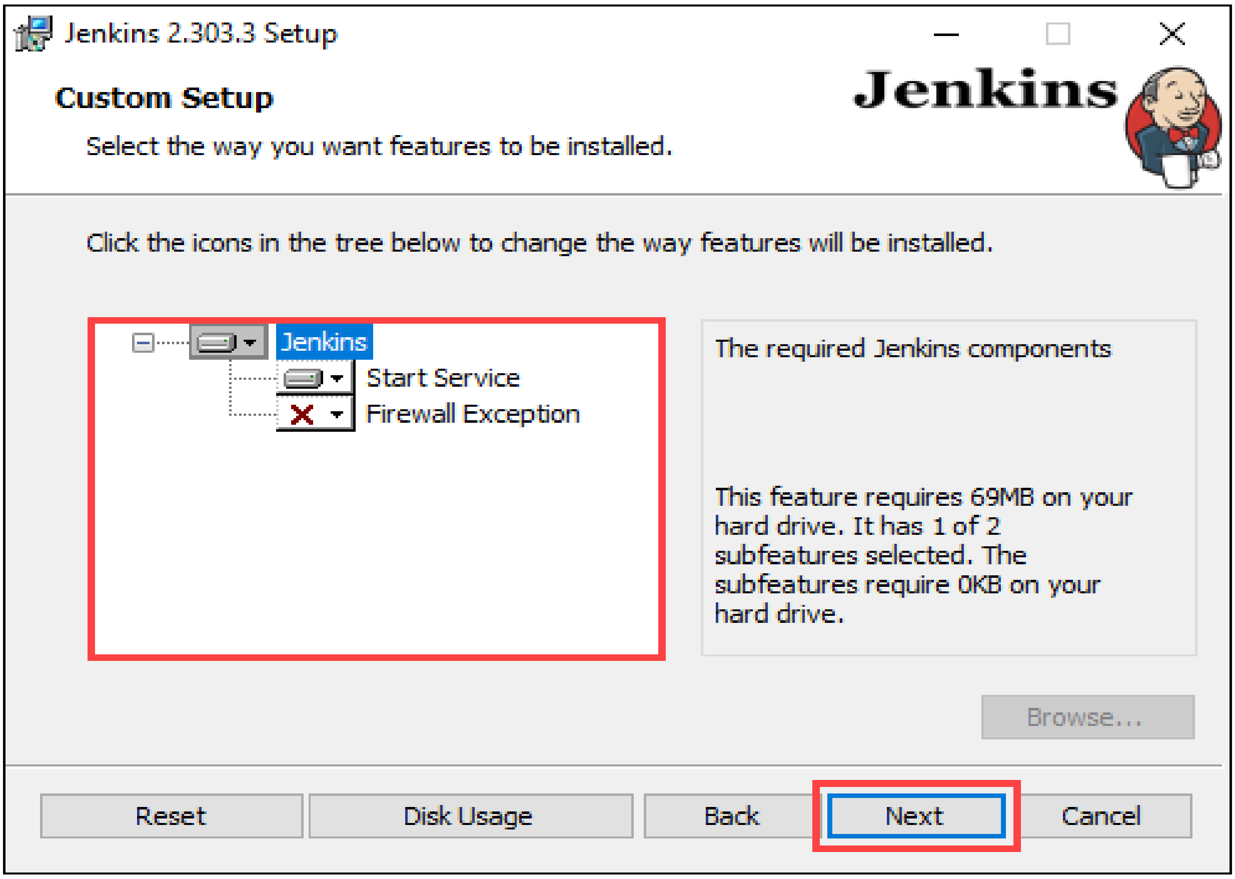
Jenkins 2.303.3 Setup

Select Java home directory (JDK or JRE)

Please select the path of a Java Development Kit or Java Runtime Environment.
Only Java 1.8 and 11 are supported by Jenkins.

C:\Program Files (x86)\Java\jre1.8.0_311\

8. Select the features you want to install with Jenkins and click **Next** to continue.



Jenkins 2.303.3 Setup

Custom Setup

Select the way you want features to be installed.

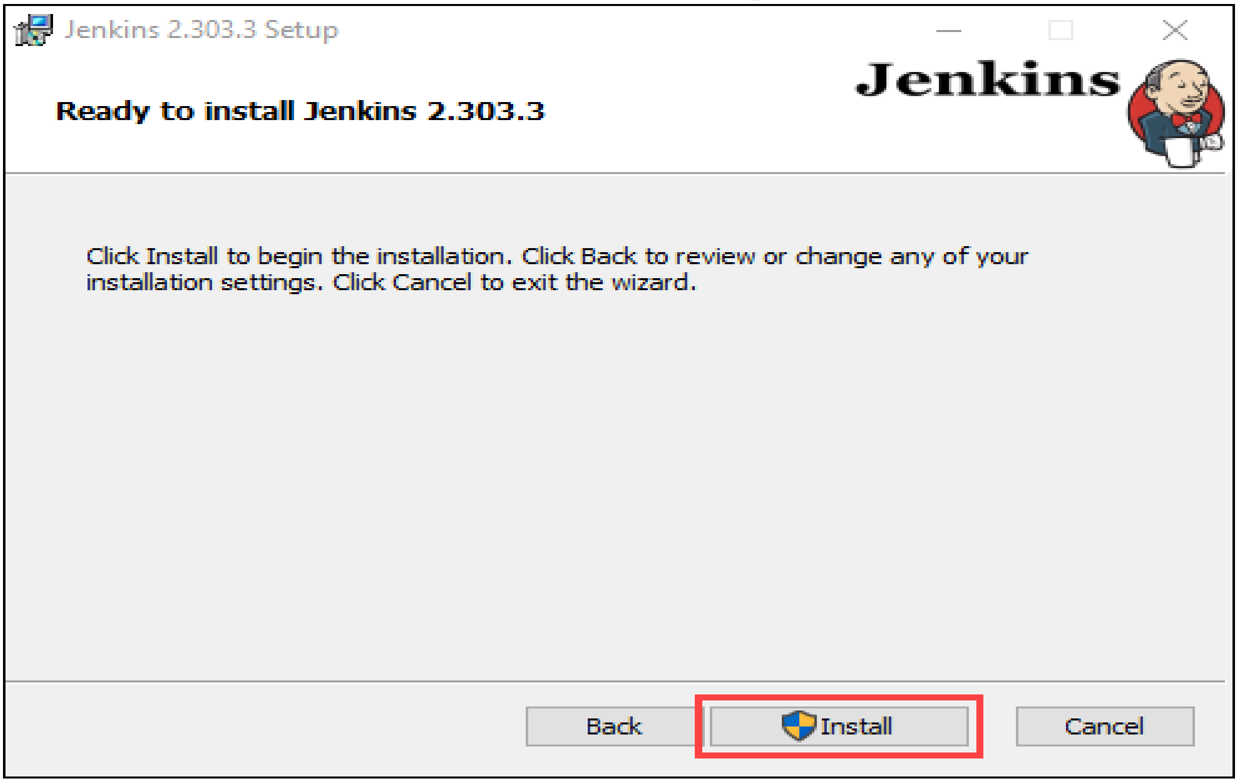
Click the icons in the tree below to change the way features will be installed.

- Jenkins
  - Start Service
  - Firewall Exception

The required Jenkins components

This feature requires 69MB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.

9. Click **Install** to start the installation process.

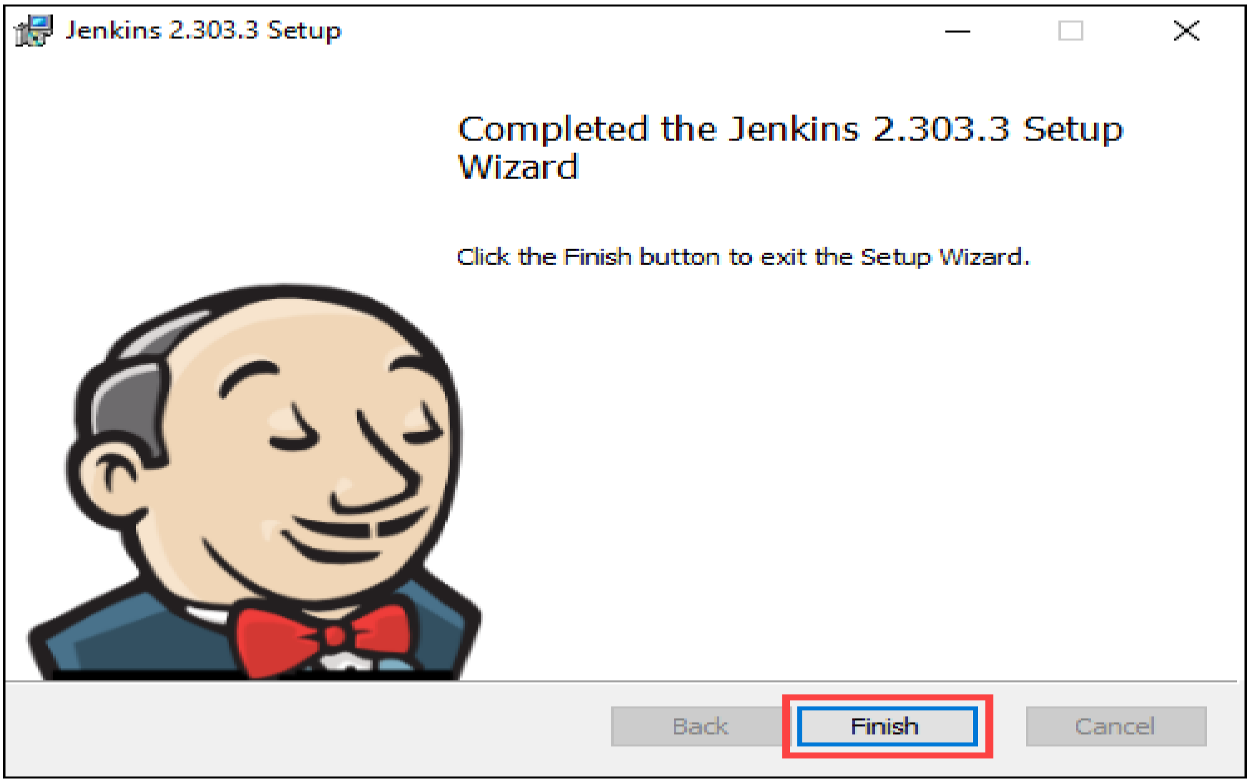


Jenkins 2.303.3 Setup

Ready to install Jenkins 2.303.3

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

10. Once the installation is complete, click **Finish** to exit the install wizard.



Jenkins 2.303.3 Setup

Completed the Jenkins 2.303.3 Setup Wizard

Click the Finish button to exit the Setup Wizard.

How to Configure Jenkins :-

- After completing the installation process, you have to unblock Jenkins before you can customize and start using it.

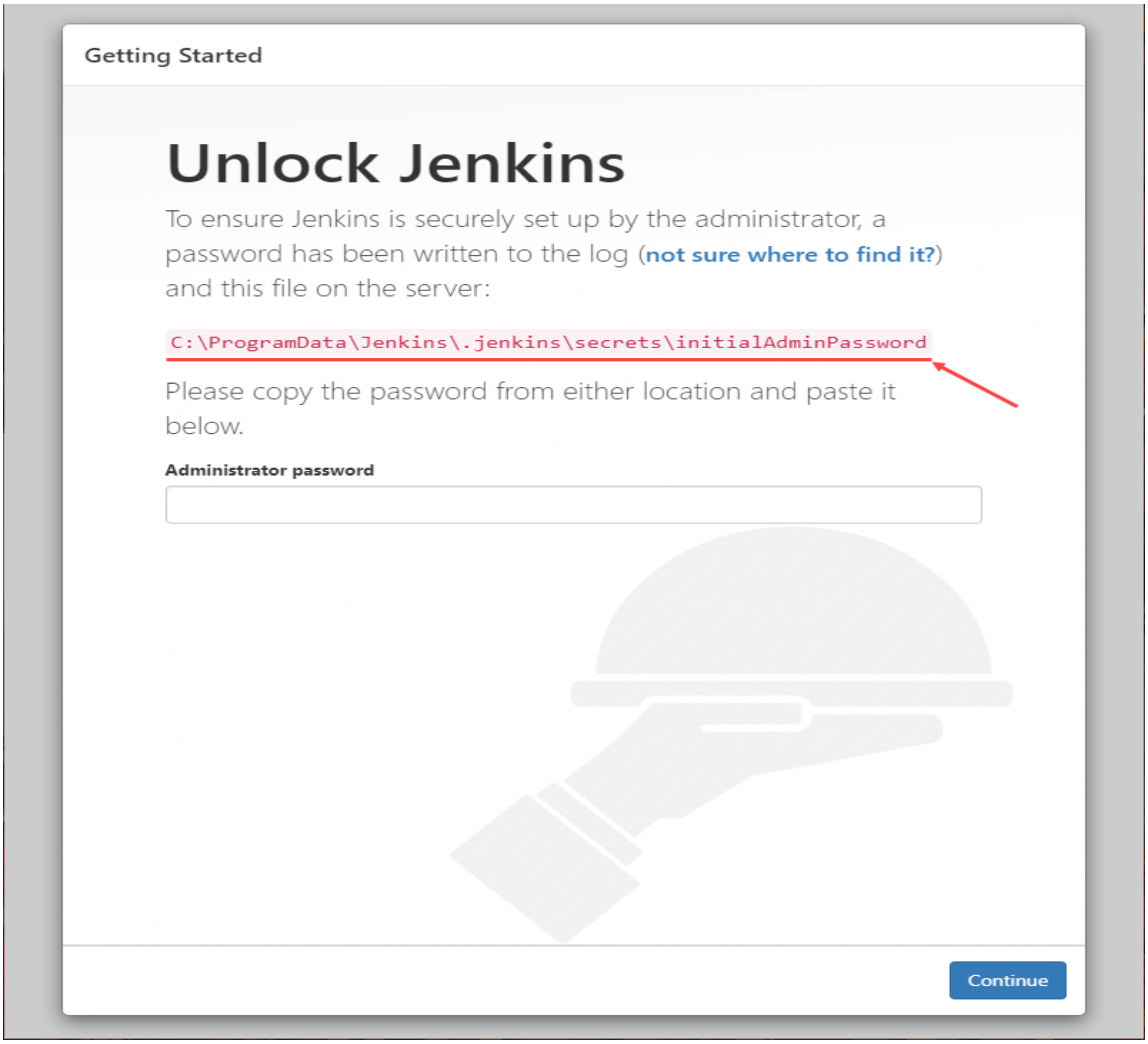
1.Browser, navigate to the port number you selected during

the installation using the following address:

- http://localhost:[port number]
- ex :- http://localhost:[8080]
- 127.0.0.1:8080

2.Navigate to the location on your system specified by the

Unblock Jenkins page.



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:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword
```

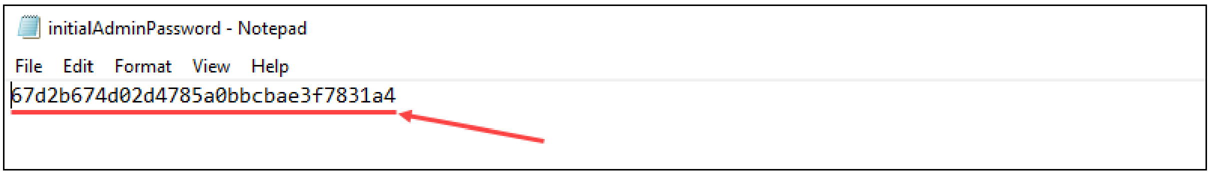
Please copy the password from either location and paste it below.

Administrator password

3.Open the **initialAdminPassword** file using a text editor such

as Notepad.

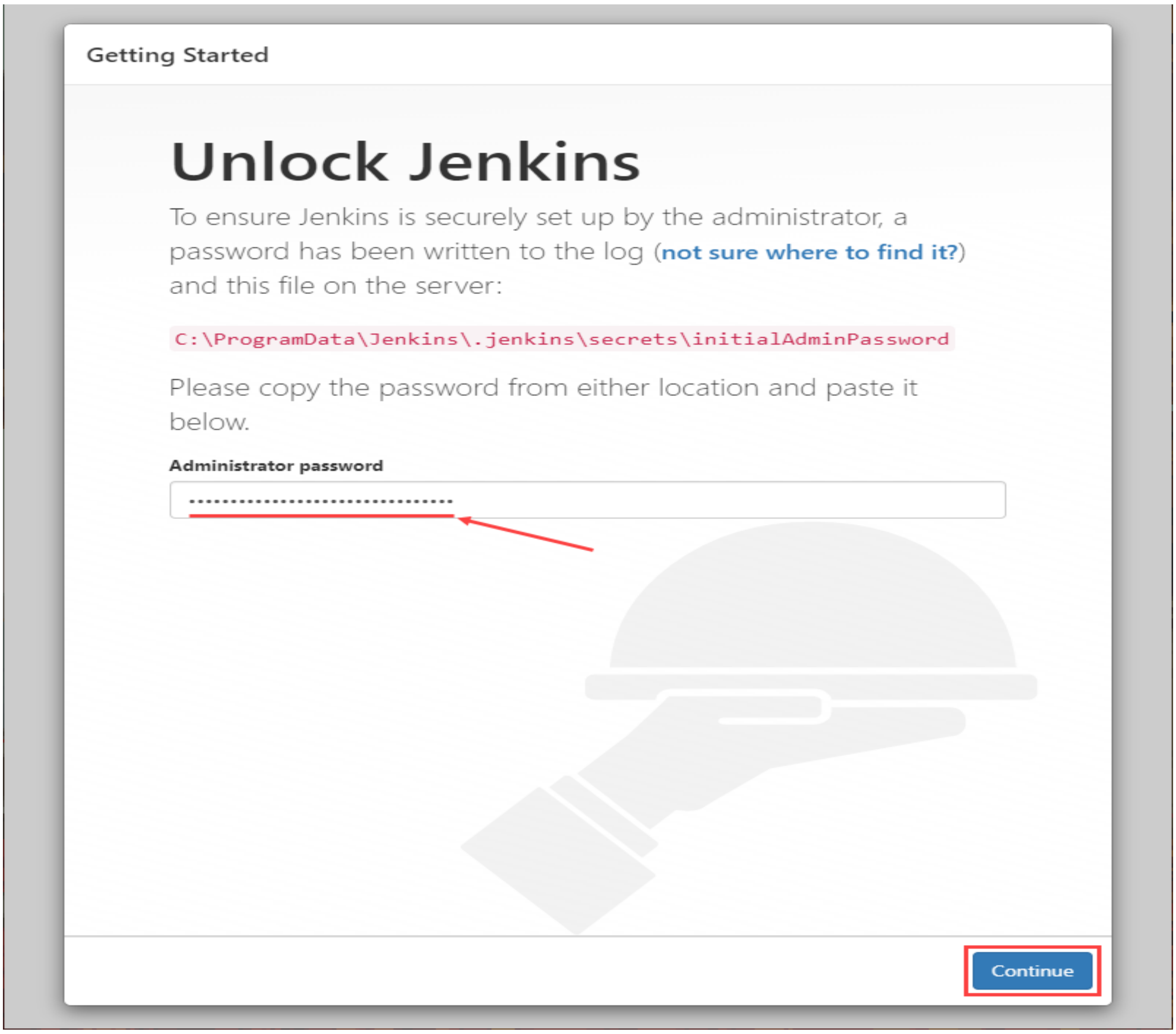
4. Copy the password from the **initialAdminPassword** file.



initialAdminPassword - Notepad

```
67d2b674d02d4785a0bbcbae3f7831a4
```

5. Paste the password in the **Administrator password** field on the Unblock Jenkins page and click **Continue** to proceed.



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:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword`

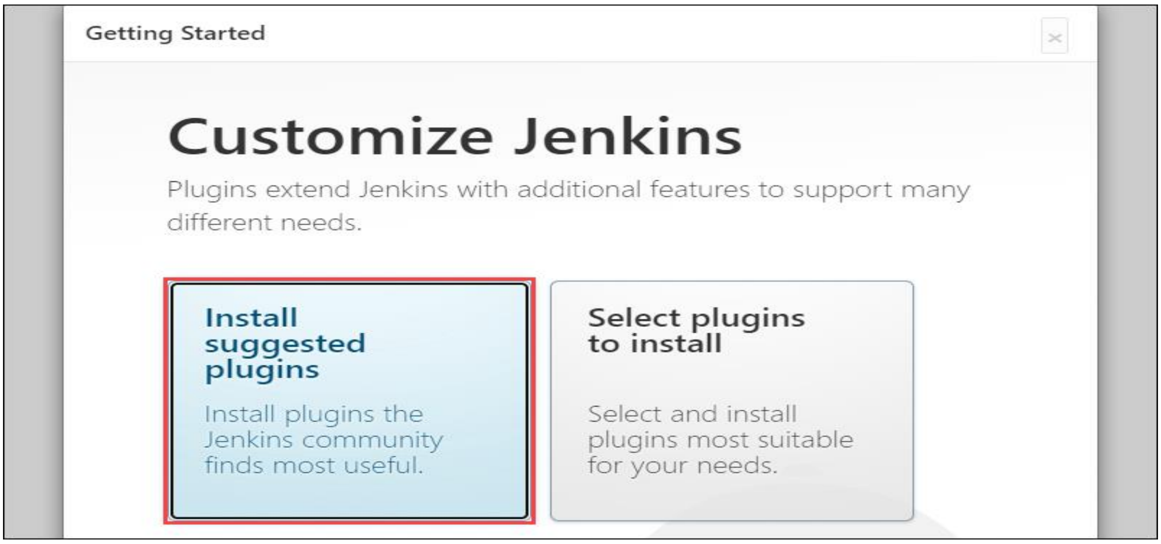
Please copy the password from either location and paste it below.

Administrator password

••••••••••••••••••••••••••••••••

Customize Jenkins :-

1. Click the Install suggested plugins button to have Jenkins automatically install the most frequently used plugins.



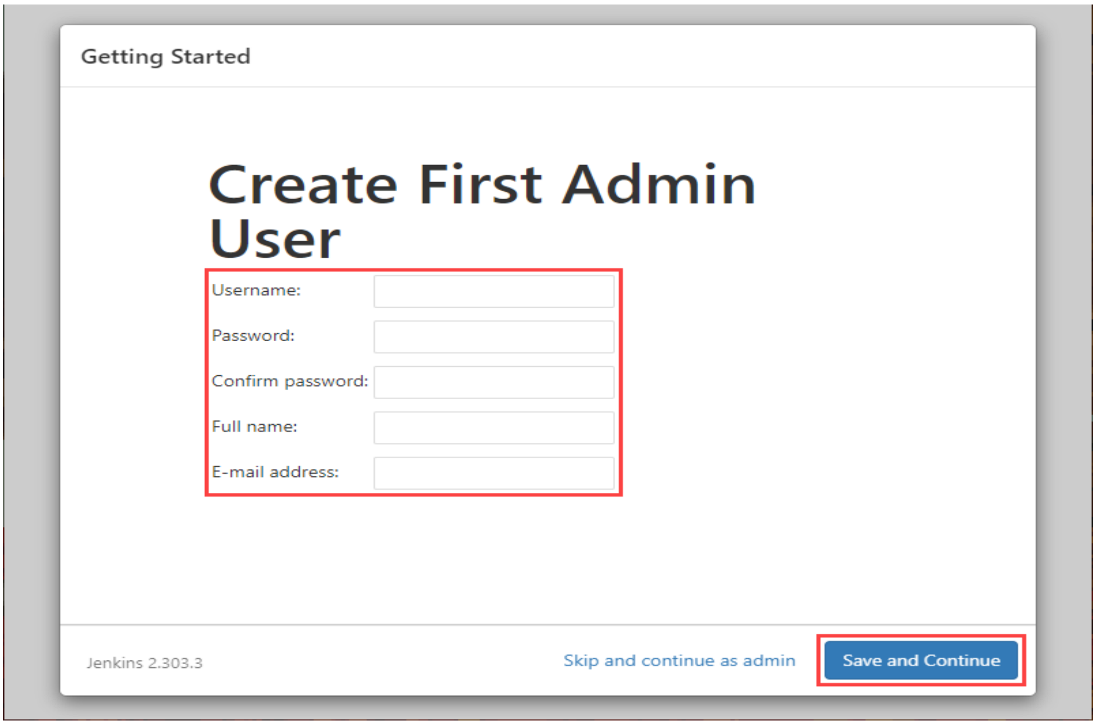
Getting Started

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.

2. After Jenkins finishes installing the plugins, enter the required information on the **Create First Admin User** page. Click **Save and Continue** to proceed.



Getting Started

Create First Admin User

Username:

Password:

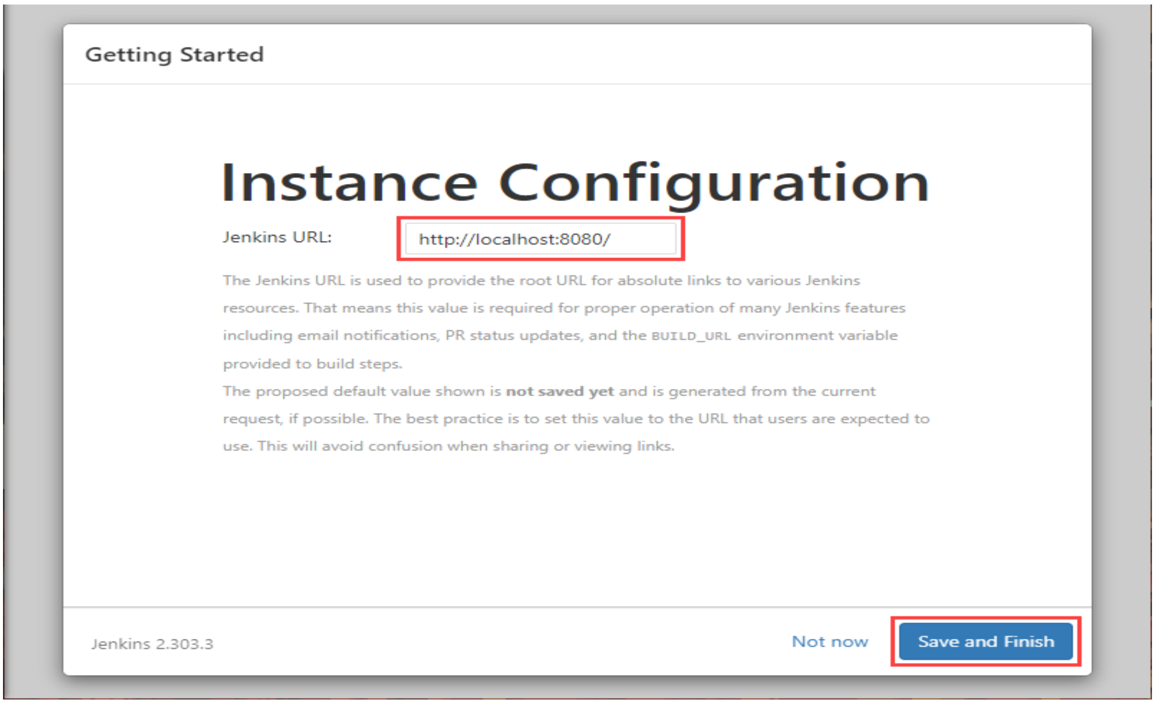
Confirm password:

Full name:

E-mail address:

Jenkins 2.303.3

3. On the **Instance Configuration** page, confirm the port number you want Jenkins to use and click **Save and Finish** to finish the initial customization.



Getting Started

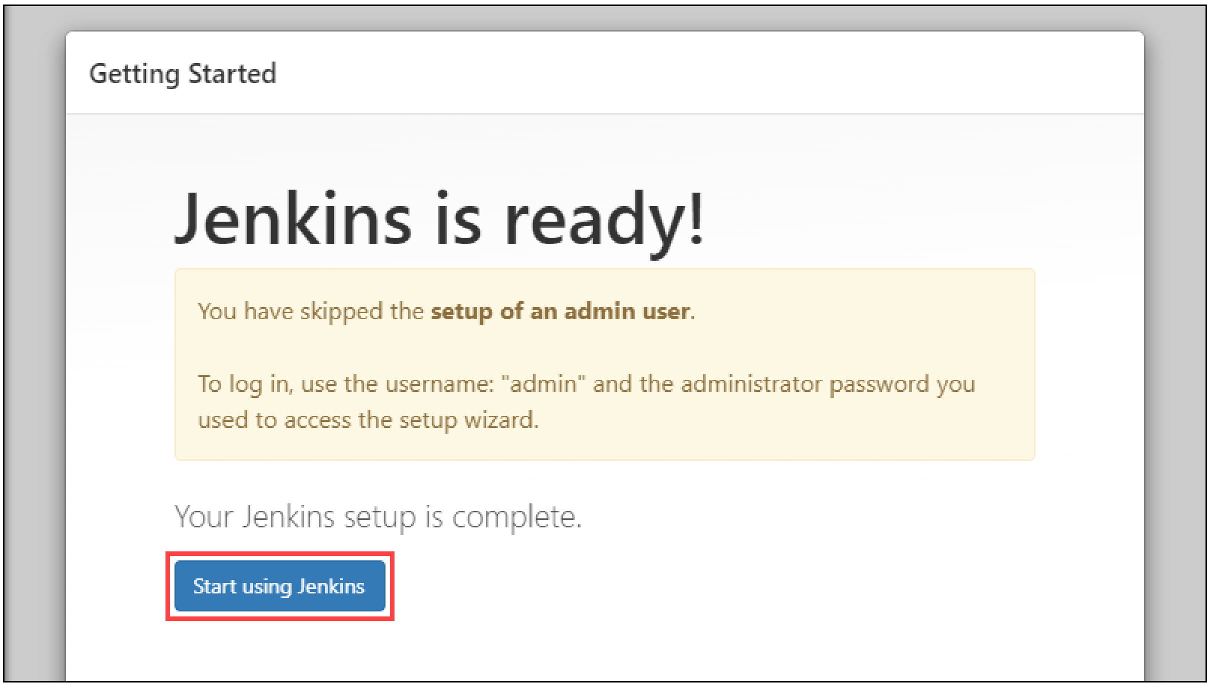
Instance Configuration

Jenkins URL: http://localhost:8080/

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

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

4. Click the **Start using Jenkins** button to move to the Jenkins dashboard.



Getting Started

Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

How to Stop Jenkins server in windows :-

- In windows search for services & select (or) search for J
- Click on this stop (or) restart button

How to Restart the Jenkins :-

- If we go to URL http://localhost:8080/safeRestart
- Jenkins will try to pause jobs and restart once all running jobs are either finished or paused

Other way :-

- Localhost:8080/restart (don’t use)

(06/02/2025)

Plugin Installation :-

There are 2 ways :-

  - Automatically
  - Manualy

- Once we login to Jenkins GUI , left side you will see the Option of manage Jenkins
- Click On manage Jenkins In first tab i.e., system configuration , under that we can see plugins
- Go to available Junkins , search for the plugin which is required , then select the plugin under the list
- Once we select the plugin then Install utton will be highlighted
- The Click on Install button

2. Manually :-

- Under Plugins tab , go for available plugins & search for it , Click on the plugin which is required
- Once we click on the plugin , click on Releases

- Under releases we can see the various versions of the Plugins
- Go for the Plugins which is required Specific version
- Under installation options we can get the direct link , click on direct link plugin will get downloaded to our local machine
- When we download Plugin in manual way again we need to deploy to Jenkins
- In Jenkins GUI Plugins tab , click on advance settings
- Their you can search for the choose file upload
- Which is download to local machine (or) local system

Plugins :-

- Add , remove , disable (or) enable Plugins that can extend the functionality of Jenkins
- When we install Plugin in automatic way Plugins extention is JPI (Jenkins Plugins)
- When we Install Plugin in Mannual way is HPI (Hudson Plugin)

How to Uninstall the Plugin ?

- Manage Jenkins → Plugins then install Plugin , then search for Plugin name which we want to uninstall
- Select the plugin and then uninstall button will be highlighted once clicked on uninstall , plugins will get uninstalled

How to Update the Plugin ?

- Manage Jenkins →Plugins →Updates
- Select the Plugin and then Update button will be Highlighted

- Once clicked on update, Plugin will get Updated

❖ **In real time we shouldn't update the plugin directly (because there maybe challenges (or) encounter)**

- It will recommended to in your local machines where other team member willn't get effect if anythink goes wrong

How to Create the Users ?

- Manage Jenkins → Security → Users (Create / Delete / Modify users that can log in to Jenkins)
- Click on the users we can see the create user option after clicking on the users option we can see the username , password , confirm password , full name , email address the click on create users

How to give permissions to the Users ?

- Manage Jenkins → security (secure Jenkins , define who is allowed to access / use the system)
- Add a user button , click on it
- It will ask userID , give user ID & click on OK button
- You can see user is add then give the required permission to the user by checking the box
- The minimum access to us is over all read
- By default what ever we create users for Genkins own user Database
- This is suitable for smaller set up where you have no existing user database
- The other option is LADP (Lightweight Directory Access Protocol)

Authorization strategy :-

- By default we will be “ Project – based Matrix Authorizate Strategy ” , with the we can manage the authorization baced the option available under Authorization strategy
- Cradentials , agent , Job etc

Matrix Authorization Strategy :-

Use Cases :-

- Matrix , Authorization , allows Configuring the lowest level permission , such as starting new builds iteams or deleting configuration items

Jenkins Projects (or) jobs :-

Free style project :-

- Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Maven Project

Pipeline (workflows) :-

- 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.

Multi branch pipeline :-

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

Organization folder :-

- Creates a set of multibranch project subfolders by scanning for repositories.

RBAC (Roll Base Access control) , where we can secure the Jenkins

How to Create Job (or) Project ?

- In Jenkins GUI , you have “ + New item ” , click on the new item , Enter item name (or) job name
- Then select Item type , click on ok button it will be highlighted

- Job configuration is consists of “ General ” (Where we can give information (or) description about the project)
- Source code management where we will be passing URL’s of the repositories , build triggers (In what way we wanted to trigger (or) build the Job) “ Build environment ” , “ Build steps ” (where we can add tasks) , post build actions

How to Create Free style Job ?

- New item → Job name → Select click on Ok
- Go to Build steps configuration of Job
- Click on “ Add Build step ” drop down button
- Select “ Execute windows batch command ”
- Select DIR , click on save
- Click on Buildnow , click for the console output

Blue Ocean :-

- Blue Ocean re thinks the Jankins user experience Designed from the ground up for Jenkins Pipeline and Compatible with freestyle jobs

07-02-2025

What if Jenkins user forget the Password ?

- As a Jenkins administrator there is a provision to reset the users Creditials
- Manage Jenkins → Security → Users → Go to user which need to be reset → Click on security → And change the Password → Save → Refresh → Restart

- Once as a administrator again you need to inform to the user please follow below steps to change the password
- Click on Username → Security → reset the password to your wish → Save → Refresh → Restart
- Slaves , Nodes , Agents these three Terminologies are same
- Jenkins folder in windows OS , we can see all Jenkins information & configuration details under
- **Path :-** c:\ProgramData\Jenkins\.jenkins
- Under .jenkins folder we can see users folder (where all the Jenkins user data available)
- **Workspace :** Where all the configured jobs information is available
- **Secrets :** Where all the configured secrets are available
- **Plugins :** All the Plugins (which are installed manualy & automatically) Available
- **Nodes :** Where all the configured nodes are Available
- **Logs :** We can see slaves (if configured) and tasks logs are available
- Apart from the above folder their Jenkins configurational file namely " config.xml " (if at all changing config.xml file first take the backup of that file (or) copy to some where else)
- And we can also see the Jenkins logs in Jenkins
  GUI Path : Manage Jenkins → Status Information → System log → All Jenkins logs → Click on it you will see all logs details
- We can see the Jenkins Version bottom right Corner in Jenkins GUI

- **And also Path :** Manage Jenkins → Status information → About Jenkins

✚ Ways to trigger Jenkins Jobs : (Interview Question)

- There are 5 types of ways :-

❖ Setup automated actions that start your build based on specific events like code changes (or) Scheduled time

Type – 1 : Trigger builds remotely (e.g., from scripts)

- In Order to use this trigger first we need to generate the authentication token of the Job
- Enable this option if you would like to trigger new builds by accessing a special predefined URL
- You'll need to provide an authorization token in the form of a string so that only those who know it would be able to remotely trigger this project's builds.
- This is most useful when your Jenkins instance grants read access to this job to anonymous users.
- When that's not the case, Jenkins will reject requests sent to the trigger URL even when the correct token is specified.
- Use the following URL to Trigger build remotely : JENKINS-URL/job/ajafreestyle/build:token=TOKEN_NAME (or) /build with Parameters ? token = TOKEN_NAME

Type – 2 : Build after other projects are built (upstream & downstream)

- Set up a trigger so that when some other projects finish building, a new build is scheduled for this project. This is convenient for running an extensive test after a build is complete
- We need to pass the Job name (which are configured already) under , Projects to watch
- Trigger only if build is stable
- Trigger even if the build is unstable
- Trigger even if the build fails
- Always trigger, even if the build is aborted

Type – 3 : Build periodically (Schedule)

- This field follows the syntax of cron (with minor difference) specifically each line consists of 5 fields separated by TAB (or) whitespace

MINUTE HOUR DOM MONTH DOW

- ❖ **MINUTE :** Minutes within the hour (0-59)
- ❖ **HOUR :** The hour of the day (0-23)
- ❖ **DOM :** The day of the month (1-31)
- ❖ **MONTH :** The month (1-12)
- ❖ **DOW :** The day of the week (0-7) , where 0 and 7 are Sunday

- To specify multiple values for one field , the following operators are available in the order of precedence
- For reference (or) Practice go with the website “ corntab.guru ”

Type – 4: GitHub hook trigger for GITScm polling

- When Jenkins receives a GitHub push hook, GitHub Plugin checks to see whether the hook came from a GitHub repository which matches the Git repository defined in SCM/Git section of this job.

Type – 5:

- Configure Jenkins to poll changes in SCM.
- Note that this is going to be an expensive operation for CVS, as every polling requires Jenkins to scan the entire workspace and verify it with the server.

❖ Apart from the above 5 trigger options / ways we can also tripper jobs manualy

<u>08-02-2025</u>

- February 8th Saturday Every minute Every Hour

MINUTE	HOUR	DOM	MONTH	DOW
*	*	8	2	6

Scheduling the Job: -

- Job configuration of build periodically triggers & POLSCM trigger

- Create New item → triggers → Build Periodically → (or) POLSCM → give the Schedule to Build the Project (or) Job → Windows bath → dir command (or) any → Save → refresh

- Then the job will be build accordingly based on schedule

Build after other Projects are build: -

- Create more than on Job In triggers option build after other Projects are build once checks that need to pass Projects to watch name then click on Save and Check
  Eg :- Job1 , Job2 , Job3
- For Job2 , put the upstream project as Job1
- For Job3 , put the upstream Job as Job 2
- When we build the Job1 , once Job1 Completed automatically down streams Projects has to be build
- For Job2 , Once select the build after other Projects are build and select the other options even if the build fails → Save
- For Job1 , Do some mistake wantedly and Check for the all the Jobs

Pipeline Project: -

- We are going for pipeline projects in order to Secure the CICD pipeline code by storing in any of the repositories (Github , bit bucket , Git Lab)
- Sample,Hello World pipelines script Creation
- Click on new item select pipeline and click on OK then ,

- Go for Pipeline in Configuration where we can define (or) write the pipeline script , we can write the script in script tab (or) else , we can take it from SCM
- In the sript tab , click on drop down option to try sample Pipeline

  Syntax:

```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
    }
}
```

- Click on Save
- Make sure that stage view Plugin should be installed , So that we can see Stages

Rebuild v/s Replay :

- When we use Rebuild we aren't Changing anything but Where as Replay gives us Provision to Change the code then & their
- Replay we can also use it as one of the trouble shooting mechanism

- If we don't know how to write Pipelines Script code we can take the advantage of Snippet generator in Pipeline Syntax option

(Pipeline syntax will only visiable when you create Pipeline Job (or) Project

- The Snippet Generator will help you learn the Pipeline Script code which can be used to define various steps pick a step you are interested in form the list , Configure it click Generate Scripts , and you will see a Pipeline Script Statement that would call the step with that configuration , you may copy and paste the whole statement into your script (or) pickup just the option you care about

Ex : To Print message in sample step of Snippet Generator we can search for the echo , echo: print Message then , what message we want to print we have to pass it in message block , then click on Generate Pipeline Script , which will give us Script to print message

- By default Pipeline execution will work on Serial mode (one by one mode only) . If the Previous stages pass it will go for next Stages .