

Jenkins



Documentation

1. Overview:

Jenkins is an open-source Continuous Integration (CI) and Continuous Delivery (CD) server, primarily used to automate repetitive tasks such as building, testing, and deploying software. It provides a robust set of features and plug-ins to support the build, deploy, and automation needs of virtually any project.

Jenkins plays a pivotal role in DevOps by facilitating the integration of automated tests, code quality assessments, and early reporting on defects that forms part of the Continuous Integration and Continuous Delivery process. It can integrate project changes more easily by quickly finding issues as soon as a commit is made in a version control system (VCS), which leads to easier problem diagnosis and increased quality of software delivered.

2. Features:

Easy Installation: Jenkins is a self-contained Java program that comes ready to run out of the box. It can be installed through native system packages, Docker, or run standalone by any machine with a Java Runtime Environment installed.

Platform Agnostic: Jenkins can be run on any operating system (Linux, Windows, macOS) or on a cloud-based infrastructure.

Over 1500 Plugins: Jenkins supports a broad array of plugins to provide a versatile environment that can be tailored to your specific requirements.

Easy Configuration: Jenkins offers a simple setup process and user-friendly web interface for configuring the system and jobs.

Distributed Build Architecture: Jenkins can distribute build and test loads for multiple machines with different operating systems to speed up jobs, increase capacity, or run separate jobs on isolated environments.

Pipeline as Code: The Jenkins pipeline feature provides an extensible set of tools for modelling simple to complex delivery pipelines as code, thereby enforcing version control and consistency.

3. Jenkins Architecture:

Jenkins follows master-slave architecture to manage distributed builds. The role of the Jenkins master is to handle scheduling of build jobs, dispatching builds to the slaves for the actual execution, monitor the slaves, and record and present the build results. A slave is a computer set up to offload build projects from the master.

4. Conclusion:

Jenkins is a powerful and flexible tool that enables teams to automate their software lifecycle from integration, testing, delivery to deployment. Its vibrant open-source community constantly contributes to its growth, making it an integral part of many DevOps environments today.

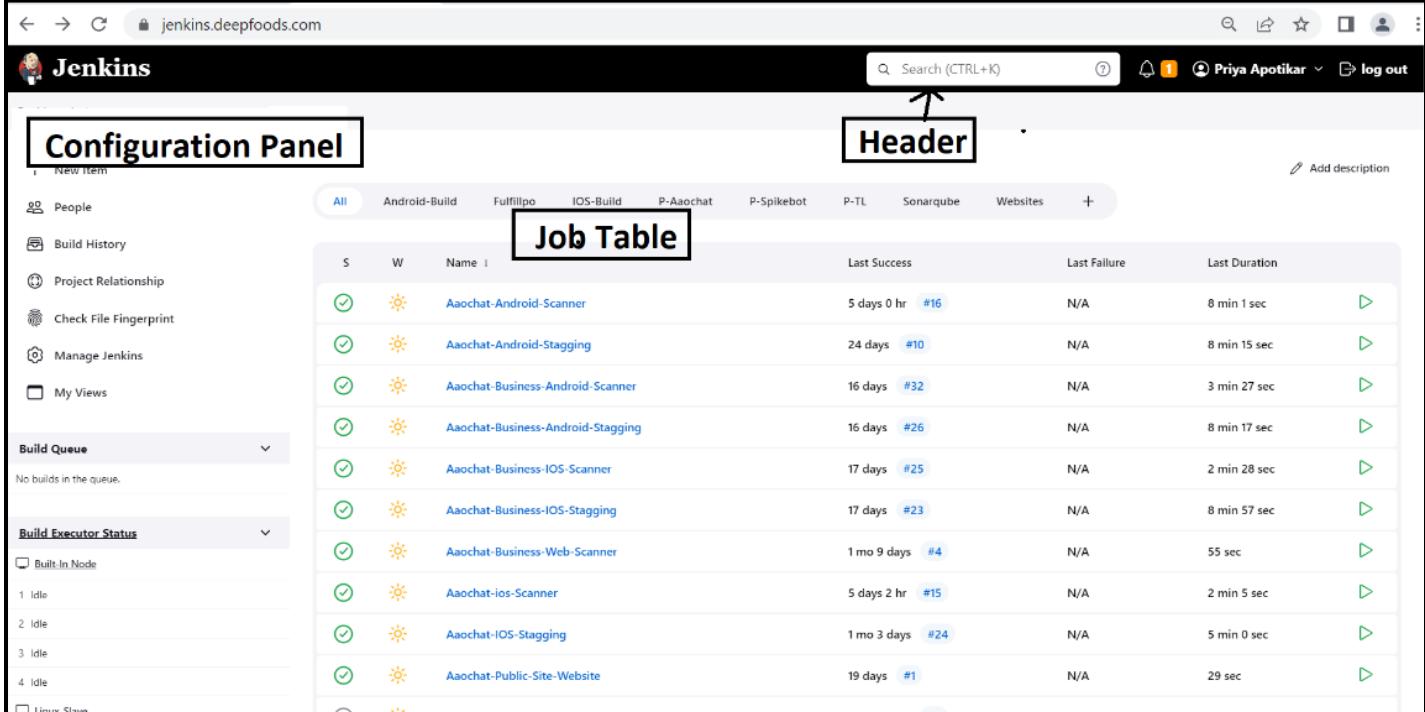
Index

Sr. No.	Parameter	Page No.
1	Introduction	4
2	Procedure	6
2.1	Jenkins Master	6
3	Create Jobs and Views	14
4	Manage Jenkins	18
4.1	System Configuration	20
4.2	Tools Configuration	24
4.2.1	Add JDK	25
4.2.2	Add Git	25
4.2.3	Gradle installation	26
4.2.4	Add Maven	27
4.2.5	Add Node.js	27
4.3	Node	28
4.4	Plugins	33
4.5	Security	35
4.6	Credentials	37
4.7	Users	39

1. Introduction

What Is Jenkins

Jenkins is an open-source solution comprising an automation server to enable continuous integration and continuous delivery (CI/CD), automating the various stages of software development such as build, test, and deployment.



The screenshot shows the Jenkins web interface. At the top, there's a header bar with a search field, notifications, and user information. Below the header is a navigation menu with links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. There are also sections for 'Build Queue' (empty) and 'Build Executor Status' (showing 1 idle node). The main area is titled 'Configuration Panel' and contains a 'Job Table'. The table has columns for 'S' (Status), 'W' (Last Success), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. It lists ten jobs, all of which are green (success) and have 'N/A' for failure and duration. The jobs are: Aaochat-Android-Scanner, Aaochat-Android-Stagging, Aaochat-Business-Android-Scanner, Aaochat-Business-Android-Stagging, Aaochat-Business-IOS-Scanner, Aaochat-Business-IOS-Stagging, Aaochat-Business-Web-Scanner, Aaochat-ios-Scanner, Aaochat-IOS-Stagging, and Aaochat-Public-Site-Website. Each job entry includes a link icon.

A Pictorial Representation of the Jenkins Interface

Jenkins Used for

- Continuous Integration and Continuous Delivery
- Building and Testing Projects
- Automating Deployment & Execution of Scripts
- Pipeline as Code
- Supporting Various Plugins
- Reporting

Jenkins is aimed at enabling swift and safe delivery of software, from integrating and building to testing and deploying.

How Does Jenkins Work?

Jenkins may be operated as a server on various operating systems, including Windows, macOS, Unix versions, and, most notably, Linux. It also runs on the Oracle JRE or OpenJDK and requires a Java 8 virtual machine or higher. Jenkins is often executed as a Java servlet within a Jetty application server, and other Java application servers, such as Apache Tomcat, Nginx can be used to run it.

What Is a Jenkins Pipeline?

Pipelines are needed to run Jenkins. A pipeline is a set of steps the Jenkins server will execute to complete the CI/CD process's necessary tasks. In the context of Jenkins, a pipeline refers to a collection of jobs (or events) connected in a specific order.

Jenkins Pipelines comprise a powerful technology that includes a set of tools for hosting, monitoring, compiling, and testing code or code modifications across various tools such as:

Continuous integration server: Jenkins

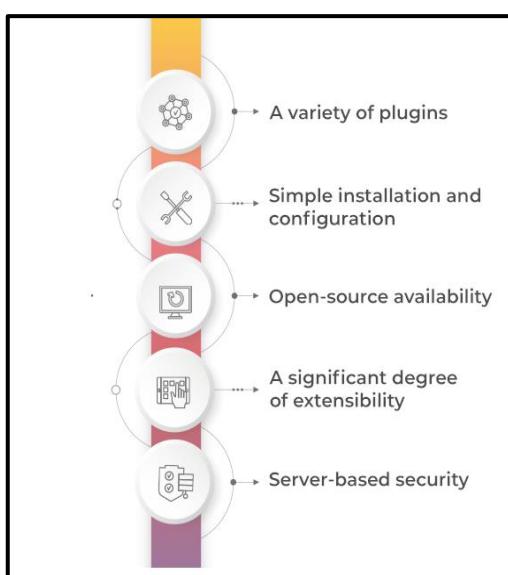
Source control software: GIT

Build tools: Gradle, Ant

Automation testing framework:



Key Features of Jenkins



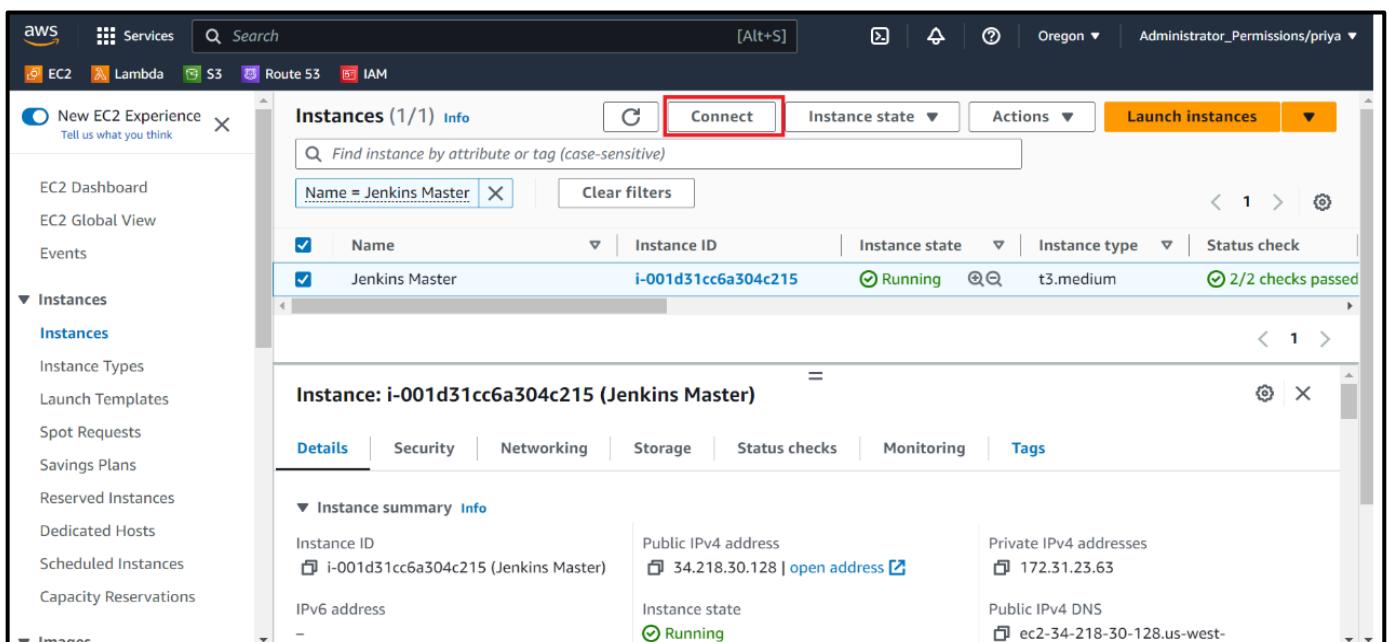
2. Procedure:

2.1 Jenkins Master

- Login to Your Account, and Launch an EC2 Instance

Login into your AWS account and type “EC2” in the search box. Click on “EC2” and select the “Ubuntu” image.

Give a name for your instance Jenkins Master Next, create a security group and make sure to have **port 22, port 80** and **port 8080** open so that the instance can “**send and receive**” traffic to the internet. Leave the rest of the default settings as and **click launch to deploy your ubuntu instance**. You have the option to connect to your instance remotely via SSH or through “**instance connects**” within the instance console.



Steps to Setup Jenkins Master server:

- Update Your System

Command1: sudo apt update

- Install Java

Jenkins requires Java to run., you can install the OpenJDK package:

Command2: sudo apt install openjdk-11-jdk

- Check the Java Version:

Command3: java --version

```
* Support:      https://ubuntu.com/advantage

System information as of Tue Jul 18 11:59:35 UTC 2023

System load: 0.41          Processes:        101
Usage of /: 21.0% of 7.57GB  Users logged in:   0
Memory usage: 23%          IPv4 address for eth0: [REDACTED]

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[REDACTED]:~$ █
```

- Add the repository key to the system:

Command4: wget -q -O - <https://pkg.jenkins.io/debian-stable/jenkins.io.key> | sudo apt-key add -

- Append the Debian package repository address to the server's sources.list:

Command5: sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >/etc/apt/sources.list.d/jenkins.list'

- Update the repository:

Command6: sudo apt update

Command7: curl -fsSL <https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key> | sudo tee /usr/share/keyrings/jenkins-keyring.asc

Command8: echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] <https://pkg.jenkins.io/debian-stable binary/> | sudo tee /etc/apt/sources.list.d/jenkins.list

- Install Jenkins:

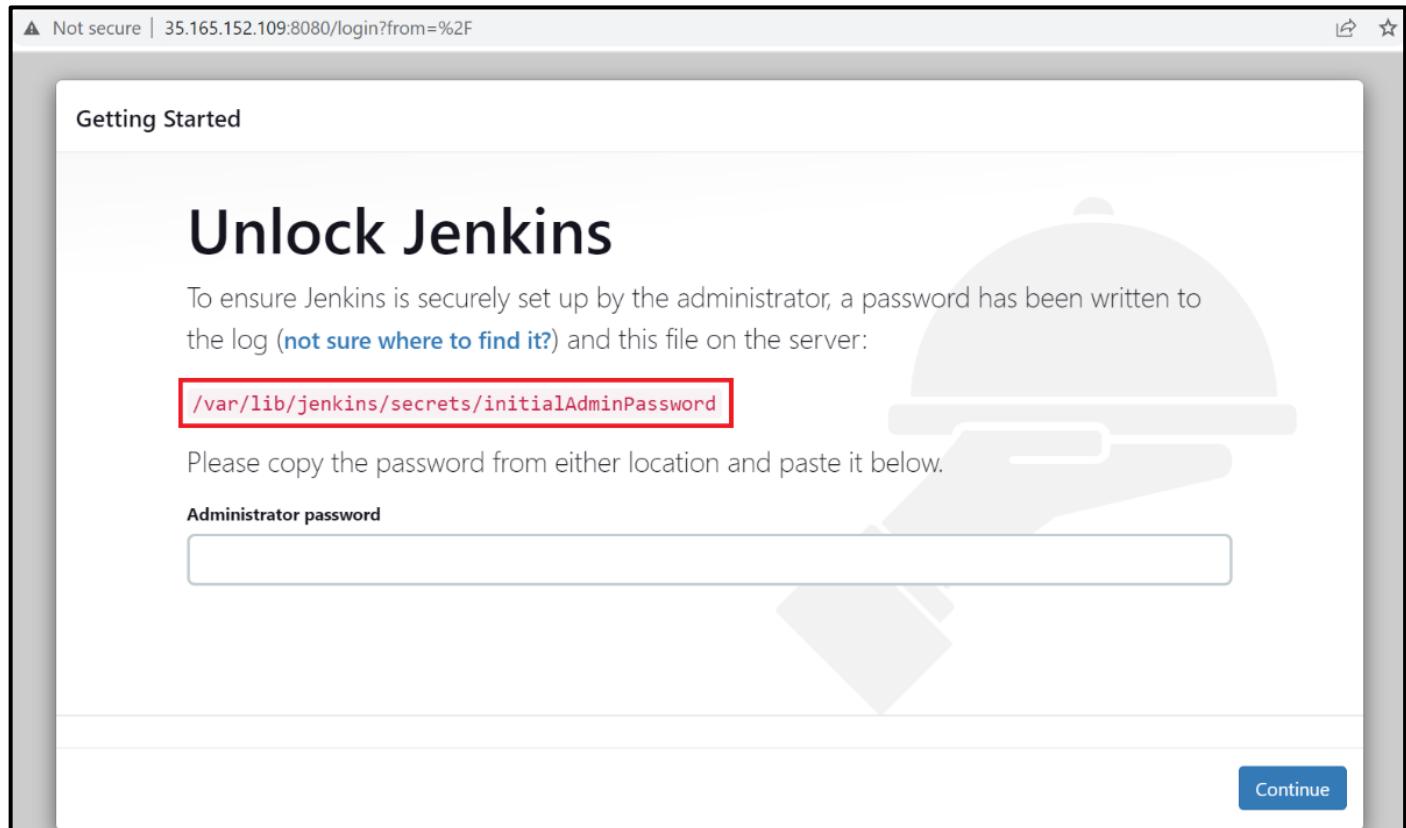
Command9: sudo apt install Jenkins

- Start Jenkins:

Command10: sudo systemctl start jenkins

- Enable Jenkins:

Command11: sudo systemctl enable jenkins



- Setting up Jenkins

Command12: cat /var/lib/jenkins/secrets/initialAdminPassword

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:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

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

Administrator password

```
.....
```

Continue

- Customize jenkins

install suggested 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.

Getting Started

Create First Admin User

Username

Password

Confirm password

Jenkins 2.401.2

[Skip and 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.

Jenkins 2.401.2

[Not now](#)[Save and Finish](#)

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

Jenkins 2.401.2

 Jenkins

Dashboard >

- + New Item
- People
- Build History
- Manage Jenkins
- My Views

Build Queue ▼
No builds in the queue.

Build Executor Status ▼
1 Idle
2 Idle

Search (CTRL+K) ? 1 Admin log out

Welcome to Jenkins!

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

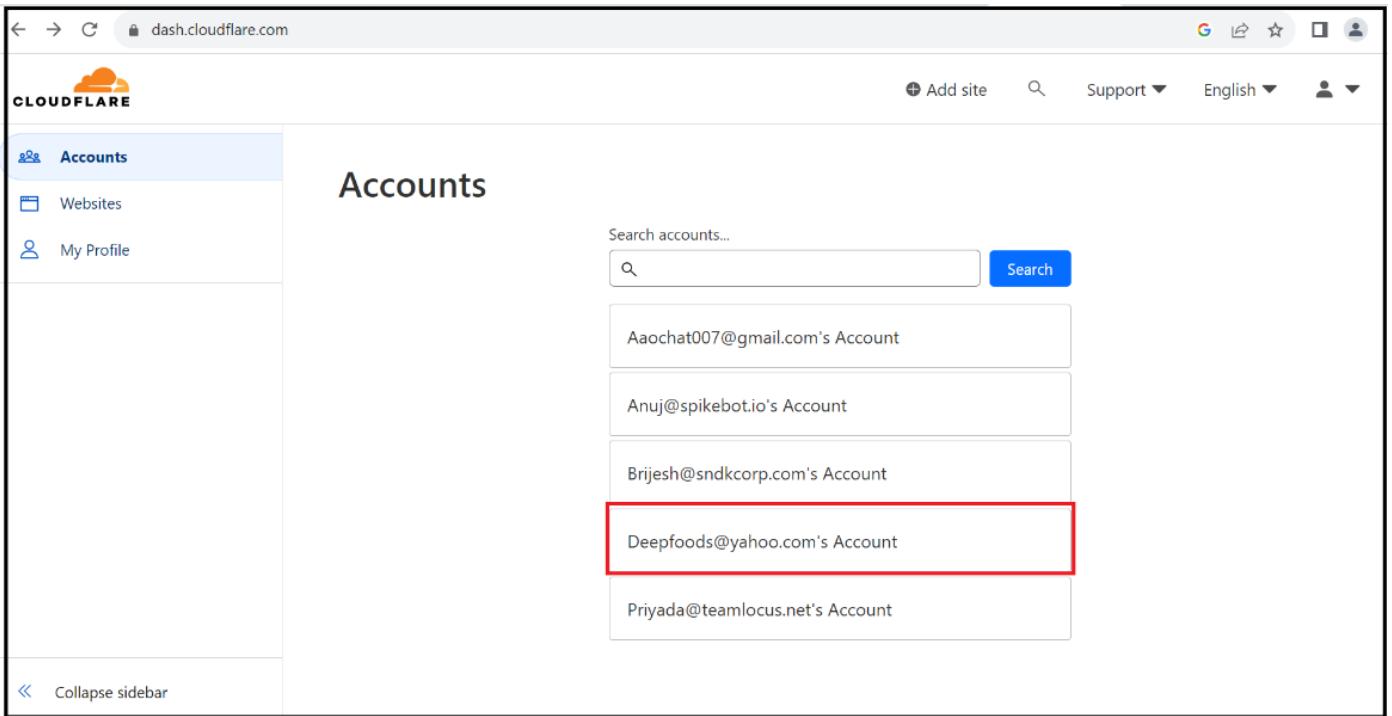
Start building your software project

Create a job →

Set up a distributed build

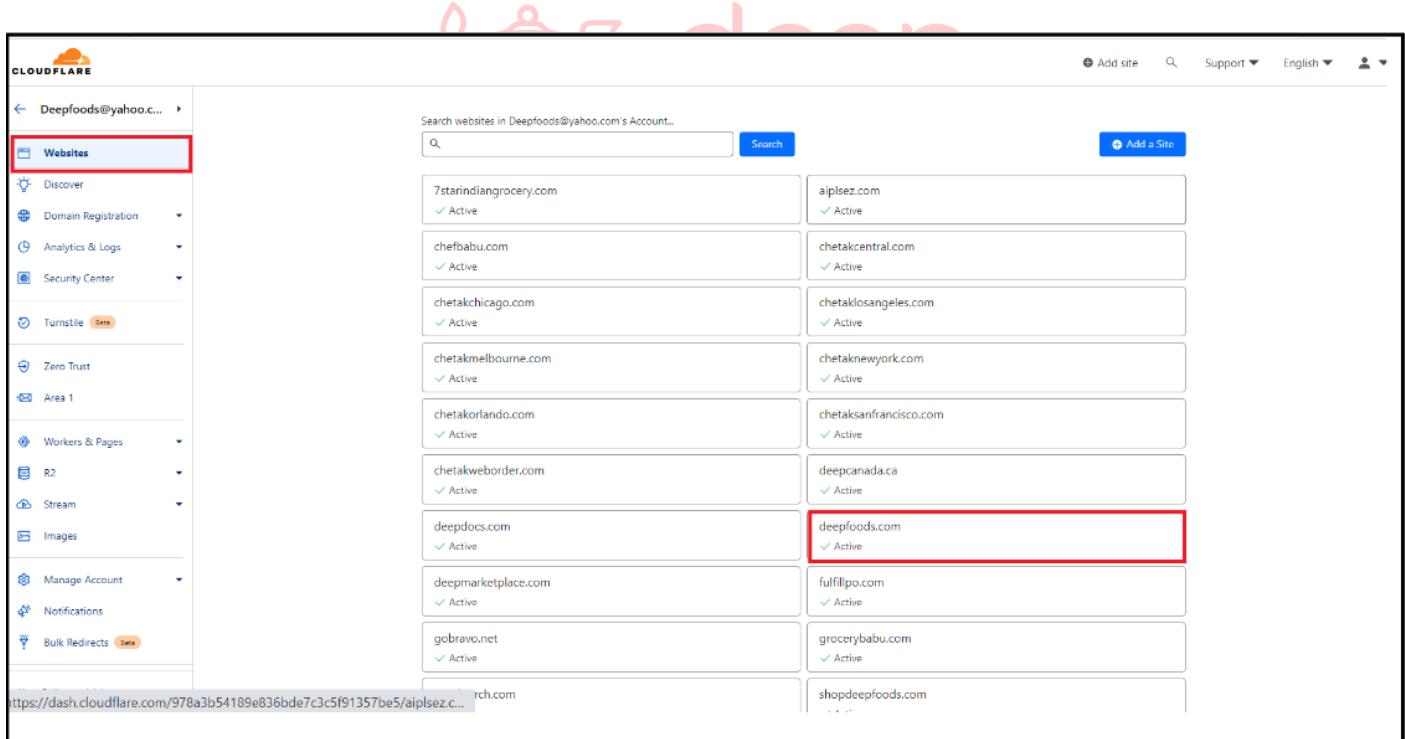
Set up an agent →
Configure a cloud →

- Add domain in CLOUDFLARE
- Select Deepfoods@yahoo.com's Account



The screenshot shows the Cloudflare Accounts page. On the left, there's a sidebar with 'Accounts' selected, followed by 'Websites' and 'My Profile'. The main area is titled 'Accounts' and contains a search bar. Below it, a list of accounts is shown in boxes: 'Aaochat007@gmail.com's Account', 'Anuj@spikebot.io's Account', 'Brijesh@sndkcorp.com's Account', 'Deepfoods@yahoo.com's Account' (which is highlighted with a red border), and 'Priyada@teamlocus.net's Account'.

- Navigate to the Cloudflare website and 'Log In' Deepfoods@yahoo.com account.
- Once you are logged in, click on the '+Add a Site' or 'Select the site' at the top of the page.



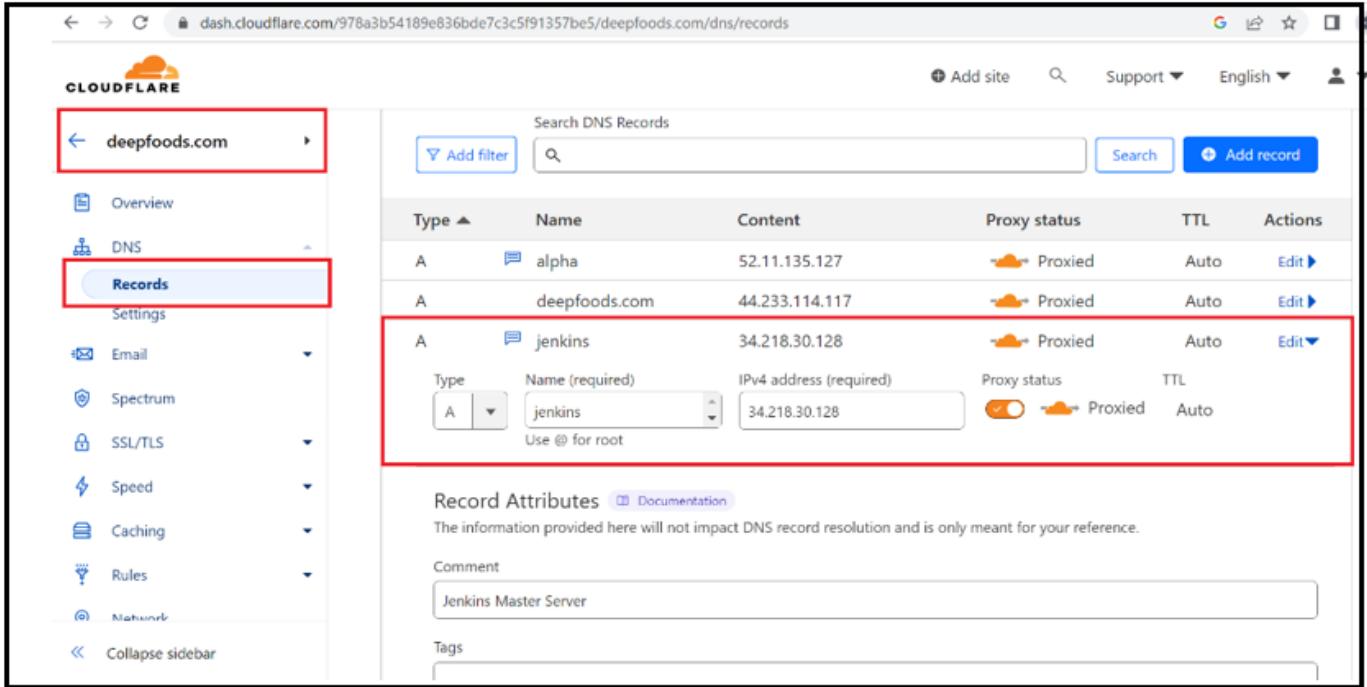
The screenshot shows the Cloudflare Websites page for the account 'Deepfoods@yahoo.com'. The sidebar on the left has 'Websites' selected (highlighted with a red box). The main area shows a grid of websites under the heading 'Search websites in Deepfoods@yahoo.com's Account...'. The websites listed are: 7starindiangrocery.com, aipsez.com, chefbabu.com, chetakcentral.com, chetakchicago.com, chetaklosangeles.com, chetakmelbourne.com, chetakknewyork.com, chetakkorlando.com, chetaksanfrancisco.com, chetakwborder.com, deepcanada.ca, deepdocs.com, deepfoods.com (highlighted with a red box), deepmarketplace.com, fulfillpo.com, gobravo.net, grocerybabu.com, and shopdeepfoods.com.

- Scan DNS Records

Cloudflare will automatically scan the DNS records of your domain. This process can take up to a minute.

After the scan, Cloudflare will display the DNS records they found. Check them carefully to ensure they match the records you have. If they don't, manually add them. Click 'Continue' when you are done.

- Add records for Jenkins



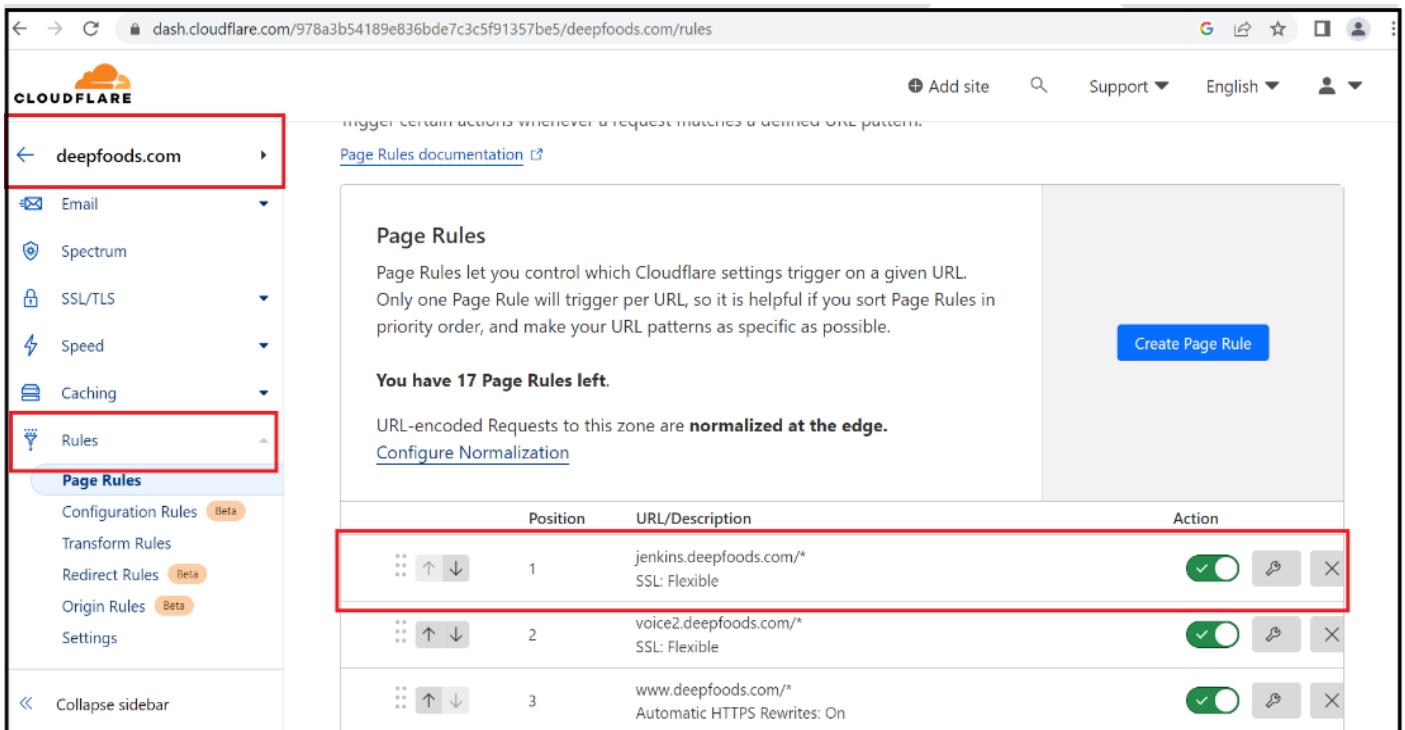
The screenshot shows the Cloudflare DNS Records page for the zone 'deepfoods.com'. The left sidebar is collapsed. The 'DNS' section is selected in the main menu. Under 'Records', a new A record is being created for the name 'jenkins' with the content '34.218.30.128'. The 'Proxy status' is set to 'Proxied' and 'TTL' is 'Auto'. The table lists existing records: 'alpha' at 52.11.135.127 and 'deepfoods.com' at 44.233.114.117.

Type	Name	Content	Proxy status	TTL	Actions
A	alpha	52.11.135.127	Proxied	Auto	Edit
A	deepfoods.com	44.233.114.117	Proxied	Auto	Edit
A	jenkins	34.218.30.128	Proxied	Auto	Edit

- Add Rules

Jenkins.deepfoods.com/*

SSL Flexible

The screenshot shows the Cloudflare Page Rules page for the zone 'deepfoods.com'. The left sidebar is collapsed. The 'Rules' section is selected in the main menu. Under 'Page Rules', there is one rule listed: 'jenkins.deepfoods.com/*' with 'SSL: Flexible'. The 'Action' column shows a green toggle switch and edit/cross icons.

Position	URL/Description	Action
1	jenkins.deepfoods.com/* SSL: Flexible	<input checked="" type="checkbox"/>
2	voice2.deepfoods.com/* SSL: Flexible	<input checked="" type="checkbox"/>
3	www.deepfoods.com/* Automatic HTTPS Rewrites: On	<input checked="" type="checkbox"/>

3. Create Jobs and Views

Creating views help you to organize your jobs.

Crons:

All jobs will get automatically to default view, when you create number of jobs its hard to maintain.

Best Practice:

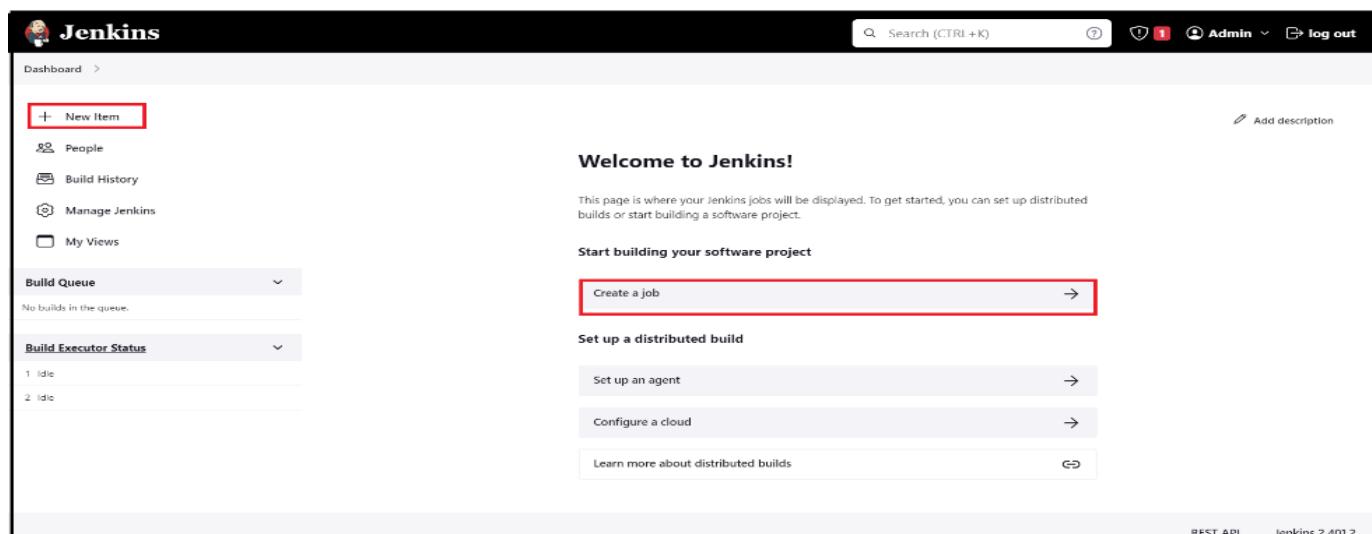
Create view per application and pull respective jobs to that view for better maintainability.

Prerequisites:

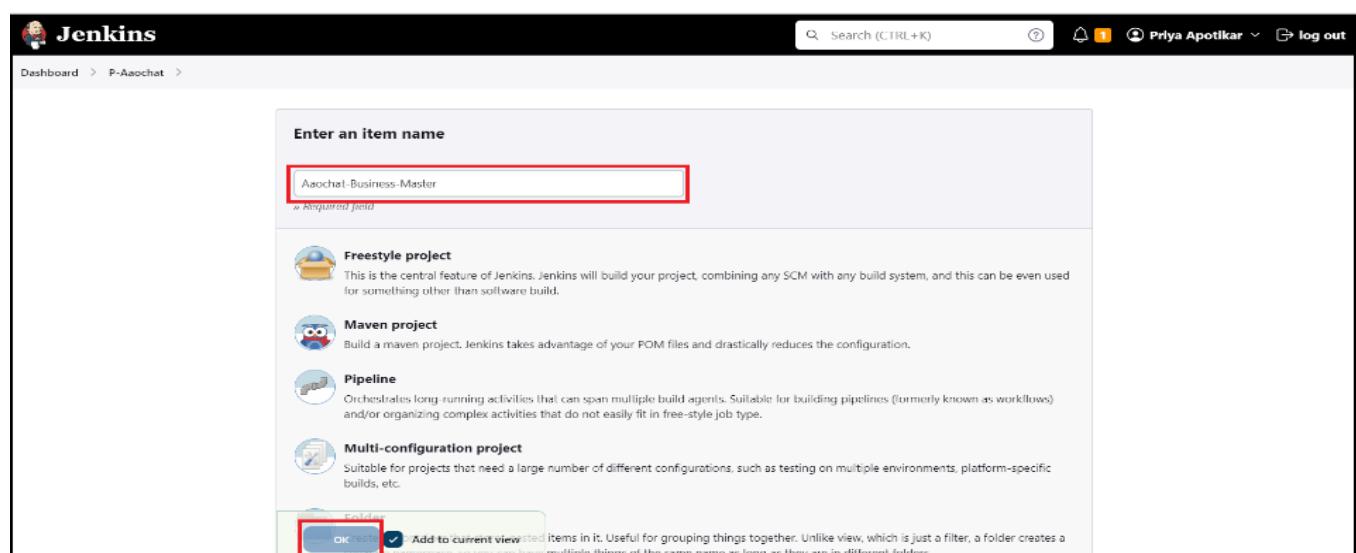
Jenkins Installed

Steps:

create view and add jobs



The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'. Below the sidebar are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 idle, 2 idle). The main content area has a heading 'Welcome to Jenkins!' followed by a paragraph about displaying Jenkins jobs. It features a 'Start building your software project' section with a 'Create a job' button, which is highlighted with a red box. Below this are sections for 'Set up a distributed build' (with 'Set up an agent' and 'Configure a cloud' buttons) and a link to 'Learn more about distributed builds'. At the bottom right, it says 'REST API Jenkins 2.401.2'.

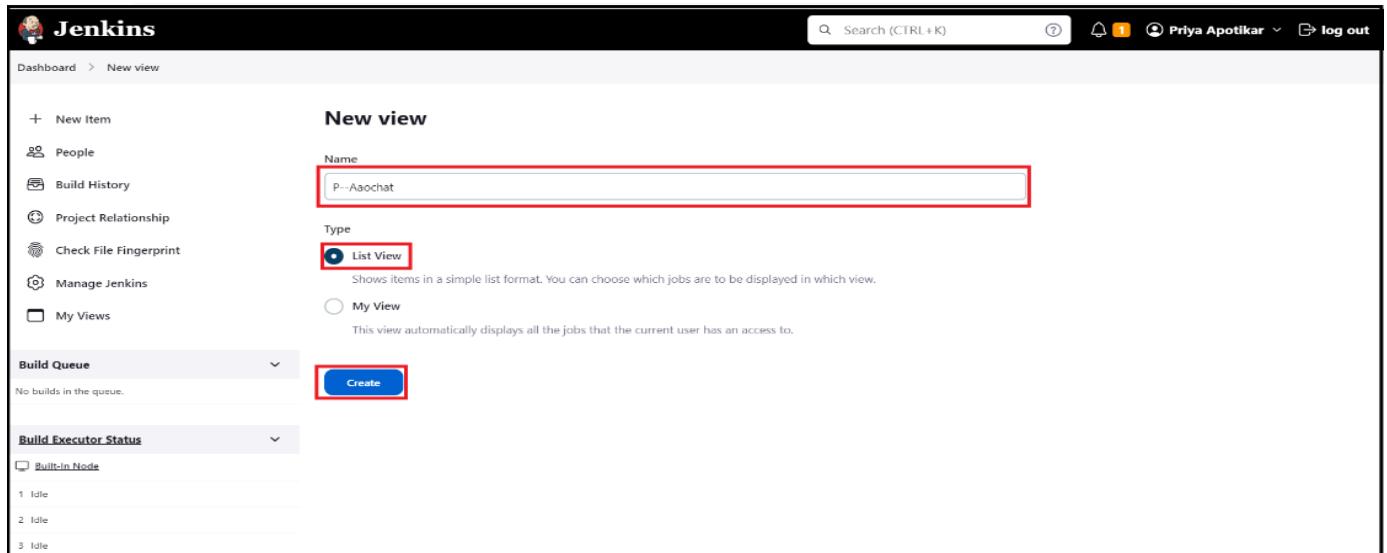


The screenshot shows a 'Enter an item name' dialog box. Inside, there's a text input field containing 'Aaochat-Business-Master' with a red border around it. Below the input field are four project types: 'Freestyle project', 'Maven project', 'Pipeline', and 'Multi-configuration project', each with a small icon and a brief description. At the bottom of the dialog is a 'Folder' section with an 'OK' button, which is also highlighted with a red box. A tooltip for the 'Folder' section explains that it's useful for grouping items together. The background shows the Jenkins dashboard with a user profile 'Priya Apotikar'.

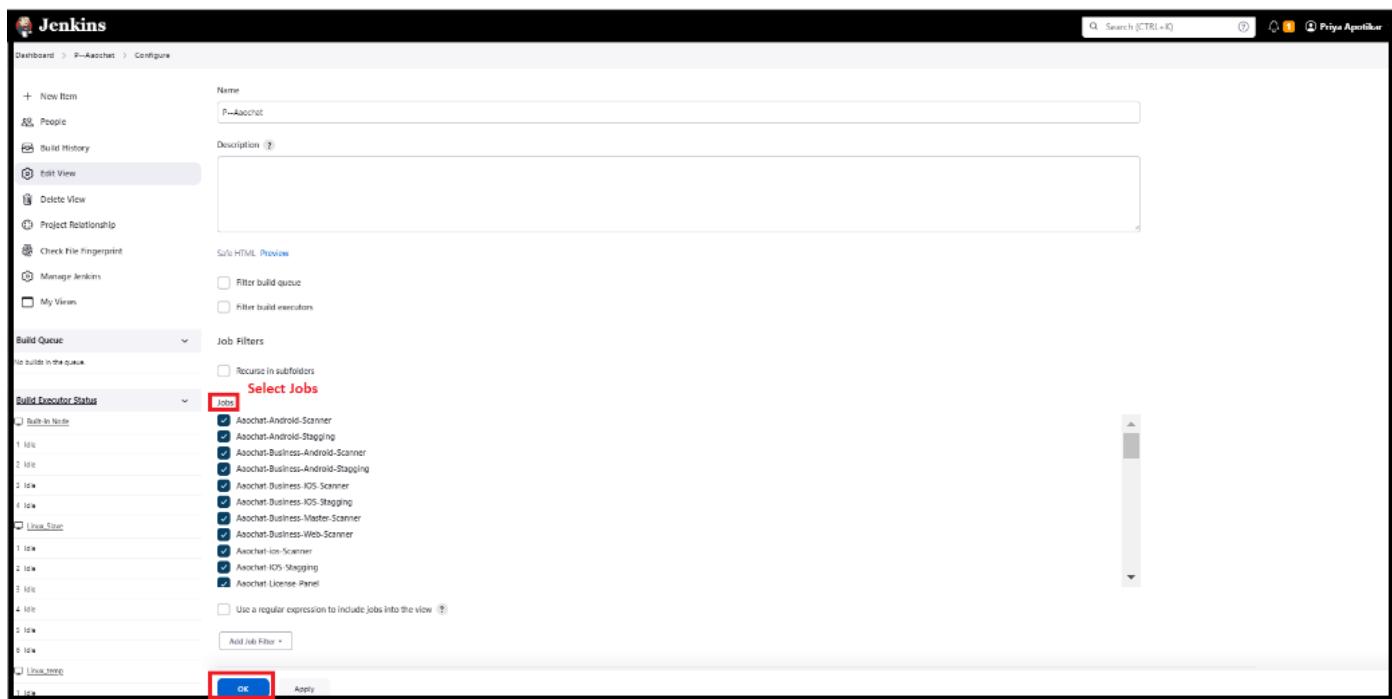
- Click on "New View"
- Click this "+" tab.



- Enter View Name and Select View Type
- You will be directed to a new page. where need to enter details for the new view.
- In the "Name" field, specify a unique name for the view.
- Below the "Name" field, you'll have to select the "View type". By default, Jenkins offers two types of views - "List View", "My View". Select the one you prefer.
- Click "create" to proceed.

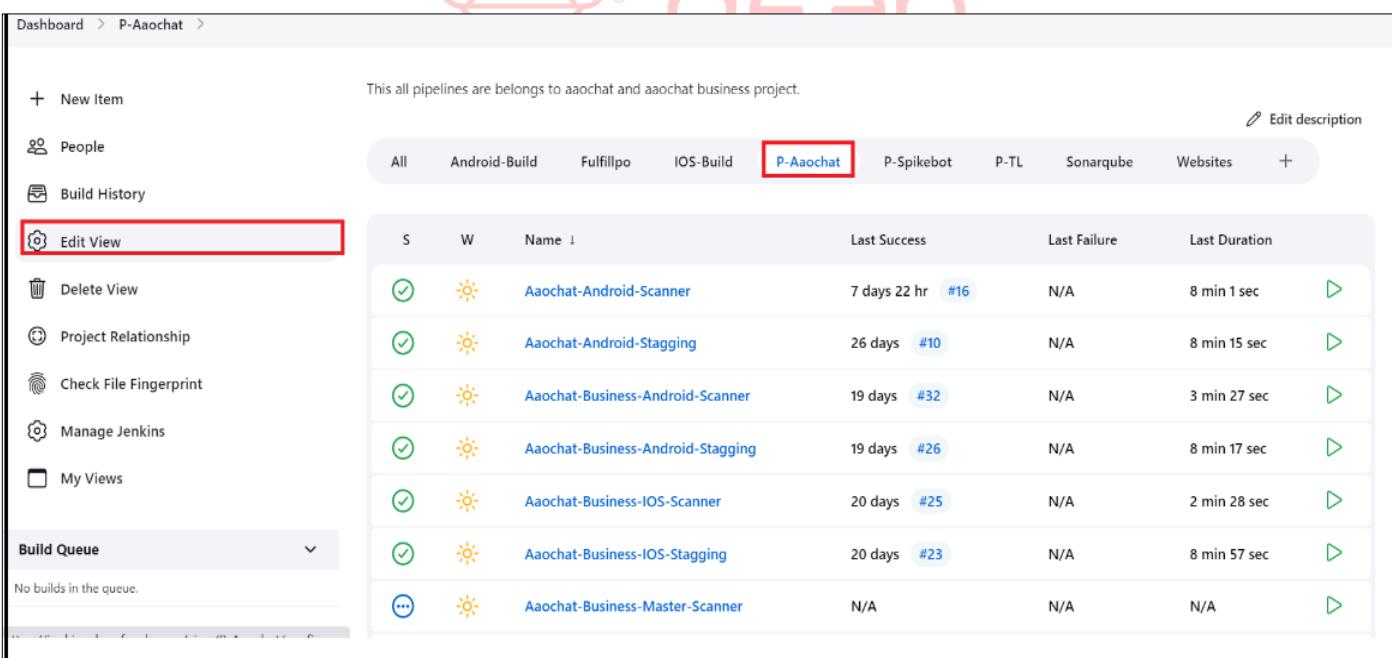


- Configure the View
- Now you'll see a configuration page where you can design your view according to your preferences.
- Under the "Job" section, you can configure which jobs to include in the view.
- Save the View
- Once all the configurations are done, click "Apply" and then "OK" to save the view.
- Now, when you go back to your Jenkins dashboard, you'll see the view you just created in the top menu. Click on it to see the jobs you've included



The screenshot shows the Jenkins 'Edit View' configuration page for a view named 'P-Aaochat'. In the 'Job Filters' section, the 'Jobs' dropdown is set to 'Jobs' and contains a list of Jenkins jobs. The 'OK' button at the bottom of the form is highlighted with a red box.

- You add jobs to a Jenkins view during the view's creation or modifying an existing view.
- On the Jenkins dashboard, click on the view you want to edit, then click on "Edit View" on the left side of the page.



S	W	Name	Last Success	Last Failure	Last Duration
		Aaochat-Android-Scanner	7 days 22 hr #16	N/A	8 min 1 sec
		Aaochat-Android-Staging	26 days #10	N/A	8 min 15 sec
		Aaochat-Business-Android-Scanner	19 days #32	N/A	3 min 27 sec
		Aaochat-Business-Android-Staging	19 days #26	N/A	8 min 17 sec
		Aaochat-Business-IOS-Scanner	20 days #25	N/A	2 min 28 sec
		Aaochat-Business-IOS-Staging	20 days #23	N/A	8 min 57 sec
		Aaochat-Business-Master-Scanner	N/A	N/A	N/A

- On the view configuration page, look for the "Job" section.
- Once you've added all the jobs you want, scroll to the bottom of the page and click "Apply", then "OK".

Dashboard > P-Aaochat > Configure

Check File Fingerprint Safe HTML [Preview](#)

Manage Jenkins Filter build queue

My Views Filter build executors

Build Queue Job Filters

No builds in the queue.

Recurse in subfolders

Build Executor Status

Node	Status	Idle
Built-in Node	Idle	1
Linux Slave	Idle	2
...

Jobs

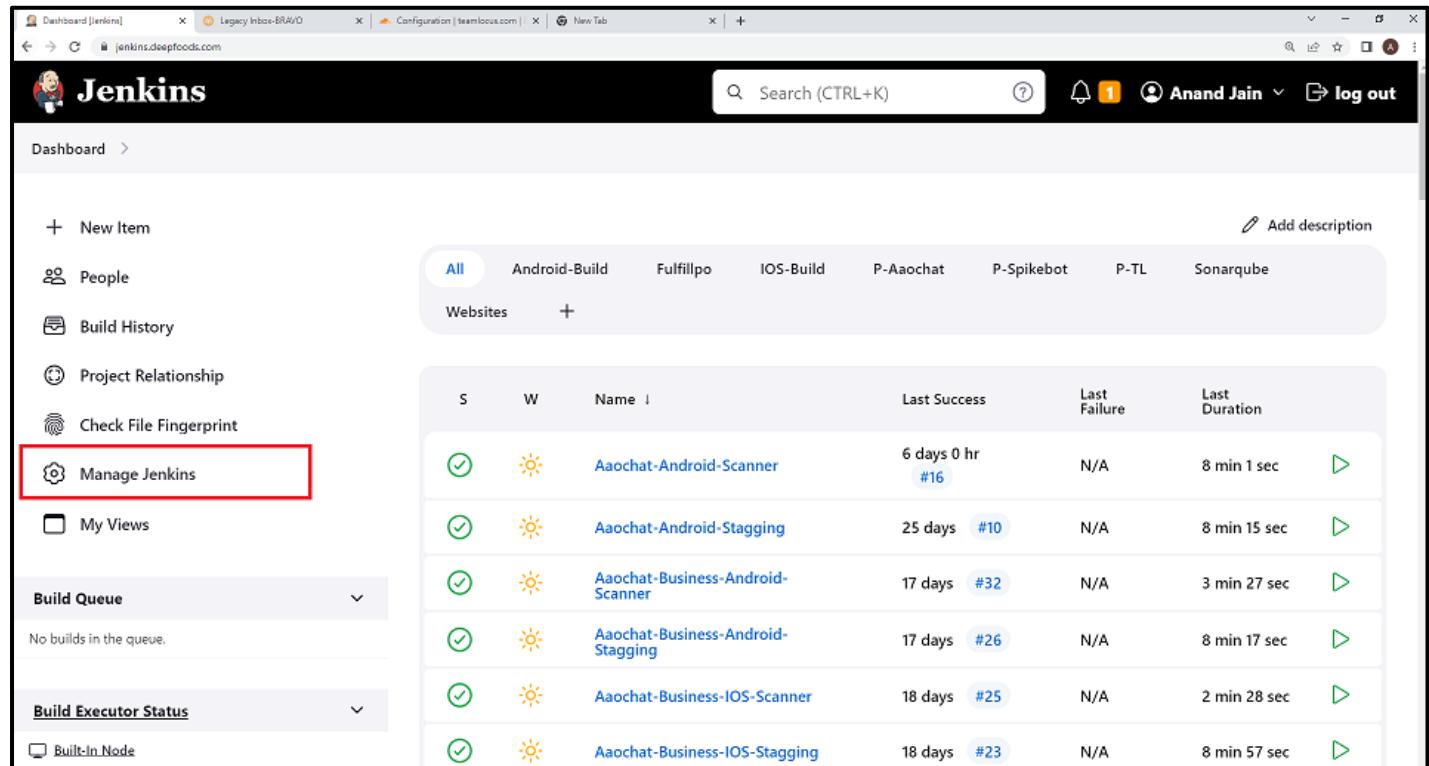
- Aaochat-ios-Scanner
- Aaochat-IOS-Stagging
- Aaochat-License-Panel
- Aaochat-License-Panel-Scanner
- Aaochat-Public-Site-Website
- Aaochat-Public-Site-Website-G
- Aaochat-Public-Site-Website-Scanner
- Aaochat-Web-Scanner
- Aaochat_Production

OK **Apply**



4. Manage Jenkins

The "Manage Jenkins" tab in Jenkins is a crucial part of the Jenkins dashboard that allows administrators to manage and configure the Jenkins instance. Here are the functions available on this tab:



The screenshot shows the Jenkins dashboard with the "Manage Jenkins" link highlighted by a red box. The dashboard includes a sidebar with links like "New Item", "People", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins" (highlighted), "My Views", "Build Queue" (with "No builds in the queue." message), and "Build Executor Status". The main area shows a table of build jobs with columns: S, W, Name, Last Success, Last Failure, and Last Duration. The table lists six jobs under the "Aaochat" project.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	Aaochat-Android-Scanner	6 days 0 hr #16	N/A	8 min 1 sec ➔
✓	☀️	Aaochat-Android-Stagging	25 days #10	N/A	8 min 15 sec ➔
✓	☀️	Aaochat-Business-Android-Scanner	17 days #32	N/A	3 min 27 sec ➔
✓	☀️	Aaochat-Business-Android-Stagging	17 days #26	N/A	8 min 17 sec ➔
✓	☀️	Aaochat-Business-IOS-Scanner	18 days #25	N/A	2 min 28 sec ➔
✓	☀️	Aaochat-Business-IOS-Stagging	18 days #23	N/A	8 min 57 sec ➔

Configure System: This is the main system configuration page where global configurations take place. Here, you set system-wide parameters such as Environment Variables, JDK installations, Ant installations, Maven installations, etc.

Configure Tools: Adding tools to Jenkins can be done via the "Global Tool Configuration" page in "Manage Jenkins". Here's how to add a few common tools, like Git, JDK, and Maven

Configure Global Security: It's dedicated to secure Jenkins; authentication, access control can be configured from this section.

Configure Credentials: Here you can manage all credentials for your jobs in one place.

System Information: This section provides information about the Jenkins environment. You can get a complete list of the system properties available on Jenkins.

Manage Plugins: This part lets you add, remove, disable, or enable any of the Jenkins plugins. It's also where you can update an existing plugin or perform an update for all the plugins installed.

System Log: It shows the Jenkins logs and other log records.

Load Statistics: It shows utilization statistics for your Jenkins setup including Executors, Queue length, and Memory usage.

Manage Nodes and Clouds: This section allows you to add and delete build nodes and manage your Cloud agents, if applicable.

Reload Configuration from Disk: If you've made direct changes to Jenkins' configuration files, you can apply those changes with this function without needing to restart the entire Jenkins instance.

Jenkins CLI: This option provides a downloadable CLI (Command Line Interface) JAR file allowing you to interact with Jenkins from a command line.

Script Console: An advanced feature allowing administrators to run arbitrary Groovy scripts on the Jenkins master.

About Jenkins: Information about the precise version of Jenkins you have installed.

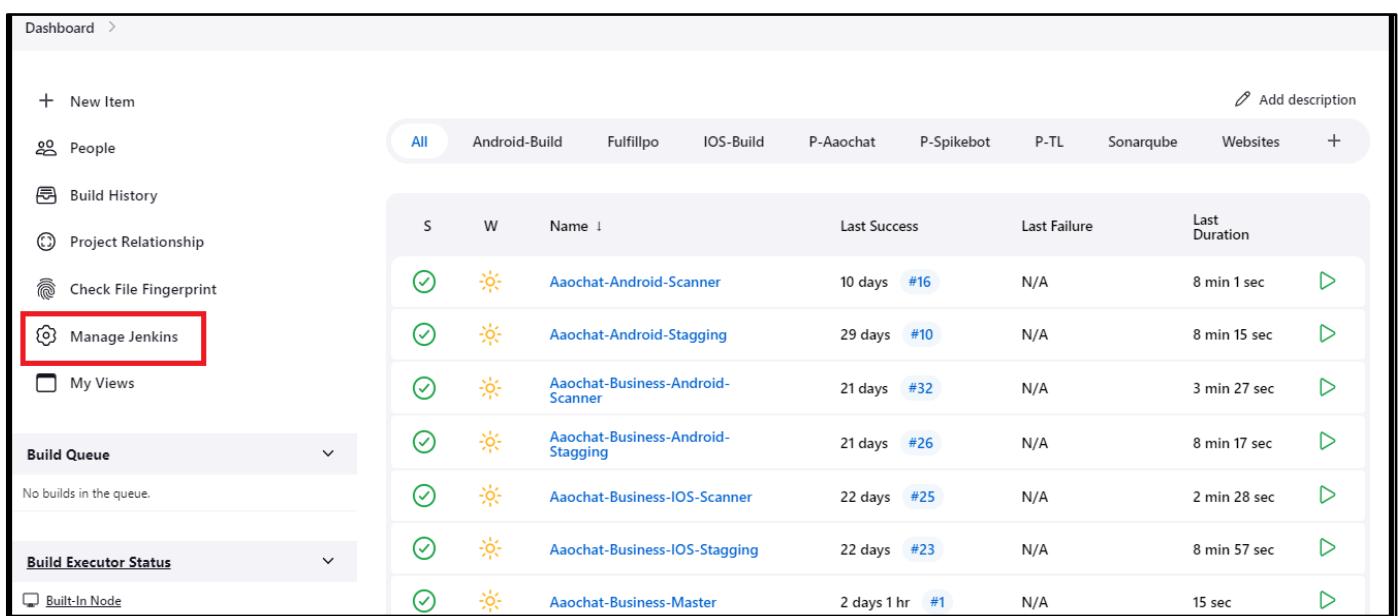
Each section has its specific role in the set-up and maintenance of your Jenkins instance. Note that access to this tab is usually limited to Jenkins administrators.



4.1 System Configuration

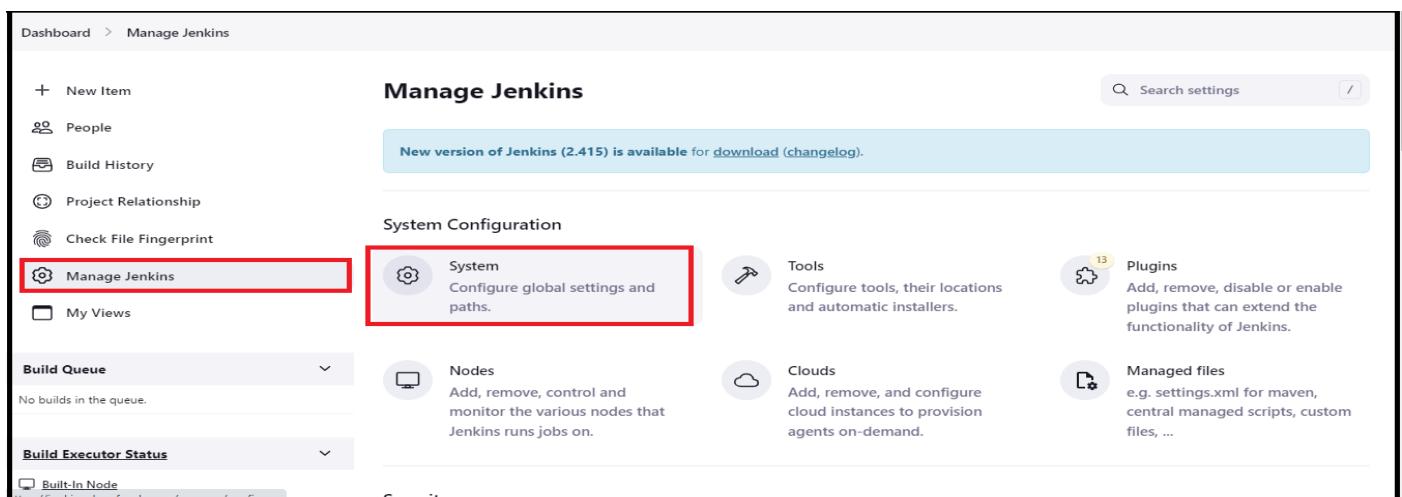
Prerequisites

1. Jenkins server
 2. You can configure system settings in Jenkins through the "Manage Jenkins" section which provides many configuration options.
- Log in to Jenkins and click on "Manage Jenkins" in the left-hand sidebar.



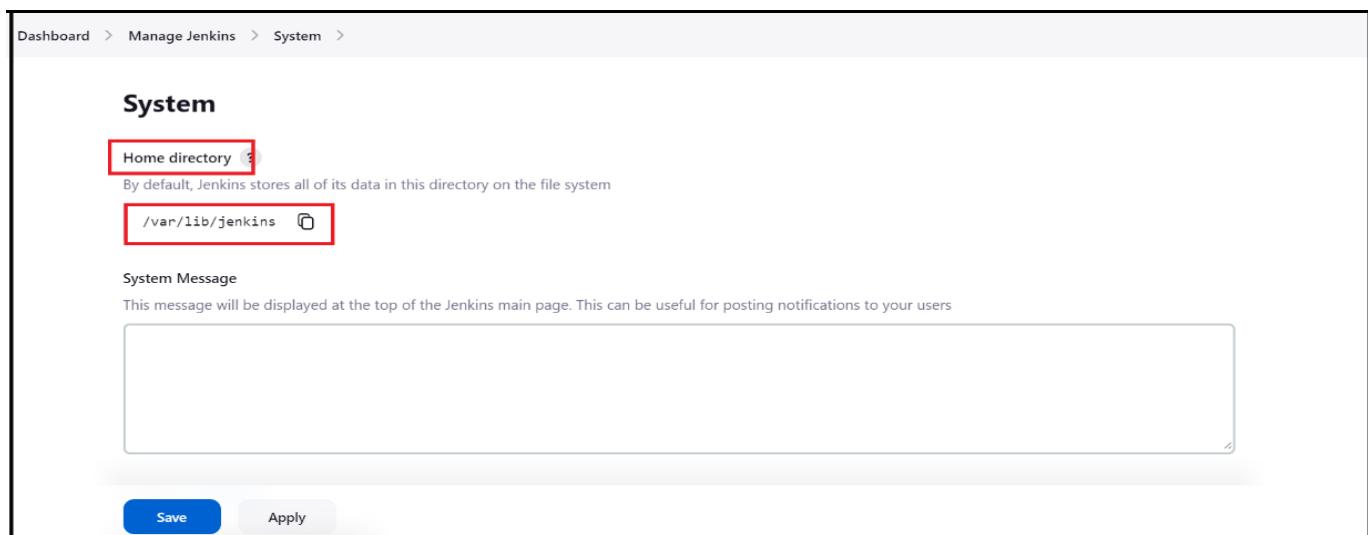
The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with various links: 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (which is highlighted with a red box), and 'My Views'. Below these are dropdown menus for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing 'Built-In Node'). The main area has tabs for 'All', 'Android-Build', 'Fulfillpo', 'IOS-Build', 'P-Aaochat', 'P-Spikebot', 'P-TL', 'Sonarqube', and 'Websites'. To the right is a table listing Jenkins jobs, each with a green checkmark icon and a sun icon. The columns are: S (Status), W (Last Success), Name, Last Success, Last Failure, Last Duration, and a 'More' button. The jobs listed are: Aaochat-Android-Scanner, Aaochat-Android-Staging, Aaochat-Business-Android-Scanner, Aaochat-Business-Android-Staging, Aaochat-Business-IOS-Scanner, Aaochat-Business-IOS-Staging, and Aaochat-Business-Master.

- In the "Manage Jenkins" page, click on the "Configure System" option. From this section, you can set up a range of different options. Some common configurations include:
- In the "Manage Jenkins" page, click on the "System" option. From this section, you can set up a range of different options. Some common configurations include:



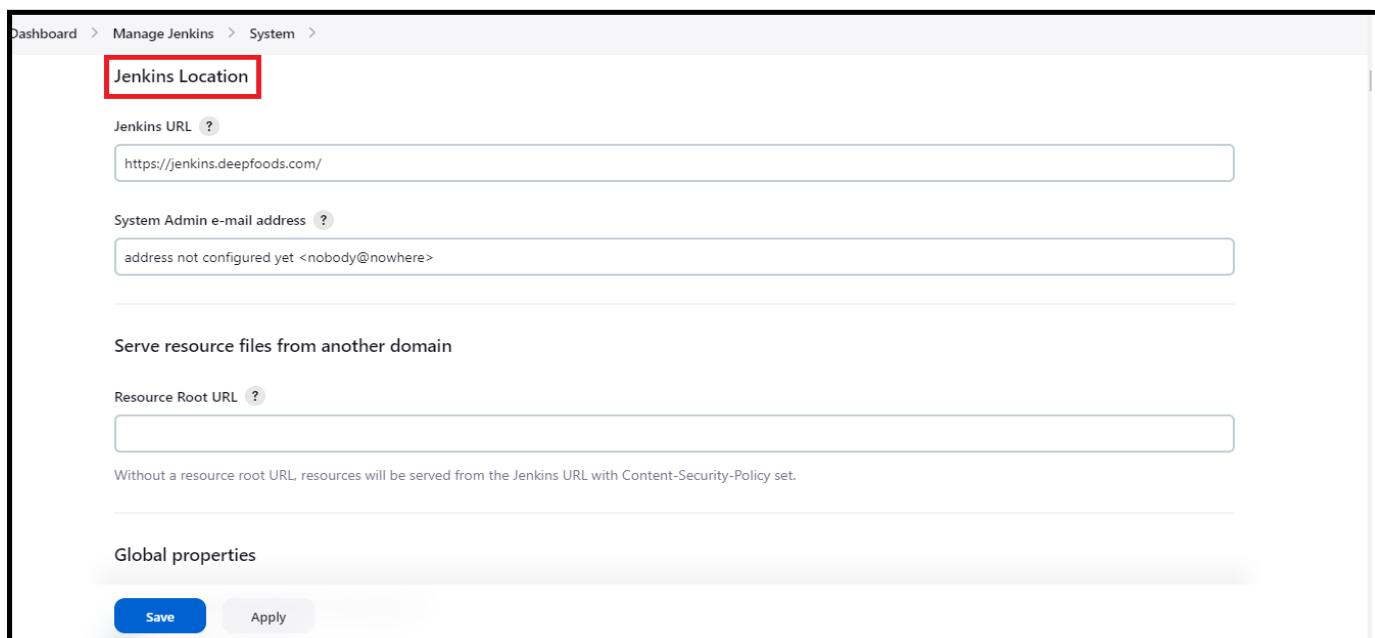
The screenshot shows the 'Manage Jenkins' page. The sidebar includes 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (highlighted with a red box), and 'My Views'. Below these are dropdown menus for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing 'Built-In Node'). The main content area has a 'Manage Jenkins' header and a message about a new Jenkins version available for download. It then shows the 'System Configuration' section with several options: 'System' (highlighted with a red box), 'Tools', 'Nodes', 'Clouds', 'Plugins' (with a count of 13), and 'Managed files'. Each option has a brief description and a link to its configuration page.

- **Jenkins Home Location:** Configure the Jenkins home directory.
- **System Message:** Edit the message that will be displayed on the top page.
- **Number of Executors:** Configure the maximum number of concurrent builds that Jenkins may perform.
- **JDK Installations, Git Installations, and Maven Installations:** Enables Jenkins to automatically install these tools when a job requires them.
- **E-mail Notification:** Configure Jenkins to send emails notifications for build results. You will need an SMTP server for this.
- **Quiet Period:** Set the number of seconds a newly triggered build waits in the queue before actual build starts.



The screenshot shows the Jenkins System configuration page. The 'Home directory' input field is highlighted with a red box. Below it, a note says 'By default, Jenkins stores all of its data in this directory on the file system' with a value of '/var/lib/jenkins'. There is also a 'System Message' section with a text area and a note about displaying messages at the top of the main page. At the bottom are 'Save' and 'Apply' buttons.

- **Jenkins Location:** Set the Jenkins URL and system-administrator email address.



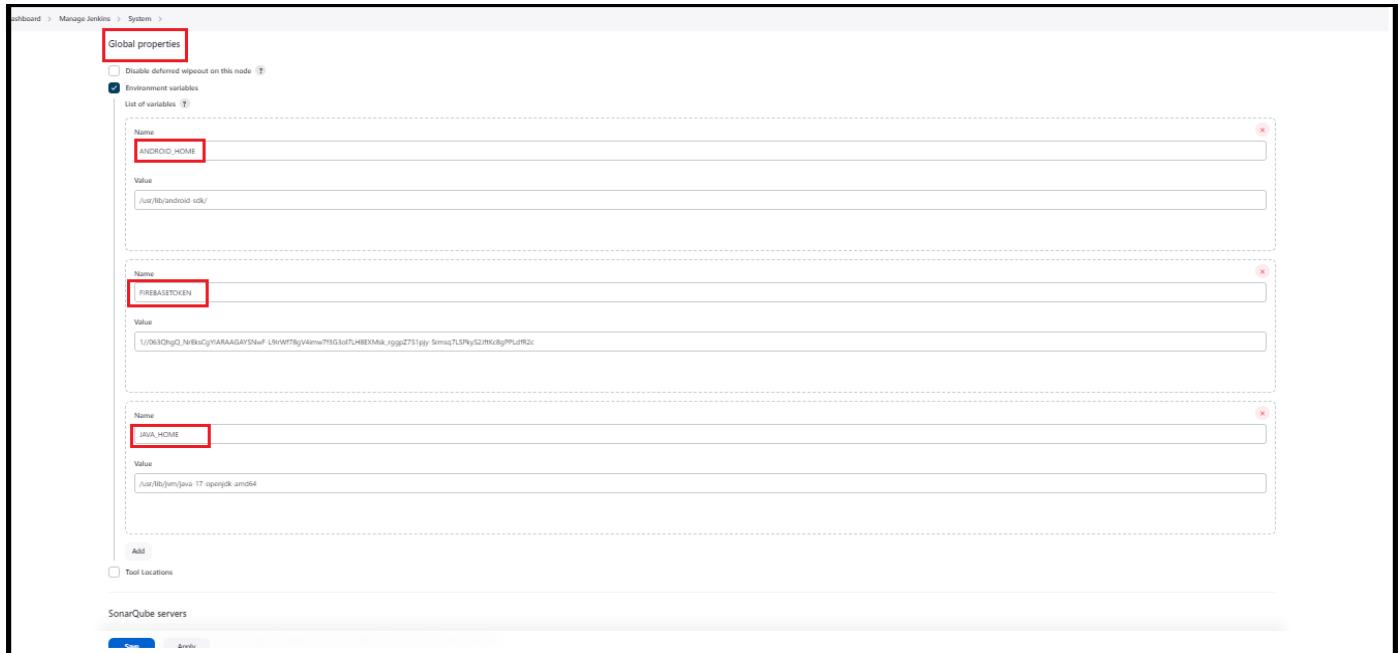
The screenshot shows the Jenkins System configuration page with the 'Jenkins Location' section highlighted. It includes fields for 'Jenkins URL' (set to https://jenkins.deepfoods.com/) and 'System Admin e-mail address' (set to address not configured yet <nobody@nowhere>). Below these are sections for serving resources from another domain ('Resource Root URL') and global properties. At the bottom are 'Save' and 'Apply' buttons.

- **Environment variables**

- Add any global environment variables which will apply to all Jenkins jobs.

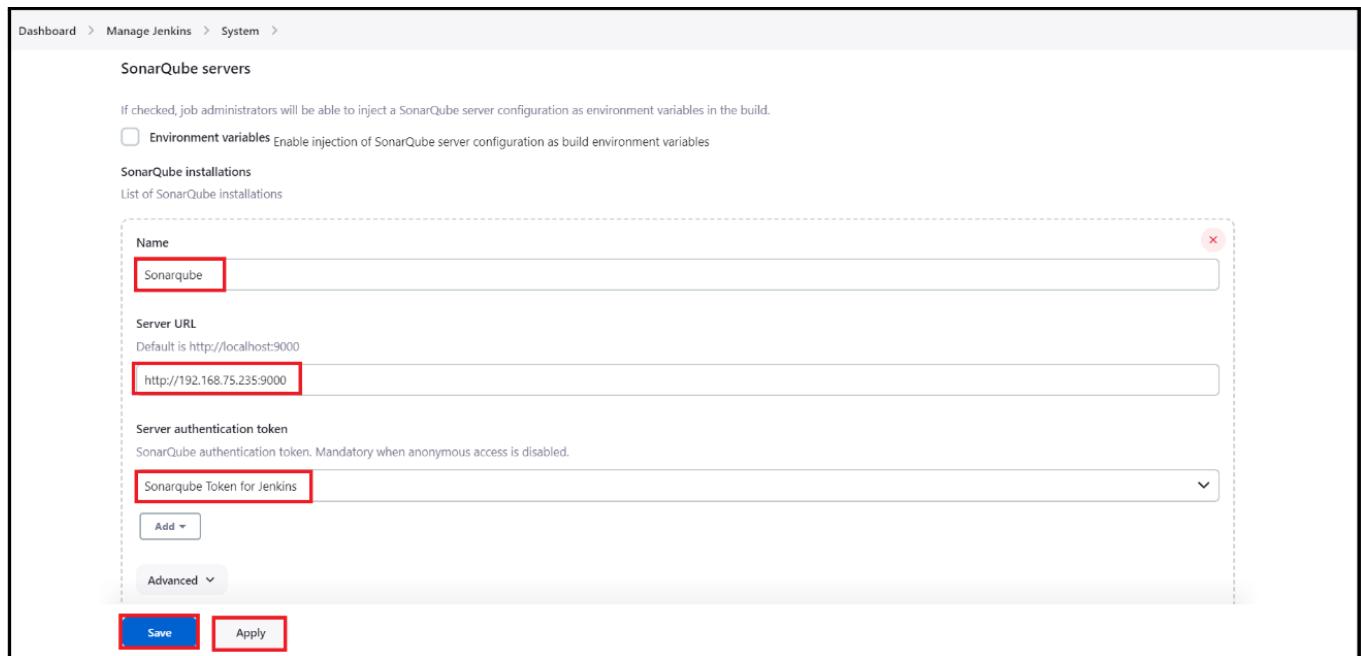
List of variables

1. ANDROID_HOME
2. FIREBASETOKEN
3. JAVA_HOME



Adding Sonarqube Server

- Create a VM for Sonarqube (ip: 192.168.75.235)
- Once the Server installation is finished, navigate to 'Manage Jenkins' > 'System'.
- Click on the 'Add SonarQube' button.
- Provide a unique name for the server in the 'Name' field.
- In the 'Server URL' field, enter the URL where SonarQube server is running.
- SonarQube servers normally require an authentication token. The token can be generated from the SonarQube server itself from "User" > "My Account" > "Security".
- Copy this token and paste it in Jenkins. For that, in the same SonarQube Server configuration in Jenkins, click on the 'Add' button next to 'Server authentication token'.
- Paste the token in 'Secret' field. Give it a unique id and optional description. Click 'Add'.
- Select the newly added credentials in 'Server authentication token' dropdown in SonarQube server configuration.
- Click 'Apply' and 'Save' at the bottom of the page to save the changes.



SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables Enable injection of SonarQube server configuration as build environment variables

SonarQube installations

List of SonarQube installations

Name	Actions
Sonarqube	X

Name
Sonarqube

Server URL
Default is <http://localhost:9000>
http://192.168.75.235:9000

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.
Sonarqube Token for Jenkins

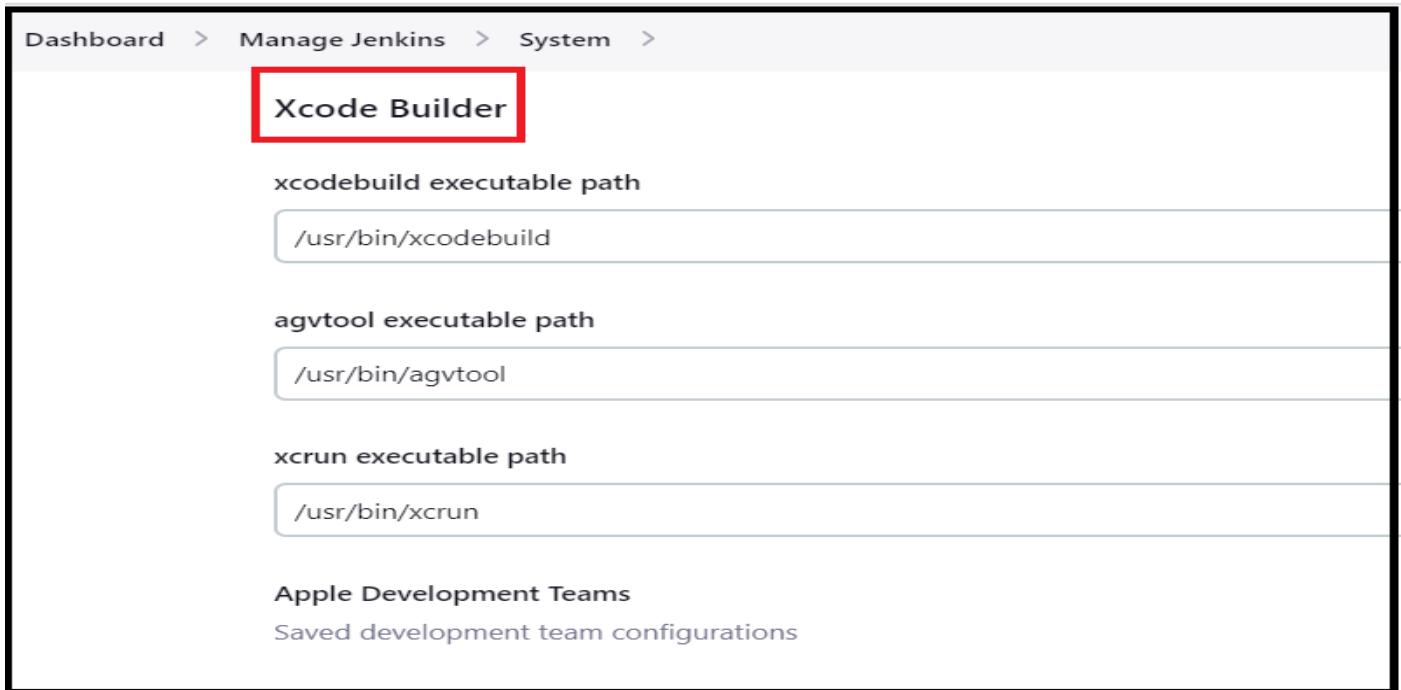
Add ▾

Advanced ▾

Save **Apply**

Xcode

- From your Jenkins dashboard, navigate to 'Manage Jenkins' > 'Manage Plugins'.
- Install Xcode Plugin
- Scroll down to find the 'Xcode' section.
- In the "Xcode Directory" field, specify the path where Xcode is installed on your build machine. (/usr/bin/xcodebuild)
- Specify the path of agvtool executable path (/usr/bin/agvtool)
- Specify the path of xcrun executable path (/usr/bin/xcrun)
- Add Apple Development Teams
 - Anuj Parikh
 - Naineshkumar Patel
 - Deep Foods Inc



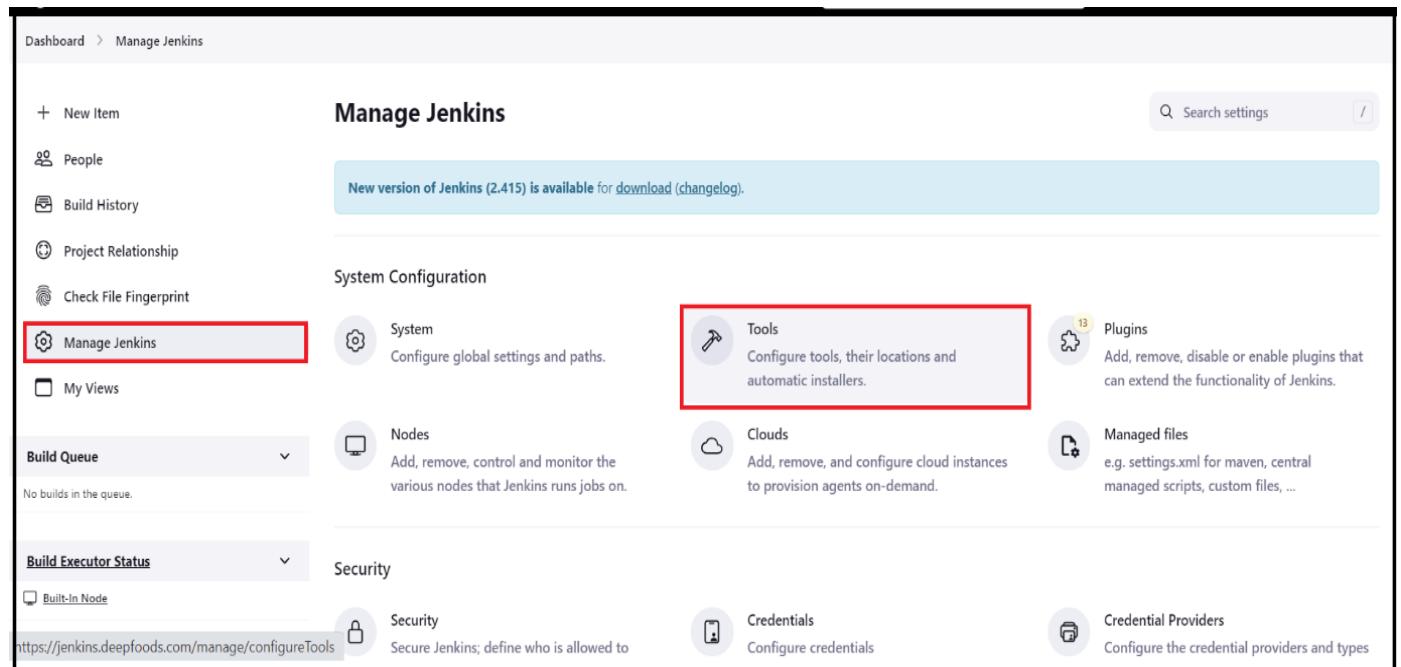
The screenshot shows the Jenkins System configuration page. At the top, the breadcrumb navigation is "Dashboard > Manage Jenkins > System". A red box highlights the "Xcode Builder" section. Below it, there are three configuration items:

- xcodebuild executable path**: Set to `/usr/bin/xcodebuild`.
- agvtool executable path**: Set to `/usr/bin/agvtool`.
- xcrun executable path**: Set to `/usr/bin/xcrun`.

Below these, under "Apple Development Teams", it says "Saved development team configurations".

4.2 Tools Configuration

- Adding tools to Jenkins can be done via the "Tools Configuration" page in "Manage Jenkins". Here's how to add a few common tools, like Git, JDK, and Maven
- Log into Jenkins. Click "Manage Jenkins" from the main dashboard, and then select "Tools Configuration"



The screenshot shows the Jenkins "Manage Jenkins" page. The left sidebar has a red box around the "Manage Jenkins" item. The main content area is titled "Manage Jenkins" and includes a message about a new version available for download. The "System Configuration" section is expanded, showing several options:

- Tools**: Configure tools, their locations and automatic installers. This option has a red box around it.
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins. It shows 13 available plugins.
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Managed files**: e.g. settings.xml for maven, central managed scripts, custom files, ...
- Credentials**: Configure credentials.
- Credential Providers**: Configure the credential providers and types.

At the bottom left, there is a link to "https://jenkins.deepfoods.com/manage/configureTools".



The screenshot shows the Jenkins 'Tools' configuration page under 'Manage Jenkins'. The 'Maven Configuration' section is highlighted with a red box. It contains fields for 'Default settings provider' (set to 'Use default maven settings') and 'Default global settings provider' (set to 'Use default maven global settings'). Below this is the 'JDK installations' section, which is also highlighted with a red box. It shows a table with one row, 'Java', and buttons for 'Save' and 'Apply'.

4.2.1 Add JDK

- Scroll down to the "JDK" section and click "Add JDK".
- Uncheck the "Install automatically" checkbox (keep it checked if you want Jenkins to automatically install JDK when needed).
- Give the JDK a name and specify "JAVA_HOME". JAVA_HOME should point to the location where JDK is installed on the Jenkins build agent machine.



The screenshot shows the 'Add JDK' dialog box. The 'JDK installations' section is highlighted with a red box. Inside, the 'Add JDK' button is highlighted with a red box. The 'JDK Name' field contains 'Java' and the 'JAVA_HOME' field contains '/usr/lib/jvm/java-17-openjdk-amd64/'. A checkbox for 'Install automatically' is present but unchecked. Buttons for 'Save' and 'Apply' are at the bottom.

4.2.2 Add Git

- Scroll down to the "Git" section and click on "Add Git".
- Give it a name (or leave the default name, which is the Git version) and, if necessary, adjust the path to the Git executable.



Dashboard > Manage Jenkins > Tools

JDK installations

JDK installations Edited

Git installations

Git

Name: Default

Path to Git executable: git

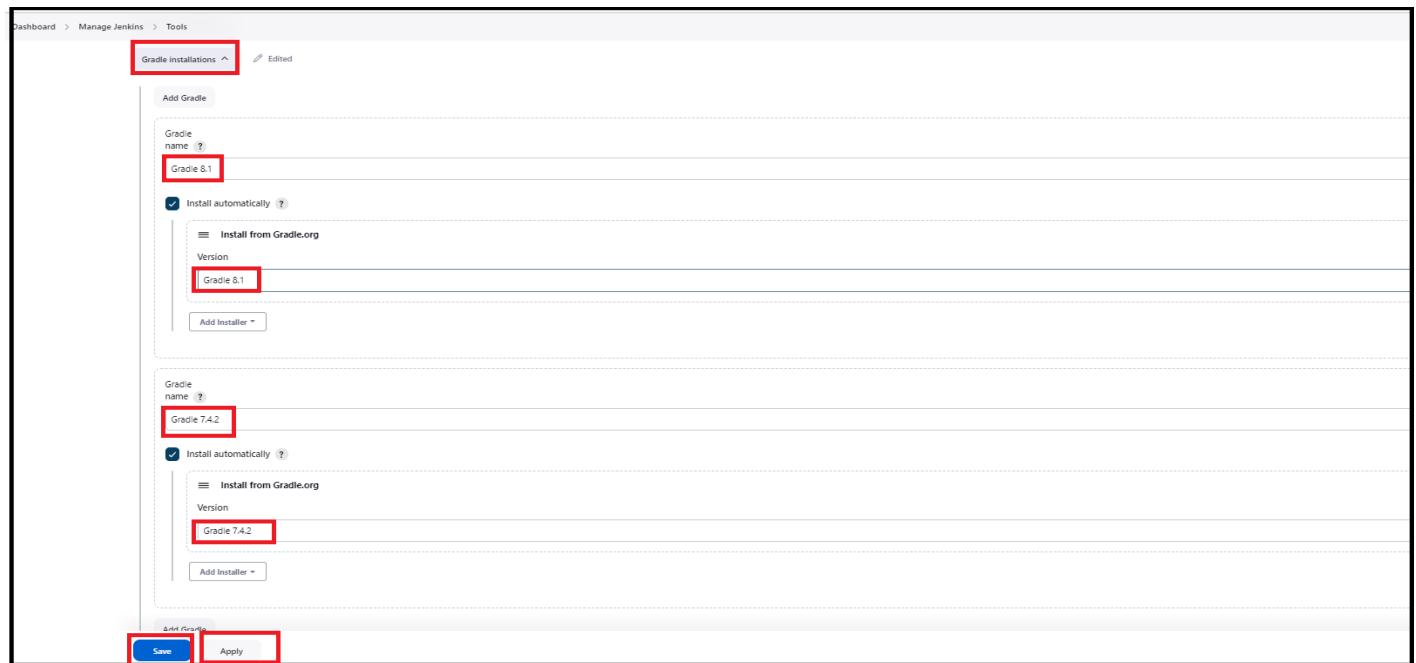
Install automatically

Add Git

Save **Apply**

4.2.3 Gradle installation

- In the 'Manage Jenkins' menu, select 'Tool Configuration'.
- Scroll down to the 'Gradle' section. Click on the 'Add Gradle' button.
- Provide a 'Name' for your Gradle installation. You can leave the 'Install automatically' option checked if you want Jenkins to automatically install Gradle when it is needed.
- If 'Install automatically' is checked, you can select the version of Gradle to install from the 'Install from Gradle Version' dropdown. If 'Install automatically' is not checked, you will have to provide the 'GRADLE_HOME' manually.
- Click 'Apply' and then 'Save' at the bottom of the page to save your configuration..



Dashboard > Manage Jenkins > Tools

Gradle installations

Add Grade

Gradle 8.1

Install automatically

Install from Gradle.org

Version: Gradle 8.1

Add Installer

Gradle 7.4.2

Install automatically

Install from Gradle.org

Version: Gradle 7.4.2

Add Installer

Save **Apply**

4.2.4 Add Maven

- Scroll down to the "Maven" section and click "Add Maven".
- Check the "Install automatically" checkbox.
- Click "Apply" and then "Save".

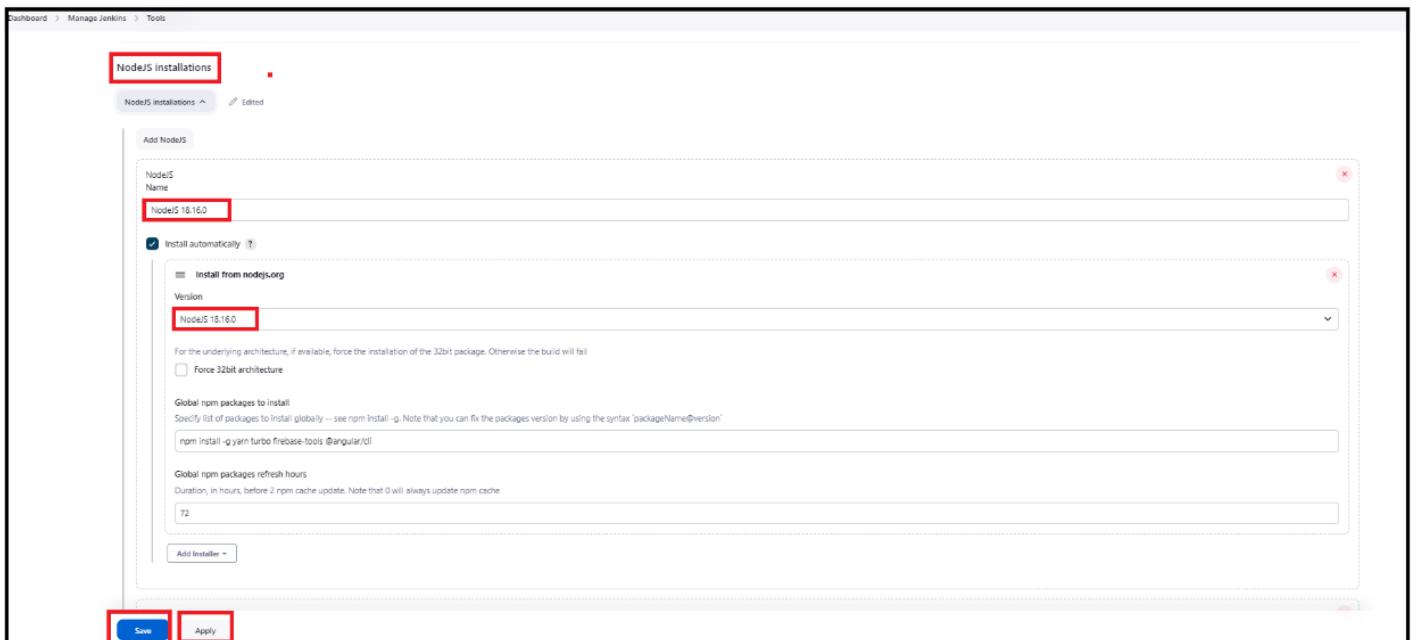


4.2.5 Add Node.js



- To add Node.js in Jenkins, you will need to install the NodeJS Plugin first.
- Install NodeJS Plugin
- Once the plugin is installed, go back to 'Manage Jenkins' > 'Global Tool Configuration'.
- Scroll down to the 'NodeJS' section.
- Click on the 'Add NodeJS' button.
- Choose a name for your Node.js installation (e.g., NodeJS 18.16.0).
- If you leave the 'Install automatically' option selected, Jenkins will download and install Node.js automatically from nodejs.org based on the configuration you set.
- You can also select any global Node.js packages to install, such as npm, angular-cli, typescript, gulp, grunt, etc, by clicking on the 'Add' button in 'Global npm packages to install' section.
- Finally, click on 'Apply' and 'Save'.
- Now, when configuring a job, you can add a 'Build Environment' step and choose 'Provide Node & npm environment', then select the name of the Node.js installation you added.
- If you plan to run your Jenkins jobs on several different agents, Node.js must be installed on each one of them, or they must be able to install it automatically.
- Node.js, we installed
 1. NodeJS 19.9.0 (Default)
 2. NodeJS 16.20.0
 3. NodeJS 12.18.3

4. NodeJS 16.14.0
5. NodeJS 18.16.20



4.3 NODE

Jenkins Master and Slave configuration

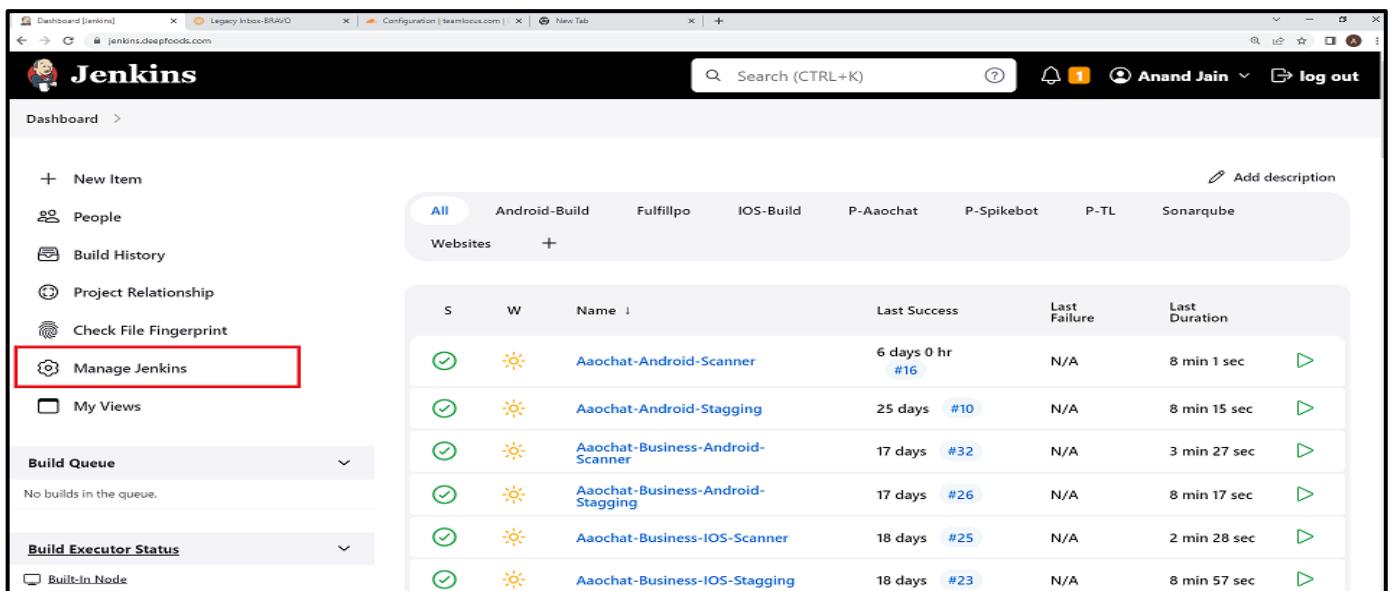
Prerequisites

- Jenkins server
- Slave server with Java installation

Procedure:

Navigate to the Jenkins dashboard.

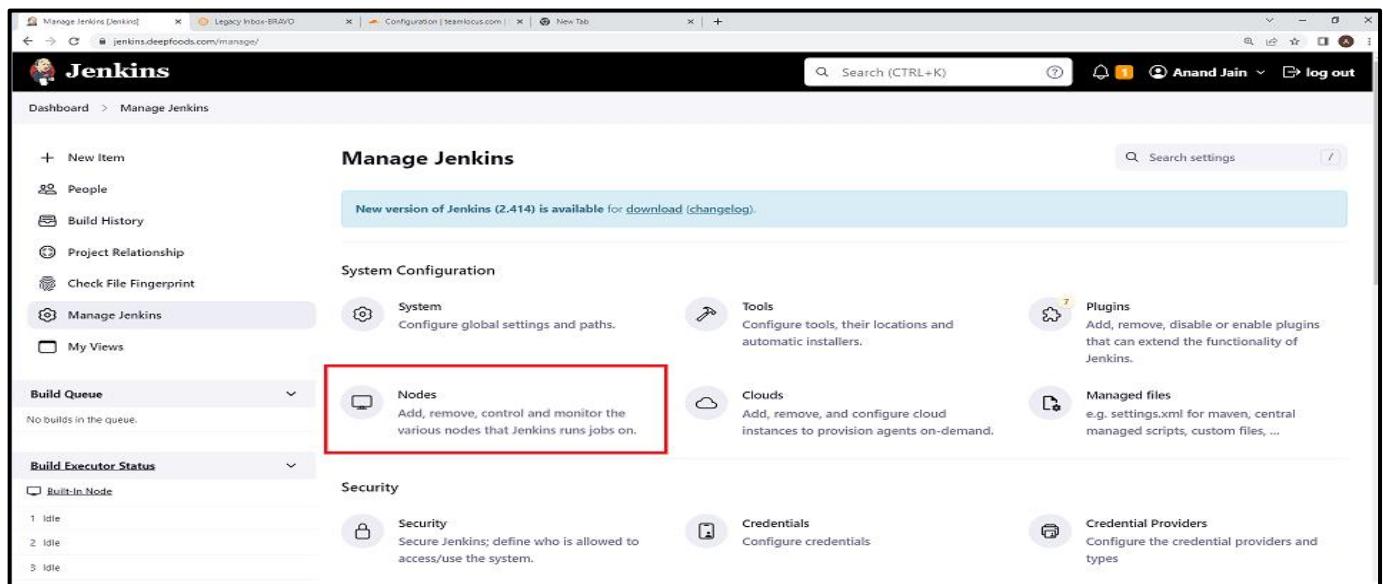
- Click on "Manage Jenkins" in the left navigation menu.



The screenshot shows the Jenkins dashboard. On the left sidebar, under 'Manage Jenkins', the 'Manage Jenkins' link is highlighted with a red box. The main area displays a table of build jobs with columns for Status, Last Success, Last Failure, and Last Duration.

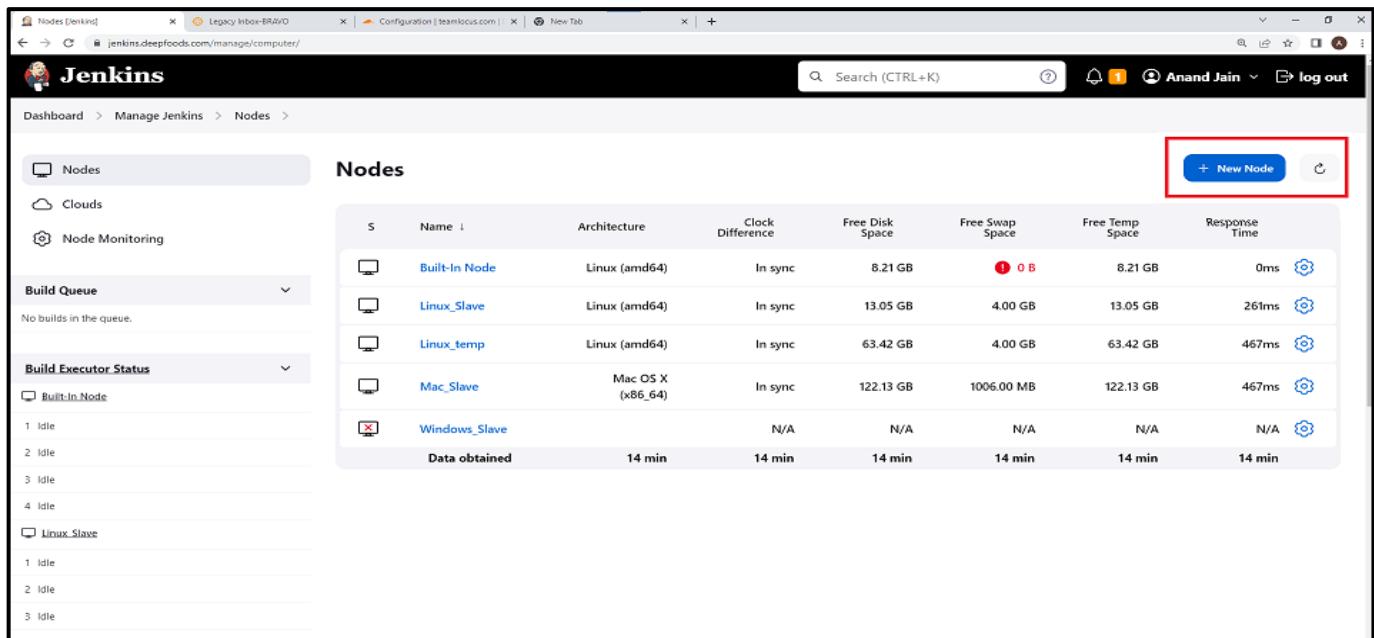
S	W	Name	Last Success	Last Failure	Last Duration
		Aaochat-Android-Scanner	6 days 0 hr #16	N/A	8 min 1 sec
		Aaochat-Android-Staging	25 days #10	N/A	8 min 15 sec
		Aaochat-Business-Android-Scanner	17 days #32	N/A	3 min 27 sec
		Aaochat-Business-Android-Staging	17 days #26	N/A	8 min 17 sec
		Aaochat-Business-IOS-Scanner	18 days #25	N/A	2 min 28 sec
		Aaochat-Business-IOS-Staging	18 days #23	N/A	8 min 57 sec

- Click on "Manage Nodes and Clouds".



The screenshot shows the 'Manage Jenkins' page. Under 'System Configuration', the 'Nodes' link is highlighted with a red box. The page also displays sections for Tools, Plugins, Security, Clouds, Managed files, and Credential Providers.

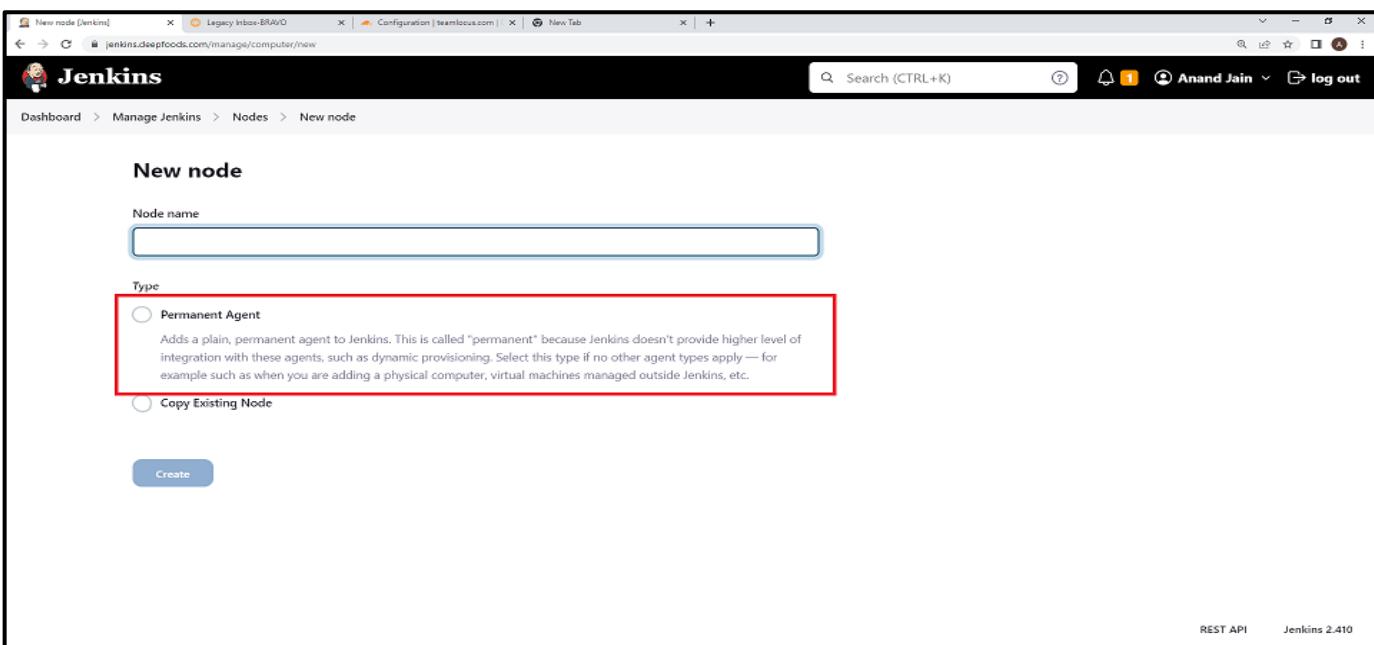
- Click on "New Node".



The screenshot shows the Jenkins 'Nodes' page. On the left, there's a sidebar with links for 'Nodes', 'Clouds', 'Node Monitoring', 'Build Queue' (which says 'No builds in the queue'), and 'Build Executor Status'. Under 'Build Executor Status', it shows 'Built-In_Node' with 4 idle instances and 'Linux_Slave' with 3 idle instances. The main area is titled 'Nodes' and lists the following agents:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	8.21 GB	1 0 B	8.21 GB	0ms
	Linux_Slave	Linux (amd64)	In sync	13.05 GB	4.00 GB	13.05 GB	261ms
	Linux_temp	Linux (amd64)	In sync	63.42 GB	4.00 GB	63.42 GB	467ms
	Mac_Slave	Mac OS X (x86_64)	In sync	122.13 GB	1006.00 MB	122.13 GB	467ms
	Windows_Slave		N/A	N/A	N/A	N/A	N/A
	Data obtained		14 min	14 min	14 min	14 min	14 min

- Give your new agent a name, select "Permanent Agent" and click "OK".



The screenshot shows the 'New node' creation page. It has a 'Node name' input field and a 'Type' section. The 'Type' section contains two options: 'Permanent Agent' (selected) and 'Copy Existing Node'. A red box highlights the 'Permanent Agent' option. Below it, a tooltip provides a description: 'Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply—for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.' At the bottom is a 'Create' button.

You will now be asked to configure the agent. Here are the details you need to fill in:

Description: A description of the agent.

of executors: Number of jobs that the agent can execute simultaneously.

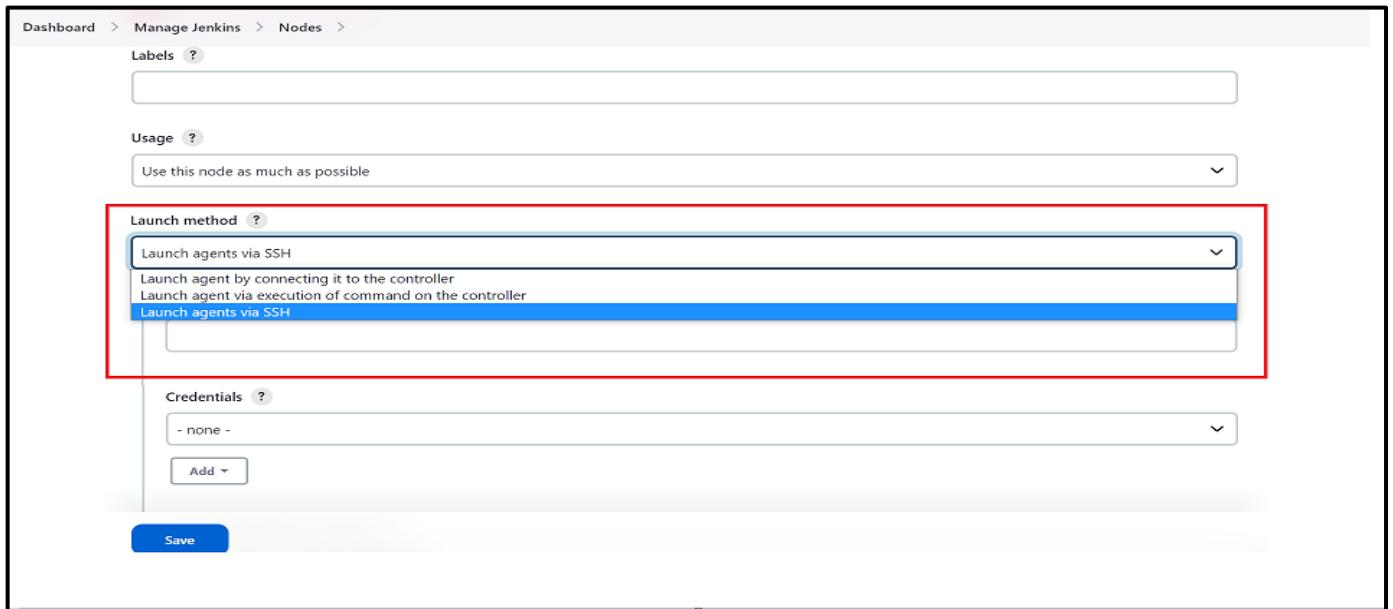
Remote root directory: An absolute path of a directory on the agent where Jenkins places its files.

Labels: Space-separated tags used to assign jobs to agents.

Next, in the 'Launch method' field, choose "Launch agent via SSH".

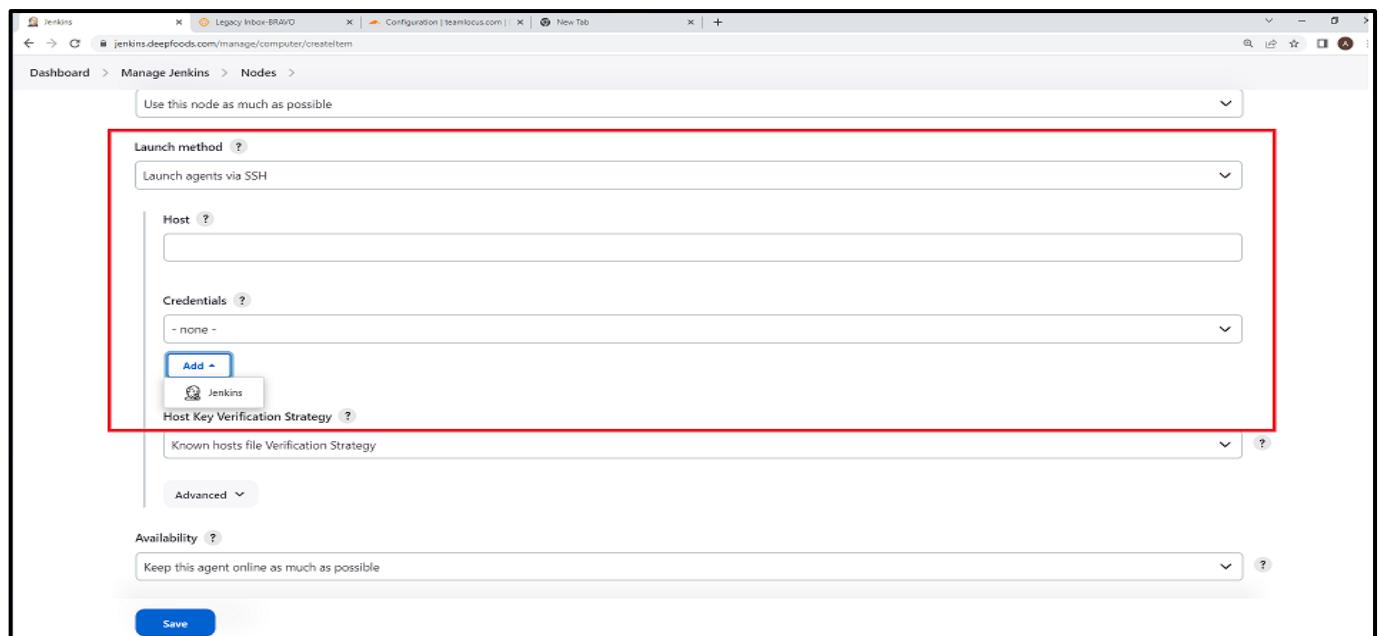
Launch method:

- Launch agent by connecting it to the controller.
- Launch agent via execution of command on the controller.
- Launch agent via SSH.



The screenshot shows the Jenkins 'Manage Jenkins > Nodes > New Node' configuration page. The 'Launch method' dropdown is highlighted with a red box. The selected option is 'Launch agents via SSH'. Other options in the dropdown are 'Launch agent by connecting it to the controller' and 'Launch agent via execution of command on the controller'. Below the dropdown, there are sections for 'Labels', 'Usage', 'Credentials', and a 'Save' button.

- Step to connect slave as **Launch agent via SSH**.



The screenshot shows the Jenkins 'Manage Jenkins > Nodes > New Node' configuration page. The 'Host' section and the 'Advanced' section are highlighted with a red box. The 'Host' section contains fields for 'Host' and 'Credentials'. The 'Advanced' section contains a 'Host Key Verification Strategy' dropdown set to 'Known hosts file Verification Strategy'. Below these sections are 'Availability' and a 'Save' button.

You will need to add further details for the SSH configuration:

- **Host:** IP address/hostname of the agent.

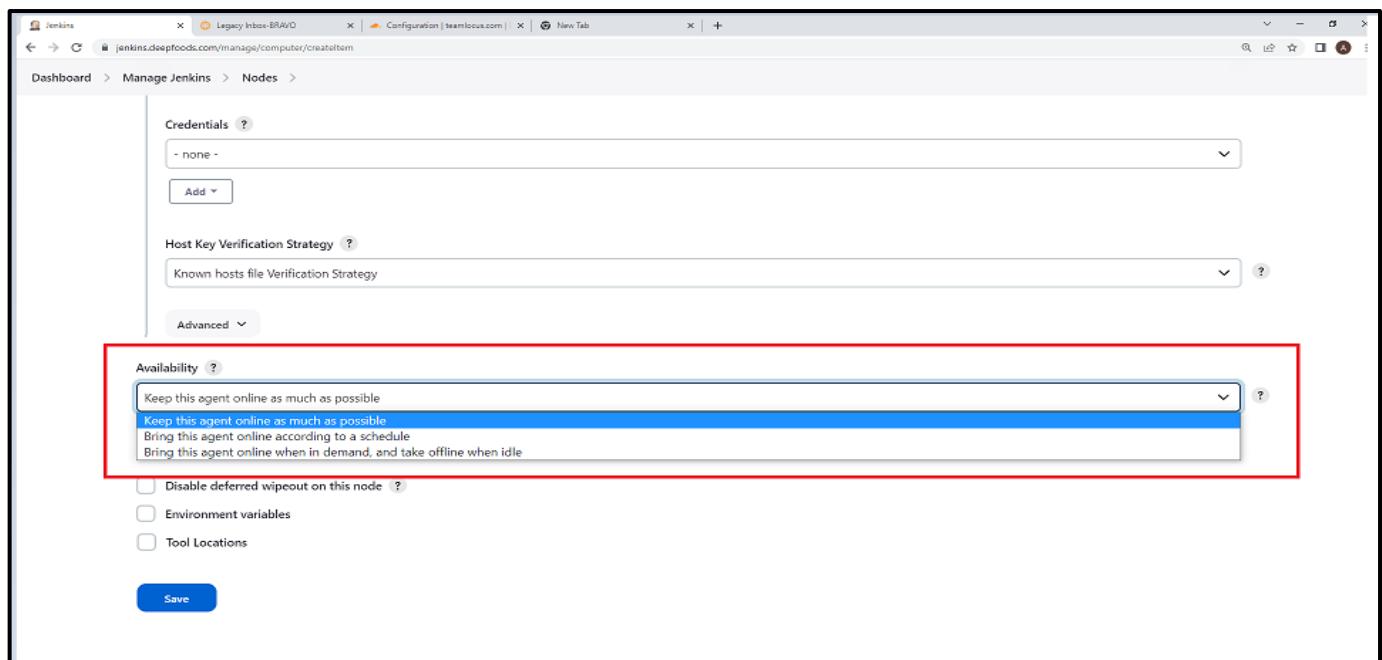
- **Credentials:** Add the agent's SSH credentials here.

Click on "Add", then "Jenkins". Input username, and then you can choose either password or private key for SSH authentication.

- **Host Key Verification Strategy:** Strategy used to verify the agent's SSH key. For a secure connection, use "**Known hosts file Verification Strategy**" or "**Manually provided key Verification Strategy**".

- **Availability:**

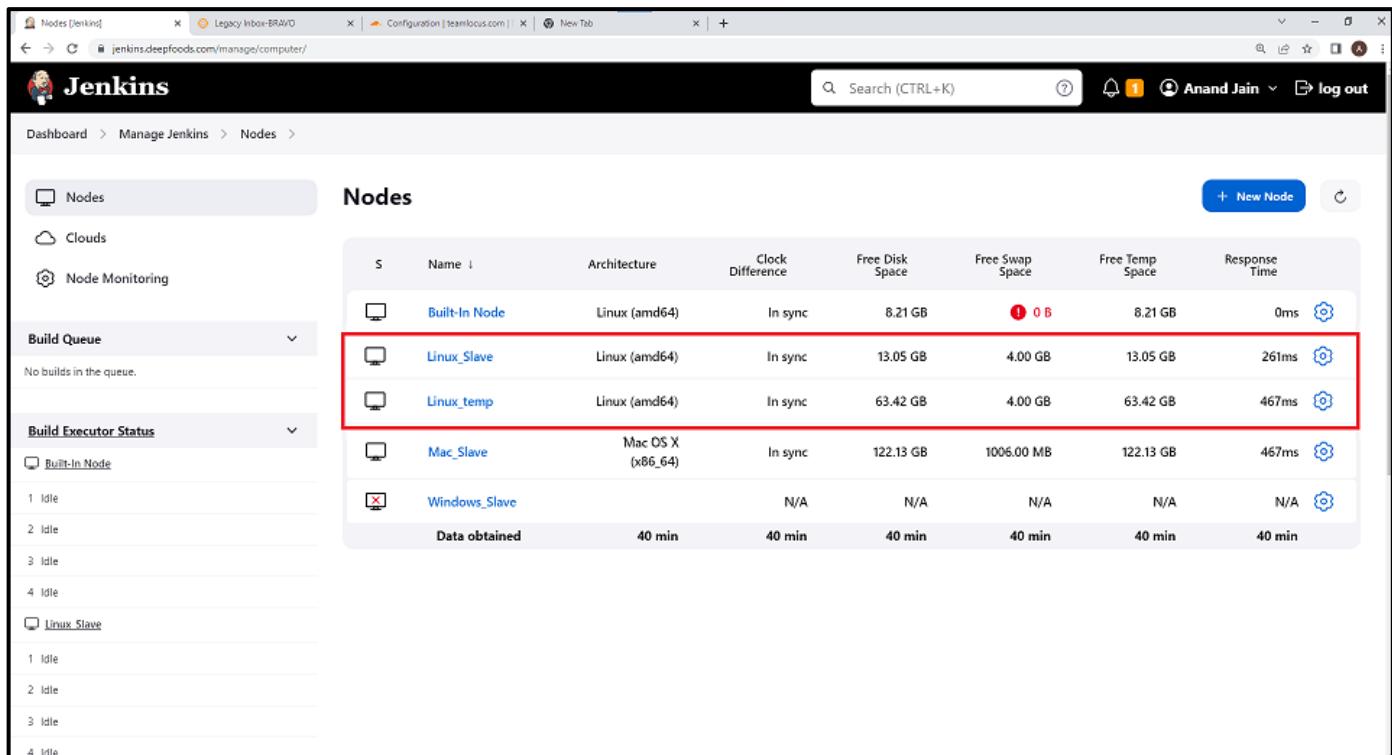
- Keep this agent online as much as possible
- Bring this agent online according to a schedule
- Bring this agent online when in demand, and take offline when idle



- Click on "Save".

Jenkins will now try to connect to the slave agent using the provided SSH details. If the connection is successful, the agent will become part of your Jenkins architecture and appear as online in your Jenkins dashboard.

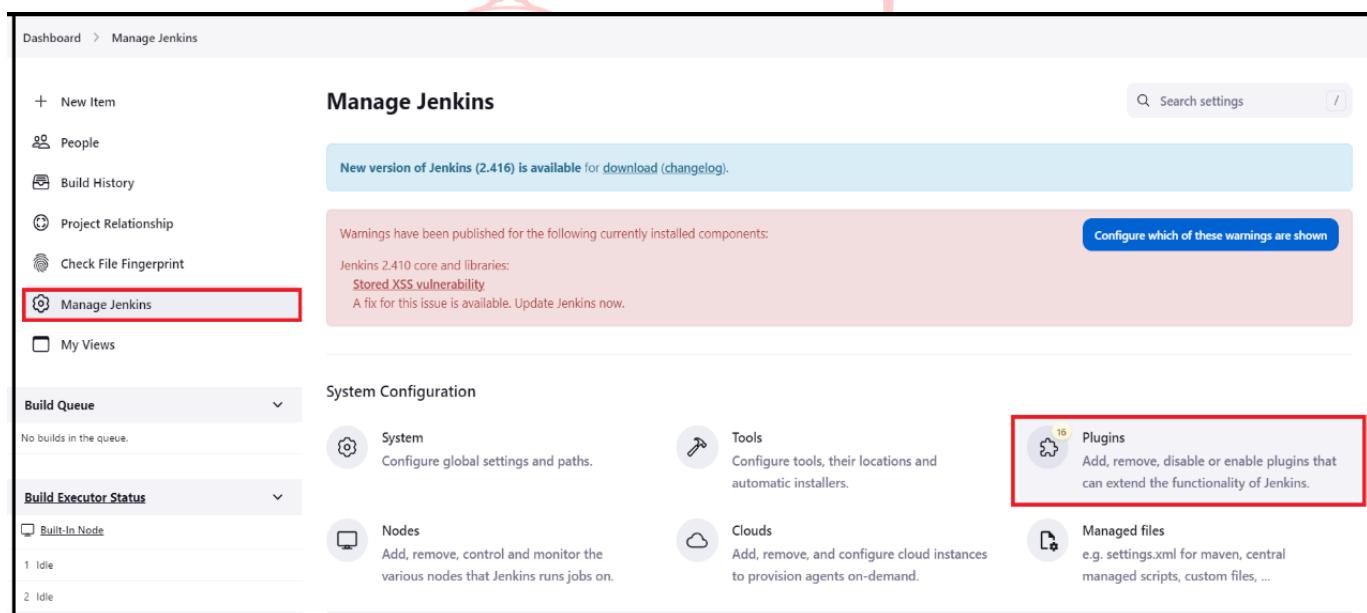
Remember to ensure that the Jenkins master can reach your agent node via SSH. You also need to have Java installed on the agent machine because Jenkins uses Java for the agent runtime.



The screenshot shows the Jenkins 'Nodes' page. On the left, there's a sidebar with links like 'Nodes', 'Clouds', 'Node Monitoring', 'Build Queue', and 'Build Executor Status'. The main area is titled 'Nodes' and contains a table with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. Five nodes are listed:

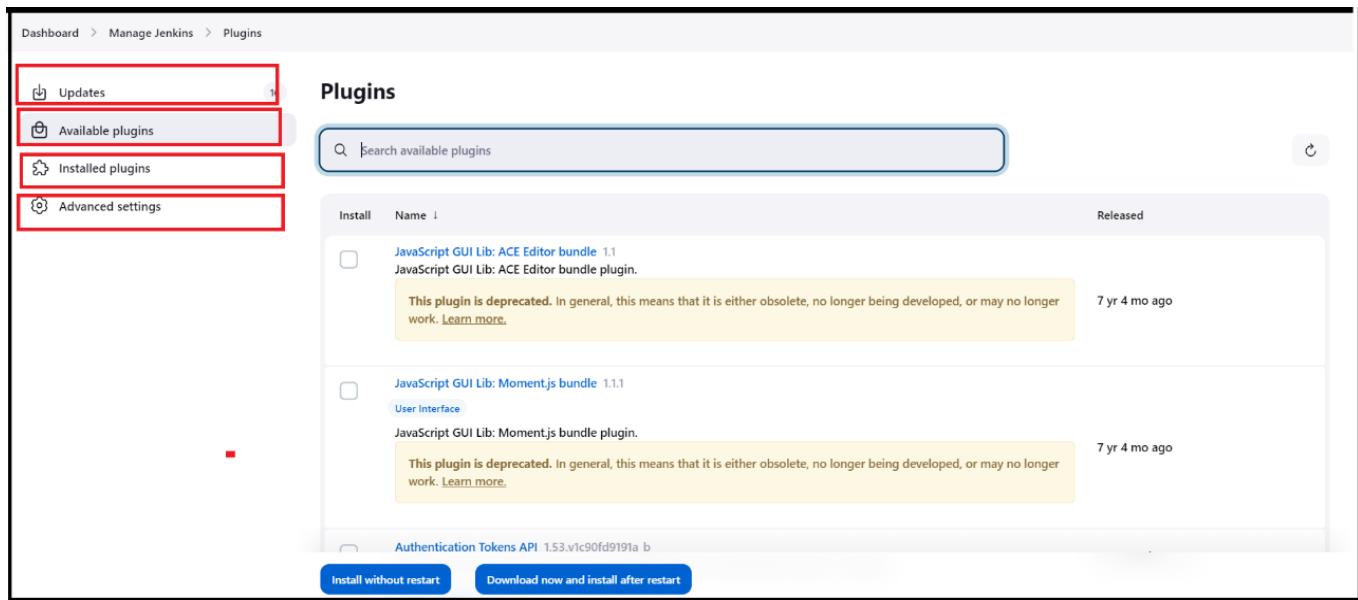
S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	8.21 GB	0 B	8.21 GB	0ms
	Linux_Slave	Linux (amd64)	In sync	13.05 GB	4.00 GB	13.05 GB	261ms
	Linux_temp	Linux (amd64)	In sync	63.42 GB	4.00 GB	63.42 GB	467ms
	Mac_Slave	Mac OS X (x86_64)	In sync	122.13 GB	1006.00 MB	122.13 GB	467ms
	Windows_Slave			N/A	N/A	N/A	N/A
	Data obtained		40 min	40 min	40 min	40 min	40 min

4.4 Plugins

The screenshot shows the Jenkins 'Manage Jenkins' page. On the left, there's a sidebar with links like '+ New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', and 'Manage Jenkins'. The 'Manage Jenkins' link is highlighted with a red border. The main area has a header 'Manage Jenkins' and a message 'New version of Jenkins (2.416) is available for download (changelog.)'. Below that, it says 'Warnings have been published for the following currently installed components: Jenkins 2.410 core and libraries: Stored XSS vulnerability. A fix for this issue is available. Update Jenkins now.' There's a button 'Configure which of these warnings are shown'. The page also features a 'System Configuration' section with icons for 'System', 'Tools', 'Nodes', 'Clouds', and 'Plugins'. The 'Plugins' section is highlighted with a red border and contains the text: 'Add, remove, disable or enable plugins that can extend the functionality of Jenkins.' It also lists 'Managed files'.

- Navigate to 'Manage Jenkins'
- In the 'Manage Jenkins' screen, click on 'Manage Plugins'.



The screenshot shows the Jenkins 'Manage Jenkins' section with the 'Plugins' tab selected. On the left, there's a sidebar with tabs: 'Updates' (highlighted with a red box), 'Available plugins' (highlighted with a red box), 'Installed plugins', and 'Advanced settings'. The main area is titled 'Plugins' and contains a search bar. Below it, a table lists three plugins:

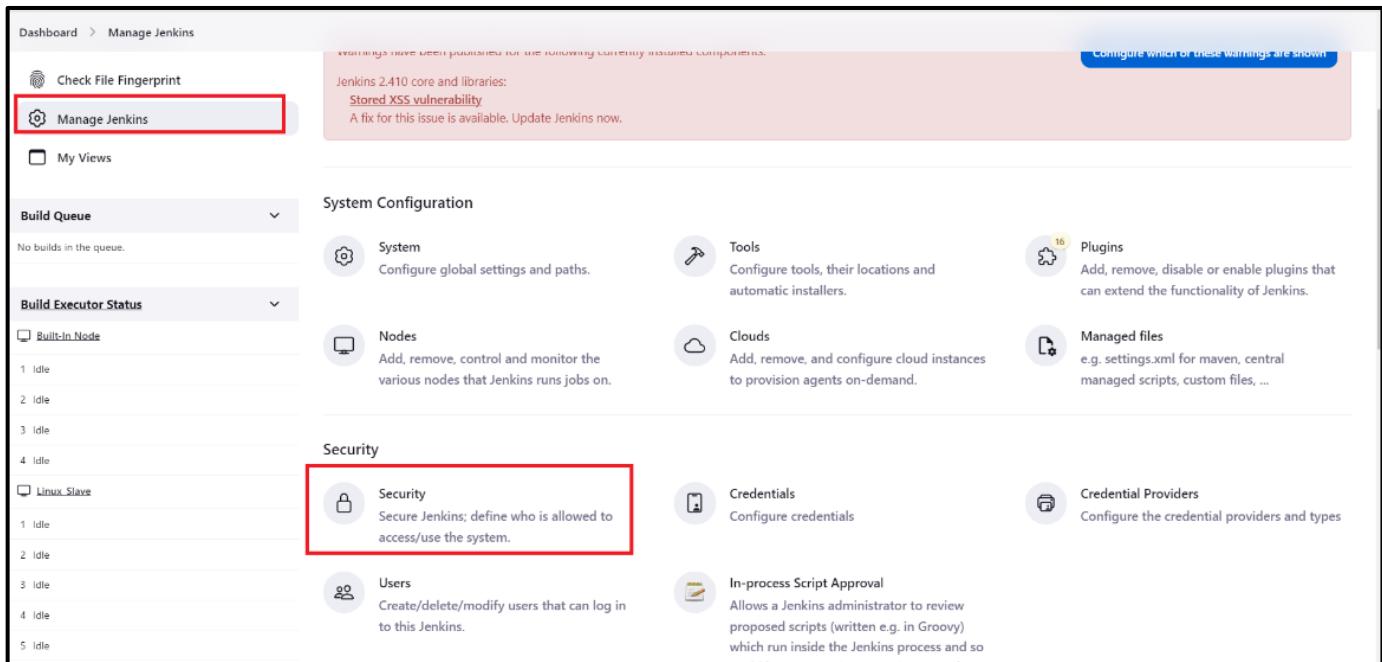
	Name	Released
<input type="checkbox"/>	JavaScript GUI Lib: ACE Editor bundle 1.1 JavaScript GUI Lib: ACE Editor bundle plugin.	7 yr 4 mo ago
<input type="checkbox"/>	JavaScript GUI Lib: Moment.js bundle 1.1.1 User Interface JavaScript GUI Lib: Moment.js bundle plugin.	7 yr 4 mo ago
<input type="checkbox"/>	Authentication Tokens API 1.53.v1c90fd9191a b	

At the bottom of the table are two buttons: 'Install without restart' and 'Download now and install after restart'.

- Go to the 'Available' tab and find the plugin you want to install. You can use the search bar to find it faster.
- Select the checkbox next to the plugin you would like to install.
- Click on 'Install without restart' or 'Download now and install after restart' button. The first option will install the plugin and makes it available immediately, but the second one will schedule the installation after the next Jenkins' restart.
- Restart Jenkins so that the changes can take effect. If you selected the "Install without restart" option, this may not be necessary. Still, in some cases, a restart is required for all features of the plugin to be correctly loaded.
- Only users **with admin permissions** can add or remove Jenkins plugins. In case the desired plugin is not found under 'Available' tab, it might be already installed, check the 'Installed' tab.

4.5 Security

- Adding security to your Jenkins setup is important to ensure that only authorized users can access it.



The screenshot shows the Jenkins Manage Jenkins dashboard. At the top left, there are links for 'Check File Fingerprint' and 'Manage Jenkins'. A red box highlights the 'Manage Jenkins' link. Below it, there's a warning about a stored XSS vulnerability in Jenkins 2.410 core and libraries, with a button to 'Update Jenkins now'. On the left, there are sections for 'Build Queue' (no builds in the queue) and 'Build Executor Status' (two nodes listed: 'Built-In Node' with 1 idle, 2 idle, 3 idle, 4 idle; and 'Linux_Slave' with 1 idle, 2 idle, 3 idle, 4 idle, 5 idle). The main area is titled 'System Configuration' and contains several sections: 'System' (Configure global settings and paths), 'Tools' (Configure tools, their locations and automatic installers), 'Nodes' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), 'Clouds' (Add, remove, and configure cloud instances to provision agents on-demand), 'Plugins' (16 available, Add, remove, disable or enable plugins that can extend the functionality of Jenkins), 'Managed files' (e.g. settings.xml for maven, central managed scripts, custom files, ...), 'Security' (Secure Jenkins: define who is allowed to access/use the system), 'Credentials' (Configure credentials), 'In-process Script Approval' (Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so), and 'Credential Providers' (Configure the credential providers and types). A red box highlights the 'Security' section.

- From your Jenkins Dashboard, click on 'Manage Jenkins'.
- On the 'Manage Jenkins' page, click on 'Configure Security'.
- Check the 'Enable security' checkbox.
- Under the 'Security Realm' section choose the desired method for user authentication.
 - 'Jenkins' own user database' - Jenkins will manage its own internal database of users.
 - 'LDAP' - For integration with LDAP as an authentication source.
- Under the 'Authorization' section, pick how will users get their permissions.
 - 'Logged-in users can do anything' - Any logged in users have full control but anonymous users cannot do anything.
 - 'Matrix-based security'** - You can provide specific permissions to each user or group.
 - 'Project-based Matrix Authorization Strategy' - You can provide specific permissions to each user or group in specific Jobs.
 - 'Role-Based Strategy' - You can provide specific permissions to a role and assign the role to a user or group. (You would need the 'Role-based Authorization Strategy' plugin for this.)
- After setting up the options as needed, click 'Apply' and 'Save' at the bottom of the page.

Now, your Jenkins instance should be secured. Depending on the options you chose, users may need to start logging in to have access. Be sure to create accounts and set up appropriate permissions as needed, and remember, having Jenkins security set up correctly is vital for securing sensitive data and operations in Jenkins.

Security

Authentication

Disable remember me

Security Realm

Jenkins' own user database

Allow users to sign up

Authorization

Matrix-based security

User/group	Overall	Credentials	Agent	Job	Run	View	SCM
Anonymous	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Authenticated Users	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Arvind Jain	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Ankit Bakarwala	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Aparav Bhavaras	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Ashu Munde	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Harish Patel	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Janini Maheshwari	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Kalpesh Chavhan	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Administrator	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Nayan Gehl	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Pankit Mistri	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Parth Patel	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Priya Apotikar	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Rakesh Solanki	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Ritesh	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Sanket Sharma	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Usama Mashayek	Allow	Allow	Allow	Allow	Allow	Allow	Allow

[Add user...](#) [Add group...](#)

Markup Formatter

Agents

TCP port for inbound agents: Fixed Random Disable

CSRF Protection

Copy Artifact

Git plugin notifyCommit access tokens

Current access tokens: There are no access tokens yet.

[Add new access token](#)

Git Hooks

Allow on Controller Allow git hooks to run on the Jenkins Controller
 Allow on Agents Allow git hooks to run on Jenkins Agents

Hidden security warnings

[Security warnings](#)

API Token

Generate a legacy API token for each newly created user (Not recommended)
 Allow users to manually create a legacy API token (Not recommended)
 Enable API Token usage statistics

SSH Server

SSHD Port: Fixed Random Disable

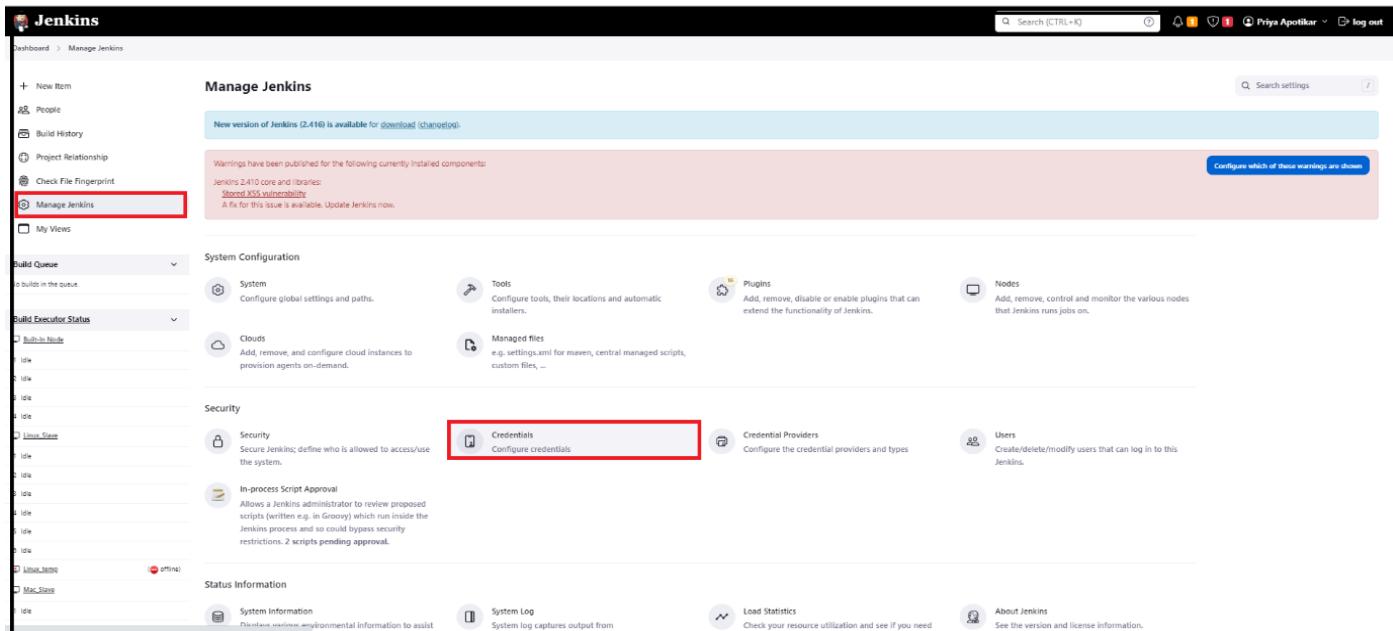
Git Host Key Verification Configuration

Host Key Verification Strategy: Known hosts file

[Save](#) [Apply](#)

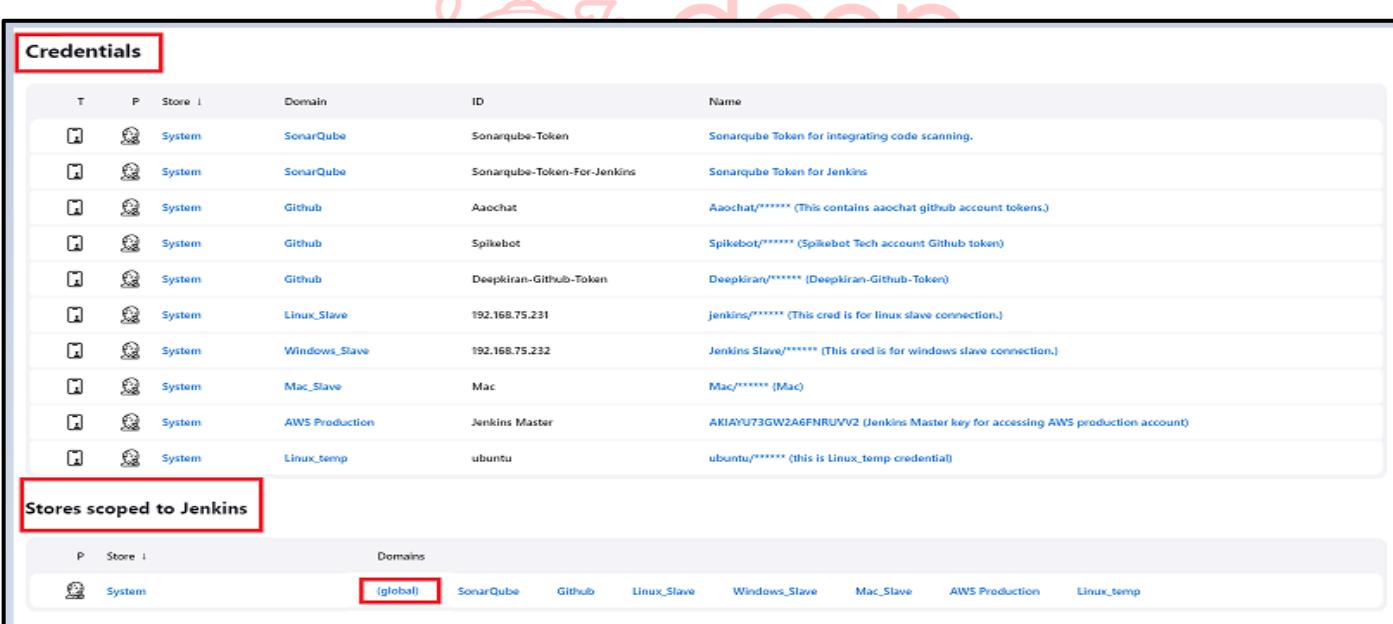
4.6 Credentials

- On your Jenkins dashboard, click on 'Credentials' in the left-hand menu.

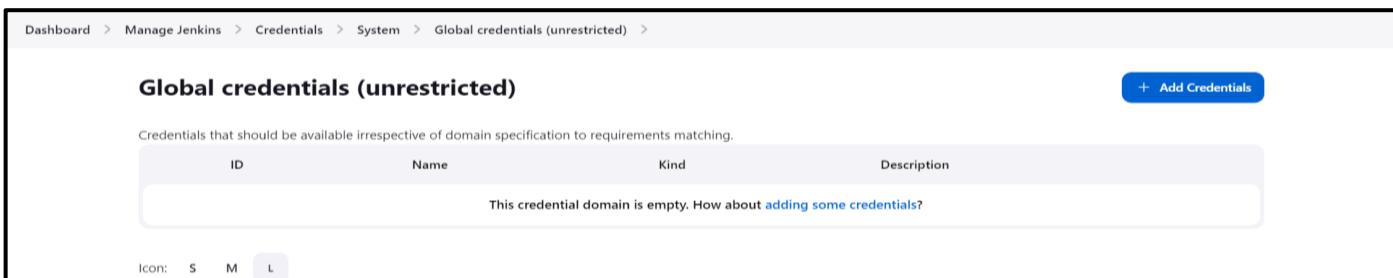


The screenshot shows the Jenkins Manage Jenkins page. On the left sidebar, under 'Manage Jenkins', the 'Manage Jenkins' item is highlighted with a red box. The main content area displays various system configuration sections: System Configuration (System, Tools, Plugins, Nodes), Security (Security, In-process Script Approval), and Status Information (System Information, System Log, Load Statistics, About Jenkins). A red box highlights the 'Credentials' section under Security.

- On the 'Credentials' page, you'll see a list of 'Stores scoped to Jenkins'. Click on '(global)' in the 'System' row.

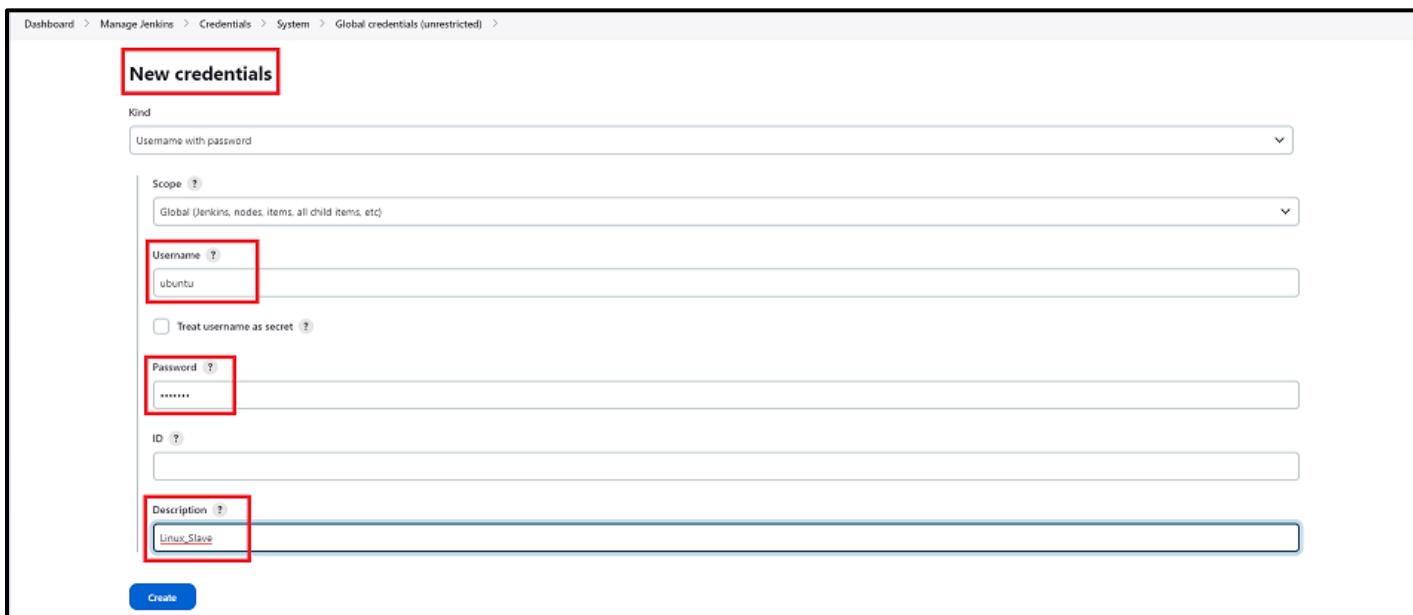


The screenshot shows the Jenkins Credentials page. At the top, the 'Credentials' menu item is highlighted with a red box. Below it, a section titled 'Stores scoped to Jenkins' contains a table of credentials. One row for 'System' is highlighted with a red box. The table columns are T, P, Store, I, Domain, ID, and Name. The 'System' row details are: ID: Sonarqube-Token, Name: Sonarqube Token for integrating code scanning. The table also lists other entries like Sonarqube-TOKEN-For-Jenkins, Aaochat, Spikebot, Deepkiran-Github-Token, jenkins, Jenkins Slave, Mac, Jenkins Master, and ubuntu.



The screenshot shows the Jenkins Global credentials (unrestricted) page. The URL in the address bar is: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted). The 'System' store is selected in the navigation bar, with '(global)' highlighted with a red box. The main content area shows a table with columns ID, Name, Kind, and Description. A message at the bottom states: 'This credential domain is empty. How about adding some credentials?' Below the table are size selection buttons: Icon: S M L.

- Choose the 'Kind' of credentials from the dropdown. For example, if you want to save a username and password, the kind will be 'Username with password'.
- Fill in the 'Username' and 'Password' details.
- In the 'ID' field, enter a unique ID that will be used to refer to these credentials.



New credentials

Kind: Username with password

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

Username: ubuntu

Treat username as secret:

Password: *****

ID:

Description: Linux_Slave

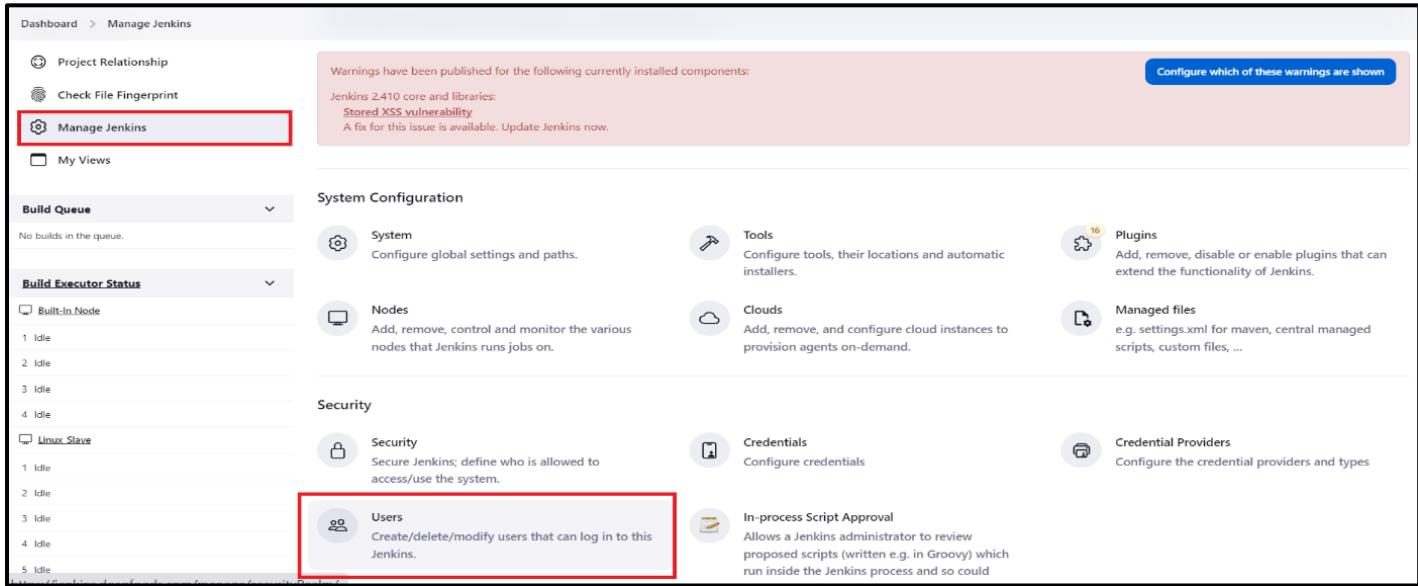
Create

- Now your credentials are stored and can be used in your Jenkins jobs. You can refer to these credentials by the unique ID that you provided when setting up the credentials.

4.7 Users

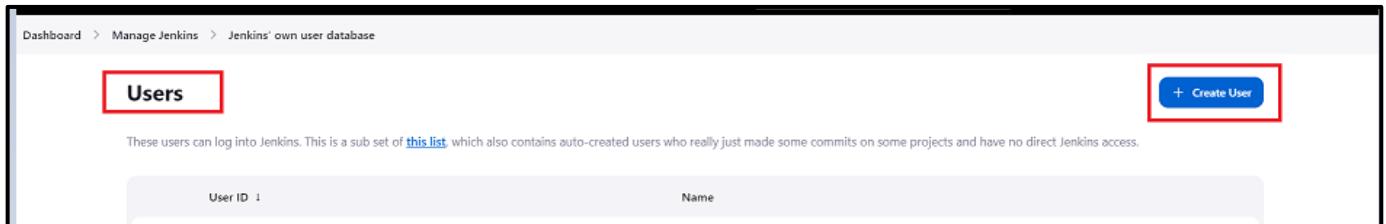
Here is a way to add users in Jenkins:

- Go to the Dashboard:
- Navigate to 'Manage Jenkins': Click on the 'Manage Jenkins' link available on the dashboard.



The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with links like 'Project Relationship', 'Check File Fingerprint', and 'Manage Jenkins'. The 'Manage Jenkins' link is highlighted with a red box. The main content area has a header 'System Configuration' with several sections: 'System' (Configure global settings and paths), 'Nodes' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), 'Tools' (Configure tools, their locations and automatic installers), 'Clouds' (Add, remove, and configure cloud instances to provision agents on-demand), 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), 'Managed files' (e.g. settings.xml for maven, central managed scripts, custom files, ...), 'Credential Providers' (Configure the credential providers and types), 'Security' (Secure Jenkins; define who is allowed to access/use the system), and 'In-process Script Approval' (Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could). A red box highlights the 'Users' section under the Security heading.

- In the 'Manage Jenkins' section, scroll down and click on 'Manage Users'.
- **Note:** If you don't see 'Manage Users', you need to ensure the 'Enable Security' is checked in 'Configure Global Security' and 'Jenkins' own user database' is selected as the 'Security Realm'.



The screenshot shows the 'Manage Jenkins > Jenkins' own user database' page. At the top, there's a 'Users' tab and a blue 'Create User' button, both of which are highlighted with red boxes. Below the tabs, there's a note: 'These users can log into Jenkins. This is a sub set of [this list](#), which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.' There's a table with columns 'User ID' and 'Name'.

- Here, you will see a list of users. There is also an option to create a user, click on 'Create User'.
- Fill up the 'Create User' form with the user name, password, confirm password, full name, and email address.

Create User

Username

Password

Confirm password

Full name

E-mail address

Create User

- After you have filled up the user form, just click on the 'Create User' button to finish creating the user account.
- The user you just created can now log in to Jenkins using the credentials you've established. You can manage users' permissions to control access from Configure Global Security, using the appropriate Authorization setting.