Jenkins is an automation tool written in Java with built-in plugins for continuous integration tasks. It is used to continuously build and test projects making it easier to integrate the changing codes to it.

Jenkins Originally Developed by Sun Micro Systems in 2004 under name **Hudson.**

The Project was later named Jenkins when oracle bought Microsystems

Jenkins Installation:

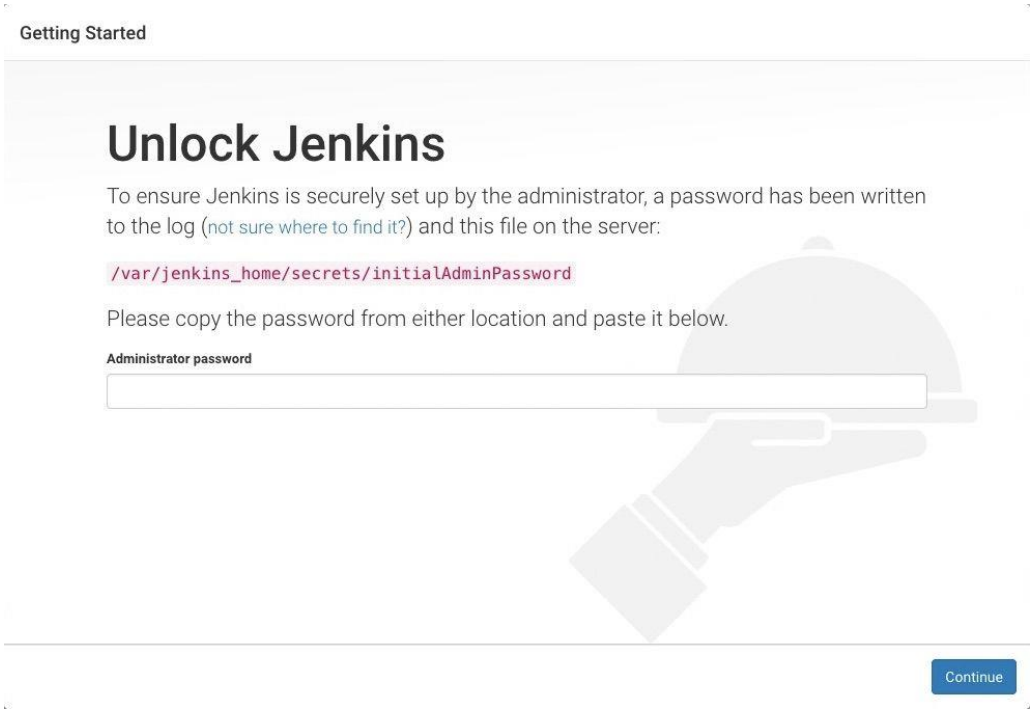
- Install Jdk 17
- sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
- sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
- yum install Jenkins

- You can enable the Jenkins service to start at boot with the command
- sudo systemctl enable Jenkins

- You can start the Jenkins service with the command:
- sudo systemctl start Jenkins

- You can check the status of the Jenkins service using the command:
- sudo systemctl status jenkins
-

Browse **to http://localhost:8080** (or whichever port you configured for Jenkins when installing it) and wait until the **Unlock Jenkins** page appears.

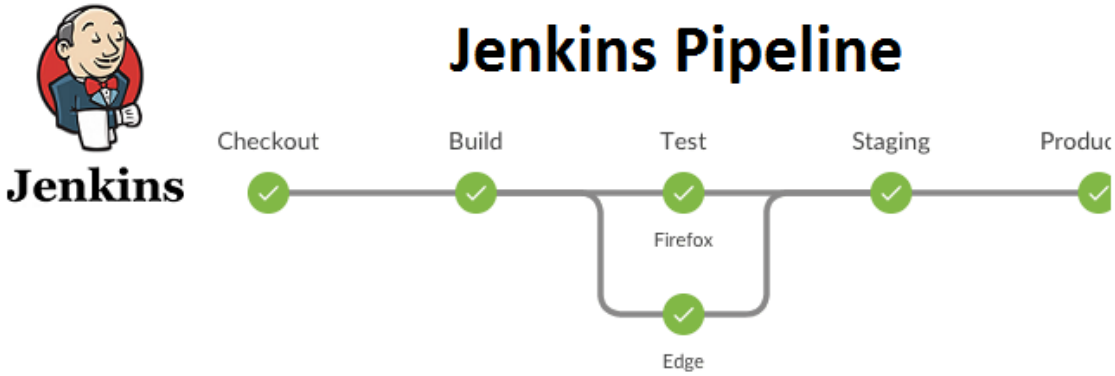


- The command**: sudo cat /var/lib/jenkins/secrets/initialAdminPassword** will print the password at console.

Other Platforms than Jenkins:

- GitLab CI/CD
- Travis CI
- CircleCI
- Bamboo

Jenkins WorkFlow:



Continuous Delivery

A software development discipline where software is built so that it CAN be released to production at any time.

Continuous Integration

A software development practice where contributors are integrating their work very frequently. This results in multiple daily integrations to a mainline. Automated testing (post-commit promotion) is commonly used.

PipeLine:

Jenkins Pipeline is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins. Pipeline provides an extensible set of tools for modeling simple-tocomplex delivery pipelines "as code"

CI/CD Pipeline

A typical CI/CD pipeline includes the following stages:

1. **Source Stage**: Code is committed to the version control system (e.g., Git).
2. **Build Stage**: Code is compiled or packaged into executable files.
3. **Test Stage**: Automated tests are run to verify the functionality and correctness of the code.
4. **Deploy Stage**: The code is deployed to a staging or production environment.
5. **Monitor Stage**: The deployed application is monitored for performance and errors.

Procedure for Adding a Jenkins Slave via Launch agent by connecting it to the controller

- Switch to the "root" user. sudo su

From the Jenkins Dashboard

- Click "Manage Jenkins" from the left panel.
- Click "Manage Nodes."
- Click "Add Node."
- Set a name for your node (e.g. "Slave 1").
- Select "Permanent Node."
- Set "Remote root directory" to /var/lib/jenkins.
- Set "Usage" to "Use this node as much as possible."
- Set "Launch Method" to "Launch agent by connecting it to the controller"
- Click "Save."

Run from agent command line: (Unix)

- **curl -sO http://3.111.29.145:8080/jnlpJars/agent.jar**
- **java -jar agent.jar -url http://3.111.29.145:8080/ -secret 64c54177a3efa0b4d285525e8ec783a0b0cae21ef776940e04b7aca9545a1042 -name jenkins -workDir "/var/lib/jenkins"**

- The agent should now be available for use.

Installing Plugins via the Jenkins Web Interface

1. **Access Jenkins Dashboard**
    - Log in to your Jenkins instance and navigate to the Jenkins dashboard.
2. **Go to Plugin Manager**
    - Click on "Manage Jenkins" from the left-hand menu.
    - Select "Manage Plugins."
3. **Choose a Plugin to Install**
    - In the "Plugin Manager" page, you can find plugins in the following tabs:
        - **Updates:** Lists available updates for installed plugins.
        - **Available:** Shows plugins that are available for installation but are not currently installed.
        - **Installed:** Displays plugins already installed on your Jenkins instance.
        - **Advanced:** Allows manual installation of plugins.
4. **Install a Plugin**
    - In the "Available" tab, you can search for a specific plugin using the search box.
    - Check the box next to the plugin(s) you want to install.
    - Click "Install without restart" to install the plugin while Jenkins is running. Alternatively, click "Download now and install after restart" to install the plugin after a Jenkins restart.

If Your Jenkins DashBoard Is Slow

- Navigate to Manage Jenkins.
- Go to your Jenkins dashboard and click on "Manage Jenkins" on the left-hand side. Access System Configuration.
- In the Manage Jenkins section, select "Configure System". Update Jenkins Location URL.
- Find the field labeled "Jenkins Location" or "Jenkins URL".

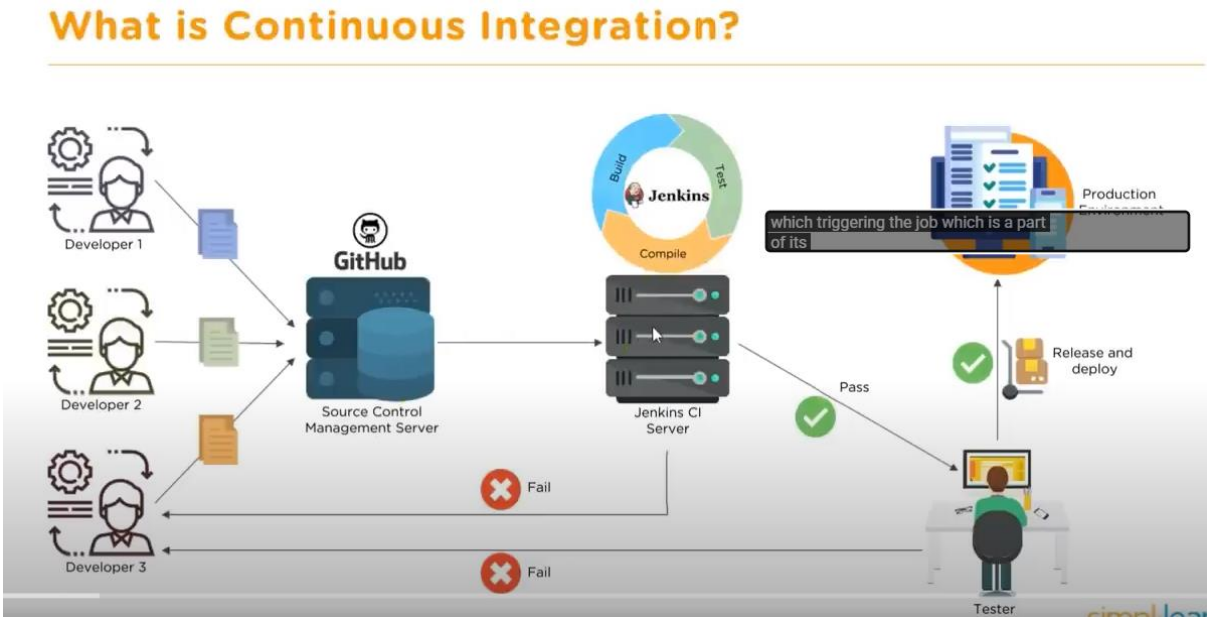
Update it to match the URL you use to access Jenkins, including the correct protocol (http:// or https://) and domain. Save Changes. Scroll down and click on "Save" to apply the changes.

OR

- cd /var/lib/Jenkins
- sudo vi jenkins.model.JenkinsLocationConfiguration.xml
- Check Public Ip
- Click save
- Restart jenkins

Create A Pipeline:

- Click Newitem
- Enter the item name
- Select Pipeline
- Click Ok
- Go to pipeline
- Select Pipeline script from scm
- Select Scm as git
- Paste Git Respository Their
- Select Branch to Build
- Select the script Path where file is
- Click on apply and save
- Click on Build Now



Credentials Setup

- Manage Jenkins
- Manage credentials
- System
- Global credentials

- Select "Add" under "Credentials."
- Set "Kind" to "SSH Username with private key."
- Set "Username" to "jenkins."
- Set "Private key" to "From the Jenkins Master."
- Click "Add."
- Choose the new credential from the "Credentials" dropdown.
- Click "Save."

MultiBranch Pipeline

The Multibranch Pipeline project type enables you to implement different Jenkinsfiles for different branches of the same project. In a Multibranch Pipeline project, Jenkins automatically discovers, manages and executes Pipelines for branches which contain a Jenkinsfile in source control

- Click "New Item" on the left navigation menu.
- Set the name to "My First Pipeline" under "Enter an Item Name"
- Then, select "Multibranch Pipeline" and click "OK".
- Under "Branch Sources" in the configuration menu, select "GitHub" or “Git” dropdown.
- In the build configuration place the path of jenkinsfile.
- Click "Save"
- If all went well, you'll see the job make an initial run and pass.

How to Add Webhooks to MultiBranch Pipeline

- Install Plugin Multibranch Scan Webhook Trigger
- Click scan by webhook
- Give me trigger token “what ever Name need to give”
- Click on save
- Move to github repo in the top of repository their will be setting click on settings left side their will be webhook
- Click webhook
- Click add webhook http://localhost:8080/jenkins/multibranch-webhook-trigger/invoke?token=”name of trigger”

First Free style Job

- Click on “New Item” at the top left-hand side of your dashboard.
- Enter Item details
- Enter Project details
- Enter repository URL
- Under build,
- Click on “Add build step”
- Click on “Execute Windows batch command”
- Click Apply Save the project.

Pipeline Script

```
pipeline {
    agent any 1
    stages {
        stage('Build') { 2
            steps {
                // 3
            }
        }
        stage('Test') { 4
            steps {
                // 5
            }
        }
        stage('Deploy') { 6
            steps {
                // 7
            }
        }
    }
}
```

Build Trigger

- Click Trigger Build Remotely
- Add Token “Give the name what ever you want”
- Click Apply save
- Go to Browser and paste the link

Building Ci/Cd pipeline real time Project

- Go to the job
- In the git check the docker-compose.yml file(see the version which is avaible in the your operating system)
- Manage Jenkins
- Go to tools check docker,nodejs,dependency,sonarcube installation

SonarQube

SonarQube *is an open-source tool to manage code quality and code analysis. It continuously inspects the code and performs automatic code reviews to detect bugs, code smells, and vulnerability issues in the code.*