Build Java Project using maven in Jenkins Pipeline

03 March 2023 01:26

Jenkins pipeline is a combination of plugins which automates number of tasks and makes the CI/CD pipeline efficient , high in quality and reliable.

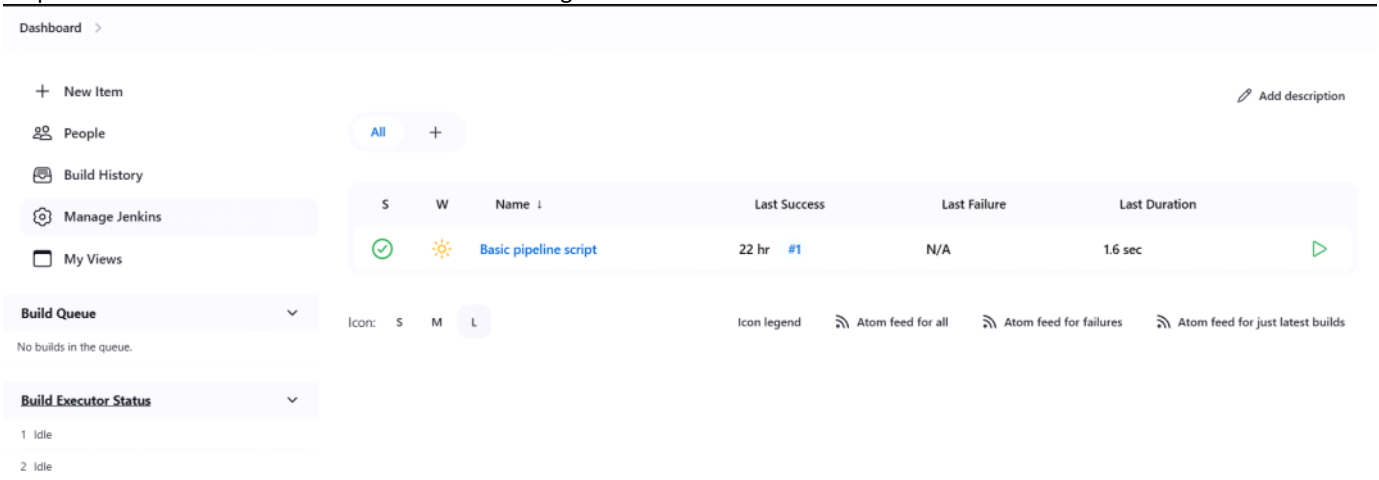
What is Jenkinsfile?

- Jenkinsfile is nothing but a simple text file which is used to write the Jenkins Pipeline and to automate the Continuous Integration process.
- Jenkinsfile works as a “Pipeline as a Code”.
- Jenkinsfile is written in couple of ways
    - Declarative pipeline syntax
    - Scripted pipeline syntax

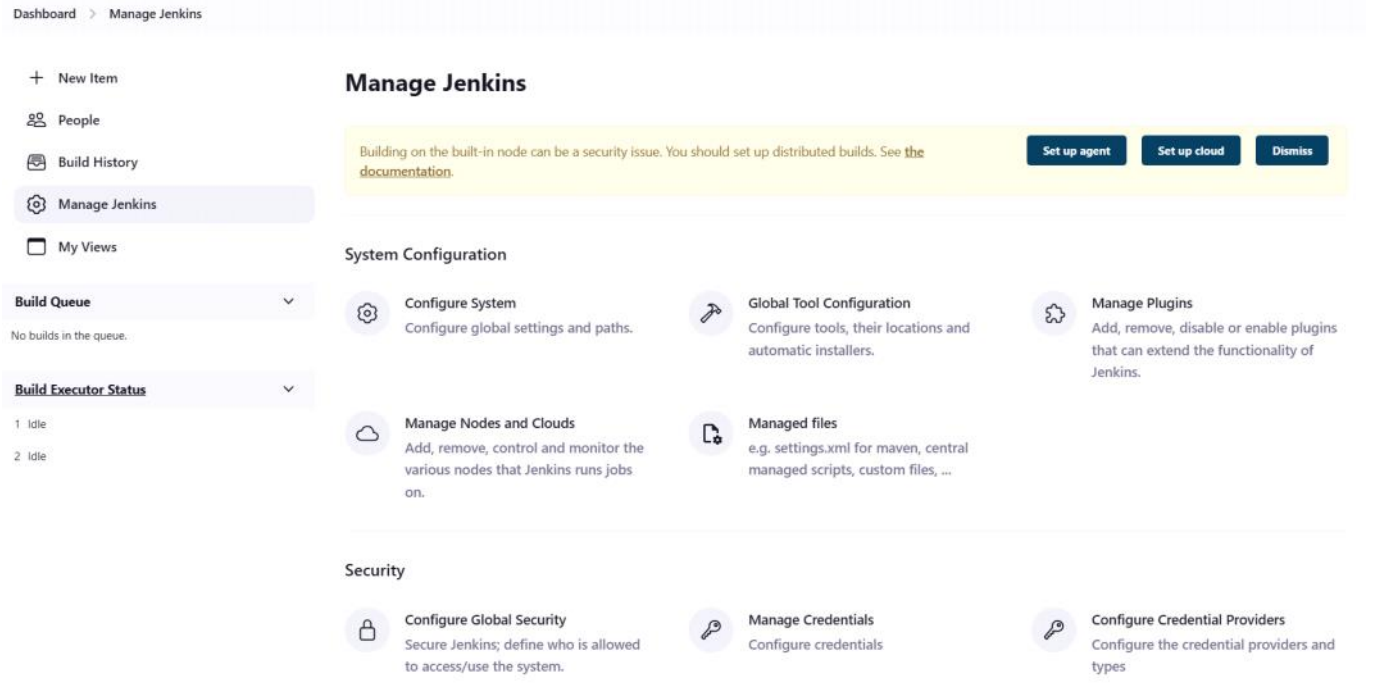
Install Plugins in Jenkins

1. Maven integration
2. Pipeline maven integration
3. Pipeline utility steps

Step 1: Go to Jenkins Dashboard and then click Manage Jenkins

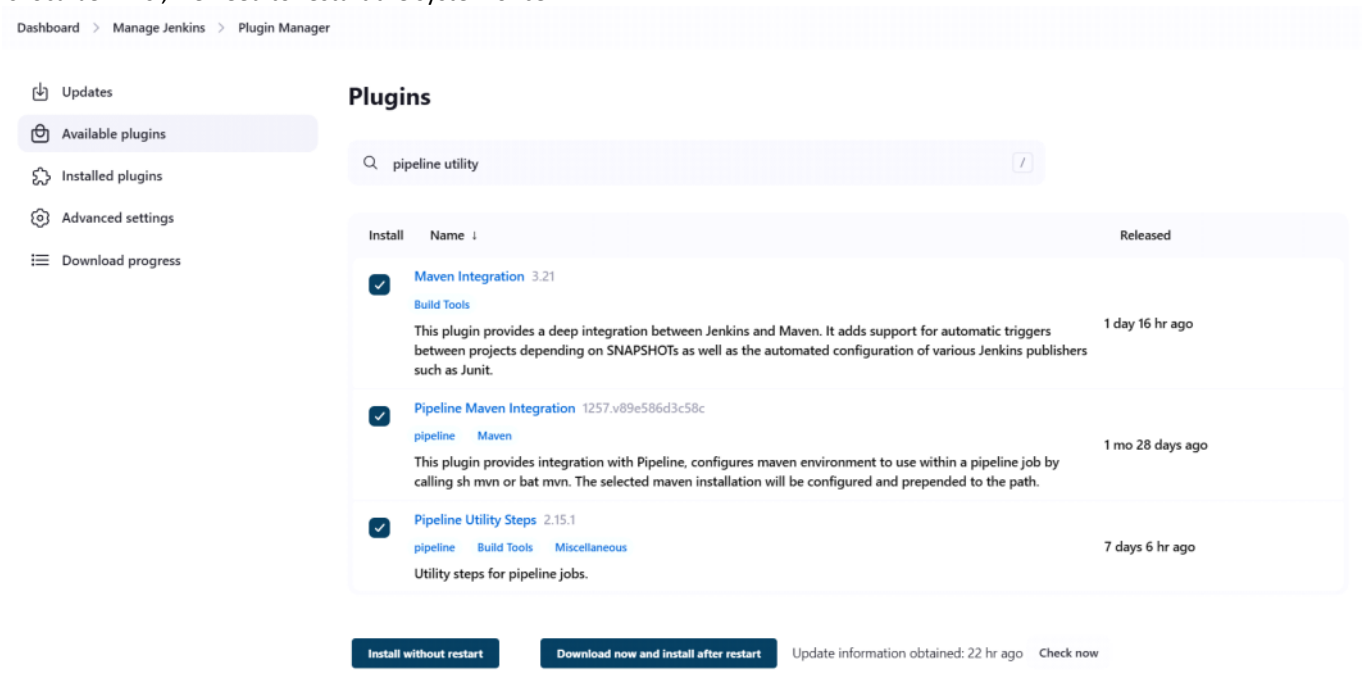


Step 2: And in here , you can see that 'Manage Plugins' under 'Manage Jenkins'



Step 3: And in here , we can select the plugins that we needed to run our project to download under the 'Available plugins' , we just need to check the plugins that are needed to be installed

and then click 'Download now and install after restart' , and then restart the Jenkins server , if it is local Jenkins , we need to restart the system once.

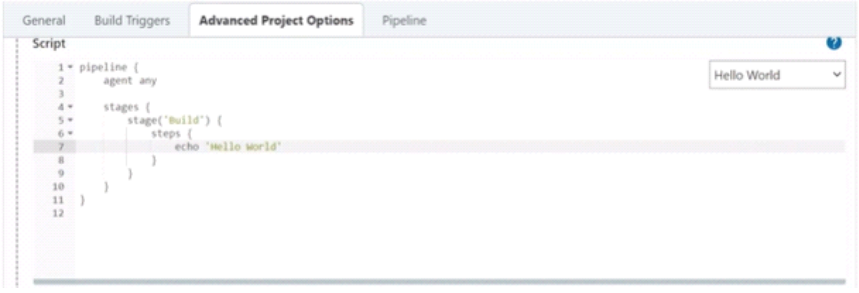


And later restart , we will be able to see the installed plugins under the 'Installed plugins'

In general , if we see any pom.xml , it consists of java project configuration and the dependencies of our java project.

This is the github repository that we are using (GitHub - devopshint/jenkins-pipeline-example) , it consists a small java project , where application print's the hello world and the test checks it prints helloworld or not.

The keyword agent in the pipeline defines that , where we need to run that pipeline. And later we give the series of steps under the 'stages' , under which we execute different steps under the individual 'stage'. Each stage have it's individual steps.



Jenkins Hands on

05 March 2023 01:09

Install ubuntu on virtual machine:

https://r2schools.com/how-to-install-and-configure-java-on-ubuntu-22-04-linux/

https://r2schools.com/how-to-install-jenkins-on-ubuntu-22-04/

https://docs.docker.com/engine/install/ubuntu/

There are different plugins of Jenkins that can help you run Docker in various ways. Here are some of them:

- **Docker plugin**[1][2]: This plugin allows you to dynamically provision containers as Jenkins nodes using Docker. You can configure a Docker cloud in Jenkins system configuration and specify the Docker image, label, credentials, etc. for each node template. Then you can use the label expression in your pipeline or freestyle job to run on a Docker node.
- **Docker Pipeline plugin**[3]: This plugin allows you to build, test, and use Docker images from Jenkins pipeline projects. You can use various steps and methods to interact with Docker in your pipeline script. For example, you can use docker.build to build an image from a Dockerfile, docker.image to access an existing image, docker.inside to run commands inside a container, etc.
- **Docker Build and Publish plugin**[4]: This plugin allows you to build an image from a Dockerfile in your workspace and push it to a registry. You can add this as a build step in your freestyle job and configure the image name, tag, registry credentials, etc.

Error:

permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/maven:3-alpine/json": dial unix /var/run/docker.sock: connect: permission denied

Solution:

This error means that the Jenkins user does not have enough permissions to access the Docker daemon socket at **/var/run/docker.sock**[1][2][3]. To fix this, you need to add Jenkins user to the Docker group on the host machine. You can do this by running this command:

`usermod -a -G docker jenkins`

Then you need to restart Jenkins and Docker services for the changes to take effect. You can do this by running these commands:

`service jenkins restart`

`service docker restart`

You may also need to change the permissions of **/var/run/docker.sock** file to make it readable

and writable by the Docker group. You can do this by running this command:

`chmod 664 /var/run/docker.sock`

This should solve your permission problem and allow you to build a Docker image in Jenkins.

Error:

```
[ERROR] Rule 0: org.apache.maven.plugins.enforcer.RequireMavenVersion failed with message:
Detected Maven Version: 3.6.3 is not in the allowed range [3.8.6,).
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
```

Solution:

There are two ways to install Maven 3.8.6 or above in Ubuntu: using apt command or direct download.

Using apt command:

- Update the system repository using: `sudo apt update`
- Install Maven from the official repository: `sudo apt install maven`
- Check the installed Maven version: `mvn -version`

Note: This method may not install the latest Maven version available on the official website[4].

Using direct download:

- Visit the Maven download page[3] and select the version of Maven you want to install. The latest version is listed in the Files section, and you can access earlier versions by using the archive link in the Previous Releases section.
- Download Apache Maven on your system using wget command. For example, to download Apache Maven 3.8.6, use: `wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.tar.gz`
- Extract the downloaded archive file to /usr/local directory using: `sudo tar xzf apache-maven-3.8.6-bin.tar.gz -C /usr/local`
- Create a symbolic link to make it easier to update Maven version in future: `sudo ln -s /usr/local/apache-maven-3.8.6 /usr/local/maven`
- Set up environment variables by editing /etc/profile.d/maven.sh file using: `sudo nano /etc/profile.d/maven.sh`
- Add the following lines to the file:

```
export M2_HOME=/usr/local/maven
export PATH=${M2_HOME}/bin:${PATH}
```

- Save and close the file.
- Make the script executable using: `sudo chmod +x /etc/profile.d/maven.sh`
- Load environment variables using: `source /etc/profile.d/maven.sh`

- Check the installed Maven version: `mvn -version`

New Concept

There are different ways to declare environmental variables in Jenkinsfile depending on your needs. One way is to use the environment directive of the declarative pipeline syntax[1][2]. For example, to set two variables named DISABLE_AUTH and DB_ENGINE, you can write:

```
pipeline {
 //Setting the environment variables DISABLE_AUTH and DB_ENGINE
 environment {
  DISABLE_AUTH = 'true'
  DB_ENGINE = 'mysql'
 }
}
```

This method sets the environment variables globally for the entire pipeline[2]. You can also set environment variables for specific stages or steps using the withEnv step[3]. For example, to set a variable named MESSAGE for a stage named Greeting, you can write:

```
pipeline {
 stages {
  stage('Greeting') {
   steps {
    withEnv(['MESSAGE=Hello World']) {
     echo "$MESSAGE"
    }
   }
  }
 }
}
```

This method sets the environment variables locally for a specific block of code[3]. You can also use pre-defined credentials as environment variables using credentials() function[4]. For example, to use a credential named MY_SECRET_KEY as an environment variable named SECRET_KEY, you can write:

```
pipeline {
 //Using credential MY_SECRET_KEY as environment variable SECRET_KEY
 environment {
  SECRET_KEY = credentials('MY_SECRET_KEY')
 }
}
```

This method allows you to access credentials securely without exposing their values[4].

Docker installation on centos

10 March 2023 02:16

Docker: Install Docker Engine on CentOS
Docker-compose: Install the Compose standalone (docker.com)

1. ***sudo curl -SL https://github.com/docker/compose/releases/download/v2.16.0/docker-compose-linux-x86_64 -o /usr/local/bin/docker-compose***
2. ***sudo chmod +x /usr/local/bin/docker-compose***
3. ***docker-compose*** (Use this command to test docker-compose is installed or not)

sudo du -sh /var/lib/docker => Tells the size of the docker occupying in our system or virtual machine or environment.
If we want to change permissions for a folder , we need to do '***sudo chown uid:gid <directory-name> -R***'
'***docker-compose up -d***' => Triggers the docker-compose file and run up the containers and perform what is in the docker-compose file.
'***docker logs -f <container-name>***' => This is used to check the logs of running container.

To start a docker service:
"sudo systemctl start docker"
To check the status of docker:
"sudo systemctl status docker"
To enable a service to start in the startup
"sudo systemctl enable docker"
If we are getting the /var/run/docker.sock error , we must add our current working user to docker group

If we want to create any local DNS for our IP address , either it can be of any service , we have to edit the hosts file , which is under the 'C:\Windows\System32\drivers\etc' , there may be accessibility permissions for that for the users , so edit them in the properties section. And add one of the line in the end as
192.168.0.132 jenkins.local
It means ,
<ip-address-that-we-have-to-map> <DNS-in-local-that-we want-to-use>
After we saving it here, we can use the same DNS to login through putty , instead of ip address. As they are mapped on local hosts file.

If we want to connect to this jenkins ip address through any other terminal on the system or out of the system if it is hosted on the cloud , we can use the ssh , for example:

"ssh username@ip-address" or "ssh username@domain-name" => In here , we use "ssh jeevan@jenkins.local"

Important links:
How to Configure Docker in Jenkins {Step-by-Step Guide} (phoenixnap.com)
Jenkins in Docker: Running Docker in a Jenkins Container | Hackmamba

docker run -it -p 8080:8080 -p 50000:50000 -v /var/run/docker.sock:/var/run/docker.sock -v jenkins_home:/var/jenkins_home custom-jenkins-docker

Email Notification in Jenkins: How to send Email from Jenkins |Send Email from Jenkins Job| Configure Email Notification In Jenkins

When we don't know to write the pipeline script , we must go to the Job Configuration , and in there the place we write the pipeline script , we will be having an option "Pipeline Syntax". In there, we will be having sample templates to write the pipeline.

1) Select a Job if created else create one.
2) Go to Job configuration.
3) Under "pipeline syntax" , we can select sample templates of Jenkins pipeline.

Install tomcat server using docker-compose file
[Deploy Apache Tomcat with Docker Compose - Linux Windows and android Tutorials (osradar.com)]