**Jenkins Practical’s**

**To start Jenkins**

Open command prompt

1. java -jar jenkins.war : By default this will start Jenkins on localhost port number 8080
2. java -jar jenkins.war --httpPort=8383 : This will start Jenkins explicitly on localhost port number 8383

**Freestyle Job**

1. Login to Jenkins
2. Click on **New** **Item**
3. Add name to job
4. In **General**, 🡪 Give **description** you want
5. In **Build**, 🡪 Select **Execute Windows Batch Command** 🡪 Give any batch command for execution 🡪 echo “Hello World”
6. Click on **Apply** & **Save**
7. Click on **Build** **now**

**Trigger Job Remotely**

1. Click on Configure
2. In **Build Triggers**, 🡪 Select **Trigger builds remotely (e.g., from scripts)** 🡪 Give **Authentication Token** 🡪 **Copy** given **Jenkins URL** 🡪 Open **new** **browser** 🡪 **Paste** link there 🡪 Instead of Jenkins\_URL : **localhost:8080** and TOKEN\_NAME : Given **Authentication Token**

When you hit enter here 🡪 Automatically Trigger will Build and you can see that on Jenkins Projects Dashborad

**How to Chain Job Execution**

1. Create Three **Freestyle Jobs** 🡪 Test1, Test2, Test3
2. Go to **Test2** 🡪 Click on **Configure** 🡪 Click on **Build Triggers** 🡪 Select **Build after other projects are built** 🡪 Give **Projects to Watch** as Test1 🡪 Select any one of the following 🡪
   1. Trigger only if build is stable : This means that Test2 will be built only after Test1’s build is stable.
   2. Trigger even if the build is unstable : This means that Test2 will be built even after Test1’s build is unstable.
   3. Trigger even if the build fails : This means that Test2 will be built even if Test1’s build is failed.
3. In **Post-Build Actions** 🡪 Select **Build Other Projects** 🡪 Give **Projects to Build** as Test3 🡪 Select any one of the following 🡪
   1. Trigger only if build is stable : This means that Test3 will be built only after Test2’s build is stable.
   2. Trigger even if the build is unstable : This means that Test3 will be built even after Test2’s build is unstable.
   3. Trigger even if the build fails : This means that Test3 will be built even if Test2’s build is failed.
4. Now, **Build** Test1 🡪 Here when first job gets executed successfully then second will be executed and if second gets executed successfully then third will get executed

This is how we chain the project

For Test1 🡪 downstream 🡪 Test2

For Test2 🡪 upstream 🡪 Test1 and downstream 🡪 Test3

For Test3 🡪 upstream 🡪 Test2

**How to add Git Credentials OR any Credentials to Jenkins?**

1. Login to Jenkins
2. Click on **Manage Jenkins**
3. Click on **Manage Plugins**
4. Install **Credentials Plugins**
5. Click on **Credentials**
6. Click on **Add** **Credentials** 🡪 Add **Username** and **Password** 🡪 Click on **OK**

This is how we add credentials to Jenkins

**Jenkins Integration with GIT (SCM)**

1. Create a java project & run in through command line
2. Add this project to Github
3. Now, Create a **freestyle job**
4. In **Build**, 🡪 Select **Execute Windows Batch Command** 🡪 Give commands to run **Java** **Program** 🡪
   1. Go to that folders location where java program is located 🡪 ‘ cd c:\Users\Gayatri\Desktop\Java ‘
   2. To compile java program 🡪 javac Hello.java
   3. Then to run java program 🡪 java Hello
5. In **SCM**, 🡪 Select **Git** 🡪 Give **Repository URL** and **Credentials** 🡪 Check **Branches to build** and **Branch Specifier (blank for 'any')**
6. In **Build Triggers** 🡪 Select **Poll SCM** 🡪 Give the Pattern of stars you want to build the job at specific time as \* \* \* \* \*
7. Click **Build now**

This is how we perform Jenkins integration with GIT

**How to do Automated Deployment? (Install Deploy to Container Plugin)**

1. Add user and password to **tomcat-user.xml**

Go to C:\Users\Gayatri\Softwares\apache-tomcat-9.0.45\conf 🡪 Open **tomcat-user.xml** 🡪 Add this to this file 🡪 <user username="user1" password="user1" roles="manager-script"/> 🡪 Save this file

1. Now start Tomcat server

C:\Users\Gayatri\Softwares\apache-tomcat-9.0.45\bin 🡪 Here open **Command Prompt** 🡪 Type = **startup** 🡪 Open browser and check tomcats link as **localhost:9090** 🡪 Here tomcat is running

1. Login to Jenkins
2. Create freestyle project
3. In **Build**, 🡪 Select **Execute Windows Batch Command** 🡪 Give Commands to execute as echo “Hello”
4. In **Post-Build Action**, 🡪 Select **Deploy war/ear to a container** 🡪 Give **WAR/EAR files** as \*\*/\*.war 🡪 Give **Context Path** as war\_file\_name.war 🡪 Click on **Add Container** 🡪 Select **Tomcat 8.x Remote** 🡪 Give **Credentials** which are added into the tomcat-users.xml file 🡪 Give **Tomcat URL** as <http://localhost:9090/>
5. Click on Apply and Save
6. Click on **Build Now**
7. Now, open **localhost:9090/war\_file\_name/**

Here we will see war file is deployed to Tomcat

1. To stop tomcat 🡪 type command : **shutdown** on command prompt

**How to send Email from Jenkins**

1. Click on **Manage Jenkins**
2. Click on **Configure System**
3. Go to **Email Notifications**
4. In **SMTP** **server** type the **smtp.gmail.com**
5. Go to **Advanced** section
6. Select **SMTP Authentication** Checkbox
7. In username give your **mail id** and **password**
8. Select the **Use SSL** checkbox
9. Enter **SMTP port** as 465
10. Check the checkbox of **Test Configuration by sending test email**
11. **Test Email Recipient** 🡪 Give any Recipient email id here
12. Click on Test configuration button (Email will be sent now)
13. Now create any freestyle project
14. In Post-Build Action 🡪 Select Email notification 🡪 Add receipts mail id and password & check any option

**How to trigger job with Email (Install Poll mailbox trigger plugin)**

1. Create a freestyle job
2. In **Build Trigger**, 🡪 Select **Poll Mailbox Trigger** 🡪 Give host username and password 🡪 host as imap.gmail.com
3. Click on **Advanced** 🡪 **Advanced email properties** (allows you to configure the plugin, using standard key=value property notation)🡪 ? click build 🡪 Execute windows batch command 🡪 Apply and Save
4. Now send the mail id to mail id by subject id we have given 🡪 Go to Jenkins and check mail triggered job or not

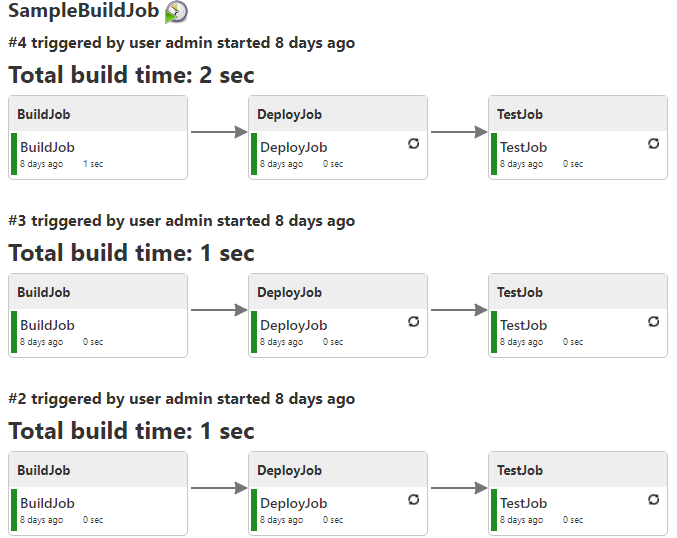
**Delivery Pipeline View**

* **For Downloading delivery pipeline view :**

Jenkins 🡪 Manage 🡪 Plugins 🡪 Available 🡪 Search **Delivery pipeline plugin**.

* **For Adding Delivery pipeline view in our Jenkins :**

Go to the Jenkins dashboard 🡪 Click on **+**  icon 🡪 select the delivery pipeline view 🡪 give any name to your delivery pipeline view Ex: **DeliveryPipelineView**  🡪 ok 🡪 go to down at pipeline section 🡪 components🡪 click initial job 🡪 select any your previous job whose pipeline view u want to see Ex: SampleBuildJob 🡪 Apply 🡪 Save 🡪It will show all past runs 🡪 click on full view 🡪 go to edit view 🡪 select no of pipelines instances per pipeline as u want here we have selected 3 instances.



**Build Monitor View**

* **For Downloading Build Monitor view :**

Jenkins 🡪 Manage 🡪 Plugins 🡪 Available 🡪 Search **Build Monitor plugin**.

* **For Adding Build Monitor view in our Jenkins :**

Jenkins dashboard 🡪 Click on **+** icon 🡪select build monitor view 🡪 Give any name to your view like Ex: BuildMonitorView 🡪Ok 🡪 Now select any no of jobs by checking the checkboxes of which you want to create view 🡪 apply 🡪 ok 🡪 we can see our build monitor view for our selected jobs.



**Build Pipeline View**

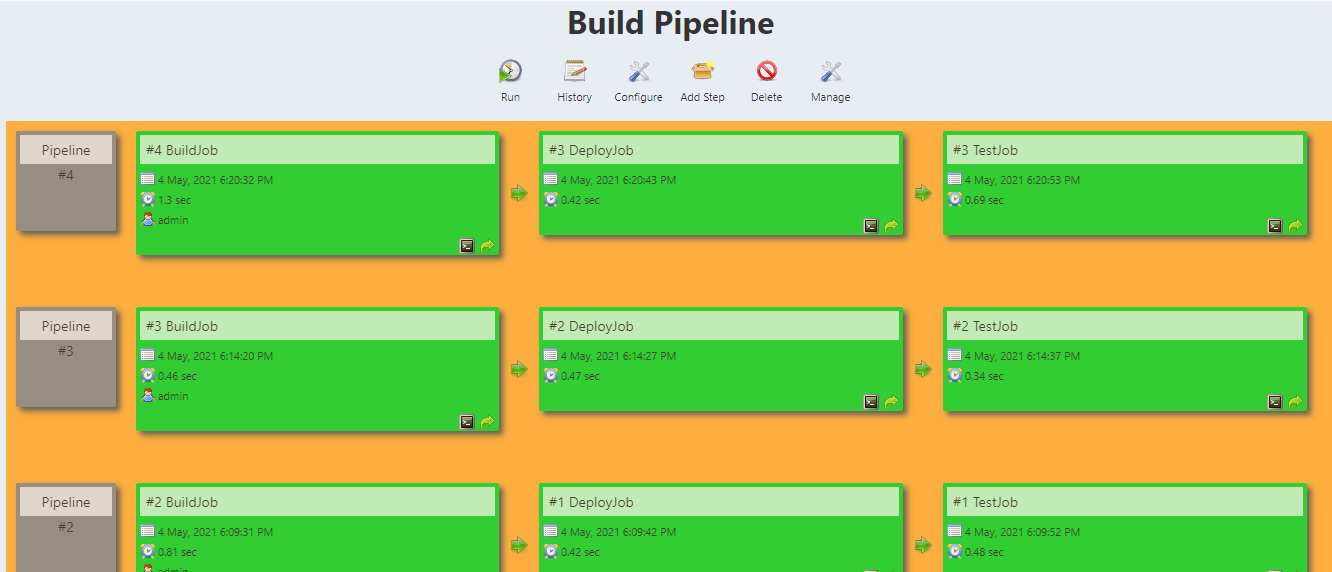
* **For Downloading Build Pipeline view :**

Jenkins 🡪 Manage 🡪 Plugins 🡪 Available 🡪 Search **Build Pipeline plugin**.

* **For Adding Build Pipeline line view in our Jenkins :**

Jenkins 🡪 click on **+** icon 🡪 Select build pipeline view 🡪 Give name to it 🡪 save 🡪 now we can see our job here like if we have chained job then it will display both jobs 🡪 Job is in green color means its execution is completed 🡪 blue means execution not yet started 🡪 yellow means currently executing task 🡪 red means failed.

To start the execution click on the run button execution will start.



**How to create Jenkins Pipeline**

1. Login Jenkins
2. Click on **New Item** 🡪 Give name to your Pipeline job (Pipeline1) 🡪 Select **Pipeline** 🡪 Click OK
3. In **Pipeline**, 🡪 In **Definition** 🡪 Select **Pipeline** **Script** 🡪 Select sample from **Try Sample Pipeline** 🡪 **Apply** and **Save**
4. Click on **Build** **Now**

**How to another Build job by building current job**

1. Create another **pipeline job** (Pipeline2)
2. Click on **Configure** (Pipeline1)
3. Select **Pipeline Syntax**
4. In **Sample step** 🡪 Select **build : build a job**
5. In **Project to Build** 🡪 Give job name you want to build (Pipeline2)
6. In **quiet** **period** 🡪 Give 0
7. Click on **Generate Pipeline Script**
8. **Copy** this script and **paste** in **Pipeline Script**
9. **Apply**, **Save** & **Build Now**

**How to get Jenkinsfile from Git SCM**

1. In **Git**, Make Repository 🡪 Place one file named **Jenkinsfile** 🡪 This should contain **pipeline** **script** to build
2. Create Pipeline Job
3. In **Pipeline**, 🡪 In **Definition**, Select **Pipeline Script from SCM** 🡪 Select **GIT** as **SCM** 🡪 Give **Repository** **URL** and **Credentials** 🡪 Check **Branch** of git 🡪 **Apply** and **Save**
4. Click on **Build now**

**How to clone a git repository using Jenkinsfile**

1. Create **Pipeline job**
2. Select **Hello world script**
3. Click on **Pipeline Syntax** 🡪 Select **git** : Git as **Sample step** 🡪 Give **Repository** and **Credentials** 🡪 Click on **Generate** **Script** 🡪 **Copy** this **script** and **Paste** it into **Pipeline** **Script** 🡪 **Apply** and **Save**
4. Click on **Build** **Now**

**How to Create Multibranch Pipeline**

1. Create a Job with **Multibranch Pipeline**
2. In **Branch Sources** 🡪 Git 🡪 Give Repository URL and Credentials
3. In **Scan Multibranch Pipeline Trigeers** 🡪 Select **Scan by webhook** 🡪 Give **Trigger Token** (Click on **?** of **Trigger Token** 🡪 **Copy** that **Link**) 🡪 Apply and Save
4. On Git 🡪 Make one repository 🡪 Add **Jenkinsfile** 🡪 Add some **script** 🡪 Click on **Settings** 🡪 Click on **Webhook** 🡪 Click on **Add** **Webhook** 🡪 **Payload** **URL** (Paste copied link here and give trigger token you given as Trigger Token) 🡪 Now for **Jenkins** **URL** 🡪 Open **ngrok.exe** 🡪 give command as **ngrok.exe http 8383** (Port where Jenkins is running) 🡪 Copy **IP address** and Paste that to **Jenkins URL** on GIT 🡪 Click on **Add Webhook**
5. Now add branch or some files and commit it 🡪 Then we will see automatically it will come to Jenkins

**Gradle Integration with Jenkins**

1. Click on **Manage Jenkins**
2. Click on **Global Tool Configuration**
3. Go to **Gradle** 🡪 Click on **Add Gradle**
4. Now, Give name to Gradle 🡪 Specify Gradle Home Directory (C:\Users\Gayatri\gradle-7.0) 🡪 Uncheck Install automatically 🡪 Apply and Save
5. Create gradle project and push it into Git and Copy git URL
6. Now Create Jenkins job as freestyle
7. In General, Select GitHub project 🡪 Paste copied link here and remove .git
8. In SCM, Select Git 🡪 Provide Repository URL and Credentials and check branch
9. In Build, Select Invoke Gradle Script 🡪 Select Gradle Version 🡪 Give tasks as build 🡪 Apply and Save
10. Click on Build now

**PetClinic Scripted pipeline**

Manage Jenkins🡪manage plugins🡪pipeline maven integration plugin 🡪Ok

Create new item 🡪 PetclinicScriptedpipeline🡪select pipeline🡪ok🡪go down select the pipeline option🡪 select pipeline script

Write this code into the block:

node

{

stage('checkout')

{

git 'https://github.com/RaikarPriyanka/maven-java-sample-app.git'

}

stage('build')

{

withMaven(maven:'M3')

{

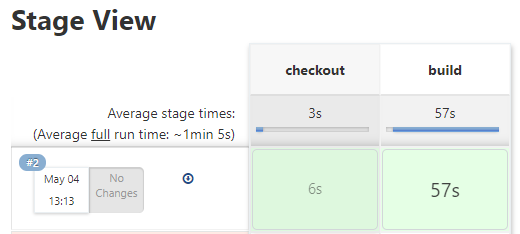
bat 'mvn compile'

}

}

}

Save this code 🡪 build now



**PetClinic Declarative pipeline**

Create new item 🡪give any name like: PetClinicDeclarativePipeline 🡪pipeline 🡪ok 🡪 select pipeline 🡪 write below script into it

pipeline {

agent any

stages {

stage('Checkout') {

steps {

git branch: 'master', url:'https://github.com/RaikarPriyanka/maven-java-sample-app.git'

}

}

stage('Build') {

steps {

bat 'mvn compile'

}

}

stage('Test') {

steps {

bat 'mvn test'

}

}

stage('Package') {

steps {

bat 'mvn package'

}

}

stage('Deploy') {

steps {

bat 'java -jar -Dserver.port=8001 target/spring-petclinic-2.3.1.BUILD-SNAPSHOT.jar'

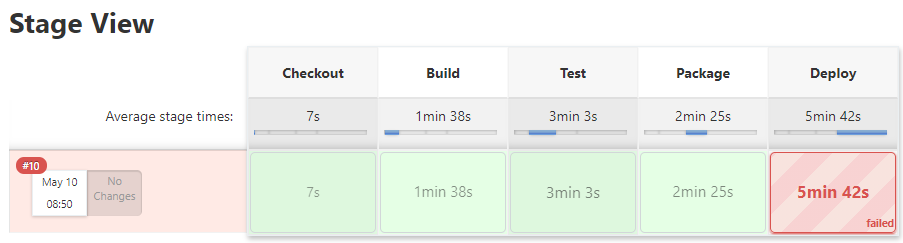
}

}

}

}

Save this 🡪build now



**How to use CATLIGHT**

Catlight will notify us when buils,bugs & tasks needs our attention.

Download catlight from internet 🡪 now once download extract the zip file 🡪 now click on it & start the catlight 🡪 open 🡪 select the Jenkins from list 🡪 Your Jenkins should be running at that time 🡪 Give our Jenkins url there like <http://localhost:8080> 🡪 select anonymus or add your name 🡪 save 🡪 It will display all the list of jobs present on your Jenkins dashboard till now --. Select all 🡪 save 🡪 it will monitor the status of all the projects & get notified on our windows desktop.

**What is BLUE OCEAN**

* A new user experience for Jenkins.
* Increases the clearity
* Provides the interactive view of the Jenkins pipeline.

Go to Jenkins dashboard 🡪 manage 🡪 plugins 🡪 install the BLUE OCEAN plugin 🡪 restart your Jenkins 🡪now you will see blue ocean icon on dashboard 🡪 click on any job u will get its all details there.