**JENKINS**

# What is Jenkins?

## Tool used to run our automation code on the build.

## Final build will be ready in the Jenkins itself.

## We must keep our code in common repo, and we must configure the path and other things in Jenkins .

## It is headless(no UI)- very fast.

## In real time DevOps team will provide the URL, username, and password we can access using that and add our project and configure.

# Run using Jenkins:

# Download Jenkins

# Configure Jenkins

## URL [**http://localhost:8080**](http://localhost:8080)( in project DevOps will share the URL)

## To unlock Jenkins in Jenkins secret folder u will be found the password.

## Install suggested plugins.

## Create admin users( in Jenkins wat are the project u r responsible for u can see only that DevOps team will give access only to that)

## Save and finish.

## We are using maven project in eclipse so install maven plugin in Jenkins.

## For that go to manage Jenkins(all plugins we can install from this and all configurations can be done) and manage plugins

## Select all maven related plugins.

## Click install without restart.

## Run automation thru Jenkins.

# Two ways to run our test in Jenkins.

# Freestyle Project

## We can run our test using run.bat file which contain our commands.

## Select new item -> give Project name->freestyle project->ok->come down to build section->dropdown select execute window batch command->give path of run.bat file along with cd->save.

## Open project and click build now to execute. It will automatically execute the commands and run the test cases.

# Maven Project

## Directly run our pom.xml( we must do some configuration in Jenkins)(

## Manage Jenkins -> Global tool configuration(Jenkins should know where our java and maven is located)

## Click JDK installation.

## Give any name and give the jdk path and give apply and save.

## Select new item -> give Project name->select Maven project-> ok->come down to build section-> give exact path of pom.xml in Root POM and clean install command in Goals and options->apply and save.

## Open project and click build now to execute. It will automatically execute the commands and run the test cases.

# Workflow

## As QA we must do and set this project in Jenkins and DevOps team will execute it all time.

## Actual Jenkins will be in DevOps environment but they will only share URL with u but actual application is in DevOps server but entire project is in local system but how will you configure your local path in remote Jenkins.

## So, for that we will keep all our code in remote repository(git and GitHub). This code can be accessed thru URL. Jenkins will access from GitHub. So instead of giving local path we must provide GitHub repository path where we maintained our project.

# Dev

## Implement the code by using certain programming language.

## All developers at the end of the day they will keep their code in common repository.

# DevOps

## Perform certain tasks from developer and perform certain task from QA.

## Basically, they will share the task between QA and Dev

## E.g. They will take all code from repository once developer checked in, they will create build(final integrated product(exe, msi etc)

## 9 to 5 – dev worked and pushed code in repo.

## 5 to till next day morning(9 am)- DevOps team run automation(integration of build happen and automation code execute parallelly)

## 9 – QA will test the complete build and test

## DevOps team will again take the automation code from the common repo execute our automation code against this build.

## Due to this process is faster.

## Once the build is successful everybody in the team will get automatic email that build is pass then QA team will download this build in their environment and install and continue the rest of the functionalities and process continues

# QA:

## Implement the test scripts by using certain programming language.

## All testers at the end of the day they will keep their automation code in common repository.

# Jobs:

## Each phase contains different jobs

## Using pipeline, we will execute each and every job in sequential order.

# Jenkins Pipeline

# Two ways:

## Using Plugin

## Using groovy script

# Deploy:

## Process of creating build based on code and testing.

# Continuous Integration:

## Developing the code, building the code (based on code build the software), deploying the code and testing the code(test the software.

# Continuous Delivery:

## Once above all is done, we will release (release the software)the particular build to the customer.

## Every activity is dependent on some other activity in Jenkins.

## Get the code from developer(jobs)-> then build it(jobs)-> after test it(jobs) -> after deploy it(jobs)-> after release(jobs)

## This process can be achieved by using pipeline.

# Pipeline:

## Group of jobs or events which are interlinked with one another in sequence. 4 phases

# BUILD

## Run build jobs(Deploy is downstream job for this)

# DEPLOY

## Run different deploy jobs(Build is upstream job for this)

# TEST

## Rest software which is already build and deployed.

# RELEASE

## Release software and run certain release jobs.

# Set up Pipeline:

## Create multiple jobs(create chain) which are related to multiple phases.

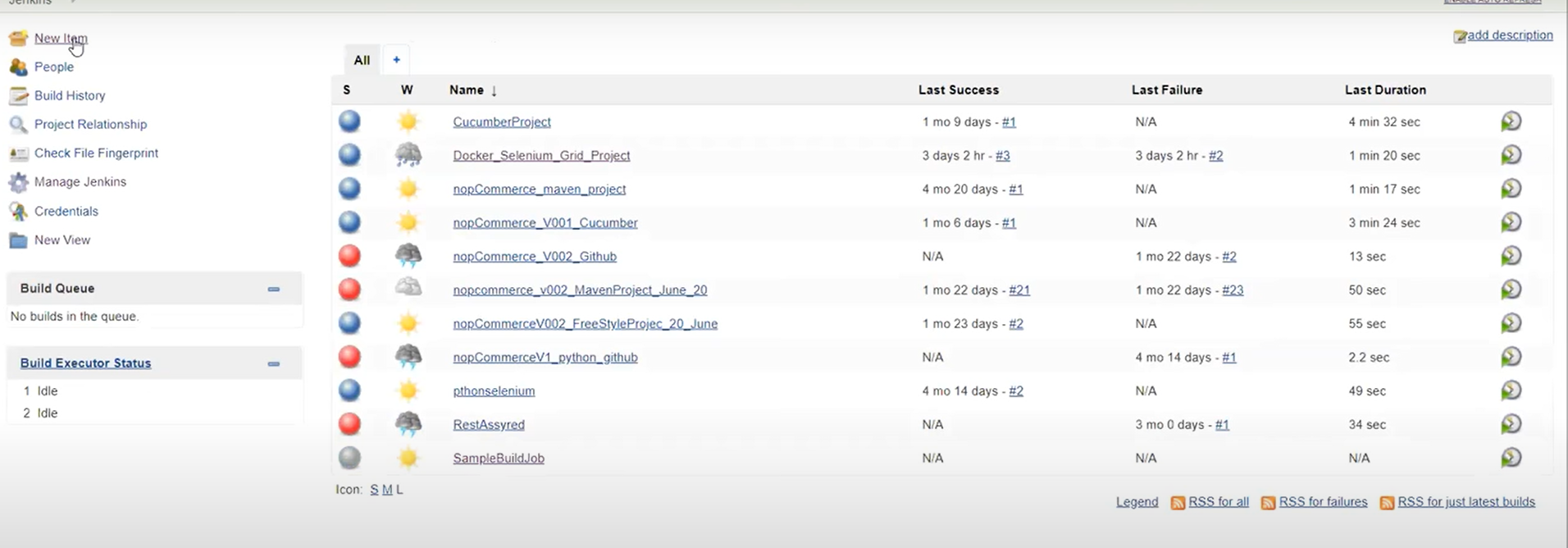
## Every job is created by using different types of commands.

## Deploy job we must configure it because it has to execute after build job .

## So, in deploy job project name dropdown select configure -> In build trigger section select “build after other projects are build”-> in projects to watch “ give Build job project name” since it must be executed before deploy job-> apply -> save.

## If we run build job after this all-other jobs will automatically execute in sequence since it is chained and interlinked

## use pipeline plugin to link (as linked above)and execute the jobs sequentially.





A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

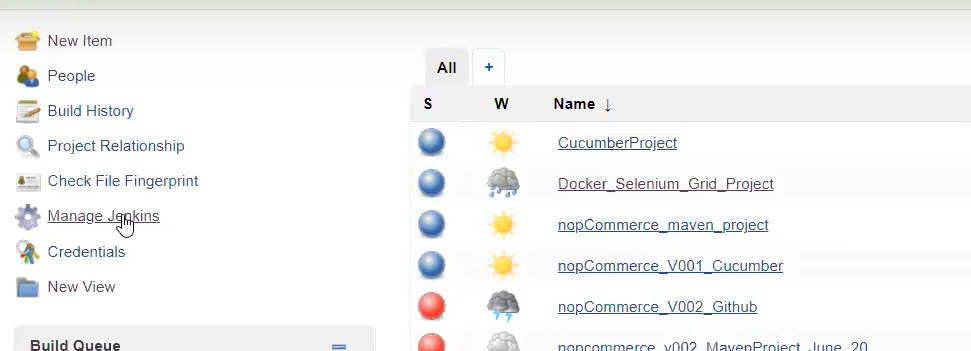
Description automatically generated

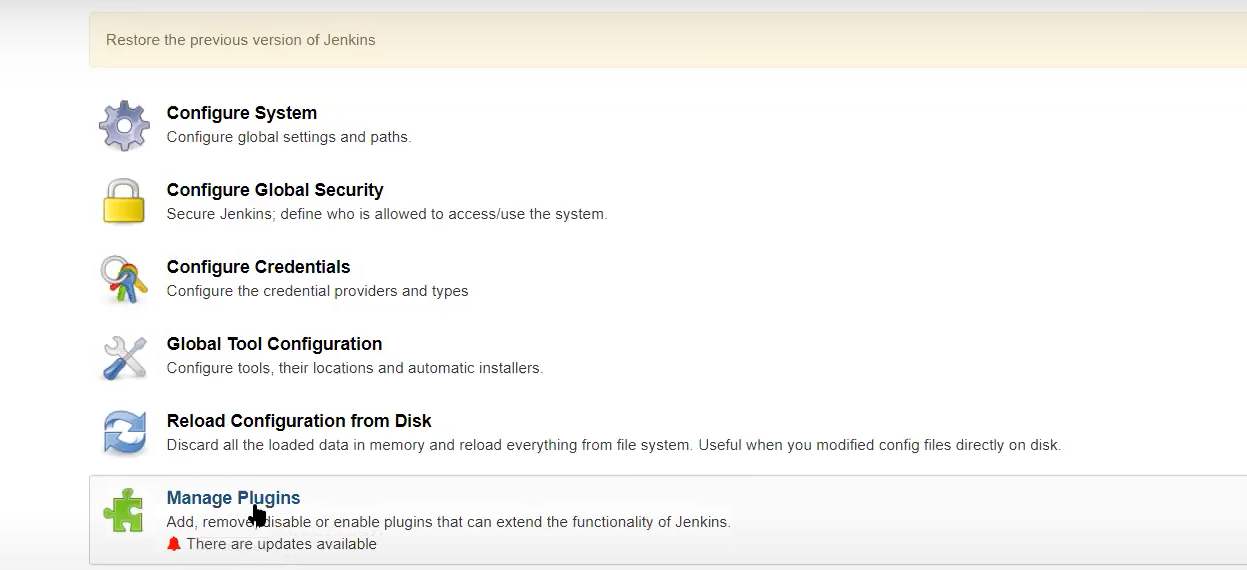
# Step 1: install pipeline plugin

A screenshot of a computer

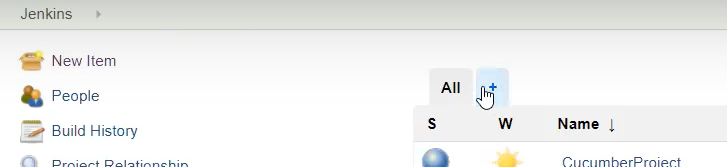
Description automatically generated

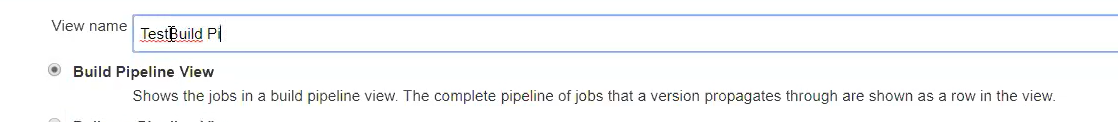
## Manage Jenkins-> mange plugins->Under Available “Build pipeline plugin”-> click install without restart.





## In dashboard click plus button and select build pipeline view-> give view name-> ok



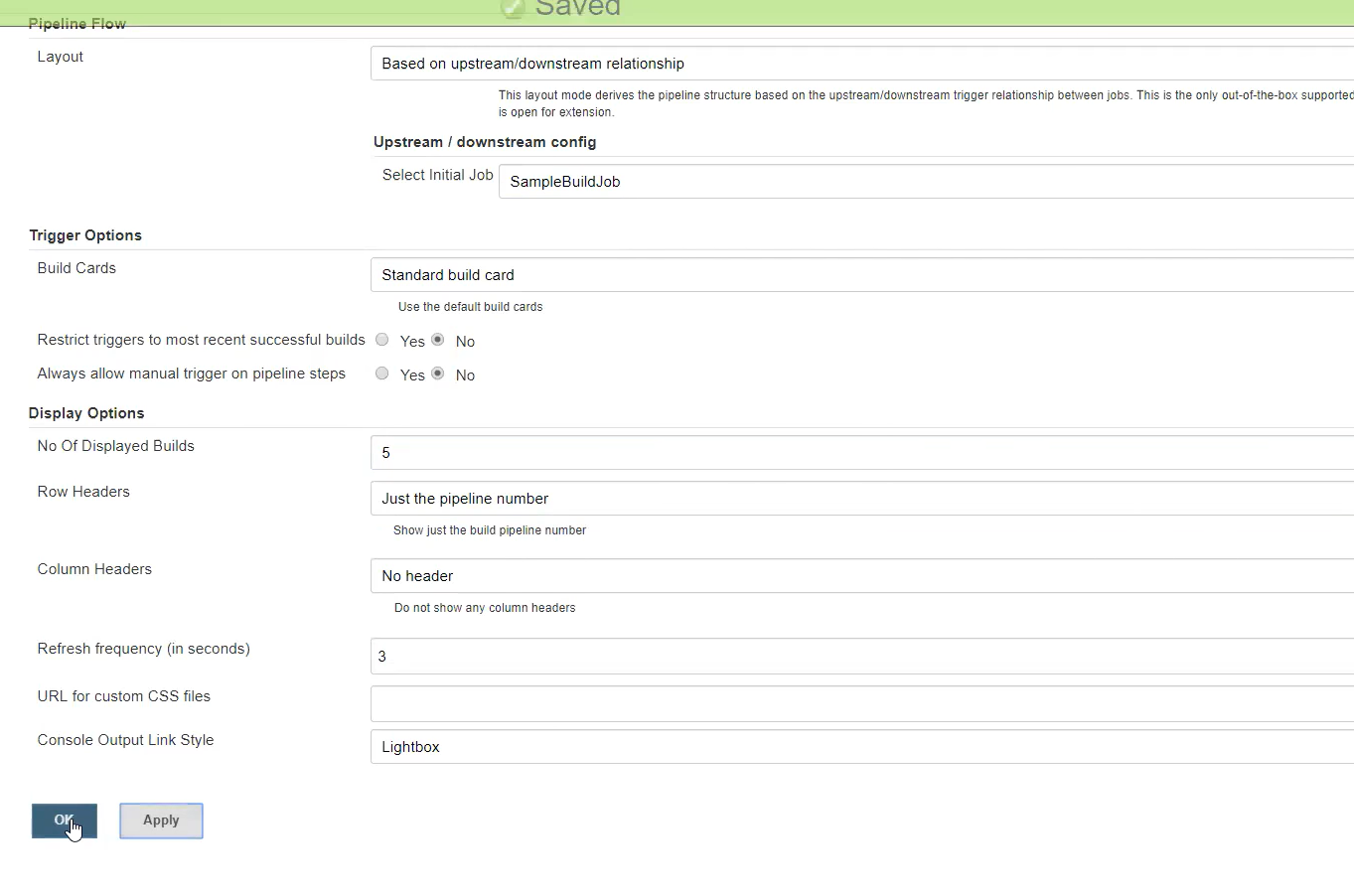


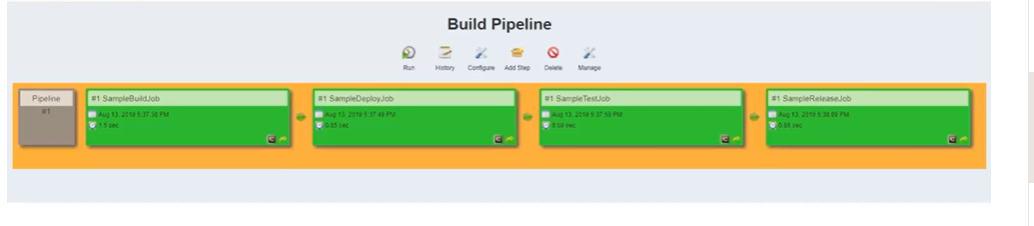
## Now it will give u a configuration page.

## Select initial job under pipeline flow->no of display as 5 -> ok.

A screenshot of a computer

Description automatically generated





## By default, it will show build pipeline.

If click on run option it will run again A computer screen shot of a computer

Description automatically generated A computer screen shot of a diagram

Description automatically generated

## If we have 4 to 5 jobs, then we create the jobs linked together and install plugin and created pipeline and execute(integrated all these jobs) the pipeline to run sequentially.

## But in real time we have n no of jobs we need to create in each phase so if jobs are increasing then it is difficult to link all those jobs

## So, if we have more no of jobs pipeline plugin is not recommended, we must use groovy script.

## So here we create only one pipeline job and inside this pipeline jobs I will create multiple activities. We call them as stages.

## If we run the groovy script it wil automatically create pipeline and execute all stages one after another

# Two Types

## Scripted pipeline

## Declarative pipeline

# Keywords to write script

# Pipeline:

## Entire build process

# Node:

## Some jobs we will run in some node some jobs in other node. Machine to execute pipelines

# Stage:

## Represents multiple steps. Build is one stage, Deploy is one stage etc

# Step:

## Single task

# Scripted pipeline

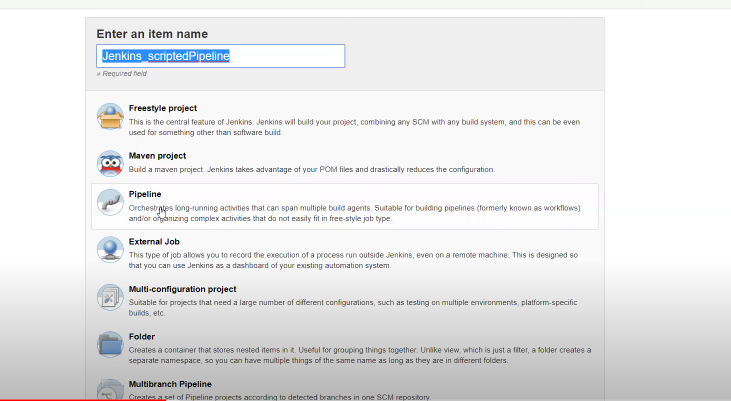
## If you have only single jenkins serve in ur local machine then prefer to use this.

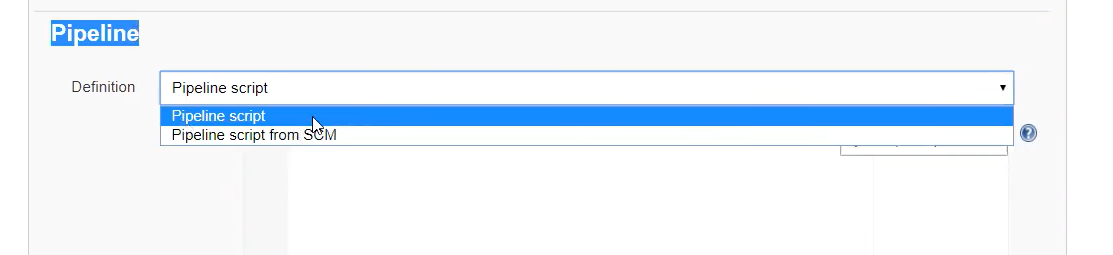
## Write groovy script inside jenkins directly.

A screenshot of a computer

Description automatically generated

## In Jenkins select new item-> give name->select pipeline(install pipeline plugin) to get this in dashboard -> ok





## In configuration page Under pipeline select “Pipeline script” if we want to write scripted Pipeline

## It will start with the node.

## We get the commands from developer or DevOps teams .

# Declarative Pipeline:

## For that we have to create Jenkins file along with project.This file should be part of project code and present in github server.

## Jenkins file is text file contains definiton of pipelineand is checked into source control

## Based on the script created It wil create the pipele and gets executed

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

<https://itzzpankaj004.medium.com/streamlining-node-js-deployment-how-to-use-jenkinsfile-for-effortless-application-delivery-48b4839a68ad>

## In dashboard create new item ->give job name->select pipeline-> under pipe select “pipeline script from SCM” -> Select git in SCM field-> Provide repo URL-> give file name(“jenkinsfile”) in script path.

