Jenkins

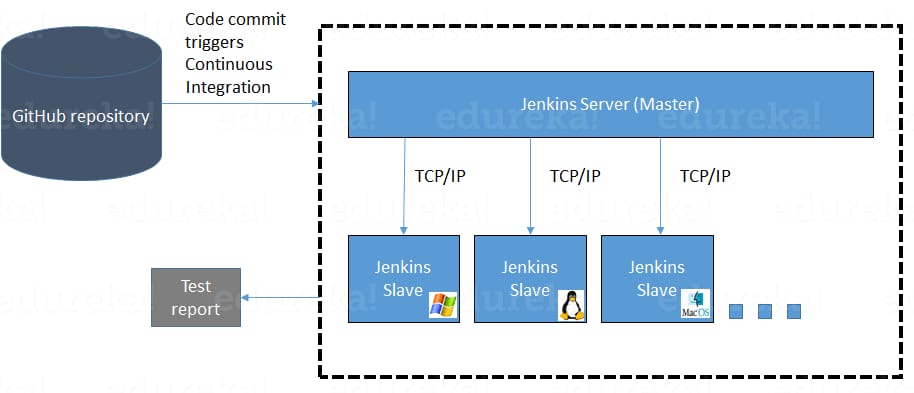
* Java based continuous build system
* Runs in servlet container
* Glass fish, tomcat
* Supported over 400 plugins
* Scm, testing, notifications, reporting artifact saving, triggers, external integration
* Under development since 2005

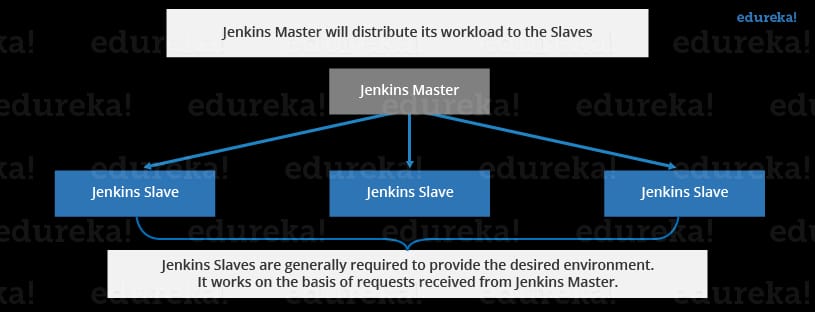
Step one

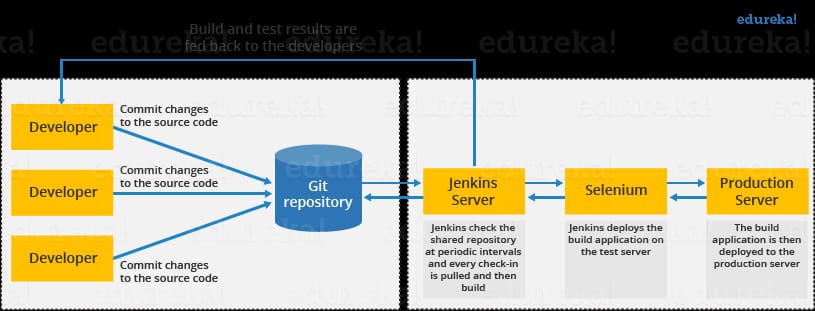
Install java

Yum install java

Step 2 install Jenkins



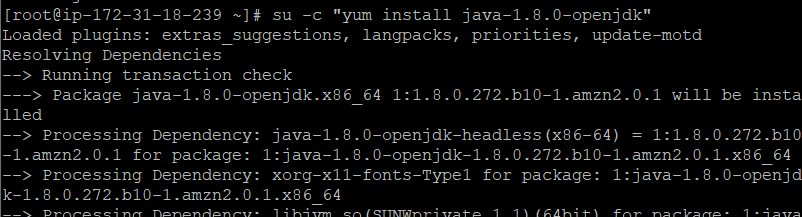


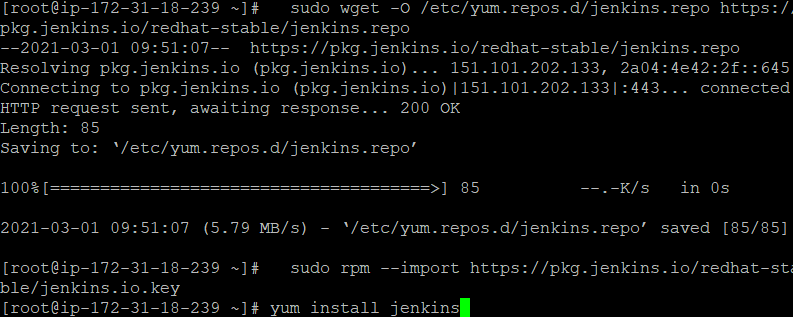


Step 1

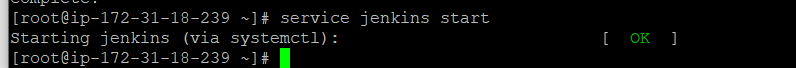
1st download java

After downloading the java install the jenkins



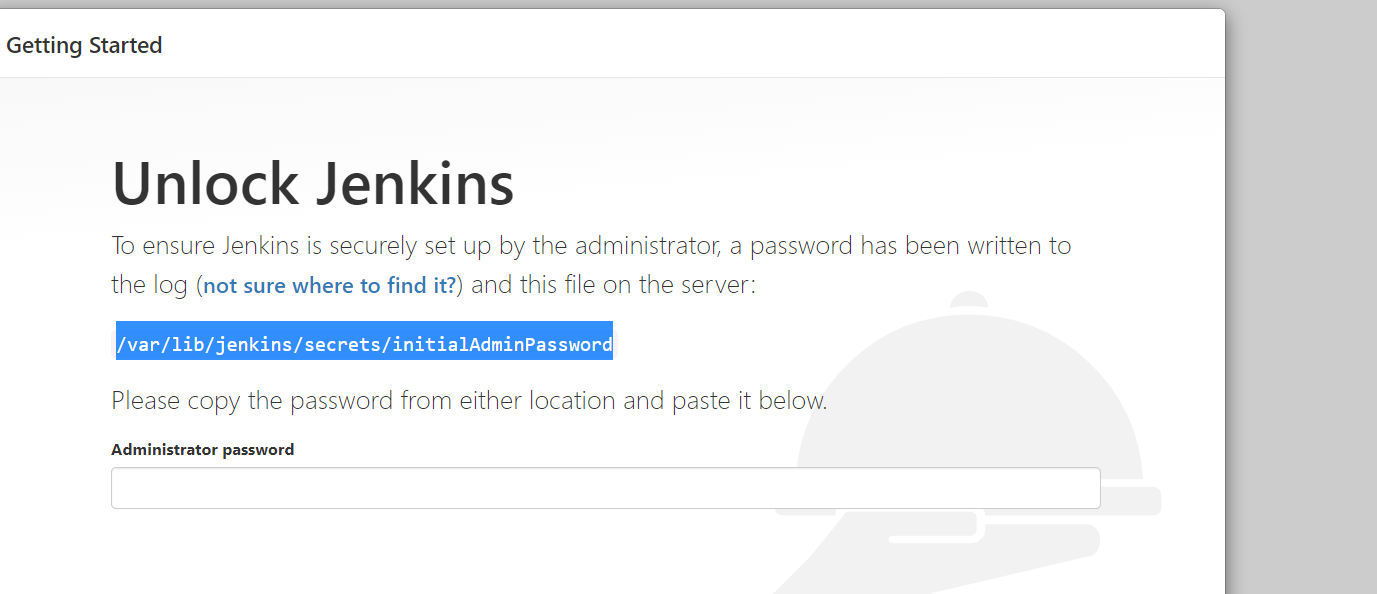


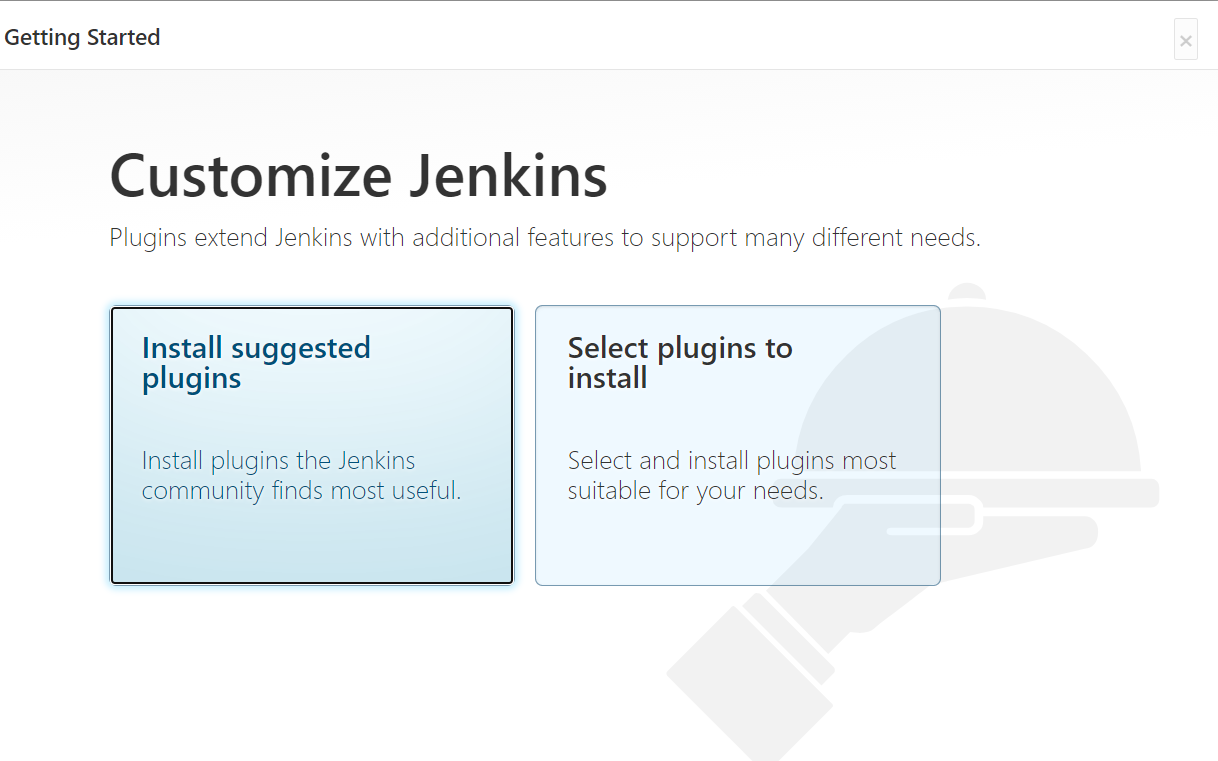
1

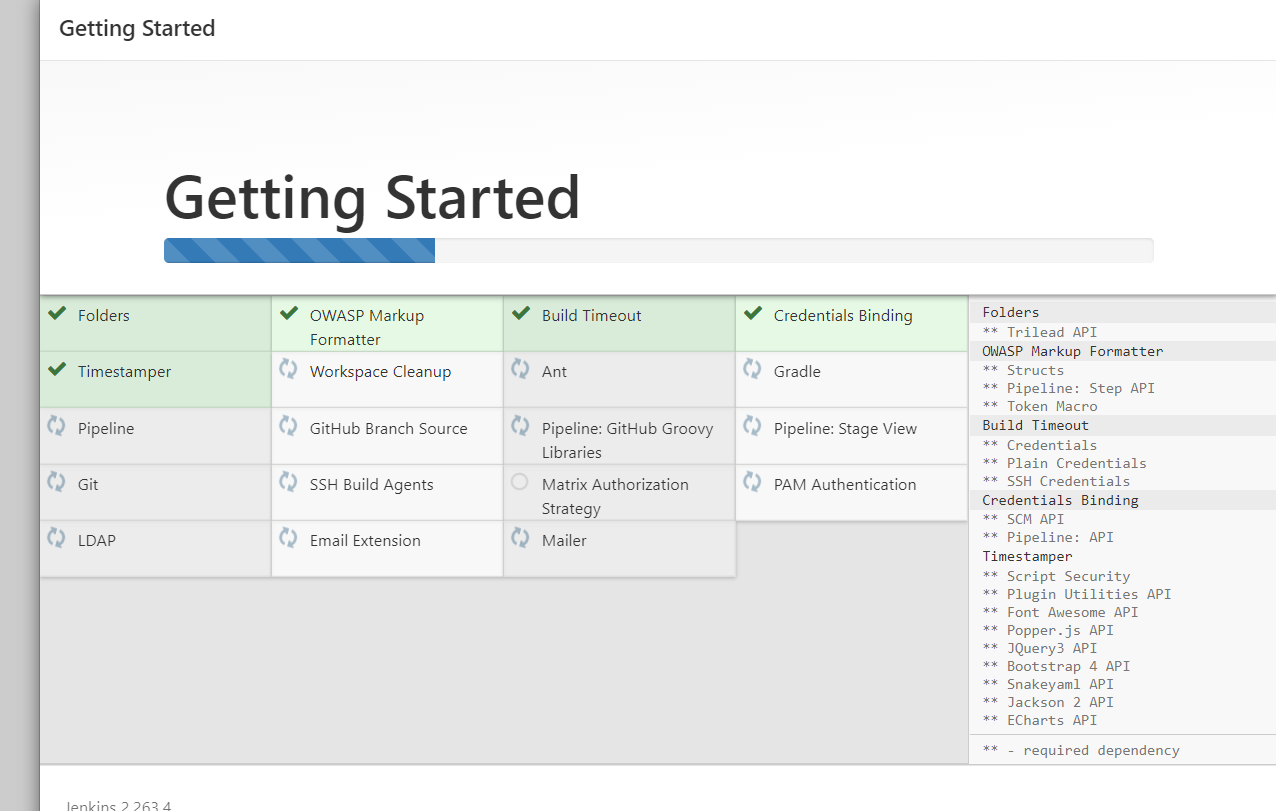


Copy and paste thepublic ip adress

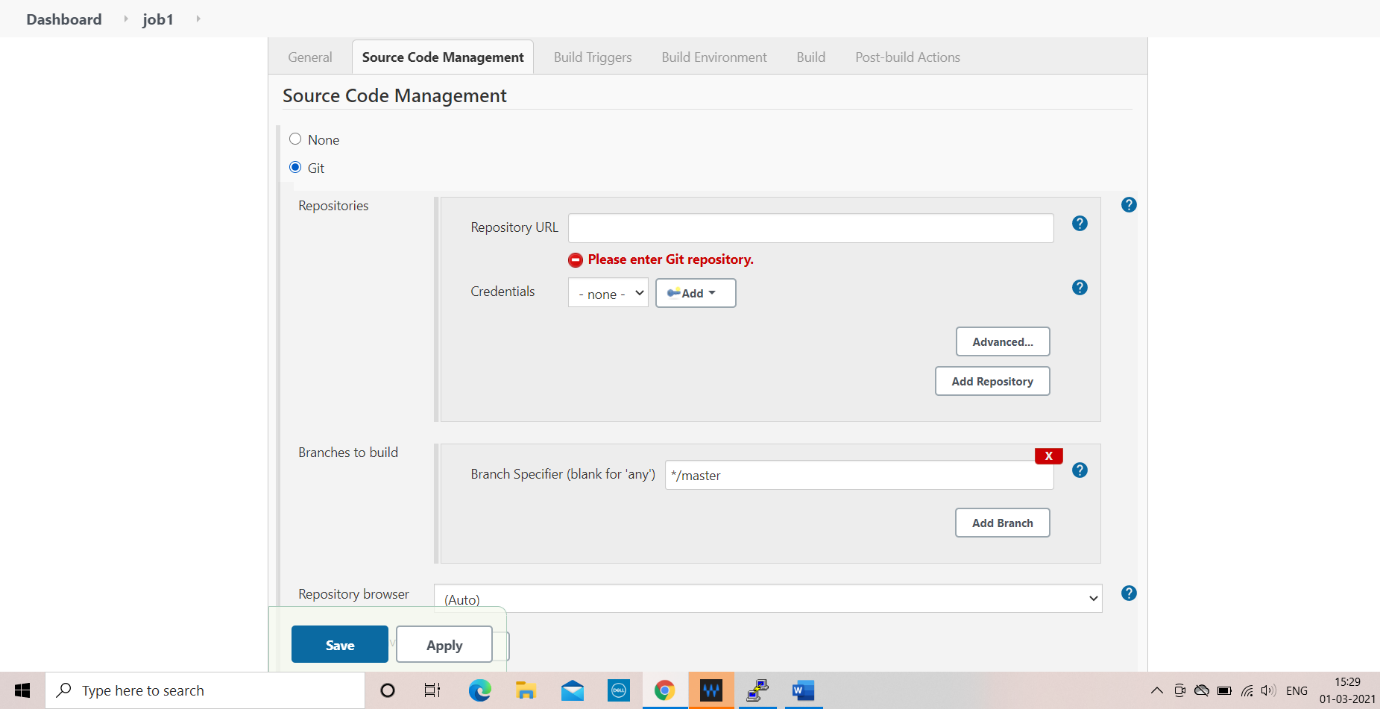


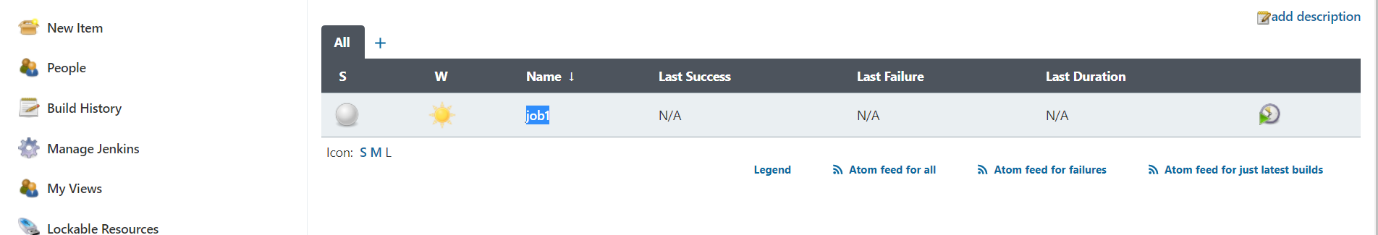


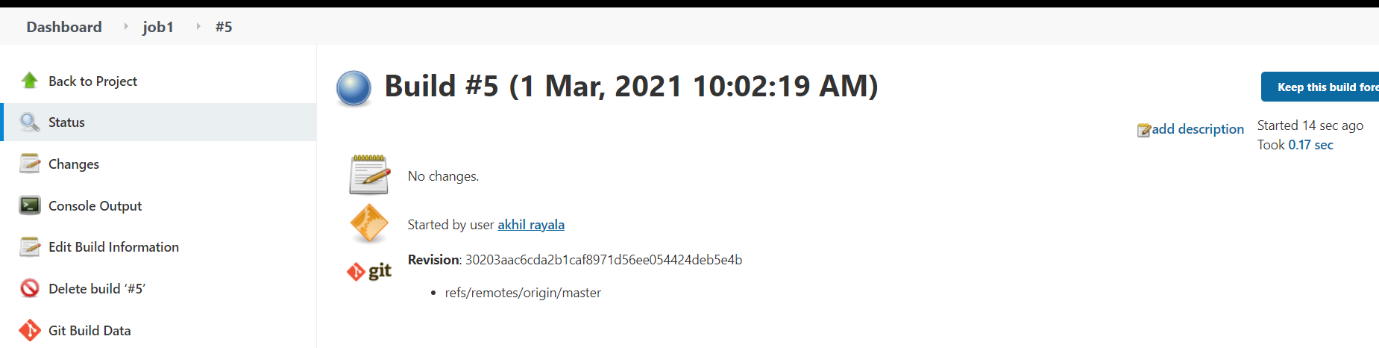




Create a free style job







Maven

* Maven is a build tool
* Java based build tool

Used for packaging

* Resolve dependence issue

Structure

* SRC => main, test
* Pom.xml =>project object model, project details are present

Step 1

* Install maven in Jenkins server

Step 2

* Download maven in GUI
* Intigrate plugin on GUI

Download maven tar or zip file

And extract it

Give short name to the maven

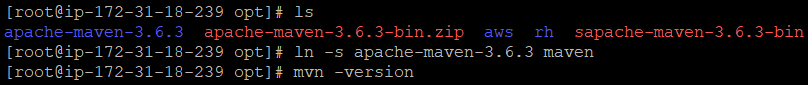
Ln -s apache-maven3.6.3 maven

Give home path to the maven

PATH M2\_HOME='/opt/apache-maven-3.6.3'

PATH="$M2\_HOME/bin:$PATH" export





To check the home path to the maven echo $M2\_HOME is the command

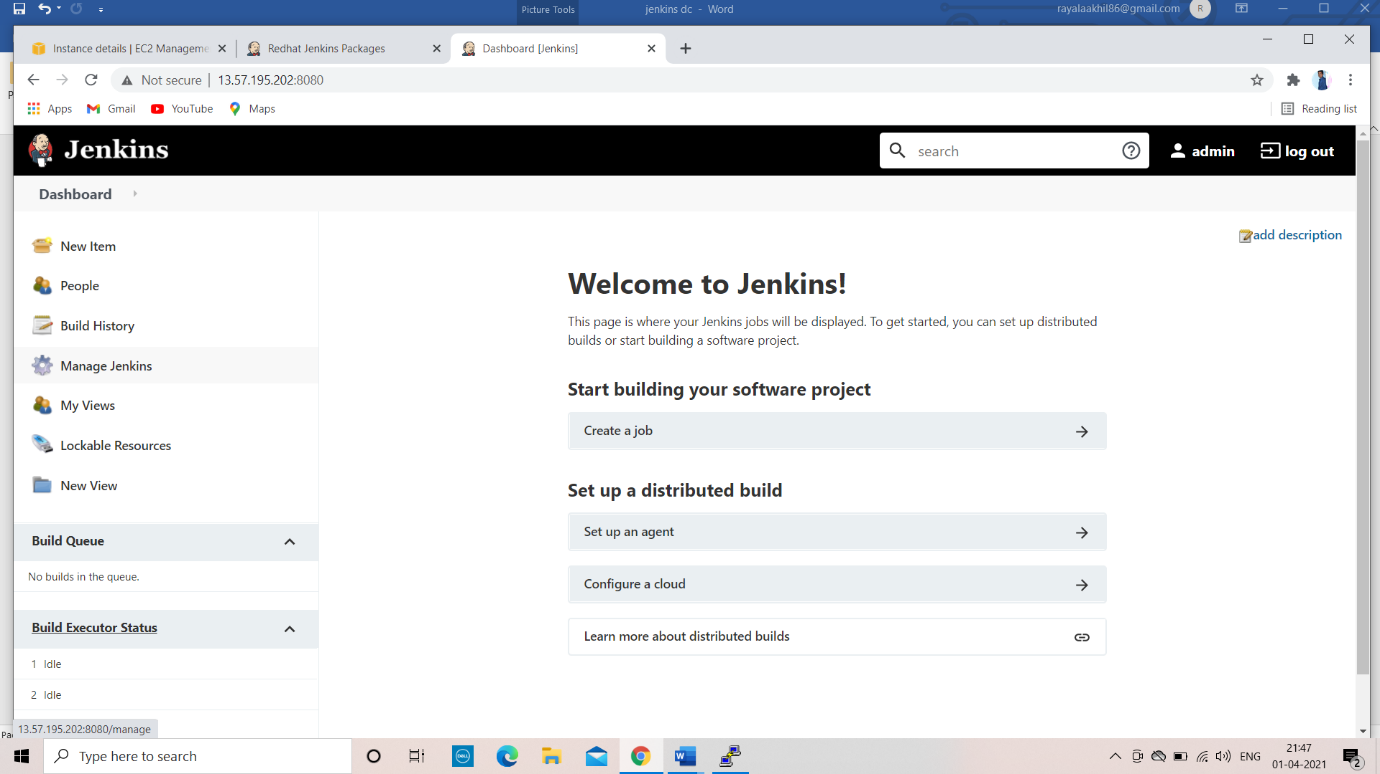


And also provide home path to the java

Echo $JAVA\_HOME is the command to check the home of the java

Start the Jenkins service



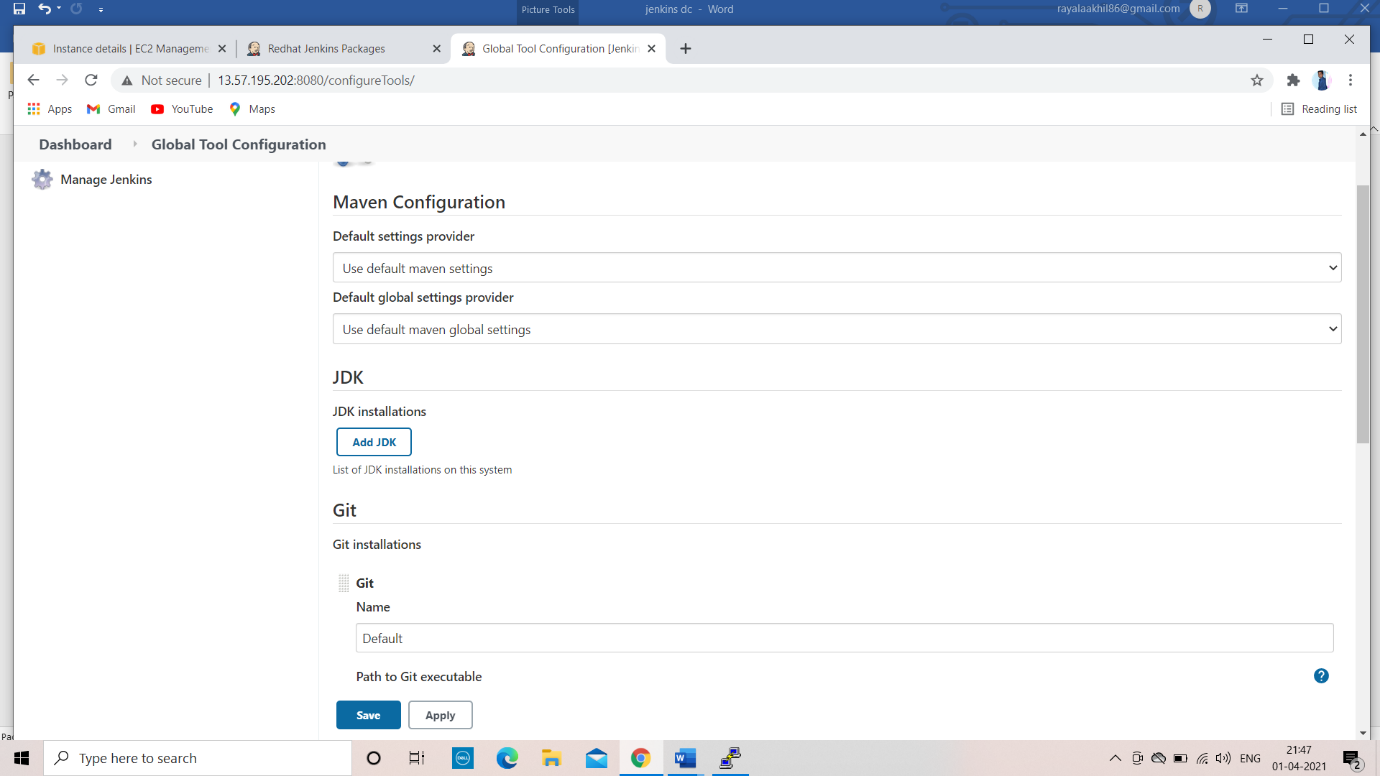


Step 1

* Go to manage Jenkins
* Global tool configuration

JDK

* Provide java home
* provide maven home



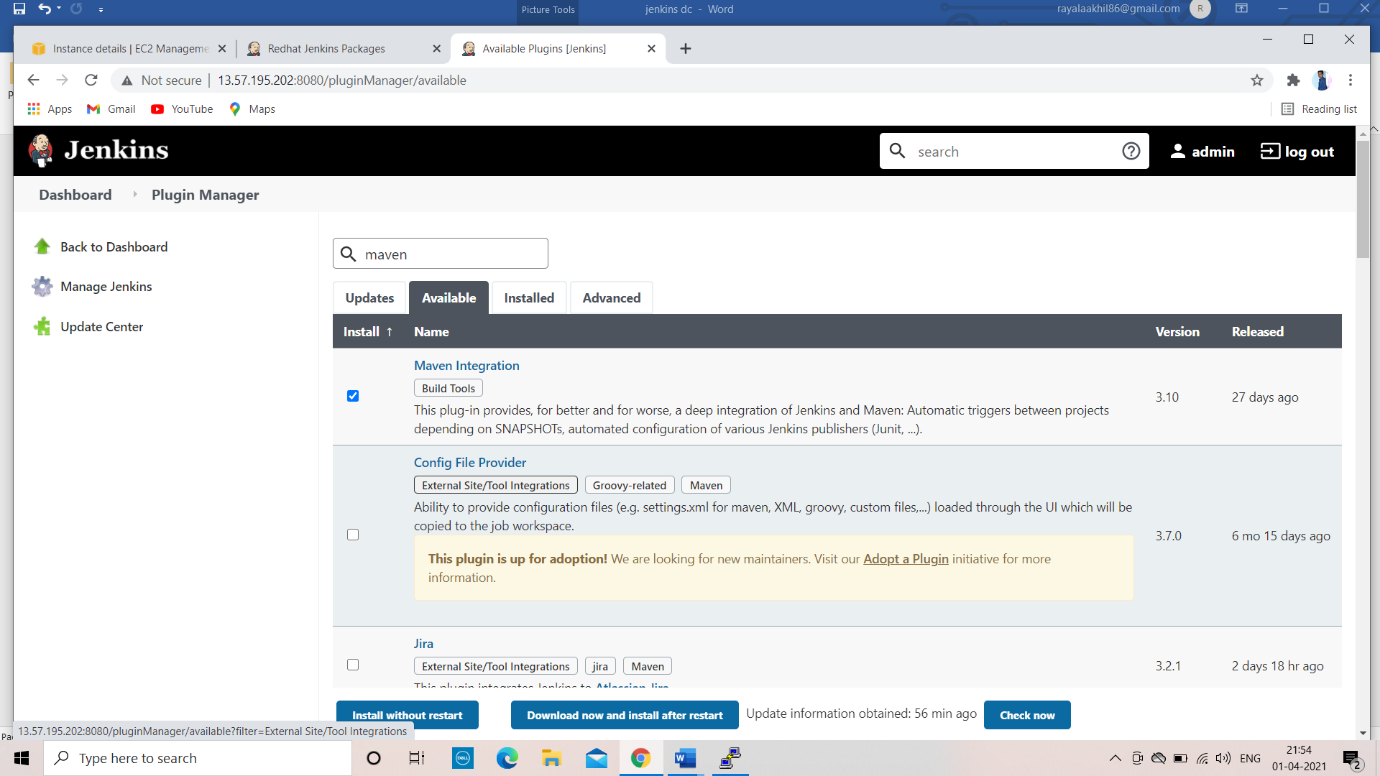
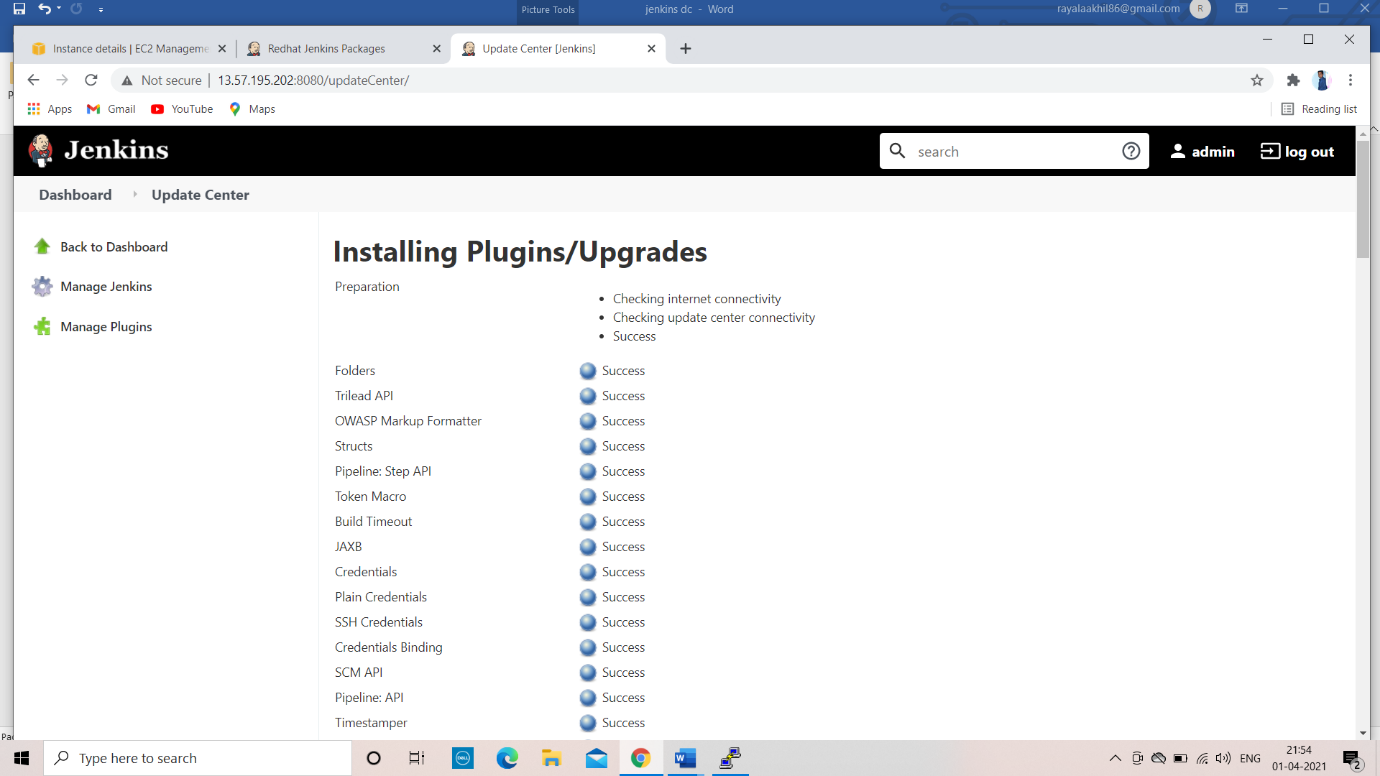
Before creating a job download maven plugin

1. Go to manage jenkins

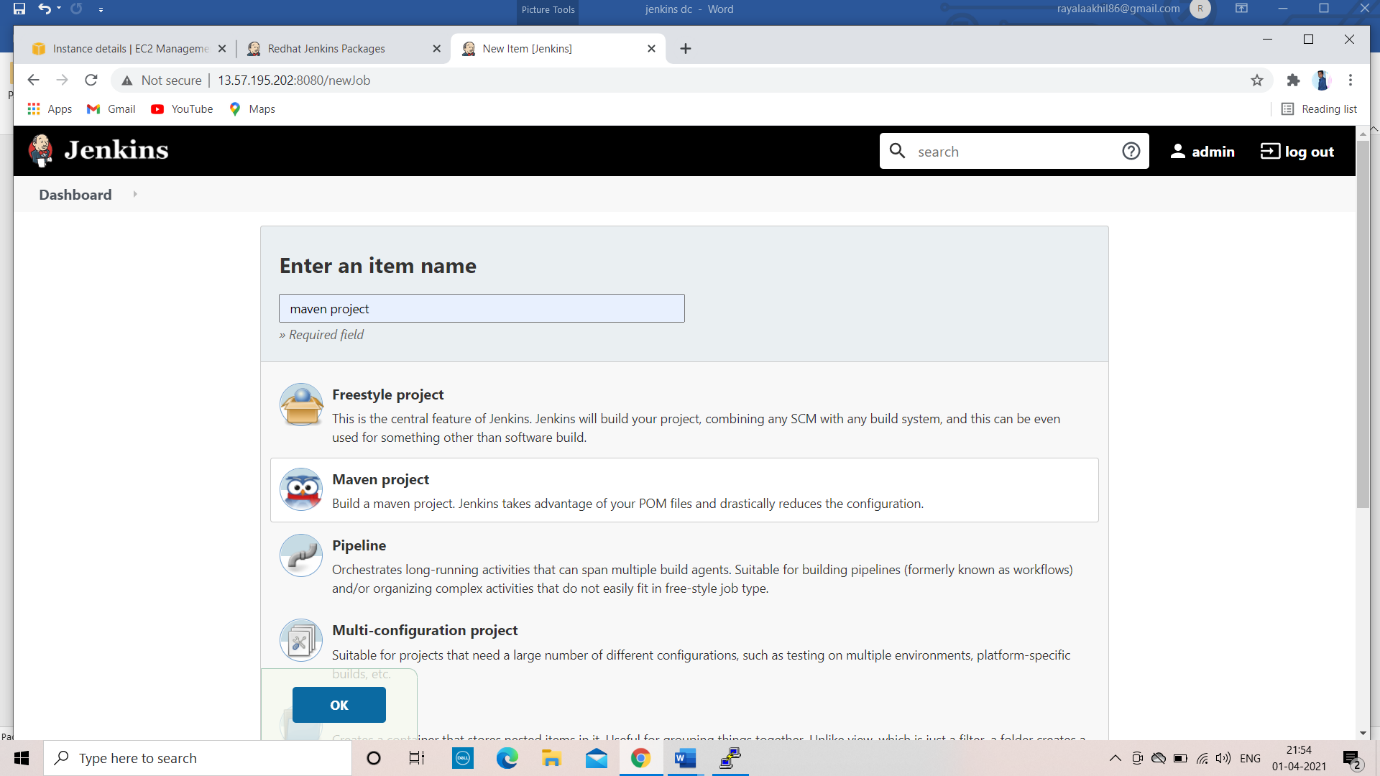
2. Go to manage plugins

3. Download maven integration {maven plugin}

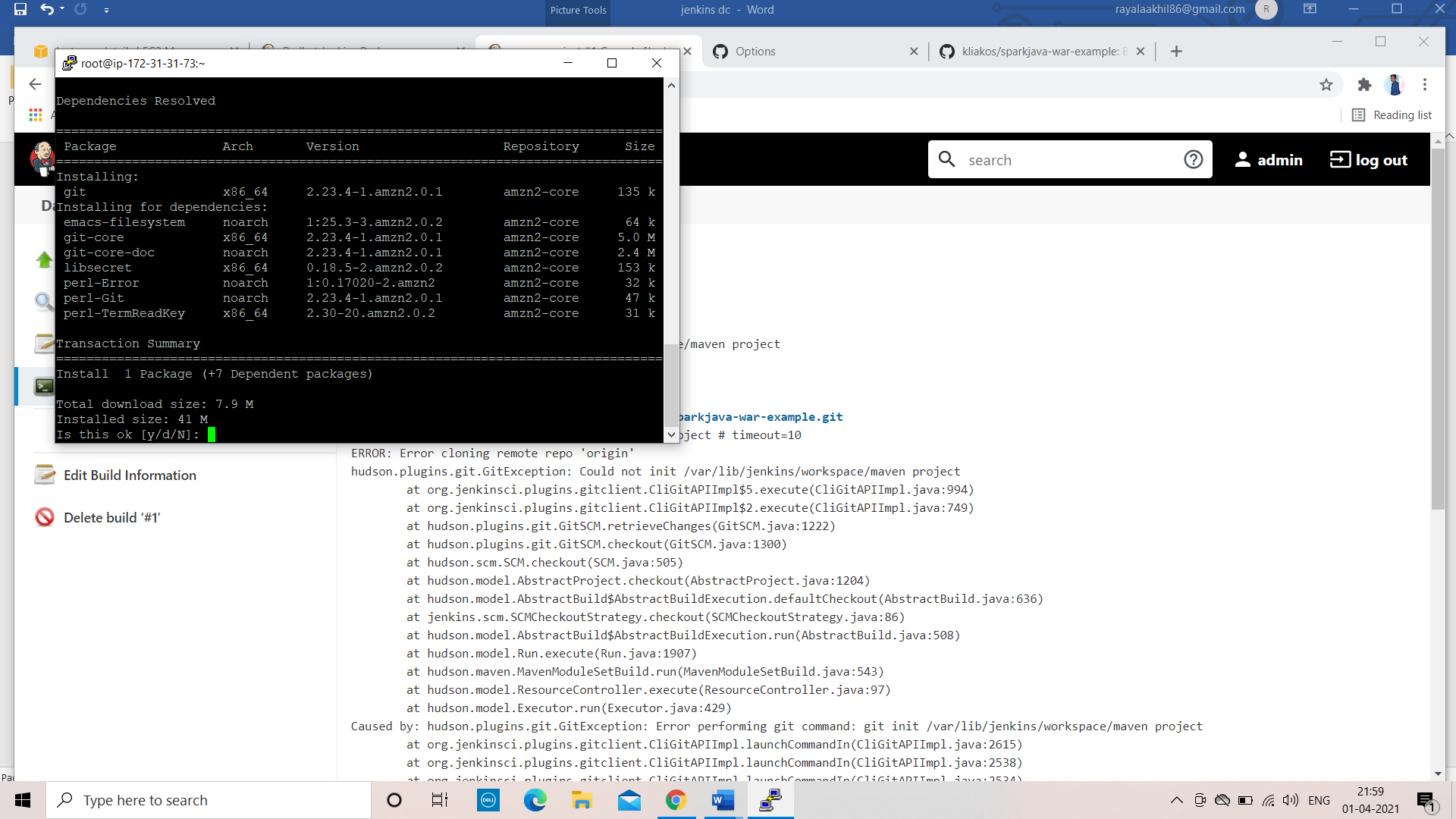
4. create a new job using maven

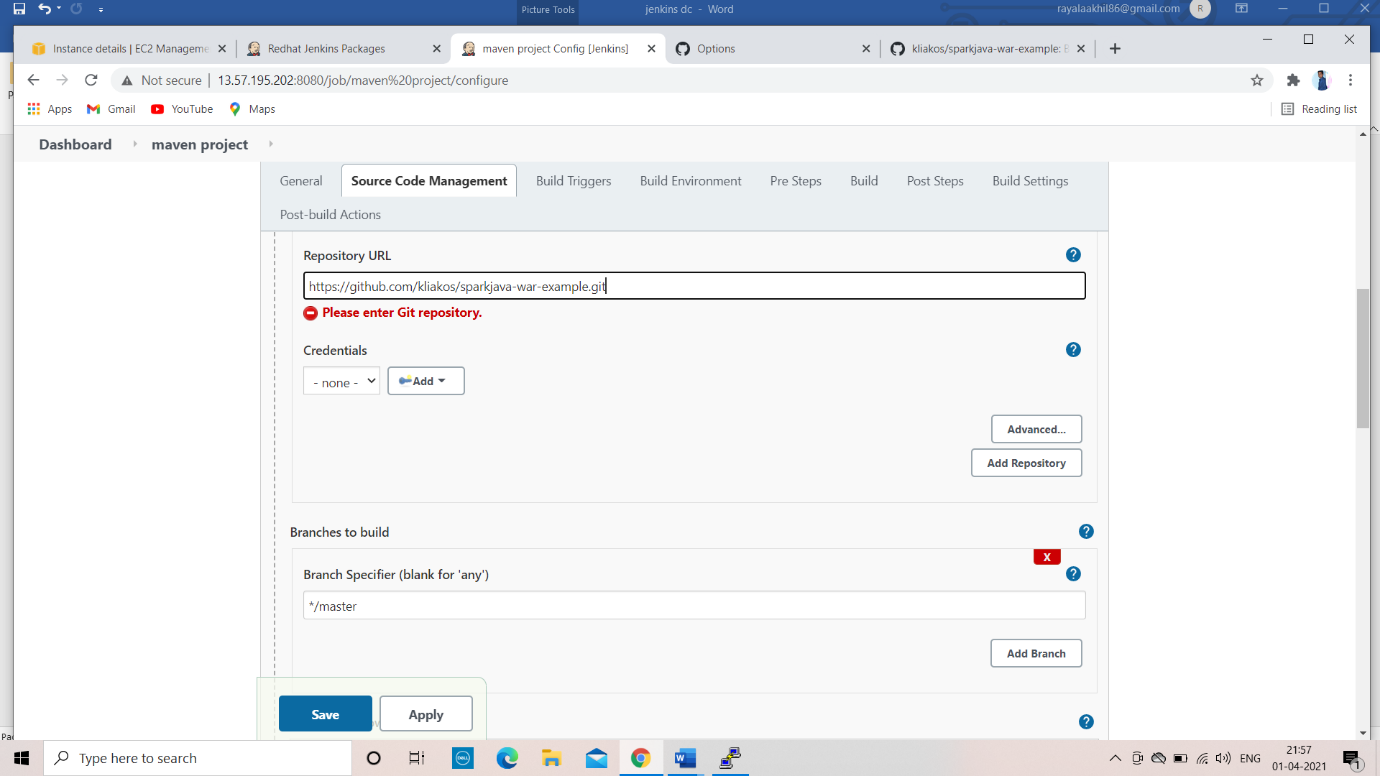
Create a new job using maven project



Download git in ur machine

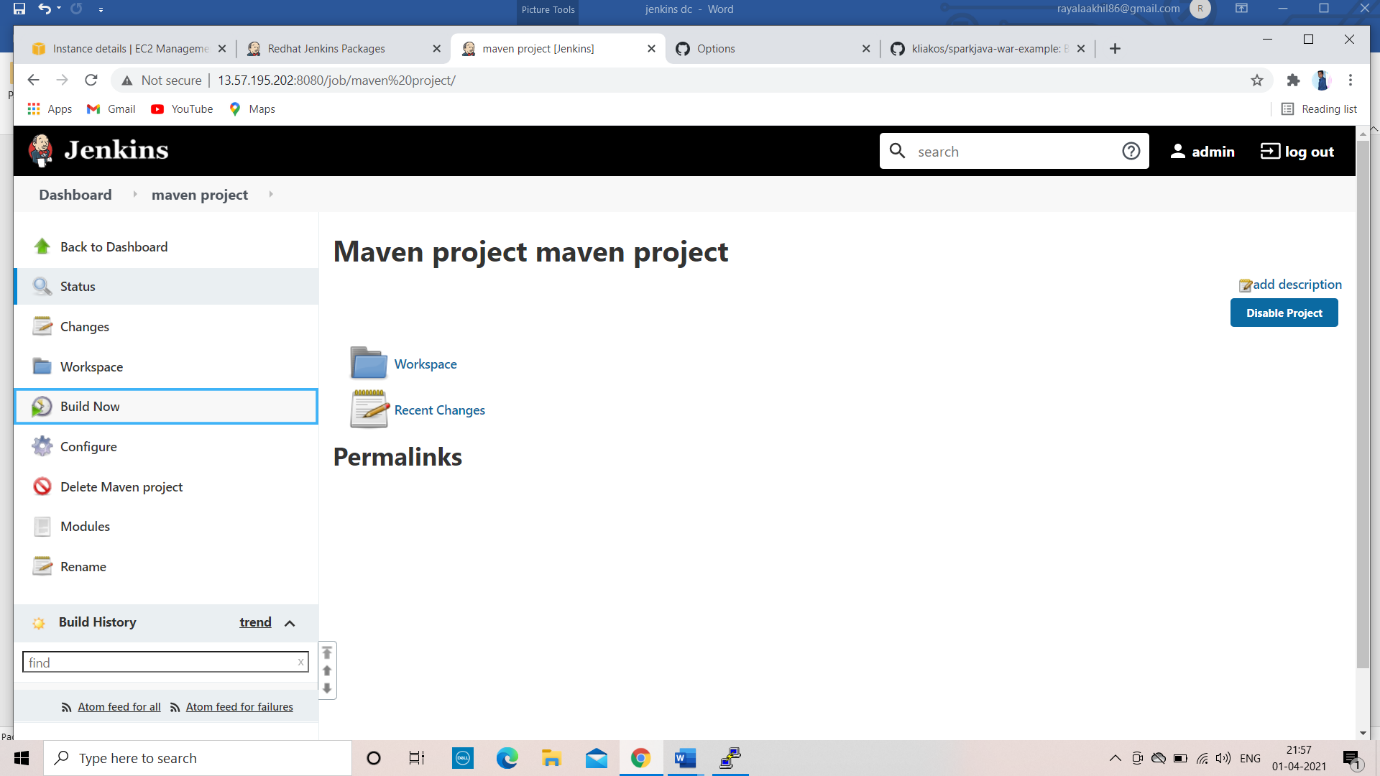


Go to source code management provide you repository url [maven project ][git]



And go to maven and give name and source as package

Then build the job





Creating MAVEN project

Use the command

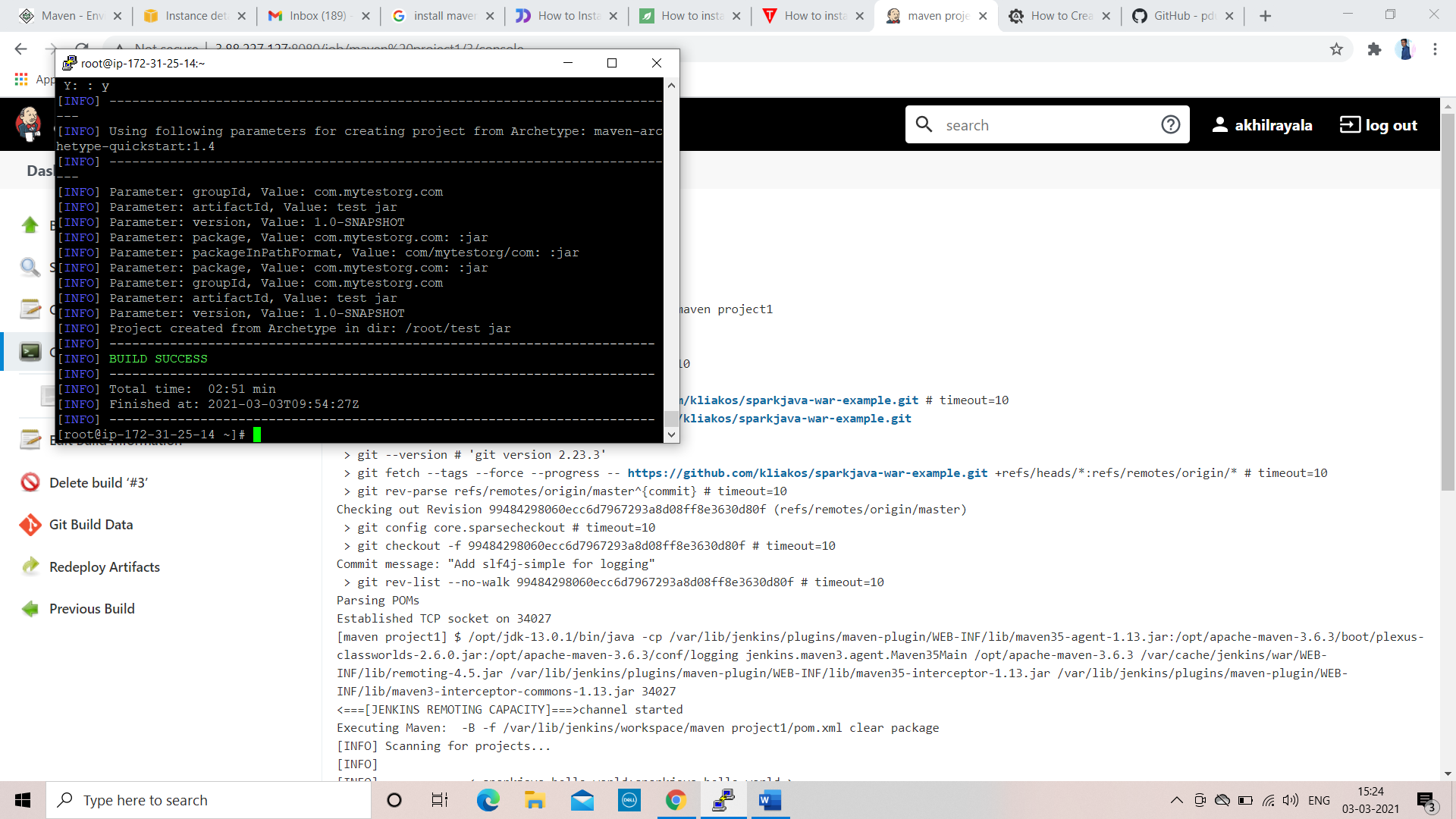
* mvn archtype:generate

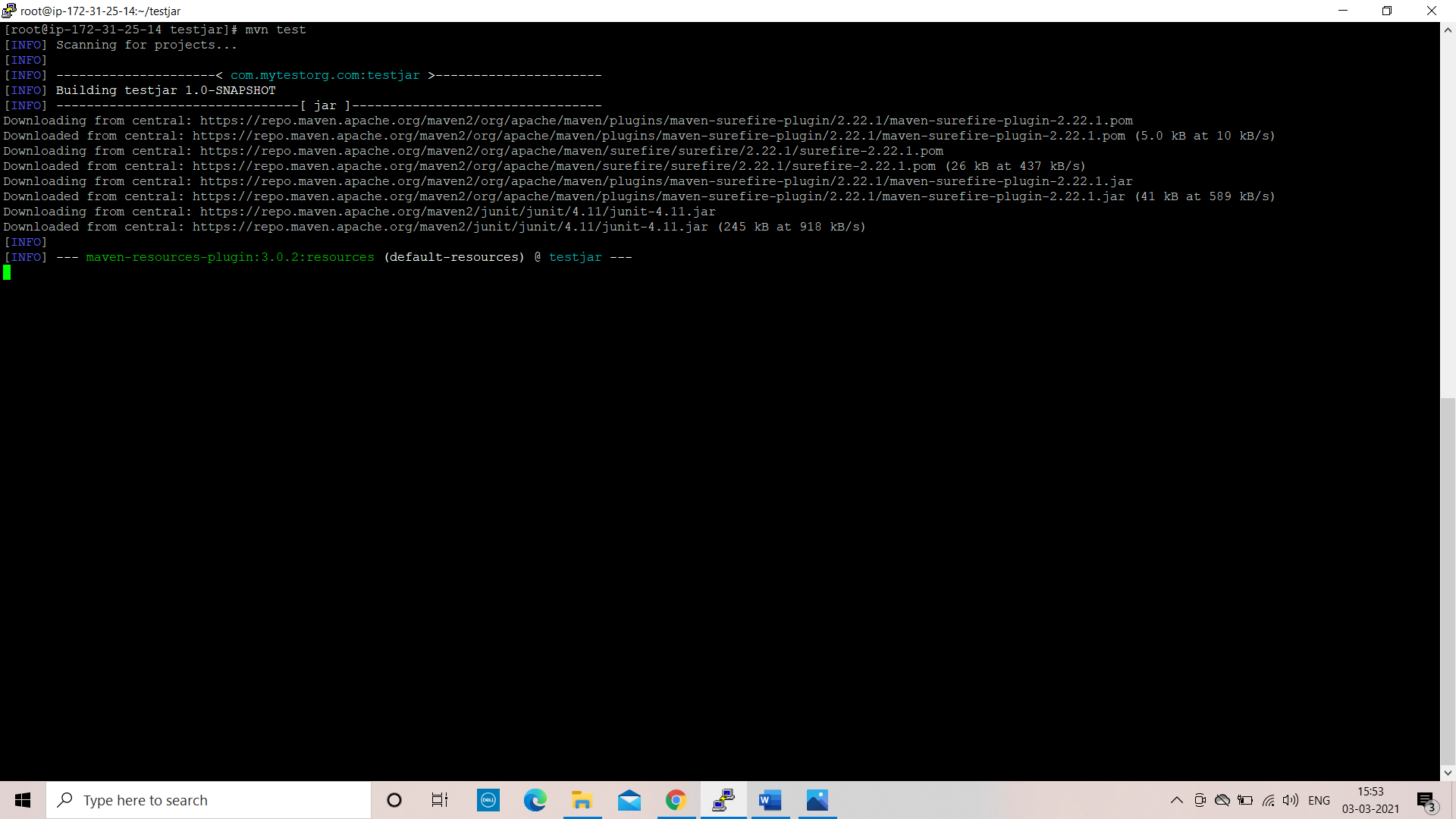
MAVEN BUILD LIFECYCLE

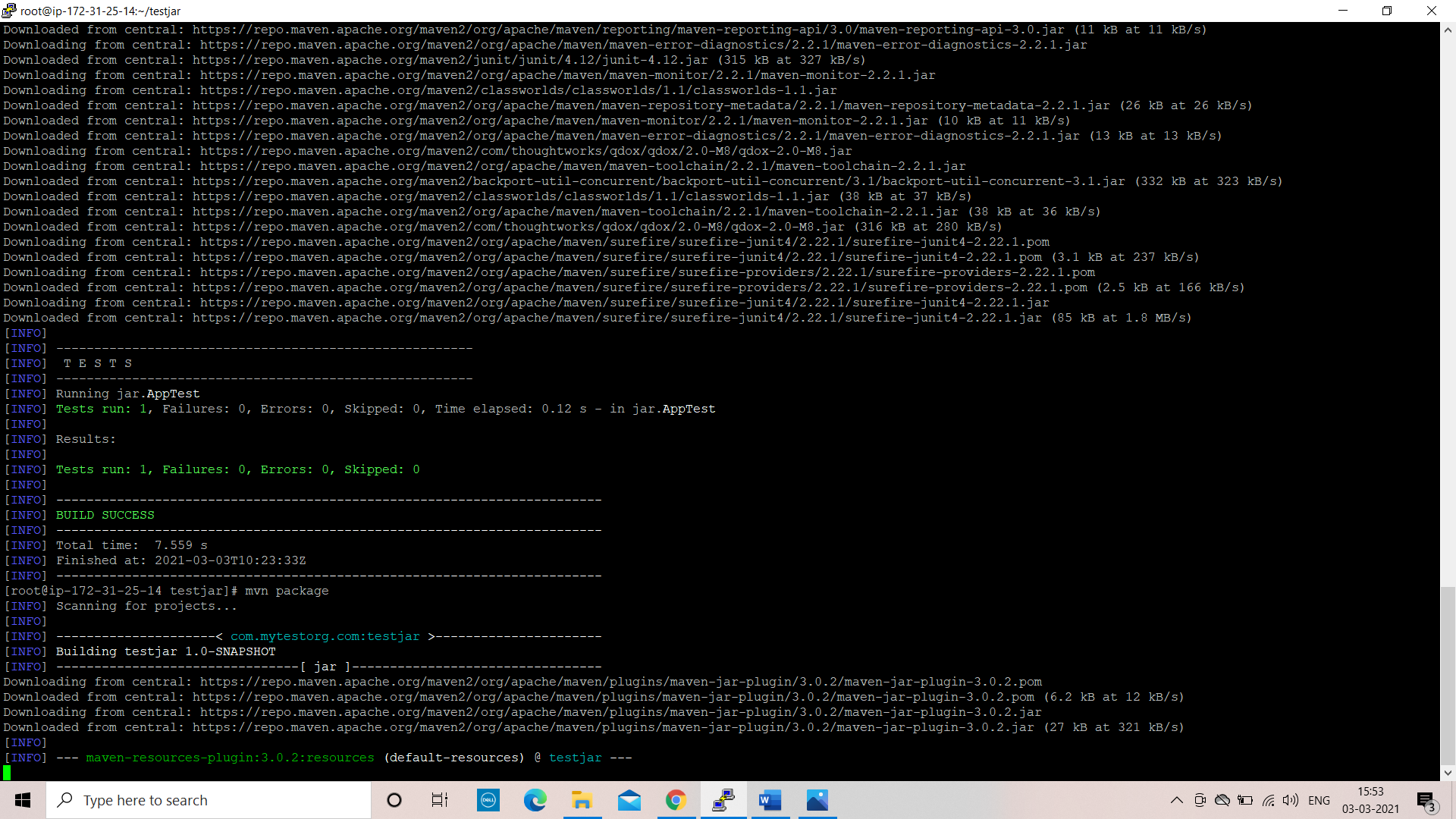
* mvn validate => the command validate maven project that everything is correct and all the necessary information available.
* mvn compile => it is used to compile the source java classes of the project
* mvn test compile =>the command compiles the test classes of the maven project
* mvn package =>the command builds the maven project and packages them into JAR, WAR etc.
* mvn install => the command builds the maven project and install the project files into {JAR, WAR, pom.xml, etc.} into local repository
* mvn deploy =>the command used to deploy the artifact to the remote repository
* the remote repository configured properly in project pom.xml file

CREATING A SAMPLE PROJECT

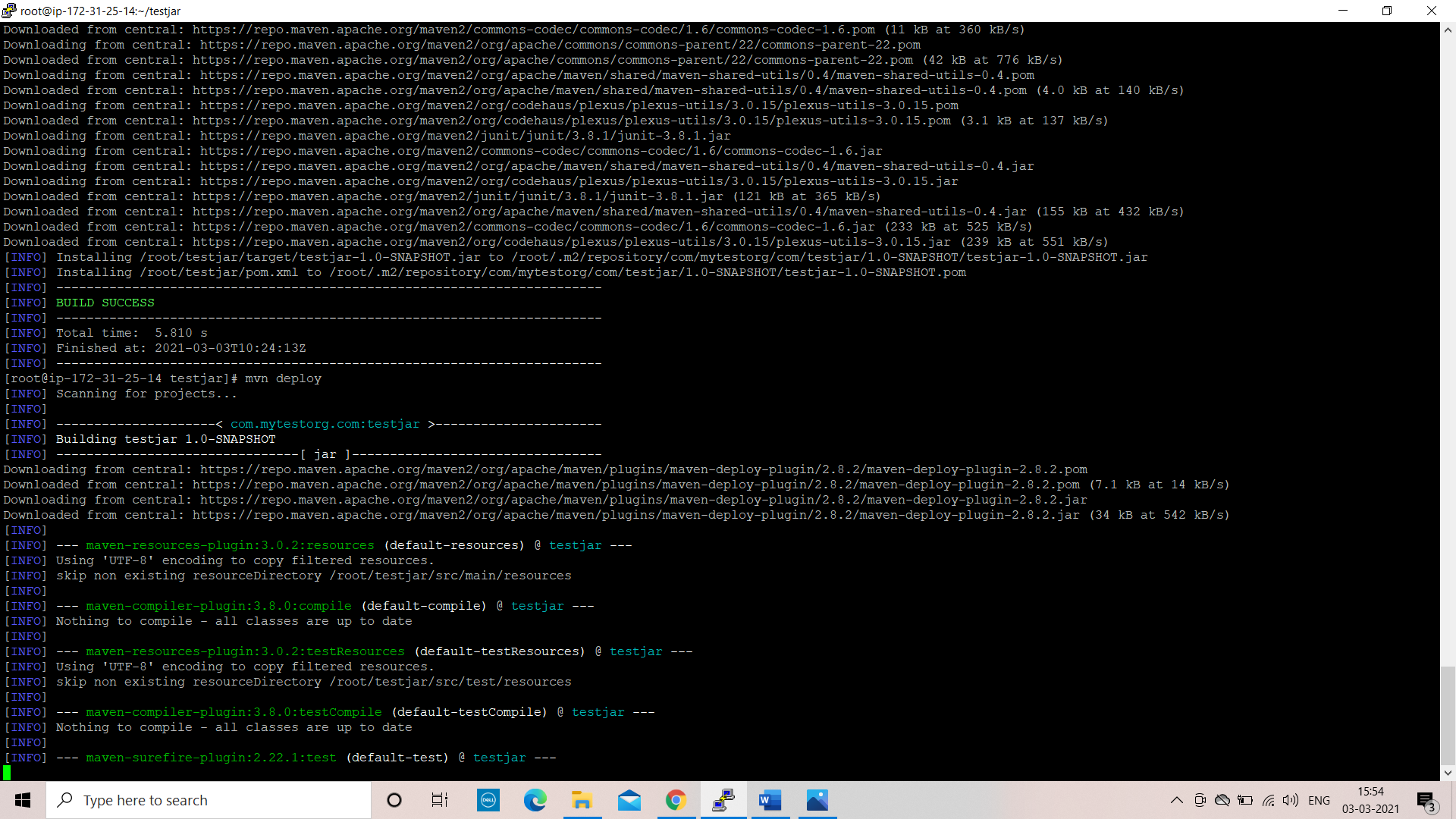
* define group id => com.mytestorg.com
* artifactid => test jar
* version => 1.0-snapshot: SNAPSHOT
* package name =>com.mystestorg.com: : jar









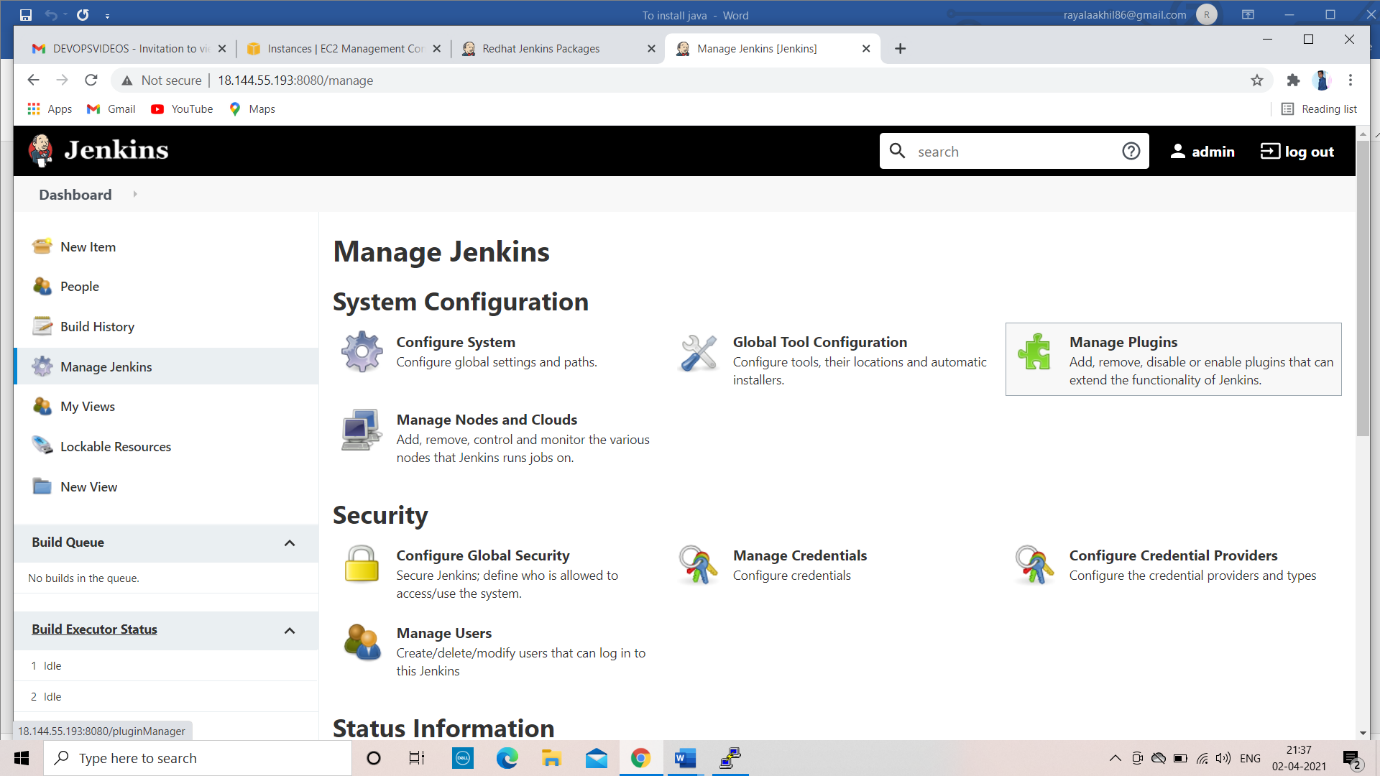


Pipeline

Sequence of the job

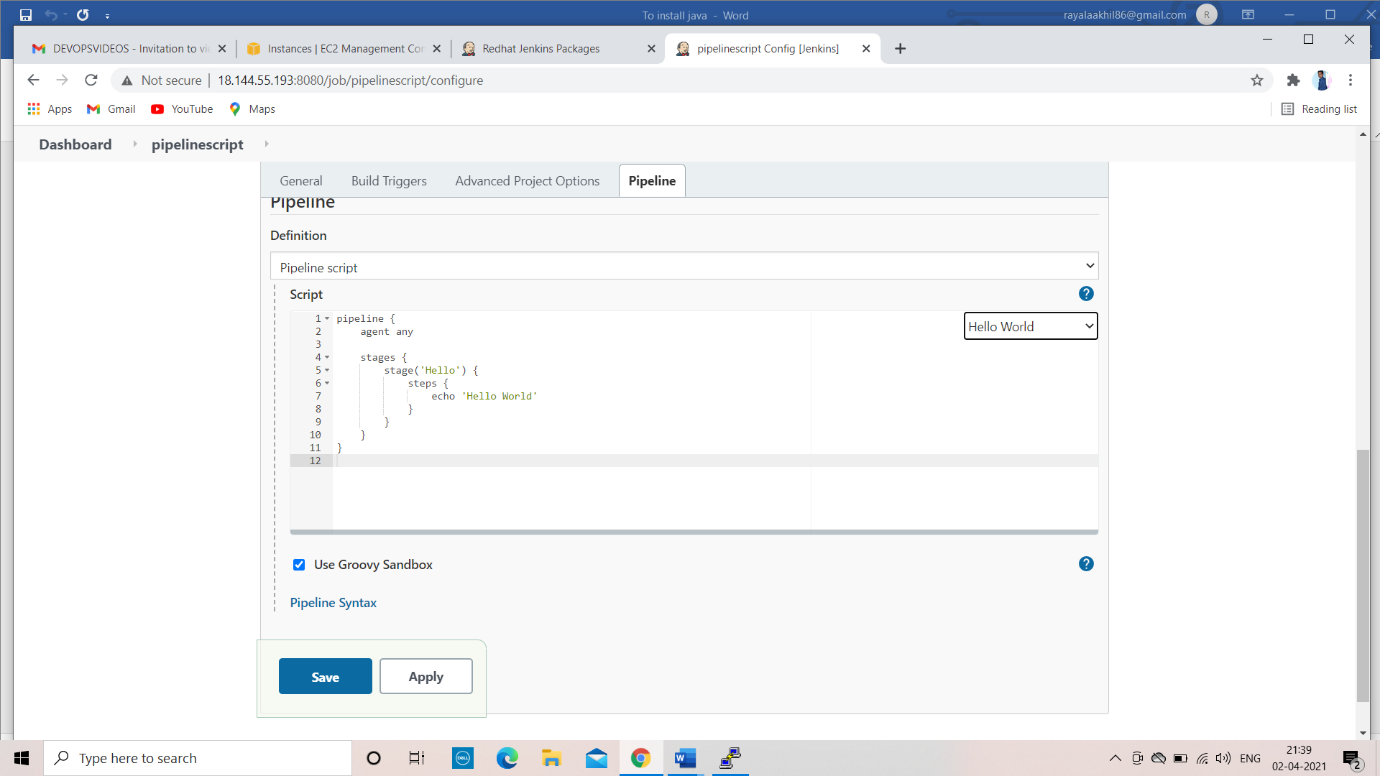
Pipeline is a workflow with group of events or jobs that are chained and integrated with each other in sequence in pipeline terms

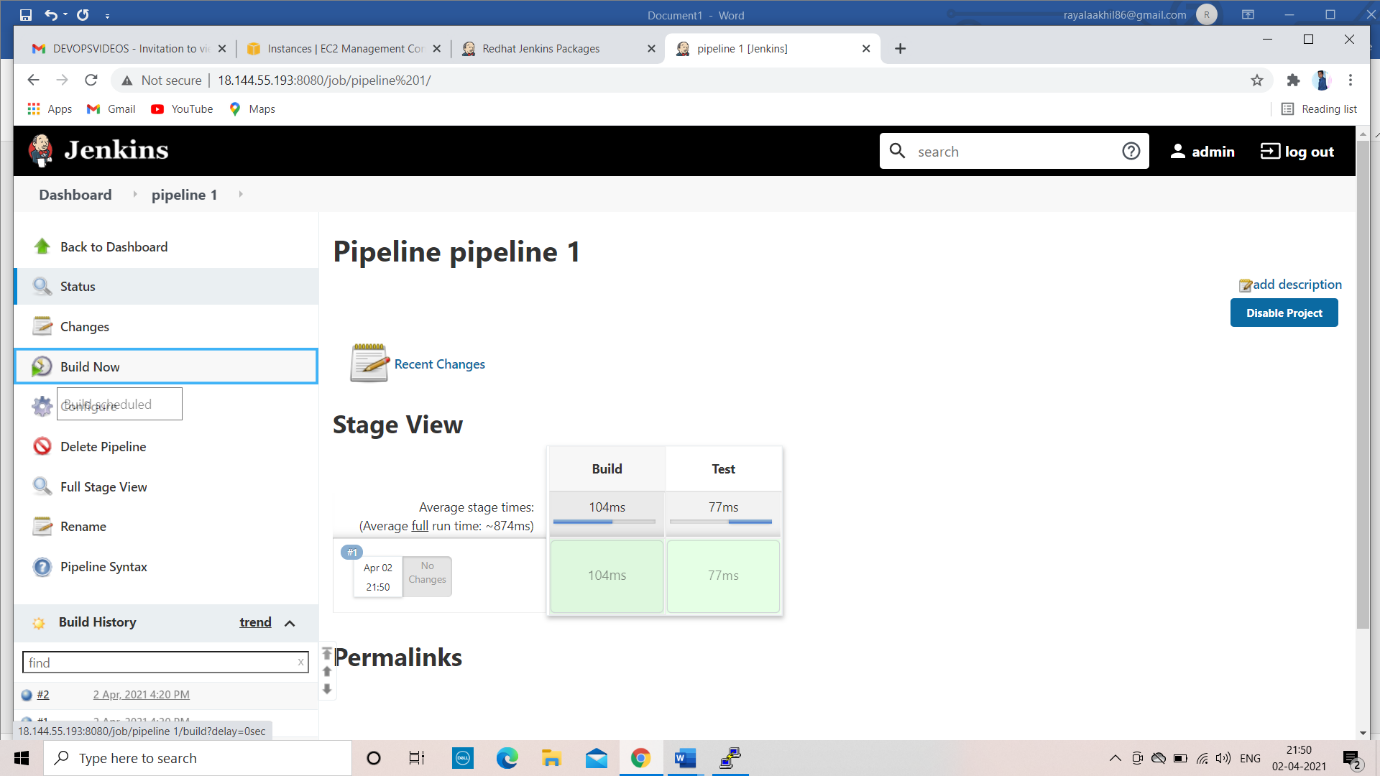
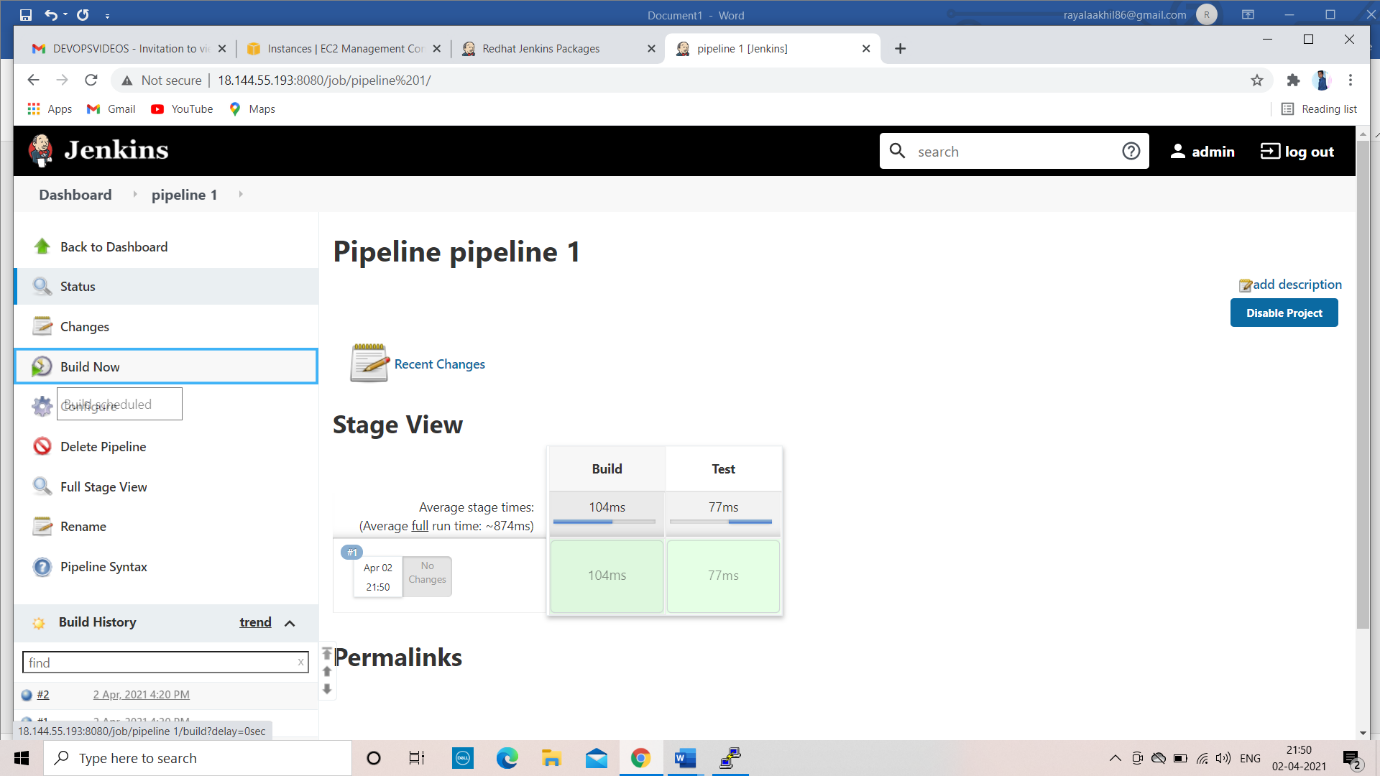
Go to manage plugins download pipeline

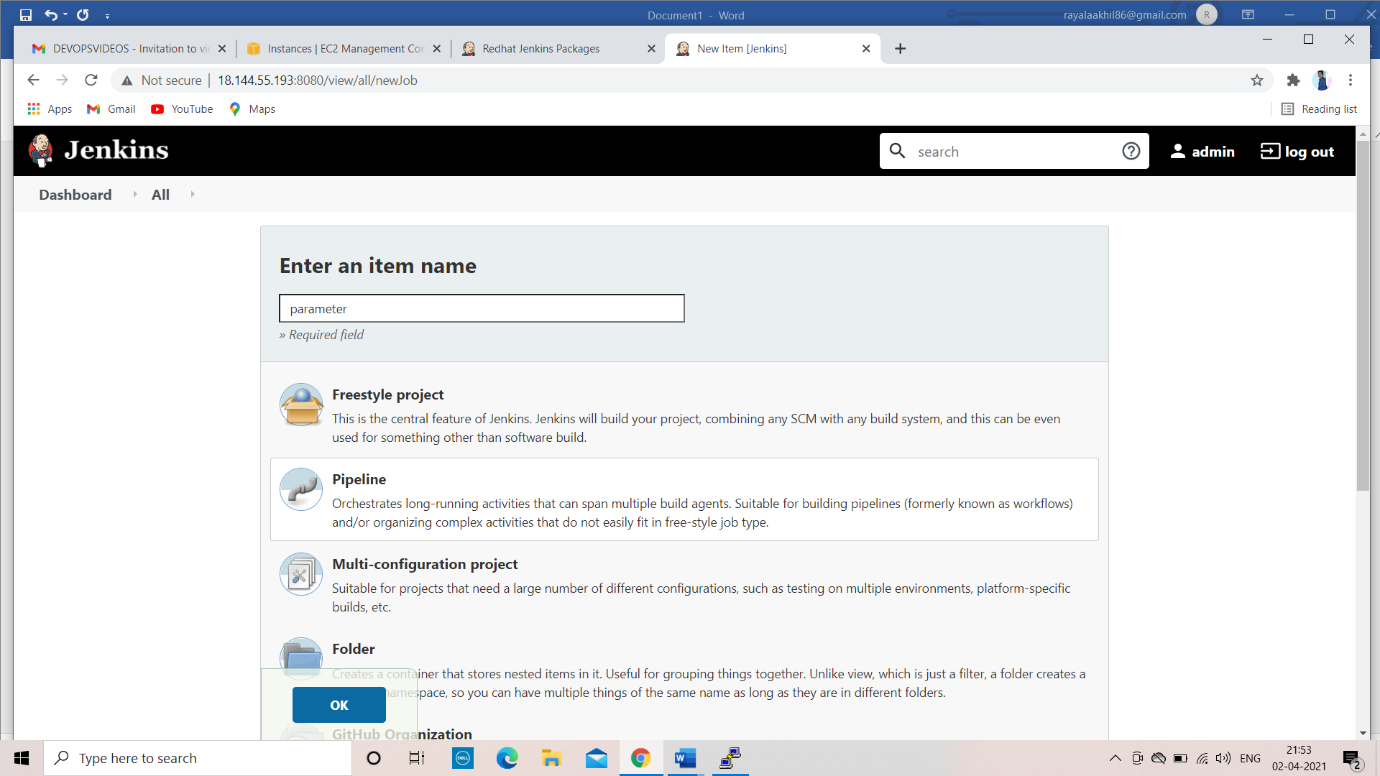


Create a pipeline script

sample





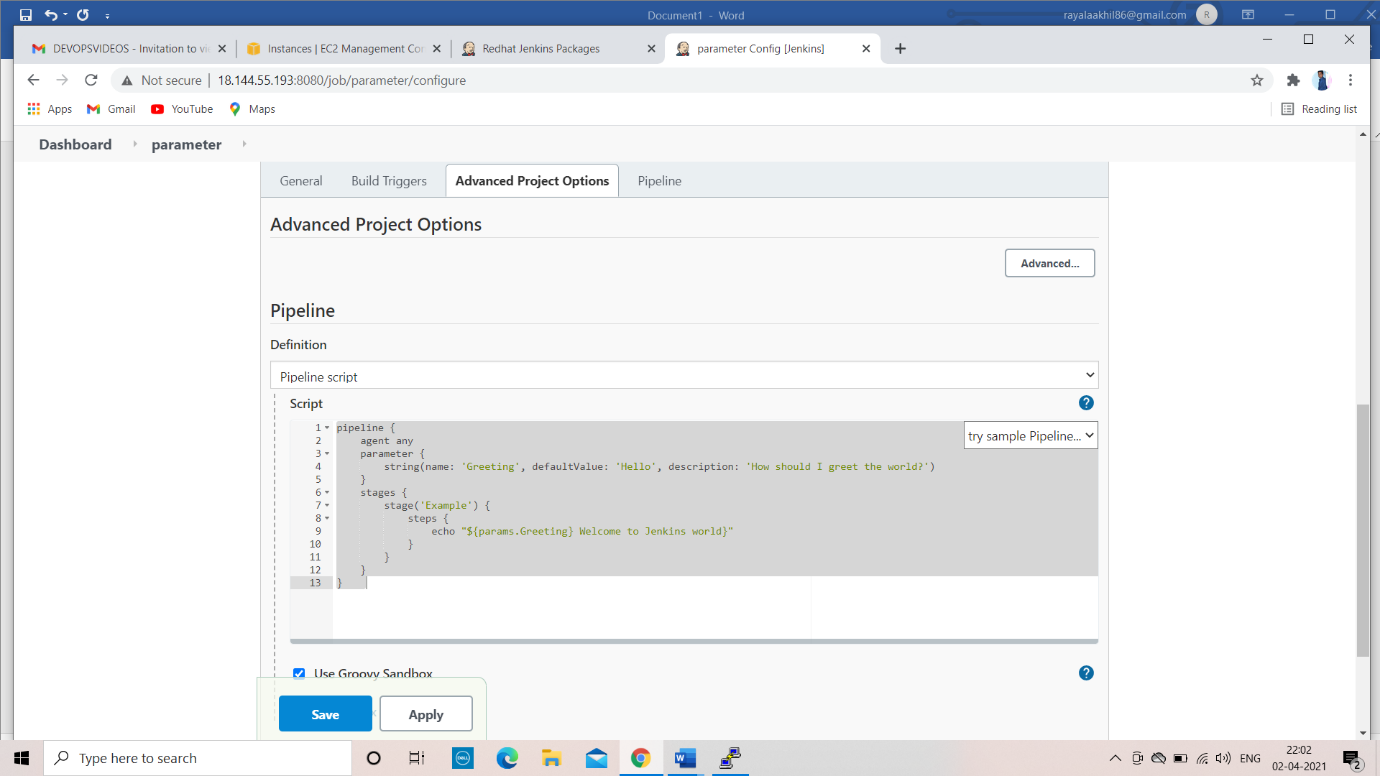


Pipe line parameter

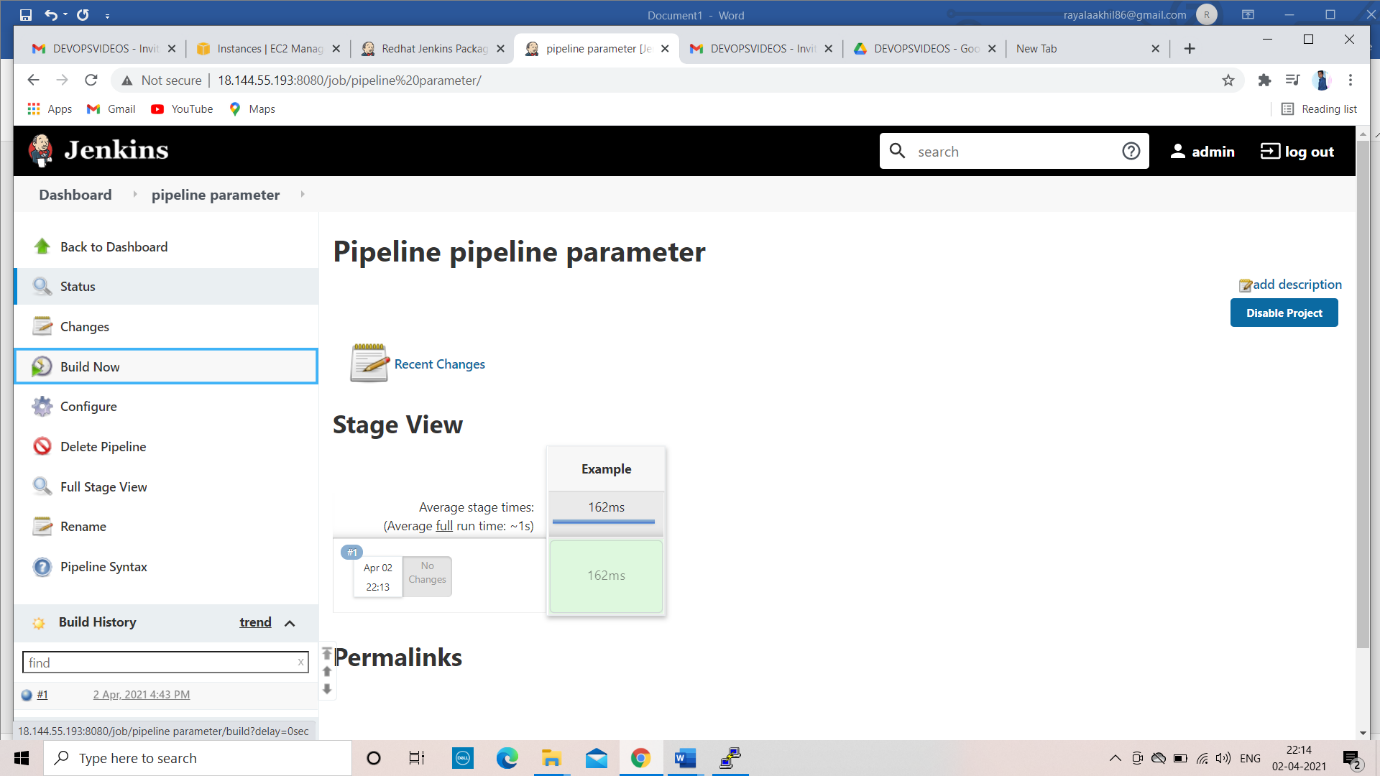
parameter is nothing but run time environment variables

you want pass some input in you build use parameter

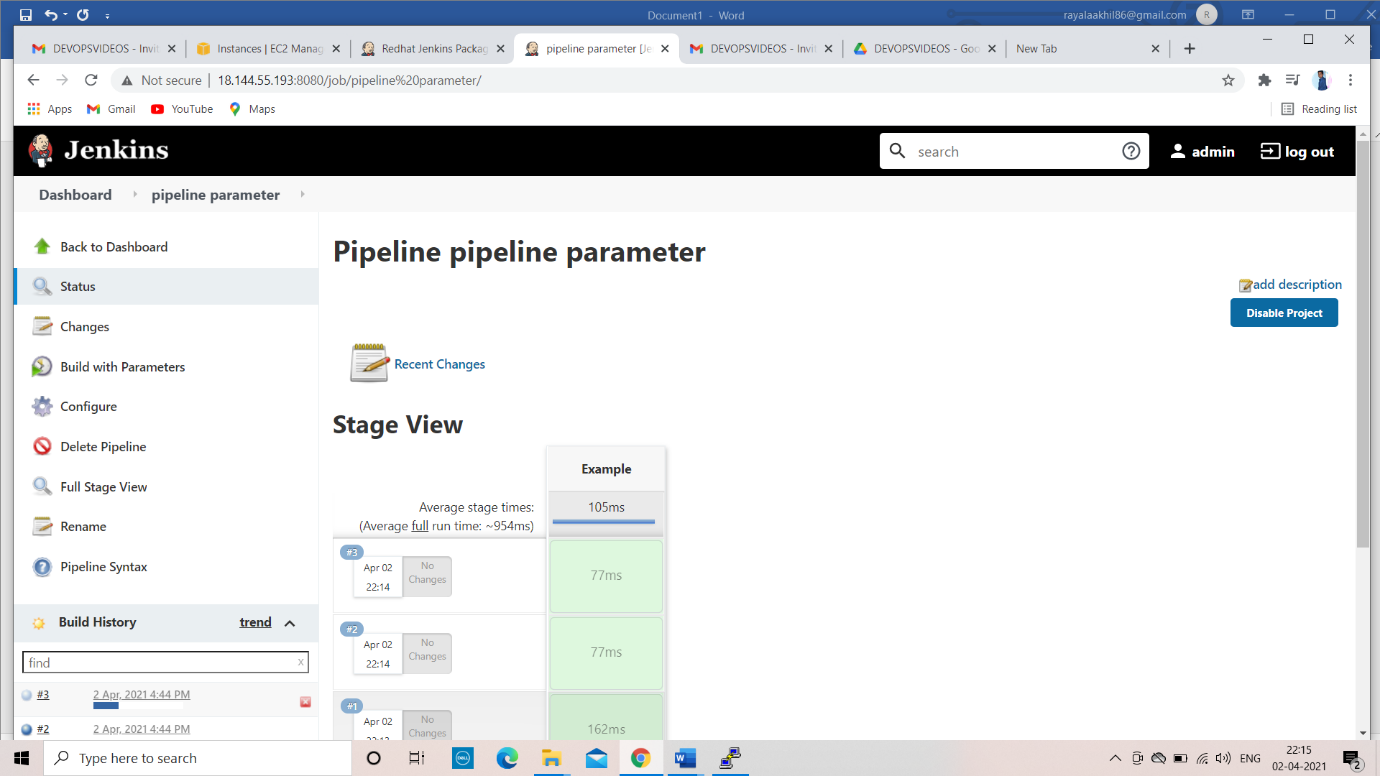
parameter have string, default value, description



First time you will see only build now option



Second time you will see the build with parameter



Deploy and QA

We cannot deploy the production our own

So before deploy the production

We given the input “does the staging environment look ok”

Then go to the next step

In the below we have the two steps 1)deploy QA 2)deploy production

By using pipleline we can build both QA and Production

