## Definition:

## CI and CD stand for continuous integration and continuous delivery/continuous deployment. In very simple terms, CI is a modern software development practice in which incremental code changes are made frequently and reliably. Automated build-and-test steps triggered by CI ensure that code changes being merged into the repository are reliable. The code is then delivered quickly and seamlessly as a part of the CD process. In the software world, the CI/CD pipeline refers to the automation that enables incremental code changes from developers’ desktops to be delivered quickly and reliably to production.

**Integration:** Combining all modules of code with your code.

**Continuous Integration:** Frequently integrate all other developers development code to your code and it installed in application server & test it in server.

\* Whenever you do integration late , then complexity is high in troubleshooting the integration issues.

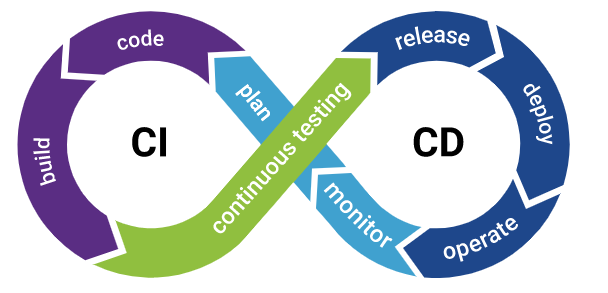
\* The more the delay in integration of your code to others code, the more time is spent on troubleshooting integration issues.

**Why is CI/CD important?**

CI/CD allows organizations to ship software quickly and efficiently. CI/CD facilitates an effective process for getting products to market faster than ever before, continuously delivering code into production, and ensuring an ongoing flow of new features and bug fixes via the most efficient delivery method.

## **Difference between CI and CD**

CI is a set of practices performed *as developers are writing* code, and CD is a set of practices performed *after* the code is completed.



## **What are the benefits of CI/CD?**

* Automated testing enables continuous delivery, which ensures software quality and security and increases the profitability of code in production.
* CI/CD pipelines enable a much shorter time to market for new product features, creating happier customers and lowering strain on development.
* The great increase in overall speed of delivery enabled by CI/CD pipelines improves an organization’s competitive edge.
* Automation frees team members to focus on what they do best, yielding the best end products.
* Organizations with a successful CI/CD pipeline can attract great talent. By moving away from traditional [waterfall methods](https://en.wikipedia.org/wiki/Waterfall_model), engineers and developers are no longer bogged down with repetitive activities that are often highly dependent on the completion of other tasks.

====Jenkins installation in Centos7=======================

yum install java-1.8.0-openjdk-devel

java

javac

yum install wget -y

wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import http://pkg.jenkins.io/redhat-stable/jenkins.io.key

yum install jenkins

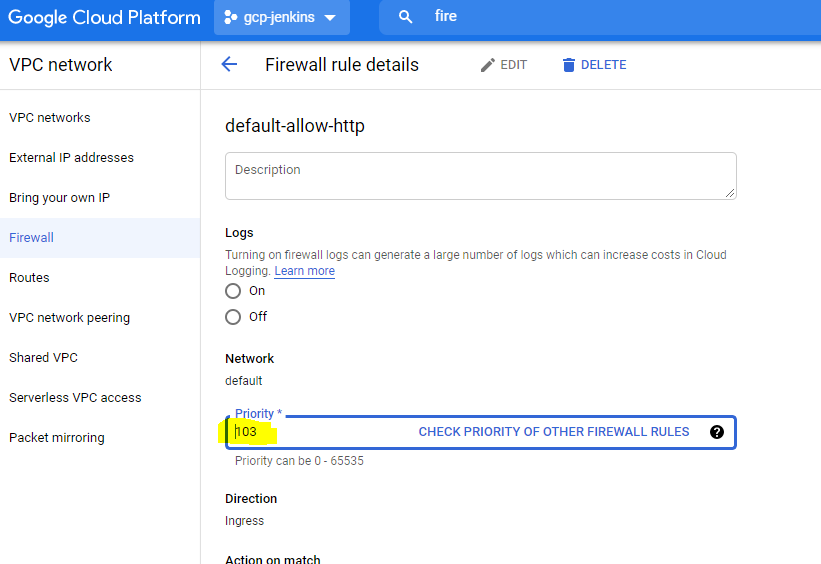
sudo systemctl start jenkins

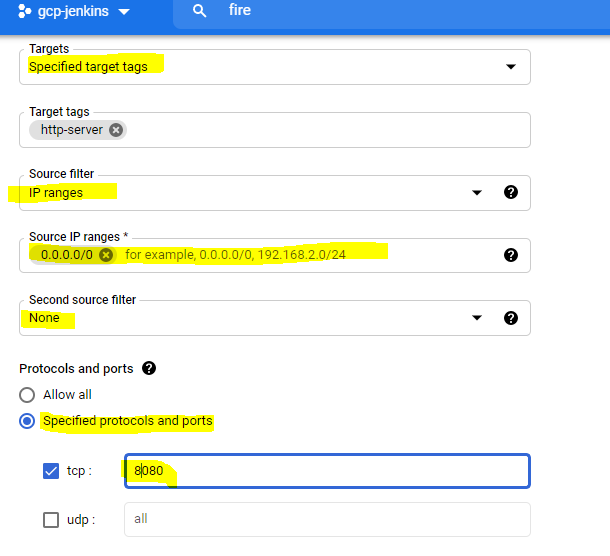
sudo systemctl enable jenkins

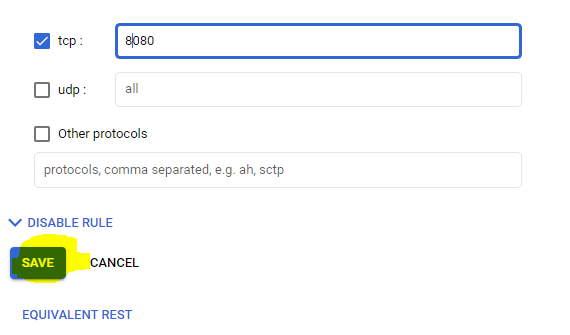
sudo systemctl status jenkins

ps -ef | grep jenkins

=**Enable port 8080==**===========







What was CI generally do?

A). It frequently builds your code that is there in your version control.

After your build, It's frequently deployed on the server.

After successful deployment, we need to test our application by running some automated test cases.

**What are the diff types of CI tools?**

Hudson

Jenkins

Bamboo

Cruise Control

Anthil Pro

IBM Buildforge.

Azure Devops

What is your role in Jenkins?

A). Administration.

.i.e, Installation, Configuration, maintenance and set up jenkins jobs.

**How to install Jenkins in tomcat server?**

A). Jenkins is a java web application, so jenkins has WAR file(jenkins.war).

Pre requirement:

you need JRE installed in your system.

You need to install a tomcat application server.

Download jenkins.war file from below URL.

<https://updates.jenkins-ci.org/download/war/>

Download tomcat application server

https://tomcat.apache.org/download-70.cgi

Extract tomcat application server and copy jenkins.war war to tomcat/webapps location.

Now start the tomcat application server.

As jenkins is a web application, you need to access jenkins using browser.

how to verify your tomcat is started or not?

A) ps -ef | grep java

or

ps -ef | grep tomcat

if you get any result, then tomcat is started.

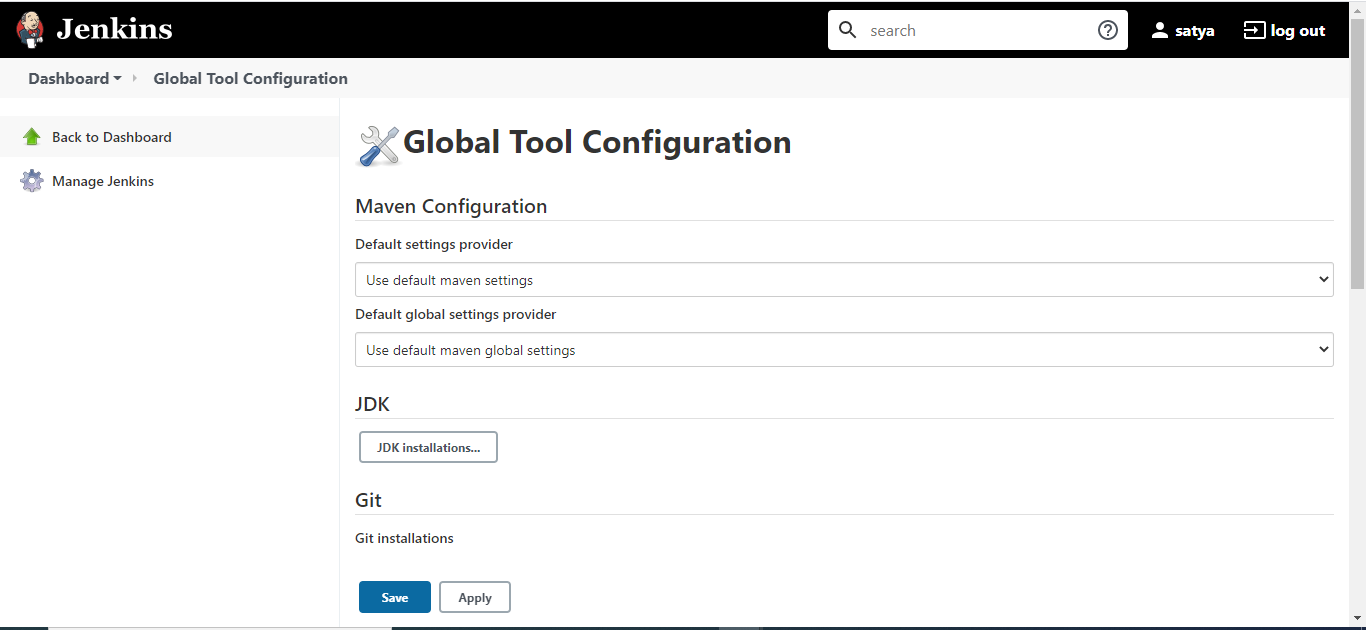
Default tomcat application server port number is 8080

**Global Tool Configuration:**

Only the Administrator have manage Jenkins option in jenkins home page.

Only Administrator can do the Global Tool Configuration in jenkins.

Manage jenkins --> Global Tool Configuration -->



JAVA Configuration

In server check where we have java

FYI  
We installed **java-1.8.0-openjdk-devel (**yum install java-1.8.0-openjdk-devel)

[root@linux ~]# which javac

/bin/javac

[root@linux ~]# ls -ln /bin/javac

lrwxrwxrwx. 1 0 0 23 Jul 10 03:29 /bin/javac -> /etc/alternatives/javac

[root@linux ~]# ls -ln /etc/alternatives/javac

lrwxrwxrwx. 1 0 0 70 Jul 10 03:29 /etc/alternatives/javac -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.292.b10-1.el7\_9.

x86\_64/bin/javac

[root@linux ~]# cd /usr/lib/jvm/java-1.8.0-openjdk

[root@linux java-1.8.0-openjdk]# pwd

/usr/lib/jvm/java-1.8.0-openjdk

[root@linux java-1.8.0-openjdk]# ll

total 180

-rw-r--r--. 1 root root 1522 Apr 21 14:28 ASSEMBLY\_EXCEPTION

drwxr-xr-x. 2 root root 4096 Jul 10 03:29 bin

drwxr-xr-x. 3 root root 132 Jul 10 03:29 include

drwxr-xr-x. 4 root root 95 Jul 10 03:29 jre

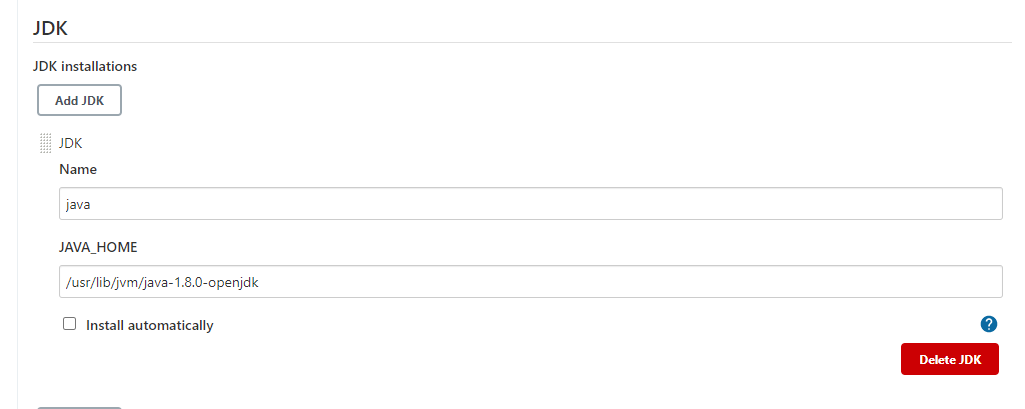
drwxr-xr-x. 3 root root 144 Jul 10 03:29 lib

-rw-r--r--. 1 root root 19274 Apr 21 14:28 LICENSE

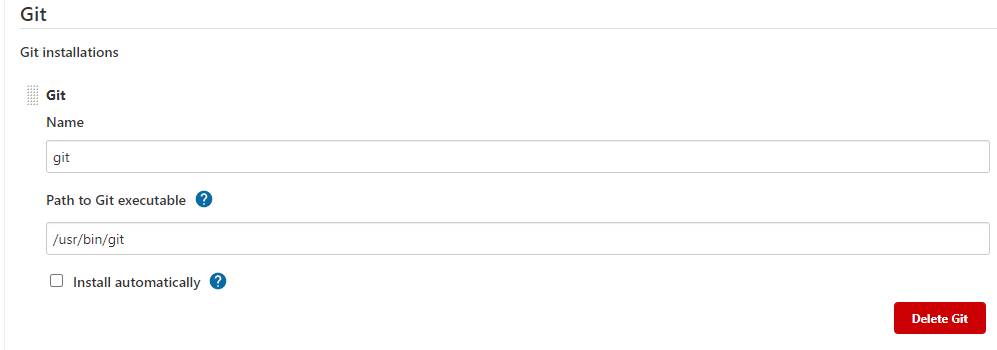
drwxr-xr-x. 2 root root 204 Jul 10 03:29 tapset

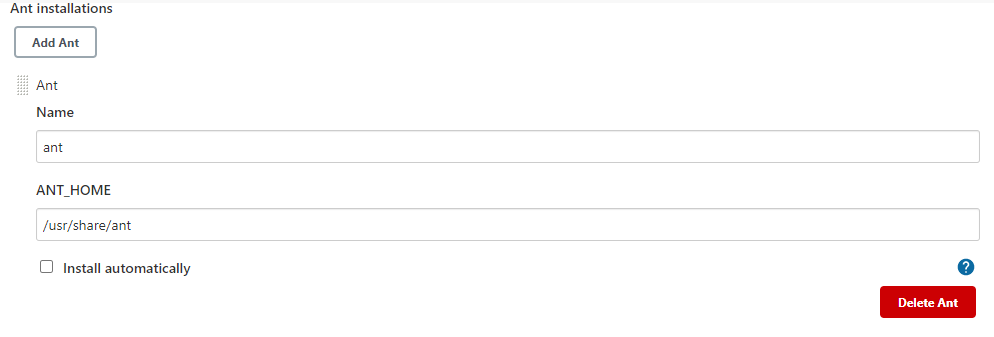
-rw-r--r--. 1 root root 155003 Apr 21 14:28 THIRD\_PARTY\_README

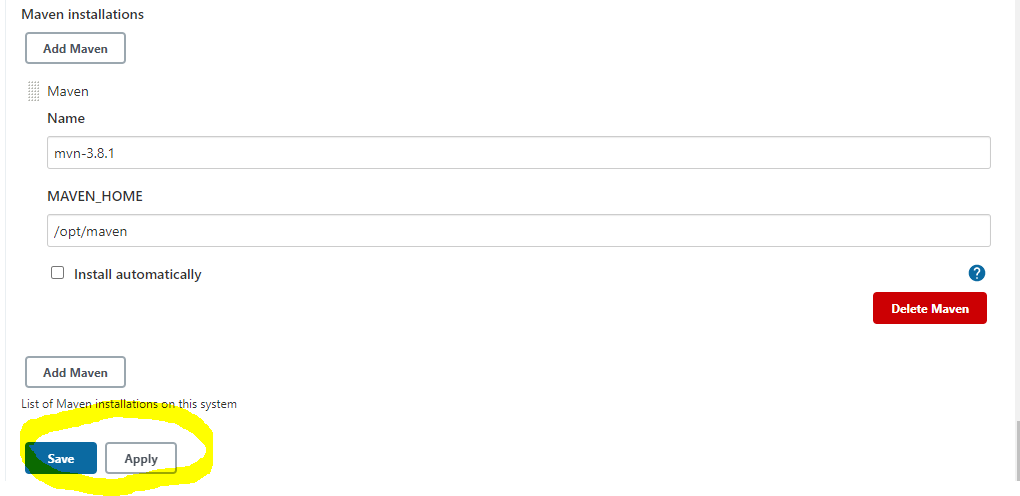
[root@linux java-1.8.0-openjdk]#



Like that add GIT /MAVEN/ANT







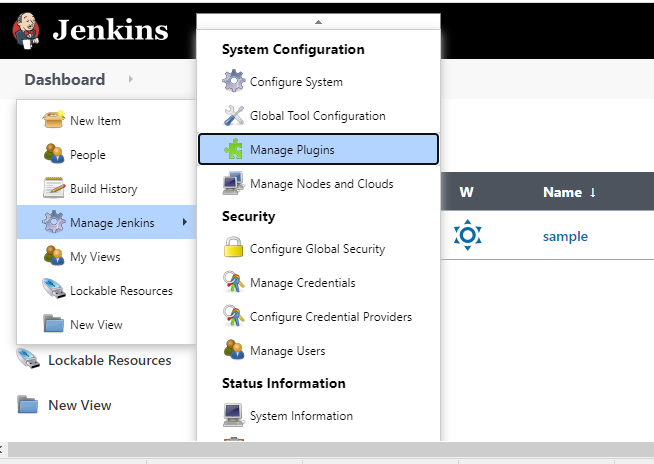
Then save it.

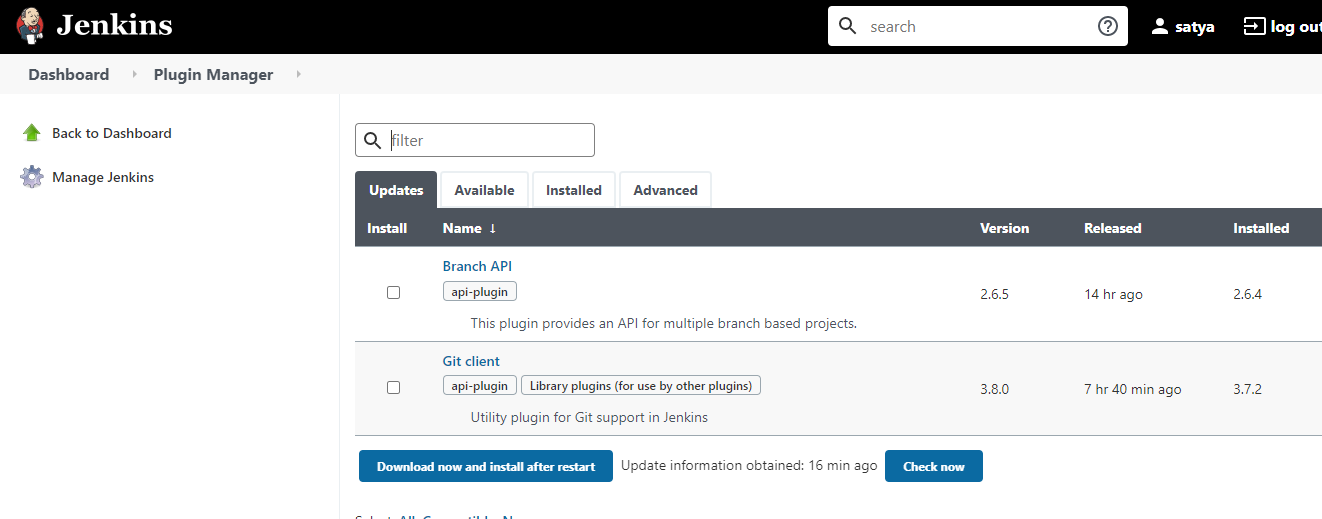
What are plug-ins in jenkins?

Plug-ins are just like addons, they are meant to extend the functionality of your jenkins.

.i.e, Jenkins comes with some default set up and if you want to extend the functionality or add new features then you need to install corresponding plugins.

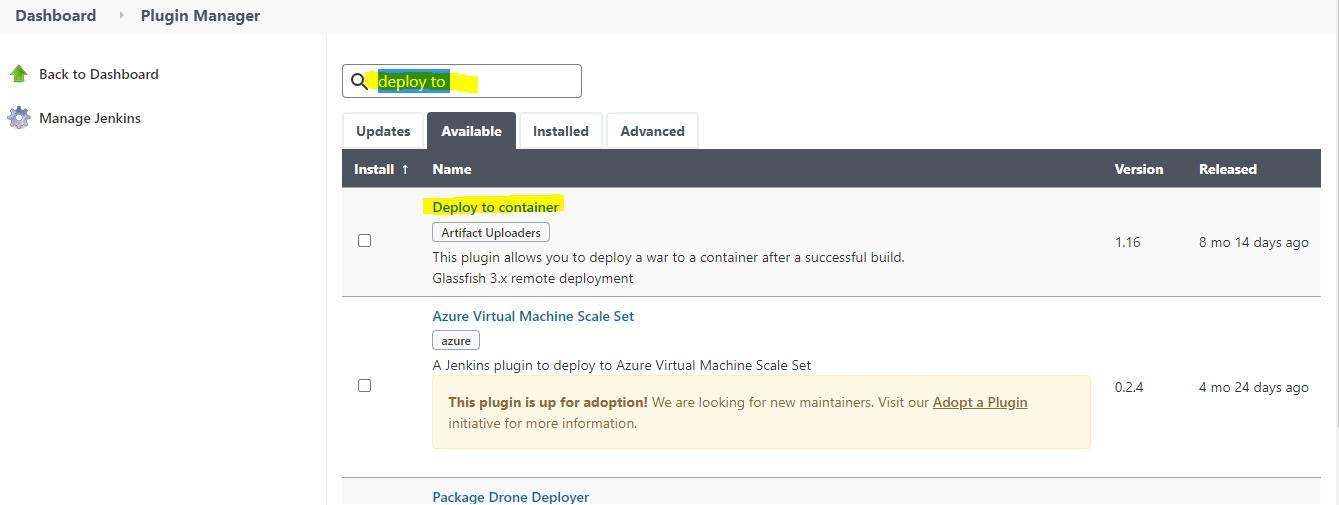
**How to install plug-ins in jenkins?**

--> only the Administrator can install plug-ins in jenkins.

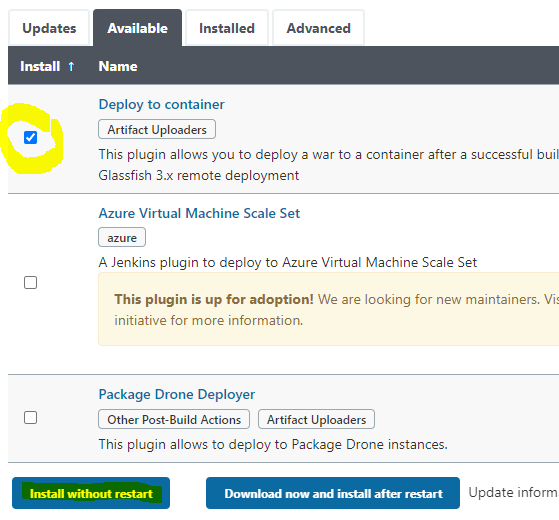


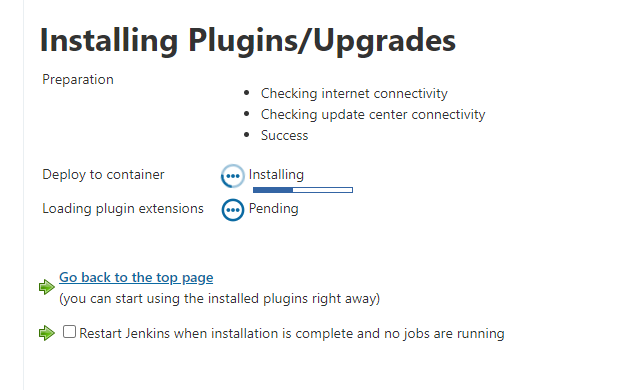
Click on Available

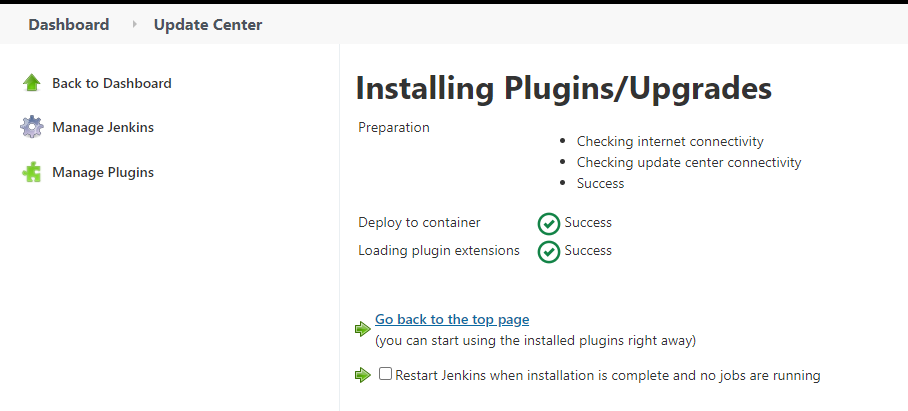
Search for require plug-in



Select that and click on install without restart







What is a job in jenkins?

If you want Jenkins to do any activity/task ,we need to create a job in jenkins.

In that job configuration, you need to define How to do & what to do?

What does a job contain?

Mainly a job contains 4 categories.

--> **Source Code Management**

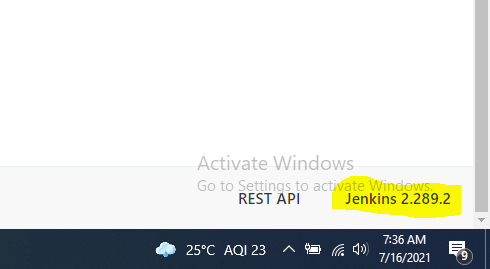
--> **Build Triggers**

--> **Build**

--> **Post-build Actions**

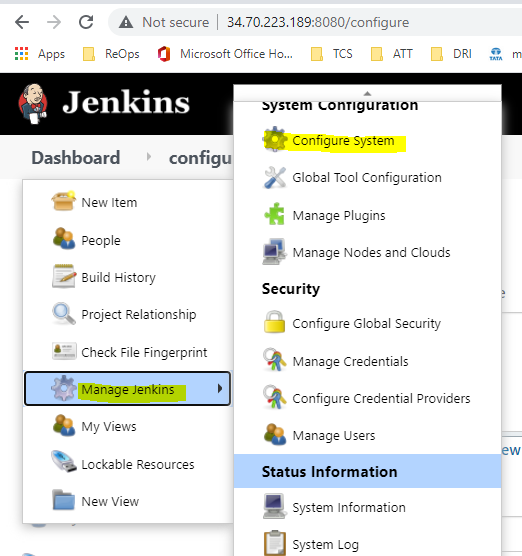
**Where you find the jenkins installed version in jenkins home page:**

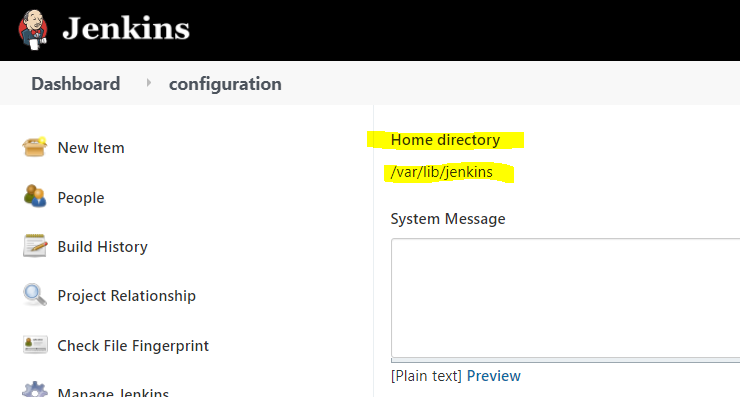
**At the right side bottom of the home page we can see the jenkins installed version.**

****

**Where do you find the jenkins Home directory?**

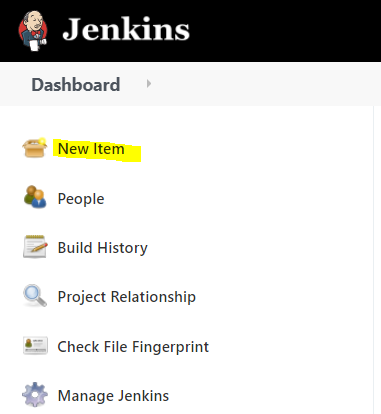
**Manage jenkins → Configure system**

****

****

**JOB:**

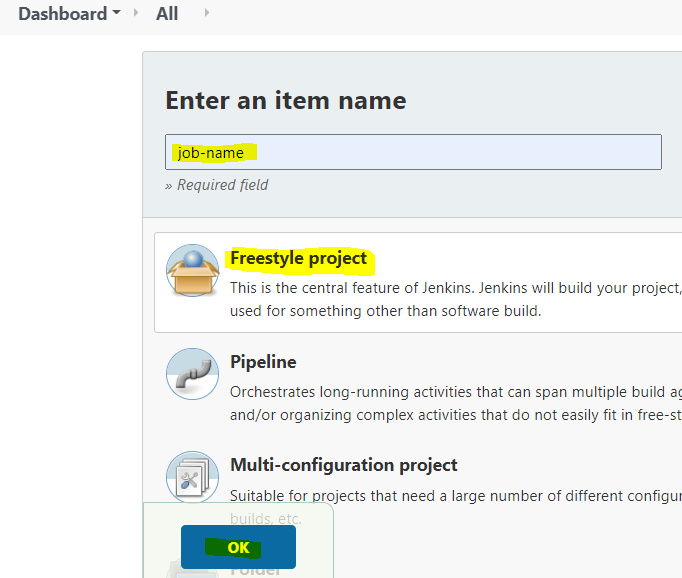
**Click on new item**

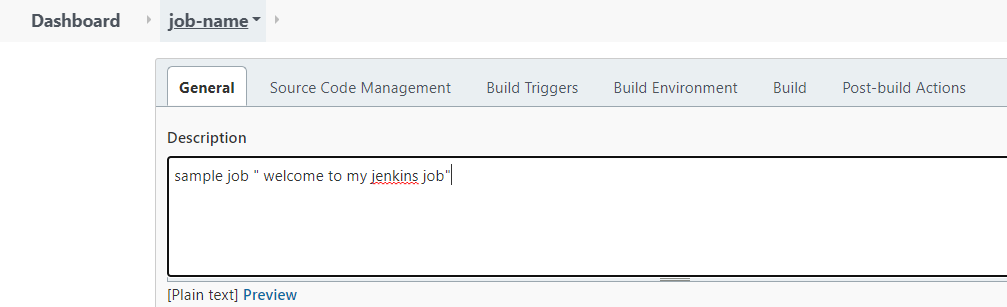
****

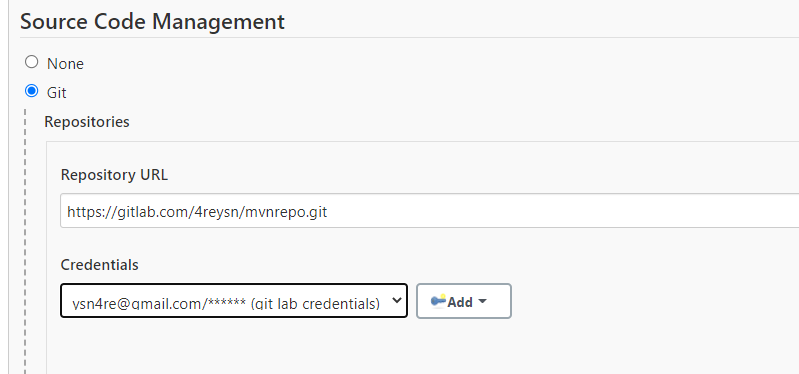
**Enter job name**

**Select Freestyle project**

**Click on OK**

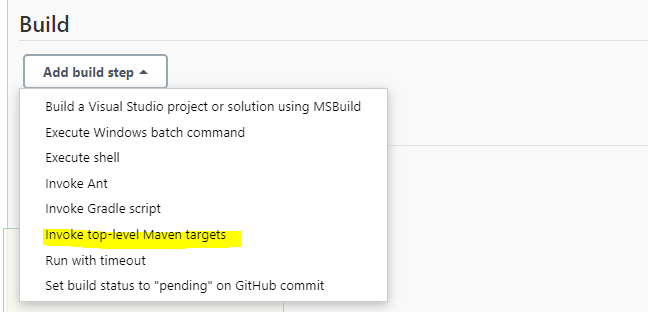
****

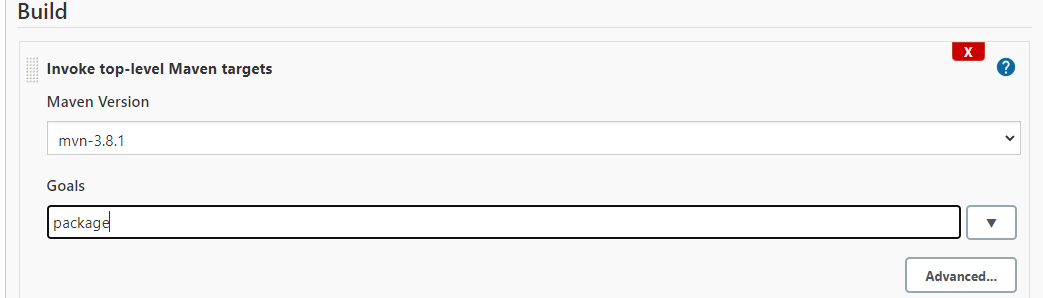
****

****

**In build phase select invoke top-level Maven targets for maven project**

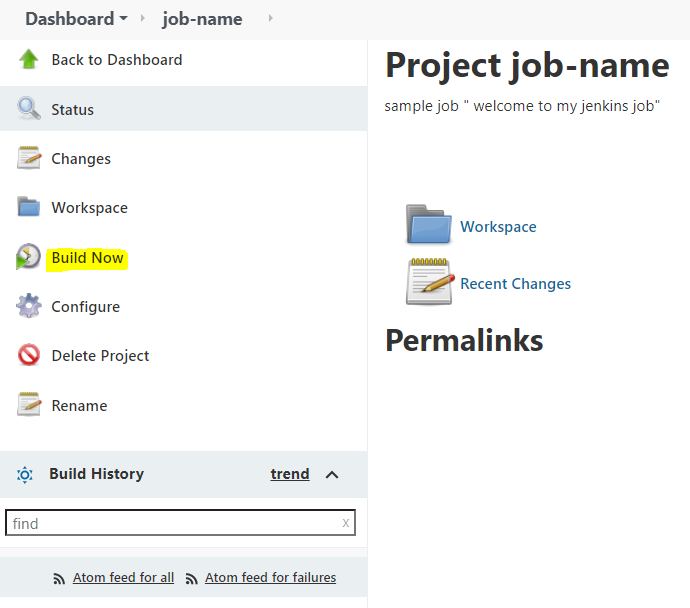
**If build file is build.xml select Invoke Ant**

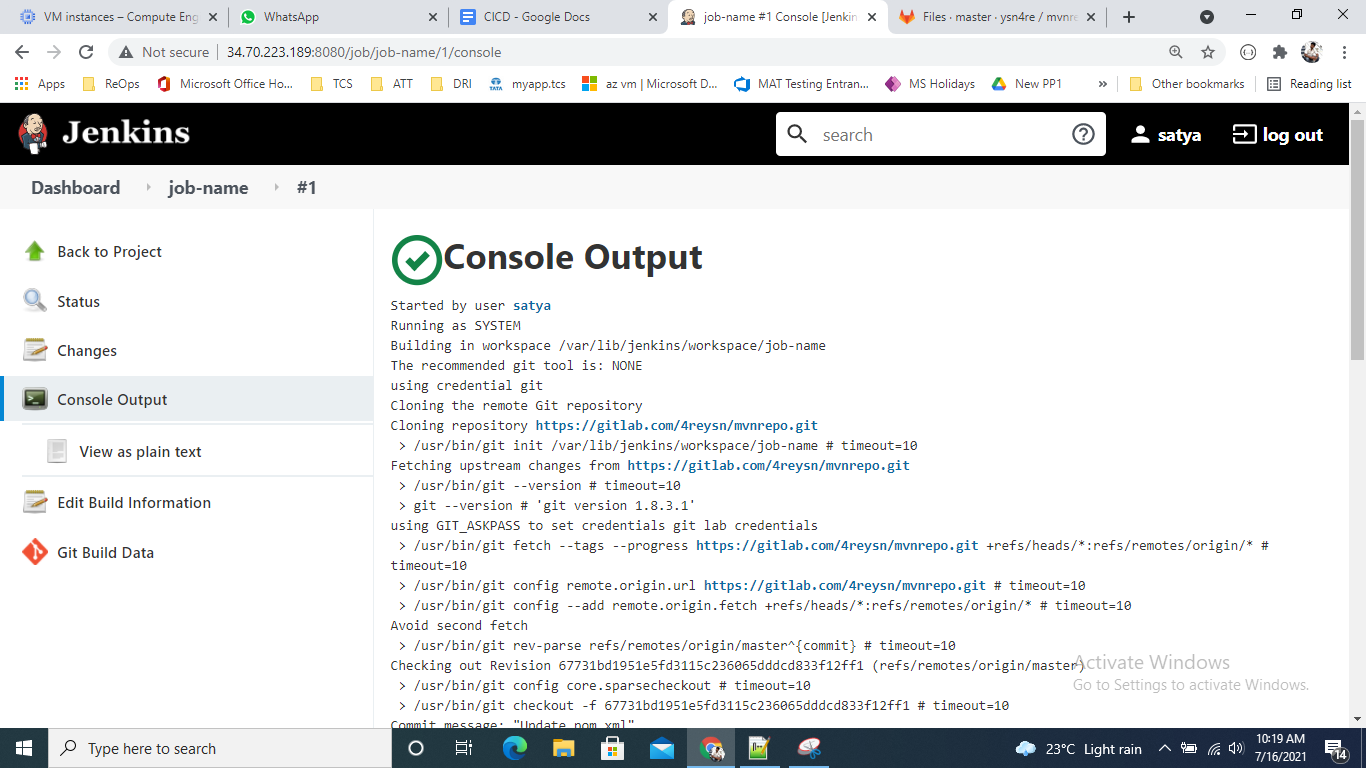
****

****

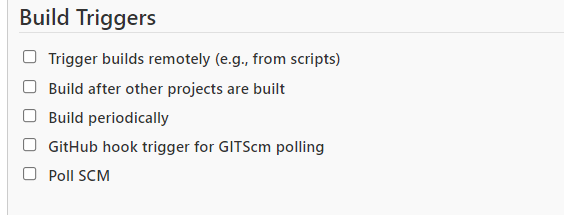
**Click on save.**

**Then build the job**

****

****

**Build triggers Category:**

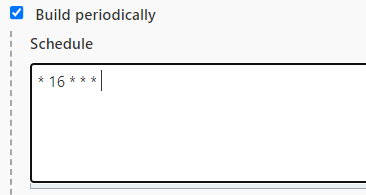
****

**If you want to run a job at a particular time**

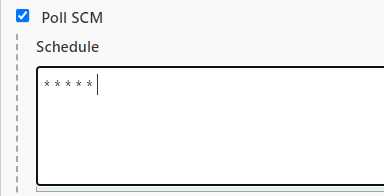
**Select Build periodically then give the at what time you want to run the job.**

**For example if you want to run the at 4pm daily**

**\* 16 \* \* \***

****

**For every code check-in into code repo, if you want to trigger a job use POLL SCM option**

****

TYPES of Jenkins jObs:

**Freestyle JOb:**

**Freestyle means improvised or unrestricted. A freestyle project in Jenkins is a project that spans multiple operations. It can be a build, a script run, or even a pipeline. According to the official Jenkins wiki, a freestyle project is a typical build job or task.**

**Pipeline JOb:**

**In Jenkins, a pipeline is a collection of events or jobs which are interlinked with one another in a sequence. It is a combination of plugins that support the integration and implementation of continuous delivery pipelines using Jenkins.**

**Multiconfiguration job:**

**The “multiconfiguration project” (also referred to as a “matrix project”) lets you run the same build job in many different configurations. This powerful feature can be useful for testing an application in many different environments, with different databases, or even on different build machines.**

**Jenkins provides multi-configuration project. With this option we can create only one job with many configurations. Each configuration will be executed as a separate job.**

**for example: if you want to run job with JDK8 which is installed in one node**

**and same job if you want to run on JDK7 which is installed in another node.**

**Folder job:**

**What are folders? Jenkins provides the ability to organize jobs into a hierarchical manner with the CloudBees Folders Plugin. This allows us to manage the jobs much like we would files on a file system. Folders can also be used to manage permissions on a per folder basis to ease security administration**

**What is GitHub organization in Jenkins?**

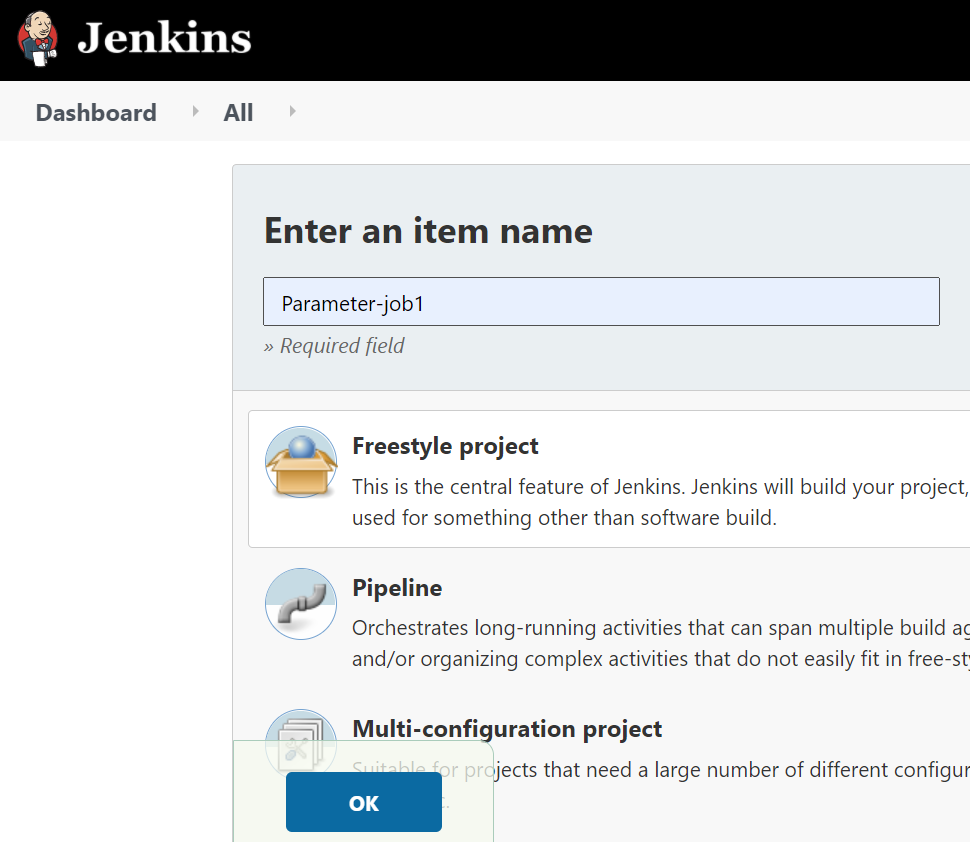
**Creating a GitHub Organization project**

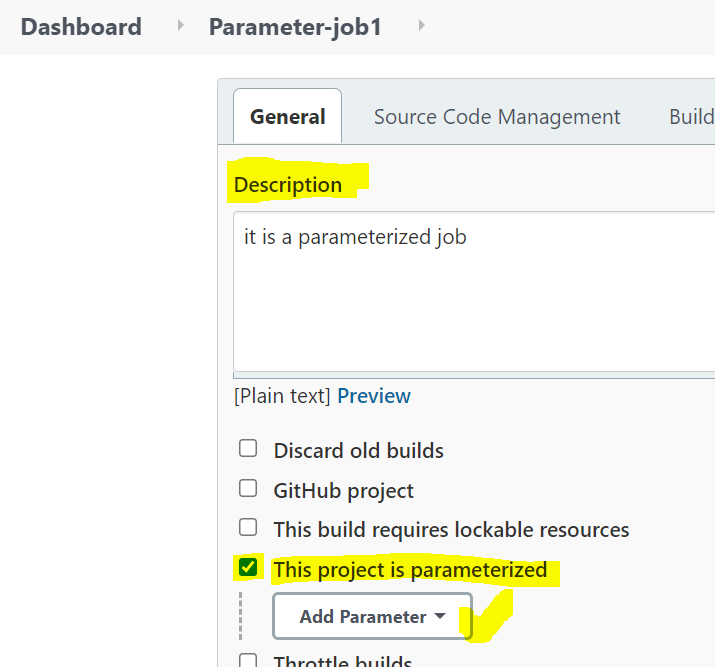
**The GitHub organization project creates a Multibranch Pipeline for each repository in an organization. This Multibranch Pipeline then creates a separate project for each branch that contains a Jenkinsfile in the top level directory.**

**What is a Multibranch pipeline in Jenkins?**

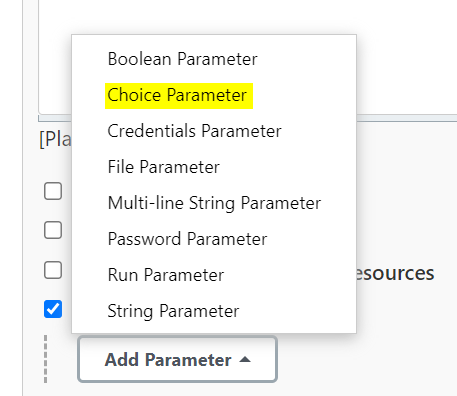
**A multi-branch pipeline is a concept of automatically creating Jenkins pipelines based on Git branches. It can automatically discover new branches in the source control (Github) and automatically create a pipeline for that branch.**

**PARAMETERIZED JOBS:**

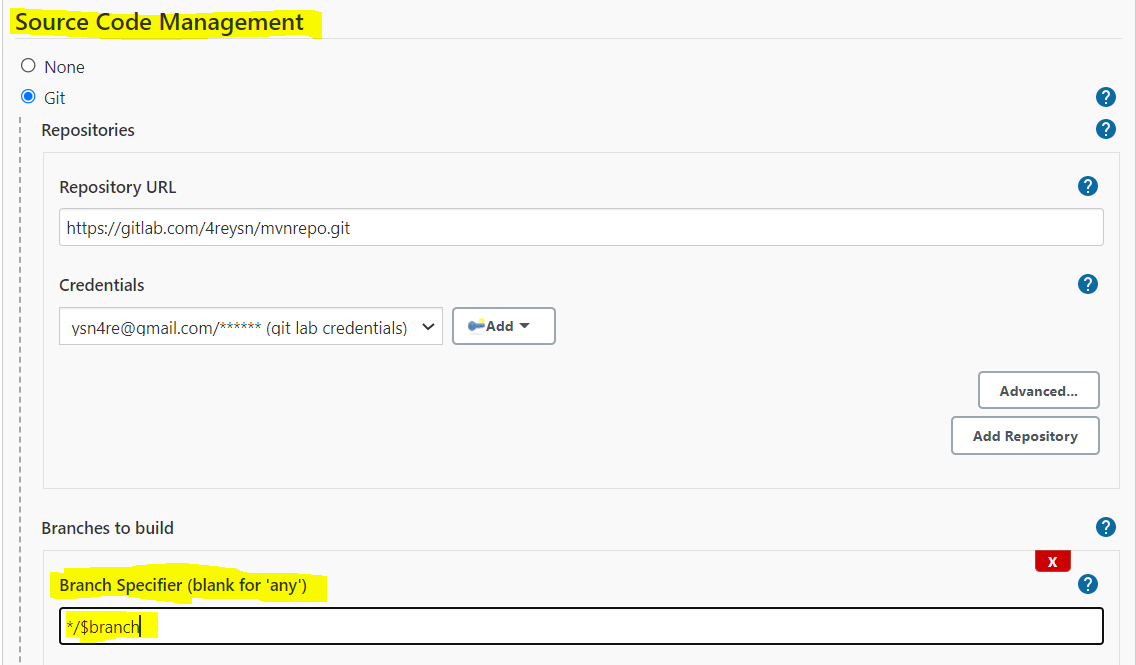
****

****

**Click on Add parameter and select require parameter .**

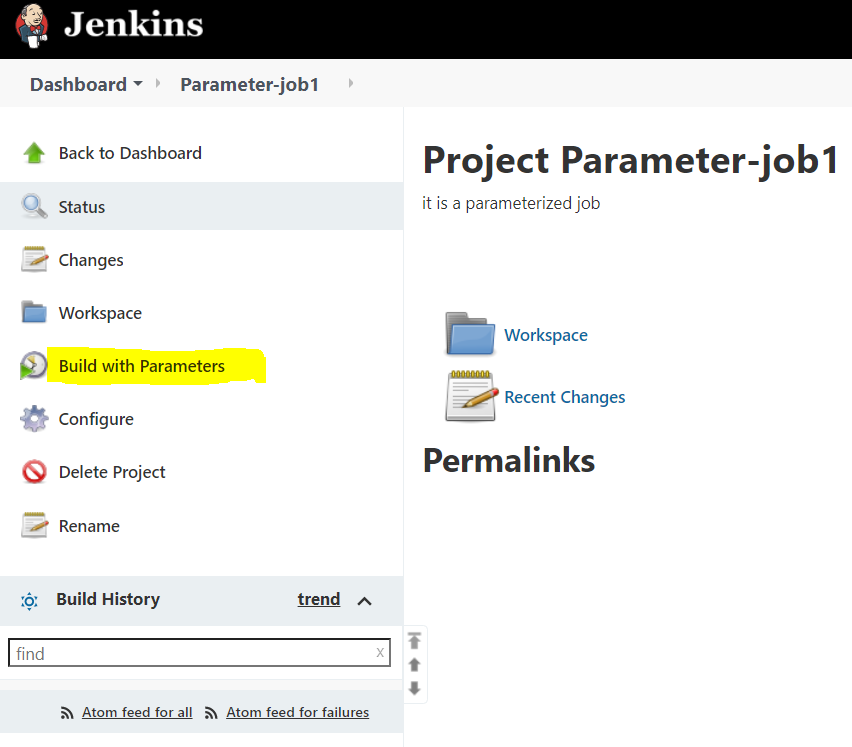
****

****

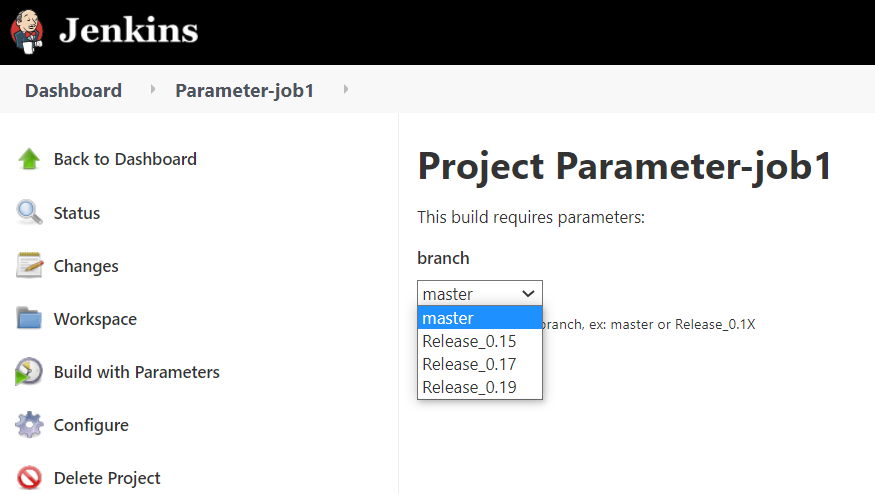
****

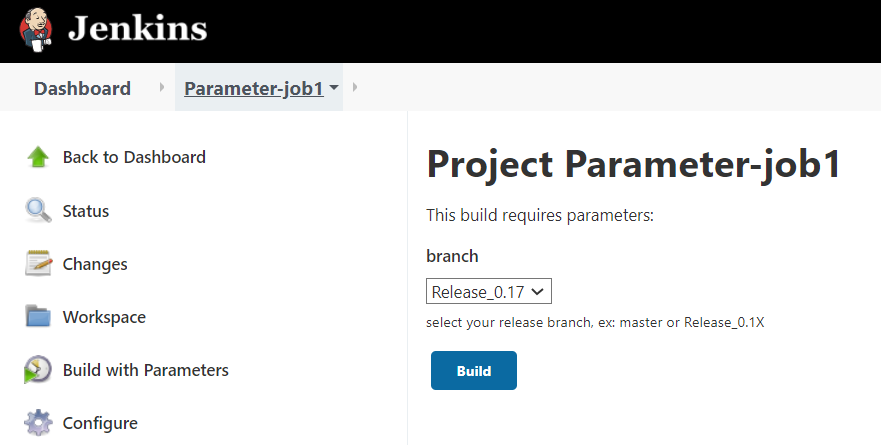
****

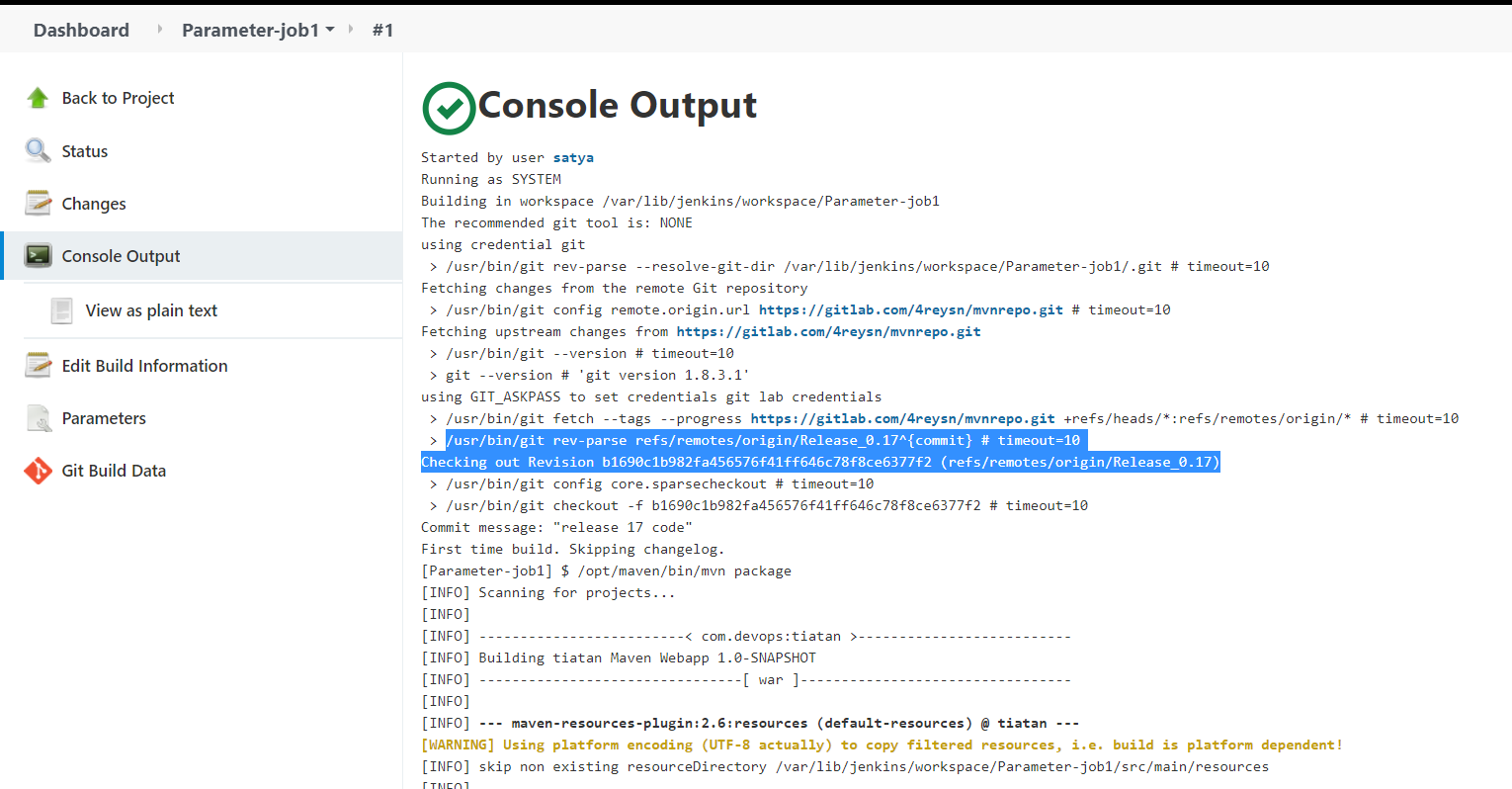
**Save the job with the above configuration and build it.**

****

**Click on Build with parameters and select require branch and click on Build.**

****

****

****

**TOMCAT App server installation:**

**Tomcat installation:**

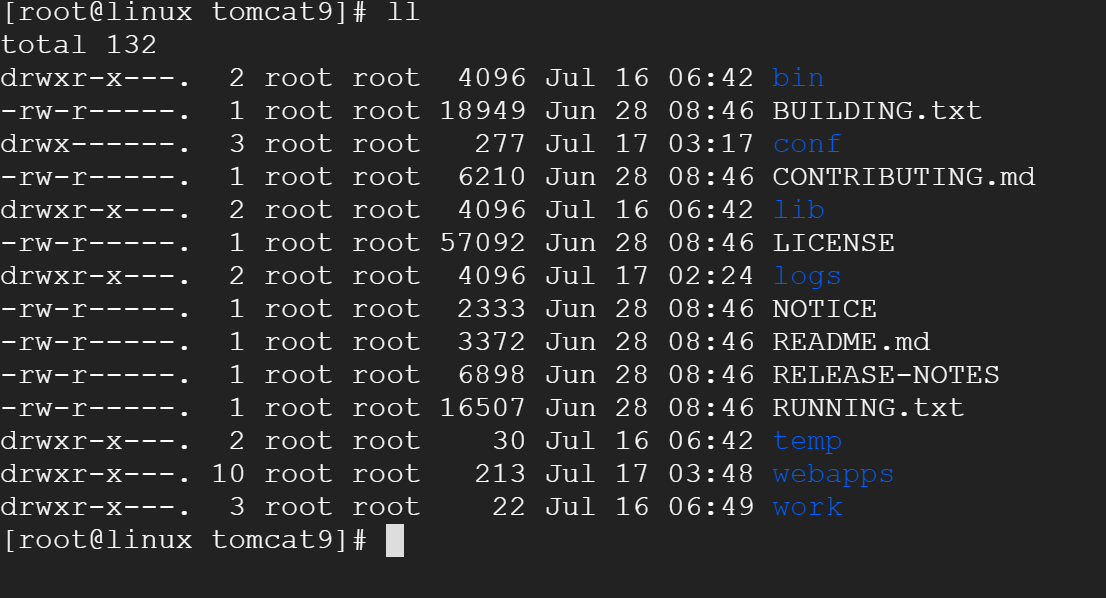
**#cd /opt**

**#wget https://mirrors.estointernet.in/apache/tomcat/tomcat-9/v9.0.50/bin/apache-tomcat-9.0.50.tar.gz**

**#tar -xvf apache-tomcat-9.0.50.tar.gz**

**#mv apache-tomcat-9.0.50 tomcat-9**

**#cd tomcat-9**

****

**Need to do below modifications:**

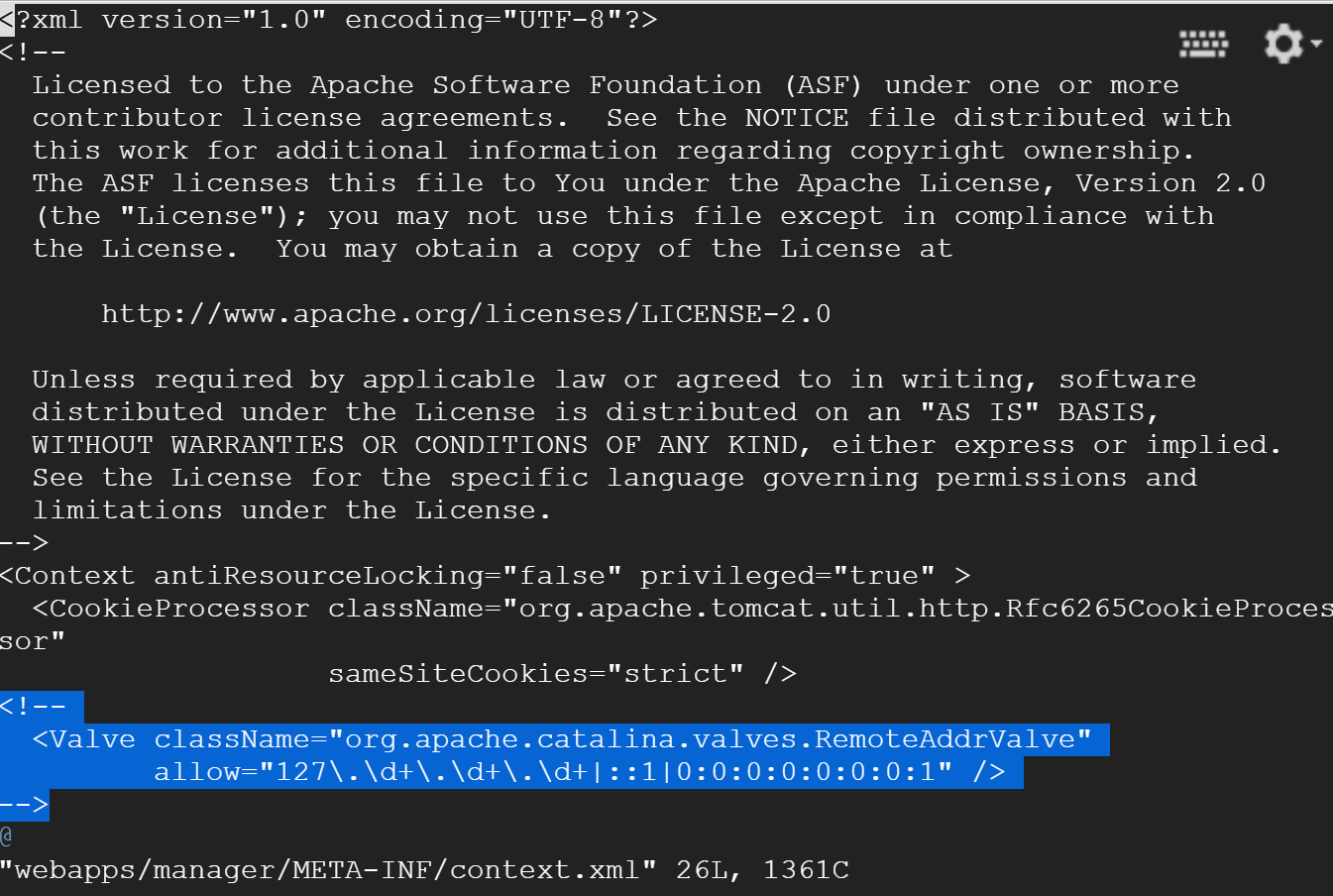
**tomcat9/webapps/manager/META-INF/context.xml**

**<!--**

**<Valve className="org.apache.catalina.valves.RemoteAddrValve"**

**allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />**

**→**

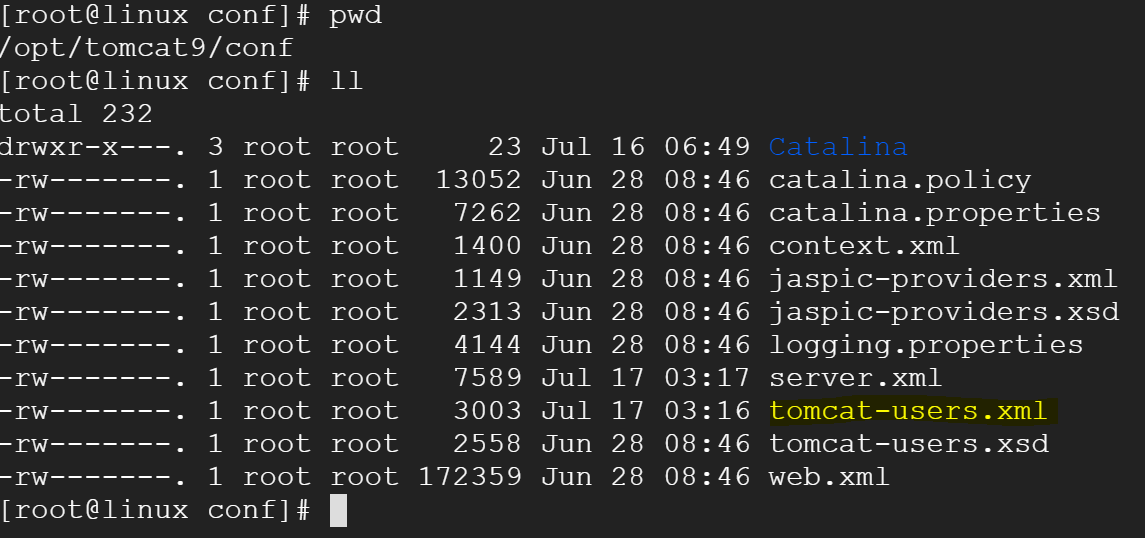
****

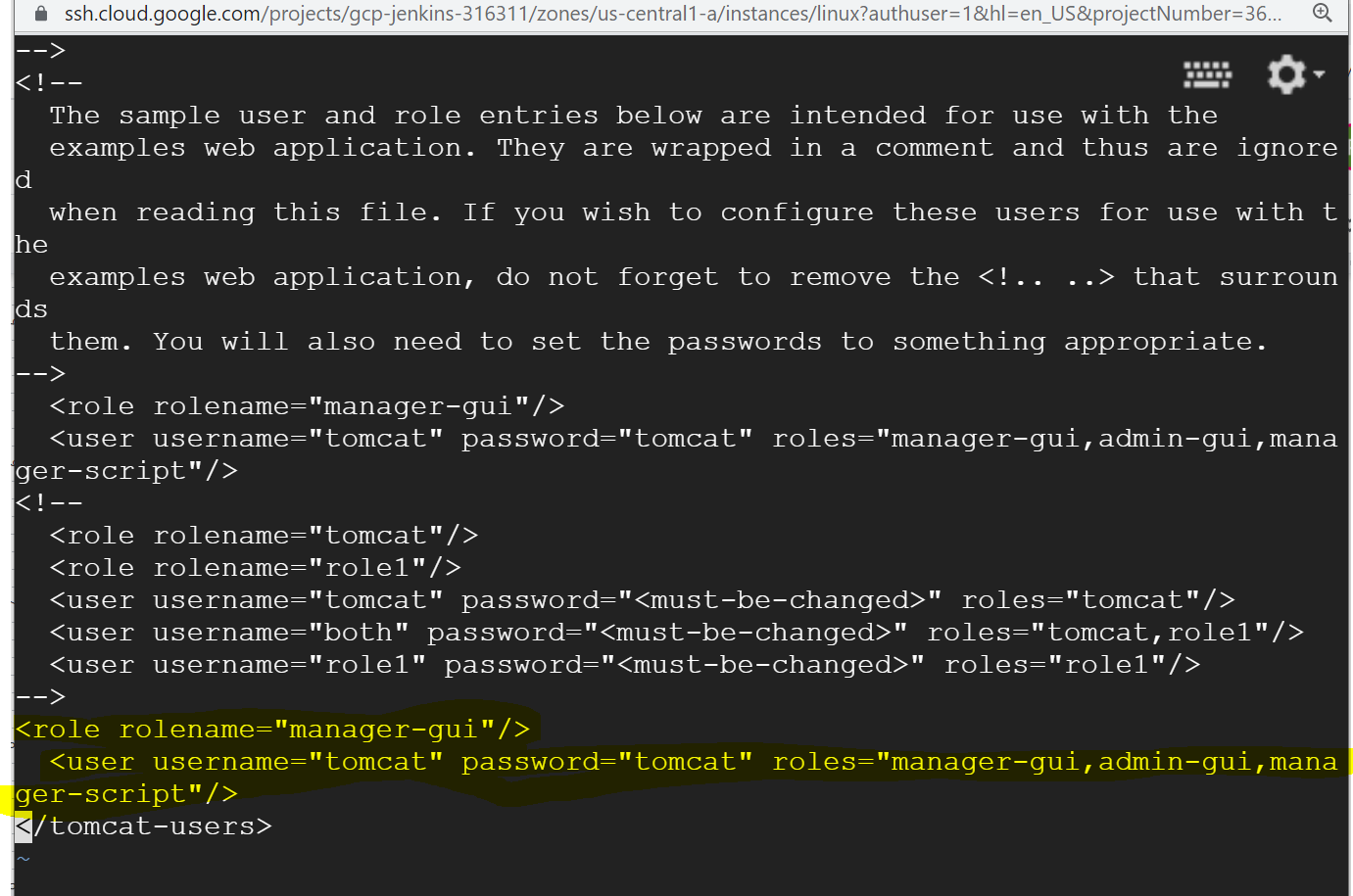
**Add user in tomcat-users.xml**

**tomcat9/conf/tomcat-users.xml**

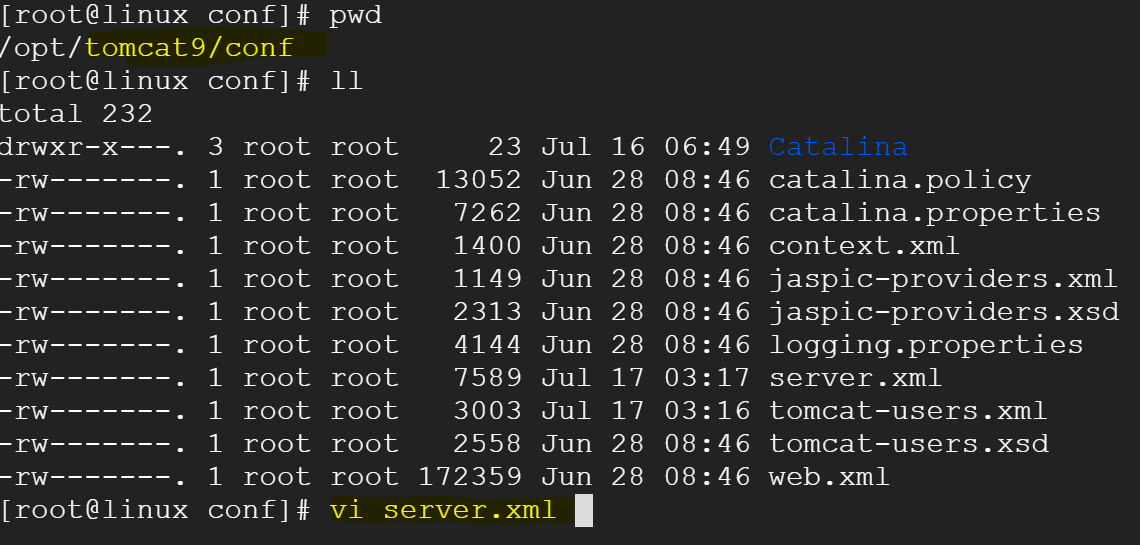
**<role rolename="manager-gui"/>**

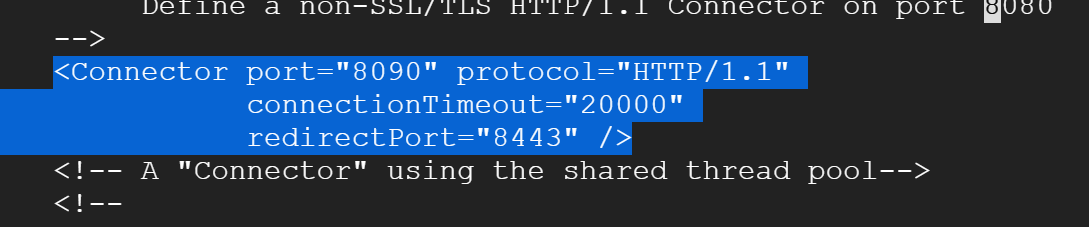
**<user username="tomcat" password="tomcat" roles="manager-gui,admin-gui,manager-script"/>**

****

****

**If you are installing TOMCAT APP server in same jenkins installed server, you have to update connector port for TOMCAT**

****

****

**NOw start the tomcat server**

**[root@linux conf]# cd /opt/tomcat9/bin/**

**[root@linux bin]# pwd**

**/opt/tomcat9/bin**

**[root@linux bin]# ./startup.sh**

**In parameterized job open configure**

**Go to** Post-build Actions

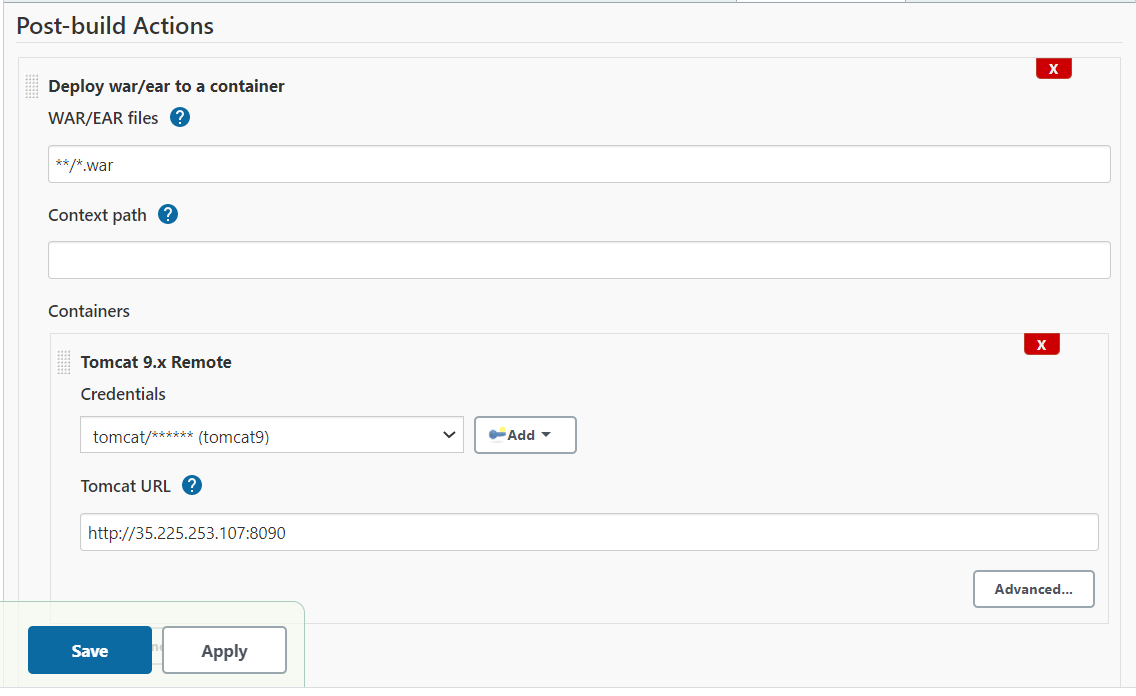
Select Deploy war/ear to container

-> WAR/EAR files -- \*\*/\*.war

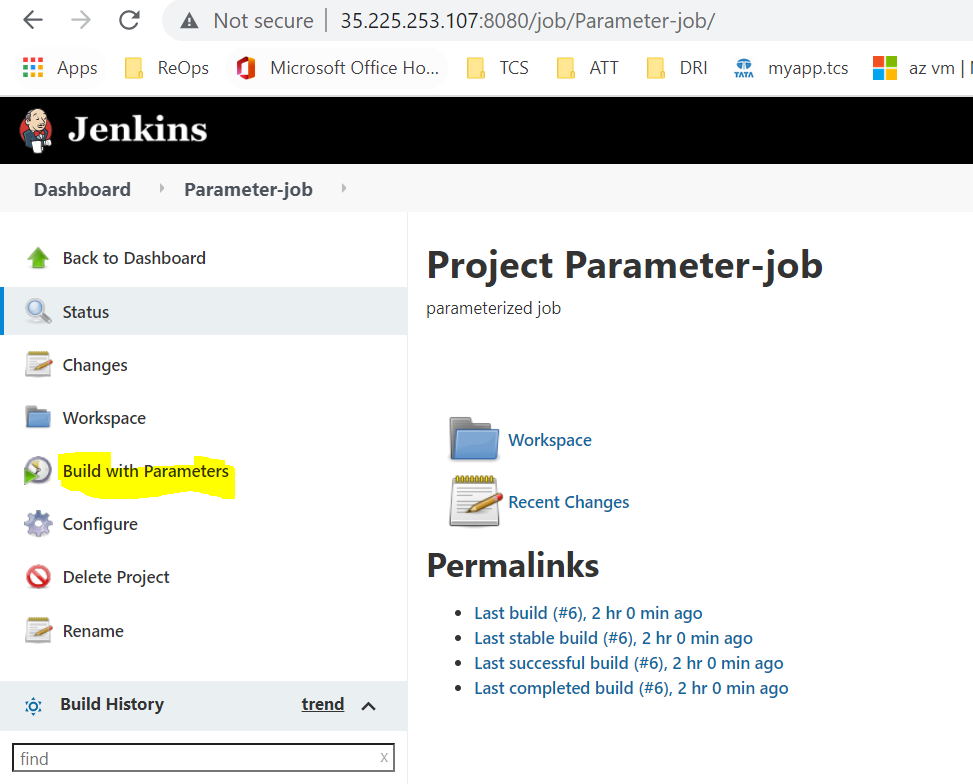
-->add tomcat credentials

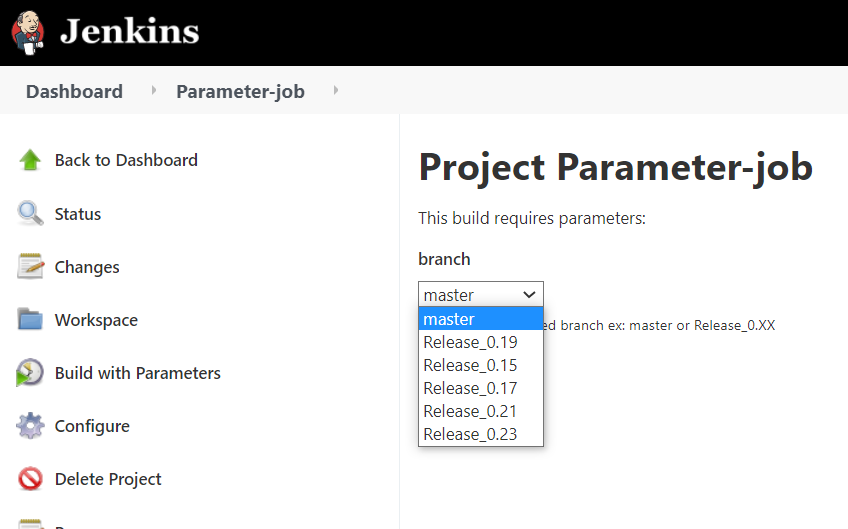
→ give tomcat URL

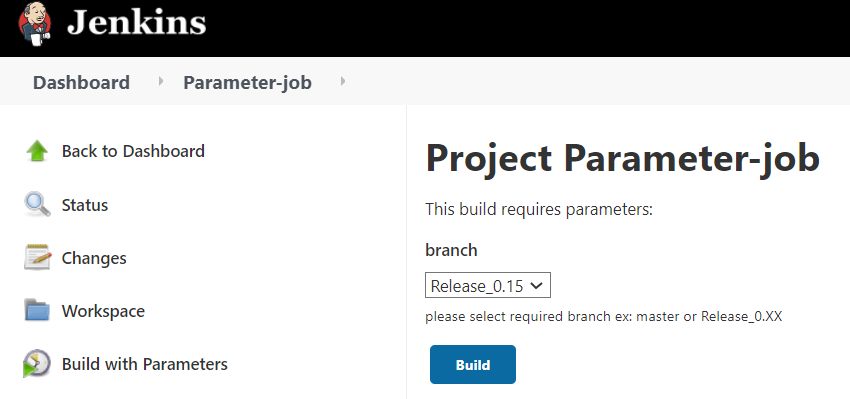
Save the job and run the parameterized job.

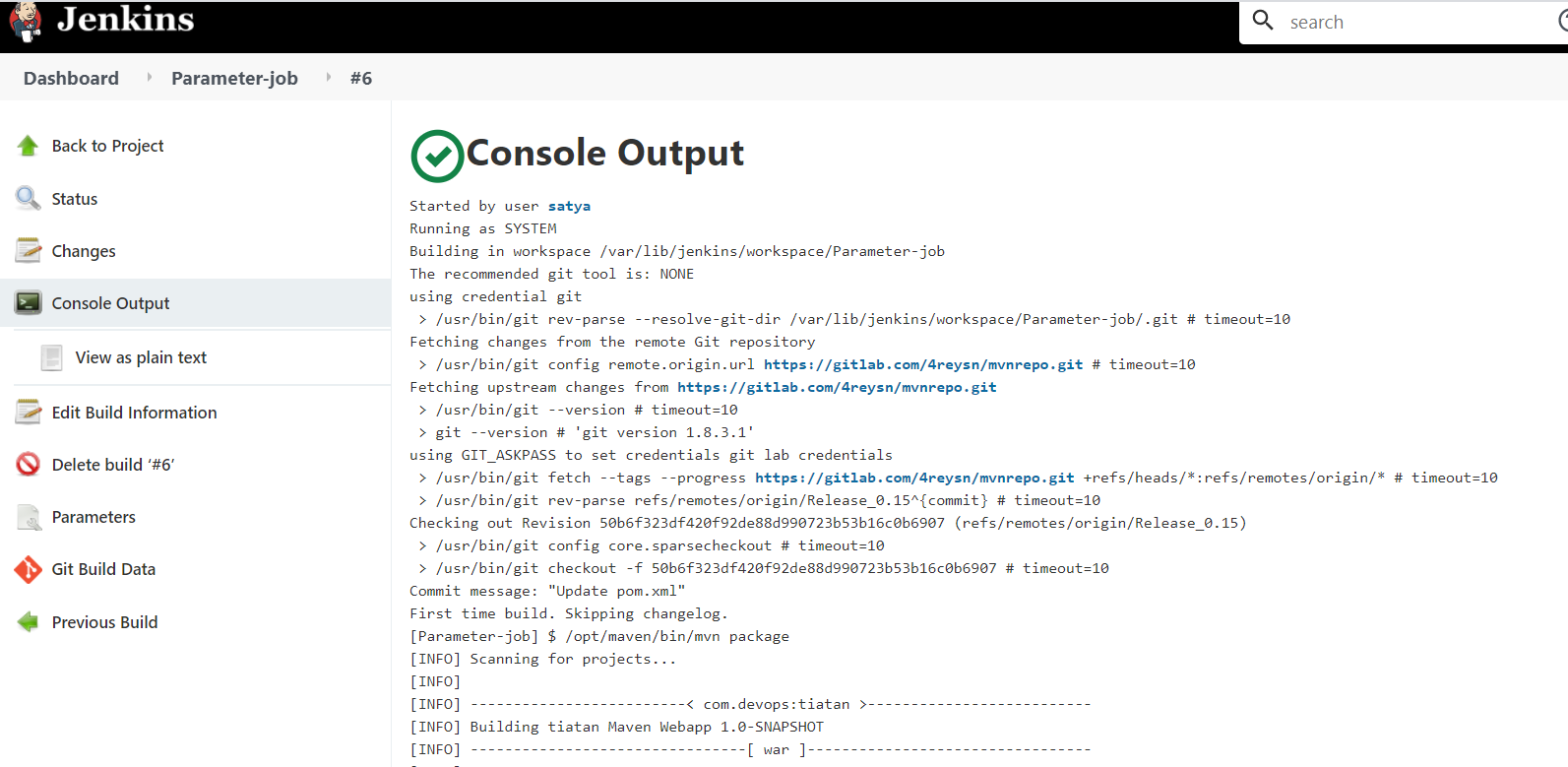


**Click on Build with Parameters**

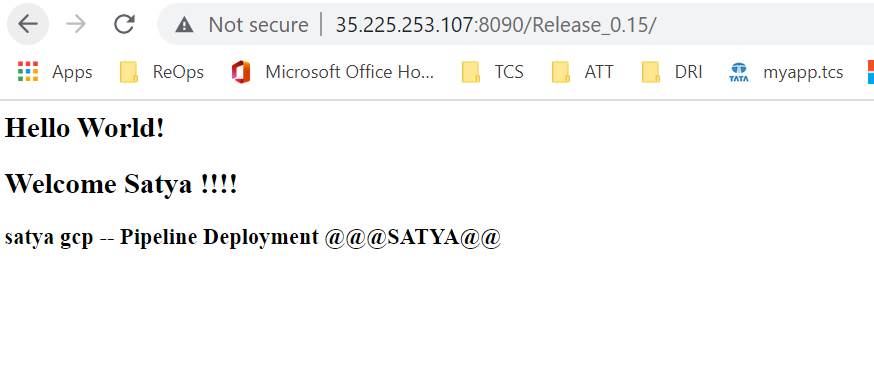
****

****

**Select the required branch and click on Build**

****

****

****

**MASTER --SLAVE(NODE):**

**Jenkins Master:**

The Jenkins master acts to schedule the jobs, assign slaves, and send builds to slaves to execute the jobs. It will also monitor the slave state (offline or online) and get back the build result responses from slaves and the display build results on the console output.

If you have only a few jobs to run on the same platform, having only a Jenkins master to build these jobs is OK.

But if you have hundreds jobs running on different platform (Linux, Windows, 32/64 bits), it's better to use Jenkins slave.

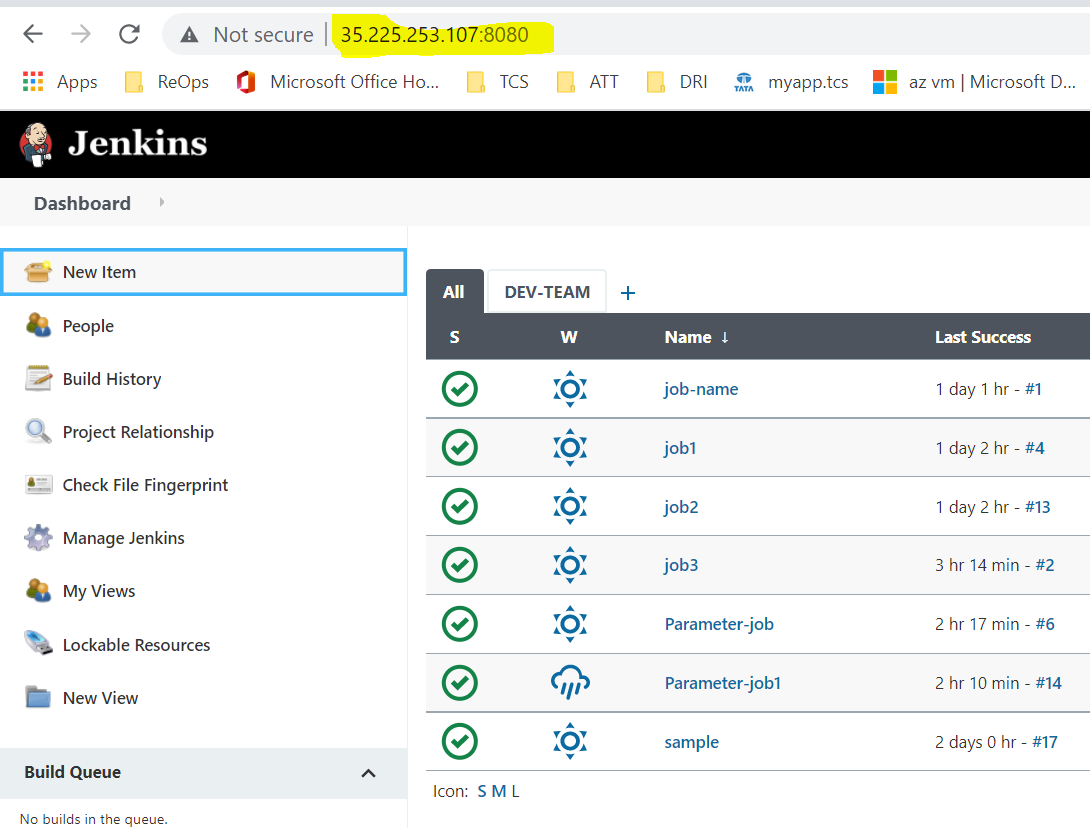
You can affect some labels per slave to sort them by Team,OS,...

Your Jenkins slaves can run on a VM or on a Docker image.

Jenkins Slave:

A slave is a Java executable that runs on a remote machine. The characteristics of the slave are : It hears requests from the Jenkins Master instance. Slaves can run on a variety of operating systems.

**Jenkins installed and running on “35.225.253.107”**

**MATER Jenkins URL -- http://35.225.253.107:8080/**

**Create new VM : node1**

**Set the password to the user**

**Enable PasswordAuthentication:**

**Restart SSHD service**

**[ysn4re@node1 ~]$ sudo su -**

**[root@node1 ~]# hostname**

**node1**

**[root@node1 ~]# passwd ysn4re**

**Changing password for user ysn4re.**

**New password:**

**BAD PASSWORD: The password is shorter than 7 characters**

**Retype new password:**

**passwd: all authentication tokens updated successfully.**

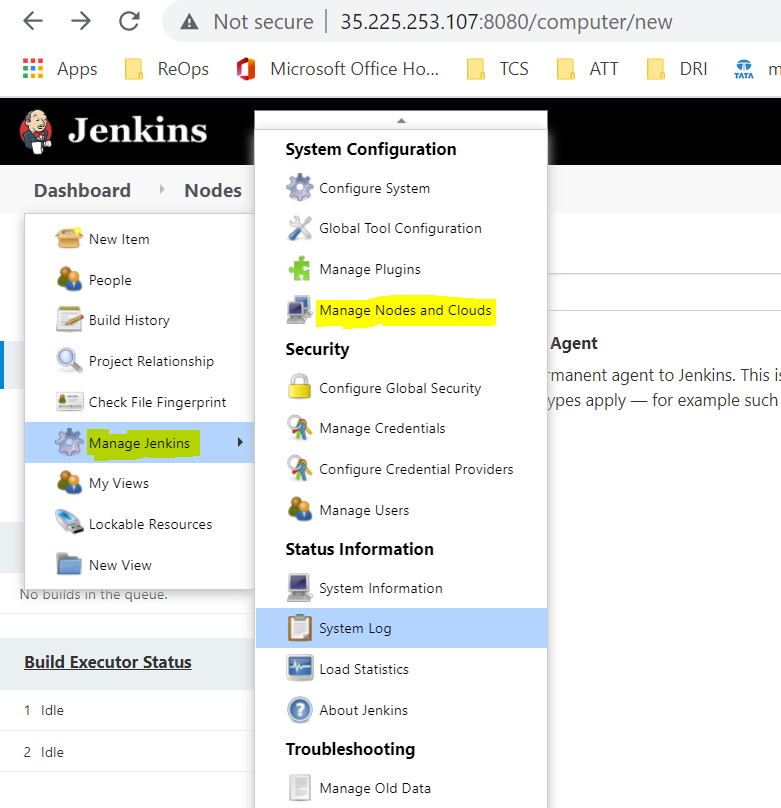
**[root@node1 ~]# sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd\_config**

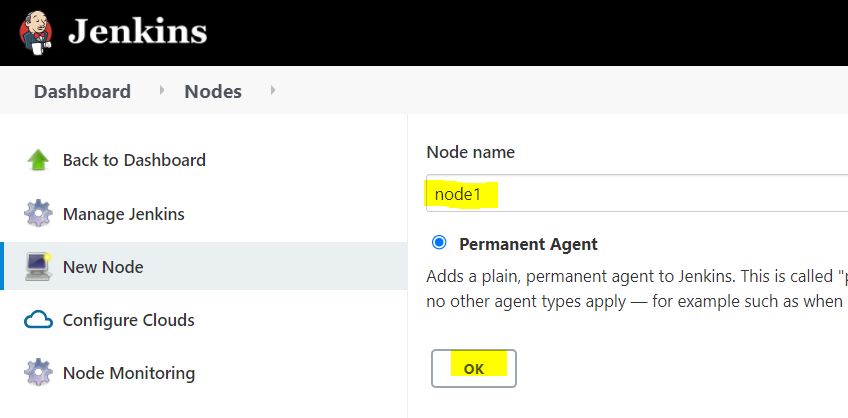
**[root@node1 ~]# sudo systemctl restart sshd**

**[root@node1 ~]#**

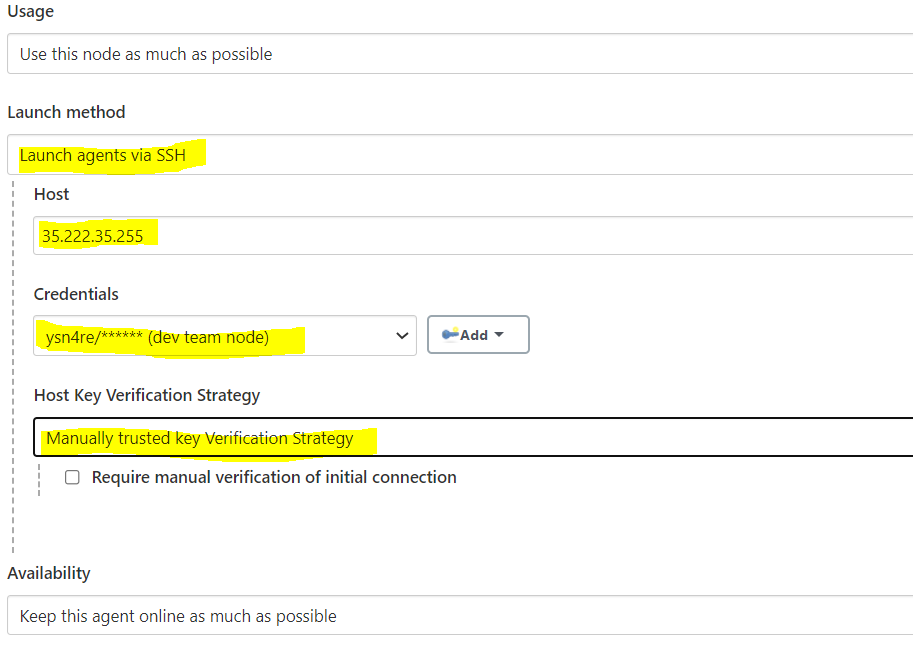
**Add node to Jenkins Master:**

**Manage jenkins → Manage Nodes and Clouds**

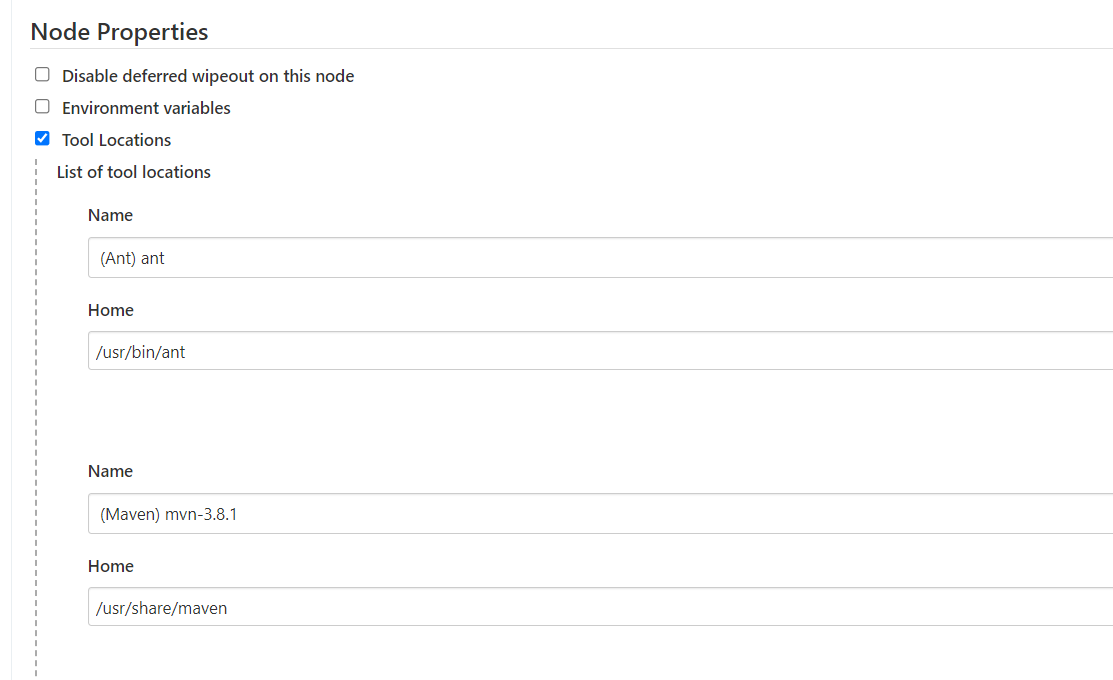
****

****

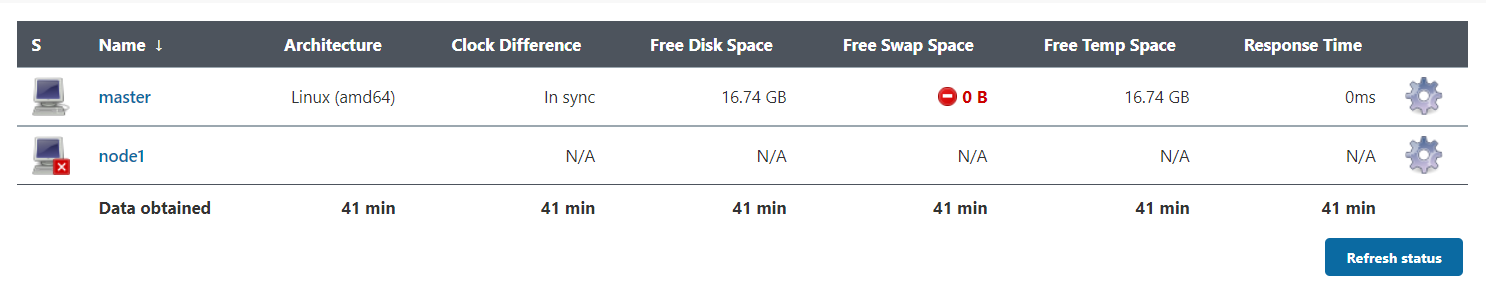
****

****

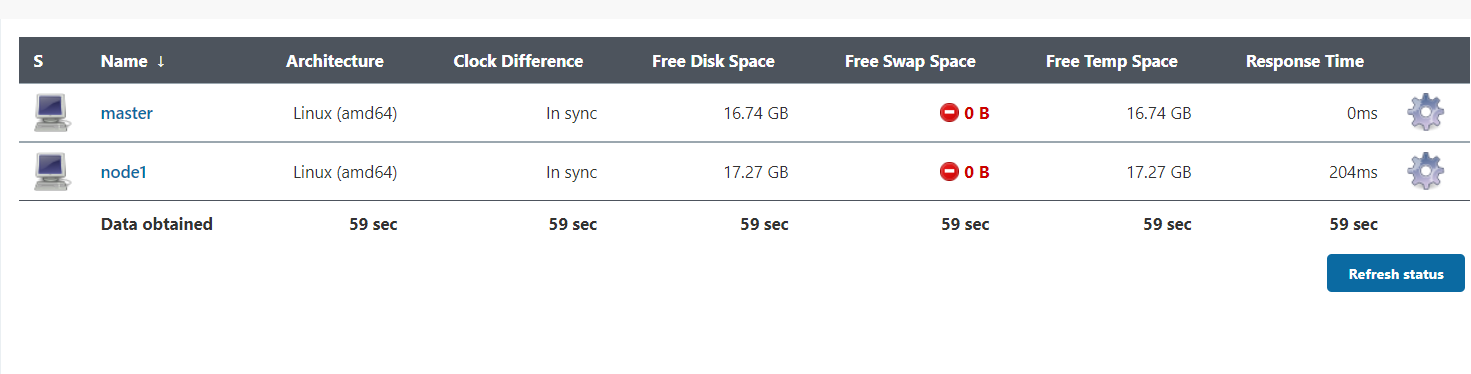
**Add tools info and click on save.**

****

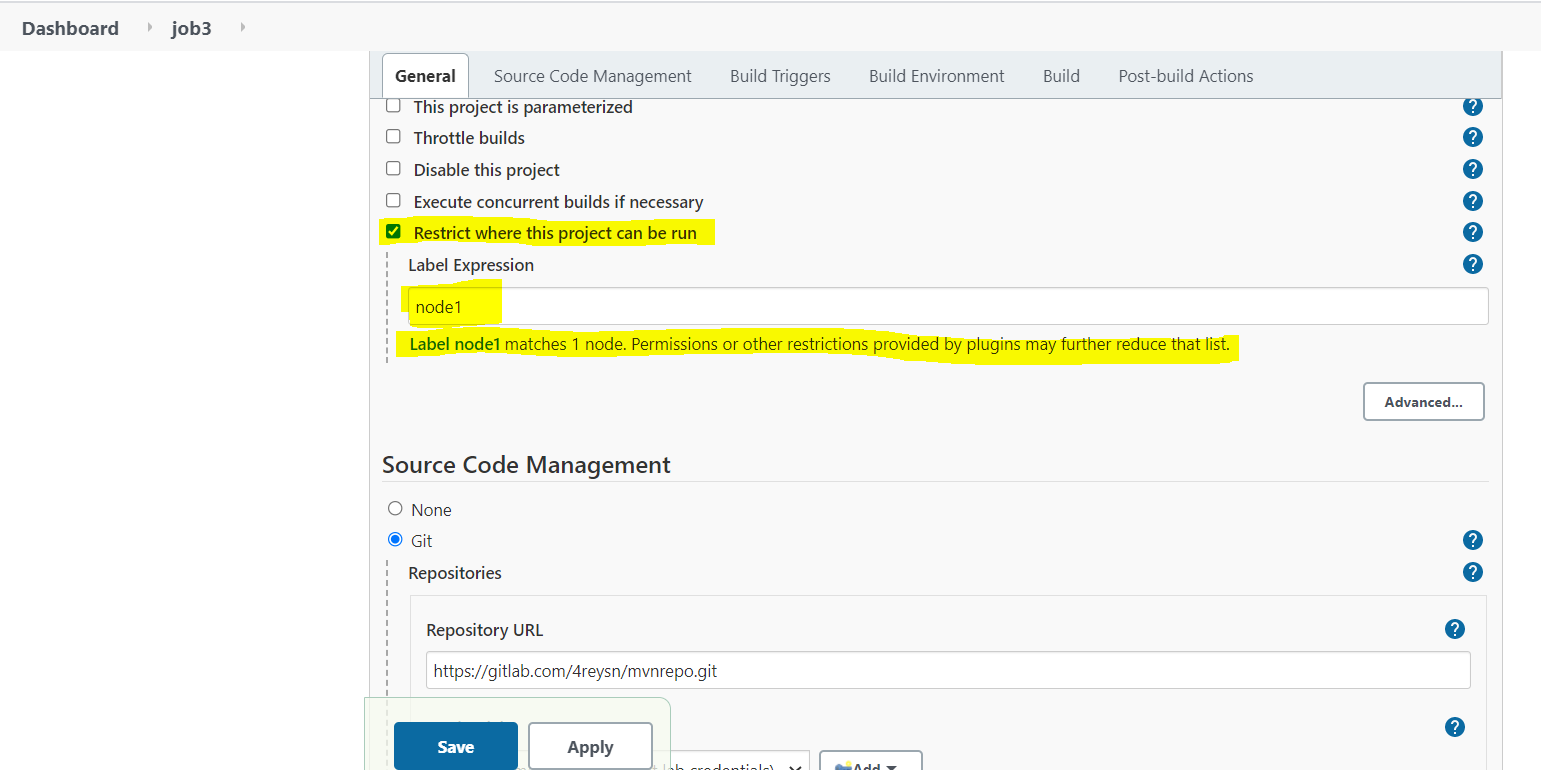
**Click on node1-34.134.2.115**

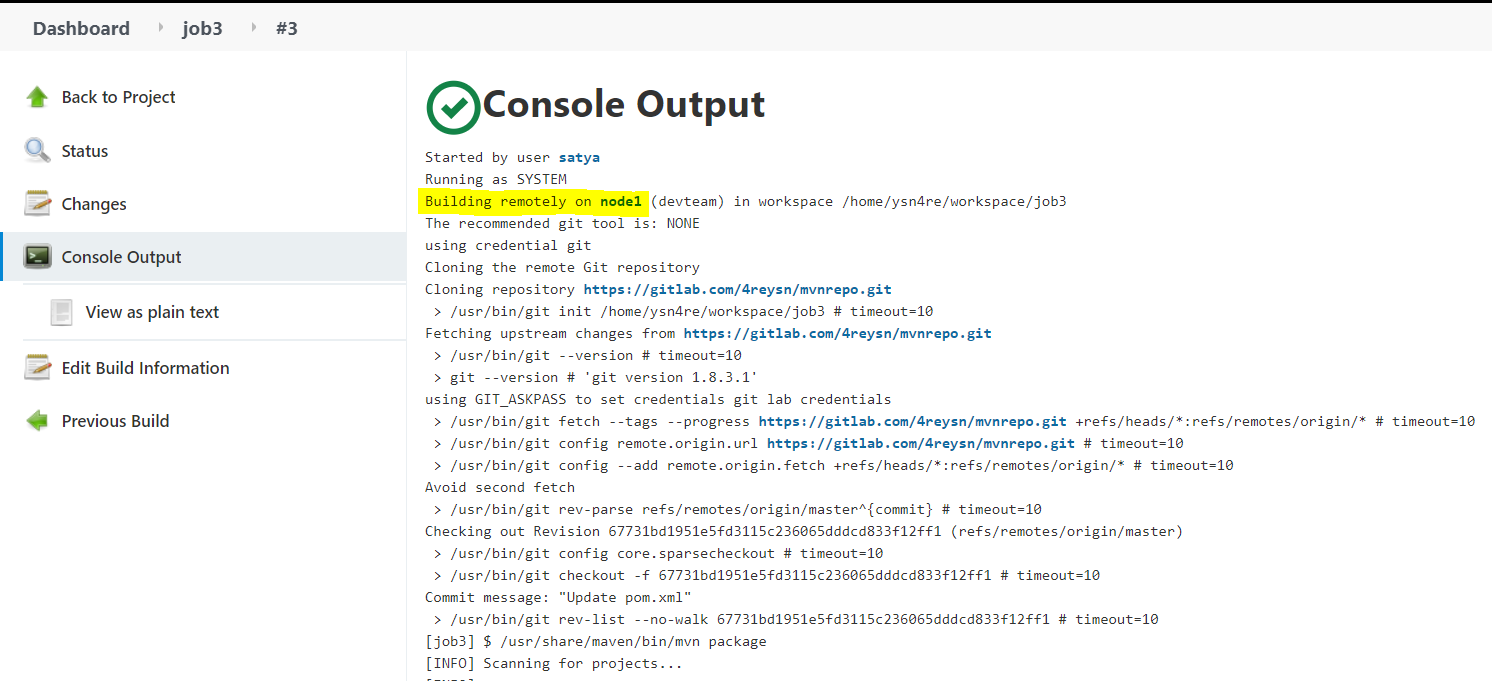
****

**It will launch the connection**

****

**Edit the job and configure to run the job on node**

****

****

**Nexus Installation Steps:**

**NOTE: In firewall rules allow 8081 port to access nexus from browser**

**1. sudo yum install wget -y**

**2. Download Java 8**

**sudo yum install java-1.8.0-openjdk.x86\_64 -y**

**you can confirm java is installed by typing the below command:**

**java -version**

**3. Download Nexus**

**cd /opt**

**sudo wget http://download.sonatype.com/nexus/3/nexus-3.26.1-02-unix.tar.gz**

**4. Extract Nexus**

**sudo tar -xvf nexus-3.26.1-02-unix.tar.gz**

**sudo mv nexus-3.26.1-02 nexus**

**5. Create a user called Nexus**

**sudo adduser nexus**

**6. Change the ownership of nexus files and nexus data directory to nexus user.**

**sudo chown -R nexus:nexus /opt/nexus**

**sudo chown -R nexus:nexus /opt/sonatype-work**

**7. Add Nexus as a user**

**sudo vi /opt/nexus/bin/nexus.rc**

**change as per above screenshot by removing # and adding nexus**

**run\_as\_user="nexus"**

**8. Configure Nexus to run as a service**

**sudo vi /etc/systemd/system/nexus.service**

**Copy the below content.**

**[Unit]**

**Description=nexus service**

**After=network.target**

**[Service]**

**Type=forking**

**LimitNOFILE=65536**

**User=nexus**

**Group=nexus**

**ExecStart=/opt/nexus/bin/nexus start**

**ExecStop=/opt/nexus/bin/nexus stop**

**User=nexus**

**Restart=on-abort**

**[Install]**

**WantedBy=multi-user.target**

**9. Create a link to Nexus**

**sudo ln -s /opt/nexus/bin/nexus /etc/init.d/nexus**

**10. Execute the following command to add nexus service to boot.**

**sudo chkconfig --add nexus**

**sudo chkconfig --levels 345 nexus on**

**11. Start Nexus**

**sudo service nexus start**

**sudo service nexus status**

**12. Once Nexus is successfully installed, you can access it in the browser by URL - http://public\_dns\_name:8081**

**14. Click on Sign in link & Reset password**

**user name is admin and password can be found by executing below command:**

**sudo cat /opt/sonatype-work/nexus3/admin.password**

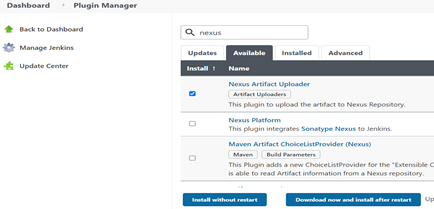
**Copy the password and click sign in.**

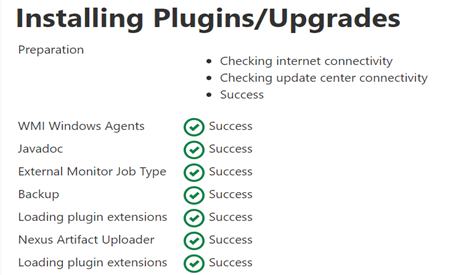
**Now setup admin password as admin123**

**Jenkins Nexus Integration - How to integrate Sonatype Nexus with Jenkins - Upload artifacts from Jenkins to Nexus repository:**

**Pre-requistes:**

**Install Nexus Artifact Uploader plugin in Jenkins.**

****

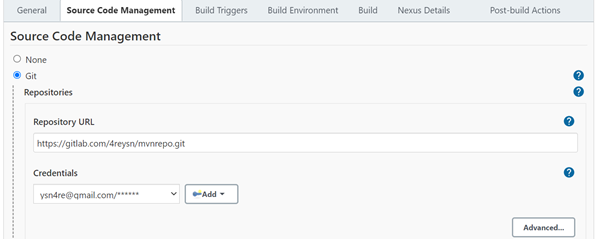
****

**Steps:**

**1. Once you installed the above plug-ins, click existing FreeStyle job configuration or create a new job.**

**2. Under source code management. enter git repo url and git credentials.**

**3. Go to build section, add Maven targets. with goals clean install. Also click advance, give path of POM.xml**

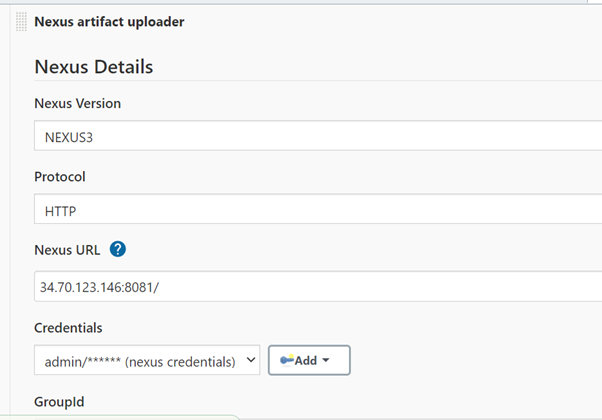
****

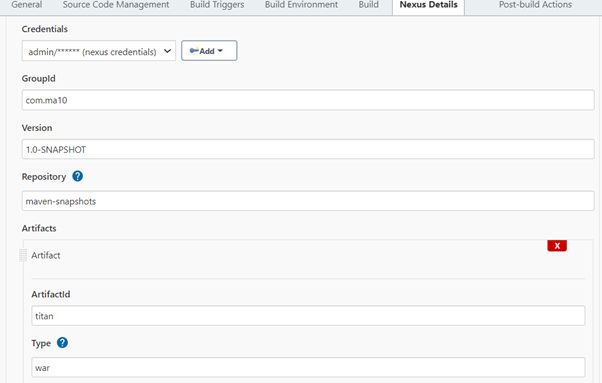
****

**4. click on Add build step, choose Nexus artifact uploader.**

**5. Enter information per below screen shot.**

**Make sure you enter version as 1.0-SNAPSHOT and repository as maven-snapshots**

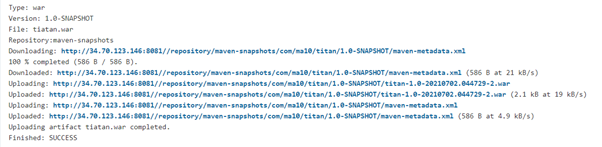
****

****

****

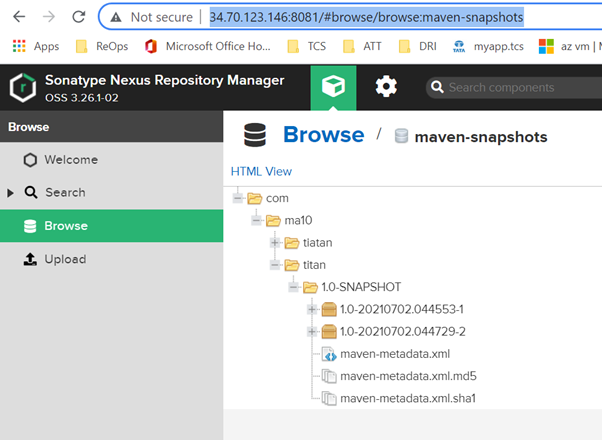
**6. Click on Apply, Save.**

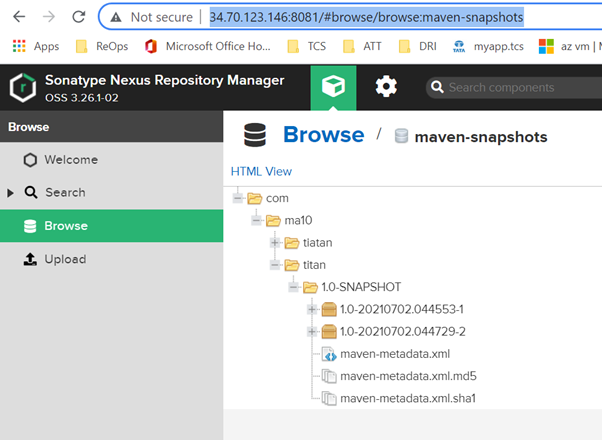
**7. trigger build**

****

**8.Now login to Nexus repo**

**http://34.70.123.146:8081/#browse/browse:maven-snapshots**

****

****

**PIPELINE:**

**What is the Jenkins pipeline?**

**Jenkins pipelines are nothing but a powerful way of implementing both CI and CD.**

**If you want to automate the build process,code quality process, artifactory repository and code deployment we will use the pipelines.**

**Jenkins has 2 pipelines**

**scripted pipeline and Declarative pipeline.**

**Scripted pipeline:**

**pipeline code in Groovy language.**

**Syntax:**

**node{**

**stage("checkout"){**

**{**

**stage("build"){**

**}**

**stage("Artifactory"){**

**}**

**stage("code deploy"){**

**}**

**}**

**Declarative Pipeline:**

**We can run each stage on a separate build agent in a declarative pipeline, which we can't do in a scripted pipeline.**

**simple pipeline syntax.**

**no complexity compared with scripted pipeline.**

**We can write the pipeline code in a jenkins file and we can push it to the source repository.**

**Syntax:**

**Pipeline{**

**agent any**

**tools {**

**}**

**stages{**

**satge("checkout"){**

**steps{**

**}**

**}**

**stage("build"){**

**steps{**

**}**

**}**

**stage("deploy"){**

**steps{**

**}**

**}**

**}**

**}**

[**https://gitlab.com/4reysn/pipeline-script.git**](https://gitlab.com/4reysn/pipeline-script.git)

**pipeline{**

**agent any**

**tools{**

**maven 'mvn-3.8.1'**

**}**

**stages{**

**stage('Checkout'){**

**steps{**

**checkout([$class: 'GitSCM', branches: [[name: '\*/Release\_0.17']], extensions: [], userRemoteConfigs: [[credentialsId: 'git', url: 'https://gitlab.com/4reysn/mvnrepo.git']]])**

**}**

**}**

**stage('Build'){**

**steps{**

**sh 'mvn clean package -f pom.xml'**

**}**

**}**

**stage('Nexus-Repo'){**

**steps{**

**nexusArtifactUploader artifacts: [[artifactId: 'tiatan', classifier: '', file: 'target/Release\_0.17.war', type: 'war']], credentialsId: 'nexus-repo', groupId: 'com.devops', nexusUrl: '34.123.187.134:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'maven-snapshots', version: '1.0-SNAPSHOT'**

**}**

**}**

**stage('PARAM-JOB'){**

**steps{**

**build job: 'PARAM-JOB', parameters: [string(name: 'branch', value: 'Release\_0.15')]**

**}**

**}**

**stage ('Approve for Deployment') {**

**steps {**

**echo "Taking approval from DEV Manager for QA Deployment"**

**timeout(time: 7, unit: 'DAYS') {**

**input message: 'Do you want to deploy?', submitter: 'satya'**

**}**

**}**

**}**

**stage('DEPLOY'){**

**steps{**

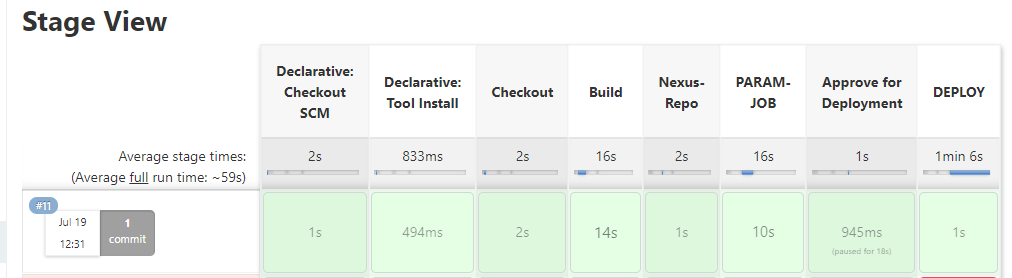
**deploy adapters: [tomcat9(credentialsId: 'tomcat9', path: '', url: 'http://34.72.229.27:8090')], contextPath: null, war: '\*\*/Release\_0.17.war'**

**}**

**}**

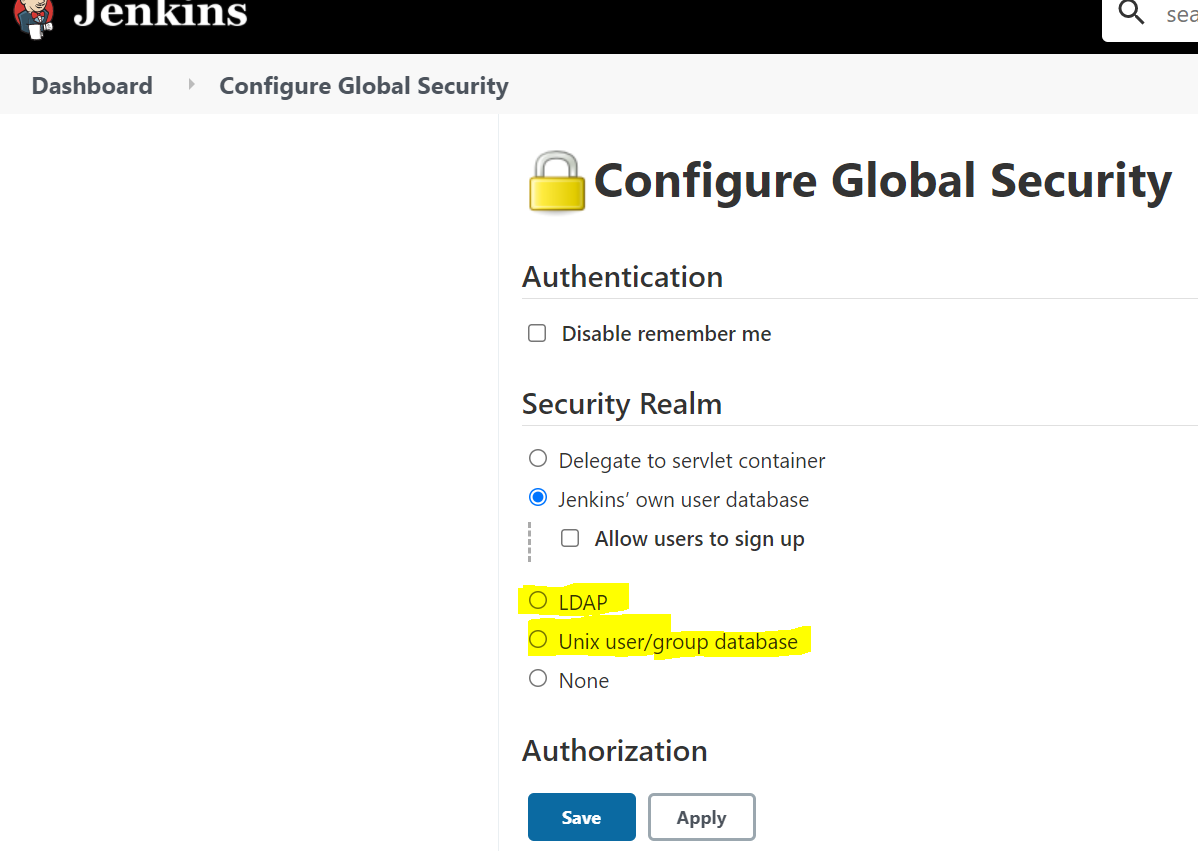
**}**

**}**

****

**USER Authentication:**

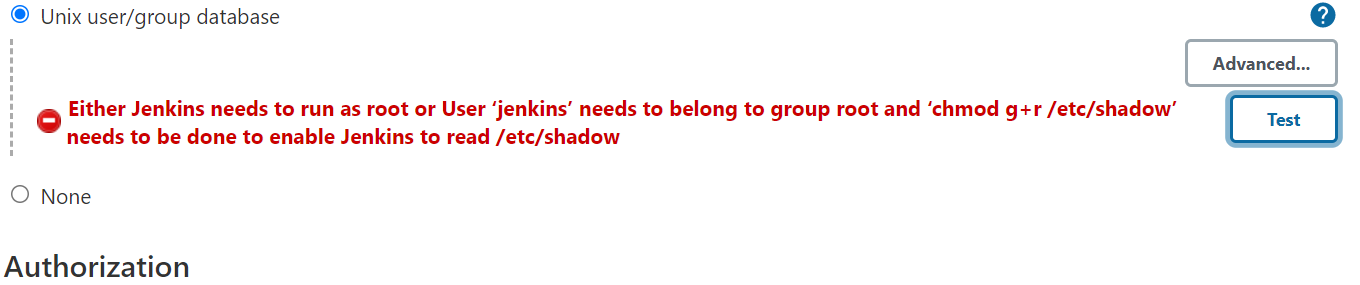
****

****

**In real time LDAP/UNIX user authentication will be used.**

**Select → Unix User/group database**

**And click on Test**

****

**TO enable jenkins user authentication as UNIX login:**

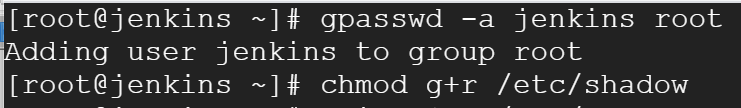
**User ‘jenkins’ needs to belong to group root to read /etc/shadow**

**[root@jenkins ~]# gpasswd -a jenkins root**

**Adding user jenkins to group root**

**[root@jenkins ~]# chmod g+r /etc/shadow**

**[root@jenkins ~]#**

****

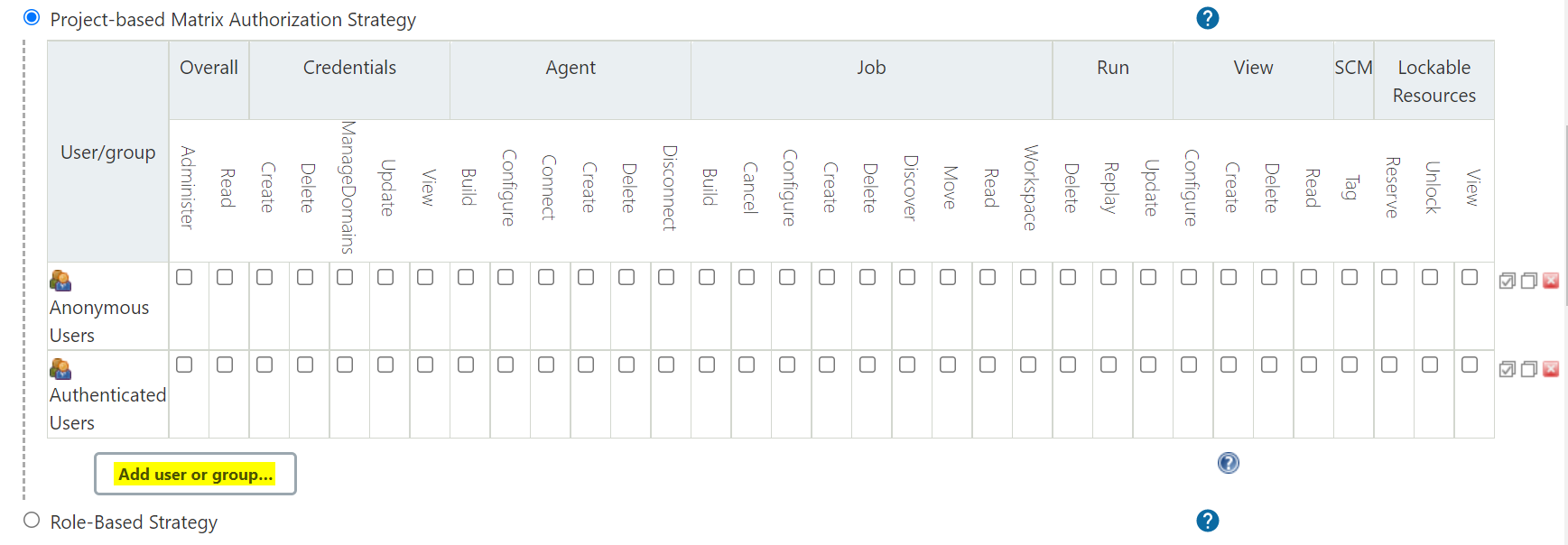
**systemctl status jenkins**

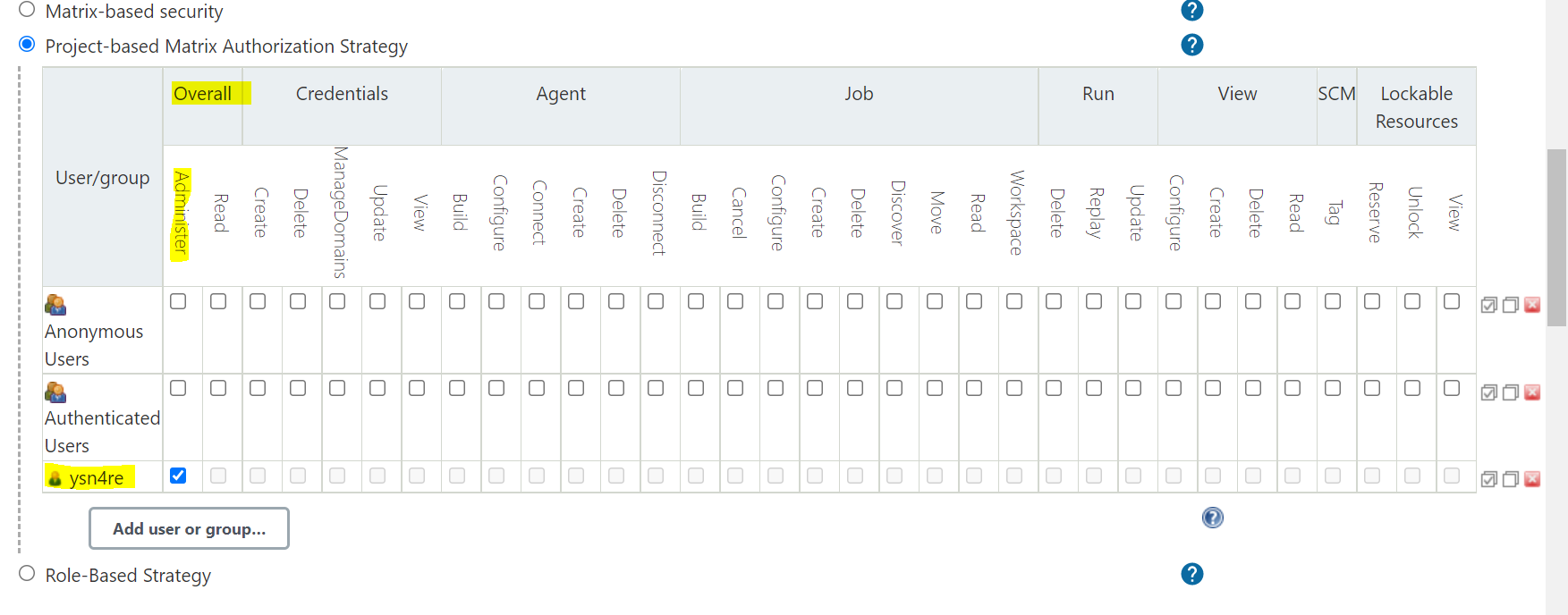
**systemctl stop jenkins**

**systemctl start jenkins**

****

**Then select Project based Matrix Authorization Strategy and click on Add user or Group**

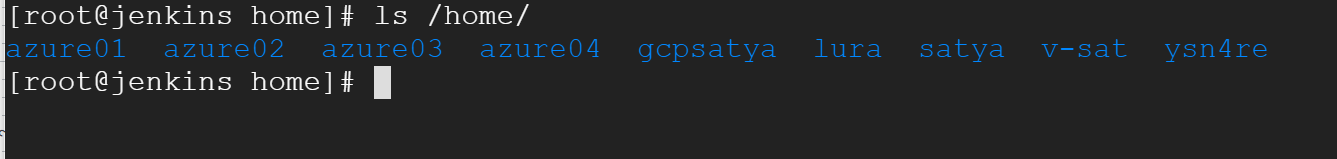
****

****

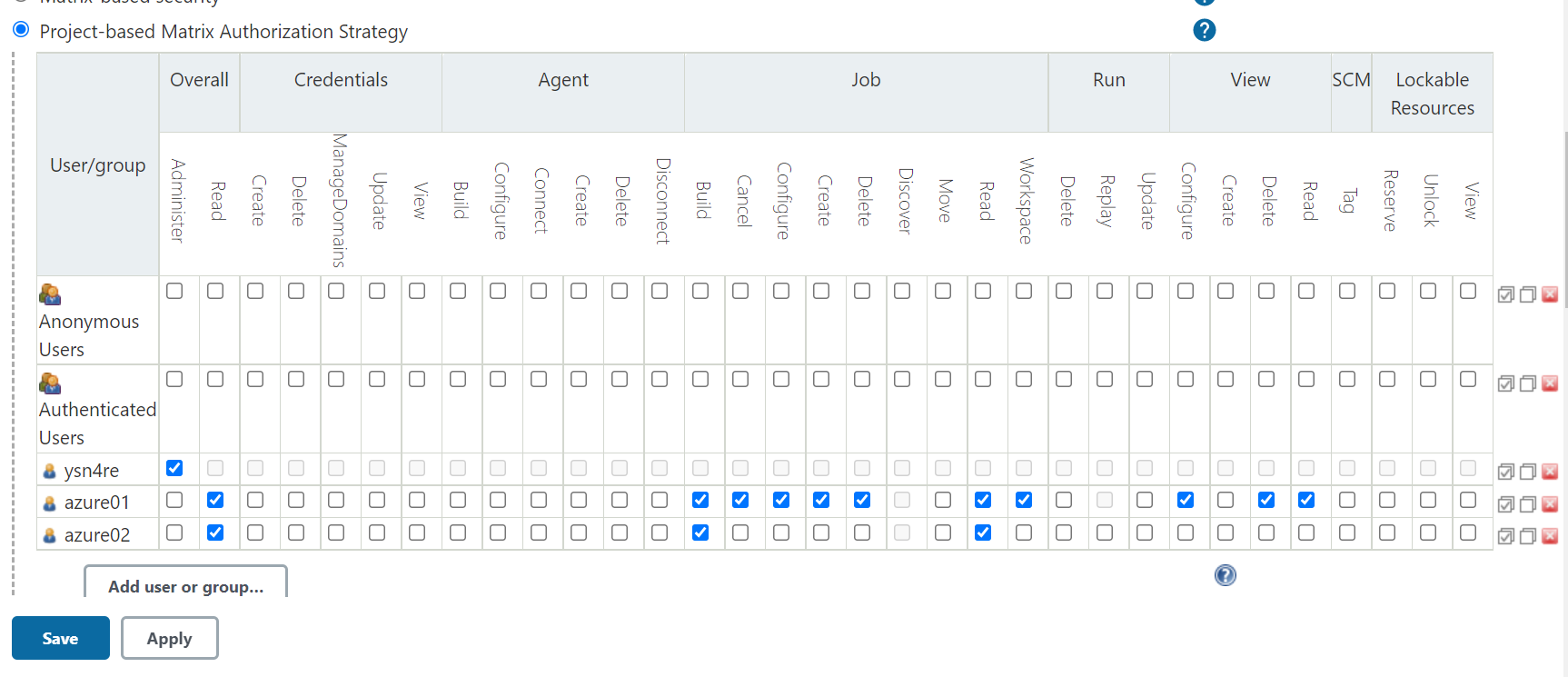
**I have added “ ysn4re” user and I gave administrator access to him.**

**Like that add other users which are available in your jenkins Server.**

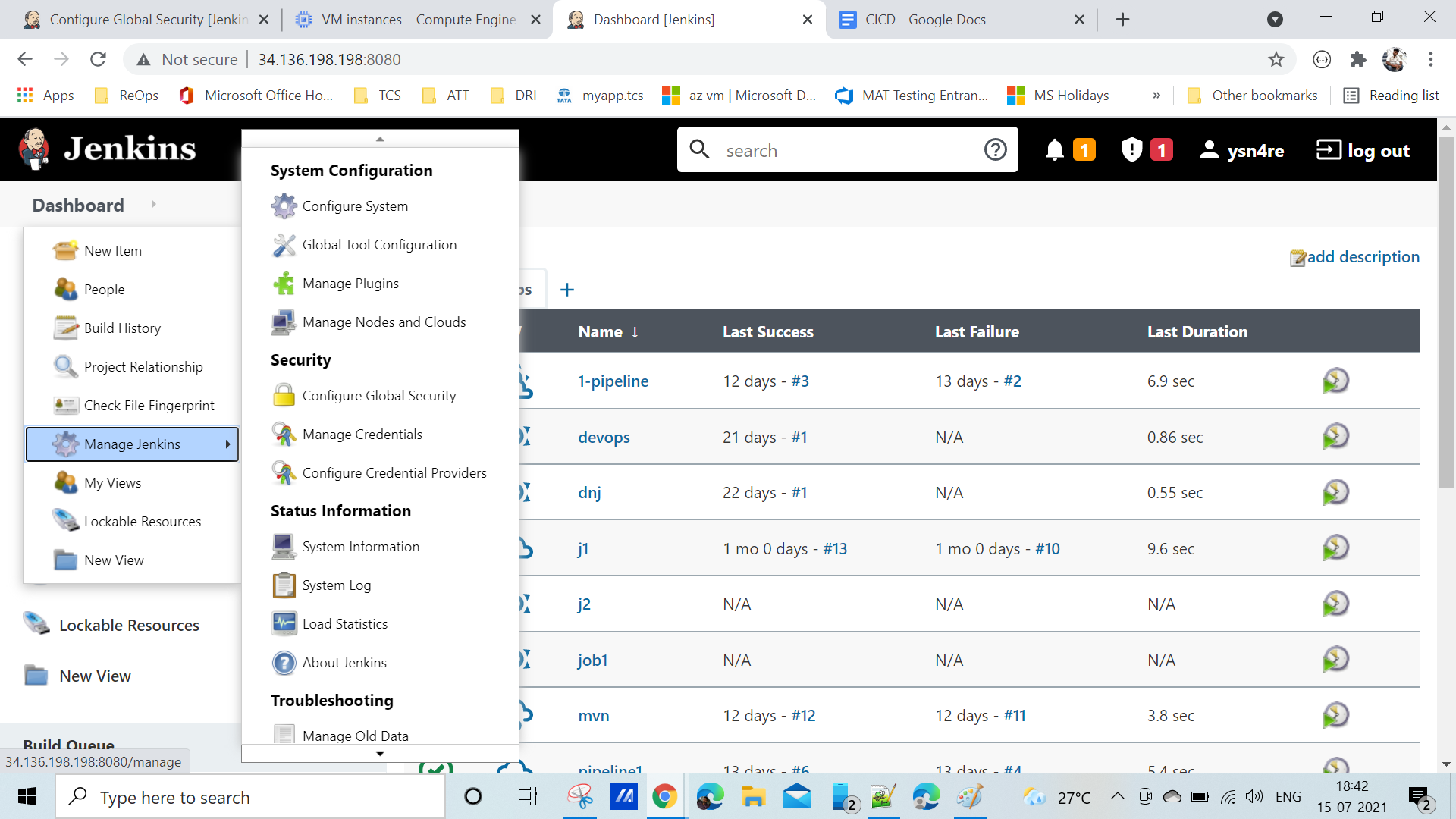
**Here i have below users in jenkins server(linux box)**

****

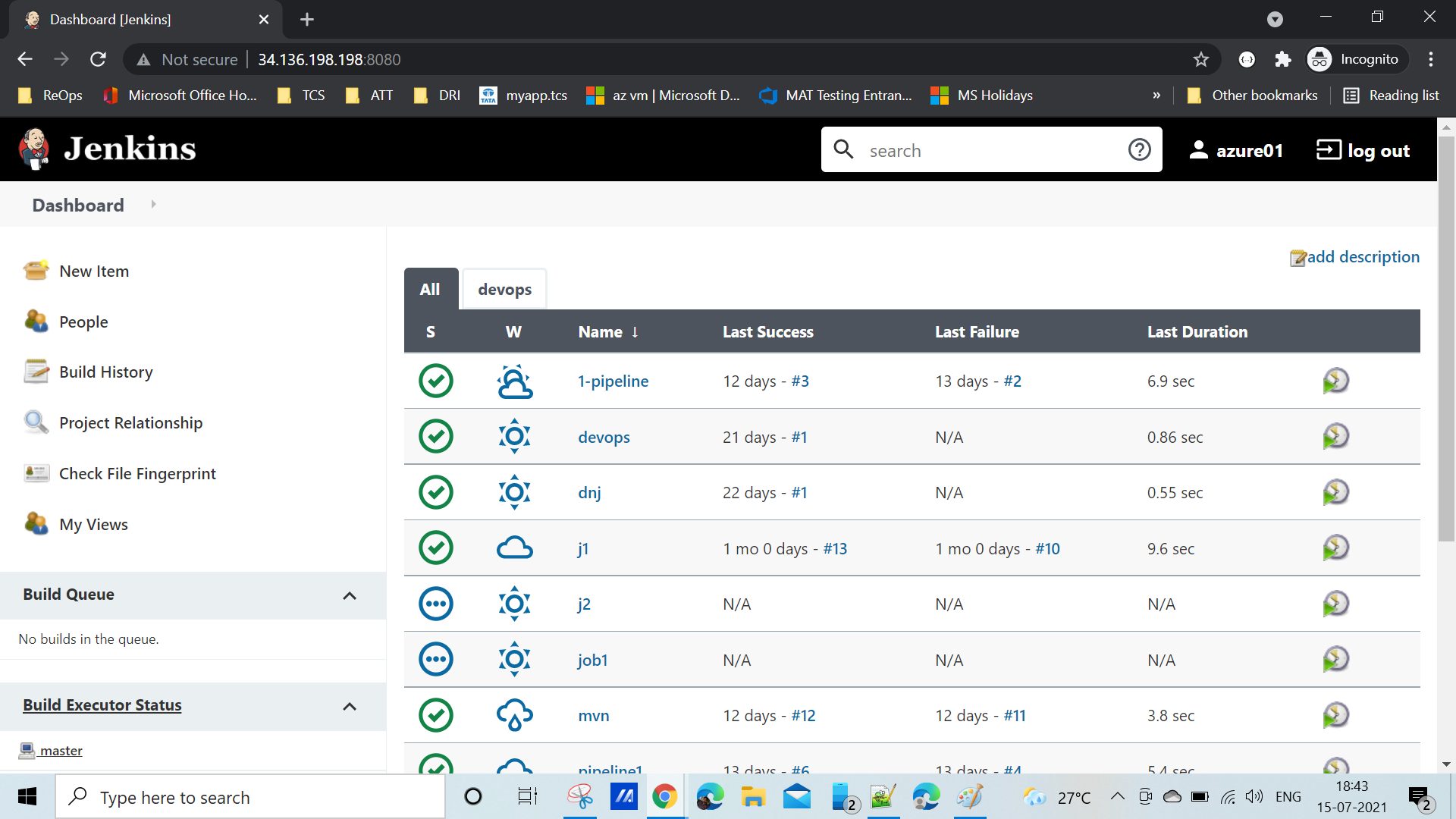
**NOw i will add azure01 and azure02 users and I will give required permissions to those users.**

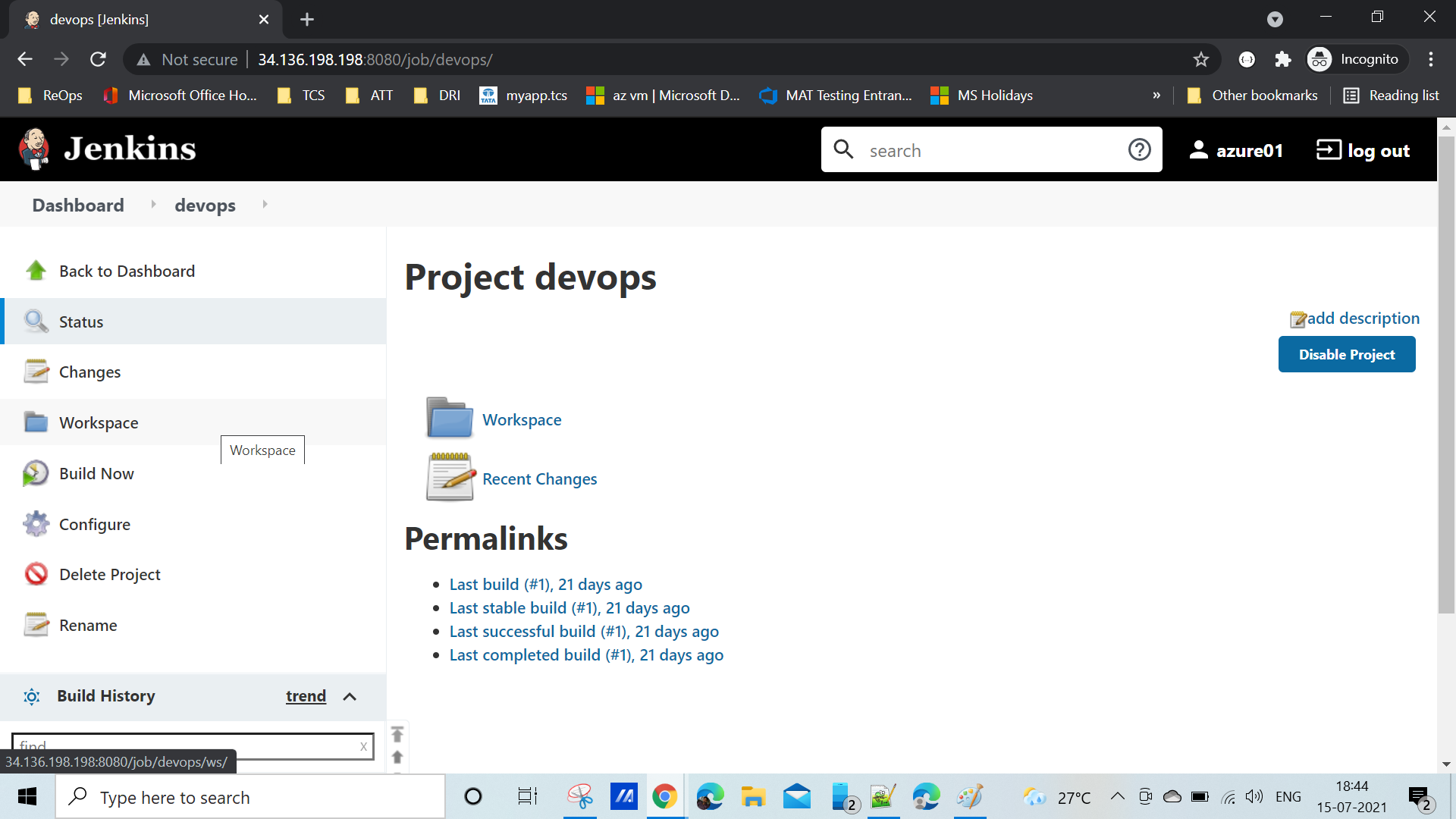
****

**User ysn4re**

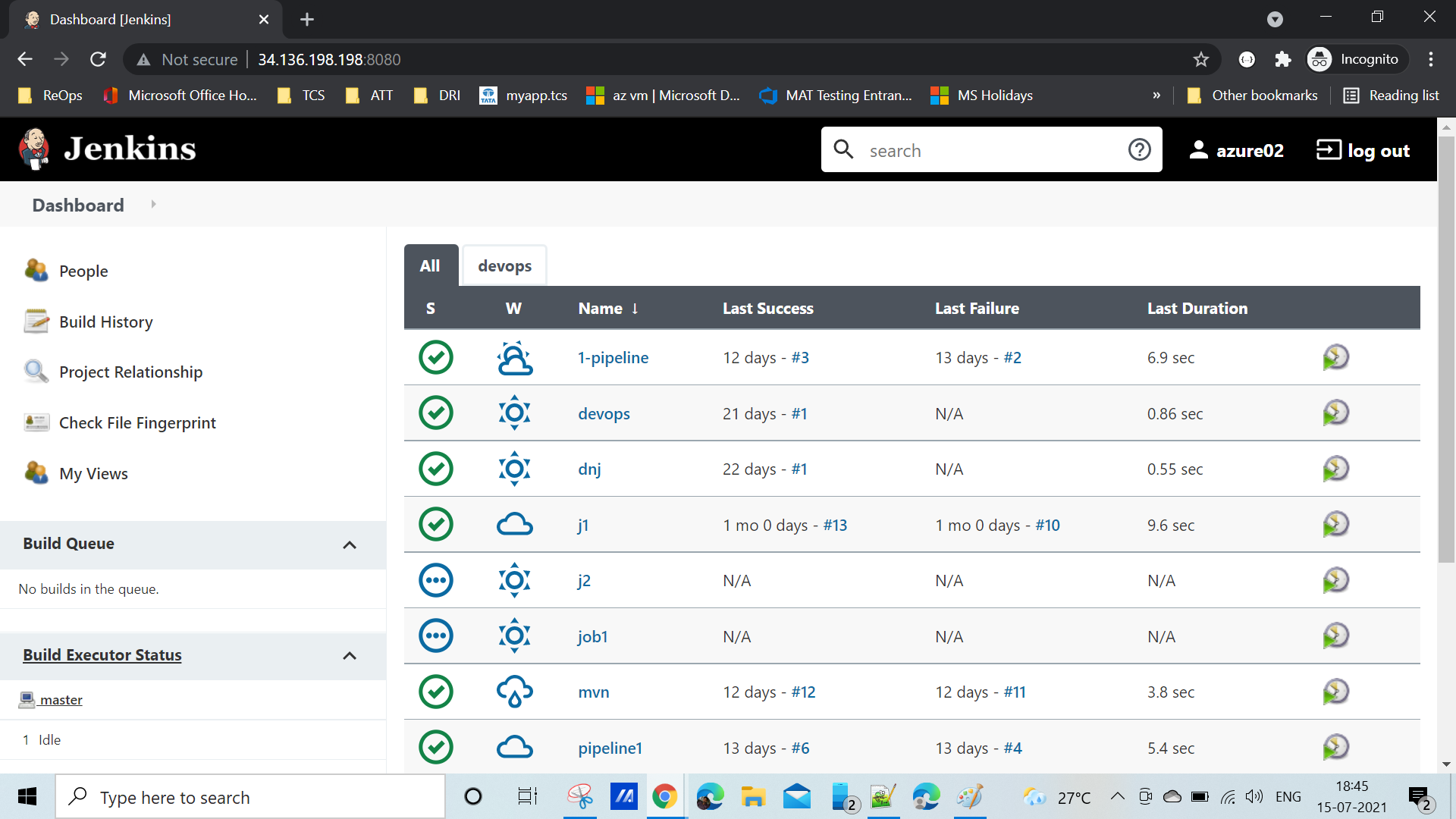
****

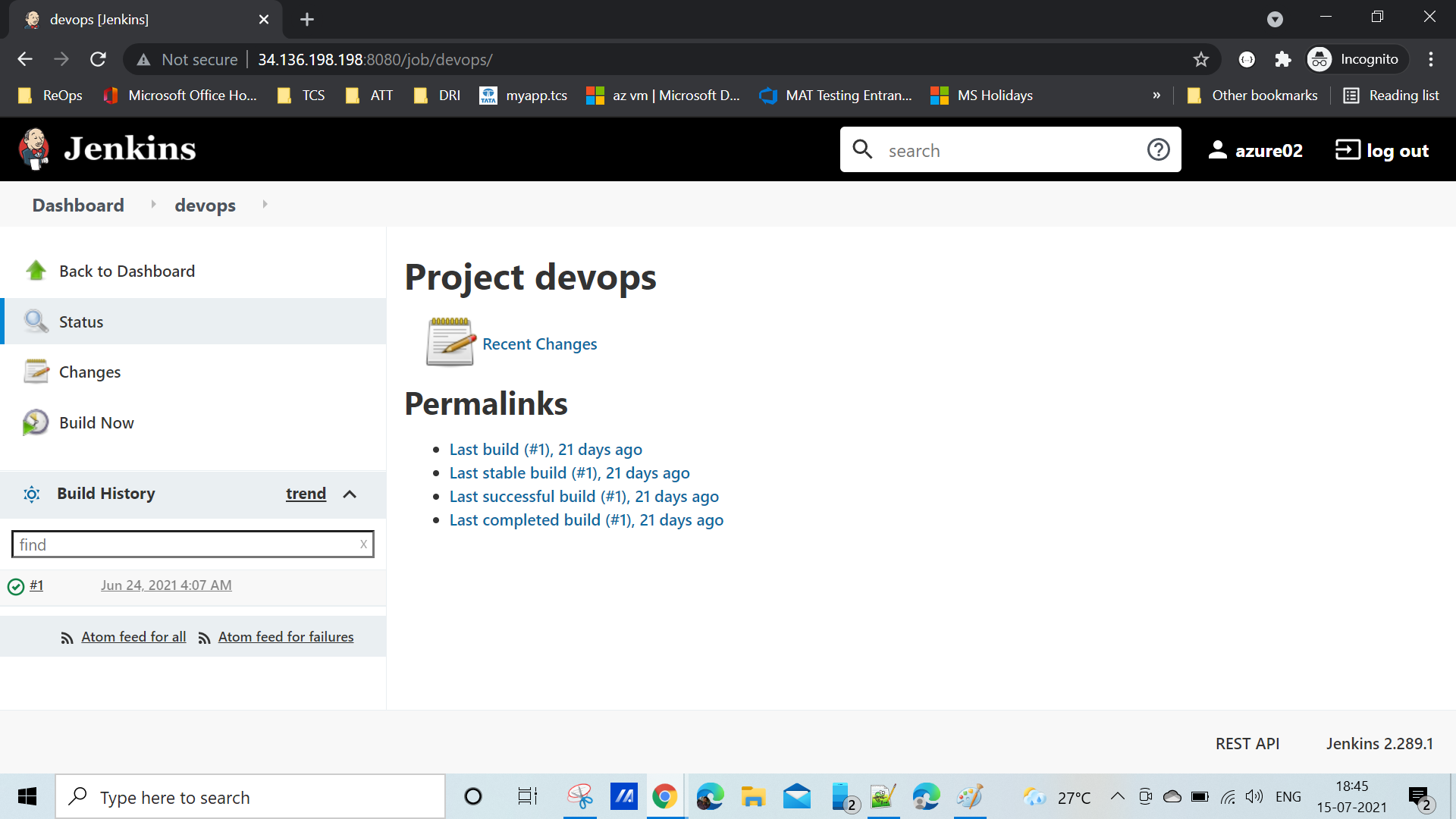
**User: azure01**

****

****

**User: azure02**

****

****

**Q & A:**

**What is Jenkins?**

**Jenkins is an open source continuous integration tool which is written in Java language. It keeps a track on the version control system and to initiate and monitor a build system if any changes occur. It monitors the whole process and provides reports and notifications to alert the concern team.**

**Explain what is continuous integration?**

**When multiple developers or teams are working on different segments of the same web**

**application, we need to perform integration tests by integrating all the modules. To do that an**

**automated process for each piece of code is performed on daily bases so that all your code gets**

**tested. And this whole process is termed as continuous integration.**

**What is the relation between Hudson and Jenkins?**

**Hudson was the earlier name of current Jenkins. After some issues were faced, the project name was changed from Hudson to Jenkins.**

**What are the advantages of Jenkins?**

**Advantage of using Jenkins**

**1. Bug tracking is easy at an early stage in a development environment.**

**2. Provides a very large number of plugin support.**

**3. Iterative improvement to the code, code is basically divided into small sprints.**

**4. Build failures are cached at integration stage.**

**5. For each code commit change an automatic build report notification gets generated.**

**6. To notify developers about build report success or failure, it can be integrated with LDAP**

**mail server.**

**7. Achieves continuous integration agile development and test-driven development**

**environment.**

**8. With simple steps, maven release projects can also be automated.**

**What is a Jenkins Pipeline?**

**Jenkins Pipeline (or simply ―Pipeline‖) is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.**

**What is the difference between Maven, Ant and Jenkins?**

**Maven and Ant are Build Technologies whereas Jenkins is a continuous integration(CI/CD)**

**tool.**

**Which SCM tools does Jenkins support?**

**Source code management tools supported by Jenkins are below:**

**1. AccuRev**

**2. CVS**

**3. Subversion**

**4. Git**

**5. Mercurial**

**6. Perforce**

**7. Clearcase**

**8. RTC and many more**

**How can you set up Jenkins jobs?**

**Steps to set up Jenkins job as follows:**

**Select new item from the menu.**

**After that enter a name for the job (it can be anything) and select a free-style job.**

**Then click OK to create a new job in Jenkins dashboard.**

**The next page enables you to configure your job, and it‘s done.**

**I have 50 jobs in the Jenkins dashboard, I want to build at a**

**time all the jobs?**

**In Jenkins there is a plugin called build after other projects build. We can provide job names over**

**there and If one parent job runs then it will automatically run all other jobs. Or we can use**

**Pipeline jobs.**

**How can I integrate all the tools with Jenkins?**

**I have to manage Jenkins and then global tool configurations. There you have to provide all the details such as Git URL , Java version, Maven version, Path etc**

**How can you avoid the waiting time for the triggered jobs in**

**Jenkins.**

**First I will check the Slave nodes capacity, If it is fully loaded then I will add the slave node by**

**doing the following process.Go to the Jenkins dashboard -> Manage Jenkins ->Manage Nodes**

**Create the new node a By giving the all required fields and launch the slave machine as you**

**want.**

**How will you take backup for Jenkins?**

**Copy JENKINS\_HOME directory and ―jobs‖ directory to replicate it in another server**

**What exactly defined is DevOps?**

**DevOps is all about bringing together the structure and process of traditional operations, so being**

**supported deployment, including any tools, also practices of traditional construction methods**

**such as source control and versioning.**

**Need for Continuous Integration:**

**Improves the quality of software.**

**Reduction in time taken to delivery**

**Allows dev team to detect and locate problems early**

**Success factor for the Continuous Integration**

**Maintain Code Repository**

**Automate the build**

**Perform daily checkin and commits to baseline**

**Test in clone environment**

**Keep the build fast**

**Make it easy to get the newest deliverables**

**Can we copy Jenkins job from one server to another server?**

**Yes, we can do that using one of the following ways:**

**We can copy the Jenkins jobs from one server to another server by copying the**

**corresponding jobs folder.**

**We can make a copy of the existing job by making clone of a job directory with different**

**Names Rename the existing job by renaming the directory**

**How can we create the backup and copy in Jenkins?**

**We can copy or backup, we need to back up the JENKINS\_HOME directory which contains the**

**details of all the job configurations, build details etc.**

**Difference between “poll scm” and “build periodically”**

**Poll SCM will trigger the build only if it detects the change in SCM, whereas Build Periodically**

**will trigger the build once the given time period is elapsed.**

### **What is Maven? What is the benefit of integrating Maven with Jenkins?**

**Maven is a build management tool. It uses a simple pom.xml to configure all the dependencies needed to build, test and run the code. Maven manages the full lifecycle of a test project. Once integrated with Jenkins, the maven Webdriver will build the project and execute all tests efficiently.**

### **What is a backup plugin? Why is it used?**

**This is a helpful plugin that backs up all the critical settings and configurations to be used in the future. This is useful in cases when there is a failure so that we don’t lose the settings.**

**What are the types of pipelines in Jenkins?**

**Scripted pipeline**

**Declarative pipeline**

### **What are the types of Jenkins pipelines?**

**Jenkins Pipelines can be either - a Declarative pipeline or a Scripted Pipeline. Declarative pipeline makes use of numerous, generic, predefined build steps/stages (i.e. code snippets) to build our job according to our build/automation needs whereas, with Scripted pipelines, the steps/stages can be custom-defined & used using a groovy syntax which provides better control & fine-tuned execution levels.**

### **What is Jenkinsfile?**

**Jenkins file is a text file that has a definition of a Jenkins pipeline and is checked into the source control repository. It enables code review and iteration on the pipeline. It also permits an audit trail for the pipeline.**

**How do you create Multibranch Pipeline in Jenkins?**

**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 that contain a Jenkinsfile in source control.**

### **What is a Jenkins Shared Library and how it is useful?**

**As an organization starts using more and more pipeline jobs, there is a chance for more and more code being duplicated in every pipeline job, since a part of the build/automation processes will be the same for most of the jobs. In such a situation, every other new upcoming job should also duplicate the same piece of code. To avoid duplications, the Jenkins project brought in the concept of Shared Libraries, to code - DRY - Don't Repeat Yourself.**

**Shared libraries are a set of code that can be common for more than one pipeline job and can be maintained separately. Such libraries improve the maintenance, modularity & readability of the pipeline code. And it also speeds up the automation for new jobs.**

**What are Triggers?**

**Trigger in Jenkins defines the way in which the pipeline should be executed frequently. PollSCM, Cron, etc are the currently available Triggers.**

**What are the differences between Continuous Integration, Continuous Delivery, and Continuous Deployment?**

| **Continuous Integration** | **Continuous Delivery** | **Continuous Deployment** |
| --- | --- | --- |
| Continuous Integration (CI) is a DevOps software development practice that permits developers to combine/merge the changes to their code in the central repository to run automated builds and tests. | Continuous Delivery (CD) refers to the building, testing, and delivering improvements to the software code. The most critical part of the CD is that the code is always in a deployable state. | Continuous Deployment (CD) is the ultimate stage in the DevOps pipeline. It refers to automatic release of any developer changes from the repository to the production stage. |

### **What is a CI/CD pipeline?**

CI/CD Pipeline or Continuous Integration/ Continuous Delivery is referred to as the DevOps approach's backbone. The pipeline is responsible for building codes, running tests, and deploying new software versions.