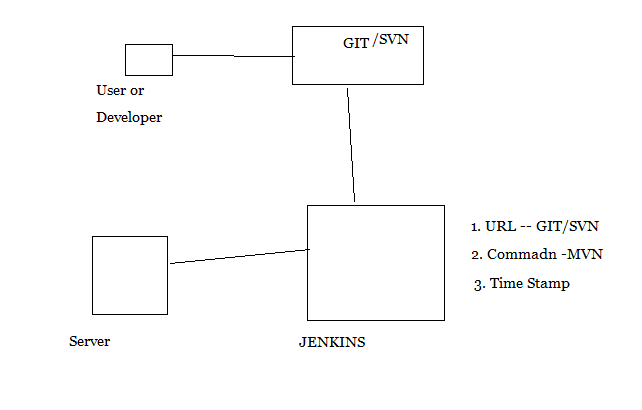
JENKINS

Jenkins called as continuous integration deployment or delivery tool ( CI tool )



**Jenkins Installation :**

→ Jenkins written in Java so JDK installation is required to execute it.

Jenkins installing in many ways

**Approach 1:**

1. Download and Install JDK

2. Download and install tomcat

3.Download Jenkins.war

4. Copy jenkins.war to $TOMCAT\_HOME/webapps ( Deploying Jenkins to Tomcat )

5. Start Tomcat Server ( $TOMCAT\_HOME(Tomcat Server Installation Location)/bin/./startup.sh ) ( To shutdown the tomcat $TOMCAT\_HOME/bin/./shutdown.sh )

6. Launch Jenkins Using below URL :

<http://localhost:8080/jenkins>

**Approach 2:**

Running Jenkins directly from command line , Not suitable for production jenkins

jenkins.war comes with a light weight server called “ jetty “

Below command runs jenkins in jetty server

$java -jar jenkins.war

Launch Jenkins using below command

http://localhost:8080

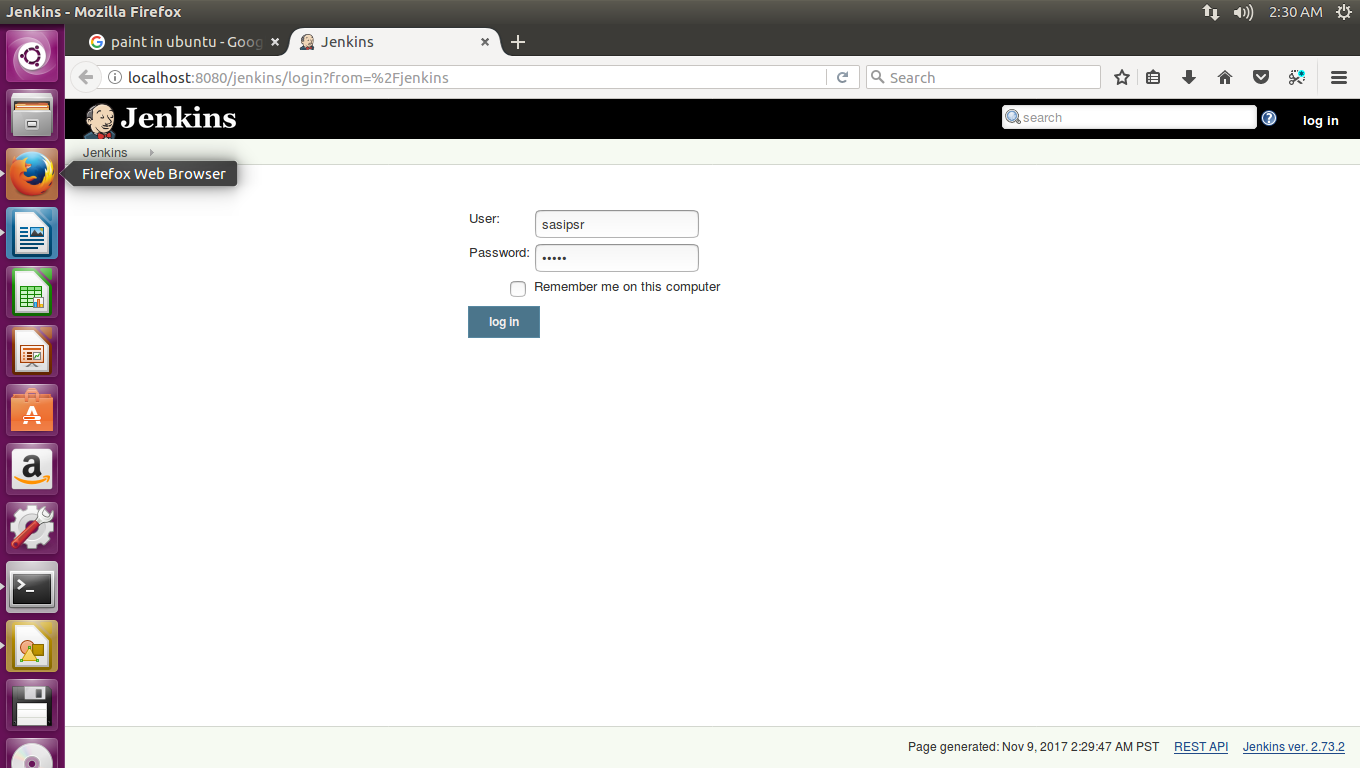
Approach 3:

sudo apt-get install jenkins

Jenkins access link is :

<http://localhost:8080/jenkins> (or) [http://IP](http://IP/) address:8080/jenkins

If it’s existed it will show like this



( Note: To start Jenkins tomcat should be running or in active state only( To check tomcat is running or not :

ps -ef | grep java ) If Tomcat server is in inactive mode need to active this using below command :

Go to tomcat installation foleder /bin and type ./startup.sh )

**Create a Sample project :**

After login click on “ create new jobs “ --→> Enter An Item Name ( Project Name ) -→> Free Style Project →> Ok

Next click on Jenkins ( Top left corner )

Next Click on Manage jenkins -→> Configure System: ( In this we have lot of options to do work on the project )

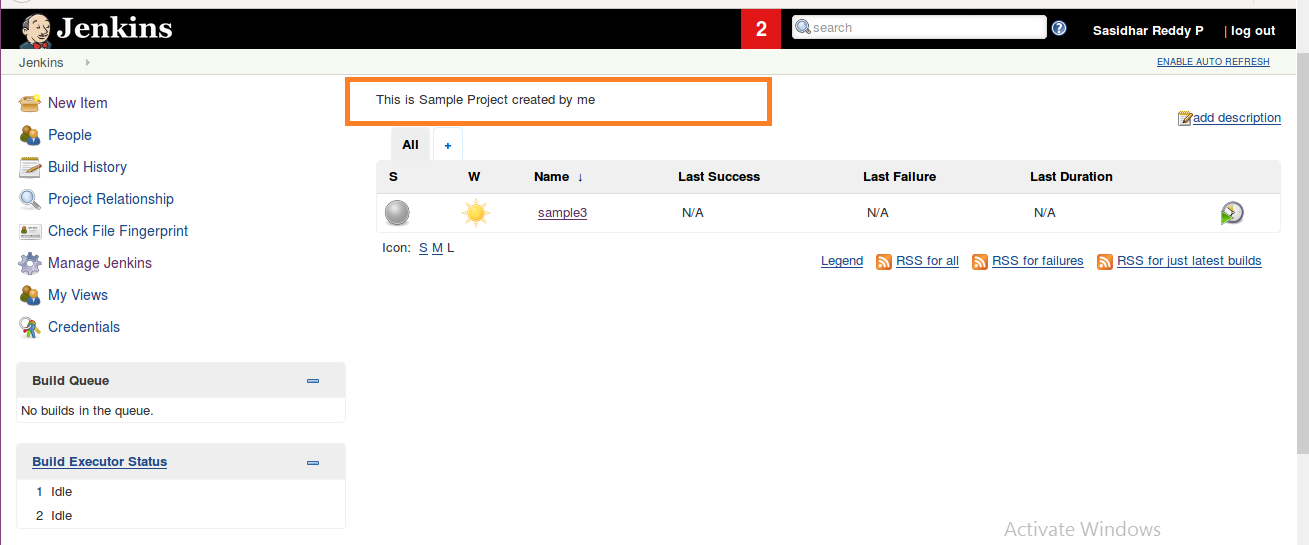
**1. home directory** , We can change if we want . To change it

$ USER\_HOME/.bashrc and add the below environment variable

export JENKINS\_HOME=/home/scripts/jenkins/.jenkins

**2. System Message** : Means we can write our own text about project update or any system message (like System will be shutdown from 3 pm to 6 pm ) in our own way. We can write it in many languages like html etc….

Ex :



If we want make it as more attractive or much suitable we can write it in html also.

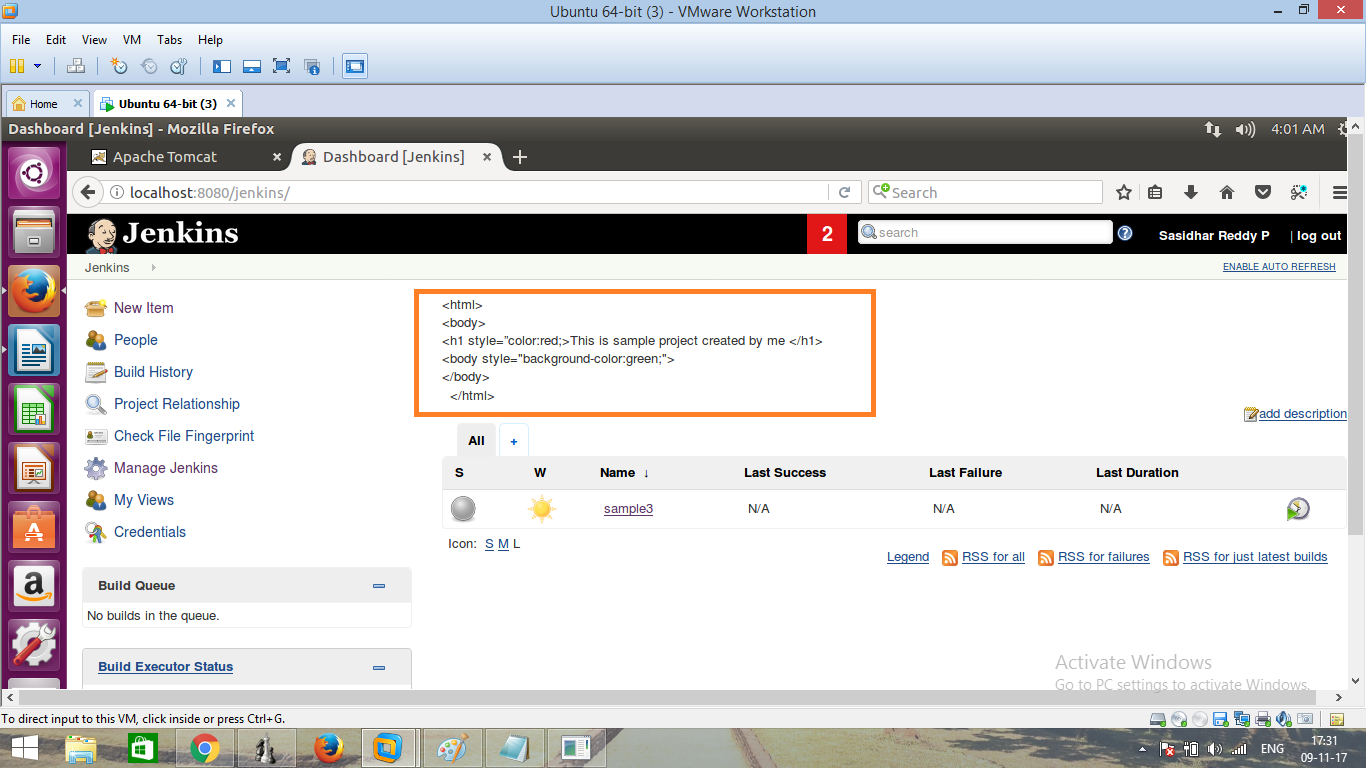
Ex: We can write same text “ This is Sample Project “ In HTML .

<html>   
 <body>

<h1 style=”color:red;>This is sample project </h1>  
</body>

</html>

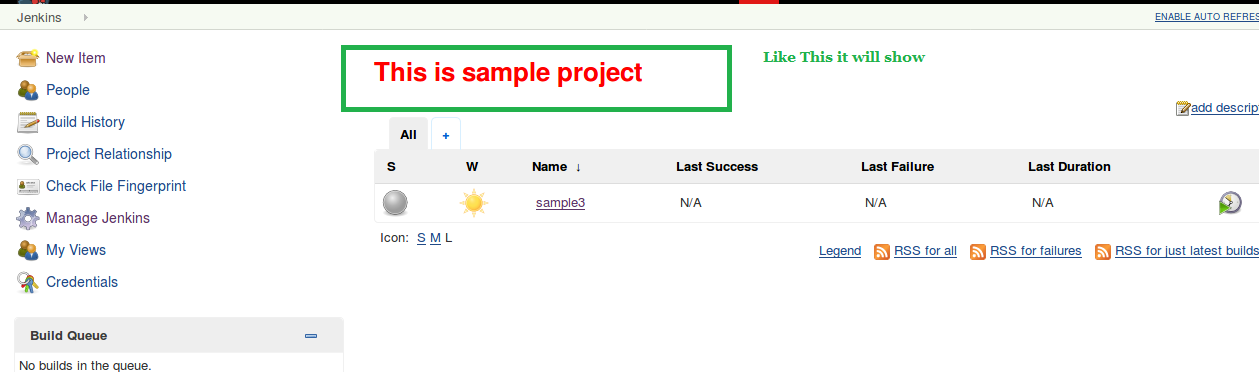
After apply and save also it’s showing same html code as shown below



So in this case we need to change markup formatter from plain text to html safe

Go to manage jenkins-→> Configure Global Security -→> Markup Formatter -→> plainhtml/safe HTML: select safe HTML-→ Apply-→. Save

Then It will show like this :



**3. # of executors** : means if # of exexutors is 2 then while building the project if we want to run 3 or 4 jobs only 2 jobs will run first and remaining 2 are in queue . So if we want to run 20 jobs at a time we need set the value is 20

**4. Lables:**

**5.Usage:**

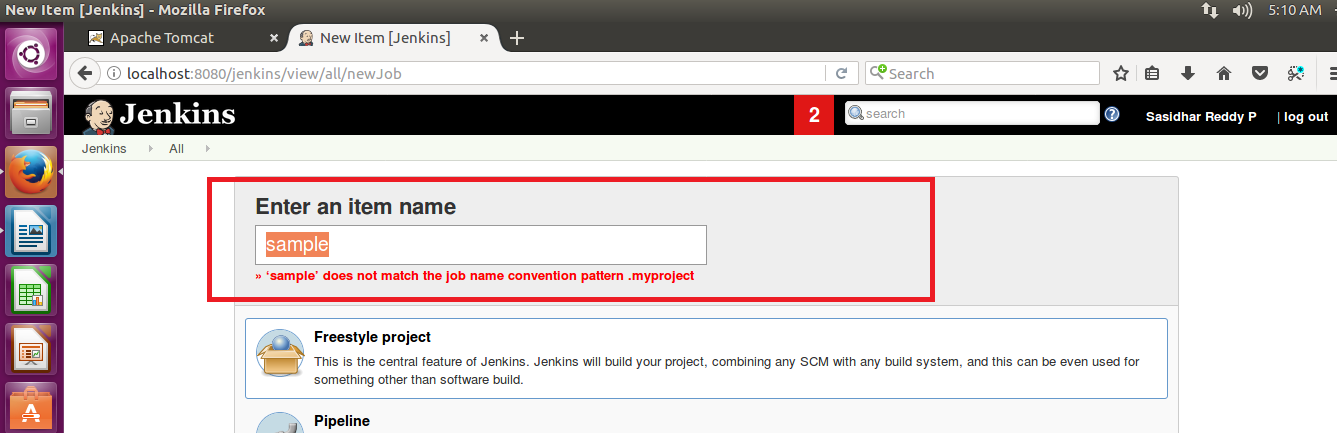
**6.Quiet Period:**

**7. SCM checkout retry count:** If any connection failure how many times need to be try to reconnect : For safer side it should be zero because If any network connection problem then only we notify that there is a connection problem immediately.

**8. check box- Restict project Name:** If we want a project name with a particulars then we can set this.

Ex: If we want every project name ends with .my project then we mentioned here “ .my project “

manage jenkins -→> configuration -→> Restict project naming →> pattern → > enter text in name patterns



**9. Global Properties :**

**10.Usage statistics :**

**11. Time Stamper :**

**12.Administrative monitors configurations** :

**13.Jenkins Location** : By default Jenkins location is <http://localhost:8080/jenkins>

System Admin email address :

**14. Git Hub :**

**a) GIT Hub server details**:

**15. GIT Hub enterprise servers:**

**16. Pipe line model defination:**

**a) Docker Label**

**b) docker registery URL**

**c) Registry credentials**

**17. Global pipe line Libraries:**

**18.Build-timeout Plugin > BuildStep Action**

**a) Check box-Enable BuildStep Action**

**19.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Git plugin** | | | |
| **a)** | **Global Config user.name Value :** |  |  |
|  | |  |  |
|  |  | |  |
| **b)** | **Global Config user.email Value** |  | Help for feature: Global Config user.email Value |
|  | |  |  |
|  |  | |  |
| **c)** | **Create new accounts based on author/committer's email – Check Box –** |  | Help for feature: Create new accounts based on author/committer's email |
|  | |  |  |
|  |  | |  |

**20.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subversion** | | | |
| a) | Subversion Workspace Version |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Help for feature: Subversion Workspace Version |  |  |  |
|  | |  |  |
|  |  | |  |
| b) | Exclusion revprop name |  | |

21.

|  |  |  |
| --- | --- | --- |
| Shell | | |
| a) | Shell executable |  |

22.

|  |  |  |  |
| --- | --- | --- | --- |
| Extended E-mail Notification | | | |
| a) | SMTP server |  | Help for feature: SMTP server |
|  | |  |  |
|  |  | |  |
| b) | Default user E-mail suffix |  | |

23.

|  |  |  |  |
| --- | --- | --- | --- |
| Extended E-mail Notificati | | | |
|  | SMTP server |  | Help for feature: SMTP server |
|  | |  |  |
|  |  | |  |
|  | Default user E-mail suffix |  | |

24. Default Content Type

25. Check box – Use List Id Email header

26. Check Box – Add 'Precedence: bulk' Email Header

27. Default Recipients

28.Reply To List

29.Emergency reroute

30.Excluded Recipients

31.Default Subject

32.Maximum Attachment Size :

33.Default Content:

34.Default Pre send script:

35.Default Post-send Script

36.Additional groovy classpath

37.Check Box –-Enable Debug Mode

38. Check Box –-Require Administrator for Template Testing

39. Check Box –-Enable watching for jobs

40.Check Box – Allow sending to unregistered users

41. Content Token Reference

42.Email Notification :

a) SMTP server:

b) Default user e-mail suffix

43.Check Box – Test configuration by sending test e-mail

**ENABLE SECURITY SETTINGS IN JENKINS:**

Manage Jenkins -→> Configure Global Security -→>

1. Enable Security : Should be enable for To enable security settings

2. Access Control :

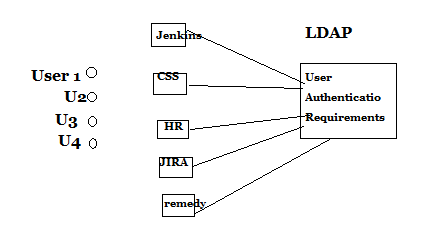
Here we concentrate mainly on two options

A) Jenkins Own User Data base : It allows

B) LDAP :

LDAP means In a organization different portals or there like jenkin. HR, remedy, CSS etc..

For each and every portal we need to create a user name and password to do access , So it’s very difficult to create it for all portals and delete also . To overcome this we create the credentials in LDAP server and point it to all portals like shown in below :

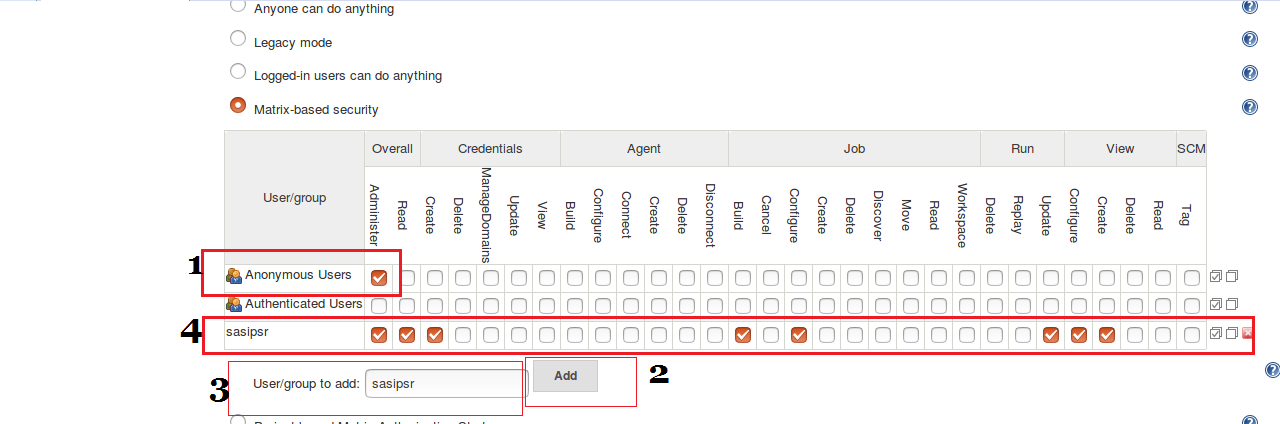


3. Authorization : In this Matrix-based security is important .

Marix-based Security : In this “ Anonymous Users “ need to be Administrative

and we can create different users and give particular credentials like shown below.

( Here we created a new user sasipsr )



4. Mark Up Formatter : It means A project heading or note mentioned in which format means in plain text or HTML format .

5.CSRF Prtotection :

6.CLI:

7. Plugin Manager :

8.Hidden Security Manager :

9.SSH server :

RELOAD CONFIGURATION FROM DISK :( Modify anything from backend )

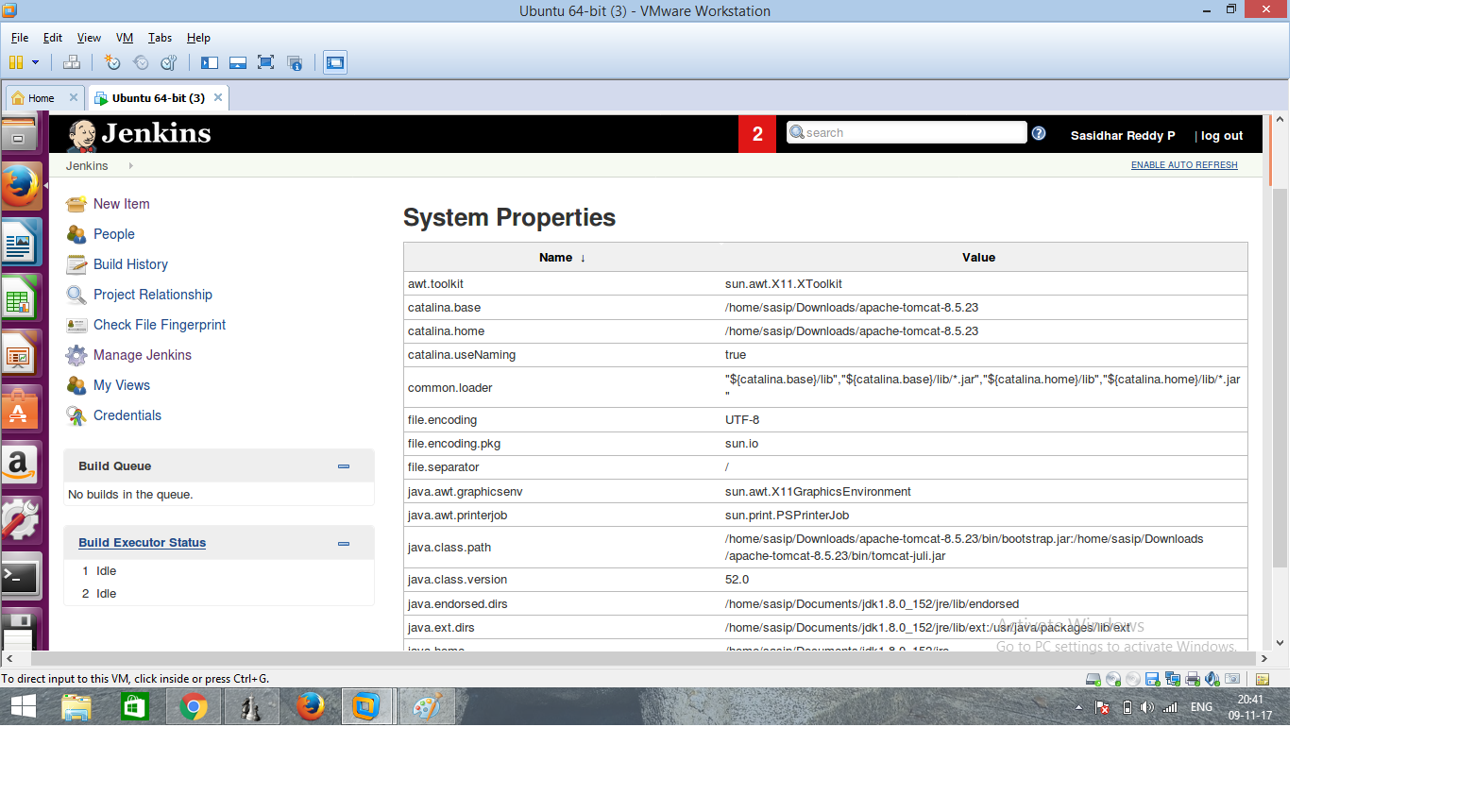
Suppose we modified a file from CLI it’s not reflecting in Jenkins web app, For this we need to click it on Reload Configuration From Disk

Ex: Suppose we modified in pom.xml as JDK version changed from JDK 7 to JDK 8

In this case we need to reflect in Jenkins We have to click on Reload Conf From Disk.

System Information:

It will show all system information like home directory etc…



System Logs :

It will show all system logs like Jenkins logs and jenkins errors and configurtion history and records of each and every job change or modification ( In CLI we can see the jenkins error logs in Jnekinslogs/jenkinserrors