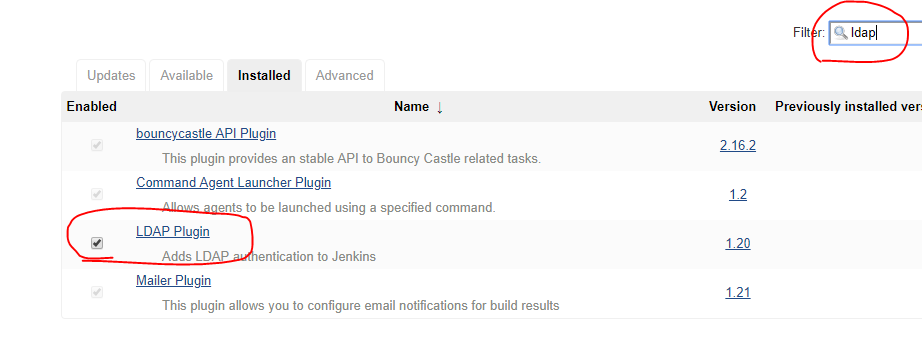
**Jenkins-Roles-and-User-management-with-LDAP**

|  |
| --- |
| Note: Please view this image in “Web Layout” to see the complete image,  right side bottom of the word doc. |

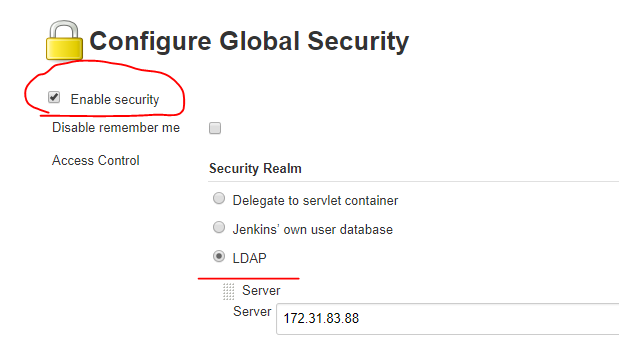
**Install LDAP Plugin in Jenkins**:

Manage Jenkins 🡪 Manage Plugin 🡪 Installed 🡪 Search for LDAP plugin. If this plugin not there under the section “installed” go to “Available” tab & search for LDAP plugin then install it.



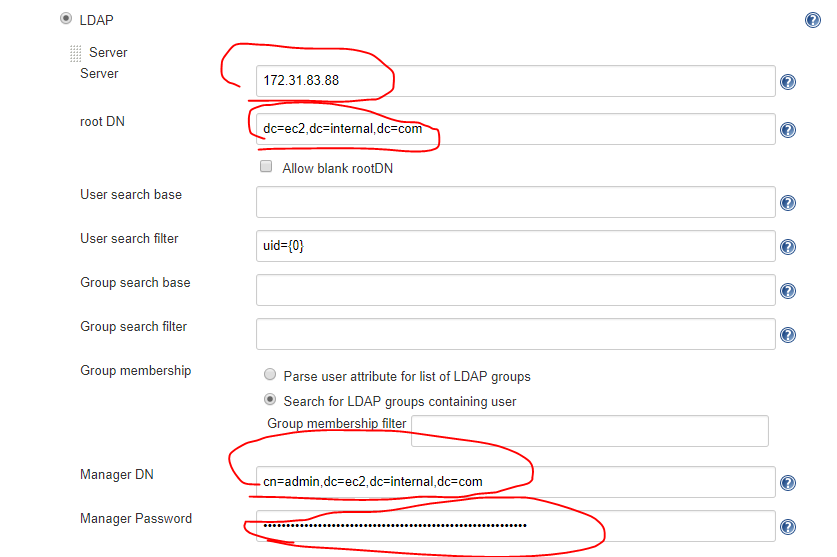
**Configure LDAP server in Jenkins**:

Manage Jenkins 🡪 Configure Global Security 🡪 Enable Security 🡪 LDAP.

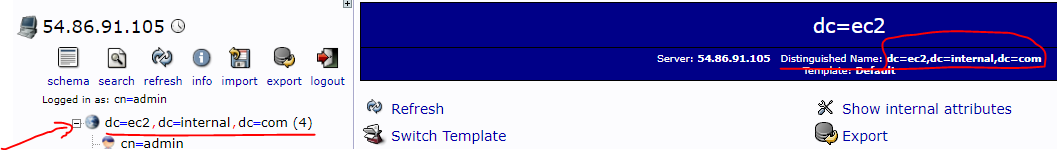


Give the LDAP server, root DN, Manager DN, manager password details (see the next image about LDAP to find & get the root DN, Manager DN details). Others are default values: (Give the Private IP address of AWS instance as the public IP frequently changing for every restart. Otherwise, public IP also works).

LDAP Config in Jenkins:



Root DN details in LDAP Server: (to copy the root DN details, just click on the “dc=ec2, dc=internal, dc=com” as arrow mark showing in below image, and copy the rounded one as showing in right side).

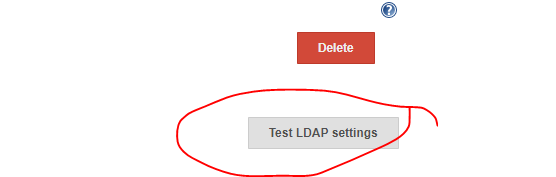


Root DN details in LDAP Server: Click on cn=admin and copy the DN.



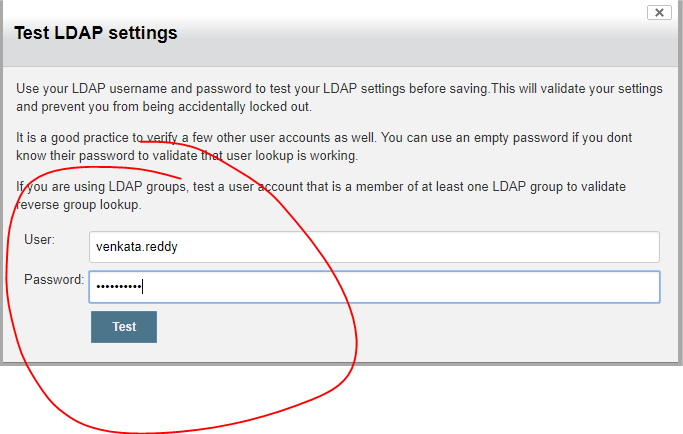
|  |
| --- |
| Additionally, if you want, you can give the access on Jenkins to the specific group: (Optional Configuration), so that Jenkins will load the users list from the specified LDAP group. This is one way of restricting access to the other groups members. For our practice I am not configuring this. But you can test it later once this practice completed. |

Test the connectivity & authentication: Click on “Test LDAP Settings”.

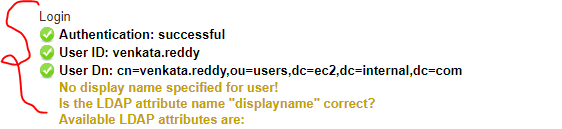




Enter any username & password which you created in LDAP server. Enter only CN.



Connectivity from Jenkins to LDAP server is success:



**Authorization**: There are several ways to provide authentication to the Jenkins user. (Please see the below image and follow the same numbers below which shown on the image).

1. Anyone can do anything: If you choose this, without login, anyone can have admin rights on Jenkins. See the below image with the name “Image-1-Anyone-can-do-anything”. Without login anyone can manage Jenkins. (This is not recommended at all).
2. Logged-in users can do anything: If you choose this, any logged in user can do anything. i.e., any authenticated user can access Manage Jenkins, then user can change anything in Jenkins. (This is also not recommended).
3. Matrix-based security: As sown in below image, there you can add the usernames which you created in LDAP. Add the usernames & give the permission read, delete, Administer, Configure etc.
   1. See the below image: “Image-3.a.-Adding-server-Matrix-based security” : Adding one user “venkata.reddy” (this username already created in LDAP server) and giving permissions as “Administer”. See the image “Image-3.a.-Login”.
   2. As shown in the below image, there are two default groups “Anonymous Users” & “Authenticated Users” under the “User/group”.
      1. “Anonymous Users” means any un-authorized user. No sign-in required. Basically, in realtime, we give “read” access as to access the Jenkins, and can able to see the Jenkins job etc. See the below image “Image-3.b.i-adding-Anonymous-read-only” & “Image-3.b.i-accessing-Jenkins-Anonymous-read-only”. (Not recommended to give any other access apart from ‘read’ to non-logged-in user i.e., Anonymous Users.)
      2. Authenticated Users means any logged in user. Practice yourself by giving some permissions “read” etc and see what will happen. In realtime, ‘read’ only access is enough. (Not recommended to give any other access apart from ‘read’ to logged-in users i.e., Authenticated Users. For example, in an organization, everyone is a user in LDAP i.e., the user is authenticated to the entire organization. User might be a test engineer, DB admin, manager, network team etc. So, even though the user is authorized (i.e., account exists in LDAP), user should **not** be able to perform trigger/build, configure, update, delete etc. So, the Admin of the Jenkins will decide & give the appropriate access to the user based on the business justification & managers approvals.
4. Project-based Matrix Authorization Strategy: Adding users, give permission to added users/Anonymous/Authenticated is same as above. But if you choose this, each & every job wise we can provide access. Job-1 can be accessed & trigger by one team. Job-1 can be accessed & trigger by other team. We can restrict each & every job access. See ‘Image-4-Project-Based-Access’.
5. Role-Based Strategy: Please refer the other document, “Jenkins-Role-Based-Strategy.doc”

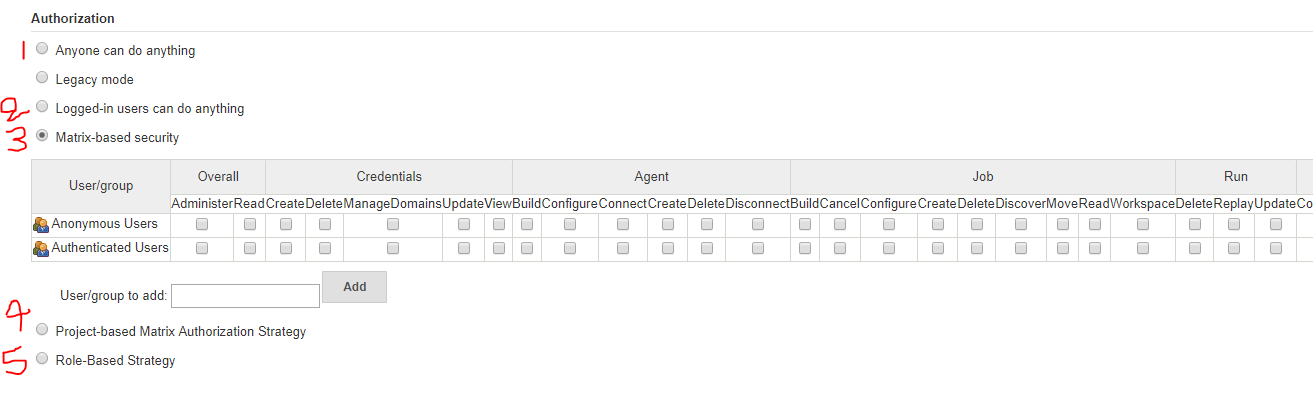


Image-1-Anyone-can-do-anything:

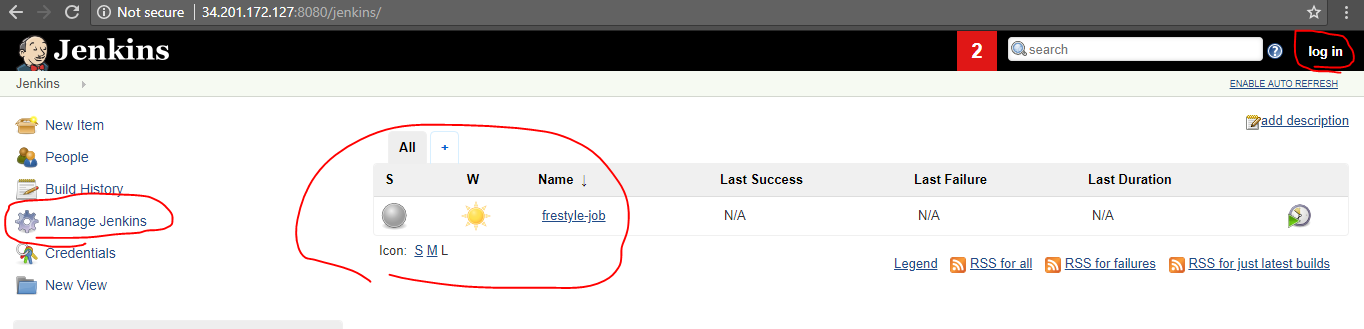


Image-3. a.-Adding-server-Matrix-based security:

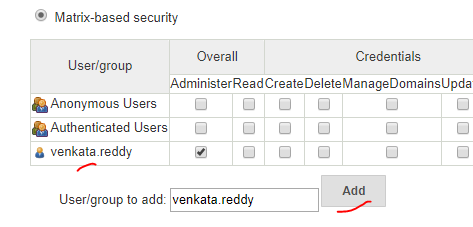


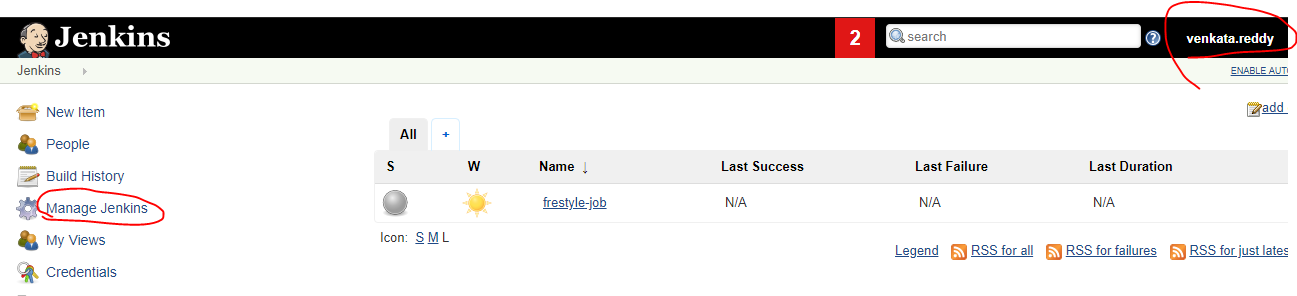
Image-3. a.-Login: 

Image-3.b.i-adding-Anonymous-read-only: Overall read access i.e., accessing Jenkins. Job read access i.e., able to see the Jenkins jobs. Jenkins view read access i.e, able to see the Jenkins views.

Try with different permissions. Un-check the below ‘read’ permission and then save and then logout. Try to access the Jenkins, see what will happen. I will ask you to login.

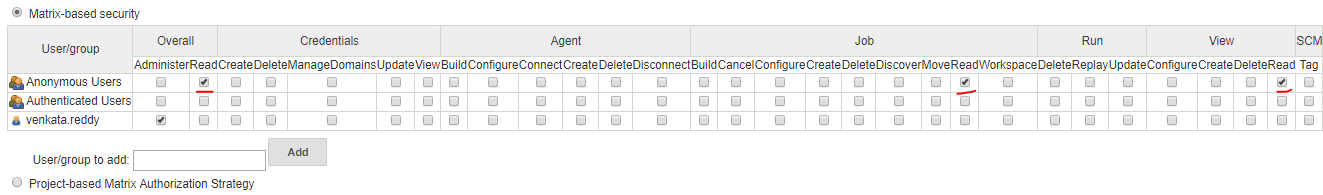


Image-3.b.i-accessing-Jenkins-Anonymous-read-only: After the user accessed the Jenkins URL, without Login the user can able to access Jenkins & user can see the Jenkins job & views.

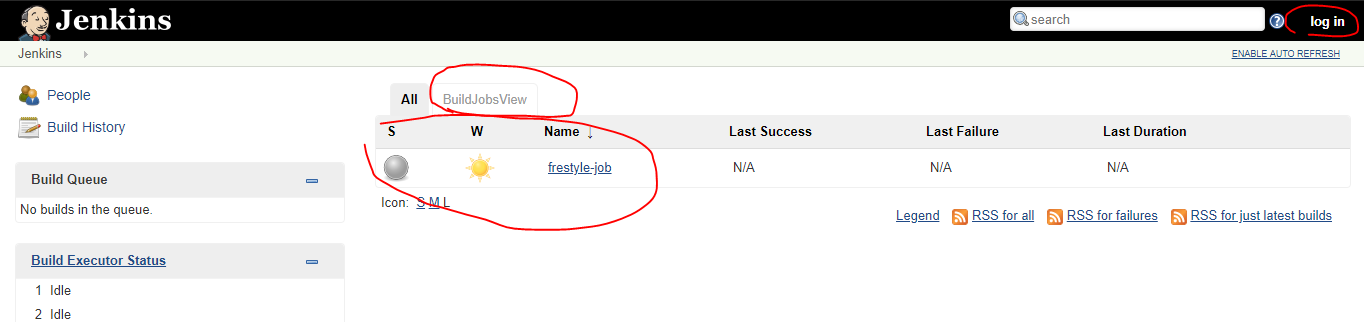


Image-4-Project-Based-Access: There is a Jenkins job, adding user “venkata.sykam” in each Jenkins job lelel (which is already exists in LDAP) & granted ‘read’ only access. Now, login as user “venkata.sykam”.

