Jenkins Role-based Authorization Strategy

The Role-based Authorization Strategy plugin is used to add a new role-based mechanism to manage user's permissions.

We use this plugin;

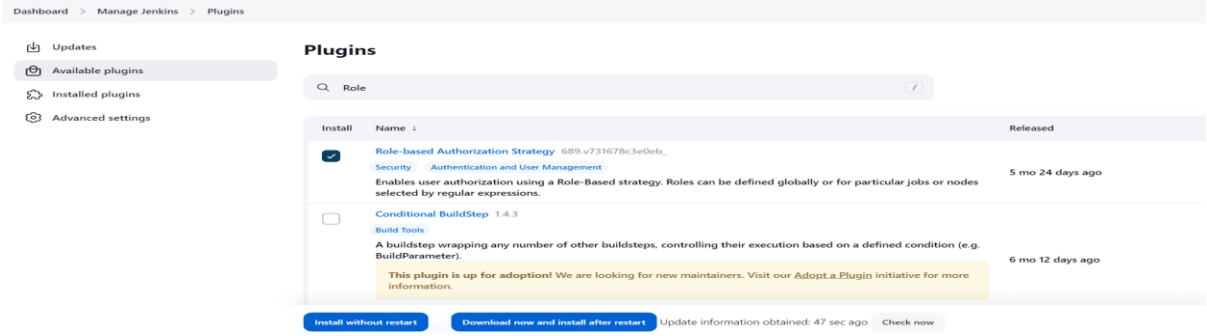
- **Global roles**, such as admin, job creator, anonymous, etc., allowing to set Overall, Agent, Job, Run, View and SCM permissions on a global basis.
- **Project roles**, allowing to set only Job and Run permissions on a project basis.
- **Agent roles**, allowing to set node-related permissions.
- Assigning roles to users and user groups

Installation and Authorization

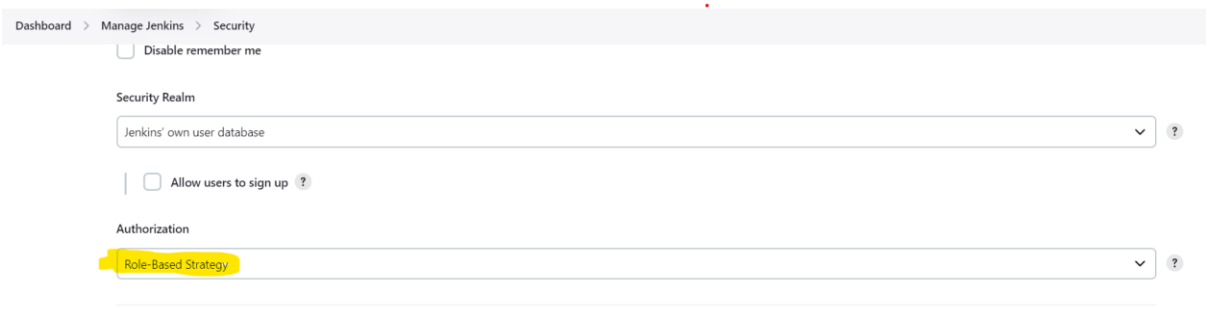
****

First, we need to install the plugin;

Go to **Manage Jenkins** > **Manage Plugins** > **Available** and search for "**Role-based Authorization Strategy**", check the box and click on **Install without restart** button, as shown in screen shot.



After successful installation, go to **Manage Jenkins > Security**, and scroll down to the **Authorization** section and select the **Role-Based Strategy** option, see the below screen shot.

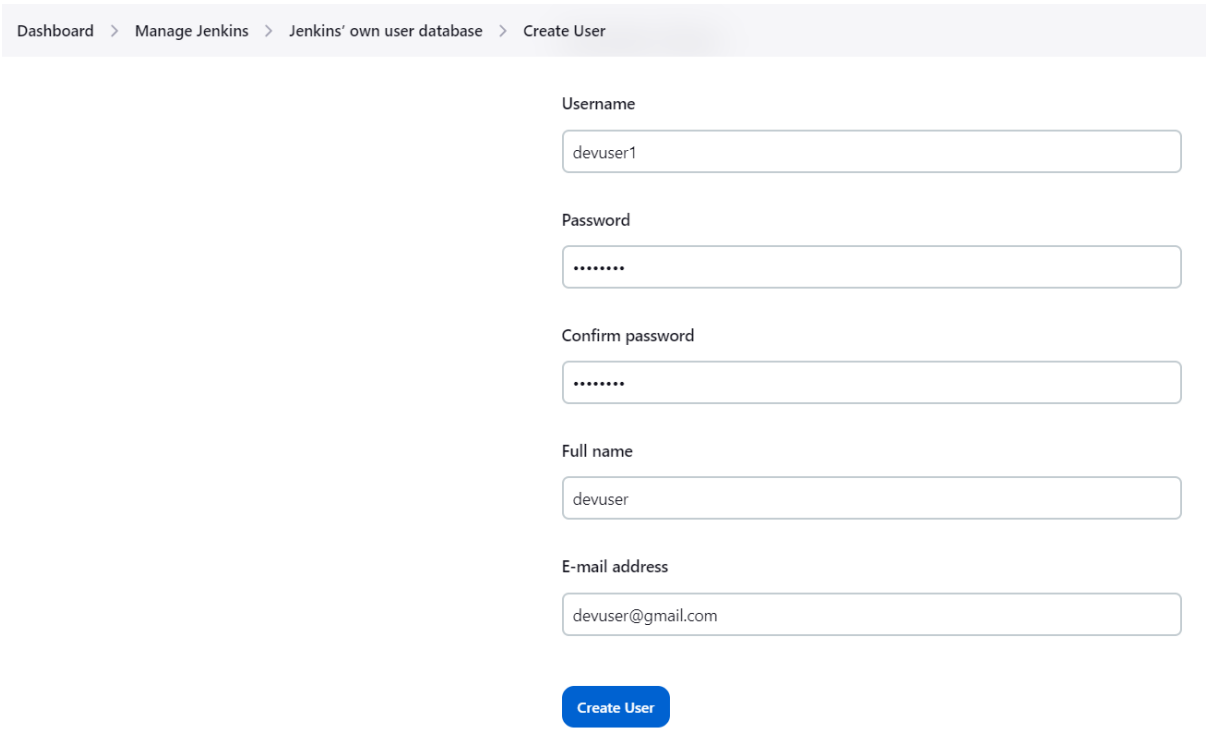


User Creation

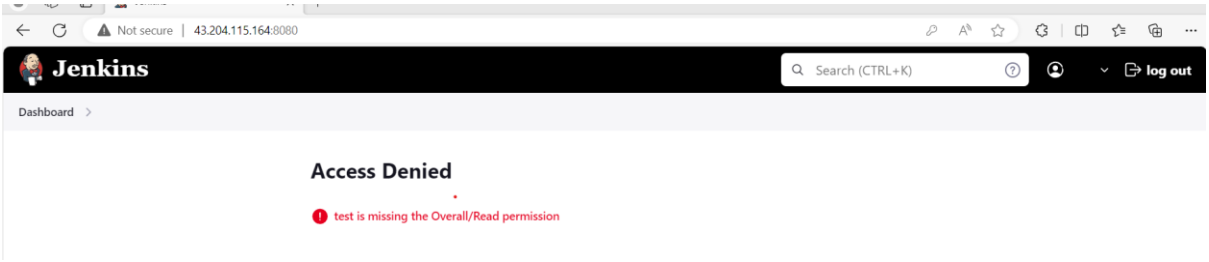
We need to create a user;

Go to **Manage Jenkins > Manage Users > Create Users**

Provide the details as in attached screen shot.



Go to the IP_Address:Port(http://43.204.115.164:8080/), provide the details for the **devuser1** you will see now permissions error.



Manage and Assign Roles:

****

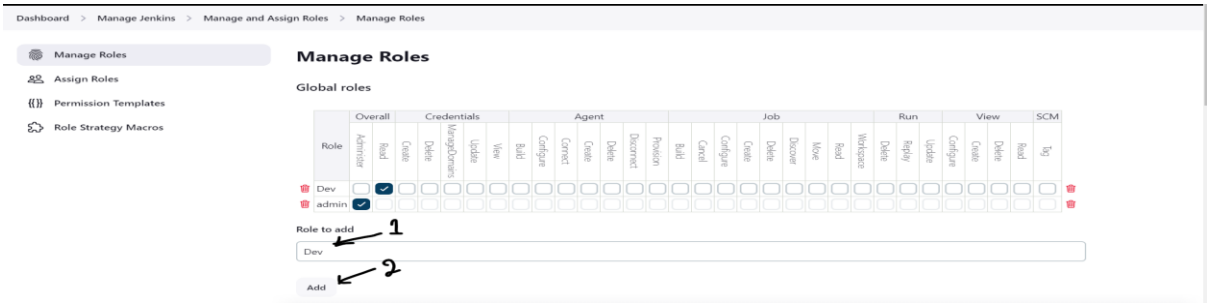
Go to **Manage Jenkins** > Under Security>click on **Manage and Assign Roles** > **Manage Roles**

Note: You won't see the **Manage and Assign Roles** unless you installed the plugin [Role-based Authorization Strategy](#).

Manage Roles

Under the **Manage Roles** page we see two sections, one is **Global roles** and the other is **Item roles**, let's start with **Global roles**;

Under the **Global roles** section add a role with the name **DEV** and assign Read access inside the **overall** category, as shown in the screenshot below;



Item roles:

Under the **Item roles** section add a role and a pattern to the role, for the role to add provide any role that is understandable with the porject in my case I have created **Dev_user** and for pattern provide the RE of the projects to be authorized to this user, as shown in the below screenshot;

Note:

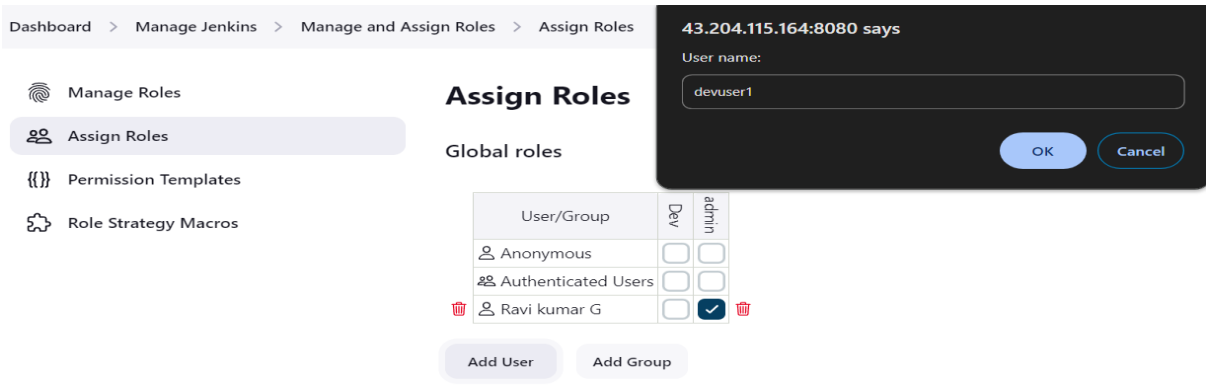
- **Dev_user** is the role to be added and **Dev-.*** is the RE, it means, only **Dev_user** can access all the project that starts with **Dev-**
- **.*** means any combination after **Dev-**.
  We have done with **Mange Roles**, scroll down to the end of the page and click on **Apply** and then **Save.**

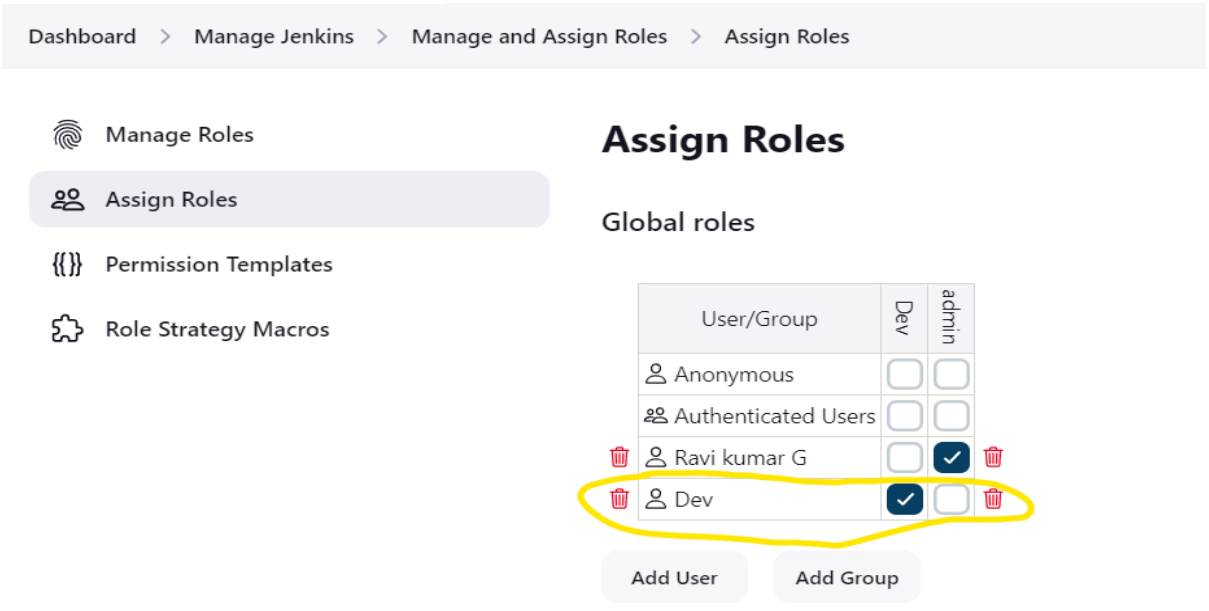
Assign Roles

Again go to **Manage Jenkins > Manage and Assign Roles > Assign Roles** under Under Assign Roles we need to work for Global roles and Item roles, lets do it;

Global role:

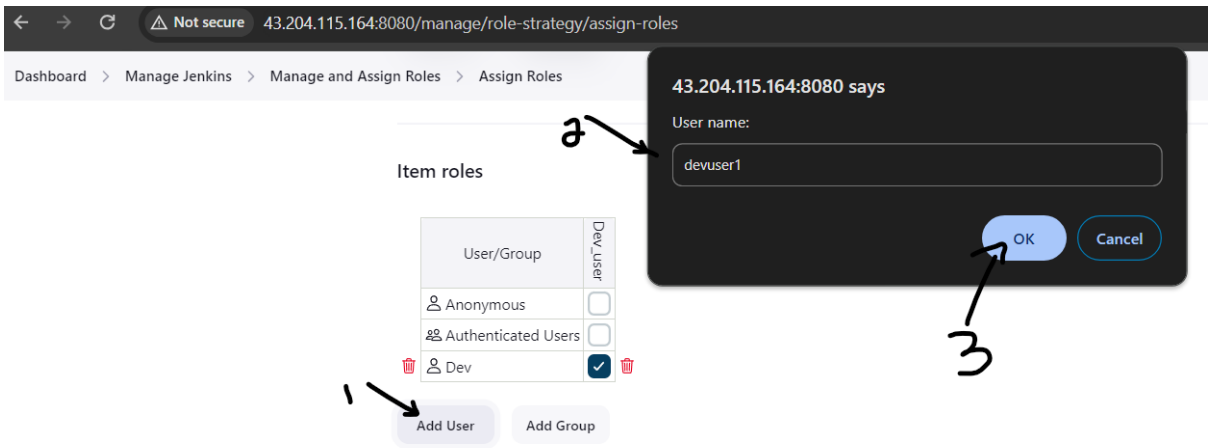
Under **Global roles**, provide the created user name i.e. Devuser1 and click on the **Add** button, user will be added **Global roles** table and select the checkbox under the Employee, please see the screen shot;



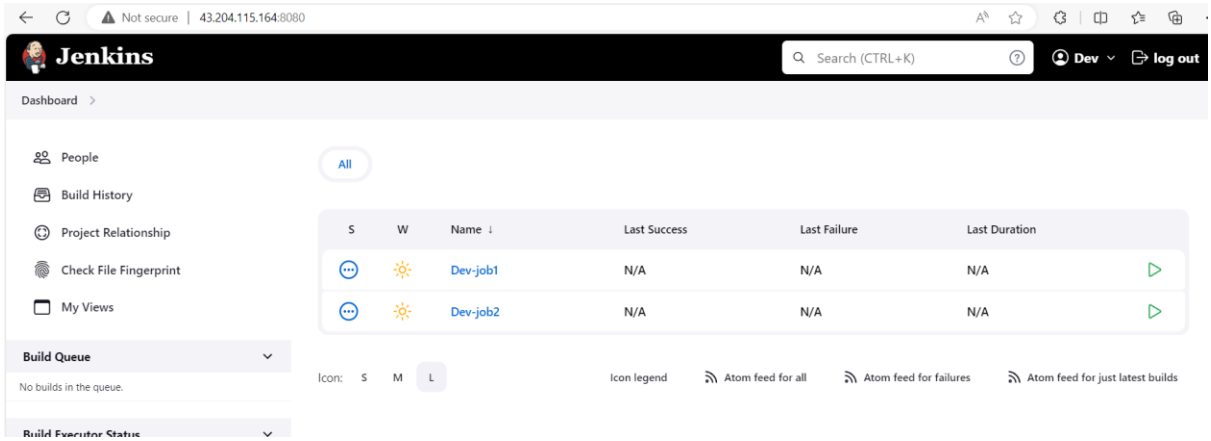


Item roles:

Under **Item roles**, provide the created user name i.e. **devuser1** and click on the **Add** button, user will be added into **Item roles** table and select the checkbox under the Dev_user in front of Dev, please see the screen shot;



Now again go to the IP_Address:Port (http://43.204.115.164:8080/), provide the details for the **devuser1**, you will see now permissions error is resolved and the user can now see all the assign projects.



1.what is RBAC in Jenkins?

RBAC, or Role-Based Access Control, in Jenkins is a way to manage who can do what in Jenkins. It lets administrators assign different roles to users, like "Admin" or "Developer", and each role has its own set of permissions, like "Create Jobs" or "Run Builds".

2.What are the main Features of RBAC?

Features:

Roles: Different roles with different permissions can be created, such as "Admin", "Developer", or "Tester".

Permissions: Each role has specific permissions, like creating jobs, running builds, or configuring Jenkins settings.

User Assignment: Users are assigned to roles based on their responsibilities. For example, a developer might be assigned to the "Developer" role.

Flexible Configuration: RBAC settings can be customized to fit the needs of the organization and its users.

Integration: RBAC can work with different authentication methods, like LDAP or Active Directory, to manage user accounts.

3.Why it's Needed:

Security: RBAC helps keep Jenkins secure by controlling who can access sensitive functions and data.

Organization: It makes it easier to organize users and manage their access based on their roles and responsibilities.

Compliance: RBAC helps organizations comply with regulations and policies by ensuring that access to Jenkins resources is controlled and auditable.

Efficiency: It streamlines user management and reduces the risk of mistakes by providing a centralized way to assign permissions.

In simple terms, RBAC in Jenkins helps keep things organized, secure, and efficient by making sure the right people have the right access to the right tools and information.