# How to enable Role-Based Strategy Plugin

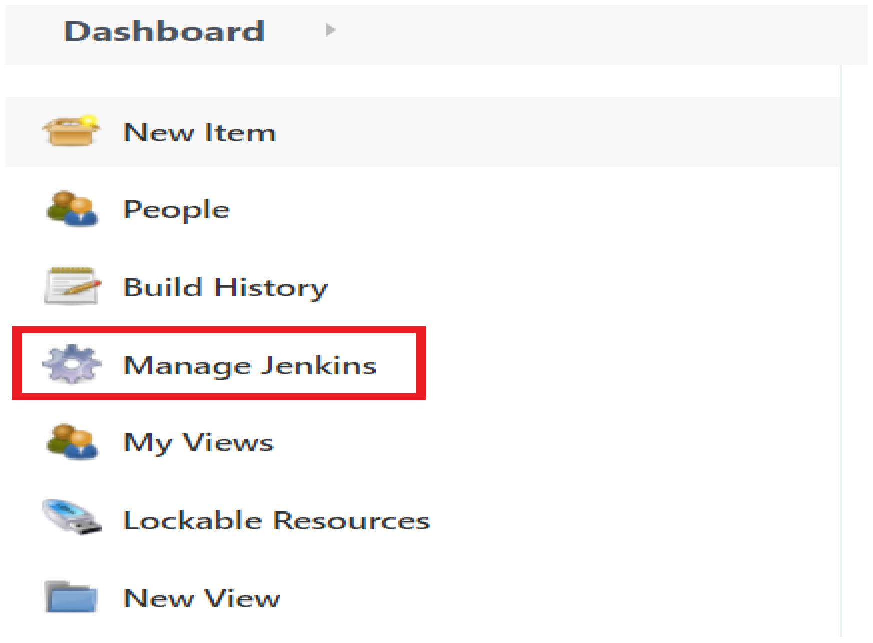
The Role-Based Strategy plugin is used when access rights and privileges need to be assigned to different users based on their role in Jenkins. For example, developers should only have access to development jobs, whereas testers should have access to testing jobs. This document can be divided into the following four sections:

1. Installing Role-Based Authorization Plugin
2. Creating users
3. Managing and assigning roles to users
4. Creating jobs and verifying roles

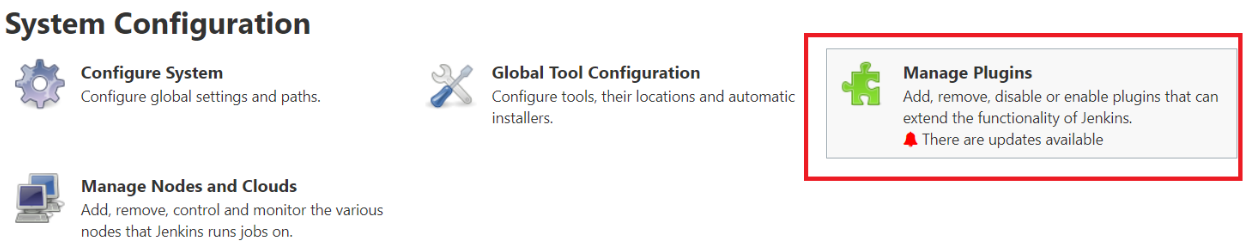
## Installing Role-Based Authorization Plug-in

First, we need to have a Role-Based Authorization Plug-in installed. Now, let's take a look at the various steps involved.

1. Select the **Manage Jenkins** option.



2. Then, select the **Manage Plugins** option.


System Configuration

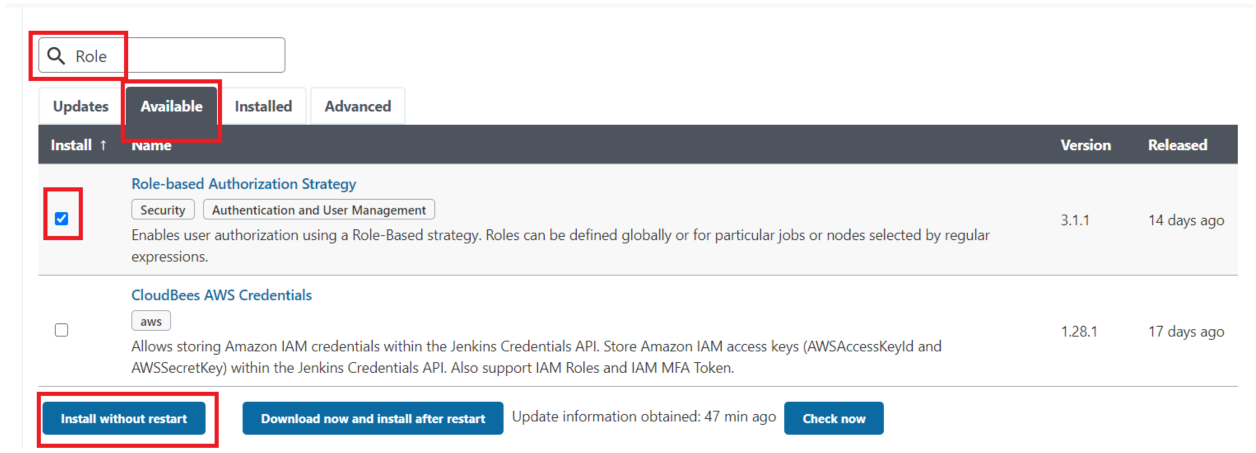
Configure System
Configure global settings and paths.

Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
There are updates available

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

3. Click on the **Available** tab.
4. Next, search for the **Role-Based Authorization Strategy** option.
5. Select it and click on the **Install without restart** option as shown in the screenshot given below.



| Install ↑ | Name | Version | Released |
|---|---|---|---|
| ✓ | Role-based Authorization Strategy<br>Security · Authentication and User Management<br>Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions. | 3.1.1 | 14 days ago |
| | CloudBees AWS Credentials<br>aws<br>Allows storing Amazon IAM credentials within the Jenkins Credentials API. Store Amazon IAM access keys (AWSAccessKeyId and AWSSecretKey) within the Jenkins Credentials API. Also support IAM Roles and IAM MFA Token. | 1.28.1 | 17 days ago |

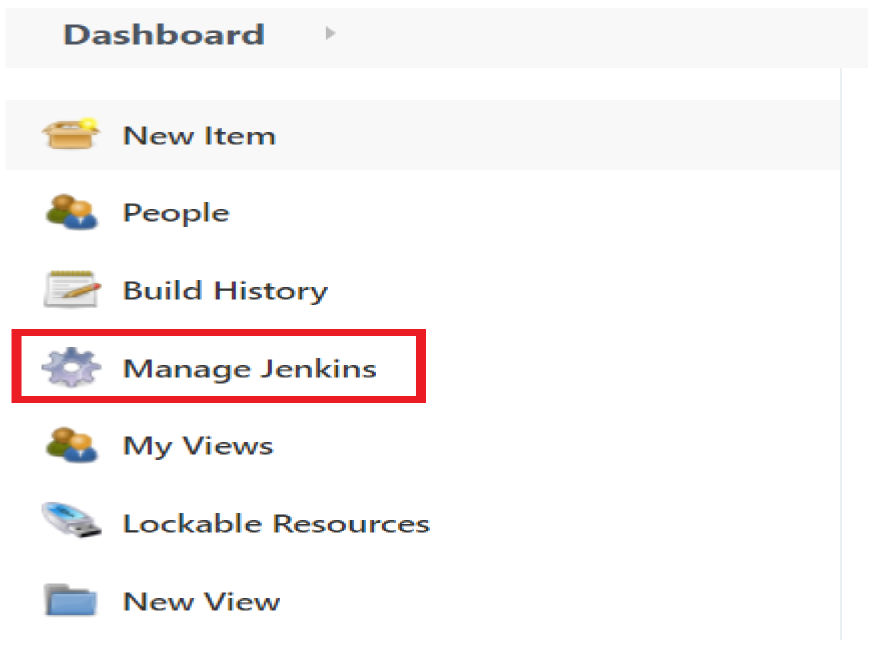
Update information obtained: 47 min ago

6. Once it is installed, go back to the **Jenkins Dashboard**.

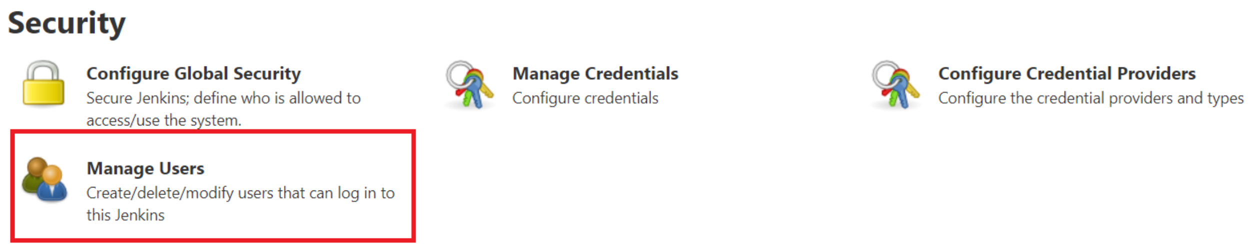
## Creating users

Now, we will create some users before assigning them to various roles. Let's take a look at the different steps involved here.

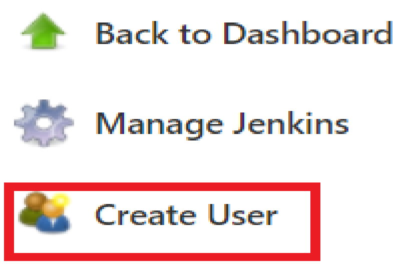
1. Select **the Manage Jenkins** option as shown in the screenshot attached below.



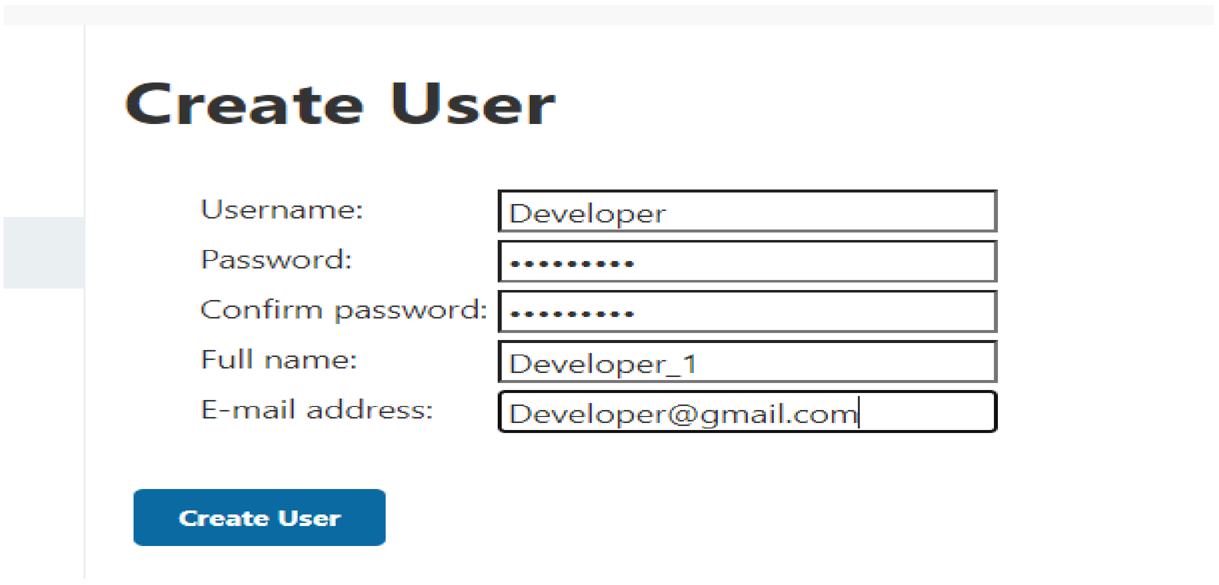
2. Then, click on **Manage Users**.


Security
Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.
Manage Credentials
Configure credentials
Configure Credential Providers
Configure the credential providers and types
Manage Users
Create/delete/modify users that can log in to this Jenkins

3. Click on the **Create User** option from the left side of the window.



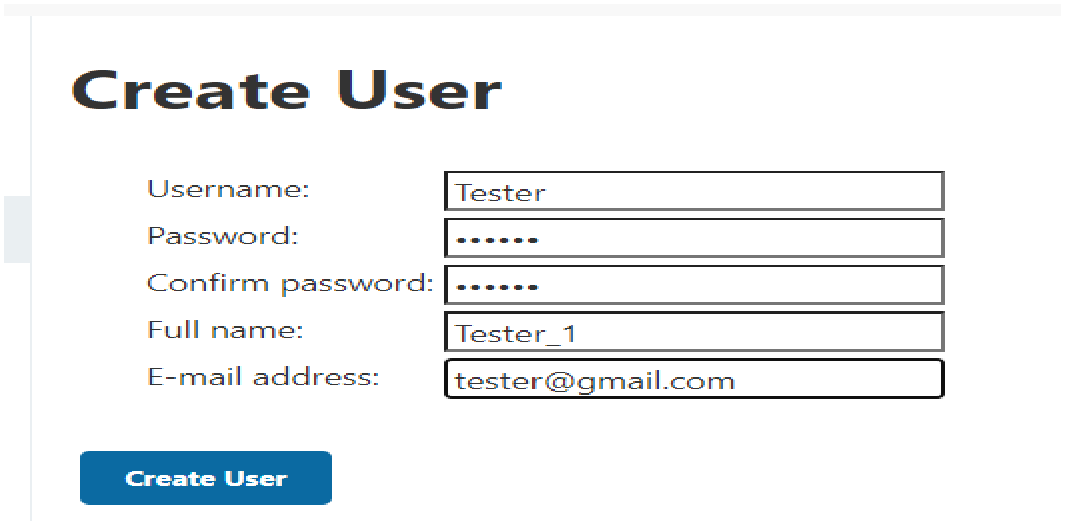
4. The first user that we are creating is **Developer.**
5. So, provide all the details and then click on **Create User.**



Create User

| | |
|---|---|
| Username: | Developer |
| Password: | •••••••••• |
| Confirm password: | •••••••••• |
| Full name: | Developer_1 |
| E-mail address: | Developer@gmail.com |

6. Similarly, create one more user named **Tester**.

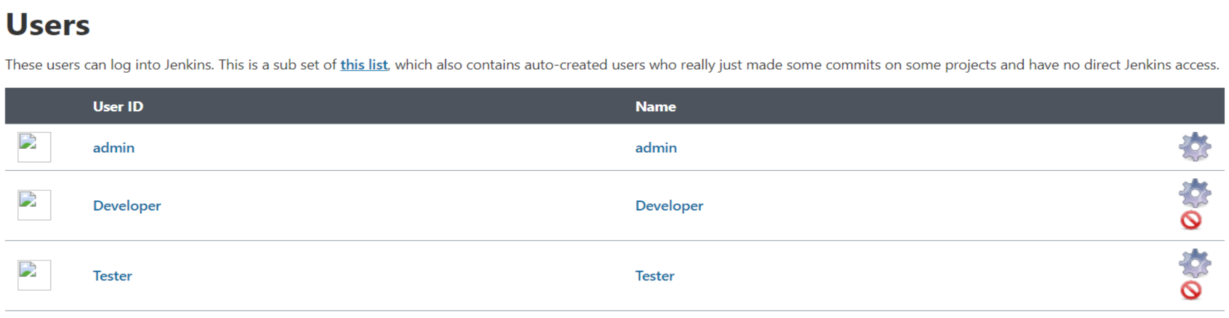


Create User

| | |
|---|---|
| Username: | Tester |
| Password: | •••••• |
| Confirm password: | •••••• |
| Full name: | Tester_1 |
| E-mail address: | tester@gmail.com |

.

7. Next, you can see that we have three users created as shown in the screenshot attached below.



Users

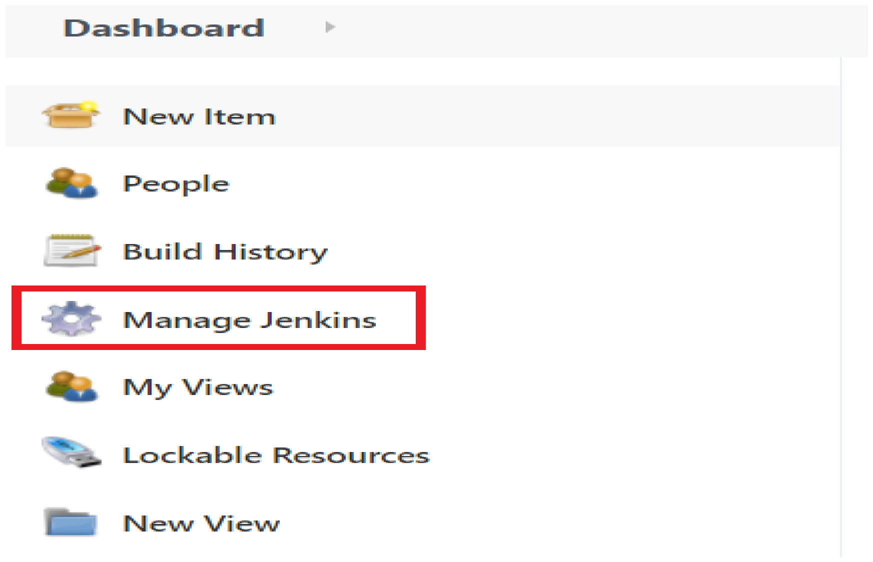
These users can log into Jenkins. This is a sub set of this list, which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.

| | User ID | Name |
|---|---|---|
| | admin | admin |
| | Developer | Developer |
| | Tester | Tester |

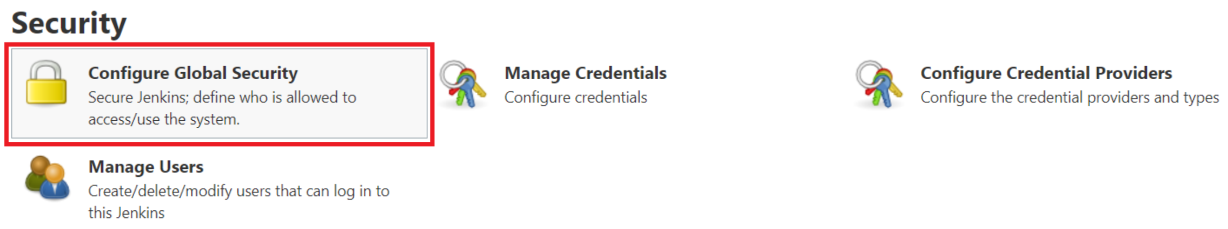
## Managing and assigning roles to users

Now, we will create different roles and assign our users to them. You will learn about the various steps involved here.

1. Now, navigate to the Jenkins dashboard and click on the **Manage Jenkins** option as shown below.



2. Next, select the **Configure Global Security** option.



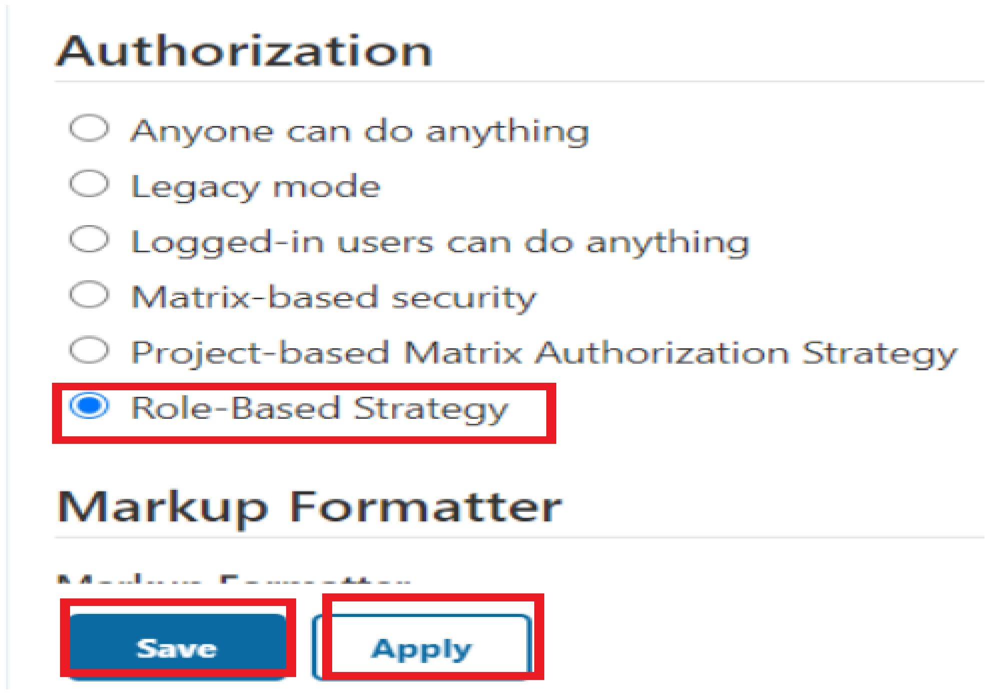
Secure Jenkins; define who is allowed to access/use the system.

Configure credentials

Configure the credential providers and types

Create/delete/modify users that can log in to this Jenkins

3. Then, in **Authorization,** select the **Role-Based Strategy** option.
4. Click on **Apply** and then **Save**.

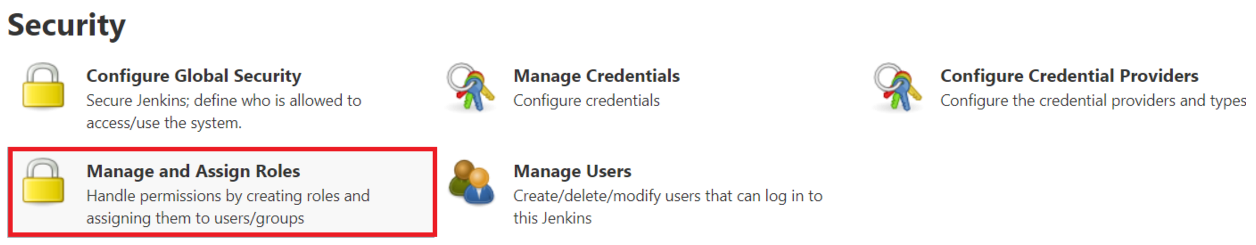


## Authorization

◯ Anyone can do anything
◯ Legacy mode
◯ Logged-in users can do anything
◯ Matrix-based security
◯ Project-based Matrix Authorization Strategy
● Role-Based Strategy

## Markup Formatter

5. Now, navigate to the **Manage Jenkins** page and select the **Manage And Assign Roles** option.



**Security**

**Configure Global Security**
Secure Jenkins; define who is allowed to access/use the system.

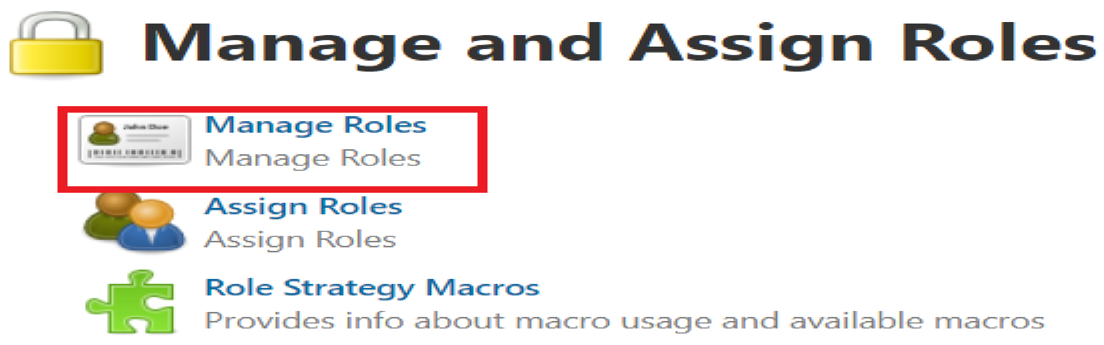
**Manage Credentials**
Configure credentials

**Configure Credential Providers**
Configure the credential providers and types

**Manage and Assign Roles**
Handle permissions by creating roles and assigning them to users/groups

**Manage Users**
Create/delete/modify users that can log in to this Jenkins

6. Select the **Manage Roles** option from here.



**Manage and Assign Roles**

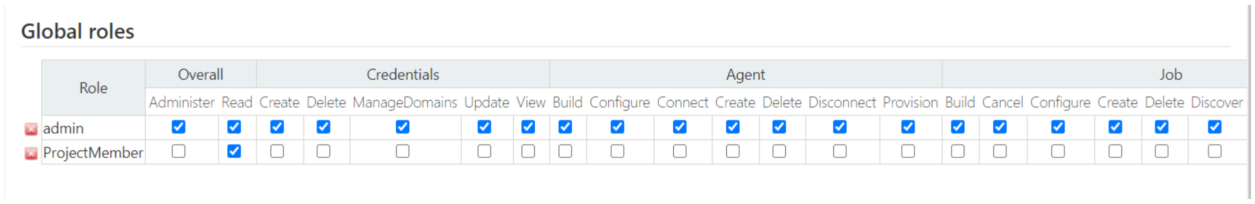
Manage Roles
Manage Roles

Assign Roles
Assign Roles

Role Strategy Macros
Provides info about macro usage and available macros

7. First, we need to add **Global roles.**

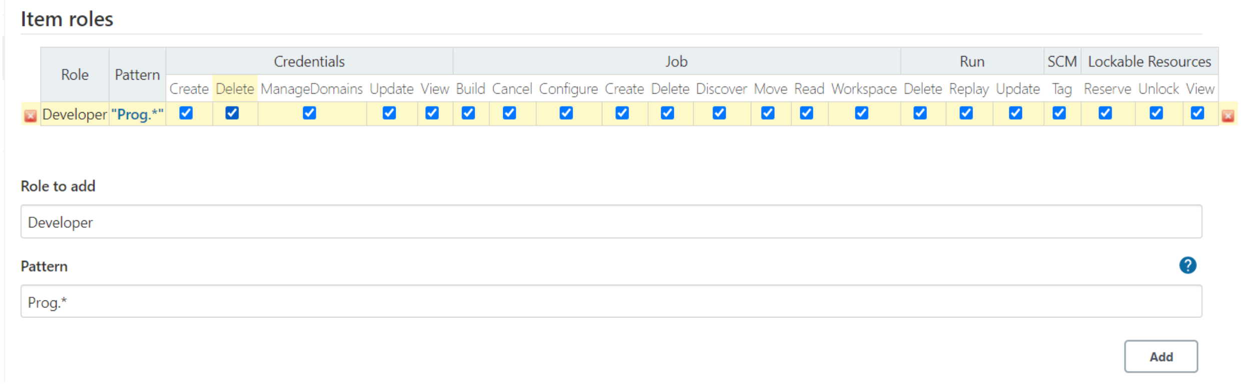
8. Enter the role to add. Here, we are creating one more role named **ProjectMember**. Click on the **Add** option.
9. Next, select the access privileges that you want to provide for the role by checking the correct boxes.



Global roles

| Role | Overall | | Credentials | | | | | Agent | | | | | | | Job | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Administer | Read | Create | Delete | ManageDomains | Update | View | Build | Configure | Connect | Create | Delete | Disconnect | Provision | Build | Cancel | Configure | Create | Delete | Discover |
| admin | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ProjectMember | ☐ | ✓ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |

10. Similarly, more roles can be added.
11. Now, we need to provide **Project roles**.
12. Here, we are creating one role named **Developer** with the pattern **'Prog.*'** that is specified.
13. Now, check all the boxes for these roles as shown in the figure given below.



Item roles

| Role | Pattern | Credentials | | | | | Job | | | | | | | | | Run | | | SCM | Lockable Resources | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Create | Delete | ManageDomains | Update | View | Build | Cancel | Configure | Create | Delete | Discover | Move | Read | Workspace | Delete | Replay | Update | Tag | Reserve | Unlock | View |
| Developer | "Prog.*" | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

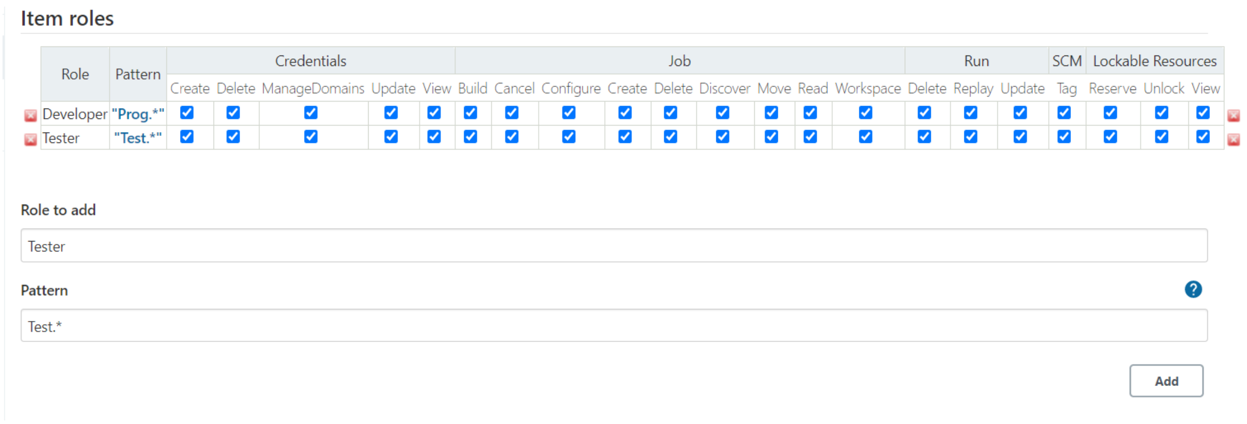
Role to add

Developer

Pattern

Prog.*

14. Similarly, we have created one more role named **Tester** with the pattern **'Test.*'**.



Item roles

| Role | Pattern | Credentials | | | | | Job | | | | | | | | | Run | | | SCM | Lockable Resources | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Create | Delete | ManageDomains | Update | View | Build | Cancel | Configure | Create | Delete | Discover | Move | Read | Workspace | Delete | Replay | Update | Tag | Reserve | Unlock | View |
| Developer | "Prog.*" | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tester | "Test.*" | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

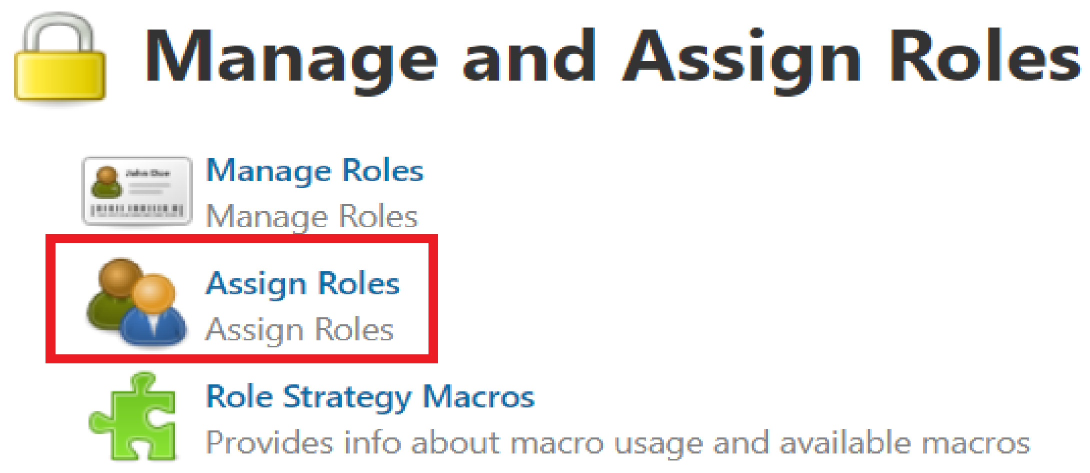
Role to add

Tester

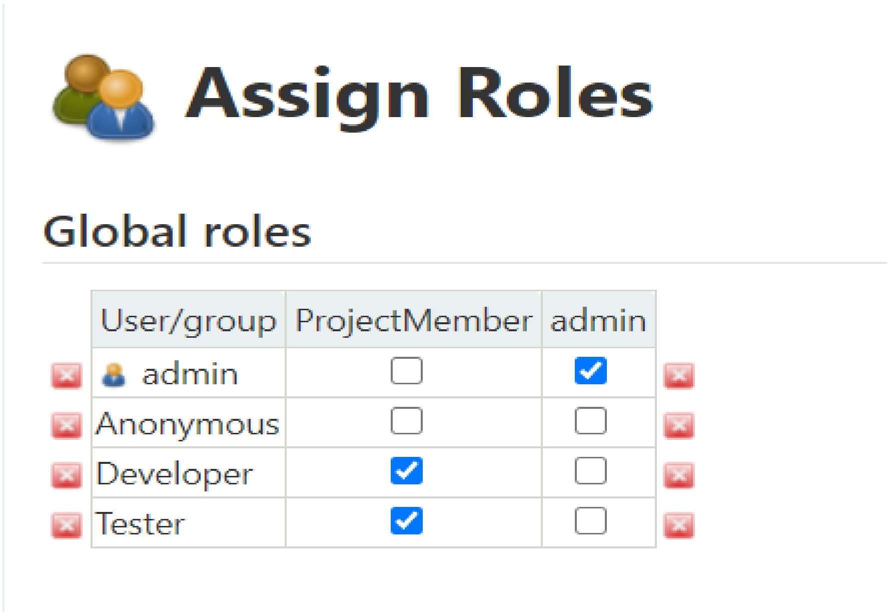
Pattern

Test.*

15. After that, click on **Apply** and then on **Save**.
16. Now, navigate to the **Manage Jenkins** page and select the **Manage and Assign Roles** option.
17. Select the **Assign Roles** option here.



18. Now, we need to add the users in the **Global role** that we created earlier to assign them roles.
19. Enter the **user to add** and click on **Add** option. Here we are adding the user **Developer**.
20. Next, assign the user to the correct role as shown in the screenshot given below. Similarly, we can add other users.

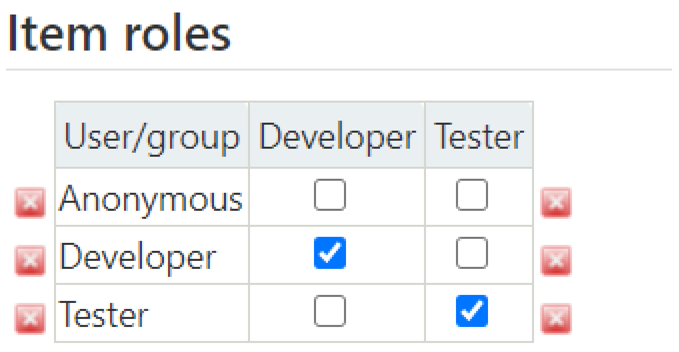


# Assign Roles

## Global roles

| User/group | ProjectMember | admin |
|---|---|---|
| admin | ☐ | ✓ |
| Anonymous | ☐ | ☐ |
| Developer | ✓ | ☐ |
| Tester | ✓ | ☐ |

21. Now, we need to add users in the **Item** roles.
22. Enter the **user to add** and then click on the **Add** option as shown below.
23. Now, assign them to the correct group as shown below.
24. Similarly, we can add other users here.



## Item roles

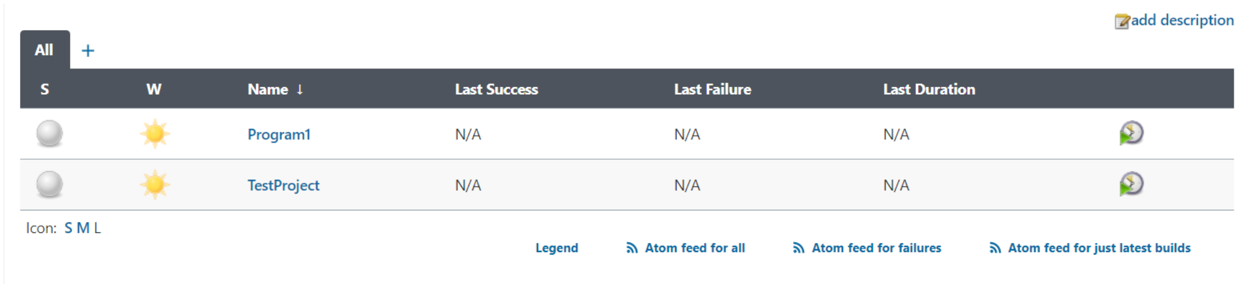
| User/group | Developer | Tester |
|---|---|---|
| Anonymous | ☐ | ☐ |
| Developer | ✓ | ☐ |
| Tester | ☐ | ✓ |

25. Then, click on **Apply** and then **Save**.

## Creating jobs and verifying roles

Now, let's create some jobs and check whether our roles are working as per the pattern.
Now, we will take a look at the steps involved here.

1. Create two jobs named **TestProject** and **Program1**.


add description

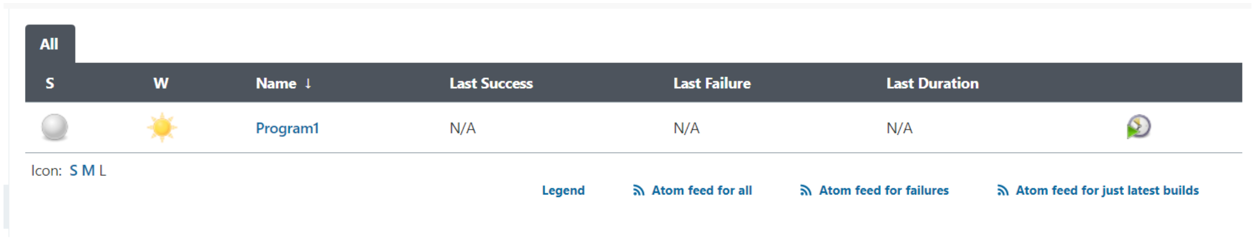
All +

| S | W | Name ↓ | Last Success | Last Failure | Last Duration | |
|---|---|---|---|---|---|---|
| | | Program1 | N/A | N/A | N/A | |
| | | TestProject | N/A | N/A | N/A | |

Icon: S M L

Legend · Atom feed for all · Atom feed for failures · Atom feed for just latest builds

2. Now, log out and then log in as **Developer**.
3. You will notice that only the job **Program1** is visible to us, as we have specified that developers can access only the jobs starting with "**Prog.***"

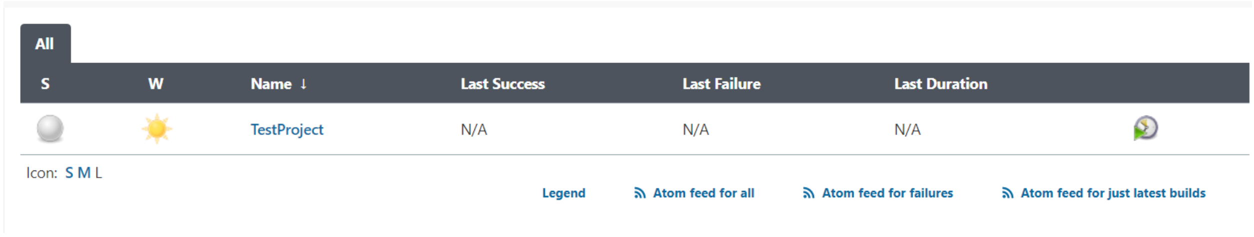

All

| S | W | Name ↓ | Last Success | Last Failure | Last Duration | |
|---|---|---|---|---|---|---|
| | | Program1 | N/A | N/A | N/A | |

Icon: S M L

Legend · Atom feed for all · Atom feed for failures · Atom feed for just latest builds

4. Log out and log in as **Tester**.
5. This time you will only see the TestProject job, as we have specified that the tester can only view the job starting with "**Test.***"
6. The tester can only create and build jobs starting with **Test**.

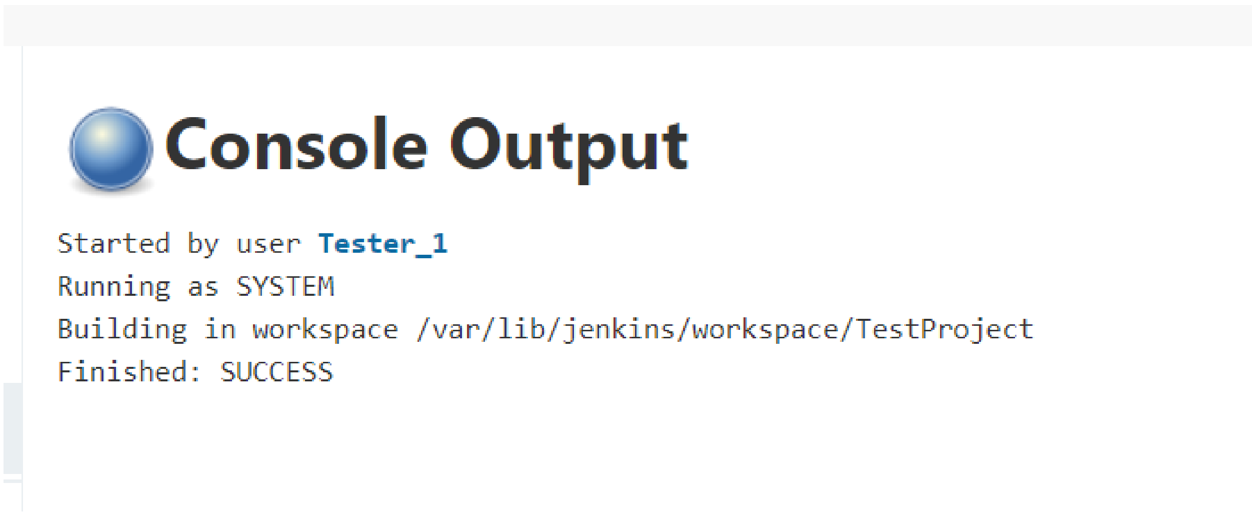

All

| S | W | Name ↓ | Last Success | Last Failure | Last Duration | |
|---|---|---|---|---|---|---|
| | | TestProject | N/A | N/A | N/A | |

Icon: S M L

Legend · Atom feed for all · Atom feed for failures · Atom feed for just latest builds

7. Finally, click on **Build Now** to view the Console Output.



## Console Output

```
Started by user Tester_1
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/TestProject
Finished: SUCCESS
```