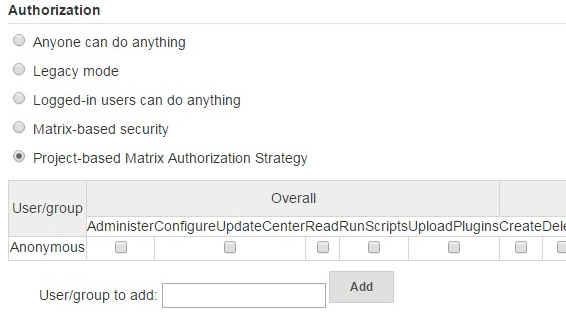
Project-based Matrix Authorization Strategy

Project-based Matrix authorization stragey is an extension to “Matrix-based security”, where you can setup ACL to individual projects. This means that you can allow a specific user to access only certain projects.

The moment you click on the “Project-based Matrix Authorization Strategy” radio-button, it will dispaly the following table right below it where you can add users and assign them individual privileges by clicking on the check-box.



4. Project-based Matrix Authorization Privileges

The following are the various privileges that are available under Project-based Matrix Authorization Strategy.

There are 7 privilege groups. Within each privilege groups, there are individual privleges that you can assign to a specific user or group.

The following explains what these individual privileges means. It is very important for you to understand what exactly each privilege means before assigning it to an user.

**Privilege Group 1: Overall**

* Administer – This permission grants the ability to make system-wide configuration changes, as well as perform highly sensitive operations that amounts to full local system access (within the scope granted by the underlying OS.)
* ConfigureUpdateCenter – This permission allows a user to configure update sites and proxy settings
* Read – The read permission is necessary for viewing almost all pages of Jenkins. This permission is useful when you don’t want unauthenticated users to see Jenkins pages: revoke this permission from the anonymous user, then add “authenticated” pseudo-user and grant the read access.
* RunScripts – Required for running scripts inside the Jenkins process, for example via the Groovy console or Groovy CLI command.
* UploadPlugin – This permission allows a user to upload arbitrary plugins

**Privilege Group 2: Credentials**

* Create – The create permission is necessary to add credentials to a credentials provider.
* Delete – The delete permission is necessary to remove credentials stored in a credentials provider.
* Manage Domains – The manage domains permission is necessary to add/remove/configure the credential domains of a credentials provider (where the credentials provider supports multiple credential domains).
* Update – The update permission is necessary to modify credentials in a credentials provider.
* View – The view permission is necessary to view the credentials stored in a credentials provider.

**Privilege Group 3: Agent**

* Build – This permission allows users to run jobs as them on agents.
* Configure – This permission allows users to configure agents.
* Connect – This permission allows users to connect agents or mark agents as online.
* Create – This permission allows users to create agents.
* Delete – This permission allows users to delete existing agents.
* Disconnect – This permission allows users to disconnect agents or mark agents as temporarily offline.

**Privilege Group 4: Job**

* Build – This permission grants the ability to start a new build.
* Cancel – This permission grants the ability to cancel a scheduled, or abort a running, build.
* Configure – Change the configuration of a job.
* Create – Create a new job.
* Delete – Delete a job.
* Discover – This permission grants discover access to jobs. Lower than read permissions, it allows you to redirect anonymous users to the login page when they try to access a job url. Without it they would get a 404 error and wouldnt be able to discover project names.
* Read – See a job. (You may deny this permission but allow Discover to force an anonymous user to log in to see the job.)
* Workspace – This permission grants the ability to retrieve the contents of a workspace Jenkins checked out for performing builds. If you don’t want a user to access files in the workspace (e.g. source code checked out from SCM or intermediate build results) through the workspace browser, you can revoke this permission.

**Privilege Group 5: Run**

* Delete – This permission allows users to manually delete specific builds from the build history.
* Replay – Ability to perform a new Pipeline build with an edited script.
* Update – This permission allows users to update description and other properties of a build, for example to leave notes about the cause of a build failure

**Privilege Group 6: View**

* Configure – This permission allows users to change the configuration of views.
* Create – This permission allows users to create new views.
* Delete – This permission allows users to delete existing views.
* Read – This permission allows users to see views (implied by generic read access).

**Privilege Group 7: SCM**

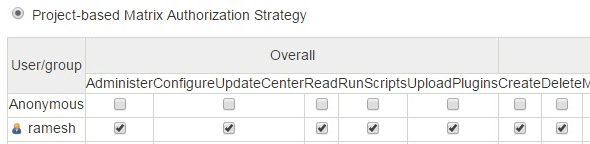
* Tag – This permission allows users to create a new tag in the source code repository for a given build.

5. Assign everything for the Admin user

This is important. If you don’t do this, you’ll get yourself (the admin user) locked out of the system when you click on “Save” in this Security screen. Also, if you don’t do this, you’ll get “user is missing the Overall/Read permission” error message when you try to click on any page even as admin user.

In the “User/group to add”, type the admin username. I.e The admin username that you created during the Jenkins intial setup, and click on Add.

This will add the admin user (for example: ramesh) to the “Project-based matrix authorization” table. From here, click on the all the individual check-boxes. This means that I’m assigning all the privileges to the username ramesh, who is the admin user.



After assigning those permissions, click on the “Save” button to save the changes.

Note: By mistake if you’ve clicked on “save” without giving admin user full-permission and got yourself locked-out of Jenkins, please leave a comment, and I’ll let you know to fix that issue.

6. Create Jenkins User

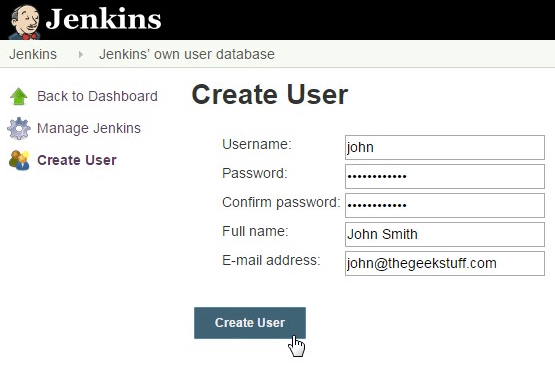
Next, if you want to assign certain privilege to individual users, follow the steps below. For example, let us create user account for a developer (john smith) who should have appropriate privileges to build a Jenkins job.

In this case, we should first create a username for John in Jenkins.

Click on the “Manage Jenkins” link on the left-side, and then click on “Manage Users”.

This will take you to a page that will list all available users. Click on “Create User” link on the left side from this users page.

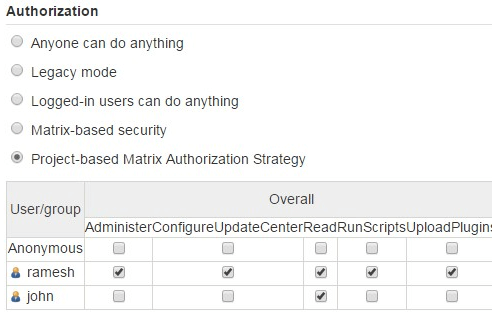
From here, create an user account for John by filling-out the following details.



7. Add User to Project-based Matrix Authorization

Go to: Jenkins -> Manage Jenkins -> Configure Global Security Option -> Under the “Authorization” section -> “Project-based Matrix Authorization Strategy” should already be selected -> You should already see the admin user here with all the privileges.

In the “User/group to add”, type “john” and click on “Add”, this will add john to the privileges table as shown below.



From here, select the privileges that you want to assign to John. If you are creating an account for a developer who will configure and run a Jenkins job, I recommend that you assign the following privileges. In this example, for John, I’ve assigned the following privileges.

* Overall – Read
* Job – Build, Configure and Read

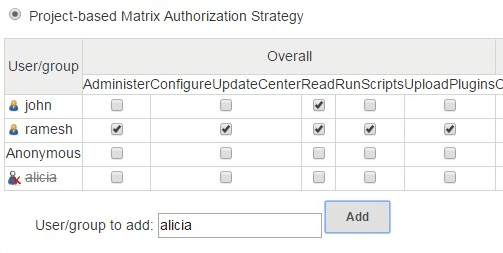
Again, note how I’ve assigned ALL privileges to user ramesh (who is the admin user), and then I assigned only specific privileges to John.

After this, click on Save to save this new configuration.

8. Assigning Privileges to non-existing User

Also, if you try to add a user who doesn’t exist to the Matrix authorization table, you’ll notice that the username will be added, but it will be crossed-out as shown in the screenshot below. In this example, we don’t have username alicia created in the system yet.

Jenkins will still let you assign privilege to an user who doesn’t exist yet. You just have to go the Manage User section and add this user later.



9. Login with Developer Account

When John login to Jenkins, he’ll not see “Manage Jenkins” or Credential options, as those are only for administrator.

If administrator has already created a job, then john will be able to run that job (build and deploy), and make changes to that job configuration.

