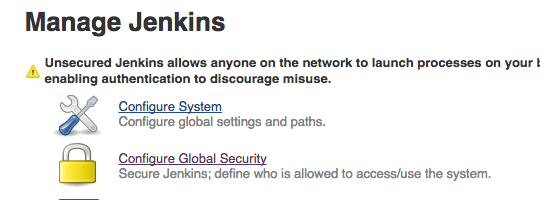
# Role-based Access Control for CloudBees Jenkins Operations Center

Goal

This lab will cover how to use the CloudBees Role-Based Access Control plugin with CloudBees Jenkins Operations Center.

Step 1. Configuring client master and centralized security

The first step is to enable security on the CloudBees Jenkins Operations Center master, so click on the “Manage Jenkins” link in the left-hand menu and then on the “Configure Global Security” main menu option.



First click on the “Enable Security” checkbox, then select the “Mock Security Realm” as the security realm and “Role-based Matrix Authorization Strategy” as the authorization policy.

For the Mock Security Realm authentication option, we will create the following users with the following text:

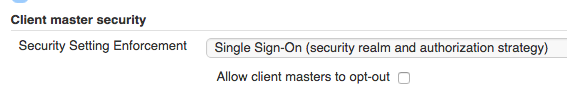
harry admin-ext

sally developer-ext

barry

Where the first column represents the username and the right column represents the group name, and both represent imported fields from an external authentication server like LDAP or Active Directory. Doing this will create 3 users with those usernames in Jenkins and Jenkins will recognize the admin and developer external groups and the users’ membership in them.

Now scroll down and enable client master security enforcement by picking the “Single Sign-On (security realm and authorization strategy)” option from the drop-down menu. This option will allow the 3 mock “imported” users to sign in to the managed client master with their same logins, and will enforce your configured security settings on the client master.



Now save your security settings by clicking the “Save” button and return to the top-level of Jenkins Operations Center. You will now see a “log in” link in the top right-hand corner, so click on that and log in as “harry” with the password “harry”.

Step 2. Creating roles

Now we will need to create roles for our mock imported users. From the top-level screen, click on the “Roles” link in the left-hand menu. This link is now available because you selected the “Role-based Matrix Authorization Strategy” option in the last section.

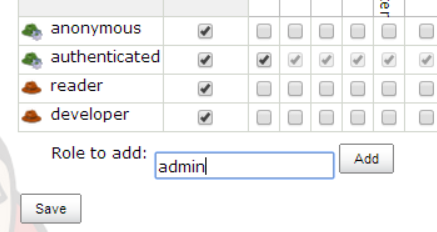
Now click on the “Manage” link in the left-hand menu and you will be taken to the “Manage Roles” screen.

Our goal here is to create a security set up that has the following characteristics:

* Anonymous users have no access whatsoever, and they need to login first before even seeing the CloudBees Jenkins Operations Center top page.
* Once logged in, users have generic read access.
* Jobs can assign some users to the “reader” role that has read-only access to jobs, their test results, build results, etc.
* Jobs can assign some users to the “developer” role, who can start a new build, configure jobs, and so on.
* A few people will be in the “admin” role, which gets irrevocable full access to the whole of Jenkins.

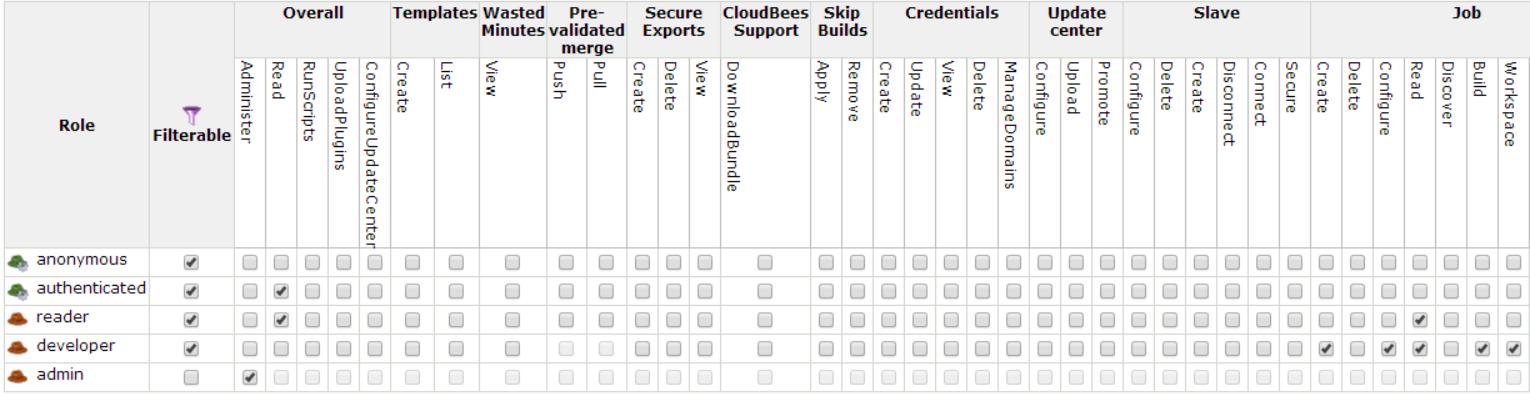
This maps **to two pre-defined system roles** (“anonymous” and “authenticated”) and **three additional roles we create** (reader, developer and admin).

To create the additional roles, type in the their titles to the “Role to add” field on the “Manage Roles” page and then hit the add button:



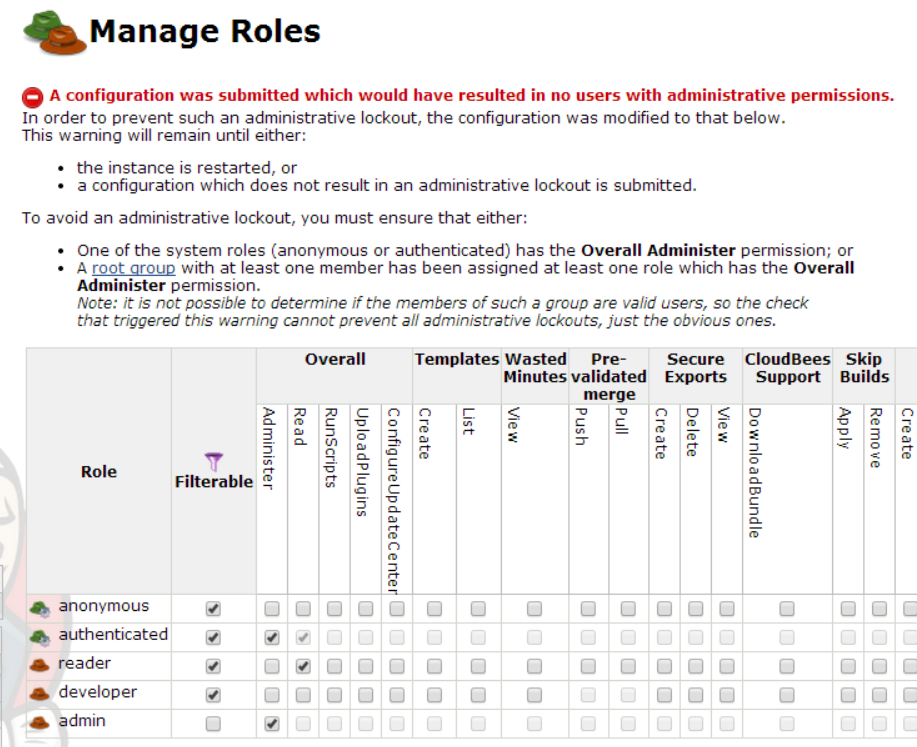
By default, “filterable” will be set to true for them all. As such, you’ll now need to edit each role’s permissions\*\*. Note that the first permissions column shows the name of the ***section*** the permission is for, while the second column shows the name of the ***checkbox*** in that section:

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Permissions (section, checkbox)** | | **Filterable** |
| Anonymous | None | None | Yes |
| Authenticated | Overall | Read | Yes |
| Reader | Overall  Job | Read  Read | Yes |
| Developer | Job  Job  Job  Job  Job | Read  Create  Configure  Build  Workspace | Yes |
| Administer | Overall | Administer | No |

Once you are ready to edit the “authenticated role”, note that you’ll be able to uncheck all boxes at once by scrolling to the right and clicking on the JEBC Screenshots:unchck.PNGbutton.

Now try to save your configuration.

Note that the “Anti-lockout” guard will kick in and change our configuration, giving “authenticated” overall Administer permissions:



This is because the “admin” role is not yet associated with a group that contains at least one member. As such, saving the configuration as-is would have locked you out of the instance.

Now let’s save these altered roles by clicking the “save” button at the bottom of the screen.

Step 3. Creating groups

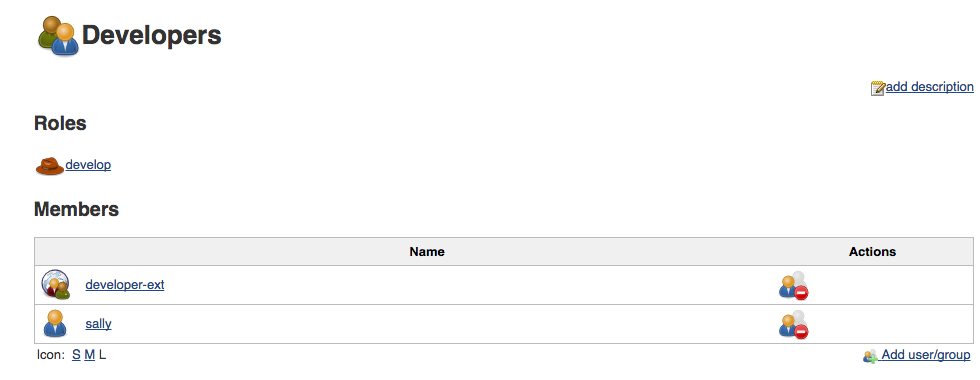
The next step is to create groups that connect users to the roles, and populate these groups.

Click on the “Groups” link in the left-hand menu, which will take you to the “Groups” list. We will create 3 groups here: “Developers”, “Administrators”, and “Browsers”.

Click on the “add some?” link and start creating these three groups by populating the following . We want to create groups to represent each role we have created, and we also want to add both our imported users *and* our imported groups as members of each group. The final configuration should look like this:

|  |  |  |  |
| --- | --- | --- | --- |
| **Group Name** | **Roles** | **Propagated** | **Members** |
| Browsers | reader | Yes | barry, harry, sally |
| Developers | developer | Yes | sally, developer-ext |
| Administrators | administer | Yes | harry, admin-ext |

For example, “Developer” is configured like this:



Repeat this process until all 3 groups have been created by hitting the “Back to groups” link in the left-hand menu and then hitting the “New Group” link in the same menu while on the “Groups” page.



Step 4. Completing configuration of the roles

Now that we have a group with the Overall/Administer permission, we can complete the configuration of the roles.

Go back to the “Manage Roles” screen (“Manage Jenkins” -> “Manage Roles”). De-select the Overall/Admin permission for the “authenticated” role and ensure that the Overall/Read permission is still on selected.

Now save the configuration. Since the “administer” role is now associated with a populated group, you should not receive an “Anti-lockout” warning.

Step 5. Single sign-on

Now that security settings have been created and enabled, try logging into the client master Jenkins instance as different users and clicking through breadcrumbs back to the CloudBees Jenkins Operations Center master. You will see that one sign on to either master will grant you access to both with no re-sign in required.