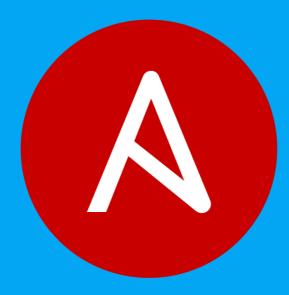Ansible Workshop - Exercises

Automation Platform

Learn to manage and run your Ansible content in AAP.

5 - Role-based access control

Objective

You have already learned how Ansible automation controller separates credentials from users. Another advantage of Ansible automation controller is the user and group rights management. This exercise demonstrates Role Based Access Control (RBAC)

Guide

Ansible automation controller users

There are three types of automation controller users:

- **Normal User**: Have read and write access limited to the inventory and projects for which that user has been granted the appropriate roles and privileges.

- **System Auditor**: Auditors implicitly inherit the read-only capability for all objects within the automation controller environment.

- **System Administrator**: Has admin, read, and write privileges over the entire automation controller installation.

Let's create a user:

- In the automation controller menu under **Access** click **Users**
- Click the **Add** button
- Fill in the values for the new user:

Parameter	Value
Username	`wweb`
Email	`wweb@example.com`
Password	`ansible`
Confirm Password	`ansible`
First Name	`Werner`
Last Name	`Web`
Organization	`Default`

- Click **Save**

Ansible automation controller teams

A Team is a subdivision of an organization with associated users, projects, credentials, and permissions. Teams provide a means to implement role-based access control schemes and delegate responsibilities across organizations. For instance, permissions may be granted to a whole Team rather than each user on the Team.

Create a Team:

- In the menu go to **Access → Teams**
- Click the **Add** button and create a team named `Web Content` within the `Default` Organization.
- Click **Save**

Add a user to the team:

- Click on the team `Web Content` and click the **Access** tab and click **Add**.
- Within the **Select a Resource Type** window, click on the **Users** resource type and click **Next**.
- Within the **Select Items from List**, select the checkbox next to the `wweb` user and click **Next**.
- Within the **Select Roles to Apply**, select **Member** as the role to apply to the `wweb` user.

Click **Save**.

Permissions allow to read, modify, and administer projects, inventories, and other automation controller elements. Permissions can be set for different resources.

Granting permissions

To allow users or teams to actually do something, you have to set permissions. The user **wweb** should only be allowed to modify content of the assigned webserver.

Add the permission to use the `Create index.html` template:

- Within **Resources** -> **Templates**, select `Create index.html`.
- Select **Access** tab from the menu and click **Add**.
- Within the **Select a Resource Type** window, click on the **Users** resource type and click **Next**.
- Within the **Select Items from List**, select the checkbox next to the `wweb` user and click **Next**.
- Within the **Select Roles to Apply**, select **Read** and **Execute** as the roles to apply to the `wweb` user.
- Click **Save**

Test permissions

Now log out of automation controller's web UI and in again as the **wweb** user.

- Go to the **Templates** view, you should notice for wweb only the `Create index.html` template is listed. He is allowed to view and launch, but not to edit the Template (no Edit button available).

- Run the Job Template by clicking the rocket icon. Enter the values for the survey questions and launch the job.

- In the following **Jobs** view have a good look around, note that there where changes to the host (as expected).

Check the result: execute `curl` again on the control host to pull the content of the webserver on `node1` (you could of course check `node2` and `node3`, too):

```
[student@ansible-1 ~]$ curl http://node1
```

Just recall what you have just done: You enabled a restricted user to run an Ansible playbook

- Without having access to the credentials

- Without being able to change the playbook itself

- But with the ability to change variables you predefined!

Effectively you provided the power to execute automation to another user without handing out your credentials or giving the user the ability to change the automation code. And yet, at the same time the user can still modify things based on the surveys you created.

This capability is one of the main strengths of Ansible automation controller!