1. **Project creation in Ansible Tower**

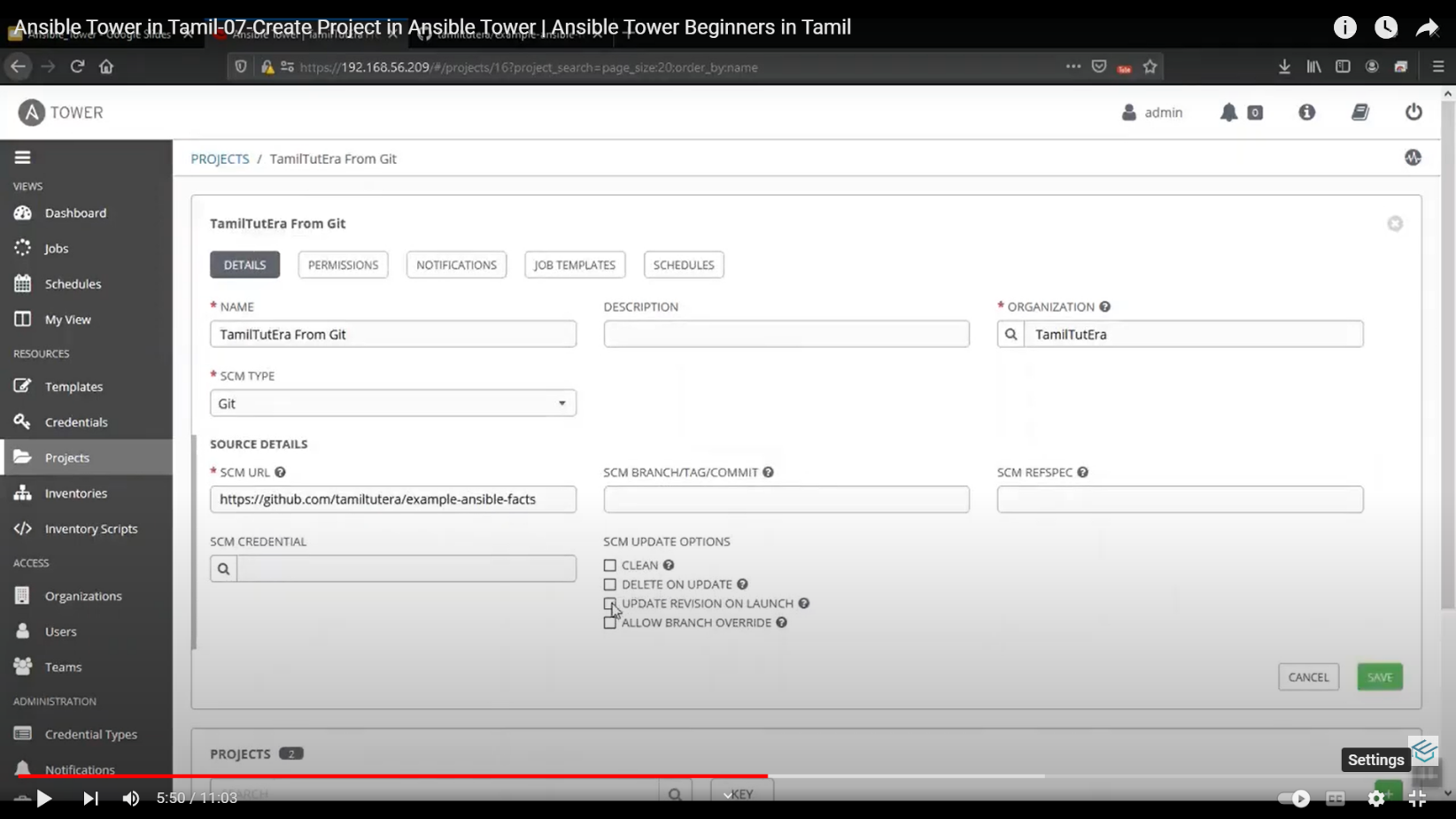
**Projects –** Logical collection is Ansible Playbooks.

By default, the Project base path: /var/lib/awx/projects

but this can be modified by Tower admin. It is configured in /etc/tower/settings.py

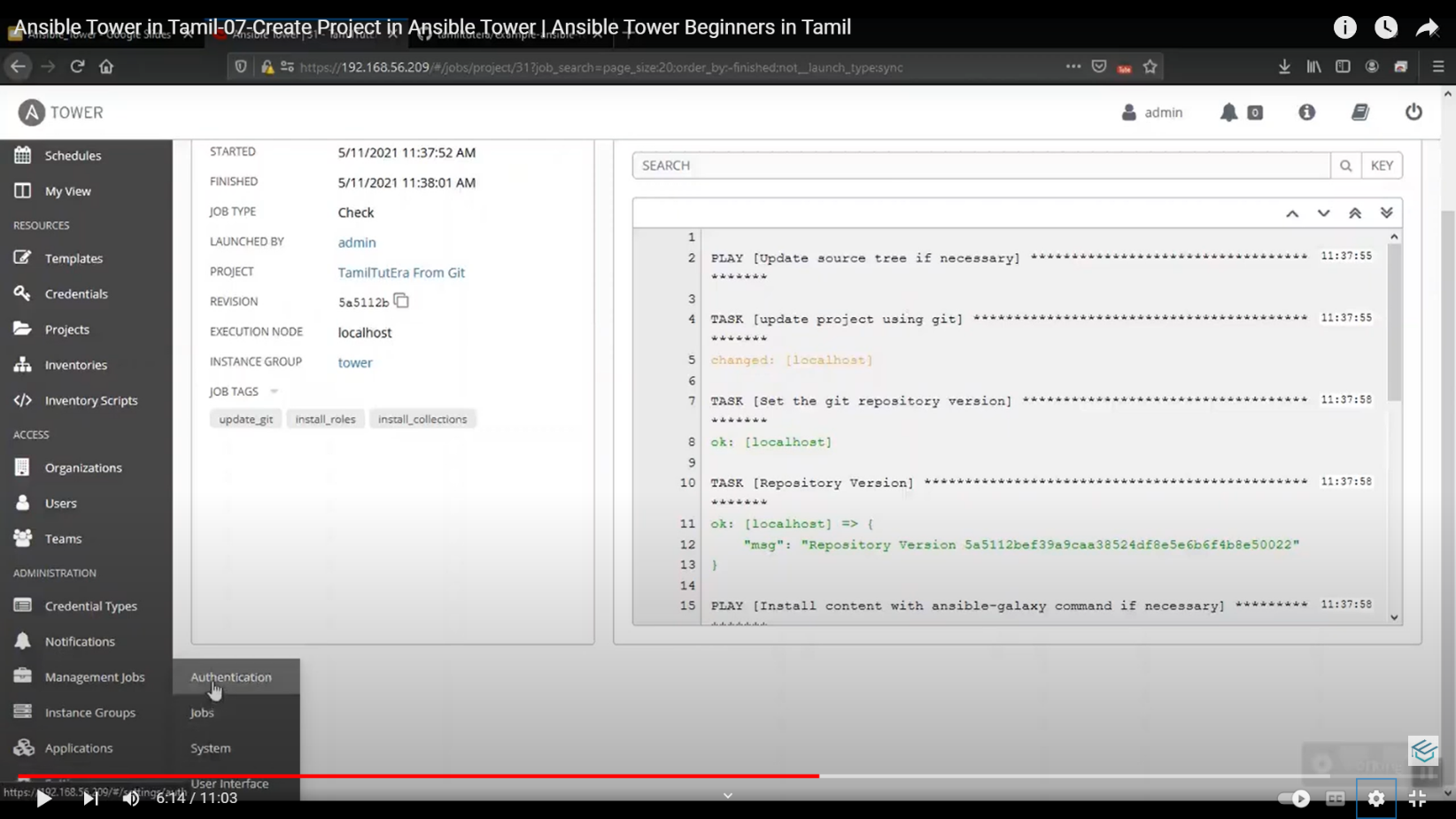
In Ansible tower Dashboard 🡪 select project option 🡪 And click (+) at top right to create a new project

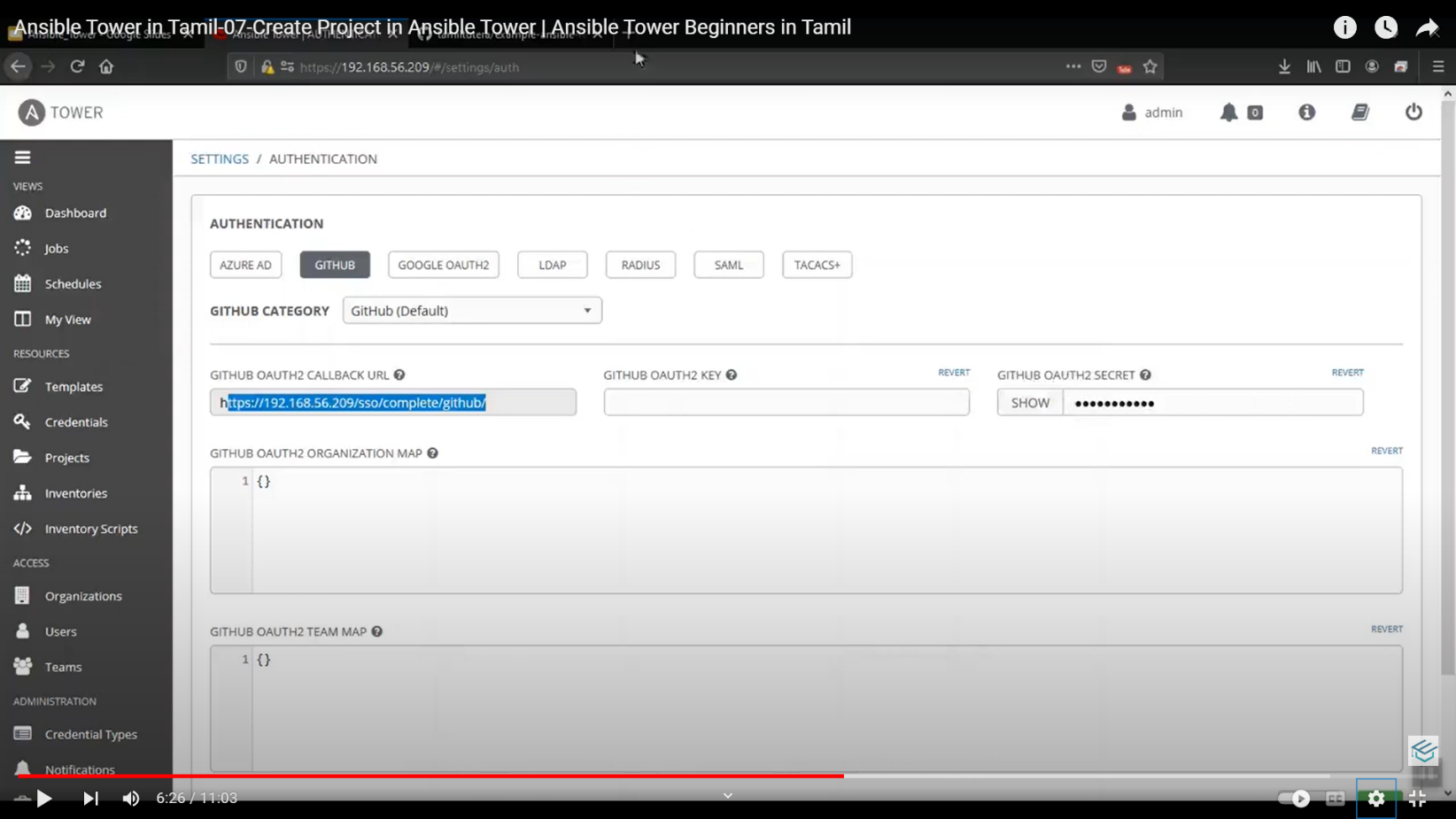
Give Project name, Organization, SCM type (GIT). Then, provide SCM URL

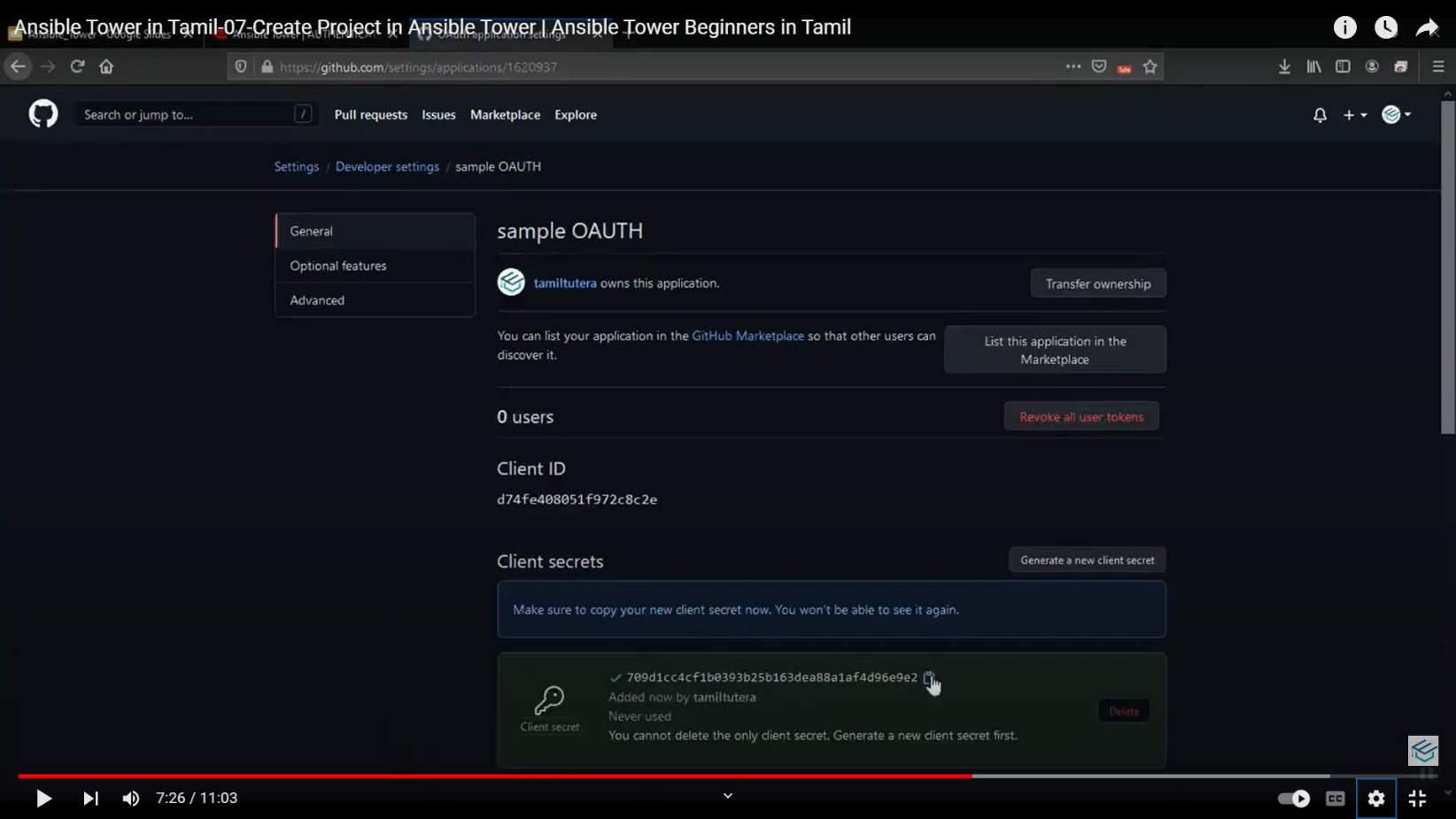


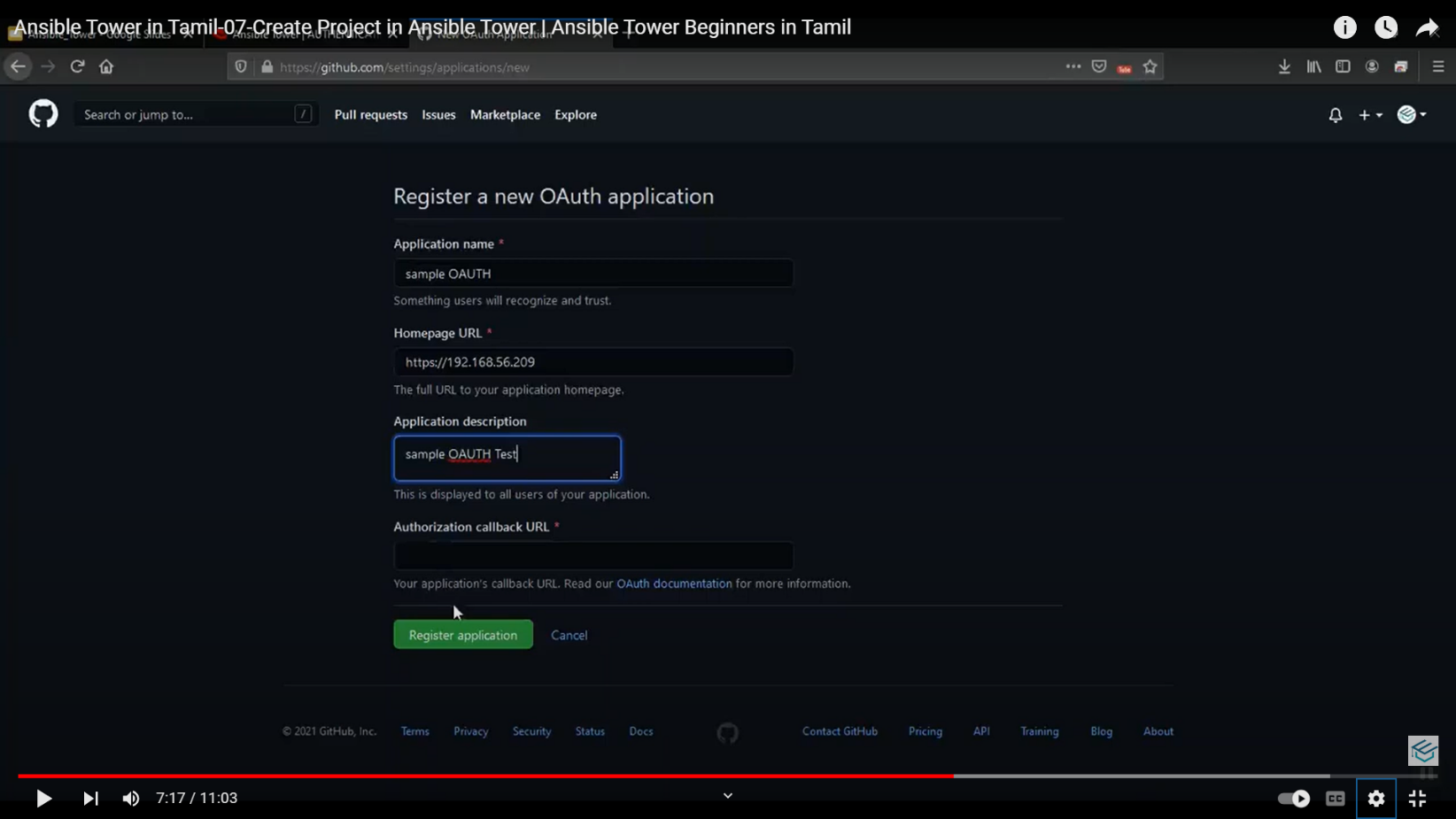
**To configure GIT in Ansible tower**

* Click settings on left panel 🡪 Authentication 🡪 select **GITHUB** tab 🡪 here provide GITHUB oauth call back URL and GITHUB oauth secret.
* Go to GitHub 🡪 top right settings 🡪 Developer settings in left panel 🡪 OAuth apps 🡪 New OAuth app on top right
* Provide application name, Homepage URL (Ansible tower URL), Auth call back URL (copy from Ansible tower) given under **GITHUB OAuth call back URL section.** Then, click on register application.
* In the next window, click on **“Generate new client secret”.**
* Then copy the secret code and paste under **“GITHUB OAuth secret”** field in Ansible tower.



****

****

****

**To check the created project**

* Open Ansible tower server instance, type **cd /var/lib/awx/projects**
* Then, type **ls 🡪** it will show the project names
* Enter into a project by giving 🡪 cd \_\_\_\_\_\_\_\_\_(project name)
* Then, type **ls 🡪** it displays the output.
* We can confirm the outputs with the Git rep

**To create a project manually**

* Projects tab in left panel 🡪 provide name, organization, SCM type as **manual** 🡪 select the project base path 🡪 then select playbook directory.
* We can also give permissions to the project like read, use, Admin.
* Go to permission tab 🡪 Click **(+)** 🡪 select the username 🡪 then select the access.

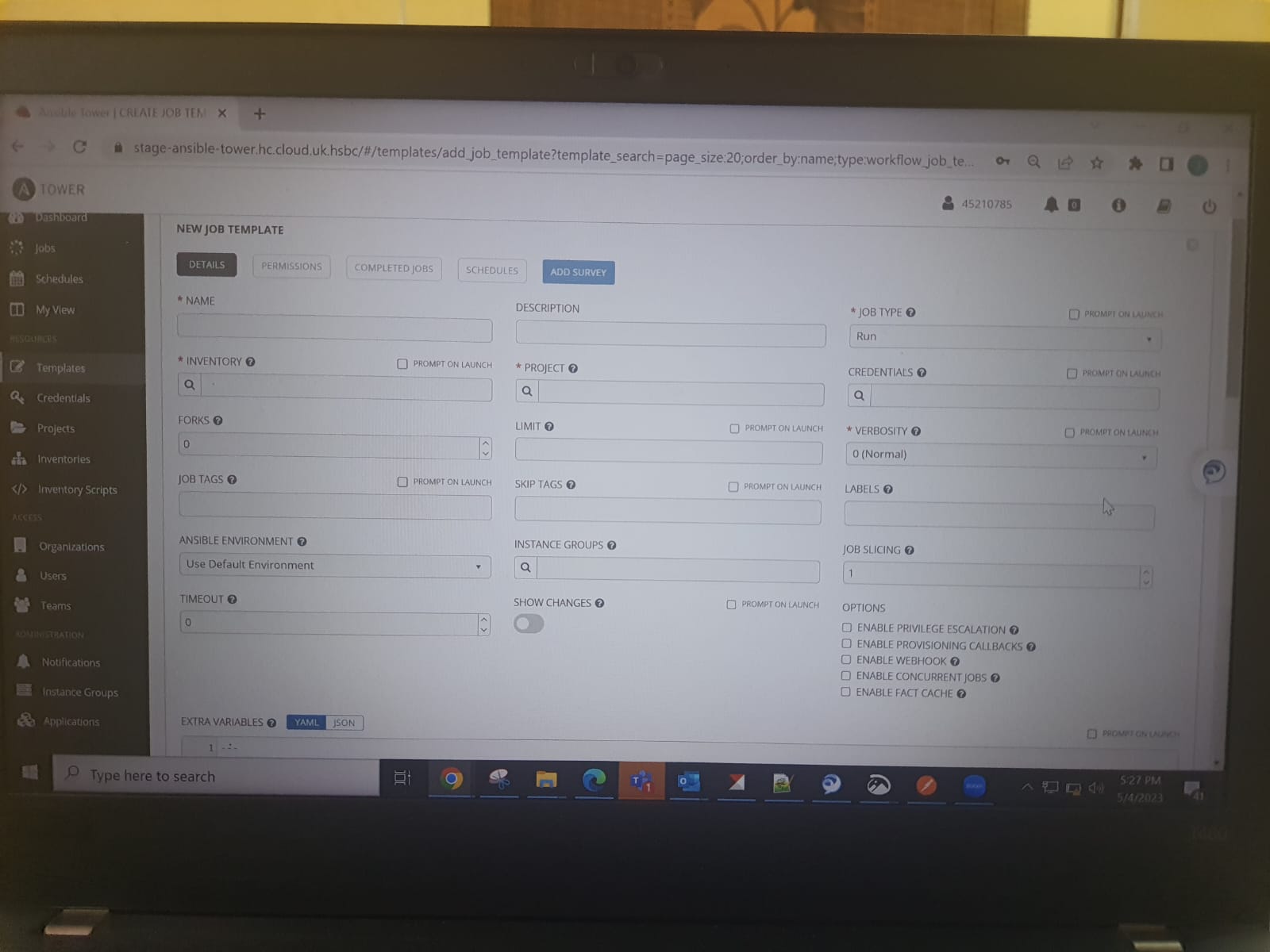
**Use –** Users can run & view the playbook o/p & status

**Read –** Can view the playbook details (playbook code, variables, inventory)

**Update –** user can modify the playbook code, variables & other settings. (Like admin).

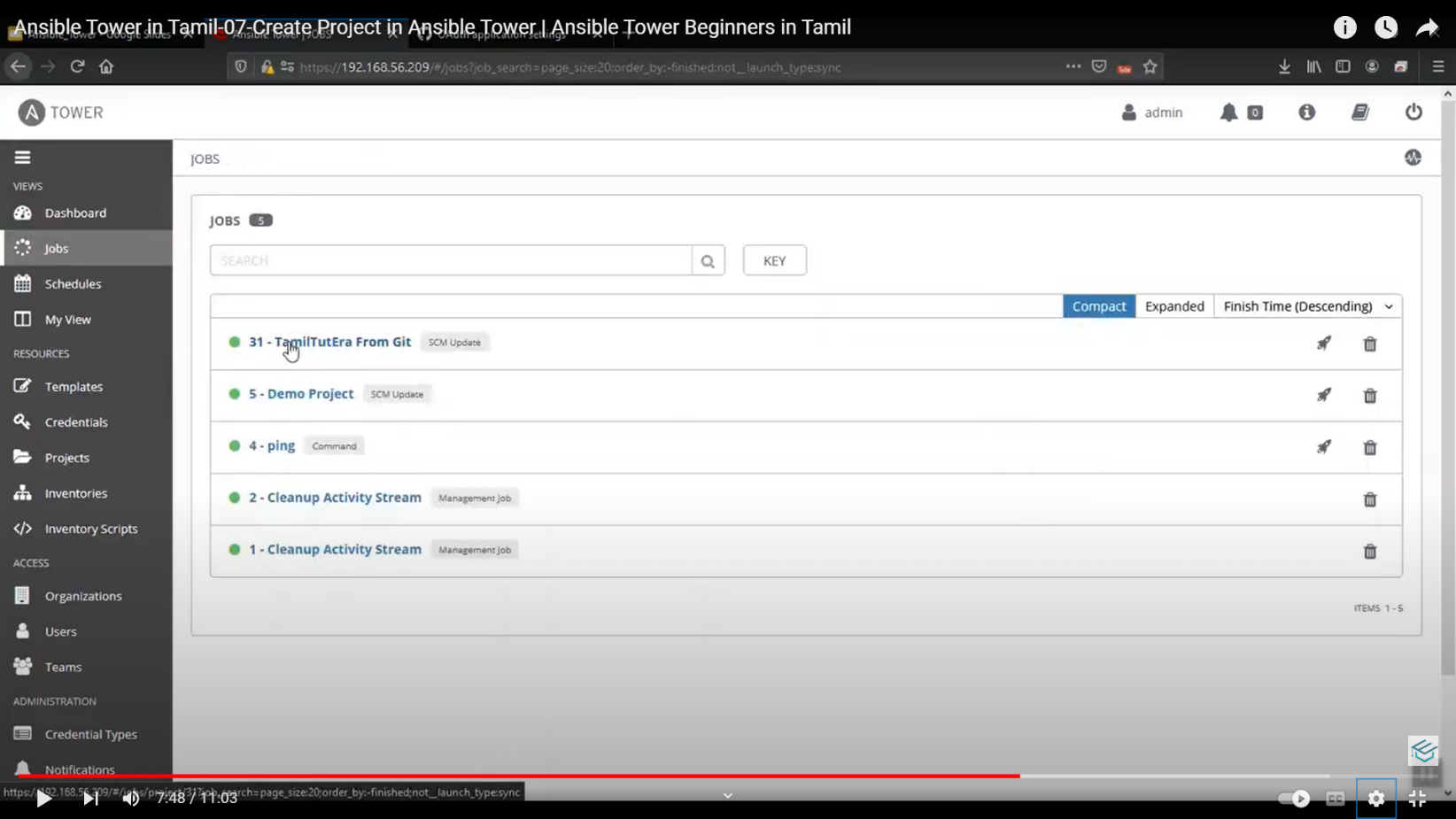
1. **Job Template creation**

* Set of parameters for running an Ansible job.
* Useful to execute the same job many times.
* Enables user to launch, schedule, modify and remove a job template.



**To create**

* Click **Template** on left panel in Ansible tower console.
* Provide the details like
  + Name
  + Job type (Run r check)
  + Inventory (host servers)
  + Project
  + Playbook (select our playbook which want to execute)
  + Verbosity
  + Credentials
  + Limit (we can select the nodes which the playbook wants to execute)
  + We can also give permission to user or to team to execute the job.
  + We can create notification to send to user about the job.
  + Also, we can schedule a job by selecting the schedule tab.



1. **Creating credentials**

* Used to authenticate remote systems.
* Credentials are encrypted before it was saved to Tower database & cannot be retrieved in clear text format.

**Types:**

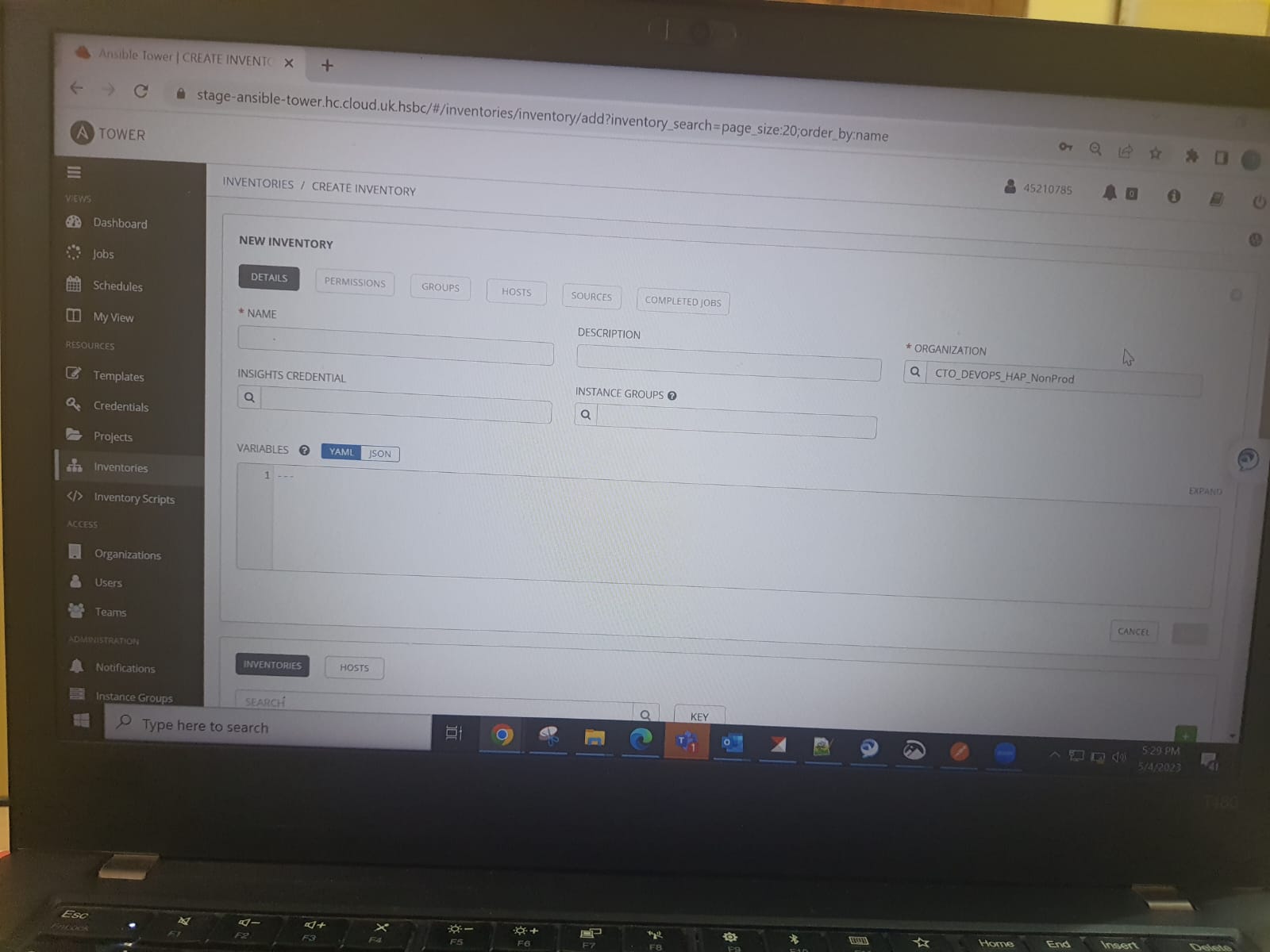
* **Machine 🡪 Used to establish connection b/w Ansible tower server nd managed hosts.**
* **Network 🡪** Used to authenticate connection b/w Ansible tower & n/w devices (router, switches, firewalls). **Why bcoz 🡪** To ensure the authorized users or system to make changes in the configuration.
* **Source Control 🡪 Used to authenticate connection b/w Tower & source control system (Git repo)**
* **Inventory 🡪** To authenticate connection b/w tower & hosts in our inventory.

Go to credential option 🡪 Details tab 🡪 Provide details (name. organization, cred type (machine), usrname, pwd r SSH keys).

For SSH key usage, we can drag nd drop the keys to required field.

1. **Adding Inventory**

**Inventory 🡪** Collection of hosts against which jobs may launched. Also, we can create inventory groups.



* Inventory menu 🡪 click Add (+) 🡪 Provide details (name, organization)
* Hosts tab 🡪 add hostnames

We can add number of hosts by clicking (+) icon in hosts tab.

**To run any ad-hoc commands in the hosts**

* Select the hosts which u want to run commands 🡪 the select **run command** option
* Select module (command, shell, run, apt, ping ,etc)
* Provide hosts names, machine cred, verbosity

**You can add the hosts IP address to the default host file**

**Path (/etc/hosts)**

**To create group in inventories**

* Go to **group** tab 🡪 provide group name 🡪 select hosts tab

Click **+** (add) 🡪 select existing r new host 🡪 then add it to the required groups.

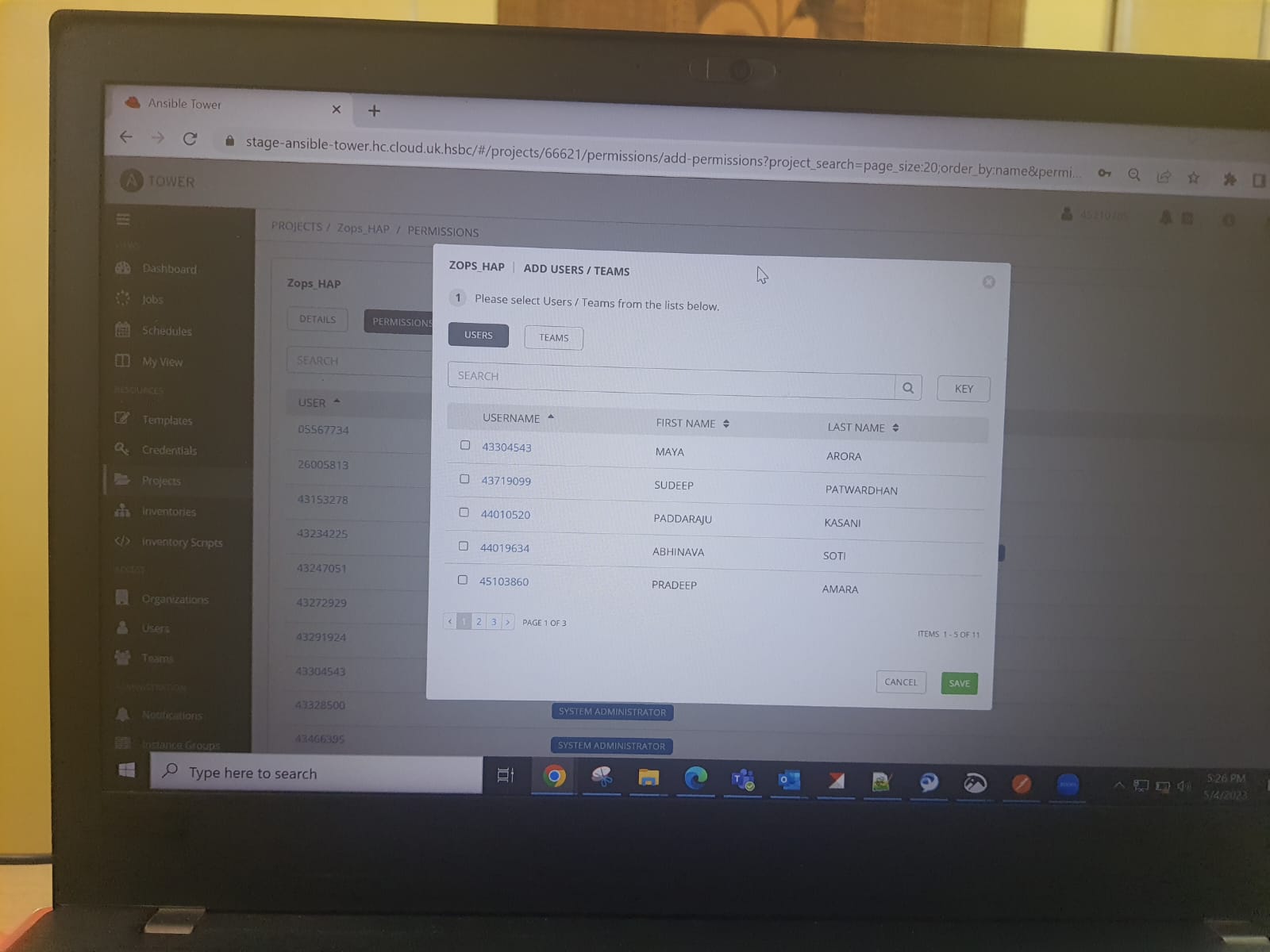
1. **Providing permission**

* Select any of the section such as Organization, Projects, Inventories, etc
* Click on **permission** tab and provide the required permission to users or to teams.
* Go to permission tab 🡪 Click **(+)** 🡪 select the username 🡪 then select the access.

**Use –** Users can run & view the playbook o/p & status

**Read –** Can view the playbook details (playbook code, variables, inventory)

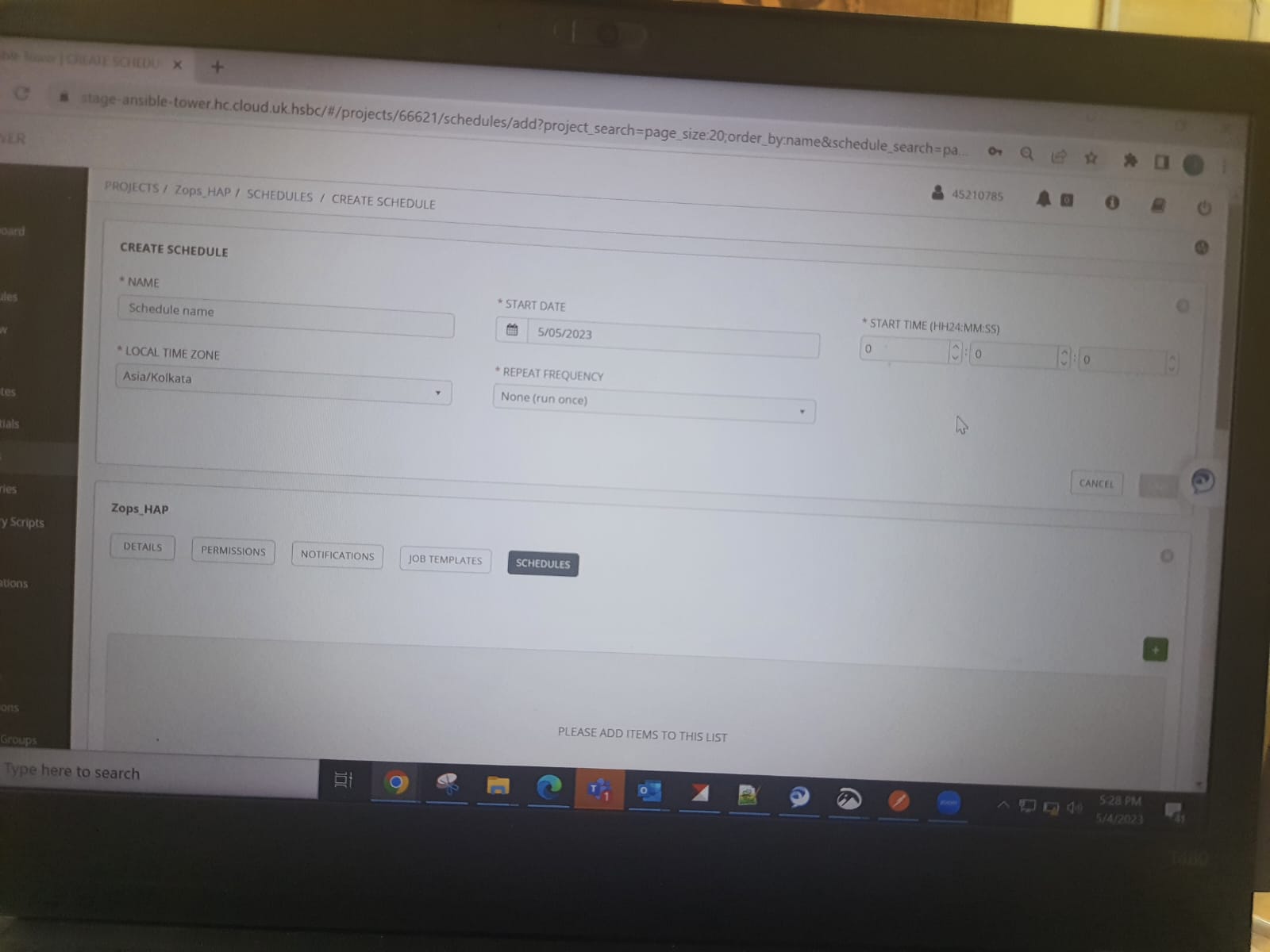
**Update –** user can modify the playbook code, variables & other settings. (Like admin).



1. **Monitoring jobs**

Here we monitor & track job progress.

* Job status 🡪 whether job is running, succeeded, failed, or cancelled (in dashboard)
* Job detail page 🡪 clicking on specific job, shows the detailed view such ass playbook used, inventory targeted, job template used….
* Real-time log 🡪 On job detail page, under **log o/p** tab 🡪shows live logs of jobs.



1. **POSTMAN TOOL**

* It is a free HTTP client-based s/w application.
* Used to perform API testing.

**API –** Application Programming Interface

API is a collection of steps, functions

**Types of API request**

* 1. GET - used to retrieve data or resource from a server. It send request to server and fetch the data.
  2. POST - Used to send data to a server to create a resource.
  3. PUT - used to send data to server to update or replace an existing resource in server
  4. PATCH – used to send partial data to server to modify r update existing resource without replacing the representation.
  5. DELETE – used to send a request to server to delete a specified data.
  6. COPY
  7. HEAD
  8. OPTIONS
  9. LINK
  10. UNLINK

1. **CURL command**

Tool used to transfer data (either upload r download) from a server using protocols (HTTP, HTTPS, SMTP, etc)

**Curl command syntax contains**

* + 1. Request type – get, post, put, etc..
    2. URL – main URL (appl)
    3. Headers – it contains (client id, secret id, authorization, x-request id, etc..) 🡪 these for accessing the API.
    4. Payloads – data which are passed to API
    5. Username & pwd – cred id.

**Eg curl cmd** : curl -k -u ${user}:${password} -X POST ${nexus URL}