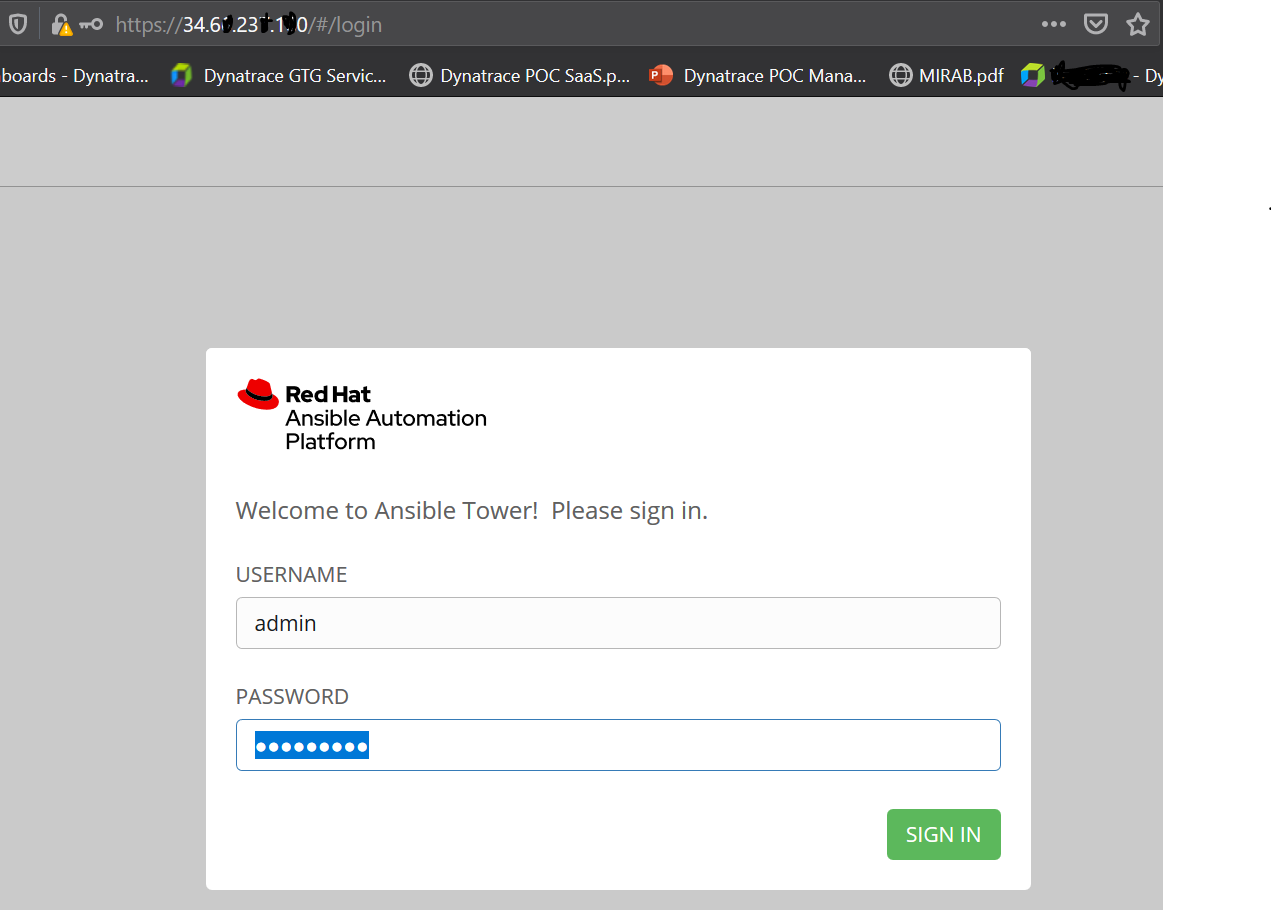
**Pre-Requisites**

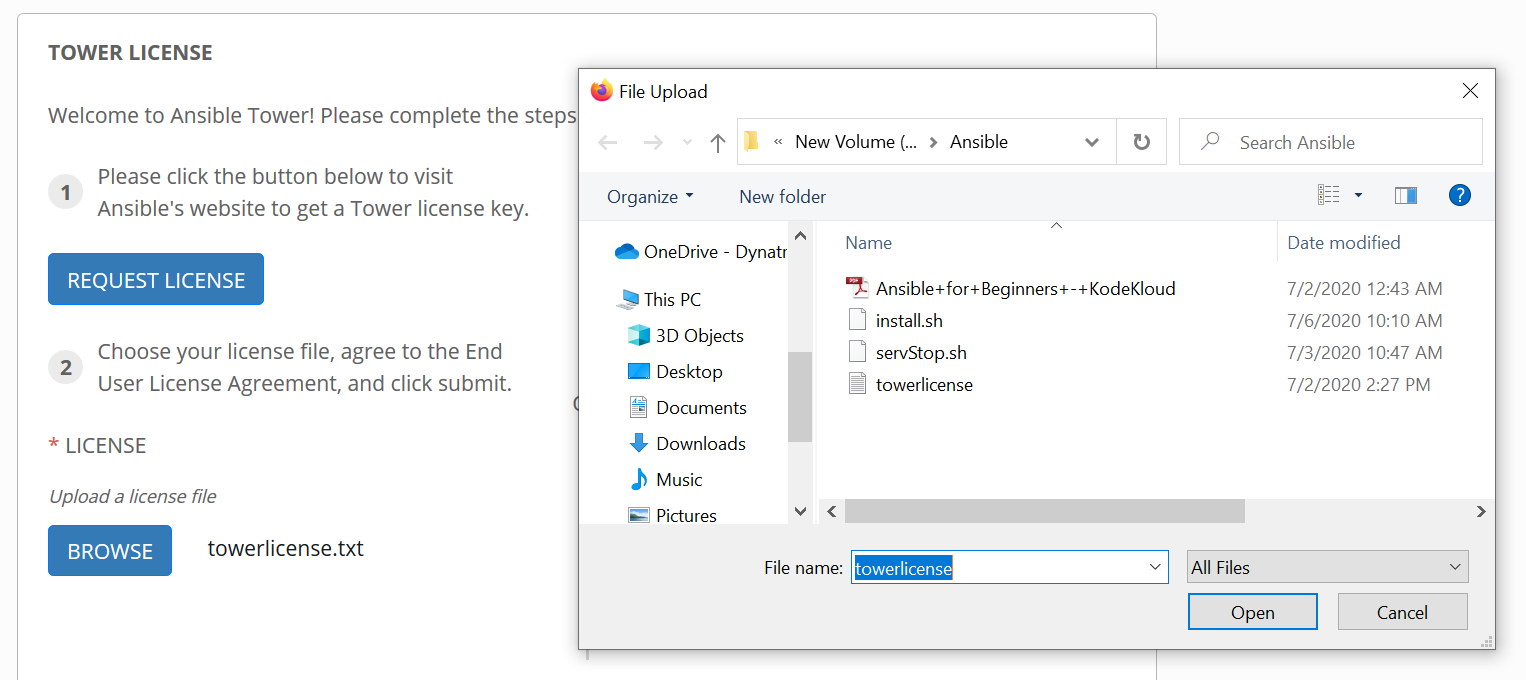
1. Ansible and Ansible Tower needs minimum of 6 GB of RAM in the server where you plan to install.
2. Linux server is required to install Ansible components. This script has been tested on RHEL 7.
3. **Yum** must be configured with all ansible and its dependent packages in the RPM.
4. Dependant Packages:
5. python-babel
6. python-crypto
7. python-crypto2.6
8. python-httplib2
9. python-jinja2-26
10. python-keyczar
11. python-markupsafe
12. python-paramiko
13. python-pyasn1
14. python-simplejson
15. sshpass
16. The ansible tower server should have accessible to internet for Github access and ansible tower package download and license verification.
17. Ansible Tower License

**Post Installation Configuration.**

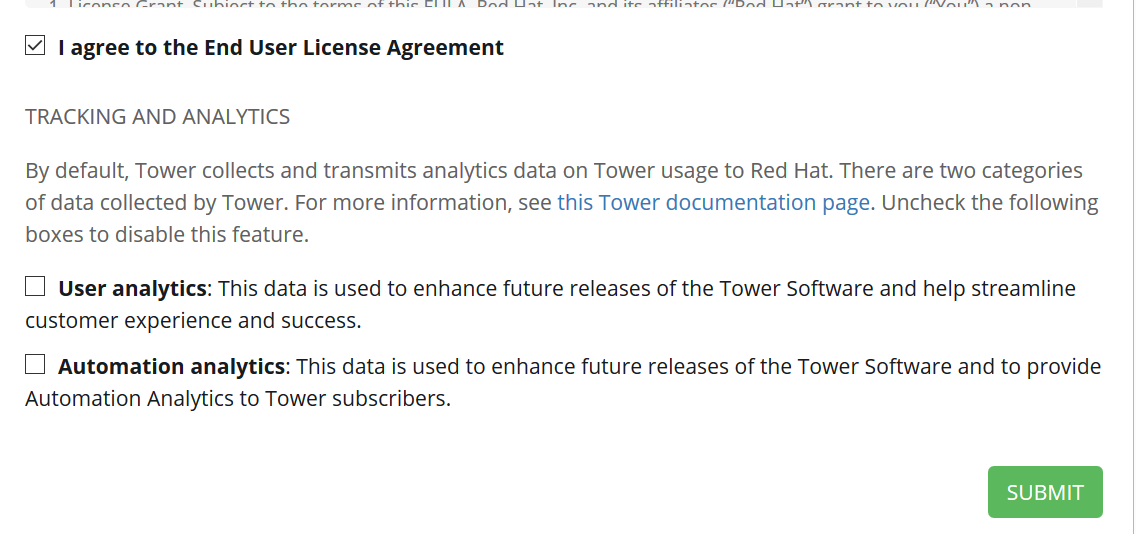
1. Access the ansible tower IP from a browser as *https://<Ansible-tower-server-IP>*
2. Login using username – admin and Password – dynatrace

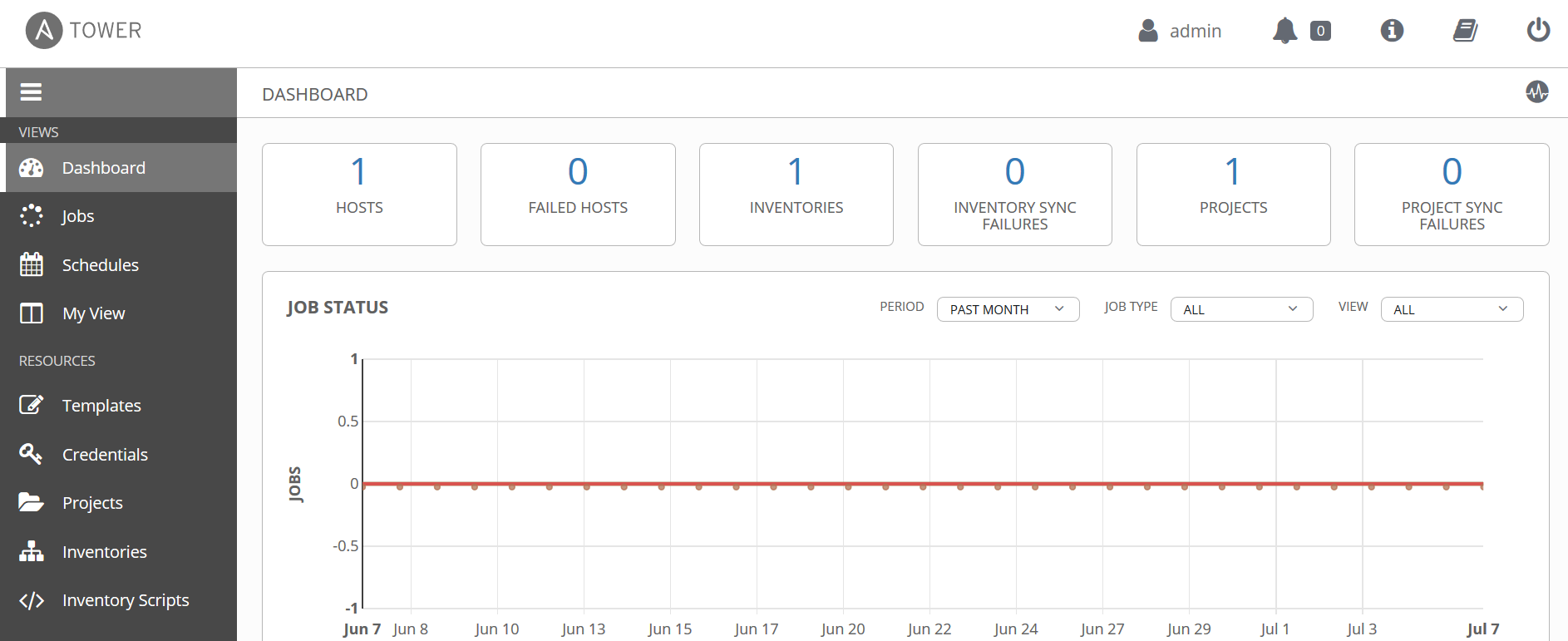


1. Upload the ansible License.



1. Check the agreement button to agree with the term and click on submit.

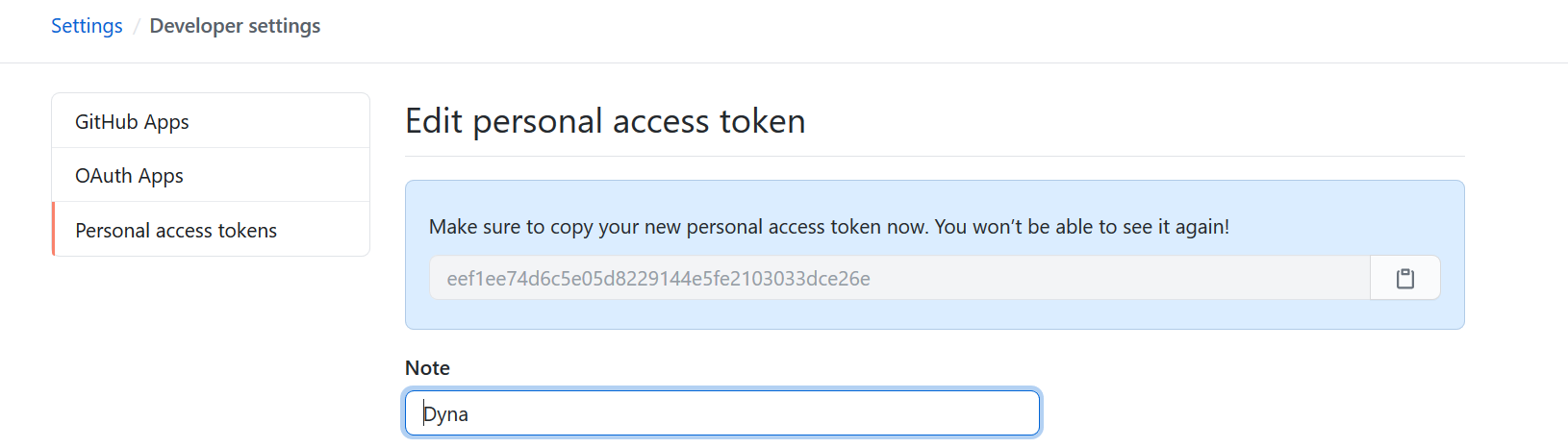


1. Your ansible is ready to configure JOBS

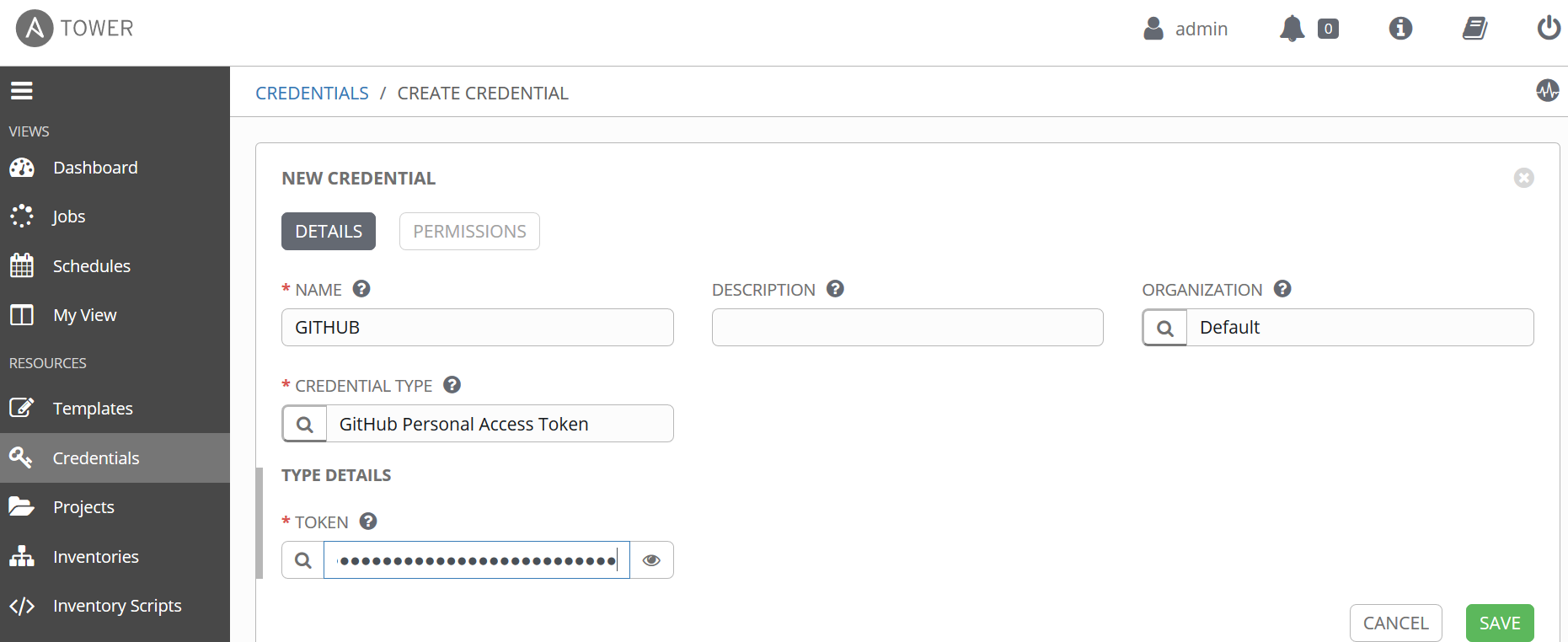
**To Configure Jobs and Integrate the same with your Dynatrace Setup**

1. save your credentials (GitHub passwords and SSH keys) so you can easily reuse them across projects. To do this, navigate to **Settings ► Credentials** and add:

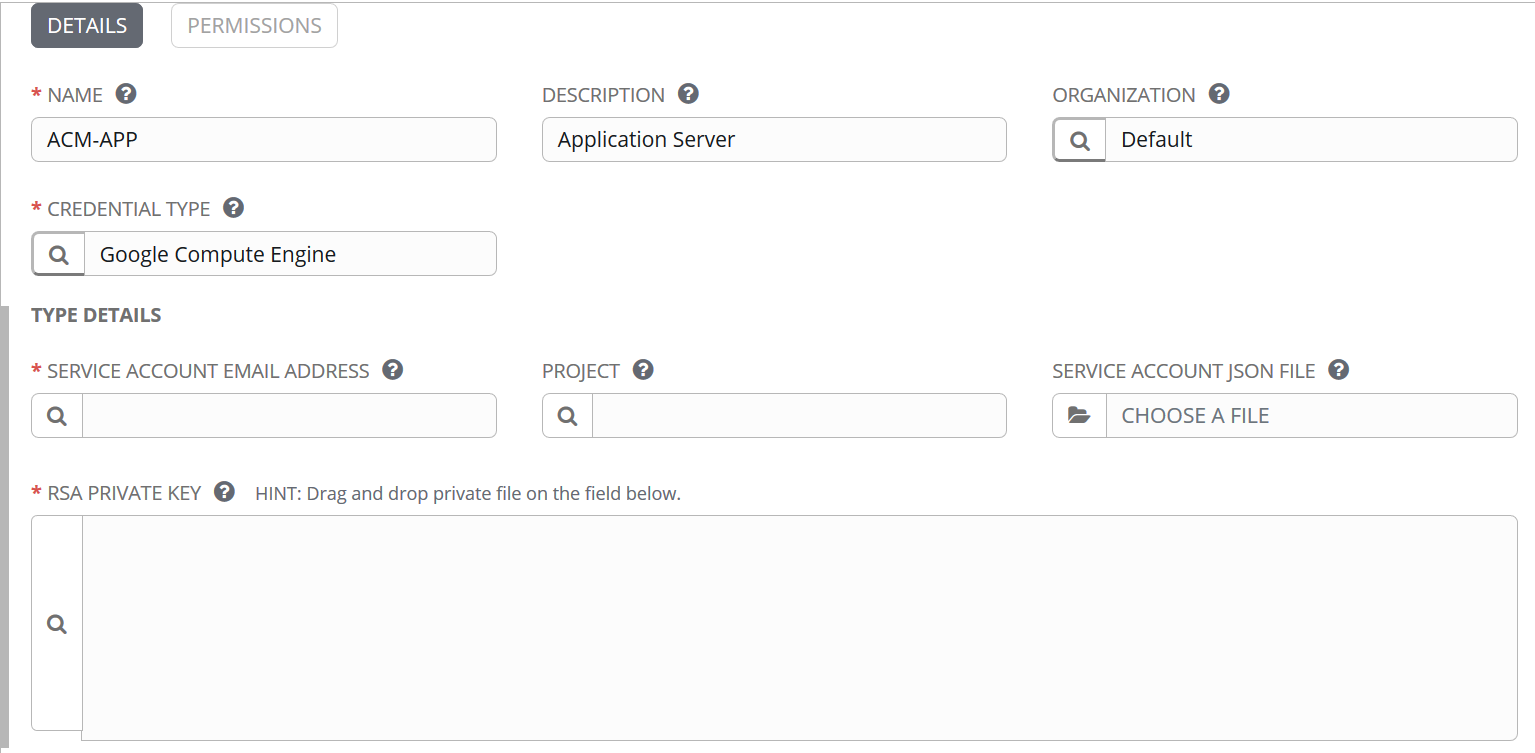
* GitHub password for your GitHub repository for check out
* An SSH key for the VM to connect to (the VM where you want to enable your self-healing applications)

To Generate Token

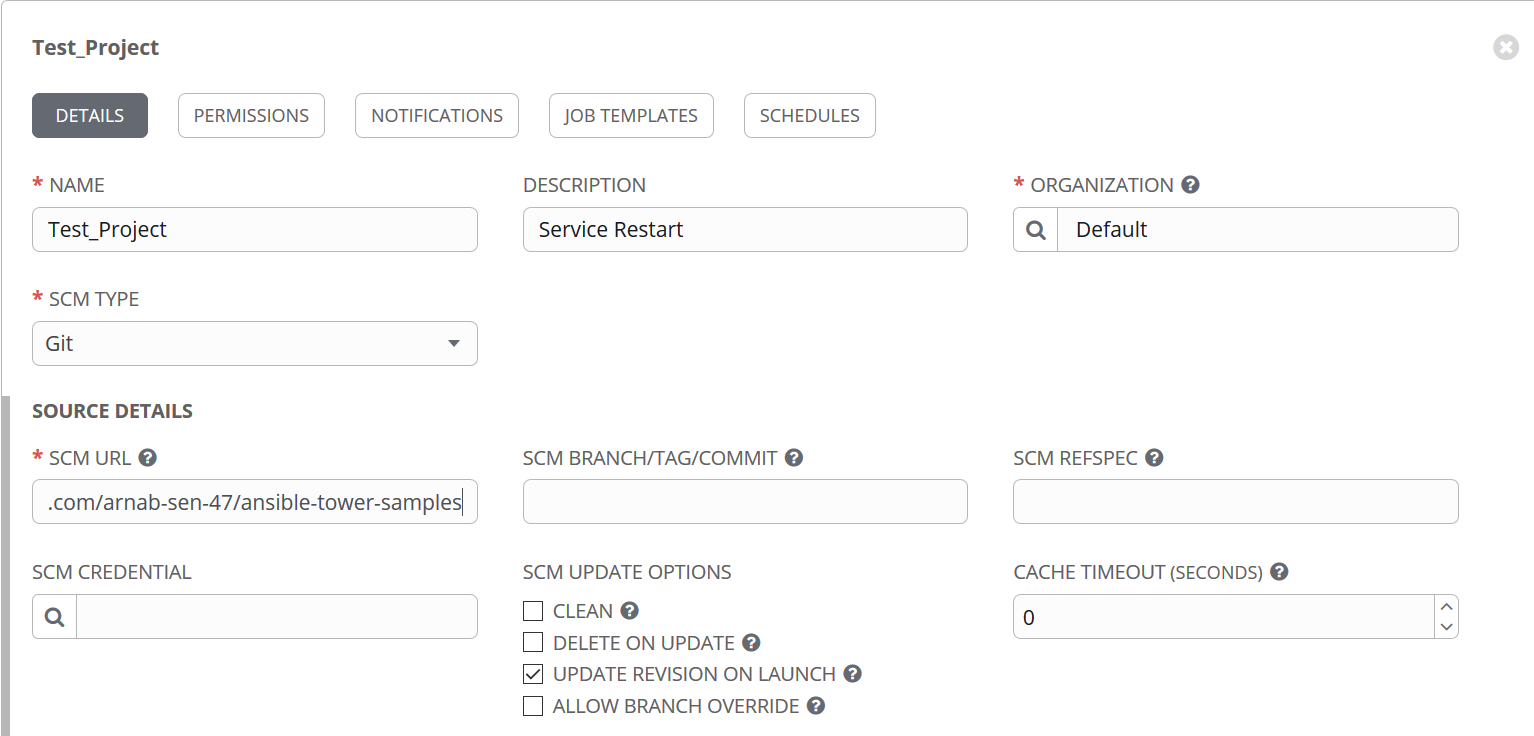
To add the same in Ansible tower and save it



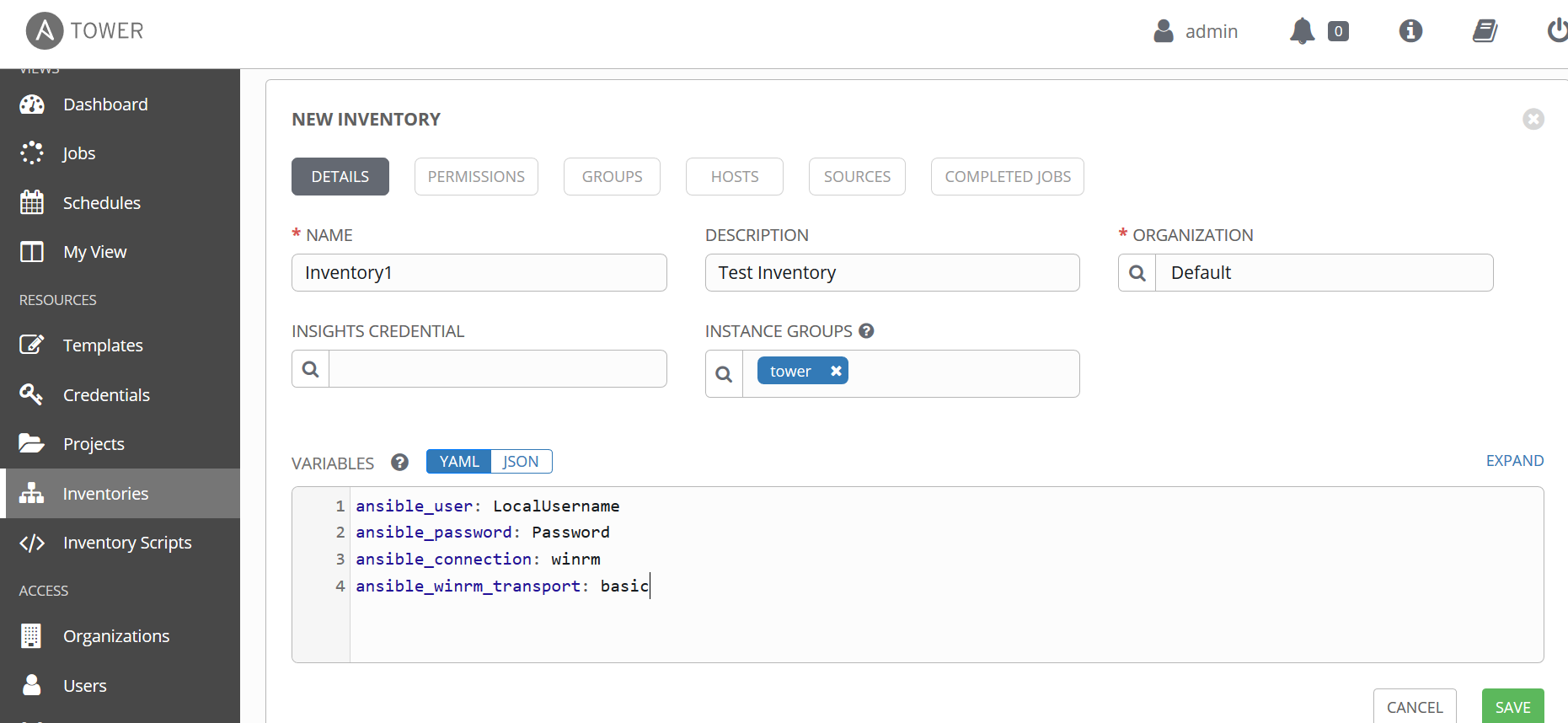
Credential is added, do the same for VMs

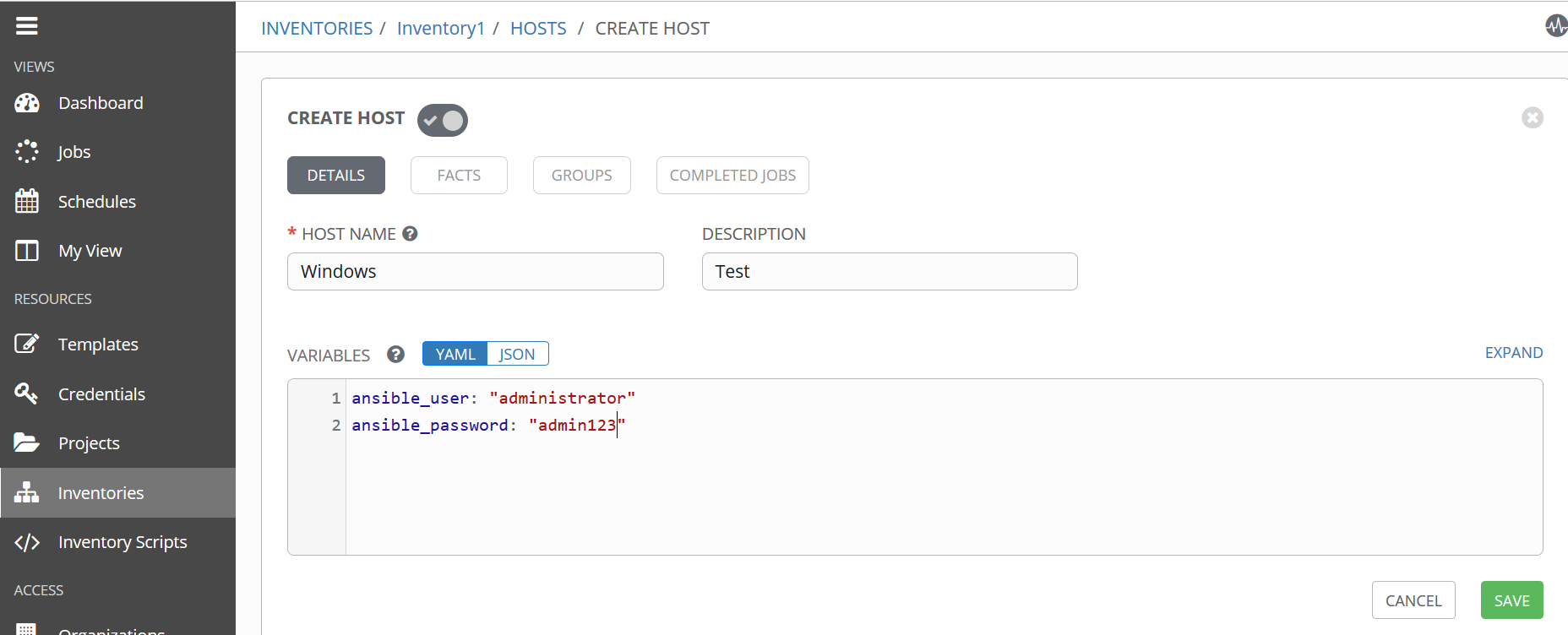


1. Navigate to the **Create Project** page.  
   Use the password stored in GitHub to check out the repository that contains the required source files.
2. Define an inventory that contains the host on which we want to deploy our source from the GitHub repository. Click the **Host** section in your environment and add a new host (IP address or host name). This is the environment where your applications will be deployed.



1. Create a Inventory for the Ansible tower to work upon

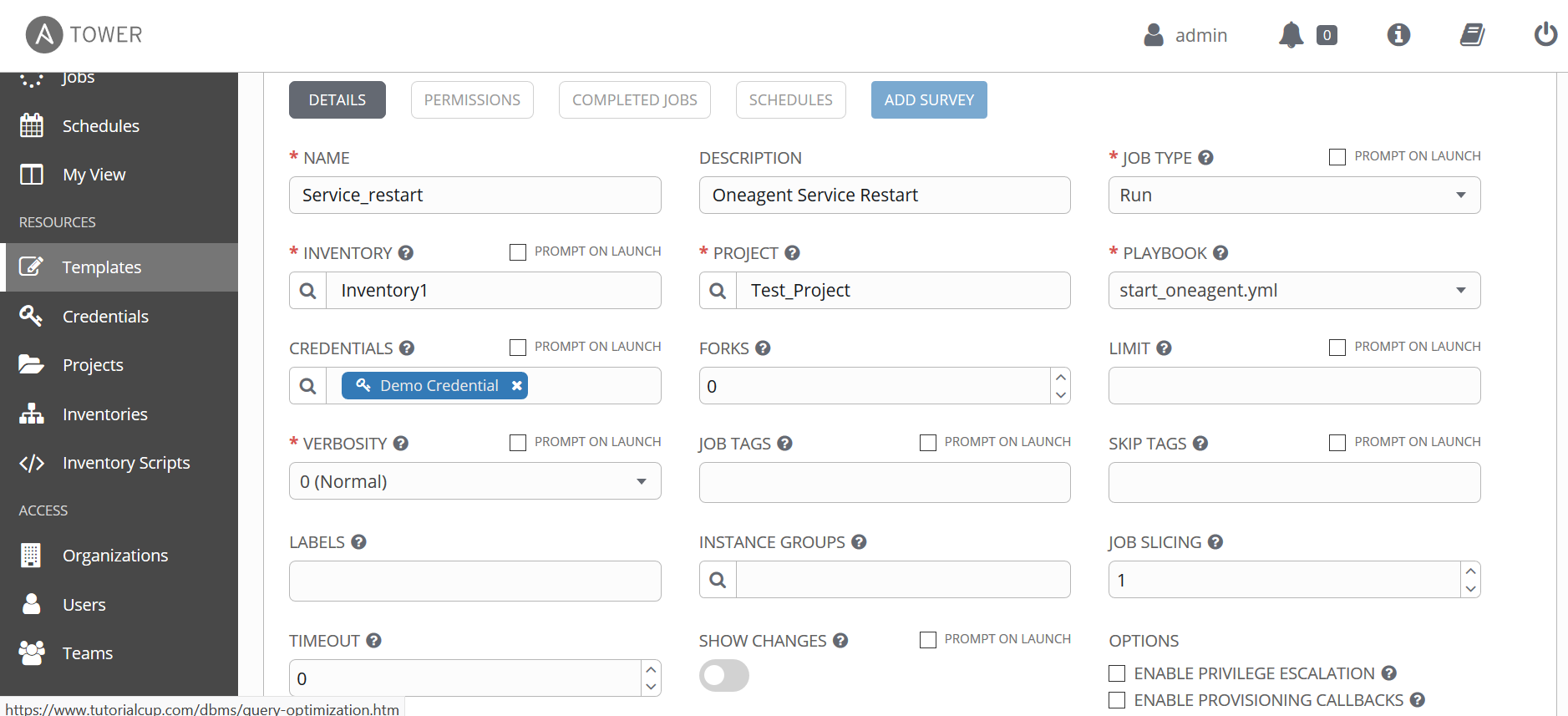




1. Create a template to define the actions that should be followed when the template is triggered by an external or internal source.

this example, we’ll:

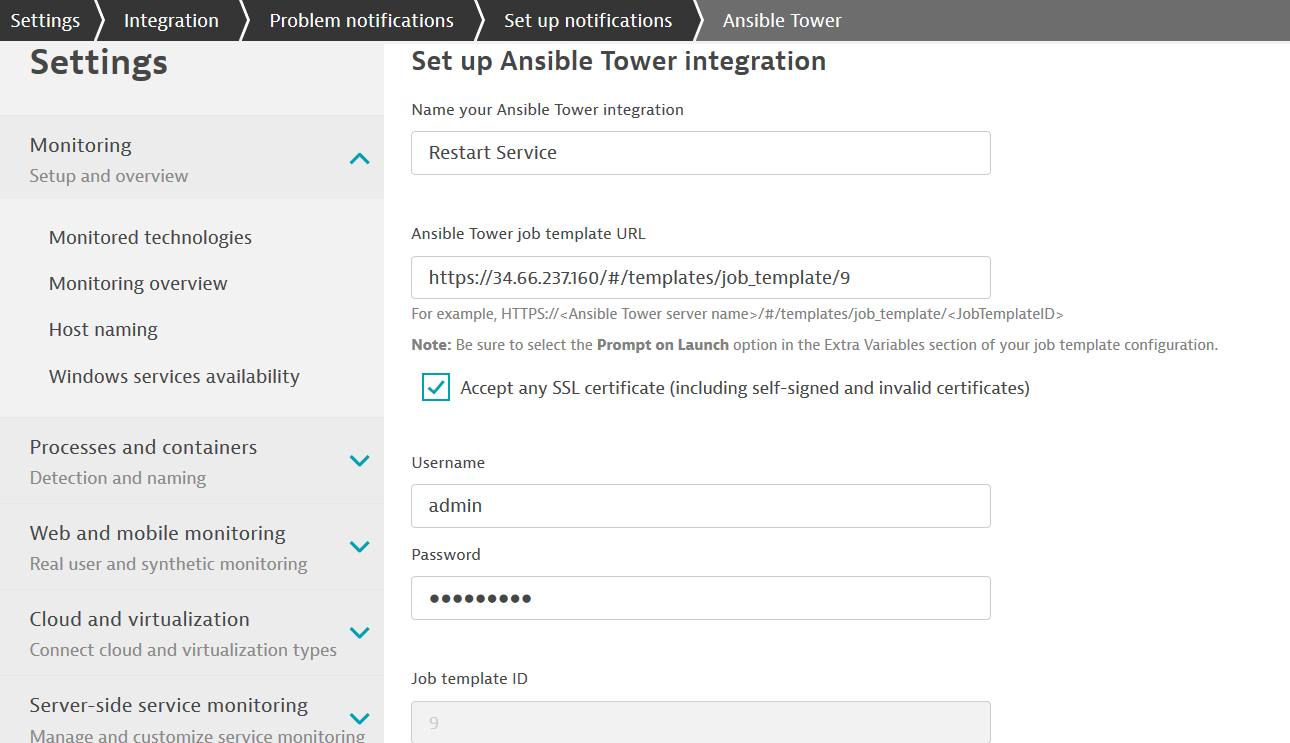
1. Run the template and execute the playbook which is stored in the GitHub repository. The playbook defines the actions that are to be taken when executing this job template.

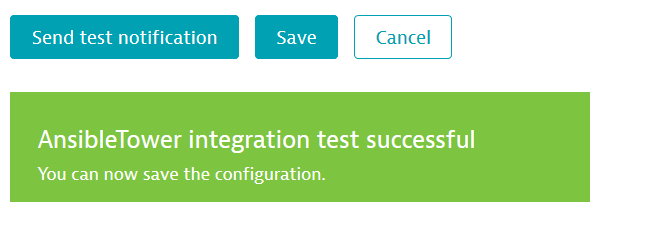


**Connect Ansible Tower to Dynatrace**

To enable applications to heal themselves when errors occur, you need the combined powers of Dynatrace (to auto-detect problems in real-time) and Ansible Tower (to run remediation tasks and bring applications back up following remediation).

1. Sign in to your Dynatrace account.
2. Select **Settings** from the navigation menu.
3. Expand **Integration** and select **Problem notifications** to display the **Problem notification setup** page.
4. Click **Set up notifications** and select **Custom integration** to open the **Set up custom integration** page.
5. Type in a **Name** for the integration (for example, Ansible Tower).
6. Paste the URL of your Ansible Tower REST API into the **Webhook URL** text field. For example, http://your-ansible-url.com/api/v2/job\_templates/<your-job-id>/launch/.  
   Be sure to you use your own job ID and don’t forget the slash at the end of the URL.
7. Click **Create basic authorization header**.  
   You’ll need the credentials of your Ansible Tower installation to allow access to your API. You won’t need a payload, so delete everything between the brackets in the **Custom payload** text field





You Ansible integration is complete

Refer Blog: <https://www.dynatrace.com/news/blog/set-up-ansible-tower-with-dynatrace-to-enable-your-self-healing-applications/>