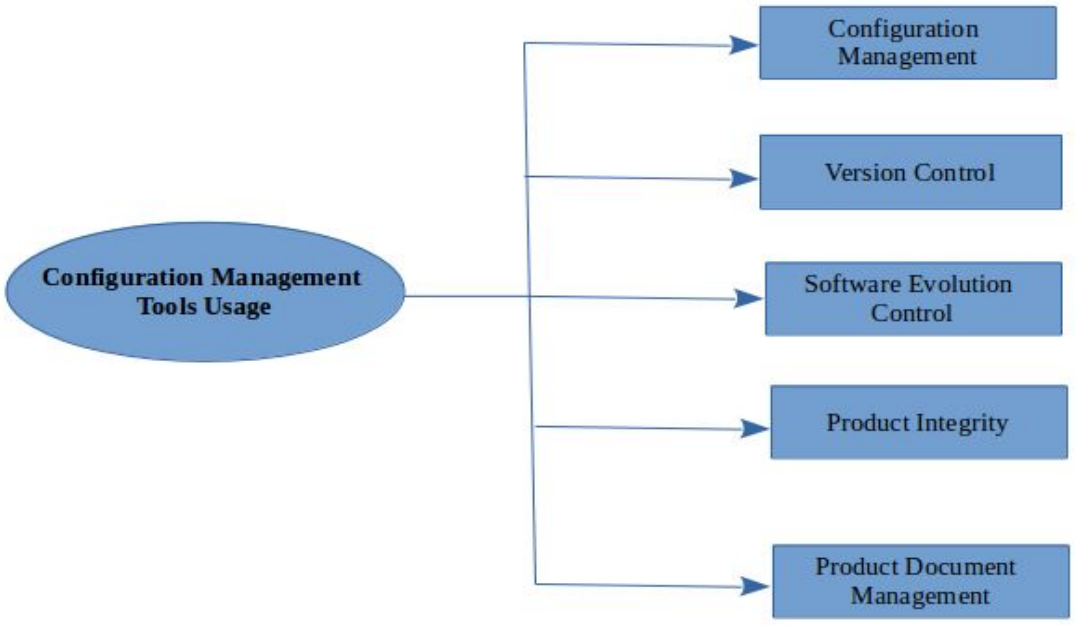
Better Visibility Into Your Infrastructure Automation Using AWX / Ansible Tower

Contents

1. Introduction

Configuration Management tools are very popular for cloud provisioning, automating repeatable tasks, configuration management, infrastructure orchestration, etc. It helps us to configure software or product and manage multiple different application's configurations on complex infrastructure which increases the efficiency.



Ansible is one of the latest configuration tools in the market which is based on python and has an **agentless architecture**. Ansible does not have master and slave setup – which differentiate it with other CM tools like Puppet, Chef, Saltstack, etc.
Ansible ecosystem has one power tool named Ansible Tower which is a web-based GUI which contains most of the features that Ansible has, plus some addon features which makes Ansible Tower different from Ansible are: **Role-based access control, Job scheduling, Notification, Cloud integration, etc.**

Now, let's briefly discuss the features and understand the Tower Dashboard.

2. Introduction to Ansible Tower:

Ansible Tower provides a web-based management GUI for Ansible. Ansible Tower is a great visualization tool for managing and scheduling Ansible Jobs. Ansible Tower also provides role-based access which in return increases the security and minimizes the administrative tasks.

3. Need of Ansible Tower:

Since configuration management tools are the first choice of DevOps and using tools with GUI is more handy. Ansible Tower makes it easy to implement features provided by Ansible and makes it more flexible by its additional features.

Below are some of the must-know features of Ansible Tower:

- **Role-based access control**:
  We can define teams which would be restricted to launch certain jobs only. Moreover, Ansible Tower also allows access control on what a team or a user can do with Inventories and Job Templates.
- **Job scheduling:**
  We can schedule Ansible jobs, dynamic inventory updates, source control updates etc. Scheduling tasks make it more powerful as we can update or run playbook as per the desired time.
- **Integrated Notifications:**
  One more feature which is important to notice is its integration with the notification channels. Some of them are Slack, Hipchat, SMS, PagerDuty and others. Also, we can configure webhook which can trigger other tools.
- **Manage and track dynamic inventory:**
  We can manage and track the inventory files of hosts on the basis of data in external inventory systems. Ansible Tower can integrate with different external inventory systems (LDAP, EC2, CMDBs, cobbler, etc).
- **Cloud integration(AWS, Rackspace, Azure):**
  Ansible Tower can be integrated with major Cloud providers such as Amazon AWS, Rackspace and Microsoft Azure.

4. Installation

AWX can be installed both in single machine scenario and also in a high availability multi-machine cluster, but for the simplicity of this blog, we are going to discuss the single machine setup.

Although you can choose from several single machine installation scenarios like OpenShift, Kubernetes, Docker, and even from distribution packages, I prefer Docker over others mainly because of the ease of management and cleanliness of the installation. Following are the prerequisites you need to have.

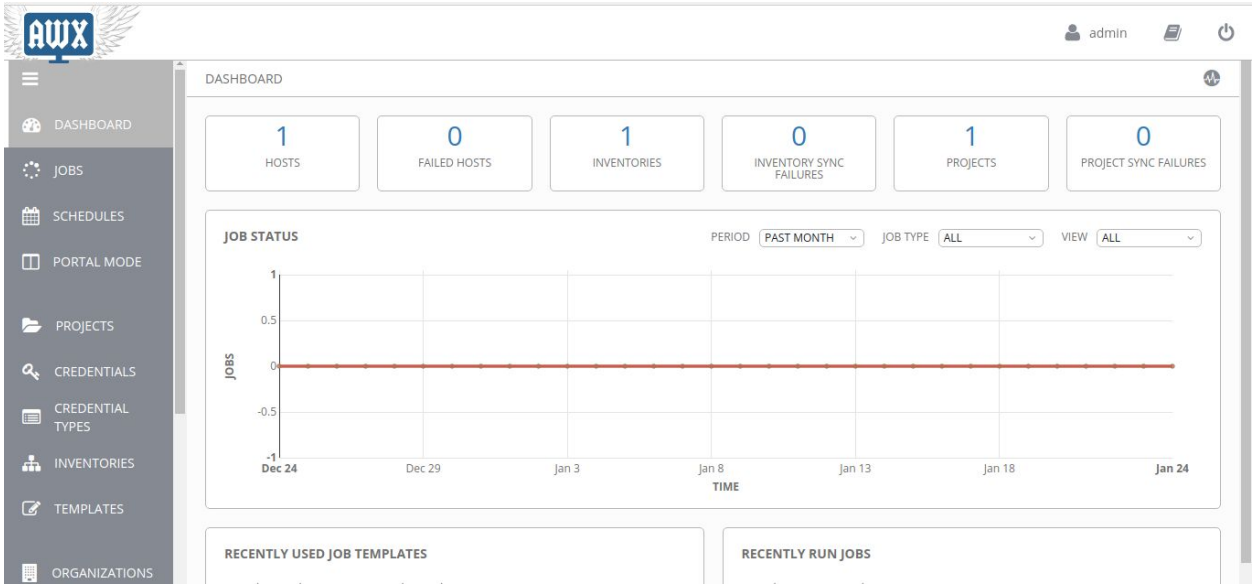
- Ansible (Version 2.4 or above)
- Docker
- Docker-py

Interesting thing is that the installation itself is done by an Ansible playbook managed by AWX project. Once you clone the github project, you just need to run the playbook and post that, 5 docker containers will be brought up.

```
# git clone https://github.com/ansible/awx
Cloning into 'awx'...
remote: Counting objects: 158907, done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 158907 (delta 54), reused 71 (delta 38), pack-reused 158800
Receiving objects: 100% (158907/158907), 191.10 MiB | 47.39 MiB/s, done.
Resolving deltas: 100% (122454/122454), done.
Checking connectivity... done.
#
#
# cd awx/installer/
#
#
# ansible-playbook -i inventory install.yml
```

5. Tower Dashboard

After a successful installation, you can simply hit your host IP or DNS on the browser and you'll see the AWX login page following the dashboard. Default admin credentials are admin & password.

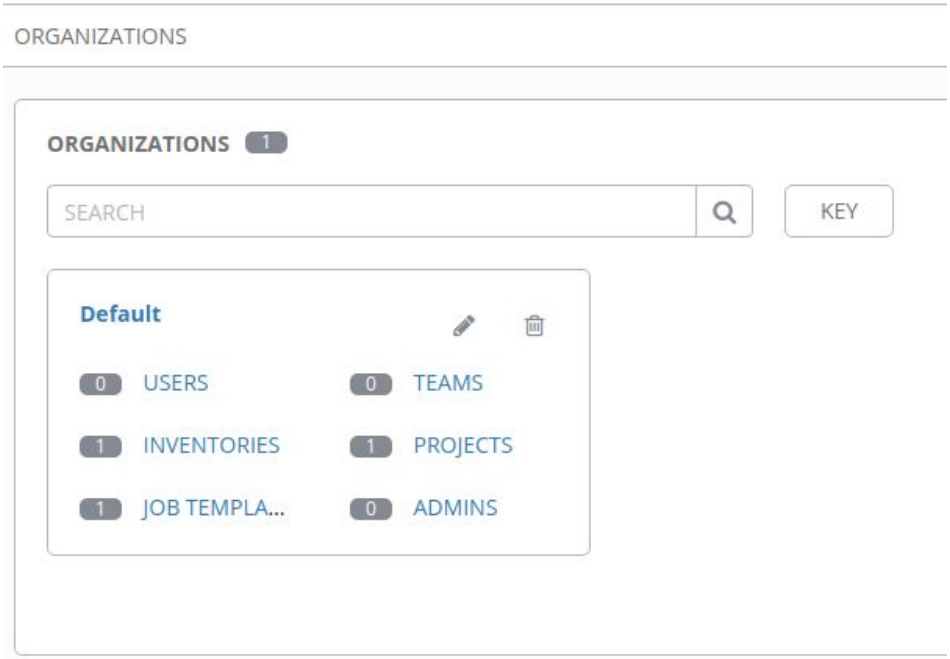


6. Terminology

If you are already versed with Ansible, there are not many terms you will take time to pick up from here, you will find the terms rather intuitive.

6.1 Organization

This is the highest level object in the hierarchy, it is a logical collection of users, teams, projects and inventories. Organization aids in implementing role based access control on the top level. Almost all the objects are associated to an organization.



6.2 Users, User Types and Teams

Users as you can guess, are the lowest level of role based access hierarchy. There are three types of users you can create

- **Normal User** have full access to the resources for which that user has been granted access to within an organization.
- **System Auditor** have read-only access to all the resources within the organization they are part of.
- **System Administrator** have superuser access to the entire awx / tower installation.

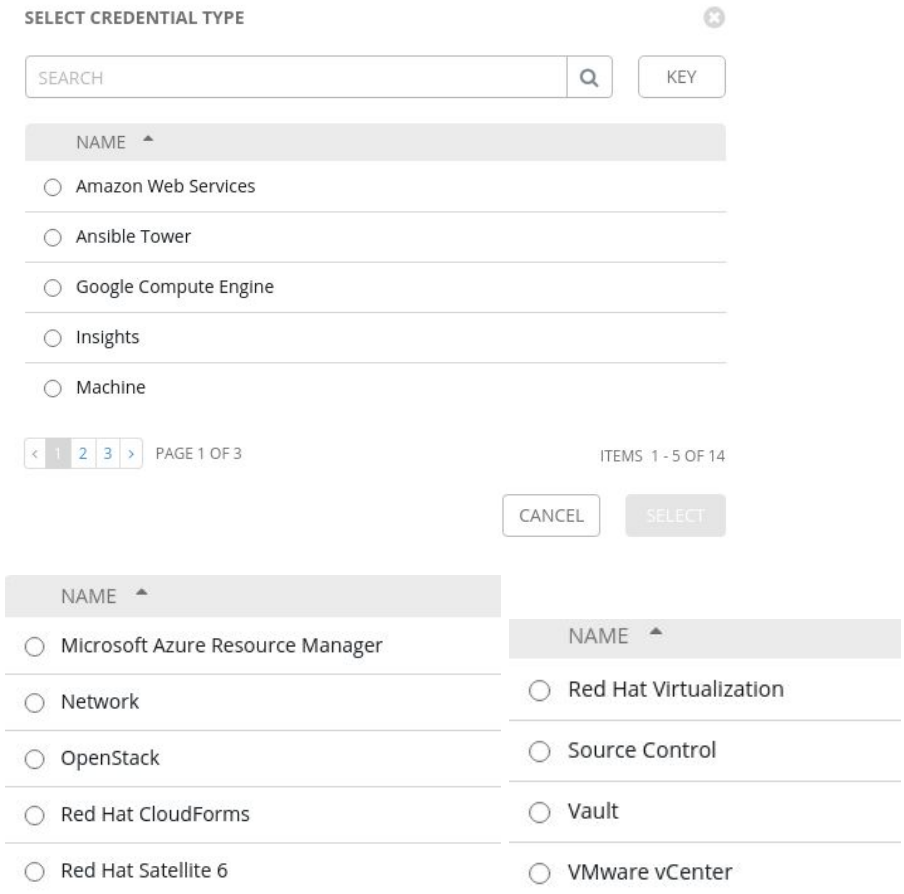
Teams are just a collection of users within an organization for cleaner access control. You can provide access to resources on team level too.

6.3 Inventories, Hosts and Groups

Inventories in AWX/tower are same as that in Ansible with a difference that hosts and groups can configured via the console, and you can control the access of every inventory just other resources in the console.

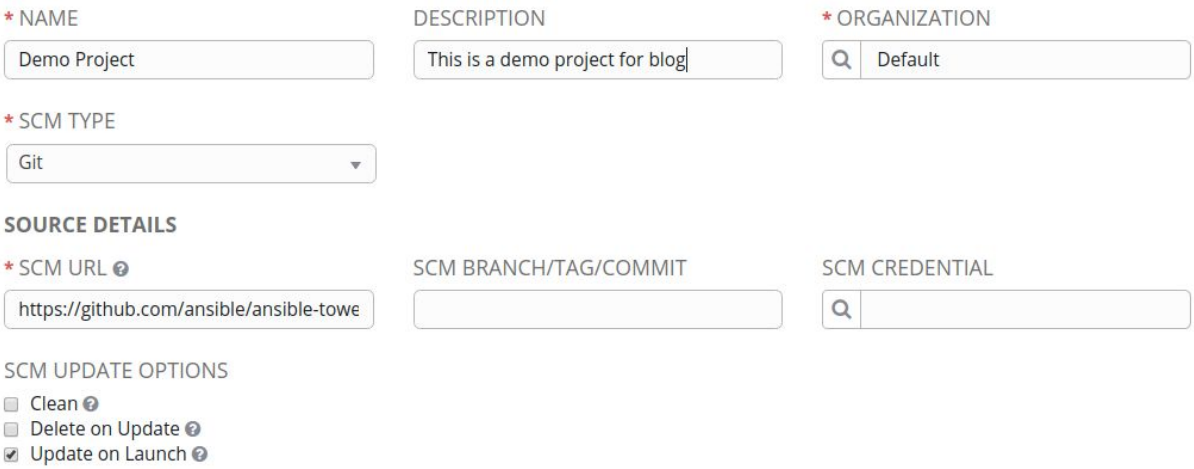
6.4 Credentials

This is where you save all your credentials like SSH key, Ansible Vault password, AWS access credentials, etc. You can also secure a playbook run with a password saved in the credentials. There are several credential types which are supported by the AWX/tower right off the bat (shown below), and you can also add new types.



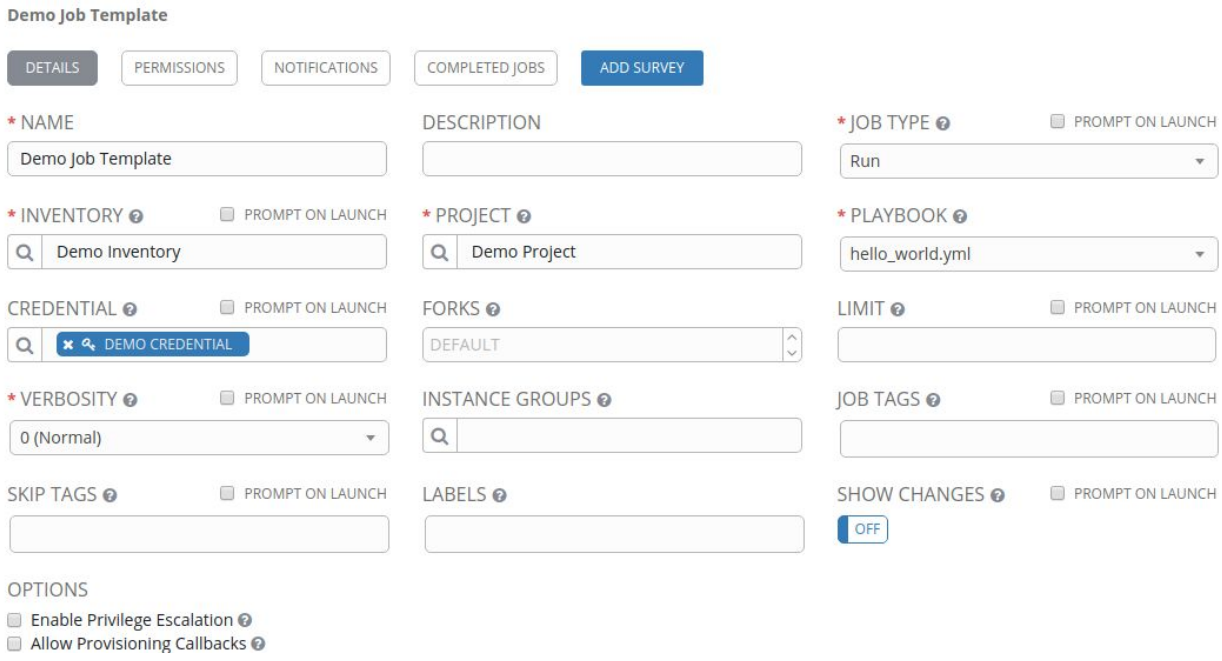
6.5 Project

A project is a collection of playbooks typically stored in an SCM. So essentially you configure a SCM repo in the project which has the playbooks and roles.

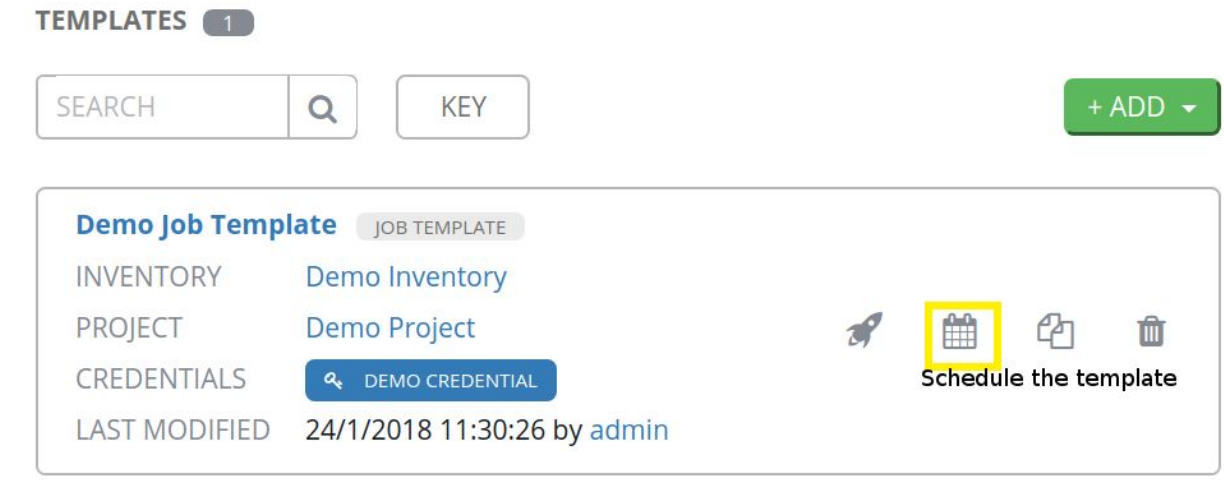


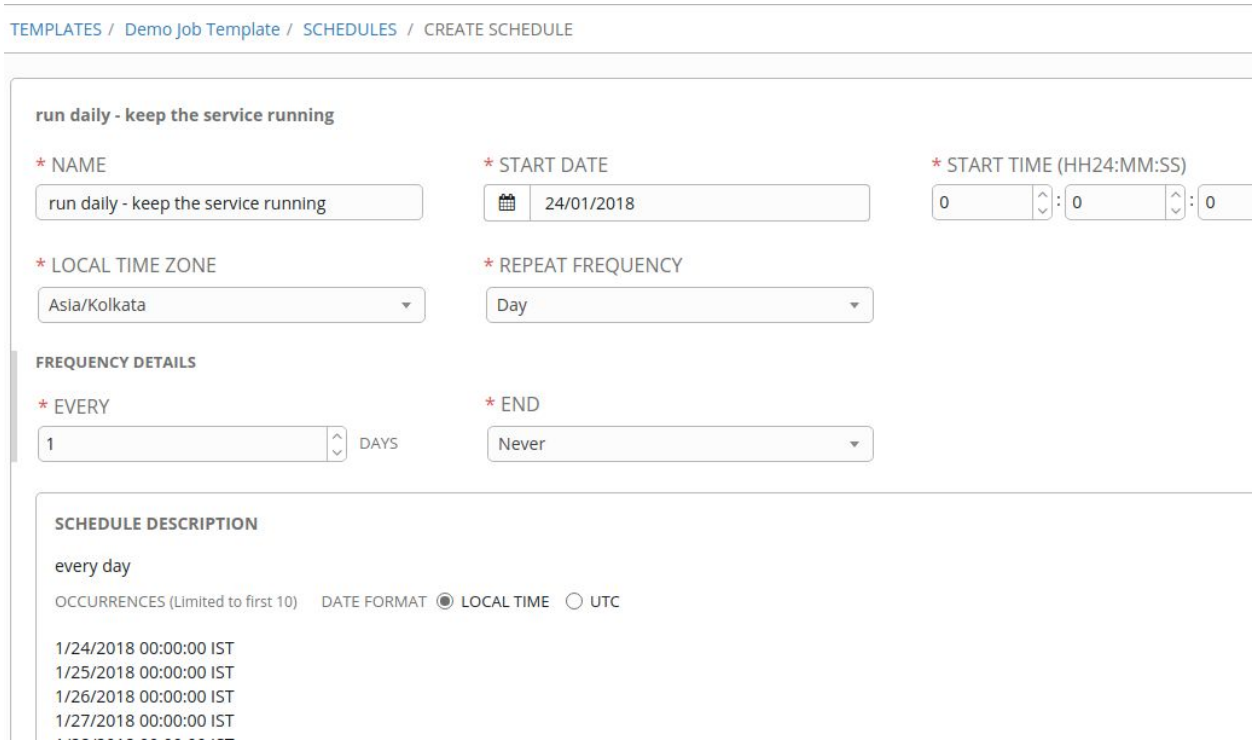
6.6 Template

This is probably the most interesting term in the list, it's where you combine your playbook with the inventory and other settings like credentials, to launch off a job from it.



A template is the object that you would schedule on a regular interval to keep all the hosts in the desired state of the playbook.





That's all you need folks, to get started with Ansible Tower or AWX. I hope you liked the post, you can also contact us if you want to your business to be equipped with this powerful yet straightforward tool.

7. About the Author

Navjot Singh - DevOps Engineer
TO THE NEW

Navjot is a Red Hat Certified DevOps Engineer and one of the leading DevOps consultants. He is also a leading aws certified solutions architect and a shell-scripting expert with hands-on-experience on multi-server environment & deployments. Being a leading AWS DevOps engineer He has strong understanding on AWS and has great ability in gathering requirements and implementing appropriate solutions for customers. He loves playing table tennis.

8. About TO THE NEW

TO THE NEW is a premium digital technology company that provides end-to-end product development services. TO THE NEW leverages the power of experience design, cutting-edge engineering and cloud to build disruptive web and mobile products and enable digital transformation for businesses.

TO THE NEW practices Agile methodologies to develop innovative products with a faster time to market. With a team of 750+ passionate technologists, TO THE NEW constantly challenges the status quo to empower Fortune 500 companies as well as startups across the globe.