
Table of Contents

---

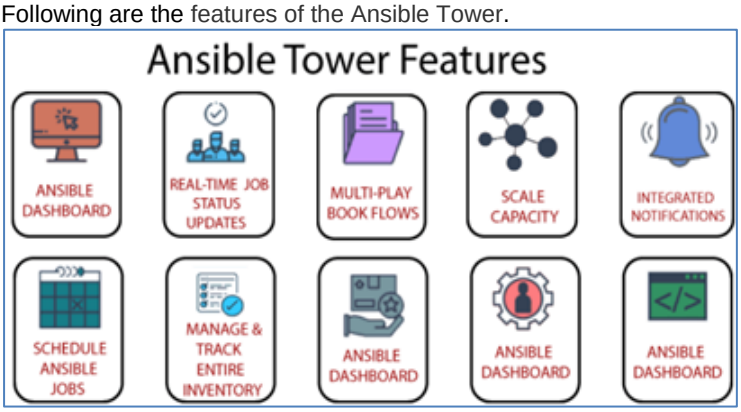
1. Introduction

- Ansible Tower is like Ansible at a more enterprise level.
- It is a web-based solution for managing your organization with an easy user interface that provides a dashboard with all the state summaries of all the hosts.
- And allows quick deployments and monitors all configurations.

1.1. Prerequisites to Install Ansible Tower

- The following OS support Ansible Tower, it required a 64-bit support kernel, runtime, and 20 GB HDD.
  - RedHat Enterprise Linux 6 64-bit
  - RedHat Enterprise Linux 7 64-bit
  - CentOS 6 64-bit
  - CentOS 7 64-bit
  - Ubuntu 12.04 LTS 64-bit
  - Ubuntu 14.04 LTS 64-bit
  - Ubuntu 16.04 LTS 64-bit
- Minimum 2 GB RAM (4 GB RAM recommended) is required.

1.2. Ansible Tower Features

- Following are the features of the Ansible Tower.
- 

1. **Ansible Tower Dashboard:** It displays everything which is going on in your Ansible environment, such as the inventory status, the recent job activity, the hosts, and so on.
2. **Real-Time Job Updates:** Ansible can automate the complete infrastructure. Also, you can see real-time job updates such as plays and tasks broken down by each machine either been successful or failure. Therefore you can see the status of your automation and know what's next in the queue.
3. **Multi-Playbook Workflows:** It allows to chain any numbers of playbooks, any way of the usage of different inventories, runs different users, or utilizes various credentials.
4. **Scale Capacity with Cluster:** You can connect multiple Ansible Tower nodes into an Ansible Tower cluster as the clusters add redundancy and capacity, which allows scaling Ansible automation across the enterprise.
5. **Self-Service:** You can launch playbooks with just a single click through this feature.
6. **Remote Command Execution:** With this command, you can run simple tasks such as restart any malfunctioning service, add users, reset passwords on any host or group of hosts in the inventory.
7. **Manage and Track Inventory:** It manages your entire infrastructure by pulling inventory from public cloud providers such as Microsoft Azure, amazon web services, etc.
8. **Integrated Notification:** This notifies you when a job succeeds or fails across the entire organization at once or customize on a pre-job basis.
9. **Schedule Ansible Jobs:** It schedule different kinds of jobs such as playbook runs, cloud inventory updates, and source control updates to run according to the need.
10. **REST API and Tower CLI Tool:** Every feature present in Ansible Tower is available through the Ansible Tower's REST API, which provides the ideal API for the systems management infrastructure. The Ansible Tower's CLI tool is available for launching jobs from CI systems such as Jenkins, or when you need to integrate with other command-line tools.

2. Installation

- Before you install Ansible Tower, you must install and configure the **Ansible** on your operating system and then install **PostgreSQL**.

➢ **Install and Configure Ansible on Ubuntu**

- As a root user, configure the Ansible PPA using the below commands.

```
# apt-get install software-properties-common
# apt-add-repository ppa:ansible/ansible
```

- After configuring, install Ansible using the below commands.

```
# apt-get update
# apt-get install ansible
```

- Install PostgreSQL

```
# apt-get update
# sudo apt-get install postgresql postgresql-contrib
```

- Download Ansible Tower - register to download Ansible - Tower. You will receive an email after you register. Open your mail and then click on the download button.
- Extract the Ansible Tower

```
# tar -xvzf ansible-tower-setup-latest.tar.gz
# ansible-tower-setup-<tower_version>
```

Note: tower-version is the version of Tower you have downloaded.

➢ **Configure the inventory file**

- After that, set up your inventory file, where you have to mention the necessary passwords (admin_password, pg_password, rabbitmq_password).

```
[tower]
localhost ansible_connection=local

[database]

[all:vars]
admin_password='pass1234'

pg_host=''
pg_port=''

pg_database='awx'
pg_username='awx'
ps_password='pass1234'

rabbitmq_port=5672
rabbitmq_vhost=tower
rabbitmq_username=tower
rabbitmq_password='pass1234'
rabbitmq_cookie-cookiemonster

#Needs to be truw for fqdn and ip address
rabbitmq_use_long-Name=false
```

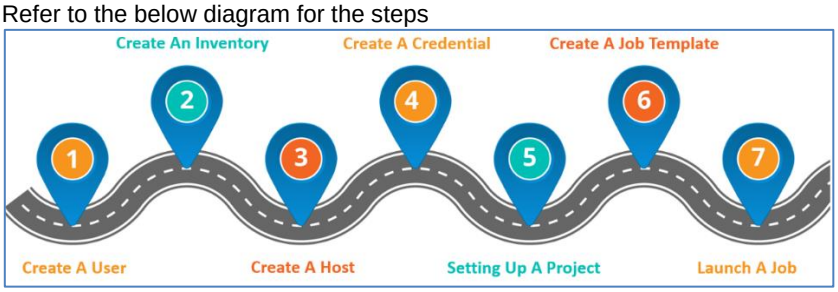
- Run the **setup.sh** script from the path where you unpacked the Tower installer tarball. This will setup the Ansible Tower with help of inventory file.

```
# ./setup.sh
```

- Once you are done setting up Tower, use the web browser to access the Tower server and view the Tower login screen, wherein you have to enter the username and password to access the Tower Dashboard.

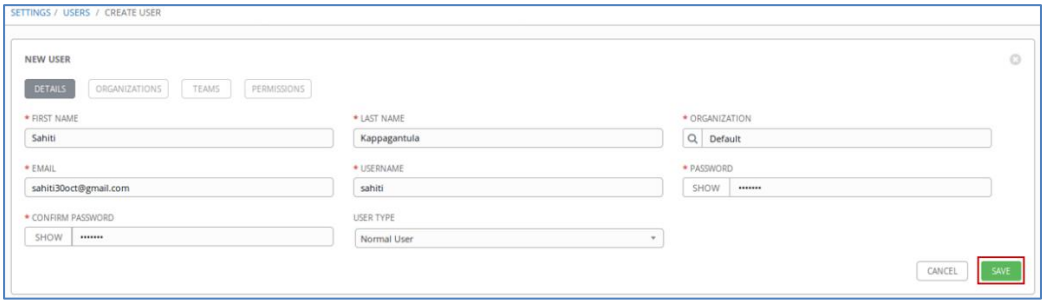
3. Job Creation

- Refer to the below diagram for the steps



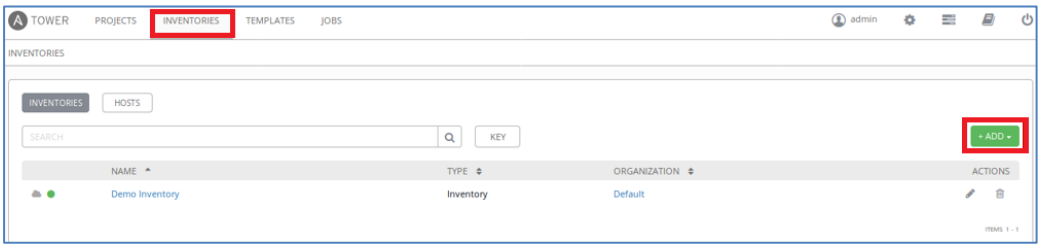
3.1. Create User

- Login into Ansible Tower from Web Browser.
- To create a user, go the **settings** option and choose the **User** tab.
- Click on the **Add** option to add a new User.
- Enter the details required and click **Save**.

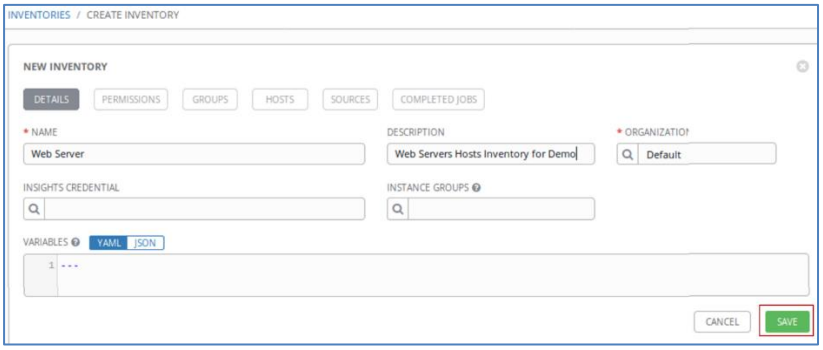


3.2. Create an Inventory

- Click on the **Inventories** option and click to the **Add** option

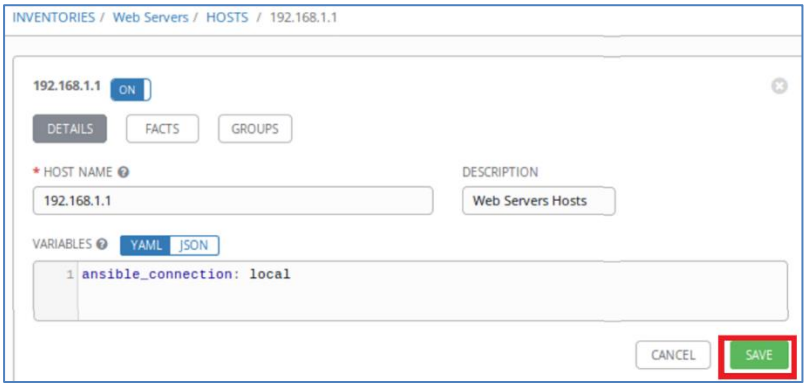


- Enter all the details required, like the name, description, and organization, then click **Save**.



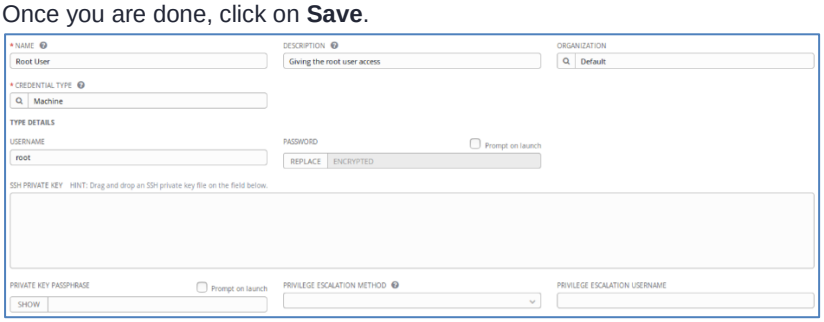
3.3. Create a Host

- To create a host, go the **Inventories** tab and choose the inventory to which you want to add hosts.
- Then choose the **Hosts** tab and click on **Add** Hosts.
- Fill the detail and click on **Save**.



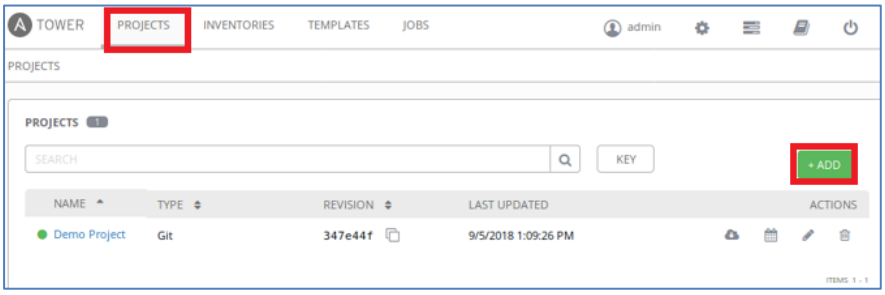
3.4. Create a credential

- Go to the **settings** options and choosing the **Credentials** tab.
- After that, go to the **Add** option and enter the details.
- Once you are done, click on **Save**.

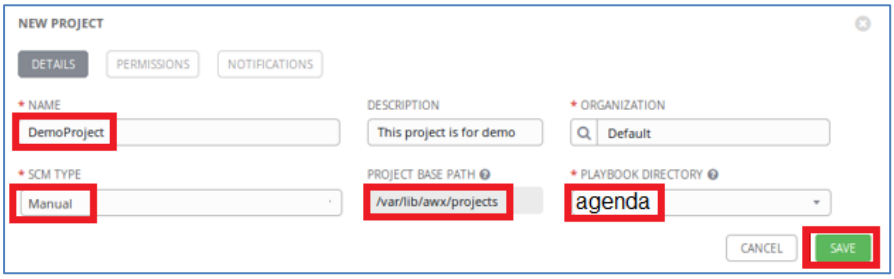


3.5. Set the Project

- Take a console login on server from **root** user and create the projects directory
  `# mkdir /var/lib/awx/projects/agenda`
- Click on the **Projects** link at the top of the Tower Dashboard
- Click on the **Add** button.

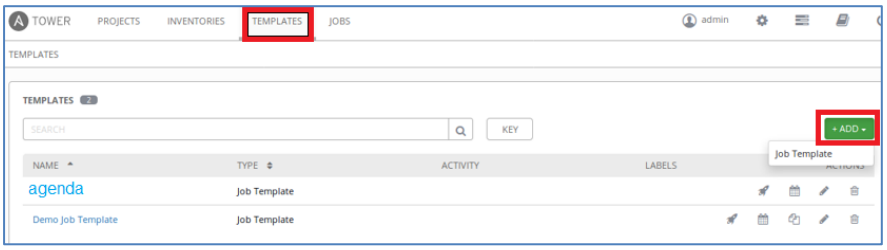


- Fill in details such as Name and Description of Project.
- Then, set the SCM type to be Manual, and for the Playbook Directory, select a value which corresponds to the subdirectory you created, then click **Save**.



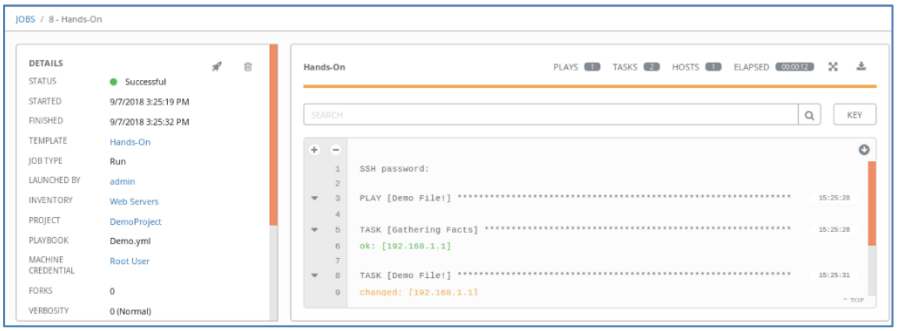
3.6. Create a Job Template

- Go to the **Job Template tab** and clicking the **Add** button.
- Fill in details such as Name, Description, Inventory name, Project, Playbooks, and Credentials.



3.7. Launch a Job

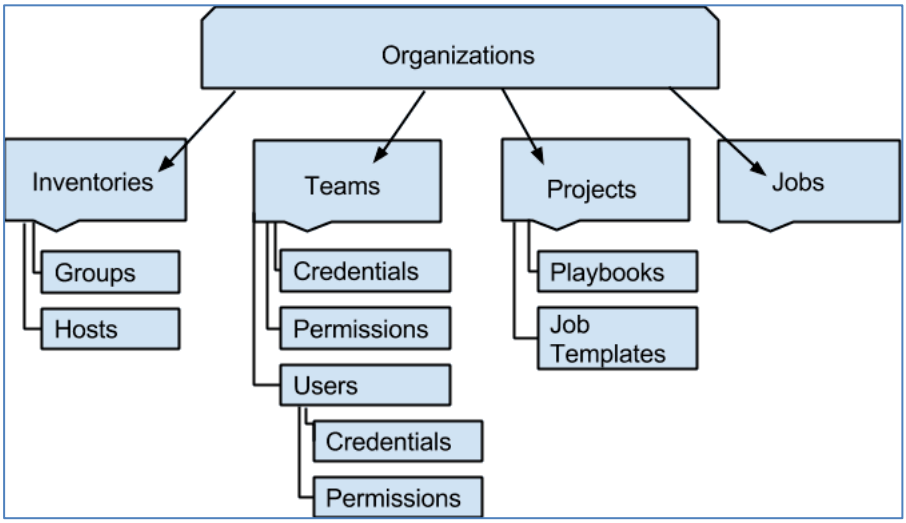
- From the **Job Templates** overview screen, click the **Launch button** (rocket symbol) to run the Job Template.



---

4. Automate Linux Patching

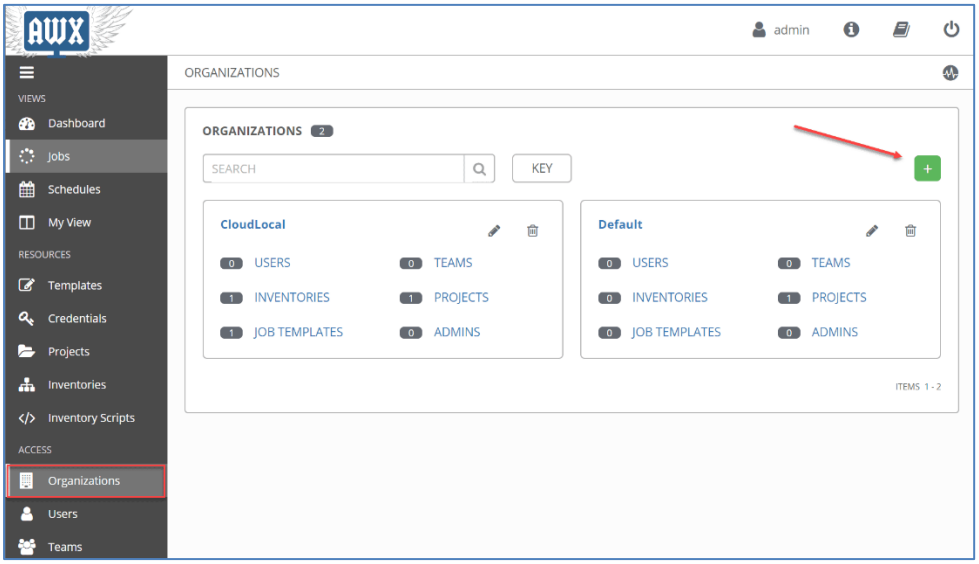
- What is an organization in Ansible Tower/AWX?
  - ➔ It is a logical collection of **Users**, **Teams**, **Projects**, and **Inventories**.



- When creating a new Job template in Ansible AWX. First things first, let's create a new **organization**.

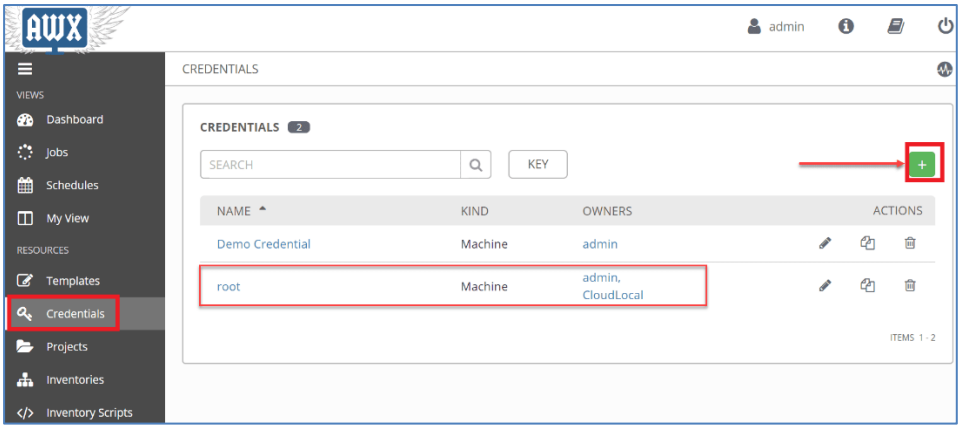
4.1. Create an organization

- Navigate to the **Organizations** node under **Access** and click the green plus sign to create a new organization.



4.2. Add the Credentials

- To connect to our Linux servers and run our job template, you will need to have credentials, Click the **Credentials** node underneath the **Resources** node and then click the **green plus sign** to add a new credential to Ansible AWX.



4.3. Create a Project.

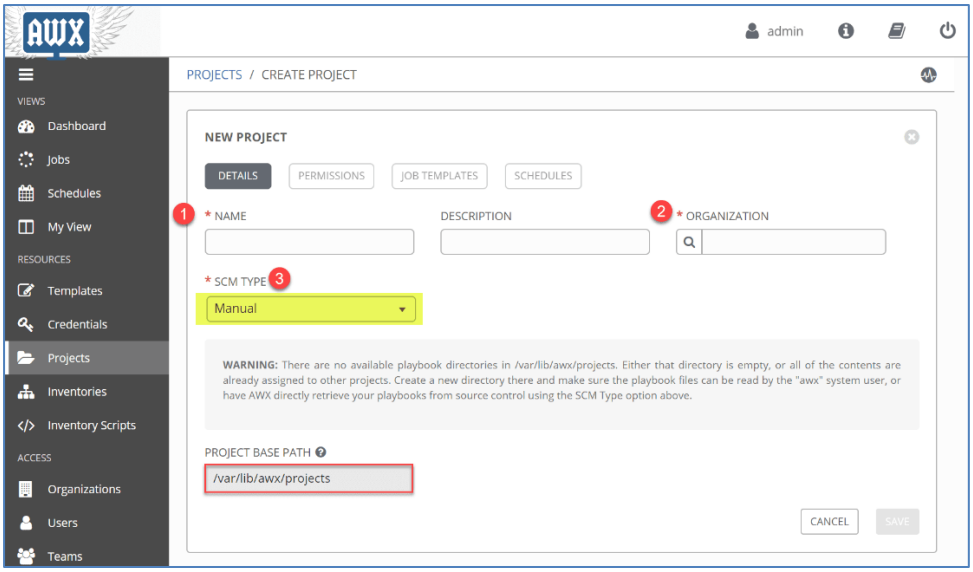
- What is a project in Tower/AWX?
  ➔ A Project is a logical collection of Ansible playbooks, represented in Tower.

- First need to manual create the project folder on server by console login.

```
# mkdir /var/lib/awx/projects/CloudUpdates
# cd /var/lib/awx/projects/CloudUpdates
# vi CloudUpdates.yml
```

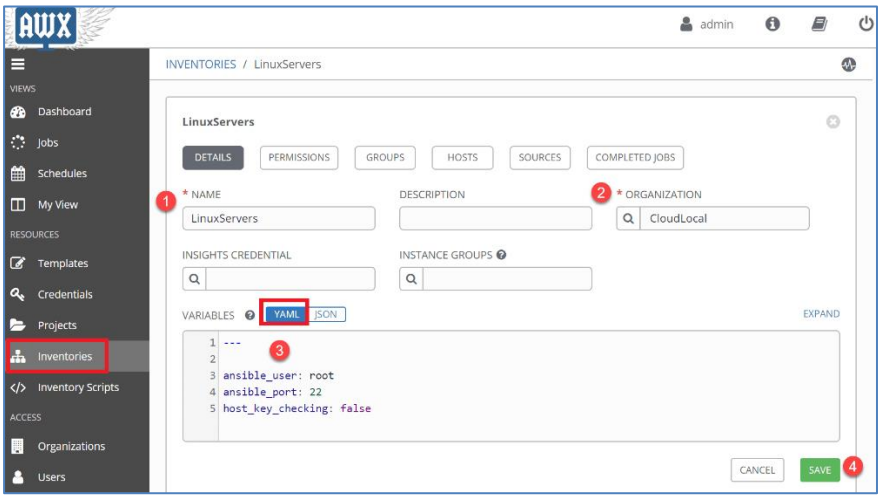
```
---
- name: Upgrade all packages to the latest version
  hosts: all
  tasks:
  - name: Upgrade all packages to the latest version
    apt:
      update_cache: yes
      upgrade: yes
```

- Click the **Project** node underneath **Resources**. -- Fill the details



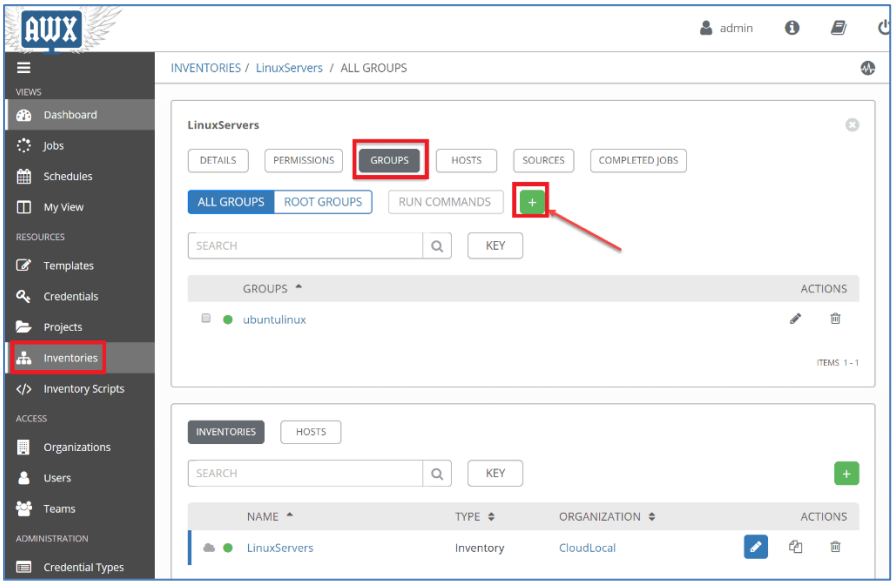
4.4. Create an Inventory

- Click the **Inventories** underneath the **Resources** → Fill the details.

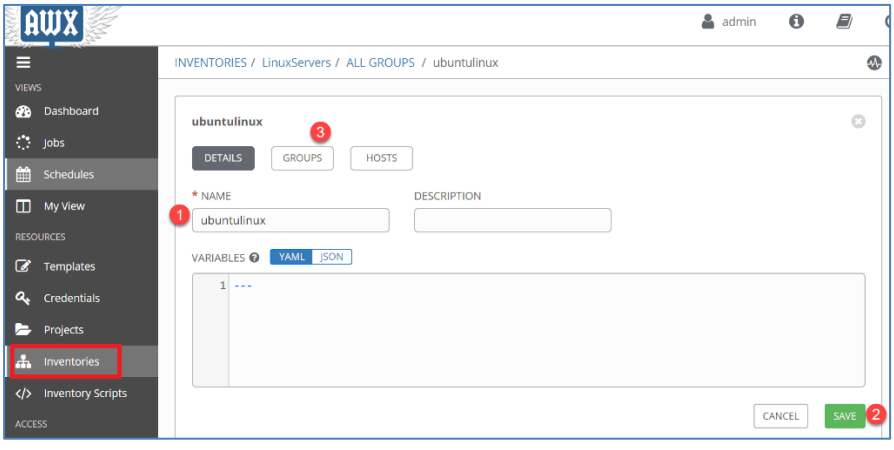


4.5. Create groups

- Click the **Groups** button and the **green plus sign** to add a new group.

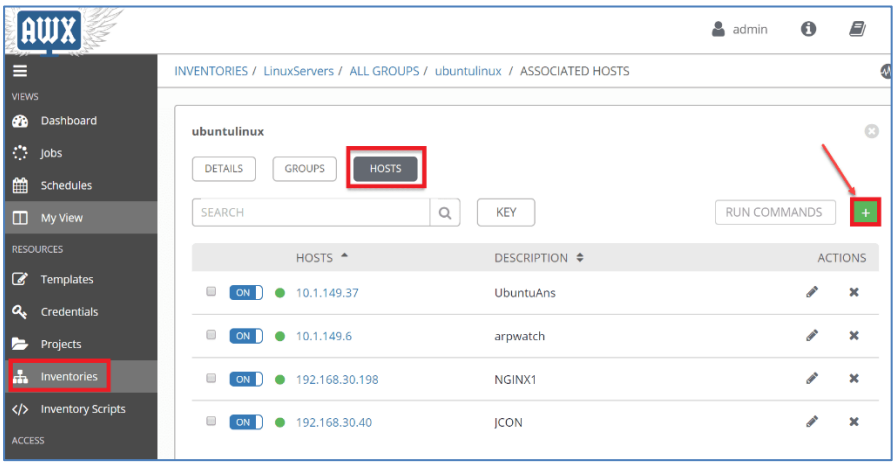


- Fill the data and click Save



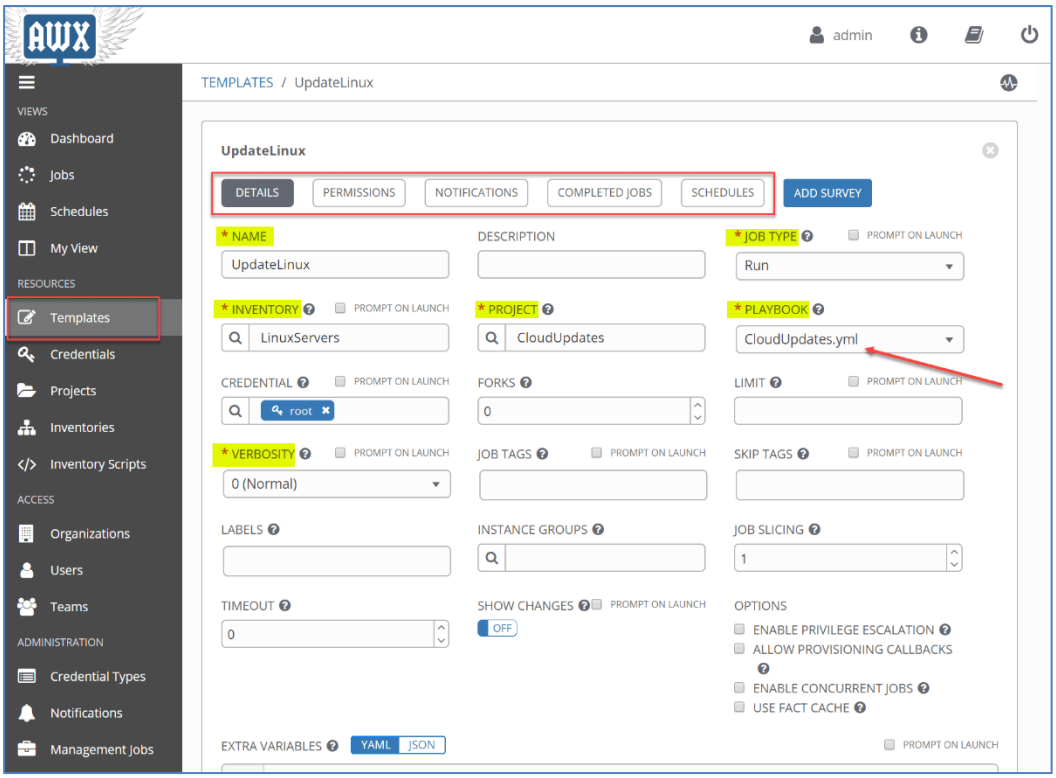
4.6. Add Hosts

- Underneath the **group**, click the **Hosts** button and click on the **green plus sign**, add the hosts.



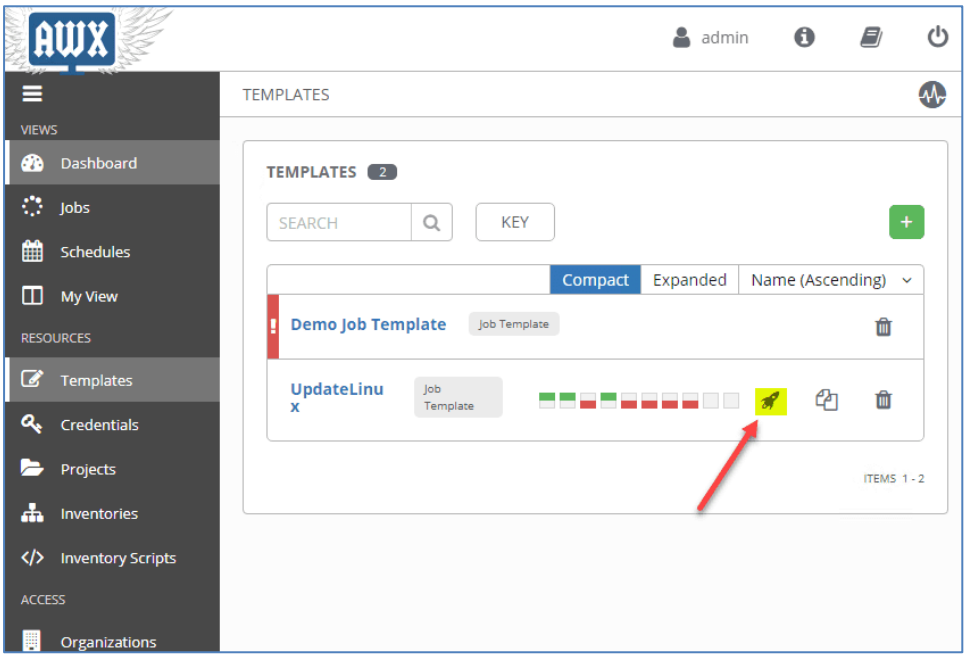
4.7. Create job template

- A job template is a definition and set of parameters for running an Ansible job.
- Underneath the **Resources**, click the **Templates** → Fill the details.

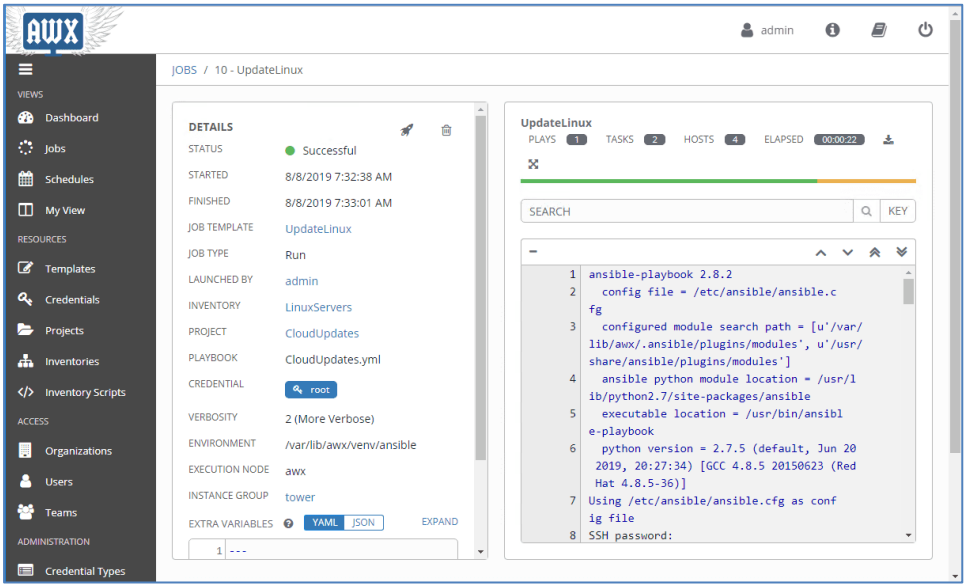


4.8. Launch a Job

- From the **Job Templates** overview screen, click the **Launch button** (**rocket** symbol) to run the Job Template.



- Below is the output.



4.9. Create Scheduler