Red Hat

Managing Ansible with Red Hat Ansible Tower

Installation and Management

Installing Red Hat Ansible Tower

Objectives

- Install Red Hat Ansible Tower in a single-server configuration.

Installation Architecture Options

Red Hat Ansible Tower can be implemented using one of the following architectures:

- Single Machine with Integrated Database
- Single Machine with Remote Database
- Multi Machine Cluster with Remote Database
- OpenShift Pod with Remote Database

Installing Red Hat Ansible Tower

- This class uses a single-node installation with integrated database
- Needs a system installed with Linux ("bare metal", virtual machine, or cloud instance)
  - Red Hat Enterprise Linux or CentOS 7.4 or later
  - 2 CPU / 4 GB RAM / at least 40 GB storage
  - On Amazon AWS EC2, can use m4.large instances

- You can get an evaluation copy of Red Hat Ansible Automation Platform (which includes Red Hat Ansible Tower) from Red Hat:
  - https://www.ansible.com/tower-trial
  - Re-download software only: https://releases.ansible.com/ansible-tower/

Installer

Two different installation packages are available for Red Hat Ansible Tower:

- Standard setup
  - https://releases.ansible.com/ansible-tower/setup/
  - Requires internet connectivity to download Red Hat Ansible Tower packages from various package repositories.

- Bundled installer
  - https://releases.ansible.com/ansible-tower/setup-bundle/
  - Includes an initial set of RPM packages for Red Hat Ansible Tower.
  - May be installed on systems disconnected from the internet.

Installing Red Hat Ansible Tower

- Downloads a compressed archive file named something like:
  - **ansible-tower-setup-bundle-3.6.2-1.tar.gz**
- Extract the archive file on your Linux system:
  - **tar zxf ansible-tower-setup-bundle-3.6.2-1.tar.gz**
- In the unpacked directory, edit the **inventory** file
  - Set your admin_password (will be used for the “admin” user to log in)
  - Set the pg_password and rabbitmq_password
- Run the install script in the unpacked directory
  - **./setup.sh**

Installing Red Hat Ansible Tower

1. As the root user, download the Red Hat Ansible Tower setup bundle.
2. Extract the Ansible Tower setup bundle and then change to the directory containing the extracted contents.

```
[root@towerhost ~]# tar xzf ansible-tower-setup-bundle-3.6.2-1.el8.tar.gz
[root@towerhost ~]# cd ansible-tower-setup-bundle-3.6.2-1.el8
```

Installing Red Hat Ansible Tower

3. Edit the inventory file to set passwords for the Ansible Tower admin account (**`admin_password`**), the PostgreSQL database user account (**`pg_password`**), and the RabbitMQ messaging user account (**`rabbitmq_password`**).

```
[tower]
localhost ansible_connection=local

[database]

[all:vars]
admin_password='myadminpassword'

pg_host=''
pg_port=''

pg_database='awx'
pg_username='awx'
pg_password='somedatabasepassword'

rabbitmq_port=5672
rabbitmq_vhost=tower
rabbitmq_username=tower
rabbitmq_password='and-a-messaging-password'
rabbitmq_cookie=cookiemonster

# Needs to be true for fqdns and ip addresses
rabbitmq_use_long_name=false
```

Installing Red Hat Ansible Tower

4.

```
[root@towerhost ansible-tower-setup-bundle-3.5.0-1.el8]# ./setup.sh
[warn] Will install bundled Ansible
...output omitted...
The setup process completed successfully.
Setup log saved to /var/log/tower/setup-2019-05-27-10:52:44.log
```

The installer will use Ansible to install and perform initial configuration of Ansible Tower.

Installing Red Hat Ansible Tower

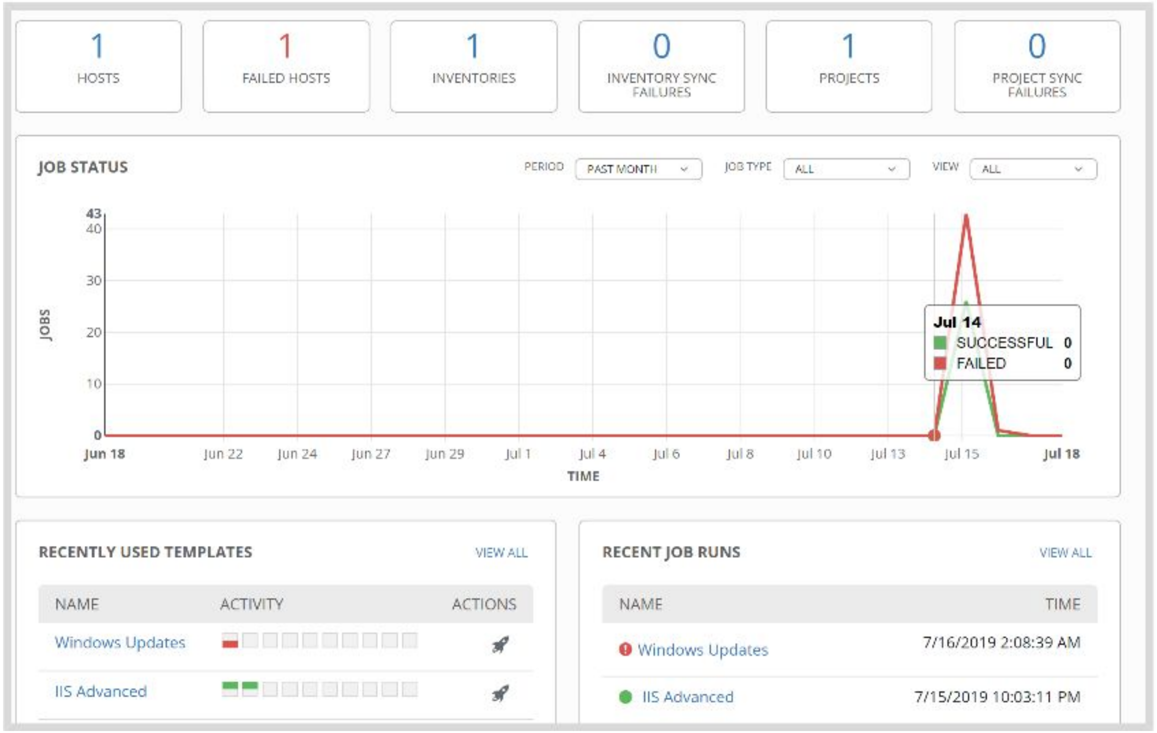
5. After the installer finishes successfully, connect to the Ansible Tower system with a web browser.

6. Login to the Ansible Tower web UI as the Ansible Tower administrator (**admin**) using the password you set for admin_password in the installer's inventory file.

7. When logging in to the Ansible Tower web UI for the first time, you are prompted to enter a license and accept the end-user license agreement. Enter the Ansible Tower license provided by Red Hat and accept the end-user license agreement.

   The Red Hat Ansible Tower Dashboard should be displayed.

Red Hat Ansible Tower Dashboard



1	1	1	0	1	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

JOB STATUS

PERIOD: PAST MONTH · JOB TYPE: ALL · VIEW: ALL

RECENTLY USED TEMPLATES — VIEW ALL

NAME	ACTIVITY	ACTIONS
Windows Updates	🟥	🚀
IIS Advanced	🟩🟩	🚀

RECENT JOB RUNS — VIEW ALL

NAME	TIME
● Windows Updates	7/16/2019 2:08:39 AM
● IIS Advanced	7/15/2019 10:03:11 PM

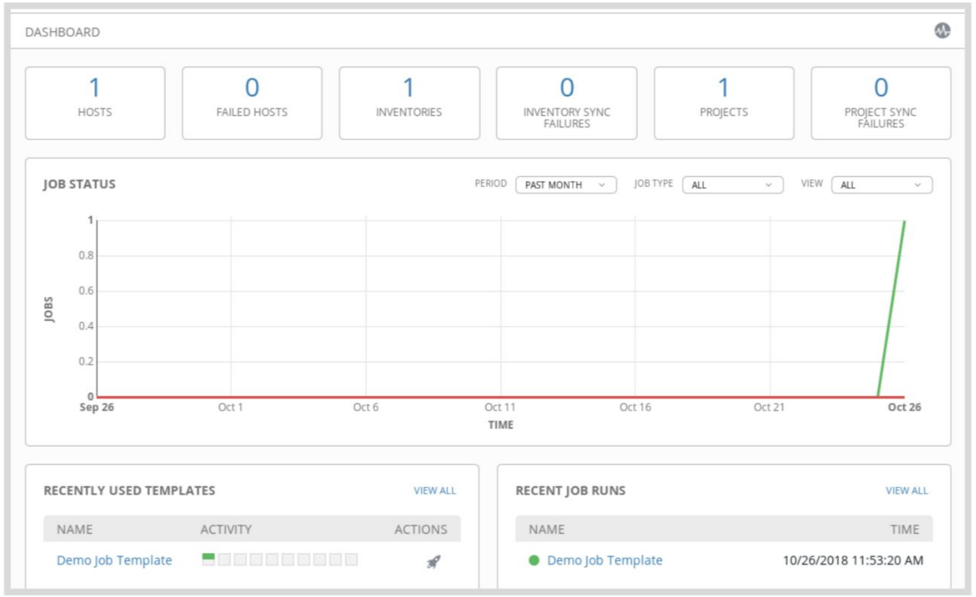
Navigating Ansible Tower’s User Interface

Objectives

- Navigate the Ansible Tower web UI.

The Ansible Tower Dashboard

- The main control center for Red Hat Ansible Tower.
- Displayed when you log in.
- Composed of four reporting sections:
  - Summary
  - Job Status
  - Recently Used Templates
  - Recent Job Runs



DASHBOARD

1	0	1	0	1	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

JOB STATUS

PERIOD: PAST MONTH · JOB TYPE: ALL · VIEW: ALL

RECENTLY USED TEMPLATES — VIEW ALL

NAME	ACTIVITY	ACTIONS
Demo Job Template	🟩	

RECENT JOB RUNS — VIEW ALL

NAME	TIME
● Demo Job Template	10/26/2018 11:53:20 AM

Navigation Bar

- A collection of navigation links to commonly used resources at the left of the web UI.

- The Ansible Tower icon takes you to the Dashboard.
- The other links provide access to the administrative page for each of the Ansible Tower resources:

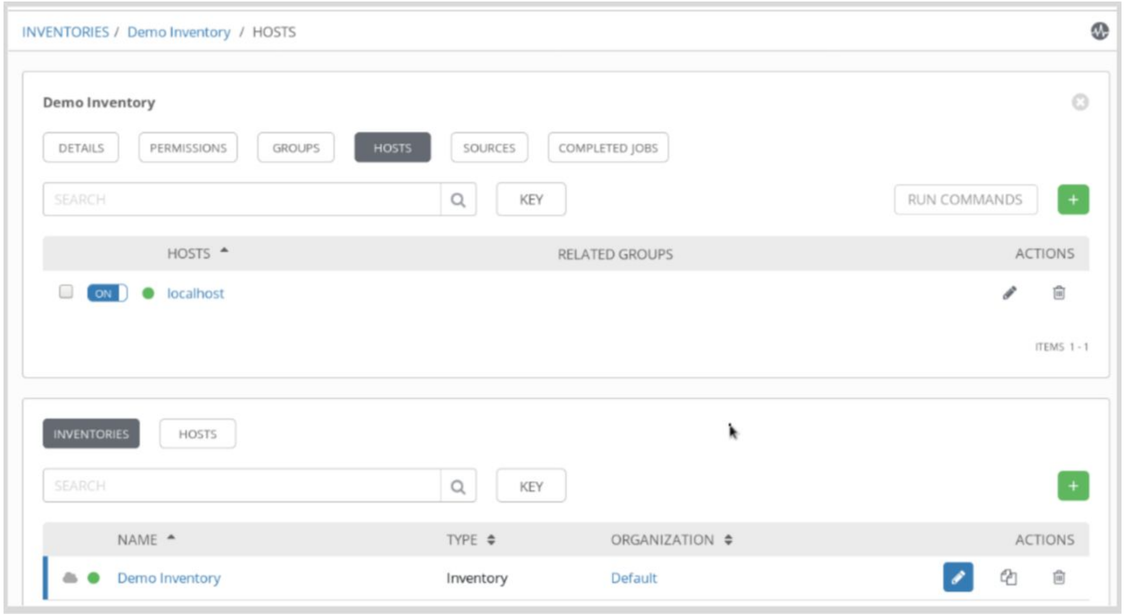
● **Jobs**	A history of previous Ansible runs.
● **Templates**	Prepared playbook settings that can be "launched" to run a job.
● **Credentials**	Authentication secrets for managed hosts and Git repositories.
● **Projects**	Sources of Ansible Playbooks (usually from a Git repo).
● **Inventories**	Inventories of managed hosts.
● **Inventory Scripts**	For use with dynamic inventories.
● **Organizations**	Used to control what resources are visible to which users.
● **Users**	User accounts.
● **Teams**	Groups of users that need to access the same resources.
● **Notifications**	You can configure notifications of job completion or failure.
● **Management Jobs**	Special jobs used to maintain the Ansible Tower installation itself.

Key Resources in Red Hat Ansible Tower

- To set up an Ansible Playbook so that you can run it, you need these basic resources:
  - An inventory of your managed hosts
  - Credentials to allow Ansible Tower to connect to and modify your managed hosts
  - A project that specifies where to get your playbook from (usually a Git repository)
  - A job template that specifies the inventory, credentials, project, and playbook from that project to run using Ansible, along with any other playbook options
- Then to run your playbook, you launch a job using that job template.
- The output from the job will be displayed in the web UI.
  - Will look like the output of **ansible-playbook** from the command line.

Demo Inventory

- Under **Inventories** in the left navigation bar, an inventory called **Demo Inventory** has been configured.
- This is a static inventory with no host groups and one host, called **`localhost`**.
- Clicking on that host in the inventory reveals that the inventory variable **`ansible_connection: local`** is set.


INVENTORIES / Demo Inventory / HOSTS

Demo Inventory

DETAILS | PERMISSIONS | GROUPS | HOSTS | SOURCES | COMPLETED JOBS

SEARCH | KEY | RUN COMMANDS | +

	HOSTS	RELATED GROUPS	ACTIONS
ON	localhost		

ITEMS 1 - 1

INVENTORIES | HOSTS

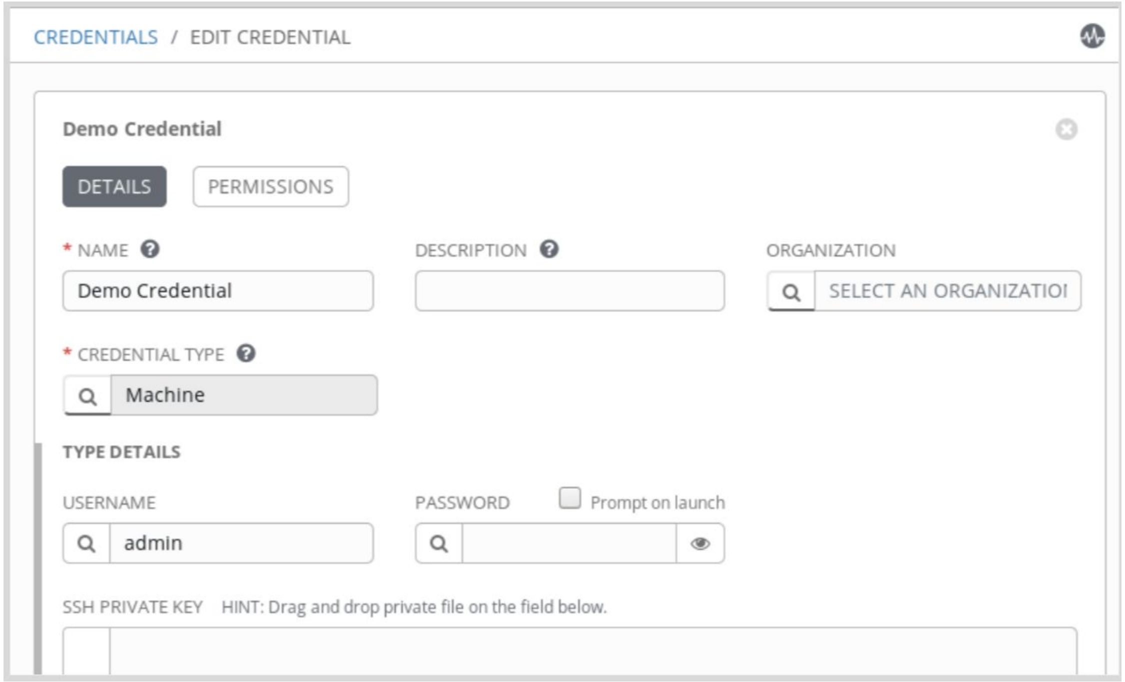
SEARCH | KEY | +

NAME	TYPE	ORGANIZATION	ACTIONS
Demo Inventory	Inventory	Default	

Demo Credential

- Under **Credentials** in the left navigation bar, a machine credential named **Demo Credential** has been created.

- This credential contains information that is used to access managed hosts in the **Demo Inventory**.



CREDENTIALS / EDIT CREDENTIAL

Demo Credential

* NAME	DESCRIPTION	ORGANIZATION
Demo Credential		SELECT AN ORGANIZATIOI
* CREDENTIAL TYPE		
Machine		

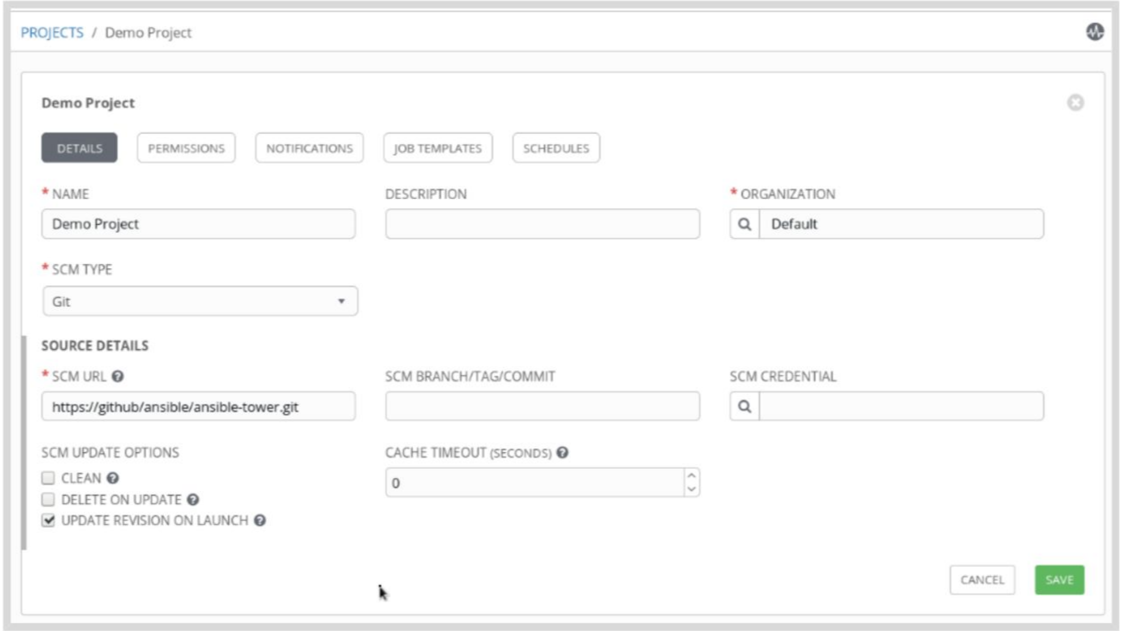
TYPE DETAILS

USERNAME	PASSWORD ☐ Prompt on launch
admin	

SSH PRIVATE KEY HINT: Drag and drop private file on the field below.

Demo Project

- Under **Projects** in the left navigation bar, a project called **Demo Project** has been configured.
- This project is configured to get Ansible project materials, including a playbook, from a public Git repository.
- You can also prepare a credential so you can access a private Git repository that needs authentication.


PROJECTS / Demo Project

Demo Project

* NAME: Demo Project

DESCRIPTION:

* ORGANIZATION: Default

* SCM TYPE: Git

SOURCE DETAILS

* SCM URL: https://github/ansible/ansible-tower.git

SCM BRANCH/TAG/COMMIT:

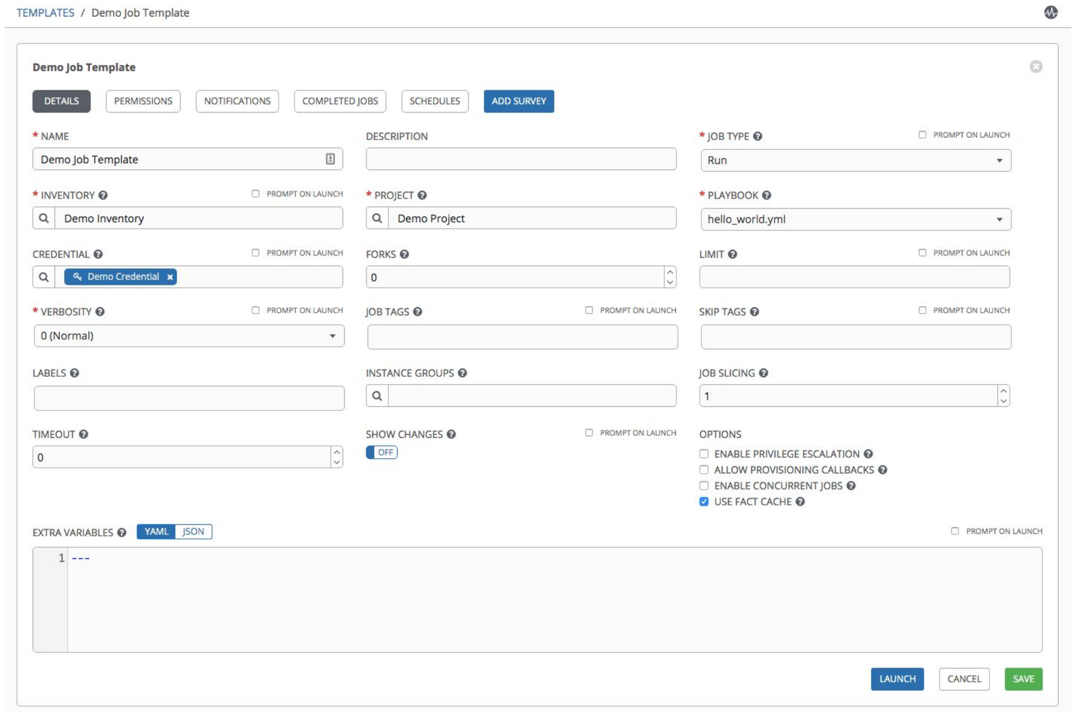
SCM CREDENTIAL:

SCM UPDATE OPTIONS
- ☐ CLEAN
- ☐ DELETE ON UPDATE
- ☑ UPDATE REVISION ON LAUNCH

CACHE TIMEOUT (SECONDS): 0

Demo Job Template

- Under **Templates** in the left navigation bar, a template called **Demo Job Template** has been configured.
- This job template runs the hello_world.yml playbook from **Demo Project** on the hosts in **Demo Inventory**, using **Demo Credential** to authenticate access.
- This initial job template can be used to test Ansible Tower.



* NAME	Demo Job Template
DESCRIPTION	
* JOB TYPE	Run
* INVENTORY	Demo Inventory
* PROJECT	Demo Project
* PLAYBOOK	hello_world.yml
CREDENTIAL	Demo Credential
FORKS	0
LIMIT	
* VERBOSITY	0 (Normal)
JOB TAGS	
SKIP TAGS	
LABELS	
INSTANCE GROUPS	
JOB SLICING	1
TIMEOUT	0
SHOW CHANGES	OFF
OPTIONS	☐ ENABLE PRIVILEGE ESCALATION ☐ ALLOW PROVISIONING CALLBACKS ☐ ENABLE CONCURRENT JOBS ☑ USE FACT CACHE
EXTRA VARIABLES (YAML / JSON)	---