Ansible Tower Quick Install

Release Ansible Tower 2.4.4

Red Hat, Inc.

Jun 06, 2017

CONTENTS

Thank you for your interest in Ansible Tower by Red Hat, the open source IT orchestration engine. Whether sharing operations tasks with your team or integrating with Ansible through the Tower REST API, Tower provides powerful tools to make your automation life easier.

The *Ansible Tower Quick Installation Guide* covers basic installation instructions for installing Ansible Tower on Red Hat Enterprise Linux, CentOS, and Ubuntu systems. This document has been updated to include information for the latest release of Ansible Tower 2.4.4.

Ansible Tower Version 2.4.4; February 22, 2016; https://access.redhat.com/

PREPARING FOR THE TOWER INSTALLATION

This guide helps you get your Ansible Tower installation up and running as quickly as possible.

At the end of the installation, using your web browser, you can access and fully utilize Tower.

1.1 Installation and Reference guide

While this guide covers the basics, you may find that you need the more detailed information available in the Installation and Reference Guide.

You should also review the General Installation Notes before starting the installation.

1.2 Prerequisites and Requirements

For platform information, refer to Supported Platforms and Requirements and Platform Specific Notes in the *Ansible Tower Installation and Reference Guide*.

Note: Tower is a full application and the installation process installs several dependencies such as PostgreSQL, Django, Apache, and others. It is required that you install Tower on a standalone VM or cloud instance and do not co-locate any other applications on that machine (beyond possible monitoring or logging software). Although Tower and Ansible are written in Python, they are not just simple Python libraries. Therefore Tower cannot be installed in a Python virtualenv, a Docker container, or any similar subsystem; you must install it as described in the installation instructions in this guide.

1.3 Primary Requirements

Ansible Tower has the following requirements:

- Supported Operating Systems:

    - Red Hat Enterprise Linux 6 64-bit

    - Red Hat 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

- An HTML5 compliant web browser
- 2 GB RAM minimum (4+ GB RAM recommended)
    - 2 GB RAM (minimum and recommended for Vagrant trial installations)
    - 4 GB RAM is recommended per 100 forks
- 20 GB hard disk
    - 10 GB of the 20 GB requirement must be dedicated to `/var/`, where Tower stores its files and working directories (dedicating less space will cause the installation to fail)
- 64-bit support required (kernel and runtime)
- For Amazon EC2:
    - Instance size of m3.medium or larger
    - An instance size of m3.xlarge or larger if there are more than 100 hosts
- For HA MongoDB setups:
    - If you plan to run MongoDB, the following guidelines provide a rough estimate for the amount of space required. The basic calculation is:

        ```
        (number of hosts in inventory) * (number of scans) *
        (average module fact size) * (number of modules scanning)
        ```
    - For example:

        hosts = 1,000

        number of scans = 365 (1 scan per day for a year)

        average module fact size = 100 kb

        number of modules = 4 (ansible, packages, services, files)

        = 146 GB

    The default scan operation has the four (4) modules listed, but you can add your own. Depending on the kinds of modules and the size of the facts you are gathering, that size might be larger.

    To help keep the size down, you can use a management job to purge old facts. Refer to Management Jobs in the *Ansible Tower Administration Guide* for more information

1.4 Ansible Software Requirements

While Ansible Tower depends on Ansible playbooks and requires the installation of the latest stable version of Ansible before installing Tower, manual installations of Ansible are no longer required.

Beginning with Ansible Tower version 2.3, the Tower installation program attempts to install Ansible as part of the installation process. Previously, Tower required manual installations of the Ansible software release package before running the Tower installation program. Now, Tower attempts to install the latest stable Ansible release package.

If performing a bundled tower installation, the installation program attempts to install Ansible (and its dependencies) from the bundle for you (refer to *Using the Bundled Tower Installation Program* for more information).

If you choose to install Ansible on your own, the Tower installation program will detect that Ansible has been installed and will not attempt to reinstall it. The latest stable version must be installed for Ansible Tower to work properly.

1.5 Additional Notes

While other operating systems may technically function, currently only the above list is supported to host an Ansible Tower installation. If you have a firm requirement to run Tower on an unsupported operating system, please contact Ansible via the Red Hat Customer Portal at https://access.redhat.com/. Management of other operating systems (nodes) is documented by the Ansible project itself and allows for a wider list.

Actual RAM requirements vary based on how many hosts Tower will manage simultaneously (which is controlled by the `forks` parameter in the job template or the system `ansible.cfg` file). To avoid possible resource conflicts, Ansible recommends 4 GB of memory per 100 forks. For example, if `forks` is set to 100, 4 GB of memory is recommended; if `forks` is set to 400, 16 GB of memory is recommended.

A larger number of hosts can of course be addressed, though if the fork number is less than the total host count, more passes across the hosts are required. These RAM limitations are avoided when using rolling updates or when using the provisioning callback system built into Tower, where each system requesting configuration enters a queue and is processed as quickly as possible; or in cases where Tower is producing or deploying images such as AMIs. All of these are great approaches to managing larger environments. For further questions, please contact Ansible Support via the Red Hat Customer Portal at https://access.redhat.com/.

The requirements for systems managed by Tower are the same as for Ansible at: http://docs.ansible.com/intro_getting_started.html

For convenience, summaries of those instructions are in the following sections.

1.6 Configuration and Installation for Ansible with Red Hat Enterprise Linux and CentOS

The following steps help you configure access to the repository as well as install Ansible.

1.6.1 Configure access to the repository for Ansible with Red Hat Enterprise Linux and CentOS (version 6 or later):

Configure the EPEL repository and any others needed.

As the root user, for Red Hat Enterprise Linux 6 and CentOS 6:

```
root@localhost:~$ yum install http://dl.fedoraproject.org/pub/epel/epel-release-
→latest-6.noarch.rpm
```

Note:

- For users of Red Hat Enterprise Linux 6, you must enable the `optional` repository.

As the root user, for Red Hat Enterprise Linux 7 and CentOS 7

```
root@localhost:~$ yum install http://dl.fedoraproject.org/pub/epel/epel-release-
→latest-7.noarch.rpm
```

Note:

- You may also need to enable the `extra` repository, named `extras` on Cen-tOS 7, `rhel-7-server-extras-rpms` on Red Hat Enterprise Linux 7, and `rhui-REGION-rhel-server-extras` when running in EC2.

- When using the official Red Hat Enterprise Linux 7 marketplace AMI, ensure that the lat-est `rh-amazon-rhui-client` package that allows enabling the optional repository (named `rhui-REGION-rhel-server-optional` in EC2) is installed.

1.6.2 Install Ansible on Red Hat Enterprise Linux and CentOS (version 6 or later):

```
root@localhost:~$ yum install ansible
```

1.7 Configuration and Installation for Ansible with Ubuntu

The following steps help you configure access to the repository as well as install Ansible.

1.7.1 Configure access to the repository for Ansible with Ubuntu 12.04 and Ubuntu 14.04:

As the root user, configure Ansible PPA:

```
root@localhost:~$ apt-get install software-properties-common
root@localhost:~$ apt-add-repository ppa:ansible/ansible
```

1.7.2 Install Ansible on Ubuntu 12.04 and Ubuntu 14.04:

```
root@localhost:~$ apt-get update
root@localhost:~$ apt-get install ansible
```

1.8 Tower Installation Scenarios

Tower installs in one of three scenarios and each of these scenarios are configurable through the Tower Installation Wizard:

- Single Machine integrated installation
- Single Machine with an external database
- High Availability Multi-Machine with an external database

For more information on these installation scenarios, refer to Tower Installation Scenarios.

TWO

DOWNLOAD THE TOWER INSTALLATION PROGRAM

Using the link provided in the email you received based on your interest in Ansible Tower, download and extract the Tower installation tarball.

Note: To obtain a trial version of Ansible Tower, visit: http://www.ansible.com/tower-trial

For pricing information, visit: http://www.ansible.com/pricing

To download the latest version of Tower directly (note, you must also obtain a license before using this), visit: https://releases.ansible.com/awx/setup/ansible-tower-setup-latest.tar.gz

---

Once extracted, `cd` into the setup directory using a command line console. In the following commands, replace the string `VERSION` with the version of Tower that you are installing (e.g., "2.4.0").

```
root@localhost:~$ tar xvzf ansible-tower-setup-latest.tar.gz
root@localhost:~$ cd ansible-tower-setup-VERSION
```

2.1 Using Vagrant/Amazon AMI Images

One easy way to try Ansible Tower is to use a Vagrant box or an Amazon EC2 instance, and launching a trial of Ansible Tower just takes a few minutes.

If you use the Vagrant box or Amazon AMI Tower images provided by Ansible, you can find the auto-generated admin password by connecting to the image and reading it from the *message of the day* (MOTD) shown at login.

2.1.1 Vagrant

For Vagrant images, use the following commands to connect:

```
$ vagrant init tower http://vms.ansible.com/ansible-tower-2.4.X-virtualbox.box
$ vagrant up
$ vagrant ssh
```

Replace `ansible-tower-2.4.X` with the version that you are trying to install. That last command provides your admin password and the Tower log-in URL. Upon login, you will receive directions on obtaining a trial license.

An up-to-date link to Ansible's Vagrant image is available from the LAUNCH TOWER IN VAGRANT section of Ansible's main website.

2.1.2 Amazon EC2

To launch the AMI, you must have an AMI ID (which varies based on you particular AWS region). A list of regions with links to AMI IDs is available in the LAUNCH TOWER IN AMAZON EC2 section of Ansible's main website.

For Amazon AMI images, use the following command to connect:

```
ssh root@<your amazon instance>
```

You must use the SSH key that you configured the instance to accept at launch time.

2.2 Using the Bundled Tower Installation Program

Beginning in Ansible Tower version 2.3.0, Tower installations can be performed using a bundled installation program. The bundled installation program is meant for customers who cannot, or would prefer not to, install Tower (and its dependencies) from online repositories. Access to Red Hat Enterprise Linux or Centos repositories is still needed.

To download the latest version of the bundled Tower installation program directly (note, you must also obtain a license before using this), visit: https://releases.ansible.com/ansible-tower/setup-bundle/

Note: The bundled installer only supports Red Hat Enterprise Linux and CentOS. Ubuntu support has not yet been added.

Next, select the installation program which matches your distribution (el6 or el7):

```
ansible-tower-setup-bundle-latest.el6.tar.gz
ansible-tower-setup-bundle-latest.el7.tar.gz
```

Note: Red Hat Enterprise Linux customers must enable the following repositories which are disabled by default:

- Red Hat Enterprise Linux 7 users must enable the extras repositories.
- Red Hat Enterprise Linux 6 users must enable the optional repository.

THE TOWER INSTALLATION WIZARD

The Tower setup process consists of two parts–a simple text-based installation wizard that determines your Tower configuration and a setup playbook that uses that information to install Tower.

The Tower Installation Wizard asks for credential information needed to access external systems where necessary.

Note: The Tower Installation Wizard and the Tower setup playbook do not need to run from the system that runs Tower, although they can.

The Tower Installation Wizard invokes as `configure` from the path where you unpacked the Tower installation tarball. It writes a file called `tower_setup_conf.yml` which contains the configuration for Tower.

```
username@localhost:~$ ./configure
```

Note: Information on which arguments the installation wizard accepts is available in the Installation Arguments section of the *Tower Installation and Reference Guide*.

For example, if you pass the `-A` or `--no-autogenerate` parameters to the Installation Wizard, you are also prompted for a PostgreSQL password and a Redis password. For more information, refer to *Passwords*.

Once invoked, the Tower Installation Wizard prompts you for configuration information.

3.1 Primary Tower machine configuration

First, the Tower wizard asks about where you intend to place the primary (or only) Tower instance.

If you are installing on the current machine, enter `localhost` or `127.0.0.1` for the current machine. If you are installing on a different machine, enter the IP address or hostname of the machine. This machine must be running and accessible via SSH when running the setup playbook later.

3.2 Database Configuration

Tower can be setup as an internal database installed on the primary Tower machine or as an external PostgreSQL database.

Enter `i` for an internal database on the same machine as Tower, or `e` for an external database. To run Tower in a high-availability configuration, you must use an external database. For more information about high availability, refer to High Availability in the Tower Administration Guide.

External database setups require additional parameters during setup.

For more information, refer to Configuring the Database in the Tower Installation and Reference Guide.

3.3 Secondary Installation

If you elected to setup an external database, you can still choose to configure any secondary Tower instances that may be necessary.

Enter the hostnames or IP addresses of any machines you want to configure as secondary Tower instances, one at a time. Enter a blank line to end the list. These machines must be running and accessible via SSH when running the setup playbook later.

3.4 Passwords

During setup, enter the administrator password. This 'admin' password is the password for the first user (and superuser) created upon installation. You must have this password for your initial login to Tower.

If you pass the `-A` or `--no-autogenerate` parameters to the Installation Wizard, you are also prompted for a PostgreSQL password and a Redis password. These passwords are internal to Tower and are not needed by the admin at runtime. The PostgreSQL and Redis passwords are normally auto-generated as a random value.

Once you have entered the required passwords, the installation wizard confirms your Tower install selection (in this case, on `localhost` with an internal database). Enter `y` to verify the installation. The installation wizard saves your configuration information to the file `tower_setup_conf.yml`. Later on, use this file to bypass the configuration step to install tower using identical configuration information and passwords.

3.5 Network Connections

If you chose to install on machines other than the current machine you are running the installation wizard on, you must also enter details on how to connect to those machines.

For more information, refer to Connection Information.

3.6 Review and Confirm

Before completing the Installation Wizard, you must review and approve your entered settings. If everything looks as expected, enter `y` to run through the setup playbook.

FOUR

THE TOWER INSTALLATION SCRIPT

4.1 The Setup Playbook

The Tower setup playbook invokes as `setup.sh` from the path where you unpacked the Tower installer tarball. It uses the `tower_setup_conf.yml` and `inventory` files written by the Tower Installation Wizard.

For information on allowable arguments, refer to The Setup Playbook in the Installation and Reference Guide.

4.2 Run the installation script

After going through the configuration wizard, invoke the installation script `setup.sh` as indicated by the setup wizard.

```
root@localhost:~$ ./setup.sh
```

If the configuration wizard suggests another command, enter this command instead.

Setup installs Tower from RPM or Deb packages using repositories hosted on **ansible.com**.

Once setup is complete, you can use your web browser to access the Tower server and view the Tower login screen. Your Tower server is accessible from port 80 (*http://tower.company.com/*).

If the installation of Tower fails, please contact Ansible via the Red Hat Customer Portal at https://access.redhat.com/.
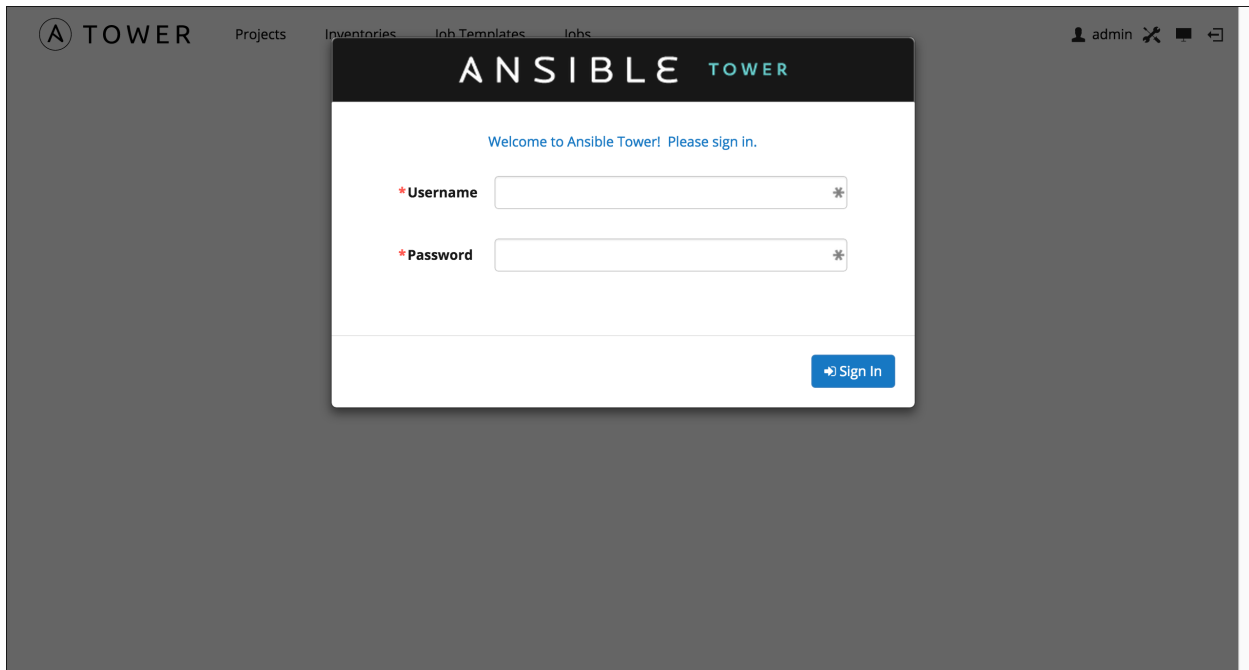
Once installed, if you log into the Tower instance via SSH, the default admin password is provided in the prompt. You can then change it with the following command (as root or as AWX user):

```
tower-manage changepassword admin
```

After that, the password you have entered will work as the admin password in the web UI.

FIVE

CONGRATULATIONS

Once the installation of Tower is complete, you are ready to set up and launch your first Ansible playbook using Tower.

If you are wondering what to do next, refer to the following list of Ansible documentation sets for information on getting started, administration, and more:

- Ansible Tower Quick Setup Guide

- Ansible Tower Installation and Reference Guide

- Ansible Tower User Guide

- Ansible Tower Administration Guide

- Ansible Tower API Guide

- http://docs.ansible.com/

INDEX

- genindex

COPYRIGHT © 2016 RED HAT, INC.