
Ansible Tower Quick Install

Release Ansible Tower 2.2.0

Ansible, Inc.

Apr 17, 2017

CONTENTS

1	Preparing for the Tower Installation	2
1.1	Installation and Reference guide	2
1.2	Prerequisites and Requirements	2
1.3	Configuration and Installation for Ansible with Red Hat Enterprise Linux and CentOS	4
1.4	Configuration and Installation for Ansible with Ubuntu	4
1.5	Tower Installation Scenarios	5
2	Download the Tower Installation Program	6
3	The Tower Installation Wizard	7
3.1	Primary Tower machine configuration	7
3.2	Database Configuration	7
3.3	Secondary Installation	8
3.4	Passwords	8
3.5	Network Connections	8
3.6	Review and Confirm	8
4	The Tower Installation Script	9
4.1	The Setup Playbook	9
4.2	Run the installation script	9
5	Congratulations	11
6	Index	12
	Index	13

Thank you for your interest in Ansible Tower, 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 2.2.0.

Ansible Tower Version 2.2.0; July 14, 2015; <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*.

Ansible Tower has the following requirements:

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 below.

- 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
- The latest stable release of Ansible
- 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
- 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:
$$\begin{aligned} &(\text{number of hosts in inventory}) * (\text{number of scans}) * \\ &(\text{average module fact size}) * (\text{number of modules scanning}) \end{aligned}$$
 - For example:
$$\begin{aligned} \text{hosts} &= 1,000 \\ \text{number of scans} &= 365 \text{ (1 scan per day for a year)} \\ \text{average module fact size} &= 100 \text{ kb} \\ \text{number of modules} &= 4 \text{ (ansible, packages, services, files)} \\ &= 146 \text{ GB} \end{aligned}$$

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

Note: Tower depends on Ansible playbooks and requires the installation of the latest stable version of Ansible before installing Tower.

Use the latest stable release of Ansible for best performance and to ensure the latest bug fixes are available. Versions of Ansible 1.8 or later are supported for Ansible Tower 2.2.

Detailed instructions on installing Ansible are available at: http://docs.ansible.com/intro_installation.html

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 as 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 via the Red Hat Customer Portal at <https://access.redhat.com/>.

Note: It is strongly recommended to use the latest stable release of Ansible for best performance and to ensure the latest bugfixes are available. However, any version of Ansible 1.8 or later is supported for Ansible Tower 2.2.

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.

1.3 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.

1.3.1 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://download.fedoraproject.org/pub/epel/6/x86_64/
↳ epel-release-6-8.noarch.rpm
```

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

```
root@localhost:~$ yum install http://download.fedoraproject.org/pub/epel/7/x86_64/e/
↳ epel-release-7-5.noarch.rpm
```

Note: You may also need to enable the “extras” repository, named “extras” on CentOS 7, “rhel-7-server-extras-rpms” on Red Hat Enterprise Linux 7, and “rhui-REGION-rhel-server-extras” when running in EC2.

Note: For users of Red Hat Enterprise Linux 7, you also need to enable the “optional” repository. When using the official Red Hat Enterprise Linux 7 marketplace AMI, be sure you install the latest “rh-amazon-rhui-client” package that allows enabling the optional repository (named “rhui-REGION-rhel-server-optional” in EC2).

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

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

1.4 Configuration and Installation for Ansible with Ubuntu

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

1.4.1 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
```

1.4.2 Install Ansible on Ubuntu 12.04 and Ubuntu 14.04:

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

1.5 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](#).

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: <http://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., “2.1.4”).

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


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 `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.

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.

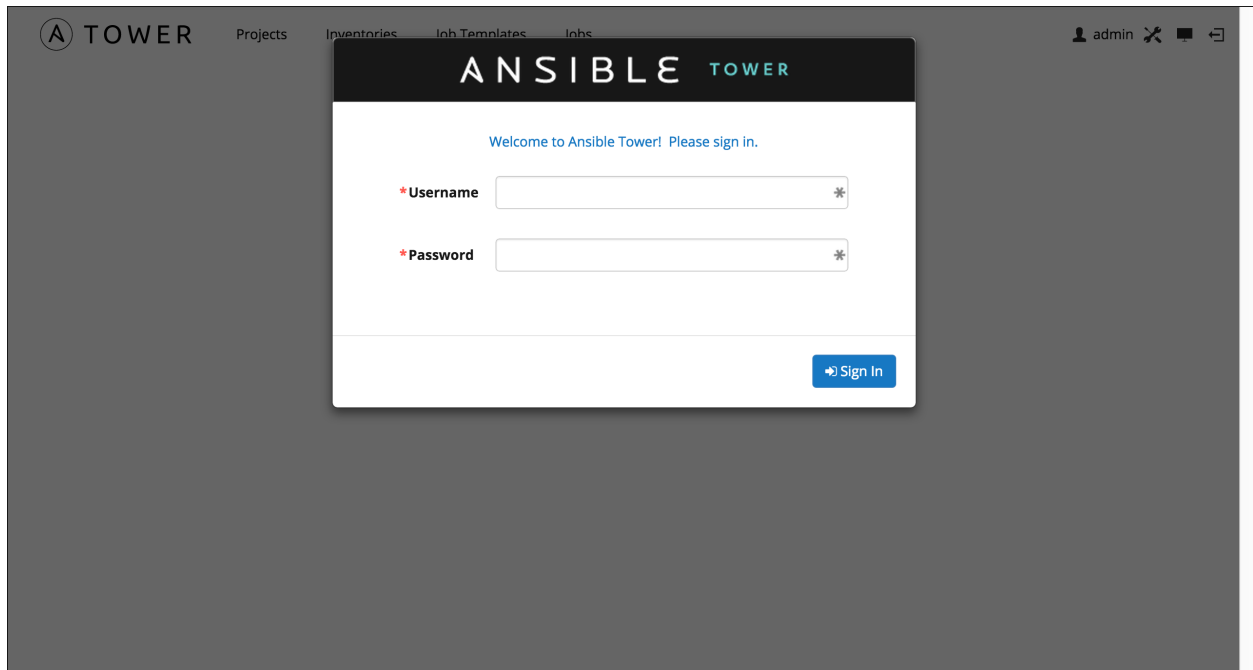
```
root@localhost:~$ ./setup.sh
```

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 us at: <http://support.ansible.com/>



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/>

- `genindex`

This document is Copyright © 2015 Ansible, Inc. All rights reserved.

Ansible and Ansible Tower are trademarks of Ansible, Inc.

If you distribute this document, or a modified version of it, you must provide attribution to Ansible, Inc. and provide a link to the original version.

Third Party Rights

Red Hat and Red Hat Enterprise Linux are trademarks of Red Hat, Inc., registered in the United States and other countries.

Ubuntu and Canonical are registered trademarks of Canonical Ltd.

The CentOS Project is copyright protected. The CentOS Marks are trademarks of Red Hat, Inc. (“Red Hat”).

Microsoft, Windows, Windows Azure, and Internet Explore are trademarks of Microsoft, Inc.

VMware is a registered trademark or trademark of VMware, Inc.

Rackspace trademarks, service marks, logos and domain names are either common-law trademarks/service marks or registered trademarks/service marks of Rackspace US, Inc., or its subsidiaries, and are protected by trademark and other laws in the United States and other countries.

Amazon Web Services”, “AWS”, “Amazon EC2”, and “EC2”, are trademarks of Amazon Web Services, Inc. or its affiliates.

OpenStack™ and OpenStack logo are trademarks of OpenStack, LLC.

Chrome™ and Google Compute Engine™ service registered trademarks of Google Inc.

Safari® is a registered trademark of Apple, Inc.

Firefox® is a registered trademark of the Mozilla Foundation.

All other trademarks are the property of their respective owners.

INDEX

A

Ansible, configure repository access, [4](#), [5](#)
Ansible, installation, [4](#), [5](#)

C

CentOS, [4](#)

D

download
 installation program, [6](#)

I

installation
 platform-specific notes, [2](#)
 scenarios, [5](#)
installation prerequisites, [2](#)
installation program
 download, [6](#)
installation requirements, [2](#)
installation script, [9](#)
 playbook setup, [9](#)
installation wizard, [7](#)
 admin password, [8](#)
 confirmation, [8](#)
 connection information, [8](#)
 database configuration, [7](#)
 high availability, [7](#)
 passwords, [8](#)
 primary machine configuration, [7](#)
 secondary installation, [8](#)

O

operating system requirements, [2](#)

P

platform-specific notes
 CentOS, [2](#)
 Red Hat Enterprise Linux, [2](#)
playbook setup, [9](#)
 setup.sh, [9](#)
prerequisites, [2](#)

R

Red Hat Enterprise Linux, [4](#)
requirements, [2](#)
requirements, Ansible, [2](#)
resources, [2](#)

U

Ubuntu, [5](#)