



# GET STARTED WITH RED HAT ANSIBLE TOWER

Follow our comprehensive guide to get you up and running with Ansible Tower in no time.

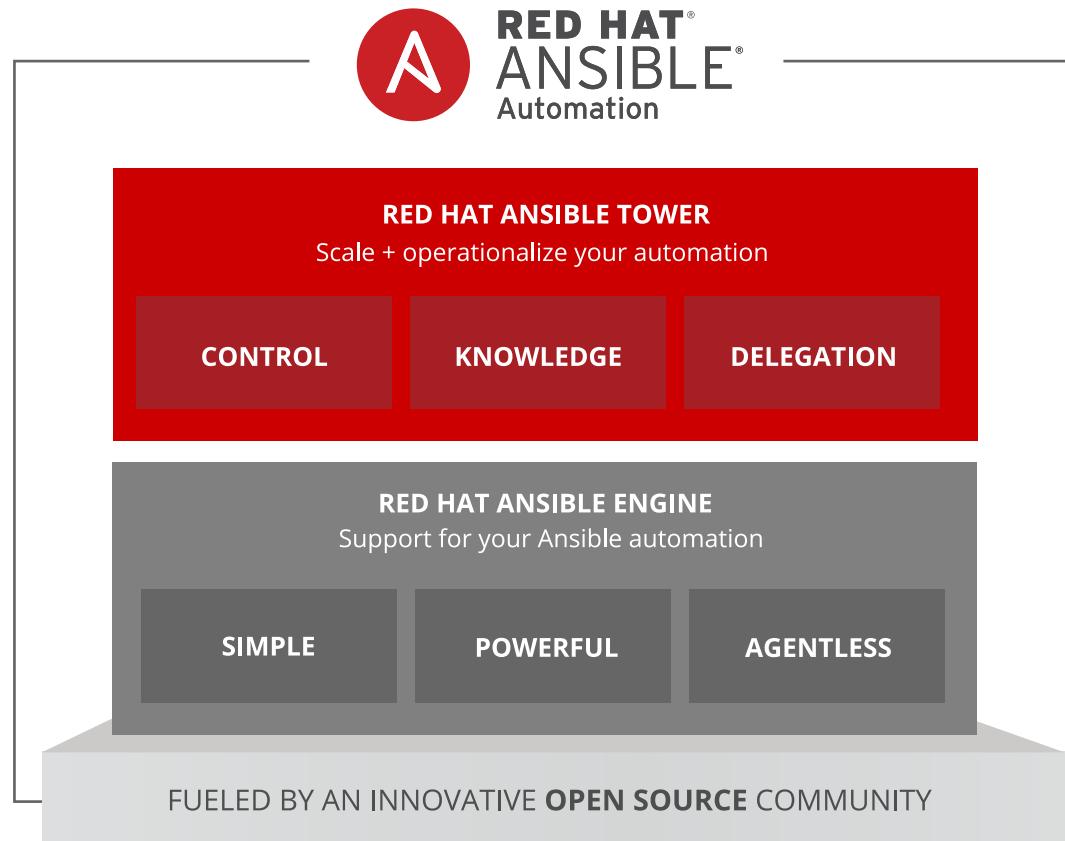
A screenshot of the Ansible Tower web interface. The left side features a dark sidebar with navigation links: Views (Dashboard, Jobs, Schedules, My View), Resources (Templates, Credentials, Projects, Inventories, Inventory Scripts), Access (Organizations, Users, Teams), Administration (Credential Types, Notifications, Management Jobs). The main area is the Dashboard. At the top, it shows summary statistics: 517 HOSTS, 0 FAILED HOSTS, 27 INVENTORIES, 1 INVENTORY SYNC FAILURES, and 23 PROJECTS. Below this is a 'JOB STATUS' chart showing the number of jobs over time (from 17:7 to 15:7). The chart has two series: a green line for active jobs (ranging from 100 to 278) and a red line for failed jobs (mostly flat near zero). Underneath the chart are sections for 'RECENTLY USED TEMPLATES' (Deploy application, Rollback, Test in development, Deploy to) and 'RECENT JOB RUNS' (Deploy application, Rollback, Test in development, Deploy to).



# Get started with the basics

Ansible is a powerful IT automation and orchestration engine that is simple enough for everyone in your IT team to use, yet powerful enough to automate even the most complex multi-tier IT applications and environments. Red Hat Ansible Engine is a fully supported product built on the foundational capabilities derived from the Ansible project.

Ansible makes automation simple. Ansible Engine gives you support, and Ansible Tower is the framework that puts you in control.



## Contents

---

[Get Started with Ansible](#)

---

[Writing a Playbook](#)

---

[Going from Ansible  
to Ansible Tower](#)

---

[Installing Ansible Tower](#)

---

[Configuring Ansible Tower](#)

---

[Ansible Galaxy](#)



## GET STARTED WITH ANSIBLE

Ansible is an IT automation engine that thousands of companies are using to drive complexity out of their environments and accelerate DevOps initiatives.

# Automation for Everyone

Ansible is designed around the way people work and the way people work together.

## **Complexity kills productivity.**

Every business is a digital business. Technology is your innovation engine, and delivering your applications faster helps you win. Historically, that required a lot of manual effort and complicated coordination.

Ansible is a simple automation language that can perfectly describe an IT application infrastructure. It's easy-to-learn, self-documenting, and doesn't require a grad-level computer science degree to read. Automation shouldn't be more complex than the tasks it's replacing.

## **Automation powered by people.**

Ansible is community powered and is the most popular open source automation tool on GitHub today with more than a quarter million downloads per month. With thousands of contributors submitting new modules all the time, rest assured that what you are automating is covered in Ansible already, or will be very soon.



## KEY TAKEAWAYS

### 1. Accelerate DevOps with Ansible.

Use Ansible to drive complexity out of your environments and accelerate DevOps initiatives.

### 2. Ansible is community powered.

Ansible is open source and created by contributors from an active open source community.

## ON-DEMAND WEBINAR

[Intro to Ansible webinar](#) 

**Are you ready to scale automation, manage complex deployments and speed up productivity?**

DOWNLOAD A FREE TOWER TRIAL



**RED HAT®  
ANSIBLE®**  
Automation



# WRITING A PLAYBOOK

Playbooks are a set of human readable instructions (plays) or commands by which automation runs within Ansible.

At a basic level, playbooks can be used to manage configurations of and deployments to remote machines. At a more advanced level, they can sequence multi-tier rollouts involving rolling updates, and can delegate actions to other hosts, interacting with monitoring servers and load balancers along the way. Once written, playbooks can be used again and again across your enterprise for easy automation.

# No Special Skills Necessary

Playbooks require no special coding skills.

Everyone in your IT organization can read an Ansible Playbook and understand it.

This simplicity means that you can start using Ansible to do real work in just minutes.

## What's in a Playbook

Playbooks start with the YAML three dashes (---) followed by:

**Name:** a brief description of the overall function of the playbook, which assists in keeping it readable and organized for all users.

**Hosts:** identifies the target(s) for Ansible to run against.

**Become:** this optional statement can be set to ‘true’/‘yes’ to activate privilege escalation (e.g., sudo, su, pfexec, doas, pbrun, dzdo, ksu, etc.).

**Tasks:** this is where actions that get executed via a call to an Ansible module get listed.

```
---
```

```
- name: Install nginx
  hosts: host.name.ip
  become: true
```

```
tasks:
```

```
- name: Add epel-release repo
  yum:
    name: epel-release
    state: present
```

```
- name: Install nginx
  yum:
    name: nginx
    state: present
```

```
- name: Insert Index Page
  template:
    src: index.html
    dest: /usr/share/nginx/html/index.html
```

```
- name: Start nginx
  service:
    name: nginx
    state: started
```

# Playbooks are Not Code

Playbooks are, at a basic level, a list of tasks.

Tasks string together to form a “desired state.” It’s this simple flow which makes a playbook readable by anybody.

The technological heavy lifting is in Ansible’s modules and plugins. Each module does one job and does it well – ensuring the desired state is met, but without doing extra work.

It’s this abstraction of the hard work that means playbooks stay accessible to non-coders and coders alike.



## KEY TAKEAWAYS

### 1. Keep your playbooks simple.

Playbooks are human readable instructions and require no special coding skills.

### 2. Playbooks are not code.

Try to keep things simple, without ‘programming’ style statements. But, gain the advantages of code-like storage in version control (using GitHub, etc).

## BLOG POST

[Writing Your First Playbook](#) ↗

**Are you ready to scale automation, manage complex deployments and speed up productivity?**

[DOWNLOAD A FREE TOWER TRIAL](#)



**RED HAT®  
ANSIBLE®  
Automation**



# GOING FROM ANSIBLE TO ANSIBLE TOWER

As organizations adopt Ansible automation technology, the requirement to govern, manage, and effectively scale IT automation increases.

Enter Ansible Tower.

# Extend the Power of Ansible with Ansible Tower

Ansible Tower is an enterprise framework for controlling, securing, and managing your Ansible automation.

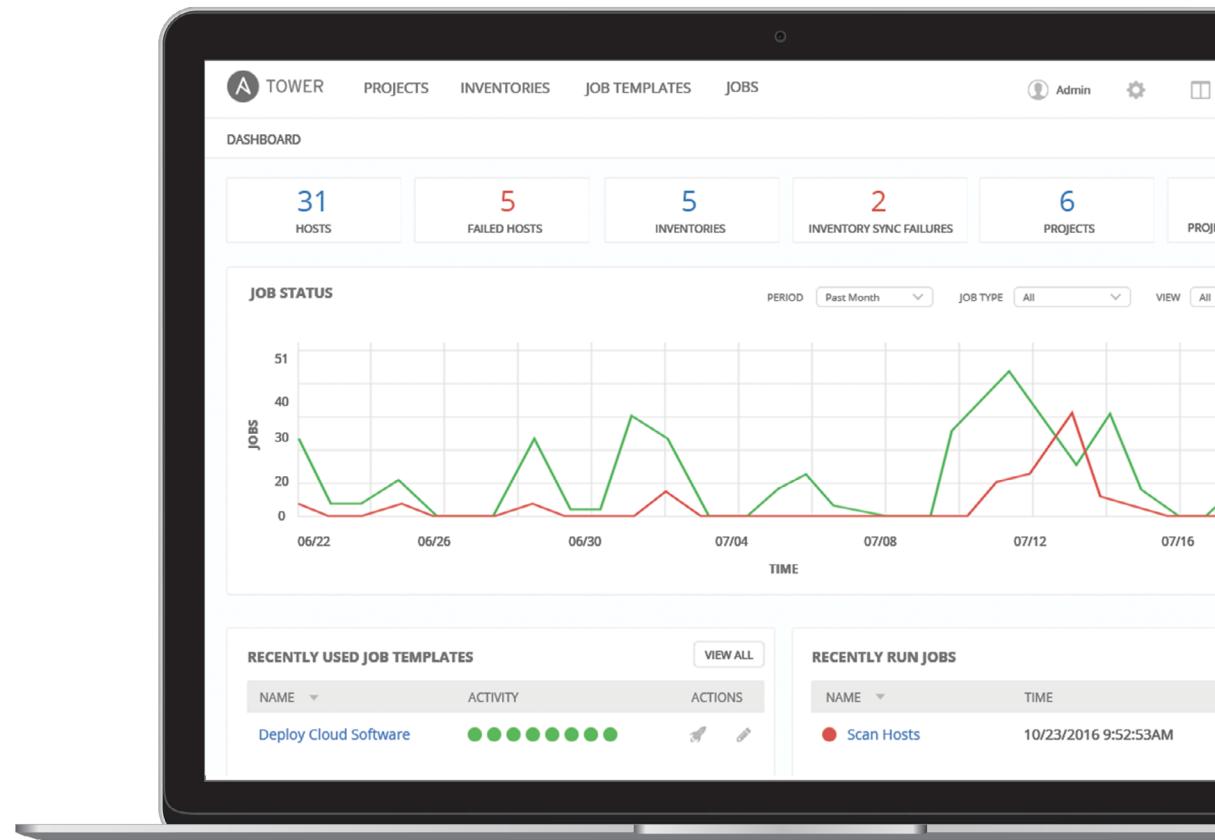
With an easy-to-use web based UI and RESTful API for managing quick deployments and monitoring configurations, Ansible Tower makes Ansible Engine even more simple for IT teams to use and is designed to be the hub for all your automation tasks.

## With Ansible Tower:

Role-based access control keeps environments secure and teams efficient.

Non-privileged users can safely deploy entire applications with push-button deployment access, even without previous Ansible knowledge.

All Ansible automations are centrally logged, ensuring auditability and compliance.



## KEY TAKEAWAYS

### 1. When you automate you accelerate.

Ansible Tower makes Ansible Engine even simpler to use with its web-based GUI and is designed to be the hub for all of your automation tasks.

### 2. Solve it. Automate it. Share it.

Ansible Tower enables easy-to-scale IT Automation, manages complex deployments and speeds productivity.

**Are you ready to scale automation, manage complex deployments and speed up productivity?**

DOWNLOAD A FREE TOWER TRIAL

## ON DEMAND WEBINAR

[Moving From Ansible CLI to Ansible Tower](#) 



**RED HAT®  
ANSIBLE®**  
Automation



# INSTALLING ANSIBLE TOWER

In keeping with our pledge of delivering simple yet powerful IT automation, installing Ansible Tower is easy.

You can have Ansible Tower installed and ready to begin automating before your coffee is brewed.

# Installation is Easy

Installing Ansible Tower can be done on a single node in four simple steps.

## Step 1:

Make sure you have the latest Ansible Tower edition.

## Step 2:

Unpack the `tar` file.

## Step 3:

Set up your passwords:

`admin_password` for Ansible Tower administration

`rabbitmq_password` for Ansible Tower messaging

`pg_password` for Ansible Tower database

**Note:** Make sure passwords do not contain special characters (! @ # \$ % ^ & \*)

## Step 4:

Run the setup script. Once installation has completed, navigate to your Ansible Tower host using Google Chrome or Mozilla Firefox by using either the hostname or IP address.

If you prefer a [clustered install](#) – where you can run more playbooks at once by sharing the load – rather than an all-in-one approach, Ansible Tower supports that, too. Clustering with Ansible Tower is quick and can be done with a few simple changes to your inventory file. Whether you are adding one extra server to make a cluster or adding three, Ansible Tower helps your team scale Ansible capacity in your organization.

For more information on [Installing an Ansible Tower Cluster](#) visit our Getting Started series.

## KEY TAKEAWAYS

### 1. Installation is easy

In the time it takes to brew your coffee, you can have Ansible Tower installed and ready to begin automating.

### 2. Customize your installation

Clustered installs are a great option for scaling out Ansible capacity.

**Are you ready to scale automation,  
manage complex deployments and  
speed up productivity?**

DOWNLOAD A FREE TOWER TRIAL

## ANSIBLE DOCUMENTATION

[Ansible Tower Quick Installation Guide](#) ↗



**RED HAT®  
ANSIBLE®  
Automation**



# CONFIGURING ANSIBLE TOWER

A blurred background image showing a desk setup. On the desk is a computer monitor displaying a graphical user interface, a keyboard, a mouse, a small potted plant, and a white mug. The overall color palette is cool and professional.

Now that you have Ansible Tower installed, you'll want to do some configuration. Much like installing Ansible Tower, you can configure your instance of Ansible Tower in a few steps.

# Set Up a User Base

First, you'll need to set up your user base and create credentials for the appropriate delegation of tasks.

**The user base can be broken up into three types:**

**1. Organization:** The top level of the user base - a logical collection of users, teams, projects and inventories.

**2. Team:** A subdivision of an organization - provides the means to set up and implement role-based access schemes as well as to delegate work across organizations.

**3. User:** Someone who has access to Ansible Tower with associated permissions and credentials.

**The user type can be broken up into three groups:**

**1. Normal User:** A user that is given no special permissions from the beginning - permissions must be granted to them by a system administrator.

**2. System Auditor:** A user who will have view-only access within Ansible Tower.

**3. System Administrator:** A user who has the power to grant and remove credentials, move users to different organizations and teams and other exclusive privileges.

# Sorting and Creating Users

The user list contains the names of everyone who has access to Ansible Tower with associated permissions and credentials. It can be sorted and searched by toggling the Username, First Name or Last Name headers.

## Step 1:

Select the *Settings* gear in the top right and select *Users*. This will take you to the users page.

**Note:** If you are starting fresh, only the “admin” account will be displayed.

## Step 2:

Select the *+Add* button and a user creation page will be displayed. From this page, you will also create their default password that can be changed by them once they login for the first time. After a password is set, you’ll set the *User Type* (System Auditor, etc.).

## Step 3:

Save the user. You can now grant the user individual credentials and add them to organizations or teams from the *Edit User* screen.

For more information on [Ansible Tower Users and Credentials](#) visit our Getting Started series.

# Creating an Ansible Tower Project

Projects are a collection of your Ansible Playbooks, commonly stored in source control. Ansible Tower lets you manage your playbook collection and assign who can run which playbooks.

## Step 1:

Click the *Project* link at the top of your Tower instance and select the *+Add* button.

## Step 2:

Upon arriving to the new project page, name the project and select which organization within your instance will be using this project.

## Step 3:

Select the source for this project, whether that be locally or through an SCM.

**Note:** The base path where projects are stored is `/var/lib/awx/projects`.

## Step 4:

Once you have declared the source, you can select *Clean*, *Delete on Update* and *Update on Launch* from the SCM update options. Once you have the correct source and you have selected the options you need, hit the save button and you will be returned to the Projects page with your new project listed.

For more information on [Ansible Tower Projects and Inventories](#) visit our Getting Started series.

# Creating a Job Template steps 1-4

Job templates are definitions and a set of parameters for running an Ansible Playbook. Job templates are linked together using a graph-like structure called nodes. Each node represents a single job template and the workflow can be set up in many ways to allow the maximization of each template.

## Step 1:

Select *Templates* from the top of the screen next to Inventories. This will bring you to the templates screen with all of your current options from which you can select. From here, navigate to the Add drop-down box and select *Job Template*.

## Step 2:

The first two options that need to be added to the template are *Name* and *Job Type*. There are two available choices:

**Check** performs a “dry run” of the playbook and reports changes that would be made without actually making them.

**Run** executes the tasks in the playbook on the selected hosts. Depending on what the playbook does, this will determine the type of job that needs to be selected.

## Step 3:

Select the inventory you want to use for your job template.

## Step 4:

Choose the project, keeping in mind that the project you select will dictate which playbook you can select.

**Note:** Ansible Tower will not display the full file name. For example, if the file name for your playbook is AWS.yml, Ansible Tower will display it as AWS.

Continue to next page to complete Creating a Job Template —>

# Creating a Job Template steps 5-7

## Step 5:

Select the credentials to save the job template. The most important option is the *Machine Credential*. The other three options are:

- *Vault*
- *Cloud*
- *Network*

Depending on what you are managing with this template, you might need either a cloud credential or a network credential.

## Step 6:

Select the *Forks*, *Limits* and *Verbosity* levels.

**Forks:** the number of parallel or simultaneous processes in use while executing the playbook.

**Limits:** host patterns to further constrain the job template's targeted hosts.

**Verbosity:** allows you to control the amount of data returned in the playbook run.

## Step 7:

You can also set a few other additional options, such as job tags and skip tags. More information about additional tags that can be set can be found in the *Job Templates* documentation page.

For more information on [Setting Up a Job Template](#) visit our Getting Started series.

## KEY TAKEAWAYS

### 1. Configuring Ansible Tower is easy

Creating users and setting up permissions using the Ansible Tower UI is straightforward and intuitive.

### 2. Organization is key

You can organize your projects visually into job templates, customizing each job to have the most efficient workflow.

## ANSIBLE DOCUMENTATION

[Configuring Ansible Tower](#) ↗

**Are you ready to scale automation,  
manage complex deployments and  
speed up productivity?**

[DOWNLOAD A FREE TOWER TRIAL](#)



**RED HAT®  
ANSIBLE®  
Automation**



# ANSIBLE GALAXY

Ansible Galaxy gives you access to thousands of user-contributed roles, playbooks, and modules.

# Galaxy Roles

A role is Ansible's way of bundling automation content and making it reusable. Instead of creating giant playbooks with hundreds of tasks, we can use roles to organize tasks, breaking them apart into smaller, more discrete units of work. A role is also all the tasks, variables and handlers needed to complete the unit of work.

## Downloading Roles

You can find the most popular roles on the Galaxy home page, or you can use the search tool to search for all available roles.

Download a role using the `ansible-galaxy` command that comes bundled with Ansible (e.g., `ansible-galaxy install username rolename`).

## Create and Share

Roles can be used for automating a wide variety of tasks, anything from the steps in your CI/CD workflow to packaging and distributing a product your company makes.

Once you finish development, push your changes to GitHub by running the following from within the project directory:

```
$ git commit -a  
$ git push
```

You will see a list of your GitHub repositories in Galaxy. If you don't see them at first, simply refresh.

## KEY TAKEAWAYS

### 1. Jump-start your automation project

Ansible Galaxy gives you access to user-contributed roles, playbooks, and modules.

### 2. Automate through sharing

Create roles and share them so other people can use them to automate a wide variety of tasks.

**Are you ready to scale automation, manage complex deployments and speed up productivity?**

DOWNLOAD A FREE TOWER TRIAL

## ANSIBLE DOCUMENTATION

[Latest Playbooks, Roles and Modules](#) 



**RED HAT®**  
**ANSIBLE®**  
Automation

# LAUNCH IT!

Now that you've used Ansible, written playbooks and installed and configured Ansible Tower, you're ready to scale your enterprise, manage complex deployments, and speed productivity.

