You want to install Ansible on CentOS 8 / RHEL 8 Linux?. Ansible is the leading Open Source configuration management system. It makes it easy for administrators and operations teams to control thousands of servers from central machine without installing agents on them.

Ansible is the simplest to use and manage when compared to other configuration management systems such as Puppet, Chef and Salt. It is easy to install, learn, and use. The only dependency required on the remote server is SSH service and Python.

**Install and Configure Ansible on CentOS 8 / RHEL 8 using the steps below.**

**Step 1: Install Python on RHEL 8 / CentOS 8**

Install and Set your default Python on RHEL 8 using the guide below.

[How to Install Python 3 / Python 2.7 on RHEL 8](https://computingforgeeks.com/how-to-install-python-3-python-2-7-on-rhel-8/)

Once it has been installed, proceed to install Pip which is a Python package manager used to install Ansible.

If you’re using Python3, install python3-pip package.

sudo dnf -y install python3-pip

sudo pip3 install --upgrade pip

For Python2 users you have to install python2-pip

sudo dnf -y install python2-pip

sudo pip2 install --upgrade pip

Step 2: Install Ansible on RHEL 8 / CentOS 8

There are two methods from which you can install Ansible on CentOS 8 / RHEL 8.

Method 1: Install Ansible on CentOS 8 / RHEL 8 from EPEL

Add EPEL repository to your CentOS 8 / RHEL 8 system.

sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm

Then Enable EPEL playground repository and install Ansible on CentOS 8 / RHEL 8 from it.

sudo dnf install --enablerepo epel-playground ansible

This will default to using Python 3, so some Python 3 packages are installed.

Dependencies resolved.

===================================================================================================================================================

Package Arch Version Repository Size

===================================================================================================================================================

Installing:

ansible noarch 2.8.5-2.epel8.playground epel-playground 15 M

Installing dependencies:

python3-jmespath noarch 0.9.0-11.el8 AppStream 45 k

python3-pyasn1 noarch 0.3.7-6.el8 AppStream 126 k

python3-bcrypt x86\_64 3.1.6-2.epel8.playground.1 epel-playground 44 k

python3-pynacl x86\_64 1.3.0-5.epel8.playground epel-playground 100 k

sshpass x86\_64 1.06-9.epel8.playground epel-playground 27 k

libsodium x86\_64 1.0.18-2.el8 epel 162 k

Installing weak dependencies:

python3-paramiko noarch 2.4.3-1.epel8.playground epel-playground 289 k

Transaction Summary

===================================================================================================================================================

Install 8 Packages

Total download size: 15 M

Installed size: 81 M

Is this ok [y/N]: y

Check the version of Ansible installed on your CentOS 8 / RHEL 8 system.

$ **ansible --version**

ansible 2.8.5

config file = /etc/ansible/ansible.cfg

configured module search path = ['/home/cloud-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python3.6/site-packages/ansible

executable location = /usr/bin/ansible

python version = 3.6.8 (default, Jul 1 2019, 16:43:04) [GCC 8.2.1 20180905 (Red Hat 8.2.1-3)]

**Method 2: Install Ansible on CentOS 8 / RHEL 8 using pip**

Once you have Pip installed, you can use it to get Ansible installed in your CentOS 8 / RHEL 8 machine.

$ pip3 install ansible --user

For Python2 pip, use:

$ pip2 install ansible --user

You can see Ansible installed using the following command:

$ **ansible --version**  
 ansible 2.7.5  
 config file = None  
 configured module search path = ['/home/jmutai/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']  
 ansible python module location = /home/jmutai/.local/lib/python3.6/site-packages/ansible  
 executable location = /home/jmutai/.local/bin/ansible  
 python version = 3.6.6 (default, Oct 16 2018, 01:53:53) [GCC 8.2.1 20180905 (Red Hat 8.2.1-3)]

**Step 3: Testing Ansible on CentOS 8 / RHEL 8 Linux**

To test Ansible, you should have OpenSSH service running on the remote server.

$ **sudo systemctl status sshd**  
 ● sshd.service - OpenSSH server daemon  
 Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)  
 Active: active (running) since Sat 2018-12-29 20:17:11 EAT; 39min ago  
 Docs: man:sshd(8)  
 man:sshd\_config(5)  
 Main PID: 820 (sshd)  
 Tasks: 1 (limit: 11510)  
 Memory: 4.6M  
 CGroup: /system.slice/sshd.service  
 └─820 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes256-cbc,aes128-gcm@openssh.com,aes128->  
 Dec 29 20:17:11 rhel8.local systemd[1]: Starting OpenSSH server daemon…  
 Dec 29 20:17:11 rhel8.local sshd[820]: Server listening on 0.0.0.0 port 22.  
 Dec 29 20:17:11 rhel8.local sshd[820]: Server listening on :: port 22.  
 Dec 29 20:17:11 rhel8.local systemd[1]: Started OpenSSH server daemon.  
 Dec 29 20:19:03 rhel8.local sshd[1499]: Accepted publickey for jmutai from 192.168.122.1 port 35902 ssh2: RSA SHA256:b/8AoYgbThoBYPcFh7CetJuGY/Tl7s4fi>  
 Dec 29 20:19:03 rhel8.local sshd[1499]: pam\_unix(sshd:session): session opened for user jmutai by (uid=0)

Create Ansible inventory file, default is /etc/ansible/hosts

I like creating inventory file in my working directory.

$ vim hosts

Copy the IP address of your remote server(s) to manage and add to Ansible inventory file.

$ echo "192.168.122.197" > hosts

You can also create a group of hosts like below:

[web]  
192.168.122.197  
  
[db]  
192.168.122.198  
  
[staging]  
192.168.122.199  
192.168.122.200  
192.168.122.201

Generate SSH key and copy it to remote servers.

$ ssh-keygen  
$ ssh-copy-id root@192.168.122.197

Use ping module to test ansible:

$ **ansible -i hosts 192.168.122.197 -m ping**    
 192.168.122.197 | SUCCESS => {  
 "changed": false,  
 "ping": "pong"  
 }

The -i option is used to provide path to inventory file. You should get the same output for hosts group name.

$ **ansible -i hosts web -m ping**   
 192.168.122.197 | SUCCESS => {  
 "changed": false,  
 "ping": "pong"  
 }

For commands that need sudo, pass the option --ask-become-pass. This will ask for privilege escalation password. This may require installation of the sshpass program.

**How to install ansible tower**

Step 1: Change directories

Change directories to /tmp

cd /tmp

Step 2: Download Red Hat Ansible Tower

Download the latest Ansible Tower package

**curl -O https://releases.ansible.com/ansible-tower/setup/ansible-tower-setup-3.7.3-1.tar.gz**

Step 3: Untar and unzip the package file

**tar xvfz /tmp/ansible-tower-setup-3.7.3-1.tar.gz**

Step 4: Change directories

Change directories into the Ansible Tower setup package

**cd /tmp/ansible-tower-setup-\*/**

Step 5: Open inventory file

Using an editor of your choice, open the inventory file

vim inventory

Step 6: Identify variables

Fill out a couple of variables in the inventory file, **admin\_password** and pg\_password. Also, fill in the **web\_server\_ssl\_cert** and **web\_server\_ssl\_key** fields, and uncomment both, as shown:

[tower]

localhost ansible\_connection=local

[database]

[all:vars]

admin\_password=**'ansibleWS' (here we have to give the password)**

pg\_host=''

pg\_port=''

pg\_database='awx'

pg\_username='awx'

pg\_password=**'ansibleWS'**

pg\_sslmode='prefer' # set to 'verify-full' for client-side enforced SSL

# Isolated Tower nodes automatically generate an RSA key for authentication;

# To disable this behavior, set this value to false

# isolated\_key\_generation=true

# SSL-related variables

# If set, this will install a custom CA certificate to the system trust store.

# custom\_ca\_cert=/path/to/ca.crt

# Certificate and key to install in nginx for the web UI and API

web\_server\_ssl\_cert=**/home/ec2-user/cert.pem (oncomment both lines)**

web\_server\_ssl\_key=**/home/ec2-user/key.pem**

# Server-side SSL settings for PostgreSQL (when we are installing it).

# postgres\_use\_ssl=False

# postgres\_ssl\_cert=/path/to/pgsql.crt

# postgres\_ssl\_key=/path/to/pgsql.key

**Step 7: Run setup**

Run the Ansible Tower setup script

**sudo ./setup.sh**

|  |  |
| --- | --- |
|  | Step 7 will take approx. 10-15 minutes to complete. This may be a good time to take a break. |

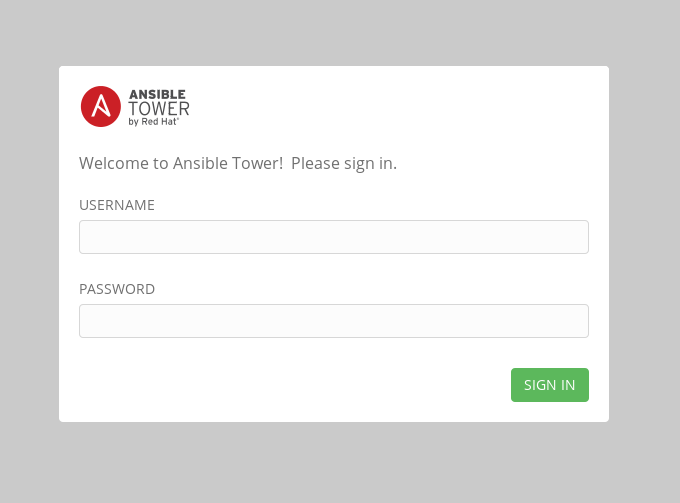
**Step 8: Confirm results**

At this point, your Ansible Tower installation should be complete. You can access your Tower workshop using the URL below:

**https://tower-0.example.rhnaps.io**

Ensuring Installation Success

You know you were successful if you are able to browse to your **Ansible Tower’s url (*control node’s IP address***) and get something like this



*Figure 2: Ansible Tower Login Screen*