**How To Install Ansible AWX on Cntos7 or RHEL7**

**First, some awx codes is not support python 2.7 anymore. So let update python to at least 3.6 version.**

**Step1:** you need to update the package manager to refresh the package index to ensure that yum repository is up to date, type the following command:

$ sudo yum update

**Step2:** you need to install SCL, which stands for [Software Collections](https://www.softwarecollections.org/en/), and is a open source project that allows you to build, install, and use multiple versions of software on the same system. To enable SCL, you need to install the **centos-release-scl** package by running the following command:

$ sudo yum install centos-release-scl

**Step3:** you can access to the SCL repository to install any python3.x version as you need on your CentOS system. Now it can be used to install the python3.6 by the following command:

$ sudo yum install rh-python36

**Step4:** Once installed, you need to launch Python36 in an new SCL shell, type:

$ scl enable rh-python36 bash

**Note:** the above command will execute the shell script **/opt/rh/rh-python36/enable**, it will change the shell environment variables.

Then you can enter the following command to check the current version of Python:

$ python -V

Outputs:

Python 3.6.3

**Now Start AWX Installation process**

**Hardware requirements**

The system that runs the Ansible AWX service will need to satisfy the following requirements:

* RAM: at leasts 4GB of memory.
* CPU: at least 2 cpu cores.
* HDD: at least 20GB of space.
* Running Docker, Openshift, or Kubernetes.
* If you choose to use an external PostgreSQL database, please note that the minimum version is 9.4.

**Install the dependency package**

**Step 1**: Disable SElinux and reboot the server.

You run the following command to replace **enforcing** with **disabled** in the config file of SElinux.

# sed -i 's|SELINUX=enforcing|SELINUX=disabled|g' /etc/selinux/config

And then reboot the server:

# reboot

**Step 2**: Install the dependency packages required for AWX.

You run the following commands in turn:

# yum -y install epel-release

# yum -y install git gcc gcc-c++ lvm2 bzip2 gettext nodejs yum-utils device-mapper-persistent-data ansible python-pip

**Step 3**: Install Docker-CE.

First, you run the command below to remove the old version of Docker on the server (if any).

# yum -y remove docker docker-common docker-selinux docker-engine

Next, you add the Docker-CE repository to the server.

# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

And now you run the command below to install Docker-CE.

# yum -y install docker-ce

After installing Docker-CE, you enable and start the docker service.

# systemctl start docker && systemctl enable docker

AWX require docker python module and you can install it via pip.

# pip install -U docker-compose

Then run the following command to check the installed version.

# pip show docker-compose

The results displayed on the screen will be similar to the following.

---

Name: docker-compose

Version: 1.25.4

Summary: Multi-container orchestration for Docker

Home-page: https://www.docker.com/

Author: Docker, Inc.

Author-email: None

License: Apache License 2.0

Location: /opt/rh/rh-python36/root/usr/lib/python3.6/site-packages

Requires: six, requests, websocket-client, docker, PyYAML, docopt, jsonschema, dockerpty, texttable, cached-property

Required-by:

**Install Ansible AWX on CentOS 7**

**Step 1**: Clone AWX from the repository.

# git clone --depth 50 https://github.com/ansible/awx.git

The result of the clone implementation will be similar to the following.

Cloning into 'awx'...

remote: Enumerating objects: 9383, done.

remote: Counting objects: 100% (9383/9383), done.

remote: Compressing objects: 100% (5148/5148), done.

remote: Total 9383 (delta 5773), reused 6229 (delta 4103), pack-reused 0

Receiving objects: 100% (9383/9383), 16.43 MiB | 7.59 MiB/s, done.

Resolving deltas: 100% (5773/5773), done.

**Step 2**: Edit file **inventory** before installation.

Move to folder awx/installer

# cd awx/installer/

Now run the command below to see what parameters are active in the inventory file.

# grep -v '^ \*#' inventory | sed '/^$/d'

Here are the parameters in the default file.

localhost ansible\_connection=local ansible\_python\_interpreter="/usr/bin/env python"

[all:vars]

dockerhub\_base=ansible

awx\_task\_hostname=awx

awx\_web\_hostname=awxweb

postgres\_data\_dir=/tmp/pgdocker

host\_port=80

docker\_compose\_dir=/var/lib/awx

pg\_username=awx

pg\_password=awxpass

pg\_database=awx

pg\_port=5432

rabbitmq\_password=awxpass

rabbitmq\_erlang\_cookie=cookiemonster

admin\_user=admin

admin\_password=password

create\_preload\_data=True

secret\_key=awxsecret

If you install AWX on the localhost (machine you are doing the command on it) then you do not care about the other parameters.

You just edit the two main parameters that are:

**Parameter 1:**admin\_password.

You run the command below to edit the parameter **admin\_password** and you change **yournewpass** using your password.

# sed -i 's|admin\_password=password|admin\_password=yournewpass|g' inventory

**Parameter 2:**secret\_key.

Next, run the following command to automatically generate a new random string, which will be used for secret\_key.

# openssl rand -base64 30

EuO3d2h8NXdb5rvfGPj0FVaZRsMV2p8dnDwY6w2Q

Then you run the command below to replace the **secret\_key** in the inventory file with the random sequence created above.

# sed -i 's|secret\_key=awxsecret|secret\_key=EuO3d2h8NXdb5rvfGPj0FVaZRsMV2p8dnDwY6w2Q|g' inventory

Now that you run the command that displays the active parameters in the inventory file again, check that the value has been replaced exactly as you did. This is just to be sure, no obligation.

# grep -v '^ \*#' inventory | sed '/^$/d'

The results:

localhost ansible\_connection=local ansible\_python\_interpreter="/usr/bin/env python"

[all:vars]

dockerhub\_base=ansible

awx\_task\_hostname=awx

awx\_web\_hostname=awxweb

postgres\_data\_dir=/tmp/pgdocker

host\_port=80

docker\_compose\_dir=/var/lib/awx

pg\_username=awx

pg\_password=awxpass

pg\_database=awx

pg\_port=5432

rabbitmq\_password=awxpass

rabbitmq\_erlang\_cookie=cookiemonster

admin\_user=admin

admin\_password=yournewpass

create\_preload\_data=True

secret\_key=EuO3d2h8NXdb5rvfGPj0FVaZRsMV2p8dnDwY6w2Q

**Step 3**: Install Ansible AWX.

After you have finished editing the inventory file, run the following command to proceed with the AWX installation.

# ansible-playbook -i inventory install.yml

The process of creating containers and settings will take about 15-20 minutes, depending on your system.

[](https://www.systemmen.com/wp-content/uploads/2018/11/install-ansible-awx-on-centos-7-01.jpg)Install Ansible AWX.

After the installation is complete, run the following command to display the created containers. If everything is OK then 5 containers are created correspondingly as below.

# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

748d34f72bd3 ansible/awx\_task:2.1.0 "/tini -- /bin/sh -c…" 2 minutes ago Up 2 minutes 8052/tcp awx\_task

a2c77e4551cc ansible/awx\_web:2.1.0 "/tini -- /bin/sh -c…" 3 minutes ago Up 3 minutes 0.0.0.0:80->8052/tcp awx\_web

529da29c4be5 memcached:alpine "docker-entrypoint.s…" 9 minutes ago Up 9 minutes 11211/tcp memcached

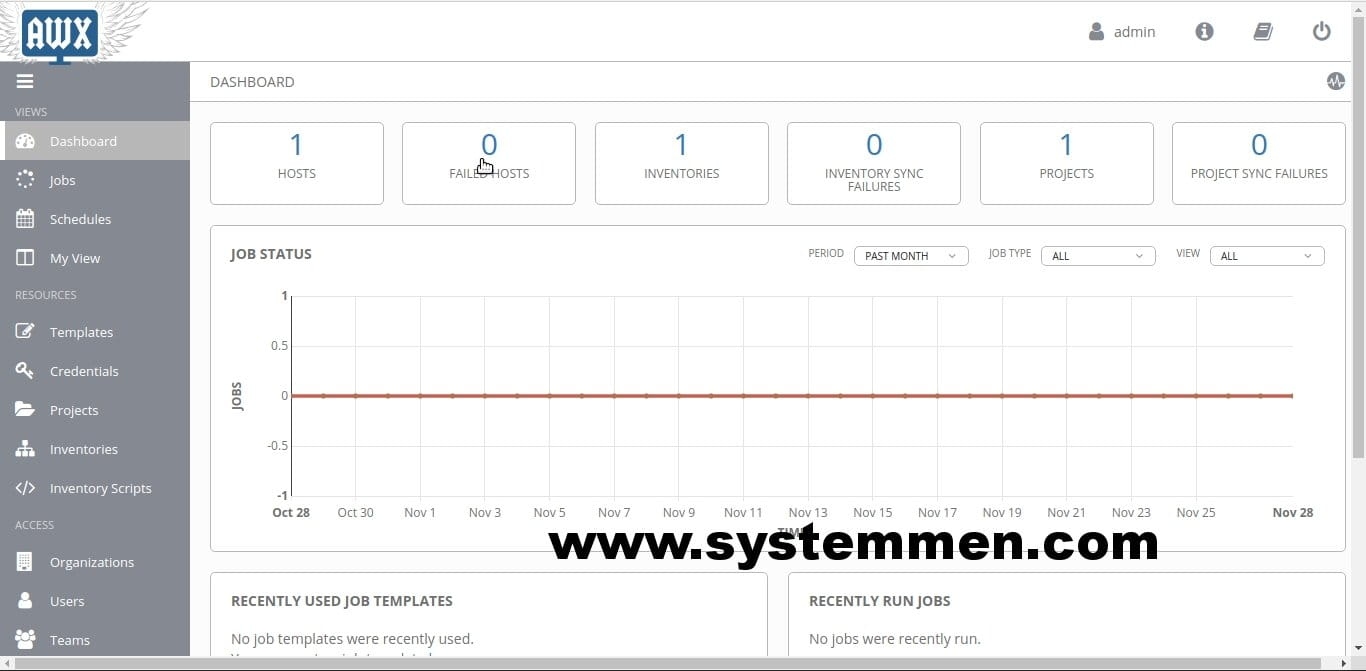
7c615c5b43b2 ansible/awx\_rabbitmq:3.7.4 "docker-entrypoint.s…" 9 minutes ago Up 9 minutes 4369/tcp, 5671-5672/tcp, 15671-15672/tcp, 25672/tcp rabbitmq

1aa8573a0416 postgres:9.6 "docker-entrypoint.s…" 11 minutes ago Up 11 minutes 5432/tcp postgres

Now open your browser and access the AWX interface with your server’s IP. Ex: http://192.168.10.10

[](https://www.systemmen.com/wp-content/uploads/2018/11/install-ansible-awx-on-centos-7-02.jpg)Access Ansible AWX login page.

Inside dashboard look like this.

[](https://www.systemmen.com/wp-content/uploads/2018/11/install-ansible-awx-on-centos-7-03.jpg)Inside Ansible AWX dashboard.