Ansible AWX Installation

Cagri Ersen
Sep 4, 2019 · 6 min read ★

As you know, automation is one of the most important aspect of the DevOPS culture. The time goes by, IT environments converted to more and more complex things and now we have more servers and network equipments that we need to manage. More importantly we need to orchestrate them without using much manpower. This is a “must have” approach to adopt DevOPS culture!

IMHO, DevOPS culture is a way to embrace “work smart, not hard” philosophy. It’s a mindset to design robust, agile and simple IT environments that produce much output with less input and a way to gaining full control over them with minimal effort!

This is why I like the clever tools that can be used to achieve to build or manage such an environment. No doubt, one of them is ansible. It’s a simple and agentless automation tool that comes with a comfortable UI like Ansible AWX (or Tower). It also has, less learning curve than others.

Anyway, Let’s focus on how to install Ansible AWX via docker-compose.

Ansible AWX

Actually AWX is an open-source version of Ansible Tower which provide a web interface to manage environments via ansible.

Here's a quick explanation from AWX project FAQ:

> *The AWX Project — AWX for short — is an open source community project, sponsored by Red Hat, that enables users to better control their Ansible project use in IT environments. AWX is the upstream project from which the Red Hat Ansible Tower offering is ultimately derived.*

So, if you're following ansible's path to manage your environments, using such a UI might be useful. Because it has some good features which mentioned in the Tower website:

> *Red Hat® Ansible® Tower helps you scale IT automation, manage complex deployments and speed productivity. Centralize and control your IT infrastructure with a visual dashboard, role-based access control, job scheduling, integrated notifications and graphical inventory management. And Ansible Tower's REST API and CLI make it easy to embed Ansible Tower into existing tools and processes.*

In particular, **job scheduling, role-based access controlling** and its **REST-API**, make it worth using.

AWX Installation

Basically, there are some options for deployment. It can be installed on docker (via docker-compose), kubernetes or openshift. But for the sake of the simplicity I'll document the installation based on docker on a latest CentOS 7 host.

Prerequisites:

In order to deploy AWX on a CentOS box, you need to have at least 4GB memory and 20GB disk space. If you decided to use AWX on production, you need to take a look Tower's requirements page for more information.

First of all let's install docker engine and docker-compose from docker repo.

```
## Install epel repo and then install jq
yum install -y epel-release -y && yum install jq

```

```
## Install docker-ce related packages
yum install -y yum-utils device-mapper-persistent-data lvm2

## Enable docker-ce repo and install docker engine.
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
yum -y install docker-ce
systemctl enable docker && systemctl start docker

## Install latest docker-compose
LATEST_VERSION=$(curl -s https://api.github.com/repos/docker/compose/releases/latest | jq -r '.
curl -L "https://github.com/docker/compose/releases/download/$LATEST_VERSION/docker-compose-$(u
chmod +x /usr/local/bin/docker-compose

## Install AWX dependencies
yum install -y python2-pip
pip install ansible
pip install docker-compose
```

gistfile1.txt hosted with ♡ by GitHub view raw

Download

Now, download the latest AWX release tarball to the home directory and extract it.

```
## Change dir to the home directory.
cd ~

## Get the latest release of ansible awx tarball and extract it.
LATEST_AWX=$(curl -s https://api.github.com/repos/ansible/awx/tags |egrep name |head -1 |awk '{
curl -L -o ansible-awx-$LATEST_AWX.tar.gz https://github.com/ansible/awx/archive/$LATEST_AWX.ta
tar xvfz ansible-awx-$LATEST_AWX.tar.gz && \
rm -f ansible-awx-$LATEST_AWX.tar.gz

## Enter awx folder.
cd awx-$LATEST_AWX
```

gistfile1.txt hosted with ♡ by GitHub view raw

Initial Configurations

The tarball includes bunch of ansible playbooks and roles to install AWX with an inventory file that holds default configuration variables. We are going to change the inventory file (*located at **installer/invetory***) in order to deploy AWX with additional configurations like https support, official logos etc.

By default, installer deploys AWX via docker-compose and uses official awx docker images which published on dockerhub to creates postgresql, rabbitmq, memcached, and a nginx based frontend container that required by AWX.

But if you want to do some additional configurations like https support; you need to build your local awx images by comment out `dockerhub_base=ansible` line as it says in the official documentation:

> *# Remove these lines if you want to run a local image build*
> *# Otherwise the setup playbook will install the official Ansible images. Versions may*
> *# be selected based on: latest, 1, 1.0, 1.0.0, 1.0.0.123*
> *# by default the base will be used to search for ansible/awx_web and ansible/awx_task*

So let's comment out that line:

```
## Disable dockerhub reference in order to build local images.
sed -i "s|^dockerhub_base=ansible|#dockerhub_base=ansible|g" installer/inventory
```

gistfile1.txt hosted with ♡ by GitHub view raw

> ***Note**: As of AWX version 6.1.0, the only way to enable https support is building local images. So, if you want to use official awx images from dockerhub, you SSL setup that we cover later in this post, cannot be done!*

As mentioned above, AWX requires a postgreSQL database and installer will automatically create a psql container for it. But in order to keep data in a persistent location, we need to create a folder to hold the db files in the host system and tell its path to the installer via `postgres_data_dir` variable.

So I'll create a folder named ***awx-psql-data*** in /opt directory and define it in the inventory.

```
## Create a folder in /opt/ to hold awx psql data
mkdir -p /opt/awx-psql-data

## Provide psql data path to installer.
sed -i "s|^postgres_data_dir.*|postgres_data_dir=/opt/awx-psql-data|g" installer/inventory
```

gistfile1.txt hosted with ♡ by GitHub view raw

> ***Note**: If you wish to use an external database, in the inventory file, set the value of* `pg_hostname`*, and update* `pg_username`*,* `pg_password`*,* `pg_database`*, and* `pg_port` *with the connection information.*

Next parameters we'll change is `ssl_certificate` which tells the installer enable https support for the frontend with the provided ssl key pair. (By default it only supports http.). With this parameter, we need to provide "a file" that should includes both a SSL certificate and its private key.

In this step, I'll create a self-signed SSL located at /etc/awx-ssl/ folder and merge the .key and .crt files in another file at */etc/awx-ssl/awx-bundled-key.crt*. Then I'll pass the file path to `ssl_certificate` variable:

```
## Create awx-ssl folder in /etc.
mkdir -p /etc/awx-ssl/

## Make a self-signed ssl certificate
openssl req -subj '/CN=secops.tech/O=Secops Tech/C=TR' \
        -new -newkey rsa:2048 \
        -sha256 -days 1365 \
        -nodes -x509 \
        -keyout /etc/awx-ssl/awx.key \
        -out /etc/awx-ssl//awx.crt

## Merge awx.key and awx.crt files
cat /etc/awx-ssl/awx.key /etc/awx-ssl/awx.crt > /etc/awx-ssl/awx-bundled-key.crt

## Pass the full path of awx-bundled-key.crt file to ssl_certificate variable in inventory.
sed -i -E "s|^#([[:space:]]?)ssl_certificate=|ssl_certificate=/etc/awx-ssl/awx-bundled-key.crt|
```

gistfile1.txt hosted with ♡ by GitHub view raw

Next changes is use of eye-candy AWX logos from awx-logos repository. I think the default logo that comes with AWX itself is horrible, so changing it might be good choice (believe me :)

In order to replace the default logo, we need to get awx-logos repository and place it to next to your awx folder (awx-6.1.0 in our example):

```
## Change dir to where awx main folder is placed:
cd ~
```

```
## Download and extract awx-logos repository.
## (We could use git to clone the repo; but it requires git to be installed on the host.)
curl -L -o awx-logos.tar.gz https://github.com/ansible/awx-logos/archive/master.tar.gz
tar xvfz awx-logos.tar.gz

## Rename awx-logos-master folder as awx-logos
mv awx-logos-master awx-logos

## Remove tarball
rm -f *awx*.tar.gz
```

gistfile1.txt hosted with ♡ by GitHub view raw

So our, working directory should be look something like below:

```
drwxrwxr-x 9 root root 4096 Jul 18 17:08 awx-6.1.0
drwxrwxr-x 3 root root 55 Sep 1 2017 awx-logos
```

> ***Note**: AWX installer which resides at **installer/install.yml**, searches for "**awx-logos**" directory as "**../../awx-logos**". So, your awx-logos folder should be located at where awx installer's parent directory is placed.*

After put the logo directory the correct location, we need to set ***awx_official*** parameter as true:

```
## Change dir to awx and replace awx_official parameter
cd ~/awx-6.1.0
sed -i -E "s|^#([[:space:]]?)awx_official=false|awx_official=true|g" installer/inventory
```

awx-logo-official hosted with ♡ by GitHub view raw

Next, we need to change ***admin_user*** and ***admin_password*** parameters. By default, installer creates a super user and you can access the AWX ui with "*admin/password*" credentials. However changing the defaults is a good habit.

So let's change them as user: ***awx-admin*** and pass: ***CHANGE_ME***

```
## Define the default admin username
sed -i "s|^admin_user=.*|admin_user=awx-admin|g" installer/inventory

## Set a password for the admin
sed -i "s|^admin_password=.*|admin_password=CHANGE_ME|g" installer/inventory
```

gistfile1.txt hosted with ♡ by GitHub view raw

> ***Note**: Set the credentials as your standards. Also there are some other credential parameters like* `secret_key` *and* `rabbitmq_password` *etc. But I don't cover all of them in this post. If you want to change them as well, you can take a look the inventory file since the comment lines explain everything nicely.*

Installation

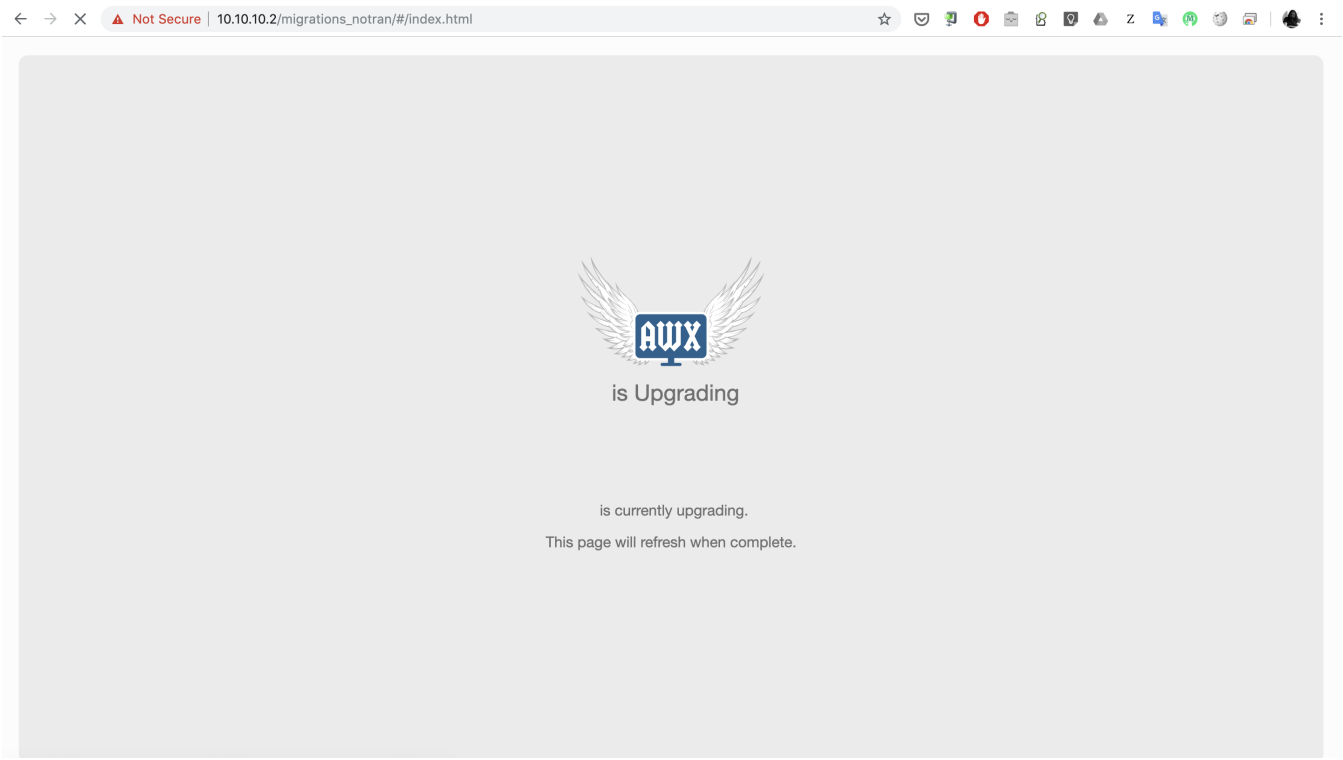
As I said before, AWX comes with a installer ansible playbooks/roles which makes the whole installation process simple. When you done with your configuration changes in the inventory file, you can just run the installer playbook like below:

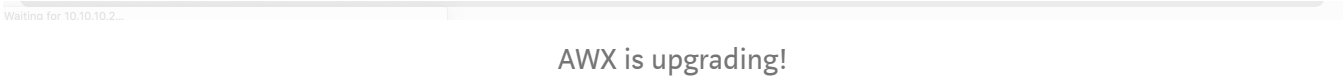
```
## Enter the installer directory.
cd ~/awx-6.1.0/installer

## Initiate install.yml
ansible-playbook -i inventory install.yml
```

gistfile1.txt hosted with ♡ by GitHub view raw

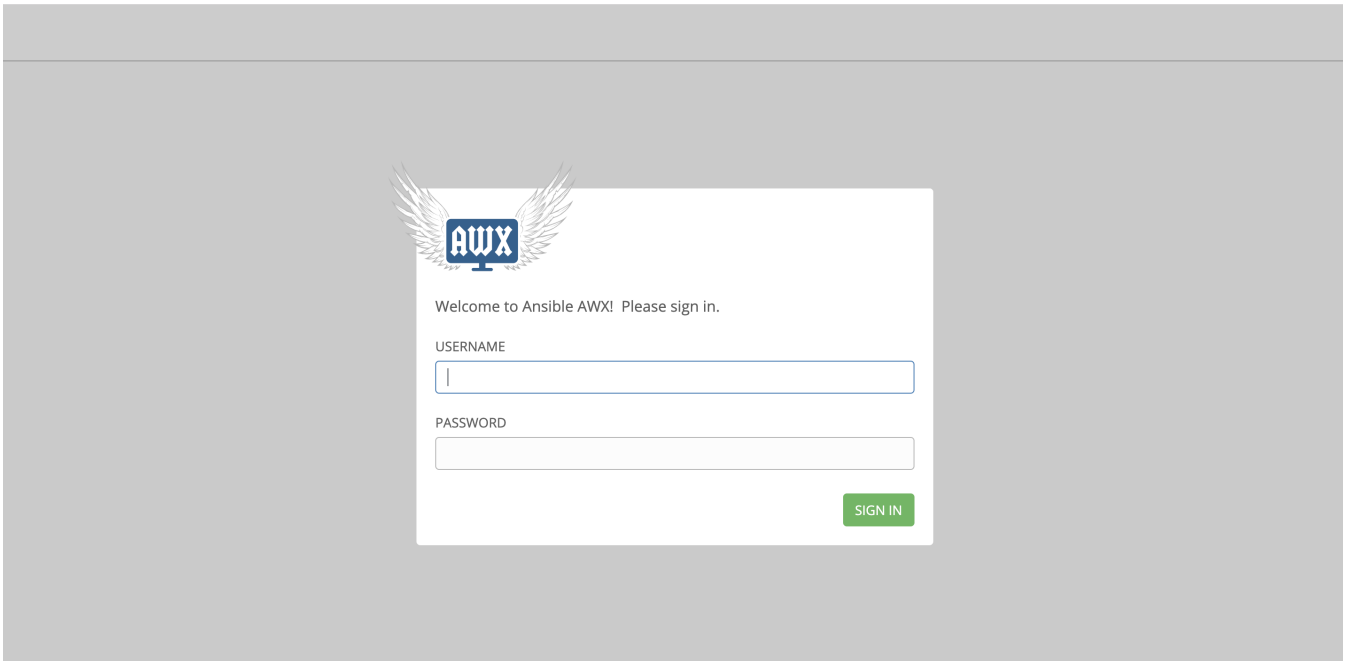
Based on compute power of your host, installation process takes a while (Since, we build local docker images for awx). But if everything goes well it should be completed successfully and you can access to AWX UI by via https://your-hosts-ip/

After initial setup, AWX upgrades itself; and when you visit the UI, you see that notice.



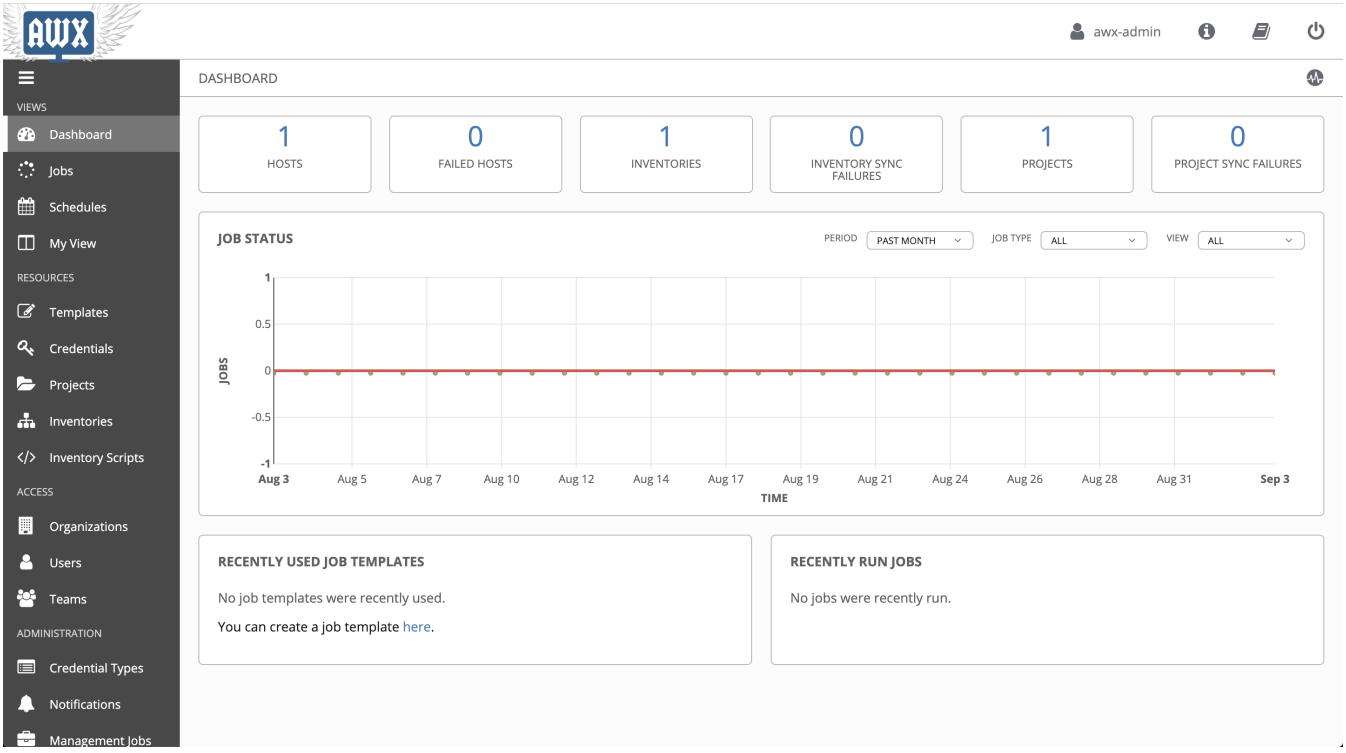


And after a while, login screen appears.



AWX login page

And AWX is ready to use:



As you noticed, there is a demo project already deployed and you can examine it as a good usage reference. But, in the next post, I'm gonna publish a document about

possible usage scenarios and best practices too.

Stay tuned.

)

Ansible Awx Tower Docker Installation