



AnsibleWorks AWX User Guide

*Version 1.4.0
November 20, 2013
awx@ansibleworks.com*

Table of Contents

Table of Contents	2
Overview	3
Licensing.....	4
Updates and Support.....	4
Requirements	4
Release Notes.....	5
Known Issues.....	6
Release History.....	6
Getting Started	7
Installation and Setup	7
Upgrade an Existing AWX Installation.....	9
Quick Start	10
User Guide	38
Logging In.....	38
Home	38
Organizations.....	39
Users.....	45
Teams.....	52
Permissions.....	60
Credentials.....	61
Projects.....	64
Inventories.....	68
Job Templates.....	79
Jobs.....	83
Best Practices.....	89
Security.....	89
Troubleshooting.....	90
Glossary.....	91
Using LDAP with AWX.....	92
Administration of AWX	92
API.....	94

Overview

AWX

AnsibleWorks AWX is a web-based user interface and REST API endpoint for Ansible, the open source IT orchestration engine. Whether sharing operations tasks with your team or integrating with Ansible through the API, AWX provides many powerful tools to make your automation life easier.

True “Push Button” Automation

Access your favorite projects and re-trigger execution from the web interface with a minimum of clicking. AWX will let you supply input variables, let you pick your credentials, will kick off and monitor the job, and allow you many great views into the results and the history of your hosts over time.

Role Based Access Control

AnsibleWorks AWX allows you to delegate specific authority to different teams or explicit users. Keep some projects private. Allow some users to edit inventory and others to run playbooks against only certain systems—either in dry run or live mode.

Cloud & Autoscaling Flexibility

AWX features a powerful callback feature that allows nodes to request configuration on demand. While optional, this is an ideal solution for an auto-scaling scenario, integrating with provisioning servers like Cobbler, or when dealing with cloud nodes with unpredictable uptimes. Requiring no software to be installed on remote nodes, the callback solution can be triggered via a simple call to ‘curl’ or ‘wget’, and is easily embeddable in init scripts, kickstarts, or preseeds.

The Ideal RESTful API

The AWX REST API is the ideal RESTful API for a systems management application, with all resources fully discoverable, paginated, searchable, and well modeled. A styled API browser allows API exploration from the API root (<http://servername/api>), showing off every resource and relation. Everything that can be done in the user interface can be done in the API—and more.

Licensing

AWX is a proprietary software product and is licensed on an annual subscription basis. There is no license fee for managing up to 10 hosts. Should you wish to acquire a license for additional servers or get support for the ones you have, please visit <http://ansibleworks.com/ansibleworks-awx> for details, <https://store.ansibleworks.com> to manage licenses, or contact awx@ansibleworks.com for assistance.

Ansible is an open source software project and is licensed under the GNU General Public License version 3, as detailed in the Ansible source code:

<https://github.com/ansible/ansible/blob-devel/COPYING>

This document is Copyright © 2013 AnsibleWorks, Inc. All rights reserved.

Updates and Support

AWX is licensed as an annual subscription, which includes:

- Standard or Premium (24x7) Support via web, email, and telephone with SLA
- All regular updates and releases of AWX and Ansible

For more information, please contact AnsibleWorks at awx@ansibleworks.com or at <http://www.ansibleworks.com/ansible-subscriptions/>.

Requirements

AnsibleWorks AWX has the following minimum requirements:

- Supported Operating Systems:
 - Red Hat Enterprise Linux 6
 - CentOS 6
 - Ubuntu 12.04 LTS
- Ansible 1.2.2
- 2 GB RAM
- 20 GB hard disk

While other operating systems may technically function, currently only this list are supported by AnsibleWorks. If you have a firm requirement to run the AWX server on an unsupported operating system, please contact awx@ansibleworks.com for further information.

NOTE: For users of Red Hat Enterprise Linux or CentOS systems, SELinux can be set to

disabled, permissive, or enforcing, but is only supported in “targeted” mode.

NOTE: Although AWX and Ansible are written in Python, they are full applications and not a simple Python library. Therefore AWX cannot be installed in a Python virtualenv or similar; you must install it as described in the installation instructions below.

NOTE: It is recommended to use Ansible version 1.4 or greater, for improved performance. However, Ansible version 1.2 is supported for AWX 1.4.

The requirements for systems *managed* by AWX are the same as for Ansible at:

<http://www.ansibleworks.com/docs/gettingstarted.html>

Release Notes

- Changes from 1.3.1
 - Added new Home tab with dashboard view of job and host status
 - Added user interface for inventory synchronization with Amazon Web Services and Rackspace Cloud Servers. Configure this in the groups editor of the Inventories tab.
 - Moved all credentials to the Credentials tab, including SSH, SCM, and cloud management. You can now create and manage all credentials from the Credentials tab. Previously, credentials were owned by a project, not a particular user. Any existing SCM synchronization jobs will be migrated such that the credentials will be owned by the admin user. If you find you can no longer synchronize SCM-based projects, please review the credentials assigned to the admin user and change their ownership to the appropriate team.
 - SCM integration dialogs are simplified
 - The hosts and groups pages have a more unified the look and feel
 - Various pages have new red and green light icons to indicate status
 - Added the Activity Stream as a beta feature available only to admin users in this release. The activity stream is accessed by an icon in the top right of most screens and shows what actions have been performed and by which users.
- Changes from 1.2.2
 - Added integration with Source Code Management systems for importing and managing AWX project playbooks
 - Added integration with LDAP and Active Directory for AWX user management. Please, see the section [Using LDAP with AWX](#) for more information.
 - The inventory display has been revised to improve the user experience

Known Issues

1. AnsibleWorks AWX implements a role based access control system. You may appear to be able to edit objects that do not belong to you (like being able to pull up an edit dialog on your team mates whom you already have permission to view). Don't worry, when you try to edit something, you'll get a 403 error, and you can't see any information you shouldn't already have access to as defined in the system.
2. This version of AWX does not support scheduling jobs from the UI, only triggering them. You may schedule jobs via a cron script via the AWX API, or using the AWX cli found at github.com/ansible/awx-cli.

Release History

The release history for this documentation is as follows:

Version	Date	Changes
1.4.0	November 21, 2013	Updated for release of AWX 1.4
1.3.0	September 13, 2013	Updated for release of AWX 1.3
1.2.2	July 31, 2013	Initial Release

Getting Started

Welcome to AnsibleWorks AWX!

To get started, first follow the installation instructions in the section entitled [Installation and Setup](#). Then, either walk through the quick start below to quickly get up and running with AWX or browse through the documentation and design an implementation plan that works for you.

We value your feedback. Please contact us at awx@ansibleworks.com and let us know what you think and your ideas for future features!

Installation and Setup

You can expect the installation of AWX to take less than fifteen minutes, depending on the speed of your network connection. (This installation will require that the AWX server be able to access the Internet.)

At the end of the installation, you will use your web browser to access AWX and utilize all of its capabilities.

NOTE: Although AWX and Ansible are written in Python, they are full applications and not a simple Python library. Therefore AWX cannot be installed in a Python virtualenv or similar; you must install it as described in the installation instructions below.

1. Install Ansible 1.3.x or later as detailed in the Ansible documentation at:
<http://www.ansibleworks.com/docs/gettingstarted.html#id3>.

For convenience, we'll summarize those installation instructions here:

- a. For Red Hat Enterprise Linux 6 and CentOS 6:
 - i. Configure the EPEL repository

```
[root@localhost ~]# yum install http://mirror.oss.ou.edu/epel/6/x86\_64/epel-release-6-8.noarch.rpm
```

- i. Install Ansible

```
[root@localhost ~]# yum install ansible
```

b. For Ubuntu 12.04:

i. Install Ansible dependencies

```
[root@localhost ~]# apt-get install python-yaml python-paramiko python-jinja2 python-pip
```

ii. Install Ansible

```
[root@localhost ~]# pip install ansible
```

2. Download the AWX installer tarball from:

<http://ansibleworks.com/releases/awx/setup/awx-setup-latest.tar.gz>

3. Extract the tarball and cd into the setup directory. Replace the string “VERSION” in the commands below with the version of AWX that you are installing e.g., “1.4.0”.

```
[root@localhost ~]# tar xvzf awx-setup-latest.tar.gz  
[root@localhost ~]# cd awx-setup-VERSION
```

4. Edit the file “group_vars/all”. Modify the variable “pg_password” to change the default database password.

NOTE: The password should not contain quotes.

5. If you wish to setup LDAP / Active Directory authentication for AWX, please review the section [Using LDAP with AWX](#).

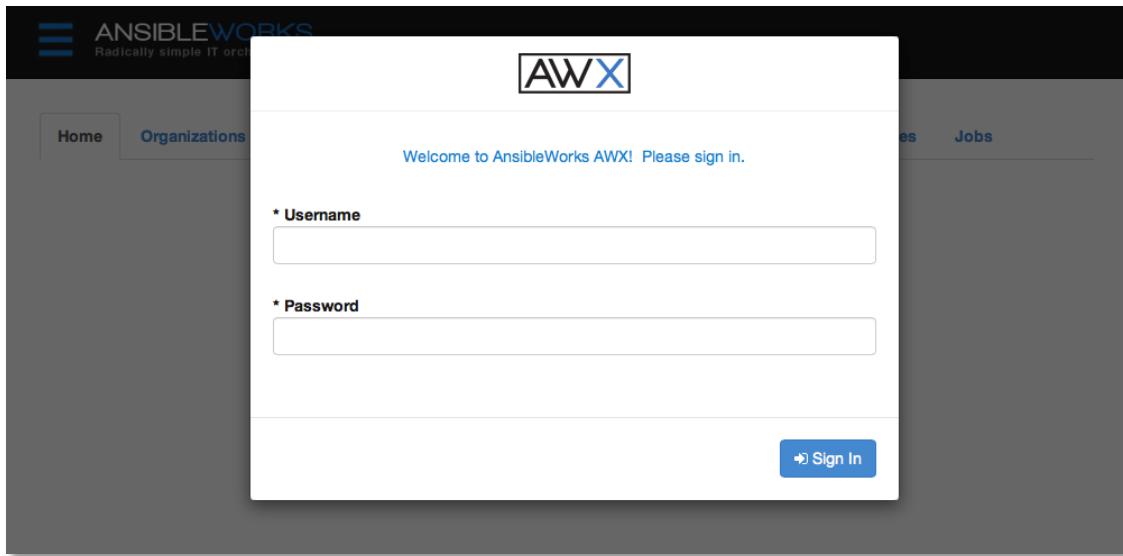
6. From the awx-setup-VERSION directory, run setup.sh

NOTE: For users of Red Hat Enterprise Linux or CentOS, PackageKit can frequently interfere with the update mechanism. Consider disabling or removing PackageKit if installed prior to running the setup process.

```
[root@localhost ~]# ./setup.sh
```

- a. Setup will install AWX from RPM or Deb packages using repos hosted on AnsibleWorks.com.

When setup completes successfully, you should be able to point your web browser to the AWX server and see the AWX login screen.



If the installation of AWX fails or if you need assistance, please contact us at awx@ansibleworks.com. AnsibleWorks subscription customers will receive a faster response by filing a support issue.

Upgrade an Existing AWX Installation

You can upgrade your existing AWX installation the latest version by running the setup playbook for the new version of AWX. All data will be preserved. However, it is important that if you changed any of the parameters in the file “group_vars/all” (e.g. “pg_password”) that you make the same changes to the new “group_vars/all” file.

You can expect the upgrade of AWX to take less than fifteen minutes, depending on the speed of your network connection. (This installation will require that the AWX server be able to access the Internet.)

At the end of the upgrade, you will use your web browser to access the AWX server and utilize all of its capabilities.

This upgrade procedure assumes that you have a working installation of Ansible and AWX.

1. Backup the existing AWX database.

```
[root@localhost ~]# awx-manage dumpdata > backup.json
```

2. Download the AWX installer tarball from:

<http://ansibleworks.com/releases/awx/setup/awx-setup-latest.tar.gz>

3. Extract the tarball and cd into the setup directory. Replace “VERSION” with the version of AWX you are installing e.g., “1.4.0”.

```
[root@localhost ~]# tar xvzf awx-setup-latest.tar.gz
[root@localhost ~]# cd awx-setup-VERSION
```

4. Edit the file “group_vars/all”. Modify the variable “pg_password” to change the default database password.
5. If you wish to setup LDAP / Active Directory authentication for AWX, please review the section [Using LDAP with AWX](#).
6. From the awx-setup-VERSION. directory, run setup.sh

NOTE: For users of Red Hat Enterprise Linux or CentOS, PackageKit can frequently interfere with the update mechanism. Consider disabling or removing PackageKit if installed prior to running the setup process.

```
[root@localhost ~]# ./setup.sh
```

- a. Setup will install AWX from RPM or Deb packages using repos hosted on AnsibleWorks.com.

When setup completes successfully, you should be able to point your web browser to the AWX server and see the AWX login screen.

If the upgrade of AWX fails or if you need assistance, please contact us at awx@ansibleworks.com. AnsibleWorks subscription customers will receive a faster response by filing a support issue.

Quick Start

After the installation of AWX is complete, we'll complete the following tasks to quickly set up and launch our first Ansible playbook using AWX. This first playbook launch will execute simple Ansible tasks to teach you how to use AWX and also ensure AWX is setup properly.

Here's a summary of the tasks we'll need to accomplish:

1. Login as Super User
2. Create an Organization
3. Add a new User to the organization
4. Add an Inventory to the organization
5. Create a set of Credentials
6. Create a Project
7. Create a new Job Template using an Ansible example playbook
8. Launch it!

You can expect the Quick Start to take less than thirty minutes, from beginning to end. At the end of the Quick Start, you'll have a functioning AWX that you can use to launch more sophisticated playbooks.

For the Quick Start, you will need to have completed the AWX installation and you will also need a target system to deploy the playbook to. This can be any sort of system that can be managed by Ansible.

NOTE: The requirements for a system to be managed by Ansible are at <http://www.ansibleworks.com/docs/gettingstarted.html>.

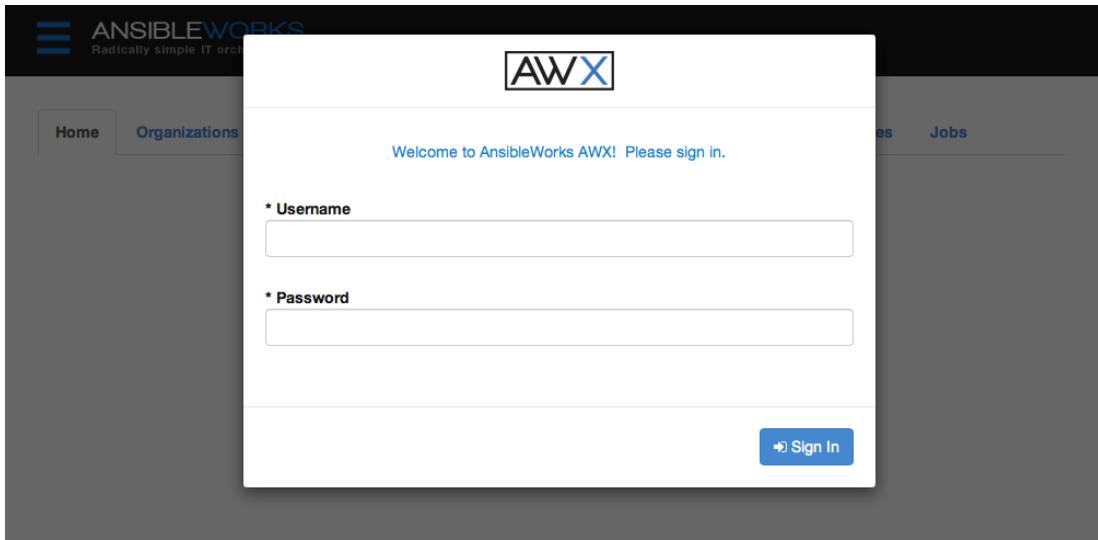
Ready? Let's go!

1. [Login as Super User](#)

First, log in to AWX by browsing to the AWX server URL at <http://<AWX server name>/>

Log in using the username and password set during the installation process. By default, this will be username: "admin" and password: "password". You can change this by clicking on the "admin" account on the users tab.

NOTE: We'll get into the details of the differences between a normal user, superuser, and organization administrator in the section [Users](#).



From this main interface, we can access all aspects of AWX, including Organizations, Users, Teams, Projects, Inventories, Credentials, Job Templates, and Jobs.

The image shows the AnsibleWorks AWX main dashboard. At the top, there's a header with the 'ANSIBLEWORKS' logo and subtext 'Radically simple IT orchestration', and a navigation bar with links for Home, Organizations, Users, Teams, Credentials, Projects, Inventories, Job Templates, and Jobs. On the far right of the header are links for 'Hello! admin', 'View License', 'Contact Support', and 'Logout'. Below the header, there are four main status boxes:

- Job Status**: A table showing counts for Jobs (Total 1, Failed 0), Inventories (Total 1, Failed 0), Groups (Total 1, Failed 0), and Hosts (Total 1, Failed 0).
- Inventory Sync Status**: A table showing 'No inventories configured for external sync'.
- System Summary**: A table showing counts for various entities: Organizations (Total 1, Failed 0), Users (Total 2, Failed 0), Teams (Total 1, Failed 0), Credentials (Total 1, Failed 0), Projects (Total 2, Failed 0), Inventories (Total 1, Failed 0), Groups (Total 1, Failed 0), Hosts (Total 1, Failed 0), Job Templates (Total 1, Failed 0), and Jobs (Total 1, Failed 0).
- Project SCM Status**: A table showing 'No projects configured for SCM sync'.

Keep in mind that the goal of this Quick Start is to launch a simple playbook. In order to do so, we'll need to set up a number of configuration options, but doing so now will ensure AWX is configured properly and allow us to easily execute more involved playbooks later while taking advantage of all the flexible role-based access control that AWX provides. You'll also get to know more about AWX along the way.

AWX provides multiple levels of role-based access, providing delegation of responsibility, but with fine-grained control over who can do what. We'll talk about that in more detail later in this document. For now, here's a simplified outline that shows the hierarchy of AWX's role based access control and the relationship between each element.

AWX Hierarchy

- Organizations
 - Inventories
 - Groups
 - Hosts
 - Teams
 - Credentials
 - Permissions
 - Users
 - Credentials
 - Permissions
- Projects
 - Playbooks
 - Job Templates
- Jobs

Now, let's create a new organization within which we can create our first user, detail our inventory of hosts, and store SSH credentials for those hosts.

2. [Create an Organization](#)

Click on the Organizations tab. An Organization is a logical collection of Users, Teams, Projects, and Inventories. It is the highest level object in the AWX object hierarchy.

and then click +Create new.

The screenshot shows the AnsibleWorks web interface. At the top, there's a navigation bar with links for Home, Organizations (which is highlighted with a red underline), Users, Teams, Credentials, Projects, Inventories, Job Templates, and Jobs. Below the navigation is a search bar with 'Name' and 'Search' fields, and a 'Create New' button. A large red arrow points from the left towards this 'Create New' button. Below the search bar is a table header with columns for '#', 'Name' (sorted by 'Name'), and 'Description'. A message 'No records matched your search.' is displayed. A second red arrow points from the right towards a button labeled 'Click to create a new Organization' which is enclosed in a rounded rectangle. At the bottom of the page are navigation buttons for 'Prev' and 'Next' and a 'Rows per page' dropdown set to 20.

Enter a simple name and description for the organization. You can edit both of these fields later, so the values aren't critical. For our example, we will create an organization for a fictitious company called Bender Products Ltd.

The screenshot shows the 'Create Organization' form. It has fields for 'Name' (Bender Products Ltd) and 'Description' (Bender operations). Below the description field is a 'Click to save your new Organization' button. At the bottom left are 'Save' and 'Reset' buttons, with a red arrow pointing to the 'Save' button.

Organizations have both normal users and organization administrators. Organization Administrators are able to modify the membership and other properties of the organization, whereas normal users cannot. They are essentially super users but only within the scope of that organization. For more about the differences between users and administrators, see the section on [Users](#).

The “admin” user is a Super User account -- a de-facto administrator for all organizations, so let’s use our admin powers to create a new user and add it to our new organization. When creating a new user, the checkbox Superuser? corresponds to this level of access. Only Super Users can create other Super Users or promote existing users to this level.

3. Create a new user and add the user to the organization

Expand the Users section (not the Users tab!) as shown here:

The screenshot shows the AnsibleWorks web interface. At the top, there is a navigation bar with links for Home, Organizations, Users, Teams, Credentials, Projects, Inventories, Job Templates, and Jobs. The 'Users' link is highlighted. The main content area shows a success message: "New organization successfully created!" above a form for creating a new organization. The form fields are labeled "Name" (Bender Products Ltd) and "Description" (Bender operations). Below the form is a button group with "Save" and "Reset" buttons. A red arrow points to the "Users" link in the sidebar below the form. The sidebar also includes an "Administrators" link. A callout bubble says "Click to expand Users section".

Add a user by clicking the +Add button.

The screenshot shows the "Users" page within the AnsibleWorks interface. The page has a search bar for "Username" and a header with columns for "#", "Username", "First Name", and "Last Name". A red arrow points to the "+Add" button in the top right corner of the header. A callout bubble says "Click to add a User to the Organization". Below the header, a message states "No records matched your search." At the bottom of the page, there are navigation links for "Prev" and "Next", and a "Rows per page" dropdown set to 10. A red arrow also points to the "Rows per page" dropdown.

A list of all existing users will be presented. Since we have not created any users, the only user listed is “admin”. Click the +Create New button to create a brand new user.

The screenshot shows the AnsibleWorks web interface. At the top, there is a navigation bar with links for Home, Organizations, Users, Teams, Credentials, Projects, Inventories, Job Templates, and Jobs. The 'Users' link is highlighted. Below the navigation bar, the URL 'Organizations / Bender Products Ltd / Add Users' is displayed. On the left, there is a search bar with 'Username' and 'Search' fields. To the right of the search bar is a green button labeled '+ Create New'. Below the search bar, there is a table header with columns for '#', 'Username', 'First Name', 'Last Name', and 'Select'. Under the 'Username' column, there is one entry: '1. admin'. In the center of the page, there is a button with the text 'Click to create a new User' enclosed in a rounded rectangle. At the bottom of the page, there are navigation links for 'Prev' and 'Next', a 'Rows per page' dropdown set to '20', and a status message 'Page: 1 of 1'.

Enter the user's details.

 **ANSIBLEWORKS**
Radically simple IT orchestration

Hello admin View License Contact Support Logout

Organizations / Bender Products Ltd / Users / Create User

* First Name
Dave

* Last Name
Smith

* Email
dsmith@myemail.com

* Organization
 Bender Products Ltd

* Username
dsmith

* Password

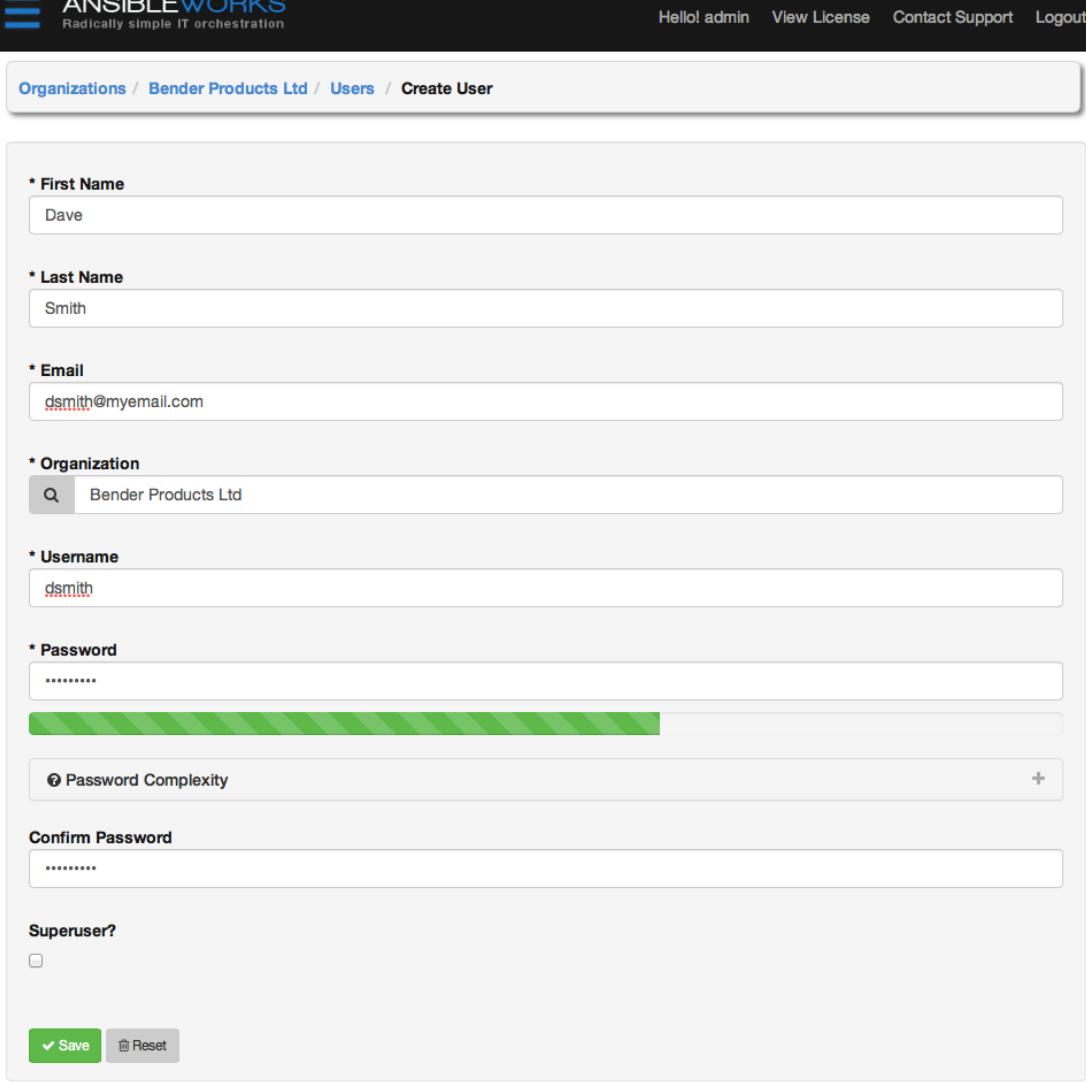
[Redacted]

>Password Complexity +

Confirm Password

Superuser?

Save **Reset**



Click the Save button to save the user. You will be taken back to the organization details, where the new user we just created now appears on the list.

The screenshot shows the AnsibleWorks web interface. At the top, there is a navigation bar with links for Home, Organizations, Users, Teams, Credentials, Projects, Inventories, Job Templates, and Jobs. The 'Organizations' tab is selected. Below the navigation bar, the URL 'Organizations / Bender Products Ltd' is displayed. The main content area shows the creation of a new organization named 'Bender Products Ltd' with a description of 'Bender operations'. There are 'Save' and 'Reset' buttons. Below this, a message says 'The new User has been added to the Organization'. A table lists a single user: 'dsmith' (Username), 'Dave' (First Name), 'Smith' (Last Name). A red arrow points to the 'dsmith' entry in the table. At the bottom, there is a link to 'Administrators'.

Now, we have an organization and a user. Let's add an inventory of hosts we'll be managing for Bender Products.

4. Create a new inventory and add it to the organization

An inventory is a collection of hosts that can be managed with AWX. Inventories are assigned to organizations and permission to launch playbooks against inventories is controlled at the user and team level. More information can be found in the [Inventories](#) and [Permissions](#) sections.

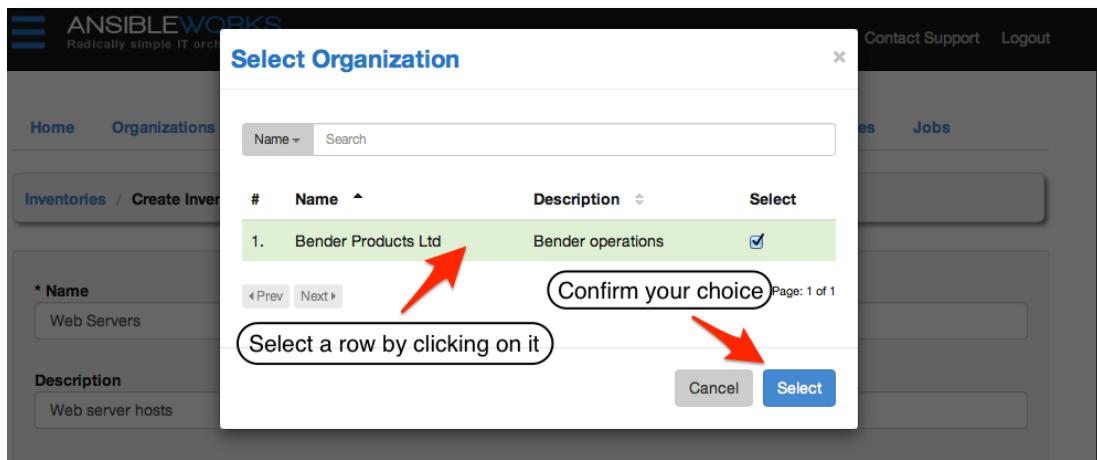
Create a new inventory by browsing to the Inventories tab and clicking +Create New.

The screenshot shows the AnsibleWorks web application interface. At the top, there's a navigation bar with links for Home, Organizations, Users, Teams, Credentials, Projects, Inventories (which is highlighted with a red wavy underline), Job Templates, and Jobs. Below the navigation is a search bar and a 'Create New' button. The main content area is titled 'Inventories' and displays a table with columns for #, Name, Organization, and Failed Hosts. A message says 'No records matched your search.' At the bottom, there are navigation buttons for Prev and Next, and a 'Rows per page' dropdown set to 20. A red arrow points from the text above to the 'Create New' button.

Enter the values for Name and Description, and then click the look-up button to the left of the Organization field to select a value. For this example, the name of our inventory will be Web Servers.

The screenshot shows the 'Create Inventory' properties form. It has fields for Name (Web Servers), Description (Web server hosts), and Organization. The Organization field includes a search icon and a dropdown menu with the placeholder 'Click to select an Organization'. At the bottom are 'Save' and 'Reset' buttons. Red arrows point to both the organization selection field and the 'Select' button in the dropdown menu.

An Inventory is assigned to an organization. For our example we'll use the organization we created earlier. Select the row from the list by clicking on it. The selected row will be highlighted with a pastel green and a checkbox in the Select column. Click the Select button to confirm your choice.



For now we will defer a discussion of variables until later and leave the Variables field alone. Click the Save button at the bottom of the page to create the inventory.

The screenshot shows the 'Create Inventory' properties page. It has fields for Name (Web Servers), Description (Web server hosts), Organization (Bender Products Ltd), and Variables (Parse as: YAML). A large red arrow points to the 'Save' button at the bottom.

Properties	
* Name	Web Servers
Description	Web server hosts
* Organization	<input type="text"/> Bender Products Ltd
Variables	Parse as: <input checked="" type="radio"/> YAML <input type="radio"/> JSON ---

Click to save the new Inventory

✓ Save ⌂ Reset

After clicking Save, you will be taken to a screen allowing you to add groups to the new inventory. A tree on the left side of the page represents the Web Servers inventory. The inventory is empty at this point, so the tree consists only of a root node.

The screenshot shows the AnsibleWorks web application interface. At the top, there is a navigation bar with links for Home, Organizations, Users, Teams, Credentials, Projects, Inventories (which is currently selected), Job Templates, and Jobs. On the far right of the top bar are links for 'Hello! admin', 'View License', 'Contact Support', and 'Logout'. Below the top bar, a breadcrumb navigation path shows 'Inventories / Web Servers / Groups'. A 'Group Selector' dropdown is open, showing 'Web Servers' as the selected item. The main content area is titled 'All Hosts' and contains a table with columns: Group, Failed Hosts, Status, and Last Updated. A message 'No records matched your search.' is displayed below the table. At the bottom of the page, there are navigation links for 'Prev' and 'Next', and a 'Rows per page' dropdown set to 20.

Inventories are divided into groups. A group might represent a particular environment (e.g. “Datacenter 1” or “Stage Testing”), a type of server (e.g. “Application Servers” or “DB Servers”), or any representation of your physical environment.

Hosts are added to groups. They cannot be added directly to the inventory root. So to begin adding hosts to the Web Servers inventory, we first need to add a group. Click the +Create New button.

Inventories / Web Servers / Groups

Group Selector: Web Servers

Group ▾ Search

All Hosts

Group Failed Hosts Status Last Updated

No records matched your search.

Bender Products has a group of web server hosts supporting the corporate CMS application. To add these hosts to the Web Servers inventory we'll create a "CMS Web" group. Again, we will defer a discussion of variables for later. Click the Save button to create the group.

ANSIBLEWORKS Radically simple IT orchestration

Contact Support Logout

Create New Group

* Name: CMS Web

Description: CMS Web Servers

Variables: Parse as: YAML JSON

Click to create the Group

Cancel Save

Finally, we'll add a host to the group. First, expand the menu to select Hosts.

Inventories / Web Servers / Groups 

Group Selector:

- Web Servers
- CMS Web

Expand this menu to select Hosts

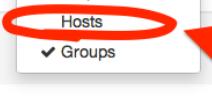
CMS Web

* Name CMS Web

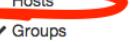
Description CMS Web Servers

Variables

Parse as: YAML JSON

Inventories / Web Servers / Groups 

Properties

Hosts 

Groups 

Group Selector:

- Web Servers
- CMS Web

Click Hosts

CMS Web

* Name CMS Web

Description CMS Web Servers

Variables

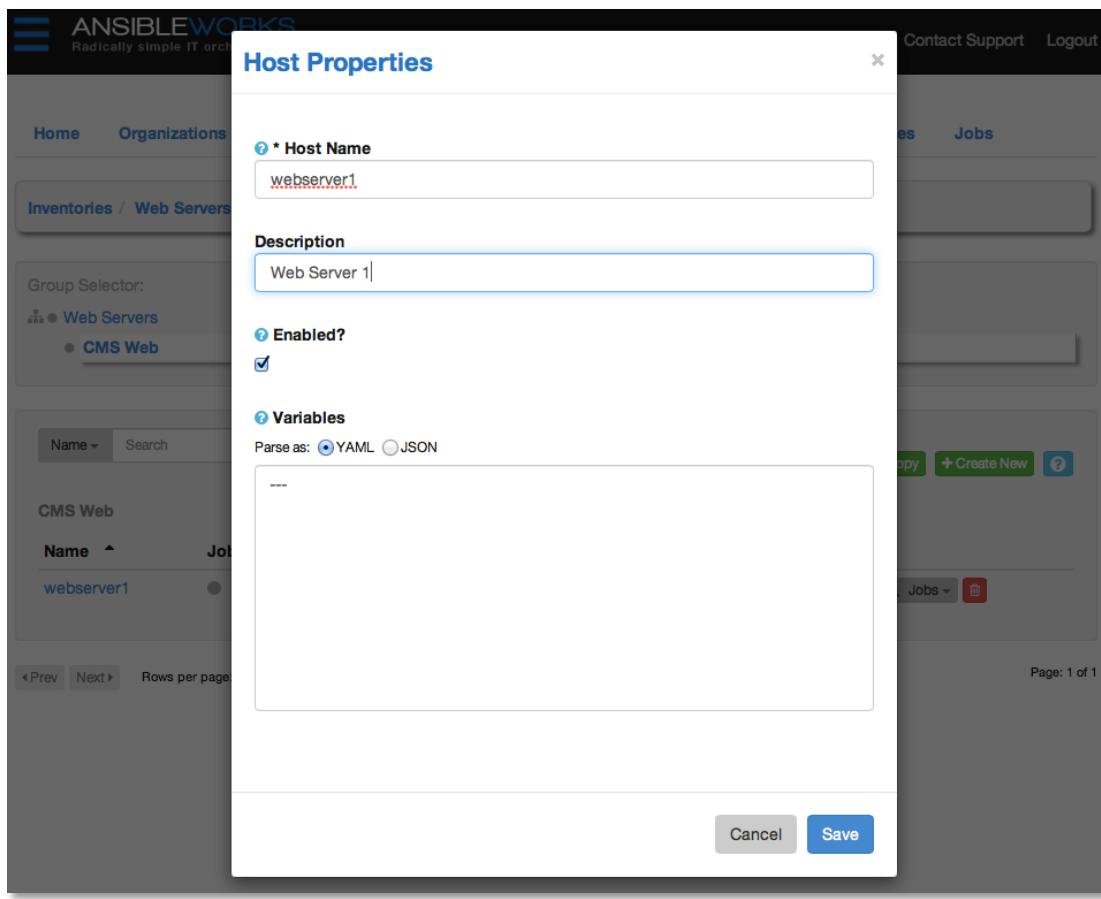
Parse as: YAML JSON

Select +Create New to create the new host and add it to the group.

The screenshot shows the AWX web interface. At the top, there's a navigation bar with 'Inventories / Web Servers / Hosts'. Below it is a 'Group Selector' section with 'Web Servers' and 'CMS Web' selected. The main area is titled 'CMS Web' and contains columns for 'Name', 'Job Status', 'Enabled', and 'Groups'. A search bar at the top says 'Name' and 'Search'. Below the table, a message says 'No records matched your search.' A prominent red arrow points to a button labeled 'Click to add a host'.

Enter the Host Name, which should either be the DNS resolvable name of the host or its IP address. This is how AWX will contact the host, so the host must be reachable using this hostname or IP address for AWX to function properly. The Description is arbitrary, as usual. (*Note, experienced Ansible users will know they could also set the ansible_ssh_host environment variable to use an alias, but that is not going to be covered here*).

For the purposes of this Quick Start, add a host that you can actually reach via SSH and manage using Ansible (i.e. that meets the Ansible [requirements](#)). We will launch a simple Ansible playbook that will not harm or modify the target in any way. Using a real target host allows us to ensure that AWX is setup properly.



Click Save to finish adding the host.

This screenshot shows the 'Hosts' list in the AnsibleWorks interface. The 'webserver1' host is now listed under the 'CMS Web' group. It has a status of 'n/a' and its 'Enabled' status is checked. Other columns include 'Name' (webserver1) and 'Groups' (CMS Web).

Next, we'll add credentials to our new user that AWX can use to access and launch Ansible playbooks for the host in our inventory.

5. Create a new set of credentials

Credentials are used to authenticate the AWX user to launch Ansible playbooks against inventory hosts and can include passwords and SSH keys. You can also require the AWX user to enter a password or key phrase when a playbook is launched using the credentials feature of AWX.

Create a new credential by browsing to the Credentials tab. Click +Create New to create a new credential.

The screenshot shows the AnsibleWorks web interface. At the top, there is a navigation bar with links for Home, Organizations, Users, Teams, **Credentials**, Projects, Inventories, Job Templates, and Jobs. The 'Credentials' link is highlighted with a red underline. Below the navigation bar, the main content area has a header 'Credentials'. Underneath is a search bar with 'Name' and 'Search' fields, and a green button labeled '+ Create New' with a magnifying glass icon. A red arrow points to this button. The main table area displays columns for '#', 'Name' (sorted by Name), and 'Description'. A message 'No records matched your search.' is shown. At the bottom of the table area, a blue button says 'Click to create a new Credential'. At the very bottom of the page, there are navigation buttons for 'Prev' and 'Next', and a dropdown for 'Rows per page' set to 20.

Enter an arbitrary Name and Description for this credential. Either an individual user or a team may own credentials. Let's associate this credential with the user we created in step #3.

Credentials / Create Credential

* Name
Root User, Ask Password

Description
root user, ask password

FAQ * Owned By?
 User Team [Select User](#)

* Type
Choose Type

Credential Type

Choose a type based on how this credential will be used:

AWS
Access keys for running cloud inventory sync with Amazon Web Services.

Machine
Define SSH and Sudo access for playbooks. Used when submitting jobs to run playbooks on a remote host.

Rackspace
Credentials for running cloud inventory sync with Rackspace.

SCM
Used on projects to clone and update local source code repositories from a remote revision control system such as Git, SVN or Mercurial.

[Save](#) [Reset](#)

ANSIBLEWORKS
Radically simple IT orchestration

Contact Support Logout

Home Organizations

Credentials / Create Credential

* Name
Root User, Ask Password

Description
root user, ask password

FAQ * Owned By?
 User Team

* User [Select User](#)

Select User

#	Username	First Name	Last Name	Select
1.	admin			<input type="checkbox"/>
2.	dsmith	Dave	Smith	<input checked="" type="checkbox"/>

Page: 1 of 1

[Cancel](#) [Select](#)

Next, select credential type Machine.

Credentials / Create Credential

* Name
Root User, Ask Password

Description
root user, ask password

? * Owned By?
 User Team

* User
dsmith Select credential type Machine

* Type
 Choose Type
 Machine Select credential type Machine
 SCM
 AWS
 Rackspace

AWS
Access keys for running cloud inventory sync with Amazon Web Services.

Machine
Define SSH and Sudo access for playbooks. Used when submitting jobs to run playbooks on a remote host.

Rackspace
Credentials for running cloud inventory sync with Rackspace.

SCM
Used on projects to clone and update local source code repositories from a remote revision control system such as Git, SVN or Mercurial.

Now, we'll enter the details of the appropriate authentication mechanism to use for the host we added to AWX in step #3. Use the actual credentials for the real host. To keep things simple, we'll use an SSH password, but ask for it at runtime. So, rather than enter the password here, we'll enter it later when we launch a playbook using these credentials. To do so, check the box Ask at runtime for SSH Password, as shown here.

NOTE: AWX supports various different options for what you want to store for credentials in this box. Uploading a locked SSH key is recommended, and AWX can prompt you for the SSH unlock password for use with ssh-agent when launching the job.

AWX encrypts passwords and key information in the AWX database and never makes secret information visible via the API.

* Type
Machine

Credential Type

SSH Username
root

SSH Password

 Ask at runtime?

Confirm SSH Password

SSH Private Key

Key Password

 Ask at runtime?

Confirm Key Password

Click Save.

Credentials			
Name	Search	Actions	
#	Name	Description	Type
1.	Root User, Ask Password	root user, ask password	Machine

Now, we'll create a new project and a job template with which to launch a simple playbook.

6. Create a new Project and select Create New.

Before we create this project, we'll need to create a subdirectory for it on the AWX server filesystem, where we will store the Ansible playbooks for this project.

NOTE: This will require you to log into the AWX server on the command line console. In a future version of AWX, this will be done without leaving the Web interface.

Create a new project directory by creating a directory on the AWX filesystem underneath the Project Base Path, by default “/var/lib/awx/projects”.

```
[root@localhost ~]# cd /var/lib/awx/projects  
[root@localhost ~]# mkdir helloworld
```

While we're here, let's go ahead and create a simple Ansible playbook. Use your favorite editor to create a file called “helloworld.yml” inside the directory we just created, “/var/lib/awx/projects”.

```
[root@localhost ~]# cd helloworld  
[root@localhost ~]# vi helloworld.yml
```

The contents of the file are below:

```
---  
- name: Hello World!  
  hosts: all  
  user: root  
  
  tasks:  
  
    - name: Hello World!  
      shell: echo "Hi! AWX is working"
```

Save this playbook file and we'll use it to test AWX running a playbook against the host in our inventory.

NOTE: Ansible playbooks utilize the YAML language. More information about Ansible playbooks may be found at: <http://www.ansibleworks.com/docs/playbooks.html>. More information on YAML can be found at: <http://yaml.org/>.

Now, create the new project by browsing to the Projects tab. Click +Create New.

The screenshot shows the AnsibleWorks web application interface. At the top, there is a navigation bar with links for Home, Organizations, Users, Teams, Credentials, Projects (which is highlighted with a red underline), Inventories, Job Templates, and Jobs. The main content area is titled "Projects". It features a search bar with "Name" and "Search" fields, and a set of action buttons including "+ Create New", a question mark icon, a refresh icon, and a magnifying glass icon. Below the search bar is a table header with columns for "#", "Name" (sorted ascending), "Description" (sorted descending), "Update Status" (sorted descending), and "Last Updated" (sorted descending). A message at the bottom of the table area states "No records matched your search."

Enter a Name and Description for the project.

The Project Base Path will display the value entered when AWX was installed and cannot be edited from this dialog. (See the section [Administration of AWX](#) for more information on how to modify this value.)

Leave SCM Type set to Manual, for now.

For the Playbook Directory, we will select a value that corresponds to the subdirectory we just created.

Projects / Create Project

* Name
HelloWorld

Description
hello world!

?

* Organization
Bender Products Ltd

* SCM Type
Manual

?

Project Base Path
/var/lib/awx/projects

?

* Playbook Directory
helloworld

Save Reset

Note: If you see the following warning, double check that the helloworld project directory and file were created correctly and that the permissions are correct. Use "chown -R awx" on the project directory if necessary. If SE Linux is enabled, check the directory and file context.

WARNING: There are no unassigned playbook directories in the base project path /var/lib/awx/projects. Either the projects directory is empty, or all of the contents are already assigned to other projects. New projects can be checked out from source control by changing the SCM type option rather than specifying checkout paths manually. To continue with manual setup, log into the AWX server and ensure content is present in a subdirectory under /var/lib/awx/projects. Run "chown -R awx" on the content directory to ensure awx can read the playbooks."

Select Save and the new project will be displayed.

Projects

#	Name	Description	Update Status	Last Updated
1.	HelloWorld	hello world!	n/a	11/19/13 12:07:56

+Create New

Finally, let's create a job template for this new playbook and launch it.

7. Create a new Job Template using an Ansible example playbook

A job template combines an Ansible playbook from a project and the settings required to launch it. Create a new job template by browsing to the Job Templates tab and clicking +Create New.

Home Organizations Users Teams Credentials Projects Inventories Job Templates Jobs

Job Templates

#	Name	Description
No records matched your search.		

+Create New

Enter values for the Name and Description. Jobs can be of type Run or Check. Select Run for this Quick Start (check corresponds to “dry run” mode.) Choose the Inventory, Project, and Credential from those we have created during this exercise.

The playbook drop-down menu will automatically populate from the project path and playbook we created in step #5. Choose the “helloworld” playbook.

* Name
HelloWorld

Description
hello world!

* Job Type
Run

* Inventory
Web Servers

* Project
HelloWorld

* Playbook
helloworld.yml

Credential
Root User, Ask Password

Click Save.

Job Templates			
Name	Search		+ Create New
#	Name	Description	
1.	HelloWorld	hello world!	Edit Launch Delete

Now, let's launch the playbook and watch it all come together.

9. Launch it!

To launch the playbook, browse to the Job Templates tab and click Launch on the template.

The screenshot shows the AWX Job Templates interface. At the top, there's a search bar and a 'Click to launch the Job' button. Below is a table with columns: #, Name, and Description. A single row is visible for 'HelloWorld' with the description 'hello world!'. To the right of the row are three buttons: 'Edit', 'Launch' (highlighted with a red arrow), and 'Delete'.

AWX will ask you for the SSH password, as we configured the credential.

The screenshot shows an 'Authentication Required' dialog box. It contains two fields: '* SSH Password' and '* Confirm SSH Password'. The 'Confirm SSH Password' field is highlighted with a red border. At the bottom are 'Cancel' and 'Continue' buttons.

AWX will then redirect the browser to the Jobs tab, where you can see the list of all jobs.

The screenshot shows the AWX Jobs interface. At the top, there's a search bar and a refresh icon. Below is a table with columns: Job ID, Date, Job Template, and Status. One job entry is listed: Job ID 8, Date 11/19/13 19:31:53, Job Template HelloWorld, Status pending. To the right of the status are three buttons: 'View', 'Launch' (highlighted with a red border), and 'Cancel'.

Select the Job ID or click View and then Status to see the details of the job. When the job is complete, you should see output similar to the following.

Jobs / 8 - HelloWorld / Status

Job Status
● successful

Date
11/19/13 19:31:53

Standard Out

```
SSH password:  
  
PLAY [Hello World!] *****  
  
GATHERING FACTS *****  
ok: [webserver1]  
  
TASK: [Hello World!] *****  
changed: [webserver1]  
  
PLAY RECAP *****  
webserver1 : ok=2 changed=1 unreachable=0 failed=0
```

Click Events from the drop-down menu:

Jobs / 8 - HelloWorld / Status

Status
Summary
Events

Job Status

Date 11/19/13 19:31:53

Standard Out

```
SSH password:  
  
PLAY [Hello World!] *****
```

The display will change to show the standard output from the host and the result of running our playbook.

Jobs / 8 - HelloWorld / Events			
Host	Search		
Created On	Status	Event	Host
11/19/13 19:32:04	● changed	⊕ Playbook Started	View
11/19/13 19:32:04	● changed	⊕ Play Started (Hello World!)	View
11/19/13 19:32:04	● success	⊕ Gathering Facts	View
11/19/13 19:32:05	● success	Host OK	webserver1 View
11/19/13 19:32:05	● changed	⊕ Task Started (Hello World!)	View
11/19/13 19:32:06	● changed	Host OK	webserver1 View
		Output:	
		Hi! AWX is working	
11/19/13 19:32:06	● changed	Playbook Complete	View

Great work! Your AWX installation is up and running properly. Now, you can browse through the [User Guide](#) and learn about all of these features of AWX in more detail.

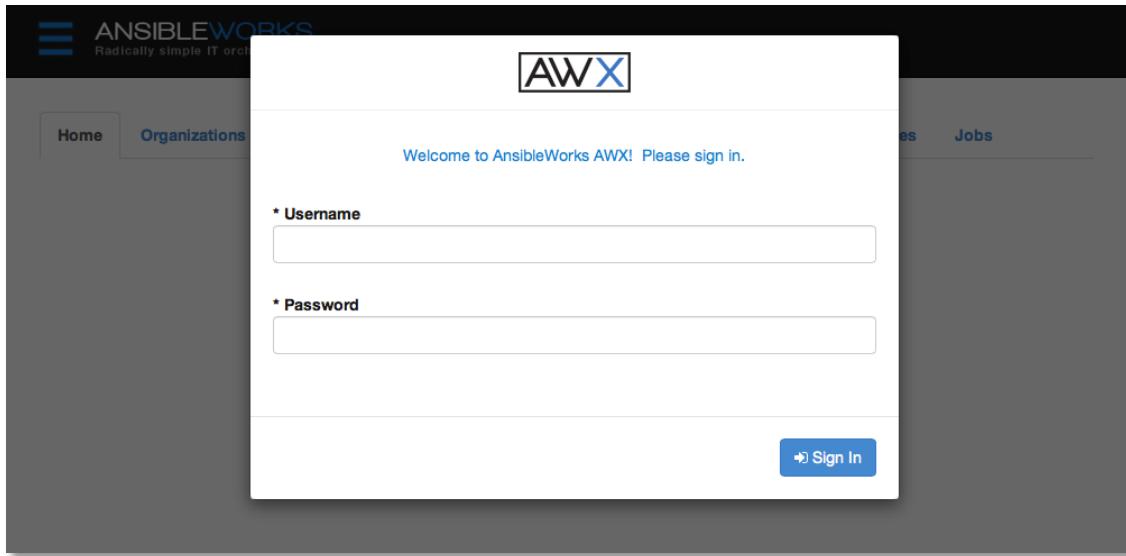
Don't hesitate to send your feedback to awx@ansibleworks.com. We appreciate your support!

User Guide

This section of the documentation will detail all of the functionality of AWX.

Logging In

To log in to AWX, browse to the AWX interface at <http://<AWX server hostname or IP Address>/>



Log in using a valid AWX username and password.

NOTE: The default username and password set during installation are “admin” and “password”, but the AWX administrator may have changed these settings during installation. If the default settings have not been changed, you can do from the Users tab.

Home

The central interface to AWX is the Home dashboard. This screen displays the status of AWX jobs, synchronization with inventory sources and source code management systems, and a summary of configured AWX objects.

The screenshot shows the AnsibleWorks Home dashboard with the following sections:

- Job Status:**

	Failed	Total
Jobs	0	3
Inventories	0	1
Groups	0	15
Hosts	0	5
- Inventory Sync Status:**

	Failed	Total
Inventories	0	1
Groups	0	1
Amazon EC2	0	1
- System Summary:**

	Total
Organizations	1
Users	3
Teams	1
Credentials	3
Projects	2
Inventories	1
Groups	15
Hosts	5
Job Templates	2
Jobs	3
- Project SCM Status:**

	Failed	Total
Git	0	1

All of the list items displayed on the Home dashboard are linked to their respective AWX objects for convenient access.

Organizations

An organization is a logical collection of Users, Teams, Projects, and Inventories and is the highest level in the AWX object hierarchy.

The Organizations tab displays all of the existing organizations for your installation of AWX. Organizations can be searched by Name or Description. Modify and remove organizations using the Edit and Delete buttons.

The screenshot shows the AWX web interface. At the top, there is a navigation bar with tabs: Home, Organizations, Users, Teams, Credentials, Projects, Inventories, Job Templates, and Jobs. The 'Organizations' tab is currently selected. Below the navigation bar is a header titled 'Organizations'. Underneath the header is a search bar with a 'Name' dropdown and a 'Search' input field, along with three icons: '+ Create New', a refresh icon, and a magnifying glass icon. A table lists one organization: 'Bender Products Ltd' (ID 1), with a description 'Bender operations'. To the right of the organization name are 'Edit' and 'Delete' buttons.

Create a new organization by selecting +Create New.

1. Enter the Name for your organization.
2. Optionally, enter a Description for the organization.

The screenshot shows the 'Create Organization' form. The title bar says 'Organizations / Create Organization'. The form has two fields: '* Name' (marked with a red asterisk) and 'Description'. Below the fields are two buttons: a green 'Save' button with a checkmark icon and a grey 'Reset' button with a trash can icon.

Click Save to finish creating the organization.

Once created, AWX will display the organization details, including two accordion-style menus below the organization name and description details that provide for managing users and administrators for the organization.

Organizations / Bender Products Ltd

* Name
Bender Products Ltd

Description
Bender operations

Save Reset

▶ Users
▶ Administrators

Users

A user is someone with access to AWX with associated permissions and credentials. For more information, please see the section [Users](#).

Expand the users menu by selecting Users.

Organizations / Bender Products Ltd

* Name
Bender Products Ltd

Description
Bender operations

Save Reset

▼ Users

#	Username	First Name	Last Name	Action
1.	dsmith	Dave	Smith	<input type="checkbox"/> Edit <input type="button" value="Delete"/>

Username ▾ Search

Rows per page: 10 Page: 1 of 1

▶ Administrators

This menu allows you to manage the user membership for this organization. (User membership may also be managed on a per-user basis via the Users tab.) The user list may be sorted and searched by Username, First Name, or Last Name. Existing users may also be modified and removed using the Edit and Delete buttons.

To add users to the organization, click the +Add button. Then, select one or more users from the list of available users by clicking the Select checkbox or clicking anywhere on the user row. Click the Select button when done.

The screenshot shows a user interface for managing organization members. At the top, there's a breadcrumb navigation: Organizations / Bender Products Ltd / Add Users. Below the header is a search bar with a dropdown for 'Username' and a 'Search' button. To the right of the search bar are three small icons: '+ Create New', a magnifying glass, and a question mark. The main area is a table with the following columns: #, Username, First Name, Last Name, and Select. There are two rows of data:

#	Username	First Name	Last Name	Select
1.	admin			<input type="checkbox"/>
2.	dsmith	Dave	Smith	<input type="checkbox"/>

At the bottom right of the table is a blue button labeled 'Select' with a checked checkbox icon.

To add a new user to AWX and to the organization, click the +Create New button, which takes us to the new user dialog.

* First Name

* Last Name

* Email

* Organization
 Bender Products Ltd

* Username

* Password

>Password Complexity

Confirm Password

Superuser?

Enter the appropriate details into the following fields:

- Username
- First Name
- Last Name
- Email
- Organization
- Password
- Confirm Password
- Superuser? (Give this user Super User privileges for AWX. *Caution!*)

All of these fields are required. Select Save when finished and the user will be added to the organization.

Organization Administrators

An organization administrator is a type of user that has the rights to create, modify, or delete objects in the organization, including projects, teams, and users in that organization. Expand the Administrators menu by selecting Administrators.

The screenshot shows a two-step process. The top part is a form for creating a new organization. It has fields for 'Name' (Bender Products Ltd) and 'Description' (Bender operations), with 'Save' and 'Reset' buttons. The bottom part shows a list of users for selecting administrators. It includes a search bar, sorting by 'Username', and a message 'No records matched your search.' with navigation buttons for 'Prev', 'Next', and 'Rows per page: 10'.

This menu displays a list of the users that are currently an organization administrator of the organization. The administrator list may be sorted and searched by Username, First Name, or Last Name.

To add an administrator to the organization, click the +Add button.

NOTE: A user must first be added to the Organization before it can be added to the list of Administrators for that Organization.

Then, select one or more users from the list of available users by clicking the Select checkbox or clicking anywhere on the user row. Click the Select button when done.

#	Username	First Name	Last Name	Select
1.	dsmith	Dave	Smith	<input type="checkbox"/>

Select

Users

A user is someone who has access to AWX with associated permissions and credentials. The Users tab allows you to manage all AWX users. The user list may be sorted and searched by Username, First Name, or Last Name.

#	Username	First Name	Last Name	Edit	Delete
1.	admin			<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
2.	dsmith	Dave	Smith	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

There are three types of AWX Users:

1. **Normal User:** read and write access is limited to the inventory and projects that the user has been granted the appropriate rights to via AWX Permissions.
2. **Organization Administrator:** the administrator of an organization has all of the rights of a normal user, as well as admin, read, and write permission over the entire organization and all of its inventories and projects, but does not have those levels of access on content belonging to other organizations. This level of user can create more users.
3. **Super User:** an AWX super user has admin, read, and write permissions over the entire AWX installation. A Super User is typically a systems administrator responsible for managing AWX, and would then delegate responsibilities for day-to-day work to various Organization Administrators.

NOTE: The initial user (usually “admin”) created by the AWX installation process is a Super User. One Super User must always exist, so if you wish to delete “admin”, first create another Super User account.

To create a new user click the +Create New button, which takes us to the new user dialog.

The screenshot shows the 'Create User' dialog box. At the top left is the breadcrumb 'Users / Create User'. Below it is a list of fields with validation stars: * First Name, * Last Name, * Email, * Organization (with a search icon), * Username, * Password, and Confirm Password. Underneath these is a 'Password Complexity' dropdown with a help icon and a plus sign. At the bottom left is a 'Superuser?' checkbox. At the bottom right are two buttons: a green 'Save' button with a checkmark icon and a grey 'Reset' button.

Enter the appropriate details into the following fields:

- Username
- First Name
- Last Name
- Email
- Organization (Choose from an existing organization)
- Password
- Confirm Password
- Superuser? (Gives this user admin privileges for AWX. *Caution!*)

All of these fields are required. Select Save when finished.

Once the user is successfully created, AWX will open the Edit User dialog. This is the same menu that is opened if the Edit button is clicked from the Users tab. Here, User Setting, Credentials, Permissions, and other user membership details may be reviewed and modified.

The screenshot shows the 'Create New User' form in AWX. The fields filled in are:

- * First Name: Dave
- * Last Name: Smith
- * Email: dsmith@myemail.com
- * Username: dsmith
- Created by LDAP?:
- * Password: (empty field)
- * Confirm Password: (empty field)
- Superuser?:

Below the form is a sidebar with the following navigation links:

- > Credentials
- > Permissions
- > Admin of Organizations
- > Organizations
- > Teams
- > Projects

Credentials

Credentials are utilized by AWX for authenticating when launching jobs against machines, to synchronize with inventory sources, and to import project content from a version control system. For details about how to use credentials, please see the section [Credentials](#).

To add credentials to user, expand the credentials menu and click the +Add button.

#	Name	Description	Actions
1.	AWS	Amazon Web Services	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2.	Root User, Ask Password	root user, ask password	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Then, select one or more credentials from the list of available credentials by clicking the Select checkbox. Click the Select button when done.

The screenshot shows a table with columns: #, Name, Description, Type, and Select. There are two rows: 1. AWS (Description: Amazon Web Services, Type: AWS) and 2. Root User, Ask Password (Description: root user, ask password, Type: Machine). A 'Select' checkbox is at the bottom right of the table.

#	Name	Description	Type	Select
1.	AWS	Amazon Web Services	AWS	<input type="checkbox"/>
2.	Root User, Ask Password	root user, ask password	Machine	<input type="checkbox"/>

To add new credentials to the user click the +Create New button, which takes us to the Create Credential dialog.

The form includes fields for Name, Description, Owned By (User selected), User (dsmith selected), Type (Choose Type dropdown), and Credential Type (dropdown with a plus sign). Buttons at the bottom include Save (green) and Reset.

Enter the appropriate details depending on the type of credential and select Save.

Permissions

Permissions are the set of privileges assigned to users and teams that provide the ability to read, modify, and administer projects, inventories, and other AWX elements. For details about how to use permissions, please see the section [Permissions](#).

This menu displays a list of the permissions that are currently available. The permissions list may be sorted and searched by Name, Inventory, Project or Permission type.

The screenshot shows a web-based application interface for managing permissions. At the top, there is a search bar labeled "Name" with a dropdown arrow. To the right of the search bar are two buttons: a green "+Add" button and a blue circular refresh/cancel button. Below the search bar is a table header with columns: #, Name (sorted ascending), Inventory (sorted descending), Project (sorted descending), and Permission (sorted descending). A blue banner at the bottom of the table area states "No records matched your search." At the bottom of the page, there are navigation links for "Prev" and "Next", and a "Rows per page" dropdown set to 10.

To add new permissions to the user, click the +Add button, which takes us to the Add Permission dialog.

The screenshot shows the "Add Permission" dialog. At the top, the breadcrumb navigation is "Users / Dsmith / Permissions / Add Permission". The dialog has several input fields:

- Permission Type**: Radio buttons for "Inventory" (selected) and "Deployment".
- * Name**: An input field with a placeholder text area below it.
- Description**: An input field with a placeholder text area below it.
- * Inventory**: A search input field with a magnifying glass icon and a placeholder text area below it.
- * Permission**: Radio buttons for "Admin", "Read", and "Write".

At the bottom of the dialog are two buttons: a green "Save" button with a checkmark icon and a white "Reset" button with a circular arrow icon.

Enter the appropriate details into the following fields:

- Permission Type
 - Inventory
 - Deployment
- Name
- Description

Selecting a Permission Type of either Inventory or Deployment will change the appearance of the Add Permission dialog to present appropriate options for each type of permission.

For a permission of type Inventory, enter the following details:

- Inventory (Select from the available inventories)
- Permission
 - Admin
 - Read
 - Write

For a permission of type Deployment, enter the following details:

- Project (Select from the available projects)
- Inventory (Select from the available inventories)
- Permission
 - Run
 - Check

Select Save.

[Admin of Organizations](#)

This displays the list of organizations that this user is an administrator of. This list may be searched by Organization Name or Description. A user cannot be made an organization administrator from this interface panel.

The screenshot shows a web-based application interface titled "Admin of Organizations". At the top, there is a search bar labeled "Name" with a dropdown arrow. Below the search bar is a table header with columns: "#", "Name" (sorted by ascending order), and "Description". A blue circular refresh button is located in the top right corner of the table area. The main content area displays a message: "No records matched your search." At the bottom of the page, there are navigation buttons for "Prev" and "Next", and a "Rows per page:" dropdown set to "10".

[Organizations](#)

This displays the list of organizations that this user is a member of. This list may be searched by Organization Name or Description. Organization membership cannot be modified from this display panel.

▼ Organizations

#	Name	Description
1.	Bender Products Ltd	Bender operations

◀Prev ▶Next Rows per page: 10 Page: 1 of 1

This screenshot shows a list of organizations. There is one entry: Bender Products Ltd, which is described as 'Bender operations'. The list includes columns for ID, Name, and Description. A search bar at the top allows filtering by Name. Navigation buttons for 'Prev' and 'Next' are present, along with a dropdown for 'Rows per page' set to 10, and a page indicator 'Page: 1 of 1'.

Teams

This displays the list of teams that this user is a member of. This list may be searched by Team Name or Description. Team membership cannot be modified from this display panel.

▼ Teams

#	Name	Description
No records matched your search.		

◀Prev ▶Next Rows per page: 10

This screenshot shows a list of teams. A message 'No records matched your search.' is displayed. The list includes columns for ID, Name, and Description. A search bar at the top allows filtering by Name. Navigation buttons for 'Prev' and 'Next' are present, along with a dropdown for 'Rows per page' set to 10.

Projects

This displays the list of projects that this user has access to. This list may be searched by Project Name or Description. Project access cannot be modified from this display. For more information about projects, please see the section [Projects](#).

▼ Projects

#	Name	Description
No records matched your search.		

◀Prev ▶Next Rows per page: 10

This screenshot shows a list of projects. A message 'No records matched your search.' is displayed. The list includes columns for ID, Name, and Description. A search bar at the top allows filtering by Name. Navigation buttons for 'Prev' and 'Next' are present, along with a dropdown for 'Rows per page' set to 10.

Teams

A team is a subdivision of an organization with associated users, projects, credentials, and permissions. Teams provide a means to implement role-based access control schemes and delegate responsibilities across organizations. For instance, permissions may be granted to a whole team rather than each user on the team.

This tab allows you to manage the teams for AWX. The user list may be sorted and searched by Username, Description, or Organization.

The screenshot shows two instances of the AWX interface, one above the other, both displaying the 'Teams' management page. The top instance has the 'Teams' tab selected in the navigation bar, while the bottom instance has the 'Users' tab selected. Both pages feature a search bar with dropdown menus for 'Name' and 'Search', and a green '+ Create New' button. Below the search bar is a table header with columns for '#', 'Name' (sorted ascending), 'Description' (sorted descending), and 'Organization' (sorted descending). A message 'No records matched your search.' is displayed in a light blue box at the bottom of the table area.

To create a new team, click the +Create New button, which takes us to the Create Team dialog.

Teams / Create Team

* Name

Description

* Organization

Save Reset

Teams / Create Team

Name

Description

Organization

Save Reset

Enter the appropriate details into the following fields:

- Name
- Description
- Organization (Choose from an existing organization)

All fields are required. Select Save.

Once the team is successfully created, AWX will open the Edit Team dialog. This is the same menu that is opened if the Edit button is clicked from the Teams tab. Here, Team Settings, Credentials, Permissions, Projects, and Users associated with this team may be reviewed and modified.

Teams / Production Operations

▼ Team Settings

New team successfully created!

* Name
Production Operations

Description
Production ops team

* Organization
Q Bender Products Ltd

✓ Save Reset

► Credentials

► Permissions

► Projects

► Users

Teams / Production Operations

▼ Team Settings

New team successfully created!

Name Production Operations

Description Production ops team

Organization Q Bender Products Ltd

✓ Save Reset

► Credentials

► Permissions

► Projects

► Users

Credentials

Credentials are utilized by AWX for authenticating when launching jobs against machines, to synchronize with inventory sources, and to import project content from a version control system. For details about how to use credentials, please see the section [Credentials](#).

The screenshot shows a table header with columns: #, Name, and Description. Below the header, a message says "No records matched your search." At the bottom, there are navigation buttons for 'Prev' and 'Next', and a 'Rows per page:' dropdown set to 10.

To add credentials to the team, click the +Add button. Then, select one or more credentials from the list of available credentials by clicking the Select checkbox. Click the Select button when done.

The screenshot shows a table with columns: #, Name, Description, Type, and Select. One row is listed: "1. Root User, Ask Password" with "root user, ask password" in the Description field, "Machine" in the Type field, and an unchecked checkbox in the Select column. A checked checkbox is shown in the Select button at the bottom right.

To create new credentials and add them to the team, click the +Create New button, which takes us to the Create Credential dialog.

The screenshot shows the 'Create Credential' page. At the top, the breadcrumb navigation is 'Teams / Production Operations / Credentials / Create Credential'. The main form has several fields:

- * Name:** A text input field.
- Description:** A text input field.
- * Owned By?**: A radio button group with 'User' and 'Team' options, where 'Team' is selected.
- * Team:** A search bar with a placeholder 'Production Operations'.
- * Type:** A dropdown menu labeled 'Choose Type'.
- Credential Type:** A dropdown menu with a '+' icon.

 At the bottom are two buttons: a green 'Save' button with a checkmark and a grey 'Reset' button.

Enter the appropriate details depending on the type of credential and select Save.

Permissions

Permissions are the set of privileges assigned to users and teams that provide the ability to read, modify, and administer projects, inventories, and other AWX elements. For details about how to use permissions, please see the section [Permissions](#).

This menu displays a list of the permissions that are currently available. The permissions list may be sorted and searched by Name, Inventory, Project or Permission type.

The screenshot shows the 'Permissions' list page. The table has the following structure:

#	Name	Inventory	Project	Permission
No records matched your search.				

 At the top left is a search bar with a 'Name' dropdown. To the right of the search bar are a green '+Add' button and a refresh/circular arrow icon. Below the table are navigation buttons for 'Prev' and 'Next' and a 'Rows per page' dropdown set to 10.

To add new permissions to the team, click the +Add button, which takes us to the Add Permission dialog.

Teams / Production Operations / Permissions / Add Permission

Permission Type

Inventory Deployment

*** Name**

Description

*** Inventory**

*** Permission**

Admin Read Write

Enter the appropriate details into the following fields:

- Permission Type
 - Inventory
 - Deployment
- Name
- Description

Selecting a Permission Type of either Inventory or Deployment will change the appearance of the Add Permission dialog to present appropriate options for each type of permission.

For a permission of type Inventory, enter the following details:

- Inventory (Select from the available inventories)
- Permission
 - Admin
 - Read
 - Write

For a permission of type Deployment, enter the following details:

- Project (Select from the available projects)
- Inventory (Select from the available inventories)
- Permission
 - Run
 - Check

Select Save.

Projects

This displays the list of projects that this team has access to. This list may be searched by Project Name or Description. For more information about projects, please see the section [Projects](#).

The screenshot shows a web-based application interface for managing projects. At the top left is a dropdown menu labeled "Projects". Below it is a search bar with a "Name" dropdown and a "Search" input field. To the right of the search bar are two buttons: a green "+Add" button and a blue circular refresh/cancel button. The main area contains a table with columns: "#", "Name" (sorted by name), and "Description" (sorted by description). A message "No records matched your search." is displayed in a light blue box. At the bottom of the table are navigation buttons for "Prev" and "Next", and a "Rows per page" dropdown set to 10.

To add a project to the team, click the “+Add” button. Then select one or more projects from the list of available credentials by clicking the Select checkbox or clicking anywhere on the user row. Click Finished when done.

The screenshot shows a "Teams / Production Operations / Add Project" dialog. At the top is a breadcrumb trail: Teams > Production Operations > Add Project. Below it is a search bar with a "Name" dropdown and a "Search" input field. To the right are buttons for "+Create New" and several icons. The main area is a table with columns: "#", "Name" (sorted by name), "Description", "Update Status" (sorted by status), "Last Updated" (sorted by last updated), and "Select". There is one row: "1. HelloWorld" with "hello world!" in the description, "n/a" in update status, and "11/19/13 12:07:56" in last updated. A "Select" checkbox is checked. At the bottom right is a "Select" button with a checked checkbox icon.

To create a new project and add it to the team, +Create New, which takes us to the Create Project dialog.

Teams / Production Operations / Projects / Create Project

* Name

Description

Organization

* SCM Type

Project Base Path

Playbook Directory

Enter the appropriate details into the following fields:

- Name
- Description
- Organization
- SCM Type
- Project Base Path
- Project Path

All fields are required. Select Save.

Users

This menu displays the list of users that are members of this team. This list may be searched by Username, First Name, or Last Name. For more information on users, please see the section [Users](#).

To add users to the team, click the +Add button. Then, select one or more users from the list of available users by clicking the Select checkbox or clicking anywhere on the user row. Click the Select button when done.

#	Username	First Name	Last Name	Select
1.	admin			<input type="checkbox"/>
2.	dmsmith	Dave	Smith	<input type="checkbox"/>

Permissions

Permissions are rights given to users to perform actions, including manage inventory and invoke Ansible playbooks / roles.

There are two permission types available to be assigned to users and teams, each with its own set of permissions available to be assigned:

- Inventory: grants permission to act on inventories, groups, and hosts
 - Admin: modify the settings for the specified inventory. This permission also grants Read and Write permissions.
 - Read: view groups and hosts within a specified inventory

- Write: create, modify, and remove groups, and hosts within a specified inventory.
Does not give permission to modify the inventory settings. This permission also grants the Read permission.
- Deployment: grants permission to launch jobs from the specified project against the specified inventory
 - Run: launch jobs of type Run. This permission also grants the Check permission.
 - Check: launch jobs of type Check.

Permissions do not have their own tab, but may be managed from either or both of the Users and Teams tabs. See those sections for information on how to modify, add, and delete permissions.

Credentials

Credentials are utilized by AWX for authenticating when launching jobs against machines, to synchronize with inventory sources, and to import project content from a version control system.

NOTE: AWX encrypts passwords and key information in the AWX database and never makes secret information visible via the API.

The Credentials tab displays a list of the credentials that are currently available. The credentials list may be sorted and searched by Name, Description, or Type.

#	Name	Description	Type	
1.	AWS	Amazon Web Services	AWS	<input checked="" type="button"/> Edit <input type="button"/> Delete
2.	Root User, Ask Password	root user, ask password	Machine	<input checked="" type="button"/> Edit <input type="button"/> Delete

Credentials may be managed from either the Teams tab or the Users tab. The Credentials tab simply provides a searchable and sortable list of credentials for your convenience.

To manage credentials for teams, please browse to the Teams tab and edit the appropriate team. Likewise, to manage credentials for a user, browse to the Users tab and edit the appropriate user.

Credentials added to a Team will be available to all members of the team, whereas credentials added to a user are only available to that user, by default.

There are four types of Credentials:

Machine

Define SSH and Sudo access for playbooks. Used when submitting jobs to run playbooks on a remote host.

Machine credentials have several attributes that may be configured:

- SSH Password

The actual password to be used to authenticate the user via SSH. This password may be stored encrypted in the AWX database, if entered. Alternatively, you may configure AWX to ask the user for the password when a job that uses this credential is launched by selecting Ask at runtime. In that case, a dialog will open when the job is launched where the user may enter the password and password confirmation.

- SSH Private Key

The actual SSH Private Key to be used to authenticate the user via SSH. This key is stored encrypted in the AWX database.

- SSH Private Key with Key Password

In addition to using an SSH private key, you may configure a Key Password associated with the private key. This password may be stored encrypted in the AWX database, if entered. Alternatively, you may configure AWX to ask the user for the password when a job that uses this credential is launched by selecting Ask at runtime. In that case, a dialog will open when the job is launched where the user may enter the password and password confirmation.

- Sudo Password

The actual password to be used to authenticate the user via sudo. This password may be stored encrypted in the AWX database, if entered. Alternatively, you may configure AWX to ask the user for the password when a job that uses this credential is launched by selecting Ask at runtime. In that case, a dialog will open when the job is launched where the user may enter the password and password confirmation.

Sudo Password must be used in combination with one of the other methods, since AWX must first establish an authenticated SSH connection with the host prior to invoking sudo to change to the sudo user.

AWS

Enables synchronization of cloud inventory with Amazon Web Services. Requires the AWS Access Key and Secret Key.

Rackspace

Enables synchronization of cloud inventory with Rackspace. Requires the Rackspace Username and API Key.

SCM

Used on projects to clone and update local source code repositories from a remote revision control system such as Git, SVN or Mercurial.

* Type
SCM

Credential Type +

Username

Password

Confirm Password

SCM Private Key

Key Password

Confirm Key Password

Projects

A Project is a logical collection of Ansible playbooks, represented in AWX.

Add your Ansible projects to the filesystem of your AWX installation under the project base path. You can do this by either managing playbooks and playbook directories manually or by using a source code management (SCM) system supported by AWX, including Git, Subversion, and Mercurial.

NOTE: By default, the Base Project Path is /var/lib/awx/projects, but this may have been modified by the AWX administrator. It is configured in /etc/ansibleworks/settings.py. Use caution when editing this file, as it is possible to disable your installation.

This menu displays a list of the projects that are currently available. The list of projects may be sorted and searched by Project Name, Description, by the time the project was Last Updated, or by Status. From the Projects tab, you can also modify and remove existing projects, using the Edit and Delete buttons.

For projects managed via Source Code Management (SCM), the Status button will display the update status for the project and Update will invoke an immediate SCM update for the project.

Status may be one of the following:

- Updating - an update is in progress
- Never updated - project has never been updated
- Failed - last update failed
- Successful - last updated succeeded
- Missing - project has a last update, but the project directory is missing, or project doesn't use SCM and the directory is missing
- Ok - project doesn't use SCM, and the directory is present

#	Name	Description	Update Status	Last Updated	Action
1.	HelloWorld	hello world!	n/a	11/19/13 12:07:56	Edit Update Delete

Refresh the list of projects with the Refresh button. To create a new project, click the +Create New button, which takes us to the Create Project dialog.

NOTE: If you have not added any Ansible playbook directories to the base project path, then you will receive the following message from AWX:

WARNING: There are no unassigned playbook directories in the base project path /var/lib/awx/projects. Either the projects directory is empty, or all of the contents are already assigned to other projects. New projects can be checked out from source control by changing the SCM type option rather than specifying checkout paths manually. To continue with manual setup, log into the AWX server and ensure content is present in a subdirectory under /var/lib/awx/projects. Run "chown -R awx" on the content directory to ensure awx can read the playbooks.

Correct this issue by creating the appropriate playbook directories and checking out playbooks from your SCM or otherwise copying playbooks into the appropriate playbook directories.

Projects / Create Project

* Name

Description

Organization

* SCM Type
 Manual

Project Base Path
 /var/lib/awx/projects

Playbook Directory
 Choose Playbook Directory

Enter the appropriate details into the following fields:

- Name
 - Description
 - Organization
- A project must have at least one organization. Pick one organization now to create the project, and then after the project is created you can add additional organizations.
- SCM Type
- Select one of Manual, Git, SVN, or Mercurial. (See the appropriate section below for more detail.)
- Project Base Path (Shown here as a convenience. A future release may make this user-editable.)
 - Project Path (The project paths shown here are automatically read from the directory tree with a root of the project base path.)

All fields are required. Select Save.

Note: Each project path can only be assigned to one project. If you receive the following message, ensure that you have not already assigned the project path to an existing project.

All of the project paths have been assigned to existing projects, or there are no directories

found in the base path. You will need to add a project path before creating a new project.

To manage playbooks manually:

1. Create one or more directories to store playbooks under the Project Base Path (e.g. “/var/lib/awx/projects/”)
2. Create or copy playbook files into the playbook directory(s).
3. Ensure that the playbook directory(s) and files are owned by the same UNIX user and group that the AWX service runs as.
4. Ensure that the permissions are appropriate for the playbook directory(s) and files.

If you have trouble adding a project path, check the permissions and SE Linux context settings for the project directory and files.

To manage playbooks using SCM, select the appropriate SCM Type.

The screenshot shows the 'Create Project' interface. At the top, there's a breadcrumb navigation: 'Projects / Create Project'. Below it is a form with the following fields:

- * Name**: A text input field.
- Description**: A text input field.
- Organization**: A dropdown menu with a search icon, currently set to 'Git'.
- * SCM Type**: A dropdown menu set to 'Git'.
- * SCM URL**: A text input field.
- GIT URLs**: A list input field with a '+' button to add items.
- SCM Branch**: A text input field.
- SCM Credential**: A dropdown menu with a search icon.
- SCM Options**: A section containing three checkboxes: 'Clean' (with a question mark icon), 'Delete on Update' (with a question mark icon), and 'Update on Launch' (with a question mark icon).

At the bottom of the form are two buttons: a green 'Save' button with a checkmark icon and a grey 'Reset' button.

Enter the appropriate details into the following fields:

- SCM URL
- SCM Branch

Optionally enter the SCM branch for Git or Mercurial

- Revision # (SVN only)

Optionally enter the Revision # for Subversion

- If authentication is required, select the appropriate SCM credentials.
- Clean

Remove any local modifications prior to performing an update.

- Delete on Update

Delete the local repository in its entirety prior to performing an update. Depending on the size of the repository this may significantly increase the amount of time required to complete an update.

- Update on Launch

Each time a job runs using this project, perform an update to the local repository prior to starting the job.

Inventories

An inventory is a collection of hosts against which jobs may be launched. Inventories are divided into groups and these groups contain the actual hosts. Groups may be sourced manually, by entering host names into AWX, or from cloud providers, including Amazon Web Services EC2 and Rackspace Cloud Servers.

This tab displays a list of the inventories that are currently available. The inventory list may be sorted and searched by Name or Organization and filtered by inventories with external source, by hosts with failed jobs, and inventories that have failed to update with an external source.

The screenshot shows the Ansible web interface with the 'Inventories' tab selected. The main content area displays a table of inventories. One row is present in the table:

#	Name	Organization	Failed Hosts	Status
1.	Web Servers	Bender Products Ltd	0 / 1	0 / 0

Below the table are buttons for '+ Create New', 'Jobs', 'Edit', and 'Delete'.

To create a new inventory click the +Create New button, which takes us to the Create Inventory dialog.

The screenshot shows the 'Create Inventory' dialog. It includes the following fields:

- * Name: A text input field.
- Description: A text input field.
- * Organization: A dropdown menu with a search icon.
- Variables:
 - Parse as: Radio buttons for YAML (selected) and JSON.
 - A large text area for defining variables in YAML or JSON syntax.

At the bottom are 'Save' and 'Reset' buttons.

Enter the appropriate details into the following fields and select Save:

- Name (required)
- Description
- Organization (Select from the available organizations)
- Variables

Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.

Existing inventories may be managed from the Inventories tab using the following actions:

- Jobs
 - All: Show *all* jobs launched against the selected inventory
 - Failed: Show *failed* jobs launched against the selected inventory
- Edit
 - Properties: Edit the properties of the selected inventory
 - Hosts: Manage the hosts belonging to the selected inventory
 - Groups: Manage the groups belonging to the selected inventory
- Delete: Delete the selected inventory. *This operation cannot be reversed!*

Inventories are divided into groups, which may contain hosts and other groups. *An inventory must contain at least one group.*

Groups

To add a group to an inventory or to manage an existing group, select Groups from the Edit menu for the selected inventory.

The screenshot shows a web-based interface for managing inventories. At the top, there's a header bar with the title 'Inventories'. Below it is a search bar with 'Name' and 'Search' fields, and buttons for '+ Create New' and a magnifying glass icon. The main area is a table with columns: '#', 'Name', 'Organization', 'Failed Hosts', and 'Status'. One row is visible, showing '1.', 'Web Servers', 'Bender Products Ltd', '0 / 1', and '0 / 0'. To the right of this row is a set of buttons: 'Jobs', 'Edit', and 'Delete'. A context menu is open over the 'Edit' button, listing 'Properties', 'Hosts', and 'Groups', with 'Groups' being the selected option. At the bottom of the table area, there are navigation buttons ('Prev', 'Next'), a 'Rows per page' dropdown set to '20', and a page indicator 'Page: 1 of 1'.

This screen displays a list of the groups that are currently available. The group list may be sorted and searched by Group Name, Status, or Source and filtered by groups with external source, by hosts with failed jobs, and groups that have failed to update with an external source.

To edit the group properties, click the group name. Additional actions may be performed on the group by selecting the buttons to the right.

- Update – Perform an update on the selected group from its source
- Cancel – Cancel any group update in progress

Create a new group by clicking the +Create New button, which opens the Create Group dialog.

Enter the appropriate details into the following fields and click Save.

- Name (required)
- Description
- Variables

Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.

Upon saving the group, the group properties are displayed and may be modified.

The screenshot shows the AWX interface for managing inventory groups. On the left, a sidebar titled 'Group Selector' shows a tree structure with 'Web Servers' expanded, showing 'AWS' and 'CMS Web' as children. The main panel is titled 'AWS' and contains the following fields:

- Name:** AWS
- Description:** Systems in AWS EC2
- Variables:** A large text area containing '---'
- Parse as:** A radio button group with 'YAML' selected.
- Source:** A dropdown menu set to 'Manual'.

At the bottom right are 'Save' and 'Reset' buttons.

By default, the group Source is manual, which means that the hosts must be entered into AWX manually. (See [Add a new host](#) for more information on managing hosts individually.)

To synchronize the inventory group from a cloud source, select the cloud from the Source menu. AWX 1.4 supports Amazon Web Services EC2 and Rackspace Cloud Servers.

[Amazon Web Services EC2](#)

To configure a group for AWS, select Amazon EC2 and enter the following details:

Source

Amazon EC2

Cloud Credential

Regions

Source Variables

Parse as: YAML JSON

```
---
```

Update Interval

none

Update Options

Overwrite [?](#) Overwrite Variables [?](#) Update on Launch [?](#)

- Cloud Credential

Choose from an existing Credential. For more information, see the [Credentials](#) section.

- Regions

A comma-separated list of regions matching the regions used at AWS. Only hosts associated with the list of regions will be included in the update process.

As of AWX 1.4, the available AWS regions include:

Region	Name
ap-northeast-1	Asia Pacific (Tokyo) Region
ap-southeast-1	Asia Pacific (Singapore) Region
ap-southeast-2	Asia Pacific (Sydney) Region
eu-west-1	EU (Ireland) Region
sa-east-1	South America (Sao Paulo) Region
us-east-1	US East (Northern Virginia) Region
us-west-1	US West (Northern California) Region
us-west-2	US West (Oregon) Region

- Source Variables

Override variables found in ec2.ini and used by the inventory update script. For a detailed description of these variables [view ec2.ini in the Ansible github repo.](#)

Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.

- Update Interval

Instruct the AWX server to automatically run the inventory update process the selected number of minutes from the last run.

With a value set, task manager will periodically compare the amount of elapsed time from the last run. If enough time has elapsed, it will go ahead and start an inventory update process.

- Update Options

- Overwrite

When checked all child groups and hosts not found on the remote source will be deleted from the local inventory.

Unchecked any local child hosts and groups not found on the external source will remain untouched by the inventory update process.

- Overwrite Variables

If checked, all variables for child groups and hosts will be removed and replaced by those found on the external source.

When not checked a merge will be performed, combining local variables with those found on the external source.

- Update on Launch

Each time a job runs using this inventory, refresh the inventory from the selected source before executing job tasks.

Rackspace Cloud Servers

To configure a group for Rackspace, select Rackspace Cloud Servers and enter the following details:

The screenshot shows the configuration interface for a new inventory source. The 'Source' dropdown is set to 'Rackspace Cloud Servers'. The 'Cloud Credential' section has a search bar and a dropdown menu. The 'Regions' section contains a text input field. The 'Update Interval' section shows 'none' selected from a dropdown. The 'Update Options' section includes three checkboxes: 'Overwrite', 'Overwrite Variables', and 'Update on Launch'.

- Cloud Credential

Choose from an existing Credential. For more information, see the [Credentials](#) section.

- Regions

A comma-separated list of regions matching the regions used at AWS. Only hosts associated with the list of regions will be included in the update process.

- Update Interval

Instruct the AWX server to automatically run the inventory update process the selected number of minutes from the last run.

With a value set, task manager will periodically compare the amount of elapsed time from the last run. If enough time has elapsed, it will go ahead and start an inventory update process.

- Update Options

- Overwrite

When checked all child groups and hosts not found on the remote source will be deleted from the local inventory.

Unchecked any local child hosts and groups not found on the external source will remain untouched by the inventory update process.

- Overwrite Variables

If checked, all variables for child groups and hosts will be removed and replaced by those found on the external source.

When not checked a merge will be performed, combining local variables with those found on the external source.

- Update on Launch

Each time a job runs using this inventory, refresh the inventory from the selected source before executing job tasks.

Hosts

Hosts are managed by first selecting Hosts from the menu and then by selecting the desired group or by selecting Hosts from the Edit menu from the Inventory tab.

The screenshot shows the 'Groups' screen in Ansible. A context menu is open over the 'Groups' button in the top navigation bar. The menu items are 'Properties', 'Hosts' (which is highlighted in blue), and 'Groups'. A red arrow points from the text 'Select Hosts from the Groups screen' to the 'Hosts' menu item. Below the menu, the main content area displays a table titled 'All Hosts' with columns: Group, Failed Hosts, Status, and Last Updated. One entry is shown: 'CMS Web' with 0/1 failed hosts and n/a status. Buttons for 'Update' and 'Cancel' are at the bottom right.

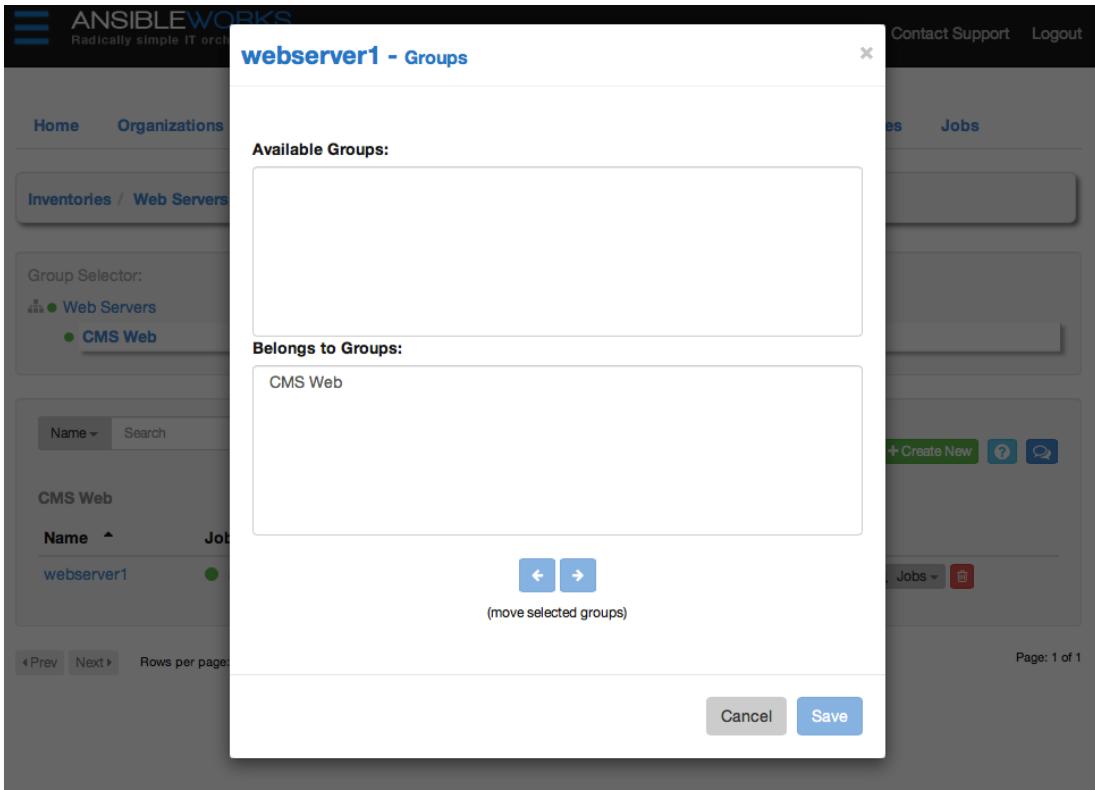
The screenshot shows the 'Inventories' screen in Ansible. A context menu is open over the 'Hosts' button in the top right corner of the table header. The menu items are 'Properties', 'Hosts' (highlighted in blue), and 'Groups'. A red arrow points from the text '...or by selecting Hosts from the Inventories screen' to the 'Hosts' menu item. Below the menu, the main content area displays a table with columns: #, Name, Organization, Failed Hosts, and Status. One entry is shown: '1. Web Servers' under 'Organization Bender Products Ltd' with 0/1 failed hosts and 0/0 status. Buttons for 'Jobs', 'Edit', and 'Delete' are visible, along with a 'Create New' button. The bottom right corner shows 'Page: 1 of 1'.

The host list may be sorted and searched by Name or Groups and filtered by hosts that are disabled, by hosts with failed jobs, and hosts synchronized with an external source.

The screenshot shows the AWX interface with the path 'Inventories / Web Servers / Hosts'. A 'Group Selector' sidebar on the left shows 'Web Servers' and 'CMS Web'. The main area displays a table for 'CMS Web' hosts. The table has columns: Name (sorted by name), Job Status (sorted by status), Enabled (sorted by enabled status), and Groups. One host, 'webserver1', is listed with a green success icon, checked enabled status, and 'CMS Web' in the groups column. Action buttons at the top right include 'Copy', '+ Create New', and others.

This list displays information about each host and provides for several actions:

- Name – Opens the Host Properties dialog
- Job Status – Can be any of pending, running, successful, or failed.
- Enabled – A toggle indicating whether the host is enabled to receive jobs from AWX. Click to toggle this setting.
- Groups – The list of groups that this host is a member of. Select the group editor button  to modify group membership for this host.
- Jobs – The Jobs menu navigates to the following Jobs views for the selected host:
 - All jobs
 - All host summaries
 - Latest job
 - Latest job events
 - Latest host summary
- Delete – Removed the host from AWX. *This operation is not reversible!*

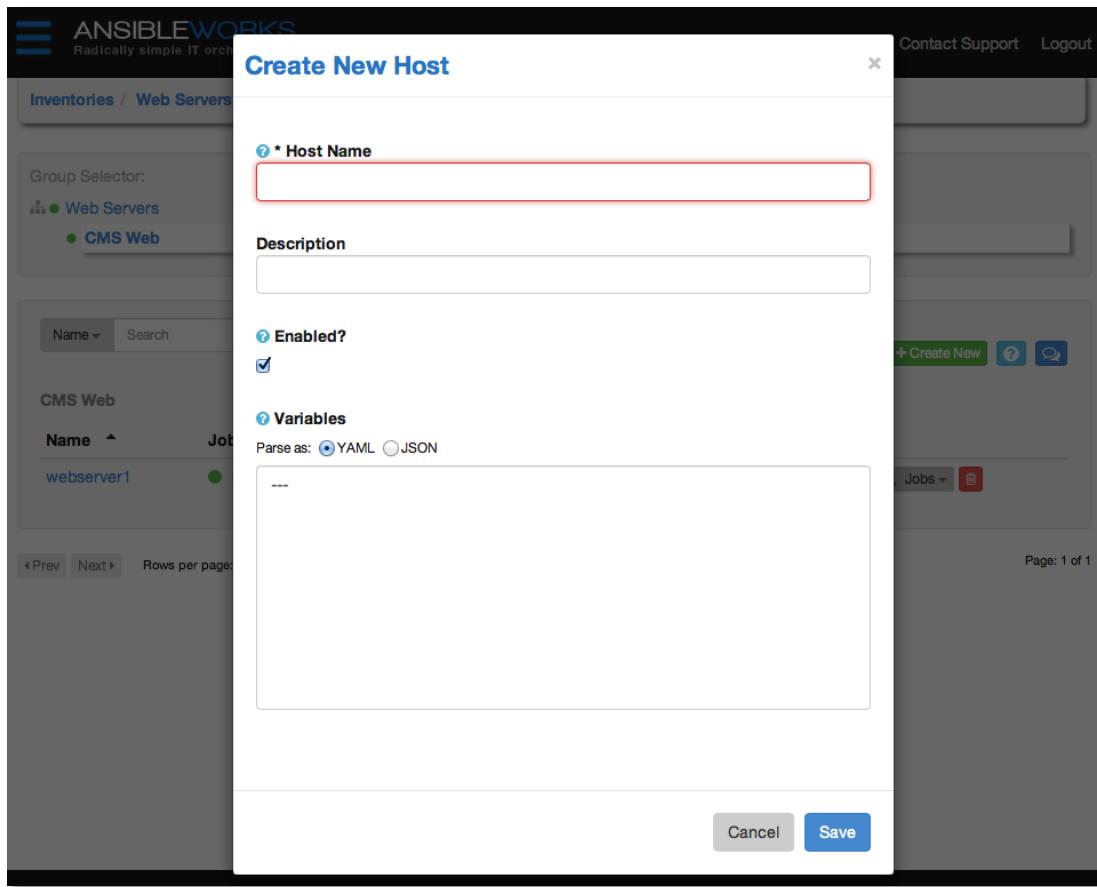


Add a new host

To create a new host and add it to an existing group, *first select the group*, and then click +Create New.

The screenshot shows the 'CMS Web' host list. At the top, there is a 'Group Selector' with 'Web Servers' and 'CMS Web' options. Below the selector is a search bar and a table with columns: Name, Job Status, Enabled, and Groups. The 'Groups' column for the 'webserver1' host shows 'CMS Web'. A red arrow points to the green '+ Create New' button at the bottom right of the host list table.

This will open to the Create Host dialog.



Enter the appropriate details into the following fields and click Save:

- Host Name - The hostname or IP address of the host
- Description
- Variables

Variable definitions and values to be applied to the selected host. Enter variables using either JSON or YAML syntax, using the radio button to toggle between JSON or YAML.

Job Templates

A job template is a definition and set of parameters for running an Ansible job. Job templates are useful to execute the same job many times. While the REST API allows executing jobs directly, the AWX User Interface requires first creating a job template.

This menu opens a list of the job templates that are currently available. The job template list may be sorted and searched by Name or Description. The Job Templates tab also enables the user to modify, launch, and remove a job template.

The screenshot shows a navigation bar with tabs: Home, Organizations, Users, Teams, Credentials, Projects, Inventories, Job Templates (which is selected and highlighted in blue), and Jobs. Below the navigation bar is a search bar with dropdown menus for 'Name' and 'Search'. To the right of the search bar are buttons for '+ Create New' and a magnifying glass icon. The main content area displays a table with one row. The columns are labeled '#', 'Name', and 'Description'. The first row contains the value '1.' under '#', 'HelloWorld' under 'Name', and 'hello world!' under 'Description'. To the right of this row are three buttons: 'Edit' (grey), 'Launch' (green), and 'Delete' (red).

To create a new job template click the +Create New button.

The screenshot shows a 'Create Job Templates' form. At the top left is the breadcrumb 'Job Templates / Create Job Templates'. The form contains several input fields: 'Name' (required), 'Description', 'Job Type' (set to 'Run'), 'Inventory', 'Project', 'Playbook', 'Credential', 'Cloud Credential', 'Forks' (set to 0), 'Limit', and 'Verbosity' (set to 'Default'). On the right side of the form are two sections: 'Extra Variables' (with a 'Parse as:' dropdown set to 'YAML') and 'Job Tags'. At the bottom of the form are 'Save' and 'Reset' buttons.

Enter the appropriate details into the following fields:

- Name (required)
- Description
- Job Type: Jobs may be of type Run or Check:

- Run: Execute the playbook when launched, running Ansible tasks on the selected hosts.
- Check: Execute the playbook in dry-run mode, reporting “changed” when an item would be changed, but not actually making changes.

More documentation on job types may be found in the [Advanced Playbook](#) section of the Ansible documentation.

- Inventory: Choose the inventory to be used with this job template from the inventories available to the currently logged in AWX user.
- Playbook: Choose the playbook to be launched with this job template from the available playbooks. This menu is automatically populated with the names of the playbooks found in the project base path. For example, a playbook named “jboss.yml” in the project path will appear in the menu as “jboss”.
- Credential: Choose the credential to be used with this job template from the credentials available to the currently logged in AWX user.
- Cloud Credential: Choose the credential to be used with this job template from the credentials available to the currently logged in AWX user.
- Forks: The number of parallel or simultaneous processes to use while executing the playbook. A value of zero will use the Ansible default setting, which is 5 parallel processes unless overridden in /etc/ansible/ansible.cfg.
- Limit: A host pattern to further constrain the list of hosts that will be managed or affected by the playbook. Multiple patterns can be separated by colons (“:”). As with core Ansible, “a:b” means “in group a or b”, “a:b:&c” means “in a or b but must be in c”, and “a:!b” means “in a, and definitely not in b”.

For more information and examples see the “Selecting Targets” section under [Inventory and Patterns](#) in the Ansible documentation.

- Verbosity: Control the level of output Ansible will produce as the playbook executes. Set the verbosity to any of Default, Verbose, or Debug. This only appears in the “details” report view. Verbose logging will include the output of all commands. Debug logging is exceedingly verbose and will include information on SSH operations that can be useful in certain support instances. Most users will not need to see debug mode output.
- Extra Variables: Pass extra command line variables to the playbook. This is the “-e” or “--extra-vars” command line parameter for ansible-playbook that is documented in the Ansible documentation at [Passing Variables on the Command Line](#). Provide key/value

pairs using either YAML or JSON. These variables have a maximum value of precedence and will override other variables specified elsewhere. An example value might be:

```
---
```

```
git_branch: production
release_version: 1.5
```

- Job Tags: Provide a comma-separated list of playbook tags with which to filter this job template. More documentation on tags may be found in the [Advanced Playbook](#) section of the Ansible documentation.
- Allow Callbacks: Enable a host to call back to AWX via the AWX API and invoke the launch of a job from this job template.

Callbacks are an important feature of AWX that allow a host to initiate a job launch, rather than waiting for a user to launch a job to manage the host. This provides for automatically configuring a system after it has been provisioned by another system (such as AWS auto-scaling, or a OS provisioning system like kickstart or preseed) or for launching a job programmatically without invoking the AWX API directly.

To enable callbacks, check the Allow Callbacks checkbox. A unique host key will be displayed that corresponds to this job template. The host key may be reused across multiple hosts to apply this job template against multiple hosts.

The URL will look like the following:

http://your.server.com:999/api/v1/job_templates/1/callback/

The request from the host must be a POST. Here is an example using curl (all on a single line):

```
[root@localhost ~]# curl --data "host_config_key=5a8ec154832b780b9bdef1061764ae5a"
http://your.server.com:999/api/v1/job\_templates/1/callback/
```

The requesting host must be defined in your inventory. If AWX fails to locate the host either by name or IP address in one of your defined inventories, the request will be denied.

Successful requests will result in an entry on the Jobs tab, where the results and history can be viewed.

Should you wish to control what hosts are able to request configuration, the key may be changed at any time.

When you have completed configuring the job template, select Save.

When editing an existing job template, by clicking the job template name or the Edit button, the bottom of the screen will display a list of all of the jobs that have been launched from this template. Please see the section [Jobs](#) for more information about this interface.

Launching Jobs

To launch a job template, click Launch.

If credentials require the user to enter additional information, such as a password or passphrase, a dialog will request this information.

Upon launch, AWX will automatically redirect the web browser to the Jobs tab.

Jobs

A job is an instance of AWX launching an Ansible playbook against an inventory of hosts.

The Jobs tab displays a list of jobs, both those currently running as well as a history of all jobs that have ever been launched. The list of jobs may be sorted and searched by Job ID, Name or Status.

The screenshot shows the AWX interface with the 'Jobs' tab selected in the top navigation bar. The main content area is titled 'Jobs'. At the top of this area is a search bar with dropdown menus for 'Job ID' and 'Search', and a refresh icon. Below the search bar is a table header with columns: 'Job ID', 'Date', 'Job Template', and 'Status'. A single job entry is listed: Job ID 8, Date 11/19/13 19:31:53, Job Template HelloWorld, Status successful (indicated by a green dot). To the right of the status are buttons for 'View', 'Launch' (highlighted in blue), and 'Delete'.

- The Job ID and Date values are automatically generated by AWX.
- Job ID: A unique integer that identifies a specific job.

- Date: The time when the job was initiated.
- Job Template: The Job Template from which this job was launched.
- Status: Can be any of pending, running, successful, or failed.

The Jobs tab does not automatically refresh to show the current status of any pending jobs. To refresh the job queue click the Refresh button.

There are several actions available for each job, all of which are detailed below:

- View
- Launch
- Delete

Summary

Selecting Summary will display a list of all of the hosts that this job was launched against, as well as useful summary information about each Ansible playbook task that was executed.

#	Host	Status	Success	Changed	Failure	Unreachable	Skipped
1.	webserver1	success	2	1	0	0	0

This information includes:

- Host: the host that the playbook was launched against. The host Name is a hotlink to the Events for this job and host. Events are described below.
- Status: Can be any of pending, running, successful, or failed.

The summary also display the number of playbook tasks that completed with an outcome of:

- Success: the playbook task returned “Ok”.
- Changed: the playbook task actually executed. Since Ansible tasks should be written to be idempotent, tasks may exit successfully without executing anything on the host. In these cases, the task would return Ok, but not Changed.
- Failure: the task failed. Further playbook execution was stopped for this host.
- Unreachable: the host was unreachable from the network or had another fatal error associated with it.

- Skipped: the playbook task was skipped because no change was necessary for the host to reach the target state.

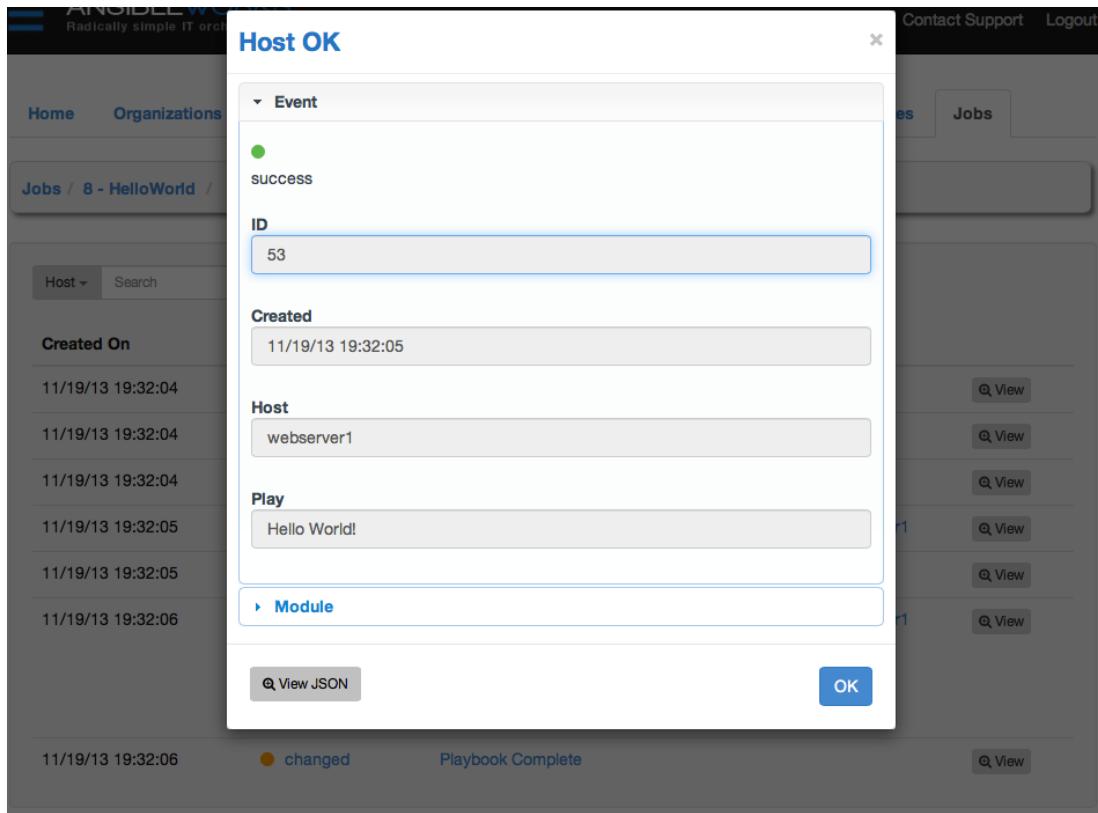
Events

Selecting Events will display details about each of the AWX events that comprised the job, including specific details on each playbook task.

Created On	Status	Event	Host
11/19/13 19:32:04	● changed	⊕ Playbook Started	
11/19/13 19:32:04	● changed	⊕ Play Started (Hello World!)	
11/19/13 19:32:04	● success	⊕ Gathering Facts	
11/19/13 19:32:05	● success	Host OK	webserver1
11/19/13 19:32:05	● changed	⊕ Task Started (Hello World!)	
11/19/13 19:32:06	● changed	Host OK Output: Hi! AWX is working	webserver1
11/19/13 19:32:06	● changed	Playbook Complete	

This information includes:

- Date: the timestamp of when this job was initiated.
- Status: Can be any of changed, success, skipped, or failed.
- Event: the name of the event, including the playbook task name, if appropriate. Clicking the event name expands and collapses the tree of events and sub-events.
- Host: the host targeted by the event.
- View: Clicking the View button on any event will display additional information about the event.



This information includes:

- Status: can be any of changed, success, skipped, or failed
- ID: the Job ID
- Created: the timestamp of when this job was initiated
- Host: the host targeted by the event
- Play: The name of the play
- Module: the name of the module invoked by the play

View: Clicking the View button on any event will display additional information about the event.

Status

Selecting the Status option from the View menu will display the job status. You can also view the same information by clicking the any of the Job ID or Status fields for the job.

The screenshot shows the AWX interface for a completed job. At the top, the navigation bar includes 'Jobs' / '8 - HelloWorld' / 'Status'. Below this, the 'Job Status' section indicates 'successful' with a green dot. The 'Date' section shows '11/19/13 19:31:53'. The 'Standard Out' section displays the Ansible playbook output:

```
SSH password:  
PLAY [Hello World!] *****  
GATHERING FACTS *****  
ok: [webserver1]  
TASK: [Hello World!] *****  
changed: [webserver1]  
PLAY RECAP *****  
webserver1 : ok=2 changed=1 unreachable=0 failed=0
```

At the bottom of the main content area, there is a link labeled 'Job Template'.

These details will include:

- **Job Status:** can be any of pending, running, successful, or failed
- **Date:** the timestamp of when the job was initiated by AWX
- **Standard Out:** displays the results of running the playbook from the standard out of the AWX server. This shows the same information you would see if you ran the Ansible playbook using Ansible from the command line.

The details of the job template that the job was launched with are shown at the bottom of this screen. This is especially useful in cases where the job template has been modified or removed since the job was executed. AWX retains this complete history of the job forever.

At the top of this screen is a menu that provides another path to navigate to the Summary and Events details for this job. Clicking Summary will display the job host summary, as described above. Clicking Events will display the events list for the job, as described above.

Launch

This button launches this job again, with the same parameters as it originally ran with. Any modification to the job template after job was launched will not be used during this re-launch.

Delete Job

This button deletes the job from AWX.

WARNING: this action cannot be undone.

Best Practices

Ansible file and directory structure

Please review the Ansible best practices from the Ansible documentation at <http://www.ansibleworks.com/docs/bestpractices.html>.

Playbooks should not use the “vars_prompt” feature, as AWX does not interactively allow Q&A for “vars_prompt” questions at this time.

Inventory Management

Keeping variable data along with the objects in AWX (see the inventory editor) is encouraged, rather than using “group_vars/” and “host_vars/”. If you use the “awx-manage inventory_import” command on an inventory source, it can sync such variables with the database.

Scaling

Using the “callback” feature to allow newly booting instances to request configuration is very useful for auto-scaling scenarios or provisioning integration.

Consider setting “forks” on a job template to larger values to increase parallelism of execution runs.

CI Integration / Continuous Deployment

For a Continuous Integration system, such as Jenkins, to spawn an AWX job, it should make a curl request to a job template. The credentials to the job template should not require prompting for any particular passwords. Using the API to spawn jobs is covered in the API section.

Security

The multi-tenancy features of AWX are sufficient to control who can run certain projects on what systems, but are not intended to hide project content from other teams. For instance, you could easily control that engineering could not push to production.

All playbooks are executed via the “awx” filesystem user. Users who have access to edit playbooks need to be trusted as playbooks do have access to the filesystem and all that implies.

Users concerned about credential security may choose to upload locked SSH keys and set the unlock password to “ask”, or choose to have the system prompt them for SSH credentials or sudo passwords rather than having the system store them in the database.

Troubleshooting

AWX server errors are logged to syslog. Apache web server errors are logged to the httpd error log.

Client-side issues may be explored using the JavaScript console built into most browsers and should be reported to awx@ansibleworks.com.

If requested by support, super users may also edit the “Django Admin” browser at <http://<AWX server name>/admin> using a super user login. Do not do this unless requested by AnsibleWorks support as you may remove objects and bypass the business logic of the application. After logging in to the admin view, users may need to clear their cookies to successfully log back into the main application.

Glossary

Organization: A logical collection of Users, Teams, Projects, and Inventories. The highest level in the AWX object hierarchy. See this description of the AWX [hierarchy](#).

User: An AWX operator with associated permissions and credentials.

Organization Administrator: An AWX user with the rights to modify the Organization's membership and settings, including making new users and projects within that organization. An organization admin can also grant permissions to other users within the organization.

Team: A sub-division of an Organization with associated Users, Projects, Credentials, and Permissions. Teams provide a means to implement role-based access control schemes and delegate responsibilities across Organizations.

Project: A logical collection of Ansible playbooks, represented in AWX.

Inventory: A collection of hosts against which Jobs may be launched.

Credentials: Authentication details that may be utilized by AWX to launch jobs against machines, to synchronize with inventory sources, and to import project content from a version control system.

Job Template: The combination of an Ansible playbook and the set of parameters required to launch it, designed to be reusable across hosts.

Job: The instantiation of a Job Template; the launch of an Ansible playbook.

Permissions: The set of privileges assigned to Users and Teams that provide the ability to read, modify, and administer Projects, Inventories, and other AWX objects.

Host: A system managed by AWX, which may include a physical, virtual, or cloud-based server, a network router, switch, or firewall, a storage device, or any unique system managed by AWX. Typically an operating system instance.

Playbook: An Ansible playbook.

Super User: An admin of the AWX server who has permission to edit any object in the system, whether associated to any organization. Super users can create organizations and other super users.

Using LDAP with AWX

As of the 1.3 release of AWX, administrators may utilize LDAP as a source for authentication information for AWX users. At this time, only user authentication is provided and not synchronization of user permissions, credentials, or team membership, however organization membership (and who is an organization admin) can be synchronized.

When so configured, a user who logs in with an LDAP username and password will automatically get an AWX account created for them and they can be automatically placed into multiple organizations as either regular users or organization administrators.

By default, if users are created via an LDAP login, by default they cannot change their username, firstname, lastname, or set a local password for themselves. This is also tunable to restrict editing of other field names.

Currently, LDAP integration for AWX is configured in the file “/etc/awx/settings.py.” No configuration is accessible via the AWX user interface. Please, review the comments in that file for information on LDAP configuration and let us know at awx@ansibleworks.com if you need assistance.

Administration of AWX

Certain command line commands are available for management of AWX. In the future, some of these may be made available via GUI tools as well, and they may be augmented with other commands. Here is a useful subset. Do not run other awx-manage commands unless instructed by AnsibleWorks Support.

NOTE: These commands should be run as the ‘awx’ user.

awx-manage inventory_import [--help]

The inventory_import command is used to synchronize an AWX inventory object with a text-based inventory file, dynamic inventory script, or a directory of one or more of the above as supported by core Ansible.

When running this command, specify either an --inventory-id or --inventory-name, and the path to the Ansible inventory source is given by --source.

By default, inventory data already stored in AWX will be blended with data from the external source. To use only the external data, specify --overwrite. To specify that any existing hosts get variable data exclusively from the --source, specify --overwrite-vars. The default behavior will

add any new variables from the external source, overwriting keys that do not already exist, but preserving any variables that were not sourced from the external data source.

awx-manage cleanup_deleted [--help]

When objects in AWX are deleted, they are not actually removed from the database. This is to ensure audit log history and referential integrity. To restore space in the database, run the ‘cleanup_deleted’ command with the --days=N flag specifying to remove objects flagged as deleted that are older than N days. You may use the --dry-run flag to list what would be deleted prior to running the command in “real” mode. You may wish to run this command nightly on cron with a value of --days=30.

API

Firebug, Chrome, and Charles Proxy

This document gives a basic understanding of the API, though you may wish to see what API calls AWX makes in sequence. To do this, using the UI from Firebug or Chrome with developer plugins is useful, though Charles Proxy (<http://www.charlesproxy.com/>) is also an outstanding visualizer that you may wish to investigate. It is commercial software but can insert itself as, for instance, an OS X proxy and intercept both requests from web browsers but also curl and other API consumers.

Browseable API

AWX features a browseable API feature.

You can visit the API in a browser at `http://<AWX server name>/api` and then click on various links in the API to explore related resources.

The screenshot shows the AnsibleWorks REST API documentation for the `/api/` endpoint. At the top, there's a header bar with the AnsibleWorks logo and the text "REST API". Below the header, the title "REST API" is displayed with a question mark icon. To the right of the title are buttons for "OPTIONS", "GET", and a dropdown menu. A sub-header "GET /api/" is present. The main content area shows the response headers and the JSON payload. The response headers are:

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS
```

The JSON payload is:

```
{
  "available_versions": {
    "v1": "/api/v1/"
  },
  "description": "AWX REST API",
  "current_version": "/api/v1/"
}
```

This screenshot is identical to the one above, showing the AnsibleWorks REST API documentation for the `/api/` endpoint. It features the same header, title, and response details, including the JSON payload:

```
{
  "available_versions": {
    "v1": "/api/v1/"
  },
  "description": "AWX REST API",
  "current_version": "/api/v1/"
}
```

Version 1[®]

GET /api/v1/

OPTIONS GET ▾

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

{
  "me": "/api/v1/me/",
  "authtoken": "/api/v1/authtoken/",
  "jobs": "/api/v1/jobs/",
  "users": "/api/v1/users/",
  "organizations": "/api/v1/organizations/",
  "job_templates": "/api/v1/job_templates/",
  "teams": "/api/v1/teams/",
  "hosts": "/api/v1/hosts/",
  "inventory": "/api/v1/inventories/",
  "groups": "/api/v1/groups/",
  "credentials": "/api/v1/credentials/",
  "config": "/api/v1/config/",
  "projects": "/api/v1/projects/"
}
```

You can also PUT and POST on the specific API pages if you so desire by formatting JSON in the various text fields.

Media type application/json

Content

POST

Conventions

With all of the basics about how to explore the API and database objects out of the way, it's now time for some general API info.

AWX uses a standard REST API, rooted at /api/ on the server. The API is versioned for compatibility reasons but only /api/v1/ is presently available. By querying /api you can see information about what API versions are available.

All data is JSON by default. You may have to specify the content/type on POST or PUT requests accordingly.

All URIs should end in "/" or you will get a 301 redirect.

Sorting

Assume the following URL, http:// <AWX server name>/api/v1/groups/

In order to sort the groups by name, access the following URL variation:

http://<AWX server name>/api/v1/groups/?order_by=name

You can order by any field in the object.

Filtering

Any collection is what the system calls a "queryset" and can be filtered via various operators.

For example, to find the groups that contain the name "foo":

http://<AWX server name>/api/v1/groups/?name__contains=foo

To do an exact match:

http://<AWX server name>/api/v1/groups/?name=foo

If a resource is of an integer type, you must add "__int" to the end to cast your string input value to an integer, like so:

http://<AWX server name>/api/v1/arbitrary_resource/?x__int=5

Related resources can also be queried, like so:

http://<AWX server name>/api/v1/groups/?user__firstname__icontains=john

This will return all groups with users with names that include the string "John" in them.

You can also filter against more than one field at once:

```
http://<AWX server name>/api/v1/groups/?user__firstname__icontains=john&group__name__icontains__foo
```

This will find all groups containing a user whose name contains John where the group contains the string foo.

For more about what types of operators are available, see:

<https://docs.djangoproject.com/en/dev/ref/models/querysets/>

You may also wish to watch the API as the UI is being used to see how it is filtering on various criteria.

Pagination

Responses for collections in the API are paginated. This means that while a collection may contain tens or hundreds of thousands of objects, in each web request, only a limited number of results are returned for API performance reasons.

When you get back the result for a collection you will see something like:

```
{'count': 25, 'next': 'http://testserver/api/v1/some_resource?page=2', 'previous': None, 'results':  
[ ... ]}
```

Where to get the next page, simply request the page given by the 'next' URL.

To request more items per page, pass the page size query string:

```
http://<AWX server name>/api/v1/some_resource?page_size=50
```

The serializer is quite efficient, but you should probably not request page sizes beyond a couple of hundred.

The user interface uses smaller values to avoid the user having to do a lot of scrolling.

Read Only Fields

Certain fields in the REST API are marked read only. These usually include the URL of a resource, the ID, and occasionally some internal fields. For instance, the 'created_by' attribute of each object indicates which user created the resource, and cannot be edited.

If you post some values and notice they are not changing, these fields may be read only.

API Example of Triggering A Job

For an example of how to launch an existing job template from a script, for integration with a Continuous Integration system or other program, please review the awx-cli project at <https://github.com/ansible/awx-cli>.

The awx-cli program is a command line utility used to send commands to the AWX API.