## Overview

Ansible AWX provides a Graphical User Interface for executing Ansible playbooks. It also has many other features around access control, job scheduling, and viewing job history.

In this lab, we will run AWX on our local workspace and then add the inventory and playbook we have developed so far. Then execute it from the interface.

[Overview](#_4l6y9k9n7exo)

[Instructions](#_hxr977vi6sdt)

[Task 0: Install AWX](#_hn4h731qqgqo)

[Step 1: Install Prerequisites](#_ogg0sfiv78u2)

[Step 2: Install AWX (with Ansible!)](#_i33kezgdqpt3)

[Step 3: View AWX](#_g657dt7s2de0)

[Task 1: Create a Job Template](#_xwkbs9hcd3ar)

[Step 1: Update the demo project](#_3aim99jnni0r)

[Step 2: Add our inventory](#_eh3ap63dkjoj)

[Step 3: Run ad hoc commands](#_jlckcwy0ssyt)

[Step 4: Upload your playbook](#_dxz6itl16u3m)

[Step 5: create a template that deploys the db portion](#_68akyljpqttc)

## 

## Instructions

Read this lab like a book, all text is there for a reason!

"→" denotes an action you must take

Use your favorite editor to edit files within the console. I suggest VI, nano, or emacs.

|  |
| --- |
| White boxes with black text denote commands and file contents |

|  |
| --- |
| Black boxes with green text denote example output |

### Task 0: Install AWX

#### Step 1: Install Prerequisites

The ansible playbook for AWX will not install some of the required dependencies. The below script should prepare your workspace.

→ Execute the following

|  |
| --- |
| sudo amazon-linux-extras install docker  sudo service docker start  sudo usermod -a -G docker $(whoami)  exec su -l $USER  sudo yum install -y libffi-devel  pip install -y --user docker docker-compose |

#### Step 2: Install AWX (with Ansible!)

→ Clone AWX

|  |
| --- |
| # Go to your home directory  cd ~  # Download awx  git clone <https://github.com/ansible/awx>  cd ansible/awx |

→ Update inventory, change the awx\_alternate\_dns\_servers line to match the following.

This will set AWX to use the google DNS servers so it can resolve websites like github.com.

|  |
| --- |
| awx\_alternate\_dns\_servers="8.8.8.8,8.8.4.4" |

→ Run the playbook

|  |
| --- |
| ansible-playbook -i inventory install.yml |

#### Step 3: View AWX

→ Go to <http://localhost> in your browser

The default administrator is 'admin' with the password 'password'

### Task 1: Create a Job Template

#### Step 1: Update the demo project

To ensure everything is set up correctly, take a look at the Demo Project under Projects in the navigation menu. This project is a link to the "https://github.com/ansible/ansible-tower-samples" git repository. You can use projects to allow AWX to clone repositories for use when creating inventories or job templates. For example, your playbook may live in a git repository and you simple want AWX to refer to that instead of maintaining its own copy.

→ Click the  "Get latest SCM revision" button for Demo Project

This will cause a job to run that will download the latest for the ansible-tower-samples project

→ Go to Jobs on the menu

You should see a Job named "Demo Project" with a green circle.

→ Click the latest job and view it's output

#### Step 2: Add our inventory

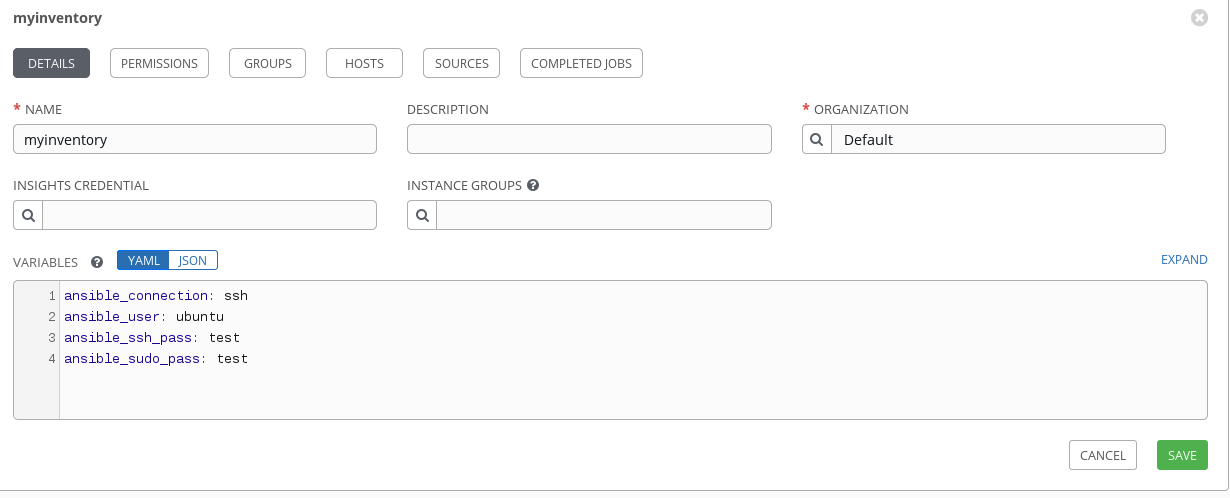
→ Go to Inventories -> 

→ Give your inventory a name

→ Create two groups, "web" and "db", just like we did in our inventory file

→ For both of our hosts, use the button under the hosts tab to add their IPs, also use the groups tab to set their correct groups.

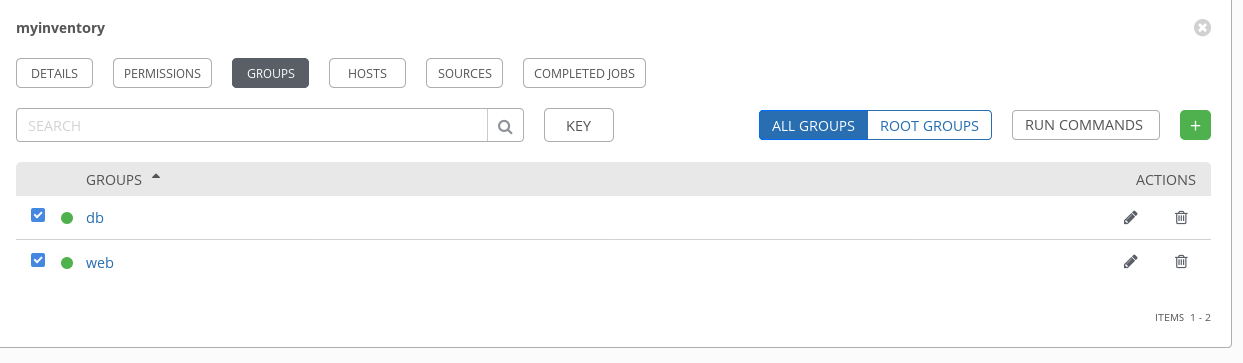
→ Add your ansible connection variables under variables for the inventory



#### Step 3: Run ad hoc commands

You can run ad hoc commands via AWX just like using the *ansible* command, but it is a little hard to find.

→ Inventories → Groups tab

→ Check the boxes for both groups

→ Click the RUN COMMANDS button in the top right

→ Choose "ping" as the module and click LAUNCH

This will create a new job running a single tasks for all selected hosts. Try it again but select DEBUG=3. You will see a lot more information in the Job log.

#### Step 4: Upload your playbook

**\*\*STOP HERE\*\* Ask your instructor where and how to put your playbook in a repository**

→ Go back to "Projects"

→ Click the green plus sign to add a project

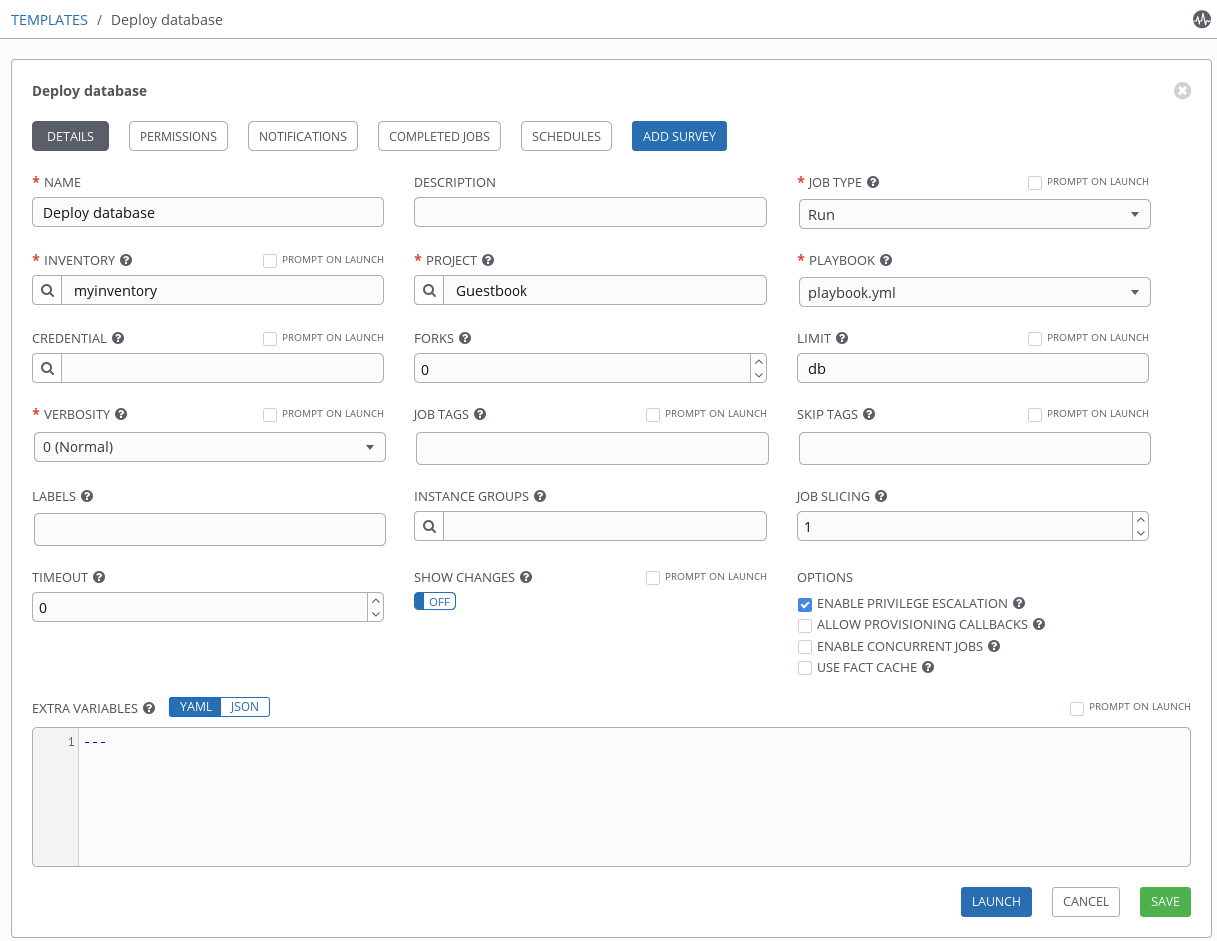
→ Ensure you have committed all of the changes in your playbook and pushed it to a repository

→ Put in the repository information

#### Step 5: Create a template that deploys the db portion

→ Templates → → Select "Job Template" in the dropdown

→ Update the following values. For inventory, use the one you created.



In the Job output, you should see some familiar text begin to appear as the job runs