



 **TechScalable**
Hands-on Guide:
Elementary Ansible

ANSIBLE

Contents

1	Ansible Installation steps on CentOS Server	3
1.1	Installing Ansible on CentOS workstation	3
2	Connecting to the Managed Hosts via SSH.....	5
2.1	Generating ssh keys.....	5
2.2	Copying SSH keys to both the managed hosts.....	5
3	Updating Ansible configuration with Host details	6
3.1	Updating hosts file	6
3.2	Validating connectivity with Managed hosts	6
4	Working with ad-hoc commands.....	8
4.1	Executing ad-hoc commands on managed hosts	8
5	Working with Ansible Playbooks – Configure webserver.....	11
5.1	Creating playbook to configure httpd web server	11
5.2	Executing playbook to configure webserver on centos servers.....	12
6	Performing file operations with Ansible Playbook.....	13
6.1	Creating playbook to perform basic file operations using file modules	13
6.2	Executing playbook to perform basic file operations using file modules.....	14
7	Setup Ansible Tower on RHEL Server	15
7.1	Download the Ansible Automation Platform Installation Program	15
8	Creating and Managing Ansible Tower Users	22
8.1	Login to ansible tower and Create Users.....	22
1.1	Managing Users Efficiently With Teams	25
9	Creating and Managing Inventories and Credentials	29
9.1	Create a new inventory called Dev within default organisation	29
1.1	Create a new Credentials for users with following information	32
10	Creating A Project for Ansible Playbooks	34
1.2	Creating a project.....	34
1.3	Create an inventory for this project.....	35
1.4	Create credentials to access the host.....	37
11	Run Ad hoc command from Ansible Tower	42
11.1	Go to inventory to run ad hoc commands.....	42
12	Install Docker on Host machine	45
12.1	Create a new Project.....	45

1 Ansible Installation steps on CentOS Server

1.1 Installing Ansible on CentOS workstation

1. Before installation of any package, we will first Update Software Repositories

```
$ sudo yum update -y
```

```
[allen@ansible-cn ~]$ sudo yum update -y
Failed to set locale, defaulting to C
Loaded plugins: langpacks
base-openlogic
extras-openlogic
updates-openlogic
base
extras
openlogic
updates
(1/9): base-openlogic/7/x86_64/group_gz          3.1 kB  00:00:00
(2/9): extras-openlogic/7/x86_64/primary_db      2.5 kB  00:00:00
(3/9): base/7/x86_64/group_gz                   2.6 kB  00:00:00
(4/9): openlogic/7/x86_64/primary_db            3.6 kB  00:00:00
(5/9): base-openlogic/7/x86_64/primary_db        2.9 kB  00:00:00
(6/9): base/7/x86_64/primary_db                 2.9 kB  00:00:00
(7/9): updates-openlogic/7/x86_64/primary_db     153 kB  00:00:00
(8/9): extras/7/x86_64/primary_db                242 kB  00:00:00
(9/9): updates/7/x86_64/primary_db               153 kB  00:00:00
                                                | 33 kB  00:00:00
                                                | 6.1 MB 00:00:00
                                                | 6.1 MB 00:00:00
                                                | 9.5 MB 00:00:00
                                                | 242 kB 00:00:00
                                                | 9.5 MB 00:00:01
```

2. Install the Extra Packages for Enterprise Linux (EPEL) repository. The EPEL repository contains various system packages, including the Ansible package, which you will install in the next step.

```
$ sudo yum install epel-release -y
```

```
Complete!
[allen@ansible-cn ~]$ sudo yum install epel-release
Failed to set locale, defaulting to C
Loaded plugins: langpacks
Resolving Dependencies
--> Running transaction check
--> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version       Repository      Size
=====
Installing:
epel-release      noarch   7-11         extras-openlogic 15 k

Transaction Summary
=====
Install 1 Package

Total download size: 15 k
Installed size: 24 k
Is this ok [y/d/N]:
```

3. Once the repository is complete, install the Ansible package.

```
$ sudo yum install ansible -y
```

```
[allen@ansible-cn ~]$ sudo yum install ansible -y
Failed to set locale, defaulting to C
Loaded plugins: langpacks
epel/x86_64/metalink
epel
(1/3): epel/x86_64/group_gz
(2/3): epel/x86_64/updateinfo
(3/3): epel/x86_64/primary_db
Resolving Dependencies
--> Running transaction check
--> Package ansible.noarch 0:2.9.23-1.el7 will be installed
--> Processing Dependency: python-httplib2 for package: ansible-2.9.23-1.el7.noarch
--> Processing Dependency: python-paramiko for package: ansible-2.9.23-1.el7.noarch
--> Processing Dependency: python2-cryptography for package: ansible-2.9.23-1.el7.noarch
--> Processing Dependency: python2-jmespath for package: ansible-2.9.23-1.el7.noarch
--> Processing Dependency: sshpass for package: ansible-2.9.23-1.el7.noarch
--> Running transaction check
--> Package python-paramiko.noarch 0:2.1.1-9.el7 will be installed
--> Package python2-cryptography.x86_64 0:1.7.2-2.el7 will be installed
--> Processing Dependency: python-idna >= 2.0 for package: python2-cryptography-1.7.2-2.el7.x86_64
--> Processing Dependency: python-cffi >= 1.4.1 for package: python2-cryptography-1.7.2-2.el7.x86_64
--> Processing Dependency: python-enum34 for package: python2-cryptography-1.7.2-2.el7.x86_64
--> Package python2-httplib2.noarch 0:0.18.1-3.el7 will be installed
--> Package python2-jmespath.noarch 0:0.9.4-2.el7 will be installed
--> Package sshpass.x86_64 0:1.86-2.el7 will be installed
--> Running transaction check
--> Package python-cffi.x86_64 0:1.6.0-5.el7 will be installed
--> Processing Dependency: python-pycparser for package: python-cffi-1.6.0-5.el7.x86_64
--> Package python-enum34.noarch 0:1.0.4-1.el7 will be installed
--> Package python-idna.noarch 0:2.4-1.el7 will be installed
--> Running transaction check
--> Package python-pycparser.noarch 0:2.14-1.el7 will be installed
--> Processing Dependency: python-ply for package: python-pycparser-2.14-1.el7.noarch
```

4. Now, confirm Ansible is installed by running ansible --version.

```
$ ansible --version
```

```
[allen@ansible-cn ~]$ ansible --version
ansible 2.9.23
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/allen/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.5 (default, Nov 16 2020, 22:23:17) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]
[allen@ansible-cn ~]$
```



2 Connecting to the Managed Hosts via SSH

2.1 Generating ssh keys

1. We will generate a new key and will share it with remaining two servers to communicate with them through our upcoming ansible commands and plays.

```
$ ssh-keygen
```

```
[[allen@ansible-cn ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/allen/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/allen/.ssh/id_rsa.
Your public key has been saved in /home/allen/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:gt6VtvHXujogS7tunYXW0xyz0/kyUPHDk0jzezs0saM allen@ansible-cn
The key's randomart image is:
+---[RSA 2048]----+
|          |
|          |
|          |
|          |
|          |
|          |
|          |
|          |
|          |
+---[SHA256]----+
[allen@ansible-cn ~]$
```

2.2 Copying SSH keys to both the managed hosts

1. Once the key is generated, copy that key to remaining two servers (managed hosts)
2. Repeat below step for both the managed hosts. We need to copy the ssh keys to both the servers.
3. Give the managed host's IP Address and while copying it, say yes to continue and then you will need to give the same password here, i.e., **Passw0rd@123j**

```
$ ssh-copy-id allen@20.106.147.196
```

```
[[allen@ansible-cn ~]$ ssh-copy-id allen@20.106.147.196
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/allen/.ssh/id_rsa.pub"
The authenticity of host '20.106.147.196 (20.106.147.196)' can't be established.
ECDSA key fingerprint is SHA256:QDcv0HlhM+XRJ/29DpYkNAoU+O1c6ExwpLmGGZiwtJ8.
ECDSA key fingerprint is MD5:77:58:00:ac:30:e8:e5:bb:45:39:47:f3:d4:d5:4f:9e.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
[allen@20.106.147.196's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'allen@20.106.147.196'"
and check to make sure that only the key(s) you wanted were added.
```

3 Updating Ansible configuration with Host details

3.1 Updating hosts file

1. Updating ansible hosts file with managed hosts details. We are updating the IP address of both the servers which we wish to configure using Ansible.
2. Add the IPs to the host file, and after that, the file will look as below:

```
$ sudo vi /etc/ansible/hosts
```

```
# This is the default ansible 'hosts' file.  
#  
# It should live in /etc/ansible/hosts  
#  
# - Comments begin with the '#' character  
# - Blank lines are ignored  
# - Groups of hosts are delimited by [header] elements  
# - You can enter hostnames or ip addresses  
# - A hostname/ip can be a member of multiple groups  
  
# Ex 1: Ungrouped hosts, specify before any group headers.  
  
20.106.147.196  
20.106.147.244  
  
## green.example.com  
## blue.example.com  
## 192.168.100.1  
## 192.168.100.10  
  
# Ex 2: A collection of hosts belonging to the 'webservers' group  
  
## [webservers]  
## alpha.example.org  
## beta.example.org  
## 192.168.1.100  
## 192.168.1.110
```



3.2 Validating connectivity with Managed hosts

1. Validate using Ansible command if we can list all the host, check if it gives 2 IPs that we added

```
$ ansible --list-hosts all
```

```
[allen@ansible-cn ~]$  
[allen@ansible-cn ~]$  
[allen@ansible-cn ~]$ ansible --list-hosts all  
  hosts (2):  
    20.106.147.196  
    20.106.147.244  
[allen@ansible-cn ~]$ █
```

2. Try pinging both the managed nodes if you can successfully do that using below command:

```
$ ansible all -m ping
```

```
[allen@ansible-cn ~]$  
[allen@ansible-cn ~]$  
[allen@ansible-cn ~]$ ansible all -m ping  
20.106.147.196 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
20.106.147.244 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
[allen@ansible-cn ~]$ █
```



4 Working with ad-hoc commands

4.1 Executing ad-hoc commands on managed hosts

1. Dumping free memory details of managed hosts using the **command module**

```
$ ansible all -m command -a "free -h"
```

```
[allen@ansible-cn ~]$ 
[[allen@ansible-cn ~]$ ansible all -m command -a "free -h"
20.106.147.196 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:       926M       256M       505M       6.8M       165M       526M
Swap:          0B          0B          0B
20.106.147.244 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:       926M       253M       508M       6.8M       164M       528M
Swap:          0B          0B          0B
[allen@ansible-cn ~]$
```

2. Using **user module** to add user and create home directory too on the managed hosts

```
$ ansible all -m user -a "name=ansible-new home=/home/ansible-new" --become
```

```
[[allen@ansible-cn ~]$ ansible all -m user -a "name=ansible-new home=/home/ansible-new" --become
20.106.147.244 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1001,
  "home": "/home/ansible-new",
  "name": "ansible-new",
  "shell": "/bin/bash",
  "state": "present",
  "system": false,
  "uid": 1001
}
20.106.147.196 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1001,
  "home": "/home/ansible-new",
  "name": "ansible-new",
  "shell": "/bin/bash",
  "state": "present",
  "system": false,
  "uid": 1001
}
[allen@ansible-cn ~]$
```

3. Using the **file module**, create a directory in one of the managed hosts with defined permissions

```
$ ansible 20.106.147.244 -m file -a  
"dest=/home/allen/ansible-dir mode=755 owner=allen  
state=directory"
```

```
[allen@ansible-cn ~]$ ansible 20.106.147.244 -m file -a "dest=/home/allen/ansible-dir mode=755 owner=allen state=directory"  
20.106.147.244 | CHANGED => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python"  
    },  
    "changed": true,  
    "gid": 1000,  
    "group": "allen",  
    "mode": "0755",  
    "owner": "allen",  
    "path": "/home/allen/ansible-dir",  
    "secontext": "unconfined_u:object_r:user_home_t:s0",  
    "size": 6,  
    "state": "directory",  
    "uid": 1000  
}  
[allen@ansible-cn ~]$
```

4. Use **file module** to create a file in a specified directory with proper permissions in the managed host

```
$ ansible 20.106.147.244 -m file -a "dest=/home/allen/ansible-dir/demo.txt mode=600 state=touch"
```

```
[allen@ansible-cn ~]$ ansible 20.106.147.244 -m file -a "dest=/home/allen/ansible-dir/demo.txt mode=600 state=touch"  
20.106.147.244 | CHANGED => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python"  
    },  
    "changed": true,  
    "dest": "/home/allen/ansible-dir/demo.txt",  
    "gid": 1000,  
    "group": "allen",  
    "mode": "0600",  
    "owner": "allen",  
    "secontext": "unconfined_u:object_r:user_home_t:s0",  
    "size": 0,  
    "state": "file",  
    "uid": 1000  
}  
[allen@ansible-cn ~]$
```

5. Use **lineinfile module** to add content to a file in the managed host

```
$ ansible 20.106.147.244 -m lineinfile -a  
'dest=/home/allen/ansible-dir/demo.txt line="This server is  
managed by Ansible'"
```

```
[allen@ansible-cn ~]$ ansible 20.106.147.244 -m lineinfile -a 'dest=/home/allen/ansible-dir/demo.txt line="This server is managed by Ansible"'  
20.106.147.244 | CHANGED => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python"  
    },  
    "backup": "",  
    "changed": true,  
    "msg": "line added"  
}  
[allen@ansible-cn ~]$ █
```

6. Let's ssh to the host in which we have created file to check if the file is created and content is updated.

```
$ ssh 20.106.147.244  
  
$ ls -lrt  
  
$ pwd  
  
$ cd ansible-dir  
  
$ ls -lrt  
  
$ cat demo.txt  
  
$ exit
```

```
[allen@ansible-cn ~]$  
[allen@ansible-cn ~]$ ssh 20.106.147.244  
Last login: Sat Jul 24 15:44:23 2021 from 137.117.45.185  
[allen@managedhost001 ~]$ ls -lrt  
total 0  
[drwxr-xr-x. 2 allen allen 22 Jul 24 15:37 ansible-dir  
[allen@managedhost001 ~]$ pwd  
/home/allen  
[allen@managedhost001 ~]$ cd ansible-dir/  
[allen@managedhost001 ansible-dir]$ ls -lrt  
total 4  
[-rw-----. 1 allen allen 34 Jul 24 15:37 demo.txt  
[allen@managedhost001 ansible-dir]$ cat demo.txt  
This server is managed by Ansible  
[allen@managedhost001 ansible-dir]$ █
```

5 Working with Ansible Playbooks – Configure webserver

5.1 Creating playbook to configure httpd web server

1. In this lab, we will be creating our first playbook to install and configure apache web server and will execute the playbook.
 - a. Create a custom index.html file
 - b. Create a playbook to install httpd package, update the custom index.html file and start the httpd service on the managed hosts
2. After creating the playbook, it should have below content:

```
---  
- hosts: all  
  tasks:  
    - name: install httpd package  
      yum:  
        name: httpd  
        update_cache: yes  
        state: present  
    - name: uploading custom index.html file to host  
      copy:  
        src: /home/allen/index.html  
        dest: /var/www/html  
    - name: Setting up attributes for file  
      file:  
        path: /var/www/html/index.html  
        owner: apache  
        group: apache  
        mode: 0644  
    - name: start httpd service  
      service:  
        name: httpd  
        state: started  
~  
~  
~  
~  
~  
~
```

5.2 Executing playbook to configure webserver on centos servers

1. Execute the playbook on both the managed hosts.

```
$ ansible-playbook apache.yml --become
```

```
[allen@ansible-cn ~]$  
[[allen@ansible-cn ~]$ ansible-playbook apache.yml --become  
  
PLAY [all] *****  
TASK [Gathering Facts] *****  
ok: [20.106.147.244]  
ok: [20.106.147.196]  
  
TASK [install httpd package] *****  
changed: [20.106.147.244]  
changed: [20.106.147.196]  
  
TASK [uploading custom index.html file to host] *****  
changed: [20.106.147.244]  
changed: [20.106.147.196]  
  
TASK [Setting up attributes for file] *****  
changed: [20.106.147.244]  
changed: [20.106.147.196]  
  
TASK [start httpd service] *****  
changed: [20.106.147.244]  
changed: [20.106.147.196]  
  
PLAY RECAP *****  
20.106.147.196 : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
20.106.147.244 : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

2. By doing curl, check the index.html on both the servers

```
$ curl 20.106.147.244
```

```
$ curl 20.106.147.196
```

```
[allen@ansible-cn ~]$  
[[allen@ansible-cn ~]$  
[[allen@ansible-cn ~]$ curl 20.106.147.244  
"Hi from Ansible configured server"  
[[allen@ansible-cn ~]$ curl 20.106.147.196  
"Hi from Ansible configured server"  
[allen@ansible-cn ~]$
```

6 Performing file operations with Ansible Playbook

6.1 Creating playbook to perform basic file operations using file modules

1. In this lab, we will be creating playbook to perform some file operations.
 - a. Use user module to create a user
 - b. We will create home directory for the testuser
 - c. Using file module we would create folder and file
 - d. We would assign proper owner and group to the file created
 - e. Use blockinfile module to write content to the newly created file
2. After creating the playbook, it should have below content:

```
---
- hosts: all
  tasks:
    - name: Creating a new user
      user:
        name: testuser
    - name: Creating a directory for the new user
      file:
        path: /home/testuser
        state: directory
    - name: creating a folder named ansible
      file:
        path: /home/testuser/ansible
        state: directory
    - name: creating a file within the folder ansible
      file:
        path: /home/testuser/ansible/hello.txt
        state: touch
    - name: Changing owner and group with permission for the file within the folder named ansible
      file:
        path: /home/testuser/ansible/hello.txt
        owner: root
        group: testuser
        mode: 0665
    - name: adding a block of string to the file created named hello.txt
      blockinfile:
        path: /home/testuser/ansible/hello.txt
        block: |
          This is line 1
          This is line 2
~
```

6.2 Executing playbook to perform basic file operations using file modules

1. Execute the playbook on both the managed hosts.

```
$ ansible-playbook file-operation.yml --become
```

```
[allen@ansible-cn ~]$  
[[allen@ansible-cn ~]$ ansible-playbook file-operation.yml --become  
  
PLAY [all] *****  
  
TASK [Gathering Facts] *****  
ok: [20.106.147.244]  
ok: [20.106.147.196]  
  
TASK [Creating a new user] *****  
changed: [20.106.147.244]  
changed: [20.106.147.196]  
  
TASK [Creating a directory for the new user] *****  
ok: [20.106.147.244]  
ok: [20.106.147.196]  
  
TASK [creating a folder named ansible] *****  
changed: [20.106.147.244]  
changed: [20.106.147.196]  
  
TASK [creating a file within the folder ansible] *****  
changed: [20.106.147.244]  
changed: [20.106.147.196]  
  
TASK [Changing owner and group with permission for the file within the folder named ansible] *****  
changed: [20.106.147.196]  
changed: [20.106.147.244]  
  
TASK [adding a block of string to the file created named hello.txt] *****  
changed: [20.106.147.244]  
changed: [20.106.147.196]  
  
PLAY RECAP *****  
20.106.147.196 : ok=7    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
20.106.147.244 : ok=7    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
  
[allen@ansible-cn ~]$
```



2. ssh to either of the servers and verify the operations performed.

7 Setup Ansible Tower on RHEL Server

7.1 Download the Ansible Automation Platform Installation Program

1. Starting with Ansible Tower 3.8, Automation Hub will act as a content provider for Ansible Tower, which requires both an Ansible Tower deployment and an Automation Hub deployment running alongside each other.
2. SSH to your RHEL server to execute installation commands.

```
$ ssh username@IPAddress  
$ sudo su  
# yum update -y
```

```
[allen@ansible-tower ~]$ sudo su  
[root@ansible-tower allen]# yum update -y  
Last metadata expiration check: 0:01:34 ago on Thu 15 Jul 2021 09:27:39 AM UTC.  
-
```

3. Download the latest version of the online installer directly from the internet.

```
# wget https://releases.ansible.com/ansible-tower/setup/ansible-tower-setup-latest.tar.gz
```

```
[root@ansible-tower allen]# wget https://releases.ansible.com/ansible-tower/setup/ansible-tower-setup-latest.tar.gz  
--2021-07-15 09:34:49-- https://releases.ansible.com/ansible-tower/setup/ansible-tower-setup-latest.tar.gz  
Resolving releases.ansible.com (releases.ansible.com)... 104.26.0.234, 104.26.1.234, 172.67.68.251, ...  
Connecting to releases.ansible.com (releases.ansible.com)|104.26.0.234|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 5070767 (4.8M) [application/x-gzip]  
Saving to: 'ansible-tower-setup-latest.tar.gz'  
  
ansible-tower-setup-latest.tar.gz 100%[=====]> 4.83M 9.87MB/s in 0.5s  
2021-07-15 09:34:50 (9.87 MB/s) - 'ansible-tower-setup-latest.tar.gz' saved [5070767/5070767]  
[root@ansible-tower allen]#
```

4. Extract files from zip package and change directory.

```
# tar xvzf ansible-tower-setup-latest.tar.gz  
# ls  
# cd ansible-tower-setup-3.8.3-1/
```

```
[root@ansible-tower allen]# tar xvzf ansible-tower-setup-latest.tar.gz
```

```
[root@ansible-tower allen]# ls  
ansible-tower-setup-3.8.3-1  ansible-tower-setup-latest.tar.gz  
[root@ansible-tower allen]# cd ansible-tower-setup-3.8.3-1/
```

5. Edit inventory file and add the password you want to set for your Ansible Tower.

```
# vi inventory
```

```
[tower]  
localhost ansible_connection=local  
  
[automationhub]  
  
[database]  
  
[all:vars]  
admin_password=''  
  
pg_host=''  
pg_port=''  
  
pg_database='awx'  
pg_username='awx'  
pg_password=''  
pg_sslmode='prefer' # set to 'verify-full' for client-side enforced SSL  
  
# Automation Hub Configuration  
#  
  
automationhub_admin_password=''  
  
automationhub_pg_host=''  
automationhub_pg_port=''  
  
automationhub_pg_database='automationhub'  
automationhub_pg_username='automationhub'  
automationhub_pg_password=''  
automationhub_pg_sslmode='prefer'  
  
# By default if the automation hub package and its dependencies  
# are installed they won't get upgraded when running the installer  
# even if newer packages are available. One needs to run the ./setup.sh  
# script with the following set to True.
```

- Update the password field and set the password string. We need to use this password string at the time of Login to Ansible tower.

```
[tower]
localhost ansible_connection=local

[automationhub]

[database]

[all:vars]
admin_password='password'

pg_host=''
pg_port=''

pg_database='awx'
pg_username='awx'
pg_password='password'
pg_sslmode= prefer # set to 'verify-full' for client-side enforced SSL

# Automation Hub Configuration
#
automationhub_admin_password=''

automationhub_pg_host=''
automationhub_pg_port=''

automationhub_pg_database='automationhub'
automationhub_pg_username='automationhub'
automationhub_pg_password='password'
automationhub_pg_sslmode= prefer

# By default if the automation hub package and its dependencies
# are installed they won't get upgraded when running the installer
# even if newer packages are available. One needs to run the ./setup.sh
# script with the following set to True.
#
# automationhub_upgrade = False
```

- Save the file and exit using ":wq". Execute setup.sh file to install Ansible tower.

```
[root@ansible-tower ansible-tower-setup-3.8.3-1]#
[root@ansible-tower ansible-tower-setup-3.8.3-1]# ./setup.sh
Ansible Tower Dependencies Repository -
Dependencies resolved.

=====
Package           Architecture      Version            Repository          Size
=====
Installing:
ansible           noarch          2.9.20-1.el8ae   ansible-tower-dependencies-temp    17 M
Installing dependencies:
sshpass           x86_64          1.06-3.el8ae    ansible-tower-dependencies-temp    27 k
Installing weak dependencies:
python3-jmespath  noarch          0.9.0-11.el8     rhel-8-for-x86_64-appstream-eus-rhui-rpms  45 k

Transaction Summary
=====
Install 3 Packages

Total download size: 17 M
Installed size: 96 M
Downloading Packages:
(1/3): python3-jmespath-0.9.0-11.el8.noarch.rpm          386 kB/s | 272 kB   00:00
(2/3): sshpass-1.06-3.el8ae.x86_64.rpm                  44 kB/s | 27 kB    00:00
(3/3): ansible-2.9.20-1.el8ae.noarch.rpm                507 kB/s | 972 kB   00:32 ETA
```

```

PLAY [Install Automation Hub node] *****
skipping: no hosts matched

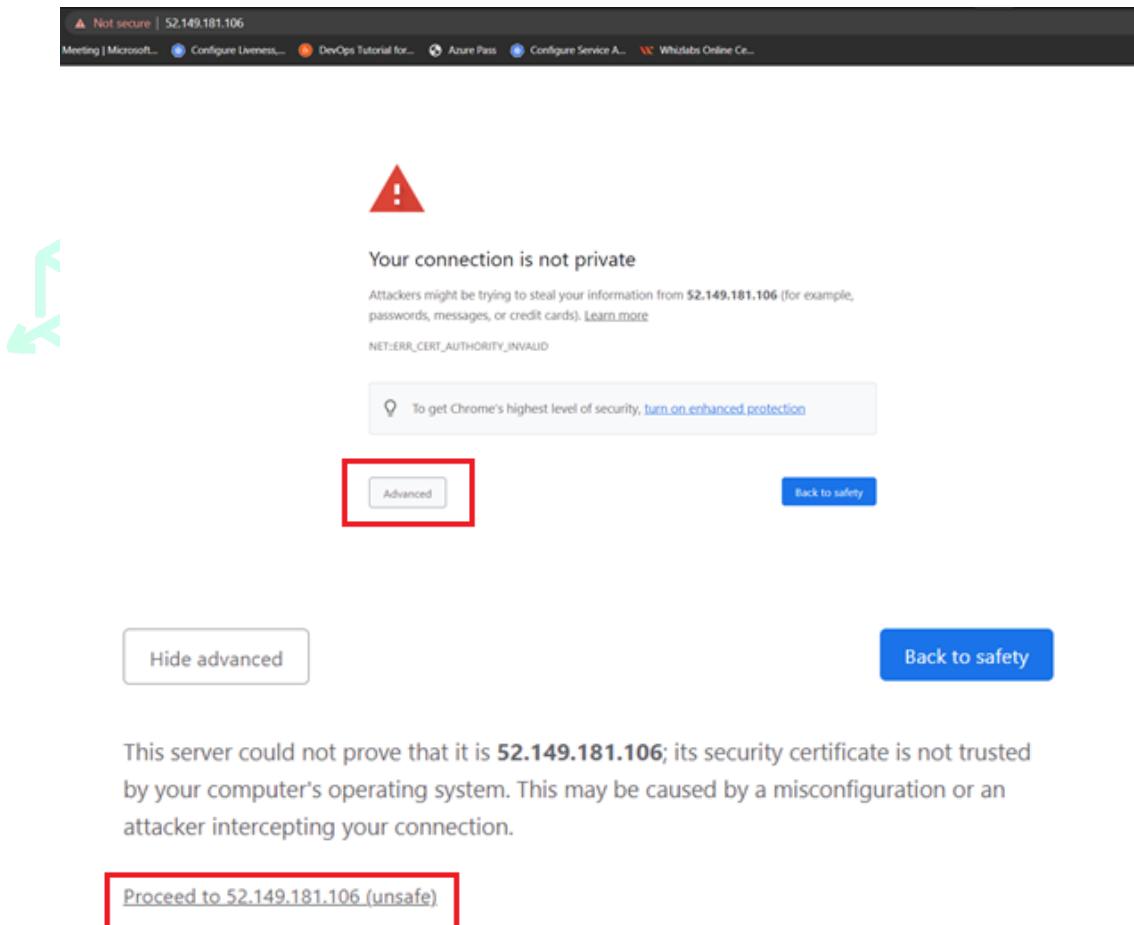
PLAY [Install Tower isolated node(s)] *****
skipping: no hosts matched

PLAY RECAP *****
: ok=175      unreachable=0    failed=0   skipped=86   rescued=0   ignored=2

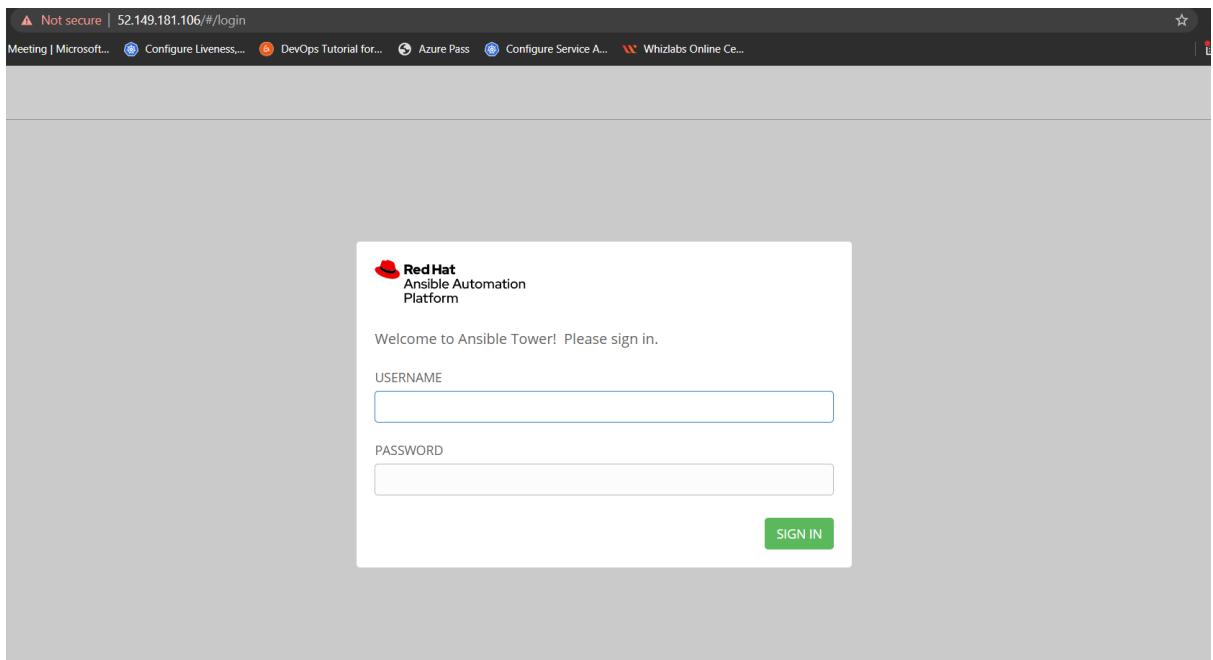
The setup process completed successfully.
Setup log saved to /var/log/tower/setup-2021-07-16-07:41:16.log.
[root@ansible-tower ansible-tower-setup-3.8.3-1]#

```

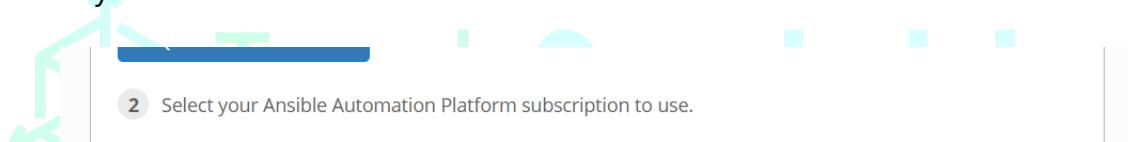
8. Try to access Ansible machine's public IP from Browser. It will show warning that your connection is not private, this is because ssl cert is not configured.
9. Click on advance and then Click on proceed to Public IP address.



10. Provide Username and Password which you have configured in ansible.conf file.



11. Note: Ansible Tower is enterprise edition tool, Red Hat gives free access for 60 days. You should have active Red Hat account to enable Ansible Tower.



2 Select your Ansible Automation Platform subscription to use.

Upload a Red Hat Subscription Manifest containing your subscription. To generate your subscription manifest, go to [subscription allocations](#) on the Red Hat Customer Portal.

* RED HAT SUBSCRIPTION MANIFEST ?

BROWSE No file selected.

Provide your Red Hat or Red Hat Satellite credentials below and you can choose from a list of your available subscriptions. The credentials you use will be stored for future use in retrieving renewal or expanded subscriptions.

USERNAME
ketan71

PASSWORD
.....

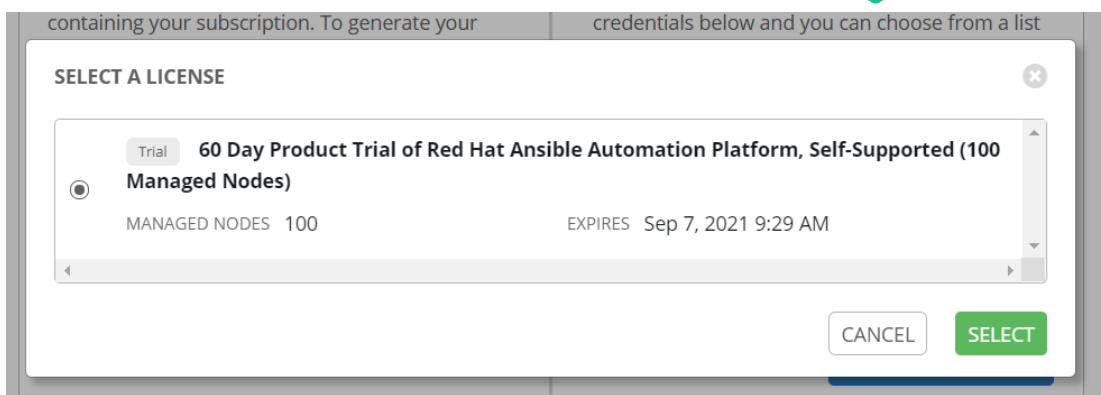
GET SUBSCRIPTIONS

SELECTED 60 Day Product Trial of Red Hat Ansible Automation Platform, Self-Supported (100 Managed Nodes)

3 Agree to the End User License Agreement, and click submit.

* END USER LICENSE AGREEMENT

12. Select 60 Days product trial and click on select.



13. Agree to terms and conditions and click on submit.



REPRESENTS THAT HE OR SHE HAS THE AUTHORITY TO ENTER INTO THIS END USER LICENSE AGREEMENT ON BEHALF OF THAT ENTITY. IF YOU DO NOT ACCEPT THE TERMS OF THIS AGREEMENT, THEN YOU MUST NOT USE THE RED HAT

I agree to the End User License Agreement

TRACKING AND ANALYTICS

By default, Tower collects and transmits analytics data on Tower usage to Red Hat. There are two categories of data collected by Tower. For more information, see [this Tower documentation page](#). Uncheck the following boxes to disable this feature.

User analytics: This data is used to enhance future releases of the Tower Software and help streamline customer experience and success.

Automation analytics: This data is used to enhance future releases of the Tower Software and to provide Automation Analytics to Tower subscribers.

SUBMIT

14. On successful completion of previous step, you will be redirected to Ansible Tower Dashboard. Click on Get Subscription button.

15. Select I agree to Licence Agreement and click on Submit button.

Agree to the End User License Agreement, and click submit.

* END USER LICENSE AGREEMENT

END USER LICENSE AGREEMENT
RED HAT STANDARD
PLEASE READ THIS END USER LICENSE AGREEMENT CAREFULLY BEFORE USING SOFTWARE FROM RED HAT. BY USING RED HAT SOFTWARE, YOU SIGNIFY YOUR ASSENT TO AND ACCEPTANCE OF THIS END USER LICENSE AGREEMENT AND ACKNOWLEDGE YOU HAVE READ AND UNDERSTAND THE TERMS. AN INDIVIDUAL ACTING ON BEHALF OF AN ENTITY REPRESENTS THAT HE OR SHE HAS THE AUTHORITY TO ENTER INTO THIS END USER LICENSE AGREEMENT ON BEHALF OF THAT ENTITY. IF YOU DO NOT ACCEPT THE TERMS OF THIS AGREEMENT, THEN YOU MUST NOT USE THE RED HAT SOFTWARE. THIS END USER LICENSE AGREEMENT DOES NOT

I agree to the End User License Agreement

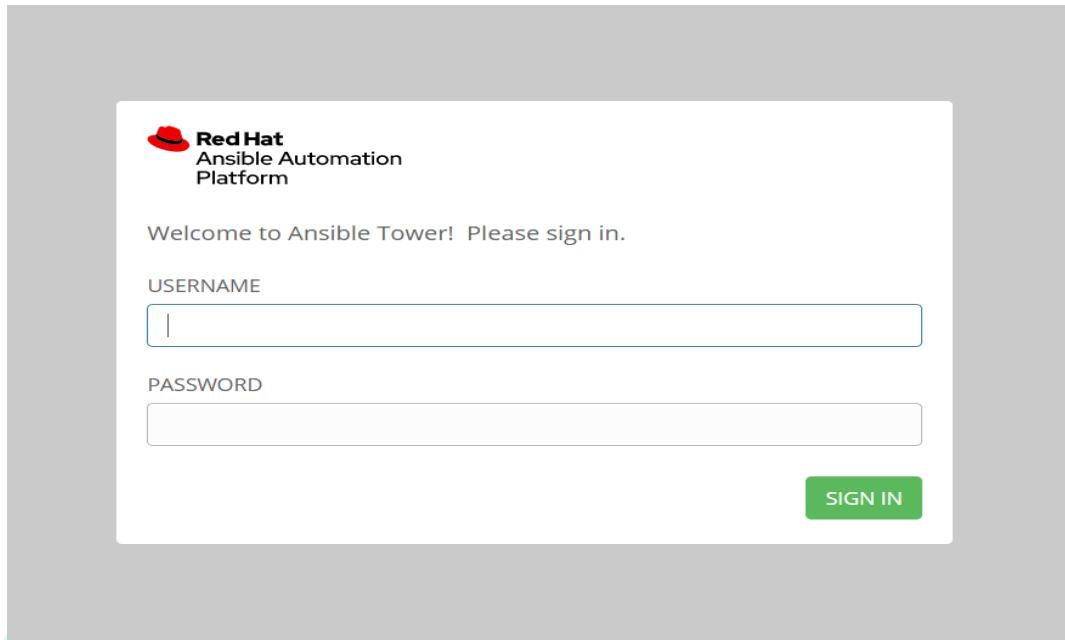
SUBMIT



8 Creating and Managing Ansible Tower Users

8.1 Login to ansible tower and Create Users

1. Log in to the Ansible Tower web interface running on the tower system using the admin account and password.

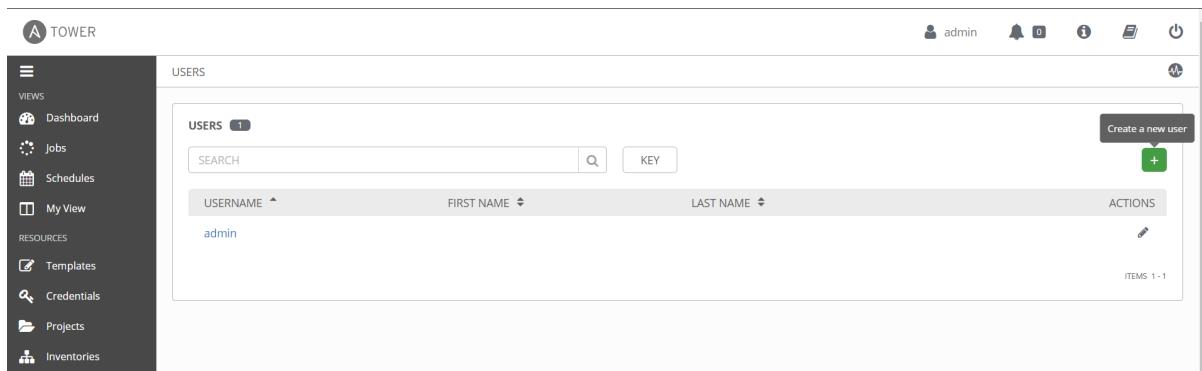


2. Create a user, sam, as a Normal User.
3. Go to Users in the left navigation bar.

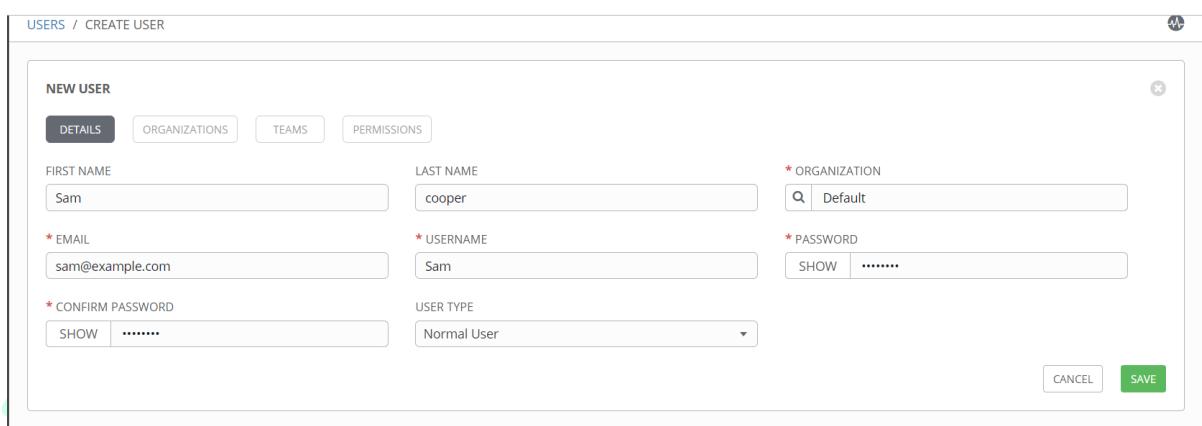


The image shows the Ansible Tower dashboard. On the left, there is a navigation sidebar with several options: Dashboard, Jobs, Schedules, My View, Templates, Credentials, Projects, Inventories, Inventory Scripts, Organizations, and Teams. The "Users" option is highlighted with a red box. The main dashboard area displays various metrics: 2 hosts, 0 failed hosts, 2 inventories, 0 inventory sync failures, 1 project, and 0 project sync failures. Below these metrics is a chart titled "JOB STATUS" showing the number of jobs over time from June 20 to July 20. At the bottom of the dashboard, there are two boxes: "RECENTLY USED JOB TEMPLATES" (empty) and "RECENTLY RUN JOBS" (empty). The URL at the bottom of the page is <https://52.249.176.125/#users>.

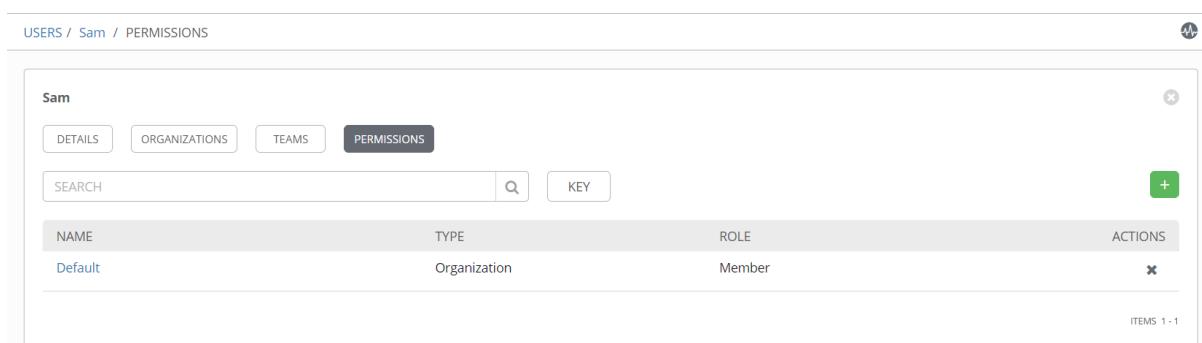
4. Click the + button to add a new user.



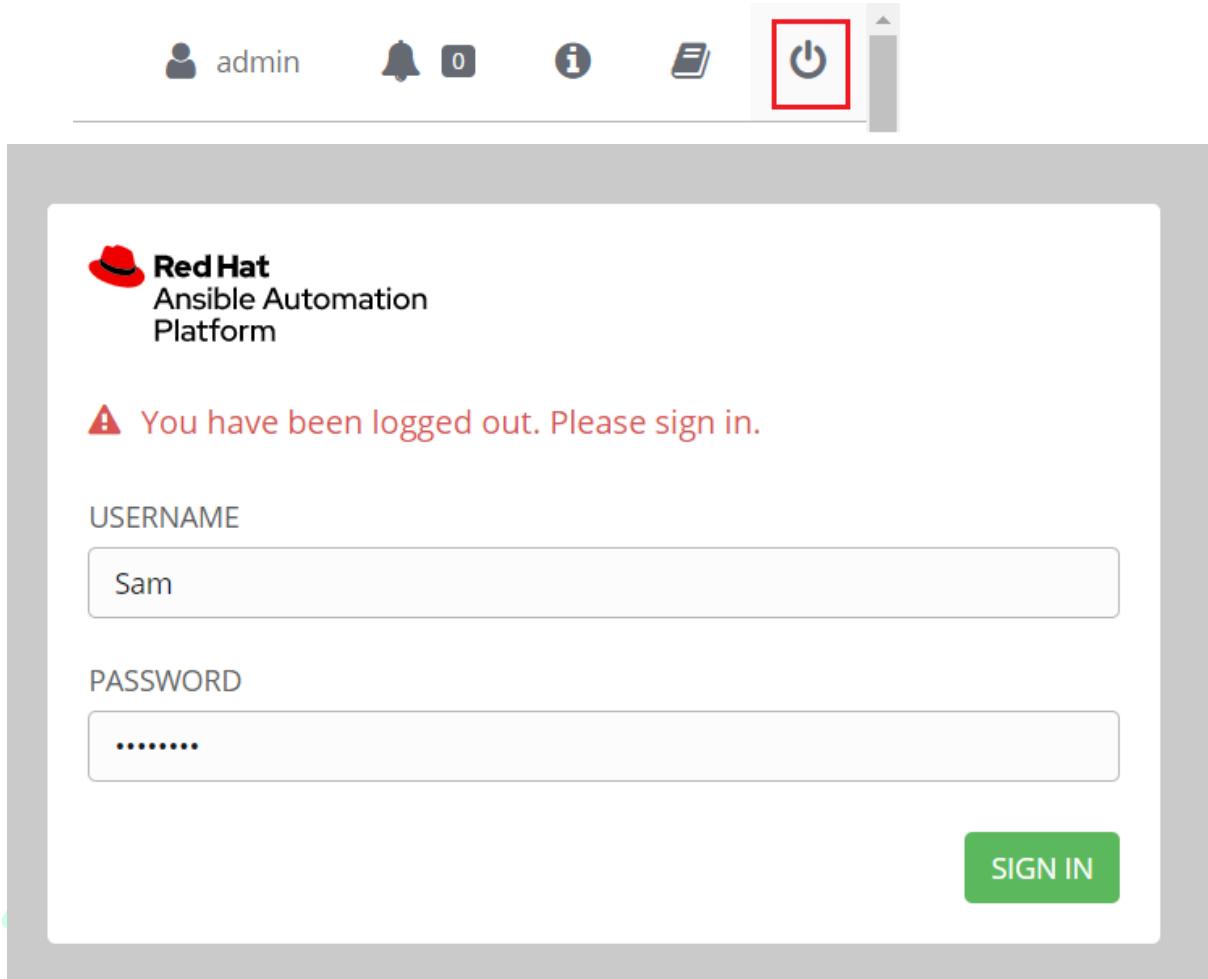
- On the next screen, fill in the details as follows:



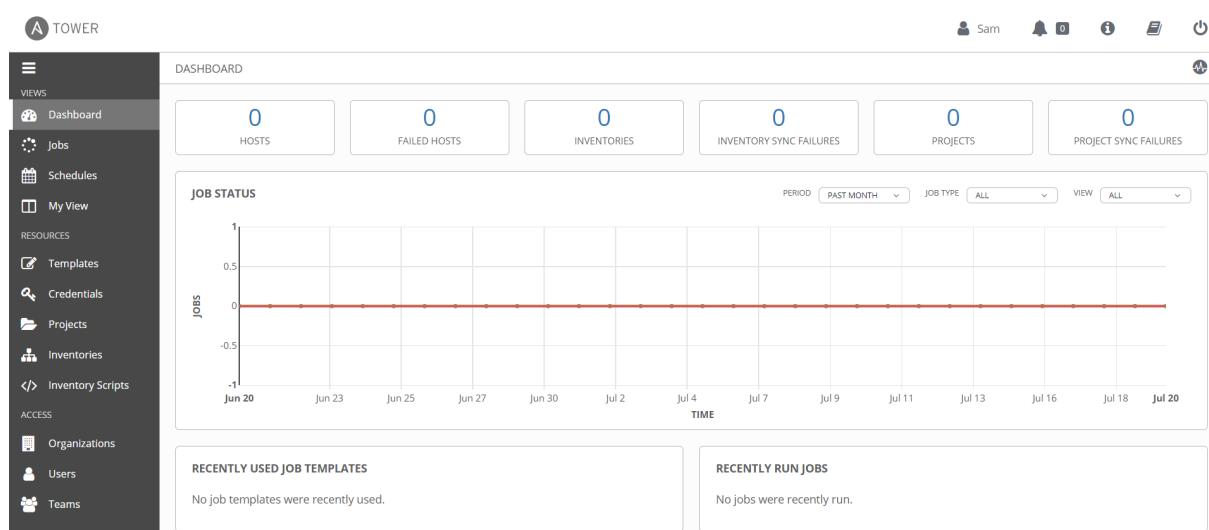
- click save to create new user.
- Verify the permissions for the newly created sam user.
- Click PERMISSIONS to see the user's permissions.



- As you can see, sam is simply a Member of the Default Organization.
- Click the Log Out icon in the top right corner to log out and then log back in as Sam with password of "password".



11. In the left navigation bar, you can see the user's access is limited.



The dashboard displays the following statistics:

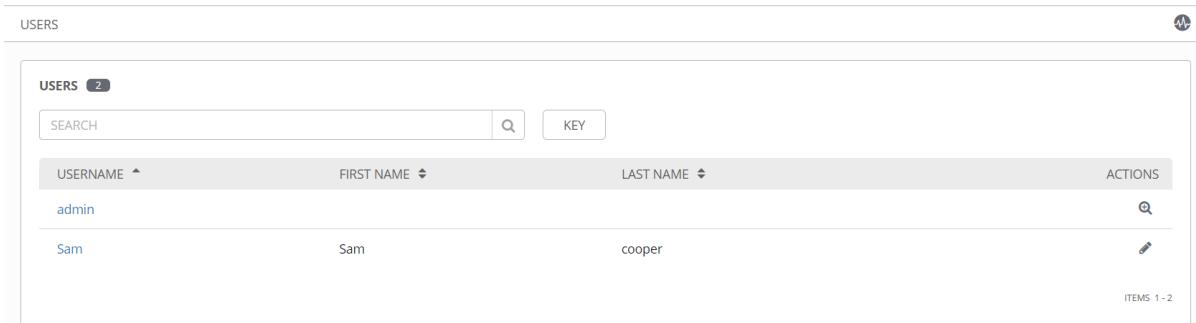
Category	Value
HOSTS	0
FAILED HOSTS	0
INVENTORIES	0
INVENTORY SYNC FAILURES	0
PROJECTS	0
PROJECT SYNC FAILURES	0

The 'JOB STATUS' chart shows zero jobs over time from June 20 to July 20.

Below the chart, two boxes show:

- RECENTLY USED JOB TEMPLATES:** No job templates were recently used.
- RECENTLY RUN JOBS:** No jobs were recently run.

12. Click Users in the left navigation bar to manage user permission. As you can see, it is not possible for Sam to add new users.



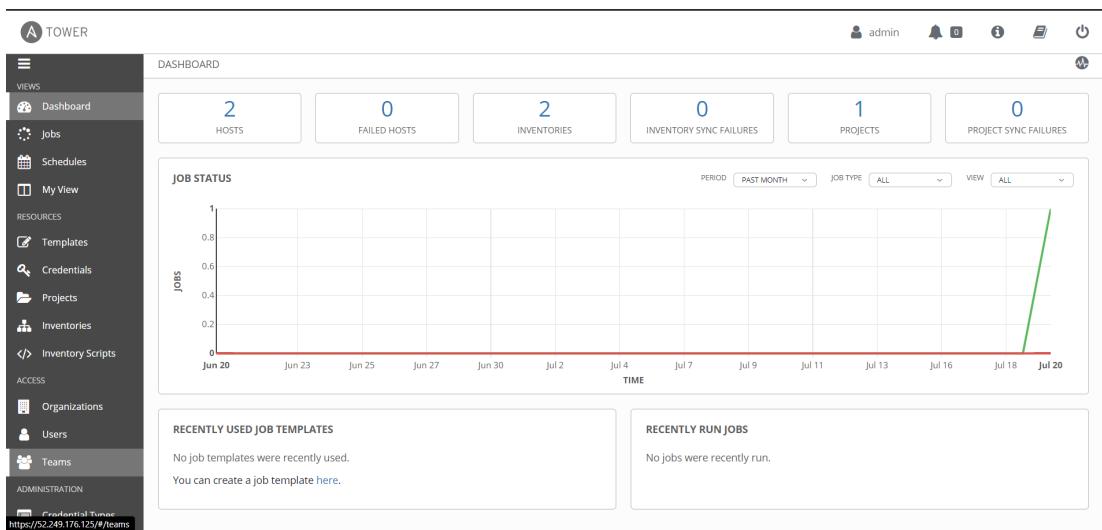
The screenshot shows the 'USERS' section of the Ansible Tower interface. At the top, there is a search bar and a 'KEY' button. Below the header, there are three columns: 'USERNAME', 'FIRST NAME', and 'LAST NAME'. Under 'USERNAME', there are two entries: 'admin' and 'Sam'. Under 'FIRST NAME', there are two entries: 'Sam' and 'cooper'. Under 'LAST NAME', there are two entries: 'cooper' and 'cooper'. On the far right, there is an 'ACTIONS' column with edit and delete icons. At the bottom right of the table, it says 'ITEMS 1 - 2'.

13. Click the Log Out icon to log out of the Tower web interface.
 14. Ansible tower provides few more User types such as System administrator and System Auditor. You can explore them by creating Users with appropriate User type and see that what level of permission they get.

1.1 Managing Users Efficiently With Teams

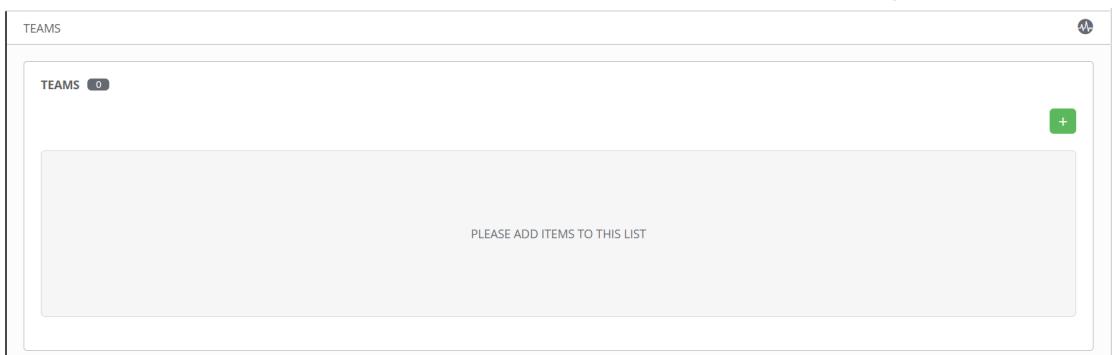
In this exercise, you will organize users into Teams and explore the access provided by different Team roles.

1. Create a new Team called Developers.
2. Click Teams in the left navigation bar to manage Teams.

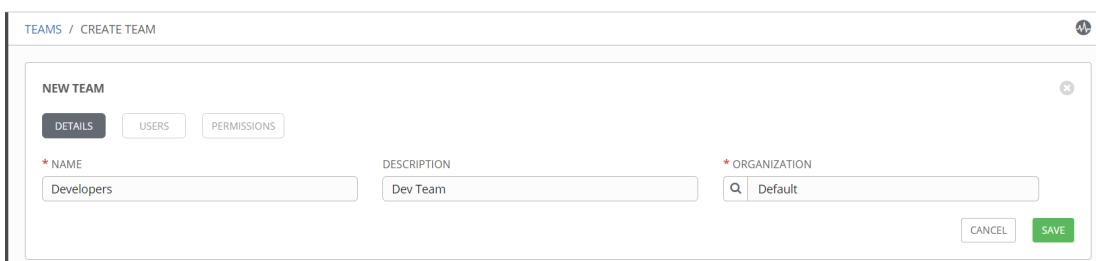


The screenshot shows the Ansible Tower dashboard. The left sidebar has several sections: 'VIEWS' (Dashboard, Jobs, Schedules, My View), 'RESOURCES' (Templates, Credentials, Projects, Inventories, Inventory Scripts), 'ACCESS' (Organizations, Users, Teams), and 'ADMINISTRATION' (Credential Types). The 'Teams' option under 'ACCESS' is highlighted with a red box. The main area shows a 'DASHBOARD' with statistics: 2 HOSTS, 0 FAILED HOSTS, 2 INVENTORIES, 0 INVENTORY SYNC FAILURES, 1 PROJECTS, and 0 PROJECT SYNC FAILURES. Below the dashboard is a 'JOB STATUS' chart showing the number of jobs over time from June 20 to July 20. The chart shows a sharp increase in job count starting around July 18. There are also two boxes: 'RECENTLY USED JOB TEMPLATES' (No job templates were recently used) and 'RECENTLY RUN JOBS' (No jobs were recently run).

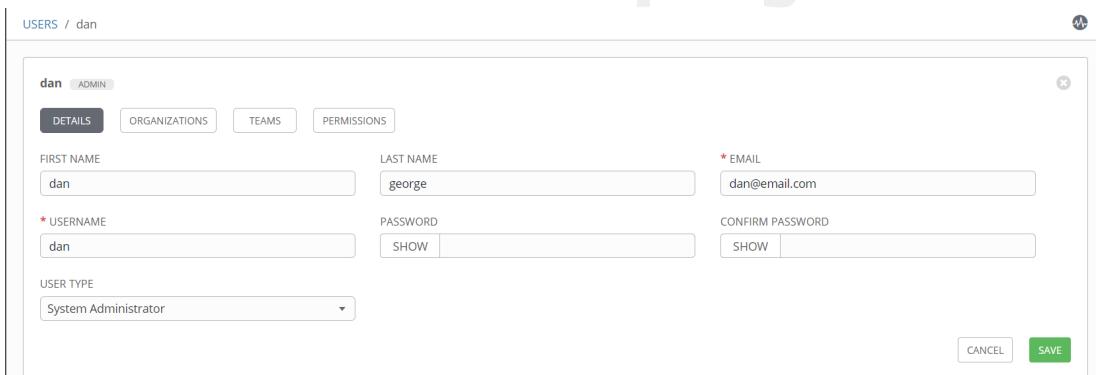
3. Click the + button to add a new Team.



4. On the next screen, fill in the details as follows:

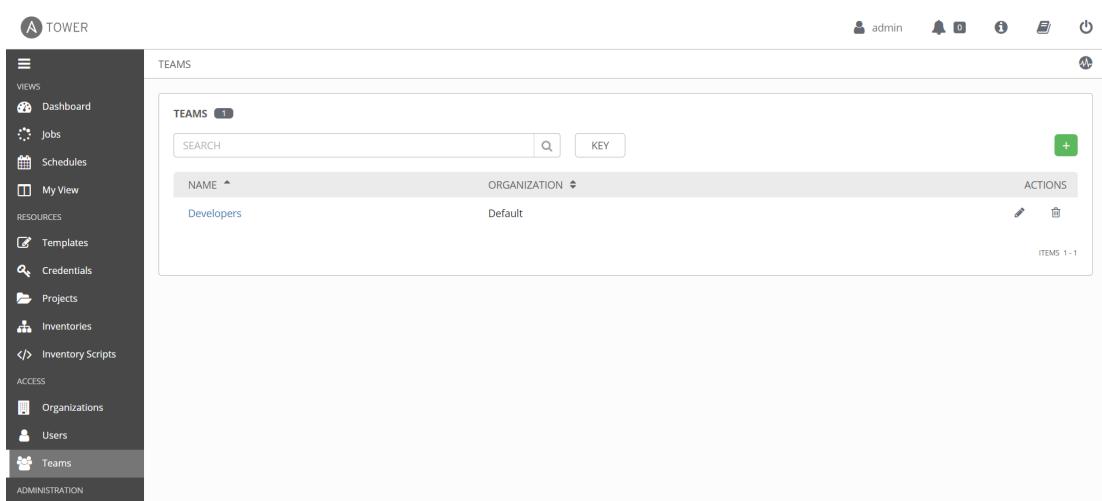


5. Create a user who will be the administrator for the newly created Developers Team.
 6. Click Users in the left navigation bar to manage users.
 7. Click + to add a new user.
 8. On the next screen, fill in the details as follows:



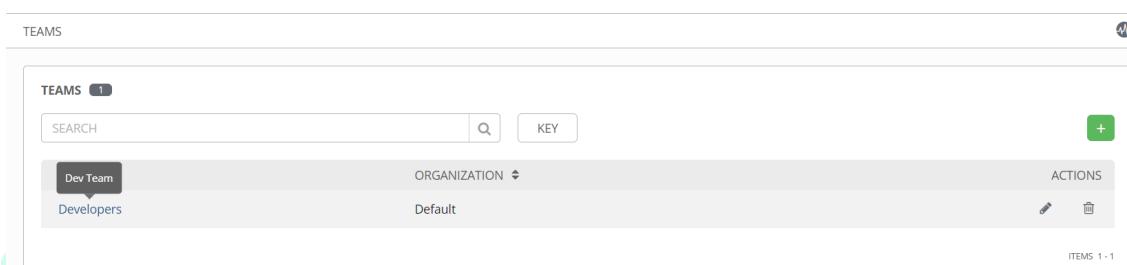
9. Click SAVE to create the new user.
 10. Assign the Admin role on the Developers Team to the dan user.
 11. Click Teams in the left navigation bar to manage Teams.

A TOWER



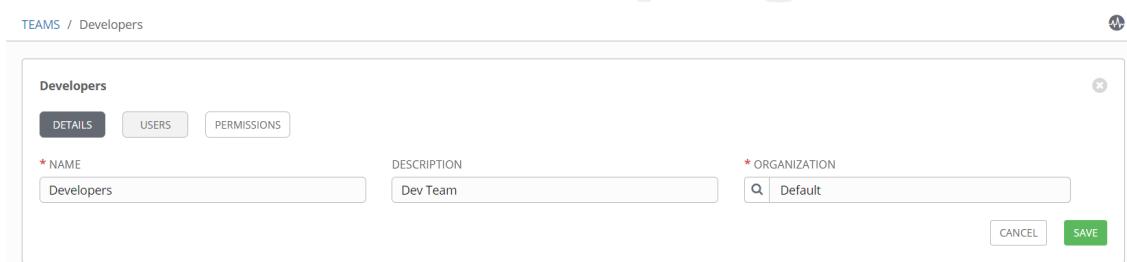
NAME	ORGANIZATION	ACTIONS
Developers	Default	

12. Click the link for the Developers Team created previously.



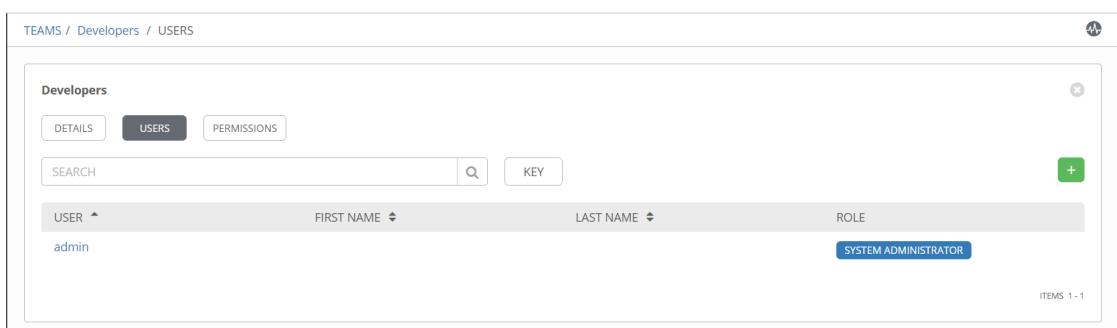
NAME	ORGANIZATION	ACTIONS
Dev Team	Default	

13. Click the USERS button to manage the users assigned to the Team.



USER	FIRST NAME	LAST NAME	ROLE
admin			SYSTEM ADMINISTRATOR

14. Click + to add a new user to the Team.



15. In the first section of the screen, check the box next to dan to select this user.
This adds Dan George to the list of users in Developer team.

DEVELOPERS | ADD USERS X

Please select Users from the list below.

SEARCH KEY

USERNAME ▲	FIRST NAME ▲	LAST NAME ▲
<input type="checkbox"/> admin		
<input checked="" type="checkbox"/> dan	dan	george
<input type="checkbox"/> Sam	Sam	cooper

ITEMS 1 - 3

CANCEL SAVE

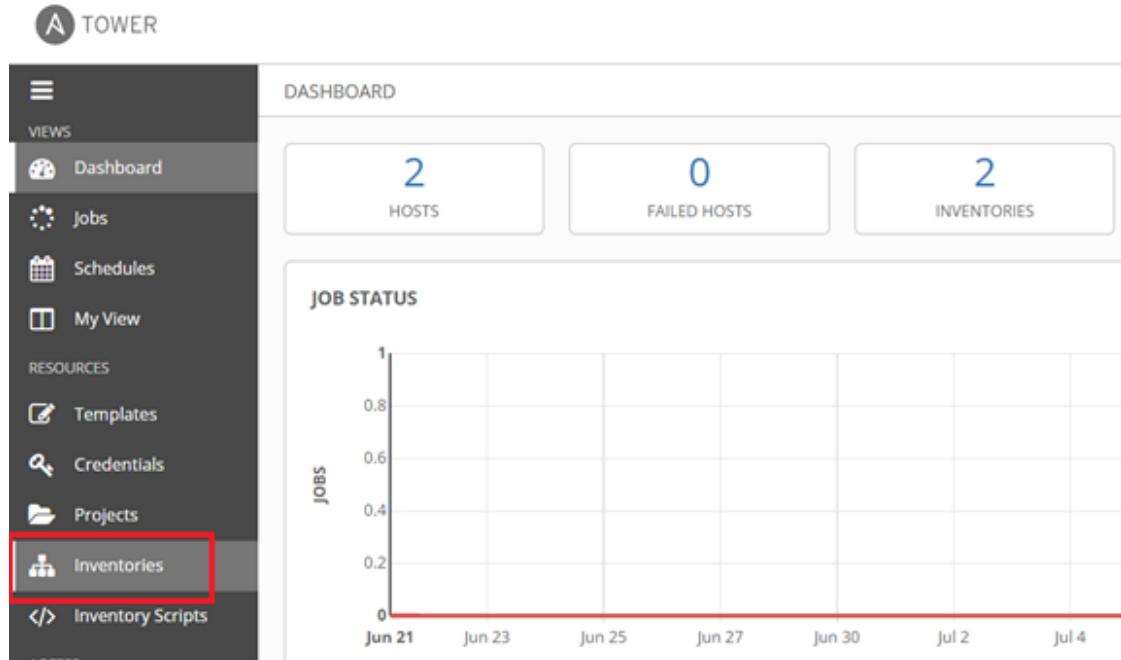
16. Click SAVE.



9 Creating and Managing Inventories and Credentials

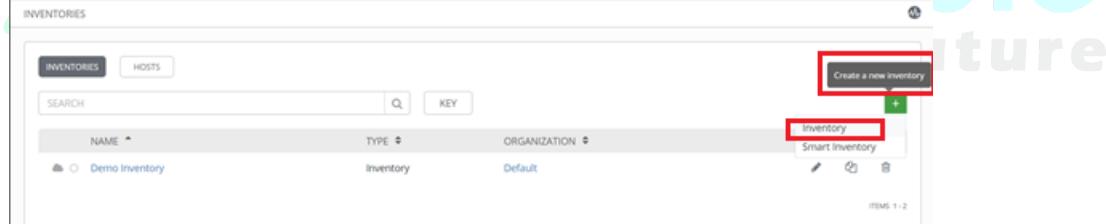
9.1 Create a new inventory called Dev within default organisation

1. Click Inventories in the left quick navigation bar.



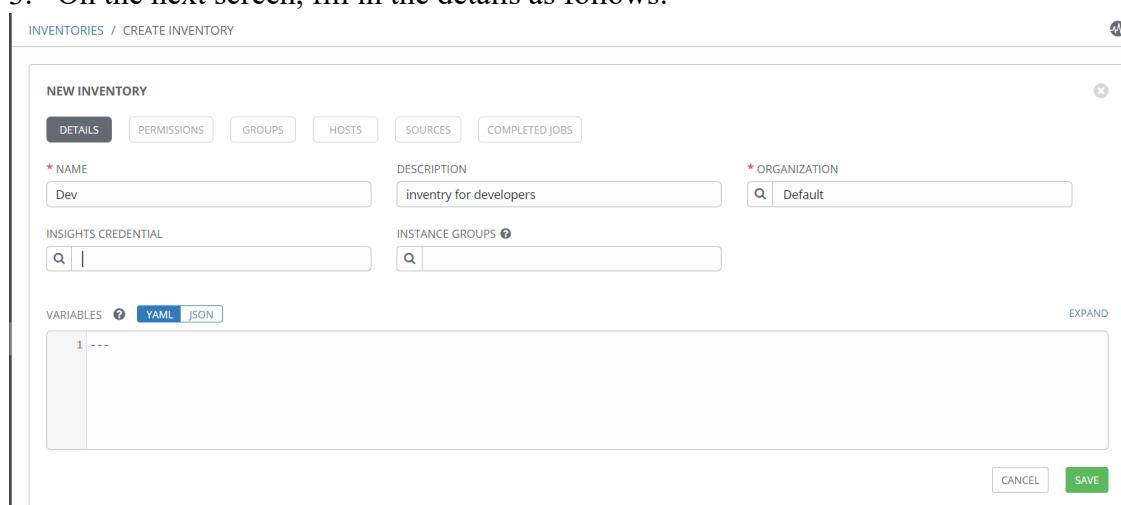
The screenshot shows the A TOWER interface. On the left, there is a navigation sidebar with various options like Dashboard, Jobs, Schedules, My View, Templates, Credentials, Projects, and Inventories. The 'Inventories' option is highlighted with a red box. The main area is a 'DASHBOARD' showing statistics: 2 HOSTS, 0 FAILED HOSTS, and 2 INVENTORIES. Below this is a 'JOB STATUS' chart with a single data point at 1.0 on June 21.

2. Click the + button. From the drop-down list, select Inventory to add a new Inventory.



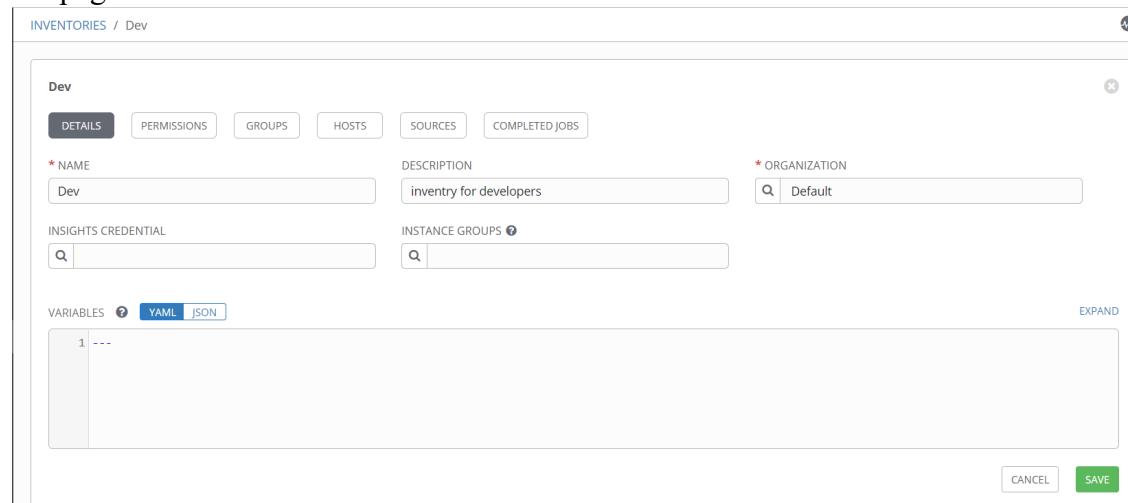
The screenshot shows the 'INVENTORIES' screen. It lists an existing 'Demo Inventory' and has a search bar and filters for NAME, TYPE, and ORGANIZATION. On the right, there is a modal window titled 'Create a new inventory'. Inside the modal, there is a green '+' button with a red box around it, and a dropdown menu showing 'Inventory' and 'Smart Inventory'.

3. On the next screen, fill in the details as follows:



The screenshot shows the 'CREATE INVENTORY' form. The 'NAME' field is set to 'Dev'. The 'DESCRIPTION' field contains 'inventory for developers'. The 'ORGANIZATION' field is set to 'Default'. Under the 'VARIABLES' section, the 'YAML' tab is selected, showing a single variable entry: '1'. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

4. Click **SAVE** to create the new Inventory. You are redirected to the Inventory details page.



INVENTORIES / Dev

Dev

DETAILS **PERMISSIONS** **GROUPS** **HOSTS** **SOURCES** **COMPLETED JOBS**

* NAME: Dev
DESCRIPTION: Inventory for developers
* ORGANIZATION: Default

INSIGHTS CREDENTIAL:
INSTANCE GROUPS:

VARIABLES: **YAML** **JSON**

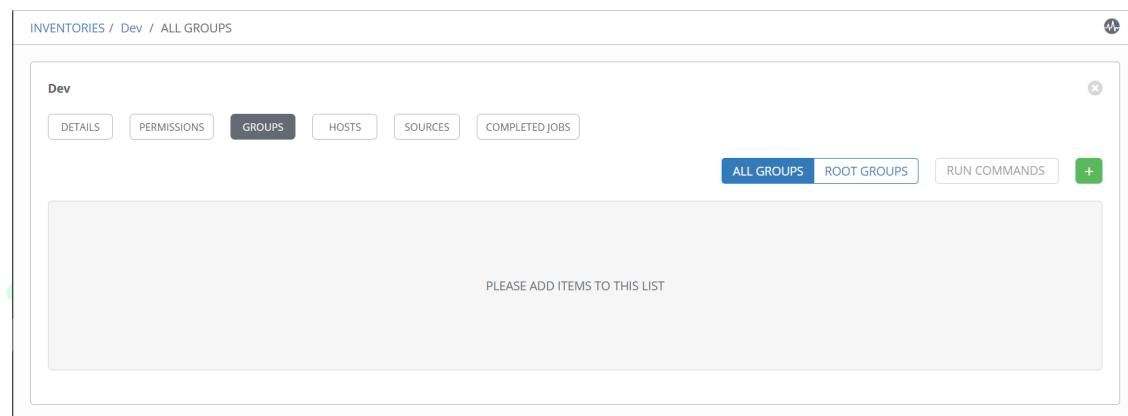
1

EXPAND

CANCEL **SAVE**

Create a group called dev-servers in the Dev Inventory.

1. Click the **GROUPS** button.



INVENTORIES / Dev / ALL GROUPS

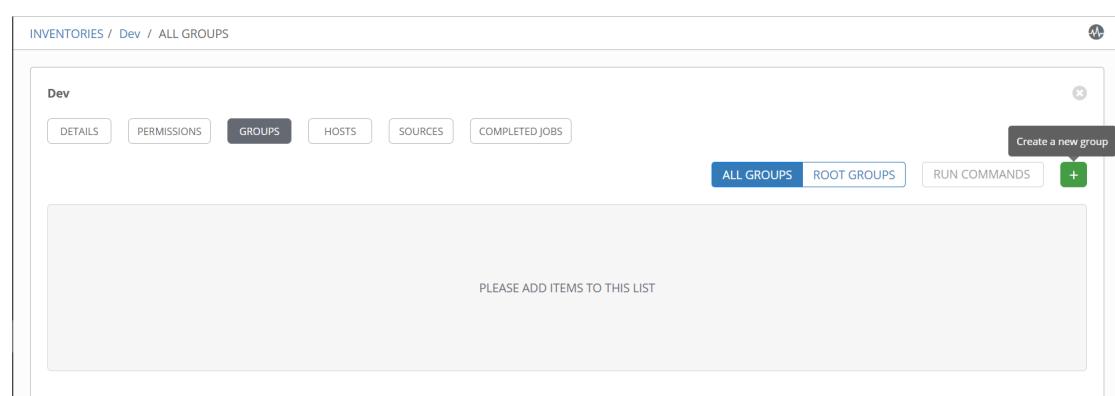
Dev

DETAILS **PERMISSIONS** **GROUPS** **HOSTS** **SOURCES** **COMPLETED JOBS**

ALL GROUPS **ROOT GROUPS** **RUN COMMANDS** **+**

PLEASE ADD ITEMS TO THIS LIST

2. Click the **+** button to add the new group.



INVENTORIES / Dev / ALL GROUPS

Dev

DETAILS **PERMISSIONS** **GROUPS** **HOSTS** **SOURCES** **COMPLETED JOBS**

ALL GROUPS **ROOT GROUPS** **RUN COMMANDS** **+**

Create a new group

PLEASE ADD ITEMS TO THIS LIST

3. On the next screen, fill in the details as follows:

INVENTORIES / Dev / ALL GROUPS / CREATE GROUP

CREATE GROUP

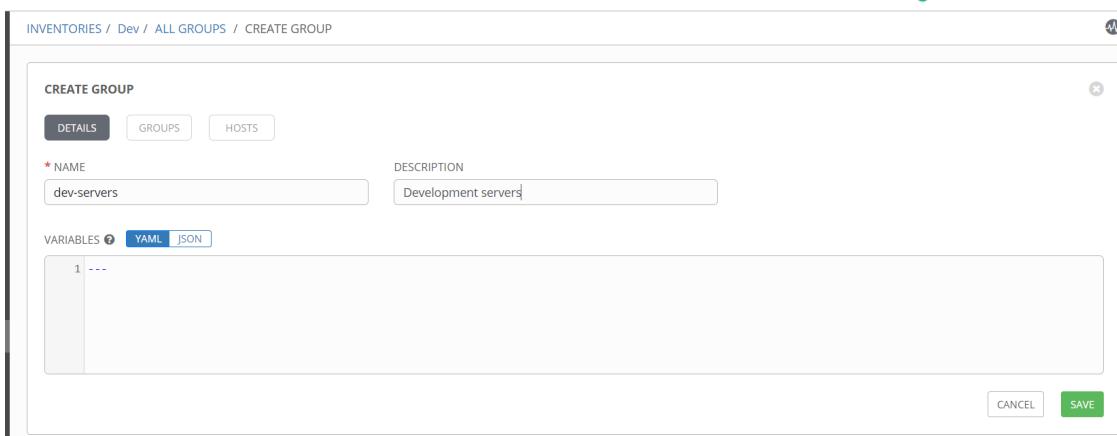
DETAILS GROUPS HOSTS

* NAME: dev-servers DESCRIPTION: Development servers

VARIABLES: YAML JSON

1

CANCEL SAVE



4. Click **SAVE** to create the new group.
 5. Add hosts with their public ip and description to dev-servers group.
- Click the **HOSTS** button, within the group you just created.

INVENTORIES / Dev / ALL GROUPS / dev-servers / ASSOCIATED HOSTS

dev-servers

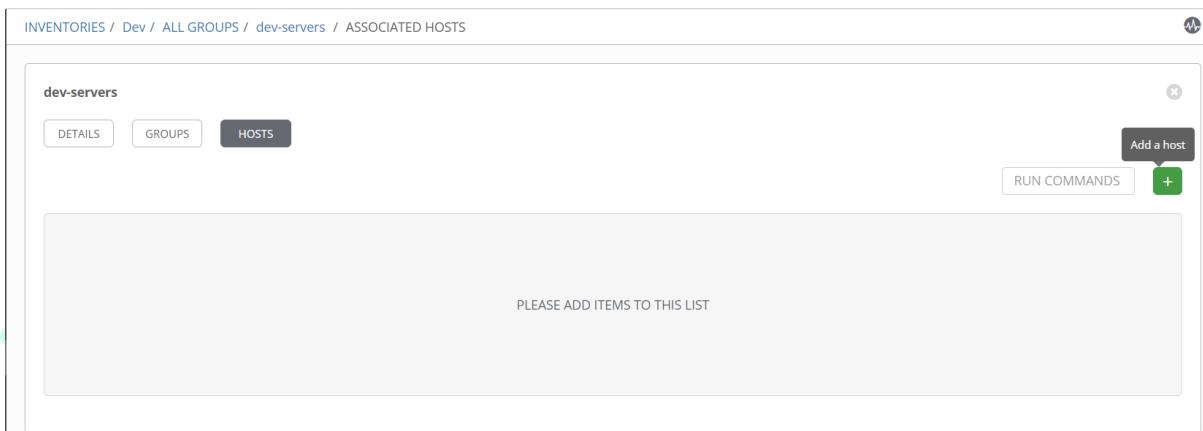
DETAILS GROUPS **HOSTS**

Add a host

RUN COMMANDS

+

PLEASE ADD ITEMS TO THIS LIST



Click the + button. From the drop-down menu, select New Host to add a new host to the group.

On the next screen, fill in the details as follows:

INVENTORIES / Dev / ALL GROUPS / dev-servers / ASSOCIATED HOSTS / CREATE HOST

CREATE HOST

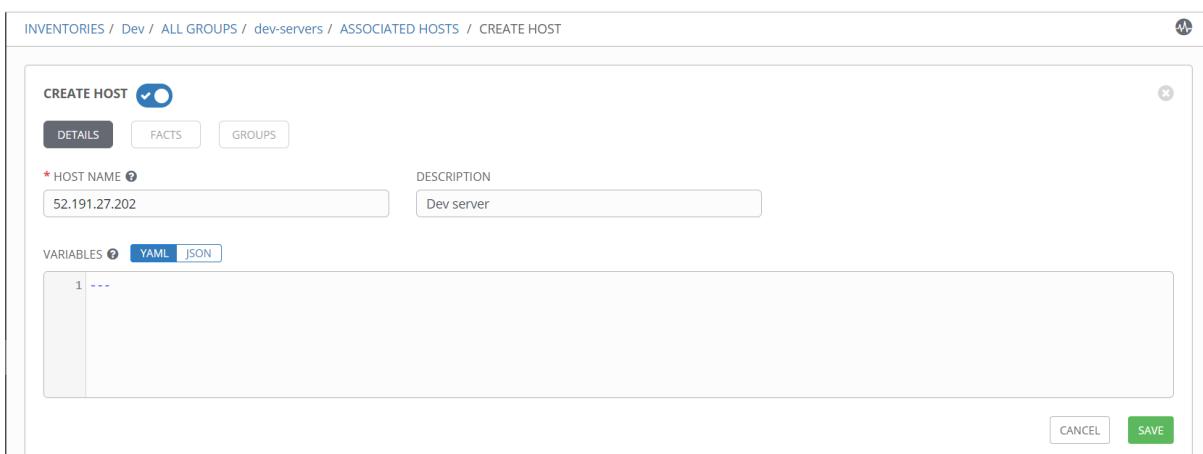
DETAILS FACTS GROUPS

* HOST NAME: 52.191.27.202 DESCRIPTION: Dev server

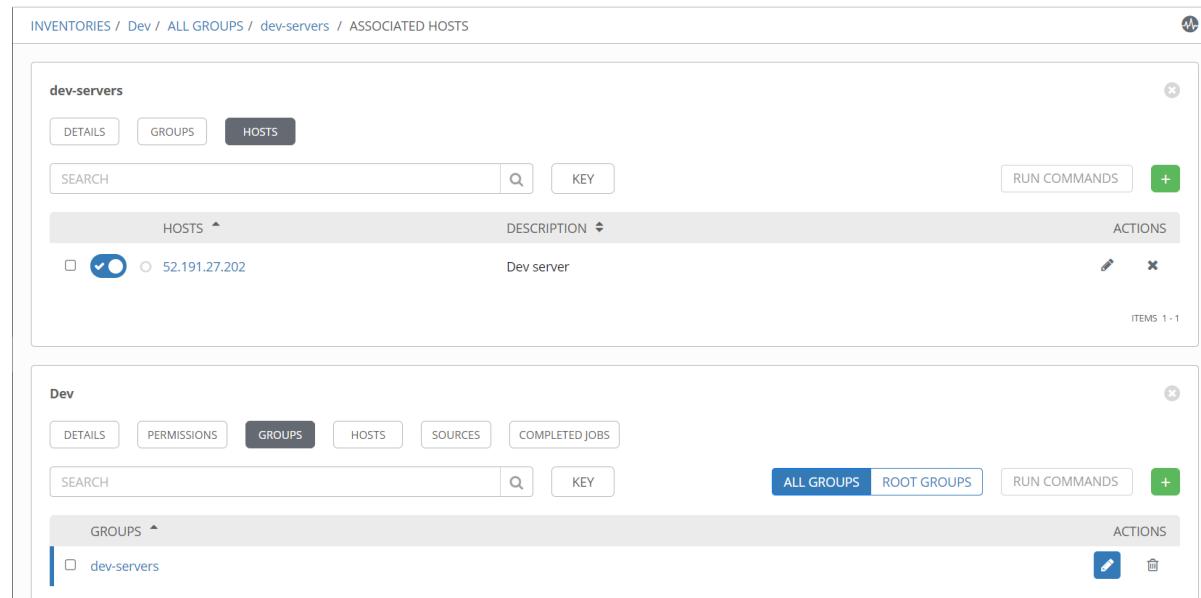
VARIABLES: YAML JSON

1

CANCEL SAVE



Click **SAVE** to create the new host.

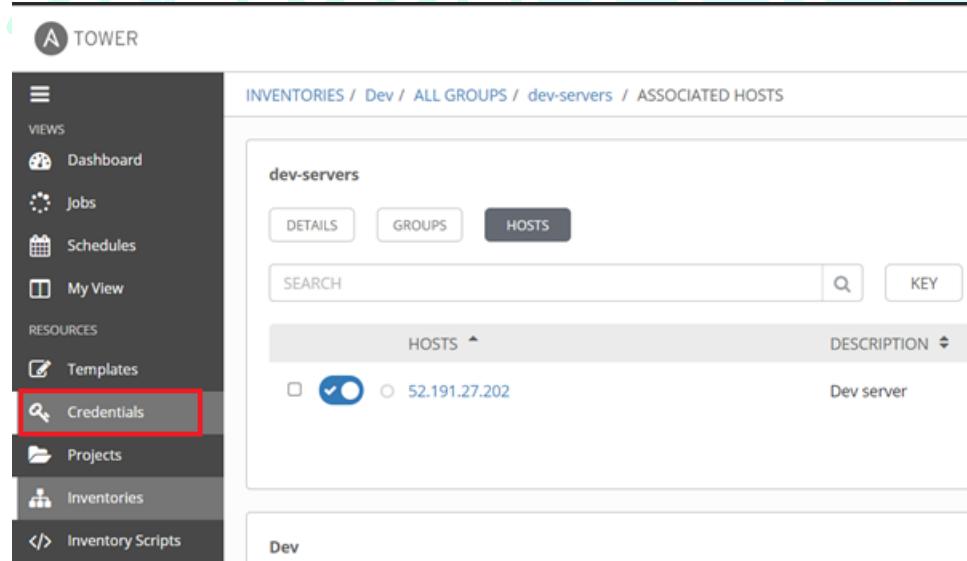


The screenshot shows the Ansible Tower interface for managing inventories. At the top, it says 'INVENTORIES / Dev / ALL GROUPS / dev-servers / ASSOCIATED HOSTS'. Below this, there are two main sections:

- dev-servers**: A table showing one host: 52.191.27.202 (labeled 'Dev server'). It has columns for DETAILS, GROUPS, and HOSTS, with the HOSTS tab selected. Buttons include SEARCH, KEY, RUN COMMANDS, and ACTIONS (edit and delete).
- Dev**: A table showing one group: dev-servers. It has tabs for DETAILS, PERMISSIONS, GROUPS, HOSTS, SOURCES, and COMPLETED JOBS, with the GROUPS tab selected. Buttons include SEARCH, KEY, ALL GROUPS, ROOT GROUPS, RUN COMMANDS, and ACTIONS (edit and delete).

1.1 Create a new Credentials for users with following information

1. Click Credentials in the left quick navigation bar.



The screenshot shows the Ansible Tower interface with a sidebar on the left containing various navigation options. The 'Credentials' option is highlighted with a red box. The main content area shows the 'INVENTORIES / Dev / ALL GROUPS / dev-servers / ASSOCIATED HOSTS' view, which is identical to the one in the previous screenshot.

2. Click the + button to add a new Credential.

CREDENTIALS

CREDENTIALS 2		
SEARCH	KEY	
NAME	KIND	OWNERS
Ansible Galaxy	Ansible Galaxy/Automation Hub API Token	
Demo Credential	Machine	admin

ITEMS 1 - 2

3. Create a new Credential, Developers, with the following information:
Provide your VM Username and Password which you use to SSH the machine.

CREDENTIALS / CREATE CREDENTIAL

NEW CREDENTIAL	
DETAILS	PERMISSIONS
* NAME	DESCRIPTION
developers	developers creds
* CREDENTIAL TYPE	ORGANIZATION
Machine	Default
TYPE DETAILS	
USERNAME	PASSWORD <input type="checkbox"/> Prompt on launch
allen
SSH PRIVATE KEY HINT: Drag and drop private file on the field below.	

4. Leave the other fields untouched and click **SAVE** to create the new Credential.
5. Click on the permission tab and see that system administrators has the access to the credentials.

CREDENTIALS / EDIT CREDENTIAL / PERMISSIONS

CREDENTIALS PERMISSIONS		
DETAILS	PERMISSIONS	
SEARCH	KEY	
USER	ROLE	TEAM ROLES
admin	SYSTEM ADMINISTRATOR	
dan	SYSTEM ADMINISTRATOR	

ITEMS 1 - 2

10 Creating A Project for Ansible Playbooks

After completing this section, students should be able to create and manage a project in Ansible Tower that gets playbooks and other project materials from an existing Git repository.

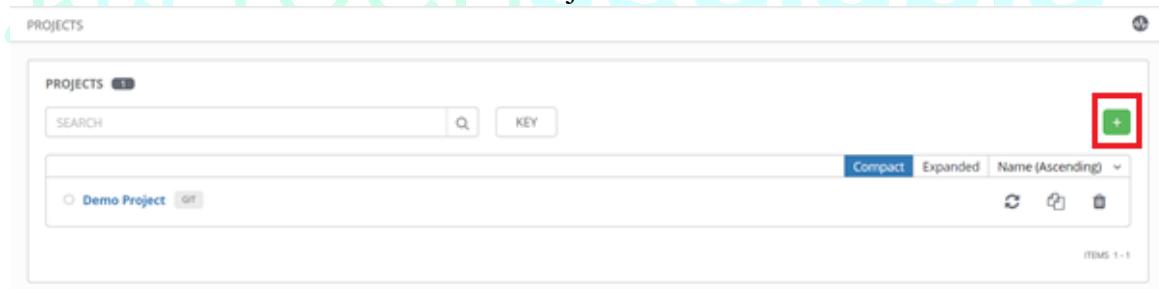
1.2 Creating a project

The following is the procedure for creating a Project to share a collection of Ansible playbooks and roles managed in an existing Git repository.

1. Click **Projects** in the left quick navigation bar to go to the Projects management screen.



2. Click the + button to create the new Project.



3. Parallelly ssh to Ansible tower host and create a file under specific path shown as above.

/var/lib/awx/projects/ is the specific path under this we can create a directory for multiple projects and store the playbook. Playbook under this specific directory will be accessible and executable from Ansible Tower.

“/var/lib/awx/projects/copyfile” create a file with below content using this command.

```
$ sudo su
# mkdir /var/lib/awx/projects/copyfile
# cd /var/lib/awx/projects/copyfile
# vi copy.yml
```

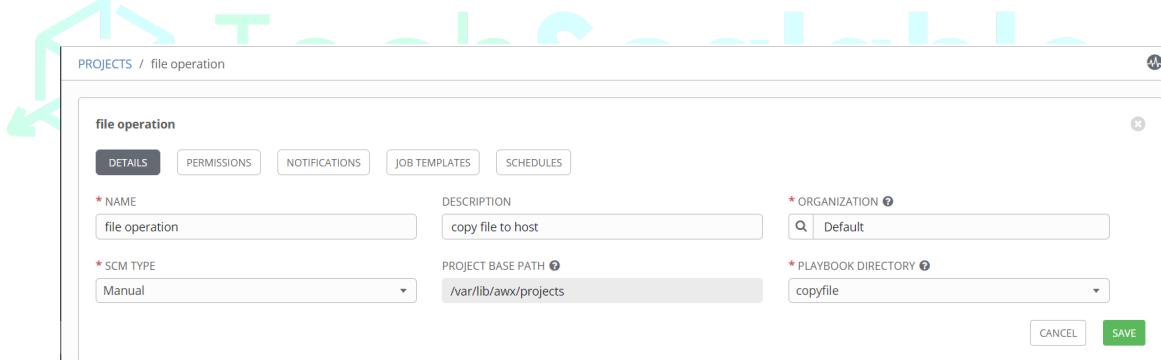
```
---
- hosts: all
  tasks:
    - name: content update
      copy:
        dest: /tmp/hello.text
        content: "hello world, copy this to all hosts"
~
```

Once you create this file and save it you will get option under playbook directory to choose copyfile directory.

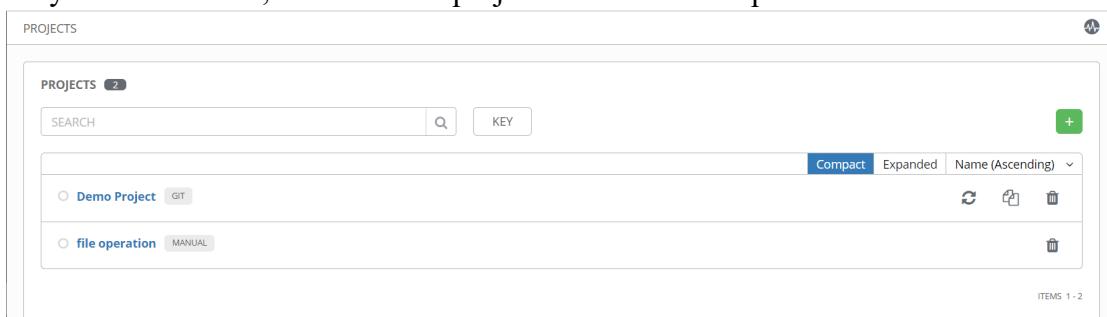
Note: one directory can be used for only one project.

4. Enter a unique name for the Project in the NAME field. Optionally, enter a description for the Project in the DESCRIPTION field.

5. Click the magnifying glass icon next to the ORGANIZATION field to display a list of Organizations within Ansible Tower. Select Default organization from the list and click SELECT.

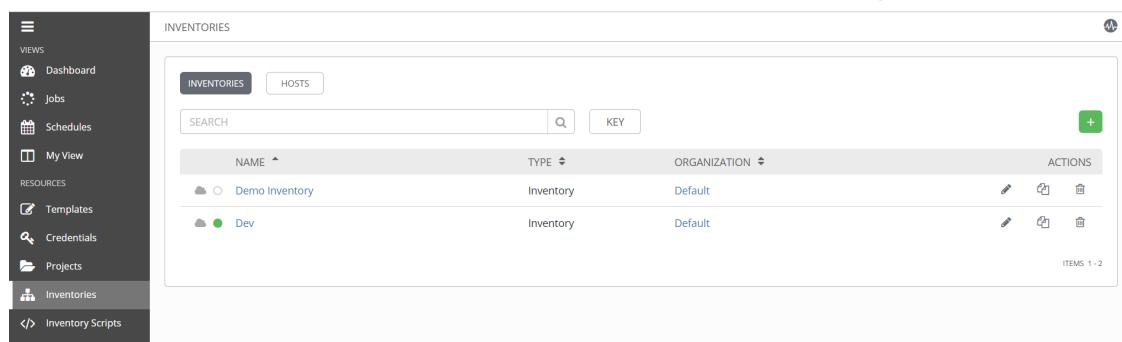


As you can see now, we have new project created as file operation.



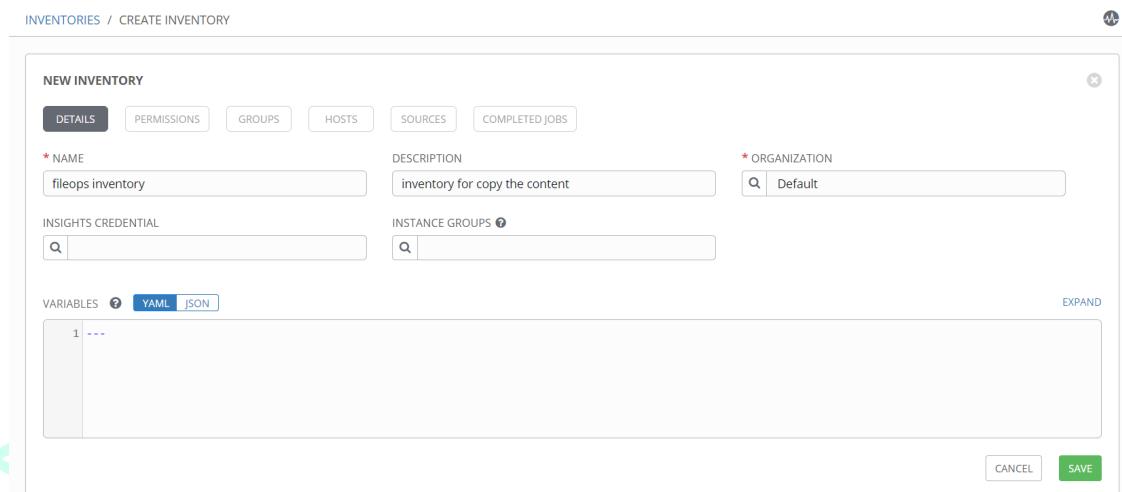
1.3 Create an inventory for this project

1. On the left panel click on **inventory**.



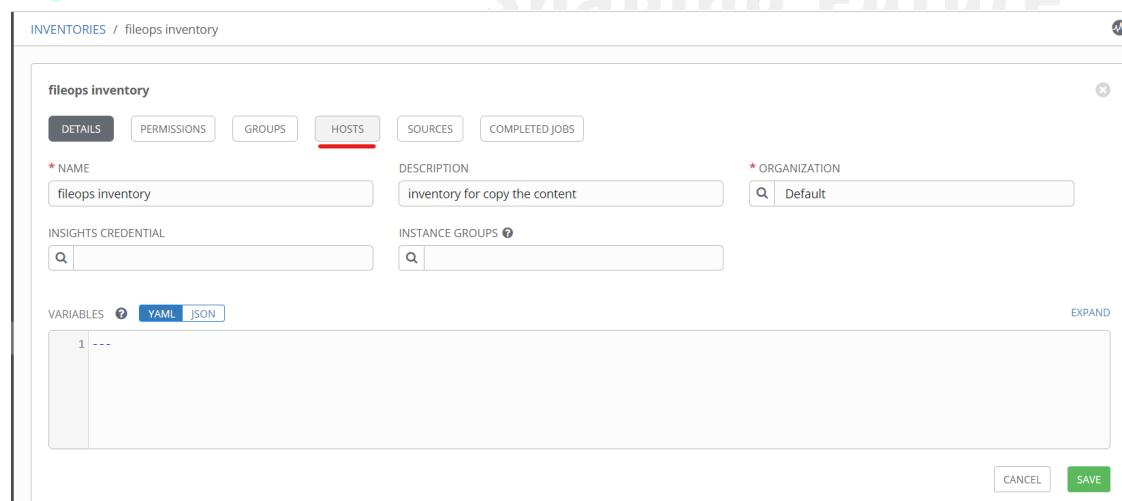
The screenshot shows the 'INVENTORIES' section of a management interface. On the left is a sidebar with options like Dashboard, Jobs, Schedules, My View, Templates, Credentials, Projects, Inventories (which is selected), and Inventory Scripts. The main area displays a table with columns for NAME, TYPE, ORGANIZATION, and ACTIONS. Two entries are listed: 'Demo Inventory' (Inventory, Default) and 'Dev' (Inventory, Default). A search bar and a 'KEY' button are also present.

- Add new inventory with below details and save it.



The screenshot shows the 'NEW INVENTORY' dialog. The 'DETAILS' tab is selected. The 'NAME' field contains 'fileops inventory'. The 'DESCRIPTION' field contains 'inventory for copy the content'. The 'ORGANIZATION' dropdown is set to 'Default'. Under 'INSIGHTS CREDENTIAL' and 'INSTANCE GROUPS', there are search fields. The 'VARIABLES' section shows a single entry '1'. At the bottom are 'CANCEL' and 'SAVE' buttons.

- Click on **host** and we will add the host to run the project.



The screenshot shows the 'fileops inventory' dialog with the 'HOSTS' tab selected. The 'DETAILS' tab is also visible. The 'NAME' field contains 'fileops inventory'. The 'DESCRIPTION' field contains 'inventory for copy the content'. The 'ORGANIZATION' dropdown is set to 'Default'. Under 'INSIGHTS CREDENTIAL' and 'INSTANCE GROUPS', there are search fields. The 'VARIABLES' section shows a single entry '1'. At the bottom are 'CANCEL' and 'SAVE' buttons.

- add new host with below details. Provide public IP of the VM as Host name and add description. Click on save to configure host.

INVENTORIES / fileops inventory / HOSTS / CREATE HOST

CREATE HOST

DETAILS **FACTS** **GROUPS** **COMPLETED JOBS**

* HOST NAME DESCRIPTION

VARIABLES **YAML** **JSON**

```
1 ---
```

CANCEL **SAVE**

We can add more host by following above steps.

1.4 Create credentials to access the host

1. Go to credentials tab and click on plus to add credentials for ubuntu host.

A TOWER

CREDENTIALS

NAME	KIND	OWNERS	ACTIONS
Ansible Galaxy	Ansible Galaxy/Automation Hub API Token		
Demo Credential	Machine	admin	
developers	Machine	Default	

ITEMS 1 - 3

2. Provide appropriate name for creds and use ssh username and password of your VM. Leave other fields blank and save it.

CREDENTIALS / CREATE CREDENTIAL

NEW CREDENTIAL

DETAILS **PERMISSIONS**

* NAME DESCRIPTION ORGANIZATION

* CREDENTIAL TYPE

TYPE DETAILS

USERNAME PASSWORD

SSH PRIVATE KEY HINT: Drag and drop private file on the field below.

3. Once you save the credentials you can see that it automatically encrypts the password. So it will not be exposed to other users.

CREDENTIALS / EDIT CREDENTIAL

ubuntu creds

DETAILS **PERMISSIONS**

* NAME DESCRIPTION ORGANIZATION

* CREDENTIAL TYPE

TYPE DETAILS

USERNAME PASSWORD Prompt on launch

SSH PRIVATE KEY HINT: Drag and drop private file on the field below.

5. As a next step you need to create a template which will run the playbook and use these credentials to access the host.
6. From the left panel click on template and add a template to run the playbook.

TEMPLATES

TEMPLATES **1**

SEARCH KEY

Demo Job Template	Job Template			
-------------------	--------------	--	--	--

ITEMS: 1 - 1

7. Click on plus button and select a job template.

TEMPLATES

TEMPLATES **1**

SEARCH KEY

Demo Job Template	Job Template			
-------------------	--------------	--	--	--

ITEMS: 1 - 1

8. Select the resources which we have created in this lab to execute this template.

Job name and description we can provide on our convenience. Job type should be run as you want to run the playbook.

TEMPLATES / CREATE JOB TEMPLATE

NEW JOB TEMPLATE

DETAILS **PERMISSIONS** **COMPLETED JOBS** **SCHEDULES** **ADD SURVEY**

* NAME <input type="text" value="sample template"/>	DESCRIPTION <input type="text" value="copy the content"/>	* JOB TYPE Run	<input type="checkbox"/> PROMPT ON LAUNCH
* INVENTORY <input type="text" value="fileops inventory"/>	* PROJECT <input type="text" value="file operation"/>	* PLAYBOOK <input type="text" value="copy.yml"/>	<input type="checkbox"/> PROMPT ON LAUNCH
CREDENTIALS <input type="text" value="ubuntu creds"/>	FORKS <input type="text" value="0"/>	LIMIT <input type="text" value="1"/>	<input type="checkbox"/> PROMPT ON LAUNCH
* VERTBOSITY <input type="text" value="0 (Normal)"/>	JOB TAGS <input type="text" value=""/>	SKIP TAGS <input type="text" value=""/>	<input type="checkbox"/> PROMPT ON LAUNCH
LABELS <input type="text" value=""/>	INSTANCE GROUPS <input type="text" value=""/>	JOB SLICING <input type="text" value="1"/>	<input type="checkbox"/> PROMPT ON LAUNCH
TIMEOUT <input type="text" value="0"/>	SHOW CHANGES <input type="checkbox"/>	OPTIONS	
		<input type="checkbox"/> ENABLE PRIVILEGE ESCALATION	
		<input type="checkbox"/> ENABLE PROVISIONING CALLBACKS	
		<input type="checkbox"/> ENABLE WEBHOOK	
		<input type="checkbox"/> ENABLE CONCURRENT JOBS	
		<input type="checkbox"/> ENABLE FACT CACHE	

Verify all the fields and save the template.

TIMEOUT

SHOW CHANGES

PROMPT ON LAUNCH

OPTIONS

ENABLE PRIVILEGE ESCALATION

ENABLE PROVISIONING CALLBACKS

ENABLE WEBHOOK

ENABLE CONCURRENT JOBS

ENABLE FACT CACHE

EXTRA VARIABLES **YAML** **JSON**

1 ---

PROMPT ON LAUNCH

LAUNCH **CANCEL** **SAVE**

9. Once you save the template you can see it in template list with some options.

TEMPLATES 2	
SEARCH	KEY
Demo Job Template	Job Template
sample template	Job Template

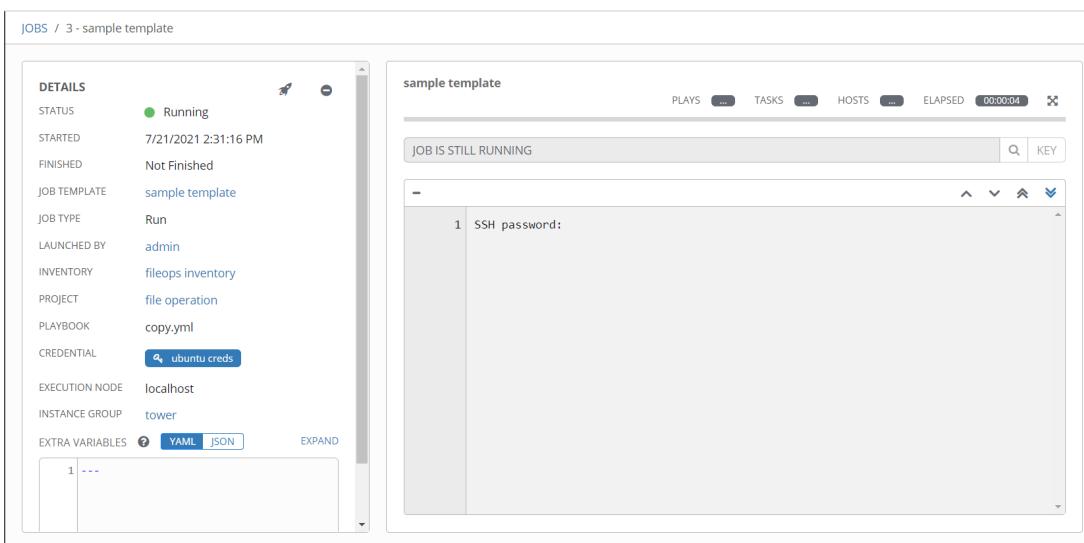
ITEMS 1 - 2

10. Click on rocket symbol to execute a template.

TEMPLATES 2	
SEARCH	KEY
Demo Job Template	Job Template
sample template	Job Template

ITEMS 1 - 2

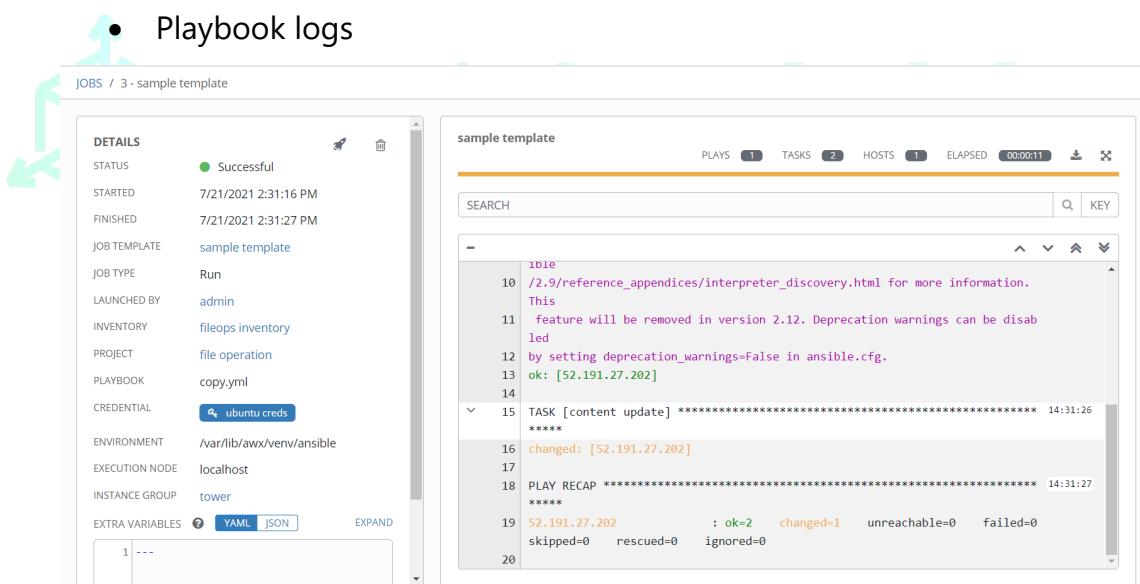
11. On the click it will redirect to the job template and show the running process.



The screenshot shows the Ansible Tower interface. On the left, a sidebar displays job details for a "sample template": Status (Running), Started at 7/21/2021 2:31:16 PM, Finished (Not Finished), Job Type (Run), Launched by (admin), Inventory (fileops inventory), Project (file operation), Playbook (copy.yml), Credential (ubuntu creds), Execution Node (localhost), and Instance Group (tower). Below this is an "EXTRA VARIABLES" section with tabs for YAML and JSON. On the right, a main panel titled "sample template" shows a progress bar with "PLAYS 0", "TASKS 0", "HOSTS 0", and "ELAPSED 00:00:04". A message "JOB IS STILL RUNNING" is displayed above a log pane. The log pane shows the command "SSH password:" followed by a blank line.

12. On the successful completion it will show the details such as

- Job Status
- Time taken to execute the job.
- Playbook logs



The screenshot shows the Ansible Tower interface after a job has completed. A green double-headed arrow points from the previous screenshot to this one. The sidebar now shows job details for a "sample template": Status (Successful), Started at 7/21/2021 2:31:16 PM, Finished at 7/21/2021 2:31:27 PM, and all other fields remain the same. The log pane on the right shows the completed playbook run, starting with "101e /2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg." It then lists tasks, changes, and a recap summary at the end.

13. Let's verify that Ansible tower has copied the file to ubuntu host or not.

ssh to your ubuntu host using Putty or terminal.

```
$ ssh allen@52.191.27.202
```

```

PS C:\Users\ketan> ssh allen@52.191.27.202
allen@52.191.27.202's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1051-azure x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Wed Jul 21 13:08:47 UTC 2021

 System load: 0.0 Processes: 113
 Usage of /: 6.1% of 28.90GB Users logged in: 0
 Memory usage: 28% IP address for eth0: 10.0.0.11
 Swap usage: 0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

 https://ubuntu.com/blog/microk8s-memory-optimisation

7 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Wed Jul 21 09:01:27 2021 from 52.191.27.216
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

allen@ansible-worker000:~$ 
  
```

```

$ cd /tmp
$ ll hello.txt
$ cat hello.txt
  
```

```

allen@ansible-worker000:~$ cd /tmp
allen@ansible-worker000:/tmp$ ll
total 48
drwxrwxrwt 9 root root 4096 Jul 21 13:14 /
drwxr-xr-x 23 root root 4096 Jul 21 06:50 ../
drwxrwxrwt 2 root root 4096 Jul 21 04:55 ICE-unix/
drwxrwxrwt 2 root root 4096 Jul 21 04:55 Test-unix/
drwxrwxrwt 2 root root 4096 Jul 21 04:55 X11-unix/
drwxrwxrwt 2 root root 4096 Jul 21 04:55 XIM-unix/
drwxrwxrwt 2 root root 4096 Jul 21 04:55 font-unix/
-rw-rw-r-- 1 allen allen 35 Jul 21 09:01 hello.txt
drwxr----- 3 root root 4096 Jul 21 06:49 systemd-private-a466bc286cec4c758fab581dc2ce85bb-systemd-resolved.service-DWzKJq/
drwxr----- 3 root root 4096 Jul 21 06:49 systemd-private-a466bc286cec4c758fab581dc2ce85bb-systemd-timesyncd.service-GqEajj/
-rw----- 1 root root 252 Jul 21 13:14 tmp.JhETLssIVM
-rw----- 1 root root 0 Jul 21 13:14 tmp.nGTwYOsNm7
-rw----- 1 root root 6 Jul 21 13:14 tmp.tWOKEvBgpF
allen@ansible-worker000:/tmp$ ll hello.txt
-rw-rw-r-- 1 allen allen 35 Jul 21 09:01 hello.txt
allen@ansible-worker000:/tmp$ 
  
```

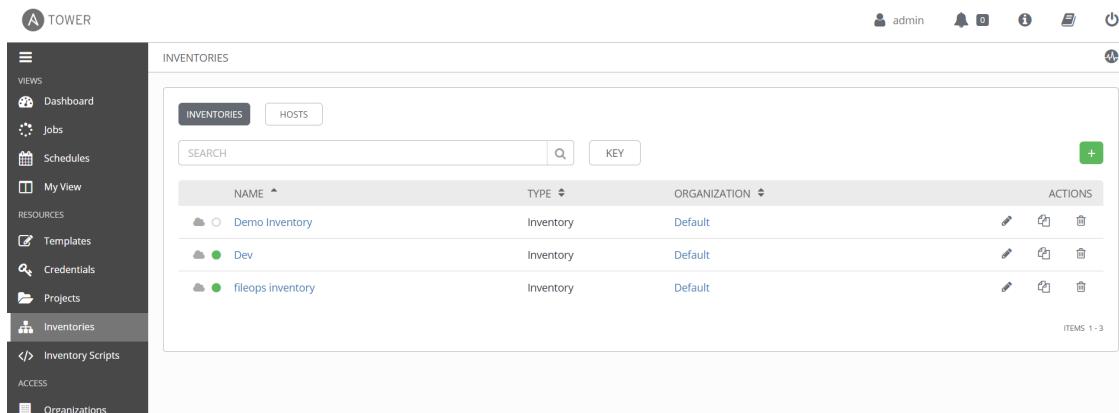
```

allen@ansible-worker000:/tmp$ cat hello.txt
hello world, copy this to all hosts allen@ansible-worker000:/tmp$ 
  
```

Keep this session running, we are going to run few more playbook in the same way.

11 Run Ad hoc command from Ansible Tower

11.1 Go to inventory to run ad hoc commands.

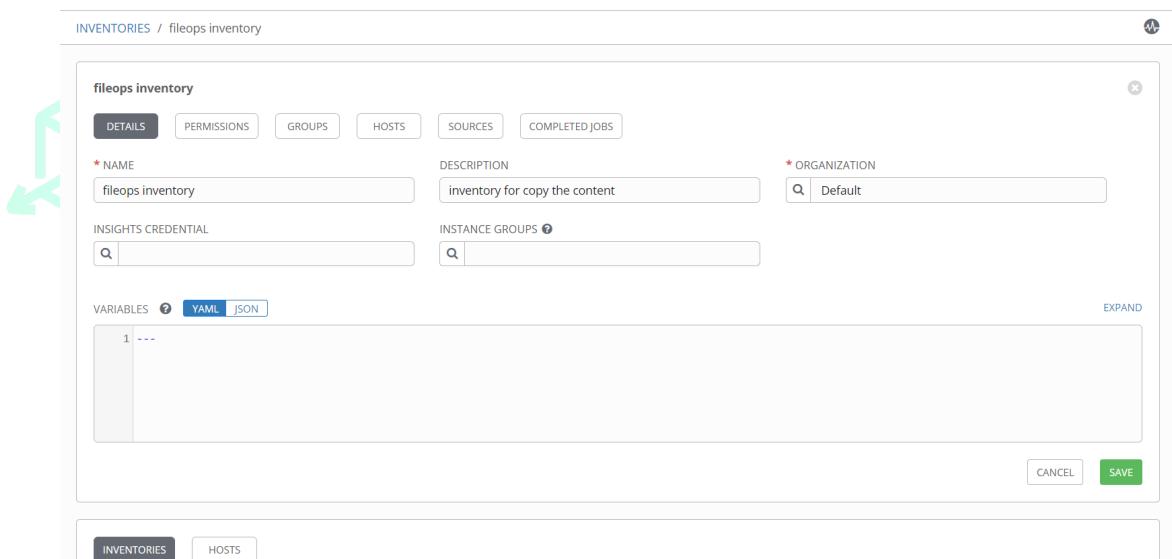


The screenshot shows the Ansible Tower interface with the title 'TOWER' at the top. On the left, there is a sidebar with 'VIEWS' (Dashboard, Jobs, Schedules, My View), 'RESOURCES' (Templates, Credentials, Projects, Inventories, Inventory Scripts), and 'ACCESS' (Organizations). The 'Inventories' tab is selected. The main area is titled 'INVENTORIES' and contains a table with three rows:

NAME	TYPE	ORGANIZATION	ACTIONS
Demo Inventory	Inventory	Default	
Dev	Inventory	Default	
fileops inventory	Inventory	Default	

ITEMS: 1 - 3

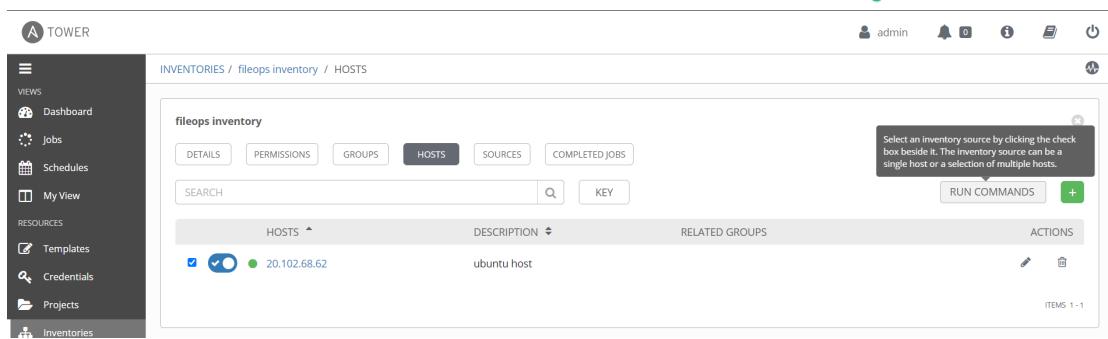
1. Click on inventory where you want to run ad-hoc command. Here we will take fileops inventory.



The screenshot shows the 'fileops inventory' configuration page. At the top, it says 'INVENTORIES / fileops inventory'. Below that is a form with fields: * NAME (fileops inventory), DESCRIPTION (inventory for copy the content), * ORGANIZATION (Default). Under 'INSIGHTS CREDENTIAL' and 'INSTANCE GROUPS', there are search input fields. At the bottom, there are tabs for 'DETAILS', 'PERMISSIONS', 'GROUPS', 'HOSTS', 'SOURCES', and 'COMPLETED JOBS'. The 'HOSTS' tab is highlighted. Below the form is a 'VARIABLES' section with a 'YAML' tab selected. At the bottom right are 'CANCEL' and 'SAVE' buttons. At the very bottom, there are tabs for 'INVENTORIES' and 'HOSTS'.

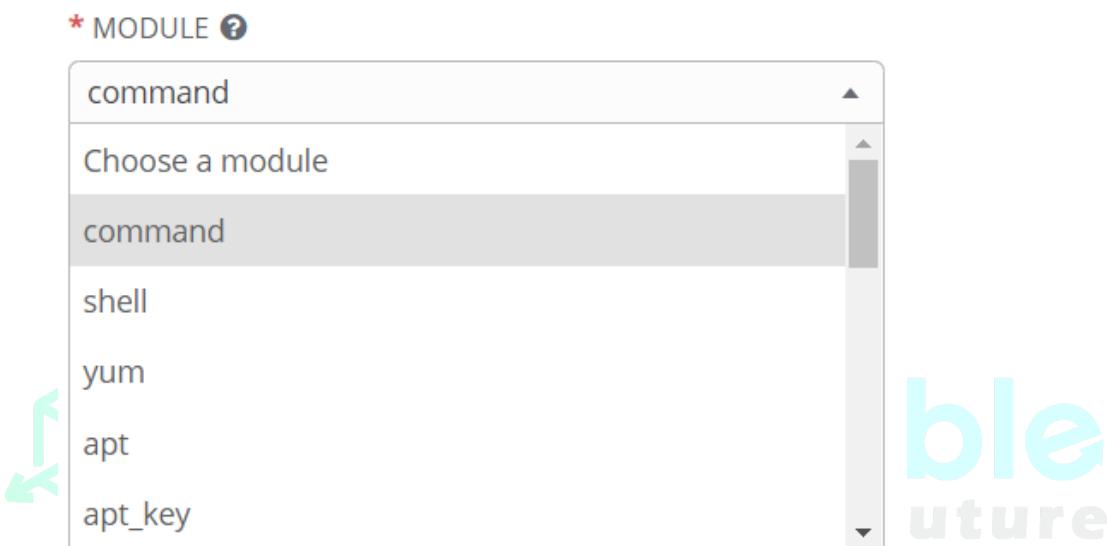
2. Click on **Host**, it will redirect to Host Dashboard.

Select the checkbox which is in front of Host name and then click on Run Commands.



The screenshot shows the Ansible Tower interface. On the left, there's a sidebar with 'TOWER' at the top, followed by 'VIEWS', 'Dashboard', 'Jobs', 'Schedules', 'My View', 'RESOURCES', 'Templates', 'Credentials', 'Projects', and 'Inventories'. The 'Inventories' option is selected. In the main area, the path 'INVENTORIES / fileops inventory / HOSTS' is shown. A modal window titled 'fileops inventory' has the 'HOSTS' tab selected. It contains a search bar and a table with one row: 'HOSTS' (20.102.68.62), 'DESCRIPTION' (ubuntu host), and 'RELATED GROUPS'. There are 'RUN COMMANDS' and '+' buttons at the bottom right. A tooltip says: 'Select an inventory source by clicking the check box beside it. The inventory source can be a single host or a selection of multiple hosts.'

Under module you will get multiple options, based on requirement you can select the module, In this exercise we will go with Command.



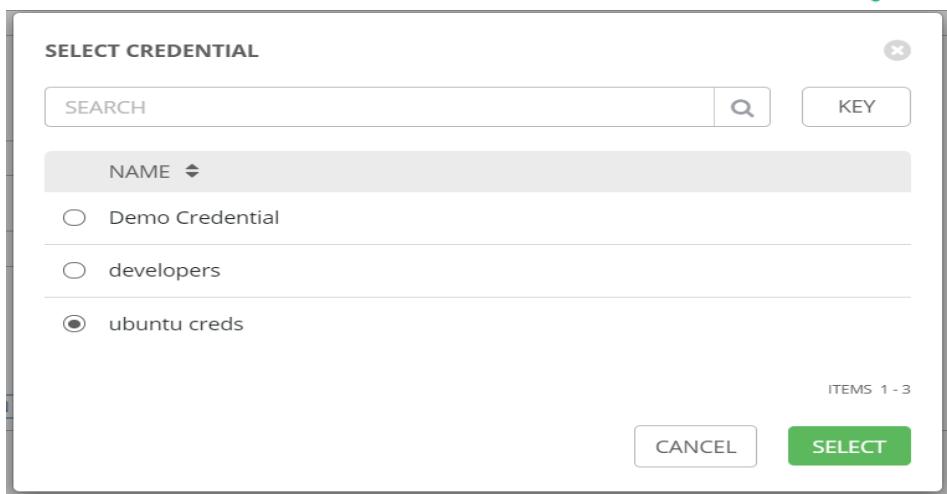
The screenshot shows a dropdown menu titled 'Choose a module'. The 'command' option is selected and highlighted with a green arrow pointing to it. Other options listed are 'shell', 'yum', 'apt', and 'apt_key'.

Here, you will run command "free -m" which should return available memory in Host VM.



The screenshot shows the 'RUN COMMAND' interface. The 'EXECUTE COMMAND' form has the following settings: 'MODULE' is set to 'Command', 'ARGUMENTS' is set to 'free -m', 'LIMIT' is set to 'all', 'MACHINE CREDENTIAL' is empty, 'VERBOSITY' is set to '0 (Normal)', 'FORKS' is set to 'DEFAULT', and 'SHOW CHANGES' is checked. At the bottom, there are tabs for 'EXTRA VARIABLES' (YAML, JSON) and 'YAML' is selected.

Select Ubuntu creds as Machine credentials, click on select.



3. Verify all the fields and click on **Launch** button.

INVENTORIES / fileops inventory / RUN COMMAND

EXECUTE COMMAND

* MODULE ARGUMENTS LIMIT

* MACHINE CREDENTIAL * VERBOSITY FORKS

SHOW CHANGES ENABLE PRIVILEGE ESCALATION

EXTRA VARIABLES

4. it will redirect you to job page, you can see the output of command. And the status is showing as successful.

DETAILS	STATUS	Successful
STARTED	7/25/2021 3:20:09 PM	
FINISHED	7/25/2021 3:20:20 PM	
MODULE ARGS	free -m	
JOB TYPE	Run	
LAUNCHED BY	admin	
INVENTORY	fileops inventory	
CREDENTIAL	<input type="button" value="ubuntu creds"/>	
LIMIT	all	
EXECUTION NODE	localhost	
INSTANCE GROUP	tower	
EXTRA VARIABLES	<input checked="" type="radio" value="YAML"/> <input type="radio" value="JSON"/>	

command

```

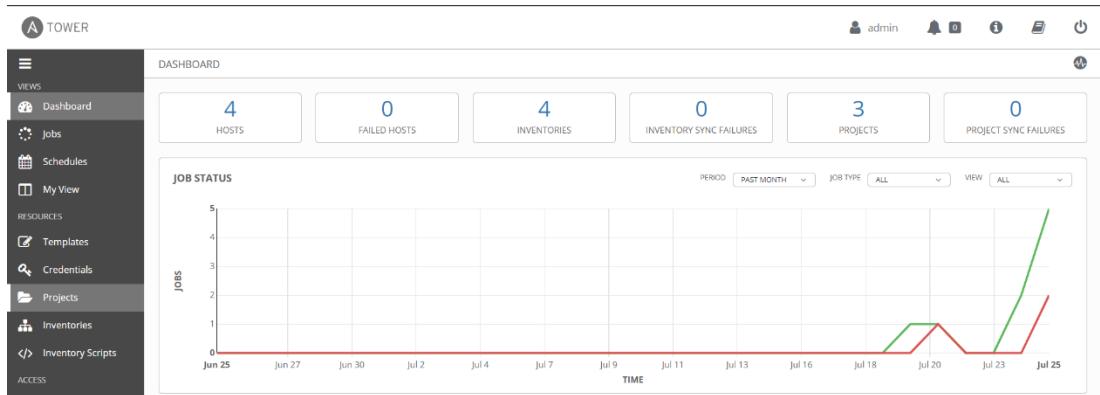
SEARCH
4 use /usr/bin/python3, but is using /usr/bin/python for backward compatibility
5 with prior Ansible releases. A future Ansible release will default to using
the
6 discovered platform python for this host. See https://docs.ansible.com/ansible
7 /2.9/reference_appendices/interpreter_discovery.html for more information.
This
8 feature will be removed in version 2.12. Deprecation warnings can be disabled
9 by setting deprecation_warnings=False in ansible.cfg.
10 20.102.68.62 | CHANGED | rc=0 >>
11      total           used         free        shared   buff/cache   avail
able
12 Mem:       917          158        268          0        490
617
13 Swap:      0            0          0          0

```

12 Install Docker on Host machine

12.1 Create a new Project

1. Create new project and configure it to install docker. Click on **Projects** and add new project.



2. Create a directory at /var/lib/awx/projects to setup Docker project.

```
$ ssh username@ansiblemachineIP
$ sudo su
# cd /var/lib/awx/projects
# mkdir docker
$ vi docker.yml
```

```
[allen@ansible-tower ~]$ sudo su
[root@ansible-tower allen]# cd /var/lib/awx/projects/
[root@ansible-tower projects]# 

[root@ansible-tower docker]#
[root@ansible-tower docker]# vi docker.yml
```

Note: Here, highlighted lines includes Ansible Variables, we have declared variables in the playbook and we need to pass their values in the coming steps.

```
#####
# DO Community Playbooks: Docker
#####
---
- hosts: all
  become: true

  tasks:
    - name: Install aptitude using apt
      apt: name=aptitude state=latest update_cache=yes force_apt_get=yes

    - name: Install required system packages
      apt: name={{ item }} state=latest update_cache=yes
      loop: [ 'apt-transport-https', 'ca-certificates', 'curl', 'software-properties-common', 'python3-pip', 'virtualenv', 'python3-setuptools']

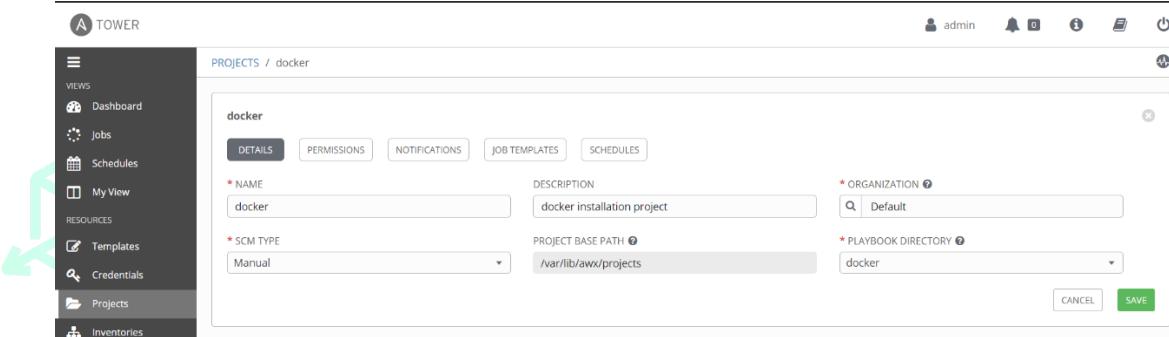
    - name: Add Docker GPG apt Key
      apt_key:
        url:"{{ docker_key }}"
        state: present

    - name: Add Docker Repository
      apt_repository:
        repo:"{{ docker_repo }}"
        state: present

    - name: Update apt and install docker-ce
      apt: update_cache=yes name=docker-ce state=latest

    - name: verify docker status
      command: systemctl status docker
  ~
  ~
  ~
```

3. Click on + button and add new project with following configuration.

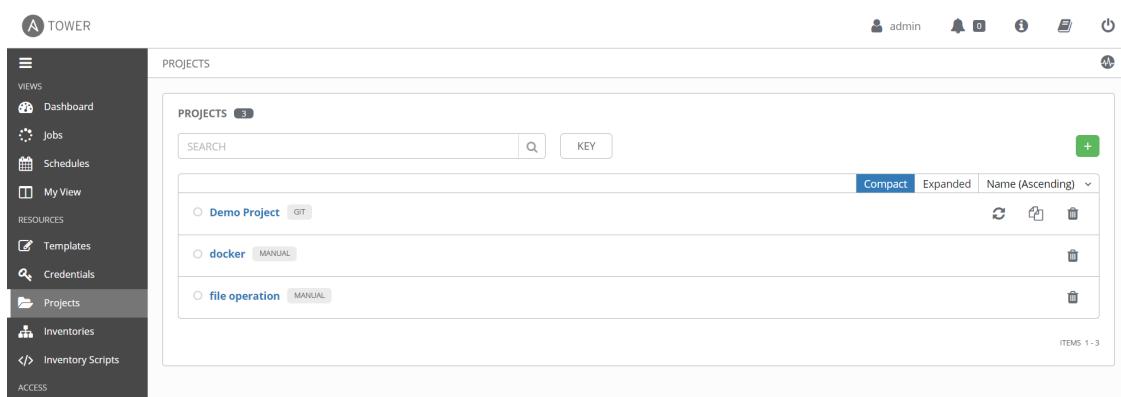


The screenshot shows the 'PROJECTS / docker' page in Ansible Tower. The 'docker' project details are displayed:

- NAME:** docker
- DESCRIPTION:** docker installation project
- ORGANIZATION:** Default
- SCM TYPE:** Manual
- PROJECT BASE PATH:** /var/lib/awx/projects
- PLAYBOOK DIRECTORY:** docker

At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

4. Make sure all details are filled up and **save** the project. Now on Project Dashboard you will be able to see docker project.



The screenshot shows the 'PROJECTS' dashboard in Ansible Tower, displaying the following projects:

- Demo Project (GIT)
- docker (MANUAL)
- file operation (MANUAL)

At the top right, there is a '+ icon' for creating new projects.

5. Create inventory to run this project. From left panel click on Inventories and add new inventory for docker project. Click on + icon to create new inventory.

A TOWER

VIEWS
Dashboard
Jobs
Schedules
My View
RESOURCES
Templates
Credentials
Projects

INVENTORIES

INVENTORIES HOSTS

SEARCH KEY

Create a new inventory +

NAME	TYPE	ORGANIZATION	ACTIONS		
Demo Inventory	Inventory	Default			
Dev	Inventory	Default			
fileops inventory	Inventory	Default			

6. Provide following details and configure the inventory.

Click on **Save**.

INVENTORIES / Management

Management

DETAILS PERMISSIONS GROUPS HOSTS SOURCES COMPLETED JOBS

* NAME Management DESCRIPTION inventory for docker and artifactory * ORGANIZATION Default

INSIGHTS CREDENTIAL INSTANCE GROUPS

VARIABLES ? YAML JSON EXPAND

1 ...

CANCEL SAVE

7. Click on Host tab and add new host to Run docker playbook. Provide below details and **save** the configuration.

INVENTORIES / Management / HOSTS / 20.106.152.215

20.106.152.215

DETAILS FACTS GROUPS COMPLETED JOBS

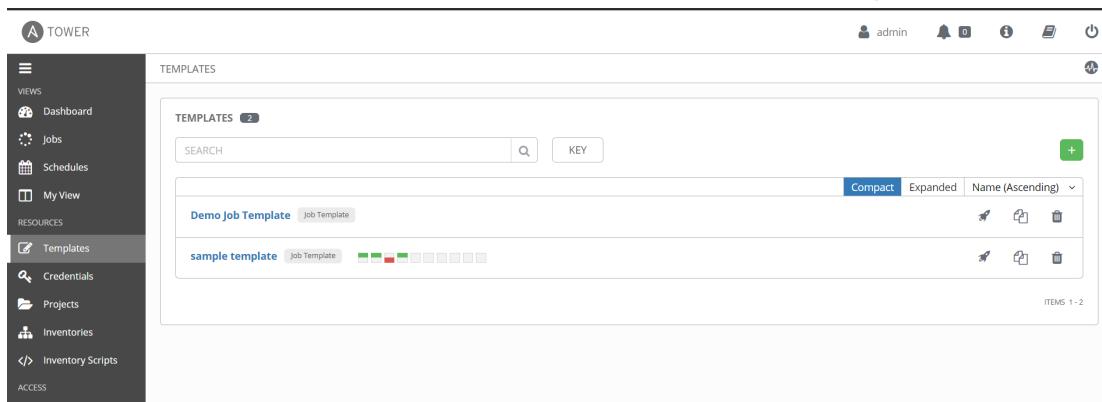
* HOST NAME 20.106.152.215 DESCRIPTION host for docker and artifactory

VARIABLES ? YAML JSON EXPAND

1 ...

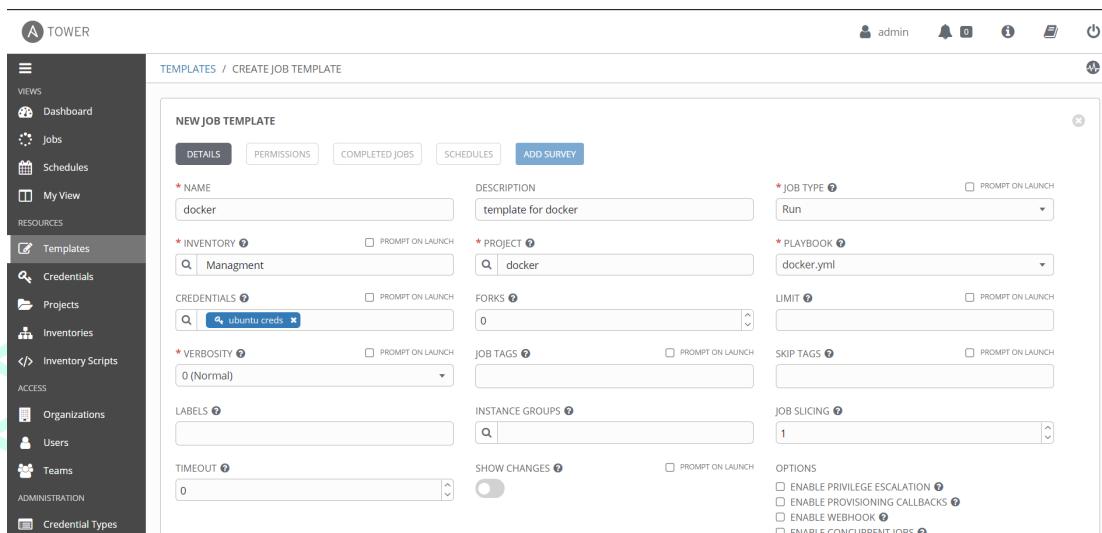
CANCEL SAVE

8. From the left panel click on **Templates** tab. On templates dashboard click on + icon and create a template to run Docker playbook.



The screenshot shows the Ansible Tower 'TEMPLATES' page. On the left is a sidebar with 'TOWER' at the top, followed by 'VIEWS', 'RESOURCES' (with 'Templates' selected), and 'ACCESS'. The main area displays a table of templates. The first row contains 'Demo Job Template' (Job Template) with a green status icon. The second row contains 'sample template' (Job Template) with a red status icon. Both rows have edit, copy, and delete icons. A search bar and a 'KEY' button are at the top of the table. Below the table is a message 'ITEMS 1 - 2'.

9. Configure the template as below. Provide appropriate details and **save** the template.



The screenshot shows the 'NEW JOB TEMPLATE' dialog. The 'DETAILS' tab is active. The configuration includes:

- NAME:** docker
- DESCRIPTION:** template for docker
- JOB TYPE:** Run
- INVENTORY:** Managment
- PROJECT:** docker
- PLAYBOOK:** docker.yml
- CREDENTIALS:** ubuntu creds
- FORKS:** 0
- LIMIT:** (empty)
- VERBOSITY:** 0 (Normal)
- JOB TAGS:** (empty)
- SKIP TAGS:** (empty)
- LABELS:** (empty)
- INSTANCE GROUPS:** (empty)
- JOB SLICING:** 1
- TIMEOUT:** 0
- OPTIONS:**
 - ENABLE PRIVILEGE ESCALATION
 - ENABLE PROVISIONING CALLBACKS
 - ENABLE WEBHOOK
 - ENABLE CONCURRENT JOBS

Scroll down and add variables for docker key and docker repository.



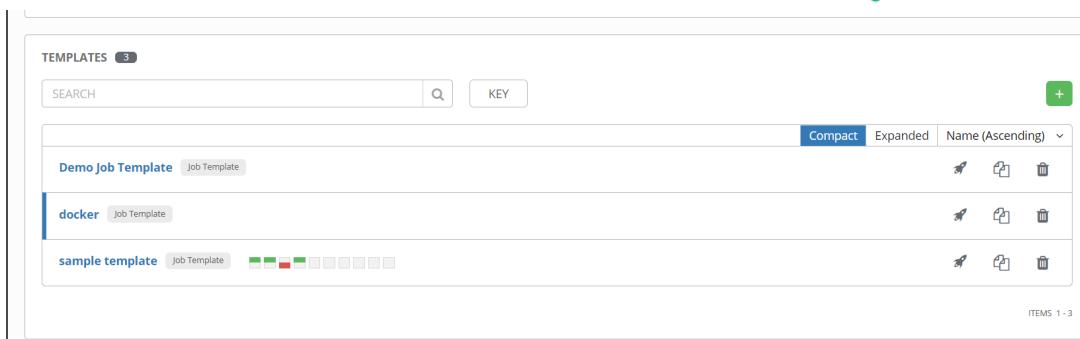
The screenshot shows the 'EXTRA VARIABLES' section. It has tabs for 'YAML' (selected) and 'JSON'. The YAML content is:

```

1 ---
2 docker_key: https://download.docker.com/linux/ubuntu/gpg
3 docker_repo: deb https://download.docker.com/linux/ubuntu bionic stable
  
```

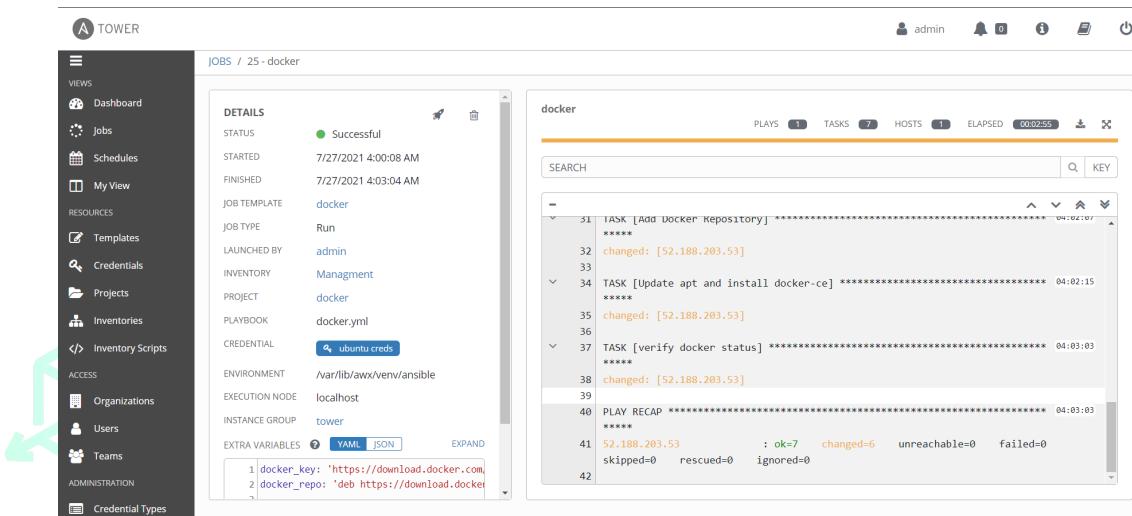
Below the code editor are 'LAUNCH', 'CANCEL', and 'SAVE' buttons.

10. On successful configuration this template will be available on Templates dashboard. You need to run this template, Click on **Rocket** icon to run the template.



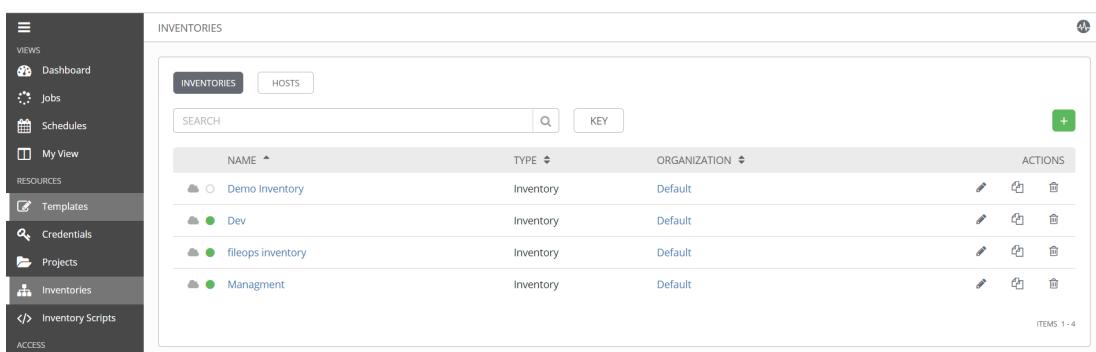
ITEMS 1 - 3

11. It will redirect to log screen, where you can see details about template execution.

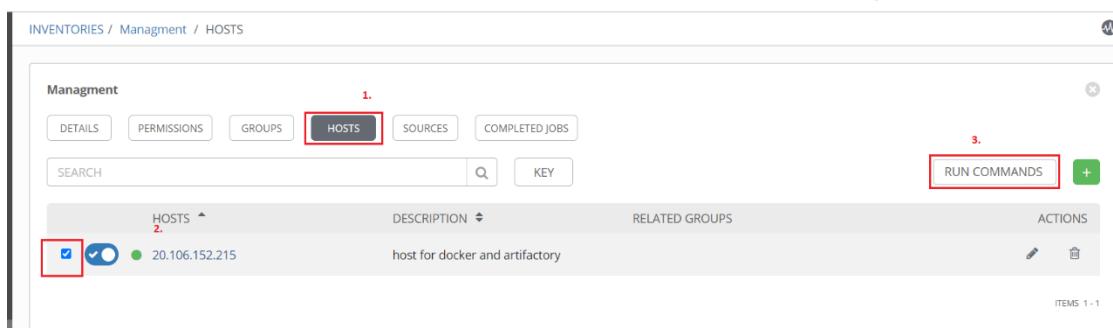


Docker.yml playbook is executed successfully, let us verify docker is being install on host machine. You need to use ad hoc command to verify docker status on hot machine.

12. Go to inventories to run ad hoc command. Select management inventory.



13. Click on host and tick on checkbox in front of Host.



INVENTORIES / Management / HOSTS

Management

HOSTS 1.

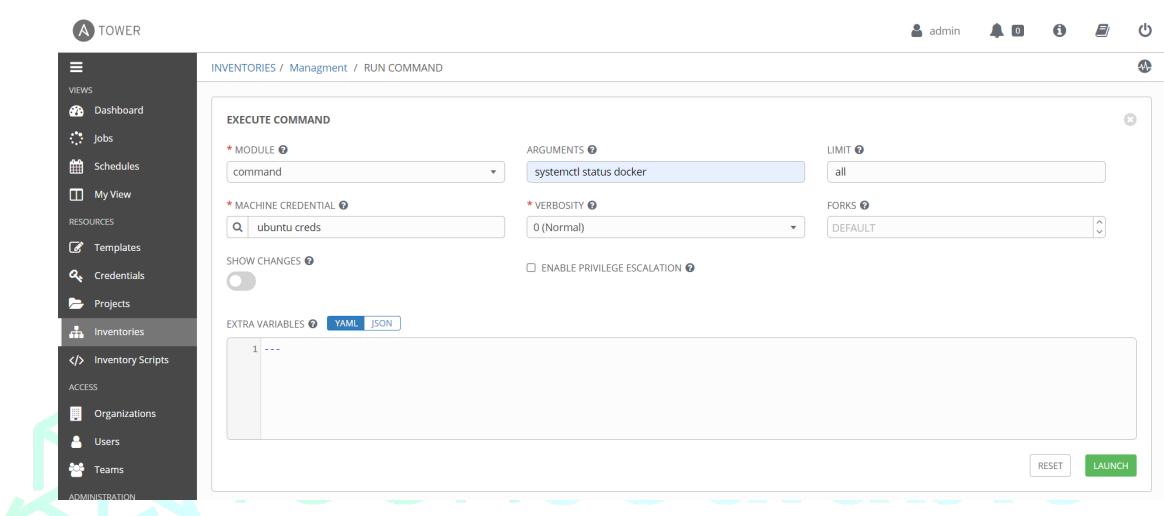
SEARCH 2.

RUN COMMANDS 3.

HOSTS	DESCRIPTION	RELATED GROUPS	ACTIONS
20.106.152.215	host for docker and artifactory		

ITEMS 1 - 1

Configure it with below given options and click on **launch**.



TOWER

INVENTORIES / Management / RUN COMMAND

EXECUTE COMMAND

* MODULE command

ARGUMENTS systemctl status docker

LIMIT all

* MACHINE CREDENTIAL ubuntu creds

* VERBOSITY 0 (Normal)

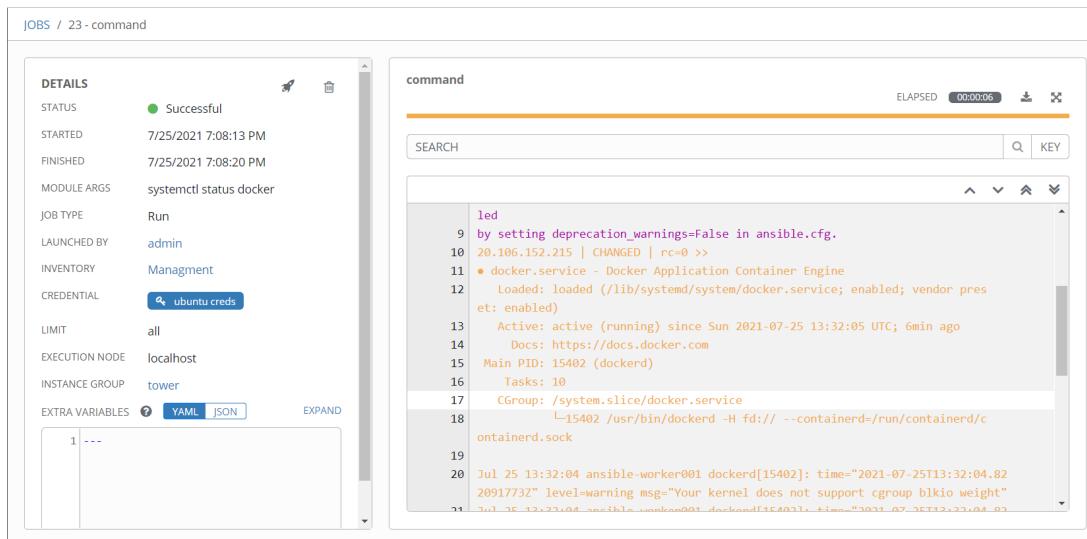
FORKS DEFAULT

SHOW CHANGES ENABLE PRIVILEGE ESCALATION

EXTRA VARIABLES

LAUNCH

14. In the logs you will be able to see the status of Docker service. The job has run successfully, and we can see that docker is up and running on the Host machine.



JOBS / 23 - command

DETAILS

STATUS: Successful

STARTED: 7/25/2021 7:08:13 PM

FINISHED: 7/25/2021 7:08:20 PM

MODULE ARGS: systemctl status docker

JOB TYPE: Run

LAUNCHED BY: admin

INVENTORY: Management

CREDENTIAL: ubuntu creds

LIMIT: all

EXECUTION NODE: localhost

INSTANCE GROUP: tower

EXTRA VARIABLES:

command

SEARCH

```

led
9 by setting deprecation_warnings=False in ansible.cfg.
10 20.106.152.215 | CHANGED | rc=0 >>
11 ● docker.service - Docker Application Container Engine
12   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor pres
et: enabled)
13     Active: active (running) since Sun 2021-07-25 13:32:05 UTC; 6min ago
14       Docs: https://docs.docker.com
15 Main PID: 15402 (dockerd)
16   Tasks: 10
17   CGroup: /system.slice/docker.service
18             └─15402 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/
19               containerd.sock
20 Jul 25 13:32:04 ansible-worker001 dockerd[15402]: time="2021-07-25T13:32:04.82
2091773Z" level=warning msg="Your kernel does not support cgroup blkio weight"
21 Jul 25 13:32:04 ansible-worker001 dockerd[15402]: time="2021-07-25T13:32:04.82
2091773Z" level=warning msg="Your kernel does not support cgroup blkio weight"

```