Lab: Installing Ansible AWX

Introduction:

AWX is a modern web UI and API to manage your organization's Ansible Playbook, Inventories, Vault, and Credentials.

It is the Open-Source upstream project of the Ansible Automation Controller (formerly Ansible Tower).

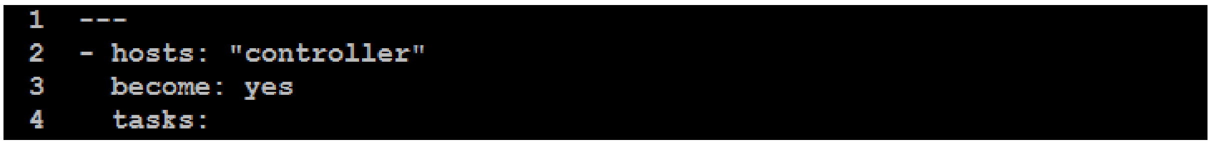
Objectives:

- **Adding kustomization tool**
- **Creating an awx instance**
- **Creating pv and accessing the AWX UI**
- **Open the AWX UI**
- **Creating an organization.**
- **Creating a sample automation.**

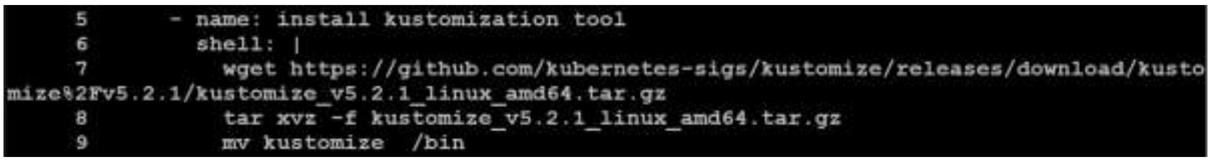
Note: Login to **eoc-controller** as **admin** user with password as **linux**

1. Adding kustomization tool.

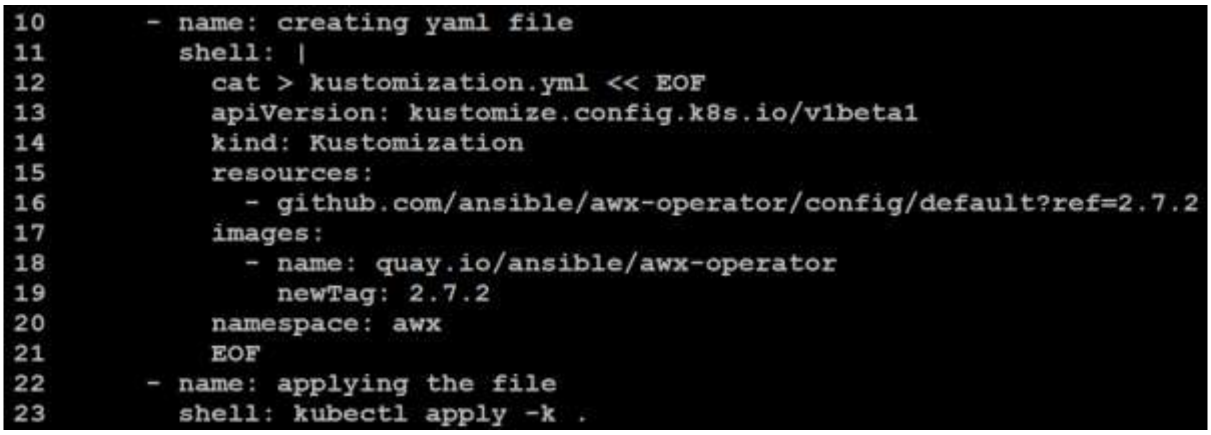
1.1 Let's create a playbook **kustom.yml** to add kustomization tool and installing awx operator.


```
---
- hosts: "controller"
  become: yes
  tasks:
```

1.2 Installing the **kustomization** tool.


```
    - name: install kustomization tool
      shell: |
        wget https://github.com/kubernetes-sigs/kustomize/releases/download/kustomize%2Fv5.2.1/kustomize_v5.2.1_linux_amd64.tar.gz
        tar xvz -f kustomize_v5.2.1_linux_amd64.tar.gz
        mv kustomize  /bin
```

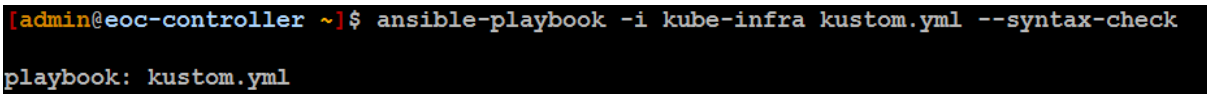
1.3 Creating and applying the **kustomization.yml** file.


```
    - name: creating yaml file
      shell: |
        cat > kustomization.yml << EOF
        apiVersion: kustomize.config.k8s.io/v1beta1
        kind: Kustomization
        resources:
          - github.com/ansible/awx-operator/config/default?ref=2.7.2
        images:
          - name: quay.io/ansible/awx-operator
            newTag: 2.7.2
        namespace: awx
        EOF
    - name: applying the file
      shell: kubectl apply -k .
```

1.4 Verify the syntax of the yaml by executing below command.

```
# ansible-playbook -i kube-infra kustom.yml --syntax-check
```

Output:

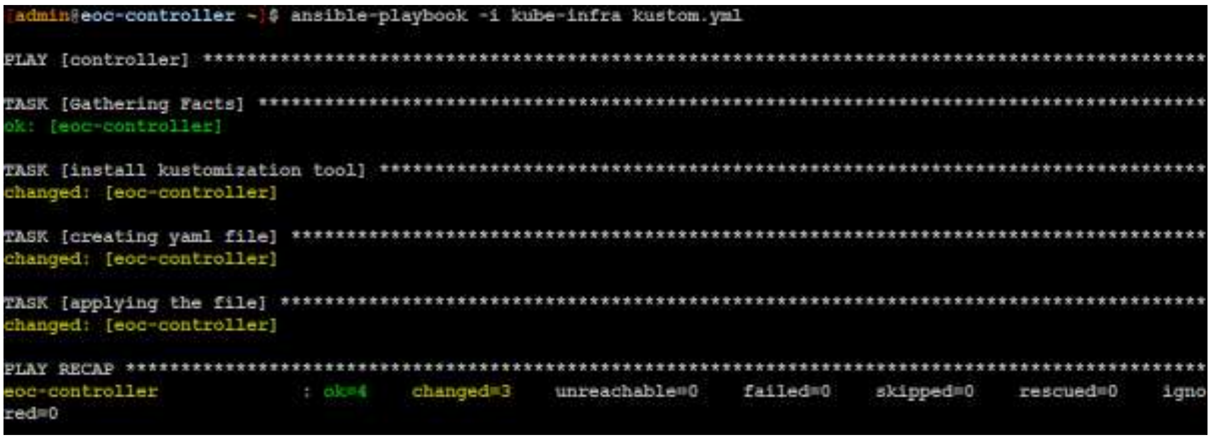


```
[admin@eoc-controller ~]$ ansible-playbook -i kube-infra kustom.yml --syntax-check

playbook: kustom.yml
```

1.5 Let’s run the playbook by executing below command.

```
# ansible-playbook -i kube-infra kustom.yml
```

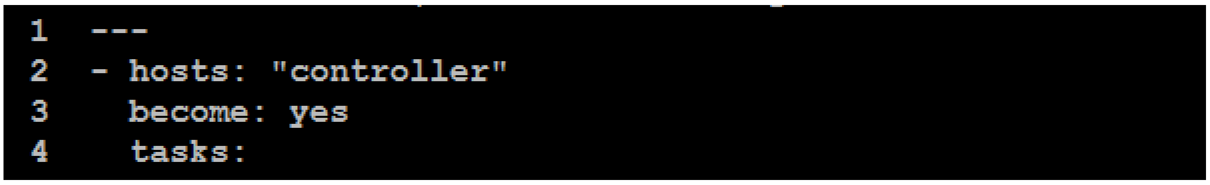
Output:



```
[admin@eoc-controller ~]$ ansible-playbook -i kube-infra kustom.yml

PLAY [controller] ***************************************************************************

TASK [Gathering Facts] **********************************************************************
ok: [eoc-controller]

TASK [install kustomization tool] ***********************************************************
changed: [eoc-controller]

TASK [creating yaml file] *******************************************************************
changed: [eoc-controller]

TASK [applying the file] ********************************************************************
changed: [eoc-controller]

PLAY RECAP **********************************************************************************
eoc-controller             : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    igno
red=0
```

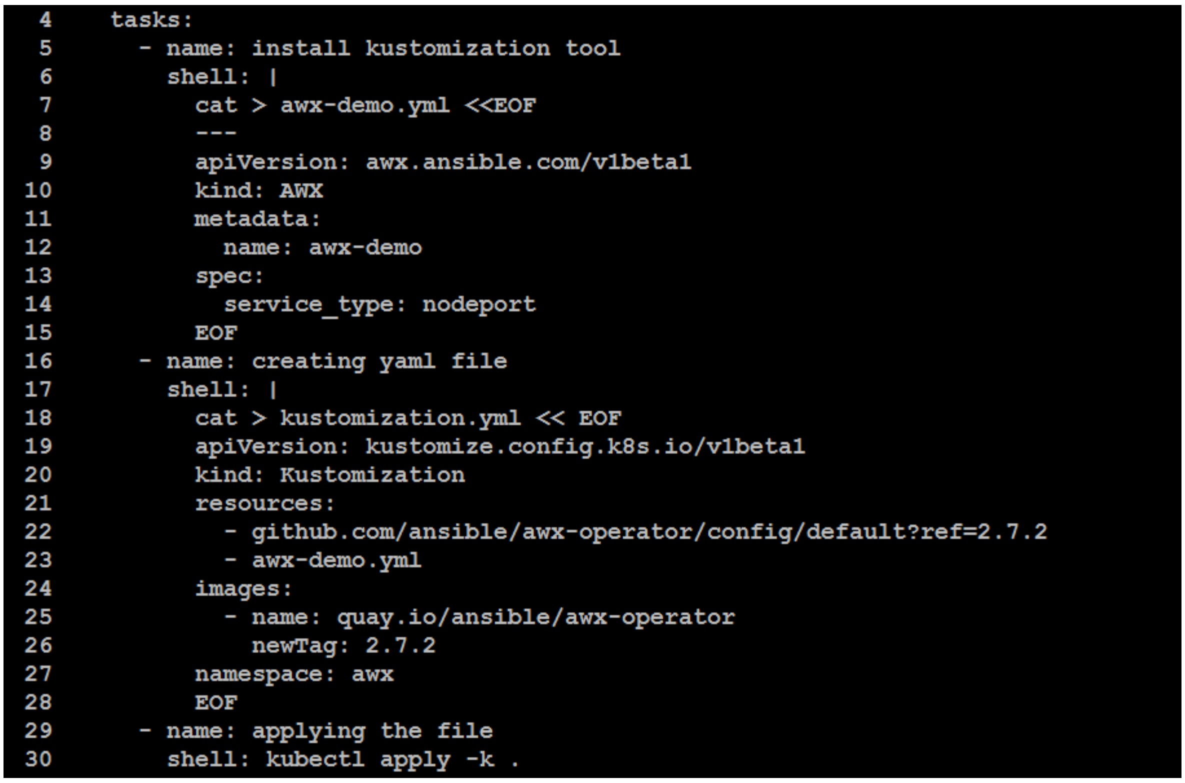
2. Creating an awx instance.

2.1 Let’s create an ansible playbook **awx.yml** which creates an awx instance.



```
---
- hosts: "controller"
  become: yes
  tasks:
```

2.2 Let’s add the **awx.yml** and configure the **kustom.yml** file and apply it.

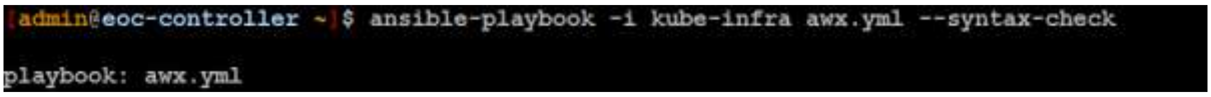


```
  tasks:
    - name: install kustomization tool
      shell: |
        cat > awx-demo.yml <<EOF
        ---
        apiVersion: awx.ansible.com/v1beta1
        kind: AWX
        metadata:
          name: awx-demo
        spec:
          service_type: nodeport
        EOF
    - name: creating yaml file
      shell: |
        cat > kustomization.yml << EOF
        apiVersion: kustomize.config.k8s.io/v1beta1
        kind: Kustomization
        resources:
          - github.com/ansible/awx-operator/config/default?ref=2.7.2
          - awx-demo.yml
        images:
          - name: quay.io/ansible/awx-operator
            newTag: 2.7.2
        namespace: awx
        EOF
    - name: applying the file
      shell: kubectl apply -k .
```

2.3 Let's verify the syntax of the file **awx.yml.**

```
# ansible-playbook -i kube-infra awx.yml --syntax-check
```

Output:

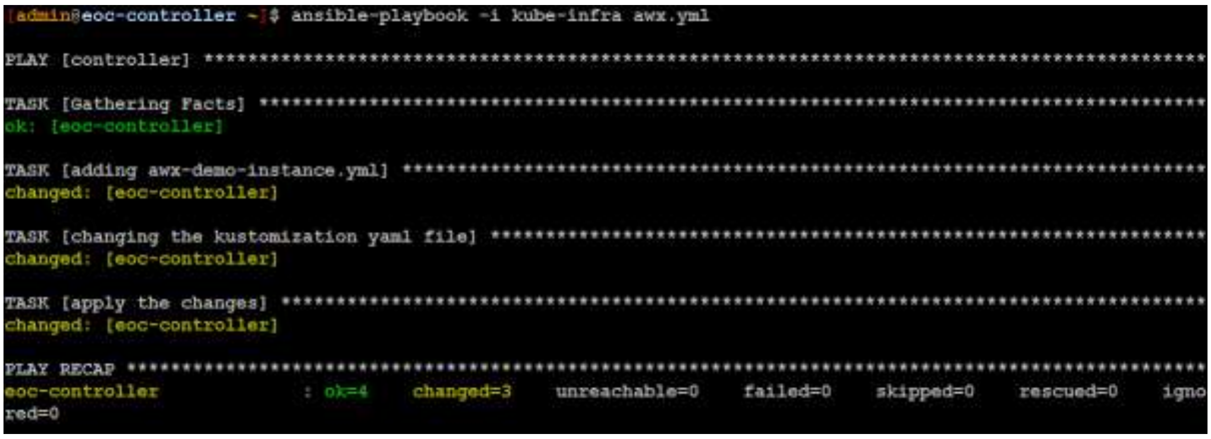


```
[admin@eoc-controller ~]$ ansible-playbook -i kube-infra awx.yml --syntax-check

playbook: awx.yml
```

2.4 Let's run the play book by executing below command.

```
# ansible-playbook -i kube-infra awx.yml
```

Output:

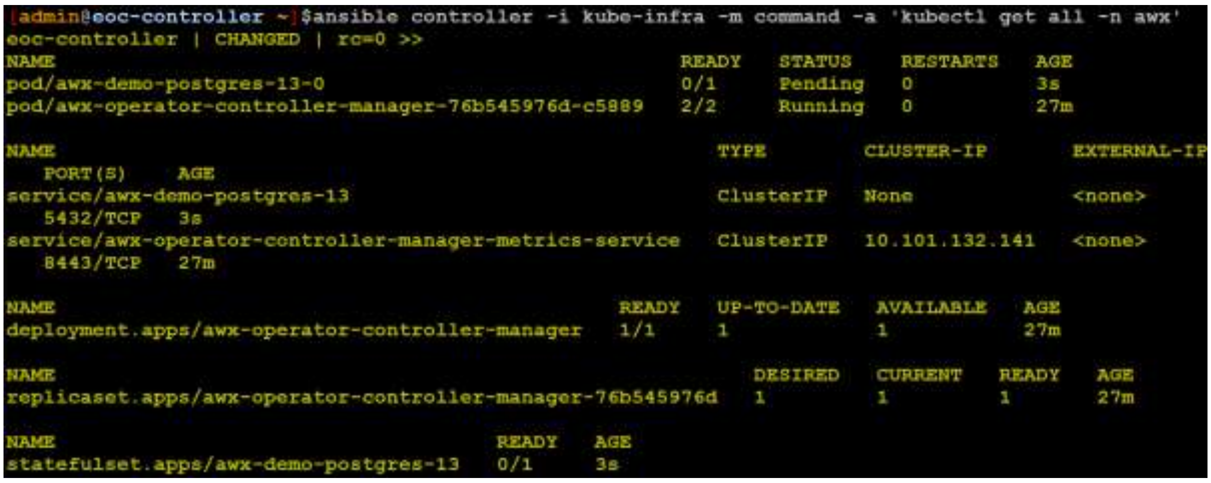


```
[admin@eoc-controller ~]$ ansible-playbook -i kube-infra awx.yml

PLAY [controller] **************************************************************************

TASK [Gathering Facts] *********************************************************************
ok: [eoc-controller]

TASK [adding awx-demo-instance.yml] ********************************************************
changed: [eoc-controller]

TASK [changing the kustomization yaml file] ************************************************
changed: [eoc-controller]

TASK [apply the changes] *******************************************************************
changed: [eoc-controller]

PLAY RECAP *********************************************************************************
eoc-controller             : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    igno
red=0
```

2.5 Check for the created resources using the **ad-hoc** commands.

```
# ansible controller -i kube-infra -m command -a 'kubectl
get all -n awx'
```

Output:

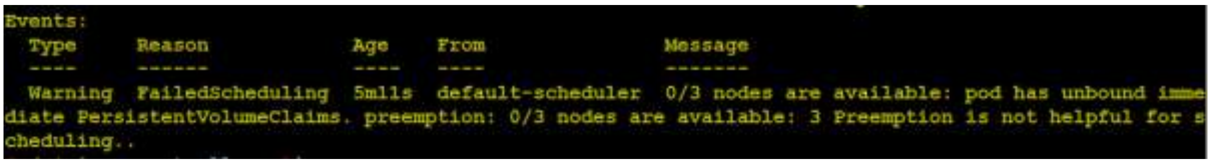


```
[admin@eoc-controller ~]$ansible controller -i kube-infra -m command -a 'kubectl get all -n awx'
eoc-controller | CHANGED | rc=0 >>
NAME                                                       READY   STATUS    RESTARTS   AGE
pod/awx-demo-postgres-13-0                                 0/1     Pending   0          3s
pod/awx-operator-controller-manager-76b545976d-c5889       2/2     Running   0          27m

NAME                                                          TYPE        CLUSTER-IP       EXTERNAL-IP
   PORT(S)     AGE
service/awx-demo-postgres-13                                  ClusterIP   None             <none>
   5432/TCP    3s
service/awx-operator-controller-manager-metrics-service       ClusterIP   10.101.132.141   <none>
   8443/TCP    27m

NAME                                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/awx-operator-controller-manager   1/1     1            1           27m

NAME                                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/awx-operator-controller-manager-76b545976d   1         1         1       27m

NAME                                        READY   AGE
statefulset.apps/awx-demo-postgres-13       0/1     3s
```

2.6 Let’s describe the pod and check.

```
# ansible controller -i kube-infra -m command -a 'kubectl
describe pod/awx-demo-postgres-13-0 -n awx'
```

Output:



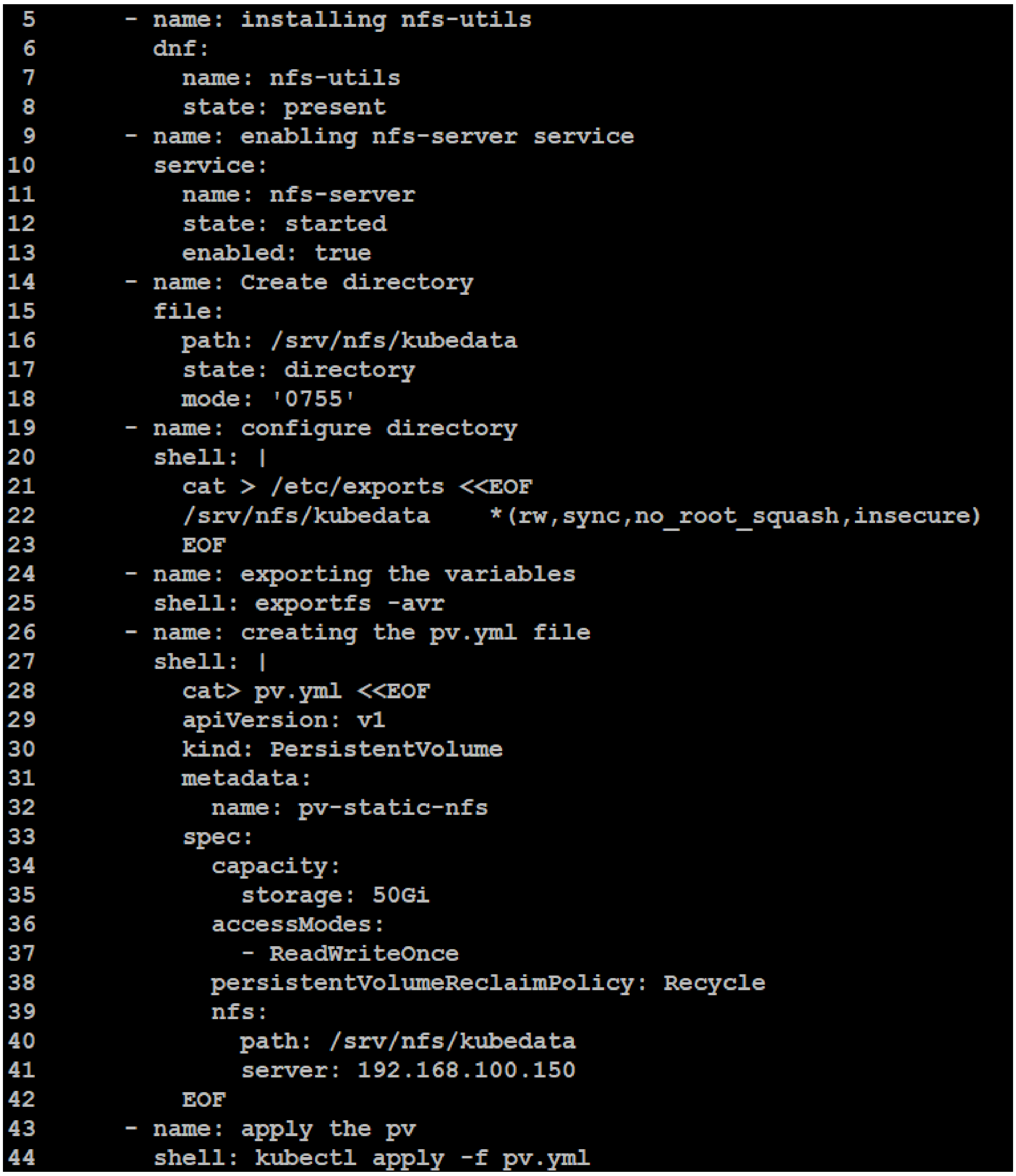
```
Events:
  Type     Reason            Age    From               Message
  ----     ------            ----   ----               -------
  Warning  FailedScheduling  5m11s  default-scheduler  0/3 nodes are available: pod has unbound imme
diate PersistentVolumeClaims. preemption: 0/3 nodes are available: 3 Preemption is not helpful for s
cheduling..
```

3. **Creating pv and accessing the AWX UI**.

3.1 Let’s create an ansible-playbook **demo-pv.yml** for creating a PV the requested storage by the pvc created by the instance is 8Gi

```
---
- hosts: "controller"
  become: yes
  tasks:
```

3.2 Creating and configuring a pv storage.

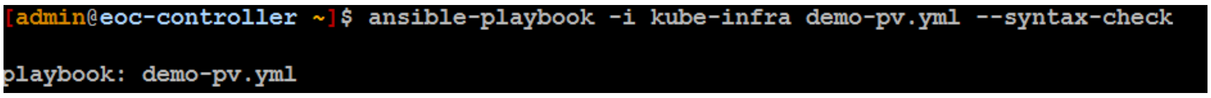


```
    - name: installing nfs-utils
      dnf:
        name: nfs-utils
        state: present
    - name: enabling nfs-server service
      service:
        name: nfs-server
        state: started
        enabled: true
    - name: Create directory
      file:
        path: /srv/nfs/kubedata
        state: directory
        mode: '0755'
    - name: configure directory
      shell: |
        cat > /etc/exports <<EOF
        /srv/nfs/kubedata    *(rw,sync,no_root_squash,insecure)
        EOF
    - name: exporting the variables
      shell: exportfs -avr
    - name: creating the pv.yml file
      shell: |
        cat> pv.yml <<EOF
        apiVersion: v1
        kind: PersistentVolume
        metadata:
          name: pv-static-nfs
        spec:
          capacity:
            storage: 50Gi
          accessModes:
            - ReadWriteOnce
          persistentVolumeReclaimPolicy: Recycle
          nfs:
            path: /srv/nfs/kubedata
            server: 192.168.100.150
        EOF
    - name: apply the pv
      shell: kubectl apply -f pv.yml
```

3.3 Let's verify the syntax of the playbook.

```
# ansible-playbook -i kube-infra demo-pv.yml --syntax-check
```

Output:

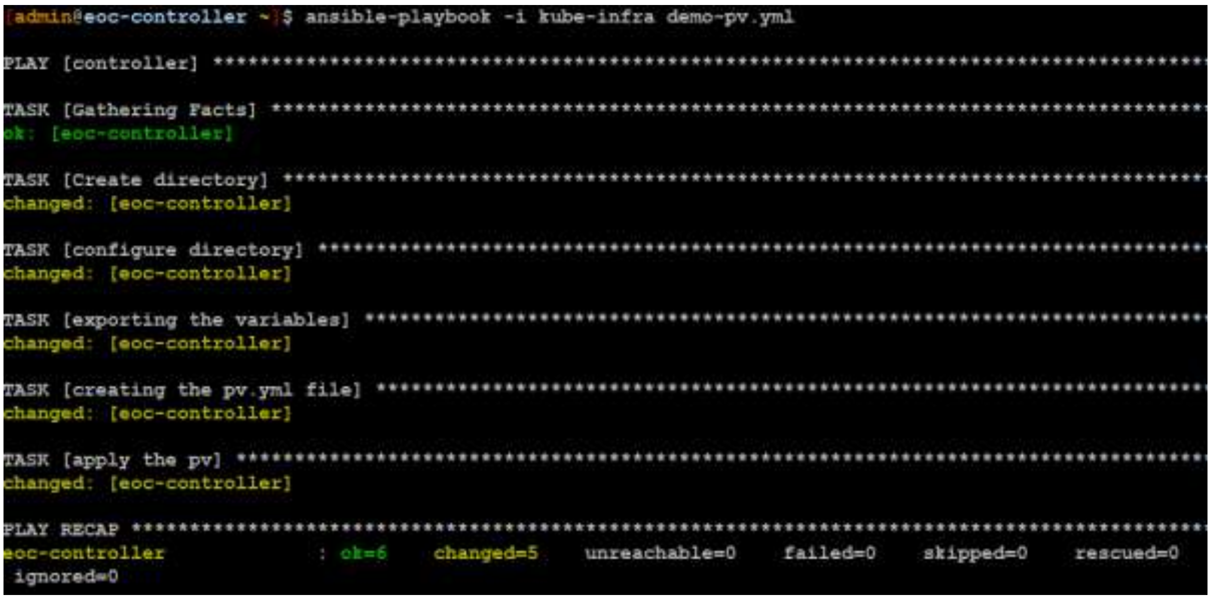


```
[admin@eoc-controller ~]$ ansible-playbook -i kube-infra demo-pv.yml --syntax-check

playbook: demo-pv.yml
```

3.4 Let's run the playbook to create a **pv** by executing below command.

```
# ansible-playbook -i kube-infra demo-pv.yml
```

Output:

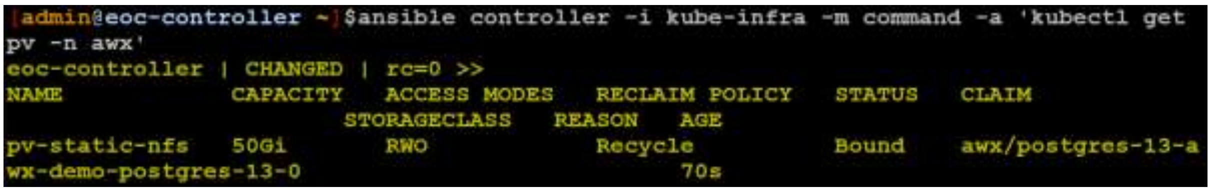


```
[admin@eoc-controller ~]$ ansible-playbook -i kube-infra demo-pv.yml

PLAY [controller] ***************************************************************************

TASK [Gathering Facts] **********************************************************************
ok: [eoc-controller]

TASK [Create directory] *********************************************************************
changed: [eoc-controller]

TASK [configure directory] ******************************************************************
changed: [eoc-controller]

TASK [exporting the variables] **************************************************************
changed: [eoc-controller]

TASK [creating the pv.yml file] *************************************************************
changed: [eoc-controller]

TASK [apply the pv] *************************************************************************
changed: [eoc-controller]

PLAY RECAP **********************************************************************************
eoc-controller             : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
```

3.5 Let's check the status of the **pv** by executing below command.

```
# ansible controller -i kube-infra -m command -a 'kubectl
get pv -n awx'
```

Output:



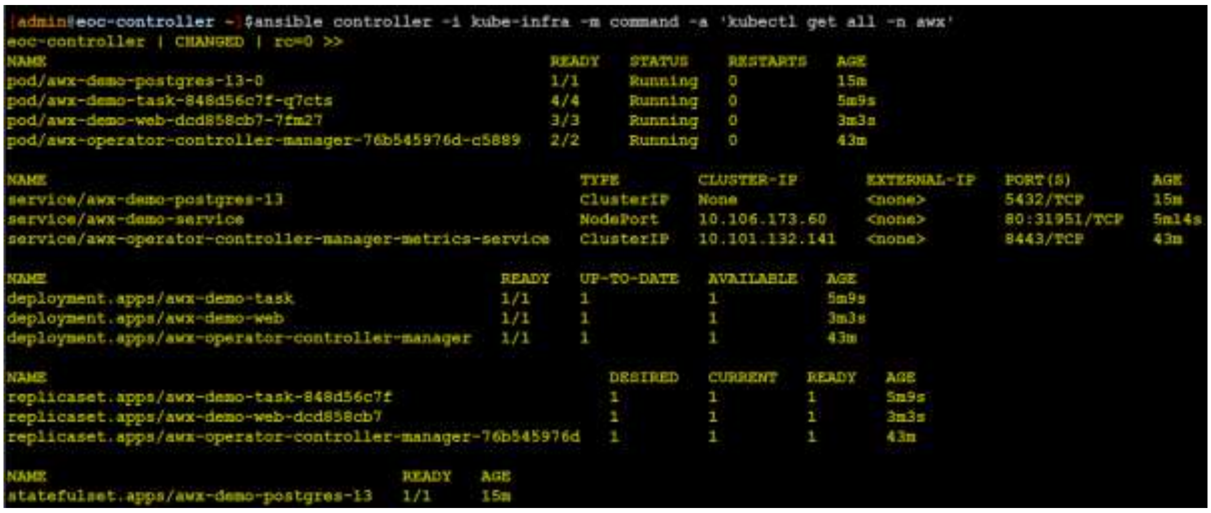
```
[admin@eoc-controller ~]$ansible controller -i kube-infra -m command -a 'kubectl get
pv -n awx'
eoc-controller | CHANGED | rc=0 >>
NAME              CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
        STORAGECLASS   REASON   AGE
pv-static-nfs     50Gi       RWO            Recycle          Bound    awx/postgres-13-a
wx-demo-postgres-13-0                       70s
```

Note: **The pv has already been bound by our pvc.**

3.6 Let's check for our resources which had been in pending state.

```
# ansible controller -i kube-infra -m command -a 'kubectl
get all -n awx'
```

Output:



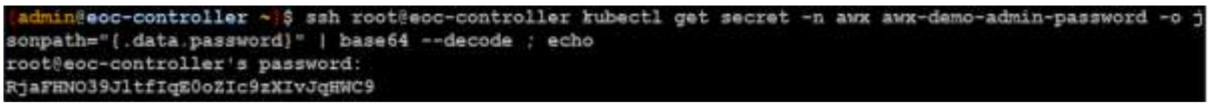
```
[admin@eoc-controller ~]$ansible controller -i kube-infra -m command -a 'kubectl get all -n awx'
eoc-controller | CHANGED | rc=0 >>
NAME                                                  READY   STATUS    RESTARTS   AGE
pod/awx-demo-postgres-13-0                            1/1     Running   0          15m
pod/awx-demo-task-848d56c7f-q7cts                     4/4     Running   0          5m9s
pod/awx-demo-web-dcd858cb7-7fm27                      3/3     Running   0          3m3s
pod/awx-operator-controller-manager-76b545976d-c5889  2/2     Running   0          43m

NAME                                                     TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
service/awx-demo-postgres-13                             ClusterIP   None             <none>        5432/TCP       15m
service/awx-demo-service                                 NodePort    10.106.173.60    <none>        80:31951/TCP   5m14s
service/awx-operator-controller-manager-metrics-service  ClusterIP   10.101.132.141   <none>        8443/TCP       43m

NAME                                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/awx-demo-task                    1/1     1            1           5m9s
deployment.apps/awx-demo-web                     1/1     1            1           3m3s
deployment.apps/awx-operator-controller-manager  1/1     1            1           43m

NAME                                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/awx-demo-task-848d56c7f                     1         1         1       5m9s
replicaset.apps/awx-demo-web-dcd858cb7                      1         1         1       3m3s
replicaset.apps/awx-operator-controller-manager-76b545976d  1         1         1       43m

NAME                                    READY   AGE
statefulset.apps/awx-demo-postgres-13   1/1     15m
```

Note: Wait for atleast 3 min to get the pods up and running. **Reboot All Servers Once after this.**

3.7 Get the initial password of the AWX UI.

```
# ssh root@eoc-controller kubectl get secret -n awx awx-
demo-admin-password -o jsonpath="{.data.password}" | base64
--decode ; echo
```

Output:



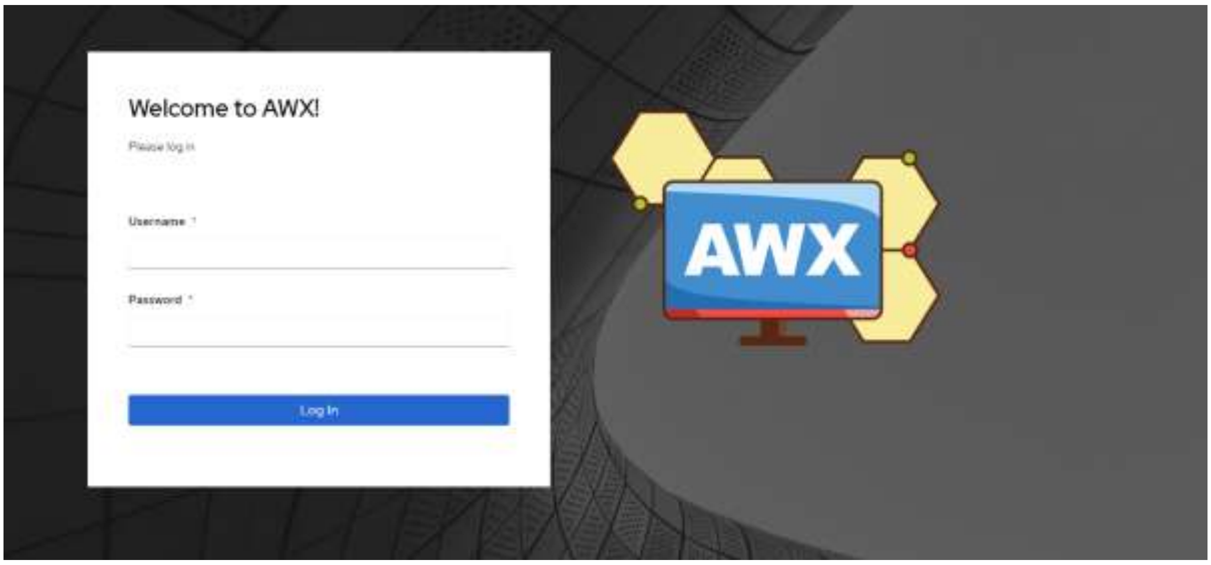
```
[admin@eoc-controller ~]$ ssh root@eoc-controller kubectl get secret -n awx awx-demo-admin-password -o j
sonpath="{.data.password}" | base64 --decode ; echo
root@eoc-controller's password:
RjaFHNO39J1tfIqE0oZIc9zXIvJqHWC9
```

Note: When prompted enter the password as **linux.**

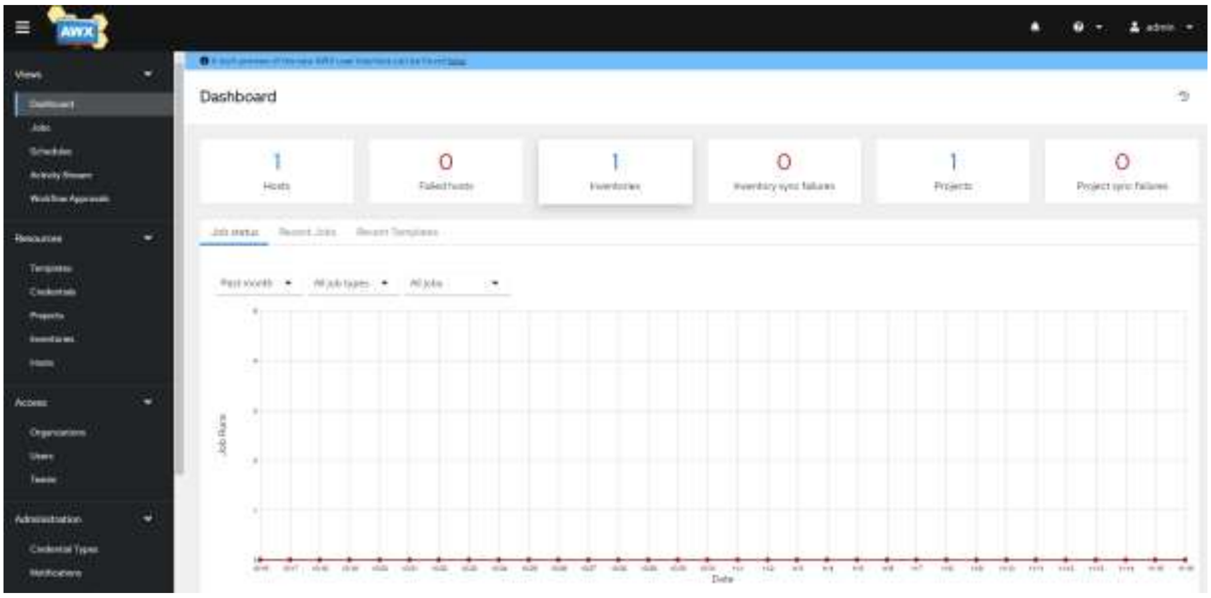
4. Open the AWX UI

4.1 In the UI using the URL http://192.168.100.150:Nodeport

Note: Change with your NodePort

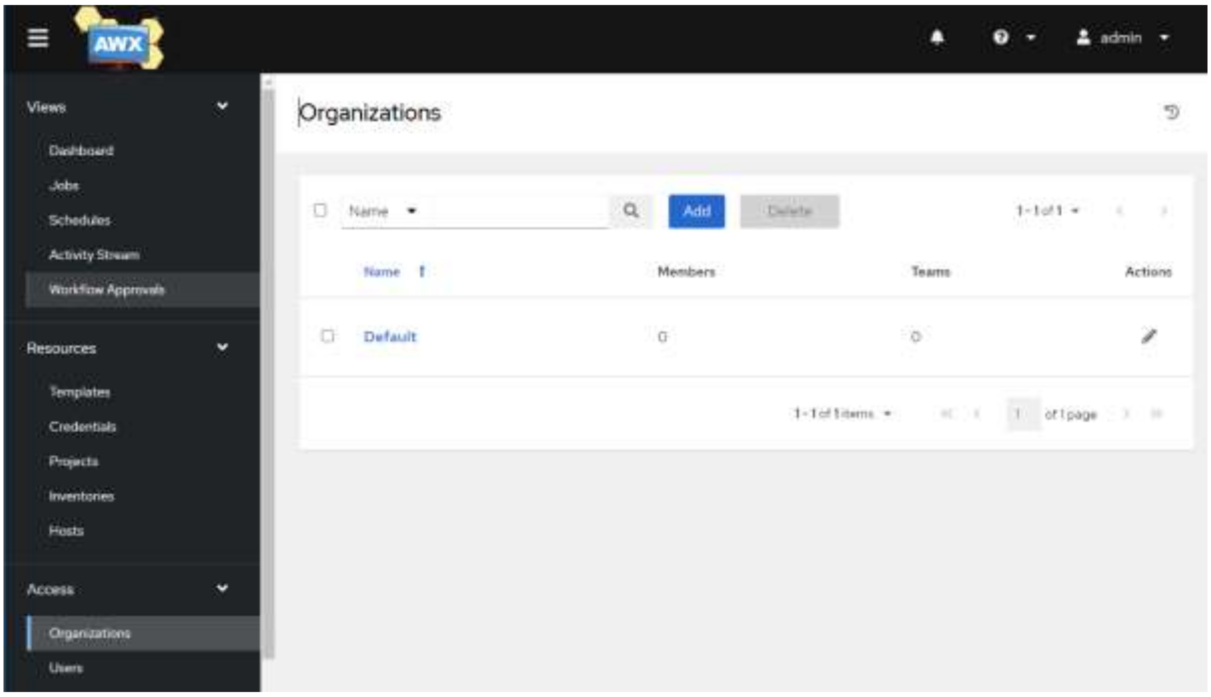


Note: Use the username as **admin** and **password** which you get from the **above step**.



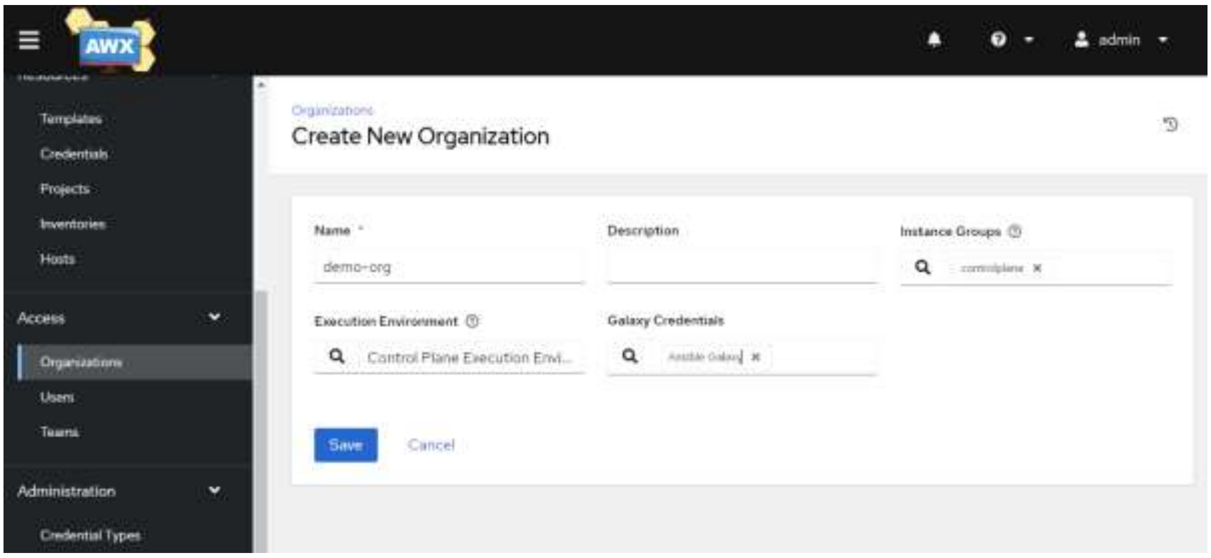
5. Creating an organization.

5.1 Select the organizations option on the left pane of the AWX and click on add new organization.



Name	Members	Teams	Actions
Default	0	0	

5.2 Add the required details and click on save.

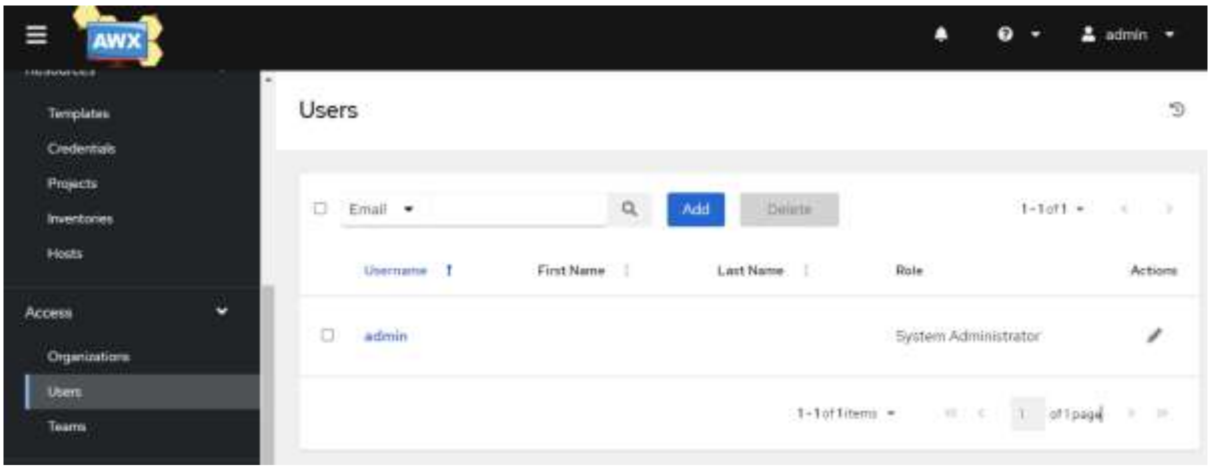


Create New Organization

Name * demo-org
Description
Instance Groups controlplane
Execution Environment Control Plane Execution Envi...
Galaxy Credentials Ansible Galaxy

5.3 Add a user to the organization so that the user can manage the org.

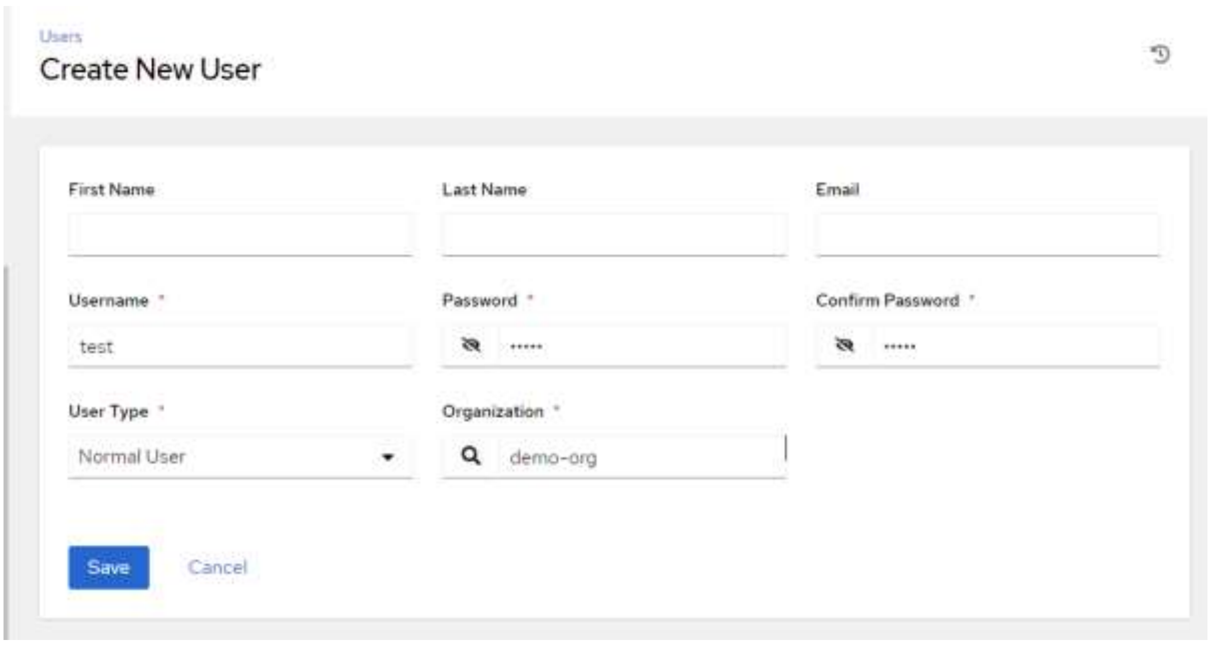
- Click on the users in the left pane and select add users.



Users

Username	First Name	Last Name	Role	Actions
admin			System Administrator	

- Fill in the required details and click on the save button select the organization which we had created earlier in this exercise.

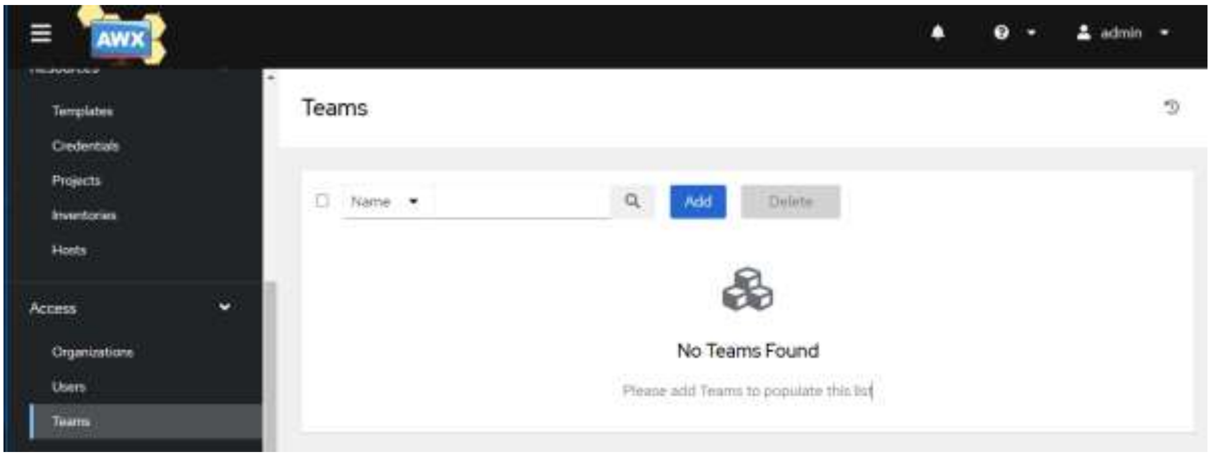

Users

Create New User

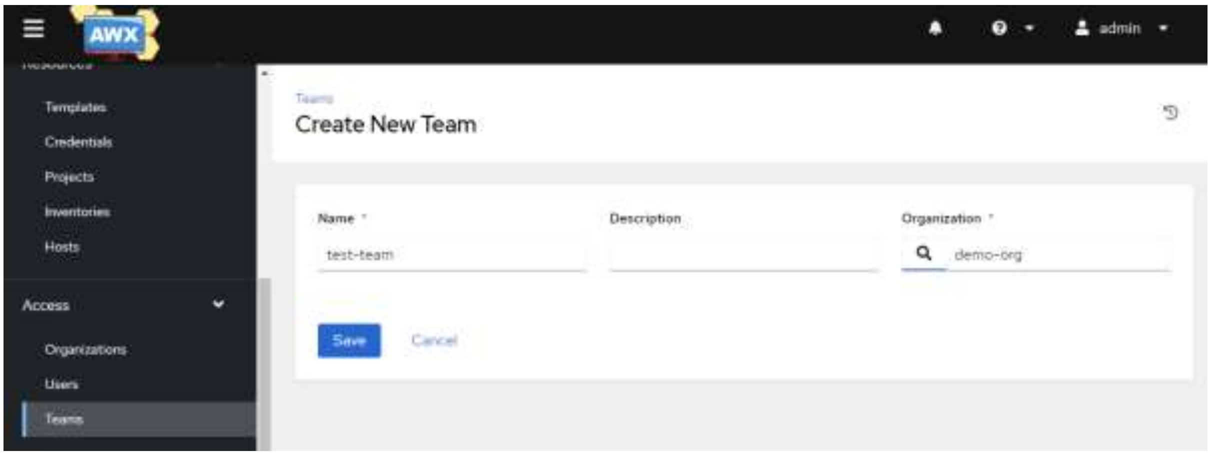
First Name	Last Name	Email
Username *	Password *	Confirm Password *
test	•••••	•••••
User Type *	Organization *	
Normal User	demo-org	

5.4 Let’s add a team for our org.

- Select the team’s option in the Access section of the left pane and click on add.



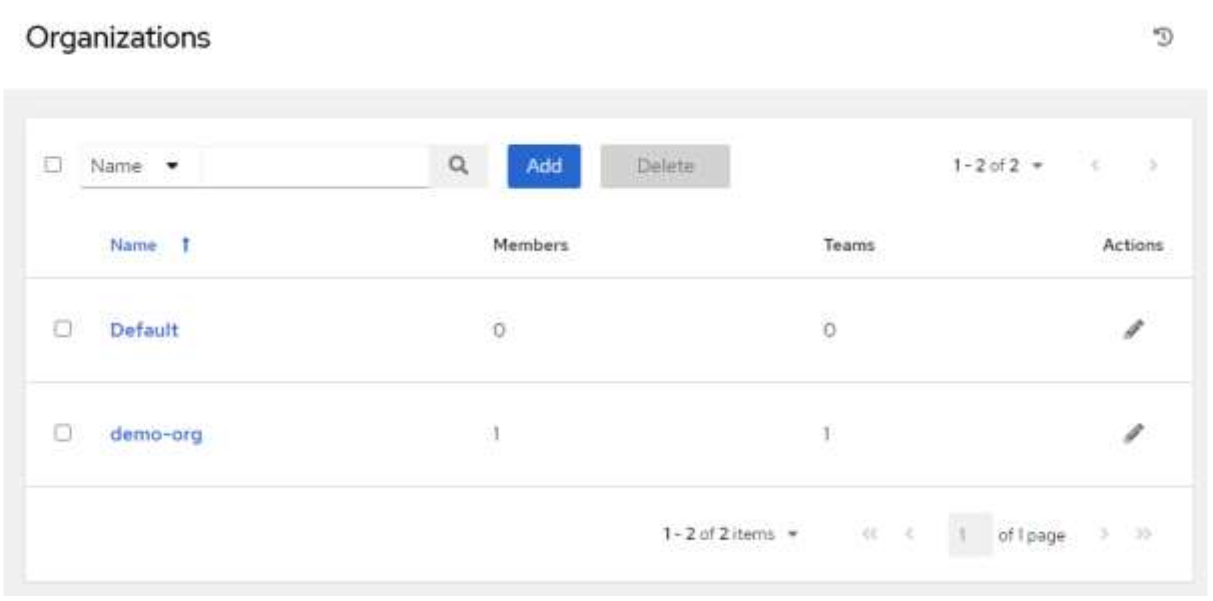
- Fill the required details and click on save.



Create New Team

Name *	Description	Organization *
test-team		demo-org

5.5 View your organization it has a new member and a team.



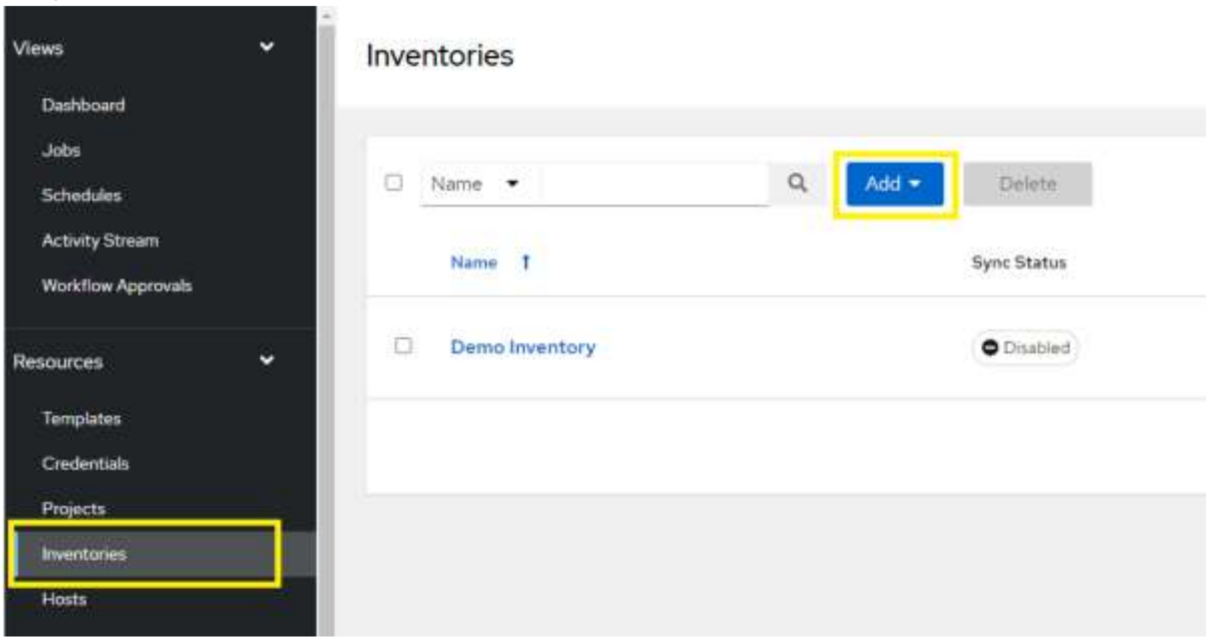
Name	Members	Teams
Default	0	0
demo-org	1	1

6. Creating a sample automation.

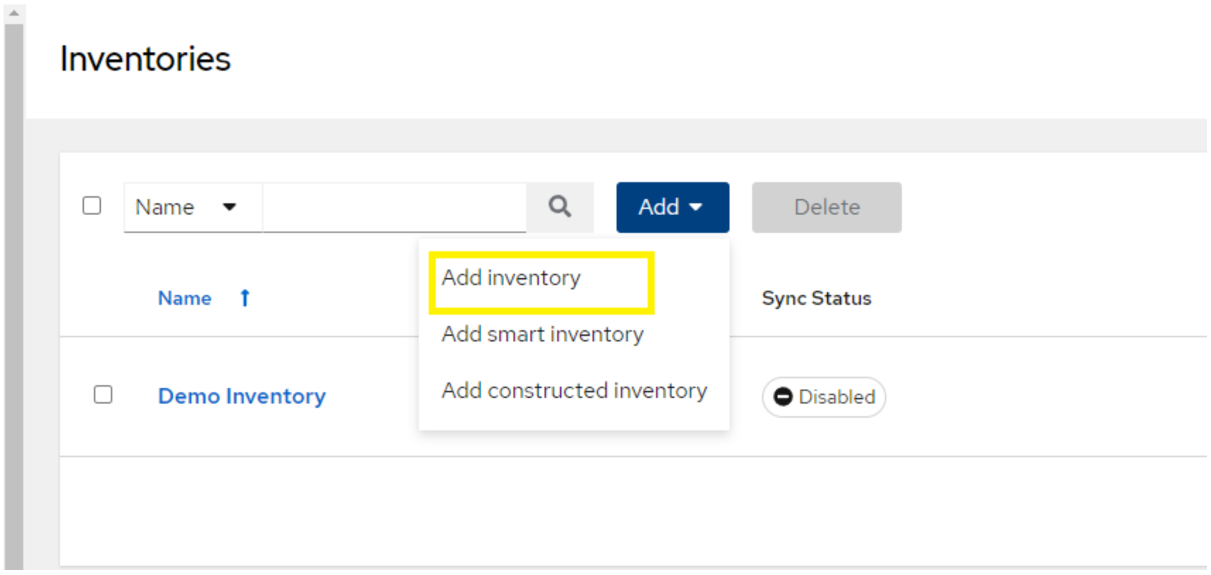
6.1 Let us add a New Inventory, Create A Host and Credential in Ansible Tower

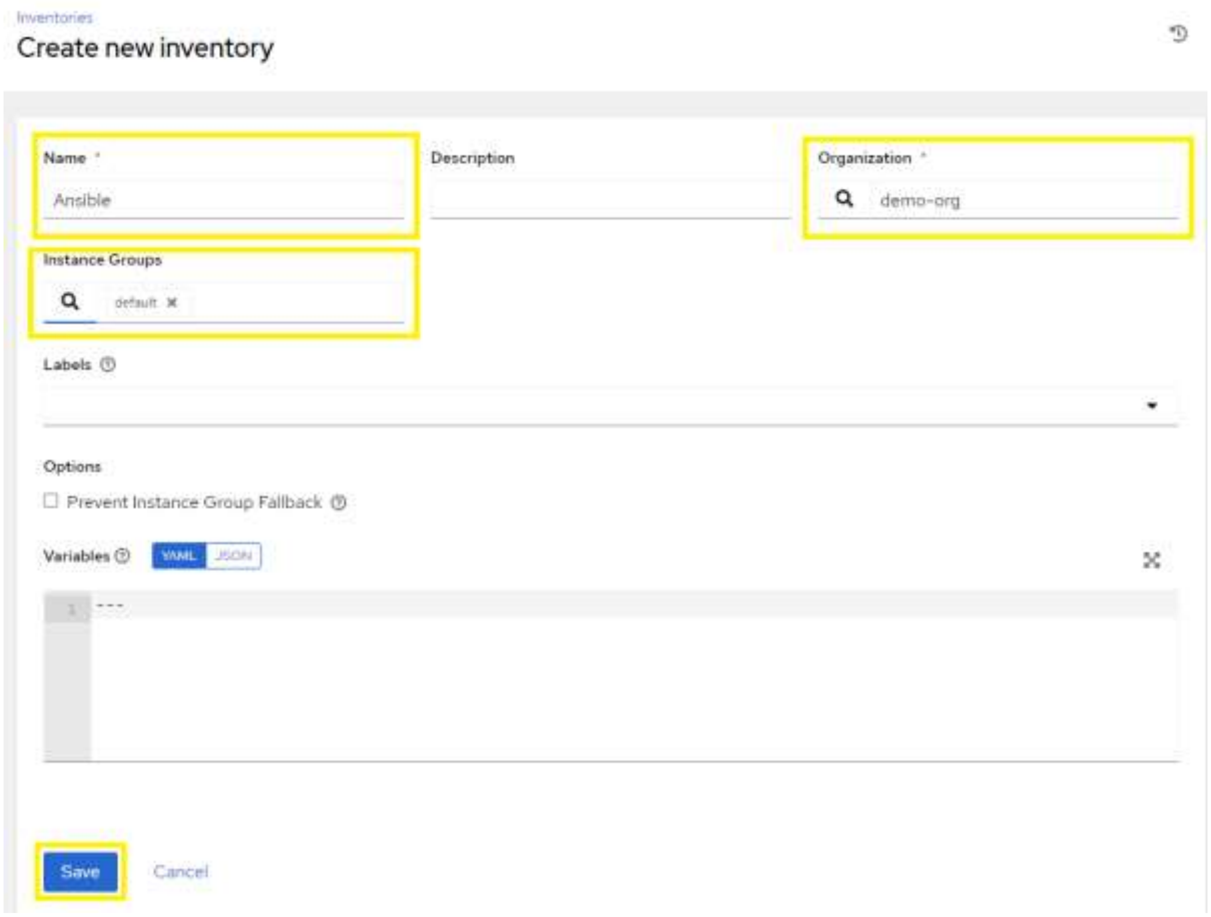
- To create a new inventory or Smart Inventory follow the below steps:
- Click the Inventories icon from the left navigation bar.
- Click the add button **+**, and select the type of inventory to create.

Output:



Name	Sync Status
Demo Inventory	Disabled




Inventories

Create new inventory

Name * Ansible

Description

Organization * demo-org

Instance Groups default

Labels

Options

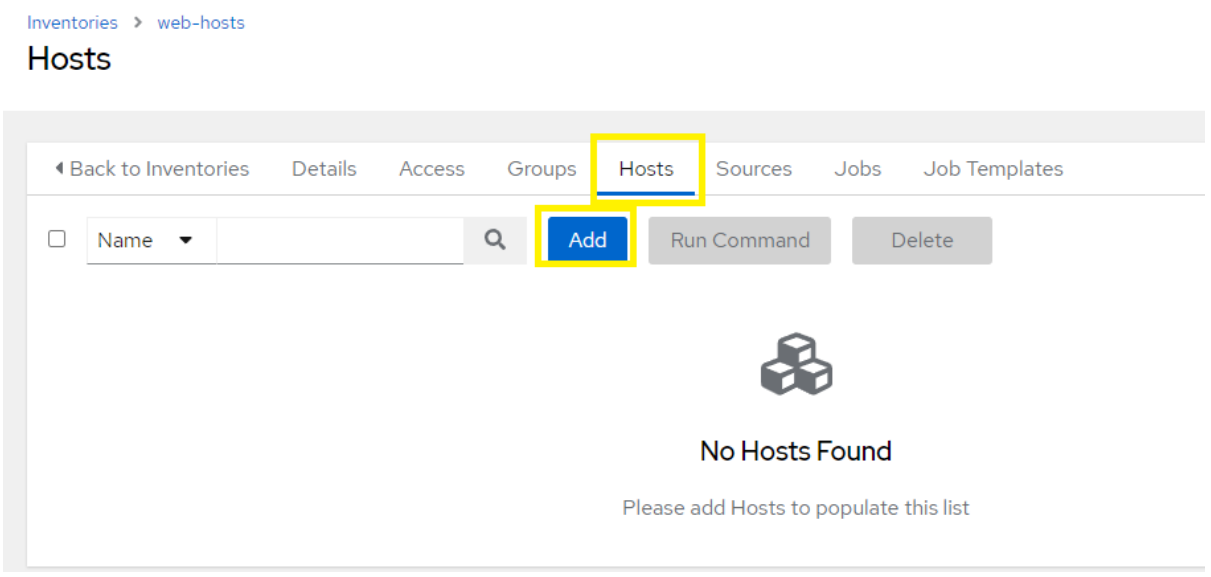
Prevent Instance Group Fallback

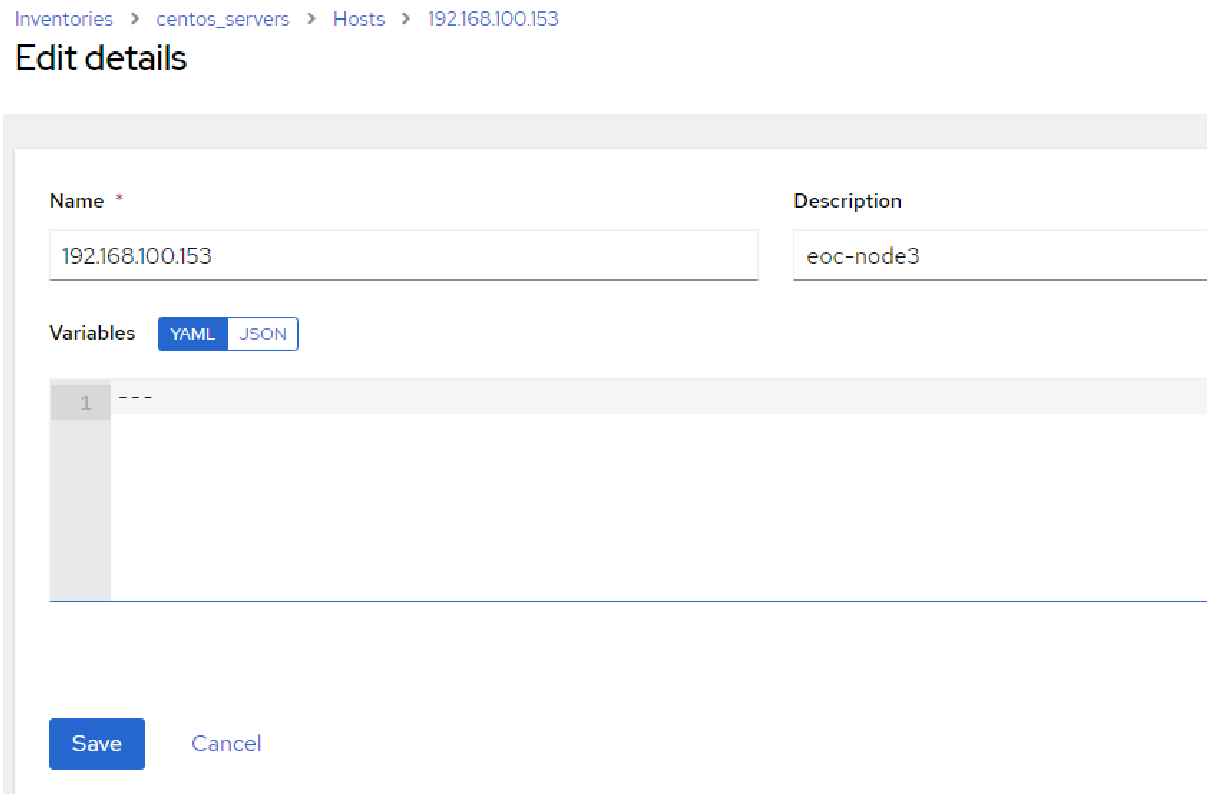
Variables YAML JSON

```
---
```

6.2 Creating a host.

- After creating a new inventory, you can proceed with configuring a new host. To create a host, follow the below steps:
- Go the Inventories tab and choose the inventory to which you want to add hosts.
- Select the Hosts tab and click on create a new host button ADD .





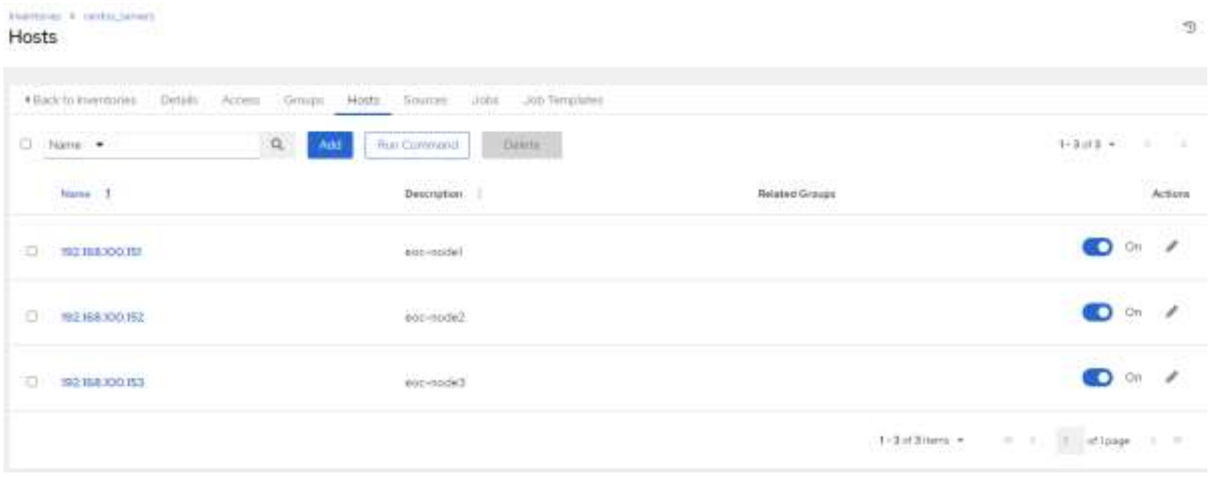
Name *: 192.168.100.153

Description: eoc-node3

Variables:

```
---
```

Note: Similarly add the other hosts also.



Inventories > centos_servers

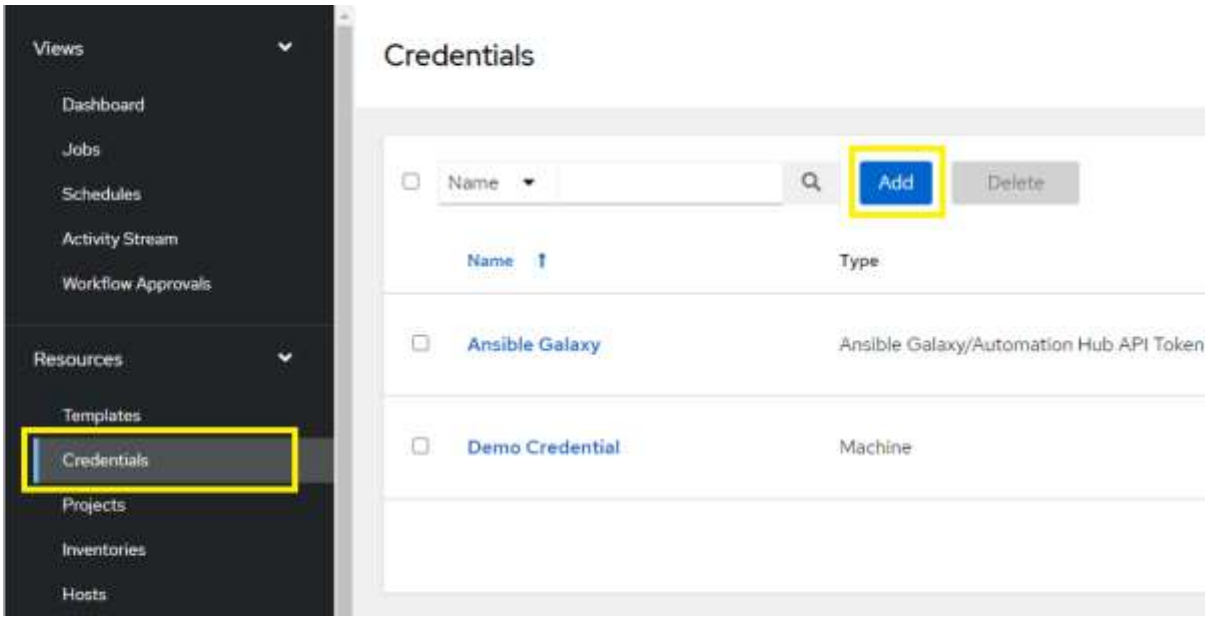
Hosts

Name	Description	Related Groups	Actions
192.168.100.151	eoc-node1		On
192.168.100.152	eoc-node2		On
192.168.100.153	eoc-node3		On

1 - 3 of 3 items

6.3 Let's create credentials.

- Credentials authenticate to launch Ansible playbooks, which can include passwords and SSH keys, against inventory hosts.
- From the left navigation bar, click the Credentials icon:
- Specify the type of credential you want to create. The credential type could be **Amazon Web Services**, **Microsoft Azure Resource Manager**, **Machine**, … In this example we will select **machine** Credential type:



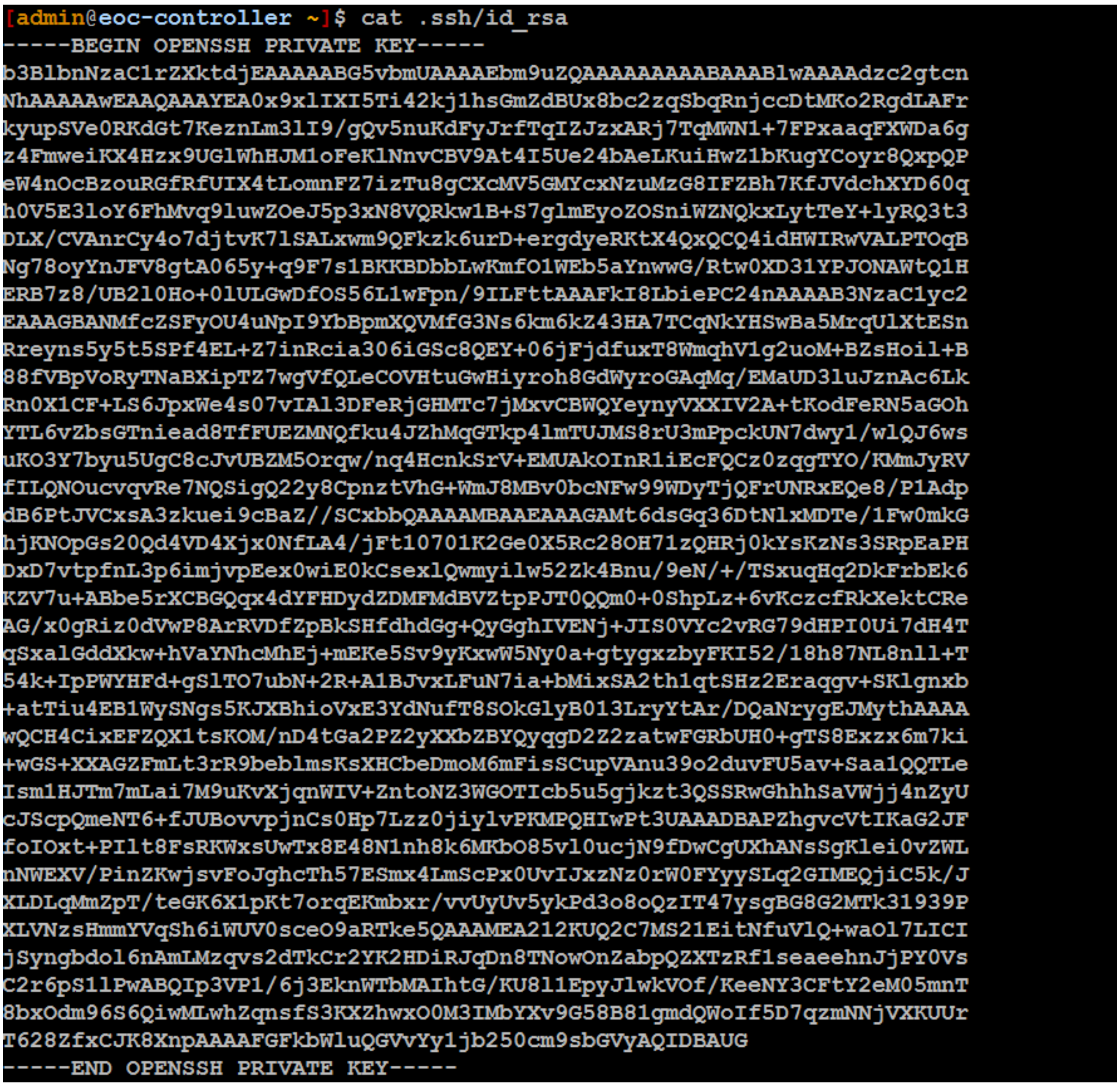
Credentials

Name	Type
Ansible Galaxy	Ansible Galaxy/Automation Hub API Token
Demo Credential	Machine

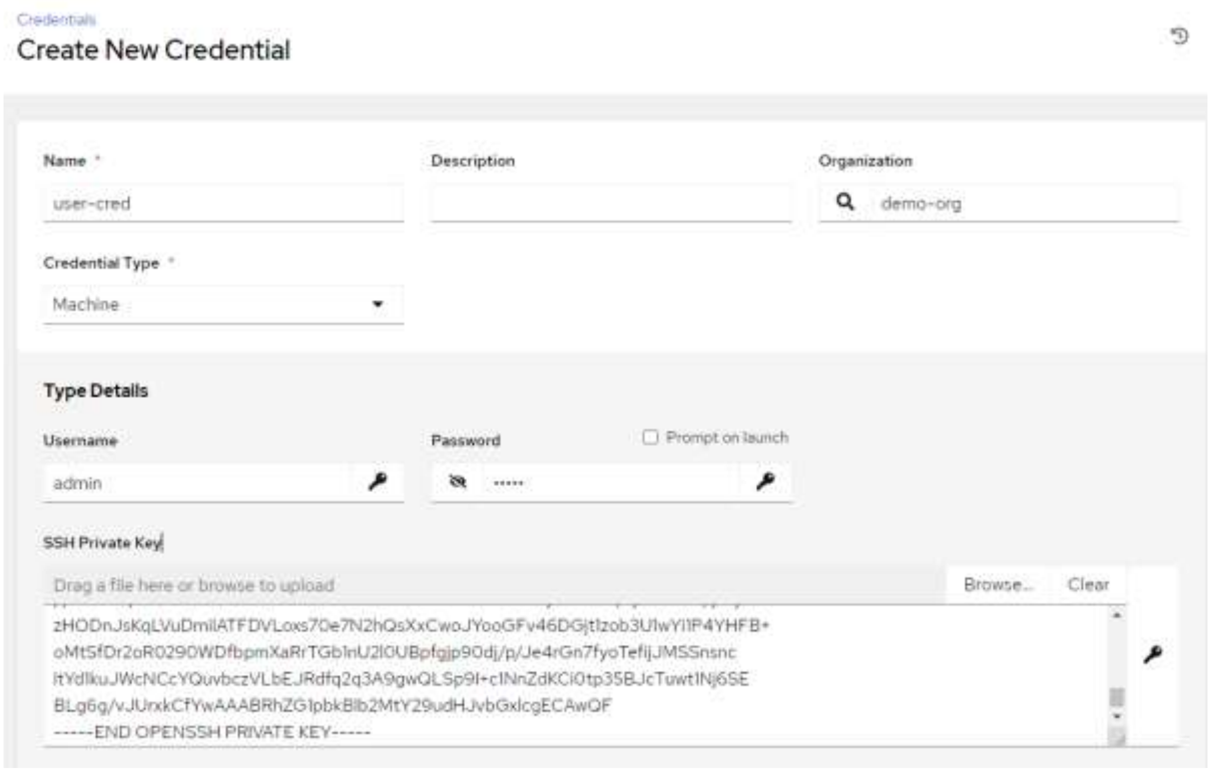
Get the ssh key file to login into all hosts.

```
# cat .ssh/id_rsa
```

Output:



```
[admin@eoc-controller ~]$ cat .ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEA0x9xlIXI5Ti42kj1hsGmZdBUx8bc2zqSbqRnjccDtMKo2RgdLAFr
kyupSVe0RKdGt7KeznLm3lI9/gQv5nuKdFyJrfTqIZJzxARj7TqMWN1+7FPxaaqFXWDa6g
z4FmweiKX4Hzx9UGlWhHJM1oFeKlNnvCBV9At4I5Ue24bAeLKuiHwZ1bKugYCoyr8QxpQP
eW4nOcBzouRGfRfUIX4tLomnFZ7izTu8gCXcMV5GMYcxNzuMzG8IFZBh7KfJVdchXYD60q
h0V5E3loY6FhMvq9luwZOeJ5p3xN8VQRkw1B+S7glmEyoZOSniWZNQkxLytTeY+lyRQ3t3
DLX/CVAnrCy4o7djtvK7lSALxwm9QFkzk6urD+ergdyeRKtX4QxQCQ4idHWIRwVALPTOqB
Ng78oyYnJFV8gtA065y+q9F7s1BKKBDbbLwKmfO1WEb5aYnwwG/Rtw0XD31YPJONAWtQ1H
ERB7z8/UB2l0Ho+0lULGwDfOS56L1wFpn/9ILFttAAAFkI8LbiePC24nAAAAB3NzaC1yc2
EAAAGBANMfcZSFyOU4uNpI9YbBpmXQVMfG3Ns6km6kZ43HA7TCqNkYHSwBa5MrqUlXtESn
Rreyns5y5t5SPf4EL+Z7inRcia306iGSc8QEY+06jFjdfuxT8WmqhV1g2uoM+BZsHoil+B
88fVBpVoRyTNaBXipTZ7wgVfQLeCOVHtuGwHiyroh8GdWyroGAqMq/EMaUD3luJznAc6Lk
Rn0X1CF+LS6JpxWe4s07vIAl3DFeRjGHMTc7jMxvCBWQYeynyVXXIV2A+tKodFeRN5aGOh
YTL6vZbsGTniead8TfFUEZMNQfku4JZhMqGTkp4lmTUJMS8rU3mPpckUN7dwy1/wlQJ6ws
uKO3Y7byu5UgC8cJvUBZM5Orqw/nq4HcnkSrV+EMUAkOInR1iEcFQCz0zqgTYO/KMmJyRV
fILQNOucvqvRe7NQSigQ22y8CpnztVhG+WmJ8MBv0bcNFw99WDyTjQFrUNRxEQe8/P1Adp
dB6PtJVCxsA3zkuei9cBaZ//SCxbbQAAAAMBAAEAAAGAMt6dsGq36DtNlxMDTe/1Fw0mkG
hjKNOpGs20Qd4VD4Xjx0NfLA4/jFt10701K2Ge0X5Rc28OH71zQHRj0kYsKzNs3SRpEaPH
DxD7vtpfnL3p6imjvpEex0wiE0kCsexlQwmyilw52Zk4Bnu/9eN/+/TSxuqHq2DkFrbEk6
KZV7u+ABbe5rXCBGQqx4dYFHDydZDMFMdBVZtpPJT0QQm0+0ShpLz+6vKczcfRkXektCRe
AG/x0gRiz0dVwP8ArRVDfZpBkSHfdhdGg+QyGghIVENj+JIS0VYc2vRG79dHPI0Ui7dH4T
qSxalGddXkw+hVaYNhcMhEj+mEKe5Sv9yKxwW5Ny0a+gtygxzbyFKI52/18h87NL8nll+T
54k+IpPWYHFd+gSlTO7ubN+2R+A1BJvxLFuN7ia+bMixSA2th1qtSHz2Eraqgv+SKlgnxb
+atTiu4EB1WySNgs5KJXBhioVxE3YdNufT8SOkGlyB013LryYtAr/DQaNrygEJMythAAAA
wQCH4CixEFZQX1tsKOM/nD4tGa2PZ2yXXbZBYQyqgD2Z2zatwFGRbUH0+gTS8Exzx6m7ki
+wGS+XXAGZFmLt3rR9beblmsKsXHCbeDmoM6mFisSCupVAnu39o2duvFU5av+Saa1QQTLe
Ism1HJTm7mLai7M9uKvXjqnWIV+ZntoNZ3WGOTIcb5u5gjkzt3QSSRwGhhhSaVWjj4nZyU
cJScpQmeNT6+fJUBovvpjnCs0Hp7Lzz0jiylvPKMPQHIwPt3UAAADBAPZhgvcVtIKaG2JF
foIOxt+PIlt8FsRKWxsUwTx8E48N1nh8k6MKbO85vl0ucjN9fDwCgUXhANsSgKlei0vZWL
nNWEXV/PinZKwjsvFoJghcTh57ESmx4LmScPx0UvIJxzNz0rW0FYyySLq2GIMEQjiC5k/J
XLDLqMmZpT/teGK6X1pKt7orqEKmbxr/vvUyUv5ykPd3o8oQzIT47ysgBG8G2MTk31939P
XLVNzsHmmYVqSh6iWUV0sceO9aRTke5QAAAMEA212KUQ2C7MS21EitNfuVlQ+waOl7LICI
jSyngbdol6nAmLMzqvs2dTkCr2YK2HDiRJqDn8TNowOnZabpQZXTzRf1seaeehnJjPY0Vs
C2r6pS1lPwABQIp3VP1/6j3EknWTbMAIhtG/KU8l1EpyJlwkVOf/KeeNY3CFtY2eM05mnT
8bxOdm96S6QiwMLwhZqnsfS3KXZhwxO0M3IMbYXv9G58B81gmdQWoIf5D7qzmNNjVXKUUr
T628ZfxCJK8XnpAAAAFGFkbWluQGVvYy1jb250cm9sbGVyAQIDBAUG
-----END OPENSSH PRIVATE KEY-----
```



Credentials

Create New Credential

Name *	Description	Organization
user-cred		demo-org
Credential Type *		
Machine		

Type Details

Username	admin
Password	•••••

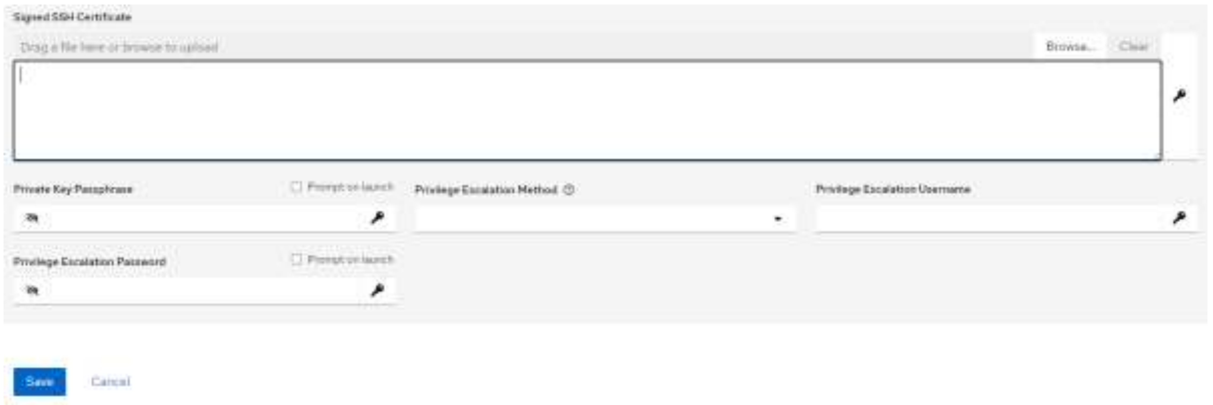
☐ Prompt on launch

SSH Private Key

Drag a file here or browse to upload

```
zHODnJsKqLVuDmilATFDVLoxs70e7N2hQsXxCwoJYooGFv46DGjtIzob3UlwYI1P4YHFB+
oMtSfDr2oR0290WDfbpmXaRrTGb1nU2l0UBpfgjp90dj/p/Je4rGn7fyoTefijJMSSnsnc
ltYdlkuJWcNCcYQuvbczVLbEJRdfq2q3A9gwQLSp9I+cINnZdKCi0tp35BJcTuwtINj6SE
BLg6g/vJUrxkCfYwAAABRhZG1pbkBlb2MtY29udHJvbGxlcgECAwQF
-----END OPENSSH PRIVATE KEY-----
```

Note: copy all the content of the ssh id and paste it in the ssh private key section.



Signed SSH Certificate

Drag a file here or browse to upload

Private Key Passphrase ☐ Prompt on launch

Privilege Escalation Method

Privilege Escalation Username

Privilege Escalation Password ☐ Prompt on launch

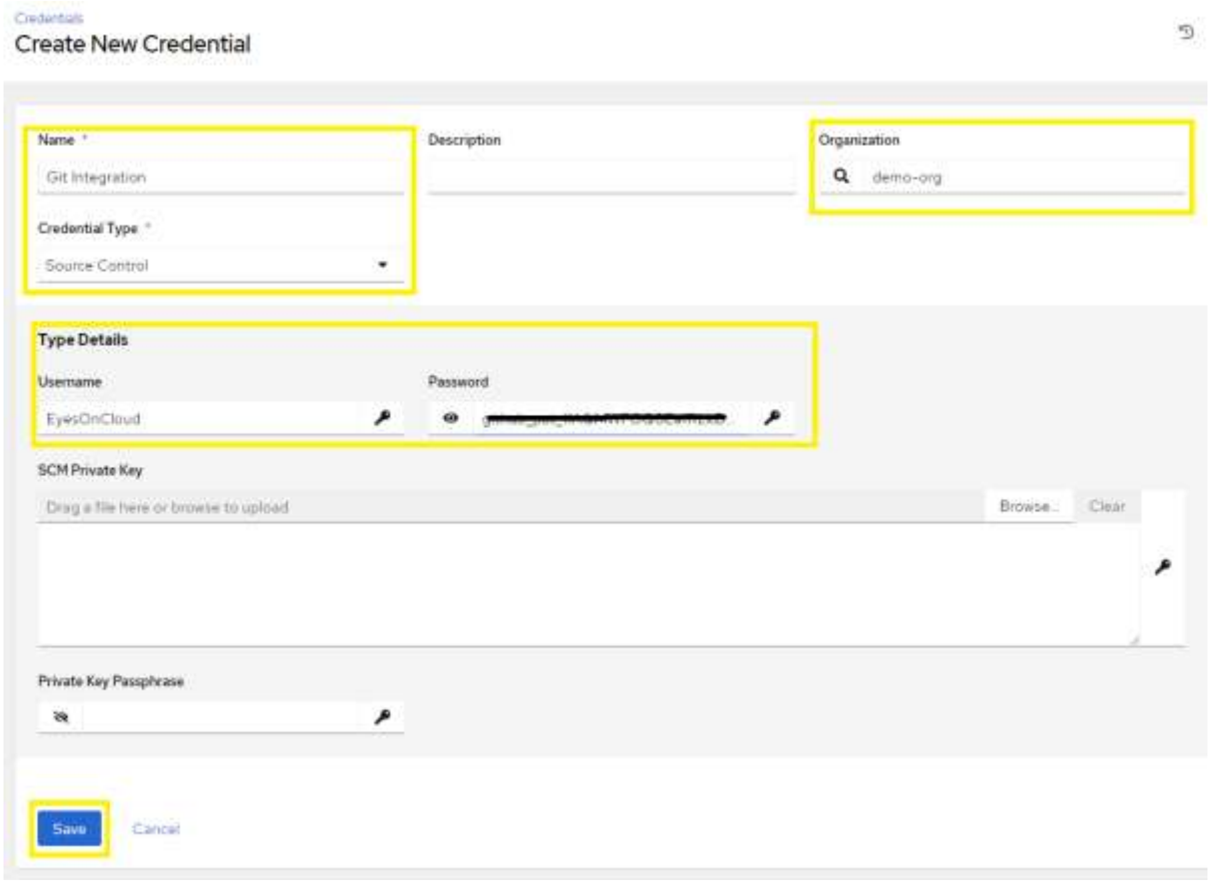
Note: Select save option

6.4 Let's Create a Credential for SCM (source content manager)

Step1: Creating a personal access token to create click the below link

https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

Let's add git hub credentials to integrate git.



Name *
Git Integration

Credential Type *
Source Control

Description

Organization
demo-org

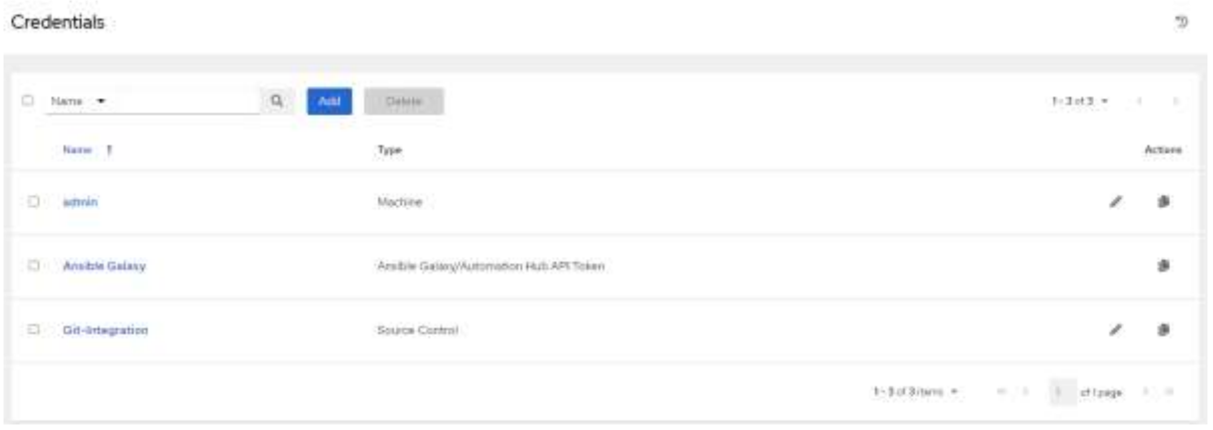
Type Details

Username
EyesOnCloud

Password
[redacted]

SCM Private Key
Drag a file here or browse to upload

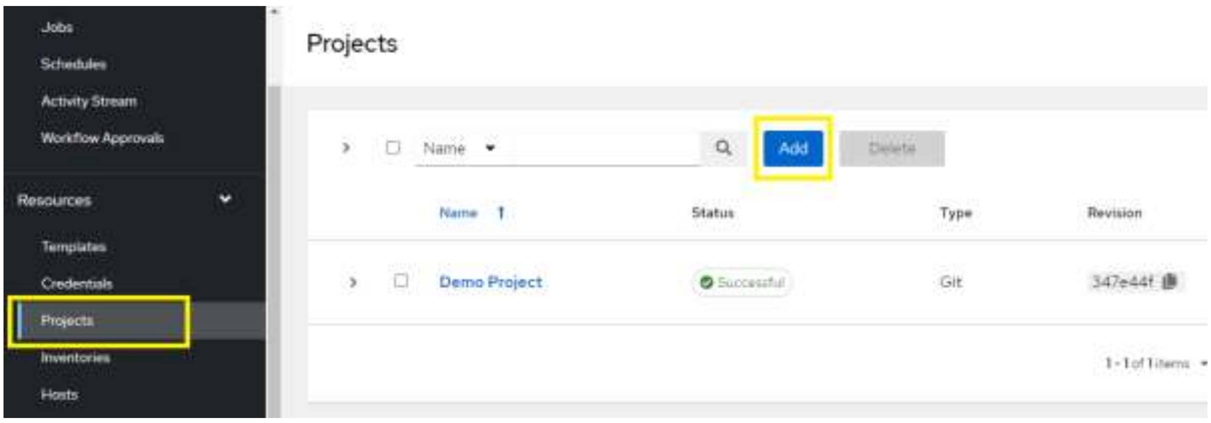
Private Key Passphrase



Name	Type	Actions
admin	Machine	
Ansible Galaxy	Ansible Galaxy/Automation Hub API Token	
Git-Integration	Source Control	

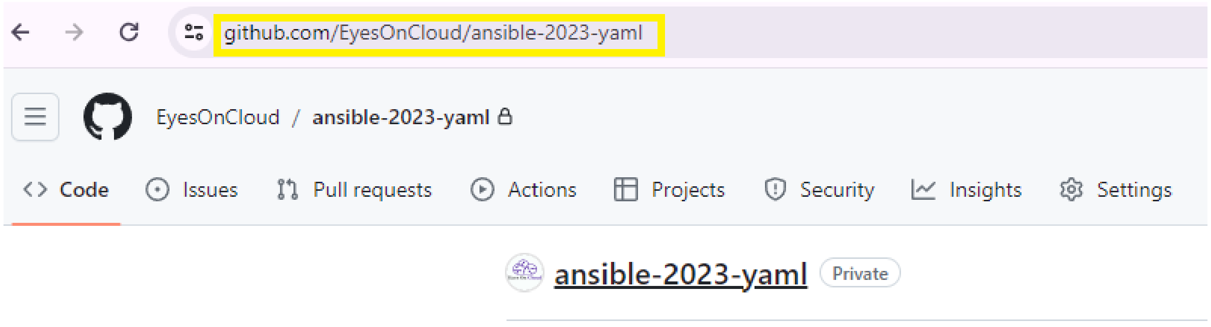
6.5 Creating a GitHub/GitLab SCM Project

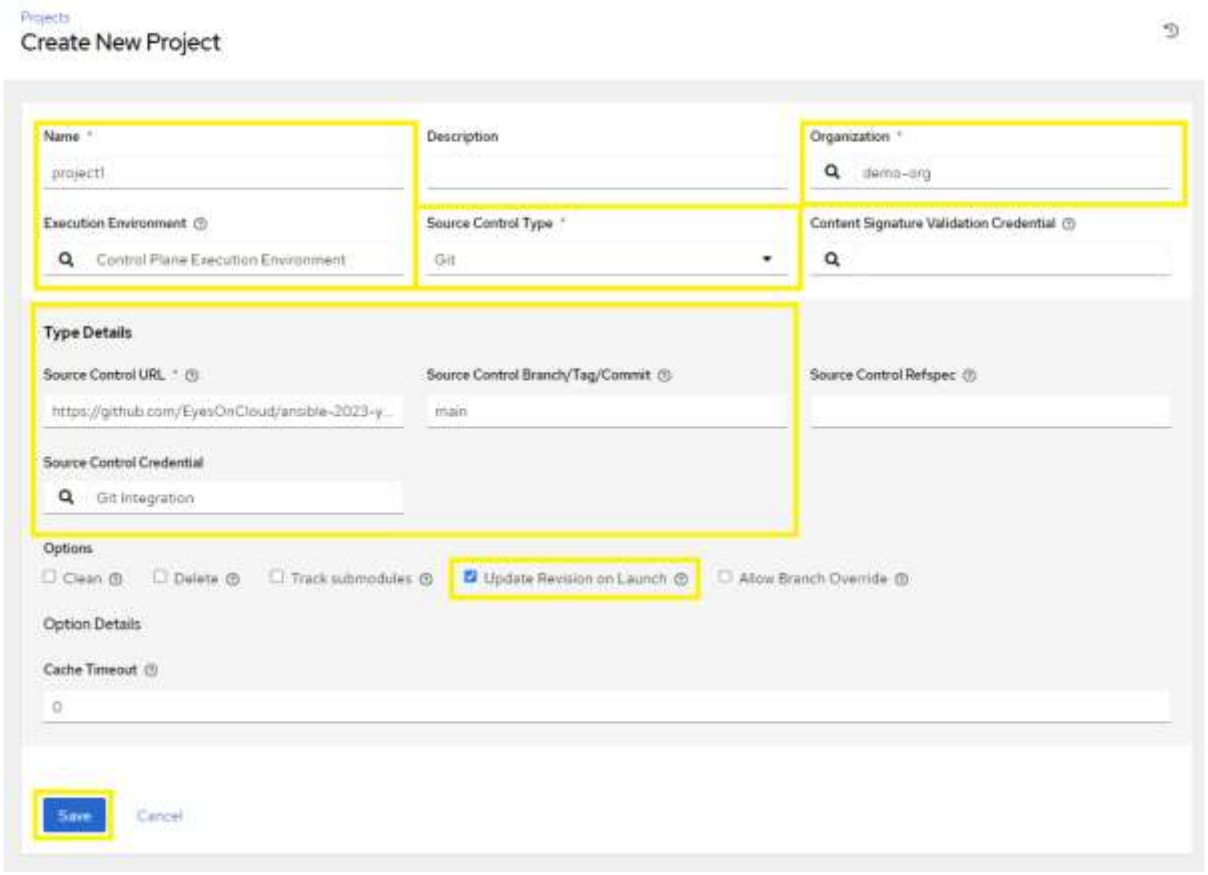
- Click the projects icon from the left navigation bar.
- Click the add button + to create a new project.
- Fill in all the necessary details such as Name, Description and Organization of Project. After that, select Git from the SCM type.



Name	Status	Type	Revision
Demo Project	Successful	Git	347e44f

1 - 1 of 1 items





Name *: project1
Description:
Organization *: demo-org
Execution Environment: Control Plane Execution Environment
Source Control Type *: Git
Content Signature Validation Credential:

Type Details
Source Control URL *: https://github.com/EyesOnCloud/ansible-2023-y...
Source Control Branch/Tag/Commit: main
Source Control Refspec:
Source Control Credential: Git Integration

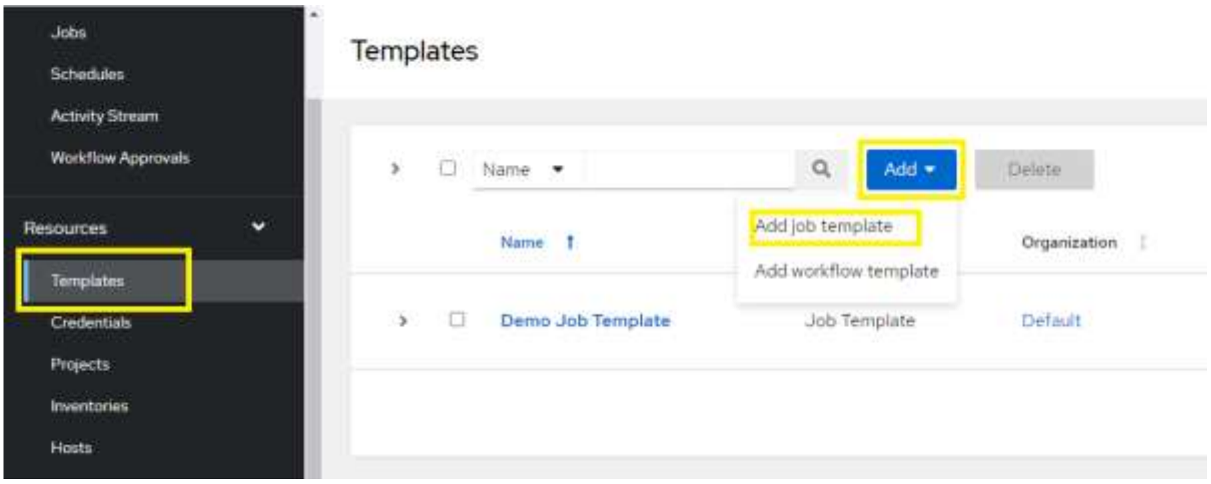
Options
☐ Clean ☐ Delete ☐ Track submodules ☑ Update Revision on Launch ☐ Allow Branch Override

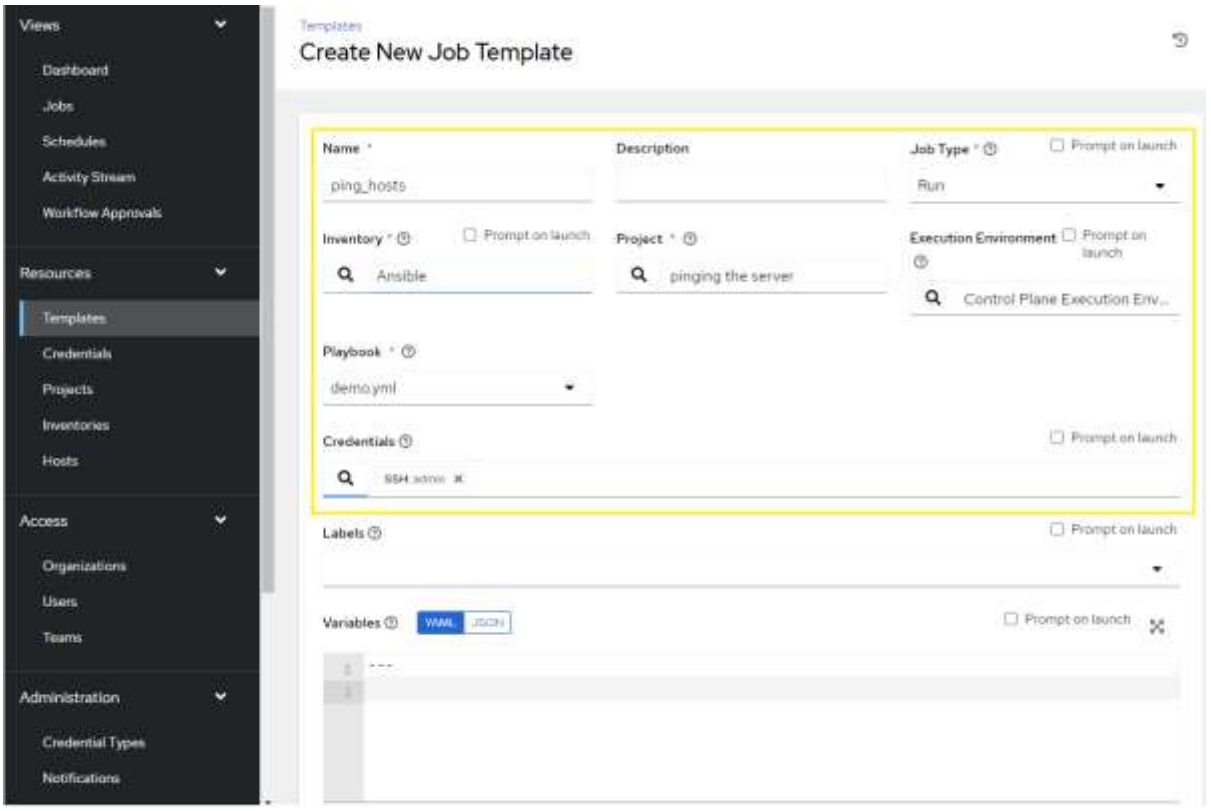
Option Details
Cache Timeout: 0

6.6 Create a new template.

- Go to templates page and select add.
- Enter the appropriate details into the following fields:

- Name: Enter a name for the job.
- Description: Enter an arbitrary description as appropriate (optional).
- Job Type: Could be Run or Check
- Inventory: Choose the inventory to be used with this job template from the inventories list.
- Project: Choose the project to be used with this job template from the projects list.
- 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 the selected project.
- Credential: Click the search button to open a separate window. Choose the credential from the available options to be used with this job template.

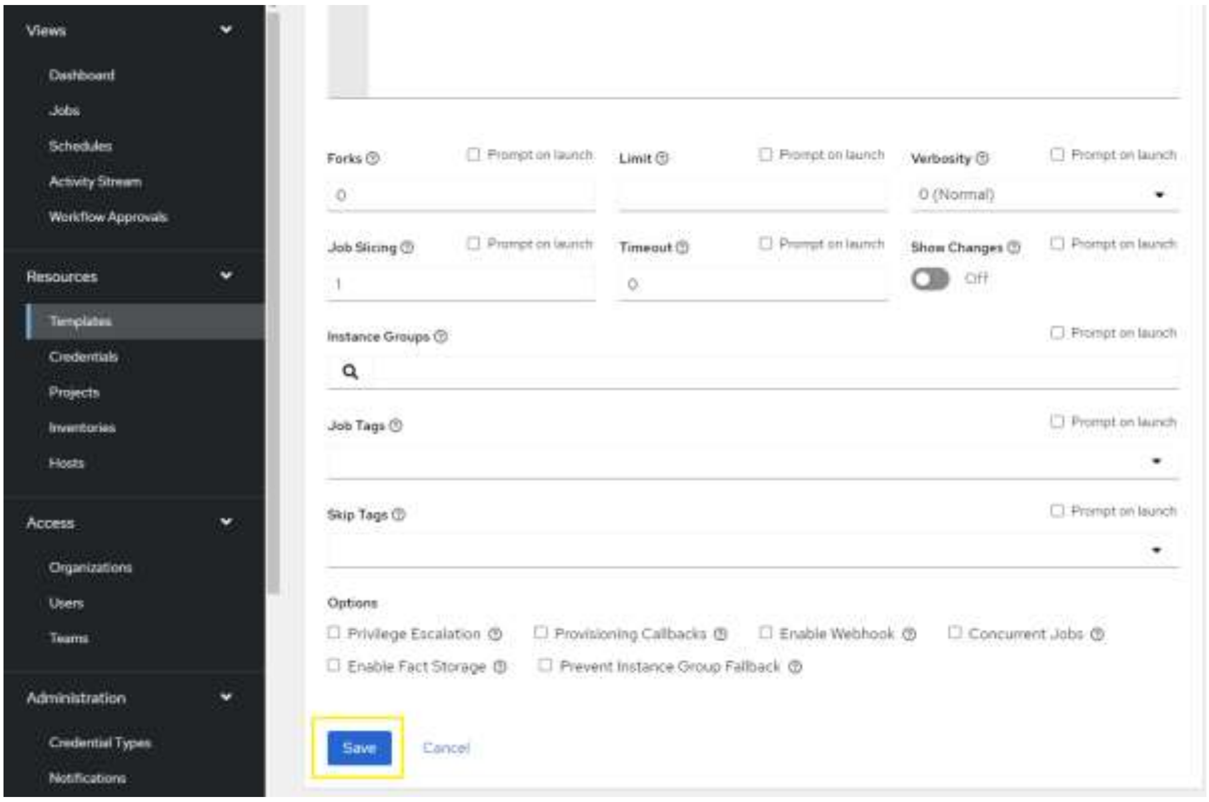




Create New Job Template

Name *: ping_hosts
Description:
Job Type *: Run
Inventory *: Ansible
Project *: pinging the server
Execution Environment: Control Plane Execution Env...
Playbook *: demo.yml
Credentials: SSH admin
Labels:
Variables: YAML JSON

```
---
```

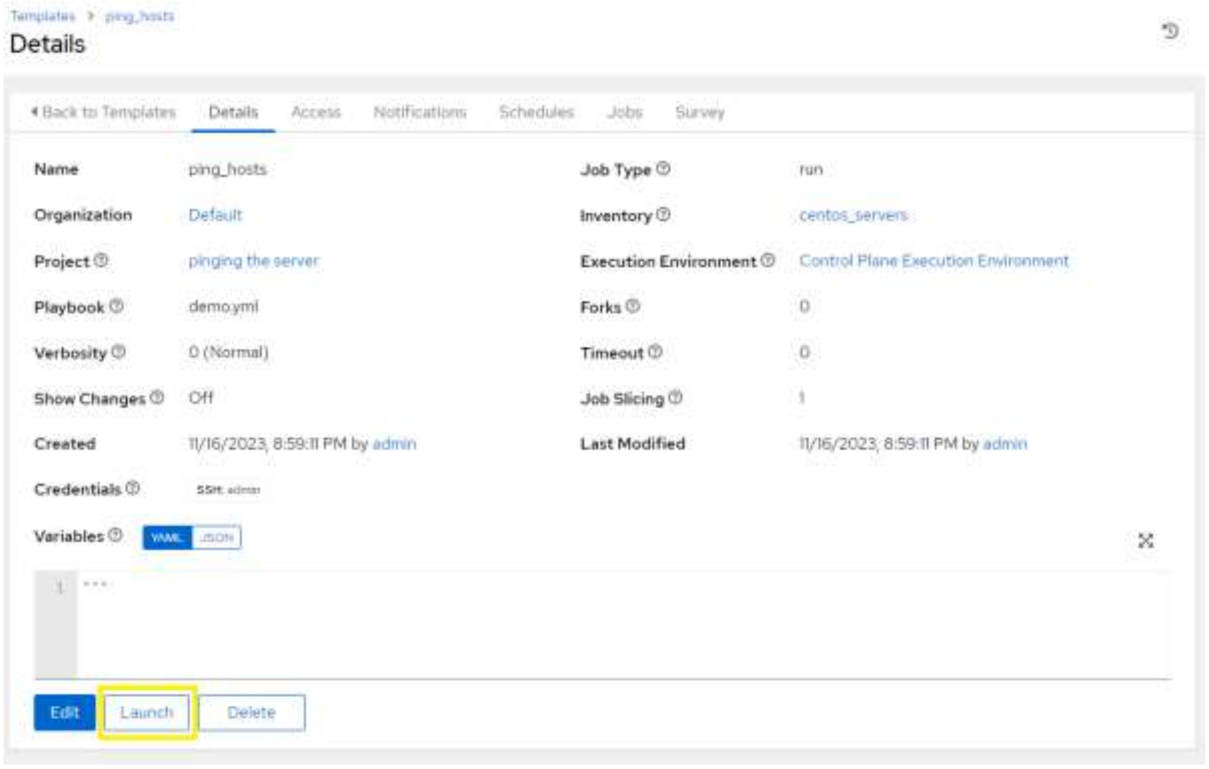


Forks: 0 ☐ Prompt on launch
Limit: ☐ Prompt on launch
Verbosity: 0 (Normal) ☐ Prompt on launch
Job Slicing: 1 ☐ Prompt on launch
Timeout: 0 ☐ Prompt on launch
Show Changes: Off ☐ Prompt on launch
Instance Groups ☐ Prompt on launch
Job Tags ☐ Prompt on launch
Skip Tags ☐ Prompt on launch

Options
☐ Privilege Escalation ☐ Provisioning Callbacks ☐ Enable Webhook ☐ Concurrent Jobs
☐ Enable Fact Storage ☐ Prevent Instance Group Fallback

Note: save the details.

6.7 Launch the job.

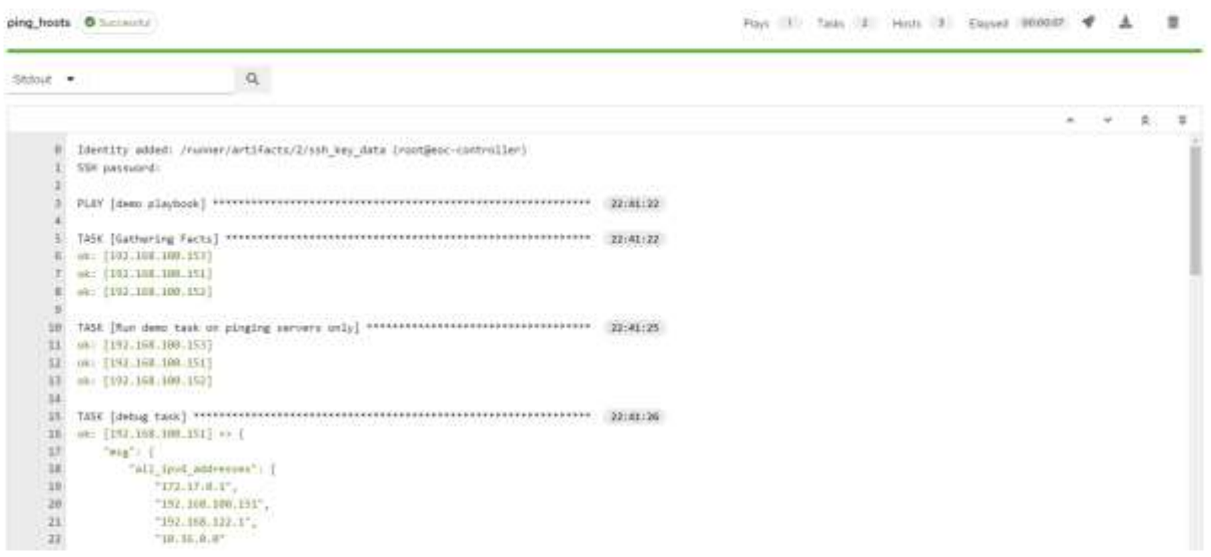


Name	ping_hosts	Job Type	run
Organization	Default	Inventory	centos_servers
Project	pinging the server	Execution Environment	Control Plane Execution Environment
Playbook	demo.yml	Forks	0
Verbosity	0 (Normal)	Timeout	0
Show Changes	Off	Job Slicing	1
Created	11/16/2023, 8:59:11 PM by admin	Last Modified	11/16/2023, 8:59:11 PM by admin
Credentials	SSH: admin		

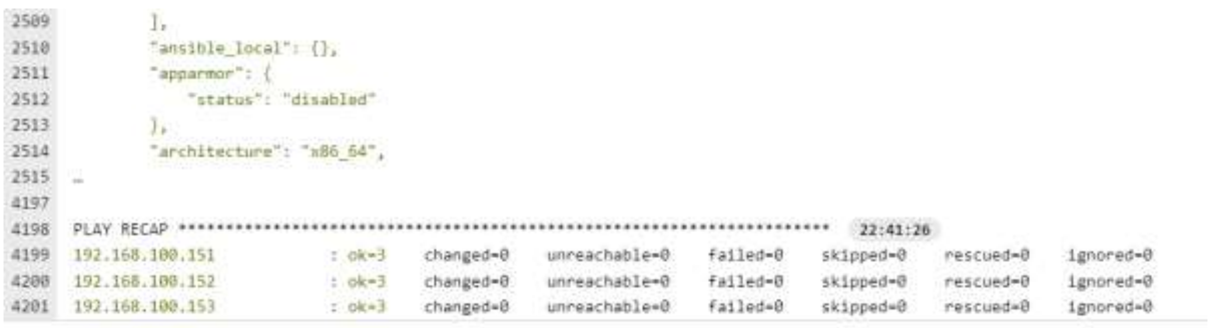
Variables

```
---
```

6.8 Let’s check the output.



```
Identity added: /runner/artifacts/2/ssh_key_data (root@eoc-controller)
SSH password:

PLAY [demo playbook] ************************************************************  22:41:22

TASK [Gathering Facts] **********************************************************  22:41:22
ok: [192.168.100.153]
ok: [192.168.100.151]
ok: [192.168.100.152]

TASK [Run demo task on pinging servers only] ************************************  22:41:25
ok: [192.168.100.153]
ok: [192.168.100.151]
ok: [192.168.100.152]

TASK [debug task] ***************************************************************  22:41:26
ok: [192.168.100.151] => {
    "msg": {
        "all_ipv4_addresses": [
            "172.17.0.1",
            "192.168.100.151",
            "192.168.122.1",
            "10.16.0.0"
```



```
            ],
            "ansible_local": {},
            "apparmor": {
                "status": "disabled"
            },
            "architecture": "x86_64",
...

PLAY RECAP **********************************************************************  22:41:26
192.168.100.151            : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.100.152            : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.100.153            : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```