### Ansible-Automation-Platform操作手册

**前言：**

1.蓝色字体表示命令行命令，正式执行时不要复制前面的#号，#号只是提示应该使用root权限操作

2.绿色字体表示注释，有时注释太多就不用绿色表示了

3.注意：本文档的所有操作请先在测环境进行实践，请不要直接在真实的服务器中操作！

**版权声明**：

本文档以开源的形式发布，所有条款如下：

（1）无担保：作者不保证文档内容的准确无误，亦不承担由于使用此文档所导致的任何后果

（2）自由使用：任何人可以出于任何目的而自由地 阅读/链接/打印/转载/引用/分发/再创作 此文档，无需任何附加条件

若您 阅读/链接/打印/转载/引用/分发/再创作 本文档，则说明接受以上2个条款。

作者：李茂福

更新日期：2024-05-16

**★第1章、安装红帽AAP自动化平台**

RedHat最早开发了Ansible Tower的自动化管理平台，在Ansible Tower 3.8及之前的版本中，执行环境与运行ansible tower的系统紧密耦合，难以管理运行ansible playbook所需的各种模块的依赖，对执行环境的更改必须直接在ansible tower系统上完成，如果不同的playbook需要不同的依赖环境，则必须手动设置和更新Python虚拟环境。后来开发了**AAP**（Ansible Automation Platform）这个自动化控制平台，AAP改进了这一架构，不再直接使用系统的环境，而是使用自动化执行环境（基于容器的一个ansible及python环境）

RedHat的Ansible Tower及Ansible Automation Platform是收费的，社区有开源免费的对应产品：Ansible-AWX，详情请参阅作者其他文档。

官方安装指导：

https://access.redhat.com/documentation/zh-cn/red\_hat\_ansible\_automation\_platform/2.2/html-single/red\_hat\_ansible\_automation\_platform\_installation\_guide/index

**★安装前准备**

**AutomationController自动化控制器配置要求：**

至少要 4核cpu，16GB内存，40GB剩余磁盘空间（20GB给/var/lib/awx存储目录）系统为rhel>=8.4

**AutomationHub自动化中心配置要求：**

至少要 2核cpu，8GB内存，60GB剩余磁盘空间（20GB专用硬盘空间）系统为rhel>=8.4

①设置主机名称

# hostnamectl set-hostname aap2.cof-lee.com

②配置ntp

# yum install chrony -y

# systemctl enable chronyd

# systemctl start chronyd

# cat > /etc/chrony.conf <<EOF

server 10.99.1.1 iburst prefer

driftfile /var/lib/chrony/drift

makestep 1.0 3

rtcsync

keyfile /etc/chrony.keys

leapsectz right/UTC

logdir /var/log/chrony

EOF

# timedatectl set-timezone Asia/Shanghai #设置时区

# systemctl restart chronyd

# chronyc sources -v

③设置ssh免密

# ssh-keygen -t rsa -b 2048 -C "root@aap2.cof-lee.com" -P "" -f ~/.ssh/id\_rsa

# ssh-copy-id aap2.cof-lee.com

**★注册系统并配置AAP订阅**

首先到红帽官网注册账号，并成为开发者账号

https://access.redhat.com/ #注册账号

https://developers.redhat.com/ #登录成为开发者账号

然后使用红帽订阅管理工具注册系统

# subscription-manager register

Registering to: subscription.rhsm.redhat.com:443/subscription

Username: Limaofu #红帽官网账号

Password: #此账号对应的密码

The system has been registered with ID: 18c94d17-a47e-4d14-8589-2e50c1646642

The registered system name is: aap2.cof-lee.com

注册系统后，可到官网查看系统注册情况： https://access.redhat.com/management/systems

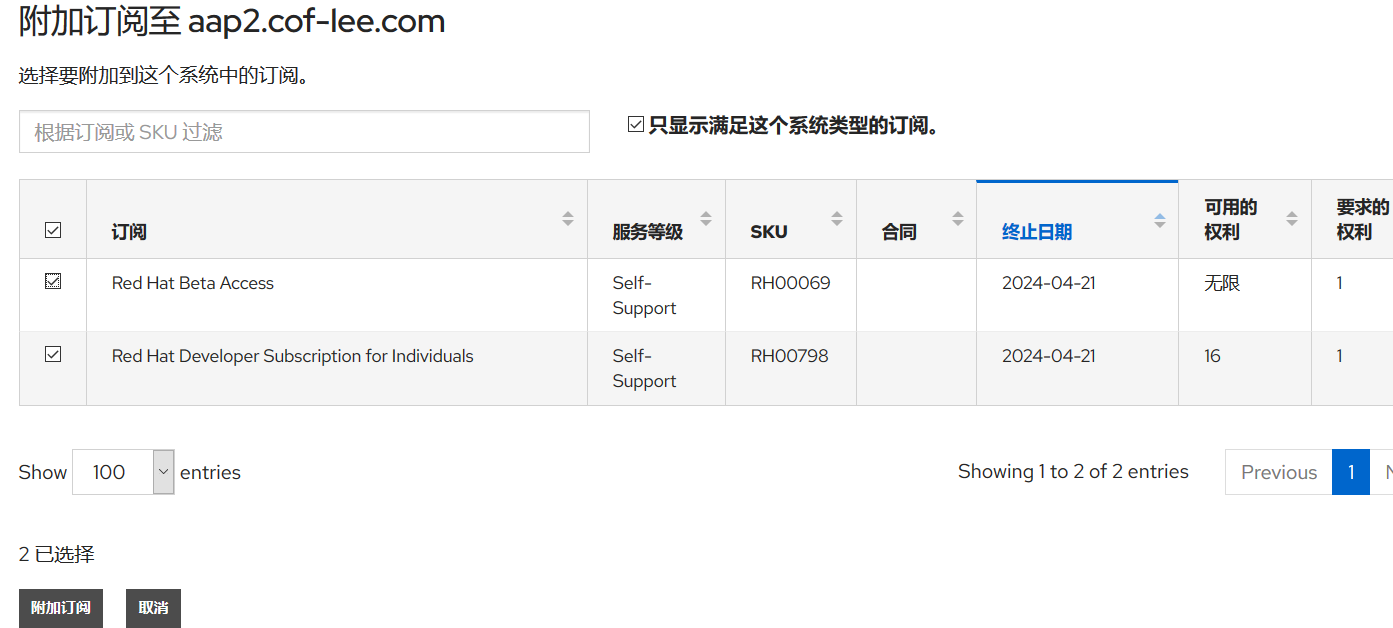
（现在已移到https://console.redhat.com/subscriptions/usage/rhel界面）



点击系统名称“aap2.cof-lee.com”，查看详情



点击“订阅”→“附加订阅”



把“Red Hat Developer Subscription for Individuals”附加到此服务器

启用红帽Ansible自动化平台2 的软件仓库

# subscription-manager list --available --all | grep -i "Ansible Automation Platform" #查看订阅中是否有AAP

Red Hat Ansible Automation Platform

#查看订阅的pool\_id

# subscription-manager list --available --all | grep -i "Ansible Automation Platform" -B6

--

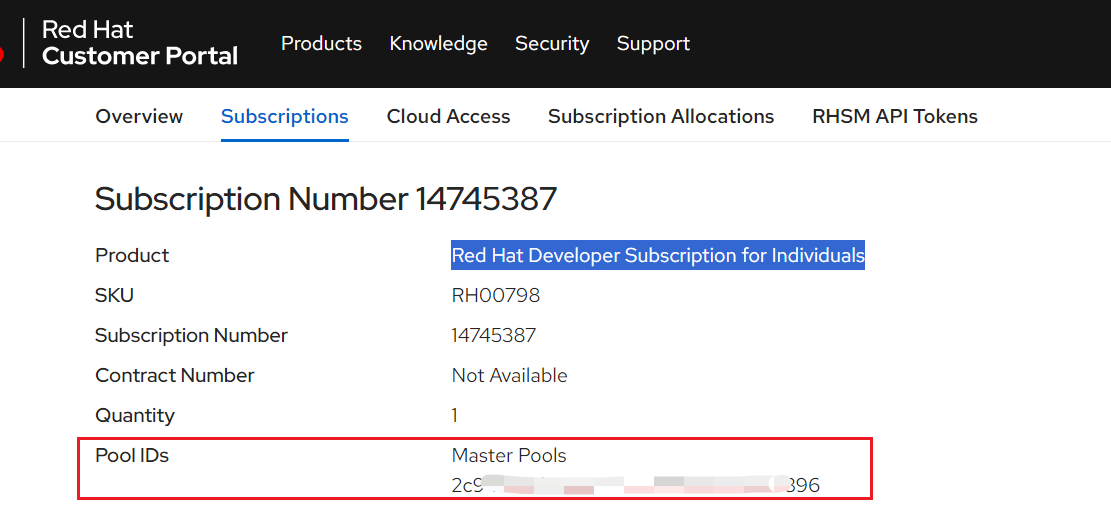
Subscription Name: Red Hat Developer Subscription for Individuals #确定是这个订阅下有AAP的授权

Provides: Red Hat Enterprise Linux Fast Datapath

Red Hat OpenShift Enterprise JBoss EAP add-on Beta

Red Hat Ansible Automation Platform

#然后去web界面查看此订阅的pool id



# subscription-manager attach --pool=<pool\_id> #附加订阅到本节点

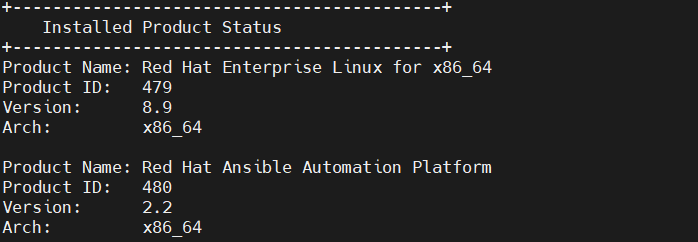
# subscription-manager attach --auto #自动附加订阅

# subscription-manager repos --list | grep "ansible-automation-platform"

# subscription-manager repos --enable ansible-automation-platform-2.2-for-rhel-8-x86\_64-rpms

Repository 'ansible-automation-platform-2.2-for-rhel-8-x86\_64-rpms' is enabled for this system.

# subscription-manager list #查看系统订阅情况



注册系统之后，会自动启用相关的软件仓库：

# cat /etc/yum.repos.d/redhat.repo | grep "enabled = 1" -C 3

**[ansible-automation-platform-2.2-for-rhel-8-x86\_64-rpms]**

name = Red Hat Ansible Automation Platform 2.2 for RHEL 8 x86\_64 (RPMs)

baseurl = https://cdn.redhat.com/content/dist/layered/rhel8/x86\_64/ansible-automation-platform/2.2/os

enabled = 1

gpgcheck = 1

gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

sslverify = 1

sslcacert = /etc/rhsm/ca/redhat-uep.pem

**[rhel-8-for-x86\_64-appstream-rpms]**

name = Red Hat Enterprise Linux 8 for x86\_64 - AppStream (RPMs)

baseurl = https://cdn.redhat.com/content/dist/rhel8/$releasever/x86\_64/appstream/os

enabled = 1

gpgcheck = 1

gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

sslverify = 1

sslcacert = /etc/rhsm/ca/redhat-uep.pem

**[rhel-8-for-x86\_64-baseos-rpms]**

name = Red Hat Enterprise Linux 8 for x86\_64 - BaseOS (RPMs)

baseurl = https://cdn.redhat.com/content/dist/rhel8/$releasever/x86\_64/baseos/os

enabled = 1

gpgcheck = 1

gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

sslverify = 1

如果没有aap的订阅，在创建repos时报错：

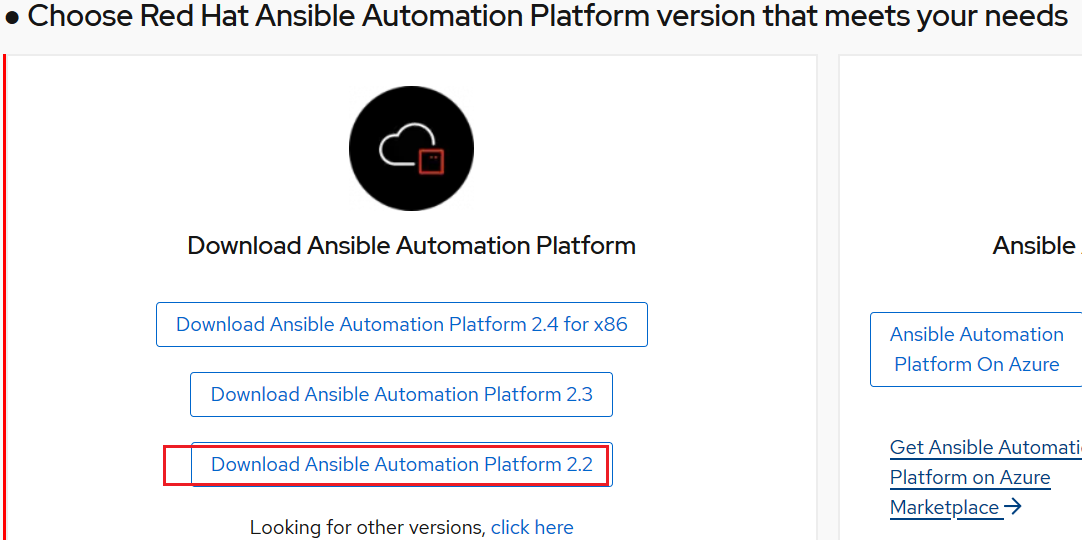
This system has no repositories available through subscriptions.

可以直接登录官网下载aap离线安装包

**★下载AAP安装包**

下载页面： https://developers.redhat.com/products/ansible/download

选择目标版本下载（本例使用AAP 2.2版本）



下载后，上传压缩包到目标服务器（要安装aap平台的服务器）

# tar -xf ansible-automation-platform-setup-bundle-2.2.0-7.tar.gz

# ls -lh ansible-automation-platform-setup-bundle-2.2.0-7/

drwxr-xr-x. 4 root root 28 Jun 30 2022 bundle #在 bundle/el8/repos 子目录下为AAP的rpm软件包

drwxr-xr-x. 3 root root 33 Jun 30 2022 collections

drwxr-xr-x. 2 root root 17 Jun 30 2022 group\_vars

drwxr-xr-x. 2 root root 120 Jun 30 2022 images

-rw-r--r--. 1 root root 8.4K Jun 30 2022 inventory

drwxr-xr-x. 3 root root 8.0K Jun 30 2022 licenses

-rw-r--r--. 1 root root 530 Jun 30 2022 README.md

-rwxr-xr-x. 1 root root 13K Jun 30 2022 setup.sh

# yum install ansible-core-2.13.0 -y #安装离线包里的ansible-core（可先把bundle/el8/repos 子目录做成本地镜像仓库）

**★安装ansible-navigator软件包**

若不想装web界面的控制器及hub自动化中心，只想装个命令行的ansible-navigator，可以在注册系统配置订阅后，直接使用yum安装ansible-navigator，也可下载离线包安装

# yum install ansible-navigator -y

# ansible-navigator --version

ansible-navigator 2.1.0

**★安装AAP自动化控制器（单机）**

使用离线包安装AAP平台的web ui界面服务，即automation-controller服务，本例中安装单机版本

首先完成前面的系统注册过程，并下载离线安装包

# tar -xf ansible-automation-platform-setup-bundle-2.2.0-7.tar.gz

# cd ansible-automation-platform-setup-bundle-2.2.0-7

# vi inventory #修改或添加以下蓝色字体的信息

[automationcontroller]

10.99.1.243

[all:vars]

admin\_password='passwdxxaap'

pg\_host=''

pg\_port=5432

pg\_database='awx'

pg\_username='awx'

pg\_password='passwdxxpg'

registry\_url='registry.redhat.io'

registry\_username='Limaofu'

registry\_password='passxxxx'

# yum install ansible-core-2.13.0 -y #安装离线包里的ansible-core

# ./setup.sh #开始安装

#安装原理是首先从红帽的软件仓库安装ansible-core及git等软件，然后使用离线包里的自动化脚本去部署整个AAP平台，如果出现以下报错：

TASK [ansible.automation\_platform\_installer.packages\_el : Disable required dnf modules] \*\*\*

fatal: [10.99.1.243]: FAILED! => {"changed": false, "msg": "Unsupported parameters for (ansible.legacy.command) module: warn. Supported parameters include: \_raw\_params, \_uses\_shell, argv, chdir, creates, executable, removes, stdin, stdin\_add\_newline, strip\_empty\_ends."}

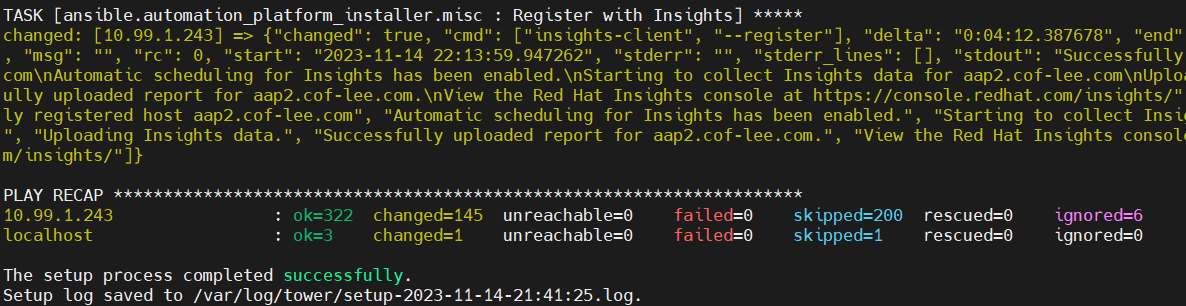
# cat collections/ansible\_collections/ansible/automation\_platform\_installer/roles/packages\_el/tasks/install\_deps.yml

脚本没啥问题，原因是前面忘记安装离线包里的ansible-core了，而setup.sh自动为我们安装的ansible-core版本较新，需要用离线包里的ansible-core版本

# yum remove ansible-core -y #卸载自动安装的ansible-core

# yum install ansible-core-2.13.0 -y #安装离线包里的ansible-core

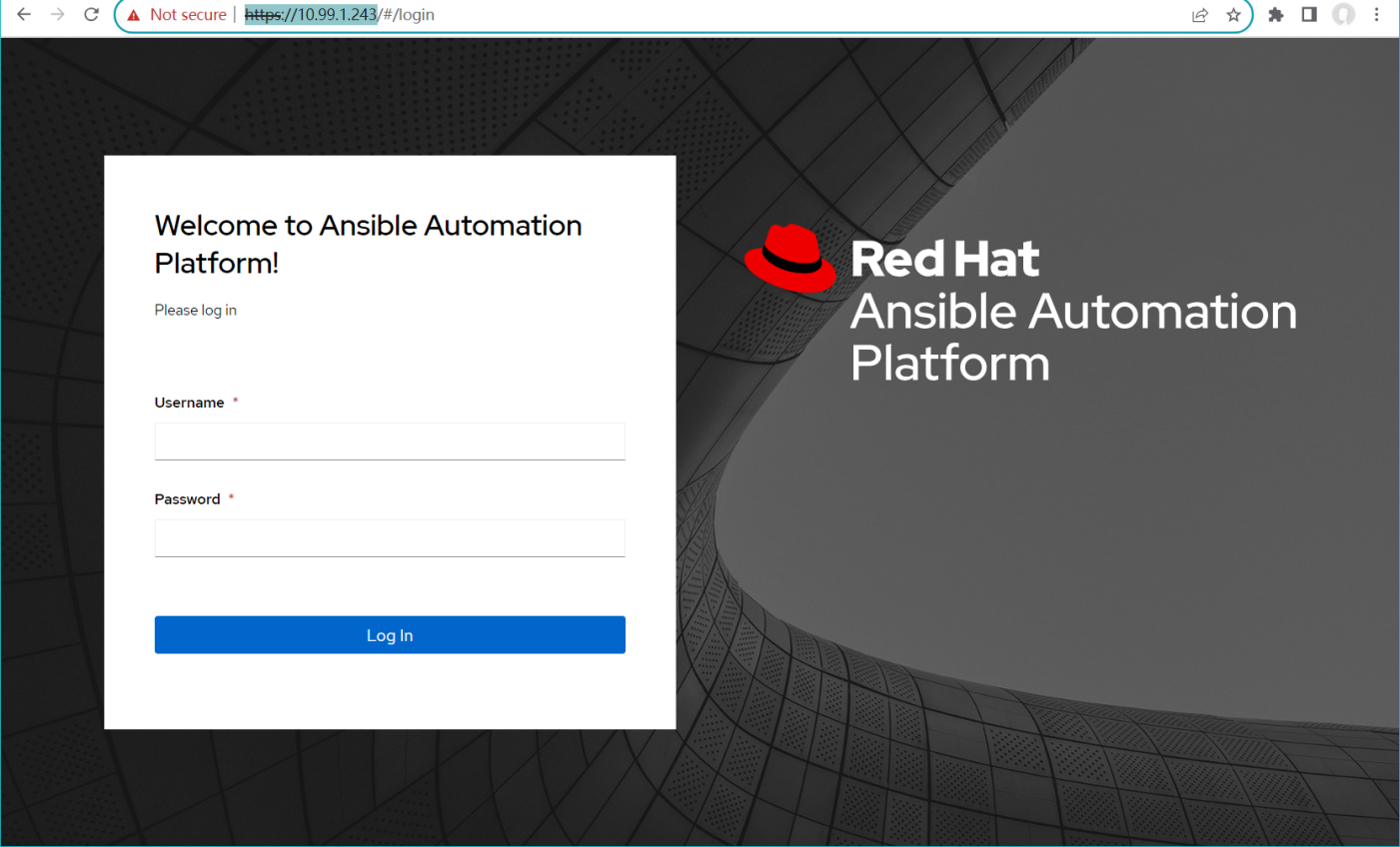
# ./setup.sh #重新开始安装，顺利的话等待10到30分钟安装完成



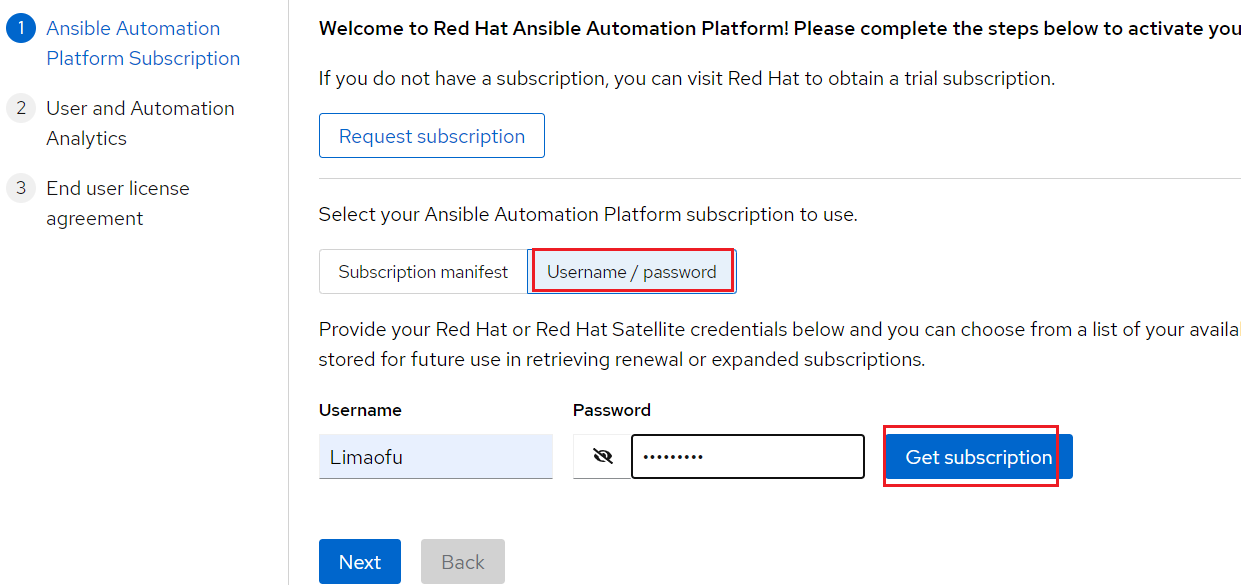
防火墙自动放通80,443端口

**★登录AAP自动化控制器web界面**

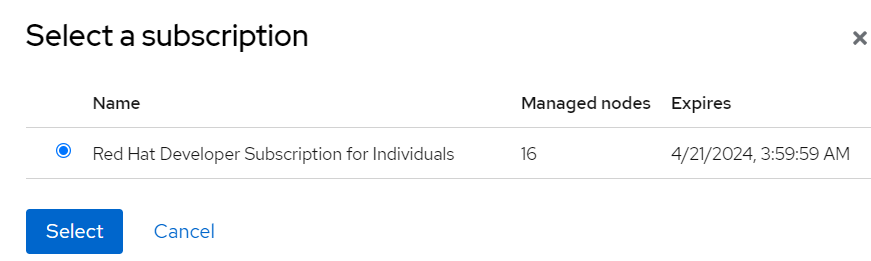
https://10.99.1.243/



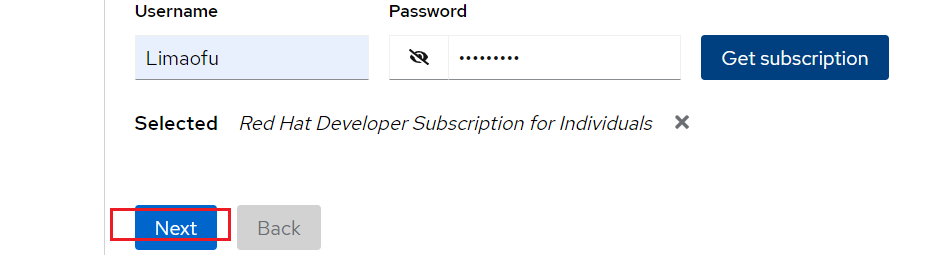
用户名admin，使用前面inventory文件里设置的密码"passwdxxaap"登录



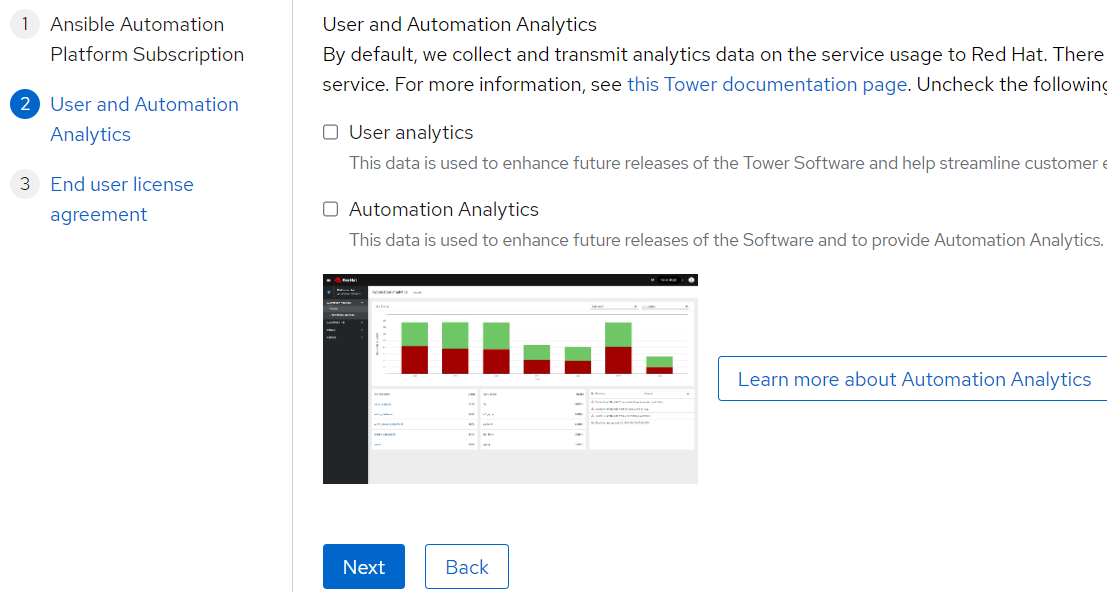
需要绑定有效的订阅，使用“username/passwor”方式，点击“Get subscription”或取此用户的订阅信息



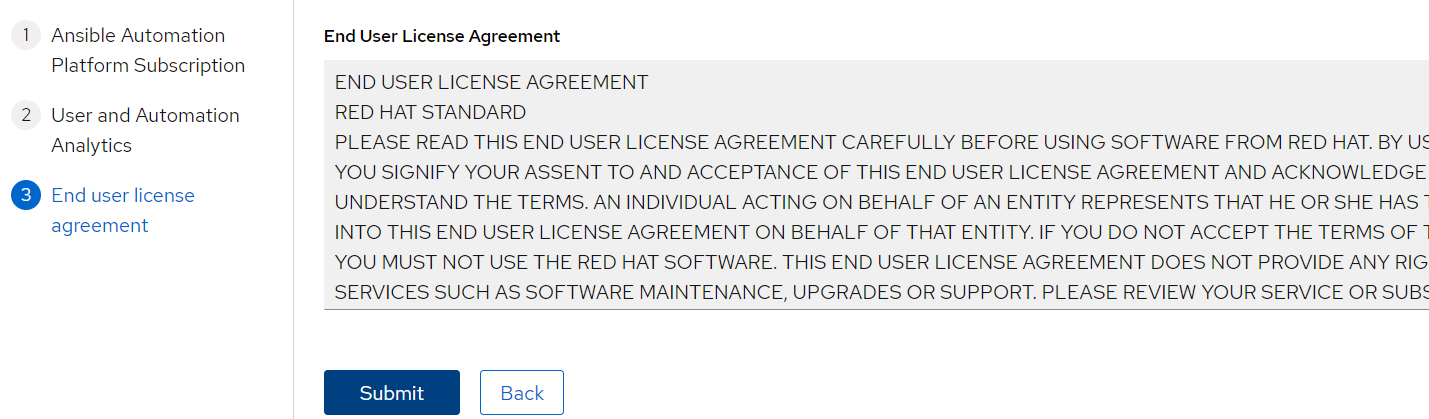
选择有效的订阅，点击“Select”



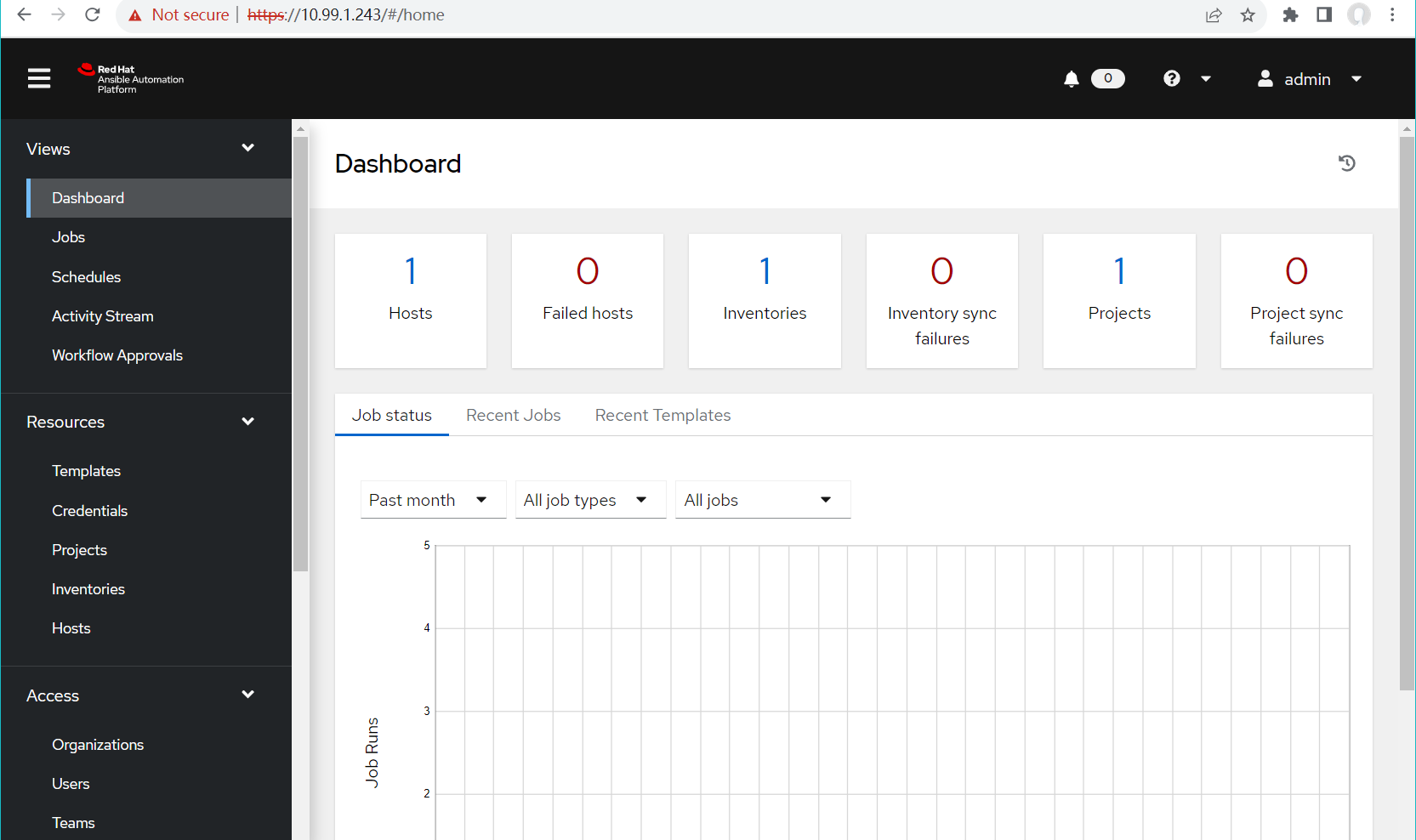
点击“Next”

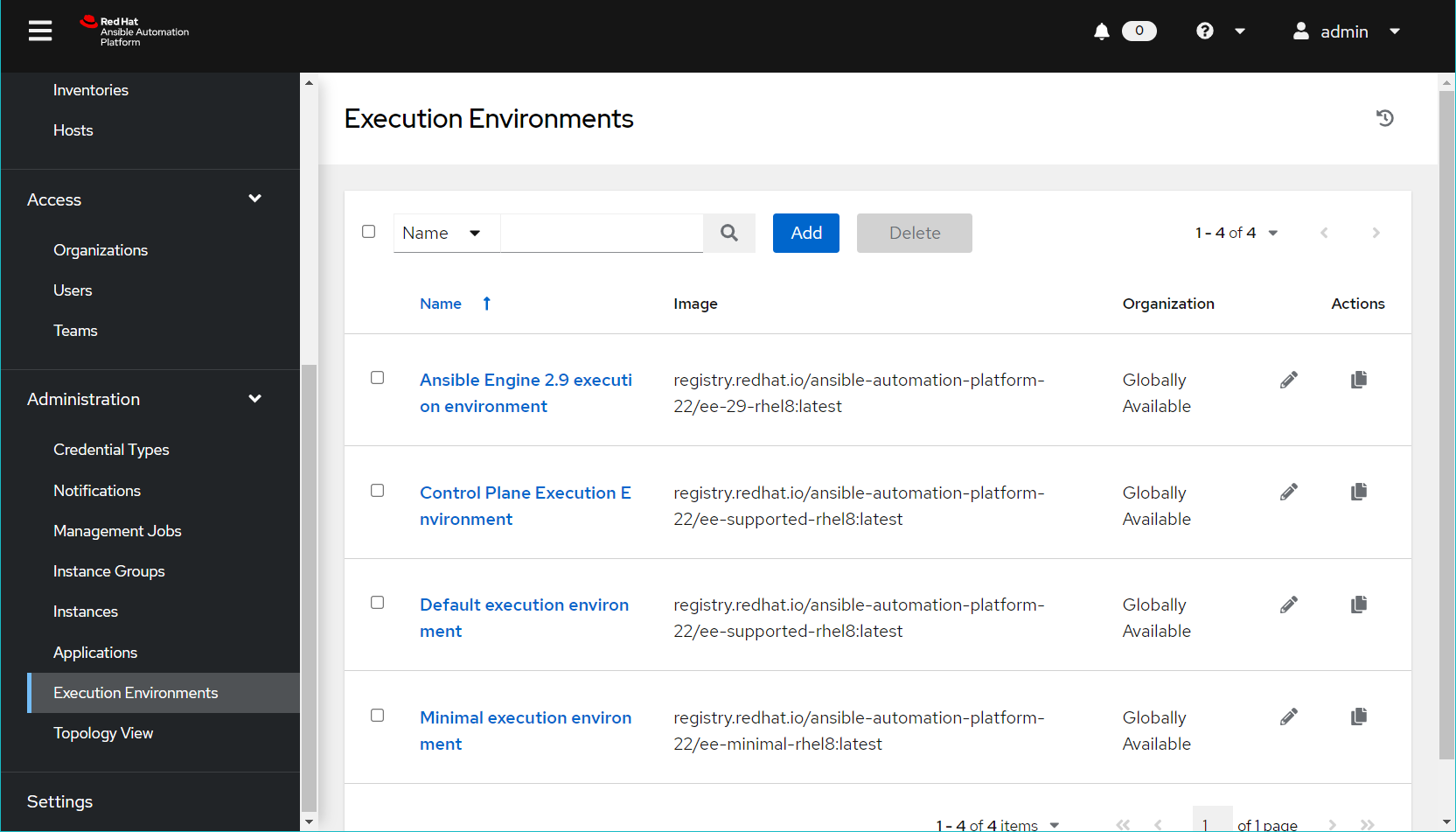


取消勾选“User analytics”及“Automation Analytics”



点击“Submit”同意用户许可，然后进入面板主页





**★安装AAP自动化控制器及自动化中心**

多节点部署（一台装AutomationController自动化控制器，另一台装AutomationHub自动化中心）

不能在同一节点上安装自动化控制器Controller和自动化中心Hub

安装前准备及注册系统步骤同上，过程这里省略了。

**★主机规划：**

|  |  |  |  |
| --- | --- | --- | --- |
| 主机名称 | ip | 角色 | 配置 |
| controller.cof-lee.com | 10.99.1.237 | AutomationController（hybrid） | 4核16G，100GB磁盘 |
| hub.cof-lee.com | 10.99.1.238 | AutomationHub | 2核8G，100GB磁盘 |

# cat >> /etc/hosts <<EOF

10.99.1.237 controller.cof-lee.com controller

10.99.1.238 hub.cof-lee.com hub

EOF

**★安装步骤**

只在controller节点上操作：

下载离线安装包并上传到controller节点上

# tar -xf ansible-automation-platform-setup-bundle-2.2.0-7.tar.gz

# cd ansible-automation-platform-setup-bundle-2.2.0-7

# vi inventory #修改或添加以下蓝色字体的信息

[automationcontroller]

controller.cof-lee.com

[automationhub]

hub.cof-lee.com

[all:vars]

admin\_password='passwdxxaap'

pg\_host=''

pg\_port=5432

pg\_database='awx'

pg\_username='awx'

pg\_password='passwdxxpg'

registry\_url='registry.redhat.io'

registry\_username='Limaofu'

registry\_password='passxxxx'

automationhub\_admin\_password='passwdxxhub'

automationhub\_pg\_host='controller.cof-lee.com'

automationhub\_pg\_port=5432

automationhub\_pg\_database='automationhub'

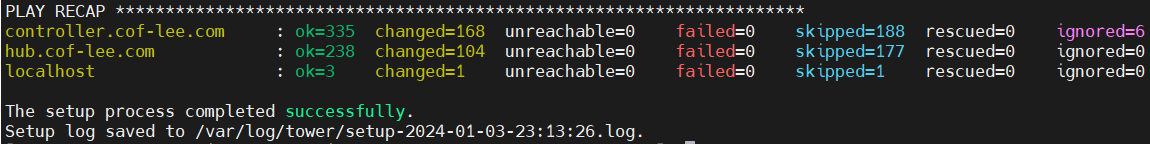
automationhub\_pg\_username='automationhub'

automationhub\_pg\_password='passwdxxhubpg'

automationhub\_pg\_sslmode='prefer'

# yum install ansible-core-2.13.0 -y #安装适配版本的ansible-core（AAP2.2对应2.13.0）

# ./setup.sh #开始安装

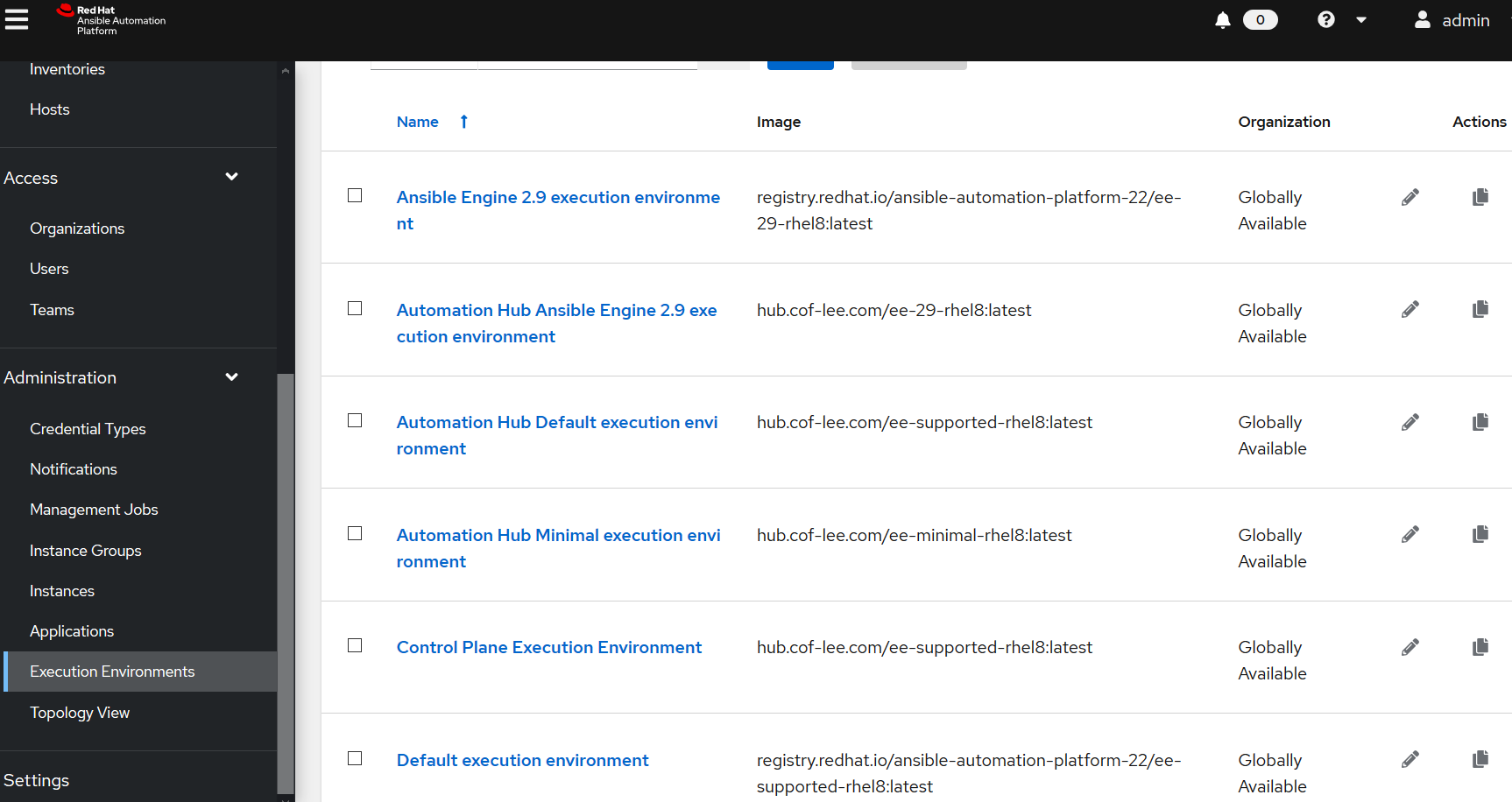


**★登录自动化控制器**

浏览器访问 https://controller地址/，请参考“安装AAP自动化控制器（单机）”里的登录过程，这里省略

和单机的自动化控制器不同之处在于：

有hub自动化中心时，其EEI执行环境镜像指向了本地的hub，比如 hub.cof-lee.com/ee-supported-rhel8:latest

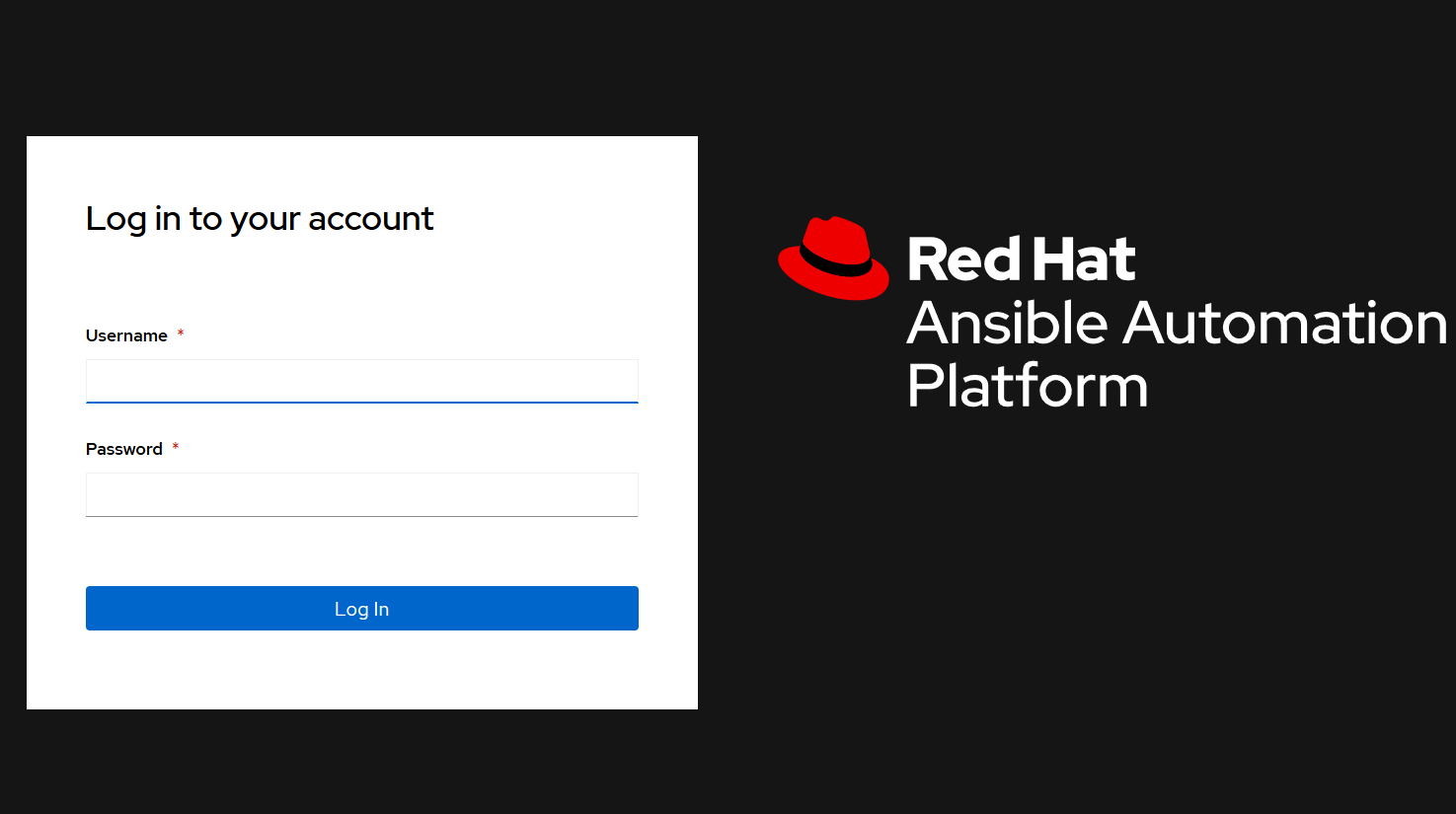


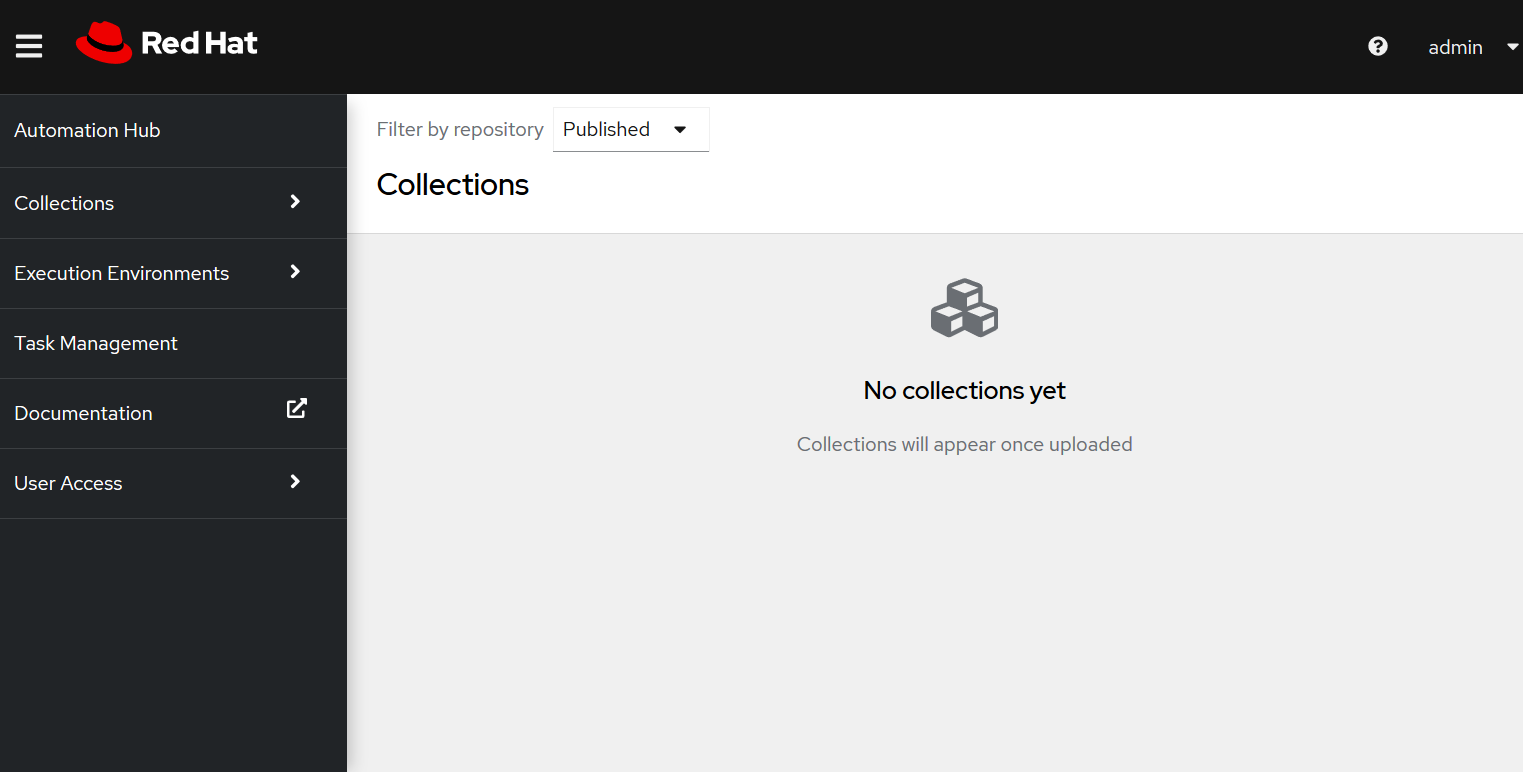


需要登录才能访问本地hub里的镜像

**★登录自动化中心**

浏览器访问 https://hub地址/，账号admin，密码同前面设置的automationhub\_admin\_password





**★第2章、ansible-navigator命令**

**★ansible-navigator运行playbook剧本**

★首先安装ansible-navigator软件包

# yum install ansible-navigator -y

# yum install --disablerepo=\\* --enablerepo=ansible-automation-platform ansible-navigator -y #非联网环境

★在工作目录下创建ansible.cfg及inventory文件

# mkdir ~/test

# cd ~/test

# cat > ansible.cfg <<EOF

[defaults]

inventory = ./inventory

host\_key\_checking = False

[privilege\_escalation]

become=True

become\_method=sudo

become\_user=root

become\_ask\_pass=False

EOF

# cat > inventory <<EOF

mariadb

EOF

# cat > play-testxx.yml <<EOF

---

- name: play testxx

hosts: mariadb

tasks:

- name: task xx

ansible.builtin.debug:

msg: this is test task xx

EOF

★使用ansible-navigator执行剧本

# ansible-navigator run play-testxx.yml -i inventory --eei ee-supported-rhel8:latest -m stdout

#run后必须先跟剧本文件，

#-i指定主机清单文件，默认为ansible.cfg里指定的inventory文件，若未指定则默认为/etc/ansible/hosts

#-eei指定执行环境镜像，默认是需要联网去registry.redhat.io拉取镜像的，默认镜像为：

# registry.redhat.io/ansible-automation-platform-22/ee-supported-rhel8:latest

#可以将镜像下载到本地，然后修改ansible-navigator.yml配置为：优先使用本地的镜像

# -m stdout表示以标准输出方式显示结果，默认进入交互模式界面

**★使用本地镜像仓库**

# cat > /etc/containers/registries.conf <<EOF

unqualified-search-registries = [ "cof-lee.com" ]

[[registry]]

prefix = "cof-lee.com"

location = "cof-lee.com"

insecure = true

EOF

#navigator配置文件指定默认使用的镜像及镜像获取方式，以及执行playbook输出模式为stdout非交互式

# cat > ./ansible-navigator.yml <<EOF

---

ansible-navigator:

mode: stdout #指定为非交互模式，默认是交互模式interactive

execution-environment:

image: hub.cof-lee.com/ee-supported-rhel8:latest #指定默认的eei执行环境镜像

pull:

policy: missing #表示优先用要地的，本地没有则网络拉取；always表示只从网络拉取镜像的，默认

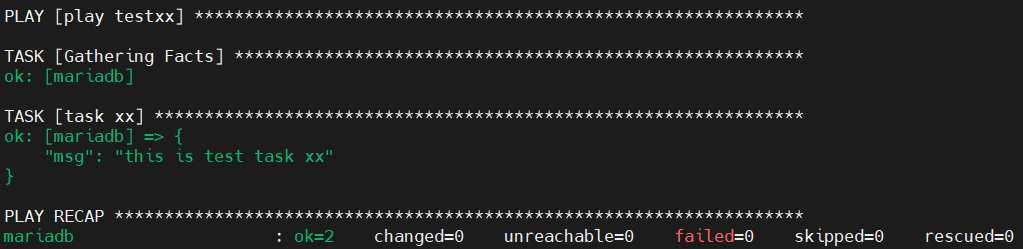
EOF

mode使用 stdout（非交互）配置时，如果要以交互模式运行，需要带上参数 -m interactive

mode使用 interactive（交互）配置时，如果要以非交互模式运行，需要带上参数 -m stdout

**★以标准输出模式执行playbook**

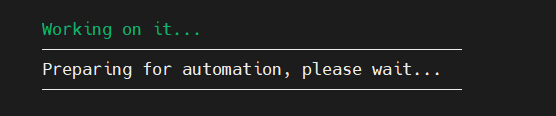
# ansible-navigator run play-testxx.yml -i inventory --eei hub.cof-lee.com/ee-supported-rhel8:latest -m stdout



和执行ansible-playbook命令一样，直接输出运行结果，不需要人工干预

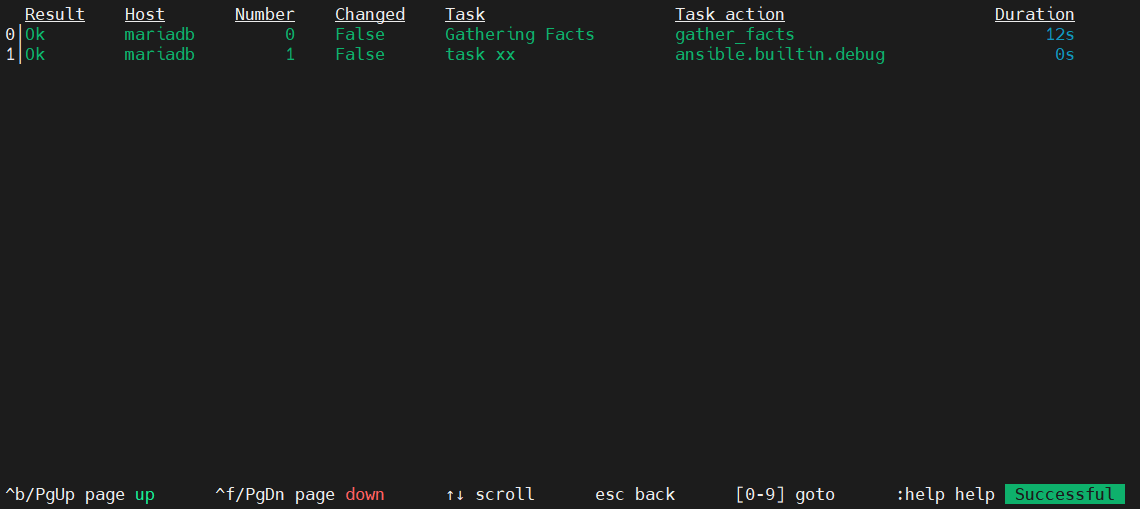
**★以交互模式执行playbook**

# ansible-navigator run play-testxx.yml -i inventory -m interactive #未指定eei时，使用配置文件里默认的eei执行环境镜像hub.cof-lee.com/ee-supported-rhel8:latest

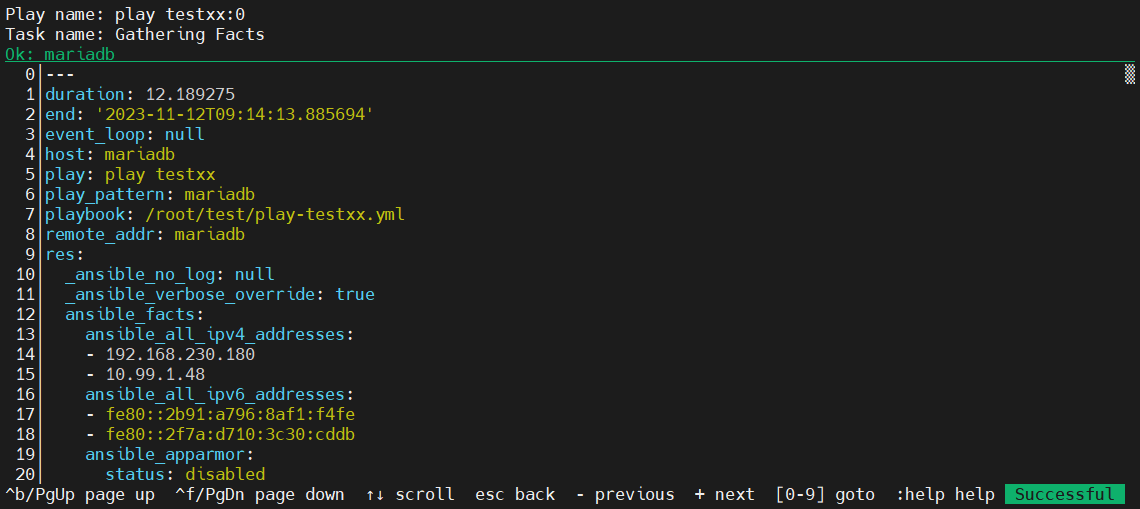




默认进入交互模式，按下数字0，进入序号为0的playbook详情页（0-9可直接按相应数字键进入，10以上需要先按下:冒号再加数字回车）



按下相应数字，查看任务详情，比如按下0查看gather\_facts详情：



可按下加号+查看下一页，按Esc键返回上一级，直到最上级后，再次按Esc则退出ansible-navigator交互界面

**★ansible-navigator运行原理**

ansible-navigator运行剧本时，首先使用podman创建一个临时容器，使用的镜像称为eei（Execution Environment Image）执行环境镜像（镜像里包含一套完整的ansible工具命令）

然后在容器里使用ansible-playbook命令执行指定的playbook剧本，剧本执行结束后，结束相应的容器

# podman ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

96fbe1f78912 cof-lee.com/aap/ee-supported-rhel8:latest ansible-playbook ... 6 minutes ago Up 6 minutes ago ansible\_runner\_6be863a9-b071-46b9-9328-23b6ff6e0079

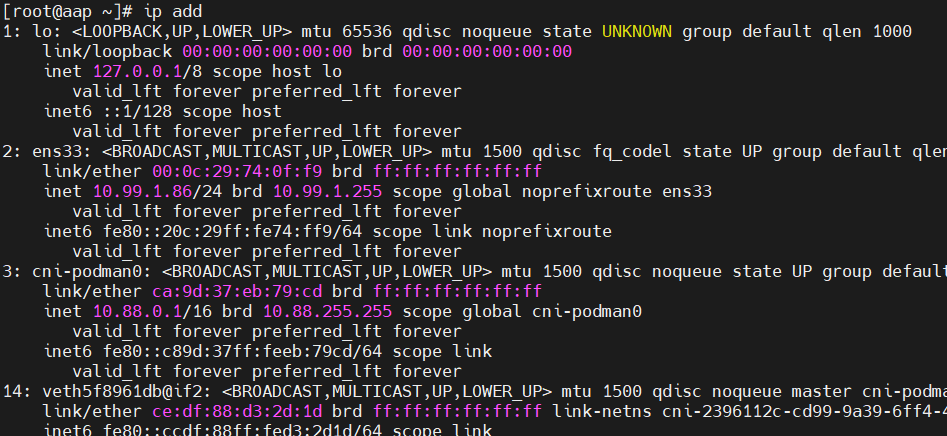
宿主机（执行机）# ip netns exec 容器id ip addr #查看容器的ip

2: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP group default

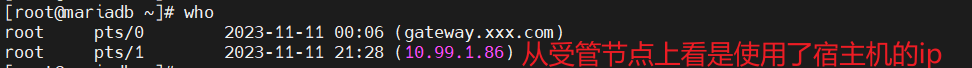
link/ether 5a:e2:ef:f8:fe:bd brd ff:ff:ff:ff:ff:ff link-netnsid 0

inet 10.88.0.11/16 brd 10.88.255.255 scope global eth0

宿主机（执行机）# ip addr #执行机的ip为10.99.1.86，容器网段10.88.0.0/16



登录到受管节点查看，下图可见当容器访问外部节点时，做了nat地址转换，转为执行机的ip了



ansible-navigator执行时，自动将执行机（宿主节点）的当前工作路径映射到eei容器里，以及当前用户的~/.ssh路径也映射到容器里了

# podman inspect 容器id #查看目录映射情况

"HostConfig": {

"Binds": [

"/tmp/ansible-navigator\_q64mmldm/artifacts:/runner/artifacts/:rw,rprivate,rbind",

"/tmp/ansible-navigator\_q64mmldm:/runner/:rw,rprivate,rbind",

"/root/test:/root/test/:rw,rprivate,rbind", #当前工作路径

"/tmp/ssh-I45OMkJzGDWy:/tmp/ssh-I45OMkJzGDWy/:rw,rprivate,rbind",

"/root/.ssh:/home/runner/.ssh/:rw,rprivate,rbind", #映射ssh密钥路径

"/root/.ssh:/root/.ssh/:rw,rprivate,rbind"

],

#执行机的/etc/hosts文件内容也添加到了容器里的/etc/hosts里

sh-4.4# more /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

10.99.1.86 controller.cof-lee.com controller

10.99.1.238 hub.cof-lee.com hub

10.99.1.246 harbor.cof-lee.com cof-lee.com cof-net.com

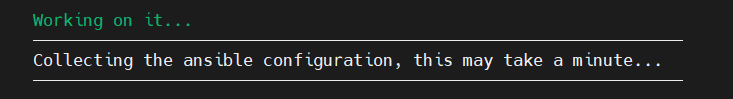
10.99.1.86 host.containers.internal

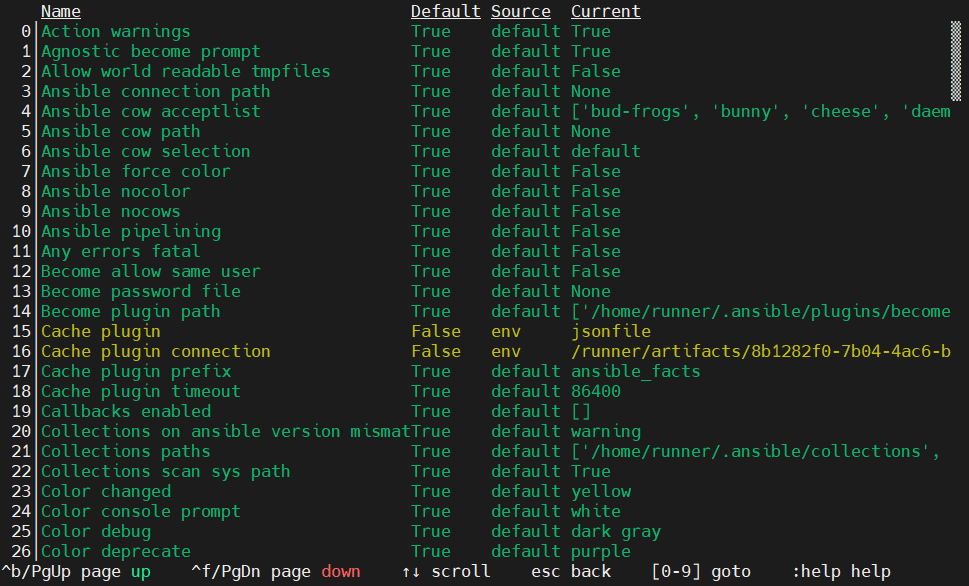
10.0.2.100 b79fdf7c6d36 ansible\_runner\_d09821d2-8146-4122-845e-f545f11efbfb

**★ansible-navigator配置**

# ansible-navigator config dump -m stdout #以标准输出方式列出所有配置项（也会创建临时eei环境）

# ansible-navigator config -m interactive #进入交互界面，查看配置项



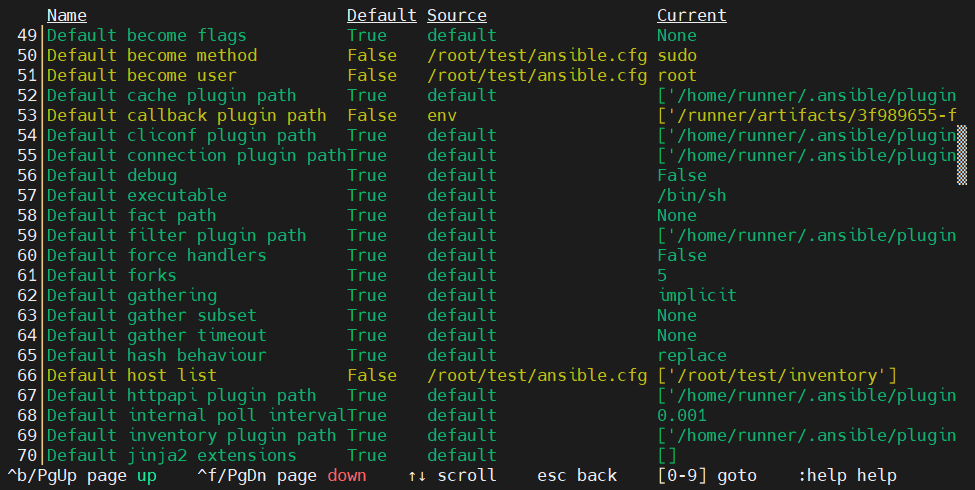


可按下冒号:f 加配置项，查看对应的配置信息，这个f表示filter（过滤，查找）例如：

:f forks #查看含有名为forks的配置项







如上图，最左侧数字表示行数，接下来各列说明：

Name这列表示配置项名称，

Default这列表示是否使用默认配置

Source这列表示配置的来源，值为default表示使用的是默认配置，其他为配置源文件或来源

Current这列表示该配置项的当前生效值

2.2版本的AAP大概默认有194行配置项

**★使用本地镜像仓库及输出模式为stdout非交互式**

# cat > /etc/containers/registries.conf <<EOF

unqualified-search-registries = [ "cof-lee.com" ]

[[registry]]

prefix = "cof-lee.com"

location = "cof-lee.com"

insecure = true

EOF

#navigator配置文件指定默认使用的镜像及镜像获取方式，以及执行playbook输出模式为stdout非交互式

# cat > ./ansible-navigator.yml <<EOF

---

ansible-navigator:

mode: stdout #指定为非交互模式，默认是交互模式interactive

execution-environment:

image: hub.cof-lee.com/ee-supported-rhel8:latest #指定默认的eei执行环境镜像

pull:

policy: missing #表示优先用要地的，本地没有则网络拉取；always表示只从网络拉取镜像的，默认

EOF

mode使用 stdout（非交互）配置时，如果要以交互模式运行，需要带上参数 -m interactive

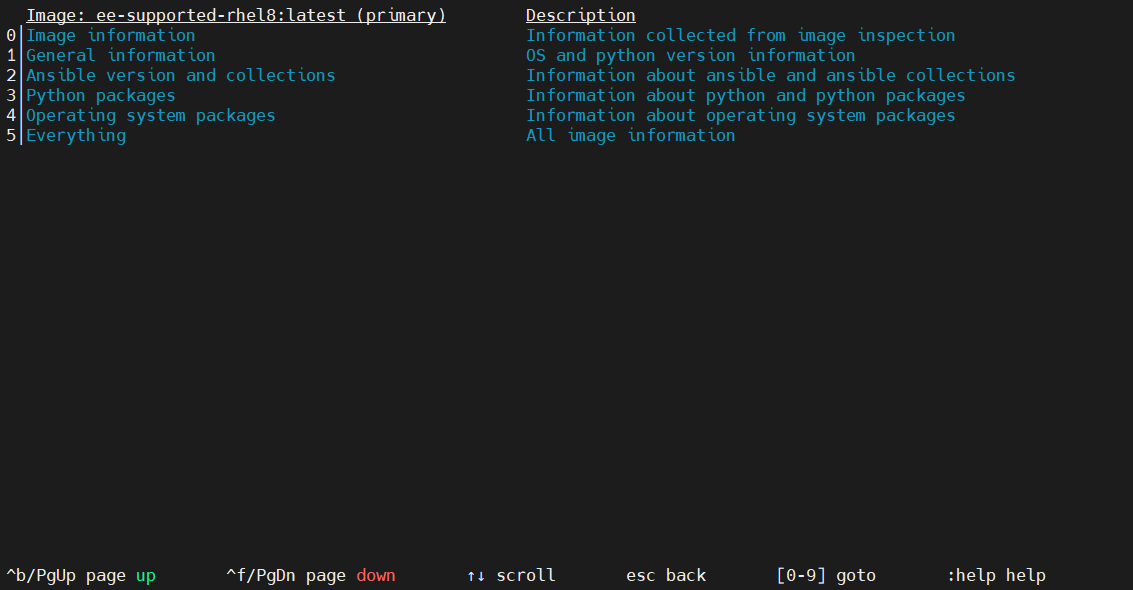
mode使用 interactive（交互）配置时，如果要以非交互模式运行，需要带上参数 -m stdout

**★构建EEI执行环境镜像**

# podman pull registry.redhat.io/ansible-automation-platform-22/ansible-builder-rhel8 #红帽官方的builder镜像

★假如需要基于'hub.cof-lee.com/ee-supported-rhel8:latest'镜像，安装某个role/collection，再生成新的镜像：

# ansible-navigator images --eei hub.cof-lee.com/ee-supported-rhel8:latest #查看基础镜像的信息



★创建用于构建镜像的清单文件

# mkdir ~/test

# cd ~/test

# cat > execution-environment.yml <<EOF

---

version: 1

build\_arg\_defaults:

EE\_BASE\_IMAGE: 'hub.cof-lee.com/ee-supported-rhel8:latest' #基于此镜像进行自定义添加新功能

EE\_BUILDER\_IMAGE: 'hub.cof-lee.com/ansible-builder-rhel8:latest' #此镜像用于builder操作

dependencies:

galaxy: requirements.yml # galaxy表示需要安装ansible-galaxy的内容，可为role及collection【可选参数】

python: requirements.txt # python表示需要安装的python包【可选参数】

system: bindep.txt # system表示需要安装软件包，可为rpm软件包【可选参数】

EOF

# cat > requirements.yml <<EOF

collections:

- name: /build/limaofu-collectionyy-1.0.0.tar.gz #依赖的collection，或者说要安装的collection

EOF

#说明，因为构建镜像时，是将context/\_build/映射到容器里的/build，且将/build设置为容器的WORKDIR，所以依赖的文件路径只能以/build目录开头

# cat > bindep.txt <<EOF

python36 [platform:rpm]

python3-pip [platform:rpm]

EOF

# cat > requirements.txt <<EOF

paramiko

EOF

# ansible-builder create #创建构建配置信息，会自动生成context目录

Complete! The build context can be found at: /root/test/context

# tree context/ #查看context目录结构

context/

├── \_build

│ └── bindep.txt #这个文件就是我们前面写的system软件清单，自动复制过来了

│ └── requirements.txt #这个文件就是我们前面写的python依赖清单，自动复制过来了

│ └── requirements.yml #这个文件就是我们前面写的galaxy依赖清单，自动复制过来了

└── Containerfile #这个是ansible-builer要用到的配置文件，根据execution-environment.yml生成的

# cat context/Containerfile

ARG EE\_BASE\_IMAGE=hub.cof-lee.com/ee-supported-rhel8:latest #基础镜像

ARG EE\_BUILDER\_IMAGE=hub.cof-lee.com/ansible-builder-rhel8:latest #构建镜像

FROM $EE\_BASE\_IMAGE as galaxy

ARG ANSIBLE\_GALAXY\_CLI\_COLLECTION\_OPTS=

USER root

ADD \_build /build #路径映射，将执行机本地的context/\_build/映射到容器里的/build

WORKDIR /build

#接下来开始进行构建操作，首先安装ansible依赖的role及collection

RUN ansible-galaxy role install -r requirements.yml --roles-path "/usr/share/ansible/roles"

RUN ANSIBLE\_GALAXY\_DISABLE\_GPG\_VERIFY=1 ansible-galaxy collection install $ANSIBLE\_GALAXY\_CLI\_COLLECTION\_OPTS -r requirements.yml --collections-path "/usr/share/ansible/collections"

FROM $EE\_BUILDER\_IMAGE as builder

COPY --from=galaxy /usr/share/ansible /usr/share/ansible

RUN ansible-builder introspect --sanitize --user-pip=requirements.txt --user-bindep=bindep.txt --write-bindep=/tmp/src/bindep.txt --write-pip=/tmp/src/requirements.txt

RUN assemble

FROM $EE\_BASE\_IMAGE

USER root

COPY --from=galaxy /usr/share/ansible /usr/share/ansible

COPY --from=builder /output/ /output/

RUN /output/install-from-bindep && rm -rf /output/wheels

#将本次构建依赖的collection压缩包复制到context/\_build/目录下

# cp ~/limaofu-collectionyy-1.0.0.tar.gz context/\_build/

# podman login hub.cof-lee.com -u admin -p passwdxxhub #登录hub，使我们有权限下载需要用到的镜像，如果要获取的镜像为public公共镜像，则可以不用登录hub

# ansible-builder build -t ee-user-supported:2.2 #构建镜像，大概需要5-10分钟，期间需要联网更新依赖

Running command:

podman build -f context/Containerfile -t ee-user-supported:2.2 context

Complete! The build context can be found at: /root/test/context

# podman images

REPOSITORY TAG IMAGE ID CREATED SIZE

localhost/ee-user-supported 2.2 ed5eec026371 16 minutes ago 1.77 GB

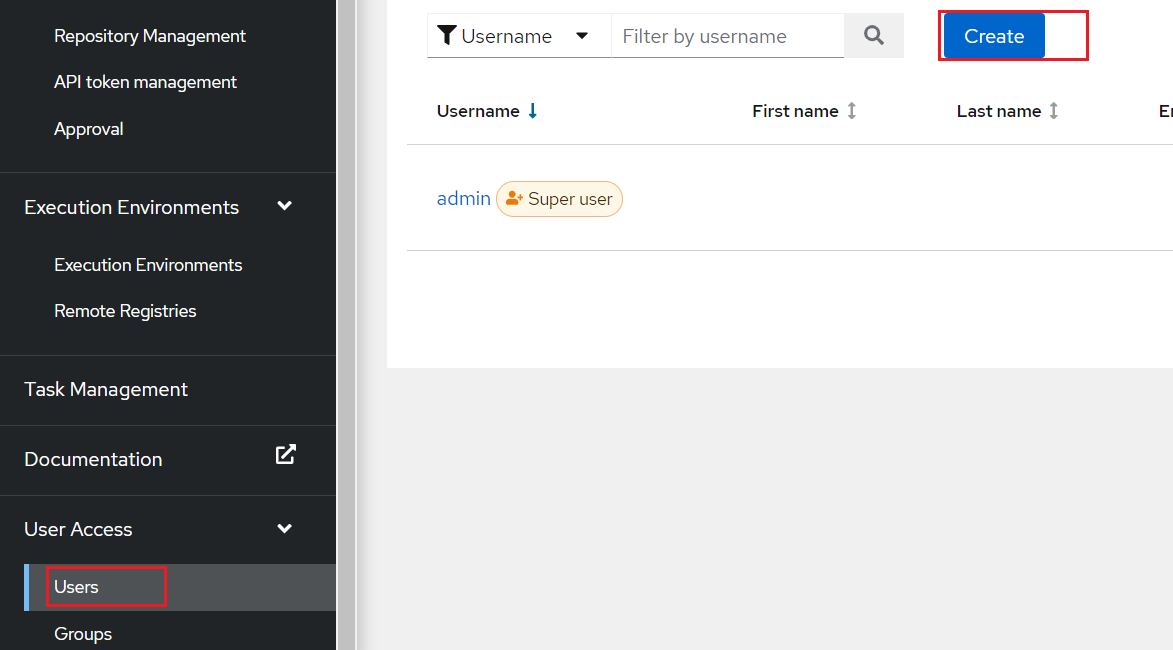
# podman tag localhost/ee-user-supported:2.2 hub.cof-lee.com/ee-user-supported:2.2

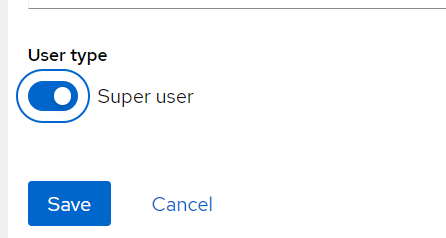
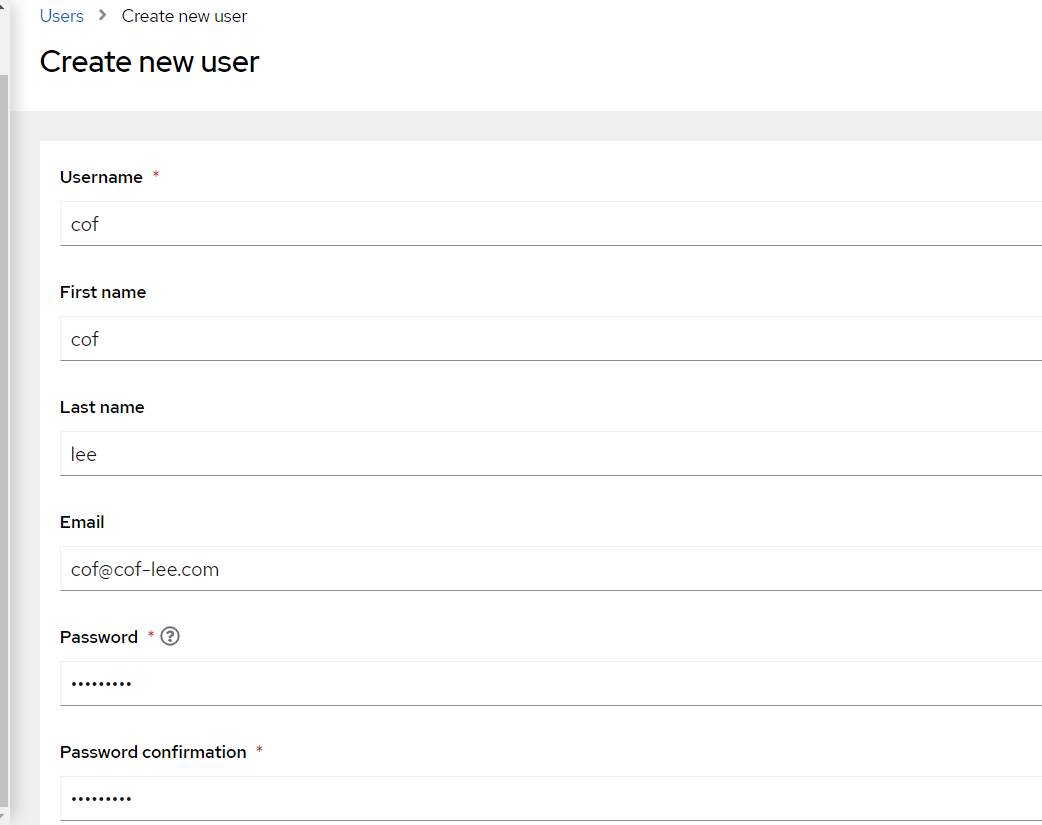
# podman push hub.cof-lee.com/ee-user-supported:2.2

**★第3章、Automation Hub平台的使用**

**★创建用户**

使用admin账号登录Hub https://10.99.1.238/



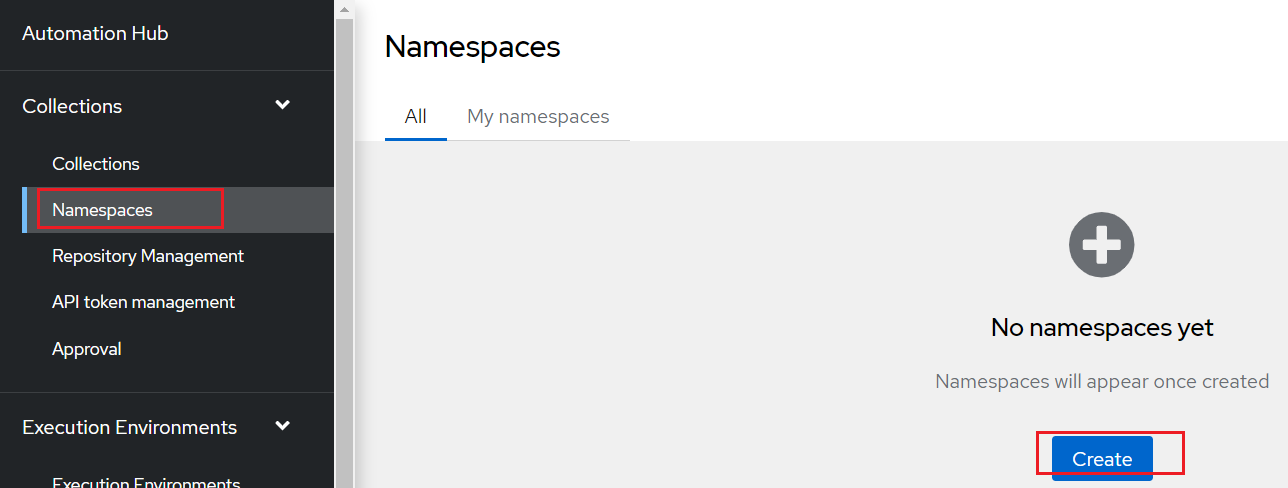


填写用户名及密码等相关信息，勾选“Super user”超级用户才可创建命名空间及发布collection

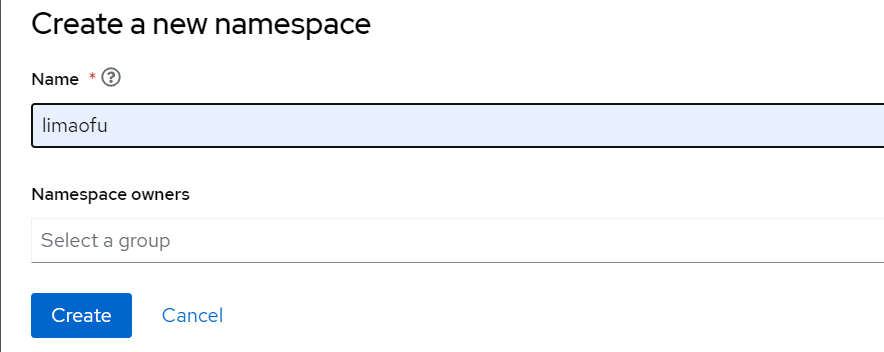
点击“save”保存

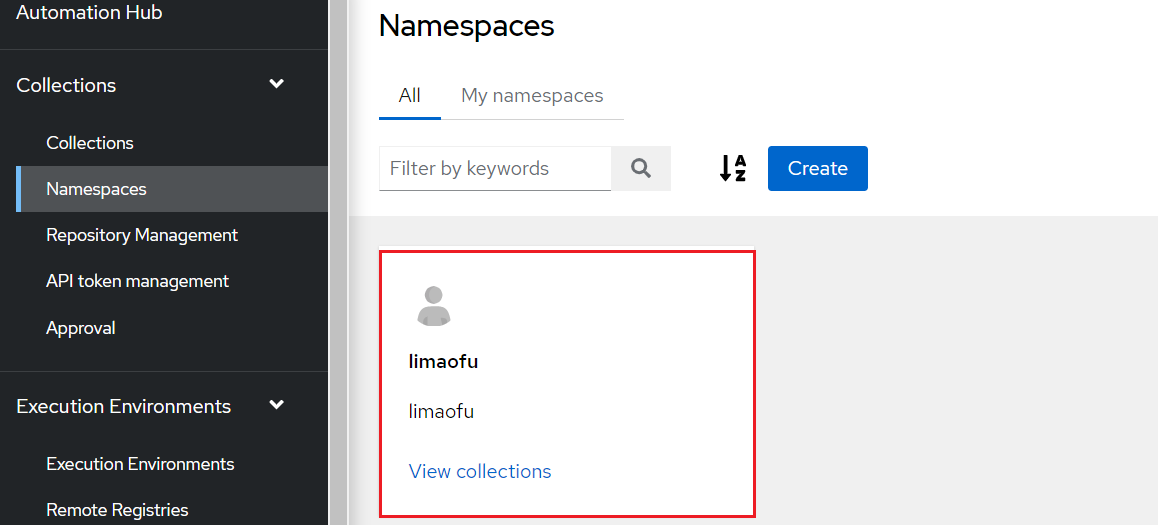
**★创建namespace**

Collections→Namespaces→点击“Create”创建



Name写集合的命名空间，比如我们创建了一个集合，名为limaofu.collectionyy，这里就定前面那部分：limaofu

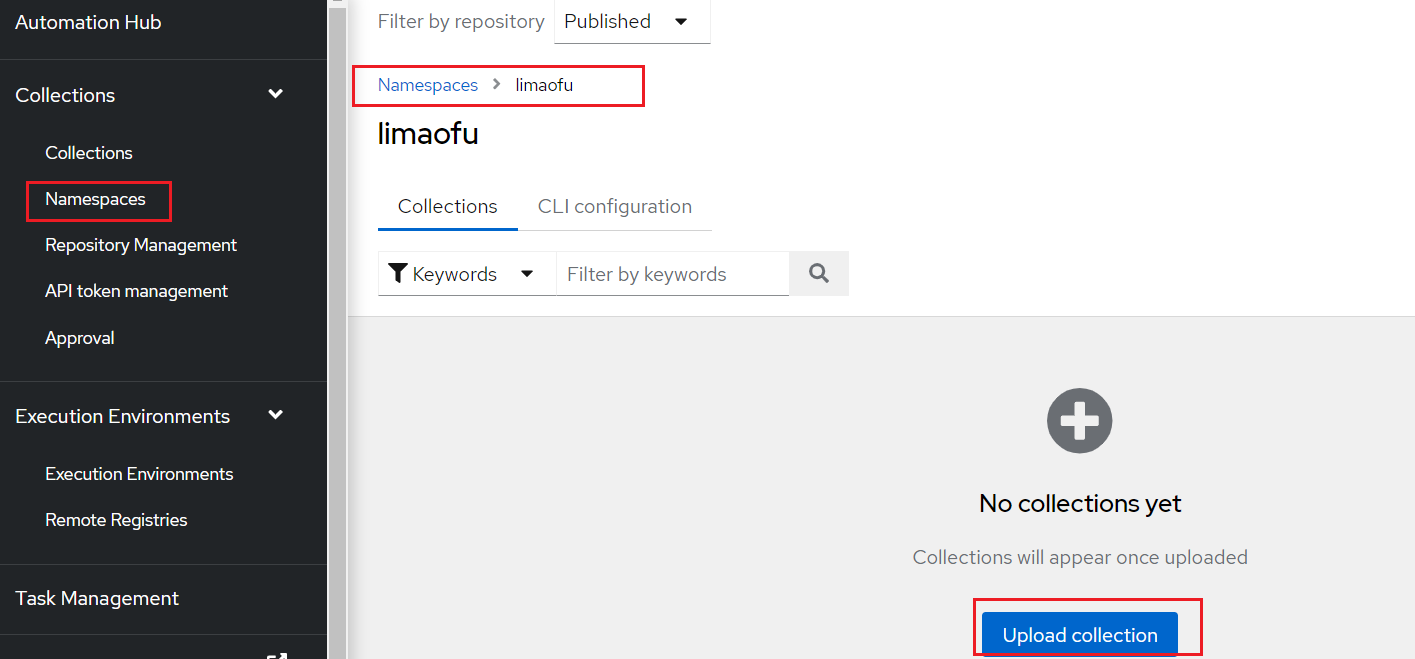




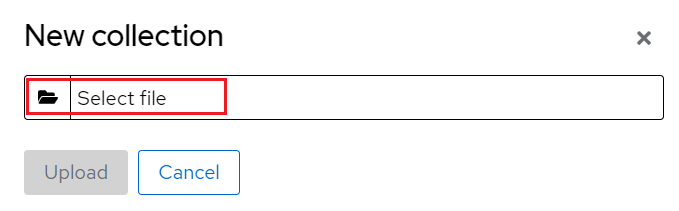
在Collections→Namespaces里可见刚刚创建的命名空间

**★发布collection**

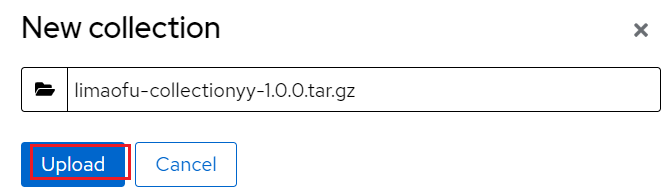
Collections→Namespaces里，点击进入目标命名空间，点击“Upload collection”

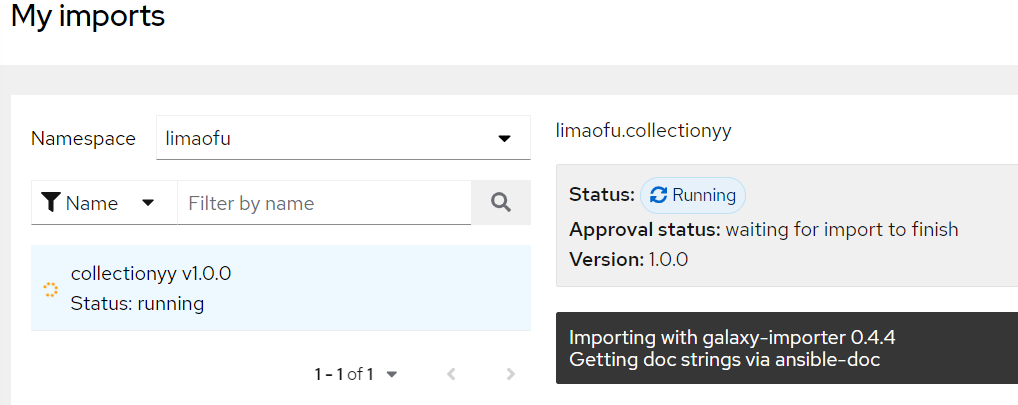


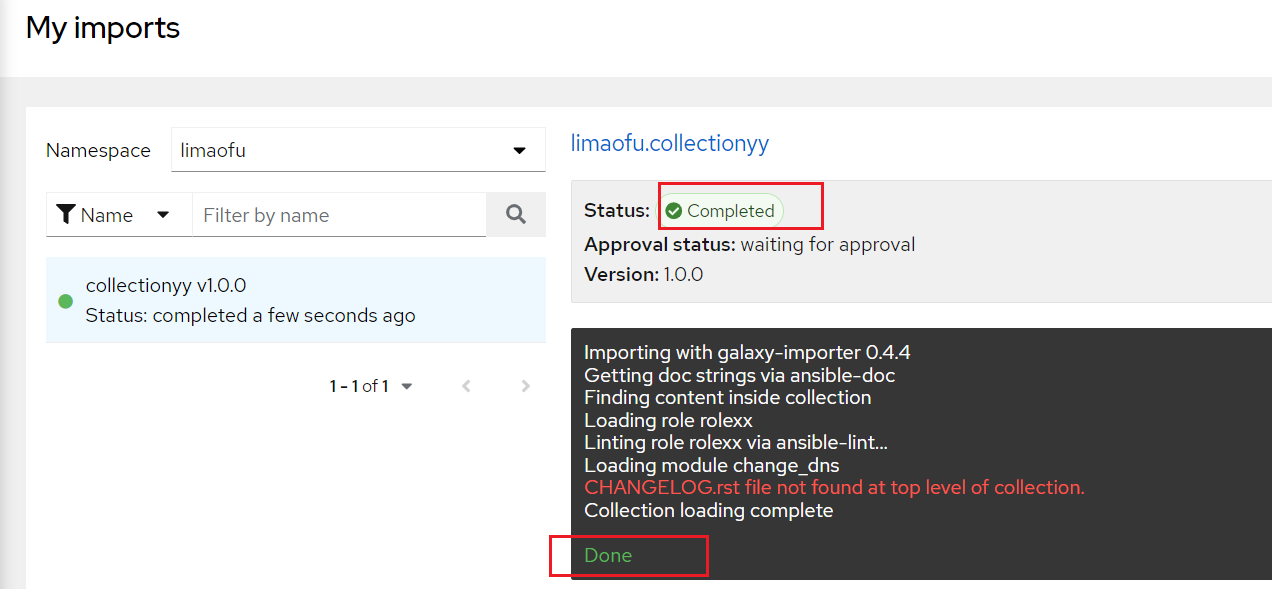
选择目标文件



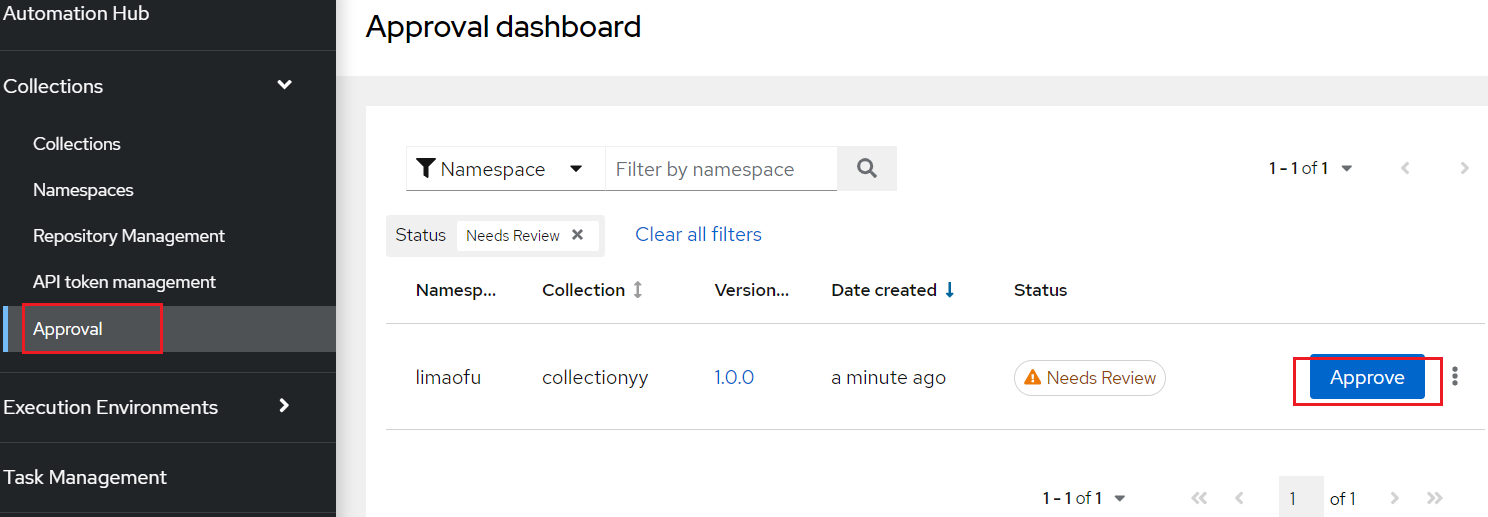
点击“Upload”上传



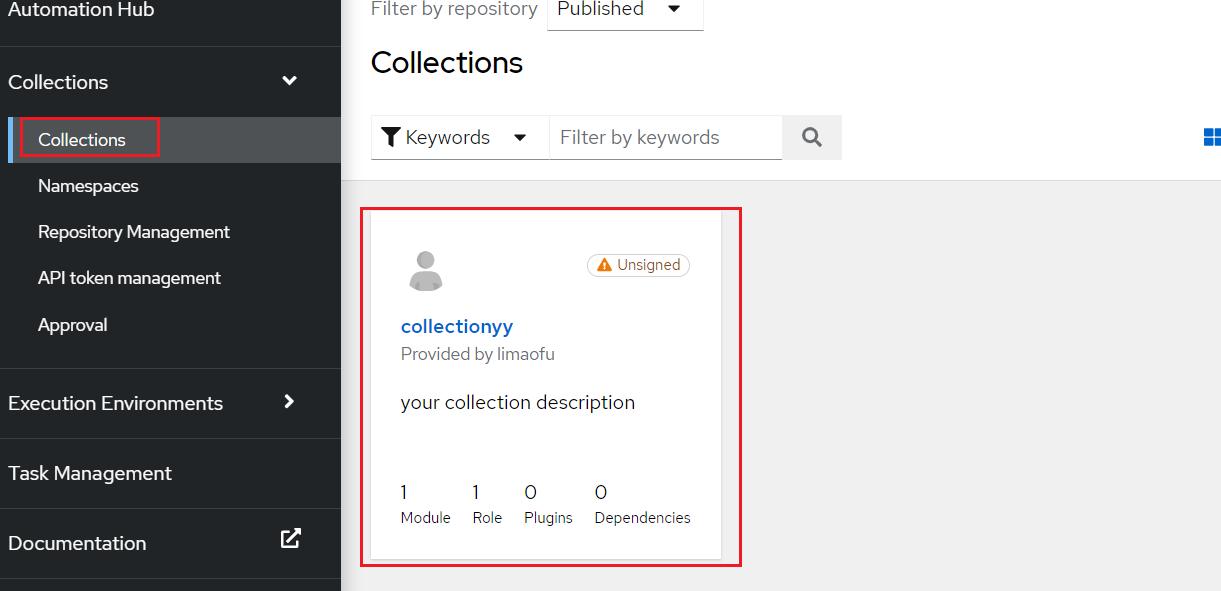




上传成功后，再到Collections→Approval里点击“Aprove”进行发布



发布成功后，在Collections里进见刚刚发布的集合

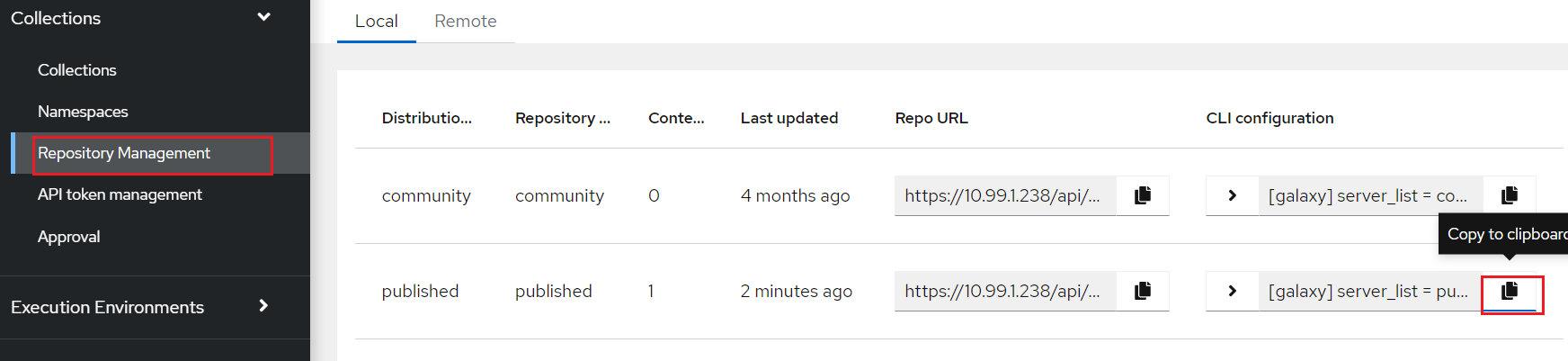


**★ansible使用私有hub**

前面2小节，我们创建了namespace以及发布了collection，现在有ansible执行机想使用这个collection，

可以在ansible.cfg里指定使用我们私有的automation hub仓库

首先在Hub的Collections→Repository Management里，点击published这一行右边的Copy to clipboard按钮



内容如下：

[galaxy]

server\_list = published\_repo

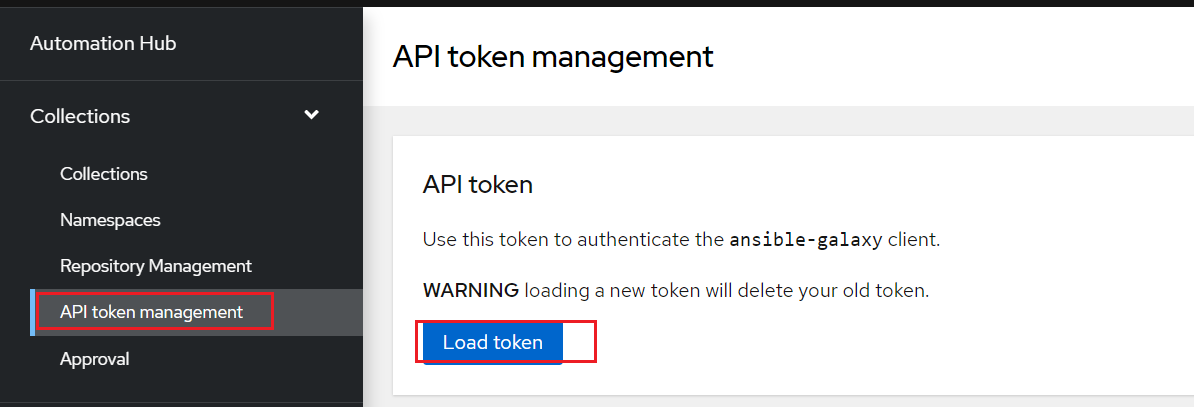
[galaxy\_server.published\_repo]

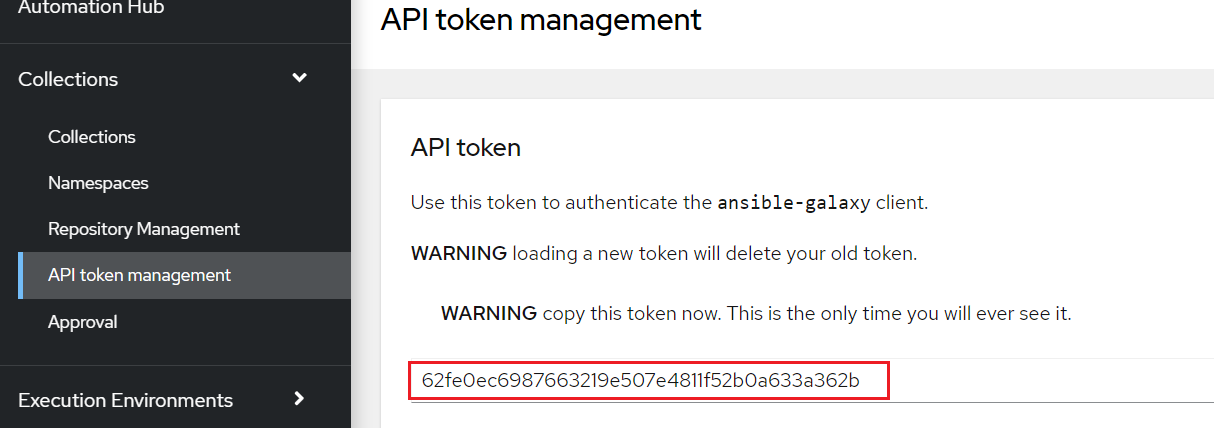
url=https://10.99.1.238/api/galaxy/content/published/

token=<put your token here>

再去获取此仓库的token

Collection→API token management→Load token





复制token: 62fe0ec6987663219e507e4811f52b0a633a362b

（token只可load一次，此界面只可打开一次，再次打开token就变了）

执行机配置# vi ansible.cfg

[defaults]

inventory = ./inventory

collections\_paths = /home/cof/mycollections

[galaxy]

server\_list = published\_repo

[galaxy\_server.published\_repo]

url=https://10.99.1.238/api/galaxy/content/published/

token=62fe0ec6987663219e507e4811f52b0a633a362b

# ansible-galaxy collection install limaofu.collectionyy -p /home/cof/mycollections/

执行后报错了：

Starting galaxy collection install process

Process install dependency map

[WARNING]: Skipping Galaxy server https://10.99.1.238/api/galaxy/content/published/. Got an unexpected error when getting available versions of

collection limaofu.collectionyy: Unknown error when attempting to call Galaxy at 'https://10.99.1.238/api/galaxy/content/published/api': <urlopen

error [SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verify failed: IP address mismatch, certificate is not valid for '10.99.1.238'. (\_ssl.c:1129)>

ERROR! Unknown error when attempting to call Galaxy at 'https://10.99.1.238/api/galaxy/content/published/api': <urlopen error [SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verify failed: IP address mismatch, certificate is not valid for '10.99.1.238'. (\_ssl.c:1129)>

因为ssl证书是匹配的域名，写ip会导致检验失败。

将ansible.cfg里的url=https://10.99.1.238/api/galaxy/content/published/改为域名：

url=https://hub.cof-lee.com/api/galaxy/content/published/

# ansible-galaxy collection install limaofu.collectionyy -p /home/cof/mycollections/ #再次安装，成功了

Starting galaxy collection install process

Process install dependency map

Starting collection install process

Downloading https://hub.cof-lee.com/api/galaxy/v3/plugin/ansible/content/published/collections/artifacts/limaofu-collectionyy-1.0.0.tar.gz to /home/cof/.ansible/tmp/ansible-local-2016621emut4x/tmpfu32j9fn/limaofu-collectionyy-1.0.0-xat94z88

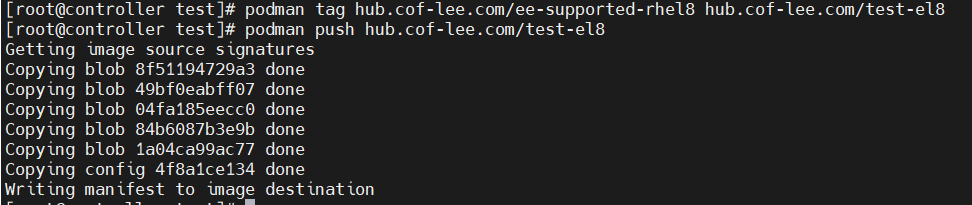
Installing 'limaofu.collectionyy:1.0.0' to '/home/cof/mycollections/ansible\_collections/limaofu/collectionyy'

limaofu.collectionyy:1.0.0 was installed successfully

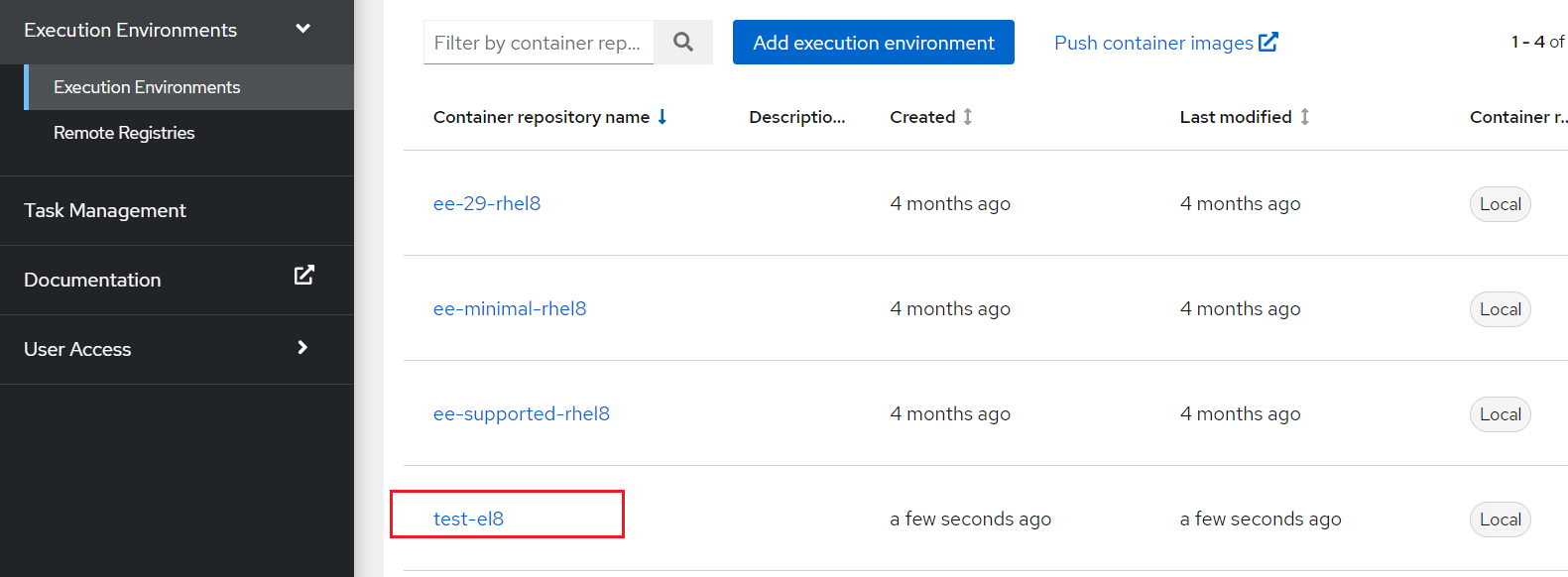
**★新增执行环境镜像（eei）**

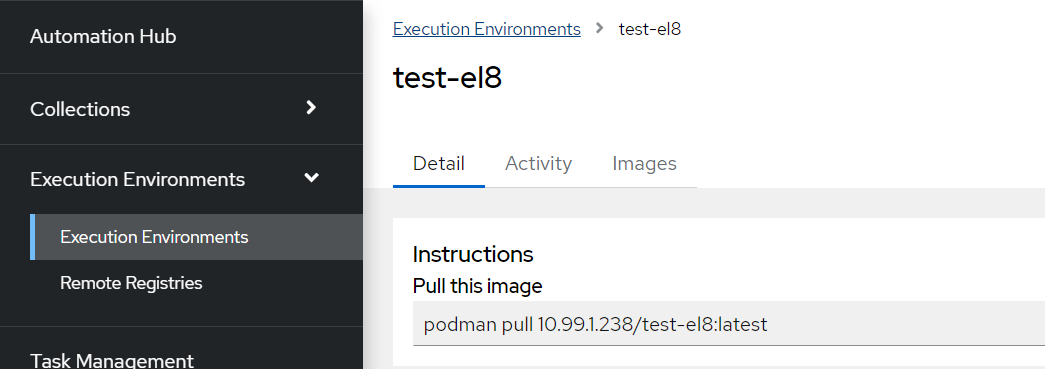
直接在执行机上传镜像到hub

# podman push hub.cof-lee.com/test-el8



然后到 Execution Environments→Execution Environments里查看

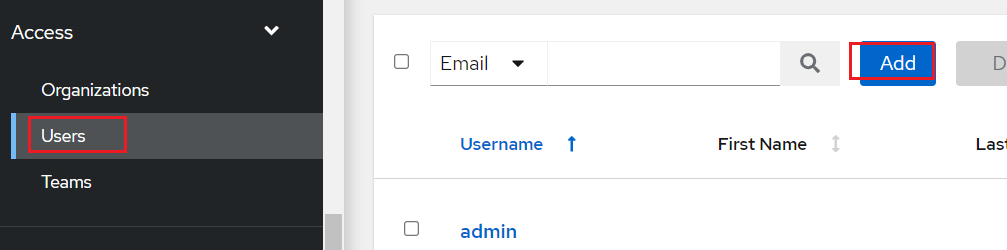


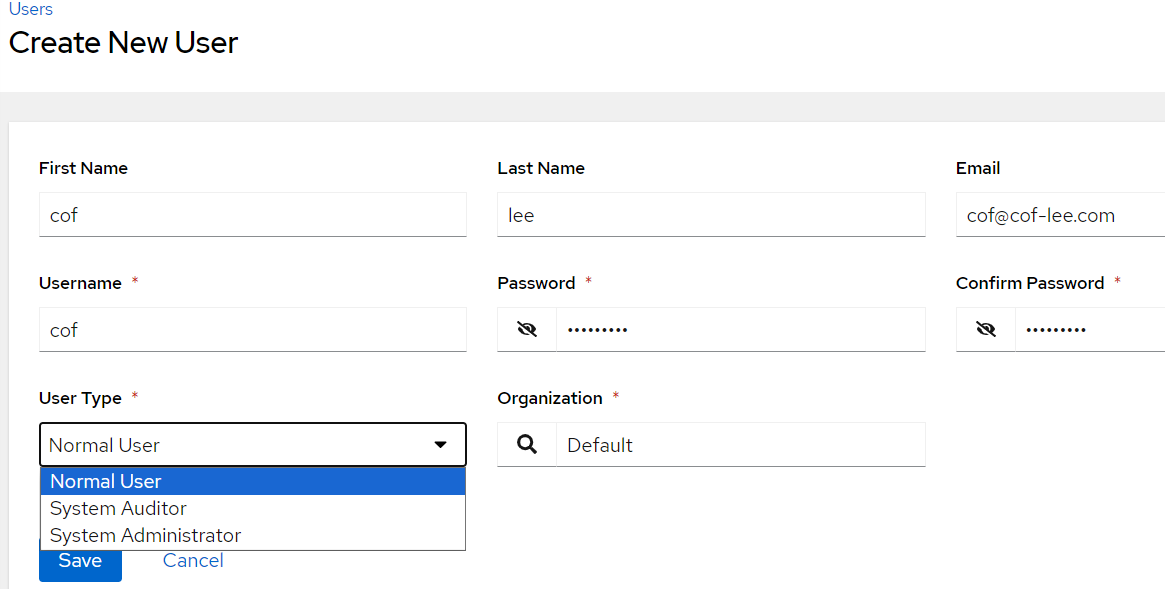


**★第4章、AAP平台的使用**

**★创建用户**

默认已有admin账号，可再创建其他账号，Access→Users→Add

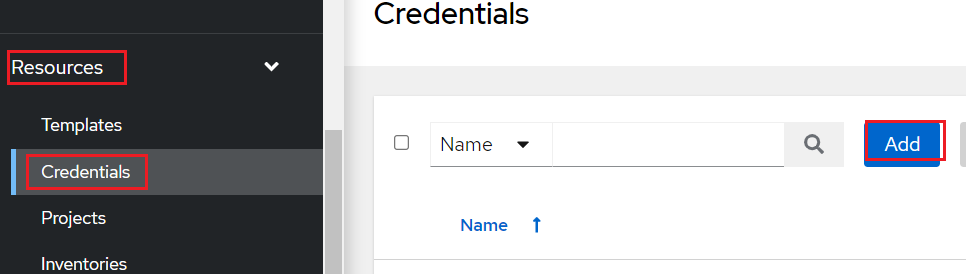


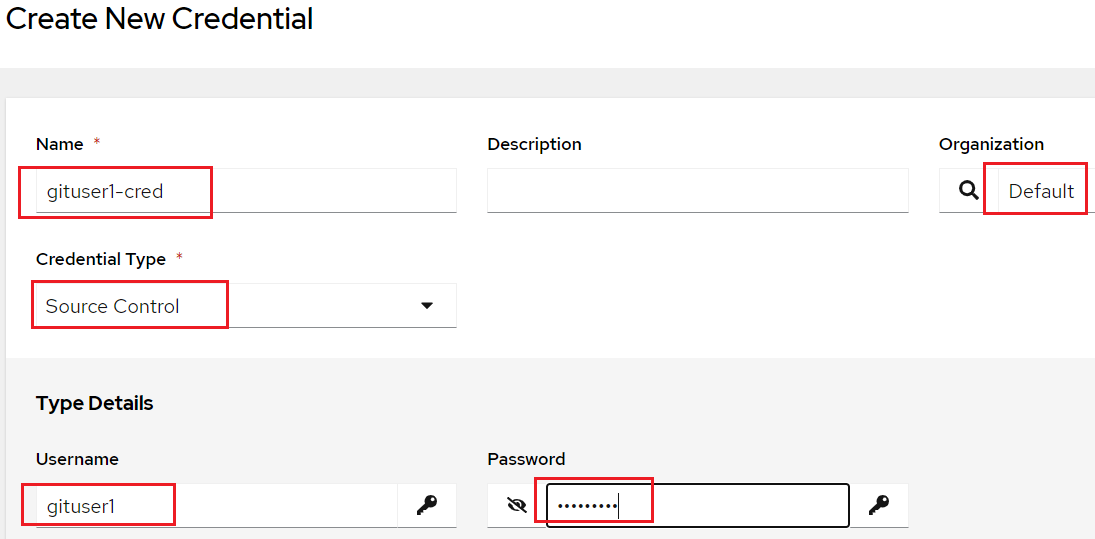


**★创建凭据**

**①创建git用户凭据**

Resources→Credentials→Add

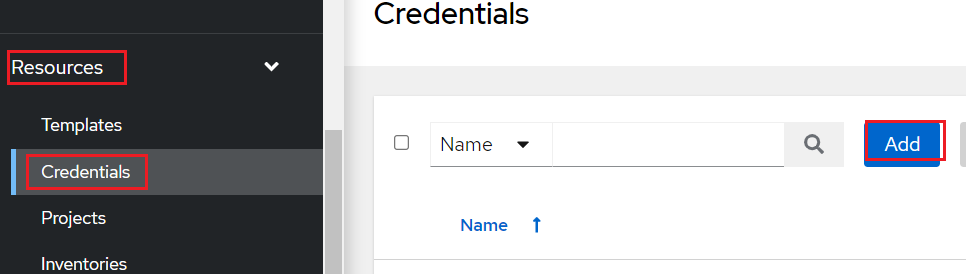


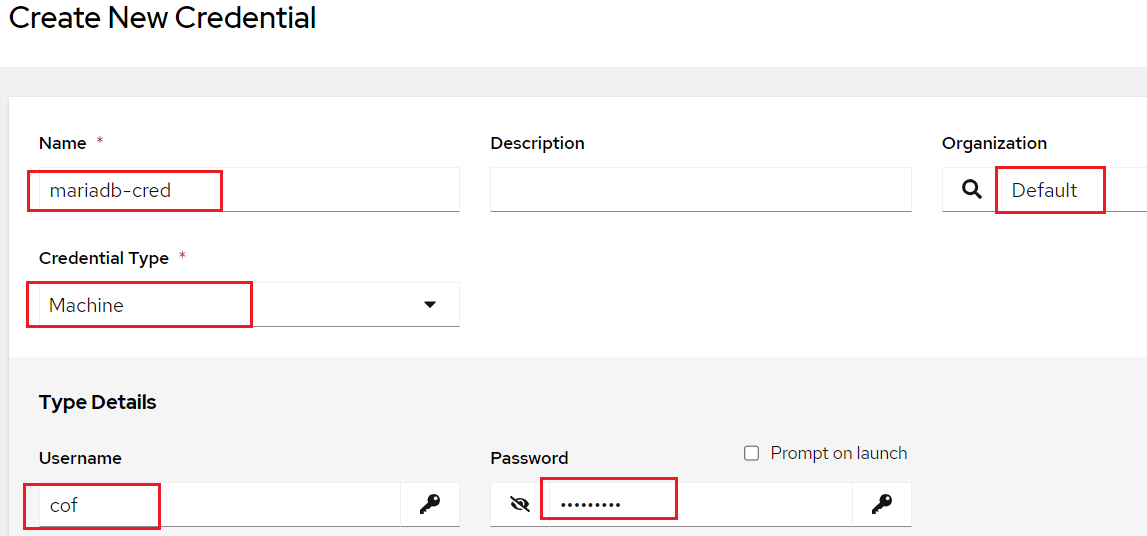


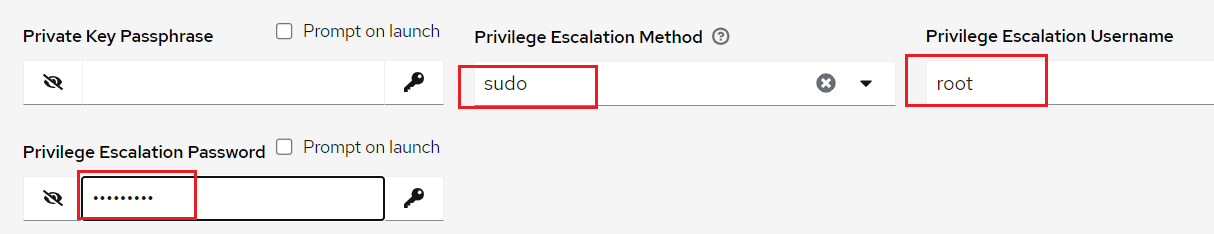
点击“Save”

**②创建受管主机登录凭据**

Resources→Credentials→Add



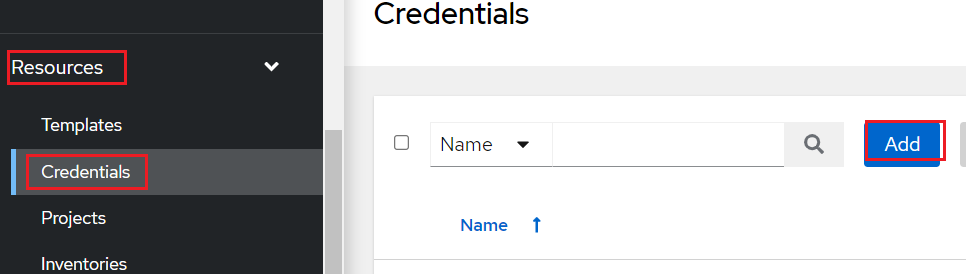


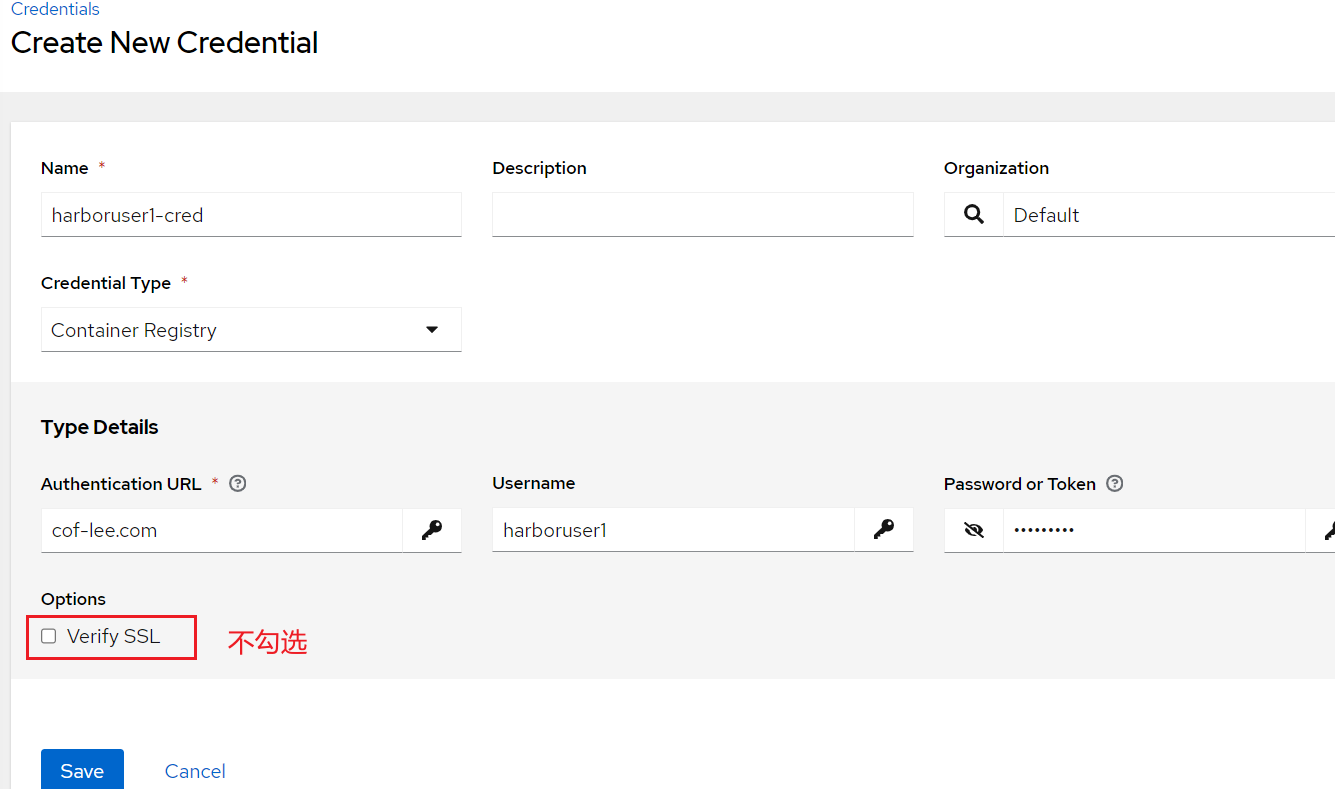


点击“Save”

**③创建docker-registry用户凭据**

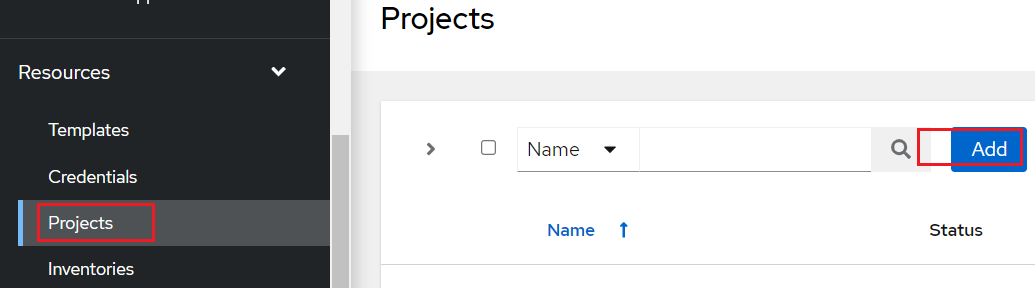
Resources→Credentials→Add



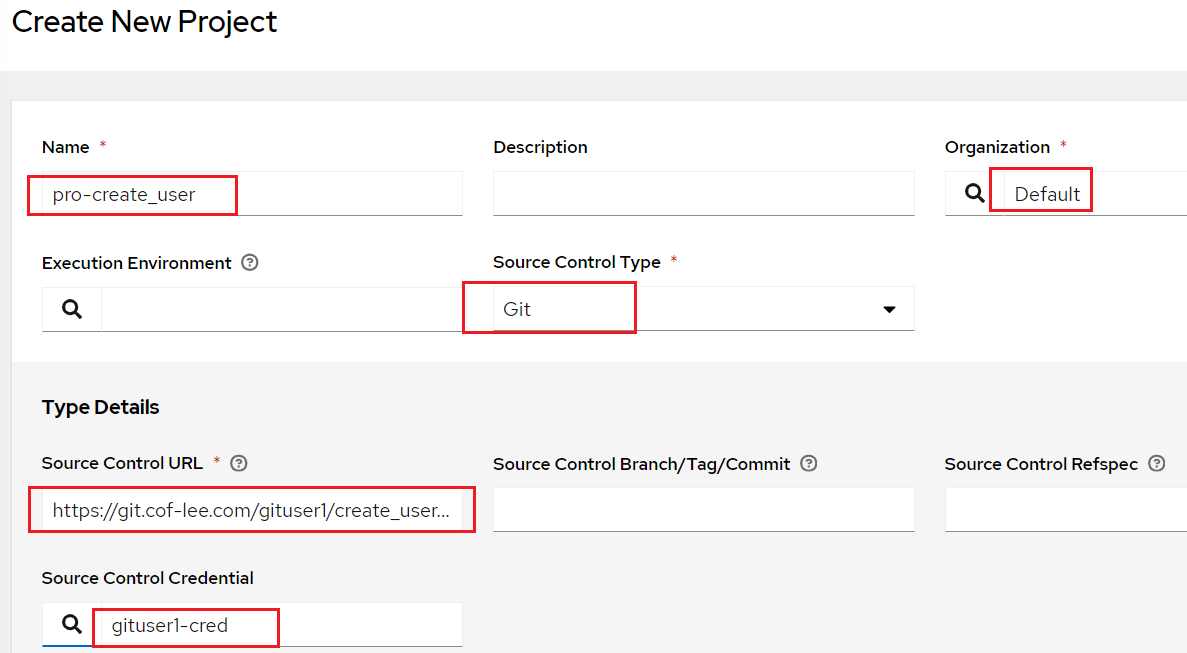


点击“Save”保存

**★创建项目**



Resources→Projects→Add

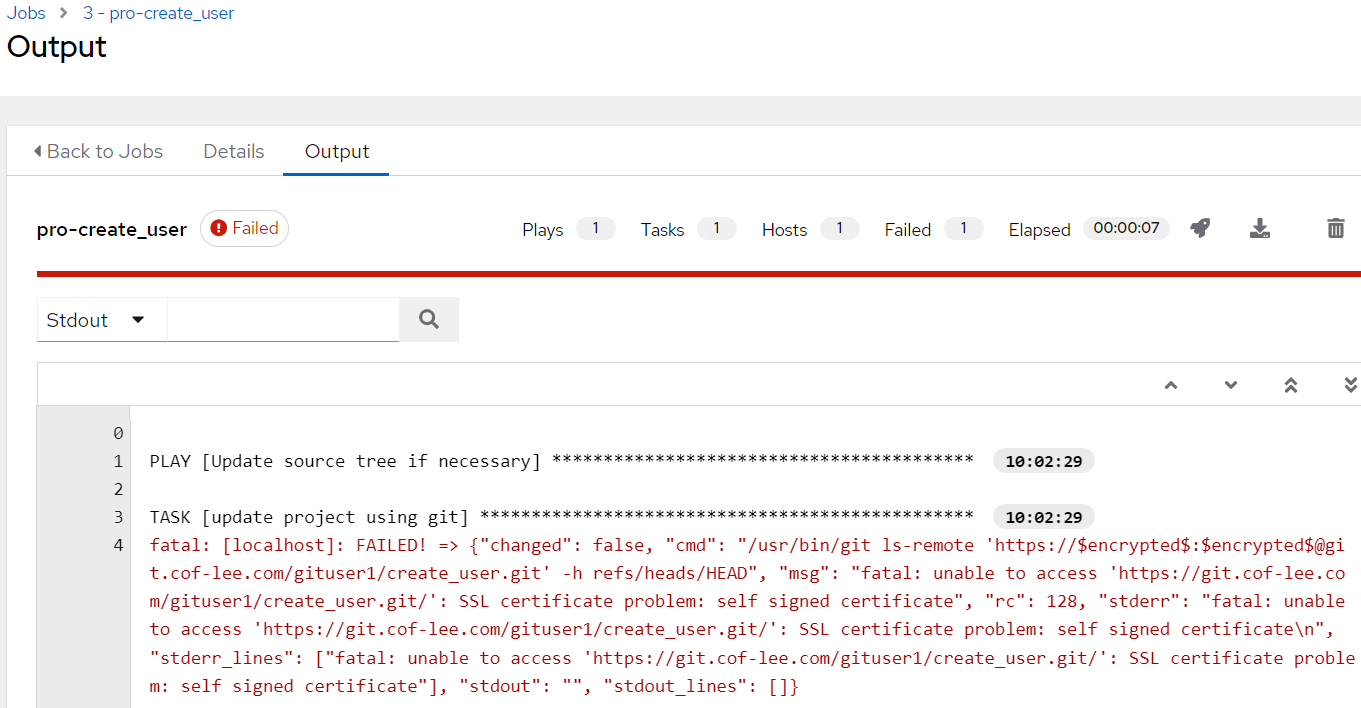


https://git.cof-lee.com/gituser1/create\_user.git

点击“Save”

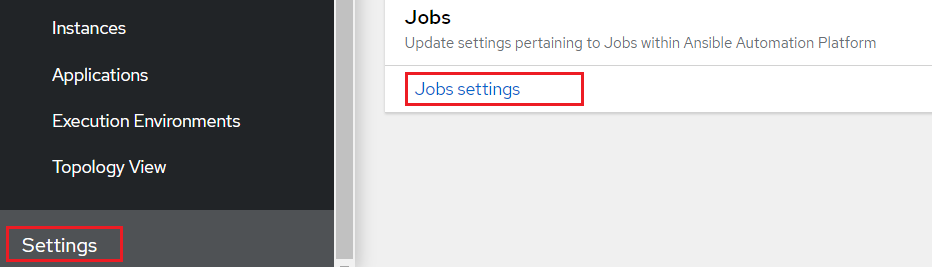
创建完项目后，AAP平台默认会运行一次此项目进行各项检查

如果出现以下报错，则是使用的git仓库为私有仓库，其ssl证书是自签名的

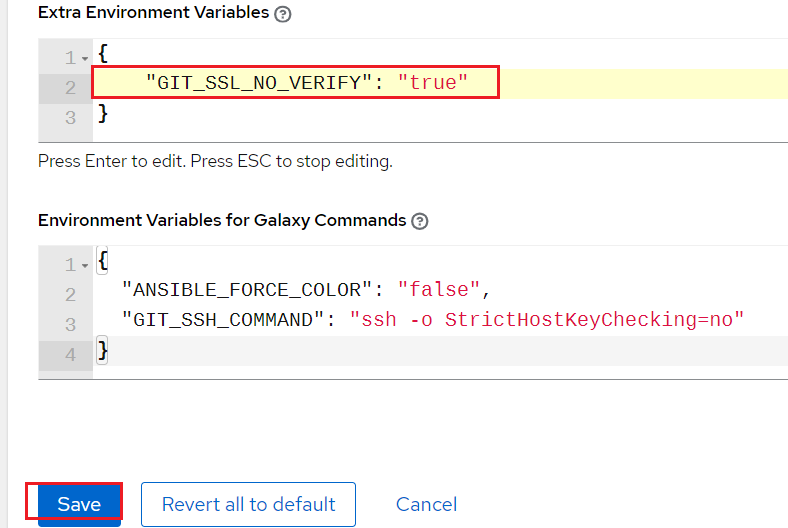


需要让执行更新项目的任务job不校验git仓库的ssl证书，方法如下：

Settings→Jobs settings



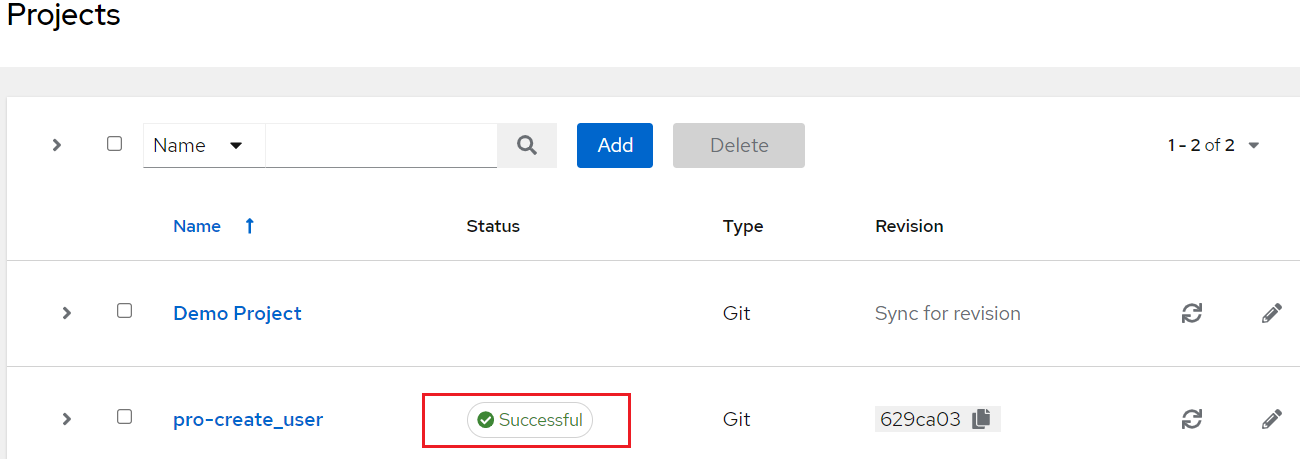
编辑设置：



在“Extra Environment Variables”里新增一条配置："GIT\_SSL\_NO\_VERIFY": "true" 表示不校验git仓库的ssl证书

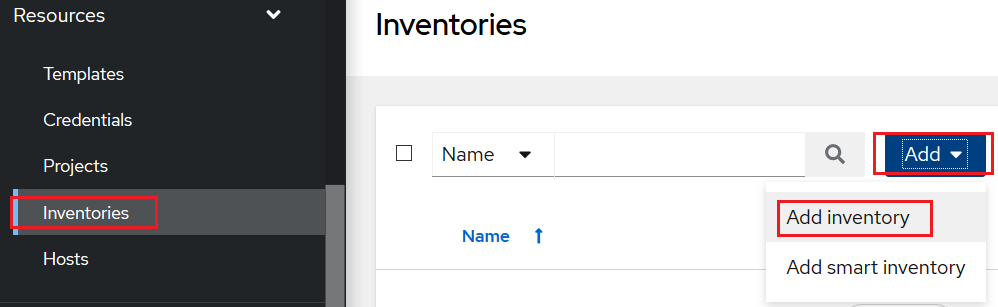
点击“Save”保存

再次点击测试，测试通过

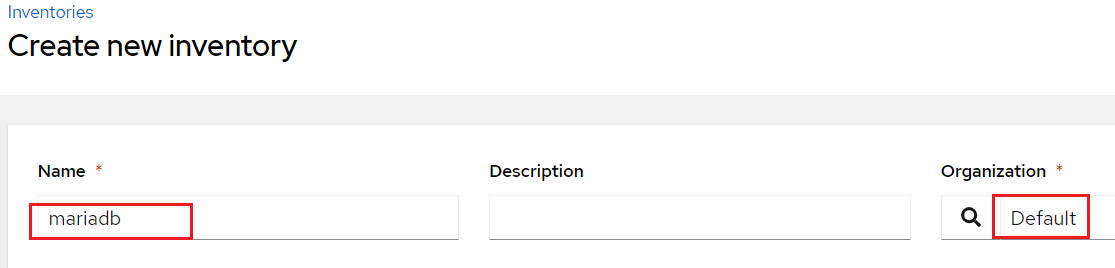


**★创建主机清单**

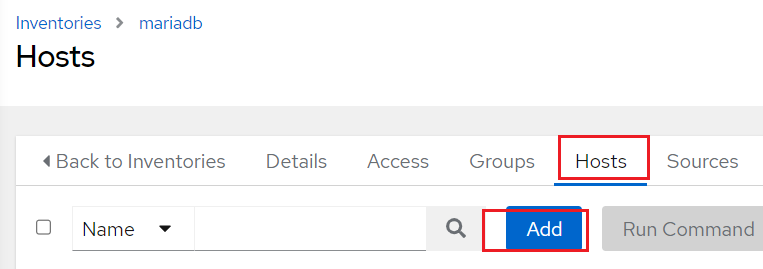
①创建静态主机清单

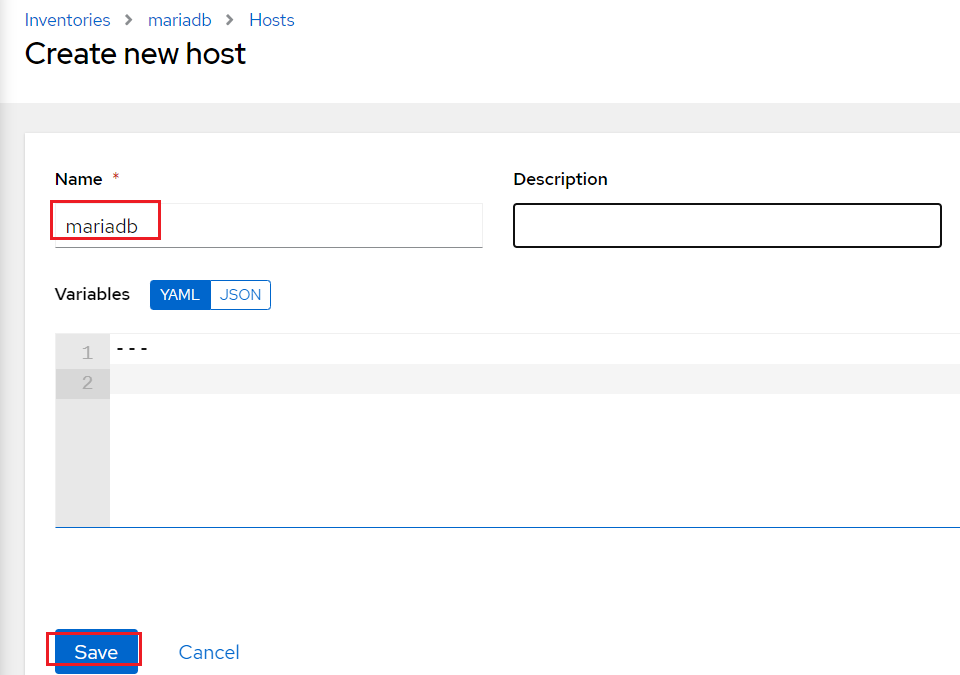


Resources→Inventories→Add→Add inventory



Save保存后可添加groups及hosts：

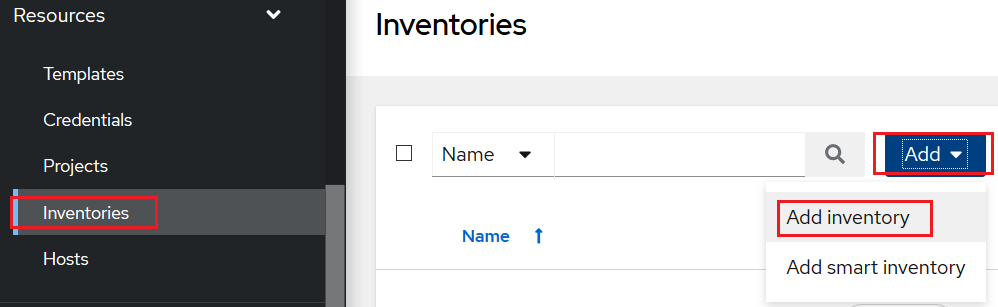




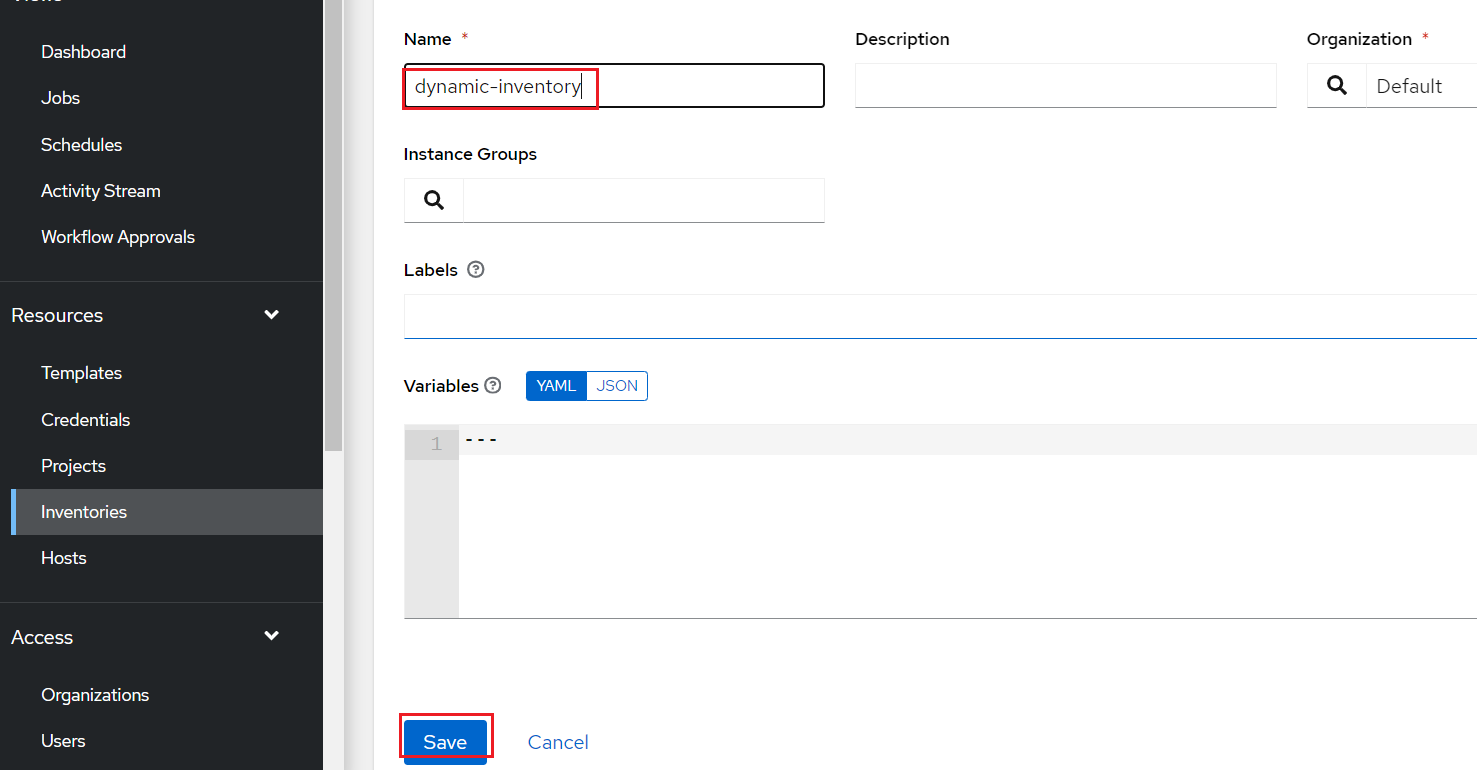
主机name可写主机名称，也可写ip地址

点击“Save”保存

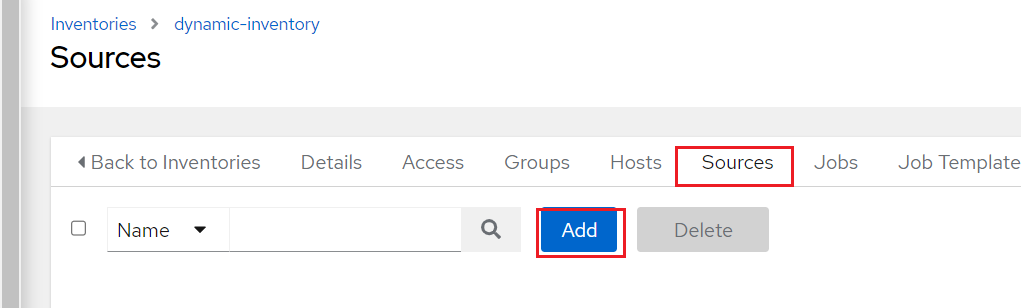
**②创建动态主机清单**



Resources→Inventories→Add→Add inventory

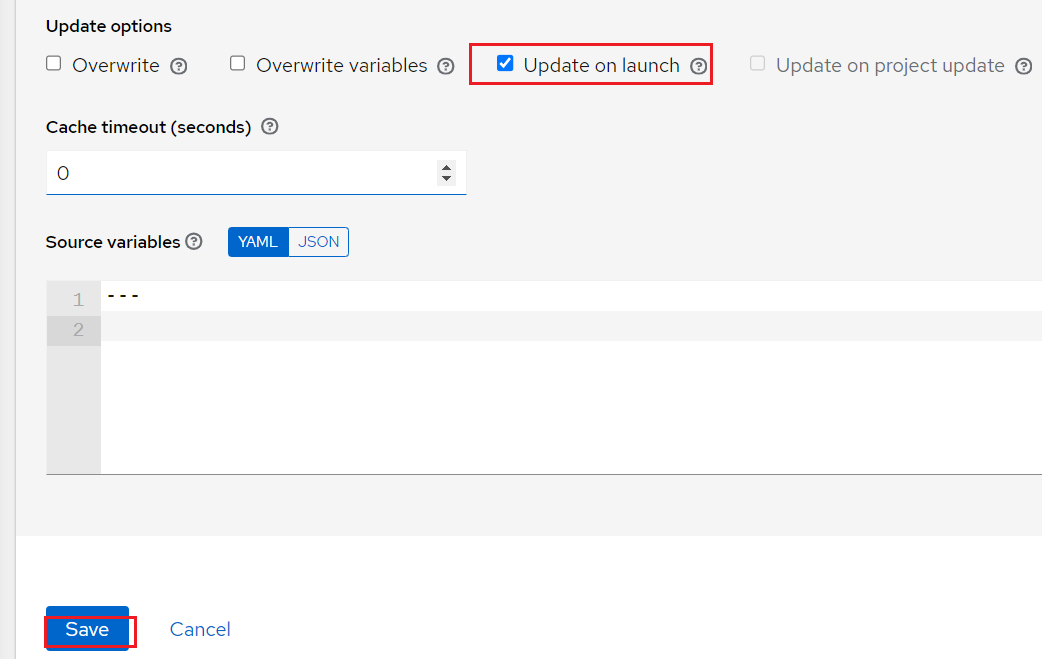


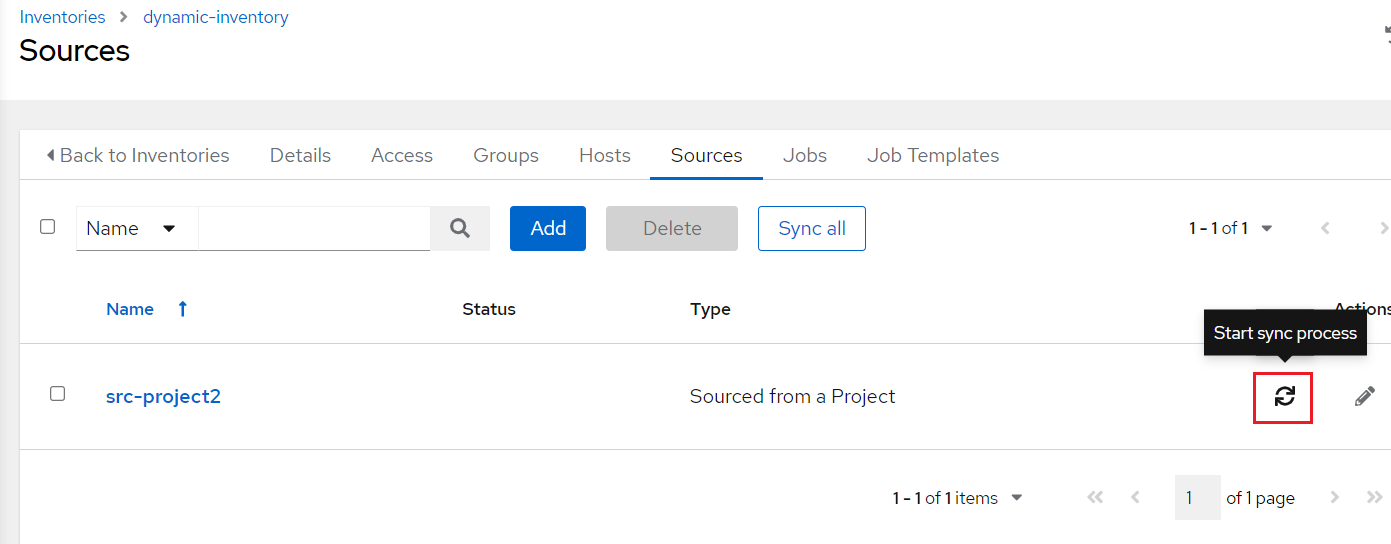
创建之后，在Sources里添加源



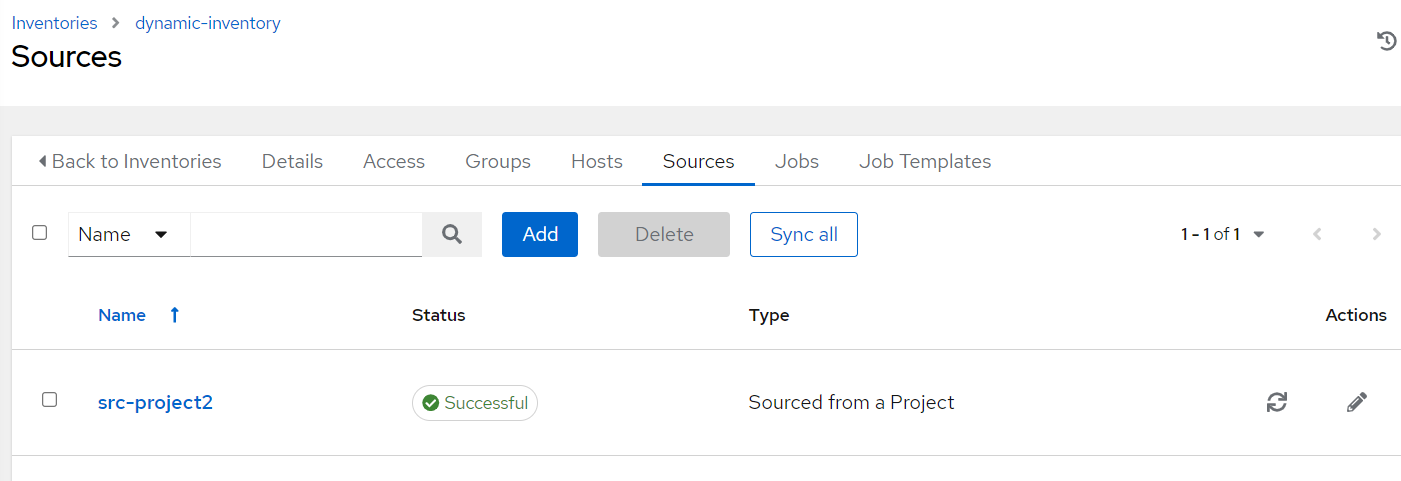


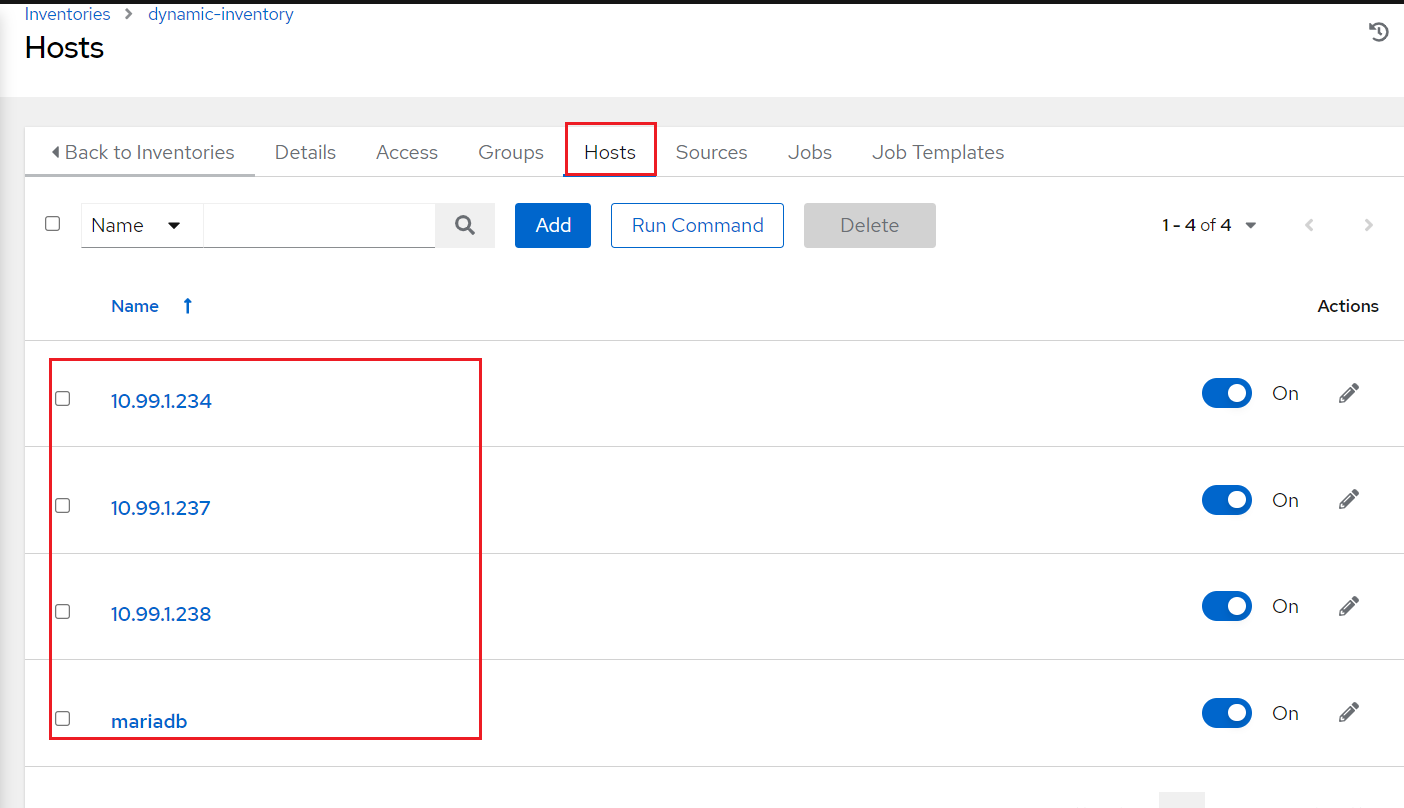
从项目里获取动态inventory文件



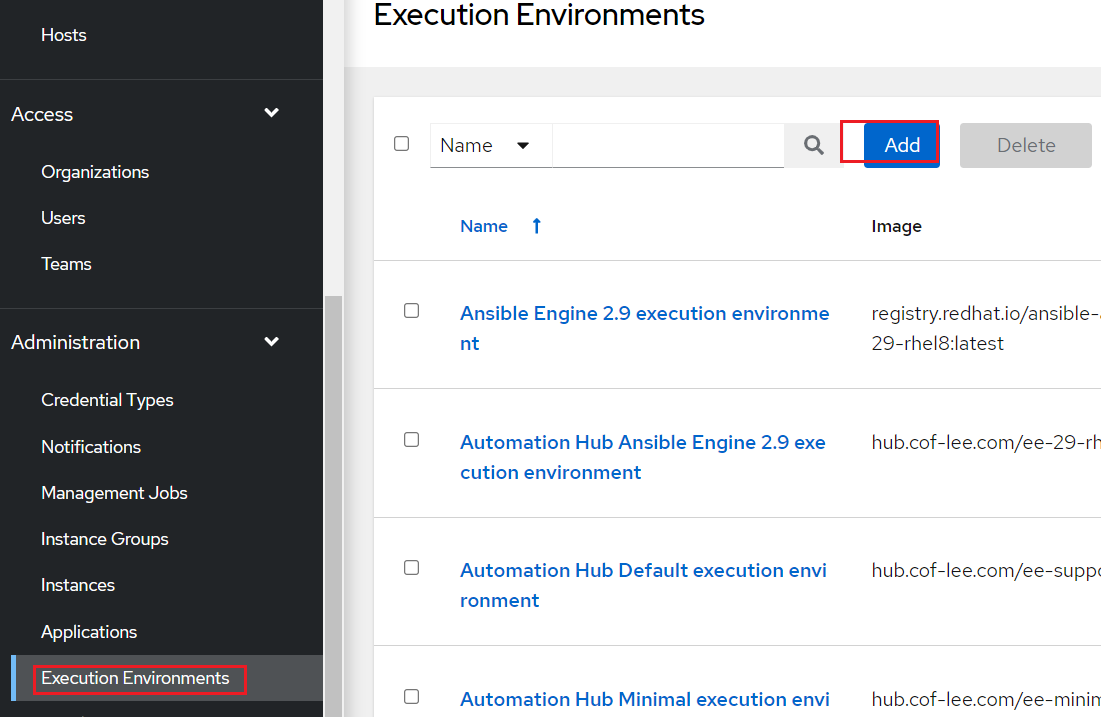


添加源之后，需要同步一下，同步成功才会有主机信息

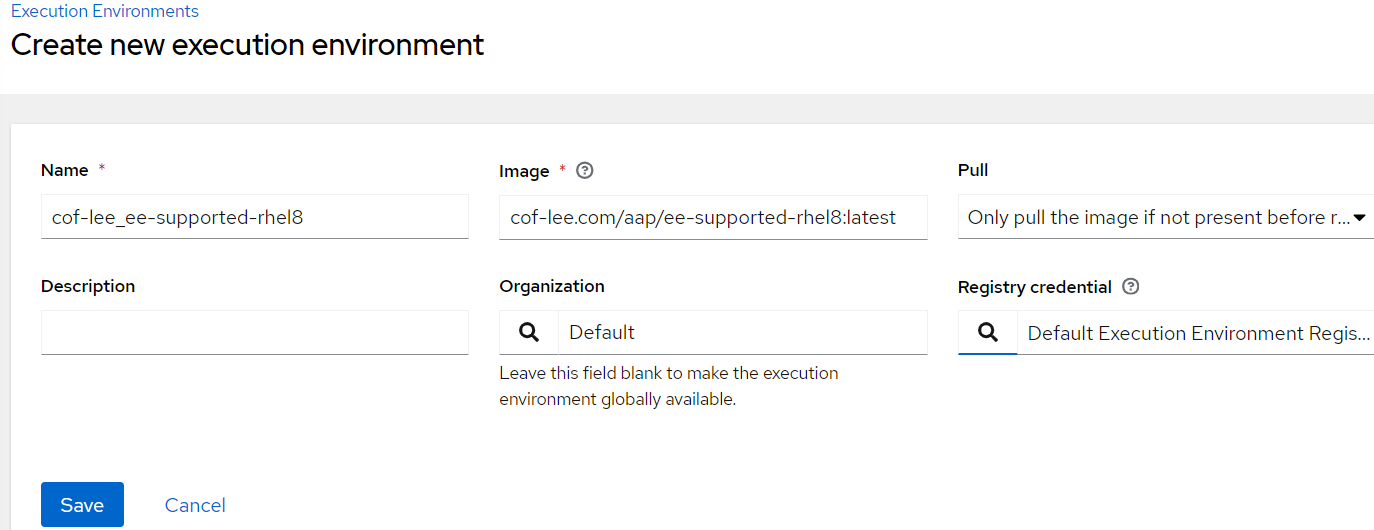




**★新增eei执行环境镜像**



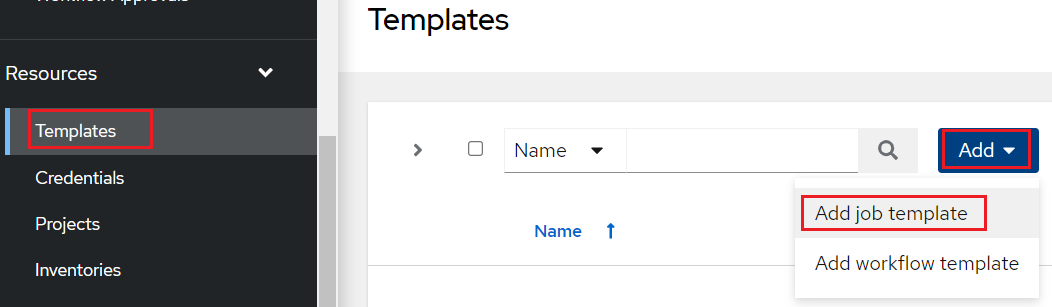
Administration→Execution Environments→Add



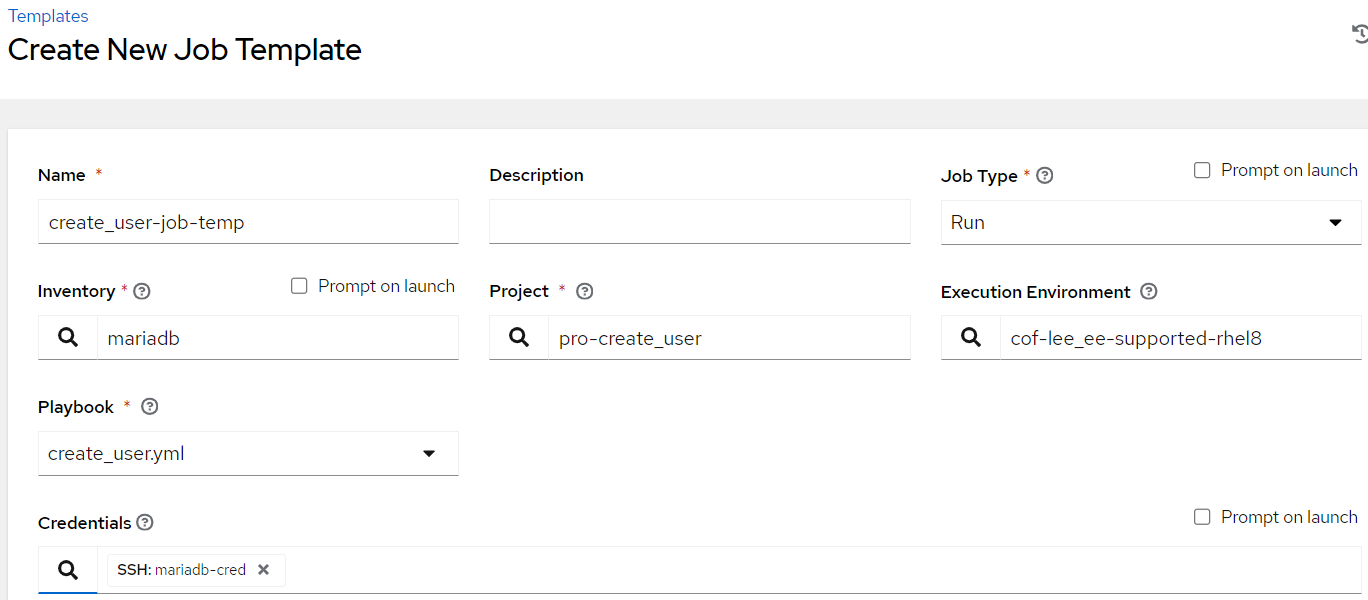
镜像为：cof-lee.com/aap/ee-supported-rhel8:latest （如果是私有仓库配置了ssl，需要在控制器节点上配置podman信任私有镜像仓库）

点击“Save”保存

**★创建作业模板**

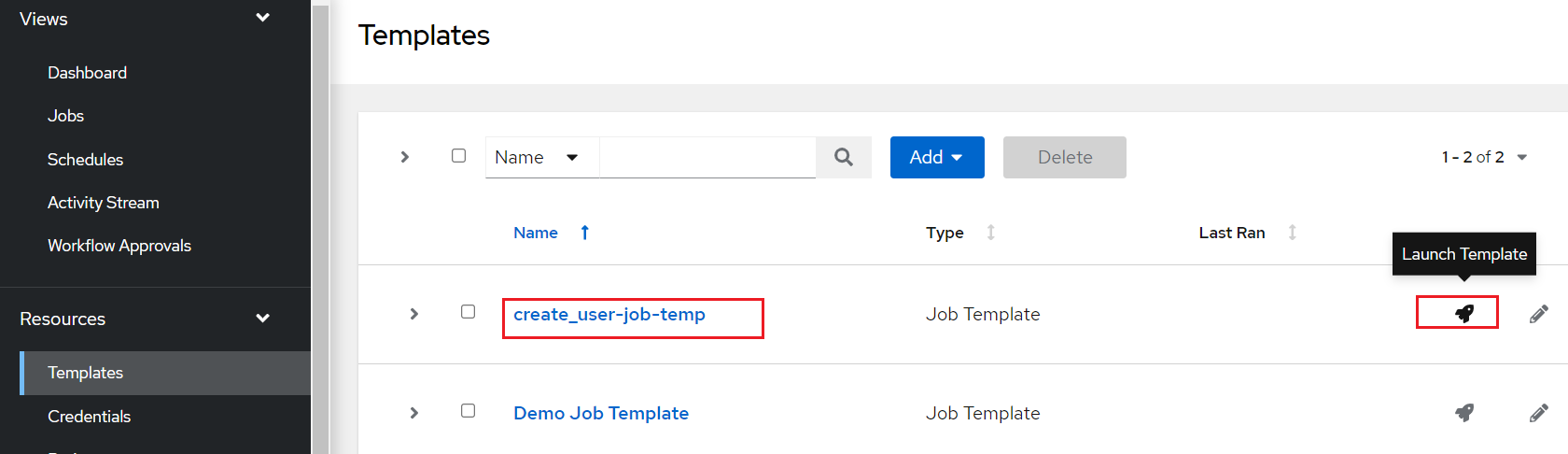


Resources→Templates→Add→Add job template



点击“Save”保存

★运行作业模板



如果报错如下：



原因是此eei镜像存储在本地镜像仓库里，使用的ssl证书为自签名证书，配置podman信任此镜像仓库即可

# cat > /etc/containers/registries.conf <<EOF

unqualified-search-registries = [ "docker.io" ]

[[registry]]

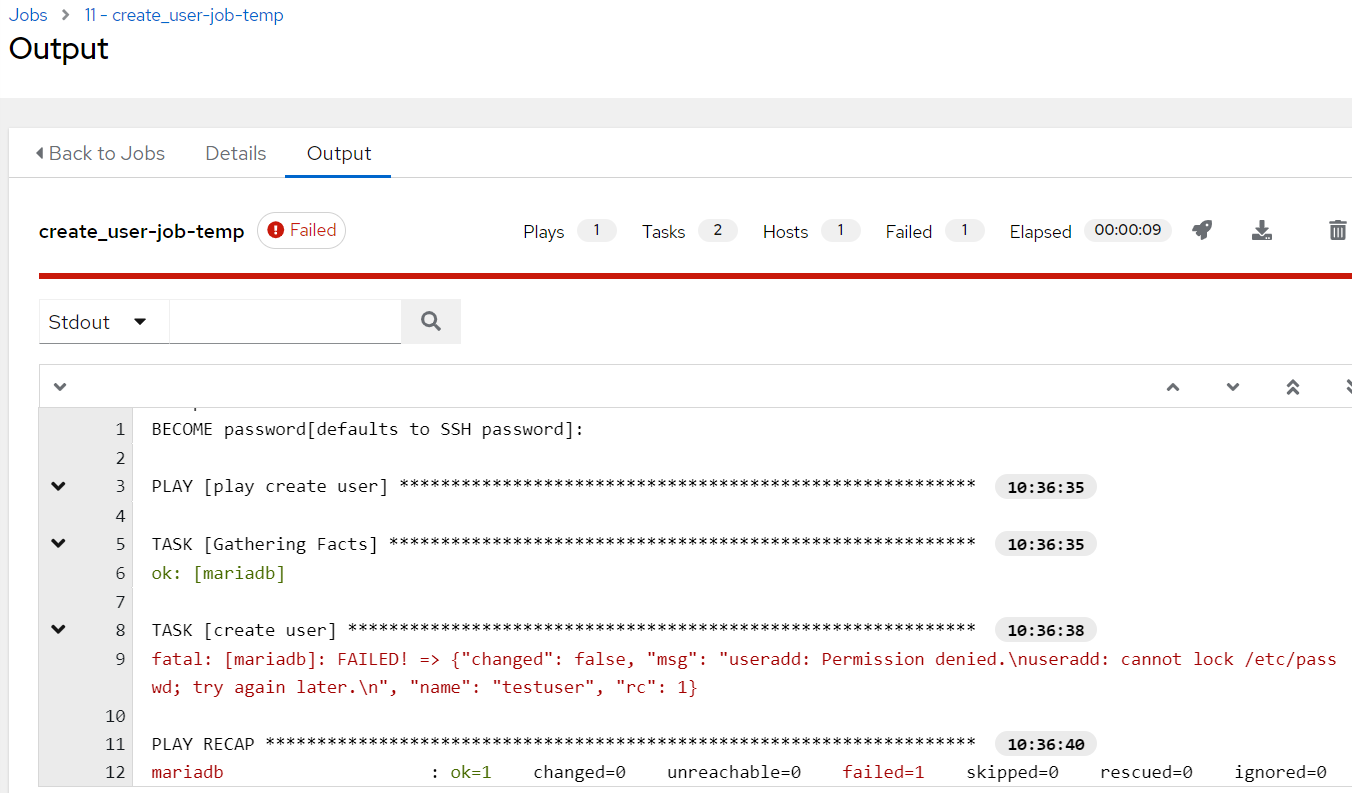
prefix = "cof-lee.com"

location = "cof-lee.com"

insecure = true

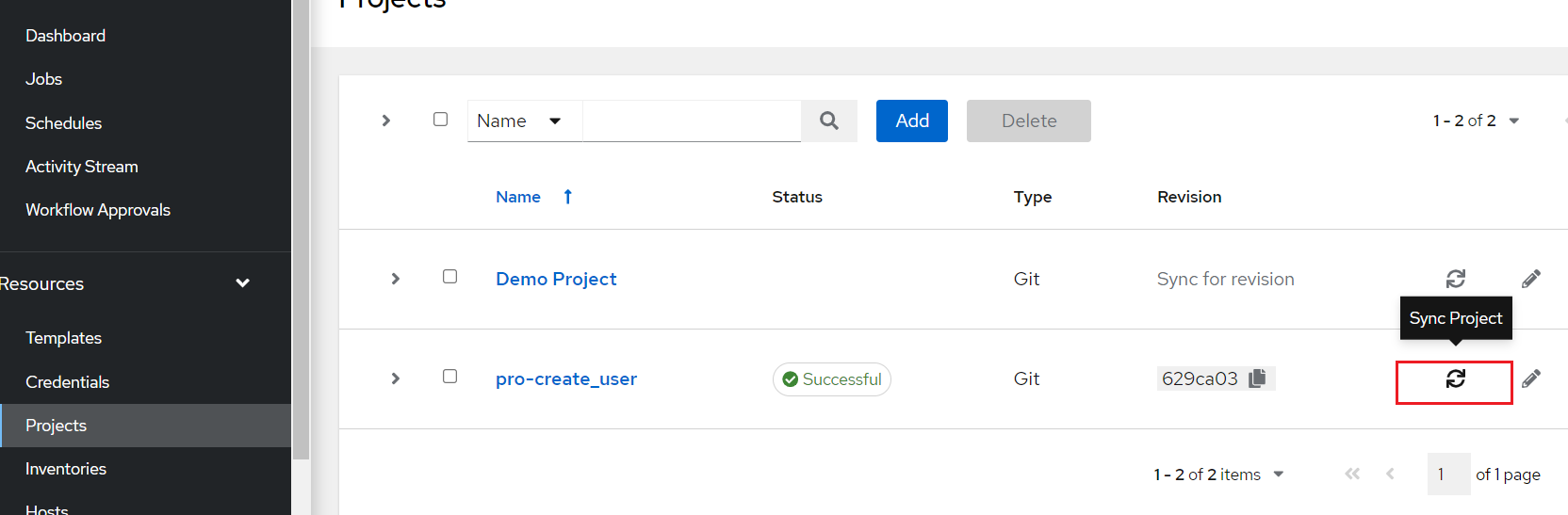
EOF

再次运行模板：

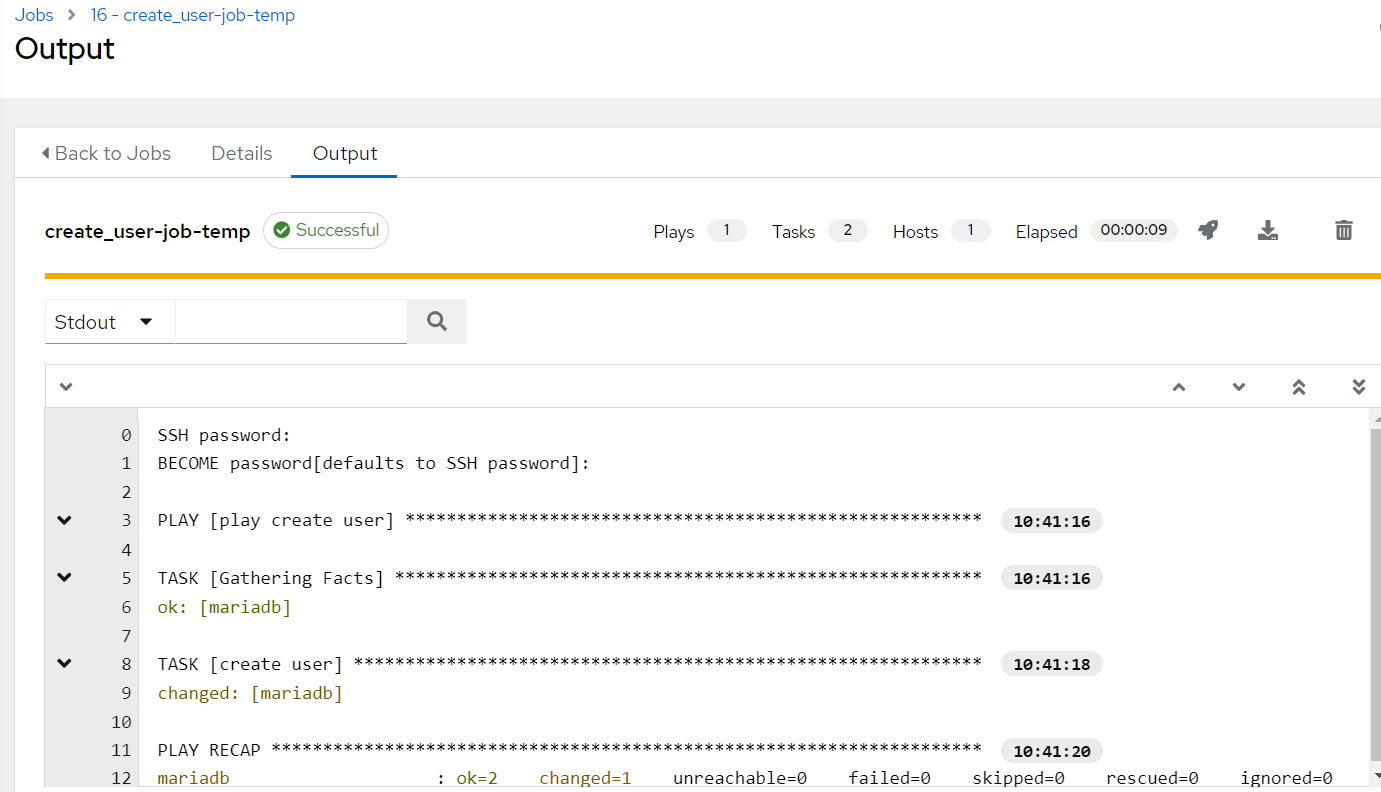


原因是使用AAP平台执行的playbook剧本，默认提权时要求手动输入become\_pass的，即become\_ask\_pass=true

需要在ansible.cfg里添加become\_ask\_pass=false配置项，再更新git仓库，最后在AAP平台同步此项目

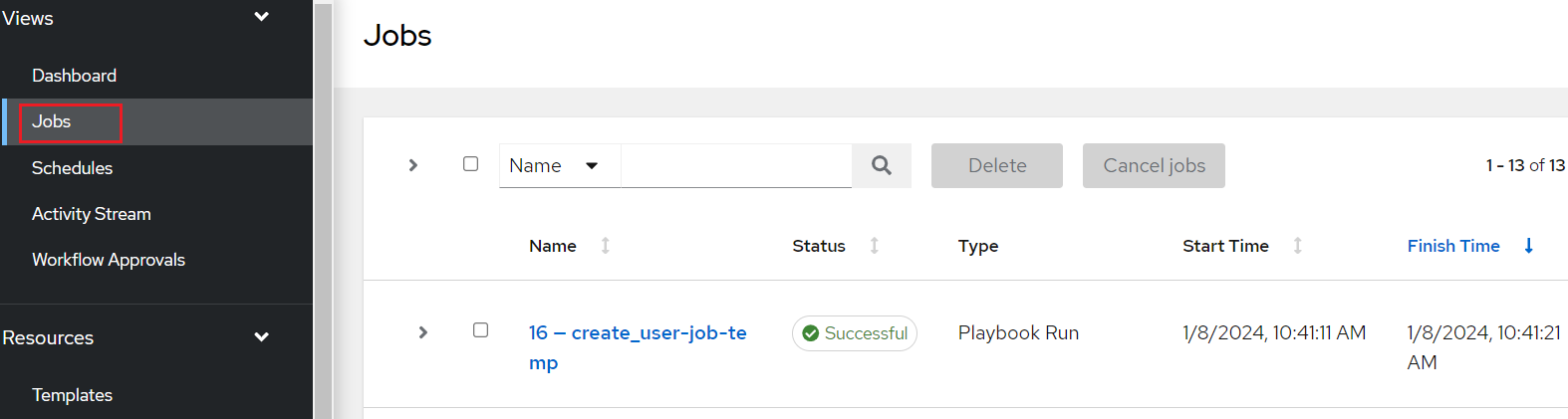


再次执行作业模板：



运行成功

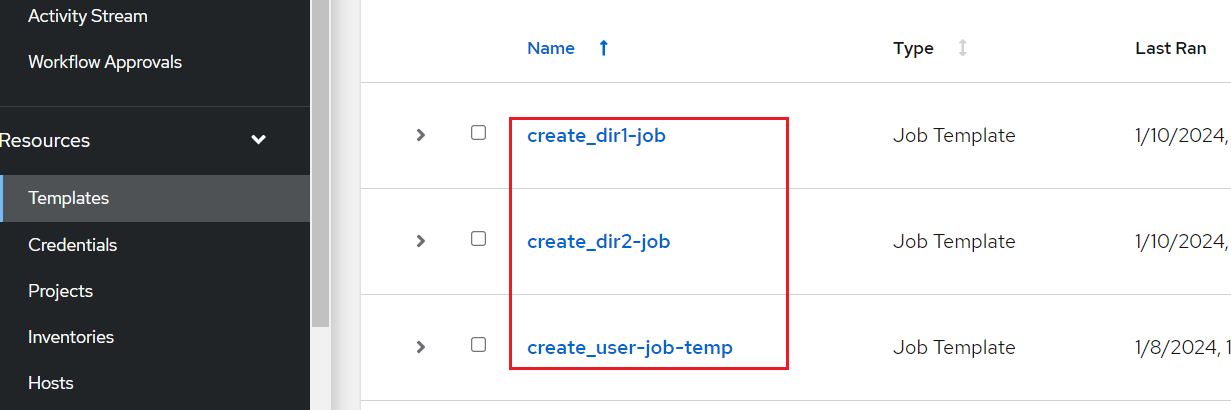
每次运行作业模板，都会生成一个job，可在以下地方查看：Views→Jobs



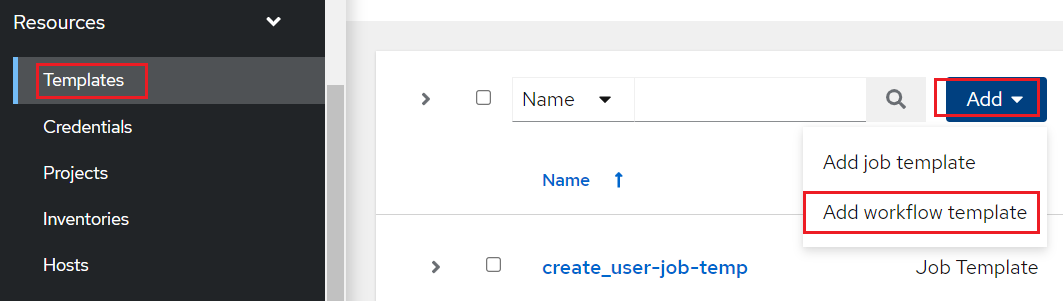
**★创建工作流模板**

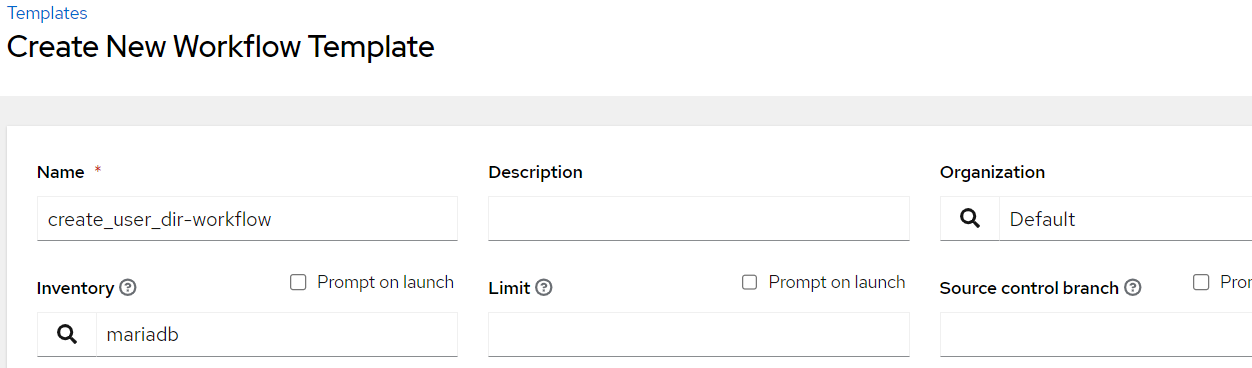
前面创建的工作模板是针对单个作业任务的，假如要执行一系列的作业任务，相互之间有依赖或者说需要按顺序及某些条件判断执行这些作业，这时可创建工作流（workflow）模板

现在有3个作业模板，想创建一个工作流程，首先执行“create\_user-job-temp”，如果此作业运行成功则继续运行“create\_dir1-job”；如果“create\_user-job-temp”运行失败，则继续运行“create\_dir2-job”

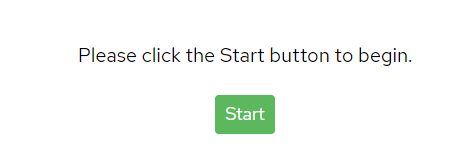


Resources→Templates→Add→Add workflow template

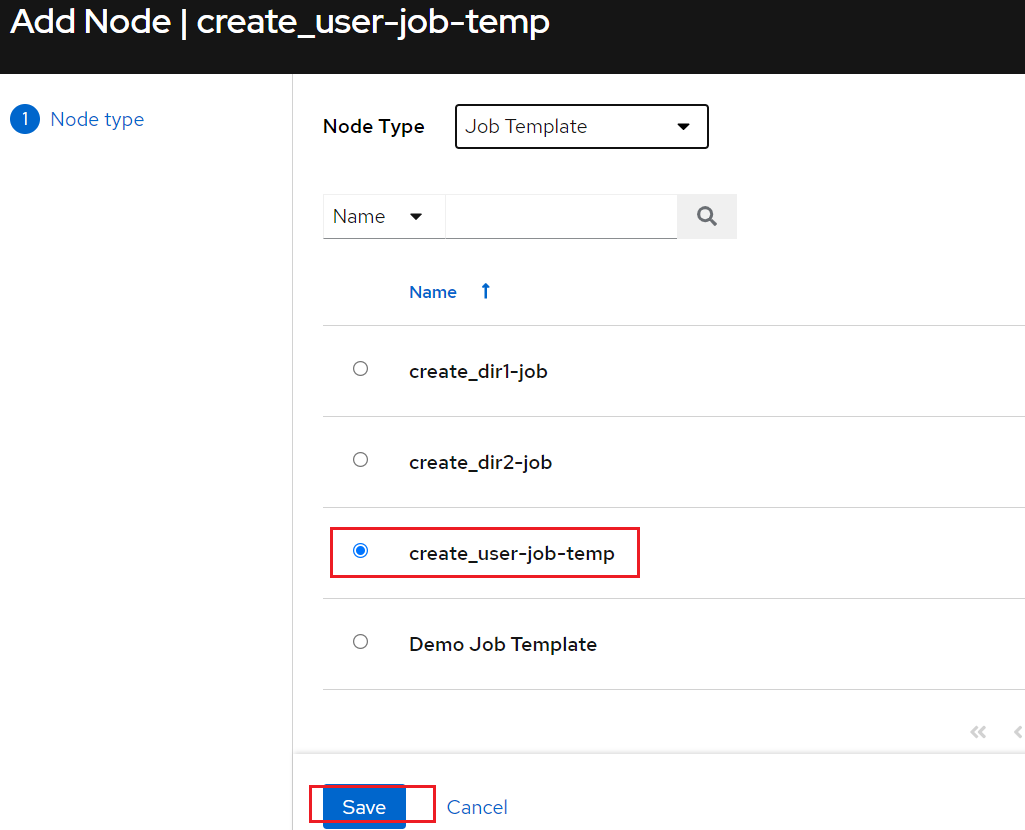


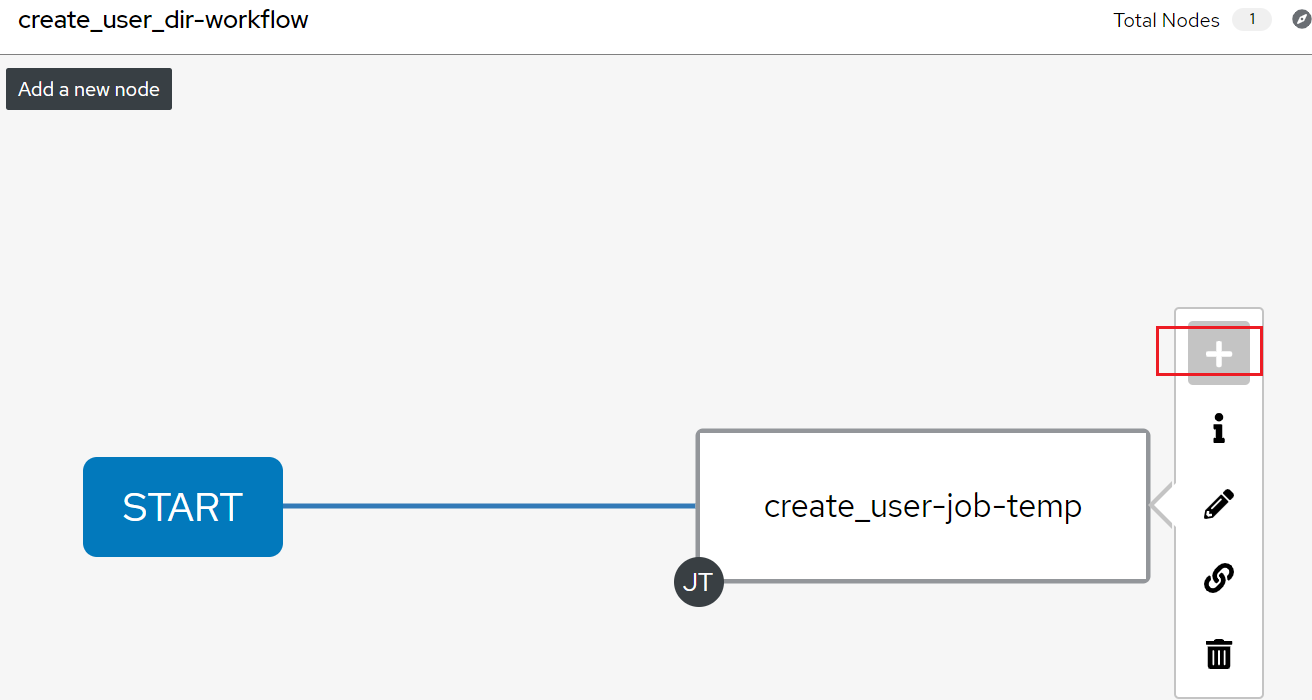


点击“Save”后：

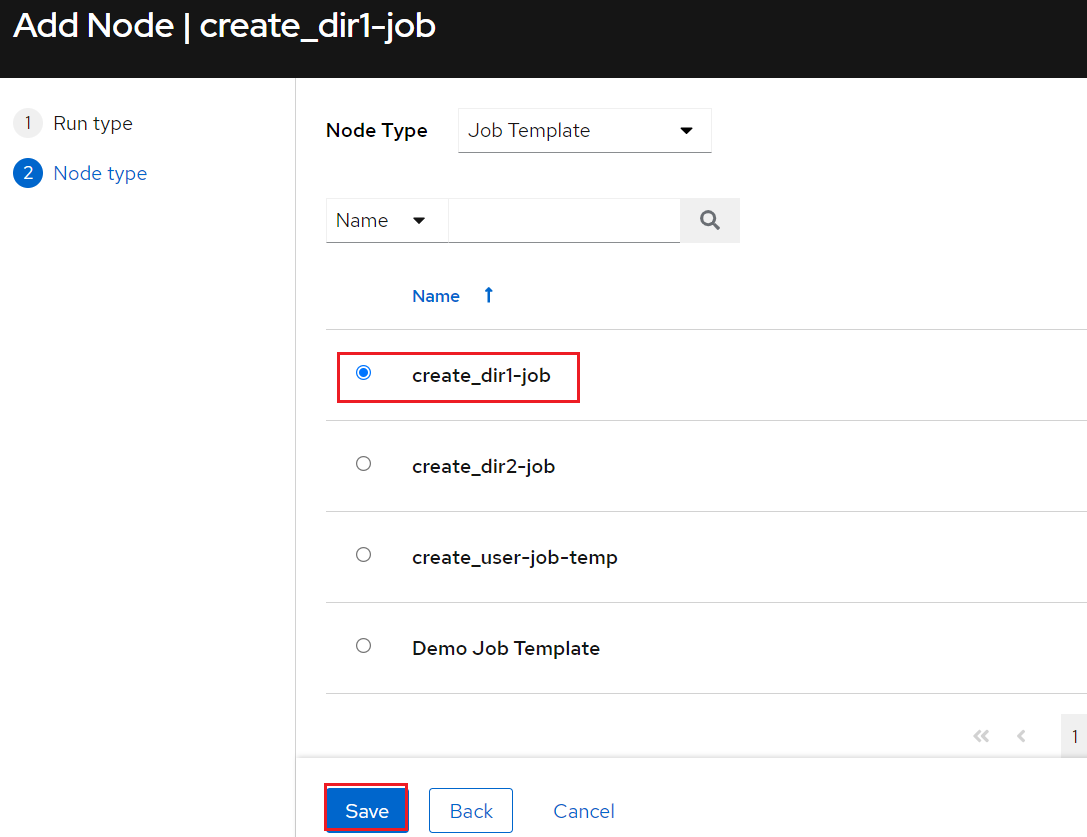


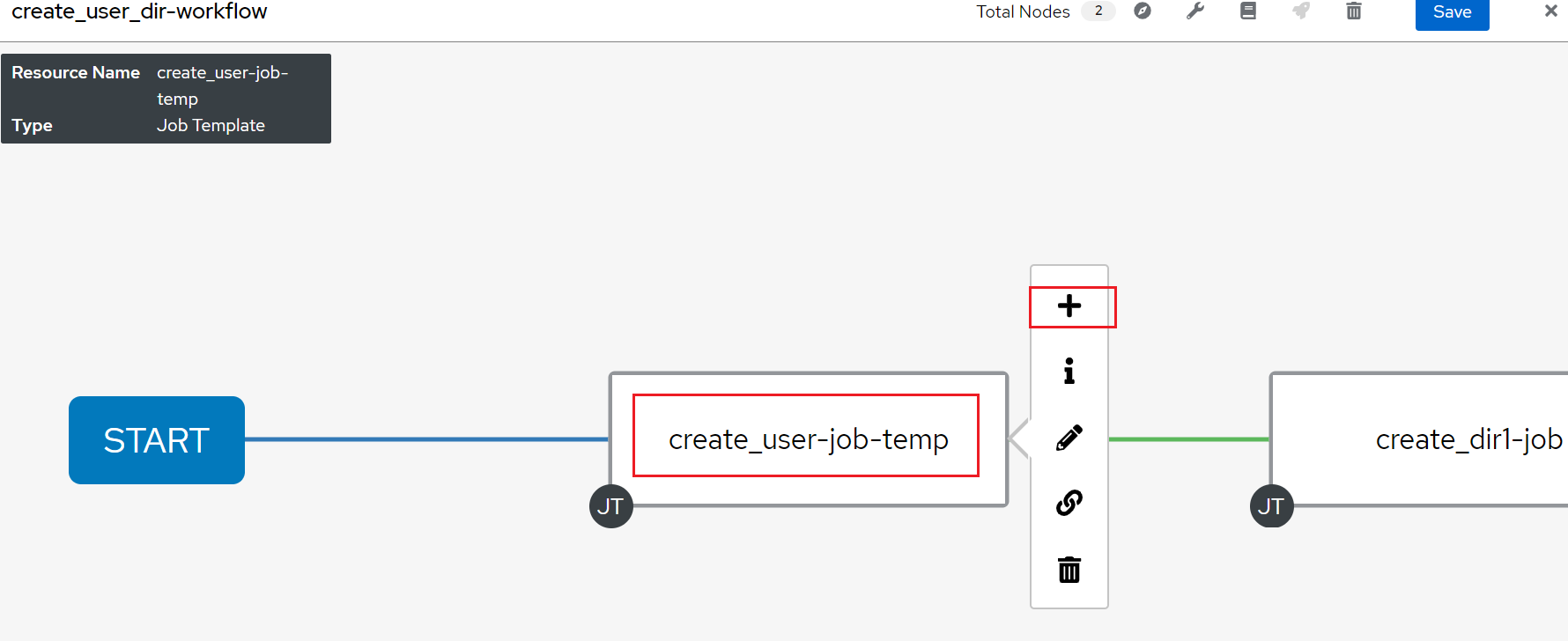
点击“Start”后进入工作流编辑界面：

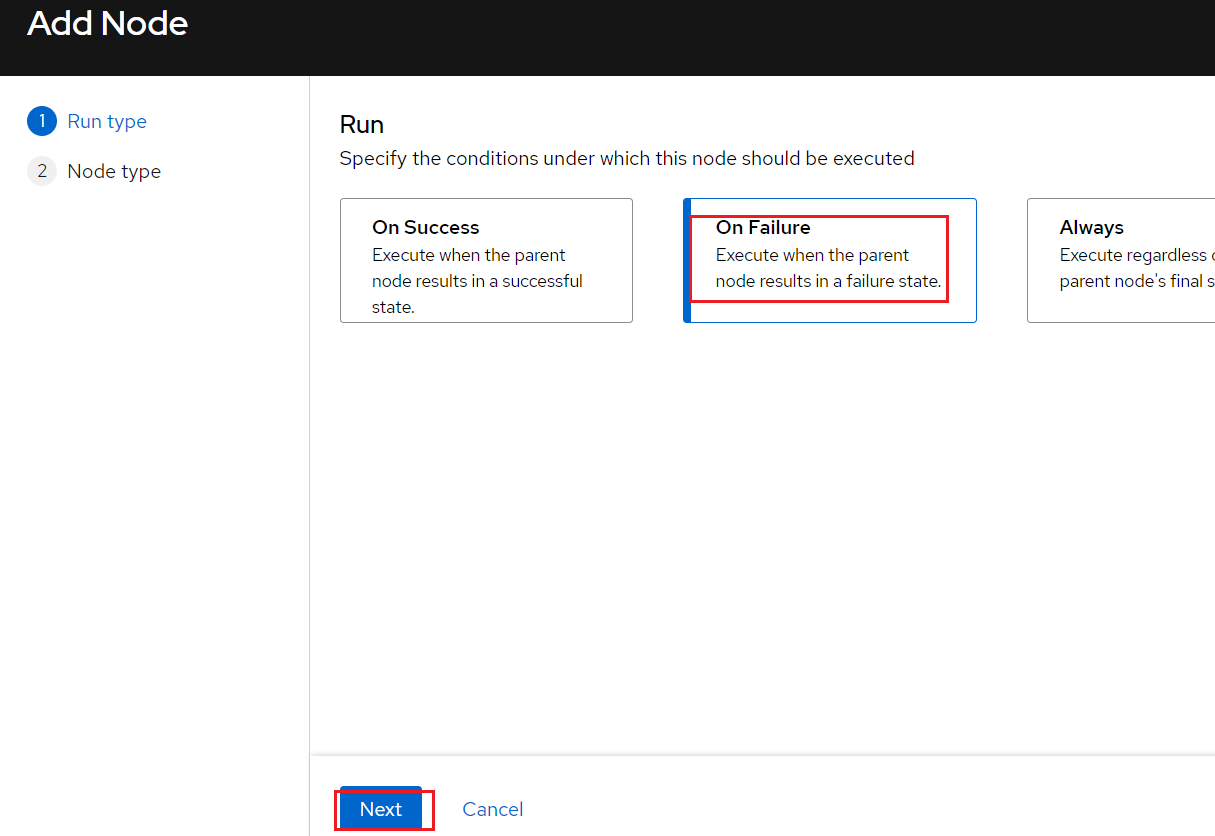


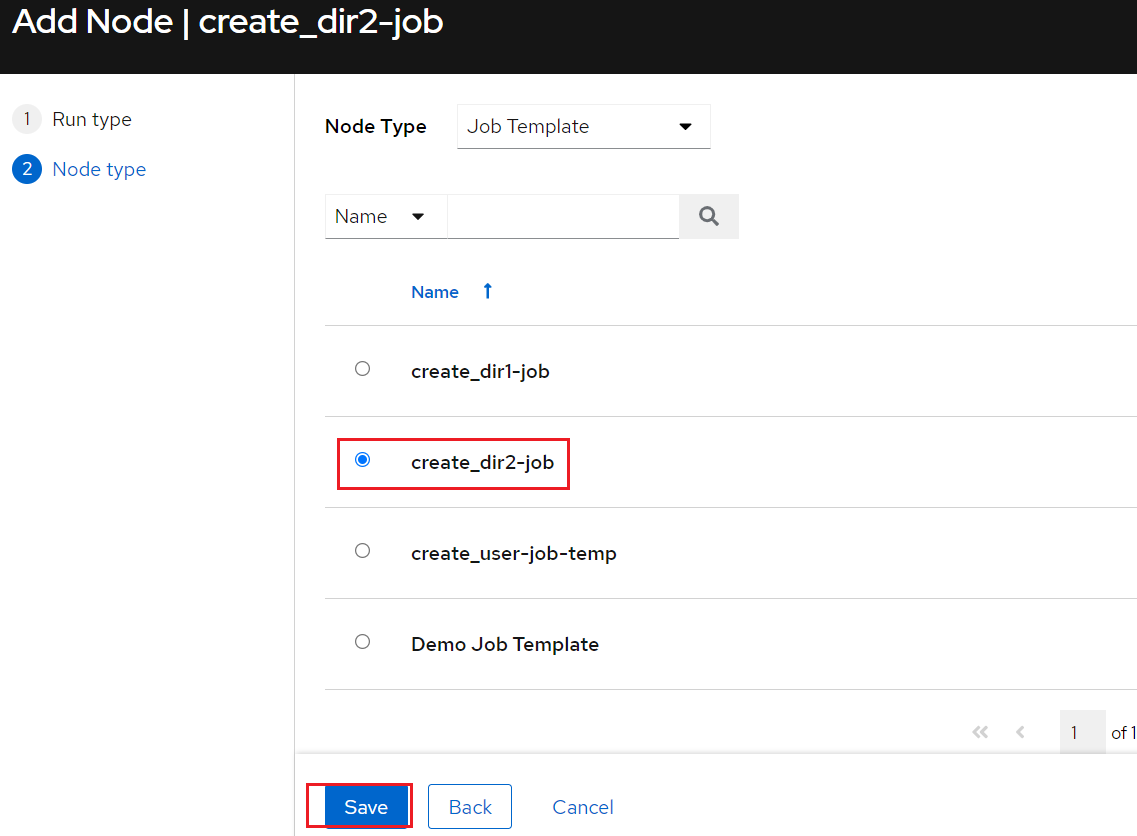


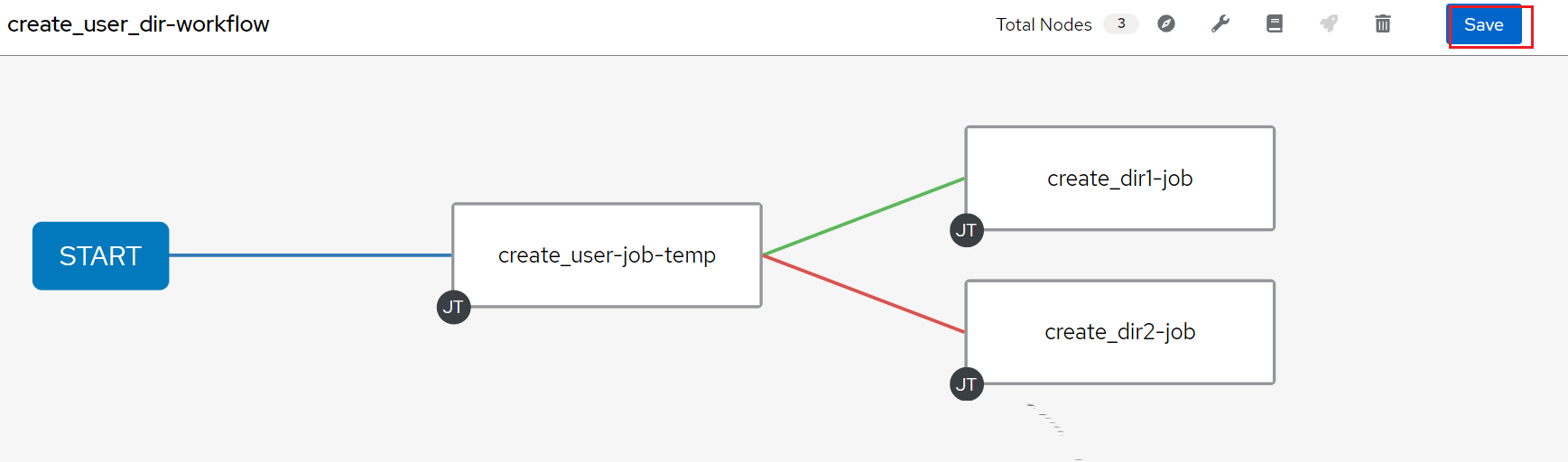












**执行工作流：**

