### 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：2023-11-14

**★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，这使得管理运行ansible playbook所需的各种模块的依赖变得困难，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虚拟环境来管理它们。于是后来开发了**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。

**★安装前准备**

服务器配置至少要 4核cpu，16GB内存，20GB剩余磁盘空间（/var/lib/awx存储目录）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

# vi /etc/chrony.conf

server 10.99.1.1 iburst prefer #配置可访问的ntp服务器

# 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 localhost

**★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

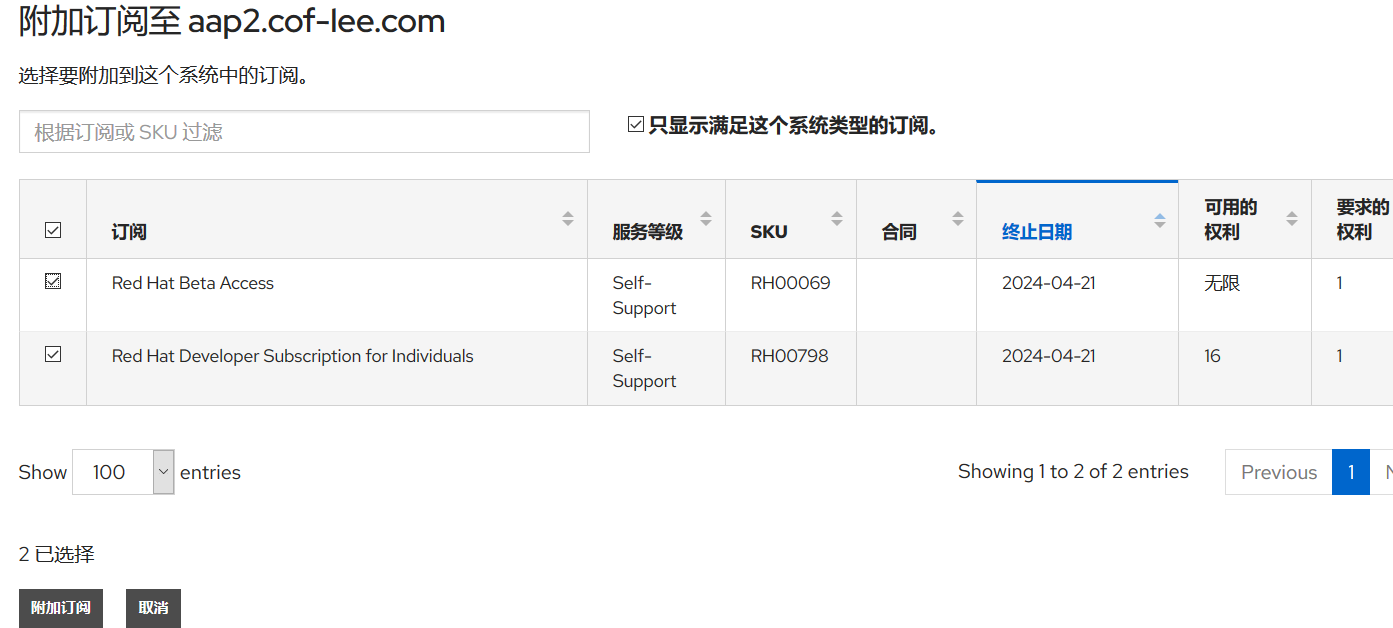
注册系统后，可到官网查看系统注册情况： https://access.redhat.com/management/systems



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

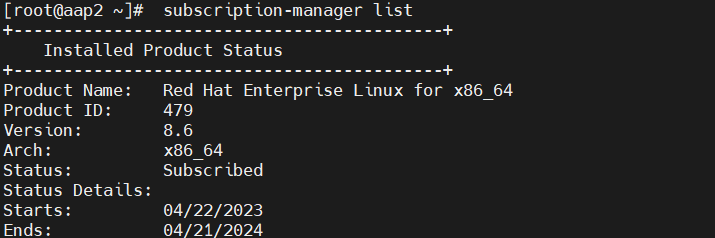
# subscription-manager list --available --all | grep -i "Ansible Automation Platform" #查看订阅中是否有aap

Red Hat Ansible Automation Platform

# 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



安装ansible-navigator软件包

# yum install ansible-navigator -y

# ansible-navigator --version

ansible-navigator 2.1.0

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

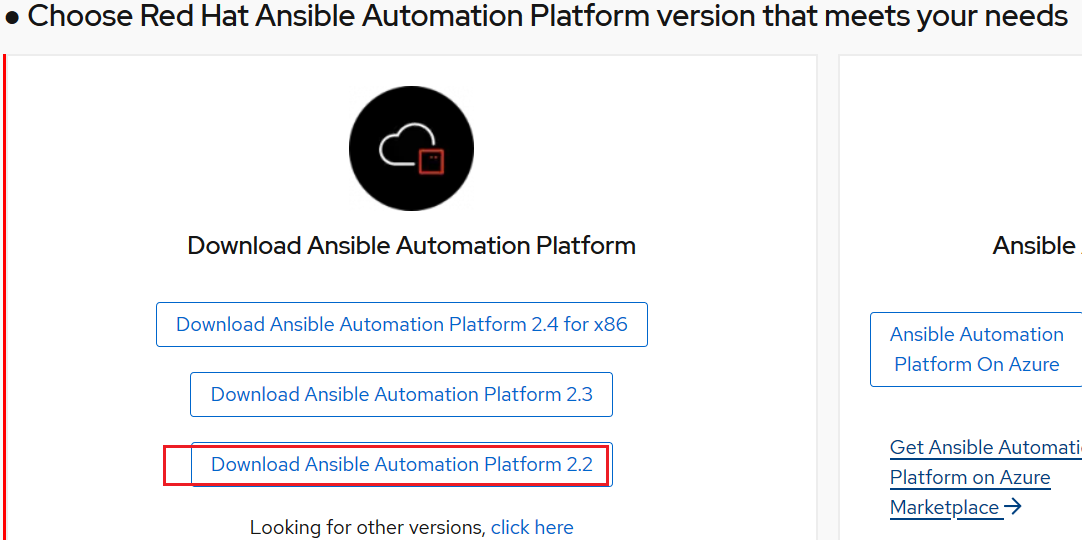
如果没有aap的订阅，在创建repos时报错：

This system has no repositories available through subscriptions.

可以直接登录官网下载aap离线安装包

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

选择目标版本下载



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

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

# ls -lh

drwxr-xr-x. 4 root root 28 Jun 30 2022 bundle

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

在 bundle/el8/repos 子目录下为AAP软件包

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

**★安装AAP的web界面服务**

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'

# ./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."}

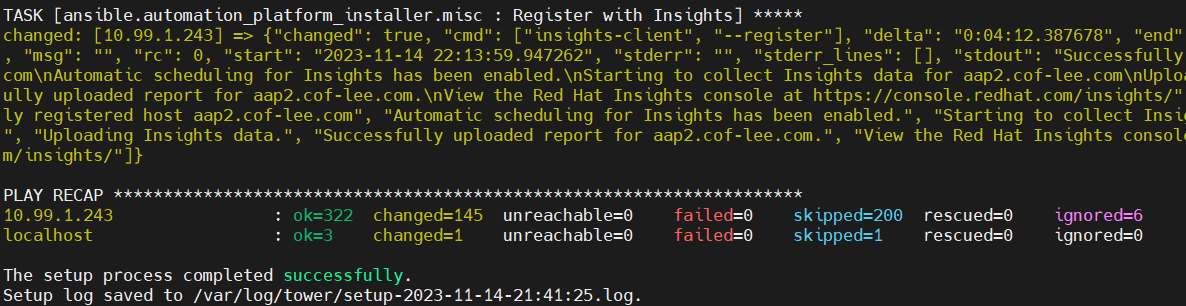
# cat collections/ansible\_collections/ansible/automation\_platform\_installer/roles/packages\_el/tasks/install\_deps.yml

脚本没啥问题，原因是之前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

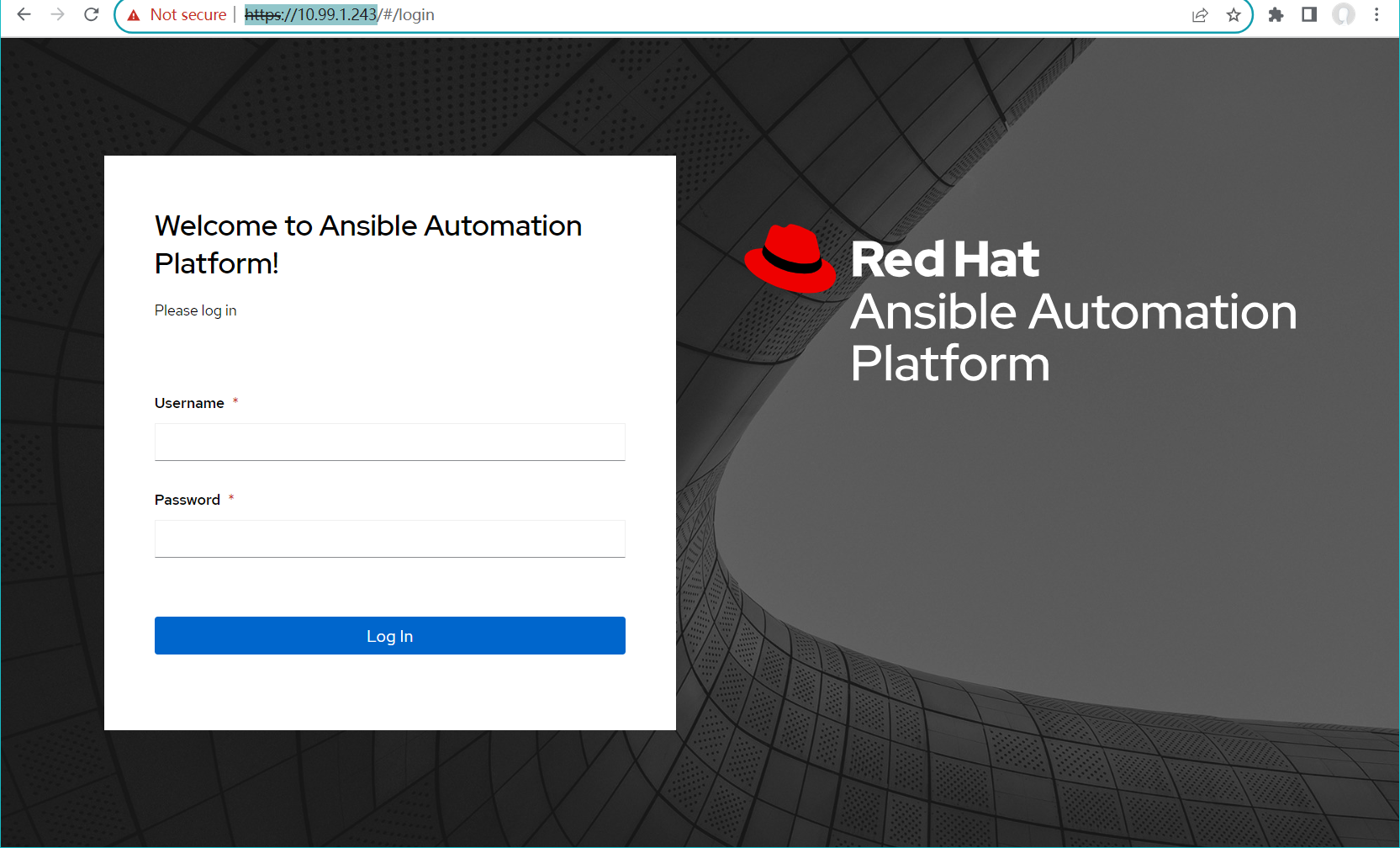
# ./setup.sh #重新开始安装，顺利的话等待10到30分钟安装完成

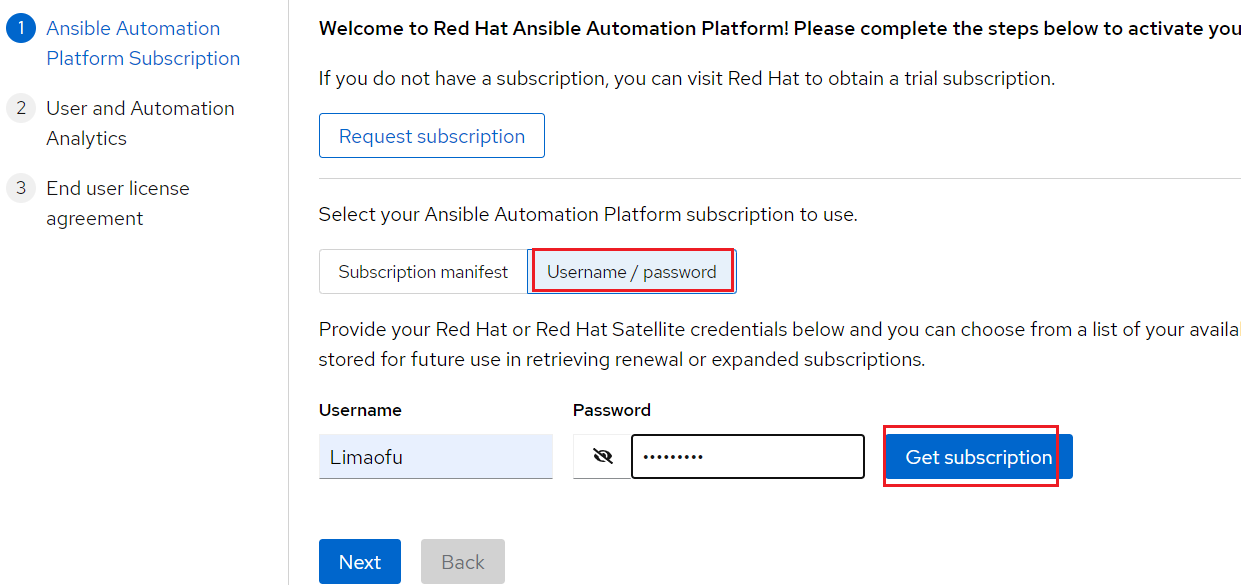


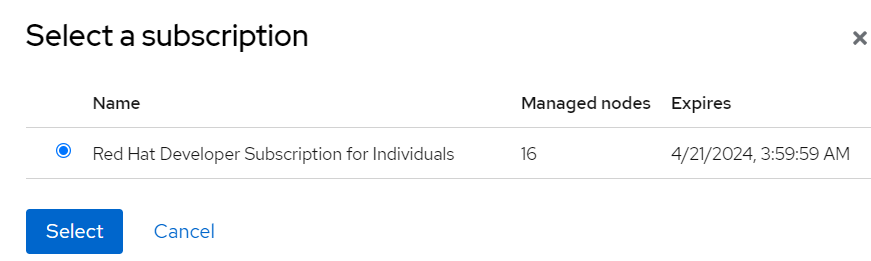
防火墙自动放通80,443端口

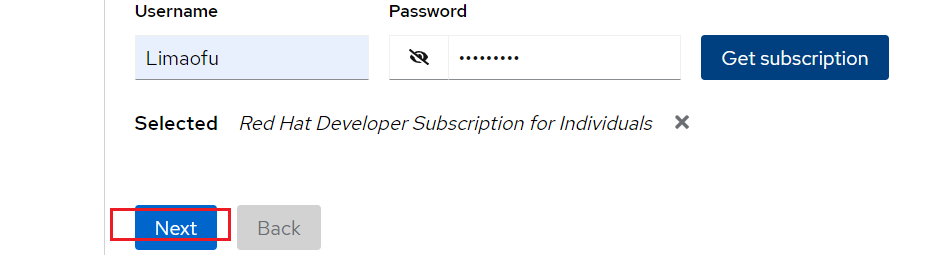
★登录AAP web界面

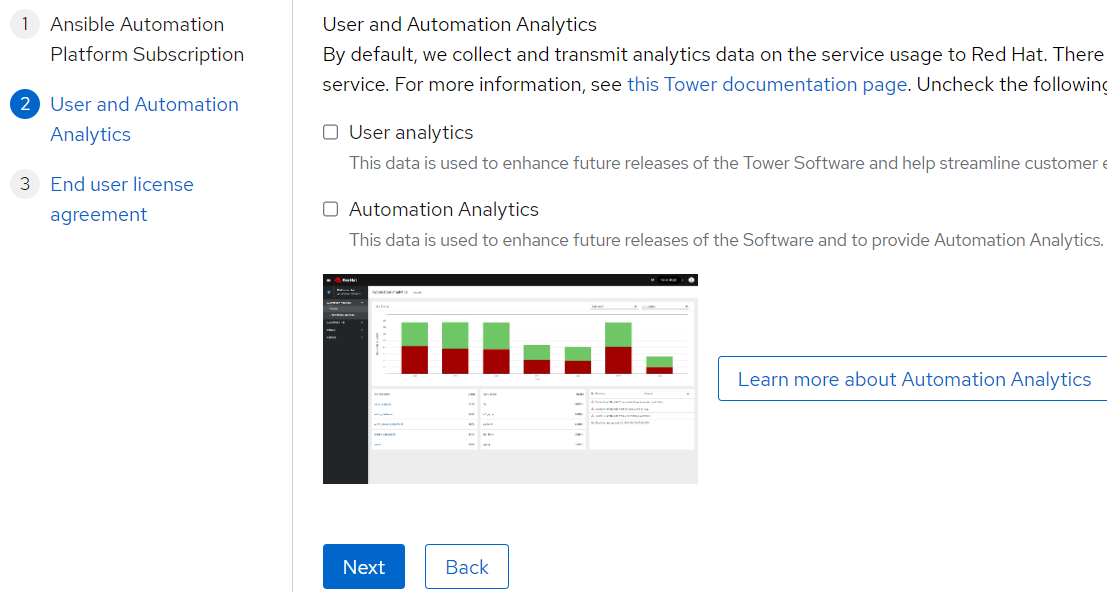
https://10.99.1.243/

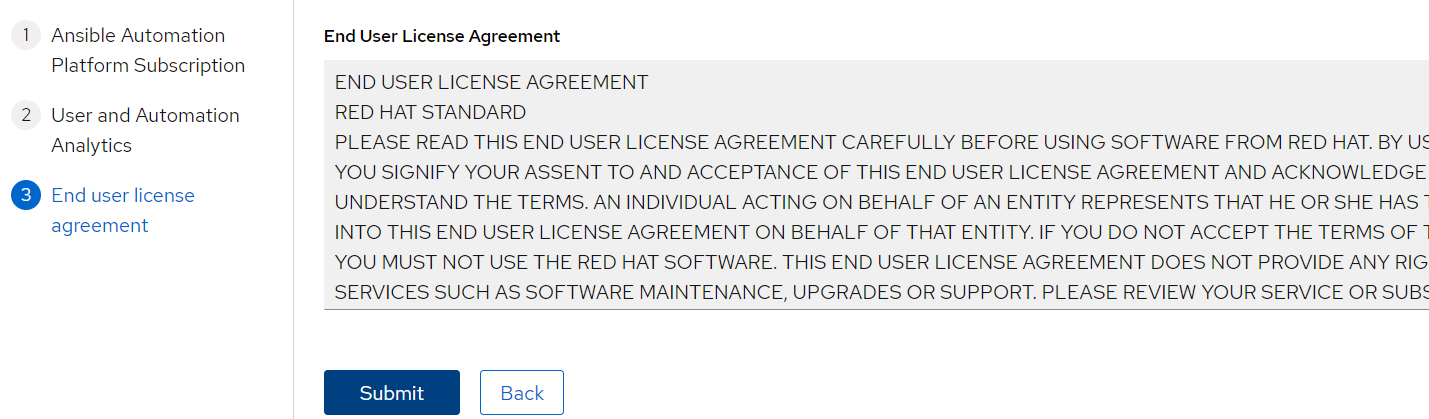




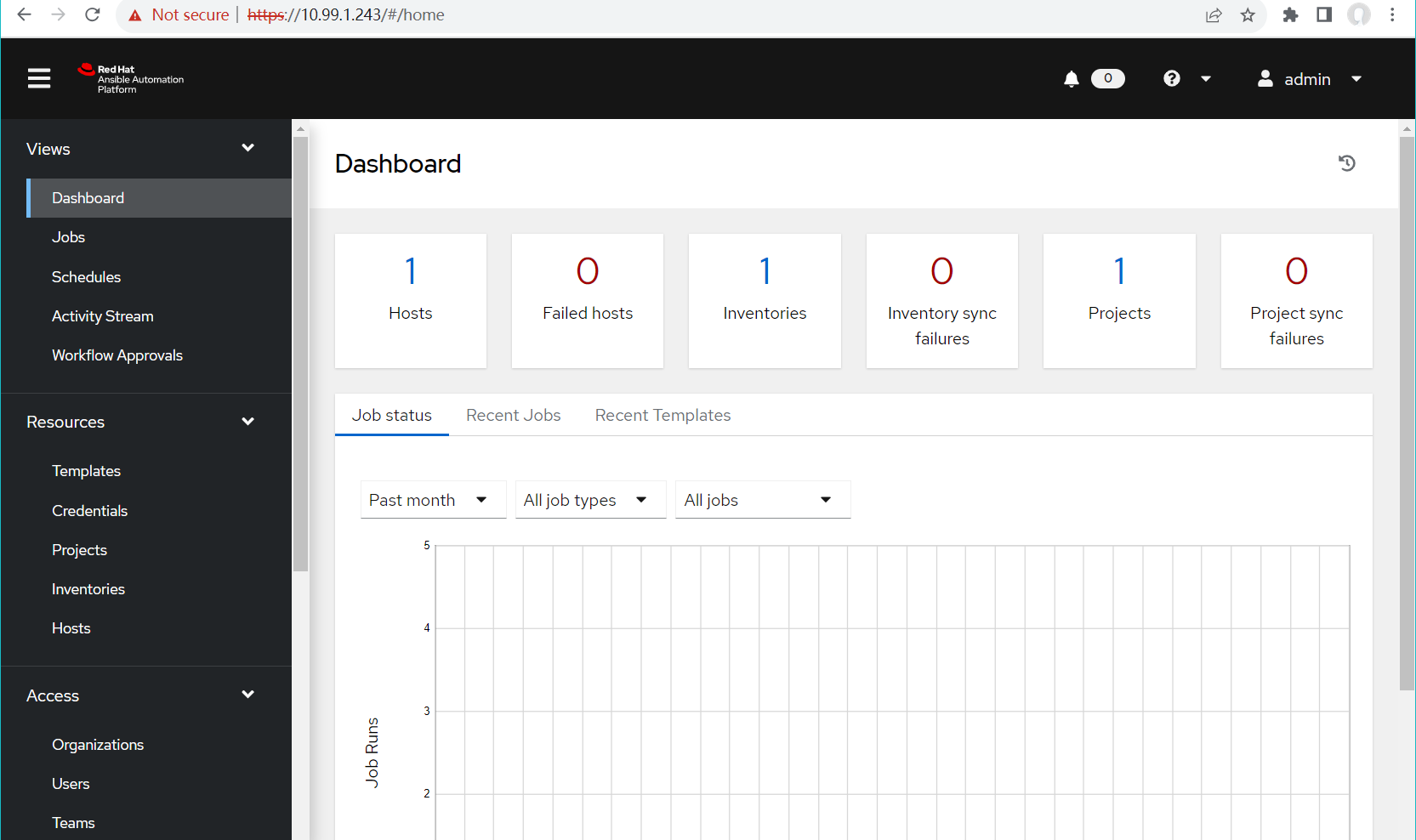


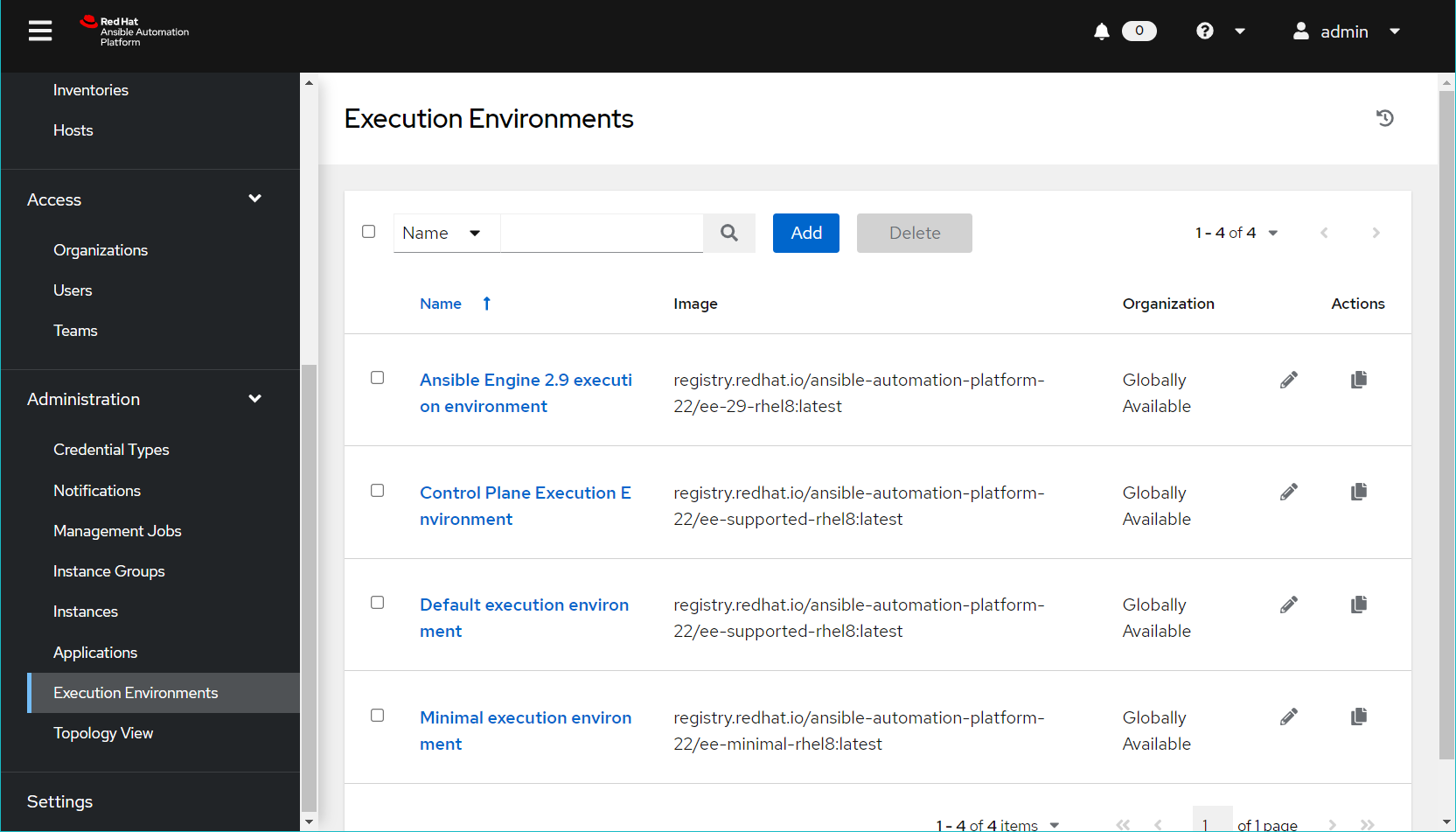






然后进入面板主页





https://console.redhat.com/ansible/ansible-dashboard/trial

**★ansible-navigator运行playbook剧本**

安装ansible-navigator软件包

# yum install ansible-navigator -y

#首先在工作目录下创建ansible.cfg及inventory文件

# 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

# 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

#-m stdout表示以标准输出方式显示结果，默认进入交互模式

#-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，

#可以将镜像下载到本地，然后修改配置为：优先使用本地的镜像

**★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

# 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: cof-lee.com/aap/ee-supported-rhel8:latest

pull:

policy: missing #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

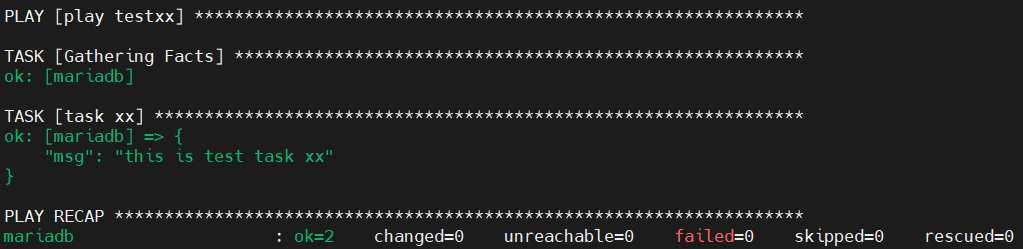
EOF

mode使用 stdout 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 配置时，如果要以交互模式运行，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 ee-supported-rhel8:latest -m stdout



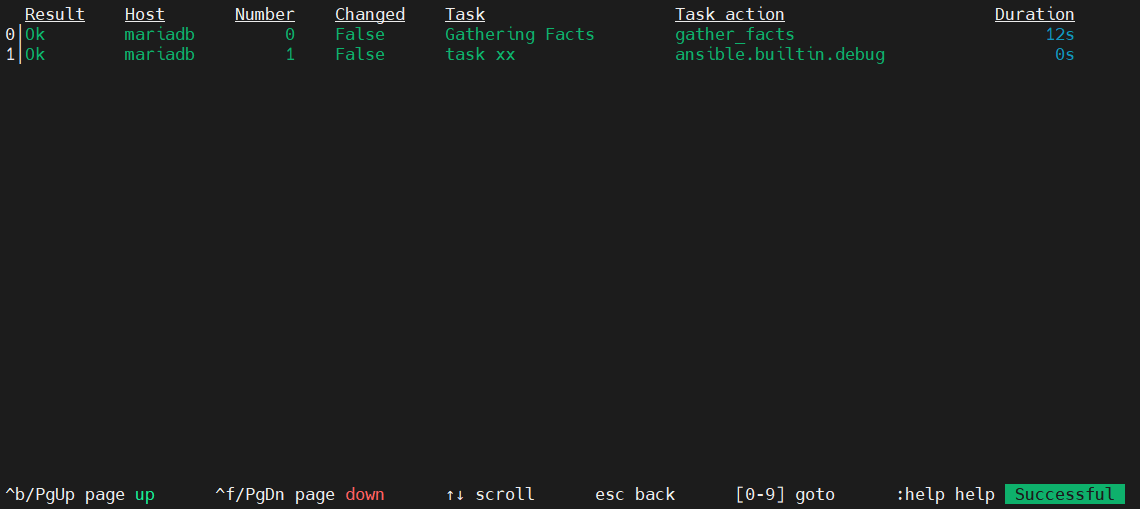
和执行ansible-playbook命令一样，直接输出运行结果，不需要人工干预

★以交互模式执行playbook

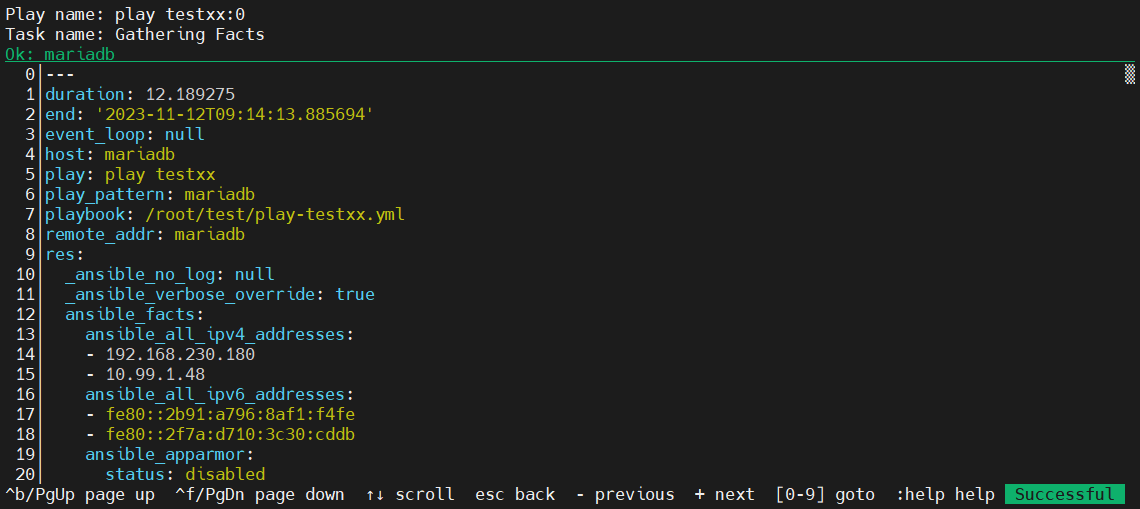
# ansible-navigator run play-testxx.yml -i inventory --eei ee-supported-rhel8:latest -m interactive



默认进入交互模式，按下数字0，进入序号为0的playbook详情页：



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运行时，会使用podman创建一个容器，使用的镜像称为eei（Execution Environment Image）

然后在容器里使用ansible-playbook命令执行指定的playbook剧本

# 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

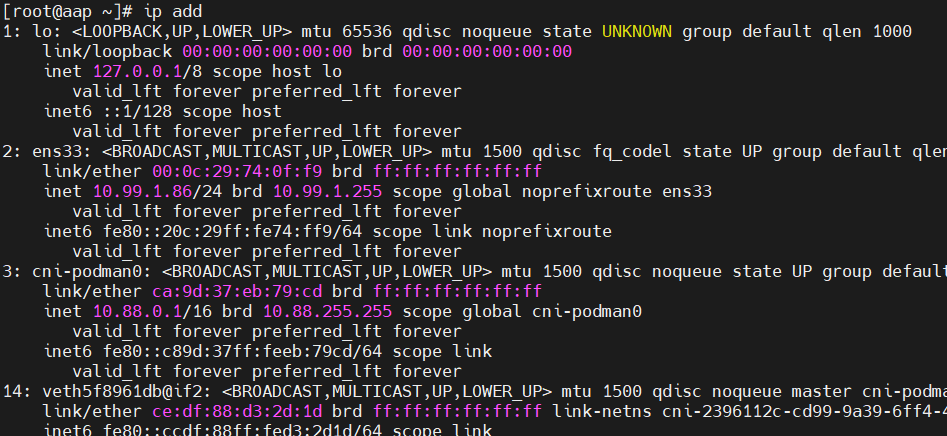
宿主机（执行机）# ip netns exec 容器id ip addr #查看容器的ip

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

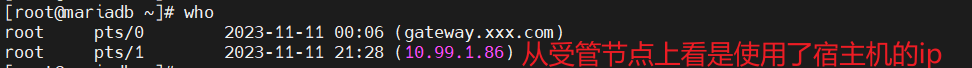
link/ether 5a:e2:ef:f8:fe:bd brd ff:ff:ff:ff:ff:ff link-netnsid 0

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

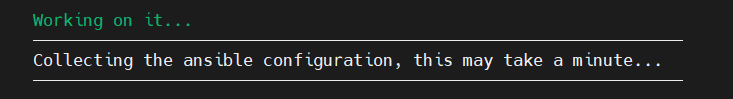
"/root/.ssh:/root/.ssh/:rw,rprivate,rbind"

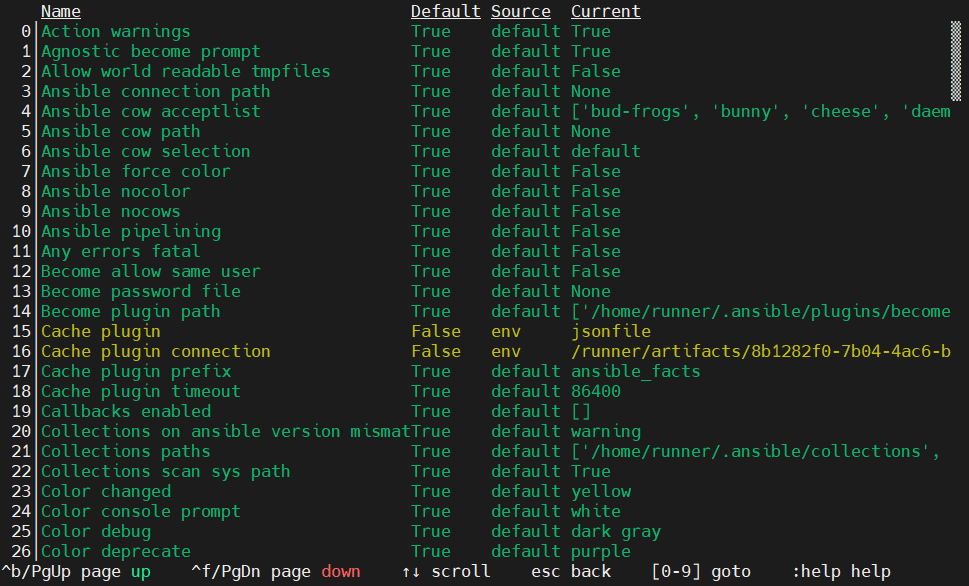
],

**★ansible-navigator配置**

# ansible-navigator config dump -m stdout #以标准输出方式列出所有配置项

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



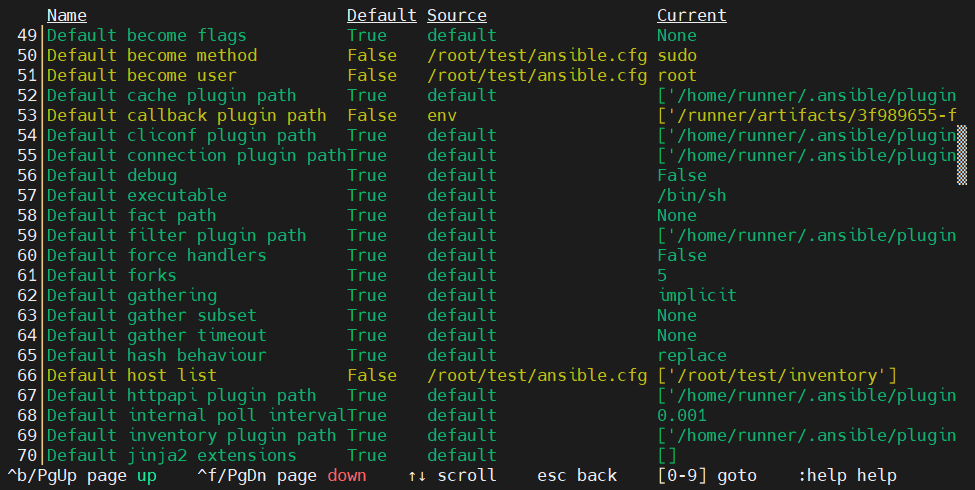


可按下冒号f 加配置项，查看对应的配置信息

:f forks #查看配置项







如上图，最左侧数字表示行数，

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：default表示默认配置，其他为配置源文件或来源

Current这列表示配置项的当前生效值