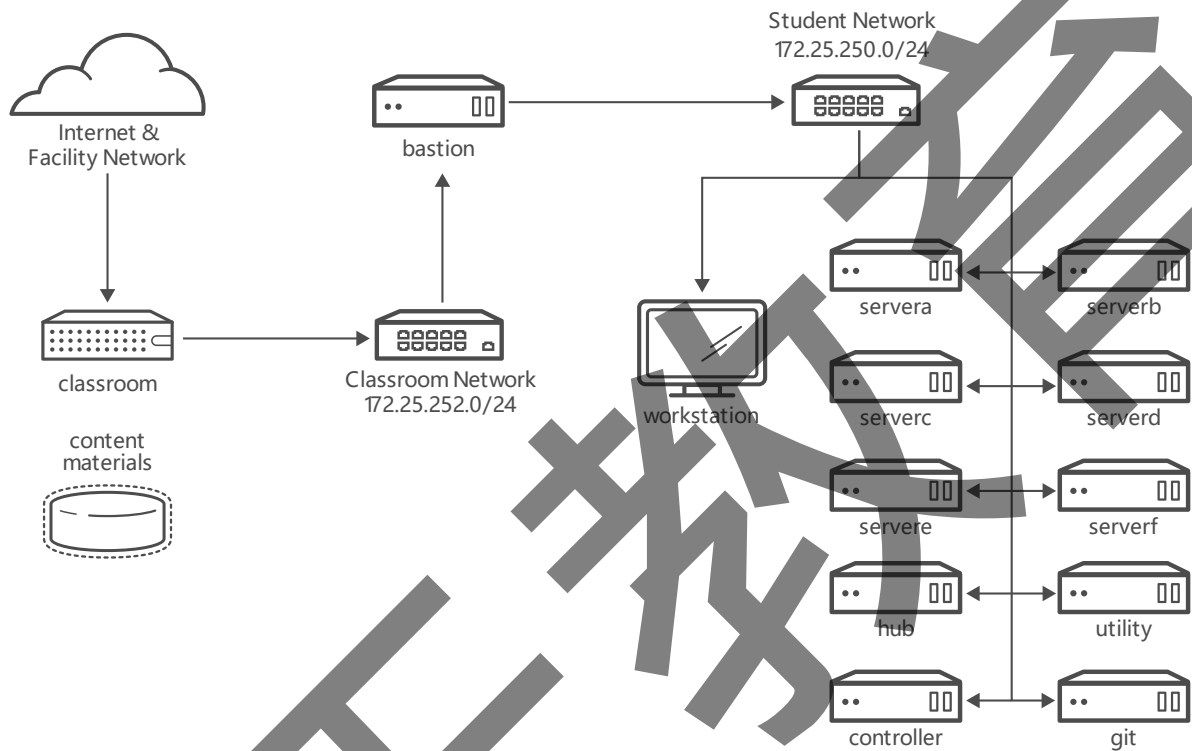


DO374备考资料

基本信息

系统信息



系统	IP地址	角色	ID	开启
classroom	172.25.254.254	content, materials	1	Y
bastion	172.25.250.254	gateway 服务器	2	Y
utility	172.25.250.220	提供其他服务的服务器	3	Y
git	172.25.250.5	git 服务器	4	Y
hub	172.25.250.6	Ansible 私有自动化中心	5	Y

LAN

系统	IP地址	角色	ID
workstation.lab.example.com	172.25.250.9	Ansible control node	1
servera.lab.example.com	172.25.250.10	Ansible managed node	2
serverb.lab.example.com	172.25.250.11	Ansible managed node	3

系统	IP地址	角色	ID
serverc.lab.example.com	172.25.250.12	Ansible managed node	4

系统	IP地址	角色	ID
controller.lab.example.com	172.25.250.7	Ansible Automation Controller control node	5
serverd.lab.example.com	172.25.250.13	Ansible Automation Controller managed node	6
servere.lab.example.com	172.25.250.14	Ansible Automation Controller managed node	7
serverf.lab.example.com	172.25.250.15	Ansible Automation Controller managed node	8

这些系统的IP地址采用静态设置。请勿更改这些设置。主机名称解析已配置为解析上方列出的完全限定主机名，同时也解析主机短名称。

帐户信息

- 登录 workstation 用户名/密码：
student/student
- Git 仓库地址：<https://git.lab.example.com>
用户名 student，密码 Student@123
- Ansible 项目
默认使用 devops 作为远程登录用户
- Execution Environment 使用 ee-supported-rhel8:latest 镜像
- Ansible Automation Controller 地址：
<https://controller.lab.example.com>
用户名admin，密码 redhat
- Ansible Hub 地址：
<https://hub.lab.example.com>
用户名 student，密码 redhat123

其他信息

- 一些考试项目可能需要修改 Ansible 主机清单。您要负责确保所有以前的清单组和项目保留下来，与任何其他更改共存。您还要有确保清单中所有默认的组和主机保留您进行的任何更改。
- 考试系统上的防火墙默认为不启用，SELinux 则处于 Disabled 模式。
- 所有节点，yum存储库已正确配置。
- 有些考试项目会将项目特定信息存储在 Git 存储库中。这些 Git 存储库已在 <http://git.lab.example.com> 上进行了配置。考试项目 Git 存储库的确切位置将在使用 Git 存储库的考试项目中指定。每个项目 Git 存储库都独立于任何其他考试项目 Git 存储库，且与它们无关。除非另有指定，否则您在 Ansible 控制节点上为管理

Ansible 托管节点所做的所有工作（包括 Ansible playbook、配置文件、主机清单等）都应上传到相应的项目 Git 存储库中，具体由各考试项目指定。

- 一些项目需要额外的文件，这些文件已在以下位置提供：<http://materials/classroom/ansible/>
- 产品文档可从以下位置找到：<http://materilas/docs/>
- 其他资源也进行了配置，供您在考试期间使用。关于这些资源的具体信息将在需要这些资源的项目中提供。

虚拟系统管理

考试期间，您可以随时关闭或重新引导虚拟机系统。您可以从虚拟系统本身进行这项操作，也可以从物理系统控制虚拟系统。要从物理系统访问或控制考试系统，单击桌面上 VM 控制台图标。这会显示一个表格，包含每个虚拟机系统的对应按钮，单击特定虚拟机系统的按钮将弹出一个菜单，包含用来控制该系统选项：

- 启动节点 VM-如果指定的虚拟系统未在运行，该选项将启动指定系统。如果系统已经在运行-则该选项无任何作用。
- 重新引导节点 VM-正常关闭考试虚拟系统，然后重启。
- 关闭节点 VM-正常关闭指定虚拟系统。
- 关闭节点 VM 电源-立即关闭指定虚拟系统。
- VM 控制台节点-这将打开一个窗口，用于连接到指定虚拟系统的控制台。请注意，如果将焦点移动到此窗口，控制台将抓住您的鼠标。要恢复鼠标，同时键入 Ctrl+Alt。
- 重建节点 VM-将当前 VM 还原为原始状态。系统将弹出一个单独的窗口，要求您确认操作。警告！！！您在 VM 上完成的所有操作都将丢失。仅当系统无法使用时才应使用这个功能。在使用这个功能之前，确保关闭 VM。

重要评测信息

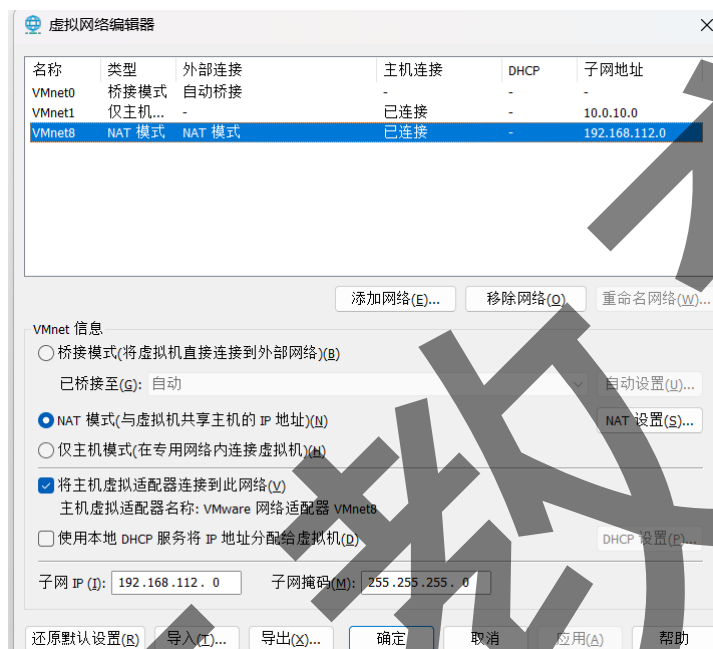
请注意，在评分之前，您的 Ansible 托管节点系统 将重置为考试开始时的初始状态，您编写的 Ansible playbook 将通过以 **student** 用户身份在控制节点上运行来加以应用。在 playbook 运行后，系统会重新启动您的托管节点，然后进行评估，以判断它们是否按照规定进行了配置。

请注意，在评分之前，您的 Ansible 托管节点系统 将重置为考试开始时的初始状态，您创建的 Ansible Tower 作业将通过以指定的用户身份运行来加以应用。在作业运行后，系统会重新启动 Ansible Tower 托管节点，然后进行评估，以判断它们是否按照规定进行了配置。

准备工作

配置机器外网通信

默认情况下，所有机器无法连接外网，请进行f0的网络配置，完成所有虚拟机外部网络的通信。在VMware虚拟机的情况下，你应当使用NAT网卡模式，并确定其网段信息。



如图确定NAT网络模式的网络信息，本资料为192.168.112.0/24网段，网关为192.168.112.2。为F0配置网络。

重置并启动所有虚拟机

```
1 [root@foundation0 ~]# rht-vmctl fullreset classroom -y
2 [root@foundation0 ~]# rht-vmctl fullreset all -y
```

确认物理网卡名称，清除原有配置后生成新的连接配置。

```
1 # 确认物理网卡
2 [root@foundation0 ~]# nmcli connection show | grep eth
3      System ens160          f95d1d34-5e3c-4f1b-b951-11eaa1ba63f0
      ethernet  ens160
4 # 删除原有配置
5 [root@foundation0 ~]# nmcli connection delete "System
      ens160"
6 # 新增配置
7 [root@foundation0 ~]# nmcli connection add ifname ens160
      con-name ens160 type ethernet \
```

```

8      ipv4.method manual ipv4.addresses 192.168.112.90/24
9  # 停止 F0 DHCP服务
10     [root@foundation0 ~]# systemctl disable --now dhcpd.service
11  # 启动nat网络
12     [root@foundation0 ~]# rht-external --configure
ens160/192.168.112.100/24/192.168.112.2
13                                     # 配置文件名/IP/掩码/网
关
14  # 配置DNS
15     [root@foundation0 ~]# rht-external --dns 114.114.114.114
16  # 测试网络连接
17     [root@foundation0 ~]# ping baidu.com -c 1
18     PING baidu.com (39.156.66.10) 56(84) bytes of data.
19     64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1
ttl=128 time=25.9 ms
20     --- baidu.com ping statistics ---
21     1 packets transmitted, 1 received, 0% packet loss, time
0ms
22     rtt min/avg/max/mdev = 25.889/25.889/25.889/0.000 ms
23     [root@foundation0 ~]# ssh workstation ping baidu.com -c 1
24     PING baidu.com (110.242.68.66) 56(84) bytes of data.
25     64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=1
ttl=125 time=24.6 ms
26     --- baidu.com ping statistics ---
27     1 packets transmitted, 1 received, 0% packet loss, time
0ms
28     rtt min/avg/max/mdev = 24.558/24.558/24.558/0.000 ms

```

上传部分文件

你应当拥有 DO374.zip 文件，应将其上传至 F0 主机。

```

1  # 上传至 F0
2  [root@foundation0 ~]# ls -l | grep zip
3      -rw-r--r--. 1 root root 24700 Apr  3 14:36 DO374.zip
4  # 拷贝至 utility 主机
5  [root@foundation0 ~]# scp DO374.zip root@utility:~
6  # 登录至 utility 机器进行上传操作
7  [root@foundation0 ~]# ssh root@utility
8  [root@utility ~]# unzip DO374.zip
9  [root@utility ~]# ls
10     DO374.zip git git_prepare.sh init.sh __MACOSX
11     # 文件解析:
12     git : 题目所需文件，应上传至git服务器

```

```
13 git_prepare.sh : 上传脚本
14 init.sh: 环境初始化脚本
15 # 安装git并进行配置
16 [root@utility ~]# yum -y install git
17 [root@utility ~]# git config --global push.default simple
18 # 执行脚本完成配置
19 [root@utility ~]# chmod +x init.sh git_prepare.sh
20 [root@utility ~]# ./init.sh
21 [root@utility ~]# ./git_prepare.sh
```

题目部分

1. 为用户配置Git

在 workstation.lab.example.com 节点上为用户管理员配置Git:

- ☐ Git user name: student
- ☐ Git 用户电子邮箱: student@lab.example.com
- ☐ Default push method: simple

除上述要求外, 请勿做任何更改

```
1 # 解题
2 [student@workstation ~]$ git config --global user.name student
3 [student@workstation ~]$ git config --global user.email student@lab.example.com
4 [student@workstation ~]$ git config --global push.default simple
5 # 推荐加入配置: 在7200秒内无需重复验证git账户信息
6 [student@workstation ~]$ git config --global credential.helper cache --timeout=7200
```

```
1 # 检查
2 [student@workstation ~]$ git config -l
3 user.name=student
4 user.email=student@lab.example.com
5 push.default=simple
```

2、创建用户

在 https://git.lab.example.com/student/create_users.git 找到 create_users 的 git 项目

- ☐ create_users.yml 是该项目的脚本文件名
- ☐ 在 developer 组中添加用户 greg
- ☐ 在 dev 组中添加 serverc 节点
- ☐ commit 并 push 变更到 git 仓库

除上面列出的以外，请勿进行任何其他更改

```
1 # 解题
2 # 安装ansible
3 [student@workstation create_users]$ sudo dnf -y install
ansible-navigator
4
5 # 修改导航器配置
6 [student@workstation ~]$ cat ~/.ansible-navigator.yml
7     ansible-navigator:
8         execution-environment:
9             image: hub.lab.example.com/ee-supported-
rhe18
10             pull:
11                 policy: missing
12
13 # 登录至仓库
14 [student@workstation ~]$ podman login
15 Username: admin
16 Password: redhat
17 Login Succeeded!
18
19 # 拉取文件
20 [student@workstation ~]$ git clone
https://git.lab.example.com/student/create_users.git
21 Cloning into 'create_users'...
22 Username for 'https://git.lab.example.com': student
23 Password for 'https://student@git.lab.example.com':
Student@123
24 remote: Enumerating objects: 6, done.
25 remote: Counting objects: 100% (6/6), done.
```

```
26         remote: Compressing objects: 100% (5/5), done.
27         remote: Total 6 (delta 0), reused 0 (delta 0), pack-
reused 0
28         Receiving objects: 100% (6/6), done.
29         [student@workstation ~]$ ls
30         create_users
31         [student@workstation ~]$ cd create_users/
32
33         # 修改主机清单
34         [student@workstation create_users]$ sed -i '/dev/a
serverc' inventory
35         [student@workstation create_users]$ cat inventory
36         [dev]
37         serverc
38         servera
39         [prod]
40         serverb
41
42         # 修改变量文件文件: 新增两行内容, 原有内容不修改
43         [student@workstation create_users]$ cat user_list.yml
44         users:
45             - name: david
46               group: developer
47             - name: fred
48               group: manager
49             - name: sally
50               group: developer
51             - name: greg
52               group: developer
53
54         # 试运行
55         [student@workstation create_users]$ ansible-navigator run
create_users.yml -m stdout
56         # 无问题即可上传
57         [student@workstation create_users]$ git add .
58         [student@workstation create_users]$ git commit -m 2
59         [master 4a35d64] 2
60         4 files changed, 5240 insertions(+)
61         create mode 100644 ansible-navigator.log
62         create mode 100644 create_users-artifact-2024-04-
03T15:01:03.358747+00:00.json
63         [student@workstation create_users]$ git push
64         Username for 'https://git.lab.example.com': student
65         Password for
'https://student@git.lab.example.com': Student@123
```



```
66 Enumerating objects: 9, done.
67 Counting objects: 100% (9/9), done.
68 Delta compression using up to 4 threads
69 Compressing objects: 100% (5/5), done.
70 Writing objects: 100% (6/6), 30.74 KiB | 5.12 MiB/s,
done.
71 Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
72 To
https://git.lab.example.com/student/create_users.git
73 96f99de..4a35d64 master -> master
74 [student@workstation create_users]$
75
```

```
1 # 验证
2 [student@workstation create_users]$ ansible servera -m shell
-a 'groupmems -g developer -l' -b
3 servera | CHANGED | rc=0 >>
4 david sally greg
5 [student@workstation create_users]$ ansible serverb -m shell
-a 'groupmems -g developer -l' -b
6 serverb | CHANGED | rc=0 >>
7 david sally greg
8 [student@workstation create_users]$ ansible serverc -m shell
-a 'groupmems -g developer -l' -b
9 serverc | CHANGED | rc=0 >>
10 david sally greg
11
```

3. 管理 web 服务器

在 Git 仓库 `httpd_alias` 中管理的 Web 服务器，添加别名：

- ☐ Git 项目在 https://git.lab.example.com/student/httpd_alias.git
- ☐ 部署新别名的脚本文件是 `install_httpd_alias.yml`
- ☐ 只有在安装别名时，才会重启 `httpd` 服务器。也就是说，如果已经安装了别名，再运行脚本，则不会重新启动 `httpd` 服务
- ☐ `commit` 并 `push` 变更到 git 仓库

除上面列出的以外，请勿进行任何其他更改

```

1 # 解题
2 # 下载本题文件
3 [student@workstation ~]$ git clone
https://git.lab.example.com/student/httpd_alias.git
4 [student@workstation ~]$ ls
5 create_users httpd_alias
6 [student@workstation ~]$ cd httpd_alias/
7 # 修改脚本文件
8 [student@workstation httpd_alias]$ cat
install_httpd_alias.yml
9 ---
10 - name: Add Apache alias
11   hosts: prod
12   become: yes
13   tasks:
14     - name: copy alias file
15       copy:
16         src: alias.conf
17         dest: /etc/httpd/conf.d
18     # 新增以下6行
19     notify: Restart apache
20   handlers:
21     - name: Restart apache
22       ansible.builtin.service:
23         name: httpd
24         state: restarted
25 # 试运行
26 [student@workstation httpd_alias]$ ansible-navigator run
install_httpd_alias.yml -m stdout
27 # 上传至git
28 [student@workstation httpd_alias]$ git add .
29 [student@workstation httpd_alias]$ git commit -m 3
30 [student@workstation httpd_alias]$ git push
31

```

4. 管理网站内容

在部署到生产之前，dev web 服务器用于测试网站内容

Git 项目 `httpd_alias` 包含一个不完整的剧本，名为 `manage_content.yml`，用于管理 dev web 服务器的内容。在 Git 仓库中 (https://git.lab.example.com/student/manage_content.git) 完善该剧本，实现：

- ☐ 当使用标签 `alpha` 来运行该剧本时，将生成内容 `Que sera, sera` 并部署到 dev 主机上的 `/var/www/html/index.html` 文件中
- ☐ 当使用标签 `beta` 来运行该剧本时，将生成内容 `whatever will be, will be`，并且部署到 dev 主机的 `/var/www/html/index.html` 文件中
- ☐ 如果没有使用以上任何一个标签运行剧本，则在受管主机上既不产生也不保存任何信息
- ☐ `commit` 并 `push` 变更到 git 仓库

除上面列出的以外，请勿进行任何其他更改

```
1 # 解题
2 # 下载文件至本地
3 [student@workstation ~]$ git clone
  https://git.lab.example.com/student/manage_content.git
4 [student@workstation ~]$ cd manage_content/
5 # 修改剧本文件
6 [student@workstation manage_content]$ cat
  manage_content.yml
7 ---
8 - name: Deploy content
9   hosts: dev
10  become: yes
11  # 新增以下内容
12  tasks:
13    - name: Copy file
14      copy:
15        content: "Que Sera, Sera\n"
16        dest: /var/www/html/index.html
17        tags: [ never, alpha ]
18    - name: Copy file
19      copy:
20        content: "Whatever will be, will be\n"
21        dest: /var/www/html/index.html
22        tags: [ never, beta ]
23 # 试运行
24 [student@workstation manage_content]$ ansible-navigator
  run manage_content.yml -m stdout
25 # 上传至git
```

```
26 [student@workstation manage_content]$ git add .
27 [student@workstation manage_content]$ git commit -m 4
28 [student@workstation manage_content]$ git push
29
```

```
1 # 验证
2 [student@workstation manage_content]$ ansible-navigator run
  manage_content.yml -m stdout -t alpha
3 [student@workstation manage_content]$ curl servera
4     Que Sera, Sera
5
6 [student@workstation manage_content]$ ansible-navigator run
  manage_content.yml -m stdout -t beta
7 [student@workstation manage_content]$ curl servera
8     whatever will be, will be
```

5. Ansible调优

按照以下要求更新Git仓库（https://git.lab.example.com/student/tune_ansible.git）中的Ansible配置文件:

- ☐ 默认情况下，禁用事实收集
- ☐ 最大并发主机连接数为 45
- ☐ commit 并 push 变更到 git 仓库

除上面列出的以外，请勿进行任何其他更改

```
1 # 解题
2 # 下载文件至本地
3 [student@workstation ~]$ git clone
  https://git.lab.example.com/student/tune_ansible.git
4 [student@workstation ~]$ cd tune_ansible/
5 # 修改配置文件
6 [student@workstation tune_ansible]$ cat ansible.cfg
7     [defaults]
8     collections_paths = /home/student/mycollections
9     inventory = ./inventory
10    remote_user = devops
11    ask_pass = false
```

```

12      # 添加以下两行
13      gathering = explicit
14      forks = 45
15
16      [privilege_escalation]
17      become = false
18      become_method = sudo
19      become_user = root
20      become_ask_pass = false
21
22      # 获取帮助
23      导出ansible默认配置: [student@workstation tune_ansible]$
ansible-config init > 1.txt
24      过滤关键字: [student@workstation tune_ansible]$ cat
1.txt | grep -E "(forks|gather)"
25      # 关键信息:
26      forks=5
27      gathering=implicit
28      ...but does always affect the 'gather_facts'
action (implicit or explicit)
29
30      # 上传至git
31      [student@workstation tune_ansible]$ git add .
32      [student@workstation tune_ansible]$ git commit -m 5
33      [student@workstation tune_ansible]$ git push
34

```

6. 从列表创建用户

Git 仓库(https://git.lab.example.com/student/create_users_complex.git)
包含以下资源:

`user_information.yml` 是一个用户账户清单, 该文件包含多字段:

- ☐
 - `name` 字段指定账户的用户名和登录ID
 - `first` 字段指定用户的 first name
 - `middle` 字段指定用户的 middle name
 - `last` 字段指定用户的 last name
 - `uid` 字段指定账户关联的 用户 ID

- ☐ `inventory` 是主机清单文件
不要对以上文件做任何修改

创建剧本，使用以上的文件，实现以下操作：

剧本文件名为 `manage_accounts.yml`，在 `inventory` 规定的主机上运行时，该剧本会根据 `user_information.yml` 文件内容，使用指定的用户 ID 创建用户账户

针对每个账户，该剧本会按照以下要求生成随机的 6 位数字密码

- 必须使用 SHA-512 对密码进行加密
- 密码的纯文本版本和用于生成密码的随机值“salt”必须存储在名为 `password-` 的文件中，其中是与该帐户关联的用户名。
例如，针对名为“frederick”用户，密码和“salt”存储在文件 `password-frederick` 中
- 剧本需要在其运行的目录中生成 `password-` 文件

针对每个账户，`user comment (GECOS)` 字段需要按照以下格式要求为

- ☐ 用户设置恰当的名字：First Middle Last(中间有单空格符)，如上所示，名字每个组成部分都必须大写
- ☐ `commit` 并 `push` 变更到 `git` 仓库

```
1 # 解题
2 # 下载文件至本地
3 [student@workstation ~]$ git clone
  https://git.lab.example.com/student/create_users_complex.git
4 [student@workstation ~]$ cd create_users_complex/
5 # 编写剧本文件
6 [student@workstation create_users_complex]$ cat
  manage_accounts.yml
7 ---
8 - hosts: all
9   become: true
10  vars_files:
11    - user_information.yml
12  tasks:
13    - name: Add the user
14      user:
15        name: "{{ item.name }}"
16        uid: "{{ item.uid }}"
17        comment: "{{ item.first | capitalize
18  }} {{ item.middle | capitalize }} {{ item.last | capitalize }}"
19        password: "{{
  lookup('ansible.builtin.password', 'password-{{ item.name }}',
  chars='digits', length=6, encrypt='sha512_crypt') }}"
```

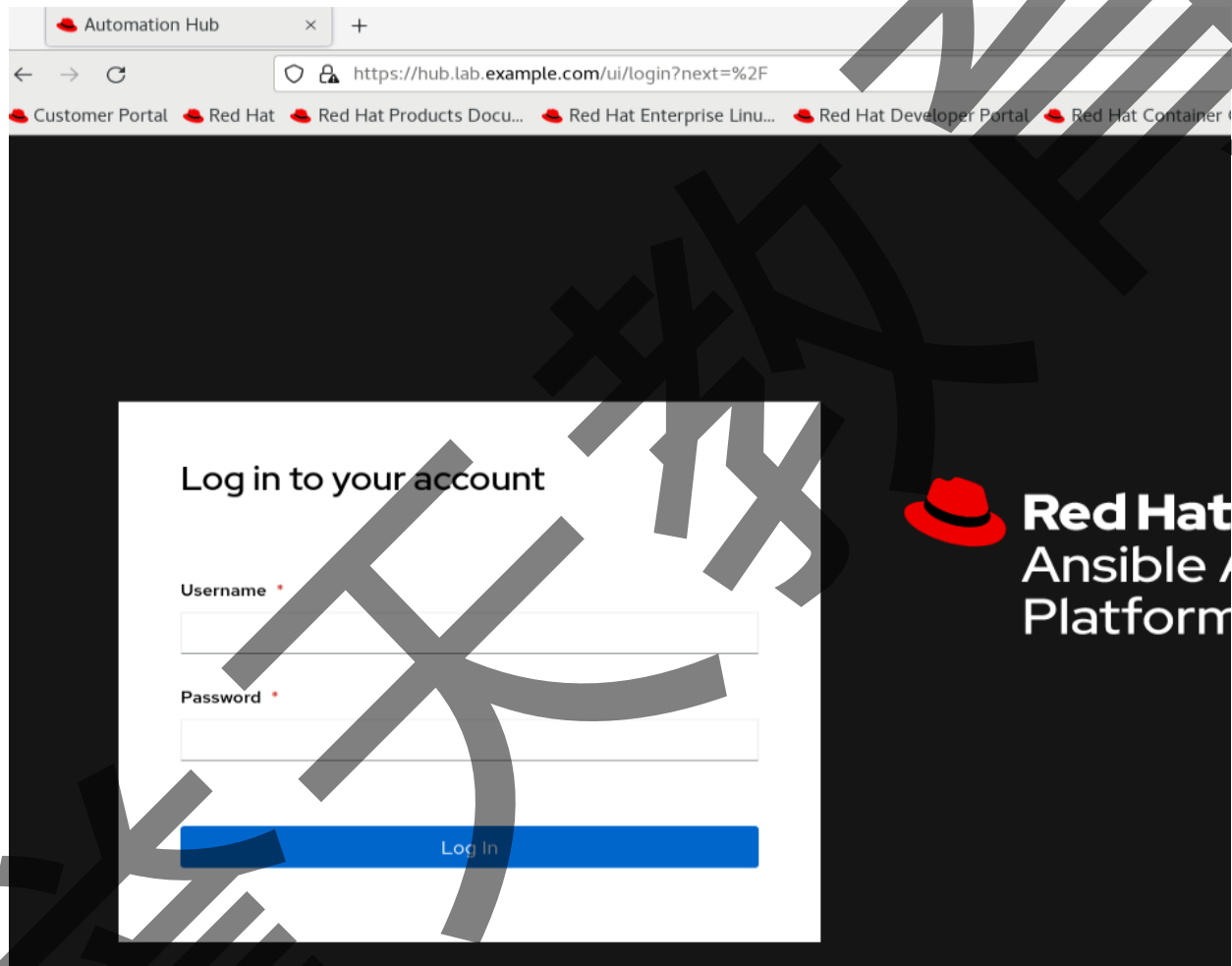
```
19         loop: "{{ users }}"
20     # 试运行
21     [student@workstation create_users_complex]$ ansible-
navigato run manage_accounts.yml -m stdout
22     # 上传至git
23     [student@workstation create_users_complex]$ git add .
24     [student@workstation create_users_complex]$ git commit -
m 6
25     [student@workstation create_users_complex]$ git push
26
```

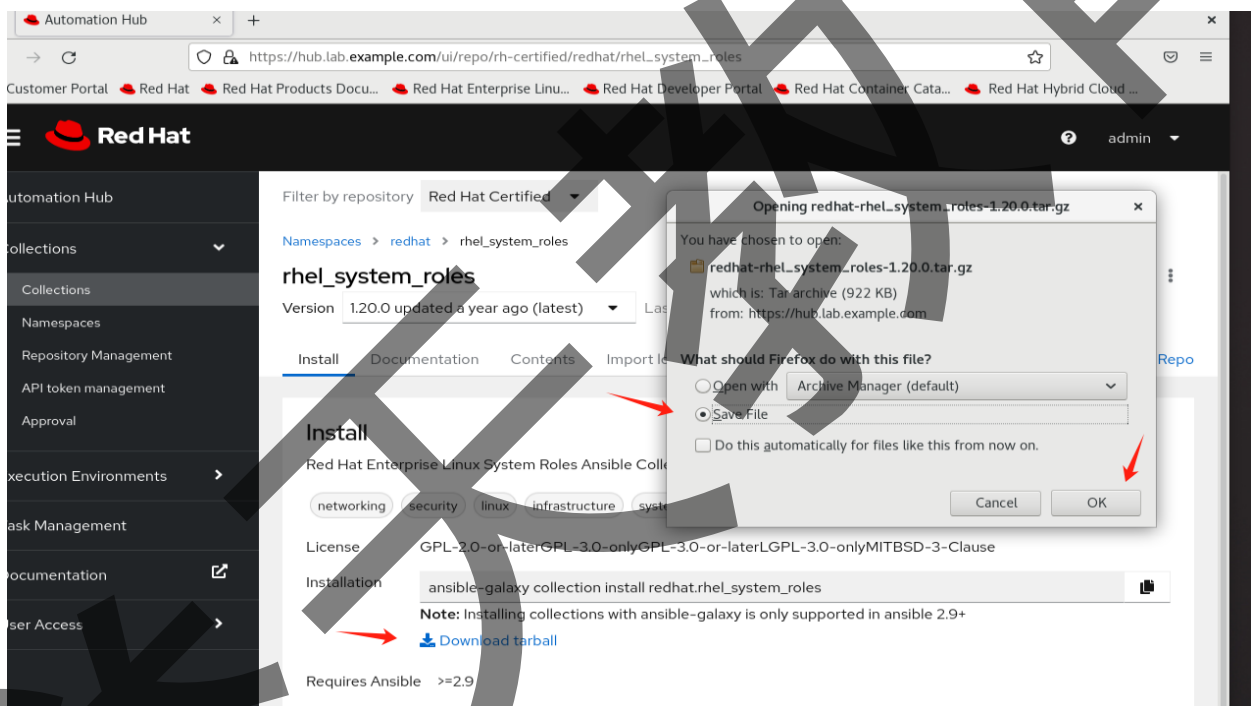
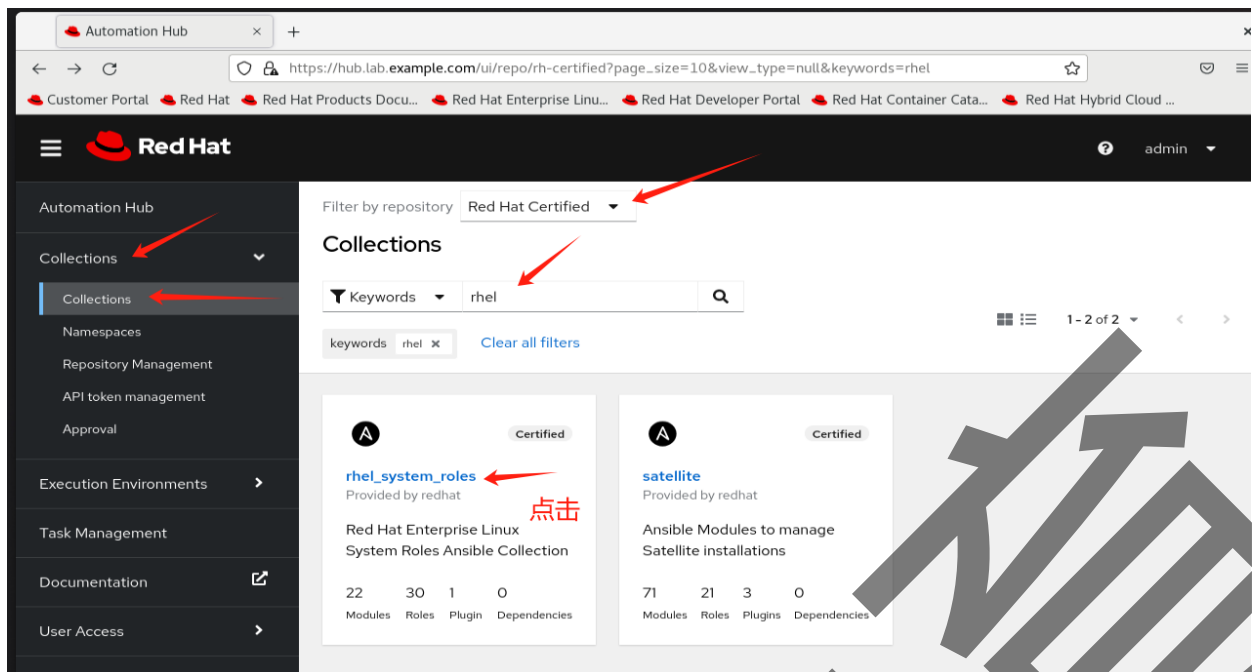
```
1 # 测试
2     [student@workstation create_users_complex]$ ansible all
-a 'tail -n 3 /etc/passwd'
3         servera | CHANGED | rc=0 >>
4         bach:x:2000:2000:Johann Sebastian
Bach:/home/bach:/bin/bash
5         handel:x:2001:2001:George Frideric
Handel:/home/handel:/bin/bash
6         mozart:x:2002:2002:Wolfgang Amadeus
Mozart:/home/mozart:/bin/bash
7         serverb | CHANGED | rc=0 >>
8         bach:x:2000:2000:Johann Sebastian
Bach:/home/bach:/bin/bash
9         handel:x:2001:2001:George Frideric
Handel:/home/handel:/bin/bash
10        mozart:x:2002:2002:Wolfgang Amadeus
Mozart:/home/mozart:/bin/bash
11        serverc | CHANGED | rc=0 >>
12        bach:x:2000:2000:Johann Sebastian
Bach:/home/bach:/bin/bash
13        handel:x:2001:2001:George Frideric
Handel:/home/handel:/bin/bash
14        mozart:x:2002:2002:Wolfgang Amadeus
Mozart:/home/mozart:/bin/bash
15    [student@workstation create_users_complex]$ ls
password-*
16        password-bach password-handel password-mozart
17    [student@workstation create_users_complex]$ cat
password-bach
18        170519 salt=qpeeoz6XoUaVA6.q
19
```

7. 安装Collection

安装 hub.lab.example.com 上的 redhat-rhel_system_roles Collection,
使用 admin 用户,
Collection 安装在 /home/student/mycollections 目录中

登录至 hub.lab.example.com 下载指定 collection。admin/redhat





```

1 # 解题
2 # 将collection文件拷贝至workstation
3 [root@foundation0 ~]# scp Downloads/redhat-
  rhel_system_roles-1.20.0.tar.gz workstation:~ student的家目录下
4 redhat-rhel_system_roles-1.20.0.tar.gz
   100% 922KB 126.1MB/s 00:00
5 # 创建角色目录并安装角色
6 [student@workstation ~]$ mkdir -p
  /home/student/mycollections
7 [student@workstation ~]$ ansible-galaxy collection
  install -p /home/student/mycollections \
8 redhat-rhel_system_roles-1.20.0.tar.gz
  
```

```
1 # 验证
2 [student@workstation ~]$ ansible-galaxy collection list
3 # /usr/share/ansible/collections/ansible_collections
4 collection version
5 -----
6 redhat.rhel_system_roles 1.16.2
```

8. 创建自定义Collection

使用 https://git.lab.example.com/student/custom_collection.git 完成以下要求：

在项目中包含以下资源

- ☐ tasks_main.yml
- ☐ users.conf

不要对以上文件做任何修改

在控制节点创建自定义 Collection 并满足以下要求：

- ☐ Collection 名称叫做 rhel.user
- ☐ Collection 包含一个角色叫做 newuser
- ☐ 文件 tasks_main.yml 存放在 roles/newuser/tasks/main.yml
- ☐ 文件 users.conf 存放在 roles/newuser/files/users.conf
- ☐ 上传 Collection rhel.user 到 hub.lab.example.com 的 rhel 命名空间下

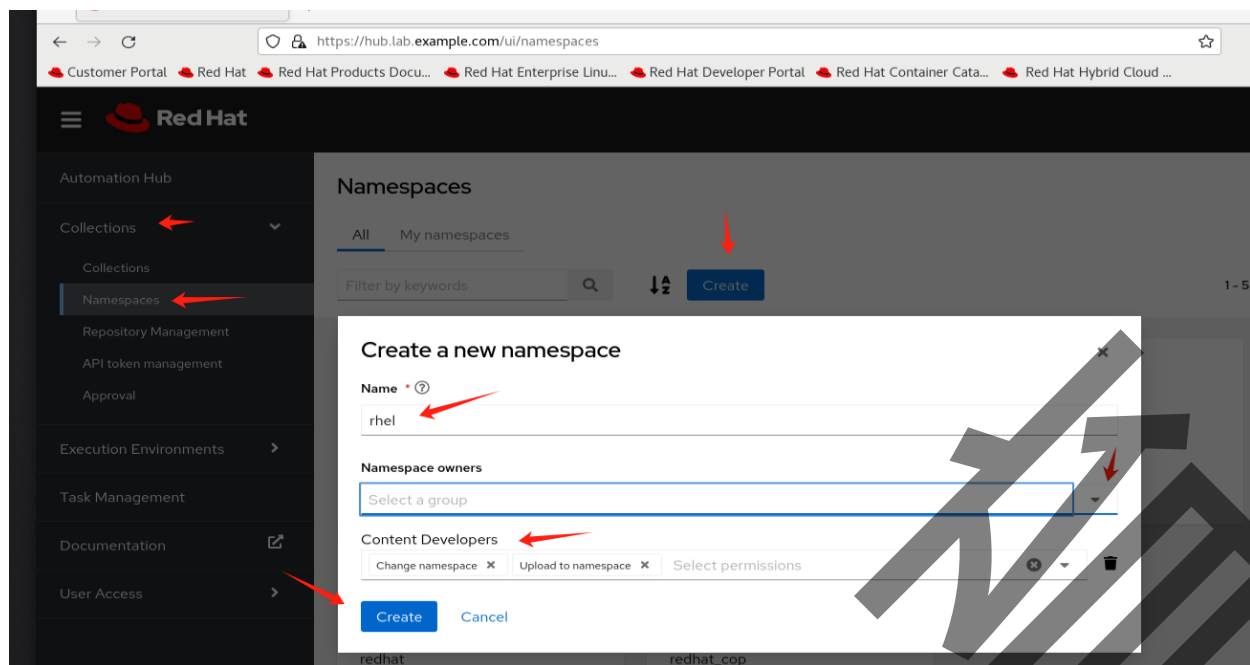
(如果 Collection 未显示在 Content Hub, 且已按照正确流程上传, 递增 Collection 的版本并再次上传)

```
1 # 解题
2 # 下载文件至本地
3 [student@workstation ~]$ git clone
https://git.lab.example.com/student/custom_collection.git
4 [student@workstation ~]$ cd custom_collection/
5 # 初始化角色
6 [student@workstation custom_collection]$ ansible-galaxy
collection init rhel.user
```

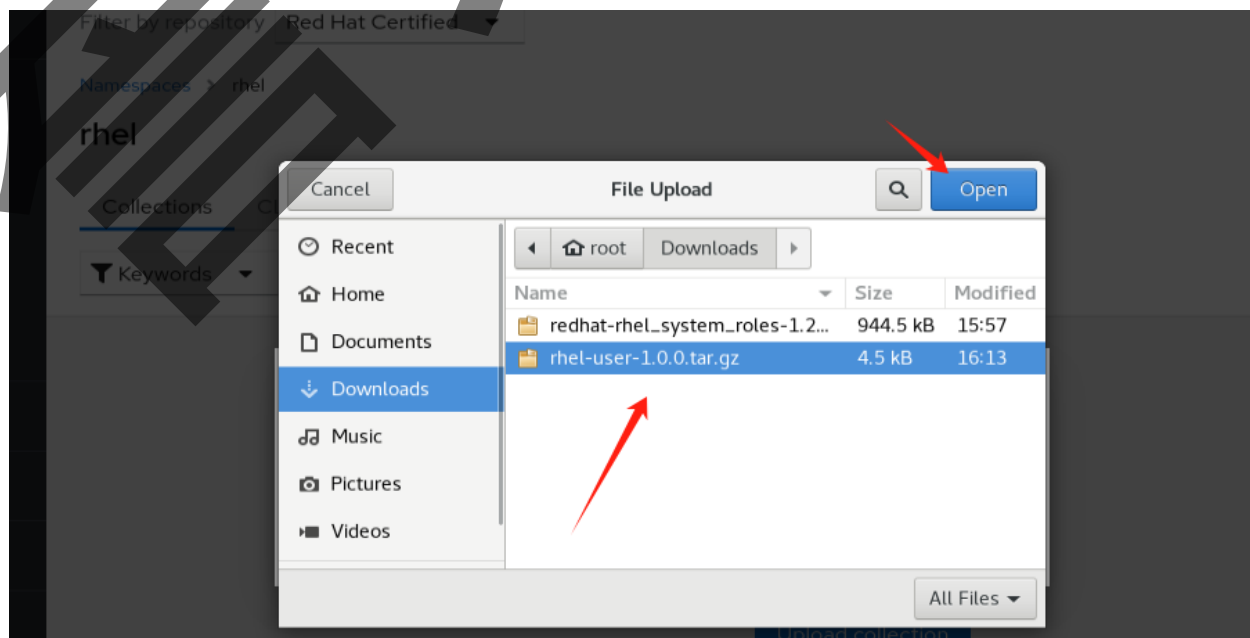
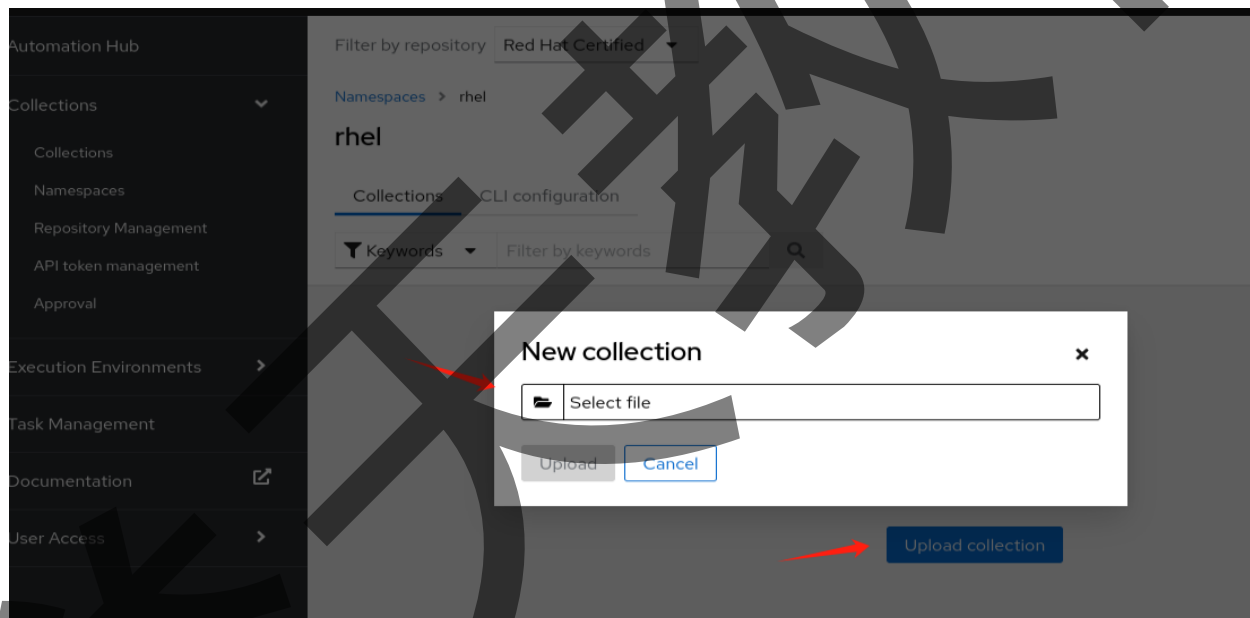
```
7         - collection rhel.user was created successfully
8         [student@workstation custom_collection]$ cd rhel/user/
9         [student@workstation user]$ ansible-galaxy role init
newuser --init-path roles
10         - Role newuser was created successfully
11         # 拷贝文件至角色内
12         [student@workstation user]$ cp
~/custom_collection/tasks_main.yml roles/newuser/tasks/main.yml
13         [student@workstation user]$ cp
~/custom_collection/users.conf roles/newuser/files/
14         # 新增meta目录文件
15         [student@workstation user]$ mkdir meta
16         [student@workstation user]$ echo 'requires_ansible:
">=2.90"' > meta/runtime.yml
17         [student@workstation user]$ cat meta/runtime.yml
18         requires_ansible: ">=2.90"
19         # 构建集合压缩包
20         [student@workstation user]$ ansible-galaxy collection
build
21         Created collection for rhel.user at
/home/student/custom_collection/rhel/user/rhel-user-1.0.0.tar.gz
22         [student@workstation user]$ ls | grep tar
23         rhel-user-1.0.0.tar.gz
24         # 拷贝至F0 /root/Downloads/ 目录
25         [root@foundation0 ~]# scp
workstation:~/custom_collection/rhel/user/rhel-user-1.0.0.tar.gz
/root/Downloads/
26         rhel-user-1.0.0.tar.gz
100% 4522 4.3MB/s 00:00
```

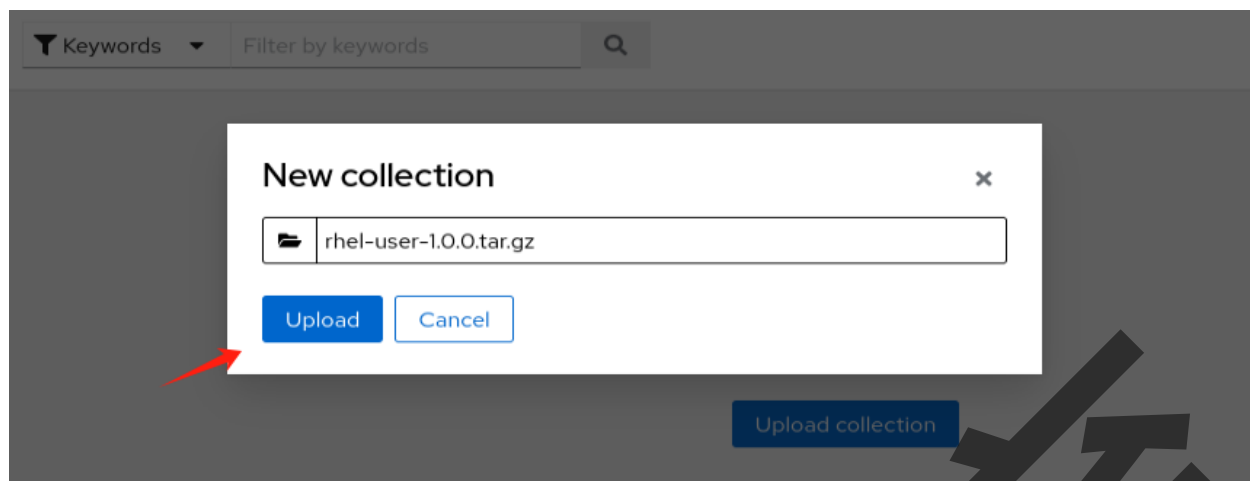
上传至hub服务器

1、创建命名空间（考试中已存在）

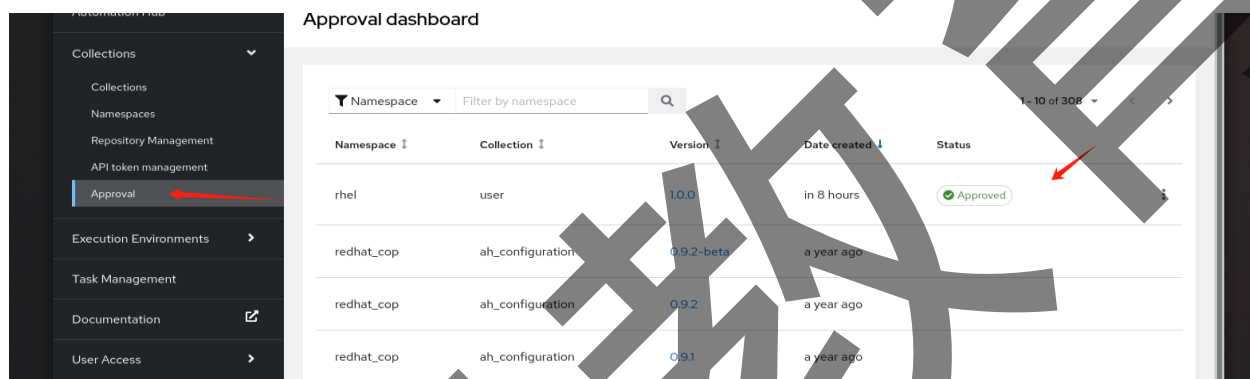


2、上传至命名空间内





3、查看集合



9. 构建自定义执行环境-1

根据下列要求构建自定义执行环境：

- ☐ 自定义执行环境命名为 `ee-user-supported:2.2`
- ☐ 执行环境基本镜像为 `ee-29-rhel8:latest`
- ☐ 执行环境的构建镜像为 `ansible-builder-rhel8:latest`
- ☐ 自定义执行环境包含 `rhel.user` 的自定义 Collection

上传新的执行环境到 `hub.lab.example.com` 中

```
1 # 解题
2 # 创建基础目录
3 [student@workstation ~]$ mkdir bascis1
4 [student@workstation ~]$ cd bascis1/
5 # 编辑yaml文件
```

```

6      [student@workstation bascis1]$ cat execution-
environment.yml
7      ---
8      version: 1
9      build_arg_defaults:
10         EE_BASE_IMAGE: ee-29-rhel8:latest
11         EE_BUILDER_IMAGE: ansible-builder-
rhel8:latest
12         dependencies:
13             galaxy: requirements.yml
14      [student@workstation bascis1]$ cat requirements.yml
15         collections:
16             - name: /build/rhel-user-1.0.0.tar.gz
17
18      # 初步构建目录
19      [student@workstation bascis1]$ ansible-builder create
20      Complete! The build context can be found at:
/home/student/bascis1/context
21      # 拷贝所需文件
22      [student@workstation bascis1]$ cp
~/custom_collection/rhel/user/rhel-user-1.0.0.tar.gz
context/_build/
23      # 构建导航器
24      [student@workstation bascis1]$ ansible-builder build -t
hub.lab.example.com/ee-user-supported:2.2
25      Running command:
26      podman build -f context/Containerfile -t
hub.lab.example.com/ee-user-supported:2.2 context
27      Complete! The build context can be found at:
/home/student/bascis1/context
28      # 上传导航器至hub
29      [student@workstation bascis1]$ podman push
hub.lab.example.com/ee-user-supported:2.2

```

10. 构建自定义执行环境-2

按照下列要求创建自定义执行环境：

- ☐ 执行环境叫做 ee-dyninventory:1.0
- ☐ 执行环境使用的基本镜像为 ee-29-rhel8:latest
- ☐ 执行环境使用的构建镜像为 ansible-builder-rhel8:latest

☐ 执行环境包含 Python36 软件包: python3-ldap 和 python36

上传执行环境到 hub.lab.example.com

```
1 # 解题
2 # 复制上题目录减少工作量
3 [student@workstation ~]$ cp -r bascis1/ bascis2/
4 [student@workstation ~]$ cd bascis2
5 # 修改构建文件
6 [student@workstation bascis2]$ cat execution-
environment.yml
7 ---
8 version: 1
9 build_arg_defaults:
10     EE_BASE_IMAGE: ee-29-rhel8:latest
11     EE_BUILDER_IMAGE: ansible-builder-
rhel8:latest
12 dependencies:
13     system: bindep.txt
14 [student@workstation bascis2]$ cat bindep.txt
15     python36 [platform:rpm]
16     python3-ldap [platform:rpm]
17 # 初步构建目录
18 [student@workstation bascis2]$ ansible-builder create
19 # 构建导航器
20 [student@workstation bascis2]$ ansible-builder build -t
hub.lab.example.com/ee-dyninventory:1.0
21 Running command:
22     podman build -f context/Containerfile -t ee-
dyninventory:1.0 context
23 Complete! The build context can be found at:
/home/student/bascis2/context
24 # 上传至hub
25 [student@workstation bascis2]$ podman push
hub.lab.example.com/ee-dyninventory:1.0
```

11. 在执行环境中运行剧本

使用 https://git.lab.example.com/student/dynamic_inventory.git 完成下列要求:

该仓库中包含以下资源:

- ☐ ansible 默认配置文件 `ansible.cfg`
- ☐ 动态清单脚本 `ldap-freeipa.py`

注意: `ldap-freeipa.py` 要求包含 `python36` 和 `python3-pip`

创建一个脚本命名为 `main.sh` , 用于在执行环境中运行 `main.yml` 的剧本, 满足以下要求:

- ☐ 使用 `ldap-freeipa.py` 作为清单
- ☐ 剧本叫做 `main.yml` 用于部署 `/etc/motd.d/banner` 文件, 内容为 `The Sun comes up and then it goes down`
- ☐ 剧本 `main.yml` 用在主机组 `testing`, 除此之外其他主机不应接受该文件
- ☐ `commit` 并 `push` 变更到 `git` 仓库

```
1 # 解题
2 # 下载文件至本地
3 [student@workstation ~]$ git clone
https://git.lab.example.com/student/dynamic_inventory.git
4 [student@workstation ~]$ cd dynamic_inventory
5 # 编写剧本文件
6 [student@workstation dynamic_inventory]$ cat main.yml
7 ---
8 - hosts: testing
9   become: yes
10  tasks:
11    - name: Create a directory if it does not
12      exist
13      ansible.builtin.file:
14        path: /etc/motd.d
15        state: directory
16    - name: Copy using inline content
17      ansible.builtin.copy:
18        content: 'The Sun comes up and then it
19        goes down'
20        dest: /etc/motd.d/banner
21 # 编写脚本文件
22 [student@workstation dynamic_inventory]$ cat main.sh
23 #!/bin/bash
```



```

22         ansible-navigator run main.yml -i ldap-
    freeipa.py --eei hub.lab.example.com/ee-dyninventory:1.0
23 # 修改权限试运行
24         [student@workstation dynamic_inventory]$ chmod +x
    main.sh ldap-freeipa.py
25         [student@workstation dynamic_inventory]$ ./main.sh
26 # 上传至git
27         [student@workstation dynamic_inventory]$ git add .
28         [student@workstation dynamic_inventory]$ git commit -m
    11
29         [student@workstation dynamic_inventory]$ git push

```

```

1 # 验证
2         [student@workstation dynamic_inventory]$ ansible -i ldap-
    freeipa.py testing -a 'cat /etc/motd.d/banner'
3             serverf | CHANGED | rc=0 >>
4             The Sun comes up and then it goes down
5             serverf.lab.example.com | CHANGED | rc=0 >>
6             The Sun comes up and then it goes down

```

12. 在剧本中使用变量

使用 https://git.lab.example.com/student/master_playbook.git 完成以下要求:

仓库内容包含以下资源:

- ☒ ansible 默认配置文件 ansible.cfg
- ☐ 动态清单脚本 inventory.py

创建剧本用于部署文件并且使用下列变量:

- ☐ 剧本命名为 master_playbook.yml
- ☐ 剧本运行在主机组 testing 上

剧本有 3 个变量

- ☐
 - content
 - directory
 - file

- ☐ 剧本部署在 directory 目录下的包含有 content 内容的 file 文件
- ☐ 其他主机不应接受该文件
- ☐ commit 并 push 变更到 git 仓库

```
1 # 解题
2 # 下载文件至本地
3 [student@workstation ~]$ git clone
https://git.lab.example.com/student/master_playbook.git
4 [student@workstation ~]$ cd master_playbook/
5 # 编写剧本文件
6 [student@workstation master_playbook]$ cat
master_playbook.yml
7 ---
8 - hosts: testing
9   become: yes
10  tasks:
11    - name: Create a directory
12      ansible.builtin.file:
13        path: "{{ directory }}"
14        state: directory
15        when: directory is defined
16    - ansible.builtin.debug:
17      msg: please define directory
18      when: directory is undefined
19    - name: Copy using inline content
20      ansible.builtin.copy:
21        content: "{{ content }}"
22        dest: "{{ directory }}/{{ file }}"
23      when:
24        - directory is defined
25        - file is defined
26        - content is defined
27    - ansible.builtin.debug:
28      msg: please define file variable
29      when:
30        - directory is defined
31        - file is undefined
32        - content is defined
33    - ansible.builtin.debug:
34      msg: please define content
variable
```

```

35         when:
36             - directory is defined
37             - file is defined
38             - content is undefined
39     # 修改主机清单脚本权限
40     [student@workstation master_playbook]$ chmod +x
inventory.py
41     # 试运行
42     [student@workstation master_playbook]$ ansible-navigator
run master_playbook.yml \
43         -i inventory.py \
44         -e directory=/tmp/yutian \
45         -e content="yutian" \
46         -e file=yutian \
47         --eei hub.lab.example.com/ee-dyninventory:1.0
48     # 上传至git
49     [student@workstation master_playbook]$ git add .
50     [student@workstation master_playbook]$ git commit -m 13
51     [student@workstation master_playbook]$ git push

```

```

1  # 验证
2      [student@workstation master_playbook]$ ansible testing -
i inventory.py -a 'cat /tmp/yutian/yutian'
3      serverf.lab.example.com | CHANGED | rc=0 >>
4      yutian
5

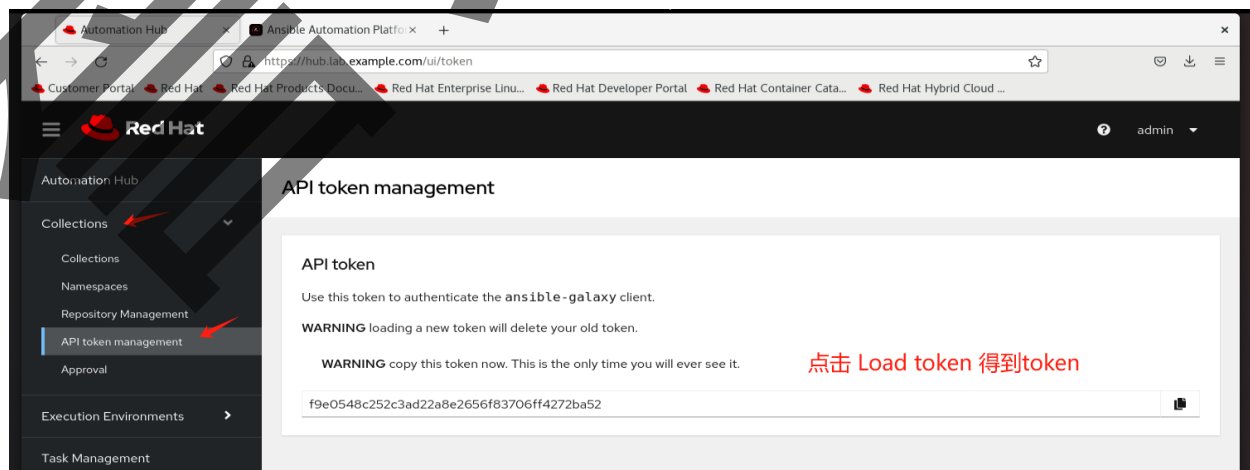
```

13. 创建剧本

使用 git 仓库 https://git.lab.example.com/student/master_user.git 完成以下内容:

- ☐ 创建叫做 main.yml 的剧本
- ☐ 剧本使用 rhel.user Collection
- ☐ 当剧本运行时, 使用 newuser 角色创建用户
- ☐ 用户仅在清单 testing 主机组中创建
- ☐ commit 并 push 变更到 git 仓库

```
1 # 解题
2 # 下载文件至本地
3 [student@workstation ~]$ git clone
https://git.lab.example.com/student/master_user.git
4 [student@workstation ~]$ cd master_user/
5 # 修改主机清单
6 [student@workstation master_user]$ cat inventory
7 [testing]
8 servera
9 # 修改ansible配置文件
10 [student@workstation master_user]$ cat ansible.cfg
11 [defaults]
12 collections_paths = /home/student/mycollections
13 inventory = ./inventory
14 remote_user = devops
15 ask_pass = false
16
17 [privilege_escalation]
18 become = false
19 become_method = sudo
20 become_user = root
21 become_ask_pass = false
22
23 [galaxy]
24 server_list = published_repo
25 [galaxy_server.published_repo]
26
27 url=https://hub.lab.example.com/api/galaxy/content/published/
token=f9e0548c252c3ad22a8e2656f83706ff4272ba52
# 注意替换，方式为下图
```



```
1 # 安装集合
```

```

2      [student@workstation master_user]$ ansible-galaxy
collection install rhel.user -p /home/student/mycollections/
3      # 编写剧本使用角色
4      [student@workstation master_user]$ cat main.yml
5      ---
6      - hosts: testing
7        become: yes
8        roles:
9          - rhel.user.newuser
10     # 试运行
11     [student@workstation master_user]$ ansible-navigator run
main.yml --ee false
12     [student@workstation master_user]$ rm -rf inventory
13     # 上传至git
14     [student@workstation master_user]$ git add .
15     [student@workstation master_user]$ git commit -m 13
16     [student@workstation master_user]$ git push
17

```

图形化题目

14. 配置项目

创建 Ansible automation controller 项目:

Name: D0374 copy file project

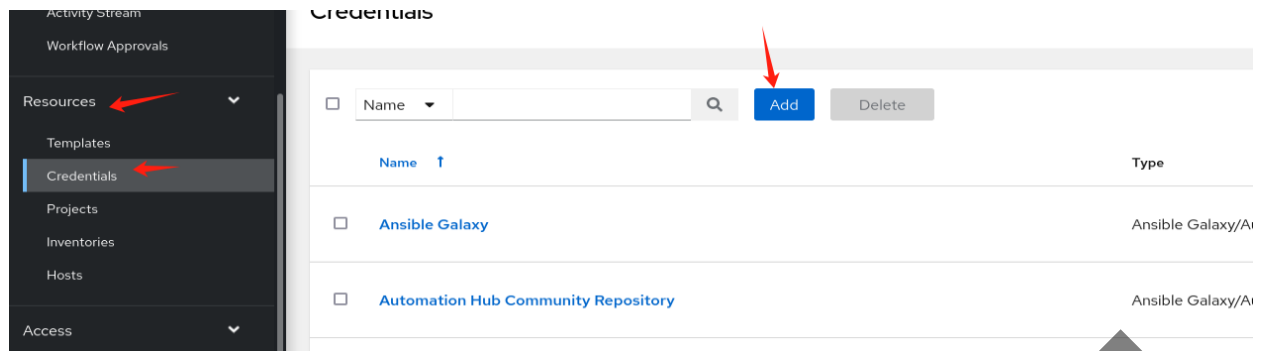
- ☐ Organization: Default
- ☐ Credential Type: Git
- ☐ URL: https://git.lab.example.com/student/master_playbook

Name: D0374 user project

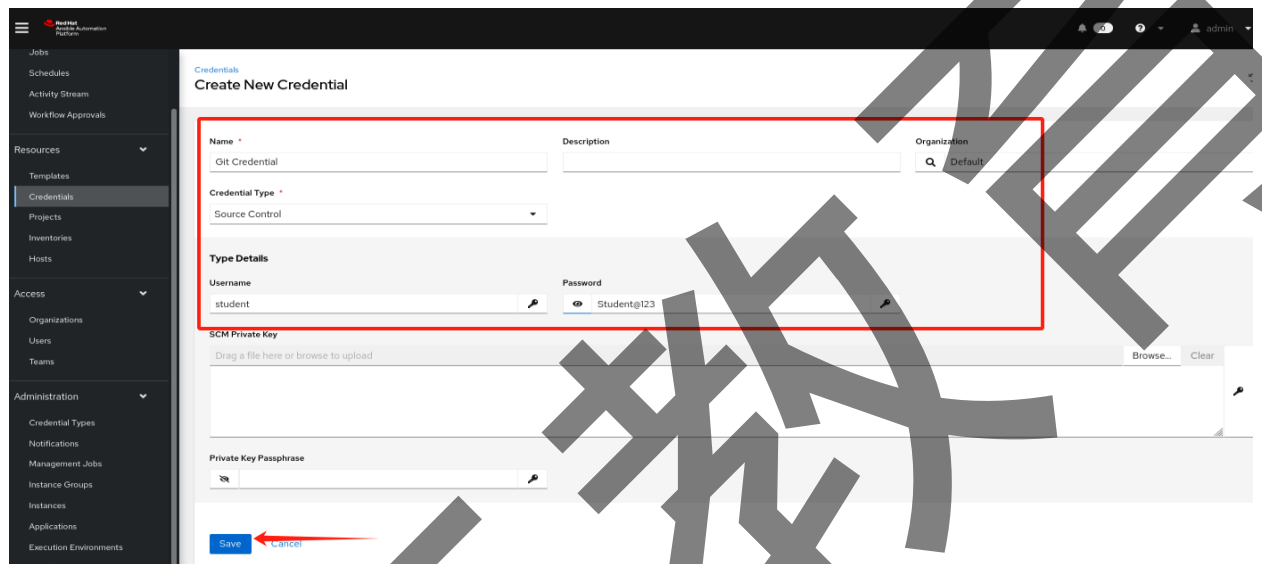
- ☐ Organization: Default
- ☐ Credential Type: Git
- ☐ URL: https://git.lab.example.com/student/master_user

登录至<https://controller.lab.example.com> 进行完成, admin/redhat

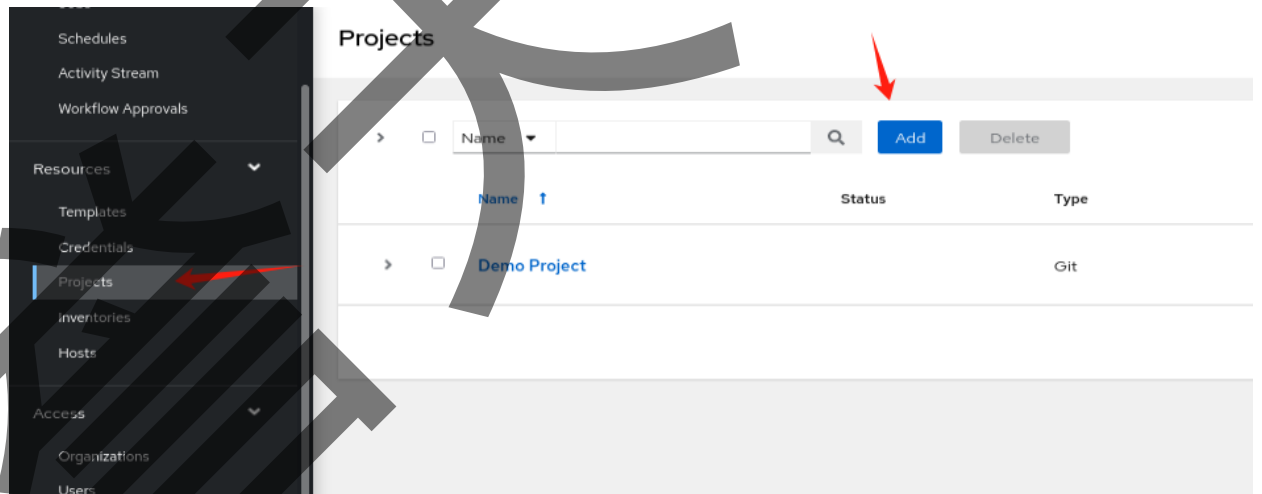
1、添加证书信息



2、保存证书信息



3、添加DO374 copy file project 项目



Projects

Create New Project

Name *

DO374 copy file project

Description

Organization *

Q Default

Execution Environment ⓘ

Q

Source Control Type *

Git

Type Details

Source Control URL * ⓘ

https://git.lab.example.com/student/master_playbook

Source Control Branch/Tag/Commit ⓘ

Source Control Refspec ⓘ

Source Control Credential

Q Git Credential

Options

☐ Clean ⓘ

☐ Delete ⓘ

☐ Track submodules ⓘ

☐ Update Revision on Launch ⓘ

☐ Allow Branch Override ⓘ

Save

Cancel

4、添加 DO374 user project 项目

Projects

>	<input type="checkbox"/>	Name	Q	Add	Delete		1 - 2 of 2
		Name ↑		Status	Type	Revision	
>	<input type="checkbox"/>	Demo Project			Git	Sync for revision	↻
>	<input type="checkbox"/>	DO374 copy file project		Successful	Git	4f173d2	↻

1 - 2 of 2 items

Projects

Create New Project

Name *

DO374 user project

Description

Organization *

Q Default

Execution Environment ⓘ

Q

Source Control Type *

Git

Type Details

Source Control URL * ⓘ

https://git.lab.example.com/student/master_user

Source Control Branch/Tag/Commit ⓘ

Source Control Refspec ⓘ

Source Control Credential

Q Git Credential

Options

☐ Clean ⓘ

☐ Delete ⓘ

☐ Track submodules ⓘ

☐ Update Revision on Launch ⓘ

☐ Allow Branch Override ⓘ

Save

Cancel

15. 配置清单

按照下列要求配置 Ansible automation controller 清单:

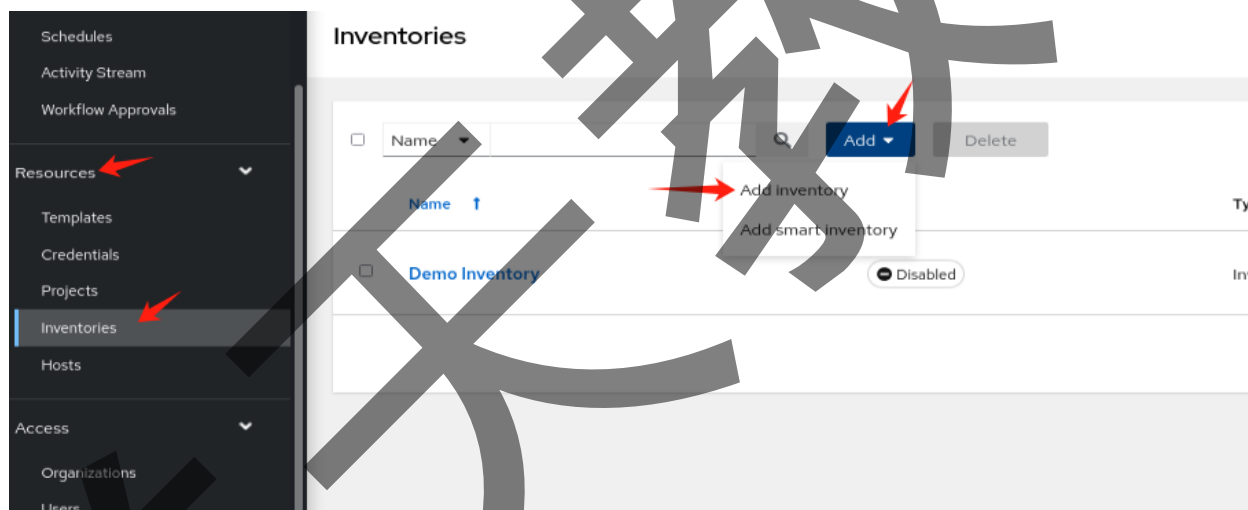
D0374 static inventory 包含下列主机组:

- ☐
 - 主机组 development 包含 serverb.lab.example.com
 - 主机组 testing 包含 servera.lab.example.com
- ☐ D0374 dynamic inventory 包含 D0374 custom source, 其中包含项目 D0374 copy file project 中的清单脚本 inventory.py

注意: 在启动前清单源会自动更新

除了上面提到的资源外, 不要创建任何其他资源

1、创建D0374 static inventory



Create new inventory

Name *

DO374 static inventory

Description

Organization *

Q

Default

Instance Groups

Q

Labels ⓘ

Variables ⓘ

YAML

JSON

1

Save

Cancel

Inventories > DO374 static inventory

Groups

Back to Inventories

Details

Access

Groups

Hosts

Sources

Jobs

Job Templates

☐

Name


▼

Q

Add

Run Command

Delete



No Items Found

Please add items to populate this list

Inventories > DO374 static inventory > Groups

Create new group

Name *

development

Variables

YAML

JSON

1

Save

Cancel

Inventories > DO374 static inventory > Groups > development

Hosts

Back to Groups

Details

Related Groups

Hosts



Name



Add

Run Command

Disassociate

Add existing host

Add new host



No Hosts Found

Please add Hosts to populate this list

[Inventories](#) > [DO374 static inventory](#) > [Groups](#) > [development](#) > [Hosts](#)

Create new host

Name *

serverb.lab.example.com

Description

Variables

YAML

JSON

1 ---

2

Save

Cancel

[Inventories](#) > [DO374 static inventory](#)

Groups

[Back to Inventories](#)

[Details](#)

[Access](#)

[Groups](#)

[Hosts](#)

[Sources](#)

[Jobs](#)

[Job Templates](#)



Name



Add

Run Command

Delete

Name ↑



development

1 - 1 of 1

Inventories > DO374 static inventory > Groups

Create new group

Name *

testing

Variables

YAML

JSON

1 ---

Save

Cancel

Inventories > DO374 static inventory > Groups > testing

Hosts

Back to Groups

Details

Related Groups

Hosts

☐ Name



Add

Run Command

Disas

Add existing host

Add new host



No Hosts F

Please add Hosts to p

Inventories > DO374 static inventory > Groups > testing > Hosts

Create new host

Name * Description

Variables YAML JSON

1	---
2	

Save Cancel

2、创建DO374 dynamic inventory

Red Hat Ansible Automation Platform

Views

- Dashboard
- Jobs
- Schedules
- Activity Stream
- Workflow Approvals

Resources

- Templates
- Credentials
- Projects
- Inventories**
- Hosts

Access

Inventories

☐ Name

Name ↑

- ☐ **Demo Inventory** Disabled
- ☐ **DO374 static inventory** Disabled

Add

- Add inventory
- Add smart inventory

Inventories

Create new inventory

Name *

DO374 dynamic inventory

Description

Organization *

Q

Default

Instance Groups

Q

Labels ⓘ

Variables ⓘ

YAML

JSON

1

Save

Cancel

Inventories > DO374 dynamic inventory

Sources

Back to Inventories

Details

Access

Groups

Hosts

Sources

Jobs

Job Templates


☐

Name

Q

Add

Delete



No Inventory Sources Found

Please add inventory Sources to populate this list

Inventories > DO374 dynamic inventory > Sources

Create new source

Name *

DO374 custom source

Description

Execution Environment

Q

Source *

Sourced from a Project

Source details

Created at

Q

Verify ⓘ

1 (Info)

Enabled Value ⓘ

Update options

☐ Overwrite ⓘ

☐ Overwrite variables ⓘ

☒ Update on search ⓘ

☐ Update on project update ⓘ

Check timeout (seconds) ⓘ

0

Source variables ⓘ

YAML

JSON

1

2

Save

Cancel

Project *

Q

DO374 my file project

Host Filter ⓘ

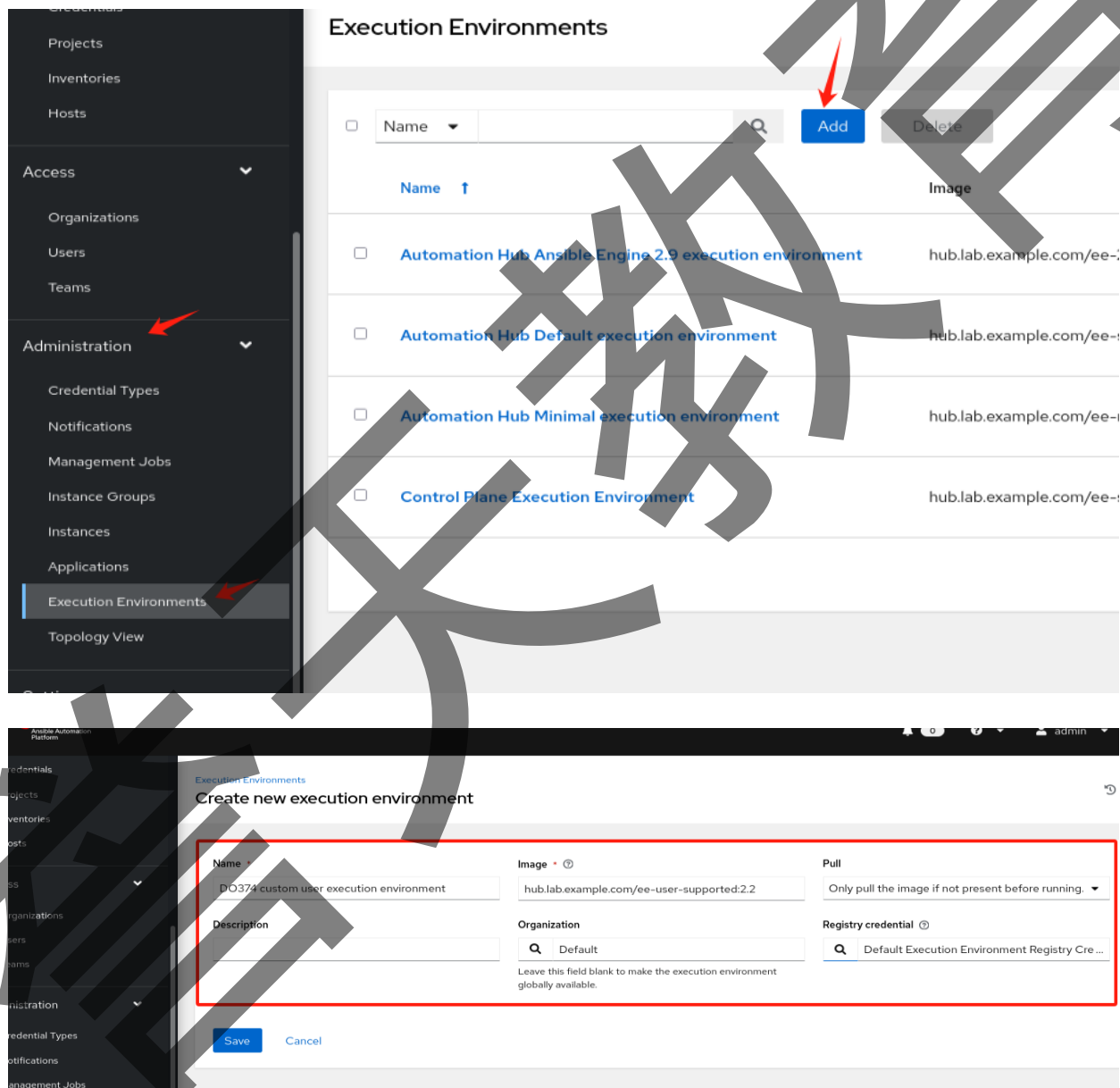
Inventory file ⓘ

inventory.py

Enabled Variable ⓘ

16. 配置 Ansible 执行环境控制器执行环境

- ☐ 在 Ansible automation controller 中使用 `hub.lab.example.com/ee-user-supported:2.2` 执行镜像
 - 创建一个名为 `DO374 custom user execution environment` 的执行环境
 - ☐ 境
- 如果在运行前镜像不存在，请先下载



17. 配置模板

按照下列要求创建 Ansible automation controller 作业模板:

模板 `DO374 static copy project template`

- 当启动模板，则会以 DO374 static inventory 为清单中主机运行在项目 DO374 copy file project 中的 master_playbook.yml
- 设置 DO374 static copy project 模板的变量

```
1 ---
2 directory: "/etc/motd.d"
3 file: "todays_message"
4 content: "where there is a will, there is a way"
```

模板 DO374 dynamic copy project template

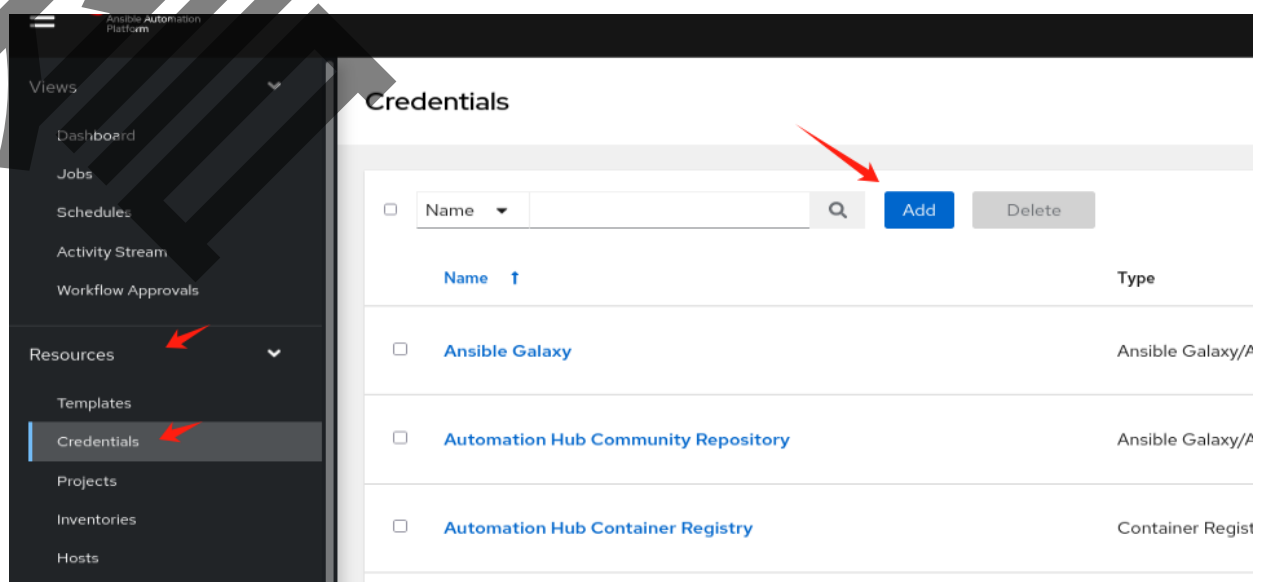
- 当启动模板，则以 DO374 dynamic inventory 为清单中主机运行在项目 DO374 copy file project 中的 master_playbook.yml
- 设置 DO374 dynamic copy project 模板的变量

```
1 ---
2 directory: "/etc/issue.d"
3 file: "todays_issue"
4 content: "Experience is the mother of wisdom."
```

模板 DO374 user project template

- 当启动模板，则以 DO374 static inventory 为清单主机运行在项目 DO374 user project 中的 main.yml 剧本
- 模板使用 DO374 custom user execution execution environment 执行环境

1、创建证书



Create New Credential

Name *
host credential

Description

Organization
Default

Credential Type *
Machine

Type Details

Username
student

Password
student

☐ Prompt on launch

SSH Private Key
Drag a file here or browse to upload

Signed SSH Certificate
Drag a file here or browse to upload

Private Key Passphrase
Prompt on launch

Privilege Escalation Method
sudo

Privilege Escalation Username
root

Privilege Escalation Password
student

☐ Prompt on launch

Save Cancel

2、创建 DO374 static copy project template

Templates

Views
Dashboard
Jobs
Schedules
Activity Stream
Workflow Approvals

Resources
Templates
Credentials
Projects
Inventories
Hosts

Templates

Name
Add job template
Add workflow template

Delete

Type
Job Template

Last Ran

Demo Job Template

1 - 1 of 1 items

Create New Job Template

Name *
DO374 static copy project template

Description

Job Type *
Run

☐ Prompt on launch

Inventory *
DO374 static inventory

☐ Prompt on launch

Project *
DO374 copy file project

Execution Environment *

Playbook *
master_playbook.yml

Credentials *
SSH root credential

Labels

Variables *
1. directory: "/etc/passwd"
2. file: "/etc/passwd"
3. content: "The sun goes down, and then the moon comes up"

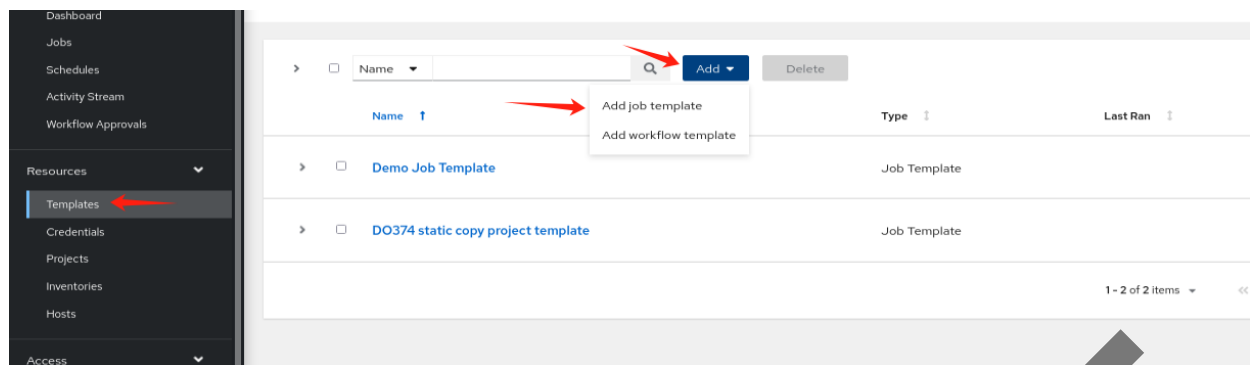
☐ Prompt on launch

Press Enter to edit. Press ESC to stop editing.

Forks
Limit
Verbosity

点击save

3、创建 DO374 dynamic copy project template



Templates

Create New Job Template

Name * DO374 dynamic copy project template

Description

Job Type * Ⓞ Run

Inventory * Ⓞ DO374 dynamic inventory

Project * Ⓞ DO374 copy file project

Execution Environment * Ⓞ

Playbook * Ⓞ master_playbook.yml

Credentials * Ⓞ SSH host credential X

Labels * Ⓞ

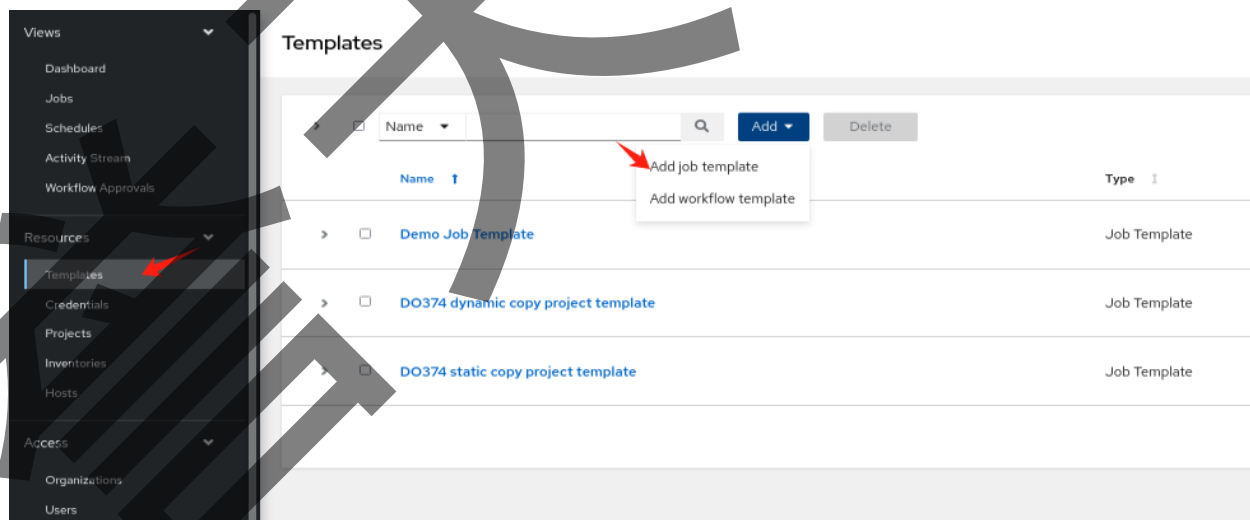
Variables * Ⓞ YAML JSON

```
1 ---
2 directory: "/etc/issue.d"
3 file: "todays issue"
4 content: "After the moon goes down, the sun comes up"
```

Forks * Ⓞ Limit * Ⓞ Prompt on launch Verbose * Ⓞ Prompt on launch

点击save

4、创建 DO374 user project template



Views

Dashboard

Jobs

Schedules

Activity Stream

Workflow Approvals

Resources

Templates

Credentials

Projects

Inventories

Hosts

Access

Organizations

Users

Teams

Administration

Create New Job Template

Name *

DO374 user project template

Description

Job Type *

Run

Prompt on launch

Inventory *

DO374 static inventory

Prompt on launch

Project *

DO374 user project

Execution Environment *

DO374 custom user execution environment

Playbook *

main.yml

Credentials *

SSH host credential X

Prompt on launch

Labels *

Variables *

YAML JSON

Prompt on launch

1

2

点击save

盗用天擎病毒