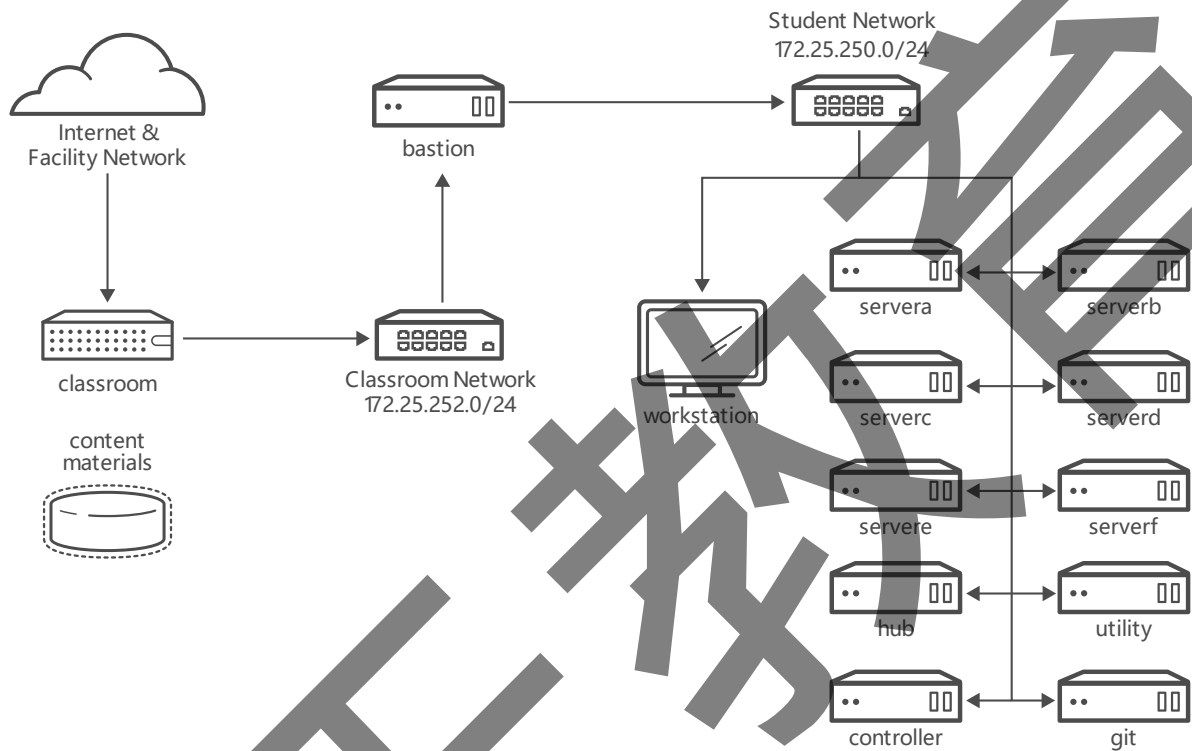


DO374备考资料

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

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
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

系统	IP地址	角色	ID
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 s://git.lab.example.com](http://s://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

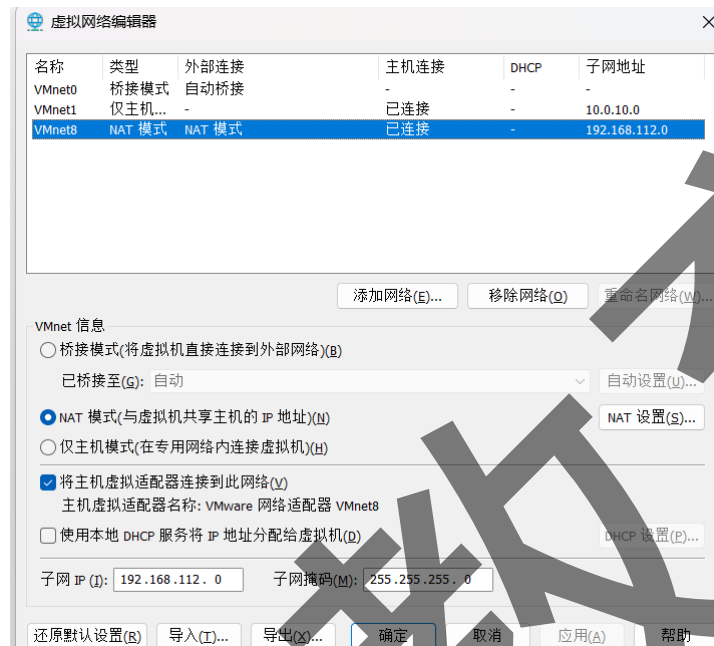
请注意，在评分之前，您的 Ansible 托管节点系统 将重置为考试开始时的初始状态，您编写的 Ansible playbook 将通过以 **student** 用户身份在控制节点上运行来加以应用。在 playbook 运行后，系统会重新启动您的托管节点，然后进行评估，以判断它们是否按照规定进行了配置。

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
4      ethernet  ens160
5 # 删除原有配置
6 [root@foundation0 ~]# nmcli connection delete "System
7      ens160"
8 # 新增配置
9 [root@foundation0 ~]# nmcli connection add ifname ens160
10      con-name ens160 type ethernet \
11      ipv4.method manual ipv4.addresses 192.168.112.90/24
12 # 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

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

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

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` 包含一个不完整的脚本，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）完善该脚本，实现：

- ☐ 当使用标签 `alpha` 来运行该脚本时，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` 来运行该脚本时，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脚本，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

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

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

- 脚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

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

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)

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

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`

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

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

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

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

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

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 D0374 custom user execution environment 的执行环境
- ☐ 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 D0374 static copy project template

- ☐ 当启动模板，则会以 D0374 static inventory 为清单中主机运行在项目 D0374 copy file project 中的 master_playbook.yml
- ☐ 设置 D0374 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 D0374 dynamic copy project template

- ☐ 当启动模板, 则以 D0374 dynamic inventory 为清单中主机运行在
项目 D0374 copy file project 中的master_playbook.yml
- ☐ 设置 D0374 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 D0374 user project template

- ☐ 当启动模板, 则以 D0374 static inventory 为清单主机运行在项
目 D0374 user project 中的main.yml 剧本
- ☐ 模板使用 D0374 custom user execution execution
environment 执行环境