【内部资料，请勿外传!!!!】

环境说明：

真机（RHEL8.2）：

server1.lab0.example.com 172.25.0.254/24
预设 root 口令为 tedu
提供 RHEL8 软件源 http://server1.lab0.example.com/rhel8/BaseOS
提供 RHEL8 软件源 http://server1.lab0.example.com/rhel8/AppStream
提供 DNS 服务，为区域 lab0.example.com 中相关站点提供解析
提供 NTP 网络时间服务；提供 NFS 文件服务，共享/rhome/ldapuser0 目录

虚拟机 control（RHEL8.2） —— 控制机：

control.lab0.example.com 172.25.254.100/24
预设授权 sudo 用户 alice，密码为 Asimov
预设 IP 地址 172.25.254.100/24，已做好主机名映射

虚拟机 node1~node5（RHEL8.2） —— 受管机：

node[1-5].lab0.example.com 172.25.254.10[1-5]/24
预设授权 sudo 用户 alice，密码为 Asimov
预设 IP 地址 172.25.254.101~105/24，已做好主机名映射

其他信息：

除非另有指定，否则所有工作都需要归属在控制机的 /home/alice/ansible/ 目录下

01. 安装和配置 ansible 环境

1）安装所需软件包
2）在/home/alice/ansible/inventory 文件中设置主机清单，要求：
node1 属于 test01 主机组
node2 属于 test02 主机组
node3 和 node4 属于 web 主机组
node5 属于 test05 主机组
web 组属于 webtest 主机组
3）在/home/alice/ansible 目录中创建 ansible.cfg，满足以下需求：
主机清单文件为 /home/alice/ansible/inventory
playbook 中角色位置为 /home/alice/ansible/roles

解题参考：

```
[alice@control ~]$ sudo  yum  -y  install  ansible              //安装 ansible 软件包
[alice@control ~]$ mkdir  -p  ~/ansible ; cd  ~/ansible/        //创建并进入 ansible 工作目录
[alice@control ansible]$ mkdir  ~/ansible/roles                 //创建角色目录

[alice@control ansible]$ vim  ansible.cfg
[defaults]
inventory = inventory                                           //主机清单文件
remote_user = alice                                             //连接受管机的远程用户名
roles_path = roles                                              //指定默认的角色目录

[privilege_escalation]                                          //设置用户 sudo 提权
become=True                                                     //需要提权
become_method=sudo                                              //提权方式为 sudo
```

```
become_user=root                                    //提权为 root
become_ask_pass=False                               //无需验证密码

[alice@control ansible]$ vim inventory              //配置主机清单
[test01]
node1
[test02]
node2
[web]
node3
node4
[test05]
node5
[webtest:children]
web
```

02. 创建和运行 Ansible 临时命令

编写脚本 /home/alice/ansible/adhoc.sh，用来为所有受管机配置 2 个 yum 仓库。

仓库 1：

名称为 BASE，描述为 software base URL 为 http://study.lab0.example.com/rhel8/BaseOS GPG 签名启用，GPG 秘钥 URL 为 http://study.lab0.example.com/rhel8/RPM-GPG-KEY-redhat-release 仓库为启用状态

仓库 2：

名称为 STREAM，描述为 software stream URL 为 http://study.lab0.example.com/rhel8/AppStream GPG 签名启用，GPG 秘钥 URL 为 http://study.lab0.example.com/rhel8/RPM-GPG-KEY-redhat-release 仓库为启用状态

解题参考：

```
[alice@control ansible]$ vim  adhoc.sh
#!/bin/bash
ansible all -m yum_repository -a 'name=BASE description="software base"
baseurl=http://study.lab0.example.com/rhel8/BaseOS                    gpgcheck=yes
gpgkey=http://study.lab0.example.com/rhel8/RPM-GPG-KEY-redhat-release enabled=yes'
ansible all -m yum_repository -a 'name=STREAM description="software stream"
baseurl=http://study.lab0.example.com/rhel8/AppStream                 gpgcheck=yes
gpgkey=http://study.lab0.example.com/rhel8/RPM-GPG-KEY-redhat-release enabled=yes'

[alice@control ansible]$ chmod  +x  adhoc.sh
[alice@control ansible]$ ./adhoc.sh
```

03. 编写剧本远程安装软件

创建名为/home/alice/ansible/tools.yml 的 playbook，能够实现以下目的：

1）将 php 和 tftp 软件包安装到 test01、test02 和 web 主机组中的主机上

2）将 RPM Development Tools 软件包组安装到 test01 主机组中的主机上

3）将 test01 主机组中的主机上所有软件包升级到最新版本

解题参考：

```
[alice@control ansible]$ vim  tools.yml
- hosts: test01, test02, web
  tasks:
    - yum: pkg="php,tftp" state=present                    //安装 php 和 tftp 软件包
- hosts: test01
  tasks:
    - yum: name="@RPM Development Tools" state=present     //安装 xx 包组
    - yum: name="*" state="latest"                         //升级所有包

[alice@control ansible]$ ansible-playbook tools.yml        //验证剧本
```

04. 安装并使用系统角色

安装 RHEL 角色软件包，并创建剧本 /home/alice/ansible/timesync.yml，满足以下要求：

1）在所有受管理节点运行
2）使用 timesync 角色
3）配置该角色,使用时间服务器 172.25.254.250,并启用 iburst 参数

解题参考：

```
[alice@control ansible]$ sudo  yum  -y install rhel-system-roles    //安装 rhel 系统角色
[alice@control ansible]$ cp -r /usr/share/ansible/roles/rhel-system-roles.timesync  roles/
                                                           //复制角色目录
[alice@control ansible]$ vim  timesync.yml
- hosts: all
  vars:
    - timesync_ntp_servers:                                //设置 NTD 服务器变量
      - hostname: 172.25.254.250
        iburst: yes
  roles:
    - rhel-system-roles.timesync                           //调用角色

[alice@control ansible]$ ansible-playbook timesync.yml     //验证剧本
```

05. 通过 galaxy 安装角色

创建剧本 /home/alice/ansible/roles/down.yml，用来从以下 URL 下载角色，并安装到 /home/alice/ansible/roles 目录下：

http://study.lab0.example.com/roles/haproxy.tar 此角色名为 haproxy
http://study.lab0.example.com/roles/myphp.tar 此角色名为 myphp

解题参考：

```
[alice@control ansible]$ vim  /home/alice/ansible/roles/down.yml     //配置角色导入信息
- name: haproxy
  src: http://study.lab0.example.com/roles/haproxy.tar
- name: myphp
  src: http://study.lab0.example.com/roles/myphp.tar

[alice@control ansible]$ ansible-galaxy  install  -r  roles/down.yml    //导入角色
```

06. 创建及使用自定义角色

根据下列要求,在/home/alice/ansible/roles 中创建名为 httpd 的角色:

1）安装 httpd 软件，并能够开机自动运行

2）开启防火墙，并允许 httpd 通过

3）使用模板 index.html.j2，用来创建/var/www/html/index.html 网页，内容如下（其中，HOSTNAME 是受管理节点的完全域名，IPADDRESS 是 IP 地址）：

```
Welcome to HOSTNAME on IPADDRESS
```

然后创建剧本 /home/alice/ansible/myrole.yml，为 webtest 主机组启用 httpd 角色。

解题参考：

```
[alice@control ~]$ ansible-galaxy  init  roles/httpd     //为新角色创建初始目录结构
[alice@control ~]$ vim  roles/httpd/templates/index.html.j2     //编写角色模板（网页）
Welcome to {{ ansible_fqdn }} on {{ ansible_eth0.ipv4.address }}

[alice@control ~]$ vim  roles/httpd/tasks/main.yml          //编写角色主任务
- yum: pkg=httpd state=present                              //装 httpd 包
- template: src=index.html.j2 dest=/var/www/html/index.html //配置网页
- service: name=httpd state=restarted enabled=yes           //起 httpd 服务
- firewalld: service=http state=enabled permanent=yes immediate=yes
                                                            //配置防火墙，允许访问 web 端口
[alice@control ansible]$ vim  myrole.yml                    //编写启动脚本
- hosts: webtest
  roles:
    - httpd

[alice@control ansible]$ ansible-playbook myrole.yml        //验证剧本
```

07. 使用之前通过 galaxy 下载的角色

创建剧本 /home/alice/ansible/web.yml，满足下列需求：

1）该剧本中包含一个 play，可以在 test05 主机组运行 haproxy 角色（此角色已经配置好网站的负载均衡服务）

2）多次访问 http://node5.lab0.example.com 可以输出不同主机的欢迎页面

3）该剧本中包含另一个 play，可以在 webtest 主机组运行 myphp 角色（此角色已经配置好网站的 php 页面）

4）多次访问 http://node5.lab0.example.com/index.php 也输出不同主机的欢迎页面

解题参考：

```
[alice@control ansible]$ vim  web.yml
- name: use role B                                          //先部署 web 节点（node3、node4）
  hosts: webtest
  roles:
    - myphp
- name: use role A
  hosts: test05                                             //再配置负载均衡器（node5）
  roles:
    - haproxy
  tasks:
    - firewalld: service=http state=enabled permanent=yes immediate=yes
                                        //配置防火墙，允许访问负载均衡器的 web 端口
[alice@control ansible]$ ansible-playbook web.yml           //验证剧本
```

08. 编写剧本远程管理逻辑卷

创建剧本 /home/alice/ansible/lvm.yml，用来为所有受管机完成以下部署：

1）在卷组 search 中创建名为 mylv 的逻辑卷，大小为 1000MiB
2）使用 ext4 文件系统格式化该逻辑卷
3）如果无法创建要求的大小，应显示错误信息 insufficient free space，并改为 500MiB
4）如果卷组 search 不存在，应显示错误信息 VG not found
5）不需要挂载逻辑卷

解题参考：

```
[alice@control ansible]$ vim  lvm.yml
- name: manager volume
  hosts: all
  tasks:
    - name: 1. failed when VG not found
      debug: msg="VG not found"
      when: "'search' not in ansible_lvm.vgs"                    //目标 VG 不存在时报错
      failed_when: "'search' not in ansible_lvm.vgs"             //停止后续任务
    - name: 2. lvcreate
      block:                                                     //配置指令块
        - lvol: lv=mylv size=1000M vg=search  force=yes
      rescue:                                                    //若块操作失败，则执行补救
        - debug: msg="insufficient free space"
        - lvol: lv=mylv size=500M vg=search  force=yes
      always:                                                    //始终需要执行的任务
        - filesystem: dev=/dev/search/mylv fstype=ext4 force=yes

[alice@control ansible]$ ansible-playbook lvm.yml              //验证剧本
```

09. 根据模板部署主机文件

1）从 http://study.lab0.example.com/materials/newhosts.j2 下载模板文件
2）完成该模板，用来生成新主机清单（主机的显示顺序没有要求），结构如下

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

172.25.254.101 node1.lab0.example.com node1
172.25.254.102 node2.lab0.example.com node2
172.25.254.103 node3.lab0.example.com node3
172.25.254.104 node4.lab0.example.com node4
172.25.254.105 node5.lab0.example.com node5
```

3）创建剧本 /home/alice/ansible/newhosts.yml，它将使用上述模板在 test01 主机组的主机上生成文件/etc/newhosts

解题参考：

```
[alice@control ansible]$ sudo yum -y  install   wget
[alice@control ansible]$ wget http://study.lab0.example.com/materials/newhosts.j2
[alice@control ansible]$ vim  newhosts.j2                      //制作 J2 动态模板文件
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

```
{% for id in groups.all %}
{{hostvars[id].ansible_eth0.ipv4.address}}                    {{hostvars[id].ansible_fqdn}}
{{hostvars[id].ansible_hostname}}
{% endfor %}

[alice@control ansible]$ vim  newhosts.yml
- hosts: all                                                  //搜集所有主机的信息
- hosts: test01                                               //只为 xx 主机组部署
  tasks:
    - template: src=newhosts.j2 dest=/etc/newhosts force=yes  //通过模板部署文件

[alice@control ansible]$ ansible-playbook newhosts.yml       //验证剧本
```

10. 编写剧本修改远程文件内容

创建剧本 /home/alice/ansible/newissue.yml，满足下列要求：

1）在所有清单主机上运行，替换/etc/issue 的内容
2）对于 test01 主机组中的主机，/etc/issue 文件内容为 test01
3）对于 test02 主机组中的主机，/etc/issue 文件内容为 test02
4）对于 web 主机组中的主机，/etc/issue 文件内容为 Webserver

解题参考：

```
[alice@control ansible]$ vim  newissue.yml
- name: deploy /etc/issue
  hosts: all
  tasks:
    - template: src=newissue.j2 dest=/etc/issue force=yes     //根据模板部署远程文件

[alice@control ansible]$ vim  newissue.j2                     //编写模板内容
{% if "test01" in group_names %}                              //如果所在组包括 test01
test01
{% elif "test02" in group_names %}                            //如果所在组包括 test02
test02
{% elif "web" in group_names %}                               //如果所在组包括 web
Webserver
{% endif %}

[alice@control ansible]$ ansible-playbook newissue.yml        //验证剧本
```

11. 编写剧本部署远程 Web 目录

创建剧本 /home/alice/ansible/webdev.yml，满足下列要求：

1）在 test01 主机组运行
2）创建目录/webdev，属于 webdev 组，常规权限为 rwxrwxr-x，具有 SetGID 特殊权限
3）使用符号链接/var/www/html/webdev 链接到/webdev 目录
4）创建文件/webdev/index.html，内容是 It's works!
5）查看 test01 主机组的 web 页面 http://node1/webdev/ 将显示 It's works!

解题参考：

```
[alice@control ansible]$ vim  webdev.yml
- name: Prepare Web Directory
```

```
  hosts: test01
  tasks:
    - yum: name=httpd state=present                                          //装 httpd 包
    - group: name=webdev state=present                                       //配置组账号
    - file: name=/webdev group=webdev mode=2775 state=directory              //配置目录
    - file: src=/webdev name=/var/www/html/webdev state=link force=yes       //配置链接
    - copy: content="It's works!" dest=/webdev/index.html force=yes          //配置网页
    - firewalld: service=http state=enabled permanent=yes immediate=yes //配置防火墙
    - service: name=httpd state=restarted enabled=yes                        //起 httpd 服务

    - command: setenforce 0                                                  //取消 SELinux 限制
    - command: sed -i '/^SELINUX=/cSELINUX=Permissive'  /etc/selinux/config //永久取消

[alice@control ansible]$ ansible-playbook webdev.yml                         //验证剧本
```

12. 编写剧本为受管机生成硬件报告

创建名为/home/alice/ansible/hardware.yml 的 playbook，满足下列要求：

1）使所有受管理节点从以下 URL 下载文件：

http://study.lab0.example.com/materials/hardware.empty

2）并用来生成以下硬件报告信息，存储在各自的/root/hardware.txt 文件中

```
清单主机名称
以 MB 表示的总内存大小
BIOS 版本
硬盘 vda 的大小
硬盘 vdb 的大小
```

其中，文件的每一行含有一个 key=value 对，如果项目不存在，则显示 NONE。

解题参考：

```
[alice@control ansible]$ vim  hardware.yml
- name: hardware report
  hosts: all
  vars:
    - host: "{{inventory_hostname}}"                                  //提取清单主机名
    - mem: "{{ansible_memtotal_mb}}"                                  //提取总内存大小（MB）
    - bios: "{{ansible_bios_version}}"                                //提取 BIOS 版本
    - vdasize: "{{ansible_devices.vda.size}}"                         //提取磁盘 vda 大小
    -  vdbsize:  "{{ansible_devices.vdb.size  if  ansible_devices.vdb.size  is  defined  else
'NONE' }}"                                                            //提取磁盘 vdb 大小，或 NONE
  tasks:                                                              //制作硬件报告
    -          get_url:          url=http://study.lab0.example.com/materials/hardware.empty
dest=/root/hardware.txt force=yes
    - replace: path=/root/hardware.txt regexp=inventoryhostname replace={{host}}
    - replace: path=/root/hardware.txt regexp=memory_in_MB replace={{mem}}
    - replace: path=/root/hardware.txt regexp=BIOS_version replace={{bios}}
    - replace: path=/root/hardware.txt regexp=disk_vda_size replace={{vdasize}}
    - replace: path=/root/hardware.txt regexp=disk_vdb_size replace={{vdbsize}}

[alice@control ansible]$ ansible-playbook hardware.yml          //验证剧本
```

13. 创建保险库文件

1）创建 ansible 保险库 /home/alice/ansible/passdb.yml，其中有 2 个变量：

```
pw_dev，值为 ab1234
pw_man，值为 cd5678
```

2）加密和解密该库的密码是 pwd@1234 ，密码存在/home/alice/ansible/secret.txt 中

解题参考：

```
[alice@control ansible]$ echo  'pwd@1234'  > secret.txt          //创建保险库钥匙文件
[alice@control ansible]$ ansible-vault  create  passdb.yml  --vault-password-file=secret.txt
pw_dev: ab1234                                                   //创建保险库文件
pw_man: cd5678
```

14. 编写剧本为受管机批量创建用户，要求使用保险库中的密码

从以下 URL 下载用户列表，保存到/home/alice/ansible 目录下：

http://study.lab0.example.com/materials/name_list.yml

创建剧本 /home/alice/ansible/users.yml 的 playbook，满足下列要求：

1）使用之前题目中的 passdb.yml 保险库文件

2）职位描述为 dev 的用户应在 test01、test02 主机组的受管机上创建，从 pw_dev 变量分配密码，是补充组 devops 的成员

3）职位描述为 man 的用户应在 web 主机组的受管机上创建，从 pw_man 变量分配密码，是补充组 opsmgr 的成员

4）该 playbook 可以使用之前题目创建的 secret.txt 密码文件运行

解题参考：

```
[alice@control ansible]$ wget http://study.lab0.example.com/materials/name_list.yml
                                                          //获取用户列表文件
[alice@control ansible]$ cat  name_list.yml               //确认列表内容
users:
  - name: tom
    job: dev                                              //用户岗位 a
  - name: jerry
    job: man                                              //用户岗位 b

[alice@control ansible]$ vim  users.yml
- name: batch users
  hosts: test01,test02,web
  vars_files:
    - passdb.yml                                          //加载密码变量
    - name_list.yml                                       //加载用户名变量
  tasks:
    - group: name=devops                                  //确保补充组 1 在指定主机已存在
    - group: name=opsmgr                                  //确保补充组 2 在指定主机已存在
    - user: name={{item.name}} password={{pw_dev|password_hash('sha512')}} groups=devops
      when: (item.job == 'dev') and ('test01' in group_names or 'test02' in group_names)
      loop: "{{users}}"                                   //按条件添加 a 岗用户
    - user: name={{item.name}} password={{pw_man|password_hash('sha512')}} groups=opsmgr
      when: (item.job == 'man') and ('web' in group_names)
```

```
      loop: "{{users}}"                                    //按条件添加 b 岗用户

[alice@control      ansible]$      ansible-playbook      users.yml      --vault-password-file
=/home/alice/ansible/secret.txt                            //验证剧本
```

15. 重设保险库密码

1）从以下 URL 下载保险库文件到/home/alice/ansible 目录：
http://study.lab0.example.com/materials/topsec.yml

2）当前的库密码是 banana，新密码是 big_banana，请更新该库密码

解题参考：

```
[alice@control ansible]$ wget   http://study.lab0.example.com/materials/topsec.yml
                                                           //下载指定保险库文件
[alice@control ansible]$ ansible-vault  rekey  topsec.yml       //为保险库设置新的密码
Vault password: 输入当前的库密码
New Vault password: 输入新的库密码
confirm New Vault password: 再次输入新的库密码确认
Rekey successful
```