D0407 EXAM, Ansible version 2.7.x

EXAM Environment	LAB Environment
control.labx.example.com	workstation
node1.labx.example.com	servera
node2.labx.example.com	serverb
node3.labx.example.com	serverc
node4.labx.example.com	serverd
node5.labx.example.com	serverb
/home/matthew	/home/student

v1.4

更新记录：
20190806
1．第4题，第7题，第8题，第9题更新了考题变动
2．第12题增加了解法2供参考

1. Install and configure ansible

install and configure ansible on the control node **control.labx.example.com** as follows:

1. install the required packages

2. create a static inventory file called /home/matthew/ansible/inventory as follows:

2.1 **node1** is a member of the **dev** host group

2.2 **node2** is a member of the **test** host group

2.3 **node3** and **node4** are members of the **prod** host group

2.4 **node5** is a member of the **balancers** host group

2.5 the **prod** group is a member of the **webservers** host group

3. create a configuration file called **/home/matthew/ansible/ansible.cfg** as follows:

the host inventory file **/home/matthew/ansible/inventory** is defined

the location of roles used in playbooks is defined as **/home/matthew/ansible/roles**

```
At workstation login as student
$ mkdir -p /home/student/ansible/roles
$ su - student
$ sudo yum -y install ansible
$ ansible -version
$ cd /home/student/ansible
$ vi inventory
[dev]
servera
[test]
serverb
[prod]
serverc
serverd
[balancers]
serverb
[webservers:children]
prod
$ less /etc/ansible/ansible.cfg  #find defaults and escalation parts#
$ vi /home/student/ansible/ansible.cfg
[defaults]
remote_user = devops
ask_pass = false
inventory = /home/student/ansible/inventory
roles_path = /home/student/ansible/roles
[privilege_escalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=false
#考试环境中remote_user = matthew
$ ansible all -m ping
```

2. Create and run an ansible ad-hoc command

As a system administrator, you will need to install software on the managed nodes.

Create a shell script called /home/matthew/ansible/adhoc.sh that runs an ansible ad-hoc command to create a yum repository on each of the managed nodes as follows:

1. The name of repository is **Exam_RHEL**
2. The description is **EX407 software**
3. the base URL is http://rhgls.labx.example.com/rhel

4. GPG signature checking is enabled

5. The GPG key URL is http://rhgls.lab.example.com/rhel/RPM-GPG-KEY-redhat-release

6. The repository is enabled

```
$ vi adhoc.sh
#!/bin/bash

ansible all -m yum_repository -a 'name=Exam_RHEL description="EX407 software"
baseurl=http://content.example.com/rhel7.6/x86_64/dvd/ gpgcheck=yes
gpgkey=http://content.example.com/rhel7.6/x86_64/dvd/RPM-GPG-KEY-redhat-release
enabled=yes'
$ sudo chmod 0755 adhoc.sh
$ ./adhoc.sh
$ ansible all -m shell -a 'cat /etc/yum.repos.d/Exam_RHEL.repo'
```

3. Install packages

Create a playbook called /home/matthew/ansible/packages.yml that:

1. Install the **php** and **mariadb** packages on hosts in the **dev**, **test**, and **prod** host groups
2. Installs the **Development Tools** package group on hosts in the **dev** host group
3. Updates all packages to the latest version on hosts in the dev host group

```
$vi packages.yml
```

Method 1:

```
- hosts: dev,test,prod
  vars:
    pkgs:
      - php
      - mariadb
  tasks:
    - name: Installs the php and mariadb packages on hosts
      yum:
        name: "{{ pkgs }}"
        state: latest
    - name: installs the Development Tools package group on hosts
      yum:
        name: "@Development Tools"
        state: latest
      when: ansible_hostname in groups.dev
    - name: updates all packages to the latest version on hosts
      yum:
        name: '*'
        state: latest
        update_only: true
      when: ansible_hostname in groups.dev
```

Method 2 (better):

```
---
```

```
- hosts: dev,test,prod
  tasks:
  - name: install php mariadb
    yum:
      name: "{{ item }}"
      state: present
    loop:
    - php
    - mariadb
  - name: install group Dev
    yum:
      name: "@Development Tools"
      state: present
    when: ansible_hostname in groups["dev"]
  - name: update
    yum:
      name: "*"
      state: latest
    when: ansible_hostname in groups["dev"]
```

```
$ ansible-playbook --syntax-check packages.yml
$ ansible-playbook packages.yml
$ ansible dev,test,prod -m shell -a 'rpm -qa | grep php'
$ ansible dev,test,prod -m shell -a 'rpm -qa | grep mariadb'
```

4. Use a RHEL system role

Install the RHEL system roles package and create a playbook called /home/matthew/ansible/timesync.yml that:

1. Runs on all managed hosts
2. Uses the **timesync** role
3. Configures the role to use the time server **172.24.1.254** (in our lab, it's **172.25.254.254**)
4. Configures the role to set the **iburst** parameter as enabled

```
$ sudo yum -y install rhel-system-roles
$ cp -rf /usr/share/ansible/roles/linux-system-roles.timesync/ /home/student/ansible/roles/
$ cd /home/student/ansible
$ vi timesync.yml
```

```
---
- name: Configure time synchronization with NTP servers
  hosts: all
  become: true
  vars:
    timesync_ntp_servers:
      - hostname: 172.25.254.254
        iburst: yes
```

```
  roles:
    - role: linux-system-roles.timesync
```

```
$ ansible-playbook --syntax-check timesync.yml
$ ansible-playbook timesync.yml
$ ansible all -m shell -a 'chronyc sources'
```

更新点：

如果看到题目脚本执行更改的配置为/etc/ntp.conf

则执行检查命令为：

```
$ ansible all -m shell -a 'ntpq -q'
```

```
$ ansible all -m shell -a 'cat /etc/ntp.conf'
```

5. Install roles using Ansible Galaxy

Use Ansible Galaxy with a requirements file called /home/matthew/ansible/roles/requirements.yml to download and install roles to /home/matthew/ansible/roles from the following

1. http://rhgls.labx.example.com/materials/haproxy.tar

   The name of this role should be **balancer**

2. http://rhgls.labx.example.com/materials.phpinfo.tar

   The name of this role should be **phpinfo**

```
$vi /home/student/ansible/roles/requirements.yml
```

```
- src: http://materials.example.com/labs/role-system/roles/haproxy.tar.gz
  name: balancer

- src: http://materials.example.com/labs/role-system/roles/phpinfo.tar.gz
  name: phpinfo
```

```
##实验中将这两个tar.gz上传到foundation的
/content/courses/do407/ansible2.7/materials/labs/role-system/role/
然后即可完成该题.
$ansible-galaxy install -r /home/student/ansible/roles/requirements.yml  -p
/home/student/ansible/roles/
```


haproxy.tar.gz


phpinfo.tar.gz

6. Create and use a role

Create a role called apache in /home/matthew/ansible/role with the following requirements:

1. The httpd package is installed, enabled on boot, and started
2. the firewall is enabled and running with a rule to allow access to the web server
3. A template file index.html.j2 exists and is used to create the file /var/www/html/index.html with the following output:

   Welcome to **HOSTNAME** on **IPADDRESS**

   where **HOSTNAME** is the fully qualified domain name of the managed node and **IPADDRESS** is the IP address of the managed node.

Create a playbook called /home/matthew/ansible/newrole.yml that uses this role as follows

4. The playbook runs on hosts in the webservers host group

```
$ ansible-galaxy init apache --init-path /home/student/ansible/roles
$ cd /home/student/ansible/roles/apache

$vi tasks/main.yml
---
- name: Install httpd
  yum:
    name: httpd
    state: present

- name: Start httpd
  service:
    name: httpd
    state: started
    enabled: yes

- name: start firewalld
  service:
    name: firewalld
    state: started
    enabled: yes

- name: filrewall permits http service
  firewalld:
    service: http
    state: enabled
    permanent: true
    immediate: yes

- name: create /var/www/html/index.html
  template:
    src: index.html.j2
    dest: /var/www/html/index.html
    setype: httpd_sys_content_t
```

```
$vi templates/index.html.j2

Welcome to {{ ansible_fqdn }} on {{ ansible_default_ipv4.address }}

vi /home/matthew/ansible/newrole.yml
---
- hosts: webservers
  roles:
    - apache
$ ansible-playbook --syntax-check newrole.yml
$ ansible-playbook newrole.yml
$ curl http://serverc
Welcome to serverc.lab.example.com on 172.25.250.12
$ curl http://serverd
Welcome to serverd.lab.example.com on 172.25.250.13
```

7. Use roles from Ansible Galaxy

Create a playbook called /home/matthew/ansible/roles.yml as follows:

1. the playbook contains a play that runs on hosts in the **balancers** host group and uses the **balancer** role.

   1.1 This role configures a service to load balance web server requests between hosts in the **webservers** host group.

   1.2 When implemented, browsing to hosts in the **balancers** host group (for example http://node5.labx.example.com) should produce the following output:

   Welcome to node3.labx.example.com on 172.24.1.8

   Reloading the browser should return output from the alternate web server:

   Welcome to node4.labx.example.com on 172.24.1.9

2. The playbook contains a play that runs on hosts in the webservers host group and uses the phpinfo role.

   2.1 When implemented, browsing to hosts in the webservers host group with the url /hello.php should produce the following output:

   Hello PHP World from FQDN

   where FQDN is the fully qualified domain name of the host.

   2.2 For example, browsing to http://node3.labx.example.com/hello.php, should produce the following output

   Hello PHP World from node3.labx.example.com

   along with various details of the PHP configuration including the version PHP that is installed.

2.3 Similarly, browsing to http://node4.labx.example.com/hello.php. should produce the following output:

Hello PHP World from node4.labx.example.com

along with various details of the PHP configuration including the version PHP that is installed.

```
$ vi /home/student/ansible/roles.yml
- hosts: balancers,webservers
  roles:
  - { role: balancer ,when: "ansible_hostname in groups['balancers']" }

- hosts: webservers
  roles:
  - phpinfo
$ ansible-playbook --syntax-check roles.yml
$ ansible-playbook roles.yml
$ curl http://serverb
Welcome to serverc.lab.example.com on 172.25.250.12
$ curl http://serverb
Welcome to serverd.lab.example.com on 172.25.250.13
$ curl http://serverb/hello.php
Hello PHP World form serverc.lab.example.com
$ curl http://serverb/hello.php
Hello PHP World form serverd.lab.example.com
```

更新：

该题部署完后可能会报错，或者效果没有出来， 需要具体检查 balancer 和 phpinfo 下的 j2 模板及 main.yml 文件，相关参数需要手动修改。

如不知道如何修改，可放弃，预计扣 3~5 分。

8. Create and use a partition

Create a playbook called /home/matthew/ansible/partition.yml that runs on all managed nodes that does the following:

1. Creates a single primary partition number 1 of size **1500 MiB** on device **vdb**
2. Formats the partition with the **ext4** filesystem
3. Mounts the filesystem persistently at **/newpart**
4. If the requested partition size cannot be created, the error message

   Could not create partition of that size

   should be displayed and the size **800 MiB** should be used instead

5. If the device vdb does not exist, the error message

   Disk does not exist

   should be displayed.

```
$ vi /home/student/ansible/partition.yml
```

```
- hosts: all
  tasks:
  - name: check vdb
    shell: ls -l /dev/vdb
    register: disk
    ignore_errors: yes
  - name: debug
    debug:
      msg: "Disk does not exist"
    failed_when: disk.rc != 0
  - block:
    - name: create a partion
      parted:
        device: /dev/vdb
        number: 1
        part_end: 1500MiB
        state: present
    rescue:
    - name: debug error information
      debug:
        msg: "Could not create partition of that size"
    - name: create a new size partion
      parted:
        device: /dev/vdb
        number: 1
        part_end: 800MiB
        state: present
  - name: create filesystem
    filesystem:
      fstype: ext4
      dev: /dev/vdb1
  - name: mount
    mount:
      path: /newpart
      src: /dev/vdb1
      fstype: ext4
      state: mounted
```

```
$ ansible-playbook --syntax-check partition.yml
$ ansible-playbook partition.yml
$ ansible all -m shell -a 'lsblk'
servera | CHANGED | rc=0 >>
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    252:0    0   40G  0 disk
└─vda1 252:1    0   40G  0 part /
vdb    252:16   0    1G  0 disk
└─vdb1 252:17   0  799M  0 part /newpart
```

```
serverb | CHANGED | rc=0 >>
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    252:0    0   40G  0 disk
└─vda1 252:1    0   40G  0 part /
vdb    252:16   0    1G  0 disk
└─vdb1 252:17   0  799M  0 part /newpart

serverc | CHANGED | rc=0 >>
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    252:0    0   40G  0 disk
└─vda1 252:1    0   40G  0 part /
vdb    252:16   0    1G  0 disk
└─vdb1 252:17   0  799M  0 part /newpart

serverd | CHANGED | rc=0 >>
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    252:0    0   40G  0 disk
└─vda1 252:1    0   40G  0 part /
vdb    252:16   0    1G  0 disk
└─vdb1 252:17   0  799M  0 part /newpart
```

8.1 Create and use a LVM

create a playbook called /home/Curtis/ansible/lv.yml that runs on all managed nodes that does the following:

1. creates a single logical volume of size **1500MiB** in volume group **research**

2. formats the logical volume with the **ext4** filesystem

3. mount the filesystem **persistently** at **/data** on hosts in the **qa** host **group ONLY**

4. if the request lv size cannot be created, the error message:

   Could not create logical volume of that size

   should be displayed and the size 800 MiB should be used instead

5. if the volume group research does not exist, the error message

   Volume group does not exist

   should be displayed

该题替换了原来的创建分区，需要大家注意参数变动。

测试机上操作(serverX):

```
# fdisk /dev/vdb
```

分区 900M

类型 8e

```
# partprobe
# pvcreate /dev/vdb1
# pvdisplay
# vgcreate research /dev/vdb1
# vgdisplay |grep research
```

$ vi /home/student/ansible/lv.yml

```
- hosts: all
  tasks:
    - name: check vg
      shell: vgdisplay | grep research
      register: check_result
      ignore_errors: yes
    - name: debug
      debug:
        msg: "Volume group does not exist"
      Faild_when: check_result.failed
    - block:
      - name:  lvcreate
        lvol:
          vg: research
          lv:  lvx
          size: 1500m
      rescue:
        - name: debug
          debug:
            msg: "Could not create logical volume of that size"
        - name: lvcreate
          lvol:
            vg: research
            lv:  lvx
            size: 800m
    - name: filesystem
      filesystem:
        fstype: ext4
        dev: /dev/research/lvx
    - name: directory
      file:
         path: /data
         State: directory
    - name: mount
      mount:
        fstype: ext4
        path: /data
        src: /dev/research/lvx
        state: mounted
      when: ansible_name in groups [‘qa’]
```

验证:

```
[student@workstation ansible]$ ansible all -a 'lsblk'
serverc | CHANGED | rc=0 >>
NAME             MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda              252:0    0   40G  0 disk
└─vda1           252:1    0   40G  0 part /
vdb              252:16   0    1G  0 disk
└─vdb1           252:17   0 1023M  0 part
  └─research-lvx 253:0    0  800M  0 lvm  /data
```

```
[student@workstation ansible]$ ansible all -m shell -a 'cat /etc/fstab'
serverc | CHANGED | rc=0 >>
#
# /etc/fstab
# Created by anaconda on Wed Oct 10 18:12:32 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
```

```
UUID=6c248666-70f5-4037-8b24-17100c2f5c1e /                       xfs     defaults        0 0
/dev/research/lvx /data ext4 defaults 0 0
```

VG 也有可能需要自己创建

lvg，lvol 两个模块使用参照如下案例

```
- hosts: test
  vars:
    vg_name: "vgtest"
    lv_name: "lvtest"
    pvs_name: "/dev/sdb"
    dir_path: "/data"
  tasks:
    - name: 创建一个 vg
      lvg:
        vg: "{{vg_name}}"
        pvs: "{{pvs_name}}"
        pesize: 4
    - name: 创建一个 lv
      lvol:
        vg: "{{vg_name}}"
        lv: "{{lv_name}}"
        size: 100%PVS
    - name: 格式化 lv
      filesystem:
        fstype: ext4
        dev: "/dev/{{vg_name}}/{{lv_name}}"
    - name: 获取 UUID
      shell: "blkid /dev/{{vg_name}}/{{lv_name}} |awk '{print $2}'"
      register: result
      ignore_errors: True
    - name: 创建挂载目录
      file:
        path: "{{dir_path}}"
        state: directory
        mode: 0755
    - name: 使用 UUID 挂载 lvm 分区
      mount:
         path: /data
         src:  "{{result.stdout}}"
         fstype: ext4
         state: mounted
```

9. Generate a hosts file

1. Download an initial template file call hosts.j2 from http://rhgls.labx.example.com/materials to /home/matthew/ansible
2. Complete the template so that it can be used to generate a file with a line for each inventory host in the same format as /etc/hosts
3. Create a playbook called /home/matthew/ansible/hosts.yml that uses this template to generate the file /etc/myhosts on hosts in the dev host group.

When completed, the file /etc/myhosts on hosts in the dev host group shoud have a line for each managed host:

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

```
172.24.1.6 node1.lab.example.com server1
172.24.1.7 node2.lab.example.com server2
172.24.1.8 node3.lab.example.com server3
172.24.1.9 node4.lab.example.com server4
172.24.1.10 node5.lab.example.com server5
```

```
$ cd /home/student/ansible/
$ wget http://rhgls.labx.example.com/materials/hosts.j2
```

##实验环境自建一个hosts.j2

更新：这一段要自己背下来，考试这个j2文件变成了空的

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

{% for host in groups['all'] %}
{{ hostvars[host]['ansible_default_ipv4']['address'] }}
{{ hostvars[host]['ansible_fqdn'] }} {{ hostvars[host]['ansible_hostname'] }}
{% endfor %}
```

```
$ vi hosts.yml
---
- hosts: all
  - name: copy j2
    template:
      src: hosts.j2
      dest: /etc/myhosts
    when: ansible_hostname in groups[‘dev’]
$ ansible-playbook --syntax-check hosts.yml
$ ansible-playbook  hosts.yml
$ ansible all -a 'cat /etc/myhosts'
```

10. Modify file content

Create a playbook called /home/matthew/ansible/issure.yml as follows:

1. The playbook runs on all inventory hosts
2. The playbook replaces the contents of /etc/issue with a single line of text as follows:
   - 2.1 On hosts in the dev host group, the line reads: Development
   - 2.2 On hosts in the test host group, the line reads: Test
   - 2.3 On hosts in the prod host group, the line reads: Production

```
$ vi /home/student/ansible/issue.yml
---
- hosts: all
  tasks:
  - name: replace content1
    copy:
```

```
      content: "Development"
      dest: /etc/issue
    when: ansible_hostname in groups["dev"]
  - name: replace content2
    copy:
      content: "Test"
      dest: /etc/issue
    when: ansible_hostname in groups["test"]
  - name: replace content3
    copy:
      content: "Production"
      dest: /etc/issue
    when: ansible_hostname in groups["prod"]
```

```
$ ansible-playbook --syntax-check issue.yml
$ ansible-playbook  issue.yml
$ ansible all -a 'cat /etc/issue'
```

11. Create a web content directory

Create a playbook called /home/matthew/ansible/webcontent.yml as follows:

1. The playbook runs on managed nodes in the dev host group
2. Create the directory /webdev with the following requirements:
   2.1 membership in the webdev group
   2.2 regular permissions:owner=read+write+excute,group=read+write+excute,other=read+excute
   2.3 special permissions: set group ID
3. Symbolically link /var/www/html/webdev to /webdev
4. Create the file /webdev/index.html with a single line of text that reads: Development

```
$ vi /home/student/ansible/webcontent.yml
```

```
- hosts: dev
  become: true
  tasks:
    - name: create a group
      group:
        name: webdev
        gid: 1111
        state: present

    - name: create a directory
      file:
        path: /webdev
        state: directory
        mode: 2775
        state: directory

    - name: create a link
```

```
      file:
        src: /webdev
        dest: /var/www/html/webdev
        state: link

    - name: copy content
      copy:
        content: "Development\n"
        dest: /webdev/index.html
```

```
$ ansible-playbook --syntax-check webcontent.yml
$ ansible-playbook  webcontent.yml
$ curl http://servera/webdev/index.html
```

12. Generate a hardware report

Create a playbook called /home/matthew/ansible/hwreport.yml that produces an output file called /root/hwreport.txt on all managed node with the following informations:

1. Inventory host name
2. Total memory in MB
3. BIOS version
4. Size of disk device vda
5. Size of disk device vdb
6. Each line of the output file contains a single key=value pair

Your playbook should:

1. Download the file hwreport.empty from the url http://rhgls.labx.example.com/materials and save it as /root/hwreport.txt
2. Modify /root/hwreport.txt with the correct values
3. If a hardware item does not exist, the associated value should be set to NONE

##create hwreport.txt in lab environment, LAB 环境中缺少该文件##

```
$ vi hwreport.txt
Template for exam407
scp hwreport.txt root@foundation0:/content/courses/do407/ansible2.7/materials/labs/
```

```
$ vi /home/student/ansible/hwreport.yml
```

```
- hosts: all
  tasks:
    - name: Download the file hwreport.empty
      get_url:
        url: http://materials.example.com/labs/hwreport.txt
        dest: /root/hwreport.txt

    - name:  set inventory_hostname
      lineinfile:
        path: /root/hwreport.txt
        line: "inventory_hostname = {{ inventory_hostname |
default('NONE') }}"

    - name: set total memory
      lineinfile:
        path: /root/hwreport.txt
        line: "Total_Mem = {{ ansible_memtotal_mb |default('NONE')}}"

    - name: print bios version
      lineinfile:
        path: /root/hwreport.txt
        line: "BIOS_ver = {{ ansible_bios_version | default('NONE')}}"

    - name: print vda size
      lineinfile:
        path: /root/hwreport.txt
        line: "vda_size = {{ ansible_devices.vda.size | default('NONE')}}"
    - name: print vdb size
      lineinfile:
        path: /root/hwreport.txt
        line: "vdb_size = {{ ansible_devices.vdb.size | default('NONE')}}"
```

```
$ ansible-playbook --syntax-check hwreport.yml
$ ansible-playbook  hwreport.yml
$ ansible all -a 'cat /root/hwreport.txt'
```

13. Create a password vault

Create an Ansible vault to store user passwords as follows

1. The name of the vault is /home/matthew/ansible/locker.yml
2. The vault contains two variables as follows:

   2.1 **pw_developer** with value **Imadev**

   2.2 **pw_manager** with value **Imamgr**

3. The password to encrypt and decrypt the vault is **whenyouwishuponastar**
4. The password is stored in the file /home/matthew/ansible/secret.txt

```
$ vi /home/student/ansible/secret.txt
whenyouwishuponastar

$ ansible-vault --vault-password-file=secret.txt create
/home/student/ansible/locker.yml
pw_developer: Imadev
pw_manager: Imamgr

$ ansible-vault view locker.yml --vault-password-file=/home/student/ansible/secret.txt
```

14. Create user accounts

1. A list of user be created can be found in the file called user_list.yml which you should download form http://rhgls.labx.example.com/materials/ and save to /home/matthew/ansible
2. Using the password vault /home/matthew/ansible/locker.yml created elsewhere in this exam， create a playbook called /home/matthew/ansible/users.yml that create user accounts as follows:

   2.1 User with a job description of **developer** should be:

   2.1.1 Create on managed nodes in the **dev** and **test** host groups

   2.1.2 Assigned the password form the **pw_developer** variable

   2.1.3 A member of supplementary group **devops**

   2.2 User with a job description of manager should be:

   2.2.1 Create on managed nodes in the **prod** host group

   2.2.2 Assigned the password from the **pw_manager** variable

   2.2.3 A member of supplementary group **opsmgr**
3. Password should use the SHA512 hash format.
4. Your playbook should work using the vault password file created elsewhere in this exam.

##create user_list.yml in lab environment, LAB 环境中缺少该文件##

```
$ vi user_list.yml
---
users:
  - name: node1
    job: developer
```

```
  - name: node2
    job: developer
  - name: node3
    job: manager
```

```
$ cd /home/student/ansible/
$ wget http://rhgls.labx.example.com/materials/user_list.yml
$ cat user_list.yml
```

```
$ vi users.yml
```

Method 1:

```
---
- hosts: dev,test,prod
  vars_files:
  - user_list.yml
  - locker.yml
  tasks:
  - name: create group for dev and test
    group:
      name: devops
      state: present
    when: ansible_hostname in groups["dev"] or ansible_hostname in groups["test"]
  - name: create group for prod
    group:
      name: opsmgr
      state: present
    when: ansible_hostname in groups["prod"]
  - name: create user for dev and test
    user:
      name: "{{ item.name }}"
      groups: devops
      password: "{{ pw_developer | password_hash('sha512') }}"
      comment: "{{ item.job }}"
    loop: "{{ users }}"
    when: ansible_hostname in groups["dev"] or ansible_hostname in groups["test"]
  - name: create user for prod
    user:
      name: "{{ item.name }}"
      groups: opsmgr
      password: "{{ pw_manager | password_hash('sha512') }}"
      comment: "{{ item.job }}"
    loop: "{{ users }}"
    when: ansible_hostname in groups["prod"]
```

```
$ ansible-playbook --syntax-check users.yml --vault-password-
file=/home/student/ansible/secret.txt
$ ansible-playbook users.yml --vault-password-file=/home/student/ansible/secret.txt
```

```
$ ansible all -a 'id nodex'
```

Method 2:

```
---
- hosts: all
  vars_files:
    - user_list.yml
    - locker.yml

  tasks:

    - name: group devops
      group:
        name: devops
        state: present
      when: inventory_hostname in groups['dev'] or inventory_hostname in
groups['test']

    - name: group opsmgr
      group:
        name: opsmgr
        state: present
      when: inventory_hostname in groups['prod']

    - name: user for developer
      user:
        name: "{{ item.name }}"
        state: present
        groups: devops
        password: "{{ pw_developer | password_hash('sha512') }}"
      loop: "{{ users }}"
      when: (inventory_hostname in groups['dev'] or inventory_hostname in
groups['test']) and item.job=="developers"

    - name:
      user:
        name: "{{ item.name }}"
        state: present
        groups: opsmgr
        password: "{{ pw_manager | password_hash('sha512') }}"
      loop: "{{ users }}"
      when: inventory_hostname in groups['prod'] and item.job=="manager"
```

15. Rekey and Ansible vault

Rekey an existing ansible vault as follows:

1. Download the ansible vault from http://rhgls.labx.example.com/materials/salaries.yml and save it as /home/matthew/ansible/salaries.yml；

2. The current vault password is **insecure4sure**;

3. The new vault password is **bbe2de98389b**;

4. The vault remains in an encrypted state with the new password

考试中注意 `salaries.yml` 的文件权限，可能需要修改才能操作

##create `salaries.yml` in lab environment, LAB 环境中缺少该文件##

```
$ ansible-vault create salaries.yml
New Vault password: insecure4sure
Confirm New Vault password: insecure4sure
RED HAT ANSIBLE 2.7 EXAM
GOOD LUCK
```

```
$ cd /home/student/ansible/
$ wget http://workstation.lab.example.com/salaries.yml -o /home/student/ansible/salaries.yml
```

```
$ ansible-vault rekey salaries.yml
Vault password: insecure4sure
New Vault password: bbe2de98389b
Confirm New Vault password: bbe2de98389b
$ ansible-vault view salaries.yml
Vault password: bbe2de98389b
```