

RH294解法第1题-安装和配置Ansible





安装和配置Ansible

照下方所述，在控制节点 `control.domainx.example.com` 上安装和配置 Ansible：

- ☐ 1. 安装所需的软件包
- ☐ 2. 创建名为 `/home/greg/ansible/inventory` 的静态清单文件，以满足以下要求：
 - `node1` 是 `dev` 主机组的成员
 - `node2` 是 `test` 主机组的成员
 - `node3` 和 `node4` 是 `prod` 主机组的成员
 - `node5` 是 `balancers` 主机组的成员
 - `prod` 组是 `webserver`s 主机组的成员
- ☐ 3. 创建名为 `/home/greg/ansible/ansible.cfg` 的配置文件，以满足以下要求：
 - 清单文件为 `/home/greg/ansible/inventory`
 - `playbook` 中使用的角色的位置包括 `/home/greg/ansible/roles`

```
1 #题目要求我们要在控制节点上安装Ansible
2 #是否有软件仓库，如果控制节点没有配置软件仓库，你要看其他配置信息，或者题干里面是否给你提供了软件仓库的url。
3
4 #连接到控制节点检查软件仓库
5 [greg@control ~]$ yum repolist
6 Red Hat Enterprise Linux 8.0 AppStream (dvd)      122 kB/s | 3.2 kB      00:00
7
8 Red Hat Enterprise Linux 8.0 BaseOS (dvd)         82 kB/s | 2.7 kB      00:00
9
10 repo id                                repo name
11 status
12 rhel-8.0-for-x86_64-appstream-rpms Red Hat Enterprise Linux 8.0 AppStream
13 4,672
14 rhel-8.0-for-x86_64-baseos-rpms      Red Hat Enterprise Linux 8.0 BaseOS (d
15 1,658
16
17 #检查软件仓库中是否有ansible的软件包
18 [greg@control ~]$ yum list | grep -i ansible
```

Index of /rhel8.0/x86_64/ucfupdates

Name	Last modified	Size	Description
 Parent Directory		-	
 Packages/	2019-08-28 23:40	-	
 TRANS.TBL	2019-08-28 23:40	440	
 repodata/	2019-08-28 23:40	-	

```
1 #通过浏览器访问http://content.example.com/rhel8.0/x86_64/ucfupdates/
2 将该链接作为软件仓库的baseurl
3 [greg@control ~]$ sudo yum-config-manager --add-
4 repo=http://content.example.com/rhel8.0/x86_64/ucfupdates/
5 Adding repo from: http://content.example.com/rhel8.0/x86_64/ucfupdates/
6
7 [greg@control yum.repos.d]$ sudo vim
8 content.example.com_rhel8.0_x86_64_ucfupdates_.repo
9 [greg@control yum.repos.d]$ cat
10 content.example.com_rhel8.0_x86_64_ucfupdates_.repo
11 [content.example.com_rhel8.0_x86_64_ucfupdates_]
12 name=created by dnf config-manager from
13 http://content.example.com/rhel8.0/x86_64/ucfupdates/
14 baseurl=http://content.example.com/rhel8.0/x86_64/ucfupdates/
15 enabled=1
16 gpgcheck=0
17
18 [greg@control yum.repos.d]$ yum repolist
19 created by dnf config-manager from http://conte 628 kB/s | 2.9 kB 00:00
20
21 repo id                                repo name
22 status
23 content.example.com_rhel8.0_x86_64_ucfupdates_ created by dnf config-mana
24 8
25 rhel-8.0-for-x86_64-appstream-rpms      Red Hat Enterprise Linux 8
26 4,672
27 rhel-8.0-for-x86_64-baseos-rpms        Red Hat Enterprise Linux 8
28 1,658
29
30
31
32
```

安装ansible

```
1 [greg@control ~]$ sudo yum -y install ansible
```

写inventory文件(普通用户作为remote_user)

```
1 [greg@control ~]$ mkdir ansible
2 [greg@control ~]$ ls
3 ansible
4 [greg@control ~]$ cd ansible/
5
6 [greg@control ansible]$ cat inventory
7 node1
8 node2
9 node3
10 node4
11 node5
12 [dev]
13 node1
14
15 [test]
16 node2
17
18 [prod]
19 node3
20 node4
21
22 [balancers]
23 node5
24
25 [webservers:children]
26 prod
27
28
29
30
```

```
1 #配置文件(普通用户作为remote_user)
2 [greg@control ansible]$ cat ansible.cfg
3 [defaults]
4 inventory = /home/greg/ansible/inventory
5 remote_user = greg
6 ask_pass = false
7 roles_path = /home/greg/ansible/roles
8
9
10 [privilege_escalation]
11 become = yes
12 become_method = sudo
13 become_user = root
14 become_ask_pass = false
15
```

remote_user是root用户

```
1 [greg@control ansible-root]$ cat ansible.cfg
2 [defaults]
3 inventory = /home/greg/ansible/inventory
4 remote_user = root
5 roles_path = /home/greg/ansible/roles
6
```

```
7 [greg@control ansible-root]$ cat inventory
8 node1
9 node2
10 node3
11 node4
12 node5
13 [dev]
14 node1
15
16 [test]
17 node2
18
19 [prod]
20 node3
21 node4
22
23 [balancers]
24 node5
25
26 [webserver:children]
27 prod
28
29 [all:vars]
30 ansible_ssh_pass = redhat
31 [greg@control ansible-root]$ cat group_vars/all
32 ansible_ssh_pass: redhat
```

RH294解法第二题-创建和运行ansible的临时命令

创建和运行Ansible临时命令

创建一个名为 /home/greg/ansible/adhoc.sh 的 shell 脚本，该脚本将使用 Ansible 临时命令在各个受管节点上安装 yum 存储库：

- ☐ 存储库1：
 - 存储库的名称为 EX294_BASE
 - 述为 EX294 base software
 - 基础 URL 为 http://repo.domainx.example.com/BaseOS
 - GPG 签名检查为启用状态
 - GPG 密钥 URL 为 http://repo.domainx.example.com/RPM-GPG-KEY-redhat-release
 - 存储库为启用状态
- ☐ 存储库2：
 - 存储库的名称为 EX294_STREAM
 - 描述为 EX294 stream software
 - 基础 URL 为 http://rhgls.domainx.example.com/AppStream
 - GPG 签名检查为启用状态
 - GPG 密钥 URL 为 http://rhgls.domainx.example.com/RPM-GPG-KEY-redhat-release
 - 储库为启用状态

```
1 [greg@control ansible]$ vim adhoc.sh
2 [greg@control ansible]$ cat adhoc.sh
3 #!/bin/bash
4 ansible all -m yum_repository -a 'name="EX294_BASE" description="EX294 base
5 software" baseurl="http://repo.domainx.example.com/BaseOS" gpgcheck=yes
6 gpgkey="http://repo.domainx.example.com/RPM-GPG-KEY-redhat-release"'
7
8
9 ansible all -m yum_repository -a 'name="EX294_STREAM" description="EX294
10 stream software" baseurl="http://repo.domainx.example.com/AppStream"
11 gpgcheck=yes gpgkey="http://repo.domainx.example.com/RPM-GPG-KEY-redhat-
12 release"'
13
14
15 [greg@control ansible]$ chmod +x adhoc.sh
16 [greg@control ansible]$ ls -l
17 total 12
18 -rwxrwxr-x. 1 greg greg 453 May  5 23:41 adhoc.sh
19 -rw-rw-r--. 1 greg greg 220 May  5 23:19 ansible.cfg
20 -rw-rw-r--. 1 greg greg 124 May  5 23:10 inventory
```

检查现象

```
1 1. 运行脚本之前查看一下节点的软件仓库信息
2 [greg@control ansible]$ ansible all -a 'yum repolist'
3
4
5 2. 运行脚本
6 [greg@control ansible]$ ./adhoc.sh
7
8 3. 运行脚本之后查看一下节点的软件仓库信息
9 [greg@control ansible]$ ansible all -a 'yum repolist'
```


3-RH294解法第三题-安装软件包

创建一个名为 `/home/greg/ansible/packages.yml` 的 playbook :

- ☐ 1. 将 `php` 和 `mariadb` 软件包安装到 `dev`、`test` 和 `prod` 主机组中的主机上
- ☐ 2. `RPM Development Tools` 软件包组安装到 `dev` 主机组中的主机上
- ☐ 3. `dev` 主机组中主机上的所有软件包更新为最新版本

```
1  #第三题要求创建一个名字为packages.yml的playbook
2  [greg@control ansible]$ cat packages.yml
3  ---
4  - name: install pkg
5    hosts: dev,test,prod
6    tasks:
7      - name: use yum module to install pkg
8        yum:
9          name:
10            - php
11            - mariadb
12          state: latest
13
14  - name: install pkg
15    hosts: dev
16    tasks:
17      - name: use yum module
18        yum:
19          name: "@RPM Development Tools"
20          state: latest
21      - name: use yum module
22        yum:
23          name: "*"
24          state: latest
25
```

```
1  [greg@control ansible]$ cat packages-second.yml
2  ---
3  - name: install pkg
4    hosts: dev,test,prod
5    tasks:
6      - name: use yum module to install pkg
7        yum:
8          name:
9            - php
10            - mariadb
11          state: latest
12      - name: use yum module
13        yum:
14          name: "@RPM Development Tools"
15          state: latest
16          when: "'dev' in group_names"
```

```
17 - name: use yum module
18   yum:
19     name: "*"
20     state: latest
21     when: "'dev' in group_names"
```


4-RH294解法第四题-使用系统role

使用RHEL系统角色

装 RHEL 系统角色软件包，并创建符合以下条件的 playbook /home/greg/ansible/timesync.yml:

- ☐ 1. 在所有受管节点上运行
- ☐ 2. 用 timesync 角色
- ☐ 3. 置该角色，以使用当前有效的 NTP 提供商
- ☐ 4. 置该角色，以使用时间服务器 172.25.254.254
- ☐ 5. 配置该角色，以后用 iburst 参数

```
1 #如果想使用rhel系统的role, 需要安装软件包
2 [greg@control ansible]$ yum list | grep role
3 [greg@control ansible]$ sudo yum -y install rhel-system-roles
4
5 #修改配置文件, 让ansible配置文件加载rhel system role
6 [greg@control ansible]$ cat ansible.cfg
7 [defaults]
8 inventory = /home/greg/ansible/inventory
9 roles_path = /home/greg/ansible/roles:/usr/share/ansible/roles
10 remote_user = greg
11 ask_pass = false
12
13 [privilege_escalation]
14 become=true
15 become_method=sudo
16 become_user=root
17 become_ask_pass=false
18
19 [greg@control ansible]$ ansible-galaxy list
20 # /home/greg/ansible/roles
21 # /usr/share/ansible/roles
22 - linux-system-roles.kdump, (unknown version)
23 - linux-system-roles.network, (unknown version)
24 - linux-system-roles.postfix, (unknown version)
25 - linux-system-roles.selinux, (unknown version)
26 - linux-system-roles.timesync, (unknown version)
27 - rhel-system-roles.kdump, (unknown version)
28 - rhel-system-roles.network, (unknown version)
29 - rhel-system-roles.postfix, (unknown version)
30 - rhel-system-roles.selinux, (unknown version)
31 - rhel-system-roles.timesync, (unknown version)
```

```
1 [greg@control ansible]$ cat timesync.yml
2 ---
3 - name: use system role
4   hosts: all
5   vars:
6     timesync_ntp_servers:
7       - hostname: 172.25.254.254
```

```
8         iburst: yes
9     roles:
10         - rhel-system-roles.timesync
11
12
13
14     #在运行之前，使用adhoc检查一下
15     [greg@control ansible]$ ansible all -a 'chronyc sources'
16
17     [greg@control ansible]$ ansible-playbook timesysnc.yml
18
19
20     #在运行之后，使用adhoc检查一下
21     [greg@control ansible]$ ansible all -a 'chronyc sources'
22     [greg@control ansible]$ ansible all -m shell -a 'grep -i iburst
    /etc/chrony.conf'
```

5-RH294解法第5题-使用ansible galaxy创建role

使用Ansible Galaxy安装角色

用Ansible Galaxy安装角色:

- ☐ 1. 创建playbook为/home/greg/ansible/roles/requirements.yml。从以下URL下载角色并安装到/home/greg/ansible/roles :
- ☐ 2. <http://rhgls.domainx.example.com/materials/haproxy.tar>此角色的名字设置为balancer
- ☐ 3. <http://rhgls.domainx.example.com/materials/phpinfo.tar>此角色的名字设置为phpinfo

```
1  #创建一个playbook, 这个playbook下载并且安装两个role
2  [greg@control ansible]$ cat requirements.yml
3  - src: http://rhgls.domainx.example.com/materials/haproxy.tar
4    name: balancer
5
6  - src: http://rhgls.domainx.example.com/materials/phpinfo.tar
7    name: phpinfo
8
9
10 #运行这个playbook之前
11 [greg@control ansible]$ ansible-galaxy list
12
13 [greg@control ansible]$ ls roles/
14
15
16 #运行playbook安装role
17
18 #运行之后
19 [greg@control ansible]$ ansible-galaxy list
20 # /home/greg/ansible/roles
21 - balancer, (unknown version)
22 - phpinfo, (unknown version)
23 # /usr/share/ansible/roles
24 - linux-system-roles.kdump, (unknown version)
25 - linux-system-roles.network, (unknown version)
26 - linux-system-roles.postfix, (unknown version)
27 - linux-system-roles.selinux, (unknown version)
28 - linux-system-roles.timesync, (unknown version)
29 - rhel-system-roles.kdump, (unknown version)
30 - rhel-system-roles.network, (unknown version)
31 - rhel-system-roles.postfix, (unknown version)
32 - rhel-system-roles.selinux, (unknown version)
33 - rhel-system-roles.timesync, (unknown version)
34 [greg@control ansible]$ ls roles/
35 balancer  phpinfo
36
```

6-RH294解法第6题-创建一个web role

创建一个web role

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

- ☐ 1.装httpd包，并启动httpd服务，设置服务下次开机启动
- ☐ 2.启动防火墙服务，设置防火墙服务下次开机启动，放行web流量
- ☐ 3.创建index.html.j2文件，该模板文件用于输出如下文件 /var/www/html/index.html：
 - Welcome to **HOSTNAME** on **IPADDRESS**
 - **HOSTNAME** 是受管节点的FQDN，**IPADDRESS** 则是受管节点的 IP 地址。
- ☐ 4.创建 playbook /home/greg/ansible/apache.yml ,在 webserver 主机组使用apache 的角色。

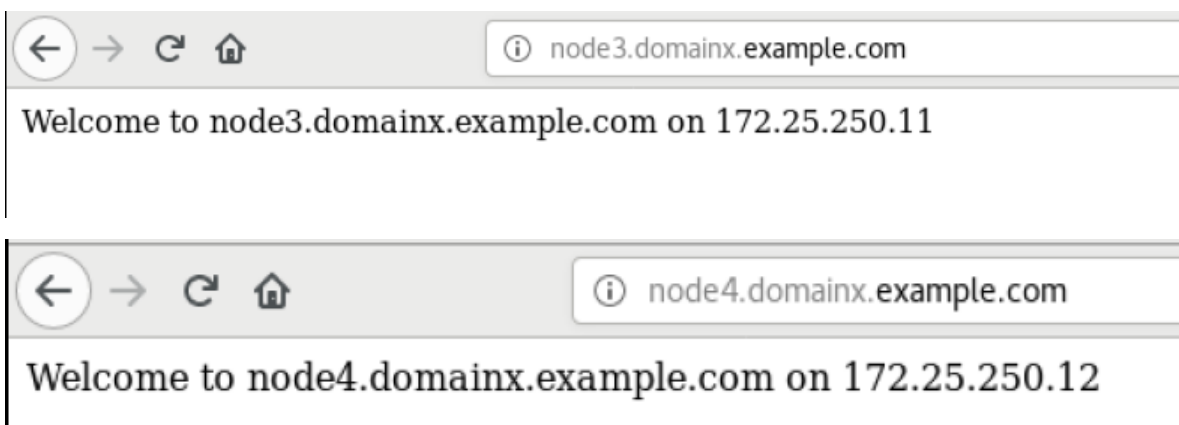
```
1 [greg@control ansible]$ cd roles/
2
3 [greg@control roles]$ ansible-galaxy init apache
4 - apache was created successfully
5
6 [greg@control roles]$ ls
7 apache balancer phpinfo
8
9 [greg@control roles]$ cd ..
10 [greg@control ansible]$ ansible-galaxy list
11 # /home/greg/ansible/roles
12 - balancer, (unknown version)
13 - phpinfo, (unknown version)
14 - apache, (unknown version)
15 # /usr/share/ansible/roles
16 - linux-system-roles.kdump, (unknown version)
17 - linux-system-roles.network, (unknown version)
18 - linux-system-roles.postfix, (unknown version)
19 - linux-system-roles.selinux, (unknown version)
20 - linux-system-roles.timesync, (unknown version)
21 - rhel-system-roles.kdump, (unknown version)
22 - rhel-system-roles.network, (unknown version)
23 - rhel-system-roles.postfix, (unknown version)
24 - rhel-system-roles.selinux, (unknown version)
25 - rhel-system-roles.timesync, (unknown version)
26
27
```

```
1 [greg@control ansible]$ vim roles/apache/templates/index.html.j2
2 [greg@control ansible]$ cat roles/apache/templates/index.html.j2
3 welcome to {{ ansible_fqdn }} on {{ ansible_default_ipv4['address'] }}
4
5 [greg@control ansible]$ vim roles/apache/tasks/main.yml
6 [greg@control ansible]$ cat roles/apache/tasks/main.yml
7 ---
8 - name: install pkg
9   yum:
10     name: httpd
11     state: latest
```

```

12
13 - name: set httpd service
14   service:
15     name: httpd
16     state: started
17     enabled: yes
18
19 - name: set firewall service
20   service:
21     name: firewall
22     state: started
23     enabled: yes
24
25 - name: set firewall to allow http traffic
26   firewall:
27     service: http
28     immediate: yes
29     permanent: yes
30     state: enabled
31
32 - name: set web content
33   template:
34     src: index.html.j2
35     dest: /var/www/html/index.html
36
37
38
39
40 [greg@control ansible]$ vim apache.yml
41 [greg@control ansible]$ cat apache.yml
42 ---
43 - name: use apache role
44   hosts: webservers
45   roles:
46     - apache
47
48 [greg@control ansible]$ ansible-playbook apache.yml
49
50
51 [greg@control ansible]$ curl node3
52 welcome to node3.domainx.example.com on 172.25.250.11
53 [greg@control ansible]$ curl node4
54 welcome to node4.domainx.example.com on 172.25.250.12
55

```



7-RH294解法第7题-从ansible galaxy使用role

```
1 [greg@control ansible]$ cat roles.yml
2 ---
3 - name: gather facts from webserver
4   hosts: webserver
5
6 - name: use haproxy role
7   hosts: balancers
8   roles:
9     - balancer
10
11 - name: use phpinfo role
12   hosts: webserver
13   roles:
14     - phpinfo
[greg@control ansible]$ ansible-playbook roles.yml
```

```
1 [greg@control ansible]$ curl node5
2 welcome to node3.domainx.example.com on 172.25.250.11
3 [greg@control ansible]$ curl node5
4 welcome to node4.domainx.example.com on 172.25.250.12
5 [greg@control ansible]$ curl node5
6 welcome to node3.domainx.example.com on 172.25.250.11
7 [greg@control ansible]$ curl node5
8 welcome to node4.domainx.example.com on 172.25.250.12
```

node3.domainx.example.com/hello.php

Hello PHP World from node3.domainx.example.com

PHP Version 7.2.11



System	Linux node3.domainx.example.com 4.18.0-80.el8.x86_64 #1 SMP Wed Mar 13 12:02:46 UTC 2019 x86_64
Build Date	Oct 9 2018 15:09:36
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-imagick.ini, /etc/php.d/20-ldap.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mcrypt.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-openssl.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-pdo_mysql.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-posix.ini, /etc/php.d/20-shmop.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlrpc.ini, /etc/php.d/20-zlib.ini

node4.domainx.example.com/hello.php

Hello PHP World from node4.domainx.example.com

PHP Version 7.2.11



System	Linux node4.domainx.example.com 4.18.0-80.el8.x86_64 #1 SMP Wed Mar 13 12:02:46 UTC 2019 x86_64
Build Date	Oct 9 2018 15:09:36
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-imagick.ini, /etc/php.d/20-ldap.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mcrypt.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-openssl.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-pdo_mysql.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-posix.ini, /etc/php.d/20-shmop.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlrpc.ini, /etc/php.d/20-zlib.ini

8-RH294解法第8题-创建和使用逻辑卷

创建和使用逻辑卷

创建一个叫做lv.yml的playbook，它将在所有被管理节点上执行下列任务：

- ☐ 1.创建逻辑卷
 - 通过research卷组创建逻辑卷
 - 逻辑卷的名称叫做data
 - 逻辑卷大小是1500MB
- ☐ 2.格式化逻辑卷为ext4文件系统
- ☐ 3.如果逻辑卷不能被创建（由于大小不满足），应显示错误"Could not create logical volume of that size"，然后创建800MB代替
- ☐ 4.如果卷组research不存在，应显示错误"Volume group does not exist"
- ☐ 5.不要以任何方式挂载逻辑卷

```
1 [greg@control ansible]$ cat lv.yml
2 ---
3 - name: create lv
4   hosts: all
5   tasks:
6     - block:
7       - name: create a lv use research vg
8         lvol:
9           vg: research
10          lv: data
11          size: 1500
12       - name: format ext4 fs
13         filesystem:
14           fstype: ext4
15           dev: /dev/research/data
16     rescue:
17       - name: output some info
18         debug:
19           msg: Could not create logical volume of that size
20           when: ansible_lvm.vgs.research is defined
21       - name: create a lv use research vg
22         lvol:
23           vg: research
24           lv: data
25           size: 800
26           when: ansible_lvm.vgs.research is defined
27       - name: format ext4 fs
28         filesystem:
29           fstype: ext4
30           dev: /dev/research/data
31           when: ansible_lvm.vgs.research is defined
32       - name: output some info
33         debug:
34           msg: Volume group does not exist
35           when: ansible_lvm.vgs.research is undefined
36
```

9-RH294解法第9题-生成主机文件

生成主机文件

- ☐ 1. 下载<http://rhgls.domainx.example.com/materials/hosts.j2>到/home/greg/ansible
- ☐ 2. 修改模板文件，让他能用来为每个inventory主机生成和/etc/hosts同样格式的文件
- ☐ 3. 创建名为 /home/greg/ansible/hosts.yml 的 playbook，它将使用此模板在 dev 主机组中的主机上生成文件 /etc/myhosts
- ☐ 4. 该 playbook 运行后，dev 主机组中主机上的文件 /etc/myhosts 应针对每个受管主机包含一行内容
 - 127.0.0.1 localhost localhost.localhost localhost4 localhost4.localhost4
 - ::1 localhost localhost.localhost localhost6 localhost6.localhost6
 - 172.25.250.9 node1.domainx.example.com node1
 - 172.25.250.10 node2.domainx.example.com node2
 - 172.25.250.11 node3.domainx.example.com node3
 - 172.25.250.12 node4.domainx.example.com node4
 - 172.25.250.13 node5.domainx.example.com node5

```
1 [greg@control ansible]$ cat hosts.j2
2 127.0.0.1 localhost localhost.localhost localhost4 localhost4.localhost4
3 ::1 localhost localhost.localhost localhost6 localhost6.localhost6
4 {% for host in groups['all'] %}
5 {{ hostvars[host]['ansible_default_ipv4']['address'] }} {{ hostvars[host]
6 ['ansible_fqdn'] }} {{ hostvars[host]['ansible_hostname'] }}
7 {% endfor %}
8 ---
9 - name: create a host file
10   hosts: all
11   tasks:
12     - name: template a host file
13       template:
14         src: hosts.j2
15         dest: /etc/myhosts
16         when: '"dev" in group_names'
17
18 [greg@control ansible]$ ansible all -a 'cat /etc/myhosts'
19 node5 | FAILED | rc=1 >>
20 cat: /etc/myhosts: No such file or directorynon-zero return code
21
22 node3 | FAILED | rc=1 >>
23 cat: /etc/myhosts: No such file or directorynon-zero return code
24
25 node2 | FAILED | rc=1 >>
26 cat: /etc/myhosts: No such file or directorynon-zero return code
27
28 node4 | FAILED | rc=1 >>
29 cat: /etc/myhosts: No such file or directorynon-zero return code
30
31 node1 | CHANGED | rc=0 >>
32 127.0.0.1 localhost localhost.localhost localhost4 localhost4.localhost4
33 ::1 localhost localhost.localhost localhost6 localhost6.localhost6
34 172.25.250.9 node1.domainx.example.com node1
35 172.25.250.10 node2.domainx.example.com node2
36 172.25.250.13 node5.domainx.example.com node5
37 172.25.250.11 node3.domainx.example.com node3
38 172.25.250.12 node4.domainx.example.com node4
39
```


10-RH294解法第10题-修改文件内容

修改文件内容

按照下方所述，创建一个名为 /home/greg/ansible/issue.yml 的 playbook

- ☐ 1. playbook 将在所有inventory主机上运行
- ☐ 2. 该 playbook 会将 /etc/issue 的内容替换为下方所示的一行文本：
 - 在 dev 主机组中的主机上，这行文本显示 为：Development
 - 在 test 主机组中的主机上，这行文本显示 为：Test
 - 在 prod 主机组中的主机上，这行文本显示 为：Production

```
1 [greg@control ansible]$ cat issue.yml
2 ---
3 - name: modify file content
4   hosts: all
5   tasks:
6     - copy:
7       content: Development
8       dest: /etc/issue
9       when: '"dev" in group_names'
10    - copy:
11      content: Test
12      dest: /etc/issue
13      when: '"test" in group_names'
14    - copy:
15      content: Production
16      dest: /etc/issue
17      when: '"prod" in group_names'
18
19
20
21
22 [greg@control ansible]$ ansible all -a 'cat /etc/issue'
23 node5 | CHANGED | rc=0 >>
24 \S
25 Kernel \r on an \m
26
27 node3 | CHANGED | rc=0 >>
28 Production
29
30 node2 | CHANGED | rc=0 >>
31 Test
32
33 node4 | CHANGED | rc=0 >>
34 Production
35
36 node1 | CHANGED | rc=0 >>
37 Development
38
39
```


11-RH294解法第11题-创建web内容目录

创建web内容目录

按照下方所述，创建一个名为 /home/greg/ansible/webcontent.yml 的 playbook

- ☐ 1.该 playbook 在 dev 主机组中的受管节点上运行
- ☐ 2.建符合下列要求的目录 /webdev
 - 所有者为 webdev 组
 - 具有常规权限：owner=rwx, group=rwx, other=rx
 - 有特殊权限：set gid
- ☐ 3.用符号链接将 /var/www/html/webdev 链接到 /webdev
- ☐ 4.创建文件 /webdev/index.html，其中包含如下所示的单行文件：Development
- ☐ 5.在 dev 主机组中主机上浏览此目录（例如 <http://node1.domainx.example.com/webdev>）将生成以下输出：Development

```
1 [greg@control ansible]$ cat webcontent.yml
2 ---
3 - name: set web content
4   hosts: dev
5   tasks:
6     - name: create a directory
7       file:
8         path: /webdev
9         state: directory
10        group: webdev
11        mode: "2775"
12        setype: "httpd_sys_content_t"
13    - name: create a soft link
14      file:
15        src: /webdev
16        dest: /var/www/html/webdev
17        state: link
18    - name: set web content
19      copy:
20        content: Development
21        dest: /webdev/index.html
22        setype: "httpd_sys_content_t"
23    - name: start httpd service
24      service:
25        name: httpd
26        state: started
27        enabled: yes
28    - name: set firewall rule to allow http traffic
29      firewallld:
30        service: http
31        permanent: yes
32        immediate: yes
33        state: enabled
34
35 [greg@control ansible]$ curl http://node1.domainx.example.com/webdev/
36 Development
```

12-RH294解法第12题-生成硬件报告

生成硬件报告

- ☐ 1. 建一个名为 `/home/greg/ansible/hwreport.yml` 的 playbook，它将在所有受管节点上生成含有以下信息的输出文件 `/root/hwreport.txt`
 - 清单主机名称
 - 以 MB 表示的总内存大小
 - BIOS 版本
 - 磁盘设备 `vda` 的大小
 - 磁盘设备 `vdb` 的大小
 - 输出文件中的每一行含有一个 `key=value` 对
- ☐ 你的 playbook 应当
 - 从 `http://rhgl.s.domainx.example.com/materials/hwreport.empty` 下载文件，并将它保存为 `/root/hwreport.txt`
 - 使用正确的值改为 `/root/hwreport.txt`
 - 果硬件项不存在，相关的值应设为 `NONE`

```
1 [greg@control ansible]$ cat hwreport.yml
2 ---
3 - name: create hardware report
4   hosts: all
5   vars:
6     hardware:
7       - hw_name: HOST
8         hw_info: "{{ ansible_hostname }}"
9       - hw_name: MEMORY
10        hw_info: "{{ ansible_memtotal_mb }}"
11       - hw_name: BIOS
12        hw_info: "{{ ansible_bios_version }}"
13       - hw_name: DISK_SIZE_VDA
14        hw_info: "{{ ansible_devices['vda']['size'] | default('NONE') }}"
15       - hw_name: DISK_SIZE_VDB
16        hw_info: "{{ ansible_devices['vdb']['size'] | default('NONE') }}"
17   tasks:
18     - name: get hw report from url
19       get_url:
20         url: http://rhgl.s.domainx.example.com/materials/hwreport.empty
21         dest: /root/hwreport.txt
22     - name: set hw report content
23       lineinfile:
24         path: /root/hwreport.txt
25         regexp: "^{{ item['hw_name'] }}"
26         line: "{{ item['hw_name'] }}={{ item['hw_info'] }}"
27         loop: "{{ hardware }}"
28
```

13-RH294解法第13题-使用ansible vault

使用Ansible Vault

- ☐ 1.vault的名字是locker.yml
- ☐ 2.vault包含两个变量
 - pw_developer的值是Imadev
 - pw_manager的值是Imamgr
- ☐ 3.加密解密的密码为whenyouwishuponastar
- ☐ 4.密码存放在/home/greg/ansible/secret.txt

```
1 [greg@control ansible]$ cat secret.txt
2 whenyouwishuponastar
3
4 [greg@control ansible]$ cat > locker.yml <<END
5 pw_developer: Imadev
6 pw_manager: Imamgr
7 END
8
9 [greg@control ansible]$ ansible-vault encrypt --vault-id=./secret.txt
locker.yml
10
11 [greg@control ansible]$ ansible-vault view --vault-id=./secret.txt
locker.yml
12 pw_developer: Imadev
13 pw_manager: Imamgr
14
```


14-RH294解法第14题-批量创建用户

创建批量添加用户role

☐ 1.在<https://rhgl.s.domainx.example.com/materials>中下载user_list.yml到/home/greg/ansible中

☐ 2.用locker.yml变量文件，创建users.yml来创建账号

- 带developer描述的job的用户应该满足如下条件

在dev和test主机组下创建用户

设置密码为pw_developer变量的值

附加组为devops

- 带manager描述的job的用户应该满足如下条件

在prod主机组下创建用户

设置密码为pw_manager变量的值

附加组为opsmgr

☐ 3.密码应该是SHA512的格式

☐ 4.playbook应该使用vault密码文件来工作

```
1 [greg@control ansible]$ cat users.yml
2 ---
3 - name: create user on dev and test
4   hosts: dev,test
5   vars_files:
6     - locker.yml
7     - user_list.yml
8   tasks:
9     - name: create group
10      group:
11        name: devops
12    - name: create user
13      user:
14        name: "{{ item['name'] }}"
15        password: "{{ pw_developer | password_hash('sha512',
16 'mysecretsalt') }}"
17      groups: devops
18      loop: "{{ users }}"
19      when: item.job == 'developer'
20 - name: create user on prod
21   hosts: prod
22   vars_files:
23     - locker.yml
24     - user_list.yml
25   tasks:
26     - name: create group
27      group:
28        name: opsmgr
29    - name: create user
```

```
30         user:
31             name: "{{ item['name'] }}"
32             password: "{{ pw_manager | password_hash('sha512',
'mysecretsalt') }}"
33             groups: opsmgr
34             loop: "{{ users }}"
35             when: item.job == 'manager'
36
37
38
39
40
41 [greg@control ansible]$ ansible-playbook --vault-id=./secret.txt users.yml
```

15-RH294解法第15题-重新设置vault密码

重新设置Ansible Vault密码

- ☐ 1. 载<http://rhgls.domainx.example.com/materials/salaries.yml>, 保存到/home/greg/ansible/salaries.yml
- ☐ 2. 当前的密码是insecure4sure
- ☐ 3. 新的密码是bbe2de98389b
- ☐ 4. vault文件使用新的密码保持为加密的状态

```
1 [greg@control ansible]$ wget
  http://rhgls.domainx.example.com/materials/salaries.yml
2
3
4 [greg@control ansible]$ ansible-vault rekey salaries.yml
5 vault password: insecure4sure
6 New Vault password: bbe2de98389b
7 Confirm New Vault password: bbe2de98389b
8
```