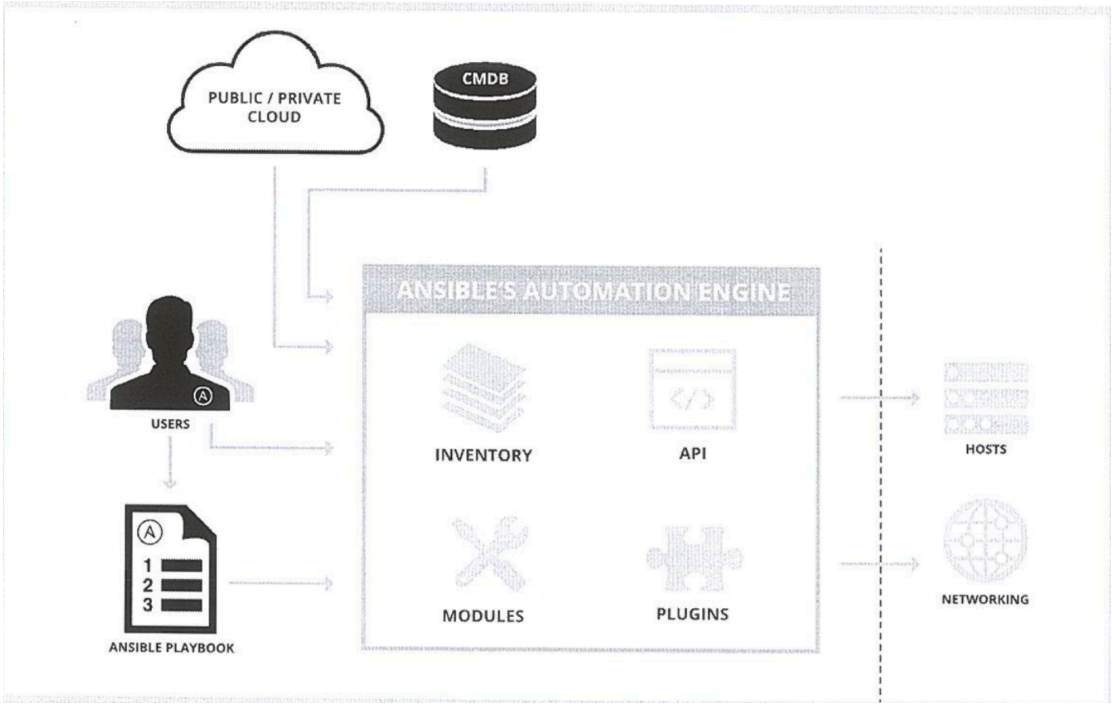
管理 III：RH294/DO407　考试 4 小时

#rht-vmtl. reset all

Ansible 是什么：
Ansible 自动化运维工具，基于 Python 开发，实现批量系统配置（用户，软件，磁盘，网络）

Ansible 基于模块化工作：
本身没有批量部署能力，ansible 只提供了一个框架，真正实现批量系统配置通过模块实现，ansible 不需要在远程节点（被管理的节点）安装 agent，它是基于 ssh 协议和远程主机进行通讯
SSH 身份验证：
1 基于用户名和密码
2 基于密钥的身份验证
控制节点：
ssh-keygen.　一路回车，生成一对密钥，然后把公钥拷贝到被管理的节点
Ansible 架构：
由几个核心组件组成： CMDB 配置管理数据库　RHN
1 ansible(主体) ansible 的引擎，提供了一个命令的接口给到用户对 ansible 进行管理操作
2 Host Inventory(清单文件)　动态清单文件（公有云，私有云）　理解成有点类似于 hosts
3 Core Modules(核心模块)　Ansible 执行命令的功能模块
4 Custom Modules(扩展模块)　用户自定义相应的模块为满足企业的需求
5 Connection Plugin(连接插件) 模块的补充，变量插件，过滤的插件，循环插件
6 Playbook （任务剧本） 一个剧本文件中定义多个任务，由 ansible 顺序依次执行 YAML
7 API：给第三方程序调用应用程序编程的接口



-m setup 模块，收集远程主机的一些基本信息

第二章 部署 ANSIBLE

清单文件：

1 静态清单文件

2 动态清单文件

清单文件定义 ansible 将要管理的一批主机，主机进行分组，进行集中管理

组可以包含子组，一个主机可以属于多个组的成员

静态清单文件：

192.168.0.1
192.168.0.2
192.168.0.3
rds1.cloudshelledu.com
rds2.cloudshelledu.com
db1.cloudshelledu.com
db2.cloudshelledu.com

静态清单文件中定义了两个组：rds. db

[rds]
rds1.cloudshelledu.com
rds2.cloudshelledu.com
192.168.0.3
[db]
db1.cloudshelledu.com
db2.cloudshelledu.com

默认组始终存在：

all. 清单文件中列出的每一个主机

ungrouped. 主机组含有清单中明确列出、但不属于任何其它组的每一个主机

定义嵌套组

[rds]
rds1.cloudshelledu.com
rds2.cloudshelledu.com
192.168.0.3
[db]
db1.cloudshelledu.com
db2.cloudshelledu.com
[server:children]
rds
db

简化主机的定义：
[start:end]
192.168.[4:7].[0:255]
db[01:20].example.com. server01.example.com 到 server20.example.com
[a:c].cloudshelledu.com. a.cloudshelledu.com 到 c.cloudshelled.com

/etc/ansible/hosts. 默认清单文件
mkdir ~student/playbooks.
cd ~student/playbooks
vi inventory
[start:end]
192.168.[4:7].[0:255]
db[01:20].example.com. server01.example.com 到 server20.example.com
[a:c].cloudshelledu.com. a.cloudshelledu.com 到 c.cloudshelled.com

ansible a.cloudshelledu.com –list-hosts -i inventory
-i --inventory. 清单文件 ,而不是系统/etc/ansible/hosts 清单文件

动态清单：
使用外部数据库提供的信息生成动态清单文件 RHN

Ansible 配置文件:
/etc/ansible/ansible.cfg 主配置文件，由 ansible 软件包提供 优先级最低
~/.ansible.cfg. 在用户家目录下创建，覆盖全局的配置文件
./ansible.cfg 如果执行 ansible 命令目录中中存在 ansible.cfg 文件，使用它 （推荐使用）

配置文件设置：
[defaults] 默认值
inventory. = /home/student/playbooks/inventory 清单文件绝对路径
inventory. = /home/student/inventory. 清单文件目录绝对路径
remote_user = student 在被管理节点上登录的用户名，如果没有指定用户，使用当前用户
become = true 连接被管理节点上切换哪个用户,总开关
become_method = su
become_method = sudo 默认值
become_user = root 在被管理节点上切换到哪个用户, 默认值 root
become_ask_pass= true or false 提示输入密码 默认值 false

运行临时的命令：
$ ansible. host-pattern. -m module.[-a ‘module arguments’]. [-i inventory]

host-pattern. 运行临时命令是在哪些主机上执行
-m setup. ping. command
-a 参数 通常情况使用模块都是加参数
-I 清单文件

ansible-doc. -l. 列出系统上安装的所有模块
ansible-doc ping. 查看 ping 模块帮助
文件模块：
copy 复制文件
file 修改文件属性，权限
lineinfile. 文件内容替换

软件包模块：
yum
dnf

系统模块：
reboot
service
firewalld

网络工具模块：
get_url. 通过 HTTP\HTTPS\FTP 下载文件
uri. 与 WEB 交互
nmcli. 管理网络

ansible -m user. -a “name=natasha uid=50000 shell=/sbin/nologin” servera.lab.example.com

学习过的命令：
useradd. groupadd. chmod. chown. chgrp. yum. systemctl. fdisk.

在被管理的主机上执行任意命令 command
ansible -m command. -a “yum update -y”

ansible 命令选项：
在被管理的主机上执行任意命令 command
ansible -m command. -a “yum update -y”

-I. inventory
-u remote_user.
-b. become
--become-method. become_mothod
--become-user. become_user
-K. become_ask_pass

$. ansible –help

LAB:
Create a shell script called /home/devops/ansible/adhoc.sh that runs ans Ansible ad-hoc command to create a yum repository on servera.lab.example.com
The name of the repository is CLOUDSHELL
The Description is BaseOS Software
The bash URL is http://content.example.com/rhel8.0/x86_64/dvd/BaseOS
The GPG key URL is
http://content.example.com/rhel/RPM-GPG-KEY-redhat-release
The repository is enabled

第三章 实施 PLAYBOOK

ANSIBLE PLAYBOOK 和临时命令

1 临时命令 用于一次性自动化操作或执行一些简单的自动化任务

2 ANSIBLE PLAYBOOK 重复性操作一些复杂自动化操作

Play：

针对清单中的主机运行一组任务

playbook 就是一个文本文件

方法 1：

ansible -m user. -a "name=natasha uid=50000 shell=/sbin/nologin" servera.lab.example.com

方法 2： YMAL 格式 xxxxx.yml.

```
---
-    name： Create user. 第一个 play 开头是以破折号加空格,一个 play 里可以有多个任务
     hosts:    servera.lab.example.com
     tasks:
          - name: newuser natasha
            user:
                name: natasha
                uid: 50000
                shell: /sbin/nologin
          - name: install web package
            yum:
                name: httpd
                state: latest

-    name： Create user.    第二个 play 开头是以破折号加空格
     hosts:    serverb.lab.example.com
     tasks:
          - name: newuser harry
            user:
                name: harry
                uid: 60000
                shell: /bin/bash
```

注意：使用空格字符缩进来表示数据结构，空格的数据没有要求

条件：

--- 以破折号开始

Play： 键值对的集合

name hosts. tasks 三个键值要有相同的缩进

排错：

$ ansible-playbook. new.yml

-v 显示任务结果

-vv 显示任务结果，任务配置

-vvv 被管理主机连接信息

--syntax-check. 检测语法

-C 空运行 Ansible 执行 playbook 里的任务，但不会实际修改被管理主机

实施多个 Play

Playbook 是一个 YAML 文件，包含一个或多个 PLAY

```
---
-   name： Create user. 第一个 play 开头是以破折号加空格,一个 play 里可以有多个任务
    hosts:  servera.lab.example.com      针对 servera 主机操作
    become: true.                        在 play 中特权升级
    tasks:
        - name: newuser natasha
          user:
              name: natasha
              uid: 50000
              shell: /sbin/nologin
        - name: install web package
          yum:
              name: httpd
              state: latest

-   name： Create user.  第二个 play 开头是以破折号加空格
    hosts:  serverb.lab.example.com  针对 serverb 主机操作
    become: true
    tasks:
        - name: newuser harry
          user:
              name: harry
              uid: 60000
              shell: /bin/bash
```

模块帮助：http://docs.ansible.com

$ ansible-doc -l

$ ansible-doc ping

```
---
-    name：  Create user.  第一个 play 开头是以破折号加空格,一个 play 里可以有多个任务
     hosts:
        - servera.   列表
        - serverb
        - serverc
     become: true.                              在 play 中特权升级
     tasks:
         - name: newuser natasha
           user:
               name: natasha
               uid: 50000
               shell: /sbin/nologin
       - name: install web package
           yum:
               name: httpd
               state: latest


---
-    name：  Create user.  第一个 play 开头是以破折号加空格,一个 play 里可以有多个任务
     hosts:     [servera,serverb,serverc]


       - name: install web package
         yum: name=httpd.   state=latest
```

Play: 一组任务列表，这些任务在特定的主机上运行
Playbook: 文本文件，包含了一个或多个 Play 组成的列表
Ansible Playbook 以 YAML 格式编写
YAML 文件中结构使用空格进行缩进
运行 playbook 使用使用 ansible-playbook
查询帮助 ansible-doc

第四章 管理变量和事实

Ansible 变量：
Ansible 支持利用变量来存储值，在 Ansible 项目中的所有文件中重复使用这些值

变量可能包含以下内容：
用户变量
安装软件包变量
启动服务变量
删除文件变量
管理文件变量

变量使用：
变量名称必须以字母开头，并且只有包含字母、数字、下划线

无效变量	有效变量
web server	web_server
remote.file.	remote_file

定义变量：
Ansible 项目中多个位置定义变量
1 全局范围 从命令行或 Ansible 配置文件中设置变量
2 Play 范围 在 play 中定义变量
3 主机范围 在清单、事实收集

在多个范围中定义了相同名称的变量，优先级，小范围优先于更广泛的范围

Playbook 中定义变量
方法 1：

```
---
-   name: first play
    hosts: all
    vars:
      user: natasha
      shell: /sbin/nologin
```

方法 2：在外部文件中定义 playbook 变量

```
---
-   name: first play
    hosts: all
    vars_files:
     - vars/users.yml
```

$ vi. vars/users.yml
user: natasha
shell: /sbin/nologin

在 Playbook 中使用变量,引用变量，将变量名称放在又花括号 {{ }}

```
---
- name: first play
  hosts: all
  vars:
    user: natasha
    shell: /sbin/nologin
  tasks:
  user:
    name: "{{ user }}"
```

主机变量：
[webservers]
db.cloudshelledu.com. ansible_user=natasha

使用目录填充主机和组变量
host_vars
group_vars

如果大家要定义针对 webservers 主机组的组变量
group_vars/webservers
user: natasha
shell: /sbin/nologin

host_vars/servera.lab.example.com
package: httpd

host_vars/serverb.lab.example.com
package: vsftpd

命令行中定义变量：优先级最高
$ ansible-playbook. httpd.yml -e "package=httpd"

使用数组作为变量：

```
user1_first_name: Bob
user1_last_name: Cook
user1_shell: /sbin/nologin

users:      数组
  natasha:
      first_name: Bob
      last_name: Cook
      user_shell: /sbin/nologin
```

users.natasha.first_name
users.natasha.last_name
users.natasha.user_shell

注册变量：

使用 register 语句捕获命令输出
输出保存在一个临时变量中
任务 1：

```
        - name: install web package
          yum:
              name: httpd
              state: latest
          register: install_result        注册变量.  和  判断语句
        - debug:
            var: install_result
```

任务 2：

```
        - name：start httpd
          service:
              name: httpd
              state: started
```

管理机密：

ansible vault
使用 ansible vault 加密敏感的变量
ansible-vault 命令行工具创建、编辑、加密、解密和查看文件
创建加密的文件：

```
$ ansible-vault create passwd.yml
$ ansible-vault create –vault-password-file=vault-pass passwd.yml
```

查看加密文件

```
$ ansible-vault view filename
```

编辑现有加密的文件
$ ansible-vault edit password.yml

加密现有的文件
ansible-vault encrypt filename

解密现有的文件
ansible-vault decrypt filename

更改加密文件的密钥：
ansible-vault rekey filename

如果大家在 playbook 中引用加密的变量文件
ansible-playbook. –vault-password-file=pw http.yml

管理事实(facts)
主机名
内核版本
操作系统版本
CPU 数量
可用磁盘空间
可用的内存大小