playbook基础

1. 基础知识

ansible七种武器

json简介

YAML简介

jinja2模板简介

1. playbook

palybook是什么

playbook语法基础

1.ansible七种武器

第一种武器

ansible命令，用于执行临时性的工作，必须掌握

第二种武器

absible-doc 模块文档说明

第三种武器

ansible-console是ansible为用户提供的交互式工具

第四种武器

ansible-galaxy从github下载管理Roles的一款工具

第五种武器

ansible-playbook通过读取先编写好的playbook文件实现批量管理，可以理解为按一定条件组成的ansible任务集

第六种武器

ansible-vault 主要用于配置文件加密，可用加密解密这个文件

第七种武器

两种工作模式pull/push默认push

JSON简介

json是javascript对象表示法，一种基于文本独立语言的轻量及数据交换格式

json中分隔符仅限于（单引号、小括号、中括号、大括号、冒号、逗号）-->英文

JSON特性：

json是纯文本

json具有自我描述性

json具有层级结构

json可通过javascripts进行解析

JSON语法规则：

数据在名称/值对 中

数据由逗号分割

大括号保存对象

中括号保存数组

json数据中的书写格式是： 名称/值对

- 名称/值对包括字段名称（在双引号中），后面写一个冒号，然后是值，例如

“诗仙”: “李白”

JSON语法规则之数组

{ “诗人”:

[“李白”,”杜浦”,”白居易”,”李贺”]

}

复合复杂类型

{ “诗人”:

[{“李白”:”诗仙”,”年代”:”唐”},

{“李贺”:”诗鬼”,”年代”:”唐”}

]

}

YAML简介

--是一个可读性高，用来表达数据序列的格式

YAML基础语法

--YAML的结构通过空格来展示

--数组使用“- ”来表示 ----> ’- 空格‘

--键值对使用“： ”表示 --->‘- 空格’

--YAML使用一个固定的缩进风格表示数据层级结构关系

--一般每个缩进级别有两个以上空格组成

--#表示注释

注意：

不要使用tab，同一层级缩进必须对齐

YAML的键值表示方法

--采用冒号分隔

-- ：后面必须有一个空格

--键值对例子

“诗人”: “李白”

或 “李白”:

“诗人”

复杂的YAML的键值对嵌套

“诗人”:

“李白”: “诗仙”

或

playbook

tasks:

-name: tasks下的name是注释

playbook进阶

变量 给主机添加用户 --> ansible-doc user

vim vars.yml

---

- hosts: db

remote\_user: root

vars:

username: xinxin

tasks:

- name : add user "{{username}}"

user:

name: "{{username}}"

- shell: echo 123 | passwd --stdin "{{username}}"

- shell: chage -d 0 "{{username}}"

vim aa 创建调用文件

{"username": "xxhh"}

[root@ansible ~]# ansible-playbook vars.yml -e "@aa"

设密码：解决密码明文问题、

变量过滤器

“{{‘123’|password\_huah(‘sha512’)}}”

vim vars.yml

---

- hosts: cache

remote\_user: root

vars:

username: xinxin

tasks:

- name : add user "{{username}}"

user:

name: "{{username}}"

password: "{{'123'|password\_hash('sha512')}}"

error

ansible-playbook对错误的处理

用户存在 $? 返回值非0

vim user.yml

---

- hosts: db2

remote\_user: root

tasks:

- shell: useradd z3 || true

- shell: adduser li4

ignore\_errors: true

- shell: echo 123456 | passwd --stdin z3

- shell: echo 123456 | passwd --stdin li4

handlers 与tasks 同一层级

notify调用的是handler段的name定义的串，必须一致

多个task触发同一个notify的时候，同一个服务只会触发一次

notify可以触发多个条件，在生产环境中往往涉及到某一个配置文件的改变要重启若干服务iu的场景，handler用到这里非常适合

tags ： 给指定的任务定义一个调用标识

调用方式：

ansible-playbook httpd.yml --tags=reload apache

vim httpd.yml

---

- hosts: web

remote\_user: root

tasks:

- name: install the latest version of Apache

yum:

name: httpd

state: latest

- lineinfile:

path: /etc/httpd/conf/httpd.conf

regexp: '^Listen '

insertafter: '^#Listen '

line: 'Listen 80'

tags: editconf

notify:

- reload apache

- lineinfile:

path: /etc/httpd/conf/httpd.conf

regexp: '^#ServerName'

line: 'ServerName localhost'

- copy:

src: index.html

dest: /var/www/html/

owner: apache

group: apache

mode: 0644

handlers:

- name: reload apache

service:

name: httpd

state: started

enabled: yes

when

解决条件判断

1）当系统负载超过0.7时，则关掉httpd

- hosts: cache

remote\_user: root

tasks:

- shell: uptime | awk '{printf("%.2f",$(NF-2))}'

register: result

- service:

name: httpd

state: stopped

when: result.stdout|float > 0.7

[root@ansible ansible]# ansible-playbook when.yml

with\_items标准循环

vim add.yml

---

- hosts: web2

remote\_user: root

tasks:

- user:

name: "{{item.name}}"

group: "{{item.group}}"

password: "{{'item.pwd'|password\_hash('sha512')}}"

with\_items:

- {name: "aa", group: "users",pwd: "aa"}

- {name: "bb", group: "mail" ,pwd: "bb"}

- {name: "cc", group: "wheel",pwd: "cc"}

- {name: "dd", group: "root" ,pwd: "dd"}

嵌套循环，循环添加多用户

[root@ansible ansible]# vim add1.yml

---

- hosts: web2

remote\_user: root

vars:

un: [a, b, c]

id: [1, 2, 3]

tasks:

- name: add users

shell: echo {{item}}

with\_nested:

- "{{un}}"

- "{{id}}"

[root@ansible ansible]# ansible-playbook add1.yml