|  |
| --- |
| 文章标题  ansible自动化批量管理 |
| **作者：**  **归档：学习笔记** |
| **快捷键：**  Ctrl + 1 标题1  Ctrl + 2 标题2  Ctrl + 3 标题3  Ctrl + 4 实例  Ctrl + 5 程序代码  Ctrl + 6 正文 |
| **格式说明：**  蓝色字体：注释  黄色背景：重要  绿色背景：注意 |

****

**老男孩教育教学核心思想6重：重目标、重思路、重方法、重实践、重习惯、重总结**

**学无止境，老男孩教育成就你人生的起点！**

# ansible自动化软件使用

## ansible软件介绍

 python语言是运维人员必会的语言！

 ansible是一个基于Python开发的自动化运维工具！

 其功能实现基于SSH远程连接服务！

 ansible可以实现批量系统配置、批量软件部署、批量文件拷贝、批量运行命令等功能

#### ansible 官网

http://docs.ansible.com/ 官网

## ansible软件特点

 不需要单独安装客户端（no agents），基于系统自带的sshd服务，sshd就相当于ansible的客户端。

 不需要服务端(no servers)。

 需要依靠大量的模块实现批量管理。?

 配置文件/etc/ansible/ansible.cfg

### ansible软件部署：

#### 受控主机软件部署：

yum install libselinux-python -y

#### 管理端主机软件部署：

yum install ansible -y

说明：安装ansible软件需要具有epel源

至此ansible软件部署完毕

#### 查看版本信息

[root@m01 home]# ansible --version

ansible 2.3.1.0

config file = /etc/ansible/ansible.cfg

configured module search path = Default w/o overrides

python version = 2.6.6 (r266:84292, Aug 18 2016, 15:13:37) [GCC 4.4.7 20120313 (Red Hat 4.4.7-17)]

## ansible软件目录结构信息：

[root@m01 home]# rpm -ql ansible|egrep -v '/usr/share|/usr/lib/'

/etc/ansible

/etc/ansible/ansible.cfg --- ansible配置文件

/etc/ansible/hosts --- 定义ansible可以管理的主机信息

/etc/ansible/roles --- 主要在自动化部署多台主机时应用

/usr/bin/ansible

/usr/bin/ansible-playbook --- 执行ansible剧本命令？

### /etc/ansible/hosts文件配置

加入一个模块，在模块下加入批量管理主机IP

[root@m01 ansible]# tail -10 /etc/ansible/hosts

# Here's another example of host ranges, this time there are no

# leading 0s:

## db-[99:101]-node.example.com

[oldboy]

172.16.1.41

172.16.1.31

172.16.1.8

# 模块知识总结：



ansible软件常用参数表

|  |  |
| --- | --- |
| 命令参数 | 参数说明 |
| -m  MODULE-NAME | * 相应名称的模块被执行（默认为command） * -m 后边是模块的名字 |
| -a  MODULE-ARGS | * 模块的参数信息 * -a后面是要执行的命令；也可以写一个ip，针对一台机器来执行命令 |
| -C  --check | * 只是测试一下会改变什么内容，不会真正去执行;相反,试图预测一些可能发生的变化 |
| --syntax-check | * 执行语法检查在剧本上，但是并不执行剧本 |

## command远程执行命令

注意：

1. 命令模块中多个参数设置用空格进行分割
2. 命令模块中不能出现 "<", ">", "|", ";" and "&"

|  |  |
| --- | --- |
| 参数选项 | 选项说明 |
| chdir | 在执行命令之前，通过cd命令进入到指定目录中 |
| creates | 定义一个文件是否存在，如果不存在运行相应命令；如果存在跳过此步骤 |
| executable | 改变shell使用command进行执行，并且执行时要使用绝对路径 |
| free\_form | 命令模块采用自由形式命令运行；即可以输入任意linux命令 |
| removes | 定义一个文件是否存在，如果存在运行相应命令；如果不存在跳过此步骤 |
| warn | 如果ansible配置文件中定义了命令警告，如果参数设置了no/false，将不会警告此行命令 |

#### chdir--在执行命令前进入到指定目录中

[root@m01 ~]# ansible test -m command -a "chdir=/tmp/ pwd"

172.16.1.222 | SUCCESS | rc=0 >>

/tmp

#### creates--定义一个文件是否存在，如果不存在运行相应命令

[root@m01 ~]# ansible test -m command -a "creates=/etc/hosts ls"

172.16.1.222 | SUCCESS | rc=0 >>

skipped, since /etc/hosts exists

#### removes--定义一个文件是否存在，如果存在运行

[root@m01 ~]# ansible test -m command -a "removes=/etc/hosts ls"

172.16.1.222 | SUCCESS | rc=0 >>

1

anaconda-ks.cfg

install.log

install.log.syslog

#### ansible 模块 -m command -a "执行命令"

[root@m01 ansible]# ansible oldboy -m command -a "free -m"

172.16.1.8 | SUCCESS | rc=0 >>

total used free shared buffers cached

Mem: 474 269 205 0 39 125

-/+ buffers/cache: 103 370

Swap: 767 0 767

172.16.1.41 | SUCCESS | rc=0 >>

total used free shared buffers cached

Mem: 474 409 65 0 105 184

-/+ buffers/cache: 119 355

Swap: 767 0 767

172.16.1.31 | SUCCESS | rc=0 >>

total used free shared buffers cached

Mem: 474 412 61 0 27 260

-/+ buffers/cache: 124 349

Swap: 767 0 767

#### 测试ansible 批量管理，执行命令

[root@m01 ansible]# ansible oldboy -m command -a "df -h"

172.16.1.41 | SUCCESS | rc=0 >>

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 8.8G 1.5G 6.9G 18% /

tmpfs 238M 0 238M 0% /dev/shm

/dev/sda1 190M 35M 146M 19% /boot

172.16.1.8 | SUCCESS | rc=0 >>

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 8.8G 1.5G 6.9G 18% /

tmpfs 238M 0 238M 0% /dev/shm

/dev/sda1 190M 35M 146M 19% /boot

172.16.1.31:/data 8.8G 1.5G 6.9G 18% /mnt

172.16.1.31 | SUCCESS | rc=0 >>

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 8.8G 1.5G 6.9G 18% /

tmpfs 238M 0 238M 0% /dev/shm

/dev/sda1 190M 35M 146M 19% /boot

#### 可以省略command ，因为command为默认模块

[root@m01 ~]# ansible oldboy -a "date"

172.16.1.8 | SUCCESS | rc=0 >>

Wed Sep 6 23:56:45 CST 2017

172.16.1.41 | SUCCESS | rc=0 >>

Thu Sep 7 08:47:47 CST 2017

172.16.1.31 | SUCCESS | rc=0 >>

Thu Sep 7 08:47:48 CST 2017

## ping 模块，测试模块

#### ping 通，说明测试成功

[root@m01 ~]# ansible oldboy -m ping

172.16.1.31 | SUCCESS => {

"changed": false,

"ping": "pong"

}

172.16.1.41 | SUCCESS => {

"changed": false,

"ping": "pong"

}

172.16.1.8 | SUCCESS => {

"changed": false,

"ping": "pong"

}

#### ping 失败，说明存在问题

[root@m01 ~]# ansible oldboy -m ping

172.16.1.41 | UNREACHABLE! => {

"changed": false,

"msg": "Failed to connect to the host via ssh: ssh: connect to host 172.16.1.41 port 22: Connection refused\r\n",

"unreachable": true

}

172.16.1.8 | SUCCESS => {

"changed": false,

"ping": "pong"

}

172.16.1.31 | SUCCESS => {

"changed": false,

"ping": "pong"

}

说明：一个简单的测试模块，这个模块总是返回"pong"在成功连接时。这个模块在剧本中是没有意义的。但是能够使用ansible命令验证登录能力和用于python的配置

这个并不是传统的ICMP ping，而是先检查能否通过ssh登录节点，再检查python版本能否满足要求，能满足要求返回pong。这只是一个简单测试模块

## copy模块

|  |  |
| --- | --- |
| 选项参数 | 选项说明 |
| src | 被复制到远程主机的本地文件，可以是绝对路径，也可以是相对路径。如果路径是一个目录，它将递归复制。在这种情况下，如果路径使用"/"来结尾，则只复制目录里面的内容，如果没有使用"/"来结尾，则包含目录在内的整个内容全部复制，类似于rsync |
| remote\_src | 如果这个值设置为True，将到远程主机上搜索，默认为False |
| dest | 将数据复制到目标路径（远程主机） |
| owner | 设置复制传输数据后的数据属主信息 |
| group | 设置复制传输后目标的属组信息 |
| content | 用于替代“src”，可以直接设定指定文件的内容，等价于echo |
| backup | 在覆盖远端服务器之前，将远端服务器源文件备份，备份文件包括时间信息。有两个选项：yes|no |
| force | 如果目标主机包含该文件，但内容不同，如果设置为yes，则强制覆盖，如果为no，这只有当目标主机的目标位置不存在该文件时，才复制。默认为yes |
| directory-mode | 递归设定目录的权限，默认为系统默认权限 |
| others | 所有的file模块里的选项都可以在这里使用 |

#### copy 分发文件时，对源文件备份

ansible oldboy -m copy -a "src=/etc/hosts dest=/opt/" ---src源头 dest 目标地址

ansible oldboy -m copy -a "src=/etc/hosts dest=/opt/ backup=yes" --- 在分发文件前，对已有源文件进行备份

[root@m01 ~]# ansible oldboy -m copy -a "src=/etc/hosts dest=/tmp/ backup=yes"

172.16.1.31 | SUCCESS => {

"changed": true,

"checksum": "545f587595b5c60d66243fca48e052ed34eed782",

"dest": "/tmp/hosts",

"gid": 0,

"group": "root",

"md5sum": "fe08440ffebed54cab7a9b4cb3c3beda",

"mode": "0644",

"owner": "root",

"size": 371,

"src": "/root/.ansible/tmp/ansible-tmp-1504766452.09-24671157820386/source",

"state": "file",

"uid": 0

}

172.16.1.41 | SUCCESS => {

"changed": true,

"checksum": "545f587595b5c60d66243fca48e052ed34eed782",

"dest": "/tmp/hosts",

"gid": 0,

"group": "root",

"md5sum": "fe08440ffebed54cab7a9b4cb3c3beda",

"mode": "0644",

"owner": "root",

"size": 371,

"src": "/root/.ansible/tmp/ansible-tmp-1504766452.12-176546300182774/source",

"state": "file",

"uid": 0

}

172.16.1.8 | SUCCESS => {

"changed": true,

"checksum": "545f587595b5c60d66243fca48e052ed34eed782",

"dest": "/tmp/hosts",

"gid": 0,

"group": "root",

"md5sum": "fe08440ffebed54cab7a9b4cb3c3beda",

"mode": "0644",

"owner": "root",

"size": 371,

"src": "/root/.ansible/tmp/ansible-tmp-1504766452.1-192771818071715/source",

"state": "file",

"uid": 0

}

[root@backup ~]# ll /tmp

total 8

-rw-r--r-- 1 root root 56 Sep 5 00:00 check\_info.txt

-rw-r--r-- 1 root root 371 Sep 7 14:40 hosts

-rw-------. 1 root root 0 Aug 28 14:53 yum.log

#### 分发文件时修改分发复制后的文件属性信息，修改文件权限及所有者

ansible oldboy -m copy -a "src=/etc/hosts dest=/opt/ mode=600 owner=oldboy group=oldboy backup=yes"

## shell 模块，执行脚本，万能模块

#### 执行前需copy 模块分发脚本

[root@m01 ~]# ansible oldboy -m copy -a "src=/server/scripts/shift.sh dest=/server/scripts"

172.16.1.41 | SUCCESS => {

"changed": true,

"checksum": "a87be3180aa837043d2eafd7de09d2778eb8b632",

"dest": "/server/scripts/shift.sh",

"gid": 0,

"group": "root",

"md5sum": "5276faab3517e7760496f0bf8251fe5b",

"mode": "0644",

"owner": "root",

"size": 60,

"src": "/root/.ansible/tmp/ansible-tmp-1504767235.7-5947842560328/source",

"state": "file",

"uid": 0

}

172.16.1.8 | SUCCESS => {

"changed": true,

"checksum": "a87be3180aa837043d2eafd7de09d2778eb8b632",

"dest": "/server/scripts/shift.sh",

"gid": 0,

"group": "root",

"md5sum": "5276faab3517e7760496f0bf8251fe5b",

"mode": "0644",

"owner": "root",

"size": 60,

"src": "/root/.ansible/tmp/ansible-tmp-1504767235.71-132320225246859/source",

"state": "file",

"uid": 0

}

172.16.1.31 | SUCCESS => {

"changed": true,

"checksum": "a87be3180aa837043d2eafd7de09d2778eb8b632",

"dest": "/server/scripts/shift.sh",

"gid": 0,

"group": "root",

"md5sum": "5276faab3517e7760496f0bf8251fe5b",

"mode": "0644",

"owner": "root",

"size": 60,

"src": "/root/.ansible/tmp/ansible-tmp-1504767235.62-219844832669526/source",

"state": "file",

"uid": 0

}

#### 执行脚本

[root@m01 ~]# ansible oldboy -m shell -a "/bin/sh /server/scripts/shift.sh"

172.16.1.8 | SUCCESS | rc=0 >>

172.16.1.31 | SUCCESS | rc=0 >>

172.16.1.41 | SUCCESS | rc=0 >>

#### shell 模块万能功能

[root@m01 ~]# ansible oldboy -m shell -a "ls /server/scripts"

172.16.1.8 | SUCCESS | rc=0 >>

backup.sh

name.sh

shift.sh

172.16.1.41 | SUCCESS | rc=0 >>

hostname.sh

mail.sh

shift.sh

172.16.1.31 | SUCCESS | rc=0 >>

backup.sh

hostname.sh

inotify.sh

shift.sh

[root@m01 ~]# ansible oldboy -m shell -a "date"

172.16.1.31 | SUCCESS | rc=0 >>

Thu Sep 7 15:00:49 CST 2017

172.16.1.8 | SUCCESS | rc=0 >>

Thu Sep 7 15:00:49 CST 2017

172.16.1.41 | SUCCESS | rc=0 >>

Thu Sep 7 15:00:49 CST 2017

## script模块（远程运行脚本模块）

|  |  |
| --- | --- |
| 选项参数 | 选项说明 |
| chdir | 切换到指定路径 |
| creates | 一个文件名，它已经存在时，此步骤将不会运行。 |
| decrypt | 脚本的本地路径 |
| removes | 一个文件名，当它不存在，此步骤将不会运行。 |

ansible oldboy -m script -a "/server/scripts/yum.sh" --- 远程执行本地脚本中的命令

[root@m01 scripts]# ansible oldboy -m script -a "/server/scripts/shift.sh 1 2 3 4 "

172.16.1.41 | SUCCESS => {

"changed": true,

"rc": 0,

"stderr": "Shared connection to 172.16.1.41 closed.\r\n",

"stdout": "1 2 3 4\r\n2 3 4\r\n3 4\r\n4\r\n",

"stdout\_lines": [

"1 2 3 4",

"2 3 4",

"3 4",

"4"

]

}

172.16.1.8 | SUCCESS => {

"changed": true,

"rc": 0,

"stderr": "Shared connection to 172.16.1.8 closed.\r\n",

"stdout": "1 2 3 4\r\n2 3 4\r\n3 4\r\n4\r\n",

"stdout\_lines": [

"1 2 3 4",

"2 3 4",

"3 4",

"4"

]

}

172.16.1.31 | SUCCESS => {

"changed": true,

"rc": 0,

"stderr": "Shared connection to 172.16.1.31 closed.\r\n",

"stdout": "1 2 3 4\r\n2 3 4\r\n3 4\r\n4\r\n",

"stdout\_lines": [

"1 2 3 4",

"2 3 4",

"3 4",

"4"

]

}

## file模块--设置文件属性

|  |  |
| --- | --- |
| 选项参数 | 选项说明 |
| force | 强制创建软链接；可选yes/no需要在两种情况下强制创建软连接，一种是源文件不存在但是之后会建立的情况下；另一种是目标软连接已经存在，需要取消之前的软连接， |
| owner | 设置复制传输后的属主信息 |
| group | 设置复制传输后的数据属组信息 |
| mode | 设置文件数据权限信息 |
| dest | 设置要创建的文件或目录命令，以及路径信息 |
| state | directory创建目录，支持多级目录  file创建文件，如果文件不存在就不会创建  link创建软连接  hard创建硬链接  absent目录或文件将被递归删除不存在将显示绿色  touch修改时间时间戳不存在将创建出来 |
| src | 制定要创建软连接的文件信息 |

#### 创建目录 directory

[root@m01 scripts]# ansible oldboy -m file -a "dest=/tmp/dir\_01 state=directory"

172.16.1.8 | SUCCESS => {

"changed": true,

"gid": 0,

"group": "root",

"mode": "0755",

"owner": "root",

"path": "/tmp/dir\_01",

"size": 4096,

"state": "directory",

"uid": 0

}

172.16.1.31 | SUCCESS => {

"changed": true,

"gid": 0,

"group": "root",

"mode": "0755",

"owner": "root",

"path": "/tmp/dir\_01",

"size": 4096,

"state": "directory",

"uid": 0

}

172.16.1.41 | SUCCESS => {

"changed": true,

"gid": 0,

"group": "root",

"mode": "0755",

"owner": "root",

"path": "/tmp/dir\_01",

"size": 4096,

"state": "directory",

"uid": 0

}

[root@m01 scripts]# ansible oldboy -m shell -a "ls /tmp/"

172.16.1.8 | SUCCESS | rc=0 >>

ansible\_sZ77GM

dir\_01

hosts

yum.log

172.16.1.31 | SUCCESS | rc=0 >>

ansible\_EwsoNQ

crontab.xe4jHS

crontab.yR6S9j

crontab.ZcnFVX

dir\_01

hosts

rsync\_fail\_log.sh

yum.log

172.16.1.41 | SUCCESS | rc=0 >>

ansible\_msuWpo

check\_info.txt

dir\_01

hosts

yum.log

#### 创建文件 touch

[root@m01 scripts]# ansible oldboy -m file -a "dest=/tmp/yy.txt state=touch"

172.16.1.8 | SUCCESS => {

"changed": true,

"dest": "/tmp/yy.txt",

"gid": 0,

"group": "root",

"mode": "0644",

"owner": "root",

"size": 0,

"state": "file",

"uid": 0

}

172.16.1.41 | SUCCESS => {

"changed": true,

"dest": "/tmp/yy.txt",

"gid": 0,

"group": "root",

"mode": "0644",

"owner": "root",

"size": 0,

"state": "file",

"uid": 0

}

172.16.1.31 | SUCCESS => {

"changed": true,

"dest": "/tmp/yy.txt",

"gid": 0,

"group": "root",

"mode": "0644",

"owner": "root",

"size": 0,

"state": "file",

"uid": 0

}

[root@nfs01 ~]# ll /tmp

total 20

-rw------- 1 root root 163 Sep 1 21:46 crontab.xe4jHS

-rw------- 1 root root 163 Sep 1 21:47 crontab.yR6S9j

-rw------- 1 root root 163 Sep 1 21:46 crontab.ZcnFVX

drwxr-xr-x 2 root root 4096 Sep 7 15:20 dir\_01

-rw-r--r-- 1 root root 371 Sep 7 14:40 hosts

-rwxrwxrwx 1 root root 0 Sep 5 22:53 rsync\_fail\_log.sh

-rw-------. 1 root root 0 Aug 28 14:53 yum.log

-rw-r--r-- 1 root root 0 Sep 7 15:32 yy.txt

#### 删除 absent

[root@m01 scripts]# ansible oldboy -m file -a "dest=/tmp/yy.txt state=absent"

172.16.1.41 | SUCCESS => {

"changed": true,

"path": "/tmp/yy.txt",

"state": "absent"

}

172.16.1.8 | SUCCESS => {

"changed": true,

"path": "/tmp/yy.txt",

"state": "absent"

}

172.16.1.31 | SUCCESS => {

"changed": true,

"path": "/tmp/yy.txt",

"state": "absent"

}

[root@m01 scripts]# ansible oldboy -m file -a "dest=/tmp/dir\_01 state=absent"

172.16.1.41 | SUCCESS => {

"changed": true,

"path": "/tmp/dir\_01",

"state": "absent"

}

172.16.1.31 | SUCCESS => {

"changed": true,

"path": "/tmp/dir\_01",

"state": "absent"

}

172.16.1.8 | SUCCESS => {

"changed": true,

"path": "/tmp/dir\_01",

"state": "absent"

}

[root@nfs01 ~]# ll /tmp

total 16

-rw------- 1 root root 163 Sep 1 21:46 crontab.xe4jHS

-rw------- 1 root root 163 Sep 1 21:47 crontab.yR6S9j

-rw------- 1 root root 163 Sep 1 21:46 crontab.ZcnFVX

-rw-r--r-- 1 root root 371 Sep 7 14:40 hosts

-rwxrwxrwx 1 root root 0 Sep 5 22:53 rsync\_fail\_log.sh

-rw-------. 1 root root 0 Aug 28 14:53 yum.log

#### 创建软连接

[root@m01 ~]# ansible test -m file -a "src=/tmp/test.sh dest=/root/test.link state=link "

172.16.1.222 | SUCCESS => {

"changed": true,

"dest": "/root/test.link",

"gid": 0,

"group": "root",

"mode": "0777",

"owner": "root",

"size": 12,

"src": "/tmp/test.sh",

"state": "link",

"uid": 0

}

## fetch-取、拿模块

|  |  |
| --- | --- |
| 选项参数 | 参数信息 |
| dest | 将来远程主机上的文件，拿到本地 |
| src | 指定原文件信息 |
| flat | 默认设置为no如果设置为yes，将不会拉取路径信息 |

#### 从远程主机拉取文件

[root@m01 ~]# ansible test -m fetch -a "src=/tmp/test.sh dest=/tmp"

172.16.1.222 | SUCCESS => {

"changed": true,

"checksum": "d4dfddcc1b1a2b92599a6b498b1dbc0b3dee9128",

"dest": "/tmp/172.16.1.222/tmp/test.sh",

"md5sum": "e87e71d1c410c5863aadc5b9a3fe209b",

"remote\_checksum": "d4dfddcc1b1a2b92599a6b498b1dbc0b3dee9128",

"remote\_md5sum": null

}

[root@m01 ~]# tree /tmp

/tmp

└── 172.16.1.222

└── tmp

└── test.sh

2 directories, 1 file

#### 拉取时仅拉取目标文件或目录

[root@m01 tmp]# ansible test -m fetch -a "src=/tmp/test.sh dest=/tmp/ flat=yes"

172.16.1.222 | SUCCESS => {

"changed": true,

"checksum": "d4dfddcc1b1a2b92599a6b498b1dbc0b3dee9128",

"dest": "/tmp/test.sh",

"md5sum": "e87e71d1c410c5863aadc5b9a3fe209b",

"remote\_checksum": "d4dfddcc1b1a2b92599a6b498b1dbc0b3dee9128",

"remote\_md5sum": null

}

[root@m01 tmp]# tree /tmp

/tmp

└── test.sh

0 directories, 1 file

## mount-挂载相关模块

|  |  |
| --- | --- |
| 参数选项 | 参数信息 |
| fstype | 指定挂载文件类型 |
| opts | 挂载参数 |
| path | 指定挂载点 |
| src | 被挂载的目录设备信息 |
| state | present：在fstab中添加挂载配置  absent：在fstab中删除配置  mounted：1.将挂载信息添加到/etc/fstab 2.加载/etc/fstab实现挂载  unmounted：加载/etc/fstab实现挂载 |

## cron模块

|  |  |
| --- | --- |
| 参数名称 | 参数信息 |
| minute | 分钟 |
| hour | 小时 |
| day | 天 |
| month | 月 |
| weekday | 周信息，0表示周日 |
| name | 定义定时任务的描述信息 |
| disabled | 可选yes/no用于注释定时任务 |
| job | 要执行的任务依赖于state=present |
| state | absent表示删除定时任务present表示创建定时任务 |
| backup | 对远程主机修改前备份 |
| cron\_file | 如果指定该选项，则用该文件替换远程主机上的cron.d目录下的任务计划 |
| env | 如果设置，管理crontab的环境变量信息。新的环境变量信息会被增加到定时任务表顶部。描述或数值会被环境变得的名称或数值所定义 |
| reboot | 在重新启动后，运行作业。不建议使用此选项。用户应该使用 special\_time。 |
| special\_time | 特殊的时间格式 reboot，yearly，annually，monthly，weekly，daily，hourly |
| user | 指定以哪个用户的身份运行 |

#### 定时任务参数

\* \* \* \* \* /bin/sh /server/scripts/test.sh &>/dev/null

minute # Minute when the job should run ( 0-59, \*, \*/2, etc )

hour # Hour when the job should run ( 0-23, \*, \*/2, etc )

day # Day of the month the job should run ( 1-31, \*, \*/2, etc )

month # Month of the year the job should run ( 1-12, \*, \*/2, etc )

weekday # Day of the week that the job should run ( 0-6 for Sunday-Saturday, \*, etc )

job # The command to execute or, if env is set, the value of environment variable. Required if

state=present.

#### 定时任务创建

[root@m01 scripts]# ansible oldboy -m cron -a "name=oldboy01 minute=\*/5 job='/usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1'"

172.16.1.31 | SUCCESS => {

"changed": true,

"envs": [],

"jobs": [

"oldboy01"

]

}

172.16.1.8 | SUCCESS => {

"changed": true,

"envs": [],

"jobs": [

"oldboy01"

]

}

172.16.1.41 | SUCCESS => {

"changed": true,

"envs": [],

"jobs": [

"oldboy01"

]

}

[root@m01 scripts]# ansible oldboy -m shell -a "crontab -l"

172.16.1.41 | SUCCESS | rc=0 >>

#crond-id-001:time sync by oldboy

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

#mail

00 00 \* \* \* /bin/sh /server/scripts/mail.sh &>/dev/null

#Ansible: oldboy01

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

172.16.1.8 | SUCCESS | rc=0 >>

#crond-id-001:time sync by oldboy

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

#backup

#00 00 \* \* \* /bin/sh /server/scripts/backup.sh &>/dev/null

#Ansible: oldboy01

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

172.16.1.31 | SUCCESS | rc=0 >>

#crond-id-001:time sync by oldboy

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

#backup

00 00 \* \* \* /bin/sh /server/scripts/backup.sh &>/dev/null

#Ansible: oldboy01

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

#### 删除定时任务

[root@m01 scripts]# ansible oldboy -m cron -a "name=oldboy01 minute=\*/5 job='/usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1' state=absent"

172.16.1.31 | SUCCESS => {

"changed": true,

"envs": [],

"jobs": []

}

172.16.1.8 | SUCCESS => {

"changed": true,

"envs": [],

"jobs": []

}

172.16.1.41 | SUCCESS => {

"changed": true,

"envs": [],

"jobs": []

}

[root@m01 scripts]# ansible oldboy -m shell -a "crontab -l"

172.16.1.8 | SUCCESS | rc=0 >>

#crond-id-001:time sync by oldboy

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

#backup

#00 00 \* \* \* /bin/sh /server/scripts/backup.sh &>/dev/null

172.16.1.31 | SUCCESS | rc=0 >>

#crond-id-001:time sync by oldboy

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

#backup

00 00 \* \* \* /bin/sh /server/scripts/backup.sh &>/dev/null

172.16.1.41 | SUCCESS | rc=0 >>

#crond-id-001:time sync by oldboy

\*/5 \* \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

#mail

00 00 \* \* \* /bin/sh /server/scripts/mail.sh &>/dev/null

#### 注释定时任务信息

ansible oldboy -m cron -a "name=oldboy01 minute=\*/5 job='/usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1' disabled=yes"

#### 如果利用ansible进行批量管理设置定时任务，如果和原有手工设置定时任务一致，会被ansible定时任务覆盖掉

172.16.1.31 | SUCCESS | rc=0 >>

# sync time by lidao 2019

\*/5 \* \* \* \* /usr/sbin/ntpdate ntp1.aliyun.com >/dev/null 2>&1

ansible oldboy -m cron -a "name=oldboy02 minute=\*/5 job='/usr/sbin/ntpdate ntp1.aliyun.com >/dev/null 2>&1'"

#sync time by lidao 2019

#Ansible: oldboy02

\*/5 \* \* \* \* /usr/sbin/ntpdate ntp1.aliyun.com >/dev/null 2>&1

## yum--yum软件模块

|  |  |
| --- | --- |
| 参数名称 | 参数信息 |
| config | yum的配置文件 |
| disablerepo | 不启用某个源 |
| enablerepo | 启用某个源 |
| name | 要安装的软件名 |
| state | 是否安装 （'present' 或 'installed'，' latest'），或删除 （absent或`removed） 。 |

ansible oldboy -m yum -a "name=nmap state=installed"

## service--服务启动或管理模块

|  |  |
| --- | --- |
| 参数名称 | 参数信息 |
| arguments | 给命令行提供一些额外的参数 |
| enabled | 是否开机自启动 |
| name= | 服务的名称 |
| pattern | 定义一个状态如果status指令来查看状态时没有结果就会通过ps指令查找，如果找到，认为该服务执行 |
| runlevel | 运行级别 |
| sleep | 如果重启该服务，将会在这之间沉睡数秒，有助于解决stop结束时间稍长的问题 |
| state | 执行启动、停止、重启、重新加载等操作（started、stopped、restarted、reloaded） |
| use | 服务模块实际使用系统特定的模块，通常通过自动检测，此设置可以强制  一个特定的模块。它通常使用 'ansible\_service\_mgr' 事实的价值  和回退到旧的服务模块均不匹配时被发现。 |

## filesystem-在块设备上创建文件系统

|  |  |
| --- | --- |
| 参数名称 | 参数信息 |
| dev= | 目标块设备 |
| force | 在一个已有文件系统设备上强制创建 |
| fstype= | 文件系统的类型 |
| opts | 传递给mkfs命令的选项 |
| resizefs | 如果是的如果块设备和 filessytem 大小不同，长成空间的文件系统。请注意，XFS 只会  如果安装，生长。 |

## user-管理用户模块

|  |  |
| --- | --- |
| 参数名称 | 参数信息 |
| home | 指定用户家目录 |
| group | 指定用户组需要用户名 |
| groups | 指定用户组列表 |
| uid | 指定uid，当设置为空字符串 (' 组 =')，从所有组中删除用户除了主要的组。唯一允许的输入的格式是'以逗号分隔的字符串'，现在它应该能够接受 YAML 在2.3版本。 |
| password | password后面指定的密码不能是明文，后面这一串密码会被直接传送到被管理主机的/etc/shadow文件中，而登陆的时候输入的密码会被hash加密以后再去与/etc/shadow中存放的密码去做对比，会出现不一致的现象。所以需要先将密码字符串进行加密处理：openssl passwd -salt -1 "123456"，然后将得到的字符串放到password中即可。 |
| name= | 要创建，删除，修改的用户名 |
| createhome | 是否创建家目录 |
| system | 创建用户时使其成为系统用户，不能应用于现有用户 |
| state | ` present相当于useradd  state=absent',他相当于 userdel |
| shell | 指定命令解释器 |

## synchronize-使用rsync同步软件

|  |  |
| --- | --- |
| 参数名称 | 参数信息 |
| archiv | 是否采用归档模式同步，即以源文件相同属性同步到目标地址 |
| checksum | 是否校验 |
| compress | 开启压缩。默认为压缩 |
| copy\_links | 同步的时候是否复制链接 |
| delete | 增量同步 |
| dest= | 目标地址 |
| dest\_port | 目标接受的的端口优先考虑ansible中的hosts |
| mode | 模式push为推pull为拉 |
| recursive | 是否递归yes/no |
| rsync\_timeout | 指定rsync的超时时间 |
| src | 源路径 |
| dest | 目标路径 |

## ansible 补充知识

### ansible命令输出信息中：

绿色表示查询，或者没有发生任何改变时

红色表示执行命令操作出现异常

黄色表示执行命令后，对受控主机产生了影响，发生了配置改变

### ansible帮助信息系统中查看方法

ansible-doc -l

ansible-doc -s file

### ansible基础知识部分补充

#### ansible软件特点：

· 可以实现批量管理

· 可以实现批量部署

· ad-hoc(批量执行命令)---针对临时性的操作

ansible oldboy -m command -a "hostname" <- 批量执行命令举例

· 编写剧本-脚本(playbook)---针对重复性的操作

#### ansible核心功能：

· pyYAML-----用于ansible编写剧本所使用的语言格式（saltstack---python）

· paramiko---远程连接与数据传输

· Jinja2-----用于编写ansible的模板信息（扩展部分）

# ansible-playbook

## 编写ansible-playbook

01. 需要识别空格字符

02. tab==1个空格 编写ansible-playbook，一定要忘记有tab

03. - : 在剧本中使用时，后面要接一个空格

04. 剧本编写有等级划分，用两个空格表示不同等级

## nsible-playbook编写格式

# 表示注释信息

### 剧本的开头，可以不写

- hosts:空格all

### 处理所有服务器，找到所有服务器

tasks:

- command: echo hello oldboy linux.

### 编写剧本

#### 创建剧本存放目录

mkdir -p /etc/ansible/ansible-playbook/

#### 编写剧本，后缀用yml

[root@m01 scripts]# vim /etc/ansible/ansible-playbook/juben.yml

- hosts: oldboy

tasks:

- shell: "ls /data"

### 剧本检查语法方法

ansible-playbook --syntax-check

[root@m01 scripts]# ansible-playbook --syntax-check /etc/ansible/ansible-playbook/juben.yml

playbook: /etc/ansible/ansible-playbook/juben.yml

### 模拟执行剧本

ansible-playbook -C

[root@m01 scripts]# ansible-playbook -C /etc/ansible/ansible-playbook/juben.yml

PLAY [oldboy] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.16.1.8]

ok: [172.16.1.41]

ok: [172.16.1.31]

TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

skipping: [172.16.1.41]

skipping: [172.16.1.8]

skipping: [172.16.1.31]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.16.1.31 : ok=1 changed=0 unreachable=0 failed=0

172.16.1.41 : ok=1 changed=0 unreachable=0 failed=0

172.16.1.8 : ok=1 changed=0 unreachable=0 failed=0

### 执行剧本

ansible-playbook cron.yml

[root@m01 scripts]# ansible-playbook /etc/ansible/ansible-playbook/juben.yml

PLAY [oldboy] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.16.1.41]

ok: [172.16.1.8]

ok: [172.16.1.31]

TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.16.1.8]

changed: [172.16.1.31]

changed: [172.16.1.41]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.16.1.31 : ok=2 changed=1 unreachable=0 failed=0

172.16.1.41 : ok=2 changed=1 unreachable=0 failed=0

172.16.1.8 : ok=2 changed=1 unreachable=0 failed=0

### 查看剧本

[root@m01 scripts]# cat /etc/ansible/ansible-playbook/juben.yml

- hosts: oldboy

tasks:

- shell: "ls /data"

### 执行多任务剧本

cat cron.yml

- hosts: oldboy

tasks:

- cron: name=oldboy03 minute=00 hour=03 job='/bin/sh /server/scripts/test.sh &>/dev/null'

- hosts: all

tasks:

- name: cron04:oldboy04 00 03

cron: name=oldboy04 minute=00 hour=03 job='/bin/sh /server/scripts/test.sh &>/dev/null' state=absent

- name: cron-oldboy05 00 04

cron: name=oldboy05 minute=00 hour=04 job='/bin/sh /server/scripts/test.sh &>/dev/null' state=absent

### 根据不同主机或主机组，完成不同的ansible任务

[root@m01 ansible-playbook]# cat cron.yml

- hosts: oldboy

tasks:

- name: cron04:oldboy04 00 03

cron: name=oldboy04 minute=00 hour=03 job='/bin/sh /server/scripts/test.sh &>/dev/null'

- name: cron-oldboy05 00 04

cron: name=oldboy05 minute=00 hour=04 job='/bin/sh /server/scripts/test.sh &>/dev/null'

- hosts: web01

tasks:

- name: stop cron server

service: name=crond state=stopped enabled=no

### ansible的hosts文件进行如下配置，可以实现基于口令方式远程管理主机

[oldboy]

172.16.1.31 ansible\_ssh\_user=root ansible\_ssh\_pass=123456

172.16.1.41 ansible\_ssh\_user=root ansible\_ssh\_pass=123456

172.16.1.8 ansible\_ssh\_user=root ansible\_ssh\_pass=123456

## ansible编写剧本拍错思路：

1. ansible-playbook编写完，检查语法和模拟测试运行

2. 打开剧本，定位异常问题原因，将剧本中的内容转换命令执行一次

cron: name=oldboy04 minute=00 hour=03 job='/bin/sh /server/scripts/test.sh &>/dev/null'

ansible oldboy -m cron -a "name=oldboy04 minute=00 hour=03 job='/bin/sh /server/scripts/test.sh &>/dev/null'"

3. 将参数中的脚本文件推送到远程服务器，在远程服务器本地执行脚本

sh -x test.sh

说明：ansible执行时,加上-vvvv显示ansible详细执行过程，也可以定位异常原因

运维领域：

1. 越安全，效率越低

2. 效率越高，安全性越低