Ansible是一种自动化工具，用于自动化管理和配置计算机系统、网络设备和应用程序的部署、配置和编排。它是一个开源软件，广泛用于自动化任务执行和配置管理。

使用Ansible可以通过SSH协议远程管理多个目标主机，而无需在目标主机上安装任何客户端代理。Ansible使用基于Python的简单和易于理解的语法来描述和执行任务，使得编写和维护自动化脚本变得简单和高效。

确保系统更新：

yum update

安装epel-release仓库：在CentOS系统上，为了安装Ansible，需要添加epel-release仓库，

yum install -y epel-release

安装Ansible

yum install -y ansible

验证安装

ansible --version

配置SSH密钥认证：Ansible使用SSH协议与目标主机进行通信。确保你的控制机已经生成了SSH密钥对，并将公钥复制到目标主机上的~/.ssh/authorized\_keys文件中。

ssh-keygen -t rsa

ssh-copy-id root@192.168.106.131

编写 Ansible 配置文件 Ansible 的配置文件默认为 /etc/ansible/ansible.cfg，可以使用以下命令创建一个新的配置文件： vim /etc/ansible/myansible.cfg 在配置文件中可以设置一些常用的配置项，例如：

[defaults]

inventory = /etc/ansible/hosts

remote\_user = root

private\_key\_file = /path/to/private/key

解析：[defaults]：这是配置文件的一个部分，用于定义全局默认设置。

inventory = /etc/ansible/hosts：这个配置项指定了Ansible Inventory文件的路径。在示例中，Inventory文件被设置为/etc/ansible/hosts。Ansible将在该文件中查找要管理的主机列表。

remote\_user = user：这个配置项指定了远程连接到目标主机时使用的用户名。在示例中，用户名被设置为user，你可以根据实际情况更改为合适的用户名。

private\_key\_file = /path/to/private/key：这个配置项指定了连接到目标主机时使用的私钥文件的路径。在示例中，私钥文件的路径被设置为/path/to/private/key，你需要将其替换为实际的私钥文件路径。

.创建 Ansible inventory 文件 Ansible inventory 文件用于存储主机列表，可以使用以下命令创建一个新的 inventory 文件： vi /etc/ansible/inventory 在 inventory 文件中可以指定主机的 IP 地址或主机名，例如：

[webservers]

web1 ansible\_host=192.168.106.131

[dbservers]

db1 ansible\_host=192.168.106.131

测试：编写 Ansible playbook Ansible playbook 是一个 YAML 格式的文件，用于描述 Ansible 的执行过程。可以使用以下命令创建一个新的 playbook： vim /etc/ansible/playbook.yml 在 playbook 文件中可以指定要执行的任务

- name: Install Apache

become: true

hosts: webservers

tasks:

- name: Install Apache

yum:

name: httpd

state: latest

在操作主机上安装最新版本的 Apache。

**使用Ansible给两台虚拟机部署vsftp**

1. 创建一个Ansible Playbook文件来部署VSFTP，

[root@master ansible]# cat vsftpd\_playbook.yml

- name: Deploy VSFTP on target servers with Ansible installed

hosts: webservers

gather\_facts: true

become: true

tasks:

- name: Install VSFTP package

package:

name: vsftpd

state: present

when: ansible\_os\_family == 'RedHat' or ansible\_os\_family == 'Debian'

- name: Ensure VSFTP service is running and enabled

service:

name: vsftpd

state: started

enabled: yes

- name: Configure VSFTP

template:

src: vsftpd.conf.j2

dest: /etc/vsftpd.conf

notify: restart vsftpd

handlers:

- name: restart vsftpd

service:

name: vsftpd

state: restarted

1. 创建一个模板文件，其中包含VSFTP的配置，
2. 使用ansible-playbook命令启动Ansible Playbook文件部署，启动并启用VSFTP服务。

130的虚拟机ftp客户端连接131虚拟机的ftp服务端进行文件传输

IPv4 ipv6 不能同时listen.

编辑/etc/vsftpd/vsftpd.conf，发现有以下两项：

userlist\_enable=YES

userlist\_deny=NO

userlist\_enable=YES 表示启用user\_list，里面的用户禁止登录ftp（默认是禁止）

userlist\_deny=NO 表示拒绝user\_list里面的用户登录，设置为NO，那就是不拒绝

这两条加起来就等于：只允许user\_list文件中的用户登录ftp服务器

而user\_list文件里面并没有包括匿名用户，所以自然访问不了ftp服务器，产生拒绝访问报错

登录匿名FTP服务器时，使用一个特殊的用户名“anonymous”

原因找到了，于是在配置文件/etc/vsftpd/vsftpd.conf进行以下操作：

userlist\_enable=NO

#userlist\_deny=NO

另外，在配置文件/etc/vsftpd/vsftpd.conf中还得加上允许匿名用户登录、上传、下载等权限的操作：

anonymous\_enable=YES

anon\_upload\_enable=YES

anon\_mkdir\_write\_enable=YES

anon\_other\_write\_enable=YES

write\_enable=YES：启用任何形式的FTP写入命令，允许FTP用户上传文件到服务器。

userlist\_file=/etc/vsftpd.user\_list、userlist\_enable=YES、userlist\_deny=NO：这些选项用于配置用户列表文件，userlist\_file指定了存储允许或拒绝访问的用户名的文件路径，userlist\_enable启用了用户列表功能，userlist\_deny=NO表示默认情况下允许列出的用户访问FTP服务器。

local\_umask=022：设置新建文件的默认权限掩码为022，这意味着文件默认的权限为644，目录默认的权限为755。

chroot\_local\_user=YES：将FTP用户限制在其主目录下，增强安全性，防止用户访问系统其他目录。

pasv\_enable=YES、pasv\_min\_port=40000、pasv\_max\_port=50000：启用被动模式，并设置被动模式数据端口的范围为40000至50000，这是为了让FTP服务器支持被动模式，并且指定了被动模式数据传输的端口范围。

allow\_writeable\_chroot=YES：允许用户在chroot环境中创建或修改文件，即使他们被限制在自己的家目录中。

local\_root=/home/ftp：指定FTP用户的根目录，即限制用户的家目录位置。

listen=YES、listen\_ipv6=NO：启用IPv4的监听，禁用IPv6的监听。

pasv\_address=192.168.1.140：配置被动模式时，需要指定FTP服务器的外部IP地址，以便客户端能够连接到正确的IP地址。

xferlog\_enable=YES、log\_ftp\_protocol=YES：启用传输日志，并记录FTP协议的日志。

# **vsftpd.conf参数配置解析（ftp配置、ftp参数）**

参考网址：<https://blog.csdn.net/Dontla/article/details/130427094>

# **安装vsftpd[虚拟用户]及ftp相关命令**

参考网址：<https://zhuanlan.zhihu.com/p/610785204>

**使用Ansible给两台虚拟机部署tomcat**

先在虚拟机部署 Supervisor

Supervisor (http://supervisord.org) 是一个用Python写Linux下的进程集中管理工具，将非daemon程序变成deamon方式运行，对于daemon程序则不能监控。可以很方便地对管理的进程进行启动、重启、关闭操作（不仅仅是Python进程）。除了对单个进程的控制，还可以同时启动、关闭多个进程，比如很不幸的服务器出问题导致所有应用程序都被杀死，此时可以用supervisor同时启动所有应用程序而不是一个一个地敲命令启动。

**安装java-1.8.0-openjdk-devel，tomcat-9.0.10**

**1.主机清单文件（inventory.ini）包含了一个主机（IP地址为192.168.106.131），即将在该主机上部署Tomcat**

[root@master ansible]# cat inventory.ini

[tomcat\_servers]

192.168.106.131

**2.创建一个Ansible Playbook文件来部署 tomcat**

[root@master ansible]# cat de\_tomcat.yml

---

- name: Deploy Tomcat

hosts: tomcat\_servers

gather\_facts: explicit

#可以将 "gathering" 参数设置为 "explicit"，以确保 Ansible 在运行任务时显式地收集事实（facts），而不是使用默认的智能收集方式。这样做可以提高执行效率，尤其是在拥有大量目标主机的情况下

become: true

tasks:

- name: Ensure Java 8 is installed # 检验Java 8

yum:

name: java-1.8.0-openjdk-devel

state: present

- name: Download Tomcat # 下载tomcat压缩包到 /tmp目录下

get\_url:

url: "https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.10/bin/apache-tomcat-9.0.10.tar.gz"

dest: /tmp/apache-tomcat-9.0.10.tar.gz

- name: Extract Tomcat # 解压安装到 /opt/ 目录下

unarchive:

src: /tmp/apache-tomcat-9.0.10.tar.gz

dest: /opt/

remote\_src: yes

- name: Set Tomcat environment variables # 设置 tomcat 的环境变量

template:

src: tomcat.sh.j2

dest: /etc/profile.d/tomcat.sh

line: 'export CATALINA\_HOME=/opt/apache-tomcat-9.0.10'

- name: Start Tomcat # 启动 tomcat

**shell: "nohup /opt/apache-tomcat-9.0.10/bin/startup.sh &"**

**nohup 命令运行由 Command参数和任何相关的 Arg参数指定的命令，忽略所有挂断（SIGHUP）信号。在注销后使用 nohup 命令运行后台中的程序。要运行后台中的 nohup 命令，添加 & （ 表示“and”的符号）到命令的尾**

**3.创建模板文件（tomcat.sh.j2）定义了Tomcat的环境变量，导出了CATALINA\_HOME和将$CATALINA\_HOME/bin添加到PATH环境变量**

**[root@master ansible]# cat tomcat.sh.j2**

**export CATALINA\_HOME=/opt/apache-tomcat-10.1.13-deployer**

**export PATH=$PATH:$CATALINA\_HOME/bin**

**启动： ansible-playbook -i inventory.ini qd\_tomcat.yml**