原创

Centos7-Ansible-Nginx自动化部署

GIVE_ME 关注

2018-05-24 10:20:57 20780人阅读 0人评论

Ansible自动化批量部署nginx服务器

搭建环境：

主机： Ansible Web1 Web2
系统： Centos7 64Bit
网卡： Vmnet0桥接
IP： Ansible-152.158 Web1-152.159 Web2-152.160
注：清空并关闭以上环境所有主机的防火墙和selinux

关闭防火墙 systemctl stop firewalld.service
禁止防火墙开机自启 systemctl disable firewalld.service

关闭selinux sed -i 's/SELINUX=enforcing /SELINUX=disabled/g' /etc/sysconfig/selinux

重启 reboot
注：重启后如果可以互相ping通，那么就开始搭建ansible服务器

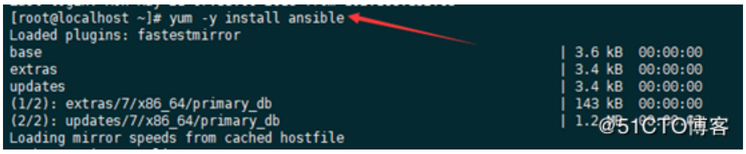
一、Ansible-server安装

安装方式：
1、从Ansible项目的GitHub源码库提取出来安装，运行Ansible不需root 权限，也不依赖于其他软件，没有后台进程运行，不需要数据库支撑。

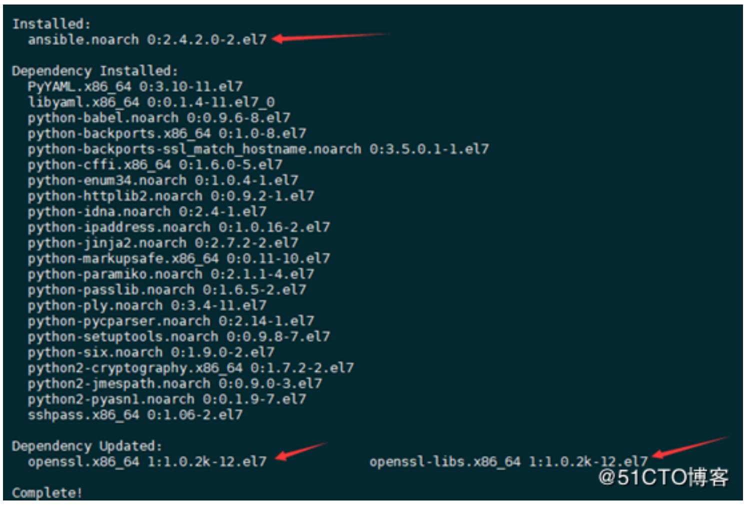
2、使用yum安装，需要有合适的yum源，对于RHEL、CentOS的官方yum源中没有 Ansible安装包，这就需要先安装支持第三方的yum仓库组件，最常用的有EPEL、 Remi、RPMForge等。可国内速度较快的高质量yum源网易 163(http://mirrors.163.com)、阿里源（https://opsx.alibaba.com/mirror）
注：这里实验使用的是默认的centos7自带的源，并使用yum直接安装

二、使用yum安装ansible

1、安装ansible yum –y install ansible

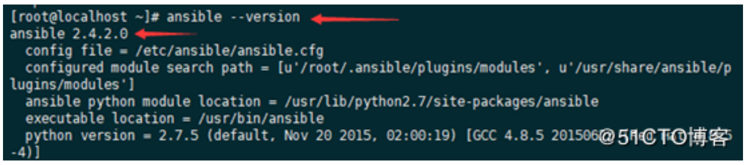


```
[root@localhost ~]# yum -y install ansible
Loaded plugins: fastestmirror
base                                                    | 3.6 kB  00:00:00
extras                                                  | 3.4 kB  00:00:00
updates                                                 | 3.4 kB  00:00:00
(1/2): extras/7/x86_64/primary_db                       | 143 kB  00:00:00
(2/2): updates/7/x86_64/primary_db                      | 1.2
Loading mirror speeds from cached hostfile
```



```
Installed:
  ansible.noarch 0:2.4.2.0-2.el7

Dependency Installed:
  PyYAML.x86_64 0:3.10-11.el7
  libyaml.x86_64 0:0.1.4-11.el7_0
  python-babel.noarch 0:0.9.6-8.el7
  python-backports.x86_64 0:1.0-8.el7
  python-backports-ssl_match_hostname.noarch 0:3.5.0.1-1.el7
  python-cffi.x86_64 0:1.6.0-5.el7
  python-enum34.noarch 0:1.0.4-1.el7
  python-httplib2.noarch 0:0.9.2-1.el7
  python-idna.noarch 0:2.4-1.el7
  python-ipaddress.noarch 0:1.0.16-2.el7
  python-jinja2.noarch 0:2.7.2-2.el7
  python-markupsafe.x86_64 0:0.11-10.el7
  python-paramiko.noarch 0:2.1.1-4.el7
  python-passlib.noarch 0:1.6.5-2.el7
  python-ply.noarch 0:3.4-11.el7
  python-pycparser.noarch 0:2.14-1.el7
  python-setuptools.noarch 0:0.9.8-7.el7
  python-six.noarch 0:1.9.0-2.el7
  python2-cryptography.x86_64 0:1.7.2-2.el7
  python2-jmespath.noarch 0:0.9.0-3.el7
  python2-pyasn1.noarch 0:0.1.9-7.el7
  sshpass.x86_64 0:1.06-2.el7

Dependency Updated:
  openssl.x86_64 1:1.0.2k-12.el7            openssl-libs.x86_64 1:1.0.2k-12.el7

Complete!
```

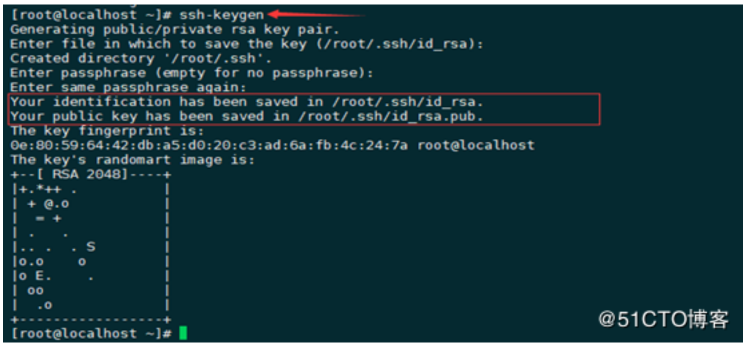
2、检查ansible版本： ansible –version



```
[root@localhost ~]# ansible --version
ansible 2.4.2.0
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/p
lugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.5 (default, Nov 20 2015, 02:00:19) [GCC 4.8.5 20150623 (Red Hat 4.8.5
-4)]
```

三、设置节点授权的ssh密钥

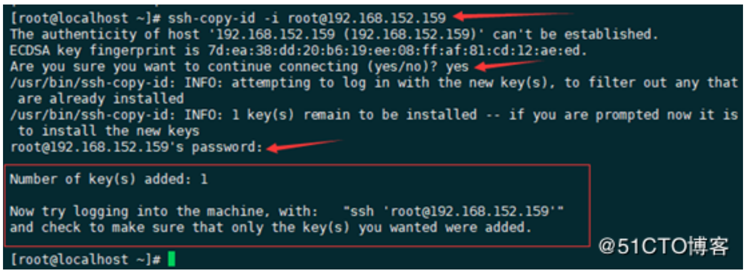
1、在Ansible服务端生成密钥 ssh-keygen



```
[root@localhost ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
0e:80:59:64:42:db:a5:d0:20:c3:ad:6a:fb:4c:24:7a root@localhost
The key's randomart image is:
+--[ RSA 2048]----+
|+.*++ .          |
| + @.o           |
|  = +            |
| .   .           |
|.. .   . S       |
|o.o     o        |
|o E.     .       |
| oo              |
|  .o             |
+-----------------+
[root@localhost ~]# 
```

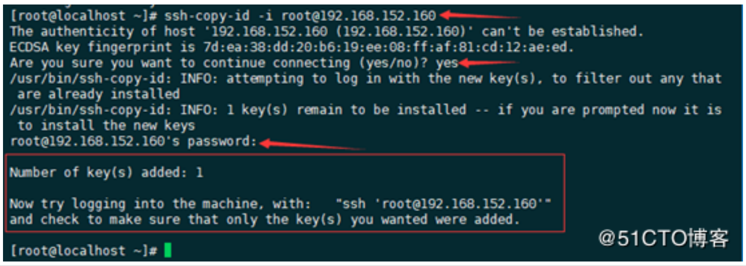
2、使用ssh-copy-id命令来复制Ansible公钥到节点web1和web2

1）复制Ansible公钥到节点web1 ssh-copy-id -i root@192.168.152.159



```
[root@localhost ~]# ssh-copy-id -i root@192.168.152.159
The authenticity of host '192.168.152.159 (192.168.152.159)' can't be established.
ECDSA key fingerprint is 7d:ea:38:dd:20:b6:19:ee:08:ff:af:81:cd:12:ae:ed.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that
 are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is
 to install the new keys
root@192.168.152.159's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@192.168.152.159'"
and check to make sure that only the key(s) you wanted were added.

[root@localhost ~]# 
```

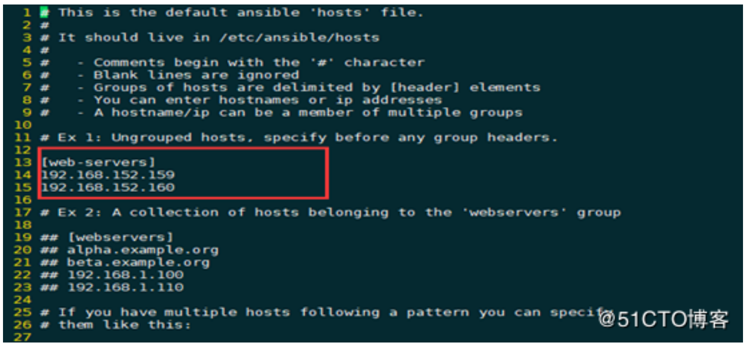
2）复制Ansible公钥到节点web2 ssh-copy-id -i root@192.168.152.160



```
[root@localhost ~]# ssh-copy-id -i root@192.168.152.160
The authenticity of host '192.168.152.160 (192.168.152.160)' can't be established.
ECDSA key fingerprint is 7d:ea:38:dd:20:b6:19:ee:08:ff:af:81:cd:12:ae:ed.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that
 are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is
 to install the new keys
root@192.168.152.160's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@192.168.152.160'"
and check to make sure that only the key(s) you wanted were added.

[root@localhost ~]# 
```

四、配置Ansible定义文件

1、编辑ansible配置文件 vi /etc/ansible/hosts

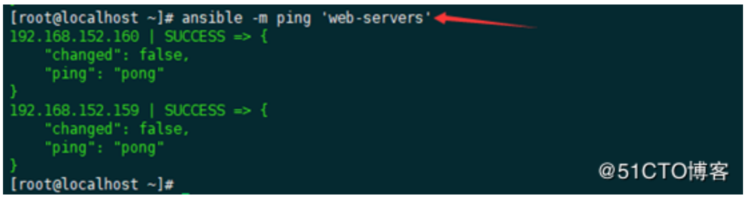


```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

[web-servers]
192.168.152.159
192.168.152.160

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can speci
# them like this:

```

注：将需要ansible自动化的节点IP添加到这里

2、测试在ansible服务端运行命令（在互相能ping通的情况下）

ansible -m ping 'web-servers'

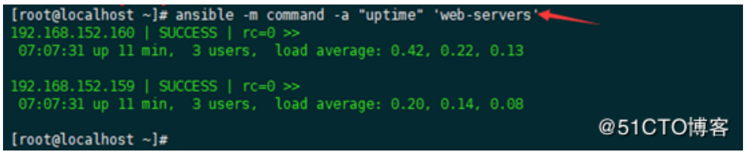


```
[root@localhost ~]# ansible -m ping 'web-servers'
192.168.152.160 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
192.168.152.159 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[root@localhost ~]#
```

五、执行shell命令

1）查看ansible节点运行时间（uptime）

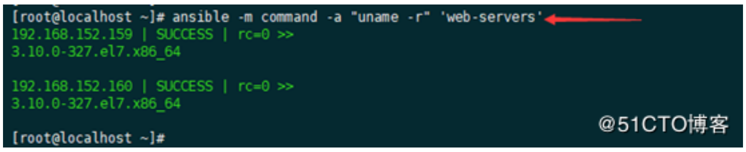
ansible -m command -a "uptime" 'web-servers'



```
[root@localhost ~]# ansible -m command -a "uptime" 'web-servers'
192.168.152.160 | SUCCESS | rc=0 >>
 07:07:31 up 11 min,  3 users,  load average: 0.42, 0.22, 0.13

192.168.152.159 | SUCCESS | rc=0 >>
 07:07:31 up 11 min,  3 users,  load average: 0.20, 0.14, 0.08

[root@localhost ~]#
```

2）查看节点内核版本（uname -r）

ansible -m command -a "uname -r" 'web-servers'



```
[root@localhost ~]# ansible -m command -a "uname -r" 'web-servers'
192.168.152.159 | SUCCESS | rc=0 >>
3.10.0-327.el7.x86_64

192.168.152.160 | SUCCESS | rc=0 >>
3.10.0-327.el7.x86_64

[root@localhost ~]#
```

注：以上操作部署已完成ansible服务搭建

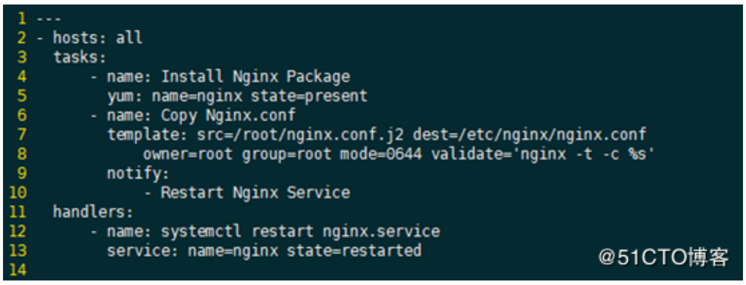
六、批量部署nginx服务器

两种Ansible批量部署nginx服务器方式

方式一：yum安装nginx，使用的是epel-release源

方式二：使用nginx.tar压缩包解压安装nginx

1、在/root/目录下创建Ansible YAML文件 vi nginx.yaml



```
1 ---
2 - hosts: all
3   tasks:
4       - name: Install Nginx Package
5         yum: name=nginx state=present
6       - name: Copy Nginx.conf
7         template: src=/root/nginx.conf.j2 dest=/etc/nginx/nginx.conf
8             owner=root group=root mode=0644 validate='nginx -t -c %s'
9         notify:
10            - Restart Nginx Service
11  handlers:
12      - name: systemctl restart nginx.service
13        service: name=nginx state=restarted
14
```

注释：

第1行表示该文件是YAML文件，非必须

第2行定义该playbook针对的目标主机，all表示针对所有主机

第3行定义该playbook所有的tasks集合，比如下面我们定义的3个task

第4行定义一个task的名称，非必须，建议根据task实际任务命名

第5行定义一个状态的action，比如这里使用yum模块实现Nginx软件包的安装

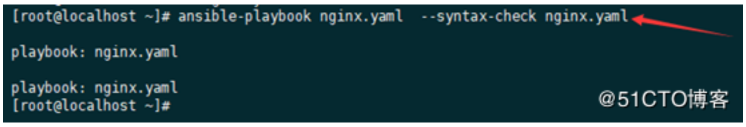
第6行到第9行使用template模板去管理/etc/nginx/nginx.conf文件，owner group定义该文件的属主以及属组，使用validate参数指文件生成后使用nginx -t -c %s命令去做Nginx文件语法验证，notify是触发handler状态，如果同步后，文件 的MD5值有变化会触发ReStart Nginx Service这个handler

第10行到第12行是定义一个handler状态让Nginx服务重启，handler的名称是 ReStart Nginx Service

注：书写yaml文件时，注意左对齐，同级别应在同一列下，并且不能使用Tab键，可以使用空格（随便空格几个都行，但是同一级别必须对齐）

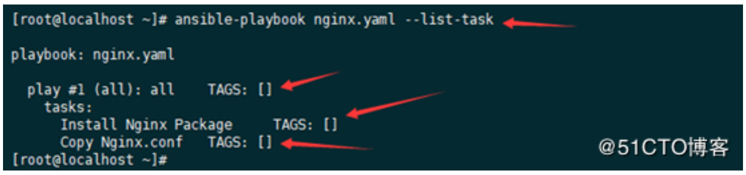
2、检测YAML文件

ansible-playbook nginx.yaml --syntax-check nginx.yaml



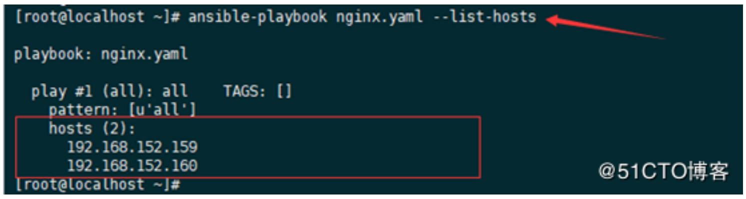
```
[root@localhost ~]# ansible-playbook nginx.yaml  --syntax-check nginx.yaml

playbook: nginx.yaml

playbook: nginx.yaml
[root@localhost ~]#
```

3、查看YAML文件任务列表 ansible-playbook nginx.yaml --list-task



```
[root@localhost ~]# ansible-playbook nginx.yaml --list-task

playbook: nginx.yaml

  play #1 (all): all    TAGS: []
    tasks:
      Install Nginx Package     TAGS: []
      Copy Nginx.conf   TAGS: []
[root@localhost ~]#
```

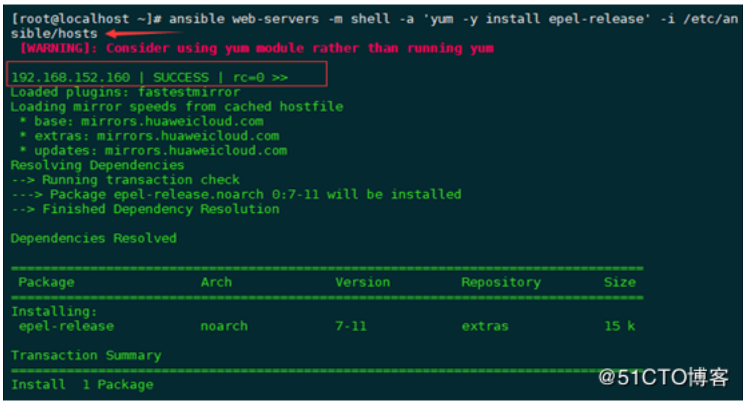
4、查看针对哪些主机做操作 ansible-playbook nginx.yaml --list-hosts



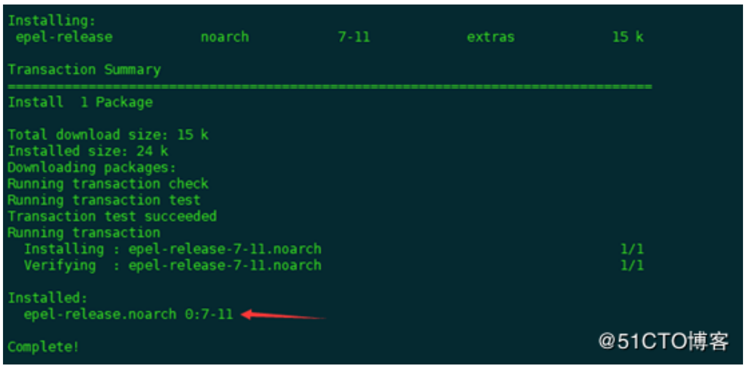
```
[root@localhost ~]# ansible-playbook nginx.yaml --list-hosts

playbook: nginx.yaml

  play #1 (all): all    TAGS: []
    pattern: [u'all']
    hosts (2):
      192.168.152.159
      192.168.152.160
[root@localhost ~]#
```

5、给两个节点安装epel-release源

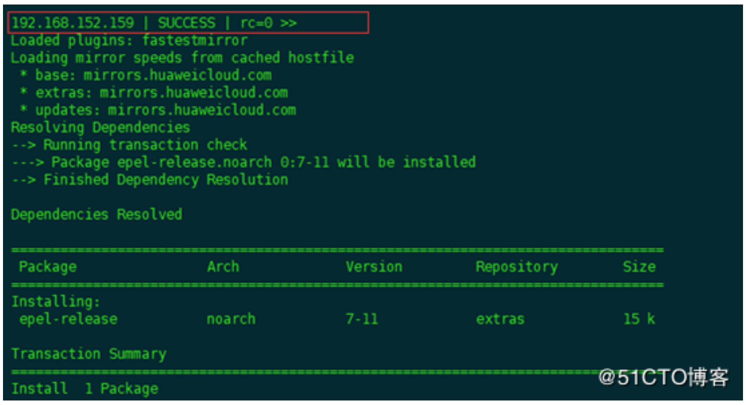
ansible web-servers -m shell -a 'yum -y install epel-release' -i /etc/ansible/hosts



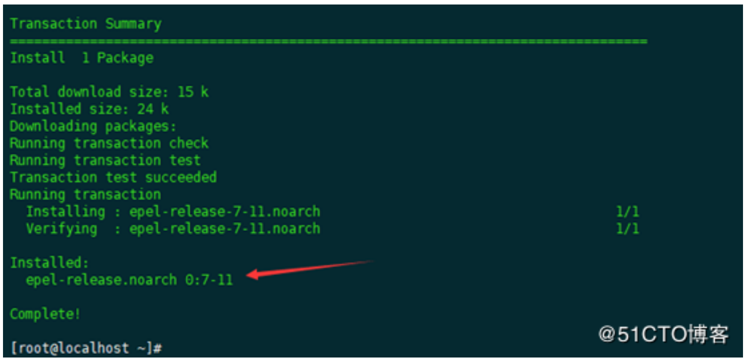
```
[root@localhost ~]# ansible web-servers -m shell -a 'yum -y install epel-release' -i /etc/an
sible/hosts
 [WARNING]: Consider using yum module rather than running yum

192.168.152.160 | SUCCESS | rc=0 >>
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.huaweicloud.com
 * extras: mirrors.huaweicloud.com
 * updates: mirrors.huaweicloud.com
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package               Arch             Version          Repository        Size
================================================================================
Installing:
 epel-release          noarch           7-11             extras            15 k

Transaction Summary
================================================================================
Install  1 Package
```



```
Installing:
 epel-release          noarch           7-11             extras            15 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 15 k
Installed size: 24 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : epel-release-7-11.noarch                                     1/1
  Verifying  : epel-release-7-11.noarch                                     1/1

Installed:
  epel-release.noarch 0:7-11

Complete!
```



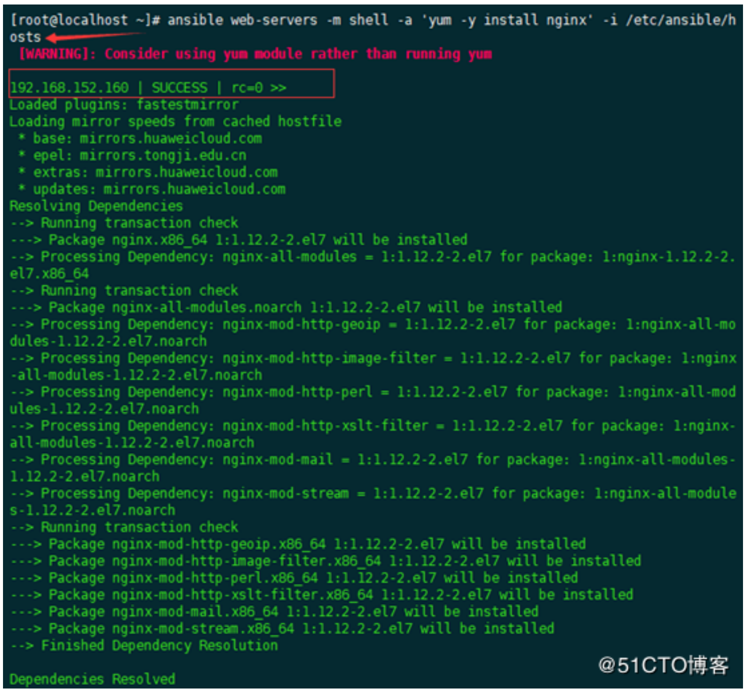
```
192.168.152.159 | SUCCESS | rc=0 >>
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.huaweicloud.com
 * extras: mirrors.huaweicloud.com
 * updates: mirrors.huaweicloud.com
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package               Arch             Version          Repository        Size
================================================================================
Installing:
 epel-release          noarch           7-11             extras            15 k

Transaction Summary
================================================================================
Install  1 Package
```



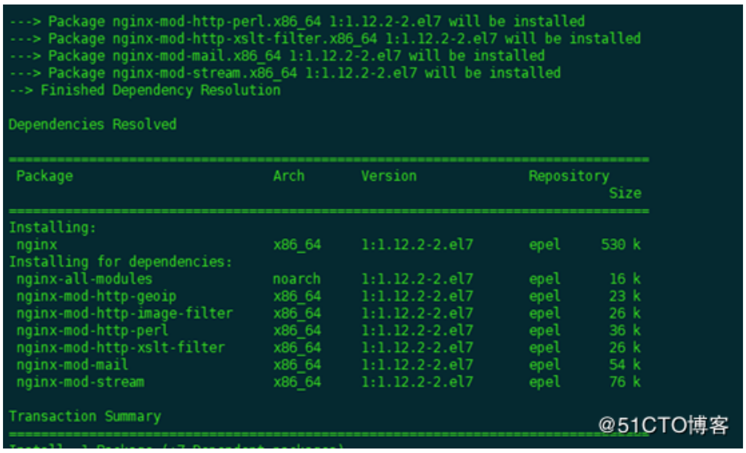
```
Transaction Summary
================================================================================
Install  1 Package

Total download size: 15 k
Installed size: 24 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : epel-release-7-11.noarch                                     1/1
  Verifying  : epel-release-7-11.noarch                                     1/1

Installed:
  epel-release.noarch 0:7-11

Complete!
[root@localhost ~]#
```

6、给两个节点安装nginx

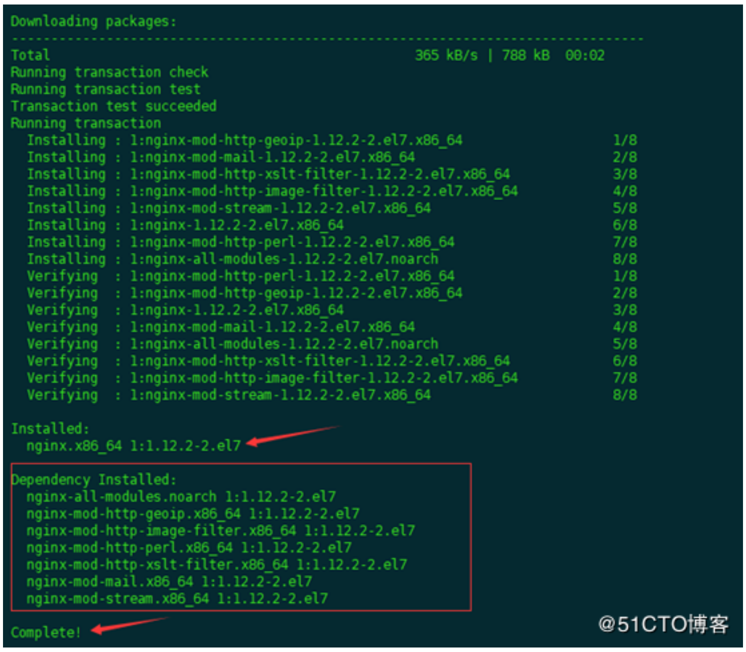
ansible web-servers -m shell -a 'yum -y install nginx' -i /etc/ansible/hosts



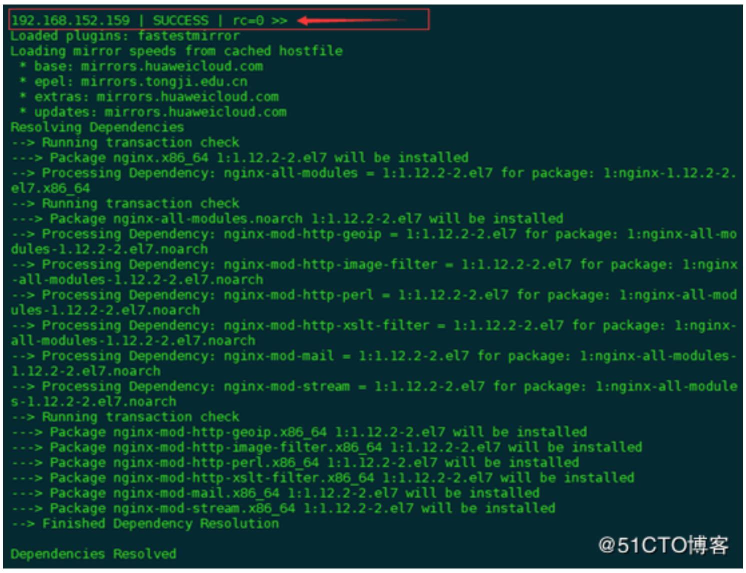
```
[root@localhost ~]# ansible web-servers -m shell -a 'yum -y install nginx' -i /etc/ansible/h
osts
 [WARNING]: Consider using yum module rather than running yum

192.168.152.160 | SUCCESS | rc=0 >>
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.huaweicloud.com
 * epel: mirrors.tongji.edu.cn
 * extras: mirrors.huaweicloud.com
 * updates: mirrors.huaweicloud.com
Resolving Dependencies
--> Running transaction check
---> Package nginx.x86_64 1:1.12.2-2.el7 will be installed
--> Processing Dependency: nginx-all-modules = 1:1.12.2-2.el7 for package: 1:nginx-1.12.2-2.
el7.x86_64
--> Running transaction check
---> Package nginx-all-modules.noarch 1:1.12.2-2.el7 will be installed
--> Processing Dependency: nginx-mod-http-geoip = 1:1.12.2-2.el7 for package: 1:nginx-all-mo
dules-1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-http-image-filter = 1:1.12.2-2.el7 for package: 1:nginx
-all-modules-1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-http-perl = 1:1.12.2-2.el7 for package: 1:nginx-all-mod
ules-1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-http-xslt-filter = 1:1.12.2-2.el7 for package: 1:nginx-
all-modules-1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-mail = 1:1.12.2-2.el7 for package: 1:nginx-all-modules-
1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-stream = 1:1.12.2-2.el7 for package: 1:nginx-all-module
s-1.12.2-2.el7.noarch
--> Running transaction check
---> Package nginx-mod-http-geoip.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-http-image-filter.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-http-perl.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-http-xslt-filter.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-mail.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-stream.x86_64 1:1.12.2-2.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```



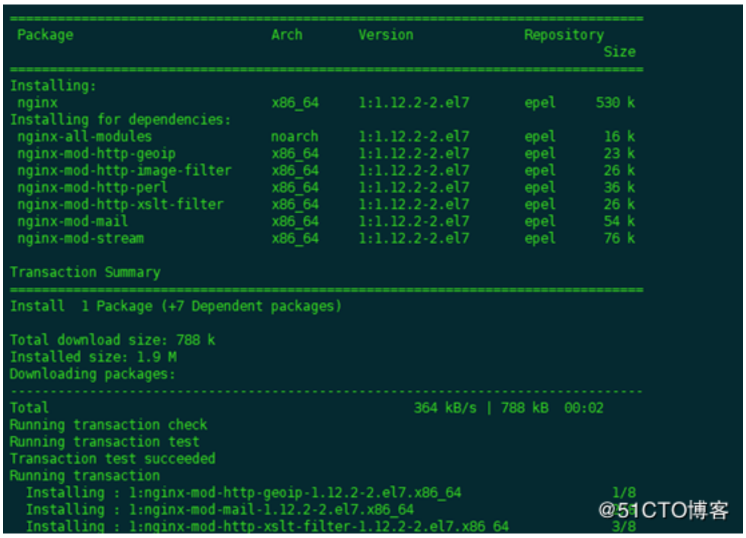
```
---> Package nginx-mod-http-perl.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-http-xslt-filter.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-mail.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-stream.x86_64 1:1.12.2-2.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=================================================================================
 Package                         Arch        Version               Repository
                                                                          Size
=================================================================================
Installing:
 nginx                           x86_64      1:1.12.2-2.el7        epel     530 k
Installing for dependencies:
 nginx-all-modules               noarch      1:1.12.2-2.el7        epel      16 k
 nginx-mod-http-geoip            x86_64      1:1.12.2-2.el7        epel      23 k
 nginx-mod-http-image-filter     x86_64      1:1.12.2-2.el7        epel      26 k
 nginx-mod-http-perl             x86_64      1:1.12.2-2.el7        epel      36 k
 nginx-mod-http-xslt-filter      x86_64      1:1.12.2-2.el7        epel      26 k
 nginx-mod-mail                  x86_64      1:1.12.2-2.el7        epel      54 k
 nginx-mod-stream                x86_64      1:1.12.2-2.el7        epel      76 k

Transaction Summary
=================================================================================
Install  1 Package (+7 Dependent packages)
```



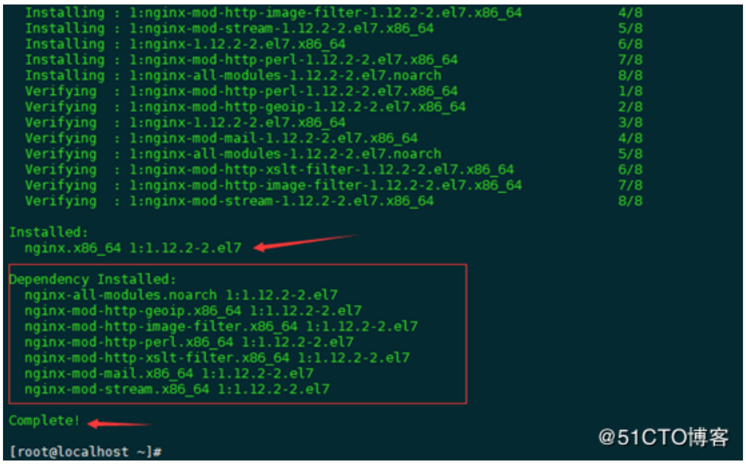
```
Downloading packages:
--------------------------------------------------------------------------------
Total                                                 365 kB/s | 788 kB  00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:nginx-mod-http-geoip-1.12.2-2.el7.x86_64                     1/8
  Installing : 1:nginx-mod-mail-1.12.2-2.el7.x86_64                           2/8
  Installing : 1:nginx-mod-http-xslt-filter-1.12.2-2.el7.x86_64               3/8
  Installing : 1:nginx-mod-http-image-filter-1.12.2-2.el7.x86_64              4/8
  Installing : 1:nginx-mod-stream-1.12.2-2.el7.x86_64                         5/8
  Installing : 1:nginx-1.12.2-2.el7.x86_64                                    6/8
  Installing : 1:nginx-mod-http-perl-1.12.2-2.el7.x86_64                      7/8
  Installing : 1:nginx-all-modules-1.12.2-2.el7.noarch                        8/8
  Verifying  : 1:nginx-mod-http-perl-1.12.2-2.el7.x86_64                      1/8
  Verifying  : 1:nginx-mod-http-geoip-1.12.2-2.el7.x86_64                     2/8
  Verifying  : 1:nginx-1.12.2-2.el7.x86_64                                    3/8
  Verifying  : 1:nginx-mod-mail-1.12.2-2.el7.x86_64                           4/8
  Verifying  : 1:nginx-all-modules-1.12.2-2.el7.noarch                        5/8
  Verifying  : 1:nginx-mod-http-xslt-filter-1.12.2-2.el7.x86_64               6/8
  Verifying  : 1:nginx-mod-http-image-filter-1.12.2-2.el7.x86_64              7/8
  Verifying  : 1:nginx-mod-stream-1.12.2-2.el7.x86_64                         8/8

Installed:
  nginx.x86_64 1:1.12.2-2.el7

Dependency Installed:
  nginx-all-modules.noarch 1:1.12.2-2.el7
  nginx-mod-http-geoip.x86_64 1:1.12.2-2.el7
  nginx-mod-http-image-filter.x86_64 1:1.12.2-2.el7
  nginx-mod-http-perl.x86_64 1:1.12.2-2.el7
  nginx-mod-http-xslt-filter.x86_64 1:1.12.2-2.el7
  nginx-mod-mail.x86_64 1:1.12.2-2.el7
  nginx-mod-stream.x86_64 1:1.12.2-2.el7

Complete!
```



```
192.168.152.159 | SUCCESS | rc=0 >>
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.huaweicloud.com
 * epel: mirrors.tongji.edu.cn
 * extras: mirrors.huaweicloud.com
 * updates: mirrors.huaweicloud.com
Resolving Dependencies
--> Running transaction check
---> Package nginx.x86_64 1:1.12.2-2.el7 will be installed
--> Processing Dependency: nginx-all-modules = 1:1.12.2-2.el7 for package: 1:nginx-1.12.2-2.
el7.x86_64
--> Running transaction check
---> Package nginx-all-modules.noarch 1:1.12.2-2.el7 will be installed
--> Processing Dependency: nginx-mod-http-geoip = 1:1.12.2-2.el7 for package: 1:nginx-all-mo
dules-1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-http-image-filter = 1:1.12.2-2.el7 for package: 1:nginx
-all-modules-1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-http-perl = 1:1.12.2-2.el7 for package: 1:nginx-all-mod
ules-1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-http-xslt-filter = 1:1.12.2-2.el7 for package: 1:nginx-
all-modules-1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-mail = 1:1.12.2-2.el7 for package: 1:nginx-all-modules-
1.12.2-2.el7.noarch
--> Processing Dependency: nginx-mod-stream = 1:1.12.2-2.el7 for package: 1:nginx-all-module
s-1.12.2-2.el7.noarch
--> Running transaction check
---> Package nginx-mod-http-geoip.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-http-image-filter.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-http-perl.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-http-xslt-filter.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-mail.x86_64 1:1.12.2-2.el7 will be installed
---> Package nginx-mod-stream.x86_64 1:1.12.2-2.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

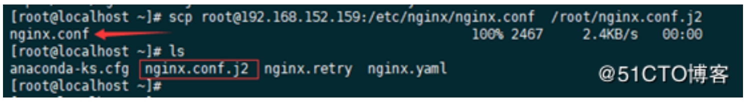


```
================================================================================
 Package                          Arch        Version               Repository
                                                                          Size
================================================================================
Installing:
 nginx                            x86_64      1:1.12.2-2.el7        epel     530 k
Installing for dependencies:
 nginx-all-modules                noarch      1:1.12.2-2.el7        epel      16 k
 nginx-mod-http-geoip             x86_64      1:1.12.2-2.el7        epel      23 k
 nginx-mod-http-image-filter      x86_64      1:1.12.2-2.el7        epel      26 k
 nginx-mod-http-perl              x86_64      1:1.12.2-2.el7        epel      36 k
 nginx-mod-http-xslt-filter       x86_64      1:1.12.2-2.el7        epel      26 k
 nginx-mod-mail                   x86_64      1:1.12.2-2.el7        epel      54 k
 nginx-mod-stream                 x86_64      1:1.12.2-2.el7        epel      76 k

Transaction Summary
================================================================================
Install  1 Package (+7 Dependent packages)

Total download size: 788 k
Installed size: 1.9 M
Downloading packages:
--------------------------------------------------------------------------------
Total                                                   364 kB/s | 788 kB  00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:nginx-mod-http-geoip-1.12.2-2.el7.x86_64                     1/8
  Installing : 1:nginx-mod-mail-1.12.2-2.el7.x86_64                           2/8
  Installing : 1:nginx-mod-http-xslt-filter-1.12.2-2.el7.x86_64               3/8
```



```
  Installing : 1:nginx-mod-http-image-filter-1.12.2-2.el7.x86_64              4/8
  Installing : 1:nginx-mod-stream-1.12.2-2.el7.x86_64                         5/8
  Installing : 1:nginx-1.12.2-2.el7.x86_64                                    6/8
  Installing : 1:nginx-mod-http-perl-1.12.2-2.el7.x86_64                      7/8
  Installing : 1:nginx-all-modules-1.12.2-2.el7.noarch                        8/8
  Verifying  : 1:nginx-mod-http-perl-1.12.2-2.el7.x86_64                      1/8
  Verifying  : 1:nginx-mod-http-geoip-1.12.2-2.el7.x86_64                     2/8
  Verifying  : 1:nginx-1.12.2-2.el7.x86_64                                    3/8
  Verifying  : 1:nginx-mod-mail-1.12.2-2.el7.x86_64                           4/8
  Verifying  : 1:nginx-all-modules-1.12.2-2.el7.noarch                        5/8
  Verifying  : 1:nginx-mod-http-xslt-filter-1.12.2-2.el7.x86_64               6/8
  Verifying  : 1:nginx-mod-http-image-filter-1.12.2-2.el7.x86_64              7/8
  Verifying  : 1:nginx-mod-stream-1.12.2-2.el7.x86_64                         8/8

Installed:
  nginx.x86_64 1:1.12.2-2.el7

Dependency Installed:
  nginx-all-modules.noarch 1:1.12.2-2.el7
  nginx-mod-http-geoip.x86_64 1:1.12.2-2.el7
  nginx-mod-http-image-filter.x86_64 1:1.12.2-2.el7
  nginx-mod-http-perl.x86_64 1:1.12.2-2.el7
  nginx-mod-http-xslt-filter.x86_64 1:1.12.2-2.el7
  nginx-mod-mail.x86_64 1:1.12.2-2.el7
  nginx-mod-stream.x86_64 1:1.12.2-2.el7

Complete!

[root@localhost ~]#
```

7、编辑本地nginx.conf.j2文件（因为本地没有安装nginx所以没有这个文件需要从节点拷贝到当前/root/下再修改，并以这个修改过的模板来下发给节点）

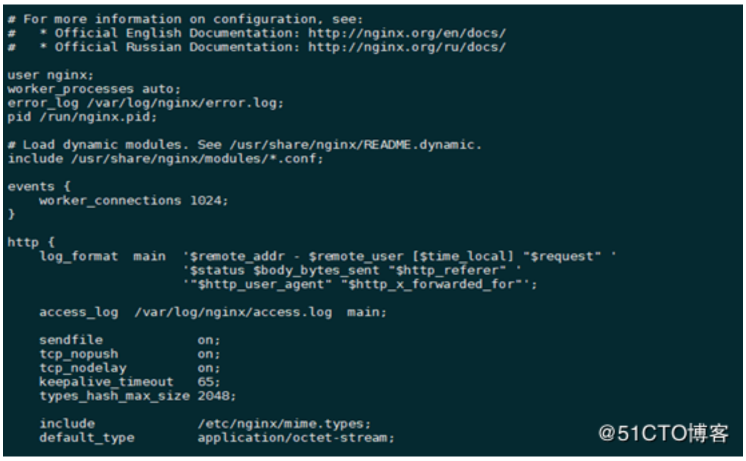


```
[root@localhost ~]# scp root@192.168.152.159:/etc/nginx/nginx.conf  /root/nginx.conf.j2
nginx.conf                                                   100% 2467     2.4KB/s   00:00
[root@localhost ~]# ls
anaconda-ks.cfg  nginx.conf.j2  nginx.retry  nginx.yaml
[root@localhost ~]#
```

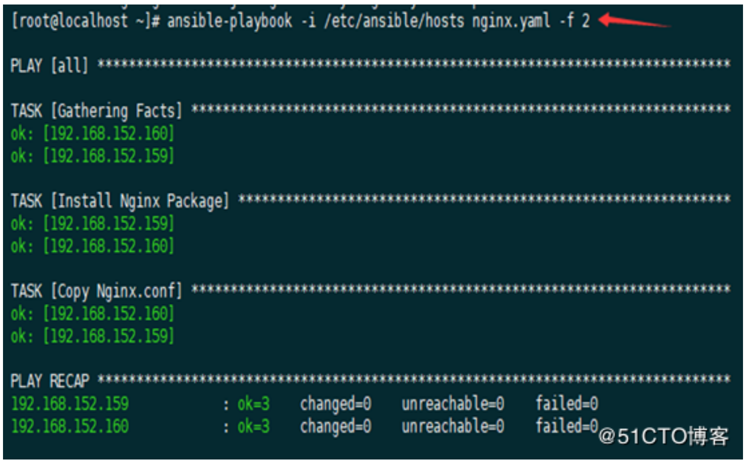
注：根据实际情况要求修改（我这里使用的默认）

8、确认信息是否正确

ansible-playbook -i /etc/ansible/hosts nginx.yaml -f 2

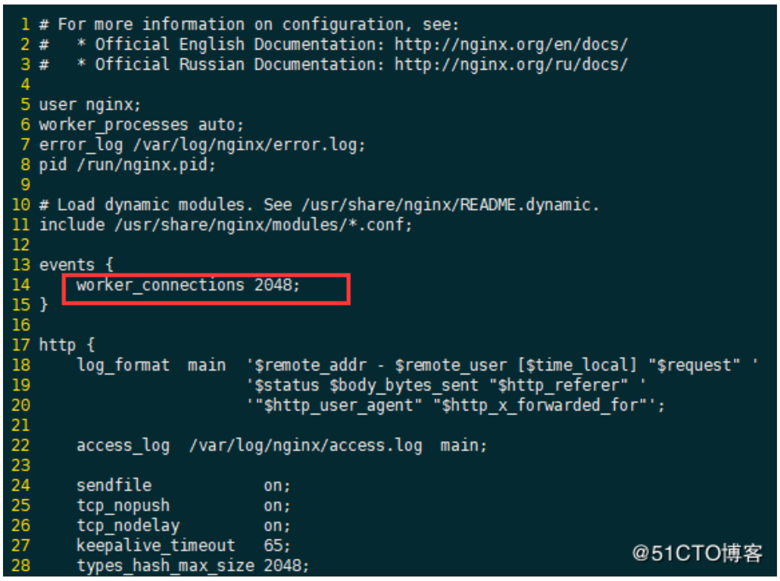


```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;
```

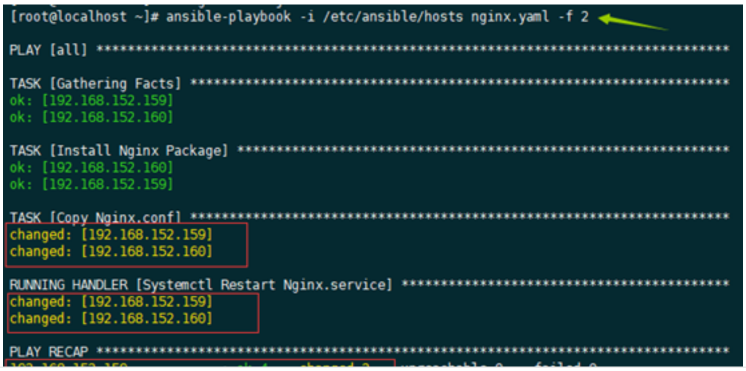


```
[root@localhost ~]# ansible-playbook -i /etc/ansible/hosts nginx.yaml -f 2

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.152.160]
ok: [192.168.152.159]

TASK [Install Nginx Package] ***************************************************
ok: [192.168.152.159]
ok: [192.168.152.160]

TASK [Copy Nginx.conf] *********************************************************
ok: [192.168.152.160]
ok: [192.168.152.159]

PLAY RECAP *********************************************************************
192.168.152.159            : ok=3    changed=0    unreachable=0    failed=0
192.168.152.160            : ok=3    changed=0    unreachable=0    failed=0
```

2）这是修改过nginx.conf.j2配置文件执行命令的状态



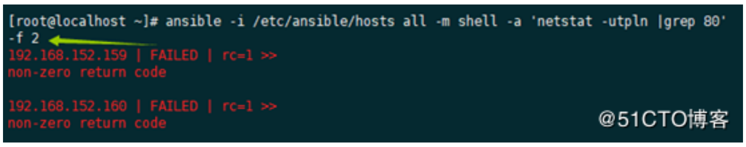
```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 2048;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;
```



```
[root@localhost ~]# ansible-playbook -i /etc/ansible/hosts nginx.yaml -f 2

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.152.159]
ok: [192.168.152.160]

TASK [Install Nginx Package] ***************************************************
ok: [192.168.152.160]
ok: [192.168.152.159]

TASK [Copy Nginx.conf] *********************************************************
changed: [192.168.152.159]
changed: [192.168.152.160]

RUNNING HANDLER [Systemctl Restart Nginx.service] ******************************
changed: [192.168.152.159]
changed: [192.168.152.160]

PLAY RECAP *********************************************************************
```

9、这样我们就完成了 3台机器的Nginx安装部署，下面需要对主机的Nginx服务进行核查，并且确认生成后nginx.conf中的worker_processes参数的值是否正确，执行命令：

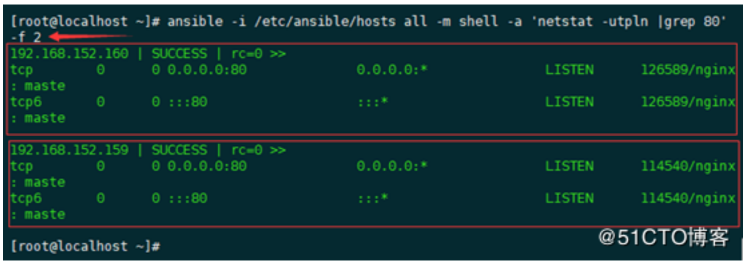
ansible -i /etc/ansible/hosts all -m shell -a 'netstat -utpln |grep 80' -f 2



```
[root@localhost ~]# ansible -i /etc/ansible/hosts all -m shell -a 'netstat -utpln |grep 80'
-f 2
192.168.152.159 | FAILED | rc=1 >>
non-zero return code

192.168.152.160 | FAILED | rc=1 >>
non-zero return code
```

@51CTO博客

注：上图红色字体说明并没有自动重启nginx，原因是nginx.conf.j2这文件默认没有修改的情况不满足执行条件，所以如果想得到下图，随意修改点东西即可。

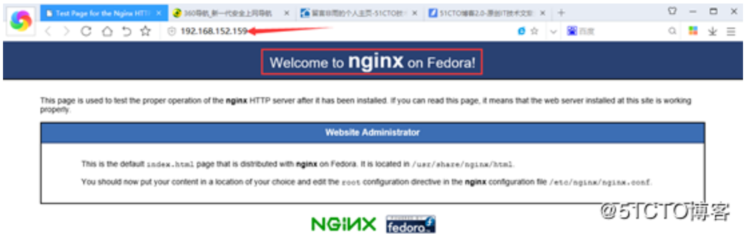


```
[root@localhost ~]# ansible -i /etc/ansible/hosts all -m shell -a 'netstat -utpln |grep 80'
-f 2
192.168.152.160 | SUCCESS | rc=0 >>
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      126589/nginx
: maste
tcp6       0      0 :::80                   :::*                    LISTEN      126589/nginx
: maste

192.168.152.159 | SUCCESS | rc=0 >>
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      114540/nginx
: maste
tcp6       0      0 :::80                   :::*                    LISTEN      114540/nginx
: maste

[root@localhost ~]#
```

@51CTO博客

10、验证：浏览器访问两个节点IP

Web1：http://192.168.152.159



Welcome to **nginx** on Fedora!

This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

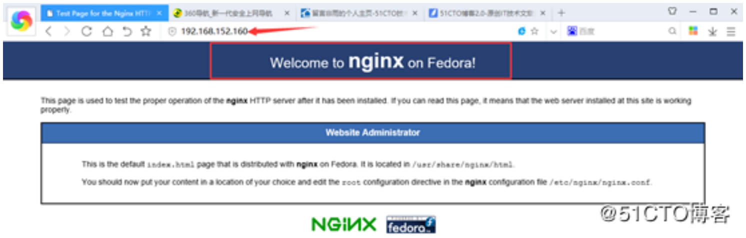
Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Fedora. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

@51CTO博客

Web2：http://192.168.152.160



Welcome to **nginx** on Fedora!

This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Fedora. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

@51CTO博客

注：nginx自动化部署完成！

好知识，才能预见未来

赞赏

0人进行了赞赏支持

Ansible 部署 nginx 自动化运维

1 收藏 分享

1 0 分享 GIVE_ME 关注

GIVE_ME

101篇文章，109W+人气，19粉丝

I am so strong！

关注

提问和评论都可以，用心的回复会被更多人看到和认可

Ctrl+Enter 发布

取消 发布

推荐专栏

更多

基于Python的DevOps实战

自动化运维开发新概念

共20章 | 抚琴煮酒

￥51.00 386人订阅

订 阅

全局视角看大型园区网

路由交换+安全+无线+优化+运维

共40章 | 51CTO夏杰

￥51.00 1210人订阅

订 阅

网工2.0晋级攻略 ——零基础入门Python/A...

网络工程师2.0进阶指南

共30章 | 姜汁啤酒

￥51.00 1413人订阅

订 阅

负载均衡高手炼成记

高并发架构之路

共15章 | sery

￥51.00 472人订阅

订 阅

带你玩转高可用

前百度高级工程师的架构高可用实战

共15章 | 曹林华

￥51.00 442人订阅

订 阅

在线
客服

猜你喜欢

MySQL5.7 命令自动补全工具mycli

shell脚本之编程规范与变量

运维自动化-Ansible (一)

nginx反向代理配置及优化

自动化运维工具Ansible详细部署

vCenter Server Appliance（VCSA ）6.7部署指南

1 0 分享

GIVE_ME 关注

关于Nginx的server_name。

CentOS7 搭建企业级NFS网络文件服务器

Django+Django-Celery+Celery的整合实战

Nginx10m+高并发内核优化详解

部署SaltStack及批量安装httpd服务

自动化运维工具Ansible实战（五）Playbooks剧本使用

k8s实践6:从解决报错开始入门RBAC

Linux 性能测试工具 sysbench 的安装与简单使用

架构师的操作系统

Kubernetes共享存储之Glusterfs+Heketi

在线
客服

1 0 分享 GIVE_ME 关注