Ansible “Ad-hoc” mode can be used to copy/delete/modify files on the specific host or Group of hosts using ansible modules. File module is used for setting file attributes like permission, ownership and creating the links. “Copy” module is used to copy the files to hosts from Ansible server. These modules are very often used in Ad-hoc mode to push the application configurations, system configurations etc.. You could quickly push the configuration to multiple hosts using these modules. Let’s demonstrate the functions of  “file” and “copy” modules.

**File operations:**

1. Login to Ansible server and list the hosts.

[sysadmin@ansible-server ~]$ ansible --list-hosts all

hosts (4):

uaans

ana-1

uaans69

ana-2

[sysadmin@ansible-server ~]$ ansible uaans69 -m ping

uaans69 | SUCCESS => {

"changed": false,

"ping": "pong"

}

[sysadmin@ansible-server ~]$

2. Create a blank file on the host (uaans69) with a given path using ansible.

[sysadmin@ansible-server ~]$ ansible uaans69 -m file -a 'path=/var/tmp/ansible\_test.txt state=touch'

uaans69 | SUCCESS => {

"changed": true,

"dest": "/var/tmp/ansible\_test.txt",

"gid": 502,

"group": "sysadmin",

"mode": "0664",

"owner": "sysadmin",

"secontext": "unconfined\_u:object\_r:user\_tmp\_t:s0",

"size": 0,

"state": "file",

"uid": 502

}

[sysadmin@ansible-server ~]$

Verify our work by logging in to uaans69 host.

[sysadmin@ansible-server ~]$ ssh uaans69

Last login: Wed Jul 11 05:09:24 2018 from 192.168.3.151

[sysadmin@uaans69 ~]$ ls -lrt /var/tmp/ansible\_test.txt

-rw-rw-r--. 1 sysadmin sysadmin 0 Jul 11 05:09 /var/tmp/ansible\_test.txt

[sysadmin@uaans69 ~]$ logout

Connection to uaans69 closed.

[sysadmin@ansible-server ~]$

Let’s change the file permission and ownership and verify by directly logging in to the host.

[sysadmin@ansible-server ~]$ ansible uaans69 -b -m file -a 'path=/var/tmp/ansible\_test.txt owner=root group=root mode=0644'

uaans69 | SUCCESS => {

"changed": true,

"gid": 0,

"group": "root",

"mode": "0644",

"owner": "root",

"path": "/var/tmp/ansible\_test.txt",

"secontext": "unconfined\_u:object\_r:user\_tmp\_t:s0",

"size": 0,

"state": "file",

"uid": 0

}

[sysadmin@ansible-server ~]$ ssh uaans69 ls -lrt /var/tmp/ansible\_test.txt

-rw-r--r--. 1 root root 0 Jul 11 05:59 /var/tmp/ansible\_test.txt

[sysadmin@ansible-server ~]$

3. Remove the newly created file and verify by directly logging in to the host.

[sysadmin@ansible-server ~]$ ansible uaans69 -m file -a 'path=/var/tmp/ansible\_test.txt state=absent'

uaans69 | SUCCESS => {

"changed": true,

"path": "/var/tmp/ansible\_test.txt",

"state": "absent"

}

[sysadmin@ansible-server ~]$ ssh uaans69

Last login: Wed Jul 11 05:09:49 2018 from 192.168.3.151

[sysadmin@uaans69 ~]$ ls -lrt /var/tmp/ansible\_test.txt

ls: cannot access /var/tmp/ansible\_test.txt: No such file or directory

[sysadmin@uaans69 ~]$

4. Let’s create a soft link on the host.

[sysadmin@ansible-server ~]$ ansible uaans69 -m file -a 'src=/etc/hosts dest=/var/tmp/hosts state=link'

uaans69 | SUCCESS => {

"changed": true,

"dest": "/var/tmp/hosts",

"gid": 502,

"group": "sysadmin",

"mode": "0777",

"owner": "sysadmin",

"secontext": "unconfined\_u:object\_r:user\_tmp\_t:s0",

"size": 10,

"src": "/etc/hosts",

"state": "link",

"uid": 502

}

Verify our work.

[sysadmin@ansible-server ~]$ ssh uaans69

Last login: Wed Jul 11 05:56:12 2018 from 192.168.3.151

[sysadmin@uaans69 ~]$ ls -lrt /var/tmp/hosts

lrwxrwxrwx. 1 sysadmin sysadmin 10 Jul 11 05:56 /var/tmp/hosts -> /etc/hosts

[sysadmin@uaans69 ~]$

5. Create a new directory on the host and verify.

[sysadmin@ansible-server ~]$ ansible uaans69 -m file -a 'path=/var/tmp/archive state=directory'

uaans69 | SUCCESS => {

"changed": true,

"gid": 502,

"group": "sysadmin",

"mode": "0775",

"owner": "sysadmin",

"path": "/var/tmp/archive",

"secontext": "unconfined\_u:object\_r:user\_tmp\_t:s0",

"size": 4096,

"state": "directory",

"uid": 502

}

[sysadmin@ansible-server ~]$ ssh uaans69

Last login: Wed Jul 11 06:03:13 2018 from 192.168.3.151

[sysadmin@uaans69 ~]$ ls -ld /var/tmp/archive

drwxrwxr-x. 2 sysadmin sysadmin 4096 Jul 11 06:03 /var/tmp/archive

[sysadmin@uaans69 ~]$

You could change the permission and ownership of the directory file module.

### ****” copy ” Module- Copies files to remote locations:****

The copy module copies a file from the local or remote machine to a location on the remote machine. Use the fetch module to copy files from remote locations to the local host.

1. Copy the Ansible server’s /etc/hosts file to remote host “uaans69”. Here are the uaans69’s “/etc/hosts” contents.

cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

The following copy action failed since we didn’t escalate the privileges.

[sysadmin@ansible-server ~]$ ansible uaans69 -m copy -a 'src=/etc/hosts dest=/etc/hosts'

uaans69 | FAILED! => {

"changed": false,

"checksum": "021b217c1b2f61e9b5faa24885ac951970e1f6e8",

"msg": "Destination /etc not writable"

}

[sysadmin@ansible-server ~]$

“/etc/hosts” modification requires root privileges. use “-b” option to become the root user.

[sysadmin@ansible-server ~]$ ansible uaans69 -b -m copy -a 'src=/etc/hosts dest=/etc/hosts'

uaans69 | SUCCESS => {

"changed": true,

"checksum": "021b217c1b2f61e9b5faa24885ac951970e1f6e8",

"dest": "/etc/hosts",

"gid": 0,

"group": "root",

"md5sum": "56aadbedb93d3be9a472f1725fa828b0",

"mode": "0644",

"owner": "root",

"secontext": "system\_u:object\_r:net\_conf\_t:s0",

"size": 557,

"src": "/home/sysadmin/.ansible/tmp/ansible-tmp-1531284115.26-70289318446057/source",

"state": "file",

"uid": 0

}

[sysadmin@ansible-server ~]$

Verify our work.

[sysadmin@ansible-server ~]$ ssh uaans69 cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.3.151 ansible-server

192.168.3.150 uaans69

192.168.3.201 ana-1

192.168.3.20 uaans

[sysadmin@ansible-server ~]$

2. We must take the backup of any configuration file before editing it. Is it possible to take a backup before overwriting it? Offcourse, You can do that using “backup” keyword.

[sysadmin@ansible-server ~]$ ansible uaans69 -b -m copy -a 'src=/etc/hosts dest=/etc/hosts backup=yes'

uaans69 | SUCCESS => {

"backup\_file": "/etc/hosts.9969.2018-07-11@06:16:49~",

"changed": true,

"checksum": "021b217c1b2f61e9b5faa24885ac951970e1f6e8",

"dest": "/etc/hosts",

"gid": 0,

"group": "root",

"md5sum": "56aadbedb93d3be9a472f1725fa828b0",

"mode": "0644",

"owner": "root",

"secontext": "system\_u:object\_r:net\_conf\_t:s0",

"size": 557,

"src": "/home/sysadmin/.ansible/tmp/ansible-tmp-1531284592.86-208978173473604/source",

"state": "file",

"uid": 0

}

Verify the backup file.

[sysadmin@ansible-server ~]$ ssh uaans69

Last login: Wed Jul 11 06:16:49 2018 from 192.168.3.151

[sysadmin@uaans69 ~]$ ls -lrt /etc/hosts\*

-rw-r--r--. 1 root root 460 Jan 12 2010 /etc/hosts.deny

-rw-r--r--. 1 root root 370 Jan 12 2010 /etc/hosts.allow

-rw-r--r--. 1 root root 158 Jul 11 06:16 /etc/hosts.9969.2018-07-11@06:16:49~

-rw-r--r--. 1 root root 557 Jul 11 06:16 /etc/hosts

[sysadmin@uaans69 ~]$ cat /etc/hosts.9969.2018-07-11\@06\:16\:49~

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

[sysadmin@uaans69 ~]$

We could also update the permissions and ownership for the copied files using “copy” module.