Modifying and Copying Files to Hosts

In this Lab we will use standard Ansible modules to **create, install, edit**, and **remove** files on managed hosts and manage the **permissions, ownership** and **SELinux contexts** of those files.

Log in to **ansi-master** as **root** using **linux** as the **password**.

1. Change to the **~/root/filemanage** working directory. Create a playbook called **securebackup.yml** in the current working directory. Configure the playbook to use the fetch module to retrieve the **/var/log/secure** log file from each of managed hosts and store them on the control node. The playbook should create the secure-backups directory with sub directories named after the hostname of each managed host.

```
# cd ansible
# mkdir filemanage
# cd filemanage
```

1.1 Add a task to **securebackup.yml** playbook that retrieves the **/var/log/secure** log file from the managed hosts and stores it in the ~/file-manage/secure-backups directory. The fetch modules creates the **~/file-manage/secure-backups directory** if it does not exist. Use the flat: no parameter to ensure the default behavior of appending the hostname, path and file name to destination.

```
# cat > securebackup.yaml <<EOF
---
- name: Use the fetch module to retrieve secure log files
  hosts: all
  remote_user: root
  become: yes
  tasks:
    - name: Fetch the /var/log/secure log file from managed
hosts
      fetch:
        src: /var/log/secure
        dest: secure-backups
        flat: no
EOF
```

1.2 Run the ansible-playbook --syntax-check securebackup.yml command to verify its syntax and correct any errors.

```
# ansible-playbook --syntax-check securebackup.yaml
```

Output:

```
[root@ansi-master ansible]# ansible-playbook --syntax-check securebackup.yaml

playbook: securebackup.yaml
```

1.3 Run ansible-playbook securebackup.yml to execute the playbook.

```
# ansible-playbook securebackup.yaml
```

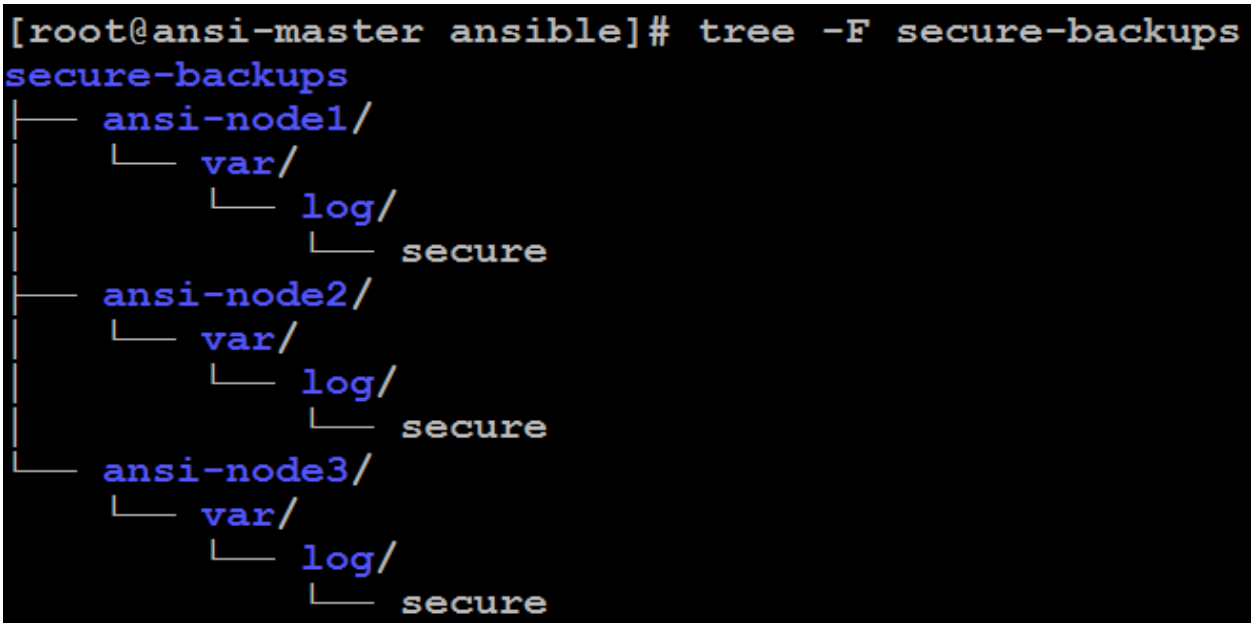
Output:

```
[root@ansi-master ansible]# ansible-playbook securebackup.yaml

PLAY [Use the fetch module to retrieve secure log files] ***********************************************

TASK [Gathering Facts] *********************************************************************************
ok: [ansi-node3]
ok: [ansi-node2]
ok: [ansi-node1]

TASK [Fetch the /var/log/secure log file from managed hosts] *******************************************
changed: [ansi-node3]
changed: [ansi-node1]
changed: [ansi-node2]

PLAY RECAP *********************************************************************************************
ansi-node1                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

1.4 Verify the playbook results:

```
$ tree -F secure-backups
```

Output:



2. Create the **copyfile.yml** playbook in the current working directory. Configure the playbook to copy the **/root/ filemanage/files/users.txt** file to all managed hosts as the devops user.

2.1 Add a task to use the copy module to copy the /home/devops/file-manage/files/users.txt file to all managed hosts. Use the copy module to set the following parameters for the users.txt file:

PARAMETER	VALUES
src	files/users.txt
dest	/home/devops/users.txt
owner	devops
group	devops
mode	u+rw,g-wx,o-rwx
setype	samba_share_t

```
$ cat > copyfile.yml <<EOF
---
- name: Using the copy module
  hosts: all
  remote_user: root
  become: yes
  tasks:
    - name: Copy a file to managed hosts and set attributes
      copy:
        src: files/users.txt
        dest: /users.txt
        owner: root
        group: root
        mode: u+rw,g-wx,o-rwx
        setype: samba_share_t
EOF
```

2.2 Create a **directory** by the name **files** and a **file** name **users.txt** inside the directory.

```
$ mkdir files
$ touch files/users.txt
```

2.2 Run the ansible-playbook --syntax-check copyfile.yml command to verify its syntax and correct any errors.

```
$ ansible-playbook --syntax-check copyfile.yml
```

Output:

```
[root@ansi-master ansible]# ansible-playbook --syntax-check copyfile.yml

playbook: copyfile.yml
```

2.3 Run ansible-playbook copyfile.yml to execute the playbook.

```
$ ansible-playbook  copyfile.yml
```

```
[root@ansi-master ansible]# ansible-playbook  copyfile.yml

PLAY [Using the copy module] ***************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [ansi-node3]
ok: [ansi-node2]
ok: [ansi-node1]

TASK [Copy a file to managed hosts and set attributes] *************************************************
ok: [ansi-node3]
ok: [ansi-node1]
ok: [ansi-node2]

PLAY RECAP *********************************************************************************************
ansi-node1                 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

2.4 Use an ad hoc command to execute the ls –Z command as user root to verify the attributes of the users.txt file on the managed hosts.

```
$ ansible all -m command -a 'ls -lZ /users.txt'
```

3. Create a playbook called **selinuxdefaults.yml** in the current working directory. Configure the playbook to use the file module to ensure the default SELinux context for user, role, type, and level fields.

```
$ cat > selinuxdefaults.yml <<EOF
---
- name: Using the file module to ensure SELinux file context
  hosts: all
  remote_user: root
  tasks:
    - name: SELinux file context is set to defaults
      file:
        path: /users.txt
        seuser: _default
        serole: _default
        setype: _default
        selevel: _default
EOF
```

3.1 Run the ansible-playbook --syntax-check selinuxdefaults.yml command to verify its syntax and correct any errors.

```
$ ansible-playbook --syntax-check selinuxdefaults.yml
```

Output:

```
[root@ansi-master ansible]# ansible-playbook --syntax-check selinuxdefaults.yml

playbook: selinuxdefaults.yml
```

3.2 Run ansible-playbook selinuxdefaults.yml to execute the playbook.

```
$ ansible-playbook  selinuxdefaults.yml
```

```
[root@ansi-master ansible]# ansible-playbook  selinuxdefaults.yml

PLAY [Using the file module to ensure SELinux file context] ********************************************

TASK [Gathering Facts] *********************************************************************************
ok: [ansi-node3]
ok: [ansi-node1]
ok: [ansi-node2]

TASK [SELinux file context is set to defaults] *********************************************************
changed: [ansi-node1]
ok: [ansi-node3]
changed: [ansi-node2]

PLAY RECAP *********************************************************************************************
ansi-node1                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

3.3 Use an ad hoc command to execute the ls –Z command as user root to verify the default file attributes on unconfined_u:object_r:user_home_t:s0

```
# ansible all -m command -a 'ls -Z'
```

Output:

```
[root@ansi-master ansible]# ansible all -m command -a 'ls -Z'
ansi-node1 | CHANGED | rc=0 >>
    system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
unconfined_u:object_r:admin_home_t:s0 Desktop
unconfined_u:object_r:admin_home_t:s0 Documents
unconfined_u:object_r:admin_home_t:s0 Downloads
    system_u:object_r:admin_home_t:s0 initial-setup-ks.cfg
unconfined_u:object_r:admin_home_t:s0 Music
unconfined_u:object_r:admin_home_t:s0 Pictures
unconfined_u:object_r:admin_home_t:s0 Public
unconfined_u:object_r:admin_home_t:s0 Templates
unconfined_u:object_r:admin_home_t:s0 Videos
ansi-node3 | CHANGED | rc=0 >>
    system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
unconfined_u:object_r:admin_home_t:s0 Desktop
unconfined_u:object_r:admin_home_t:s0 Documents
unconfined_u:object_r:admin_home_t:s0 Downloads
    system_u:object_r:admin_home_t:s0 initial-setup-ks.cfg
unconfined_u:object_r:admin_home_t:s0 Music
unconfined_u:object_r:admin_home_t:s0 Pictures
unconfined_u:object_r:admin_home_t:s0 Public
unconfined_u:object_r:admin_home_t:s0 Templates
unconfined_u:object_r:admin_home_t:s0 Videos
ansi-node2 | CHANGED | rc=0 >>
    system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
unconfined_u:object_r:admin_home_t:s0 Desktop
unconfined_u:object_r:admin_home_t:s0 Documents
unconfined_u:object_r:admin_home_t:s0 Downloads
    system_u:object_r:admin_home_t:s0 initial-setup-ks.cfg
unconfined_u:object_r:admin_home_t:s0 Music
unconfined_u:object_r:admin_home_t:s0 Pictures
unconfined_u:object_r:admin_home_t:s0 Public
unconfined_u:object_r:admin_home_t:s0 Templates
unconfined_u:object_r:admin_home_t:s0 Videos
```

4. Create a playbook called **addline.yml** in the current working directory. Configure the playbook to use the **lineinfile** module to append the line. This line wad added by the lineinfile module to the **/users.txt** file on all managed hosts.

```
$ cat > addline.yml <<EOF
---
- name: Add text to an existing file
  hosts: all
  remote_user: root
  tasks:
    - name: Add a single line of text to a file
      lineinfile:
        path: /users.txt
        line: This line was added by lineinfile
        state: present
EOF
```

4.1 Run the ansible-playbook --syntax-check addline.yml command to verify its syntax and correct any errors.

```
$ ansible-playbook --syntax-check addline.yml
```

Output:

```
[root@ansi-master ansible]# ansible-playbook --syntax-check addline.yml

playbook: addline.yml
```

4.2 Run ansible-playbook addline.yml to execute the playbook.

```
$ ansible-playbook  addline.yml
```

```
[root@ansi-master ansible]# ansible-playbook  addline.yml

PLAY [Add text to an existing file] ********************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [ansi-node1]
ok: [ansi-node3]
ok: [ansi-node2]

TASK [Add a single line of text to a file] *************************************************************
changed: [ansi-node3]
changed: [ansi-node1]
changed: [ansi-node2]

PLAY RECAP *********************************************************************************************
ansi-node1                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

4.3 Use the command module with the cat option as the devops user, to verify the content of the users.txt file on the managed hosts.

```
$ ansible all -m command -a 'cat /users.txt'
```

Output:

```
[root@ansi-master ansible]# ansible all -m command -a 'cat /users.txt'
ansi-node3 | CHANGED | rc=0 >>
This line was added by lineinfile
ansi-node1 | CHANGED | rc=0 >>
This line was added by lineinfile
ansi-node2 | CHANGED | rc=0 >>
This line was added by lineinfile
```

5. Create a playbook called **addblock.yml** in the current working directory. Configure the playbook to use the blockinfile module to append the following block to text to the /users.txt file on all managed hosts.

```
$ cat > addblock.yml <<EOF
---
- name: Add block of text to a file
  hosts: all
  remote_user: root
  tasks:
    - name: Add a block of text to an existing file
      blockinfile:
        path: /users.txt
        block: |
          This block of text consists of two lines
          They have been added by blockinfile module
        state: present
EOF
```

5.1 Run the ansible-playbook --syntax-check addblock.yml command to verify its syntax and correct any errors.

```
$ ansible-playbook --syntax-check addblock.yml
```

Output:

```
[root@ansi-master ansible]# ansible-playbook --syntax-check addblock.yml

playbook: addblock.yml
```

5.2 Run ansible-playbook addblock.yml to execute the playbook.

```
$ ansible-playbook  addblock.yml
```

Output:

```
[root@ansi-master ansible]# ansible-playbook  addblock.yml

PLAY [Add block of text to a file] *********************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [ansi-node3]
ok: [ansi-node2]
ok: [ansi-node1]

TASK [Add a block of text to an existing file] *********************************************************
changed: [ansi-node1]
changed: [ansi-node3]
changed: [ansi-node2]

PLAY RECAP *********************************************************************************************
ansi-node1                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

5.3 Use the command module with the cat command to verify the correct content of the **/users.txt** file to the managed hosts.

```
$ ansible all -m command -a 'cat /users.txt'
```

```
[root@ansi-master ansible]# ansible all -m command -a 'cat /users.txt'
ansi-node3 | CHANGED | rc=0 >>
This line was added by lineinfile
# BEGIN ANSIBLE MANAGED BLOCK
This block of text consists of two lines
They have been added by blockinfile module
# END ANSIBLE MANAGED BLOCK
ansi-node2 | CHANGED | rc=0 >>
This line was added by lineinfile
# BEGIN ANSIBLE MANAGED BLOCK
This block of text consists of two lines
They have been added by blockinfile module
# END ANSIBLE MANAGED BLOCK
ansi-node1 | CHANGED | rc=0 >>
This line was added by lineinfile
# BEGIN ANSIBLE MANAGED BLOCK
This block of text consists of two lines
They have been added by blockinfile module
# END ANSIBLE MANAGED BLOCK
```

6. Create a playbook called **removefile.yml** in the current working directory. Configure the playbook to use the file module to remove the /users.txt file from all managed hosts.

```
$ cat > removefile.yml <<EOF
---
- name: Use the file module to remove a file
  hosts: all
  remote_user: root
  tasks:
    - name: Remove a file from managed hosts
      file:
        path: /users.txt
        state: absent
EOF
```

6.1 Run the ansible-playbook --syntax-check removefile.yml command to verify its syntax and correct any errors.

```
$ ansible-playbook --syntax-check removefile.yml
```

Output:

```
[root@ansi-master ansible]# ansible-playbook --syntax-check removefile.yml

playbook: removefile.yml
```

6.2 Run ansible-playbook addblock.yml to execute the playbook

```
$ ansible-playbook  removefile.yml
```

Output:

```
[root@ansi-master ansible]# ansible-playbook  removefile.yml

PLAY [Use the file module to remove a file] ************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [ansi-node2]
ok: [ansi-node1]
ok: [ansi-node3]

TASK [Remove a file from managed hosts] ****************************************************************
changed: [ansi-node1]
changed: [ansi-node3]
changed: [ansi-node2]

PLAY RECAP *********************************************************************************************
ansi-node1                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node2                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi-node3                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

6.3 Use an ad hoc command to execute **ls -**l command to confirm that the **users.txt** file no longer exists on the managed host.

```
$ ansible all -m command -a 'ls -l'
```

Output:

```
[root@ansi-master ansible]# ansible all -m command -a 'ls -l'
ansi-node2 | CHANGED | rc=0 >>
total 8
-rw-------. 1 root root 1367 Nov  4  2020 anaconda-ks.cfg
drwxr-xr-x. 2 root root    6 Nov  4  2020 Desktop
drwxr-xr-x. 2 root root    6 Nov  4  2020 Documents
drwxr-xr-x. 2 root root    6 Nov  4  2020 Downloads
-rw-r--r--. 1 root root 1522 Nov  4  2020 initial-setup-ks.cfg
drwxr-xr-x. 2 root root    6 Nov  4  2020 Music
drwxr-xr-x. 2 root root    6 Nov  4  2020 Pictures
drwxr-xr-x. 2 root root    6 Nov  4  2020 Public
drwxr-xr-x. 2 root root    6 Nov  4  2020 Templates
drwxr-xr-x. 2 root root    6 Nov  4  2020 Videos
ansi-node1 | CHANGED | rc=0 >>
total 8
-rw-------. 1 root root 1367 Nov  4  2020 anaconda-ks.cfg
drwxr-xr-x. 2 root root    6 Nov  4  2020 Desktop
drwxr-xr-x. 2 root root    6 Nov  4  2020 Documents
drwxr-xr-x. 2 root root    6 Nov  4  2020 Downloads
-rw-r--r--. 1 root root 1522 Nov  4  2020 initial-setup-ks.cfg
drwxr-xr-x. 2 root root    6 Nov  4  2020 Music
drwxr-xr-x. 2 root root    6 Nov  4  2020 Pictures
drwxr-xr-x. 2 root root    6 Nov  4  2020 Public
drwxr-xr-x. 2 root root    6 Nov  4  2020 Templates
drwxr-xr-x. 2 root root    6 Nov  4  2020 Videos
ansi-node3 | CHANGED | rc=0 >>
total 8
-rw-------. 1 root root 1367 Nov  4  2020 anaconda-ks.cfg
drwxr-xr-x. 2 root root    6 Nov  4  2020 Desktop
drwxr-xr-x. 2 root root    6 Nov  4  2020 Documents
drwxr-xr-x. 2 root root    6 Nov  4  2020 Downloads
-rw-r--r--. 1 root root 1522 Nov  4  2020 initial-setup-ks.cfg
drwxr-xr-x. 2 root root    6 Nov  4  2020 Music
drwxr-xr-x. 2 root root    6 Nov  4  2020 Pictures
drwxr-xr-x. 2 root root    6 Nov  4  2020 Public
drwxr-xr-x. 2 root root    6 Nov  4  2020 Templates
drwxr-xr-x. 2 root root    6 Nov  4  2020 Videos
```