File Permissions Management with Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Session - 29 Agenda:
1. File Permissions Management with Ansible

Managing File Permissions in Linux:

Managing the Linux file permissions may be classified into following three categories:

1. Standard File Permissions
2. Advanced File Permissions
3. Access Control Lists (ACL)

1. Standard File Permissions:

Here we deal with the file ownership, group ownership and permissions for the owner, group & other by using the commands like chown, chgrp & chmod.

```
$ cd /tmp
$ echo "Nehra Classes Are Awesome." > nehraclasses.txt
$ ls -lh nehraclasses.txt
$ sudo chown root nehraclasses.txt
$ ls -lh nehraclasses.txt
$ sudo chgrp root nehraclasses.txt
$ ls -lh nehraclasses.txt
$ sudo chown vikasnehra:vikasnehra nehraclasses.txt
$ ls -lh nehraclasses.txt
$ chmod 0666 nehraclasses.txt
$ ls -lh nehraclasses.txt
$ chmod o-rw nehraclasses.txt
$ ls -lh nehraclasses.txt
$ chmod u=rwx,g=rw,o=r nehraclasses.txt
$ ls -lh nehraclasses.txt
$ chmod 0000 nehraclasses.txt
$ ls -lh nehraclasses.txt
$ chmod a+w nehraclasses.txt
$ ls -lh nehraclasses.txt
$ chmod +x nehraclasses.txt
$ ls -lh nehraclasses.txt
```

2. Advanced File Permissions:

Here we deal with sticky bit, sgid bit & suid bit by using the chmod command.

Sticky Bit:

```
$ sudo mkdir /project
$ sudo chmod 0777 /project
$ ls -ld /project
$ sudo useradd -m -p $(openssl passwd redhat) john
$ su - john
$ touch /project/john.txt
$ chmod 0666 /project/john.txt
$ ls -lh /project/john.txt
$ exit
$ ls -lh /project/john.txt
$ rm /project/john.txt
$ cd /project/ ; ll
$ sudo chmod +t /project
$ ls -ld /project
$ sudo chmod -t /project
$ ls -ld /project
$ sudo chmod 1777 /project
$ ls -ld /project
$ su - john
$ touch /project/john.txt
$ chmod 0666 /project/john.txt
$ ls -lh /project/john.txt
$ exit
$ ls -lh /project/john.txt
$ rm /project/john.txt
```

SGID Bit:

```
$ ls -lh /project/john.txt
$ sudo groupadd project
$ ls -ld /project
$ sudo chown root:project /project
$ ls -ld /project
$ sudo chmod g+s /project
$ ls -ld /project
$ sudo chmod g-s /project
$ ls -ld /project
$ sudo chmod 2777 /project
$ ls -ld /project
$ su - john
$ touch /project/john2.txt
$ ls -lh /project/john2.txt
$ exit
$ touch /project/vikas.txt
$ ls -lh /project/vikas.txt
```

SUID Bit:

```
$ ls -lh /etc/shadow
$ ls -lh /usr/bin/passwd
$ ls -lh /project/vikas.txt
$ chmod u+s /project/vikas.txt
$ ls -lh /project/vikas.txt
$ chmod u-s /project/vikas.txt
$ ls -lh /project/vikas.txt
$ chmod 4660 /project/vikas.txt
$ ls -lh /project/vikas.txt
```

3. Access Control Lists (ACL):

Sometime standard and advanced file permissions are not enough to provide the access to any user or group so there we use ACL by using setfacl command.

```
$ ls -lh /project/vikas.txt
$ sudo chown vikasnehra:vikasnehra /project/vikas.txt
$ getfacl /project/vikas.txt
$ setfacl -m u:john:7 /project/vikas.txt
$ getfacl /project/vikas.txt
$ setfacl -m g:project:7 /project/vikas.txt
$ getfacl /project/vikas.txt
$ setfacl -x john /project/vikas.txt
$ getfacl /project/vikas.txt
$ setfacl -b /project/vikas.txt
$ getfacl /project/vikas.txt
```

Managing Linux File Permissions Using Ansible:

We can easily manage Linux file permissions with Ansible by using either Ansible Ad-hoc commands or by using Ansible playbooks.

Managing Linux File Permissions Using Ansible Ad-hoc Commands:

Let’s try managing the file permissions by running the above command as Ansible Ad-hoc commands on the Ansible managed node.

```
$ ansible node1 -m command -a 'touch /tmp/ansible.txt'
$ ansible node1 -m command -a 'ls -lh /tmp/ansible.txt'
$ ansible node1 -m command -a 'sudo chown root:root /tmp/ansible.txt'
$ ansible node1 -m command -a 'ls -lh /tmp/ansible.txt'
$ ansible node1 -m command -a 'sudo chmod 0666 /tmp/ansible.txt'
$ ansible node1 -m command -a ‘ls -lh /tmp/ansible.txt’
$ ansible node1 -m command -a 'sudo chmod u+s /tmp/ansible.txt'
$ ansible node1 -m command -a ‘ls -lh /tmp/ansible.txt’
$ ansible node1 -m command -a 'sudo groupadd linux'
$ ansible node1 -m command -a 'sudo setfacl -m g:linux:7 /tmp/ansible.txt'
$ ansible node1 -m command -a 'getfacl /tmp/ansible.txt'
$ ansible node1 -m command -a 'ls -lh /tmp/ansible.txt'
$ ansible node1 -m command -a 'sudo ln /tmp/ansible.txt /tmp/ansible_link.txt'
$ ansible node1 -m command -a 'ls -lhi /tmp/ansible_link.txt'
$ ansible node1 -m command -a 'sudo rm /tmp/ansible_link.txt'
$ ansible node1 -m command -a 'sudo ln -s /tmp/ansible.txt /tmp/ansible_link.txt'
$ ansible node1 -m command -a 'ls -lhi /tmp/ansible_link.txt'
$ ansible node1 -m command -a 'sudo rm /tmp/ansible_link.txt'
$ ansible node1 -m command -a 'mkdir /tmp/ansible'
$ ansible node1 -m command -a 'ls -ld /tmp/ansible'
$ ansible node1 -m command -a 'sudo chmod +t /tmp/ansible'
$ ansible node1 -m command -a 'ls -ld /tmp/ansible'
$ ansible node1 -m command -a 'sudo chmod g+s /tmp/ansible'
$ ansible node1 -m command -a 'ls -ld /tmp/ansible'
```

Managing Linux File Permissions Using Ansible Playbooks:

Let’s create an ansible playbook to manage the file permissions on the managed nodes.

```
$ vim permissions.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
    - name: Create a new file with permissions
      file:
        path: "/tmp/test.txt"
        state: touch
        mode: 0755
        owner: vikasnehra
      register: task1
    - name: Printing the result of task1.
      debug:
        msg: "{{ task1 }}"
...

$ ansible-playbook permissions.yml
$ ansible node1 -m command -a 'ls -lh /tmp/test.txt'

$ vim permissions2.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
       - name: Create a directory with the owner & group owner as root having the name as
/work & sticky bit set with full permissions for everyone.
      file:
        path: /work
        owner: root
        group: root
        mode: '1777'
        state: directory
...

$ ansible-playbook permissions2.yml
$ ansible node1 -m command -a 'ls -ld /work'

$ vim permissions3.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
   - name: Create a symbolic link
```

```
     file:
        src: /etc/redhat-release
        dest: /tmp/link
        owner: vikasnehra
        group: vikasnehra
        state: link
...

$ ansible-playbook permissions3.yml
$ ansible node1 -m command -a 'ls -lhi /tmp/link'
$ ansible node1 -m command -a 'ls -lhi /etc/redhat-release'

$ vim permissions4.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
   - name: Create hard link
     file:
       src: '/etc/redhat-release'
       dest: '/tmp/hardlink'
       state: hard
...

$ ansible-playbook permissions4.yml
$ ansible node1 -m command -a 'ls -lhi /tmp/hardlink'
$ ansible node1 -m command -a 'ls -lhi /etc/redhat-release'

$ vim permissions5.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
    - name: Touch a file, using symbolic modes to set the permissions (equivalent to 0644)
      file:
        path: /etc/linux.conf
        state: touch
        mode: u=rw,g=r,o=r
...

$ ansible-playbook permissions5.yml
$ ansible node1 -m command -a 'ls -lh /etc/linux.conf'

$ vim permissions6.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
    - name: Touch the same file, but add/remove some permissions
      file:
        path: /etc/linux.conf
        state: touch
        mode: u+rw,g-wx,o-rwx
...

$ ansible-playbook permissions6.yml
$ ansible node1 -m command -a 'ls -lh /etc/linux.conf'

$ vim permissions7.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
    - name: Touch again the same file, but do not change times this makes the task idempotent
      file:
        path: /etc/linux.conf
        state: touch
        mode: u+rw,g-wx,o-rwx
        modification_time: preserve
        access_time: preserve
...

$ ansible-playbook permissions7.yml
$ ansible node1 -m command -a 'ls -lh /etc/linux.conf'

$ vim permissions8.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
    - name: Create a directory if it does not exist
      file:
          path: /etc/linux
          state: directory
          mode: '0755'
...

$ ansible-playbook permissions8.yml
$ ansible node1 -m command -a 'ls -ld /etc/linux'

$ vim permissions9.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
    - name: Update modification and access time of given file
      file:
         path: /etc/linux
         state: directory
         modification_time: now
         access_time: now
...

$ ansible-playbook permissions9.yml
$ ansible node1 -m command -a 'ls -ld /etc/linux'

$ vim permissions10.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
   - name: Change the access time of a file.
     file:
       path: /etc/linux.conf
       state: file
       access_time: '202011121015.20'
...

$ ansible-playbook permissions10.yml
$ ansible node1 -m command -a 'ls -lu /etc/linux.conf'

$ vim permissions11.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
     - name: Create a directory as vikasnehra under /etc/linux/
       file:
         path: /etc/linux/vikasnehra
         state: directory
       register: result

     - name: Printing the result of the previous task
       debug:
        msg: "{{ result }}"

     - name: Recursively change ownership of the vikasnehra directory
       file:
         path: /etc/linux
         state: directory
         recurse: yes
         owner: vikasnehra
         group: vikasnehra
...

$ ansible-playbook permissions11.yml
$ ansible node1 -m command -a 'ls -ld /etc/linux/'
$ ansible node1 -m command -a 'ls -ld /etc/linux/vikasnehra'

$ vim permissions12.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
     - name: Remove file (delete file)
       file:
          path: /etc/linux.conf
          state: absent

     - name: Recursively remove directory
       file:
        path: /etc/linux
        state: absent
...

$ ansible-playbook permissions12.yml
$ ansible node1 -m command -a 'ls -lh /etc/linux.conf'
$ ansible node1 -m command -a 'ls -ld /etc/linux/vikasnehra'
$ ansible node1 -m command -a 'ls -ld /etc/linux/'

$ ansible node1 -m command -a 'getfacl /etc/redhat-release'
$ vim permissions13.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
    - name: Grant user vikasnehra read access to a file.
      ansible.posix.acl:
        path: /etc/redhat-release
        entity: vikasnehra
        etype: user
        permissions: r
        state: present
...

$ ansible-playbook permissions13.yml
$ ansible-galaxy collection install ansible.posix
$ ansible-playbook permissions13.yml
$ ansible node1 -m command -a 'getfacl /etc/redhat-release'
$ vim permissions14.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
    - name: Grant user vikasnehra read access to a file.
      ansible.posix.acl:
        path: /etc/redhat-release
        entity: vikasnehra
        etype: user
        state: absent
...

$ ansible-playbook permissions14.yml
$ ansible node1 -m command -a 'getfacl /etc/redhat-release'

$ ansible node1 -m command -a 'sudo mkdir /etc/linux'
$ vim permissions15.yml
---
- name: managing the file permissions using ansible.
  hosts: node1
  become: true
  tasks:
    - name: Same as previous but using entry short-hand
      acl:
         path: /etc/linux/
         entry: default:user:vikasnehra:rw-
         state: present
      register: acl_info

    - name: Obtain the ACL for a specific file
      debug:
        msg: "{{ acl_info }}"
...

$ ansible-playbook permissions15.yml
$ ansible node1 -m command -a 'getfacl /etc/linux/'
```

Thank You