

Chapter: 06 Deploying files to Managed hosts

⇒ Modifying & copying files to hosts

file modules library ⇒ linux file management ⇒ creating, copying, editing, modifying permissions & other attributes.

Blockinfile ⇒ insert, update or remove a block of multiline text surrounded by customizable marker lines.

Copy ⇒ Copy files from local/remote m/c to M.N.
sets file attributes

Fetch ⇒ Reverse of copy; Used to copy like file tree.
file ⇒ Create, permissions, ownership, SELinux contexts, time stamps, remove, other attributes.

Lineinfile ⇒ Check or replace single line in file

stat ⇒ Retrieve files status information

synchronize ⇒ Similar to 'rsync' but with less power.

- name: Touch a file and set permissions
file:

path: /path/to/file

owner: xyz

group: group1

mode: 0640

state: touch

⇒ works like touch command

setype: samba_share_t

⇒ Change selinux content of file; but works like chcon.

Chcon can be undone by restorecon command. Thus to make it persistent along with file module to set the context;

- name: SELinux persistent

setcontext:

target: /path/to/samba_file

setype: samba_share_t

state: present

• setcontext does not change the context of existing files.

✎ Copy

Copies files/directories to managed node.

by default 'force: yes' i.e. it will overwrite the file if there is any.

it can be set to 'force: no'; it will copy the file only if it is not initially present there?

- name: Copy a file to managed hosts

copy:

src: file

dest: /path/to/file

✎ fetch

To retrieve files from the managed hosts.

- name: fetching file from M.N

fetch:

src: # /home/users/.ssh

dest: /path/to/file

'flat: yes' parameter
does not copy whole tree

Lineinfile

- name: Using lineinfile to add a line

lineinfile:

path: /path/to/file

line: 'Add this line'

state: present

Blockinfile

- name: Adding additional lines in a file

blockinfile:

path: /path/to/file

block: |

first line

second line

third line

state: present

The output have starting & ending commented marker to ensure idempotency.

Use 'marker' parameter on the module to edit these markers

Removing a file

state: absent parameter

Retrieving status of a file on managed hosts.

- name: verifying the file status
stat:

path: /path/to/file

register: result

- name: print stat

debug:

msg: var: result.

Synchronize

→ rsync tool should be installed on C.N & M.N.
Synchronize task originates on C.N & synchronize connects to M.N.

- name: Using synchronize
synchronize:

src: file

dest: /path/to/file.

Jinja2 TEMPLATES

— Ansible uses the jinja2 template files — Ansible also uses jinja2 syntax to reference variables in playbook

{% EXPR %} → Expressions or logic

{{ EXPR }} → Result of expression or variable

{# COMMENT #} → Comment that should not appear in the final file.

```
{# /etc/hosts line #}
{{ ansible_facts['default_ipv4']['address'] }}
```

The variables used in the template can be specified in the vars section of the playbook.

A template file does not need to have '.j2' extension. It is only for understand \rightarrow good documentation purpose.

To use template module:

tasks:

- name: template sender

template:

src: /tmp/j2-template.j2

dest: /tmp/dest-config-file.txt

Template module also have other file attributes, options. Check for 'validate' parameter, which checks the syntax of a file for correctness before copying it into place.

To avoid system admin to edit lines which were modified by ansible it is better to put a comment for that variable can be used in template.

\Rightarrow inside ansible.cfg
 ansible_managed = Ansible Managed.

\Rightarrow inside Template.j2
 {{ ansible_managed }}

Expressions or logic.

for statement \Rightarrow looping functionality.

```
{% for user in users %}
  {{ user }}
{% endfor %}
```

The 'user' variable is replaced with all the values included in the 'users' variable, one value per line.

{% for statement %}

```
{% for myuser in users if not myuser == "root" %}
  User number {{ loop.index }} - {{ myuser }}
{% endfor %}
```

loop.index \Rightarrow variable \Rightarrow index number \Rightarrow 1 2 3 ...

```
{% for myhost in groups['myhosts'] %}
  {{ myhost }}
{% endfor %}
```

myhosts is a group in inventory.

```
{% for host in groups['all'] %}
  {{ hostvars[host]['ansible_facts']['default_ipv4']['address'] }}
{% endfor %}
```

for collecting ansible facts \Rightarrow using them.

if statement

This allows you to put a line in a deployed file if certain conditions are met.

```
{% if finished %}
```

```
{% result %}
```

```
{% endif %}
```

The 'result' variable is placed in deployed file, only if the

value of 'finished' variable is true

Jinja2 provides filters to change output format for template expressions to YAML & JSON language.

```
{{ output | to_json }}
```

⇒ to_json filter formats the expression output using JSON

```
{{ output | to_yaml }}
```

⇒ to_yaml ——— using YAML

```
{{ output | to_nice_json }}
```

```
{{ output | to_nice_yaml }}
```

⇒ formats the expression to JSON or YAML.

```
{{ output | from_json }}
```

```
{{ output | from_yaml }}
```

⇒ expects strings in JSON or YAML to parse them.

Read about variable Tests???