Ansible Control Node, Inventory & Managed Nodes

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

Session - 5 Agenda:

1. Ansible Ad-Hoc Commands
2. Ansible Modules

Important Ansible Module Names:

1. *Copy*
2. *Command*
3. *Raw*
4. *Shell*
5. *File*
6. *Fetch*
7. *Get_url*
8. *Lineinfile*
9. *Replace*
10. *User*
11. *Group*
12. *Yum/Dnf/Apt*
13. *Package*
14. *Yum_repository*
15. *Stat*
16. *Mount*
17. *Setup*
18. *Service*
19. *Systemd*
20. *Debug*
21. *Uri*
22. *Parted*
23. *Cron*
24. *Script*

Quick Recap of Session – 4:

Copy Module:

Create a file nehraclasses.txt with content and copy it to /tmp directory in the managed nodes.

$ ansible all -m copy -a 'content="Nehra Classes Are Awesome." dest=/tmp/nehraclasses.txt'

Verify the file contents.
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

Now, if we put some other content in this file, it will get overwritten.
$ ansible all -m copy -a 'content="Welcome To Nehra Classes." dest=/tmp/nehraclasses.txt'

Verify the file contents.
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

Let's create a file with different contents.
$ vim nehraclasses.txt
NEHRACLASSES
We can also take backup of the existing file while overwriting it.
$ ansible all -m copy -a 'src=nehraclasses.txt dest=/tmp/nehraclasses.txt backup=yes'

Verify the backup files presence in the /tmp directory of node machines & their contents.
$ ansible all -m command -a 'ls -lh /tmp/'
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

---

8. Lineinfile Module: (This module ensures a particular line is in a file, or replace an existing line using a back-referenced regular expression.)
Let's add a line in an existing file on the managed nodes from the ansible control node.
By default, lineinfile module appends the line in any file.
$ ansible all -m lineinfile -a 'dest=/tmp/nehraclasses.txt line="This server is managed by Ansible."'

Verify the changes.
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

If you want to add a line at the beginning of the file.
$ ansible all -m lineinfile -a 'dest=/tmp/nehraclasses.txt line="Hi Everyone." insertafter=BOF'

Verify the changes.
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

If you want to add a line at the end of the file.
$ ansible all -m lineinfile -a 'dest=/tmp/nehraclasses.txt line="Thanks." insertafter=EOF'

Verify the changes.
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

If you want to add a line after a specific keyword in the file.
$ ansible all -m lineinfile -a 'dest=/tmp/nehraclasses.txt line="RHCE (EX294)." insertafter=This'

Verify the changes.
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

Let's remove a file from the file on the managed nodes using absent argument option.
$ ansible all -m lineinfile -a 'dest=/tmp/nehraclasses.txt line="RHCE (EX294)." state=absent'

Verify the changes.
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

Let's use regular expression to remove a line from the file on the managed nodes.
$ ansible all -m lineinfile -a 'dest=/tmp/nehraclasses.txt regexp=^This state=absent'

Verify the changes.
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

---

9. Replace Module: (This module will replace all instances of a pattern within a file.)

Let's replace a string with other in a file on the managed nodes using regular expressions.
$ ansible all -m replace -a 'dest=/tmp/nehraclasses.txt regexp=^Hi replace=Hello'

Verify the changes.
$ ansible all -m command -a 'cat /tmp/nehraclasses.txt'

---

10. User Module: (Manage user accounts and user attributes.) It is used for the same job that we perform using the commands like useradd, usermod, userdel etc.

Let's create a user named amit having sudo (admin) access/member of wheel group in all the managed nodes.
$ ansible all -m user -a 'name=amit state=present uid=1010 groups=wheel'
Groups: for secondary group(s)
Group: for primary group

Verify the changes.
$ ansible all -m command -a 'id -a amit'
OR
$ ansible all -m command -a 'tail -1 /etc/passwd'

To remove the user:
$ ansible all -m user -a 'name=amit state=absent'

Verify the changes.
$ ansible all -m command -a 'id -a amit'

---

11. Group Module: (Manage presence of groups on a host.) It is used for the same job that we perform using the commands like groupadd, groupmod, groupdel etc.
Let's create a group named staff having gid=1015 in all the managed nodes.
$ ansible all -m group -a 'name=staff state=present gid=1015'

Verify the changes.
$ ansible all -m command -a 'tail -1 /etc/group'

To remove the group:
$ ansible all -m group -a 'name=staff state=absent'

Verify the changes.

$ ansible all -m command -a 'grep staff /etc/group'

---

12. Yum/Dnf/Apt Module: (Installs, upgrade, downgrades, removes, and lists packages and groups with the `yum' package manager.) It is used to perform the jobs on the managed nodes that we perform on a machine using yum/dnf/apt commands.
On Debian based distros we use **apt module** for this task.
We can use **package module** which works on all linux distros.

Let's install a package e.g., zsh on the Fedora based managed nodes using yum module.
$ ansible all -m yum -a 'name=zsh state=present'
It requires subscription or yum repository already present on your node machines.
Here present or installed have the same meaning, similarly absent or removed have the same meaning. Latest means updated version of the packages need to be installed.

Verify the changes.
$ ansible all -m command -a 'rpm -qi zsh'

On Debian based nodes:
$ ansible Ubuntu -m apt -a 'name=zsh state=present'

Verify the changes.
$ ansible Ubuntu -m command -a 'apt list zsh'

---

13. Package Module:
Let's install ksh package on all the Fedora based managed nodes using dnf module.
$ ansible RHEL9 -m package -a 'name=ksh state=installed use=dnf'

Verify the changes.
$ ansible RHEL9 -m command -a 'rpm -qi ksh'

Let's create yum repository on the managed nodes.

Create a repo file first.
$ vim localrepo.repo

```
[Local-BaseOS]
name=Red Hat Enterprise Linux 9 - BaseOS
metadata_expire=-1
gpgcheck=1
enabled=1
baseurl=file:///mnt/BaseOS/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[Local-AppStream]
name=Red Hat Enterprise Linux 9 - AppStream
metadata_expire=-1
gpgcheck=1
enabled=1
baseurl=file:///mnt/AppStream/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

Copy this file all the managed nodes under /etc/yum.repos.d/ directory.
$ ansible RHEL9 -m copy -a 'src=localrepo.repo dest=/etc/yum.repos.d/'

Verify the changes.
$ ansible RHEL9 -m command -a 'yum repolist all'

Same work can also be done using yum_repository module.

---

14. Yum_repository: (Add or remove YUM repositories in RPM-based Linux distributions.)
$ ansible-doc yum_repository

Create yum repository using yum_repository module.
$ ansible all -m yum_repository -a 'name=test description="test repo" baseurl=file:///mnt/AppStream enabled=1 gpgcheck=0'

Verify the changes.
$ ansible all -m command -a 'yum repolist all'
OR
$ ansible all -m command -a 'ls -lh /etc/yum.repos.d/test.repo'

$ ansible all -m command -a 'cat /etc/yum.repos.d/test.repo'

To remove the repository:
$ ansible all -m yum_repository -a 'name=test state=absent'

OR use file module to remove the file.

Verify the changes.
$ ansible all -m command -a 'yum repolist all'
OR
$ ansible all -m command -a 'ls -lh /etc/yum.repos.d/test.repo'

Thanks