Ad-hoc commands

➢ AD HOC COMMANDS:

- ad hoc command uses the **/usr/bin/ansible command-line tool** to automate a single task on one or more managed nodes. These are quick and easy, but they are not reusable.
- ad hoc commands are great for tasks you **repeat rarely.**
  E.g.: To power off all the machines, you could execute a quick one-liner in Ansible without a playbook.
- It is used to **reboot servers, copy files, manage packages** and **users,** and much more.

$ansible [pattern] -m [module] -a "[module options]"

Here:

-m: Module
-a : Module Option

ANSIBLE MODULE:

- A module is a reusable, standalone script that Ansible runs on your behalf, either locally or remotely.
- Modules interact with your local machine, an API, or a remote system to perform specific tasks like changing a database password or spinning up a cloud instance.

$ansible --list-hosts webservers : List webservers group

To list all Modules:

$ansible-doc -l

$ansible-doc yum [Details of yum module]

$ansible -m ping webservers / $ansible -m ping all [Testing Environment]

❖ USE CASES FOR AD HOC TASKS:

REBOOT SERVERS:

- The default module for the ansible command-line utility is the **ansible.builtin.command** module. You can use an ad hoc task to call the command module and reboot all web servers at a time.

  $ansible webservers -a "/sbin/reboot"

  $ansible webservers -a "/sbin/reboot" -f 10 [By default Ansible uses only 5 simultaneous processes.To reboot the webservers with 10 parallel forks]

  $ansible webservers -a "/sbin/reboot" -f 10 -u username --become --ask become-pass

  (or)

  -K [Rebooting probably requires privilege escalation]

MANAGING FILES:

- An ad hoc task can harness the power of Ansible and SCP to transfer many files to multiple machines in parallel.

  To transfer a file directly to all servers in the [webservers] group:

  $ansible webservers -m ansible.builtin.file -a "dest=/tmp/java mode=600 state=touch"

  $ansible webservers -m ansible.builtin.copy -a "src=/opt/script.sh dest=/tmp/"

  $ ansible webservers -m ansible.builtin.file -a "dest=/tmp/ mode=600 owner=raju group=developers"

  $ansible webservers -m ansible.builtin.file -a "dest=/world/asia/india/ap/vskp mode=755 owner=raju group=developers state=directory"

  $ansible webservers -m ansible.builtin.file -a "dest=/world/asia/india/ap/vskp state=absent" [Delete a directory]

  $ansible webservers -a "free -m" [To check Ram size]

$ansible webservers -a "df -h"

SHELL MODULE:

$ansible webservers -m shell -a "cat /etc/passwd|grep -i raju" -b -K

$ansible webservers -m shell -a "cat /proc/meminfo|head -2"

MANAGING PACKAGES:

- Task to install, update, or remove packages on managed nodes using a package management module such as yum. Package management modules support common functions to install, remove, and generally manage packages.

  $ansible webservers -m ansible.builtin.yum -a "name=httpd state=present"

  ansible webservers -m ansible.builtin.yum -a "name=httpd state=present" --limit "*.4" [any node that ends with a .4 IP address.]

  $ansible webservers -m ansible.builtin.yum -a "name=httpd-2.4 state=present" --limit "webservers:!172.6.7.80”

  $ansible webservers -m ansible.builtin.yum -a "name=httpd state=absent"

MANAGING SERVICES:

- Ensure a service is started on all webservers:

  $ansible webservers -m ansible.builtin.service -a "name=httpd state=started"

  $ansible webservers -m ansible.builtin.service -a "name=httpd state=restarted"

  $ansible webservers -m ansible.builtin.service -a "name=httpd state=stopped"

MANAGING USERS AND GROUPS:

- You can create, manage, and remove user accounts on your managed nodes with ad hoc tasks:

  $ansible all -m ansible.builtin.user -a "name=jai password=<crypted password here>"

  $ansible all -m ansible.builtin.user -a "name=foo state=absent"

GATHERING FACTS:

- Facts represent discovered variables about a system. You can use facts to implement conditional execution of tasks but also just to get ad hoc information about your systems.
- To see all facts:

  $ansible all -m ansible.builtin.setup

CHECK MODE:

- In check mode, Ansible does not make any changes to remote systems. Ansible prints the commands only. It does not run the commands.
- Enabling check mode **(-C or --check)** in the bellow command means Ansible does not actually create or update the **/home/ram/script.sh** file on any remote systems.

  $ansible all -m copy -a "content=Hello dest=/home/ram/script.sh" -C