Ansible Adhoc commands Examples

==

- Ansible Ad-Hoc commands are used to accomplish tasks quickly.
- These commands are mostly used for one-off tasks.
- Every task in Playbooks can be done by using Ad-Hoc commands.

syntax:

ansible <host group> <module> <argument to the module>

Ex:

To check the communication of nodes

1)**ansible all -m ping**

Install the httpd on webserver group

2)**ansible webserver -m yum -a "install=httpd status=latest"**

Note: -

1)Linux to install packages use fallowing package manager

ubuntu-->apt

RedHat/Cent/Amazon---->yum

Amazon Linux----->yum/amazon-linux-extras (if yum command is not working then first check if the package is available in amazon-linux-extras --list if available then install with "amazon-linux-extras install <Your Package Name>"

MAC---->brew

2)package: Apache (web server)

ubuntu---->apache2

RedHat/Cent/Amazon--->httpd

3)state of the package install/uninstall

state: present----->install

state: absent----->uninstall

Ex:

ansible webserver -m yum -a "name=httpd state=latest" --become

-a //refers to the argument

-m //refers to the module

yum //refers to the module name

--become //refers to the sudo permission

Ex:

i want to find a file name as /opt/oracle that should be 10 days older and the extension should be '.log'

```
ansible all -m shell -a "find /opt/oracle -type f -mtime +10 -name '*.log'"
```

it will display the all the log file which are 10 days old in the dir. '/opt/oracle'

Ex:

i want to create user add the user into adm group and shell for user is /bin/bash

```
ansible all -m user -a "user=siva group=admin append=yes shell=/bin/bash"
```

Ex:

To restart httpd service

```
ansible all -b -m shell -a "systemctl restart httpd"
```

To check the status of the sshd service

```
ansible all -b -m shell -a "systemctl status sshd”
```

Ex:

1)Command to check connectivity of the managed node from control server

```
ansible managed node -m ping
```

2)Check managed node Uptime using commands

```
ansible all -m shell uptime
```

3)Check managed node date using commands

```
ansible managed node -m shell date
```

Note:

Ansible commands output colours:

red: --->Indicates no change done on the managed servers /it means that that the adhoc commands is not done any change/execution in the managed servers.

green: ---->Indicates changes done on the managed nodes // it means that the adhoc commands has done some changes/execution in the in the managed nodes and made some changes in the managed nodes

yellow: --->Indicates error in the command

Ex:

Check RAM on the managed node

```
ansible -m ping -a "free-h" managed node
```

Ex:

Check Disk Space on the managed node

```
ansible -m ping -a 'df-h' managed node
```

Ex:

Create user on the managed node

```
ansible -b -m shell -a "user=siva" managed node
```

Ex:

Create files on the managed node

```
ansible -b -m shell -a "touch /home/ec2-user/plain.txt" managed node
```

Ex:

Create file "devops_note.txt" on the managed node using support user ownership

Note: For sudo use --become

```
ansible -b -m shell -a "touch /home/support/devops_note.txt" --become-user=support managed node
```

Ex:

Create user "support" in managed node add the user into root group and create home directory and shell for this user is '/bin/bash'

```
ansible managed node -m user -a "name=support group=root createhome=yes append=yes shell=/bin/bash" --become-user=ubuntu
```

Ex:

command to install the httpd

```
ansible managed node -m yum -a "name=httpd state=latest" --become
```

Ex:

start and stop the services of the application in manged node

```
ansible -m service -a 'name=httpd state=started' -b all/appserver/webserver

ansible -m service -a 'name=httpd state=stopped' -b all/appserver/webserver
```

Ex:

Uninstall package on the managed node

```
ansible managed node -m yum -a "name=httpd* state=absent" --become
```

Ex:

Copy files from ansible server to managed node

```
ansible -m copy -a 'src=/home/ec2-user/plain.txt dest=/home/ec2-user/sample/' managed node
```

Ex:

Delete the files from the managed node

```
ansible managed node -m file -a 'dest=/home/ec2-user/plain.txt state=absent'
```

Ex:

Find Process consuming high memory on the managed node

```
ansible managed node -m shell -a "ps -eo pid,ppid,%mem,%cpu,cmd --sort=-%mem | head"
```

Ex:

Listing all the modules

```
ansible-doc -l
```

Ex:

Listing specific modules based on names

```
ansible-doc copy
ansible-doc file
ansible-doc yum
```

Ex:

To see the operating system configuration

```
ansible managed node -m setup
```