**Technote on Ansible basics & Patching through Ansible on Linux**

**ANSIBLE IT Automation TOOL & FEATURES**

* **Ansible is a free open source application**
* **Agent-less functionality**
* **Configuration of Servers**
* **Application Deployment**
* **Continuous testing of already installed application**
* **Automation of tasks, Provisioning and Orchestration**
* **Based on Python/YAML. There is no need to understand Python to use Ansible**
* **Tasks are written in codes, Eliminate Errors and Configuration inconsistency**
* **Highly flexible & Scalable**
* **Large number of ready to use modules for systems management**
* **Custom modules could be added if needed**
* **Configuration roll-back in case of error**
* **Simple and human readable, easy installation & configuration**
* **Self-documenting, Write once, Use multiple times**
* **Large number of ready to use modules for systems management**
* **Custom modules could be added if needed**
* **Simple and human readable, easy installation & configuration**

**ANSIBLE REQUIREMENTS:**

* **ANSIBLE package Avaibale on EPEL Repository**
* **SSh Client (OpenSSH), Password Less Authentication for non-root user (ansible) from Control System to Client Systems**
* **Sudo Previlliage: To execute all or selective commands without password prompt on client systems**
* **Python**

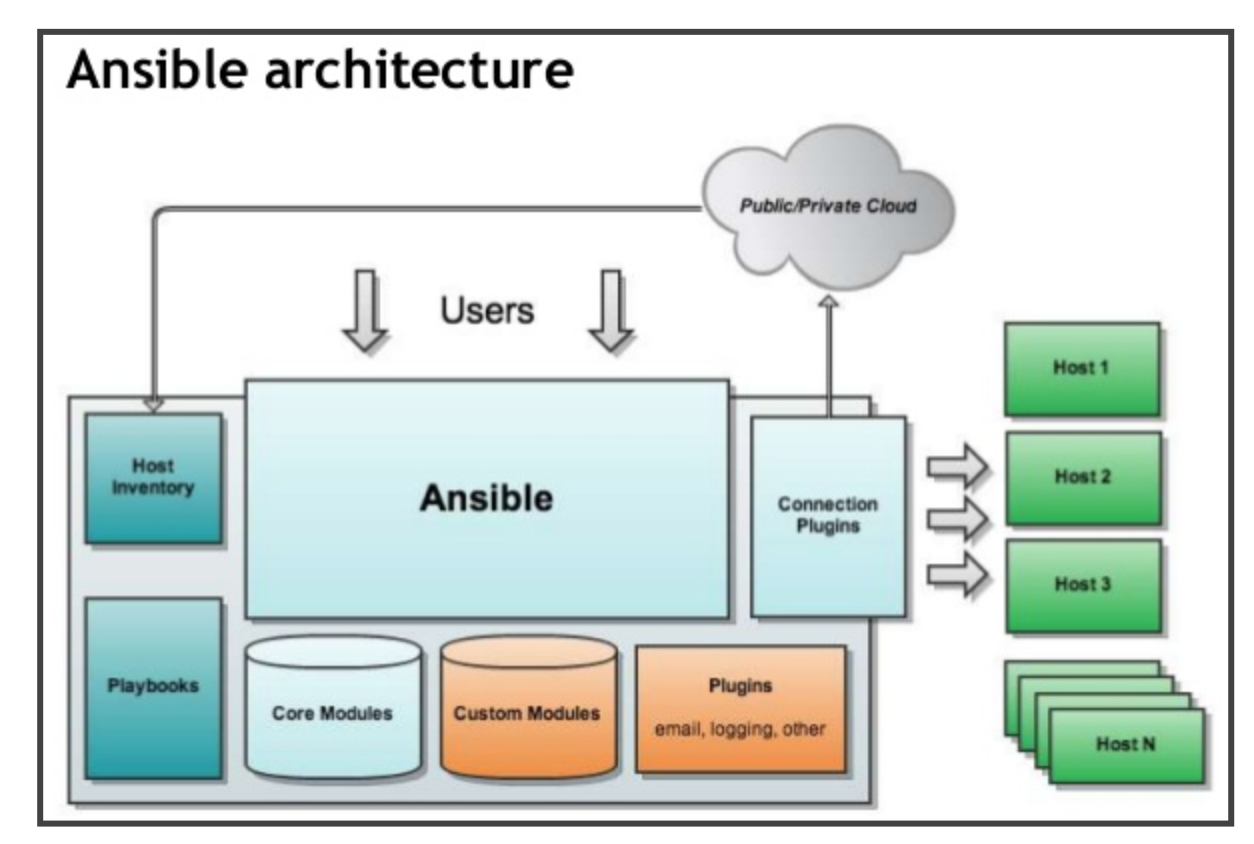
**ANSIBLE COMPONENTS**

* **Program: ansible adhoc commands & ansible playbook are Programs of Ansible**
* **Modules:** **Set of instruction for a particular type configuration management**
* **Modules perform configuration and system management examples Modules Copy, Service, File, Yum, user, Group**

**ANSIBLE PLAYBOOK**

* **In Ansible terms, play is like a task/command. A set of commands grouped together to serve a defined purpose can be termed as a playbook. A playbook written in YAML is very sensitive to “space” and “indentation”.**
* **YAML stands for Yet Another Markup Language. Ansible uses YAML to instruct the Automation Framework to perform tasks. YAML is very space sensitive. Hence, it will be better to setup indentation in text editors. Failing to do so will fail the syntax**

**ANSIBLE ARCHITECT**

****

**LAB SETUP**

**ANSIBLE CONTROLLER: RHEL7**

[root@ansible-controller ~]# cat /etc/redhat-release

CentOS Linux release 7.4.1708 (Core)

[root@ansible-controller ~]# rpm -qa|grep ansible

ansible-2.4.2.0-1.el7.noarch

[root@ansible-controller ~]#

**CLIENT SERVER: RHEL6**

[root@cman1 ~]# cat /etc/redhat-release

CentOS release 6.9 (Final)

[root@cman1 ~]#

--------------------------------------------------------------------------------------------------------------------------------------

**User “ansible” is created as ansible admin user (we can have of any name) with “/home/ansible” as home dir on Controller & across clients. Set for password less communication from Controller to all clients and permitted for sudo root access on clients**

**[root@ansible-controller ~]# ansible --version**

ansible 2.4.2.0

config file = /etc/ansible/ansible.cfg

configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /usr/bin/ansible

python version = 2.7.5 (default, Aug 4 2017, 00:39:18) [GCC 4.8.5 20150623 (Red Hat 4.8.5-16)]

**Ansible main config and binary files:**

/etc/ansible/ansible.cfg ## All ansible related configuration

/etc/ansible/hosts ### Client Inventory Groups

**Ansible binary/commands:**

/usr/bin/ansible #### Run a single task against a set of hosts

/usr/bin/ansible-playbook ## Runs Ansible playbooks, executes defined tasks on the targeted hosts

/usr/bin/ansible-doc ## displays information on modules installed in Ansible libraries.

/usr/bin/ansible-galaxy ## command to manage Ansible roles in shared repostories, the default of which is Ansible Galaxy https://galaxy.ansible.com

/usr/bin/ansible-vault ## encryption/decryption utility for Ansible data files

/usr/bin/ansible-pull ## pulls playbooks from a Version Control System(VCS) repository

/usr/bin/ansible-inventory ## used to display or dump the configured inventory as Ansible sees it

/usr/bin/ansible-config ## View, edit, and manage ansible configuration

/usr/bin/ansible-console ##REPL console, that allows for running ad-hoc tasks against a chosen inventory

**Invenstory file on ansible controller, Below as example**

[ansible@ansible-controller ~]$ cat /etc/ansible/hosts|grep -v "#"

[NONRH]

ubuntu

[APPSERVER]

appserver

[DBSERVER]

dbserver

[NEWGROUP]

cman1

**ansible adhoc commands: To achieve an task on remote machine**

**#examples adhoc commands**

#ansible-doc –l |more ###list all available modules

#ansible-doc –s yum ## List detail/options available with module

#ansible appgroup –m ping ## Check ping response for hosts under appgroup inventory grpup

#ansible all –m ping ### Check ping response for all serevrs in ansible Server inventory

#ansible all –m shell –a “uname –a;df –h” ## shell commands o/p

# ansible -m setup all ## Gathers facts about remote hosts

**##More examples of adhoc command here –s is used to allow ansible user to use sudo access on remote machine for root commands**

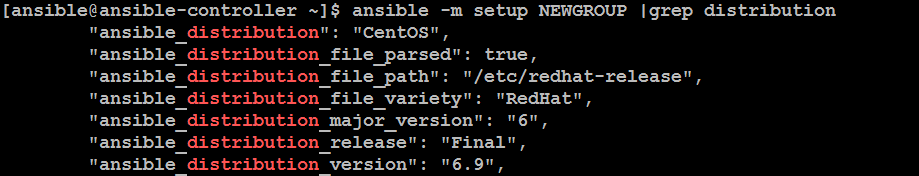
#ansible appgroup –m yum –a “name=httpd state =present” -s ## yum module to install pkg httpd

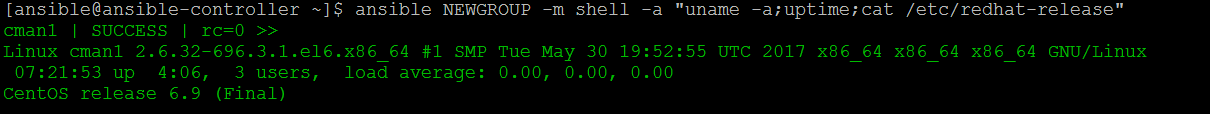
#ansible appgroup –m service –a “name=httpd state=started” -s # service module start httpd service

#ansible webgroup -m service -a "name=httpd enabled=yes" –s ## set service as enabled

#ansible appgroup –m copy –a “src=/tmp/file dest=/tmp/file” –s ##use of copy module

**##Below example is to check hosts information(OS Distribution) for Servers under NEWGROUP ansible Inventory**

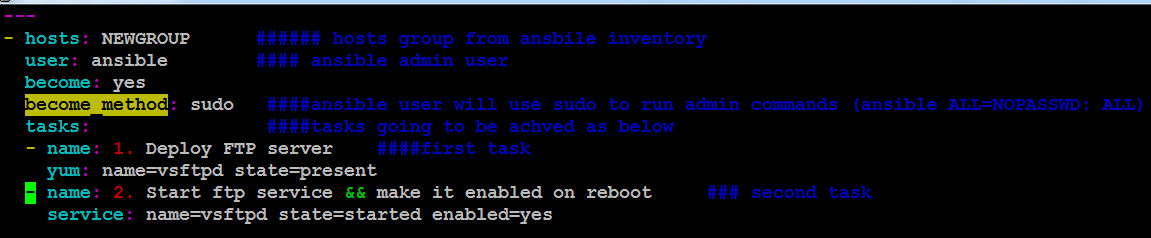


**####Example of shell Module: Here checked kernel version, uptime and OS Release of a client machine**

**Playbook in ansible: It is group of adhoc commands/tasks to be executed in scripted way.**

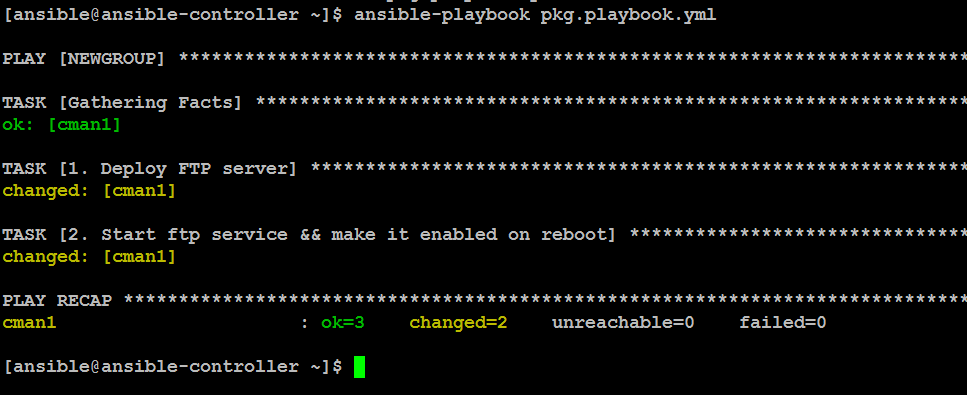
**#Example of a playbook: To install vsftpd pkg, start vsftpd service & make it persistent on Reboot**

1. **First task is to install vsftpd pkg**
2. **Second task is to start the Service & put chkconfig on**

**Pkg installation Playbook snapshot:**

**##Below o/p of running the above playbook**

**Here we could see changes made on Client on execution of both tasks and no task failed. Colour has significance like green colour no changes made and orange colour states changed made. From /etc/ansible/ansible.cfg we can customise the colours for o/p display**



**##Patching using Ansible**

* **Servers can be patched in Rolling fashion**
* **Diff Linux OS Distributions/Releases can be patched using a playbook**
* **Servers can be Patched in a Group**
* **Group can be of 2,3,4 …. Or all Servers in inventory**
* **Duration to get Server patched depends upon the network accessibility to the Patch Repository**
* **For Servers to be patched in Group must have Repository on internal Network**
* **Precheck & Post validation can be achieved by Ansible**

**####Example of Patching Playbook**

**1. First task is to verify if apps process stopped**

**2. Second task is to skip patching if apps process running**

**3. Third task is to start patching if application is stopped && OS Distribution is CentOS or RHEL**

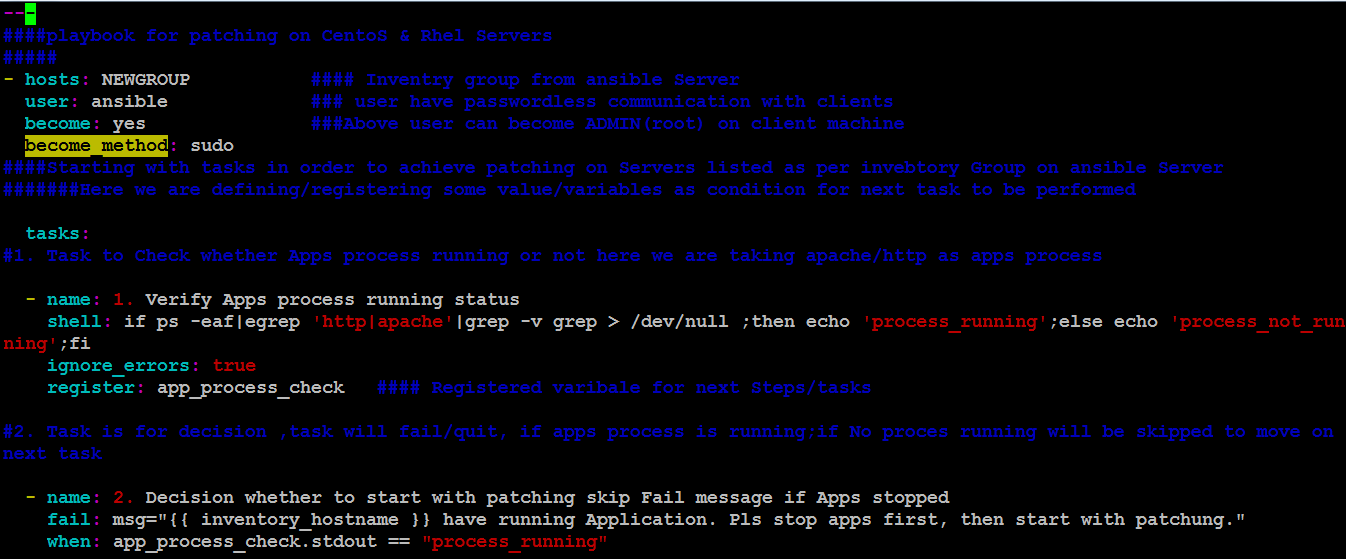
**4. Fourth task is for Server Reboot if kernel update has happened**

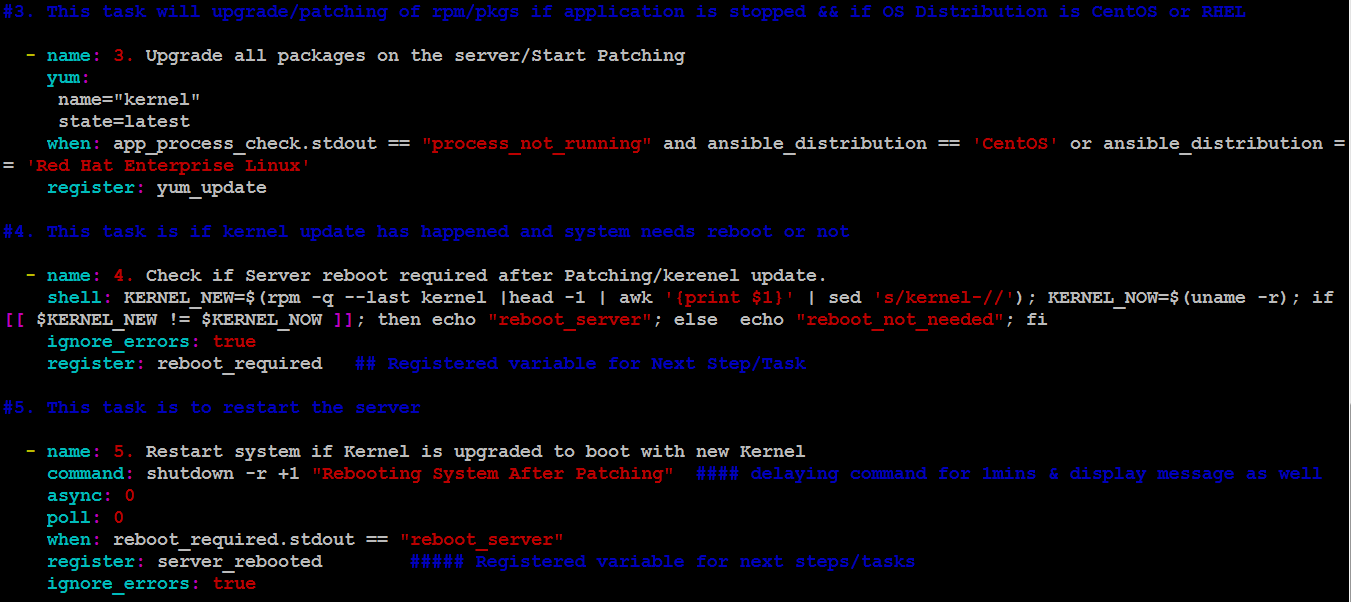
**5. Fifth task to restart the server if above condition is met**

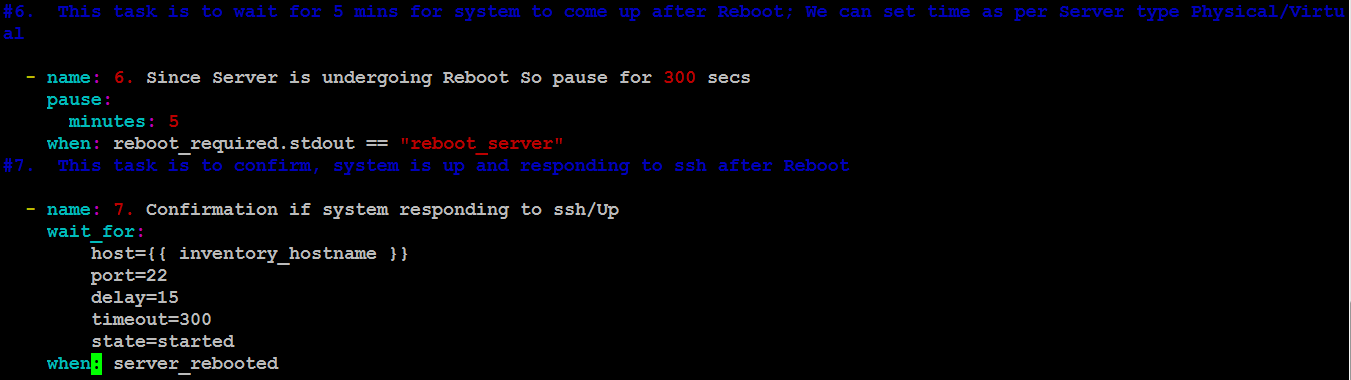
**6. Sixth task is to take pause till Server comes up after reboot**

**7. Seventh Task is to confirm system is up and responding to ssh after Reboot**

**====snapshots of patching playbook==============**







**####Starting with Patching Playbook####**

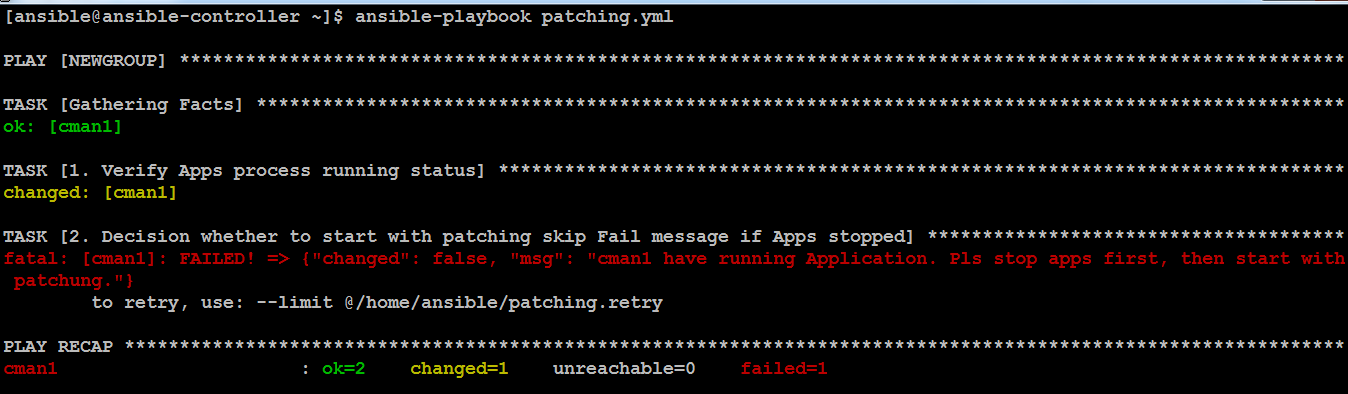
1. **Run patching Playbook when apps Process are running: It will through error after first task**

[root@cman1 ~]# ps -eaf|grep http

root 3401 1 0 13:57 ? 00:00:00 /usr/sbin/httpd

apache 3404 3401 0 13:57 ? 00:00:00 /usr/sbin/httpd

apache 3405 3401 0 13:57 ? 00:00:00 /usr/sbin/httpd



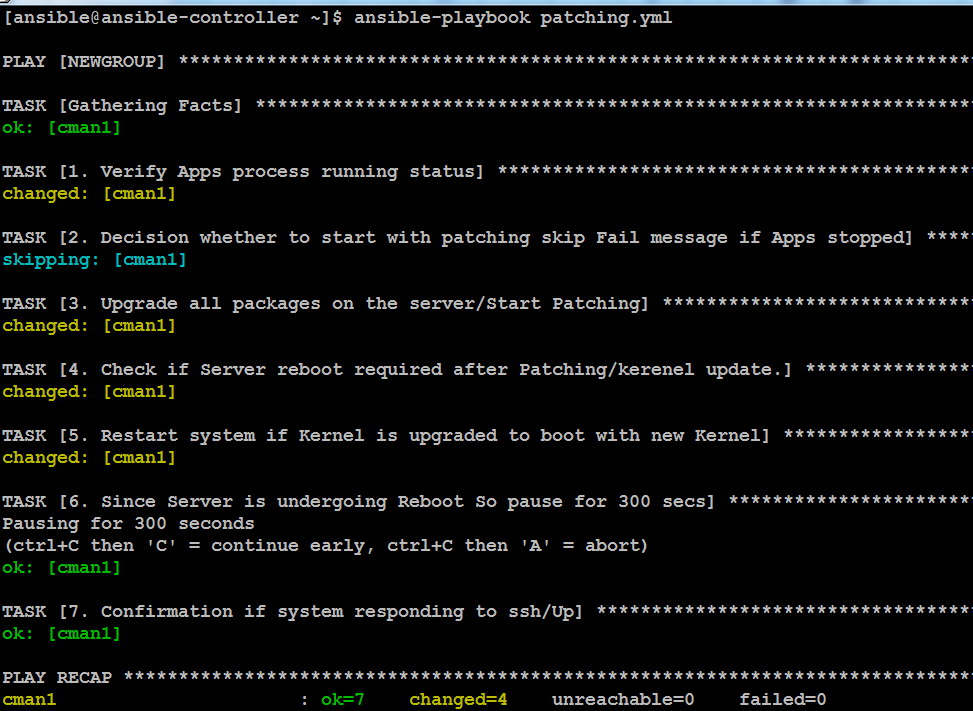
1. **Now Run patching playbook when apps are stopped: All tasks will be completed, wherever modifications are made o/p will be displayed as changed. “OK” will be displayed where no changes is made. Fail message in-case process running will be skipped here**

[root@cman1 ~]# service httpd stop

Stopping httpd: [ OK ]

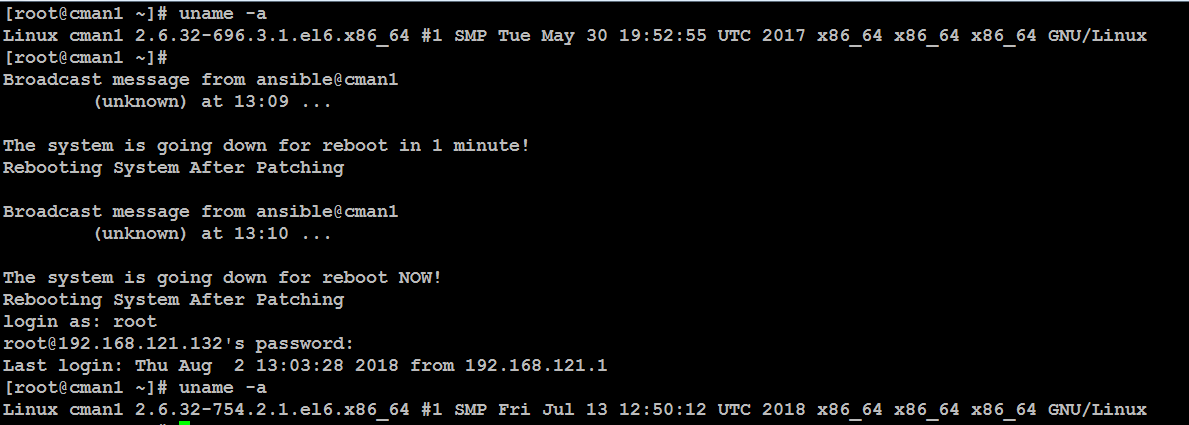
[root@cman1 ~]# ps -eaf|grep http

root 3662 3366 0 14:01 pts/0 00:00:00 grep http



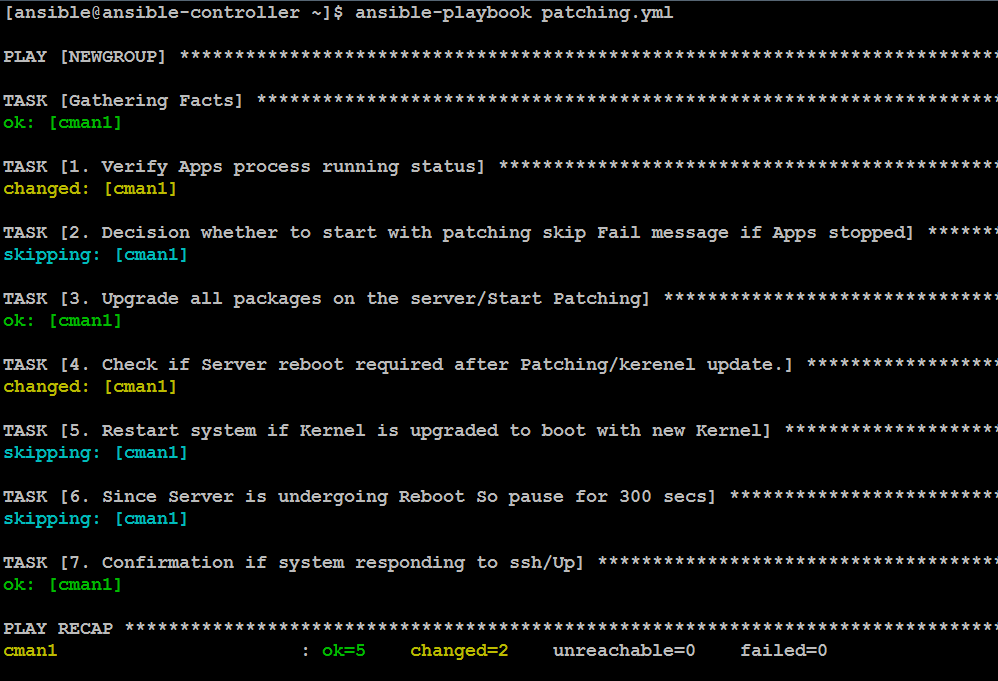
**Here you could see No task failed and and 4 tasks made changes with their action**

**Below on Client Server we could see Server going down after Patching and came up with upgraded Kernel Version**



**####Next starting patching playbook again on recent patched Server###**

**Since no Kernel/package will be upgraded Task of Server Reboot is skipped**



**Devinder Singh Lalotra**