Ansible Methods

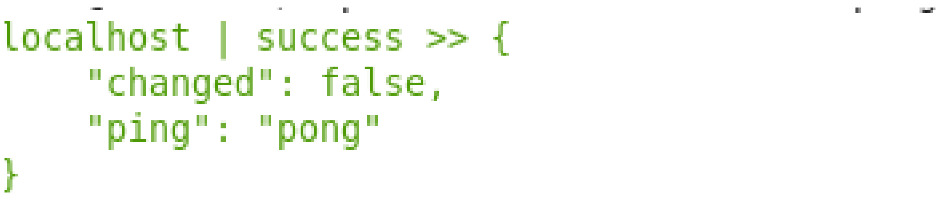
1. Checking Connectivity in every hosts

```
# ansible localhost -m ping
# ansible all -m ping
# ansible 127.0.0.1 -m ping
# ansible hosts -m ping
```

Here we can use hostname, IP , group name or all to define the hosts. 'all' means to every hosts and here 'hosts' means a group name which is given in configuration file /etc/ansible/hosts in that we can specify hosts
For example:
[hosts]
192.168.1.33
192.168.1.32
192.168.1.31

Here i'm running ansible in single node environment

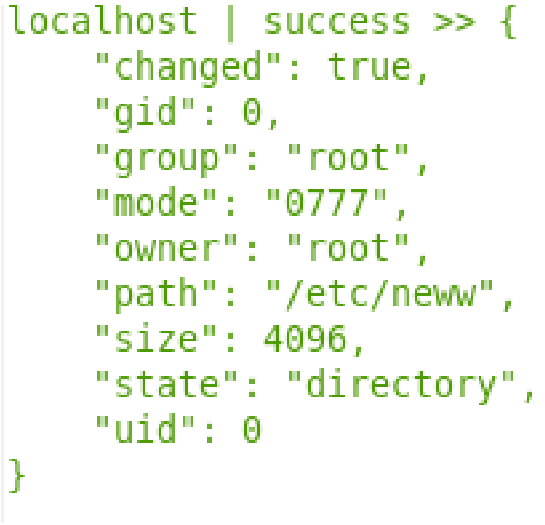
# ansible localhost -m ping



```
localhost | success >> {
    "changed": false,
    "ping": "pong"
}
```

2. Creating directory named /etc/new mode=775

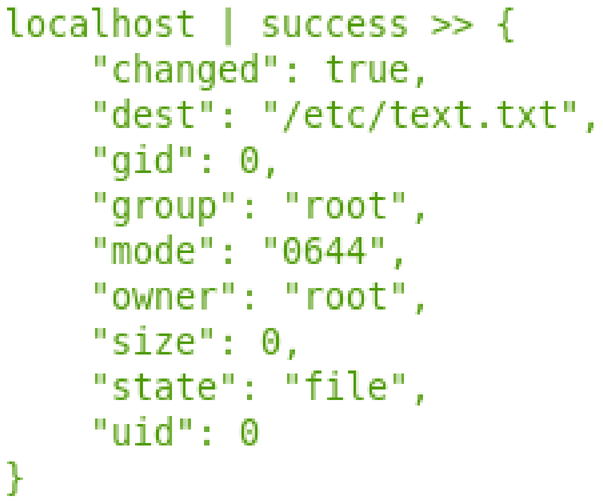
# ansible localhost -m file -a "path=/etc/new state=directory mode=0775"



```
localhost | success >> {
    "changed": true,
    "gid": 0,
    "group": "root",
    "mode": "0777",
    "owner": "root",
    "path": "/etc/neww",
    "size": 4096,
    "state": "directory",
    "uid": 0
}
```

3. Touching a file named test.txt

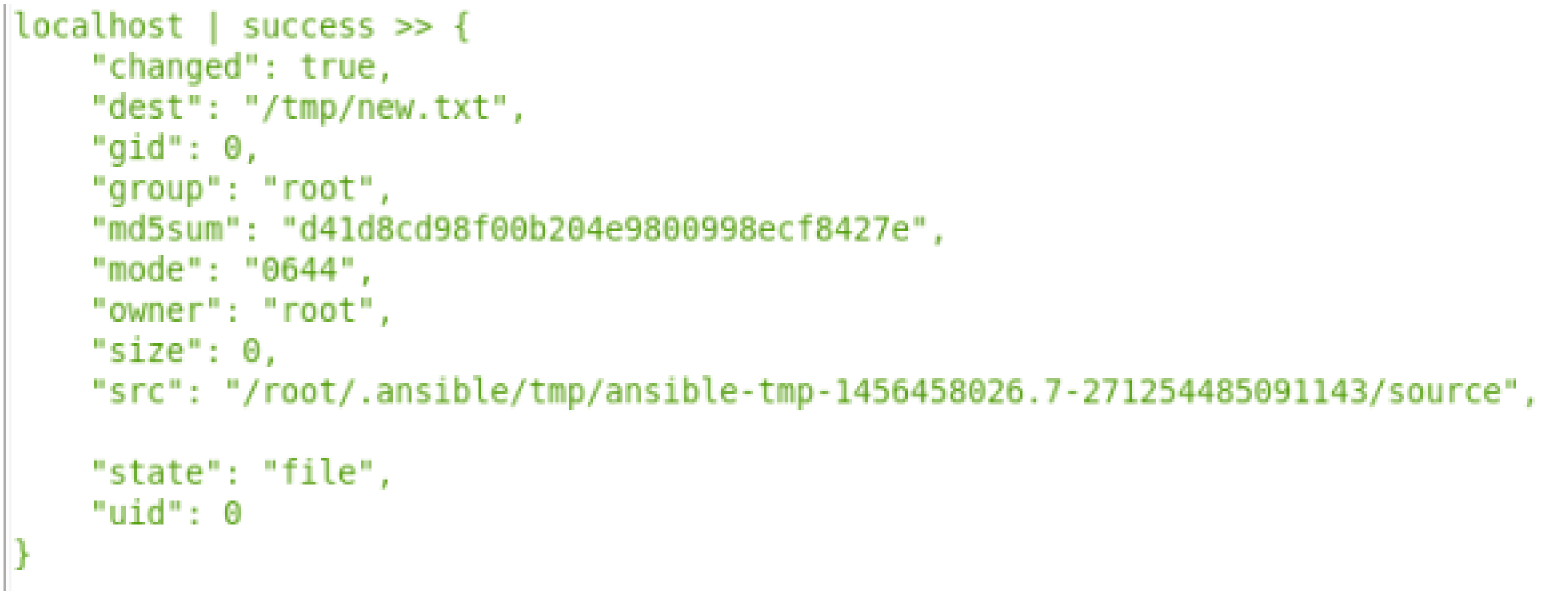
```
# ansible localhost -m file -a 'path=/etc/test.txt state=touch'
```



```
localhost | success >> {
    "changed": true,
    "dest": "/etc/text.txt",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
```

4. Copying file from /etc/new.txt to /tmp/new.txt

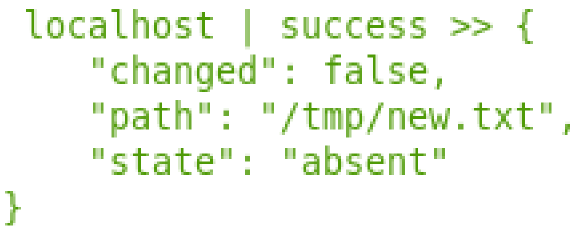
```
# ansible localhost -m copy -a 'src=/etc/new.txt dest=/tmp/
new.txt'
```



```
localhost | success >> {
    "changed": true,
    "dest": "/tmp/new.txt",
    "gid": 0,
    "group": "root",
    "md5sum": "d41d8cd98f00b204e9800998ecf8427e",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "src": "/root/.ansible/tmp/ansible-tmp-1456458026.7-271254485091143/source",

    "state": "file",
    "uid": 0
}
```

5. Deleting file /tmp/new.txt

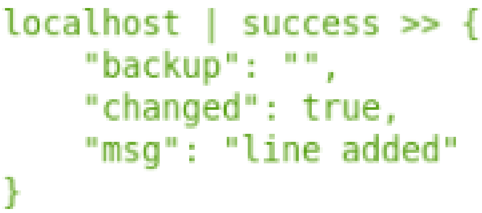
```
# ansible localhost -m file -a 'path=/tmp/new.txt state=absent'
```



```
localhost | success >> {
    "changed": false,
    "path": "/tmp/new.txt",
    "state": "absent"
}
```

6. Adding an entry to resolv.conf file

```
# ansible localhost -m lineinfile -a 'dest=/etc/resolv.conf
line="nameserver 8.8.4.4"'
```



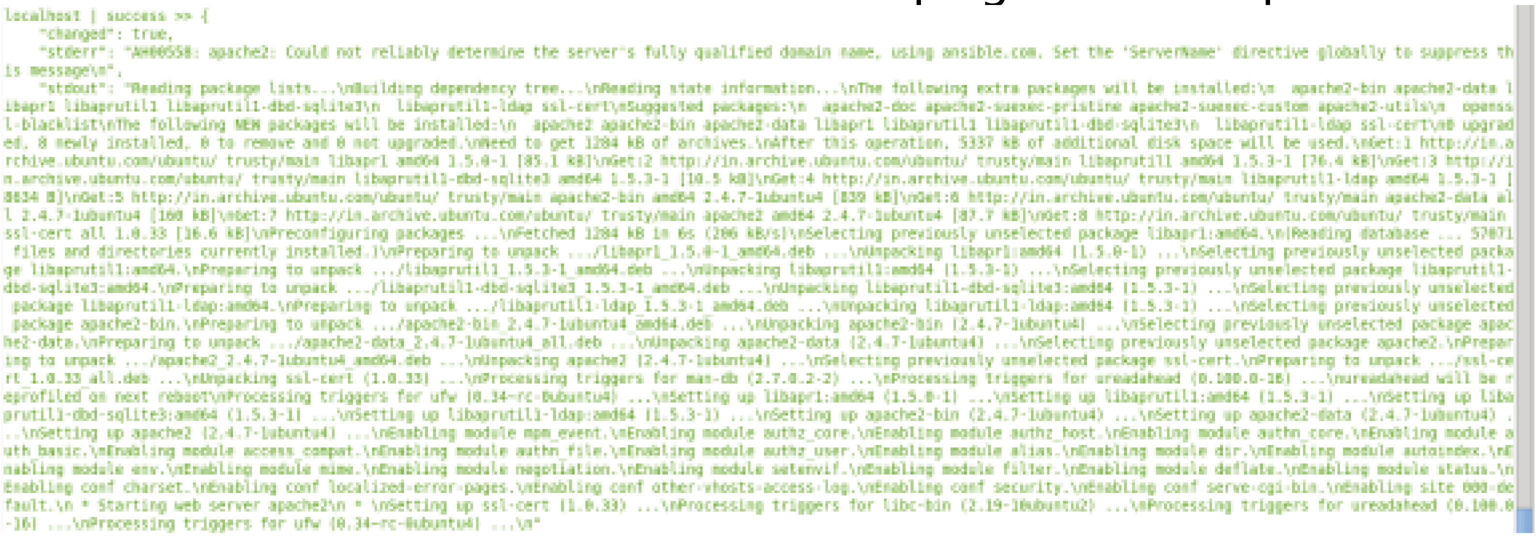
```
localhost | success >> {
    "backup": "",
    "changed": true,
    "msg": "line added"
}
```

7. Install apache

```
# ansible localhost -m apt -a 'name=apache2 state=present'
```

or You can install by using command

# ansible localhost -m command -a 'apt-get install apache'



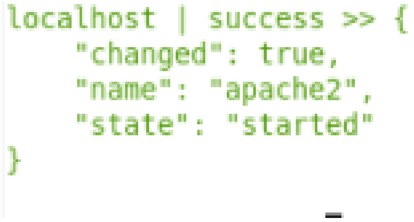
```
localhost | success >> {
    "changed": true,
    "stderr": "AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using ansible.com. Set the 'ServerName' directive globally to suppress th
is message\n",
    "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe following extra packages will be installed:\n  apache2-bin apache2-data l
ibapr1 libaprutil1 libaprutil1-dbd-sqlite3\n  libaprutil1-ldap ssl-cert\nSuggested packages:\n  apache2-doc apache2-suexec-pristine apache2-suexec-custom apache2-utils\n  openss
l-blacklist\nThe following NEW packages will be installed:\n  apache2 apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3\n  libaprutil1-ldap ssl-cert\n0 upgrad
ed, 8 newly installed, 0 to remove and 0 not upgraded.\nNeed to get 1284 kB of archives.\nAfter this operation, 5337 kB of additional disk space will be used.\nGet:1 http://in.a
rchive.ubuntu.com/ubuntu/ trusty/main libapr1 amd64 1.5.0-1 [85.1 kB]\nGet:2 http://in.archive.ubuntu.com/ubuntu/ trusty/main libaprutil1 amd64 1.5.3-1 [76.4 kB]\nGet:3 http://i
n.archive.ubuntu.com/ubuntu/ trusty/main libaprutil1-dbd-sqlite3 amd64 1.5.3-1 [10.5 kB]\nGet:4 http://in.archive.ubuntu.com/ubuntu/ trusty/main libaprutil1-ldap amd64 1.5.3-1 [
8634 B]\nGet:5 http://in.archive.ubuntu.com/ubuntu/ trusty/main apache2-bin amd64 2.4.7-1ubuntu4 [839 kB]\nGet:6 http://in.archive.ubuntu.com/ubuntu/ trusty/main apache2-data al
l 2.4.7-1ubuntu4 [160 kB]\nGet:7 http://in.archive.ubuntu.com/ubuntu/ trusty/main apache2 amd64 2.4.7-1ubuntu4 [87.7 kB]\nGet:8 http://in.archive.ubuntu.com/ubuntu/ trusty/main 
ssl-cert all 1.0.33 [16.6 kB]\nPreconfiguring packages ...\nFetched 1284 kB in 6s (206 kB/s)\nSelecting previously unselected package libapr1:amd64.\n(Reading database ... 57071
 files and directories currently installed.)\nPreparing to unpack .../libapr1_1.5.0-1_amd64.deb ...\nUnpacking libapr1:amd64 (1.5.0-1) ...\nSelecting previously unselected packa
ge libaprutil1:amd64.\nPreparing to unpack .../libaprutil1_1.5.3-1_amd64.deb ...\nUnpacking libaprutil1:amd64 (1.5.3-1) ...\nSelecting previously unselected package libaprutil1-
dbd-sqlite3:amd64.\nPreparing to unpack .../libaprutil1-dbd-sqlite3_1.5.3-1_amd64.deb ...\nUnpacking libaprutil1-dbd-sqlite3:amd64 (1.5.3-1) ...\nSelecting previously unselected
 package libaprutil1-ldap:amd64.\nPreparing to unpack .../libaprutil1-ldap_1.5.3-1_amd64.deb ...\nUnpacking libaprutil1-ldap:amd64 (1.5.3-1) ...\nSelecting previously unselected
 package apache2-bin.\nPreparing to unpack .../apache2-bin_2.4.7-1ubuntu4_amd64.deb ...\nUnpacking apache2-bin (2.4.7-1ubuntu4) ...\nSelecting previously unselected package apac
he2-data.\nPreparing to unpack .../apache2-data_2.4.7-1ubuntu4_all.deb ...\nUnpacking apache2-data (2.4.7-1ubuntu4) ...\nSelecting previously unselected package apache2.\nPrepar
ing to unpack .../apache2_2.4.7-1ubuntu4_amd64.deb ...\nUnpacking apache2 (2.4.7-1ubuntu4) ...\nSelecting previously unselected package ssl-cert.\nPreparing to unpack .../ssl-ce
rt_1.0.33_all.deb ...\nUnpacking ssl-cert (1.0.33) ...\nProcessing triggers for man-db (2.7.0.2-2) ...\nProcessing triggers for ureadahead (0.100.0-16) ...\nureadahead will be r
eprofiled on next reboot\nProcessing triggers for ufw (0.34~rc-0ubuntu4) ...\nSetting up libapr1:amd64 (1.5.0-1) ...\nSetting up libaprutil1:amd64 (1.5.3-1) ...\nSetting up liba
prutil1-dbd-sqlite3:amd64 (1.5.3-1) ...\nSetting up libaprutil1-ldap:amd64 (1.5.3-1) ...\nSetting up apache2-bin (2.4.7-1ubuntu4) ...\nSetting up apache2-data (2.4.7-1ubuntu4) .
..\nSetting up apache2 (2.4.7-1ubuntu4) ...\nEnabling module mpm_event.\nEnabling module authz_core.\nEnabling module authz_host.\nEnabling module authn_core.\nEnabling module a
uth_basic.\nEnabling module access_compat.\nEnabling module authn_file.\nEnabling module authz_user.\nEnabling module alias.\nEnabling module dir.\nEnabling module autoindex.\nE
nabling module env.\nEnabling module mime.\nEnabling module negotiation.\nEnabling module setenvif.\nEnabling module filter.\nEnabling module deflate.\nEnabling module status.\n
Enabling conf charset.\nEnabling conf localized-error-pages.\nEnabling conf other-vhosts-access-log.\nEnabling conf security.\nEnabling conf serve-cgi-bin.\nEnabling site 000-de
fault.\n * Starting web server apache2\n * \nSetting up ssl-cert (1.0.33) ...\nProcessing triggers for libc-bin (2.19-10ubuntu2) ...\nProcessing triggers for ureadahead (0.100.0
-16) ...\nProcessing triggers for ufw (0.34~rc-0ubuntu4) ...\n"
```

8. Restart apache services

# ansible localhost -m service -a 'name=apache2 state=present'

or

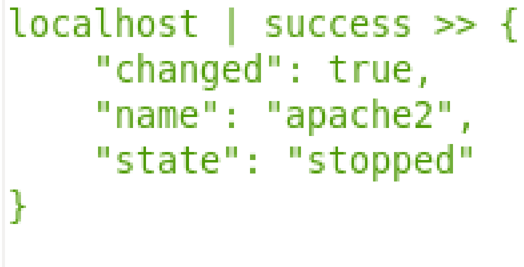
# ansible localhost -m command -a 'service apache2 restart'



```
localhost | success >> {
    "changed": true,
    "name": "apache2",
    "state": "started"
}
```

9. Stop service

# ansible localhost -m service -a 'name=apache2 state=stopped'



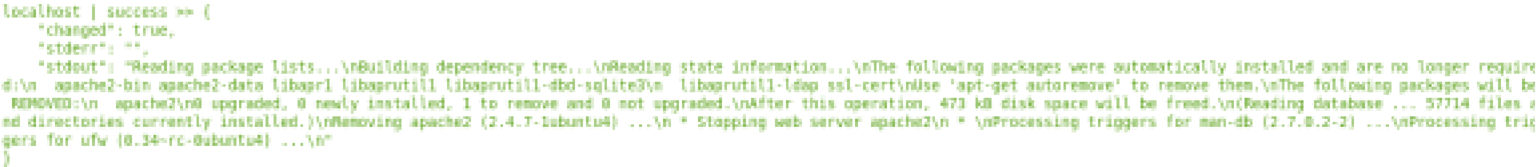
```
localhost | success >> {
    "changed": true,
    "name": "apache2",
    "state": "stopped"
}
```

10. Remove package

# ansible localhost -m apt -a 'name=apache2 state=absent'

or

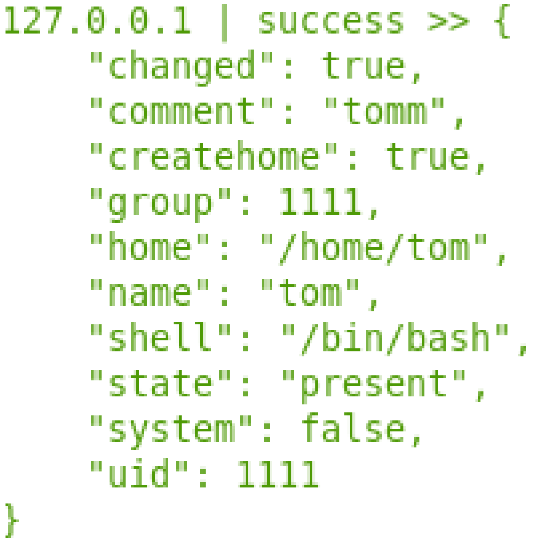
# ansible localhost -m command -a 'apt-get remove apache2'



```
localhost | success >> {
    "changed": true,
    "stderr": "",
    "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe following packages were automatically installed and are no longer require
d:\n  apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3\n  libaprutil1-ldap ssl-cert\nUse 'apt-get autoremove' to remove them.\nThe following packages will be
 REMOVED:\n  apache2\n0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.\nAfter this operation, 473 kB disk space will be freed.\n(Reading database ... 57714 files a
nd directories currently installed.)\nRemoving apache2 (2.4.7-1ubuntu4) ...\n * Stopping web server apache2\n * \nProcessing triggers for man-db (2.7.0.2-2) ...\nProcessing trig
gers for ufw (0.34~rc-0ubuntu4) ...\n"
}
```

11. Adding a user named "tom" with uid 1111 and password=123.

# ansible localhost -m user -a 'name=tom comment=tomm uid=1111 password=123'



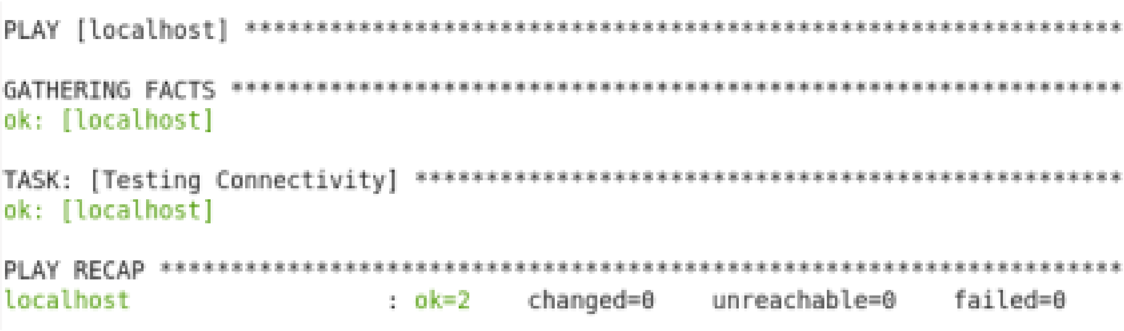
```
127.0.0.1 | success >> {
    "changed": true,
    "comment": "tomm",
    "createhome": true,
    "group": 1111,
    "home": "/home/tom",
    "name": "tom",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 1111
}
```

12. Playbook to check Connectivity

# vim ping.yml

```
---
 - hosts: localhost
   tasks:
       - name: Testing Connectivity
         ping
```

# ansible-playbook ping.yml



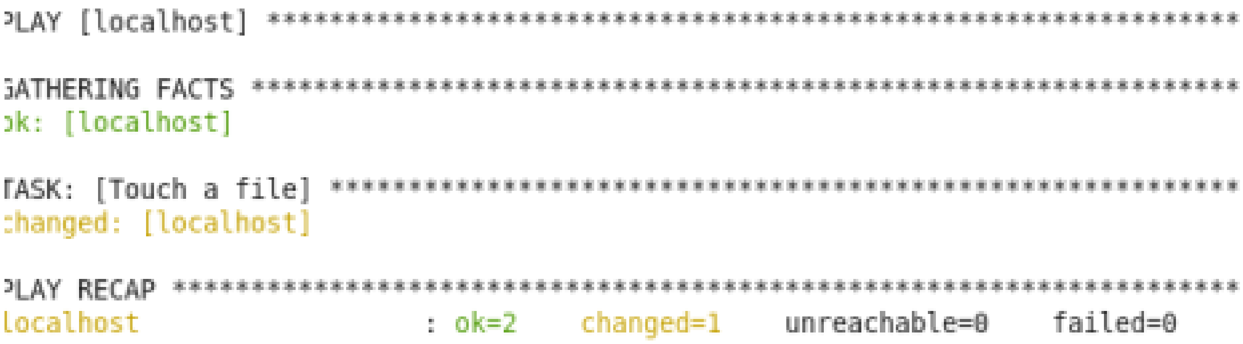
```
PLAY [localhost] *************************************************************

GATHERING FACTS **************************************************************
ok: [localhost]

TASK: [Testing Connectivity] *************************************************
ok: [localhost]

PLAY RECAP *******************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0
```

13. Playbook to touch a file named test1.txt

vim touch.yml

```
---
-
  hosts: localhost
  tasks:
     -
       name: Touching file test.txt
       file: "path=/etc/test.txt state=touch"
```

# ansible-playbook touch.yml



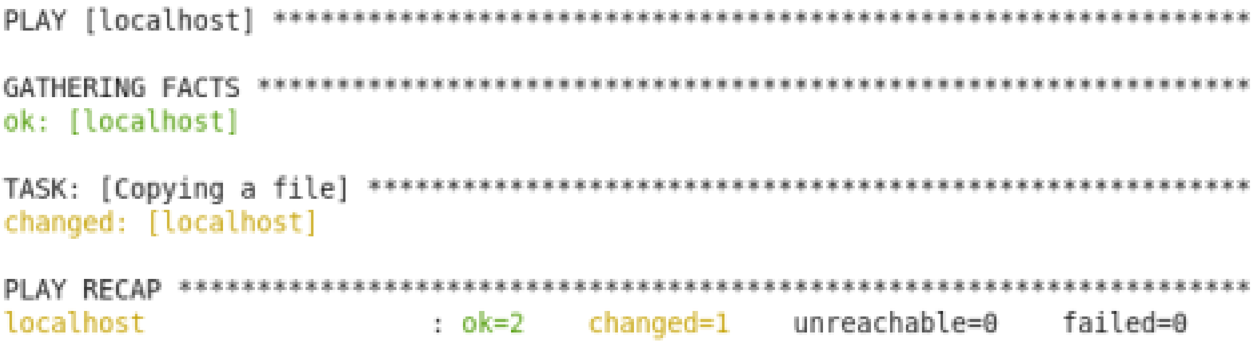
```
PLAY [localhost] **************************************************************

GATHERING FACTS ***************************************************************
ok: [localhost]

TASK: [Touch a file] **********************************************************
changed: [localhost]

PLAY RECAP ********************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0
```

14. Playbook to copy a file /etc/new.txt

# vim copy.yml

```
---
-
  hosts: localhost
  tasks:
    -
      name: Copying a file
      copy: "src=/etc/new.txt dest=/tmp/new.txt"
```

# ansible-playbook copy.yml



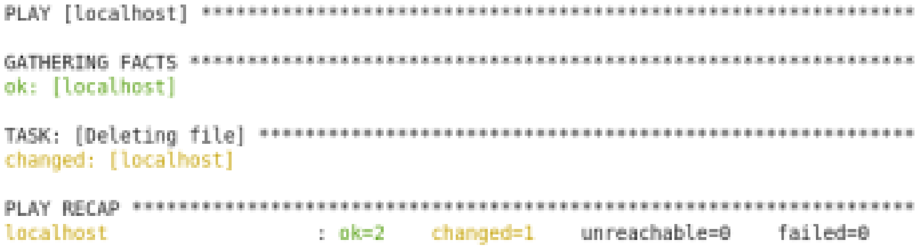
```
PLAY [localhost] **************************************************************

GATHERING FACTS ***************************************************************
ok: [localhost]

TASK: [Copying a file] ********************************************************
changed: [localhost]

PLAY RECAP ********************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0
```

15. Playbook to delete a file name /tmp/new.txt

# vim delete.yml

```
---
-
 hosts: localhost
 tasks:
    -
     name: Deleting file
     file: "path=/tmp/new.txt state=absent"
```

# ansible-playbook delete.yml



```
PLAY [localhost] *************************************************************

GATHERING FACTS **************************************************************
ok: [localhost]

TASK: [Deleting file] ********************************************************
changed: [localhost]

PLAY RECAP *******************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0
```

16. Playbook to add user named "john" uid=1010 password=123 with with 2048-bit ssh-key for user john

# vim useradd.yml

```
---
-
 hosts: localhost
 tasks:
   -
    name: Adding a user
    user: "name=john shell=/bin/bash password=123 uid=1010 generate_ssh_key=yes ssh_key_bits=2048
ssh_key_file=.ssh/id_rsa"
```

# ansible-playbook useradd.yml

Playbook for Removing added user named "john"
# vim userdel.yml

```
---
-
 hosts: localhost
 tasks:
   -
    name: Removing John
    user: "name=john state=absent remove=yes"
```

# ansible-playbook userdel.yml

17. Playbook to setup ssh

# vim sshsetup.yml

```
---
- hosts: localhost
  user: root
  vars:
    createuser: 'ansibleremote'
    createpassword: 'myamazingpassword'
  tasks:
  - name: Installing ssh-pass
    apt: "name=sshpass state=present"

  - name: Restarting Service
    service: "name=ssh state=restarted"

  - name: Setup | create user
    command: useradd -m {{ createuser }} creates=/home/{{ createuser }}
    sudo: true

  - name: Setup | set user password
    shell: usermod -p $(echo '{{ createpassword }}' | openssl passwd -1 -stdin) {{ createuser }}
    sudo: true

  - name: Setup | authorized key upload
    authorized_key: user={{ createuser }}
      key="{{ lookup('file', 'mypublickey.pub') }}"
      path='/home/{{ createuser }}/.ssh/authorized_keys'
      manage_dir=no
    sudo: true

  - name: Sudoers | update sudoers file and validate
    lineinfile: "dest=/etc/sudoers
      insertafter=EOF
      line='{{ createuser }} ALL=(ALL) NOPASSWD: ALL'
      regexp='{{ createuser }} ALL=(ALL) NOPASSWD: ALL'
      state=present"
    sudo: true
```

# ansible-playbook sshsetup.yml

18. Playbook to check server load, uptime, disk space usage on all hosts

# vim load.yml

```
---
 -
  hosts: localhost
  tasks:
    -
     name: Checking Disk Space USAGE
     shell: df -h
     register: df
    -
     debug: var=df.stdout_lines
    -
     name: Checking Server Load
     shell: w
     register: w
    -
     debug: var=w.stdout_lines
    -
     name: Uptime
     shell: uptime
     register: uptime
    -
     debug: var=uptime.stdout_lines
```

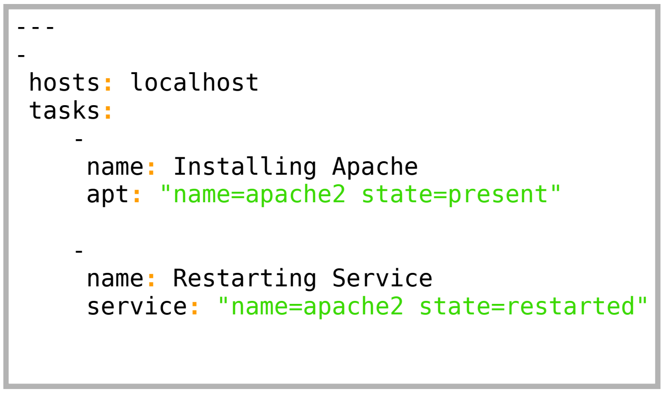
```
# ansible-playbook load.yml
```

19. Playabook to install chkrootkit

```
---
 -
  hosts: localhost
  tasks:
    -
     name: Installing Rootkit tool
     yum: "name=chkrootkit state=present"

    -
     name: Running chkrootkit
     command: "chkrootkit"
     register: chk
    -
     debug: var=chk.stdout_lines
```

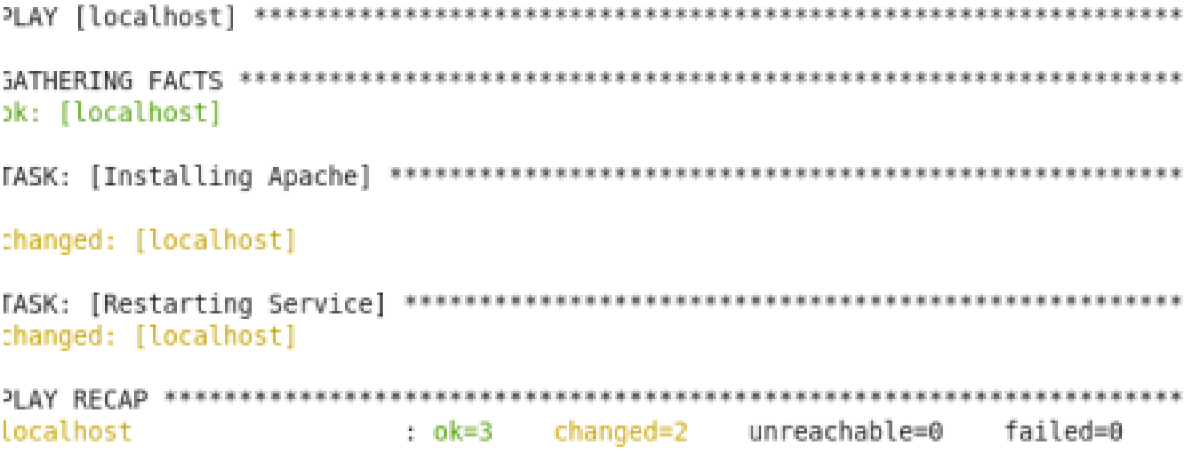
20. Playbook to install apache and restart its service

```
# vim apache.yml
```



```
---
-
 hosts: localhost
 tasks:
    -
     name: Installing Apache
     apt: "name=apache2 state=present"

    -
     name: Restarting Service
     service: "name=apache2 state=restarted"
```

# ansible-playbook apache.yml



```
PLAY [localhost] **************************************************************

GATHERING FACTS ***************************************************************
ok: [localhost]

TASK: [Installing Apache] *****************************************************

changed: [localhost]

TASK: [Restarting Service] ****************************************************
changed: [localhost]

PLAY RECAP ********************************************************************
localhost                  : ok=3    changed=2    unreachable=0    failed=0
```

Test apache by curl to localhost

# curl localhost

21. Create a playbook to add a iptables rule to allow apache to communicate on port 80

# vim iptablesrule.yml

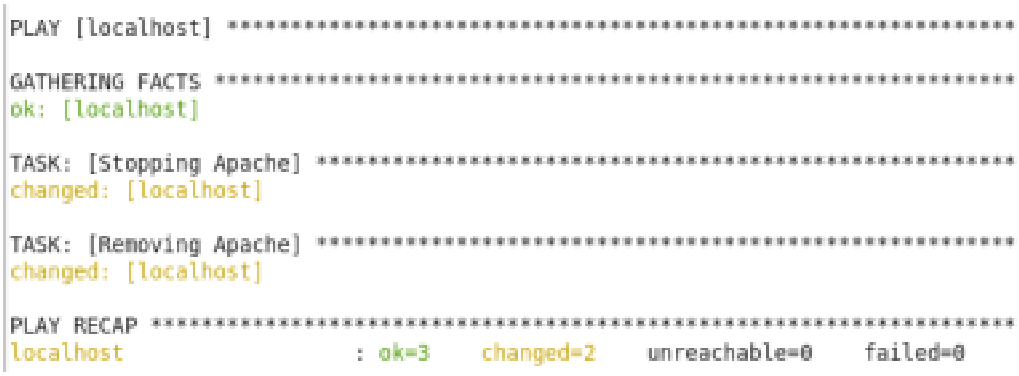
```
---
 -
  hosts: localhost
  tasks:
    -
     name: Apache | get iptables rules
     shell: iptables -L
     register: iptablesrules
     always_run: yes
     sudo: true

    -
     name: Apache | add apache iptable rule
     command: /sbin/iptables -I INPUT 1 -p tcp --dport http -j ACCEPT -m comment --comment
     "Apache"
     sudo: true
     when: iptablesrules.stdout.find("Apache") == -1

   -
    name: save iptables
    command: iptables-save
    sudo: true
```

22. Playbook to stop apache service and remove package

# vim remove.yml

```
---
 -
  hosts: localhost
  tasks:
     -
      name: Stopping Apache
      service: "name=apache2 state=stopped"
     -
      name: Removing Apache
      apt: "name=apache2 state=absent"
```

# ansible-playbook remove.yml



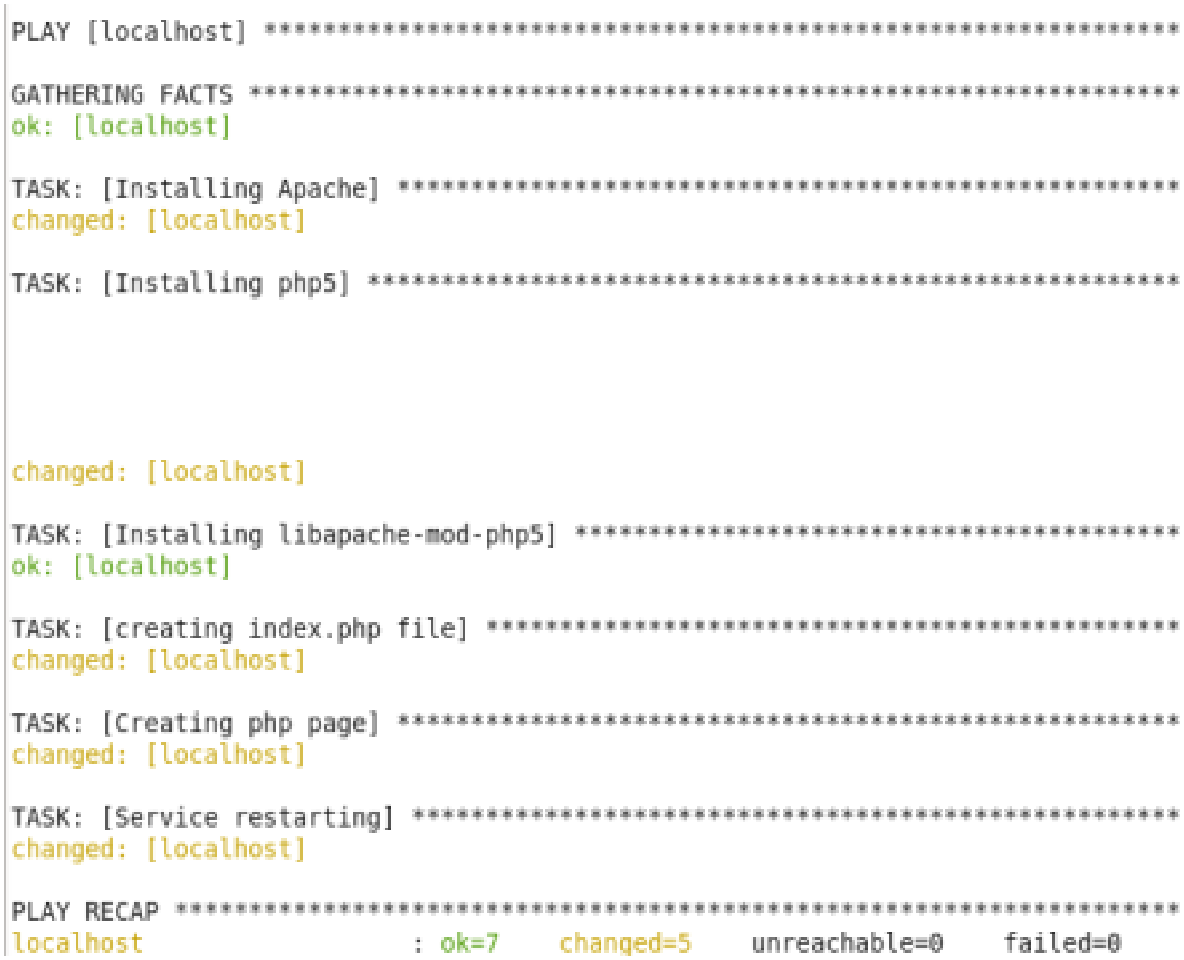
```
PLAY [localhost] ******************************************************************

GATHERING FACTS *******************************************************************
ok: [localhost]

TASK: [Stopping Apache] ***********************************************************
changed: [localhost]

TASK: [Removing Apache] ***********************************************************
changed: [localhost]

PLAY RECAP ************************************************************************
localhost                  : ok=3    changed=2    unreachable=0    failed=0
```

23. Playbook to Install apache and php

```
# vim apache-php.yml
```

```
---
 -
  hosts: localhost
  tasks:
     -
      name: Installing Apache
      apt: "name=apache2 state=present"
     -
      name: Installing php5
      apt: "name=php5 state=present"
     -
      name: Installing libapache-mod-php5
      apt: "name=libapache2-mod-php5 state=present"
     -
      name: creating index.php file
      file: "path=/var/www/html/index.php state=touch"
     -
      name: Creating php page
      lineinfile: "dest=/var/www/html/index.php line='<?php\nphpinfo();\n?>'"
     -
      name: Service restarting
      service: "name=apache2 state=restarted"
```

```
# ansible-playbook apache-php.yml
```



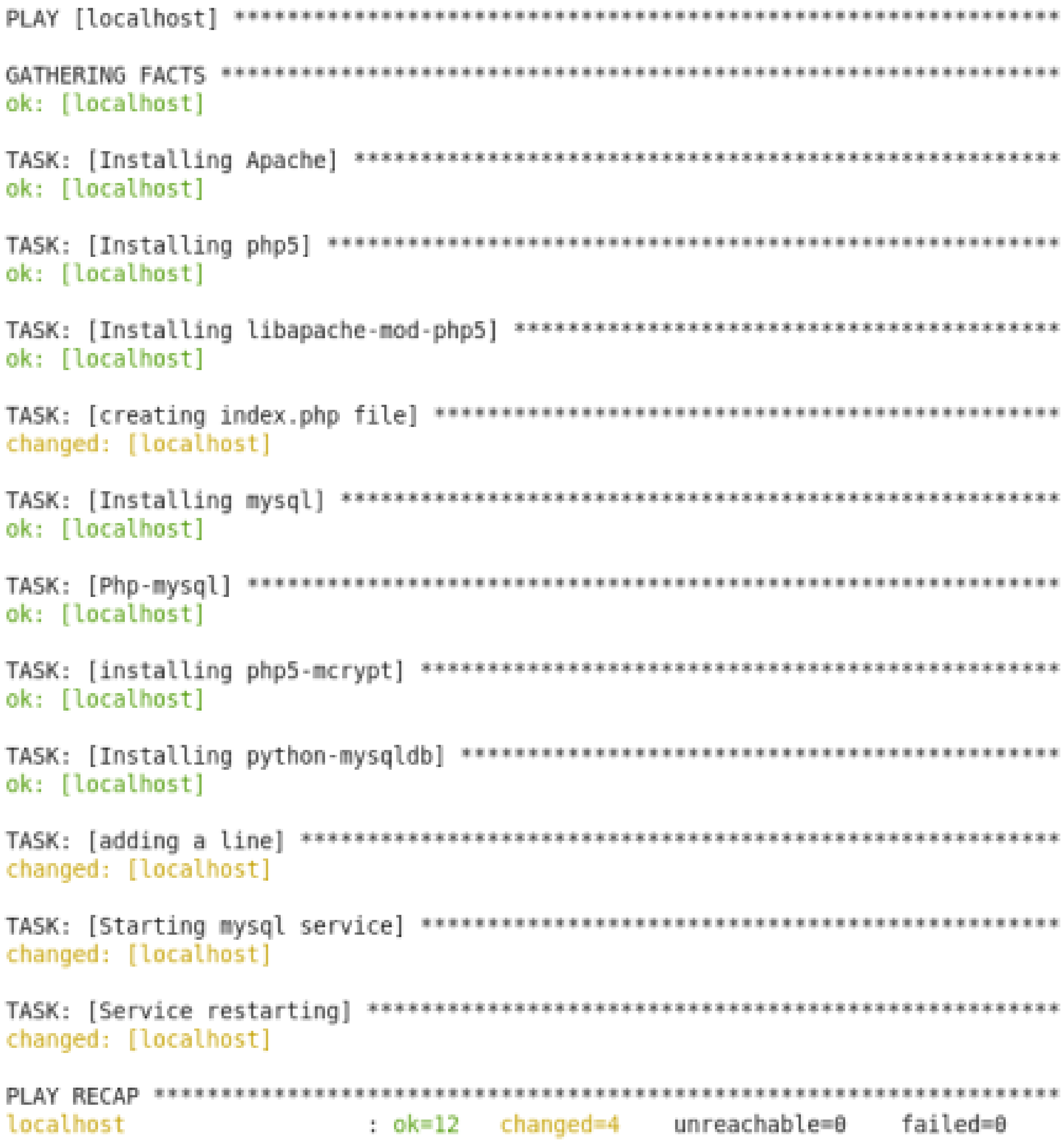
```
PLAY [localhost] ***************************************************************

GATHERING FACTS ***************************************************************
ok: [localhost]

TASK: [Installing Apache] *****************************************************
changed: [localhost]

TASK: [Installing php5] *******************************************************




changed: [localhost]

TASK: [Installing libapache-mod-php5] *****************************************
ok: [localhost]

TASK: [creating index.php file] ***********************************************
changed: [localhost]

TASK: [Creating php page] *****************************************************
changed: [localhost]

TASK: [Service restarting] ****************************************************
changed: [localhost]

PLAY RECAP ********************************************************************
localhost                  : ok=7    changed=5    unreachable=0    failed=0
```

24. Playbook for LAMP server

# vim lamp.yml

```
---
 -
  hosts: localhost
  tasks:
     -
      name: Installing Apache
      apt: "name=apache2 state=present"
     -
      name: Installing php5
      apt: "name=php5 state=present"
     -
      name: Installing libapache-mod-php5
      apt: "name=libapache2-mod-php5 state=present"
     -
      name: creating index.php file
      file: "path=/var/www/html/index.php state=touch"
     -
      name: Creating php page
      lineinfile: "dest=/var/www/html/index.php line='<?php\nphpinfo();\n?>'"
     -
      name: Installing mysql
      apt: "name=mysql-server state=present"
     -
      name: Php-mysql
      apt: "name=php5-mysql state=present"
     -
      name: installing php5-mcrypt
      apt: "name=php5-mcrypt state=present"
     -
      name: Installing python-mysqldb
      apt: "name=python-mysqldb"
       mysql_user:
       name: root
       password: "123"
       login_user: root
       login_password: "123"
       check_implicit_admin: yes
       priv: "*.*:ALL,GRANT"
     -
      name: touching ~/.my.cnf
      file: "path=/root/.my.cnf state=touch"
     -
      name: adding password in ~/.my.cnf
      lineinfile: "dest=/root/.my.cnf line='[client]\nuser=root\npassword=123'"
     -
      name: Starting mysql service
      service: "name=mysql state=restarted"
     -
      name: Service restarting
      service: "name=apache2 state=restarted"
```

# ansible-playbook lamp.yml



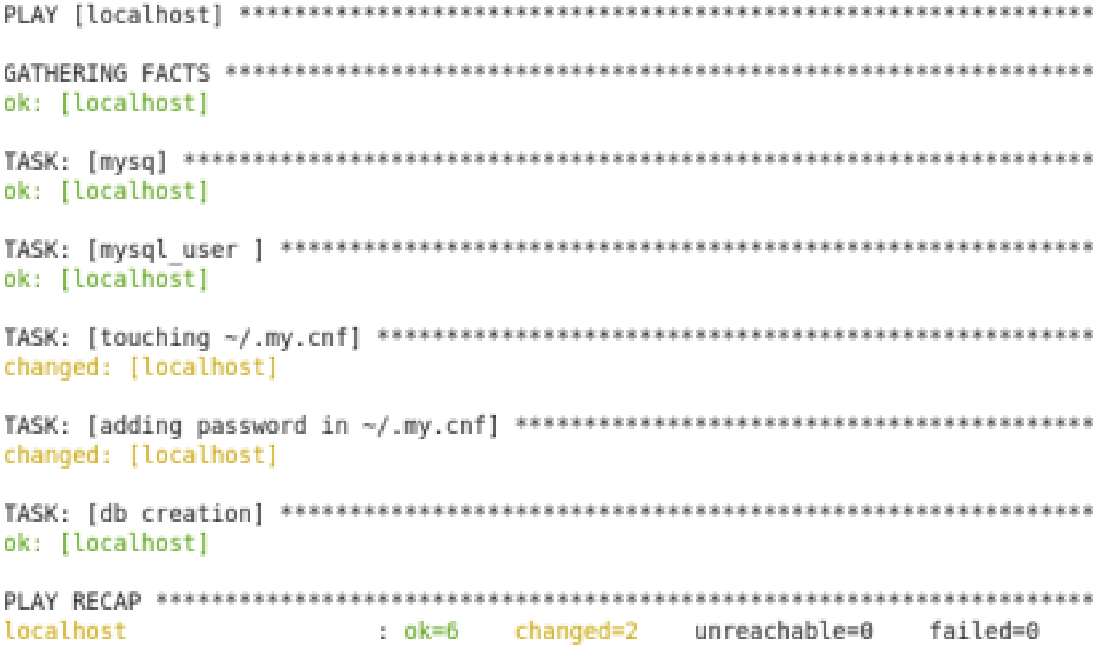
```
PLAY [localhost] **************************************************************

GATHERING FACTS ***************************************************************
ok: [localhost]

TASK: [Installing Apache] *****************************************************
ok: [localhost]

TASK: [Installing php5] *******************************************************
ok: [localhost]

TASK: [Installing libapache-mod-php5] *****************************************
ok: [localhost]

TASK: [creating index.php file] ***********************************************
changed: [localhost]

TASK: [Installing mysql] ******************************************************
ok: [localhost]

TASK: [Php-mysql] *************************************************************
ok: [localhost]

TASK: [installing php5-mcrypt] ************************************************
ok: [localhost]

TASK: [Installing python-mysqldb] *********************************************
ok: [localhost]

TASK: [adding a line] *********************************************************
changed: [localhost]

TASK: [Starting mysql service] ************************************************
changed: [localhost]

TASK: [Service restarting] ****************************************************
changed: [localhost]

PLAY RECAP ********************************************************************
localhost                  : ok=12   changed=4    unreachable=0    failed=0
```

25. Create database named "testdb"

# vim mysqldb.yml

```
---
 -
  hosts: localhost
  tasks:
-
     name: mysql
     apt: "name=mysql-server state=present"
    -
     mysql_user:
       name: root
       password: "123"
       login_user: root
       login_password: "123"
       check_implicit_admin: yes
       priv: "*.*:ALL,GRANT"
    -
     name: touching ~/.my.cnf
     file: "path=/root/.my.cnf state=touch"
    -
     name: adding password in ~/.my.cnf
     lineinfile: "dest=/root/.my.cnf line='[client]\nuser=root\npassword=123'"
    -
     name: db creation
     mysql_db: name=testdb state=present
    -


    -
     name: service restart
     service: "name=mysql state=restart"
```

# ansible-playbook mysqldb.yml



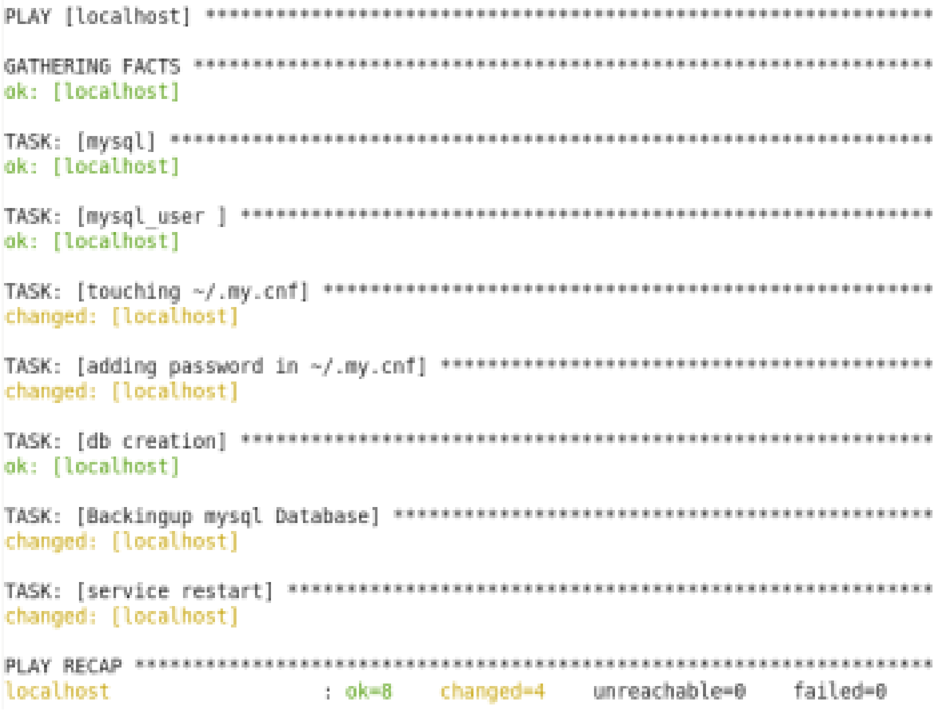
```
PLAY [localhost] ***************************************************************

GATHERING FACTS ****************************************************************
ok: [localhost]

TASK: [mysql] ******************************************************************
ok: [localhost]

TASK: [mysql_user ] ************************************************************
ok: [localhost]

TASK: [touching ~/.my.cnf] *****************************************************
changed: [localhost]

TASK: [adding password in ~/.my.cnf] *******************************************
changed: [localhost]

TASK: [db creation] ************************************************************
ok: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=6    changed=2    unreachable=0    failed=0
```

26. Backing up database to /tmp/mysql.sql

# vim dump.yml

```
---
 -
  hosts: localhost
  tasks:
    -
     name: mysql
     apt: "name=mysql-server state=present"
    -
     mysql_user:
       name: root
       password: "123"
       login_user: root
       login_password: "123"
       check_implicit_admin: yes
       priv: "*.*:ALL,GRANT"
    -
     name: touching ~/.my.cnf
     file: "path=/root/.my.cnf state=touch"
    -
     name: adding password in ~/.my.cnf
     lineinfile: "dest=/root/.my.cnf line='[client]\nuser=root\npassword=123'"
    -
     name: db creation
     mysql_db: name=testdb state=present
    -
     name: Backingup mysql Database
     mysql_db: "state=dump name=testdb target=/tmp/mysql.sql"
    -
     name: service restart
     service: "name=mysql state=restarted"
```

# ansible-playbook dump.yml



```
PLAY [localhost] **************************************************************

GATHERING FACTS ***************************************************************
ok: [localhost]

TASK: [mysql] *****************************************************************
ok: [localhost]

TASK: [mysql_user ] ***********************************************************
ok: [localhost]

TASK: [touching ~/.my.cnf] ****************************************************
changed: [localhost]

TASK: [adding password in ~/.my.cnf] ******************************************
changed: [localhost]

TASK: [db creation] ***********************************************************
ok: [localhost]

TASK: [Backingup mysql Database] **********************************************
changed: [localhost]

TASK: [service restart] *******************************************************
changed: [localhost]

PLAY RECAP ********************************************************************
localhost                  : ok=8    changed=4    unreachable=0    failed=0
```

27. Restoring dump.sql

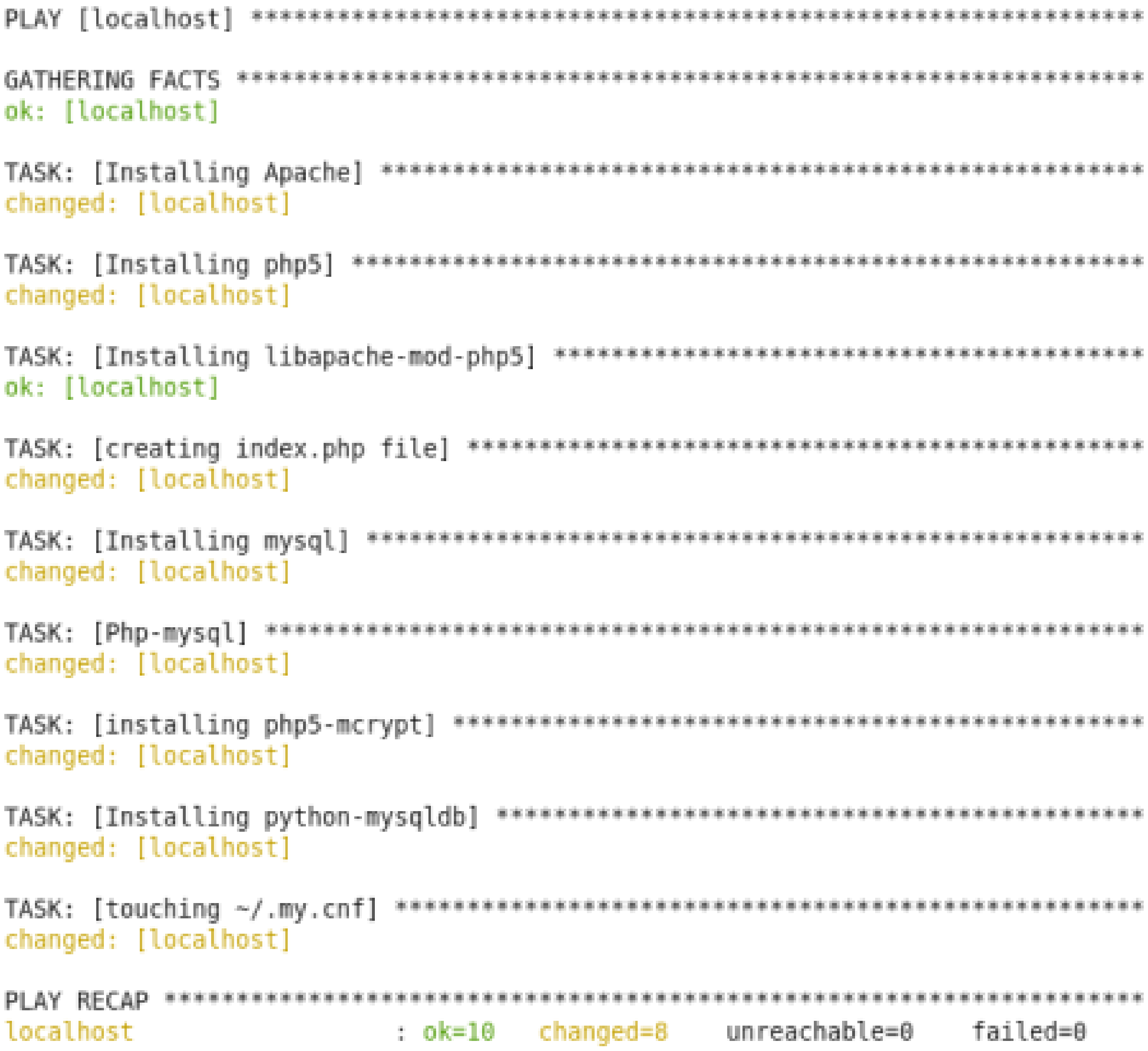
```
---
 -
  hosts: localhost
  tasks:
    -
     name: mysql
     apt: "name=mysql-server state=present"
    -
     mysql_user:
       name: root
       password: "123"
       login_user: root
       login_password: "123"
       check_implicit_admin: yes
       priv: "*.*:ALL,GRANT"
    -
     name: touching ~/.my.cnf
     file: "path=/root/.my.cnf state=touch"
    -
     name: adding password in ~/.my.cnf
     lineinfile: "dest=/root/.my.cnf line='[client]\nuser=root\npassword=123'"
    -
     name: Restoring mysql Database
     mysql_db: "state=import name=testdb target=/tmp/mysql.sql"
    -
     name: service restart
     service: "name=mysql state=restarted"
```

28. Remove LAMP server

# vim lamp-remove.yml

```
---
 -
  hosts: localhost
  tasks:
     -
      name: Installing Apache
      apt: "name=apache2 state=absent"
     -
      name: Installing php5
      apt: "name=php5 state=absent"
     -
      name: Installing libapache-mod-php5
      apt: "name=libapache2-mod-php5 state=absent"
     -
      name: creating index.php file
      file: "path=/var/www/html/index.php state=absent"
     -
      name: Installing mysql
      apt: "name=mysql-server state=absent"
     -
      name: Php-mysql
      apt: "name=php5-mysql state=absent"
     -
      name: installing php5-mcrypt
      apt: "name=php5-mcrypt state=absent"
     -
      name: Installing python-mysqldb
      apt: "name=python-mysqldb state=absent"
     -
      name: touching ~/.my.cnf
      file: "path=/root/.my.cnf state=absent"
```

```
# ansible lamp-remove.yml
```



```
PLAY [localhost] **************************************************************

GATHERING FACTS ***************************************************************
ok: [localhost]

TASK: [Installing Apache] *****************************************************
changed: [localhost]

TASK: [Installing php5] *******************************************************
changed: [localhost]

TASK: [Installing libapache-mod-php5] *****************************************
ok: [localhost]

TASK: [creating index.php file] ***********************************************
changed: [localhost]

TASK: [Installing mysql] ******************************************************
changed: [localhost]

TASK: [Php-mysql] *************************************************************
changed: [localhost]

TASK: [installing php5-mcrypt] ************************************************
changed: [localhost]

TASK: [Installing python-mysqldb] *********************************************
changed: [localhost]

TASK: [touching ~/.my.cnf] ****************************************************
changed: [localhost]

PLAY RECAP ********************************************************************
localhost                  : ok=10   changed=8    unreachable=0    failed=0
```

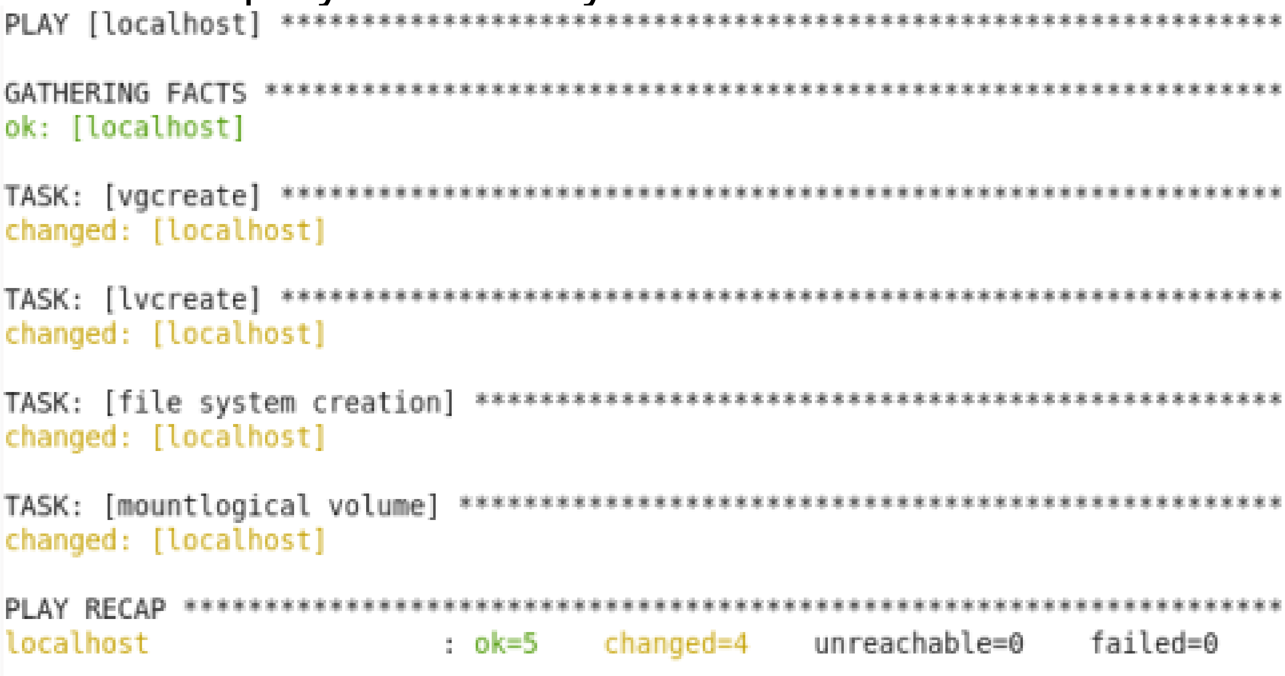
29. Playbook to configure LVM

Create a partition /dev/sdb5 and assign it 8e Linux LVM Hec code value

# vim lvm.yml

```
--
 -
  hosts: localhost
  user: root
  tasks:
    -
     name: vgcreate
     lvg:
         vg: vgdata
         pvs: /dev/sdb5
    -
     name: lvcreate
     lvol:
          vg: vgdata
          lv: dataone
          size: 50M
    -
     name: file system creation
     filesystem:
                fstype: ext4
                dev: /dev/vgdata/dataone
    -
     name: mountlogical volume
     mount:
           name: /lvdata
           src: /dev/vgdata/dataone
           fstype: ext4
           state: mounted
```

# ansible-playbook lvm.yml



```
PLAY [localhost] ***************************************************************

GATHERING FACTS ****************************************************************
ok: [localhost]

TASK: [vgcreate] ***************************************************************
changed: [localhost]

TASK: [lvcreate] ***************************************************************
changed: [localhost]

TASK: [file system creation] ***************************************************
changed: [localhost]

TASK: [mountlogical volume] ****************************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=5    changed=4    unreachable=0    failed=0
```

30. Playbook to install and configure wordpress

```
# mkdir wordpress-ansible && cd wordpress-ansible
```

Step 1 — Setting Up the File Structure

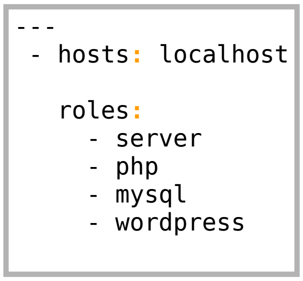
```
# touch playbook.yml
```

```
# mkdir roles && cd roles
```

We can bootstrap our roles with an Ansible tool called ansible-galaxy. For each role that we want to create, we will run ansible-galaxy init:

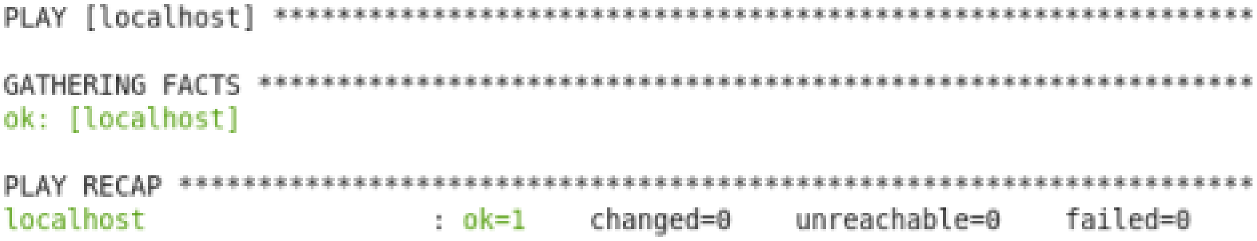
```
# ansible-galaxy init server
# ansible-galaxy init php
# ansible-galaxy init mysql
# ansible-galaxy init wordpress
```

```
# nano ~/wordpress-ansible/playbook.yml
```



```
---
 - hosts: localhost

   roles:
     - server
     - php
     - mysql
     - wordpress
```

```
# ansible-playbook playbook.yml
```



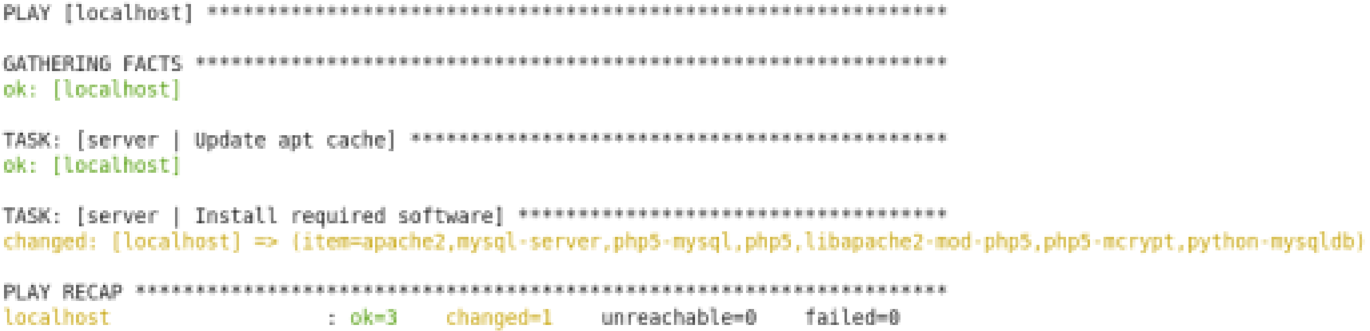
```
PLAY [localhost] **************************************************************

GATHERING FACTS ***************************************************************
ok: [localhost]

PLAY RECAP ********************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0
```

Step 2 - Creating Roles

nano roles/server/tasks/main.yml

```
---
 - name: Update apt cache
   apt: update_cache=yes cache_valid_time=3600
   sudo: yes

 - name: Install required software
   apt: name={{ item }} state=present
   sudo: yes
   with_items:
     - apache2
     - mysql-server
     - php5-mysql
     - php5
     -  libapache2-mod-php5
     - php5-mcrypt
     - python-mysqldb
```

# ansible-playbook playbook.yml



```
PLAY [localhost] **************************************************************

GATHERING FACTS ***************************************************************
ok: [localhost]

TASK: [server | Update apt cache] *********************************************
ok: [localhost]

TASK: [server | Install required software] ************************************
changed: [localhost] => (item=apache2,mysql-server,php5-mysql,php5,libapache2-mod-php5,php5-mcrypt,python-mysqldb)

PLAY RECAP ********************************************************************
localhost                  : ok=3    changed=1    unreachable=0    failed=0
```

# nano roles/php/tasks/main.yml

```
---
 - name: Install php extensions
   apt: name={{ item }} state=present
   sudo: yes
   with_items:
     - php5-gd
     - libssh2-php
```

# nano roles/mysql/defaults/main.yml

```
---
- name: Install php extensions
  apt: name={{ item }} state=present
  sudo: yes
  with_items:
    - php5-gd
    - libssh2-php
```

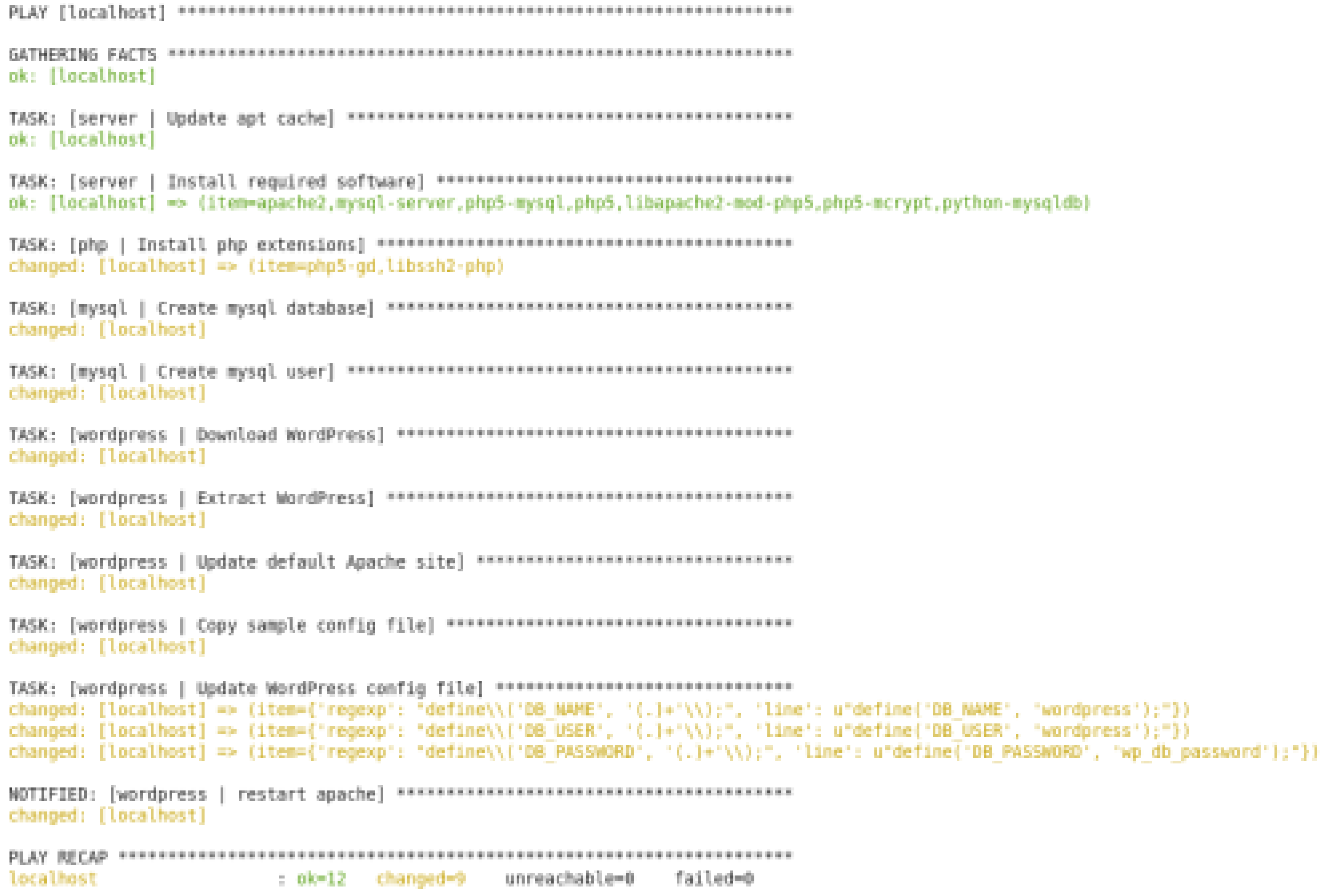
# nano roles/mysql/tasks/main.yml

```
---
# tasks file for wordpress

- name: Download WordPress
  get_url: "url=https://wordpress.org/latest.tar.gz dest=/tmp/wordpress.tar.gz validate_certs=no"

- name: Extract WordPress
  unarchive: src=/tmp/wordpress.tar.gz dest=/var/www/ copy=no
  sudo: yes

- name: Update default Apache site
  sudo: yes
  lineinfile:
    dest=/etc/apache2/sites-enabled/000-default.conf
    regexp="(.)+DocumentRoot /var/www/html"
    line="DocumentRoot /var/www/wordpress"
  notify:
    - restart apache

- name: Copy sample config file
  command: mv /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-config.php creates=/var/www/
wordpress/wp-config.php
  sudo: yes

- name: Update WordPress config file
  lineinfile:
    dest=/var/www/wordpress/wp-config.php
    regexp="{{ item.regexp }}"
    line="{{ item.line }}"
  with_items:
    - {'regexp': "define\\('DB_NAME', '(.)+'\\);", 'line': "define('DB_NAME', '{{wp_mysql_db}}');"}
    - {'regexp': "define\\('DB_USER', '(.)+'\\);", 'line': "define('DB_USER', '{{wp_mysql_user}}');"}
    - {'regexp': "define\\('DB_PASSWORD', '(.)+'\\);", 'line': "define('DB_PASSWORD',
'{{wp_mysql_password}}');"}
  sudo: yes
```

# nano roles/wordpress/handlers/main.yml

```
---
- name: restart apache
  service: name=apache2 state=restarted
  sudo: yes
```

# ansible-playbook playbook.yml



```
PLAY [localhost] ***************************************************************

GATHERING FACTS ****************************************************************
ok: [localhost]

TASK: [server | Update apt cache] **********************************************
ok: [localhost]

TASK: [server | Install required software] *************************************
ok: [localhost] => (item=apache2,mysql-server,php5-mysql,php5,libapache2-mod-php5,php5-mcrypt,python-mysqldb)

TASK: [php | Install php extensions] *******************************************
changed: [localhost] => (item=php5-gd,libssh2-php)

TASK: [mysql | Create mysql database] ******************************************
changed: [localhost]

TASK: [mysql | Create mysql user] **********************************************
changed: [localhost]

TASK: [wordpress | Download WordPress] *****************************************
changed: [localhost]

TASK: [wordpress | Extract WordPress] ******************************************
changed: [localhost]

TASK: [wordpress | Update default Apache site] *********************************
changed: [localhost]

TASK: [wordpress | Copy sample config file] ************************************
changed: [localhost]

TASK: [wordpress | Update WordPress config file] *******************************
changed: [localhost] => (item={'regexp': "define\\('DB_NAME', '(.)+'\\);", 'line': u"define('DB_NAME', 'wordpress');"})
changed: [localhost] => (item={'regexp': "define\\('DB_USER', '(.)+'\\);", 'line': u"define('DB_USER', 'wordpress');"})
changed: [localhost] => (item={'regexp': "define\\('DB_PASSWORD', '(.)+'\\);", 'line': u"define('DB_PASSWORD', 'wp_db_password');"})

NOTIFIED: [wordpress | restart apache] *****************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=12   changed=9    unreachable=0    failed=0
```

You should be able to view your WordPress site online at: http://your_server_ip.

30.