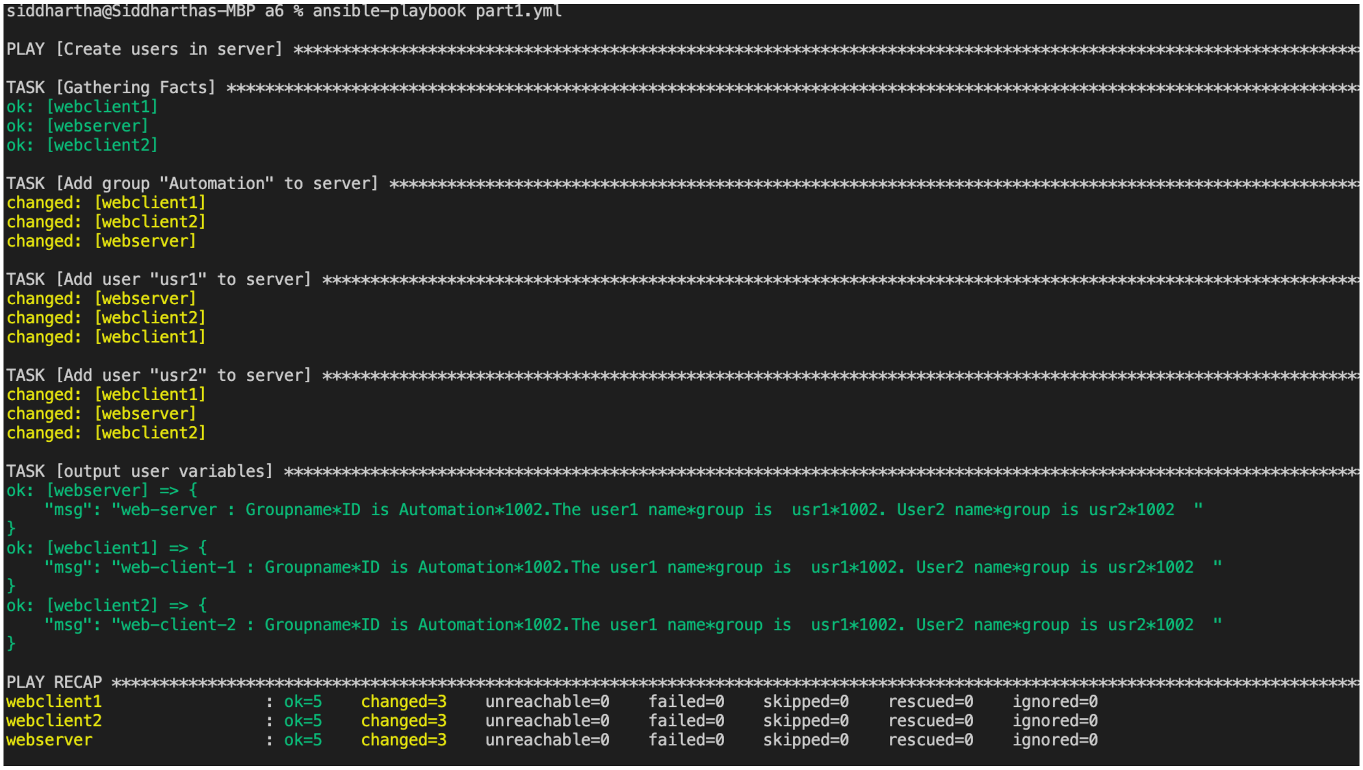
ASSIGNMENT - 6

Setup: The assignment has been done using vagrant setup.

1. **Code-**

```
---
- name : Create users in server
  hosts: assignment
  remote_user: ansible
  become: yes
  become_method: sudo
  gather_facts: true
  connection: ssh
  tasks:
     - name: Add group "Automation" to server
       group :
        name: Automation
        state: present
       register: grp1
     - name: Add user "usr1" to server
       user:
        name: usr1
        comment: "UnPrivileged User"
        uid: 2001
        group: Automation
        shell: /bin/bash
        generate_ssh_key: yes
        ssh_key_bits: 2048
        ssh_key_file: .ssh/id_rsa
       register: usr1
     - name: Add user "usr2" to server
       user:
        name: usr2
        comment: "UnPrivileged User"
        uid: 2002
        group: Automation
        shell: /bin/bash
        generate_ssh_key: yes
        ssh_key_bits: 2048
        ssh_key_file: .ssh/id_rsa
       register: usr2
     - name: output user variables
       debug:
        msg: "{{ ansible_hostname }} : Groupname*ID is {{ grp1.name }}*{{ grp1.gid }}.The user1 name*group is  {{ usr1.name }}*{{usr1.group}}. User2 name*group is {{usr2.name}}*{{usr2.group
```

Output –



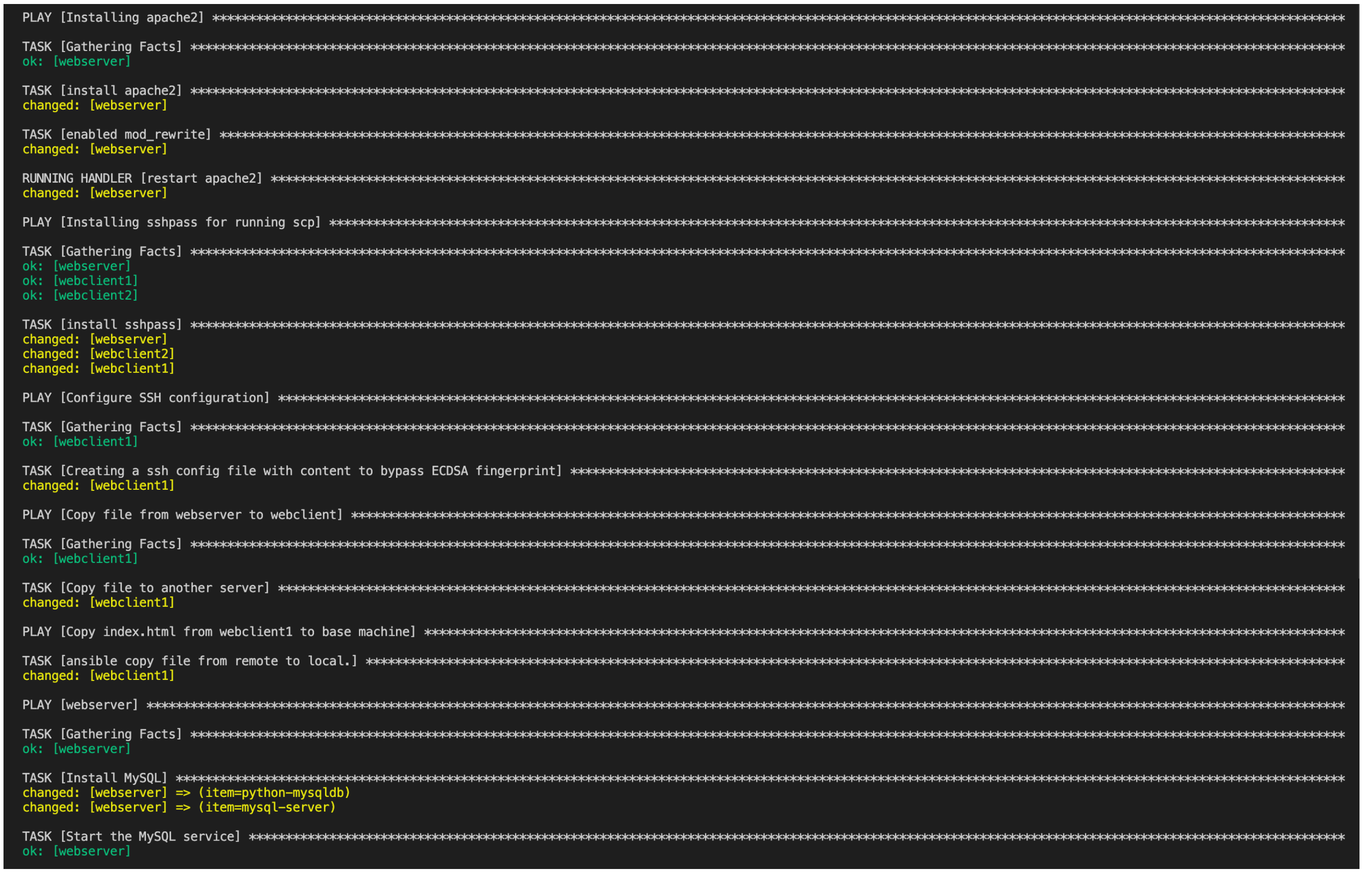
```
siddhartha@Siddharthas-MBP a6 % ansible-playbook part1.yml

PLAY [Create users in server] **************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [webclient1]
ok: [webserver]
ok: [webclient2]

TASK [Add group "Automation" to server] ****************************************************************
changed: [webclient1]
changed: [webclient2]
changed: [webserver]

TASK [Add user "usr1" to server] ***********************************************************************
changed: [webserver]
changed: [webclient2]
changed: [webclient1]

TASK [Add user "usr2" to server] ***********************************************************************
changed: [webclient1]
changed: [webserver]
changed: [webclient2]

TASK [output user variables] ***************************************************************************
ok: [webserver] => {
    "msg": "web-server : Groupname*ID is Automation*1002.The user1 name*group is  usr1*1002. User2 name*group is usr2*1002  "
}
ok: [webclient1] => {
    "msg": "web-client-1 : Groupname*ID is Automation*1002.The user1 name*group is  usr1*1002. User2 name*group is usr2*1002  "
}
ok: [webclient2] => {
    "msg": "web-client-2 : Groupname*ID is Automation*1002.The user1 name*group is  usr1*1002. User2 name*group is usr2*1002  "
}

PLAY RECAP *********************************************************************************************
webclient1                 : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
webclient2                 : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
webserver                  : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

2. Code

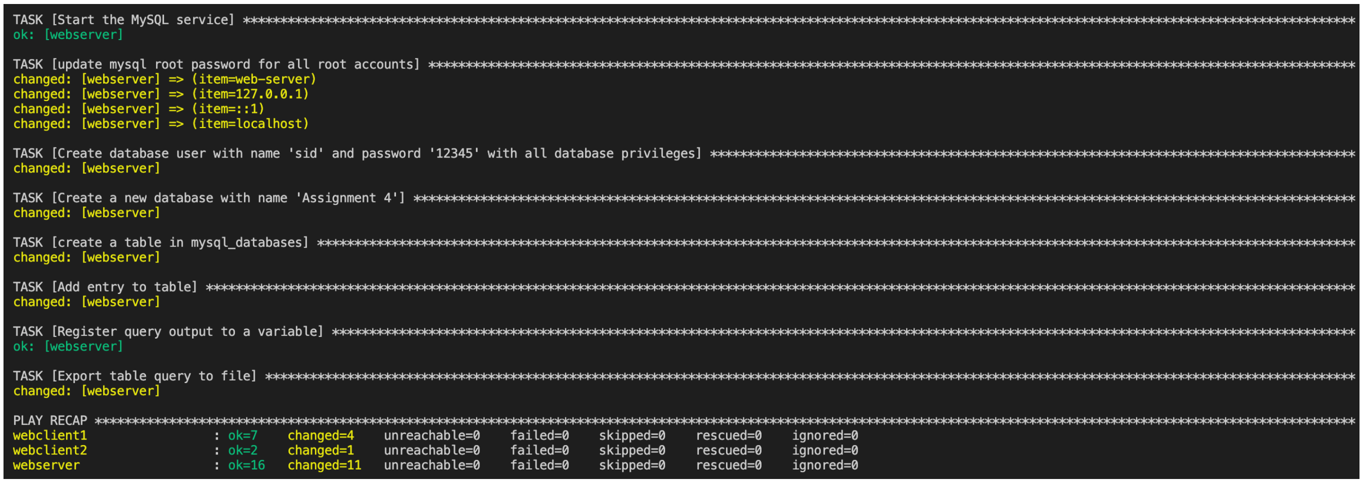
```
---
### Install Apache2

- name: Installing apache2
  hosts: webserver
  become: yes
  tasks:
    - name: install apache2
      apt: name=apache2 update_cache=yes state=latest

    - name: enabled mod_rewrite
      apache2_module: name=rewrite state=present
      notify:
        - restart apache2

  handlers:
    - name: restart apache2
```

```
      service: name=apache2 state=restarted

#### Play to Copy file from webserver to webclient

- name: Installing sshpass for running scp
  hosts: assignment
  become: yes
  tasks:
    - name: install sshpass
      apt: name=sshpass update_cache=yes state=latest

- name: Configure SSH configuration
  hosts: webclient1
  tasks:
  - name: Creating a ssh config file with content to bypass ECDSA fingerprint
    copy:
      dest: "~/.ssh/config"
      content: |
        Host *
        StrictHostKeyChecking no
        UserKnownHostsFile=/dev/null

- name: Copy file from webserver to webclient
  hosts: webclient1
  tasks:
    - name: Copy file to another server
      shell: "sshpass -p vagrant scp vagrant@192.168.50.4:/var/www/html/index.html /home/vagrant/ |cat"

- name: Copy index.html from webclient1 to base machine
  hosts: webclient1
  gather_facts: no

  tasks:
  - name: ansible copy file from remote to local.
    fetch:
      src: /home/vagrant/index.html
      dest: index.html
      flat: yes

###### Installing and setting up MySql ####### We won't add sshpass because it makes it too easy for novice SSH users to ruin SSH's security.
- hosts: webserver
  vars:
    mysql_root_password: test
  tasks:
    - name: Install MySQL
      apt: name={{ item }} update_cache=yes cache_valid_time=3600 state=present
      become: yes
      with_items:
        - python-mysqldb
        - mysql-server

    - name: Start the MySQL service
      become: yes
      service:
        name: mysql
        state: started
        enabled: true

    - name: update mysql root password for all root accounts
      become: yes
      mysql_user:
        name: root
        host: "{{ item }}"
        password: "{{ mysql_root_password }}"
        login_user: root
        login_password: "{{ mysql_root_password }}"
        check_implicit_admin: yes
        priv: "*.*:ALL,GRANT"
      with_items:
        - "{{ ansible_hostname }}"
        - 127.0.0.1
        - ::1
        - localhost

    - name: Create database user with name 'sid' and password '12345' with all database privileges
      community.mysql.mysql_user:
        login_user: root
        login_password: "{{ mysql_root_password }}"
        name: sid
        password: 12345
          priv: "*.*:ALL"
          state: present

      - name: Create a new database with name 'Assignment 4'
        community.mysql.mysql_db:
          login_user: root
          login_password: "{{ mysql_root_password }}"
          name: Assignment4
          state: present

      - name: create a table in mysql_databases
        community.mysql.mysql_query:
          login_user: root
          login_password: "{{ mysql_root_password }}"
          login_db: Assignment4
          query: create table Automation (Name varchar(255), Address varchar(255))

      - name: Add entry to table
        community.mysql.mysql_query:
          login_user: root
          login_password: "{{ mysql_root_password }}"
          login_db: Assignment4
          query: insert into Automation values("sid","Test Address")

      - name :  Register query output to a variable
        community.mysql.mysql_query:
          login_user: root
          login_password: "{{ mysql_root_password }}"
          login_db: Assignment4
          query: select * from Automation
        register: result

      - name: Export table query to file
        copy:
         content="{{ result.query_result }}"
         dest=database.txt
        delegate_to: localhost
```

Output

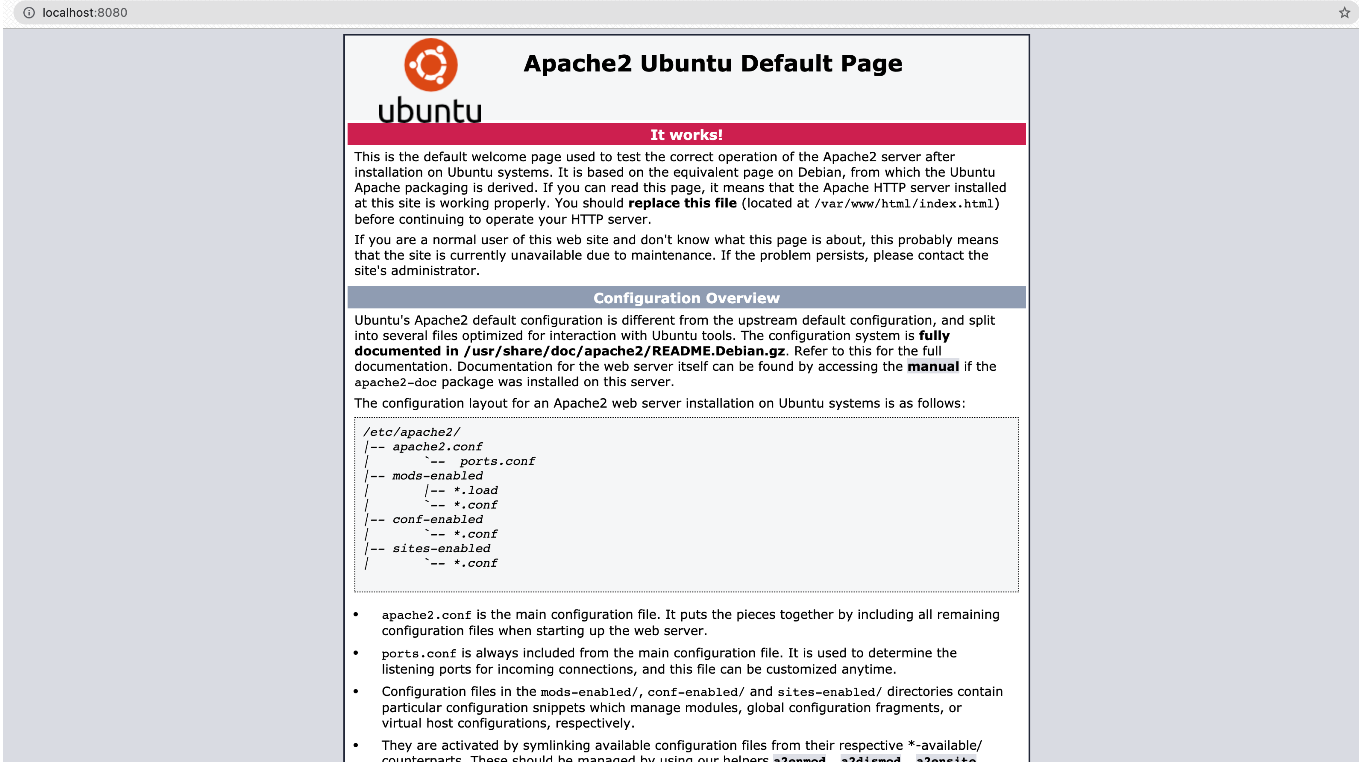


```
PLAY [Installing apache2] **********************************************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************************************************
ok: [webserver]

TASK [install apache2] *************************************************************************************************************************************************************
changed: [webserver]

TASK [enabled mod_rewrite] *********************************************************************************************************************************************************
changed: [webserver]

RUNNING HANDLER [restart apache2] **************************************************************************************************************************************************
changed: [webserver]

PLAY [Installing sshpass for running scp] ******************************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************************************************
ok: [webserver]
ok: [webclient1]
ok: [webclient2]

TASK [install sshpass] *************************************************************************************************************************************************************
changed: [webserver]
changed: [webclient2]
changed: [webclient1]

PLAY [Configure SSH configuration] *************************************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************************************************
ok: [webclient1]

TASK [Creating a ssh config file with content to bypass ECDSA fingerprint] *********************************************************************************************************
changed: [webclient1]

PLAY [Copy file from webserver to webclient] ***************************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************************************************
ok: [webclient1]

TASK [Copy file to another server] *************************************************************************************************************************************************
changed: [webclient1]

PLAY [Copy index.html from webclient1 to base machine] *****************************************************************************************************************************

TASK [ansible copy file from remote to local.] *************************************************************************************************************************************
changed: [webclient1]

PLAY [webserver] *******************************************************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************************************************
ok: [webserver]

TASK [Install MySQL] ***************************************************************************************************************************************************************
changed: [webserver] => (item=python-mysqldb)
changed: [webserver] => (item=mysql-server)

TASK [Start the MySQL service] *****************************************************************************************************************************************************
ok: [webserver]
```



```
TASK [Start the MySQL service] *****************************************************************************************************************************************************
ok: [webserver]

TASK [update mysql root password for all root accounts] ****************************************************************************************************************************
changed: [webserver] => (item=web-server)
changed: [webserver] => (item=127.0.0.1)
changed: [webserver] => (item=::1)
changed: [webserver] => (item=localhost)

TASK [Create database user with name 'sid' and password '12345' with all database privileges] **************************************************************************************
changed: [webserver]

TASK [Create a new database with name 'Assignment 4'] ******************************************************************************************************************************
changed: [webserver]

TASK [create a table in mysql_databases] *******************************************************************************************************************************************
changed: [webserver]

TASK [Add entry to table] **********************************************************************************************************************************************************
changed: [webserver]

TASK [Register query output to a variable] *****************************************************************************************************************************************
ok: [webserver]

TASK [Export table query to file] **************************************************************************************************************************************************
changed: [webserver]

PLAY RECAP *************************************************************************************************************************************************************************
webclient1                 : ok=7    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
webclient2                 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
webserver                  : ok=16   changed=11   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Webpage:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

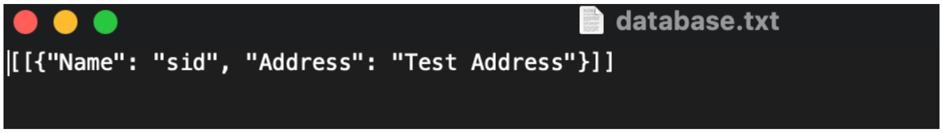
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

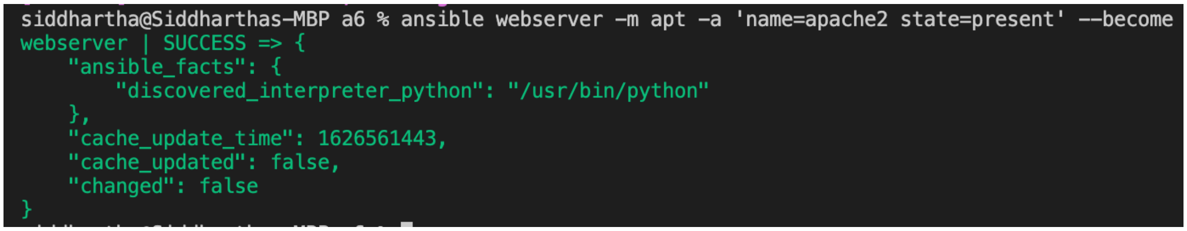
- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers **a2enmod**, **a2dismod**, **a2ensite**

Database.txt:

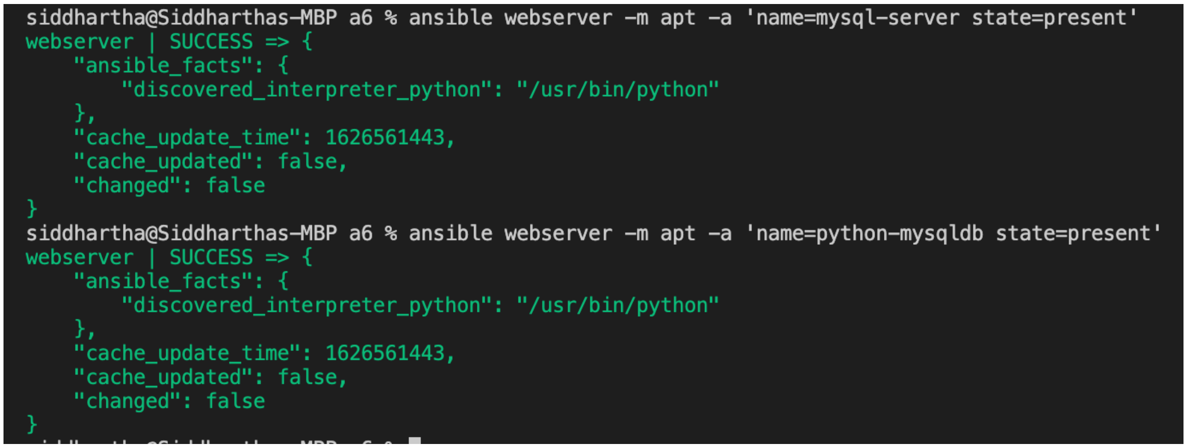


```
[[{"Name": "sid", "Address": "Test Address"}]]
```

Adhoc:



```
siddhartha@Siddharthas-MBP a6 % ansible webserver -m apt -a 'name=apache2 state=present' --become
webserver | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "cache_update_time": 1626561443,
    "cache_updated": false,
    "changed": false
}
siddhartha@Siddharthas-MBP a6 %
```



```
siddhartha@Siddharthas-MBP a6 % ansible webserver -m apt -a 'name=mysql-server state=present'
webserver | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "cache_update_time": 1626561443,
    "cache_updated": false,
    "changed": false
}
siddhartha@Siddharthas-MBP a6 % ansible webserver -m apt -a 'name=python-mysqldb state=present'
webserver | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "cache_update_time": 1626561443,
    "cache_updated": false,
    "changed": false
}
```