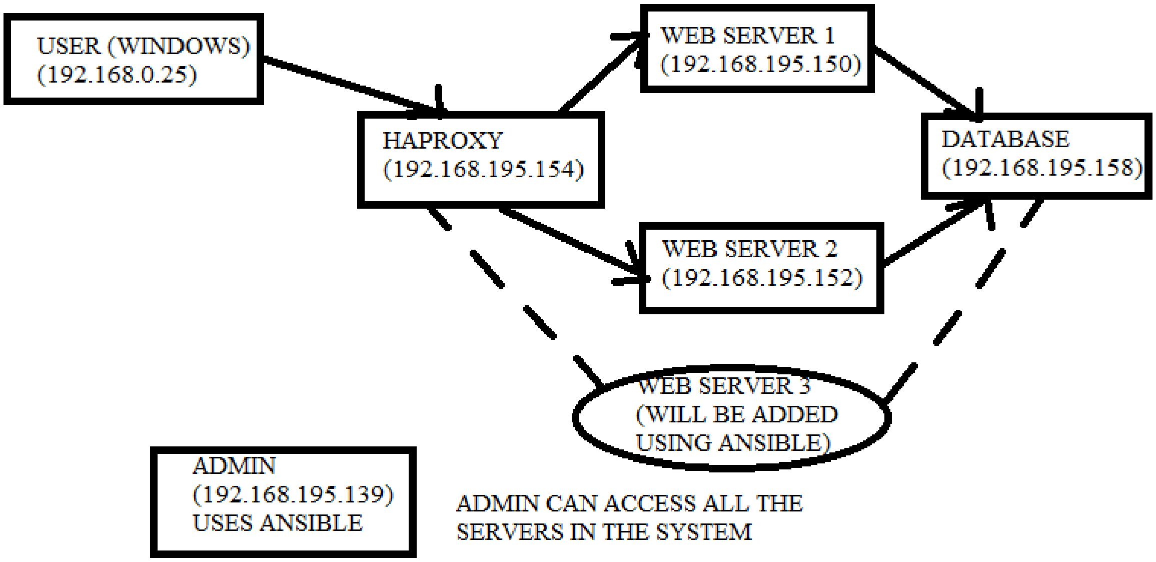
NUAGE WEB APPLICATION ASSIGNMENT

(Note: I've used Ubuntu Web Servers installed in vmPlayer for this assignment)

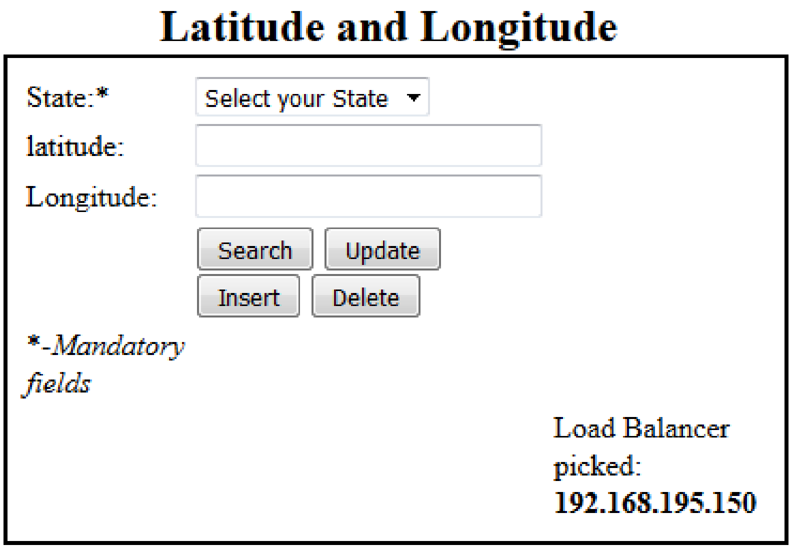
5 PARTS of the Assignment are as below:

1. **Web Application**
2. **Python Code to interact with API**
3. **HAProxy Setup (Load balance)**
4. **IPTables (Firewall)**
5. **Ansible (Automate Everything)**

THE OVERALL DESIGN IS:



1. **Web Application:**
   I’ve developed a simple application to search for Latitude and Longitude of a state listed in the drop box.



Latitude and Longitude

State:* Select your State

latitude:

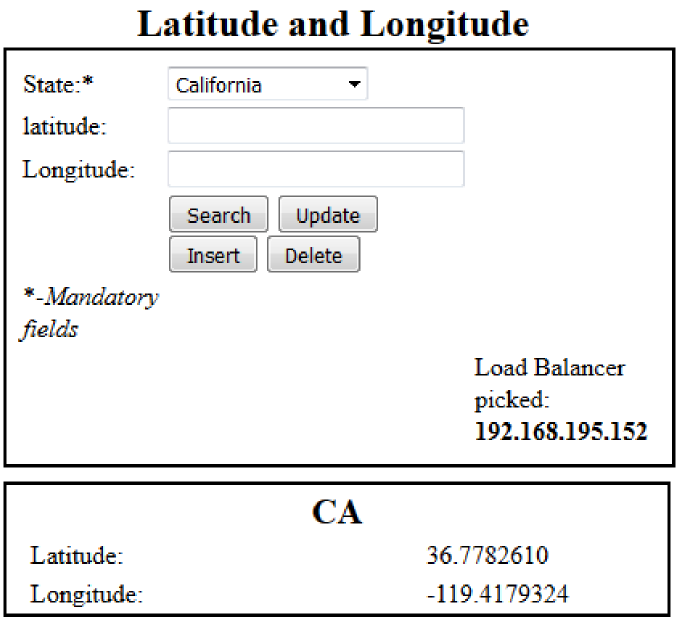
Longitude:

*-Mandatory fields

Load Balancer picked: **192.168.195.150**

Functionalities:
A state’s latitude and longitude can be

a. Searched (GET)
b. Updated (PUT)
c. Inserted (POST)
d. Deleted (DELETE)



Latitude and Longitude

State:* California

latitude:

Longitude:

*-Mandatory fields

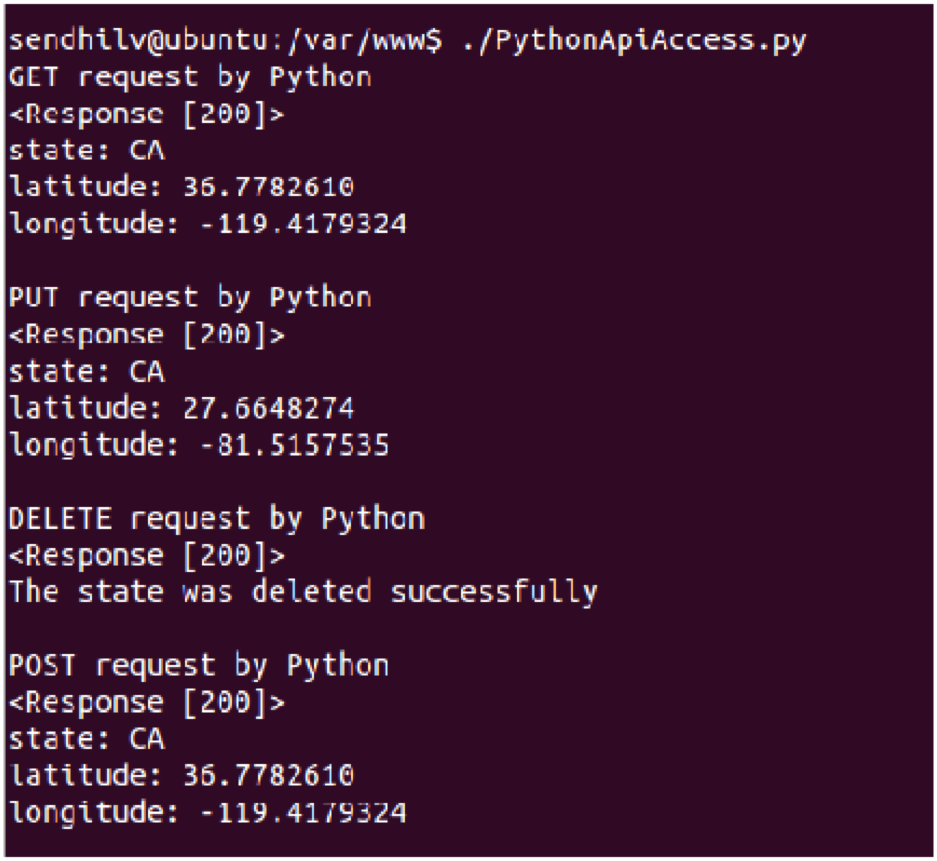
Load Balancer picked: **192.168.195.152**

CA

Latitude:	36.7782610
Longitude:	-119.4179324

2. Python Script for interacting with this API was developed. /var/www/PythonApiAccess.py

   Execute: ./PythonApiAccess.py



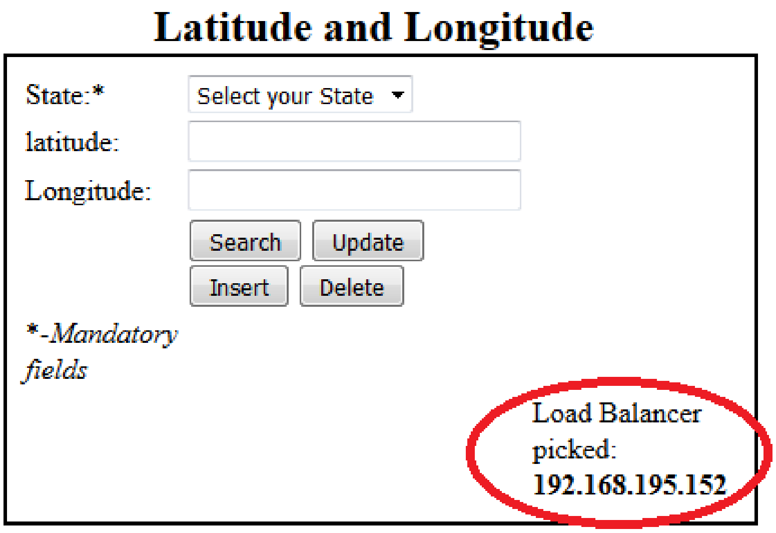
```
sendhilv@ubuntu:/var/www$ ./PythonApiAccess.py
GET request by Python
<Response [200]>
state: CA
latitude: 36.7782610
longitude: -119.4179324

PUT request by Python
<Response [200]>
state: CA
latitude: 27.6648274
longitude: -81.5157535

DELETE request by Python
<Response [200]>
The state was deleted successfully

POST request by Python
<Response [200]>
state: CA
latitude: 36.7782610
longitude: -119.4179324
```

3. HAProxy:

The HAProxy performd load balancing between all the web servers.
Load Balancer: 192.168.195.154 (balances 192.168.195.150, 192.168.195.152).
If a new webserver is added via ansible, it will automatically be load balanced.

If I reload the page http://192.168.195.154/LatLongSearch.php, we can see that it picks the 2 webservers alternatively.

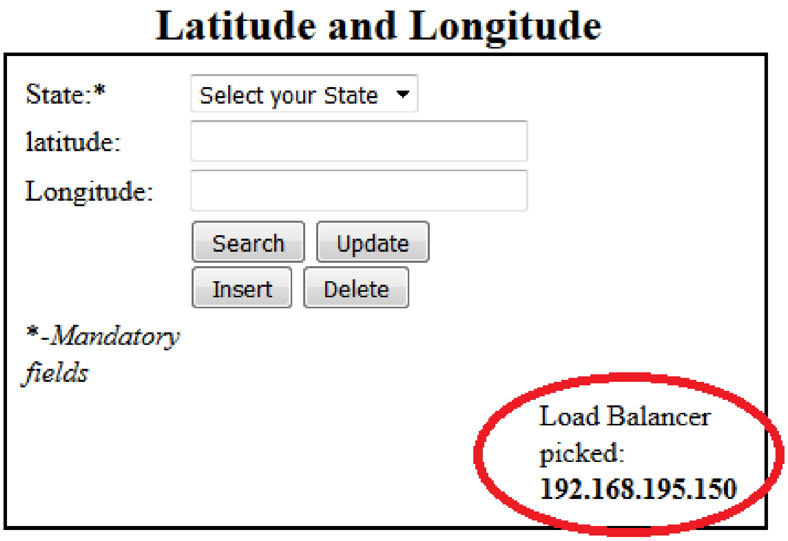
When page first loads:



Latitude and Longitude

Load Balancer picked: **192.168.195.152**

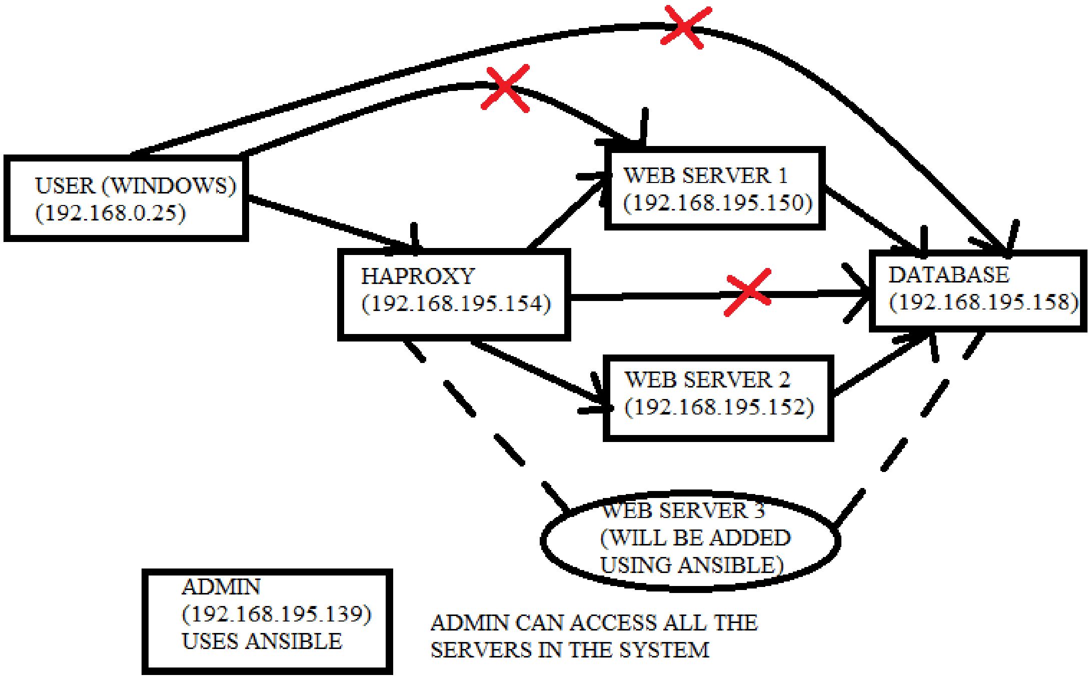
When page reloaded:



Latitude and Longitude

Load Balancer picked: **192.168.195.150**

4. IPTables (Firewall):
   By configuring the IPtables of web servers, haproxy and database we ensure the below scenario.



5. Ansible Network Automation:

   1. With Admin ready with the all the above files
   2. With 4 servers installed.

      One for database server
      Two for webservers
      One for Haproxy

      (we have to list them in the proper categories of the 'host' file)
   3. If we run "ansible-playbook --ask-pass -u root Admin.yml --ask-sudo-pass", the entire setup is completely setup by the ansible.

      Log file is:

      Ansible_Log_For_Entire_Setup.txt
   4. If we add one more Ubuntu server in the [webservers] list and run the same ansible program. Without any other effort, the new Ubuntu server is added as the web server in the web server farm.

Ansible File Structure:

- ansible.cfg
- hosts
- Admin.yml
- Roles folder
    - init **(the admin is added as sudo user for full privilege)**
        - tasks
            - main.yml
    - installations **(Complete installation for the webservers and mysql is done here)**
        - handlers
            - main.yml
        - tasks
            - main.yml
        - templates
            - iptables_basic_config.j2
    - copyFiles **(Copy all the Webserver Files from admin to all the web servers)**
        - Tasks
            - main.yml
        - files
            - WebServerFiles (LatLongSearch.php,
              PythonApiAccess.py
              phpinfo.php,
              test.php
              api.php)
    - Haproxy **(install Haproxy, config the haproxy and set default iptables config)**
        - handlers
            - main.yml
        - tasks
            - main.yml
        - templates
            - haproxy_config.j2
            - haproxy_iptables.j2
    - haproxy_iptables **(Config the Haproxy to establish firewall)**
        - handlers
            - main.yml
        - tasks
            - main.yml
    - webserver_iptables **(Config the WebServers to establish firewall)**
        - handlers
            - main.yml
        - tasks
            - main.yml
    - mysql_iptables **(Config the Mysql to establish firewall)**
        - handlers
            - main.yml
        - tasks
            - main.yml

DEMO COVERAGE:

1. Will run the entire setup ansible program.
2. Then add one more new web server using ansible (completely by automation.)
3. Load Balancing – shown by reloading the page.
4. IPtables access blocked in the wrong cases.
5. Python script execution.
6. Search using the web page and perform all 4 actions (GET, PUT, POST and DELETE).