**Network Automation**

Automation configuration of Nginx webservers and Haproxy load balancer using ansible playbook. The above files contain:

1. Site.yaml: To run Ansible playbook.
2. Hosts: Inventory file.
3. Ssh-config: configuration for ssh bastion.
4. default: Nginx configuration file to run php code.
5. Haproxy.cfg.j2: Haproxy configuration file using jinja2 template.
6. Index.php: simple php file which shows hostname, ip and port.

This assignment is to implement a simple web service using Nginx webserver and Haproxy load balancer. The basic architecture of running webserver:

Haproxy

Ssh Bastion

Webserver 3

Webserver 2

Webserver 1

user

user

**Ssh Bastion**: The ssh bastion host acts as entry point for the webservers. The webservers use only private addresses. If certified users are connected to ssh bastion then they have access to webservers. Ssh bastion uses public ip address to connect to outside world. The ssh bastion is used for security purposes for webservers. The information in the webservers will be secure and no outside user can breach, and miss use the information in the webservers.

user

Webservers

(private ip address)

Ssh bastion

(public ip address)

**Haproxy:** Haproxy acts as load balancer to reduces the load on the single webserver. When a user’s requests for service without Haproxy then the load will not be shared to all servers, to reduce the load on single webserver Haproxy was used.

**How to run ansible yaml file:**

Step1: pull this git repository

git pull https://github.com/vamsiveeramreddy/NetworkAutomation-.git

Step-2: create cloud instances with ssh key and create bastion host for all servers.

(Creating ssh key: ssh-keygen)

Copy ssh-config file at your .ssh location and change server names to your config file. Change the ip address in the ssh-config with your instances ip addresses.

Step-3: change inventory file with your server names and add ip address if it is required for your servers.

Step-4: run the ansible site.yaml file.

Ansible-playbook site. Yaml

**Nginx default file:** Mainly Nginx uses the default file in sites-enable to run configuration. The sites-enable default file links with sites-available. In the ansible playbook we change this default file in the sites-available and links it to the sites-enable.

The Nginx server listens on port 80 and it checks the php location from /var/www/html. The default Nginx configuration file is used to run the .php files.

**Haproxy config file:** The default Haproxy config file contains the information of the backend servers and frontend information. The edited Haproxy config file should be located at /etc/haproxy/haproxy.cfg.

Here to run automate haproxy through ansible, jinja2 template is used to take the server address automatically form ansible variables. To access and read the ip address of the other hosts there a template called **ansible\_default\_ipv4** which access the ip address of the other hosts(webservers) and adds to it backend servers.

The outputs of this ansible playbook, Nginx server and haproxy load balancer are shown in images.