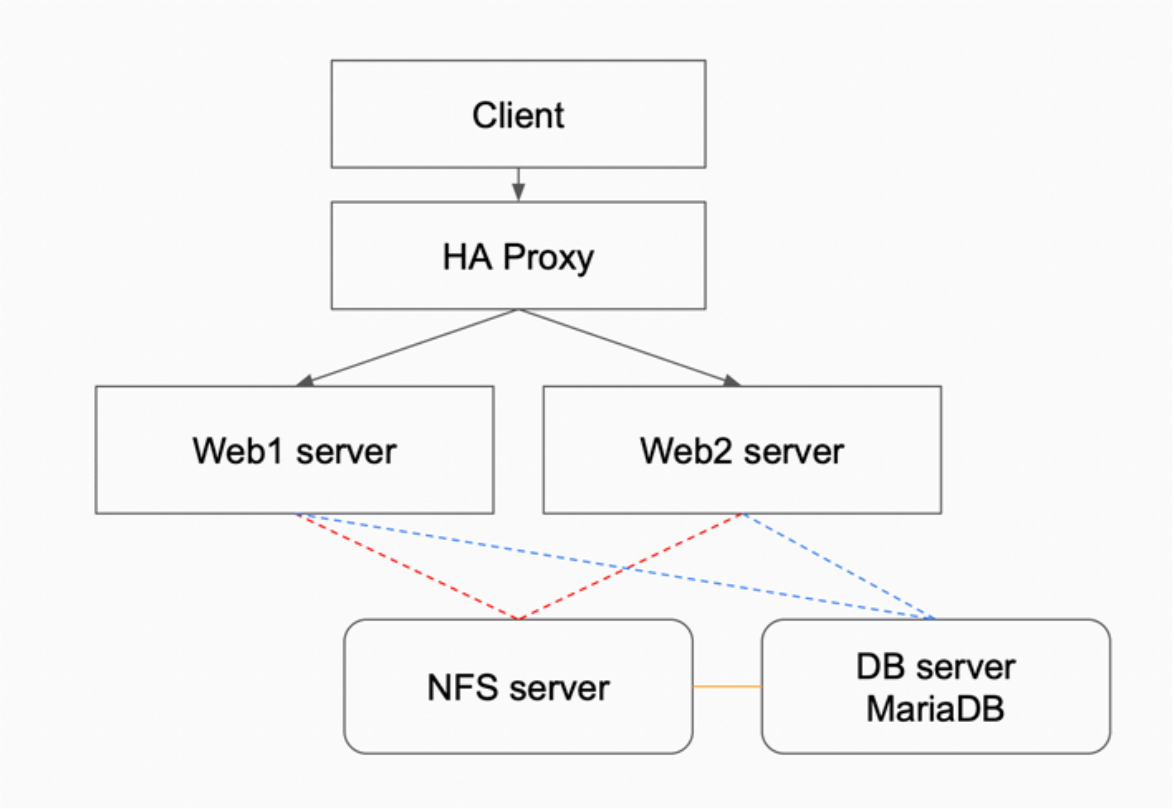


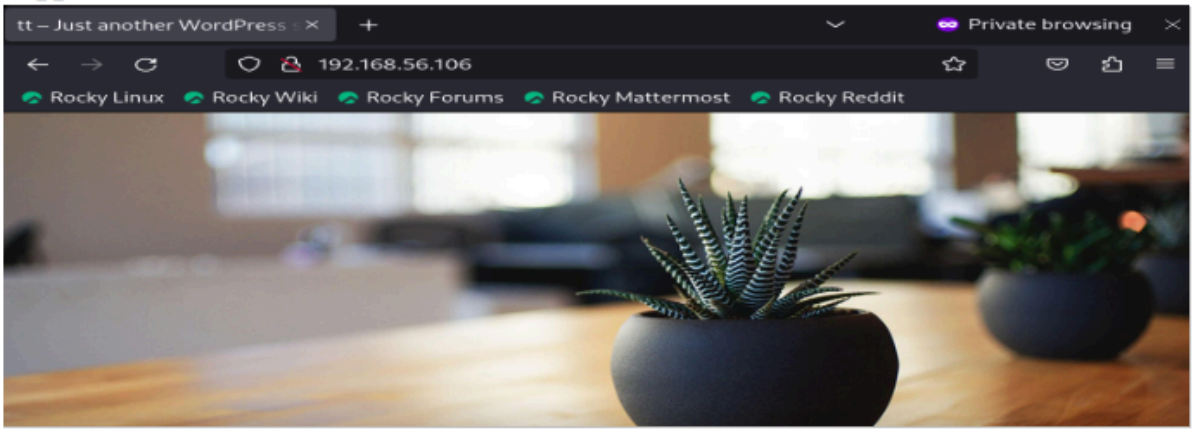
Iac를 통한 **Wordpress** 구성

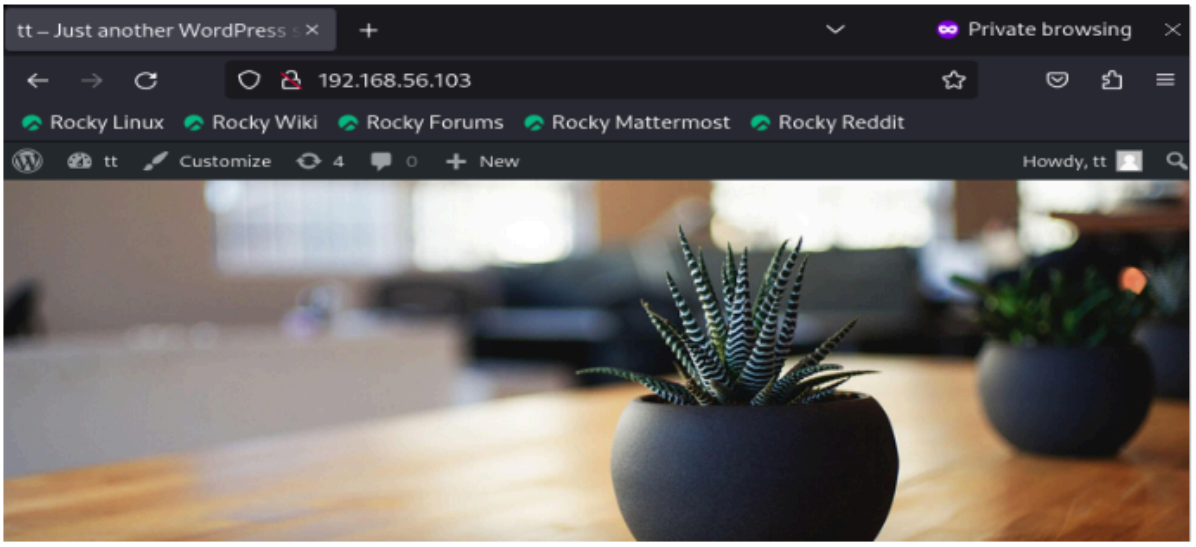
프로젝트 목표
2개의 워드프레스 서버를 haproxy를 이용해 로드밸런싱을 사용하고 하나의 db를 연동함
HA Proxy server 구성, Web1, Web2 server 구성, DB server 구성, NFS server 구성

개발환경
제어노드 Ubuntu - 192.168.56.102
관리노드 (HA Proxy) CentOS 1 - 192.168.56.101
관리노드 (web1) CentOS 2 - 192.168.56.103
관리노드 (web2) CentOS 3 - 192.168.56.106
관리노드 (DB) CentOS 4 - 192.168.56.107

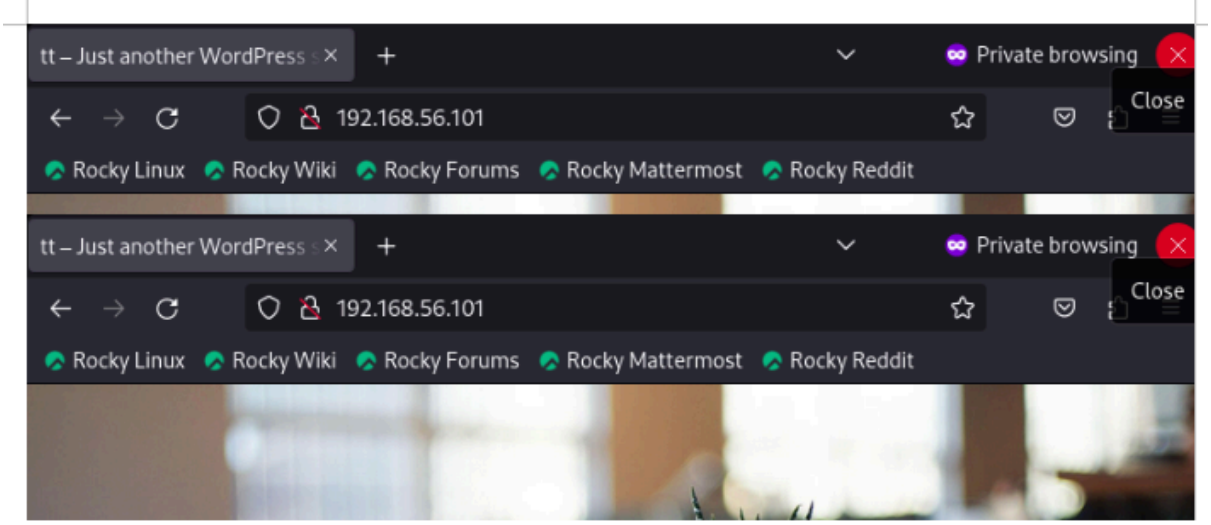
동작

web server 구성 동작





HA Proxy 를 통한 로드밸런싱



haproxy.cfg 내용에
listen stats
mode http
bind *:8080
stats enable
stats show-legends
stats uri /haproxy-status
추가 입력하여 로드밸런싱을 확인 할수있습니다.
위 사진을 보면 sessions의 total칸이 로드밸런싱 되어 번갈아가며
올라가는것을 볼수있습니다.

web_servers

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	L
app1	0	0	-	0	1		0	1	-	4	4	4s	1 418	108 126		0		0	0	0	0	4m31s UP	L4
app2	0	0	-	0	1		0	1	-	3	3	6s	1 038	160 974		0		0	0	0	0	4m31s UP	L4
Backend	0	0		0	1		0	1	300	7	7	4s	2 456	269 100	0	0		0	0	0	0	4m31s UP	

web_servers

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	L
app1	0	0	-	0	1		0	1	-	4	4	2m8s	1 418	108 126		0		0	0	0	0	6m35s UP	L4
app2	0	0	-	0	1		0	1	-	4	4	2s	1 384	214 632		0		0	0	0	0	6m35s UP	L4
Backend	0	0		0	1		0	1	300	8	8	2s	2 802	322 758	0	0		0	0	0	0	6m35s UP	

web_servers

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	L
app1	0	0	-	0	1		0	1	-	5	5	2s	1 764	161 784		0		0	0	0	0	7m3s UP	L4
app2	0	0	-	0	1		0	1	-	4	4	30s	1 384	214 632		0		0	0	0	0	7m3s UP	L4
Backend	0	0		0	1		0	1	300	9	9	2s	3 148	376 416	0	0		0	0	0	0	7m3s UP	

기초 설정

일단 haproxy server 세팅을 위해서 각 서버 별 기본 설정이 필요합니다. 우분투를 사용하여 4개의 CentOS 서버를 제어합니다. 그러기 위해서 제어 서버에 각 서버 별 ssh 키를 등록 시켜줍니다. 키 등록 후 asible.cfg와 inventory를 해당 서버에 맞게 수정했습니다. 그 이후 서버들의 selinux를 disable 하기 위해 selinux.yml 플레이북 파일을 만들어 실행합니다.

```
Ubuntu@192.168.56.102 server
#ssh-keygen -t rsa
#ssh-copy-id devops@192.168.56.101
#ssh-copy-id devops@192.168.56.103
#ssh-copy-id devops@192.168.56.106
#ssh-copy-id devops@192.168.56.107

vim ansible.afg
[defaults]
remote_user = devops
inventory = ./inventory
[privilege_escalation]
become = true
become_method = sudo
become_user = root
become_ask_pass = false

vim inventory
[internetweb]
192.168.56.101
[intranetweb]
192.168.56.103
192.168.56.106
[mariadbserver]
192.168.56.107
```

```
SELinux.yml
---
- name: disabled selinux
  hosts: all
  tasks:
    - name: Disable SELinux
      ansible.posix.selinux:
        state: disabled
```

haproxy server setting

Haproxy server setting을 위해 ansible-playbook haproxy.yml파일을 작성합니다.
haproxy.yml구성에 사용한 주요 코드

```
---
- name: haproxy setting
  hosts: 192.168.56.101
  tasks:
    - name: haproxy install
      yum:
        name:
          - firewalld
          - haproxy
        state: present
    - name: haproxy chage
      template:
        src: haproxy.cfg.j2
        dest: /etc/haproxy/haproxy.cfg
    - name: haproxy enabled
      service:
        name: haproxy
        enabled: true
        state: started
    - name: firewalld enabled and running
      service:
        name: firewalld
        enabled: true
        state: started
    - name: allow port http
      firewalld:
        port: 80/tcp
        permanent: true
        state: enabled
        immediate: yes
      notify:
       - Restart HAProxy
  handlers:
    - name: Restart HAProxy
      service:
        name: haproxy
        state: restarted
```

플레이북 중간에 사용한 haporxy.cfg.j2는 다음과 같습니다.

```
...
#---------------------------------------------------------------------
# main frontend which proxys to the backends
#---------------------------------------------------------------#frontend  main *:5000
#    acl url_static       path_beg       -i /static /images /javascript /stylesheets
#    acl url_static       path_end       -i .jpg .gif .png .css .js

#    use_backend static          if url_static
#    default_backend             app
#---------------------------------------------------------------------
# static backend for serving up images, stylesheets and such
#---------------------------------------------------------------------
#backend static
#    balance     roundrobin
#    server      static 127.0.0.1:4331 check
#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
frontend http_site
bind *:80
default_backend web_servers

backend web_servers
 balance     roundrobin
 server  app1 192.168.56.103:80 check  #//web1의 서버아이피 주소
 server  app2 192.168.56.106:80 check  #//web2의 서버아이피 주소
```

web1, web2 server setting

Web server 두 개를 구성하기 위해 가상머신 192.168.56.103과 192.168.56.106를 사용하였습니다. 마찬가지로 Web server 구성을 위해 플레이북 http.yml, wordpress.yml, nfsclient.yml을 작성하였습니다.

```
Http.yml
---
- name: Install and start HTTPD
  hosts: intranetweb
  tasks:
    - name: httpd and firewalld is present
      yum:
        name:
          - httpd
          - php
          - firewalld
        state: present
    - name: firewalld enabled and running
      service:
        name: firewalld
        enabled: true
        state: started
    - name: firewalld permits access to httpd
service
      firewalld:
        service: http
        permanent: true
        state: enabled
        immediate: yes
    - name: httpd is started
      service:
        name: httpd
        state: started
        enabled: true
    - name: allow port 80
      firewalld:
        port: 80/tcp
        permanent: yes
        immediate: yes
        state: enabled
```

```
    - name: vhost.conf
      template:
        src: vhost.conf.j2
        dest: /etc/httpd/conf.d/00-vhost.conf
    - name: httpd.conf
      template:
        src: httpd.conf.j2
        dest: /etc/httpd/conf/httpd.conf

- name: web1server index
  hosts: 192.168.56.103
  tasks:
    - name: web1server index file
      copy:
        content: '<a href="/wordpress">Go to WordPress Page</a>'
        dest: /var/www/html/index.html
- name: web2server index
  hosts: 192.168.56.106
  tasks:
    - name: web2server index file
      copy:
        content: '<a href="/wordpress">Go to WordPress Page</a>'
        dest: /var/www/html/index.html
      notify:
        - Restart httpd
  handlers:
    - name: Restart httpd
      service:
        name: httpd
        state: restarted
- name: web1server index
  hosts: 192.168.56.103
  tasks:
    - name: web1server index file
      copy:
        content: '<a href="/wordpress">Go to WordPress Page</a>'
        dest: /var/www/html/index.html
```

```
- name: web2server index
  hosts: 192.168.56.106
  tasks:
    - name: web2server index file
      copy:
        content: '<a href="/wordpress">Go to WordPress Page</a>'
        dest: /var/www/html/index.html
      notify:
        - Restart httpd
  handlers:
    - name: Restart httpd
      service:
        name: httpd
        state: restarted
```

Wordpress.yml

```yaml
---
- name: install wordpess php php-mysql
  hosts: intranetweb
  become: yes
  tasks:
    - name: php and php-mysql is present
      yum:
        name:
          - php
          - php-mysql
        state: present
    - name: download wordpress source
      get_url:
        url: https://wordpress.org/wordpress-4.9.18.tar.gz
        dest: /var/www/html
    - name: unzip
      unarchive:
        src: /var/www/html/wordpress
4.9.18.tar.gz
        remote_src: yes
        dest: /var/www/html/
    - name: copy wp-config.php
      template:
      src: wp-config.php.j2
        dest: /var/www/html/wordpress/wp-config.php
```

Nfsclient.yml

```yaml
---
- name: nfs client setup
  hosts: intranetweb
  gather_facts: no
  become: yes
  tasks:
    - name: nfs install
      yum:
       name: nfs-utils
       state: latest
    - name: start nfs
      service:
        name: nfs-utils
        enabled: true
        state: started
    - name: firewalld permits access to nfs service
      firewalld:
        service: nfs
        permanent: true
        state: enabled
        immediate: yes
    - name: mkdir nfs
      file:
        path: /var/www/html/wordpress
        state: directory
    - name: make mountpoint
      mount:
        path: /var/www/html/wordpress
        src: 192.168.56.101:/home/mnt/storage
        fstype: nfs
        opts: rw,noatime
        state: mounted
```

db server setting

Wordpress 동작을 위해서 접속을위한 db server가 필요합니다. 그렇기에 192.168.56.107 서버를 사용하여 db server를 구성하였습니다. 서버에 mariadb를 설치하고Web1, Web2 가 접속을 위해서 접근가능한 mysql 유저 생성을 위한mariadb.yml 플레이북을 작성 했습니다.

```
Mariadb.yml
- name: maria
  hosts: 192.168.56.107
  become: yes
  tasks:
    - name: Install MariaDB
      yum:
        name:
          - mariadb-server
          - MySQL-python
        state: present

    - name: Start MariaDB service
      service:
        name: mariadb
        state: started
        enabled: yes

    - name: firewalld permits access to mysql service
      firewalld:
        service: mysql
        permanent: true
        state: enabled
        immediate: yes

    - name: Ensure root has no password
      mysql_user:
        name: root
        host: localhost
        password: ""
        state: present
```

```
    - name: Remove anonymous MySQL users
     mysql_user:
        name: ''
        host: localhost
        state: absent

    - name: Ensure test database is absent
      mysql_db:
        name: test
        state: absent

    - name: Create application database
      mysql_db:
        name: my_database
        state: present
    - name: Ensure the MySQL user db01 exist
      mysql_user:
        name: db01
        host: "%"
        password: "1234"
        priv: '*.*:ALL'
        state: present

    - name: Ensure the MySQL user root has all privileges
      mysql_user:
        name: root
        host: localhost
        check_implicit_admin: yes
        priv: '*.*:ALL,GRANT'
        state: present

  handlers:
    - name: Restart MariaDB
      service:
        name: mariadb
        state: restarted
```

기타 플레이북

플레이 북에서 템플릿으로 보내기 위한 파일들 입니다.
haproxy.cfg.j2, httpd.conf.d, vhost.conf.j2, wp-config.php.j2

haproxy.cfg.j2

```
…...
frontend http_site
bind *:80
default_backend web_servers

backend web_servers
 balance     roundrobin
 server  app1 192.168.56.103:80 check  #//web1의 서버아이피 주소
 server  app2 192.168.56.106:80 check  #//web2의 서버아이피 주소
 ------------------------------------------------------------
```

vhost.conf.j2

```
…...
<VirtualHost *:80>
DocumentRoot /var/www/html/wordpress
ServerName "{{ ansible_hostname }}.com"
</VirtualHost>
<Directory /var/www/html/wordpress>
AllowOverride None
Require all granted
</Directory>
-----------------------------------------------------------------
```

httpd.conf.d

```
…..
<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
-----------------------------------------------------------------
```

wp-config.php.j2

```
…..
define( 'DB_NAME', 'my_database' );
/** Database username */
define( 'DB_USER', 'db01' );
/** Database password */
define( 'DB_PASSWORD', '1234' );
/** Database hostname */
define( 'DB_HOST', '192.168.56.107' );
/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
-----------------------------------------------------------------
```

NFS server

NFS server를 구성하여 web1,2 server의 wordpress 파일을 한번에 관리하도록 했습니다. 그러기 위해 192.168.56.101을 nfs server로 추가 설정 하기위해 nfs-server.yml 와 client의 nfs-client.yml 플레이북 파일을 작성했습니다.

```
Nfs-server.yml
---
- name: nfs server
  hosts: internetweb
  gather_facts: no
  become: yes

  tasks:
    - name: nfs install
      yum:
       name: nfs-utils
       state: latest
    - name: start nfs
      service:
        name: nfs-utils
        enabled: true
        state: started

    - name: firewalld permits access to nfs service
      firewalld:
        service: nfs
        permanent: true
        state: enabled
        immediate: yes

    - name: mkdir nfs
      file:
        path: /home/mnt/storage
        state: directory
        mode: 0777

    - name: /etc/exports configuration
      lineinfile:
       path: /etc/exports
        line: /home/mnt/storage
192.168.56.103/24(rw,sync,no_root_squash,no_subtree_check)
```

```
    - name: /etc/exports configuration
      lineinfile:
        path: /etc/exports
        line: /home/mnt/storage
192.168.56.106/24(rw,sync,no_root_squash,no_subtree_check)

    - name: exportfs -r
      command: exportfs -r
      register: exportfs_result

    - name: nfs service restart
      service:
        name: nfs
        state: restarted
```

```
Nfs-client.yml
---
- name: nfs client setup
 hosts: intranetweb
 gather_facts: no
 become: yes

 tasks:
 - name: nfs install
      yum:
       name: nfs-utils
       state: latest
    - name: start nfs
      service:
        name: nfs-utils
        enabled: true
        state: started

    - name: firewalld permits access to
nfs service
      firewalld:
        service: nfs
        permanent: true
        state: enabled
        immediate: yes

    - name: mkdir nfs
      file:
        path: /var/www/html/wordpress
        state: directory

    - name: make mountpoint
      mount:
        path: /var/www/html/wordpress
        state: directory

    - name: make mountpoint
      mount:
        path: /var/www/html/wordpress
      src: 192.168.56.101:/home/mnt/storage
        fstype: nfs
        opts: rw,noatime
        state: mounted
```