Laboratorio

“Balanceador en capa 4”

Calle Emilio Fernández N° 296 Of. 601 Urb. Santa Beatriz Lima
Teléfono: (+511)555-0008 (51)982735527

LABORATORIO

Objetivos:

- ✓ Configuración de un balanceador para aplicaciones TCP/UDP.

PROCEDIMIENTO

Previamente al laboratorio, se tiene que contar con 3 o mas servidores contemplando un cluster de galera.

Configuración de NGINX

1. **Validamos que nginx este iniciado en nuestro sistema.**

```
# systemctl status nginx
● nginx.service - nginx - high performance web server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled;
vendor preset: disabled)
   Active: active (running) since Fri 2018-02-18 02:08:20 -05; 18s ago
     Docs: http://nginx.org/en/docs/
  Process: 38471 ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf
(code=exited, status=0/SUCCESS)
 Main PID: 38472 (nginx)
   CGroup: /system.slice/nginx.service
           ├─38472 nginx: master process /usr/sbin/nginx -c
/etc/nginx/nginx.conf
           └─38473 nginx: worker process

Sep 13 02:08:20 localhost.localdomain systemd[1]: Starting nginx -
high performance web server...
```

2. **Y modificamos el archivo principal. Agregamos al final, las líneas siguientes.**
   Este procedimiento nos crea un pool de servidores que lo conforman el cluster de Galera de MariaDB, y tienen que estar dentro de la directiva de Stream ya que estamos utilizando aplicaciones TCP y no son web, por lo cual no pueden estar dentro del bloque HTTP.

```
# vim /etc/nginx/nginx.conf
...
stream {
    upstream cluster_db {
      server 192.168.1.151:3306;
      server 192.168.1.152:3306;
      server 192.168.1.152:3306;
}
```

3. **Por último, agregamos la configuración de server, porque de la misma forma que en HTTP es necesario abrir un puerto para que las solicitudes puedan llegar. De preferencia mantener el mismo puerto.**

```
# vim /etc/nginx/nginx.conf
...
stream {
    upstream cluster_db {
      server 192.168.1.151:3306;
      server 192.168.1.152:3306;
      server 192.168.1.152:3306;
    }

    server {
      listen 3306;
      proxy_pass cluster_db;
      proxy_connect_timeout 1s;
    }
}
```

4. **Validamos la configuración y si todo es correcto reiniciamos el servicio.**

```
# nginx -t
# nginx -s reload
```