

Nginx를 이용한 로드밸런싱

Load Balancing(로드밸런싱)

L4 로드밸런싱과 L7 로드밸런싱

Nginx 설치

nginx.conf 관련 내용

default.conf 관련 내용

도메인 연결

Nginx를 이용한 로드밸런서 설정

upstream 문법

upstream 적용

Nginx를 이용한 HTTP 부하분산

Load balancing methods(부하 분산 규칙)

Round Robin (라운드 로빈)

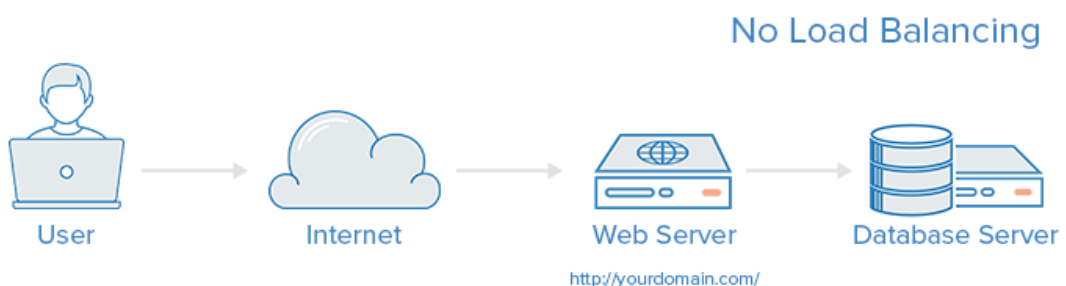
Least-connected(최소연결)

ip_hash

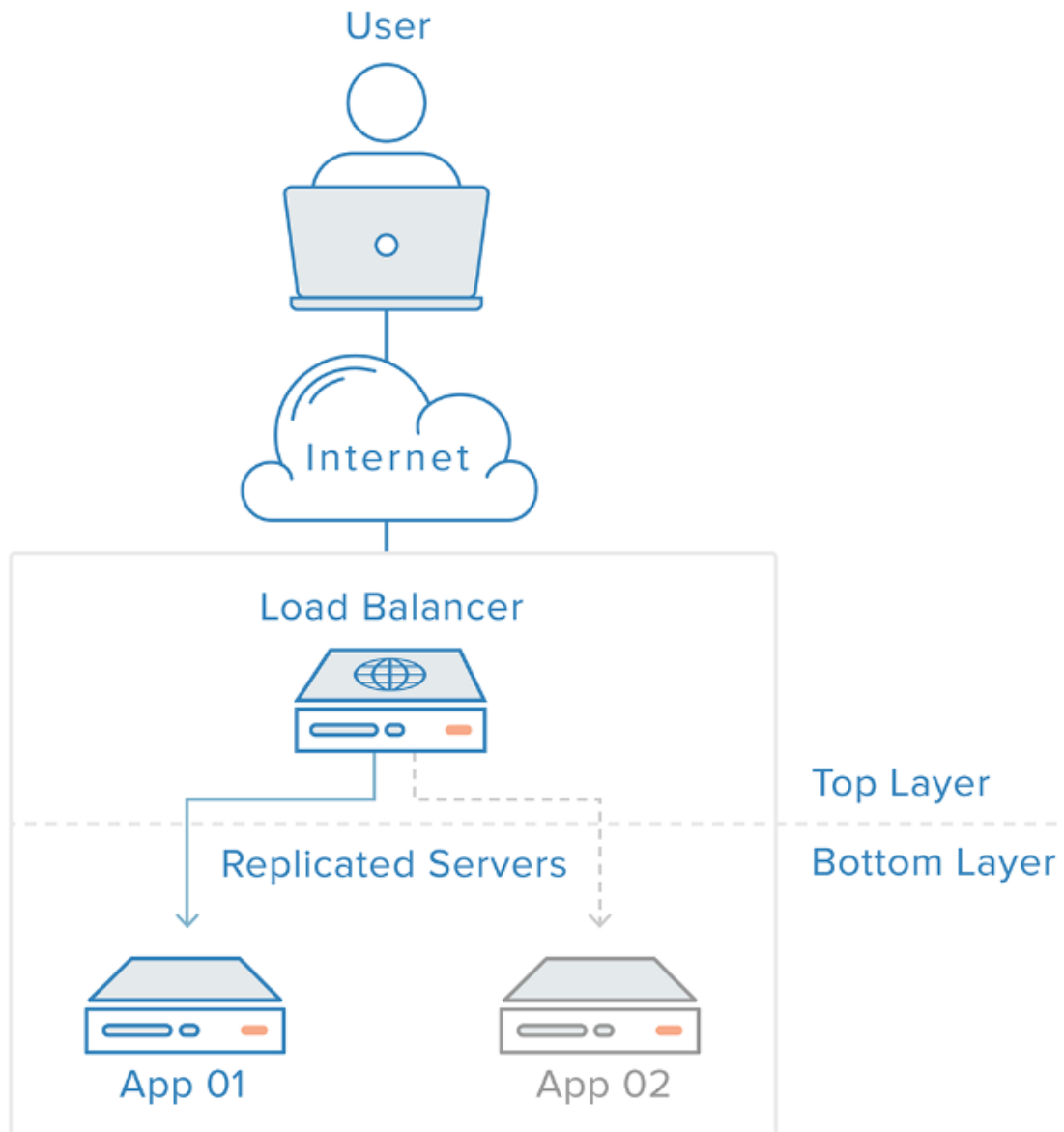
least_time

Load Balancing(로드밸런싱)

Load Balancing(로드밸런싱)은 여러 Server에 workload(작업량)를 분산 시켜 웹사이트, 애플리케이션, 데이터 베이스 그리고 다른 서비스들의 퍼포먼스(performance)와 신뢰성(reliability)을 향상하기 위한 인프라(서비스)입니다.



로드밸런싱을 적용하지 않은 웹서비스



로드밸런싱을 적용한 웹 서비스

L4 로드밸런싱과 L7 로드밸런싱

L4 로드밸런서는 IP, TCP/UDP, port 번호, MAC 주소 등을 바탕으로 부하를 분산한다.

L7 로드밸런서는 URL, payload, cookie, HTTP 헤더 등을 통해 부하를 분산한다.

Nginx 설치

```
# 저장소 생성
[centos@nginx ~]$ sudo vim /etc/yum.repos.d/nginx.repo
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/7/$basearch/
gpgcheck=0
enabled=1

# nginx 설치
[centos@nginx ~]$ sudo yum install -y nginx
....

# 방화벽 설정
[centos@nginx ~]$ sudo firewall-cmd --permanent --zone=public --add-service=http
[centos@nginx ~]$ sudo firewall-cmd --permanent --zone=public --add-service=https
[centos@nginx ~]$ sudo firewall-cmd --reload
```

nginx.conf 관련 내용

```
[centos@nginx ~]$ sudo vim /etc/nginx/nginx.conf
user  nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log notice;
pid        /var/run/nginx.pid;

events {
    worker_connections  1024;
}

http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush      on;

    keepalive_timeout  65;

    #gzip  on;

    include /etc/nginx/conf.d/*.conf;
}
```

error_log : nginx의 에러로그가 쌓이는 경로이다.

pid : nginx의 프로세스 아이디 (pid)가 저장되는 경로이다.

worker_connections : Worker Process가 동시에 처리할 수 있는 접속자의 수를 나타낸다. 기본은 1024로 설정되어져 있다.

include : 포함시킬 외부파일을 정의한다. mime.types란 파일에 작성되어진 내용들을 현재 파일로 가져오는 것을 뜻한다.

default_type : 웹서버의 기본 Content-Type을 정의한다.

log_format : 로그 형식을 지정한다. 후술한 로그 형태에 따라 로그가 작성되고 기록된다.

access_log : 접속 로그가 쌓이는 경로이다.

sendfile : sendfile() 함수의 사용여부를 지정한다. sendfile() 함수는 한 파일의 디스크립터와 다른 파일의 디스크립터 간에 데이터를 복사하는 것으로 커널 내부에서 복사가 진행된다.

default.conf 관련 내용

```
[centos@nginx ~]$ sudo vim /etc/nginx/conf.d/default.conf
server {
    listen      80;
    server_name localhost;

    location / {
        root    /usr/share/nginx/html;
        index   index.html index.htm;
    }

    error_page  500 502 503 504  /50x.html;
    location = /50x.html {
        root    /usr/share/nginx/html;
    }
}
```

listen : 해당 포트로 들어오는 요청을 해당 server { } 블록의 내용에 맞게 처리하겠다는 것을 뜻한다.

server_name : 호스트 이름을 지정한다. 가상 호스트가 있는 경우 해당 호스트명을 써넣으면 된다. 만약 로컬에서 작업하고 있는 내용을 nginx를 통해 띄우려고 하는 경우에는 localhost라고 적으면 된다.

error_page : 요청결과와 http 상태코드가 지정된 http 상태코드와 일치할 경우, 해당 url로 이동한다. 보통 403, 404, 502 등의 에러처리를 위해 사용한다.

url 결과에 따라 이후에 나오는 location = /50x.html과 일치하면 /usr/share/nginx/html 경로에 존재하는 50x.html 파일을 보여준다.

location / : 처음 요청이 들어왔을 때 (**server_name**이 127.0.0.1인 경우 -> 127.0.0.1로 요청이 들어왔을 때) 보여줄 페이지들이 속해있는 경로와 초기 페이지인 index를 지정해준다. / url로 접속했을 경우 index.html, index.htm로 정의된 파일을 보여준다.

도메인 연결

```
[centos@nginx ~]$ sudo vim /etc/nginx/conf.d/default.conf
server {
    listen      80;
    server_name  jin.com www.jin.com;

    #access_log /var/log/nginx/host.access.log  main;

    # 에러나면 밖으로 뺀다.
    location / {
        root    /usr/share/nginx/html;
        index   index.html index.htm;
    }

    [root@localhost ~]# systemctl restart nginx
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Nginx를 이용한 로드밸런서 설정

upstream 문법

```
[centos@nginx ~]$ sudo vim /etc/nginx/nginx.conf
...
http {
    ...
```

```

upstream <업스트림 이름> {
    <로드밸런스 타입: default는 round-robin>
    server <host1>:<port1>
    server <host2>:<port2>
}

```

upstream 적용

```

# DNS 서버 수정
[root@dns /var/named]# vim jin.com.zone
$TTL 1D
@      SOA      @      root. ( 0 1D 1H 1W 3H )
      IN       NS      @
      IN       A       192.168.100.10

www    IN       A       192.168.100.11

# Nginx 서버 설정
# nginx.conf를 적용하기 위해 default.conf파일의 이름을 변경한다.
[centos@nginx ~]$ sudo mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf.backup
[centos@nginx ~]$ sudo vim /etc/nginx/nginx.conf
...
http {
    ...
    upstream web {
        server 192.168.100.101;
        server 192.168.100.102;
    }
    server {
        location / {
            proxy_pass http://web;
        }
    }
}

```

Nginx를 이용한 HTTP 부하분산

```

[centos@nginx ~]$ sudo vim /etc/nginx/nginx.conf
...
http {
    ...
    upstream web {
        server 192.168.100.101:80 weight=1;
        server 192.168.100.102:80 weight=2;
        server 192.168.100.103:80 backup;
    }
    server {
        listen 80;
        server_name www.jin.com;
    }
}

```

```

        location / {
            root html;
            index index.html;
            proxy_pass http://web;
        }
    }
}

```

Load balancing methods(부하 분산 규칙)

Round Robin (라운드 로빈)

- 서버의 가중치를 고려해, 실제 서버들을 처음부터 차례로 선택해 가며 모든 서버로 균등하게 분산 됨 (default)
- 장점 : 거의 균등하게 분산 가능
- 단점 : 경로가 보장 되지 않음

Least-connected(최소연결)

- 서버의 가중치를 고려해, 활성 연결 수가 가장 적은 서버로 요청을 전송 함
- 장점 : 거의 균등하게 분산 가능
- 단점 : 경로 보장 되지 않음

ip_hash

- Hasing key 를 사용하여 IP 별로 Index 를 생성하여 경로를 지정
- 요청이 전송되는 서버는 클라이언트 IP 주소에서 결정됨
- IPv4 주소의 처음 세개의 octets 또는 전체 IPv6 주소가 해시 값을 계산하는 데 사용됨
- 장점 : 경로 보장 가능
- 단점 : 균등한 분산이 어려움

least_time

- 연결 수가 가장 적으면서 평균 응답시간이 가장 적은 쪽을 선택해서 분배 (Nginx plus에서만 가능)