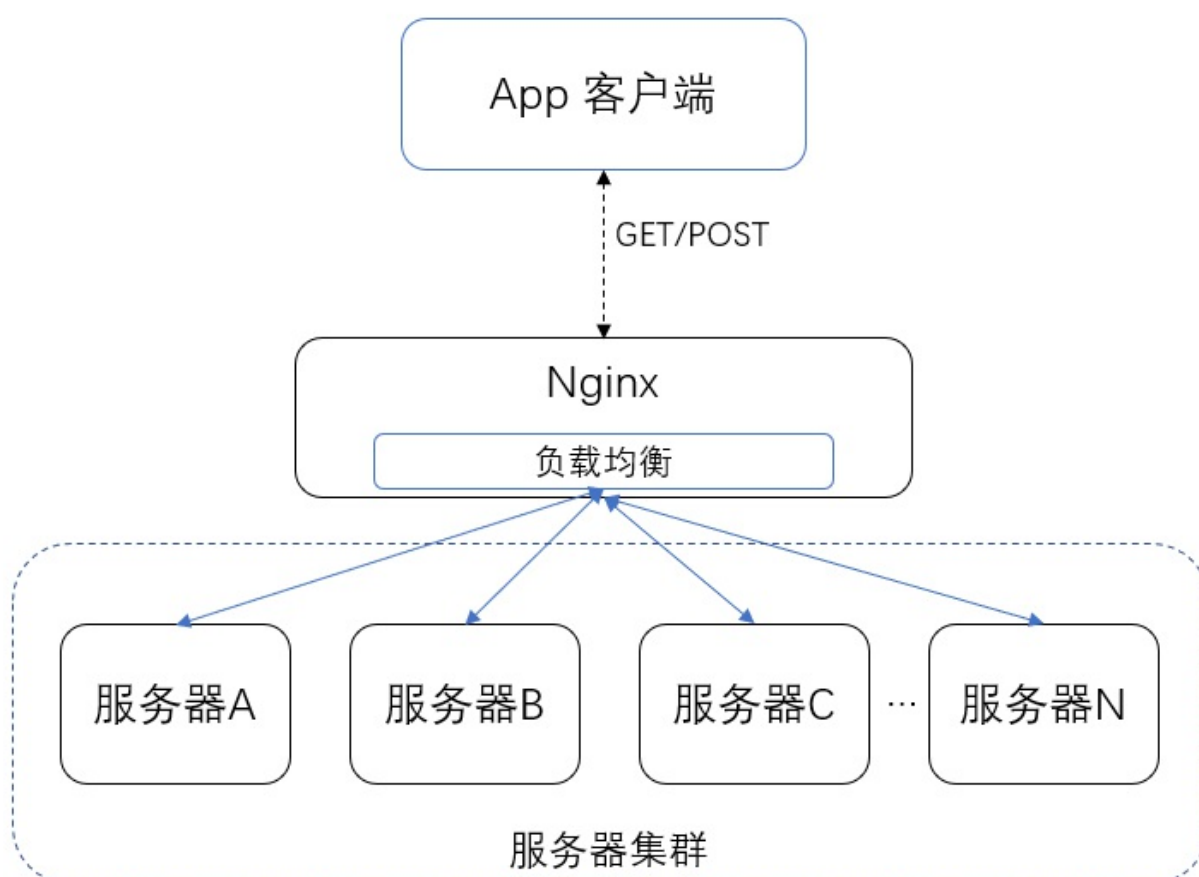


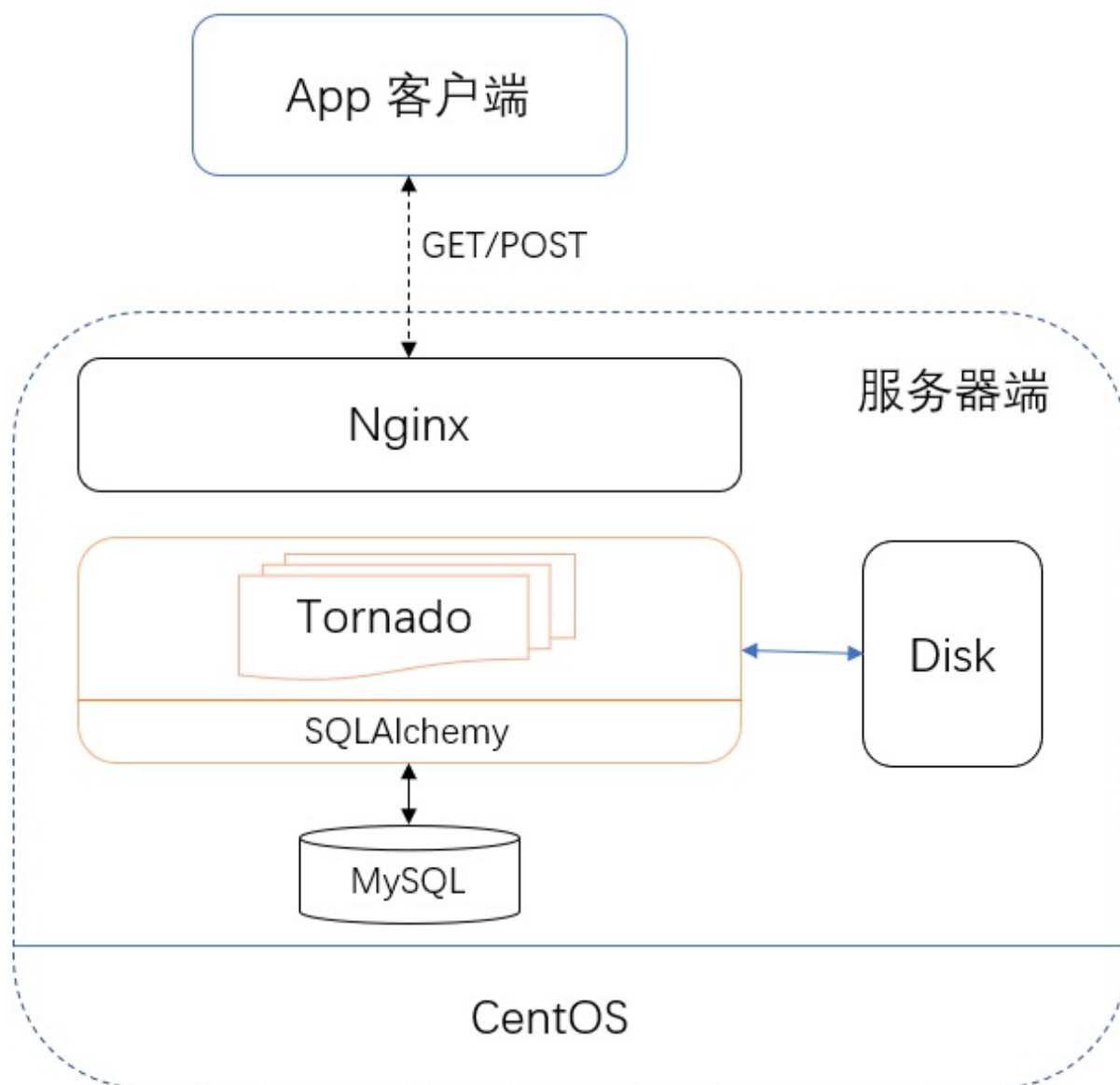
搭建基于 Nginx 的代理服务器

[Nginx \(https://nginx.org/en/\)](https://nginx.org/en/) 是俄罗斯人编写的十分轻量级的 HTTP 服务器，是一个高性能的 HTTP 和反向代理服务器。相较于 [Apache \(https://httpd.apache.org/\)](https://httpd.apache.org/)、[lighttpd \(https://www.lighttpd.net/\)](https://www.lighttpd.net/)，它具有占有内存少、稳定性高等优势。它最常见的用途是提供反向代理服务。

在本小节中，我们将利用 Nginx 的反向代理及负载均衡能力。所谓的负载均衡，是当单台服务器的性能无法满足业务需求时，需横向添加多台服务器；负载均衡就是让访问流量均匀的落在这个服务器集群的每个服务器上。具体逻辑图如下：



App 客户端将请求发送至 Nginx，Nginx 收到请求后，将其转发给后端的服务器集群。在本小册中，我们的 Demo 只有一台虚拟机，现将 Nginx 和后端服务器放在一起，Nginx 架设在 Tornado 之前，其基本框架图如下。



安装 Nginx

输入 `yum install nginx` 安装 Nginx，安装完成后，输入 `nginx -v` 查看 Nginx 是否安装成功。

```
Installed:
  nginx.x86_64 1:1.12.2-2.el7

Dependency Installed:
  fontconfig.x86_64 0:2.10.95-11.el7      font
  gperftools-libs.x86_64 0:2.4-8.el7      libX
  libXau.x86_64 0:1.0.8-2.1.el7          libX
  libunwind.x86_64 2:1.2-2.el7           libX
  lyx-fonts.noarch 0:2.2.3-1.el7         ngin
  nginx-mod-http-geoip.x86_64 1:1.12.2-2.el7  ngin
  nginx-mod-http-xslt-filter.x86_64 1:1.12.2-2.el7  ngin

Complete!
[root@VM_0_8_centos ~]# nginx -v
nginx version: nginx/1.12.2
[root@VM_0_8_centos ~]#
```

这里显示安装已成功，版本号为 1.12.2。

配置随系统自启动

配置 Nginx 随系统自启动，即 Linux 系统启动时，Nginx 能自启动，而不是人为干预启动。

```
chkconfig --levels 235 nginx on
service nginx start
```

```
[root@VM_0_8_centos local]# chkconfig --levels 235 nginx on
Note: Forwarding request to 'systemctl enable nginx.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service
```

```
[root@VM_0_8_centos local]# service nginx start
Redirecting to /bin/systemctl start nginx.service
```

配置 Nginx

进入 /etc/nginx/，编辑 nginx.conf 文件。

```
[root@VM_0_8_centos ~]#  
[root@VM_0_8_centos ~]# cd /usr/local/  
[root@VM_0_8_centos local]# mkdir nginx  
[root@VM_0_8_centos local]# ls  
bin  etc  games  include  lib  lib64  libexec  nginx  qcloud  sbin  share  
[root@VM_0_8_centos local]#
```

在 nginx.conf 中，增加后端服务器 IP 和端口，由于 Nginx 和服务在同一台机器上，这里填入：

```
upstream frontends {  
    server 127.0.0.1:8000;  
}
```

当 Nginx 和后端服务器不在同一台机器上，并有多台后端服务器设备时，则配置具体服务器的 IP:端口 即可，Nginx 会负载均衡的将流量均匀分配到这些服务器上。此时配置如下：

```
upstream frontends {  
    server x.x.x.x:nnn;  
    server y.y.y.y:mmm;  
    server z.z.z.z:lll;  
}
```

由于 Nginx 对大小超过 1MB 的文件上传有限制，这里将默认限制 1MB 修改为 50MB，即添加配置 client_max_body_size 50m;，具体位置如下图所示。

```

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

    upstream frontends {
        server 127.0.0.1:8000;
        #server 10.10.10.10:8001;
    }

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

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    client_max_body_size 50m;

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

```

在 nginx.conf 文件中的 location 下，添加如下代理配置，即所有收到的请求，都转发到 frontends（如上所述的 upstream frontends）处理，具体如下：

```

proxy_pass_header Server;
proxy_set_header Host $http_host;
proxy_redirect false;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Scheme $scheme;
proxy_pass http://frontends;

```

具体位置如下：

```
# Load configuration files for the default server block.
include /etc/nginx/default.d/*.conf;
location / {
    proxy_pass_header Server;
    proxy_set_header Host $http_host;
    proxy_redirect false;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Scheme $scheme;
    proxy_pass http://frontends;
}

error_page 404 /404.html;
location = /40x.html {
}

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

至此，我们已完成了 Nginx 的配置。

其他配置项介绍

```
# For more information on configuration, see:
#   * Official English Documentation:
http://nginx.org/en/docs/
#   * Official Russian Documentation:
http://nginx.org/ru/docs/

user nginx;
# worker_processes 一般设置与 cpu 个数相等，也可配置为
auto
worker_processes auto;
# 全局错误日志及 pid 目录
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See
/usr/share/nginx/README.dynamic.
```

```
include /usr/share/nginx/modules/*.conf;

events {
    # 单个后台 worker processes 进程的最大并发连接数
    worker_connections 1024;
}

http {
    # 设置 log 格式
    log_format main '$remote_addr -
$remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent
"$http_referer" '
                    '"$http_user_agent"
"$http_x_forwarded_for"';

    # 配置上游服务器，此处为 Tornado 服务器 IP+Port
    upstream frontends {
        server 127.0.0.1:8000;
        #server 10.10.10.10:8001;
    }

    # 访问日志
    access_log /var/log/nginx/access.log main;

    # sendfile 指令指定 nginx 是否调用 sendfile 函数
    (zero copy 方式) 来输出文件，普通应用，设置为 on 即可；
    # keepalive_timeout 配置超时时间；
    # types_hash_max_size 影响散列表的冲突率。
    types_hash_max_size 越大，就会消耗更多的内存，但散列key
    的冲突率会降低，检索速度就更快。types_hash_max_size 越
    小，消耗的内存就越小，但散列key的冲突率可能上升。
    # client_max_body_size 客户端上传的body的最大值。
```



```
sendfile                on;
tcp_nopush              on;
tcp_nodelay             on;
keepalive_timeout       65;
types_hash_max_size     2048;
client_max_body_size    50m;

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

# Load modular configuration files from the
/etc/nginx/conf.d directory.
# See
http://nginx.org/en/docs/nginx\_core\_module.html#inc
lude
# for more information.
include /etc/nginx/conf.d/*.conf;

server {
    # 监听端口为 80
    listen      80 default_server;
    listen      [::]:80 default_server;
    server_name _;
    # 默认网站根目录位置
    root        /usr/share/nginx/html;

    # Load configuration files for the
default server block.
    include /etc/nginx/default.d/*.conf;

    # 设置默认请求代理，此处使用frontends，即请求代
理至 Tronado 服务器
    location / {
```



```
        proxy_pass_header Server;
        proxy_set_header Host $http_host;
        proxy_redirect false;
        proxy_set_header X-Real-IP
$remote_addr;
        proxy_set_header X-Scheme $scheme;
        proxy_pass http://frontends;
    }

    # 定义 404 错误页
    error_page 404 /404.html;
        location = /40x.html {
    }

    # 定义 50x 错误页
    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
}
```

重启 Nginx 服务

```
service nginx stop
service nginx start
```

```
[root@VM_0_8_centos nginx]# service nginx stop
Redirecting to /bin/systemctl stop nginx.service
[root@VM_0_8_centos nginx]# service nginx start
Redirecting to /bin/systemctl start nginx.service
[root@VM_0_8_centos nginx]#
```

接下来测试从 App 客户端向 Nginx 服务器发送 HTTP 请求，查看是否能正常转发至后端服务器上。在这里，我们看到 nginx.conf 有如下配置，即表示其对外服务的端口号为 80。实际项目中，可以根据具体情况进行修改。

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    server_name _;  
    root /usr/share/nginx/html;  
  
    # Load configuration files for the default server block  
    include /etc/nginx/default.d/*.conf;  
  
    location / {  
    }
```

请求测试

我们还是以上一节 App 客户端请求加载 H5 页面为例。由于 Nginx 对外提供服务的端口号是 80，而 80 端口在 HTTP 请求中可以不用输入，此时客户端请求的 URL 为：

http://150.109.33.132/users/login?

phone=18866668888&password=demo123456

服务端输出

```
[root@VM_0_8_centos demo]# ./main.py  
Tornado server is ready for service  
  
[D 180417 00:05:36 users_views:112] LoginHandle: get user login: 18866668888  
[I 180417 00:05:36 web:2106] 200 GET /users/login?phone=18866668888&password=demo123456 (183.53.65.2) 104.27ms  
[I 180417 00:05:36 web:2106] 200 GET /static/css/index.css (183.53.65.2) 104.27ms  
[I 180417 00:05:36 web:2106] 200 GET /static/image/index/index.jpg (183.53.66.106) 61.31ms  
[I 180417 00:05:36 web:2106] 200 GET /static/image/index/3.jpg (183.53.65.2) 21.86ms  
[I 180417 00:05:36 web:2106] 200 GET /static/image/index/2.jpg (183.53.65.163) 52.32ms  
[I 180417 00:05:36 web:2106] 200 GET /static/image/index/1.jpg (183.53.66.3) 31.58ms  
[I 180417 00:05:37 web:2106] 200 GET /static/image/index/4.jpg (183.53.66.106) 10.67ms  
[I 180417 00:05:37 web:2106] 200 GET /static/image/index/5.jpg (183.53.65.2) 4.82ms  
[I 180417 00:05:37 web:2106] 200 GET /static/image/index/6.jpg (183.53.65.2) 23.51ms
```

客户端加载结果



150.109.33.132



• DEMO

About



DEMO1



DEMO2



DEMO3

"Life is like riding a bicycle. To keep your balance, you must keep moving."



Cool Demo

Make it cool



Great Demo

Make it great



Wonderful Demo

Make it wonderful



至此，我们已完成了 Nginx 的学习及服务器端的配置。

Nginx 配置下载

链接：[百度网盘 - nginx.conf \(https://pan.baidu.com/s/1-3p1N08YY5rfALqocX8YZw\)](https://pan.baidu.com/s/1-3p1N08YY5rfALqocX8YZw)

密码：xtmr

小结

本节我们完成了 Nginx 服务器的搭建，并通过一个简单的例子，讲解了 Nginx 作为反向代理服务器和负载均衡器的应用。