链滴

# docker 中的 nginx 进行添加 fair 模块

作者：MingGH
原文链接：https://ld246.com/article/1635676204674
来源网站：链滴

# 前景提要

之前的博客都是单机的，现在想在多台服务器上进行部署，一台是国外的，一台是国内的，那么能否内的请求访问时走国内的服务器，而国外的请求走国外的服务器呢，这样既可以降低延迟，还可以降压力。

答案是可以的。因为服务器是通过nginx进行负载均衡的，所以打算通过nginx 的 fair 模块（第三方来实现负载均衡，fair 采用的不是内建负载均衡使用的轮换的均衡算法，而是**可以根据页面大小、响时间智能的进行负载均衡** 。

由于nginx我也是通过docker进行部署的，所以对于之前的已经部署的nginx需要进行移除，重新buil自定义的nginx带fair模块的

# 操作步骤

在停止之前的nginx之前，我们可以先进行build自定义的nginx镜像，到时候只需要停止之前那个，动新的就完成替换，非常的简单。

## 构建自定义nginx镜像

### 在用户目录下，创建一个nginx专用的文件夹

mkdir nginx_ws

### 提前下载fair模块包，和nginx安装包

nginxupstreamfairmaster.zip

wget http://nginx.org/download/nginx-1.18.0.tar.gz

都放在刚刚创建的文件夹下

### 添加Dockerfile

touch Dockerfile

### 对Dockerfile添加内容

```
FROM centos:7.8.2003

MAINTAINER runnable.run

# 添加本地文件
ADD nginx-1.18.0.tar.gz /usr/local/src
ADD nginx-upstream-fair-master.zip /usr/local/src
# 进入指定目录
WORKDIR /usr/local/src
# 安装unzip工具，并解压fair模块
RUN yum install -y unzip && unzip nginx-upstream-fair-master.zip
```

```
# 进入指定目录
WORKDIR /usr/local/src/nginx-1.18.0

RUN yum install -y gcc gcc-c++  glibc make autoconf openssl openssl-devel \
&& yum install -y libxslt-devel -y gd gd-devel GeoIP GeoIP-devel pcre pcre-devel \
&& useradd -M -s /sbin/nologin nginx && BUILD_CONFIG="--prefix=/usr/local/nginx \
        --http-client-body-temp-path=/var/cache/nginx/client_temp \
        --http-proxy-temp-path=/var/cache/nginx/proxy_temp \
        --http-fastcgi-temp-path=/var/cache/nginx/fastcgi_temp \
        --http-uwsgi-temp-path=/var/cache/nginx/uwsgi_temp \
        --http-scgi-temp-path=/var/cache/nginx/scgi_temp \
        --with-http_stub_status_module \
        --with-http_ssl_module \
        --with-stream \
        --with-http_v2_module \
        --with-threads \
        --add-module=/usr/local/src/nginx-upstream-fair-master" && ./configure $BUILD_CONF
G \
&& mkdir -p /var/cache/nginx && make && make install


ENV PATH /usr/local/nginx/sbin:$PATH

EXPOSE 80
EXPOSE 443

ENTRYPOINT ["nginx"]

CMD ["-g","daemon off;"]
```

这里对其中的命令进行解释

FROM：指定基础镜像，必须为第一个命令

MAINTAINER: 维护者信息

ADD：将本地文件添加到容器中，tar类型文件会自动解压(网络压缩资源不会被解压)，可以访问网络源，类似wget

WORKDIR：工作目录，类似于cd命令

RUN：构建镜像时执行的命令

ENV：设置环境变量

EXPOSE：指定于外界交互的端口

ENTRYPOINT：配置容器，使其可执行化。配合CMD可省去"application"，只使用参数。

CMD：构建容器后调用，也就是在容器启动时才进行调用。

## 进行构建

docker build -t centos7.8_nginx1.18:v1 .

### 执行成功信息

当出现以下信息时，表示成功

Successfully built f91ed12a53e7

## 先启动一个最简单的nginx进行复制配置文件

docker run --name nginx -p 8081:80 -d --rm docker run --name nginx -p 80:80 -d --rm nginx

复制容器中的配置文件到指定目录

docker cp nginx:/usr/local/nginx/conf/nginx.conf /root/nginx_ws/dockerData/nginx/conf/nginx.conf

删除刚刚创建的容器,因为在创建该容器的时候已经添加了 --rm ,所以只要停止就可以自动删除

docker stop nginx

### 修改配置文件

供参考配置文件

```
#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       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  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;
```

```
#keepalive_timeout  0;
keepalive_timeout  65;

#gzip  on;
upstream blogServer {
  fair;
  #server ip1:8080;
  server ip2:8080;
}

server {
    listen       443;
    server_name  runnable.run;
    ssl on;

    proxy_set_header  Host $http_host;
    proxy_set_header  X-Real-IP  $remote_addr;

    ssl_certificate /ssl/6349085_www.runnable.run.pem;
    ssl_certificate_key /ssl/6349085_www.runnable.run.key;
    ssl_session_timeout 5m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;


    location / {
        proxy_pass http://blogServer;
        error_page  404 https://www.runnable.run;
    }

    error_page  404 https://www.runnable.run;
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}

server{
  listen 80;
  server_name runnable.run;
  proxy_set_header  Host $http_host;
  proxy_set_header  X-Real-IP  $remote_addr;
  rewrite ^(.*) https://$host$1 permanent;
}



server {
    listen       443;
    server_name  www.runnable.run;
    ssl on;

    proxy_set_header  Host $http_host;
```

```
        proxy_set_header  X-Real-IP  $remote_addr;

        ssl_certificate /ssl/6349085_www.runnable.run.pem;
        ssl_certificate_key /ssl/6349085_www.runnable.run.key;
        ssl_session_timeout 5m;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
        ssl_prefer_server_ciphers on;


        location / {
            proxy_pass http://blogServer;
            error_page  404 https://www.runnable.run;
        }

        error_page  404 https://www.runnable.run;
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   /usr/share/nginx/html;
        }
    }

    server{
      listen 80;
      server_name www.runnable.run;
      proxy_set_header  Host $http_host;
      proxy_set_header  X-Real-IP  $remote_addr;
      rewrite ^(.*) https://$host$1 permanent;
    }

}
```

## 通过自定义镜像进行启动

```
docker run -d -p 80:80 -p 443:443 --name nginx \
-v /root/nginx_ws/dockerData/nginx/conf/nginx.conf:/usr/local/nginx/conf/nginx.conf \
-v /root/nginx_ws/dockerData/nginx/ssl:/ssl/ \
-v /root/nginx_ws/dockerData/nginx/www:/usr/share/nginx/html \
centos7.8_nginx1.18:v1
```

# 参考内容

本文参考了以下文章

从零开始安装 solo 博客

Docker中编译安装nginx

Nginx 负载均衡 - fair