环境配置

nginx

若以前安装过，先卸载

```
sudo apt-get --purge remove nginx
```

```
sudo apt-get --purge remove nginx-common
```

```
dpkg --get-selections|grep nginx
```

1.前置环境

```
sudo apt-get update
sudo apt-get install gcc
sudo apt-get install ruby
sudo apt-get install zlib1g
sudo apt-get install zlib1g.dev
sudo apt-get install libpcre3 libpcre3-dev
sudo apt-get install openssl libssl-dev
```

2.下载nginx源码

下载地址在https://nginx.org
源码在压缩包中附带1.22.1版本，nginx-1.22.1.tar.gz
解压到用户主目录 ~

3.下载nginx-http-flv-module

在压缩包中附带1.2.10版本nginx-http-flv-module-1.2.10.zip
把nginx-http-flv-module解压，移动到/usr/local目录下，并将文件夹改名为：nginx-http-flv-module

```
cp nginx-http-flv-module-1.2.10 /usr/local/nginx-http-flv-module
```

进入nginx目录

```
cd ~/nginx-1.22.1
```

将nginx-http-flv-module添加到nginx中

```
./configure --prefix=/usr/local  --with-http_ssl_module --add-module=/usr/local/nginx/nginx-http-flv-module
```

生成make文件

```
make
```

安装

```
sudo make install
```

测试是否安装成功

```
nginx -v
```

4.配置nginx

需要修改nginx的配置文件，它的位置在nginx-1.22.1/conf/nginx.conf
nginx.conf文件内容如下：

```
worker_processes  10; #should be 1 for Windows, for it doesn't support Unix domain socket
#worker_processes  auto; #from versions 1.3.8 and 1.2.5

#worker_cpu_affinity  0001 0010 0100 1000; #only available on FreeBSD and Linux
#worker_cpu_affinity  auto; #from version 1.9.10

error_log logs/error.log error;

#if the module is compiled as a dynamic module and features relevant
#to RTMP are needed, the command below MUST be specified and MUST be
#located before events directive, otherwise the module won't be loaded
#or will be loaded unsuccessfully when NGINX is started

#load_module modules/ngx_http_flv_live_module.so;

events {
    worker_connections  10240;
}

http {
        include       mime.types;
        default_type  application/octet-stream;
        sendfile        on;
        keepalive_timeout  65;
        server {
                listen       9000; #http-flv的拉流端口
                server_name  localhost【这个地方写你的ip、或者说是域名都是可以的--】; #http-flv的拉流ip
                location / {
                root   html;
                index  index.html index.htm;
                }
            #http-flv的相关配置
                location /live{
                flv_live on; #打开HTTP播放FLV直播流功能
                chunked_transfer_encoding  on;
                add_header 'Access-Control-Allow-Origin' '*';
                add_header 'Access-Control-Allow-Credentials' 'true';
                }
                location /hls{
                types {
                        application/vnd.apple.mpegurl m3u8;
                        video/mp2t ts;
                 }
                 root ~/nginx-1.22.1/html/hls;
                 add_header 'Cache-Control' 'no-cache';
                }
                location /dash {
                root ~/nginx-1.22.1/html/dash;
                add_header 'Cache-Control' 'no-cache';
                }
                location /stat {
                #configuration of push & pull status
                rtmp_stat all;
                rtmp_stat_stylesheet stat.xsl;
                }
                location /stat.xsl {
                        root ~/nginx-1.22.1/html;
                }

                location /control {
                rtmp_control all; #configuration of control module of rtmp
                        }

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

}

rtmp_auto_push on;
rtmp_auto_push_reconnect 1s;
rtmp_socket_dir /tmp;


rtmp{
        out_queue 4096;
        out_cork 8;
        max_streams 128;
        timeout 15s;
        drop_idle_publisher 15s;
        log_interval 5s;
        log_size 1m;
        server{
            #推流端口
                listen 1935;
            #推流应用名称
                application myapp{
                 live on;
                 record off;
                 gop_cache on;
                }
                application hls{
                        live on;
                        hls on;
                        hls_path ~/nginx-1.22.1/html/hls;
                        hls_fragment 1s;
                        hls_playlist_length 3s;
                }
                application dash{
                        live on;
                        dash on;
                        dash_path ~/nginx-1.22.1/html/dash;
                }

        }
}
```

文件中有五处路径为~/nginx-1.22.1，在实际使用时要换成具体的目录，例如用户为admin，那么这里就要写对应的用户主目录/home/admin/nginx-1.22.1。
该文件要放到/usr/local/nginx/conf文件夹下

5.启动nginx

```
sudo cd /usr/local/nginx
sudo sbin/nginx -c conf/nginx.conf
sudo sbin/nginx -s reload
```

6.安装ffmpeg

```
sudo apt-get install ffmpeg
```

7.测试推流

```
ffmpeg -re -i rtsp://wowzaec2demo.streamlock.net/vod/mp4:BigBuckBunny_115k.mp4 -vcodec copy -acodec copy -f flv '
```

Docker

Docker安装

```
sudo apt-get install curl
curl -sSL https://get.daocloud.io/docker | sh
```

拉取镜像源

```
docker pull iludfy/predict_model
```

Docker创建容器并启动

拉取完镜像需要挂载到容器中才能启动，命令是docker run，它的参数如下：

```
docker run [Options] image

#参数说明
--name="名字"           指定容器名字
-d                      后台方式运行
-it                     使用交互方式运行,进入容器查看内容
-p                      指定容器的端口
        -p ip:主机端口:容器端口   配置主机端口映射到容器端口
        -p 主机端口:容器端口（常用）
        -p 容器端口
-P                      随机指定端口
-e                                          环境设置
-v                                          容器数据卷挂载
```

规定端口为5000，输入下面的命令：

```
docker run -p 5000:5000 iludfy/predict_model
```

一个容器就已经被创建好并启动了。

端口被占用解决办法：

以ubuntu系统为例：
先查到5000端口被哪个进程占用：

```
sudo netstat -ntlp
```

查到进程id后，使用kill命令杀死进程：

```
sudo kill {pid}
```

查看刚刚创建的容器的id：

```
docker ps -a
```

启动

```
docker start {id}
```