## nginx

### 自建nginx镜像

#### 构建镜像

Dockerfile

|  |
| --- |
| FROM debian:stretch-slim  #设置时区  RUN rm -rf /etc/localtime && ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime  #设置编码  ENV LANG=C.UTF-8  #设置aliyun apt 源  RUN set -ex \  && { \  echo 'deb http://mirrors.aliyun.com/debian stretch main contrib non-free'; \  echo 'deb http://mirrors.aliyun.com/debian stretch-proposed-updates main contrib non-free'; \  echo 'deb http://mirrors.aliyun.com/debian stretch-updates main contrib non-free'; \  echo 'deb-src http://mirrors.aliyun.com/debian stretch main contrib non-free'; \  echo 'deb-src http://mirrors.aliyun.com/debian stretch-proposed-updates main contrib non-free'; \  echo 'deb-src http://mirrors.aliyun.com/debian stretch-updates main contrib non-free'; \  echo 'deb http://mirrors.aliyun.com/debian-security/ stretch/updates main non-free contrib'; \  echo 'deb-src http://mirrors.aliyun.com/debian-security/ stretch/updates main non-free contrib'; \  } > /etc/apt/sources.list  ENV NGINX\_VERSION=1.14.0-1~stretch  ENV NJS\_VERSION=1.14.0.0.2.0-1~stretch  #安装nginx  RUN set -x \  && apt-get update \  && apt-get install --no-install-recommends --no-install-suggests -y gnupg1 apt-transport-https ca-certificates \  && \  NGINX\_GPGKEY=573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62; \  found=''; \  for server in \  ha.pool.sks-keyservers.net \  hkp://keyserver.ubuntu.com:80 \  hkp://p80.pool.sks-keyservers.net:80 \  pgp.mit.edu \  ; do \  echo "Fetching GPG key $NGINX\_GPGKEY from $server"; \  apt-key adv --keyserver "$server" --keyserver-options timeout=10 --recv-keys "$NGINX\_GPGKEY" && found=yes && break; \  done; \  test -z "$found" && echo >&2 "error: failed to fetch GPG key $NGINX\_GPGKEY" && exit 1; \  apt-get remove --purge --auto-remove -y gnupg1 && rm -rf /var/lib/apt/lists/\* \  && nginxPackages=" \  nginx=${NGINX\_VERSION} \  nginx-module-xslt=${NGINX\_VERSION} \  nginx-module-geoip=${NGINX\_VERSION} \  nginx-module-image-filter=${NGINX\_VERSION} \  nginx-module-njs=${NJS\_VERSION} \  " \  && dpkgArch="$(dpkg --print-architecture)" \  && case "$dpkgArch" in \  amd64|i386) \  echo "deb https://nginx.org/packages/debian/ stretch nginx" >> /etc/apt/sources.list.d/nginx.list \  && apt-get update \  ;; \  \*) \  echo "deb-src https://nginx.org/packages/debian/ stretch nginx" >> /etc/apt/sources.list.d/nginx.list \  \  && tempDir="$(mktemp -d)" \  && chmod 777 "$tempDir" \  \  && savedAptMark="$(apt-mark showmanual)" \  \  && apt-get update \  && apt-get build-dep -y $nginxPackages \  && ( \  cd "$tempDir" \  && DEB\_BUILD\_OPTIONS="nocheck parallel=$(nproc)" \  apt-get source --compile $nginxPackages \  ) \  \  && apt-mark showmanual | xargs apt-mark auto > /dev/null \  && { [ -z "$savedAptMark" ] || apt-mark manual $savedAptMark; } \  \  && ls -lAFh "$tempDir" \  && ( cd "$tempDir" && dpkg-scanpackages . > Packages ) \  && grep '^Package: ' "$tempDir/Packages" \  && echo "deb [ trusted=yes ] file://$tempDir ./" > /etc/apt/sources.list.d/temp.list \  && apt-get -o Acquire::GzipIndexes=false update \  ;; \  esac \  \  && apt-get install -y $nginxPackages gettext-base --no-install-recommends --no-install-suggests \  && apt-get remove --purge --auto-remove -y apt-transport-https ca-certificates \  && rm -rf /var/lib/apt/lists/\* /etc/apt/sources.list.d/nginx.list \  \  && if [ -n "$tempDir" ]; then \  apt-get purge -y --auto-remove \  && rm -rf "$tempDir" /etc/apt/sources.list.d/temp.list; \  fi  RUN set -ex \  && ln -sf /dev/stdout /var/log/nginx/access.log \  && ln -sf /dev/stderr /var/log/nginx/error.log  EXPOSE 80  STOPSIGNAL SIGTERM  CMD ["nginx", "-g", "daemon off;"] |

构建镜像

|  |
| --- |
| docker build -t litongjava/nginx .  docker push litongjava/nginx |

#### 构建过程命令讲解

1.

|  |
| --- |
| ln -sf /dev/stdout /var/log/nginx/access.log  ln -sf /dev/stderr /var/log/nginx/error.log |

确定标准输入和标准输入文件,在控制台中显示日志

2.STOPSIGNAL SIGTERM

|  |  |
| --- | --- |
| STOPSIGNAL SIGTERM | 指定停止信号是SIGTERM |

默认的stop-signal是SIGTERM，在docker stop的时给容器内PID为1的进程发送signal信号，通过--stop-signal可以设置自己需要的signal，主要的目的是为了让容器内的应用程序在接收到signal之后可以先做一些事情，实现容器的平滑退出，如果不做任何处理，容器将在一段时间之后强制退出，会造成业务的强制中断，这个时间默认是10s

3.安装nginx模块

在本次构建中安装了下面的nginx模块

|  |
| --- |
| nginx=1.14.0-1~stretch  nginx-module-xslt=1.14.0-1~stretch  nginx-module-geoip=1.14.0-1~stretch  nginx-module-image-filter=1.14.0-1~stretch  nginx-module-njs=1.14.0.0.2.0-1~stretch |

问:上面这些模块有没有安装nginx-rtmp模块

#### 启动镜像

简单运行

|  |
| --- |
| docker run -dit --restart=always --name mynginx -p 1025:80 litongjava/nginx |

export\_dir

|  |  |
| --- | --- |
| /etc/nginx | 配置文件目录 |
| /var/log/nginx | nginx日志 |
| /usr/share/nginx/ | 默认网站目录.默认访问的文件是/usr/share/nginx/html/index.html |

镜像启动后,docker-enter竟然不能进入

|  |
| --- |
| [root@docker214 nginx]# docker-enter mynginx  nsenter: failed to execute su: No such file or directory |

docker attach 2150e028e023后进入的是黑屏