← Docker 容器连接 Docker 安装 MySQL →

Docker 安装 Nginx

方法一、docker pull nginx(推荐)

查找 Docker Hub 上的 nginx 镜像

```
runoob@runoob:~/nginx$ docker search nginx
NAME                     DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
nginx                    Official build of Nginx.                        3260      [OK]
jwilder/nginx-proxy      Automated Nginx reverse proxy for docker c...   674                  [OK]
richarvey/nginx-php-fpm  Container running Nginx + PHP-FPM capable ...   207                  [OK]
million12/nginx-php      Nginx + PHP-FPM 5.5, 5.6, 7.0 (NG), CentOS...   67                   [OK]
maxexcloo/nginx-php      Docker framework container with Nginx and ...   57                   [OK]
webdevops/php-nginx      Nginx with PHP-FPM                              39                   [OK]
h3nrik/nginx-ldap        NGINX web server with LDAP/AD, SSL and pro...   27                   [OK]
bitnami/nginx            Bitnami nginx Docker Image                      19                   [OK]
maxexcloo/nginx          Docker framework container with Nginx inst...   7                    [OK]
...
```

这里我们拉取官方的镜像

```
runoob@runoob:~/nginx$ docker pull nginx
```

等待下载完成后，我们就可以在本地镜像列表里查到 REPOSITORY 为 nginx 的镜像。

```
runoob@runoob:~/nginx$ docker images nginx
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx               latest              555bbd91e13c        3 days ago          182.8 MB
```

方法二、通过 Dockerfile 构建(不推荐)

创建 Dockerfile

首先，创建目录 nginx, 用于存放后面的相关东西。

```
runoob@runoob:~$ mkdir -p ~/nginx/www ~/nginx/logs ~/nginx/conf
```

www: 目录将映射为 nginx 容器配置的虚拟目录。

logs: 目录将映射为 nginx 容器的日志目录。

conf: 目录里的配置文件将映射为 nginx 容器的配置文件。

进入创建的 nginx 目录，创建 Dockerfile 文件，内容如下：

```
FROM debian:stretch-slim

LABEL maintainer="NGINX Docker Maintainers <docker-maint@nginx.com>"

ENV NGINX_VERSION 1.14.0-1~stretch
ENV NJS_VERSION   1.14.0.0.2.0-1~stretch

RUN set -x \
    && apt-get update \
    && apt-get install --no-install-recommends --no-install-suggests -y gnupg1 apt-transport-https ca-ce
rtificates \
    && \
    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 $NGINX_GPGKEY" && exit 1; \
    apt-get remove --purge --auto-remove -y gnupg1 && rm -rf /var/lib/apt/lists/* \
    && dpkgArch="$(dpkg --print-architecture)" \
    && nginxPackages=" \
        nginx=${NGINX_VERSION} \
        nginx-module-xslt=${NGINX_VERSION} \
        nginx-module-geoip=${NGINX_VERSION} \
        nginx-module-image-filter=${NGINX_VERSION} \
        nginx-module-njs=${NJS_VERSION} \
    " \
    && case "$dpkgArch" in \
        amd64|i386) \
# arches officialy built by upstream
            echo "deb https://nginx.org/packages/debian/ stretch nginx" >> /etc/apt/sources.list.d/ngin
x.list \
            && apt-get update \
            ;; \
        *) \
# we're on an architecture upstream doesn't officially build for
# let's build binaries from the published source packages
            echo "deb-src https://nginx.org/packages/debian/ stretch nginx" >> /etc/apt/sources.list.d/n
ginx.list \
            \
# new directory for storing sources and .deb files
```

```
            && tempDir="$(mktemp -d)" \
            && chmod 777 "$tempDir" \
# (777 to ensure APT's "_apt" user can access it too)
            \
# save list of currently-installed packages so build dependencies can be cleanly removed later
            && savedAptMark="$(apt-mark showmanual)" \
            \
# build .deb files from upstream's source packages (which are verified by apt-get)
            && apt-get update \
            && apt-get build-dep -y $nginxPackages \
            && ( \
                cd "$tempDir" \
                && DEB_BUILD_OPTIONS="nocheck parallel=$(nproc)" \
                    apt-get source --compile $nginxPackages \
            ) \
# we don't remove APT lists here because they get re-downloaded and removed later
            \
# reset apt-mark's "manual" list so that "purge --auto-remove" will remove all build dependencies
# (which is done after we install the built packages so we don't have to redownload any overlapping depe
ndencies)
            && apt-mark showmanual | xargs apt-mark auto > /dev/null \
            && { [ -z "$savedAptMark" ] || apt-mark manual $savedAptMark; } \
            \
# create a temporary local APT repo to install from (so that dependency resolution can be handled by AP
T, as it should be)
            && ls -lAFh "$tempDir" \
            && ( cd "$tempDir" && dpkg-scanpackages . > Packages ) \
            && grep '^Package: ' "$tempDir/Packages" \
            && echo "deb [ trusted=yes ] file://$tempDir ./" > /etc/apt/sources.list.d/temp.list \
# work around the following APT issue by using "Acquire::GzipIndexes=false" (overriding "/etc/apt/apt.co
nf.d/docker-gzip-indexes")
#   Could not open file /var/lib/apt/lists/partial/_tmp_tmp.ODWljpQfkE_._Packages - open (13: Permission
 denied)
#   ...
#   E: Failed to fetch store:/var/lib/apt/lists/partial/_tmp_tmp.ODWljpQfkE_._Packages  Could not open f
ile /var/lib/apt/lists/partial/_tmp_tmp.ODWljpQfkE_._Packages - open (13: Permission denied)
            && apt-get -o Acquire::GzipIndexes=false update \
            ;; \
    esac \
    \
    && apt-get install --no-install-recommends --no-install-suggests -y \
                        $nginxPackages \
                        gettext-base \
    && apt-get remove --purge --auto-remove -y apt-transport-https ca-certificates && rm -rf /var/lib/ap
t/lists/* /etc/apt/sources.list.d/nginx.list \
    \
# if we have leftovers from building, let's purge them (including extra, unnecessary build deps)
    && if [ -n "$tempDir" ]; then \
        apt-get purge -y --auto-remove \
```

```
        && rm -rf "$tempDir" /etc/apt/sources.list.d/temp.list; \
    fi

# forward request and error logs to docker log collector
RUN 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;"]
```

通过 Dockerfile 创建一个镜像，替换成你自己的名字。

```
docker build -t nginx .
```

创建完成后，我们可以在本地的镜像列表里查找到刚刚创建的镜像

```
runoob@runoob:~/nginx$ docker images nginx
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx               latest              555bbd91e13c        3 days ago          182.8 MB
```

使用 nginx 镜像

运行容器

```
runoob@runoob:~/nginx$ docker run -p 80:80 --name mynginx -v $PWD/www:/www -v $PWD/conf/nginx.conf:/etc/
nginx/nginx.conf -v $PWD/logs:/wwwlogs  -d nginx
45c89fab0bf9ad643bc7ab571f3ccd65379b844498f54a7c8a4e7ca1dc3a2c1e
runoob@runoob:~/nginx$
```

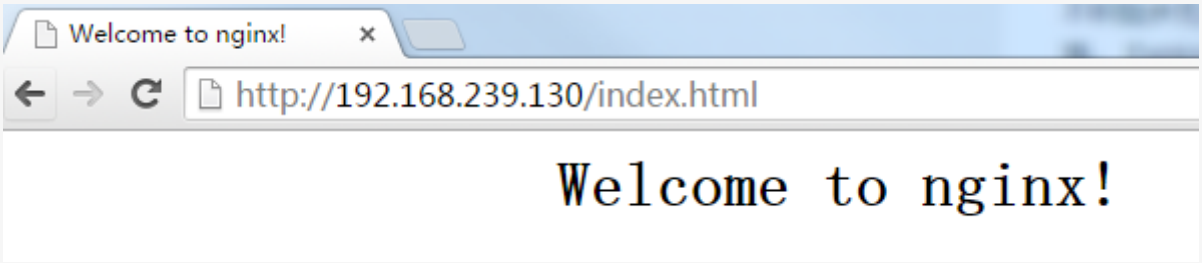
命令说明：

- **-p 80:80：**将容器的80端口映射到主机的80端口
- **--name mynginx：**将容器命名为mynginx
- **-v $PWD/www:/www：**将主机中当前目录下的www挂载到容器的/www
- **-v $PWD/conf/nginx.conf:/etc/nginx/nginx.conf：**将主机中当前目录下的nginx.conf挂载到容器的/etc/nginx/nginx.conf
- **-v $PWD/logs:/wwwlogs：**将主机中当前目录下的logs挂载到容器的/wwwlogs

查看容器启动情况

```
runoob@runoob:~/nginx$ docker ps
CONTAINER ID        IMAGE               COMMAND                  ...  PORTS                           NAMES
45c89fab0bf9        nginx               "nginx -g 'daemon off"   ...  0.0.0.0:80->80/tcp, 443/tcp     mynginx
f2fa96138d71        tomcat              "catalina.sh run"        ...  0.0.0.0:81->8080/tcp            tomcat
```

通过浏览器访问



← Docker 容器连接

Docker 安装 MySQL →

点我分享笔记