本文介绍两种安装方式，一种是基于Linux系统原生安装，一种是本地基于docker安装。

一、原生方式安装

官网下载地址为：

```
http://nginx.org/en/download.html
```

上传到linux系统上。下面开始安装。首先安装依赖环境：

1，安装gcc环境

```
yum install gcc-c++
```

2，安装PCRE库，用于解析正则表达式

```
yum install -y pcre pcre-devel
```

3，zlib压缩和解压缩依赖

```
yum install -y zlib zlib-devel
```

4，SSL安全的加密的套接字协议层，用于HTTP安全传输，也就是https

```
yum install -y openssl openssl-devel
```

5，解压，需要注意，解压后得到的是源码，源码需要编译后才能安装

```
tar -zxvf nginx-1.18.0.tar.gz
```

6，编译之前，先创建nginx临时目录，如果不创建，在启动nginx的过程中会报错

```
mkdir /var/temp/nginx -p
```

7，在nginx目录，输入如下命令进行配置，目的是为了创建makefile文件，只有生成了这个文件才能进行下面的编译和安装工作。

```
cd nginx-1.18.0
```

```
./configure \
--prefix=/usr/local/nginx \
--pid-path=/var/run/nginx/nginx.pid \
--lock-path=/var/lock/nginx.lock \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
--with-http_gzip_static_module \
--http-client-body-temp-path=/var/temp/nginx/client \
--http-proxy-temp-path=/var/temp/nginx/proxy \
--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \
--http-uwsgi-temp-path=/var/temp/nginx/uwsgi \
--http-scgi-temp-path=/var/temp/nginx/scgi
```

命令	解释
-prefix	指定nginx安装目录
-pid-path	指向nginx的pid -lock-path锁定安装文件，防止被恶意篡改或误操作
-error-log	错误日志
-http-log-path	http日志
-with-http_gzip_static_module	启用gzip模块，在线实时压缩输出数据流
-http-client-body-temp-path	设定客户端请求的临时目录
-http-proxy-temp-path	设定http代理临时目录
-http-fastcgi-temp-path	设定fastcgi临时目录
-http-uwsgi-temp-path	设定uwsgi临时目录
-http-scgi-temp-path	设定scgi临时目录

执行完毕后，会看到当前目录下多出一个 `Makefile` 文件。

8，make编译

```
make
```

9，安装

```
make install
```

10，进入sbin目录启动nginx

安装完毕之后，实际上会根据我们的第一个配置项安装到 `/usr/local/nginx` 这个目录下。我们可以用 `whereis` 命令查找下：

进入其目录下的 `sbin` 目录去启动。

```
[root@VM-0-13-centos nginx-1.18.0]# whereis nginx
nginx: /usr/local/nginx
```

```
./nginx
```

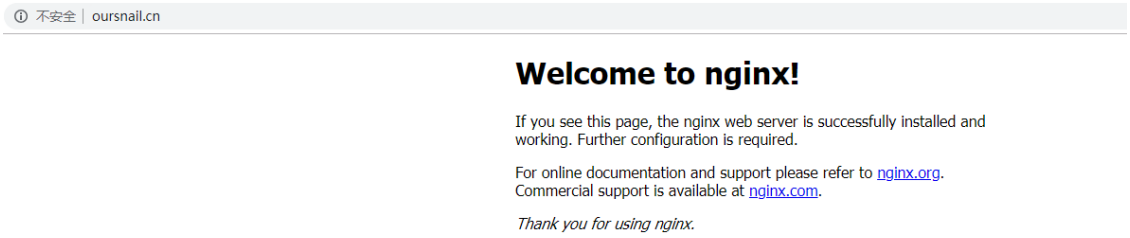
- 停止：./nginx -s stop
- 重新加载：./nginx -s reload
- 检查配置是否正常：./nginx -t

查看进程：

```
[root@VM-0-13-centos sbin]# ./nginx
[root@VM-0-13-centos sbin]# ps -ef | grep nginx
root     12124     1  0 16:07 ?        00:00:00 nginx: master
process ./nginx
nobody   12125 12124  0 16:07 ?        00:00:00 nginx: worker
process
root     12145  3126  0 16:07 pts/0    00:00:00 grep --color=auto
nginx
```

说明我们已经安装好了。

11，打开浏览器，访问虚拟机所处内网ip即可打开nginx默认页面，显示如下便表示安装成功



12，注意点：

- 如果在云服务器安装，需要在安全组中放开默认的nginx端口：80
- 如果在虚拟机安装，需要关闭防火墙
- 本地win或mac需要关闭防火墙

在我们安装好后的 `/usr/local/nginx/` 目录下有三个文件夹：

```
drwxr-xr-x 2 root root 4096 Dec  3 16:05 conf
drwxr-xr-x 2 root root 4096 Dec  3 16:05 html
drwxr-xr-x 2 root root 4096 Dec  3 16:05 sbin
```

`conf` 目录是相关的配置文件：

```
-rw-r--r-- 1 root root 1077 Dec  3 16:05 fastcgi.conf
-rw-r--r-- 1 root root 1077 Dec  3 16:05 fastcgi.conf.default
-rw-r--r-- 1 root root 1007 Dec  3 16:05 fastcgi_params
-rw-r--r-- 1 root root 1007 Dec  3 16:05 fastcgi_params.default
-rw-r--r-- 1 root root 2837 Dec  3 16:05 koi-utf
-rw-r--r-- 1 root root 2223 Dec  3 16:05 koi-win
-rw-r--r-- 1 root root 5231 Dec  3 16:05 mime.types
-rw-r--r-- 1 root root 5231 Dec  3 16:05 mime.types.default
-rw-r--r-- 1 root root 2656 Dec  3 16:05 nginx.conf
-rw-r--r-- 1 root root 2656 Dec  3 16:05 nginx.conf.default
-rw-r--r-- 1 root root  636 Dec  3 16:05 scgi_params
-rw-r--r-- 1 root root  636 Dec  3 16:05 scgi_params.default
-rw-r--r-- 1 root root  664 Dec  3 16:05 uwsgi_params
-rw-r--r-- 1 root root  664 Dec  3 16:05 uwsgi_params.default
-rw-r--r-- 1 root root 3610 Dec  3 16:05 win-utf
```

最关键也是最常用的是 `nginx.conf` 。下面要核心学习的。

`html` 目录就是默认的nginx首页。

```
-rw-r--r-- 1 root root 494 Dec  3 16:05 50x.html
-rw-r--r-- 1 root root 612 Dec  3 16:05 index.html
```

`sbin` 下面是可执行的一个文件，用来启动nginx等操作。

二、Docker方式安装

本地学习时，最方便的方式就是使用Docker来搭建了。下面我简单提炼一些docker学习的重点。

Docker 使用 Google 公司推出的 Go 语言进行开发实现，基于 Linux 内核的 cgroup，namespace，以及 OverlayFS类的 Union FS 等技术，对进程进行封装隔离，属于操作系统层面的虚拟化技术。

由于隔离的进程独立于宿主和其它的隔离的进程，因此也称其为容器。

Docker 经常拿来和虚拟机来比较，因为它们两个的用处和用法都很相似，然而实际的底层技术原理是完全不一样的。

假设你现在变身了，站在了 Docker 和 虚拟机的内部，从里面向外看，发现虚拟机有自己的 CPU(虚拟CPU)、内存、硬盘，再往外才是宿主机的 CPU、硬盘、内存等。而如果是在Docker内部向外看，发现你无论站在当前实体机的哪个容器里，看到的都是宿主机的 CPU、硬盘、内存等。说明 Dokcer 容器是直接拿宿主机的资源当自己的用，所以每个容器的硬件配置都是一样的，而虚拟机是完全虚拟出来一套。

为什么要使用Docker?

- 实现环境一致性：不用担心开发环境、测试环境、生产环境的差异了，简化了大量运维工作和问题排查成本；
- 基于云原生，资源利用更加出色，也更加便于管理；

Docker的概念有很多，但是最核心的概念就三个：

- image镜像：是一个只读模板，一个镜像可以包含一个完整的centos，镜像是用来创建Docker容器的，镜像相当于java中的Class类的概念，而一个一个new出来的对象相当于是docker中的容器概念。此外，镜像提供了简单的机制来创建镜像或更新现有镜像，用户甚至可以直接从其他人那里下载一个已经做好的镜像直接使用。
- container容器：容器就是镜像的运行实例，可以被启动、开始、停止、删除，每个容器之间互相隔离。
- repository仓库：仓库是存储镜像的地方，类似maven仓库，也分为公共仓库和私有仓库，最大的公共仓库是Docker Hub，存放了数量庞大的镜像供用户下载使用，公司或个人也可以搭建自己的私有仓库。

Docker的命令也很多，不过我们常用的就那几个：

- 【镜像相关】获取镜像：`docker pull 镜像名称`
- 【镜像相关】查看镜像列表：`docker image ls`

- 【容器相关】查看容器：docker ps -a
- 【容器相关】新建并启动容器：sudo docker run -t -i ubuntu:12.04 /bin/bash
- 【容器相关】进入容器：docker exec -it 容器id bash

记不住命令也没关系，我们可以下载可视化客户端降低我们的记忆成本，Docker Desktop是不二选择。

访问 Docker Desktop 官网 https://www.docker.com/products/docker-desktop/ ，根据你的系统下载对应的版本。

下载镜像可以通过命令行，也可以直接通过docker desktop直接下载，不过自从2023年5月中旬以来，hub.docker.com由于DNS污染原因，可以考虑走国内的镜像平台下载，比如阿里。

关于镜像加速问题：https://help.aliyun.com/zh/acr/user-guide/accelerate-the-pulls-of-docker-official-images

下面拉取镜像并启动容器步骤如下：

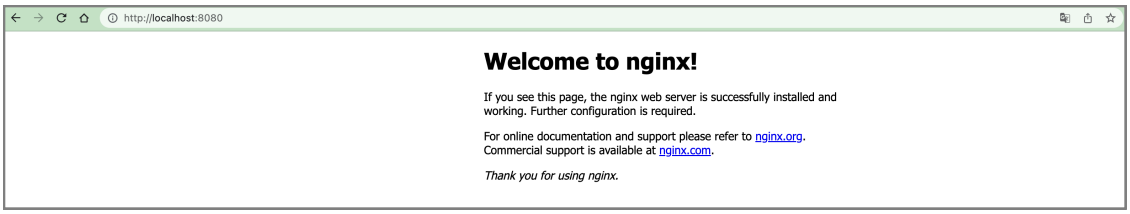
```
sunweiguo@sunweiguodeMBP ~ % docker search nginx
NAME                                             DESCRIPTION
                                 STARS     OFFICIAL   AUTOMATED
nginx                                            Official build of
Nginx.                           19103     [OK]
bitnami/nginx                                    Bitnami nginx
Docker Image                        176                  [OK]
nginxinc/nginx-unprivileged                      Unprivileged
NGINX Dockerfiles                    125
nginxproxy/acme-companion                        Automated ACME
SSL certificate generation fo…   125
nginxproxy/nginx-proxy                           Automated Nginx
reverse proxy for docker con…    110
ubuntu/nginx                                     Nginx, a high-
performance reverse proxy & we…  100
nginx/nginx-ingress                              NGINX and  NGINX
Plus Ingress Controllers fo…     81
nginx/unit                                       NGINX Unit is a
dynamic web and application …    64
nginx/nginx-prometheus-exporter                  NGINX Prometheus
Exporter for NGINX and NGIN…     33
bitnami/nginx-ingress-controller                 Bitnami Docker
Image for NGINX Ingress Contr…   30                  [OK]
unit                                             Official build of
NGINX Unit: Universal Web …      15        [OK]
```

```
nginxproxy/docker-gen                              Generate files
from docker container meta-da…   12
rancher/nginx-ingress-controller
                                 11
kasmweb/nginx                                      An Nginx image
based off nginx:alpine and in…   6
nginxinc/ingress-demo                              Ingress Demo
                                 4
nginxinc/nginx-s3-gateway                          Authenticating
and caching gateway based on …   2
rancher/nginx-ingress-controller-defaultbackend
                                 2
nginx/nginx-ingress-operator                       NGINX Ingress
Operator for NGINX and NGINX P…  1
nginxinc/amplify-agent                             NGINX Amplify
Agent docker repository          1
nginx/nginx-quic-qns                               NGINX QUIC
interop                             1
nginxinc/nginmesh_proxy_debug
                                 0
nginxinc/ngx-rust-tool
                                 0
nginxinc/mra_python_base
                                 0
nginxinc/nginmesh_proxy_init
                                 0
nginxinc/mra-fakes3
                                 0
sunweiguo@sunweiguodeMBP ~ % docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
927a35006d93: Pull complete
fc3910c70f9c: Pull complete
e11bfbf9fd54: Pull complete
fbb8b547daa2: Pull complete
0f1992aeebd8: Pull complete
f929dacee378: Pull complete
Digest:
sha256:0d17b565c37bcbd895e9d92315a05c1c3c9a29f762b011a10c54a66cd53c
9b31
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
sunweiguo@sunweiguodeMBP ~ % docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
nginx        latest    eeb9db34b331   21 months ago   134MB
rabbitmq     latest    092c6e9bcf1f   22 months ago   188MB
```

```
sunweiguo@sunweiguodeMBP ~ % docker run --name myNginx -p 8080:80 -
d nginx
7328835704c82b523681de7d3006b7520985aaf2e04e4c857539cd64a67ed9e6
sunweiguo@sunweiguodeMBP ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED
 STATUS          PORTS                  NAMES
7328835704c8   nginx     "/docker-entrypoint.…"   14 seconds ago
Up 12 seconds   0.0.0.0:8080->80/tcp   myNginx
sunweiguo@sunweiguodeMBP ~ %
```

> 参数说明:
> -name nginx-test: 容器名称。
> -p 8080:80: 端口进行映射，将本地 8080 端口映射到容器内部的 80 端口。
> -d nginx: 设置容器在在后台一直运行。

通过docker ps查看到目前有一个nginx容器正在运行，本地浏览器访问试试:

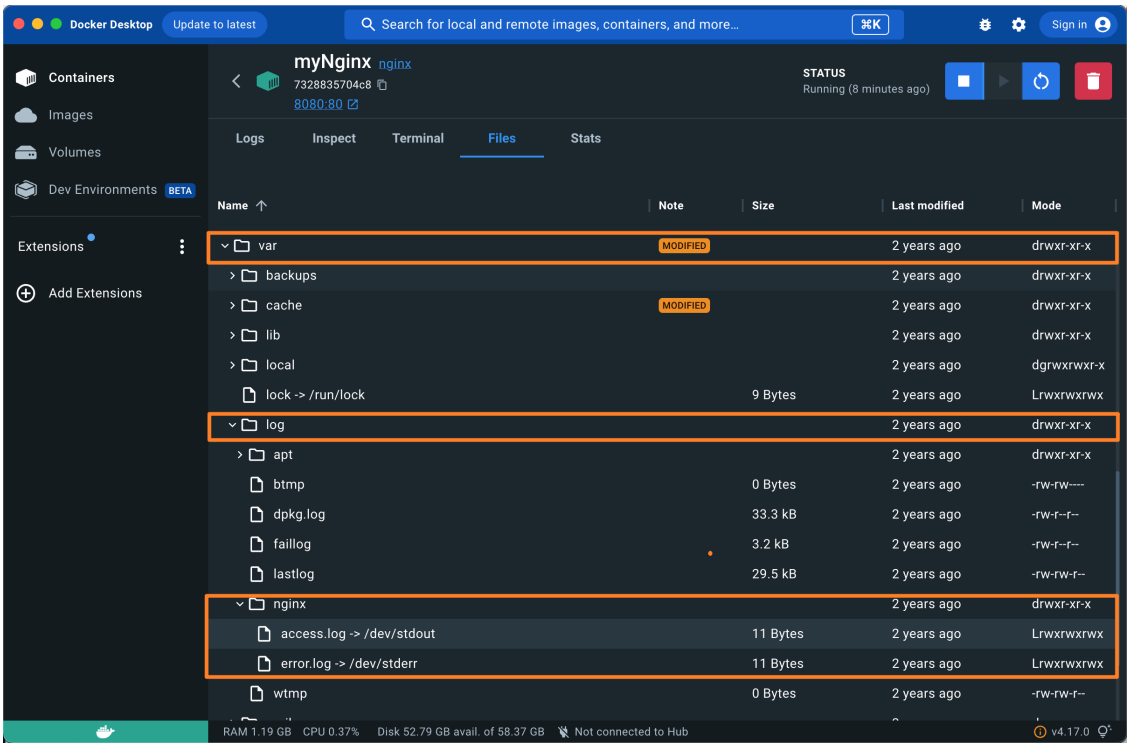


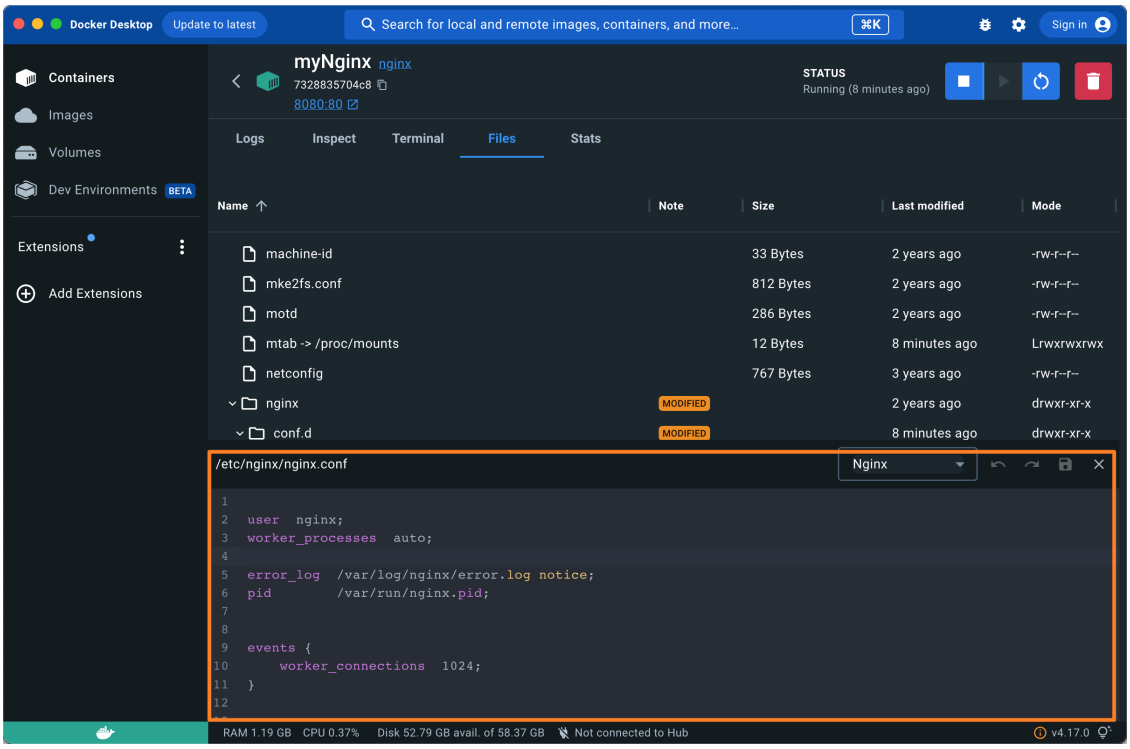
虽然我们成功启动了容器，并且成功访问到了资源，但是配置文件不方便修改，需要做一次映射，需要本地创建一个目录，将目录挂在到docker容器的nginx上，以后修改配置文件，只需要在本地配置文件中修改，重启nginx容器即可。

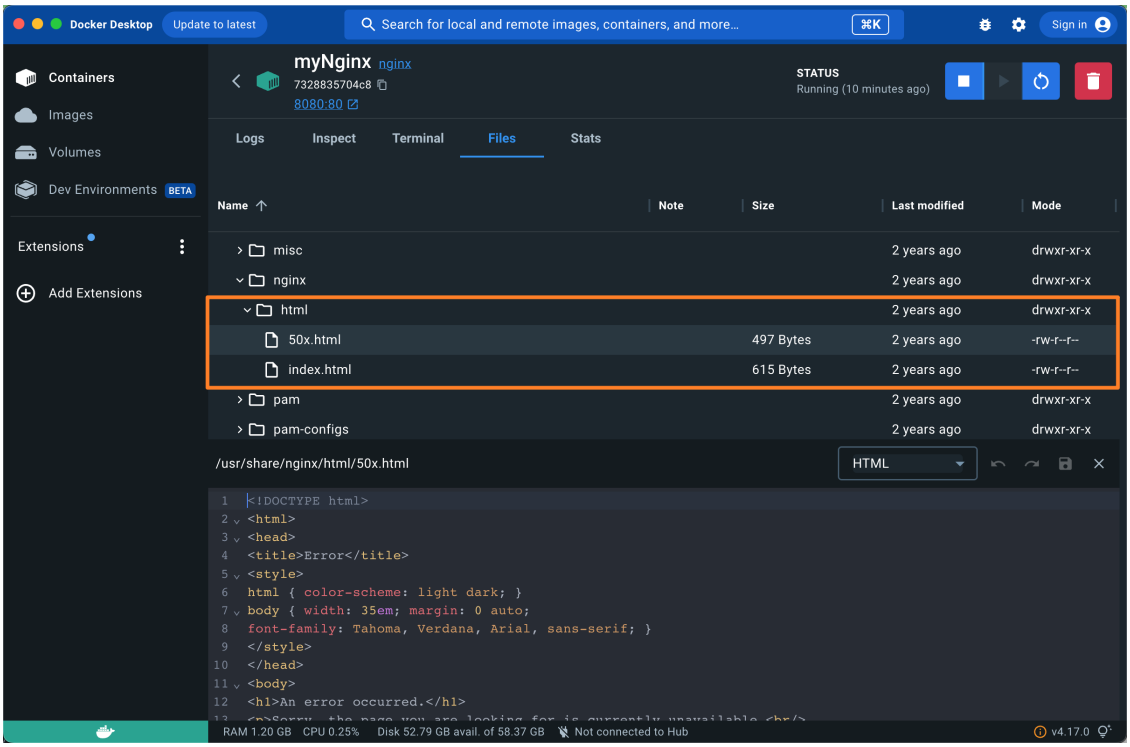
怎么做呢？也很简单，首先我们需要知道，通过docker方式安装的话，nginx配置文件一般目录是什么？

docker拉取下来的nginx配置文件路径一般情况下是:

> 日志文件位置: /var/log/nginx
> 配置文件位置: /etc/nginx
> 资源存放的位置: /usr/share/nginx/html


Docker Desktop
Update to latest
Search for local and remote images, containers, and more...
⌘K
Sign in
Containers
Images
Volumes
Dev Environments
BETA
Extensions
Add Extensions
myNginx
nginx
7328835704c8
8080:80
STATUS
Running (8 minutes ago)
Logs
Inspect
Terminal
Files
Stats
Name
Note
Size
Last modified
Mode
var
MODIFIED
2 years ago
drwxr-xr-x
backups
2 years ago
drwxr-xr-x
cache
MODIFIED
2 years ago
drwxr-xr-x
lib
2 years ago
drwxr-xr-x
local
2 years ago
dgrwxrwxr-x
lock -> /run/lock
9 Bytes
2 years ago
Lrwxrwxrwx
log
2 years ago
drwxr-xr-x
apt
2 years ago
drwxr-xr-x
btmp
0 Bytes
2 years ago
-rw-rw----
dpkg.log
33.3 kB
2 years ago
-rw-r--r--
faillog
3.2 kB
2 years ago
-rw-r--r--
lastlog
29.5 kB
2 years ago
-rw-rw-r--
nginx
2 years ago
drwxr-xr-x
access.log -> /dev/stdout
11 Bytes
2 years ago
Lrwxrwxrwx
error.log -> /dev/stderr
11 Bytes
2 years ago
Lrwxrwxrwx
wtmp
0 Bytes
2 years ago
-rw-rw-r--
RAM 1.19 GB
CPU 0.37%
Disk 52.79 GB avail. of 58.37 GB
Not connected to Hub
v4.17.0


Docker Desktop
Update to latest
Search for local and remote images, containers, and more...
⌘K
Sign in
Containers
Images
Volumes
Dev Environments
BETA
Extensions
Add Extensions
myNginx
nginx
7328835704c8
8080:80
STATUS
Running (8 minutes ago)
Logs
Inspect
Terminal
Files
Stats
Name
Note
Size
Last modified
Mode
machine-id
33 Bytes
2 years ago
-rw-r--r--
mke2fs.conf
812 Bytes
2 years ago
-rw-r--r--
motd
286 Bytes
2 years ago
-rw-r--r--
mtab -> /proc/mounts
12 Bytes
8 minutes ago
Lrwxrwxrwx
netconfig
767 Bytes
3 years ago
-rw-r--r--
nginx
MODIFIED
2 years ago
drwxr-xr-x
conf.d
MODIFIED
8 minutes ago
drwxr-xr-x
/etc/nginx/nginx.conf
Nginx
1
2 user nginx;
3 worker_processes auto;
4
5 error_log /var/log/nginx/error.log notice;
6 pid /var/run/nginx.pid;
7
8
9 events {
10 worker_connections 1024;
11 }
12
RAM 1.19 GB
CPU 0.37%
Disk 52.79 GB avail. of 58.37 GB
Not connected to Hub
v4.17.0



第一步：宿主机创建目录

```
mkdir -p  /Users/sunweiguo/Documents/docker/nginx/log
mkdir -p  /Users/sunweiguo/Documents/docker/nginx/html
mkdir -p  /Users/sunweiguo/Documents/docker/nginx/conf
mkdir -p  /Users/sunweiguo/Documents/docker/nginx/conf.d（注意：这是文件夹）
```

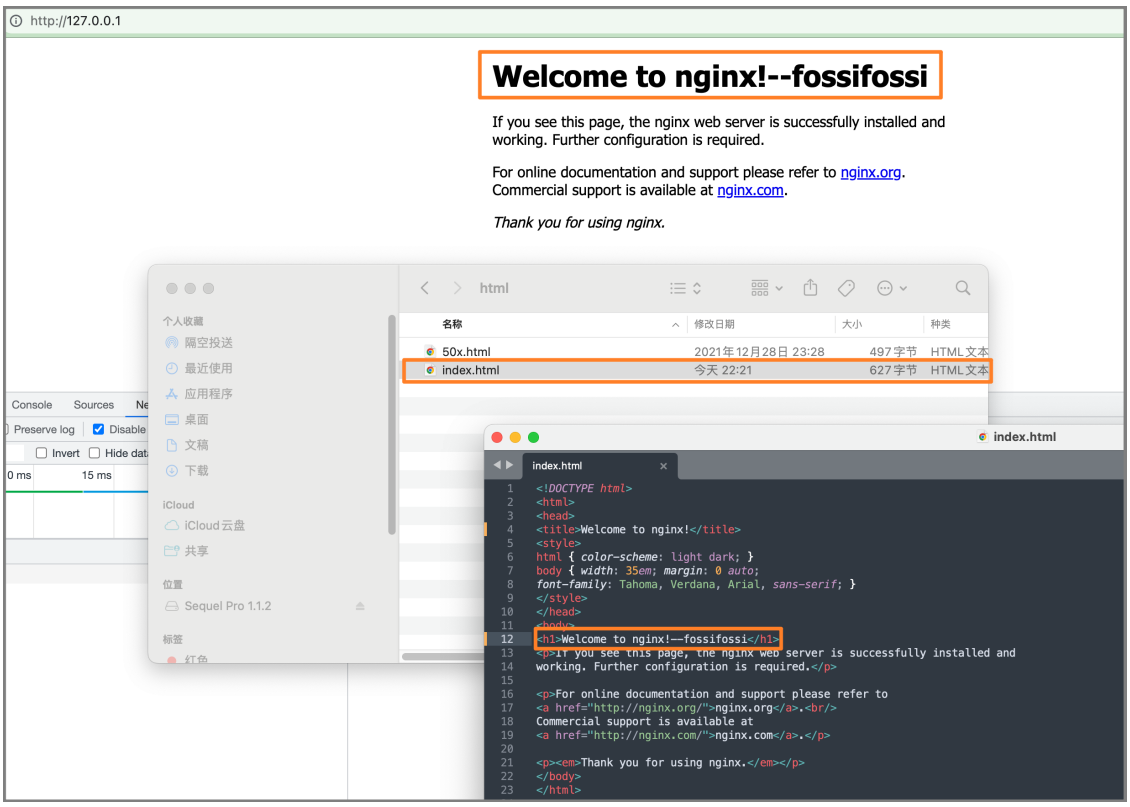
第二步：将docker安装的nginx里面文件复制到宿主机

```
# 将容器nginx.conf文件复制到宿主机
docker cp myNginx:/etc/nginx/nginx.conf
/Users/sunweiguo/Documents/docker/nginx/conf/nginx.conf
# 将容器conf.d文件夹下内容复制到宿主机
docker cp myNginx:/etc/nginx/conf.d
/Users/sunweiguo/Documents/docker/nginx
# 将容器中的html文件夹复制到宿主机
docker cp myNginx:/usr/share/nginx/html
/Users/sunweiguo/Documents/docker/nginx
#执行完以上操作后记得关闭、删除上面已经启动的nginx容器，可以直接通过desktop面板删除，也可以直接通过命令行删除
关闭该容器：docker stop nginx
删除该容器：docker rm nginx
或使用此命令，删除正在运行的nginx容器：docker rm -f nginx
```

第三步：运行镜像

```
docker run --name myNginx -p 80:80 -v
/Users/sunweiguo/Documents/docker/nginx/conf/nginx.conf:/etc/nginx
/nginx.conf -v
/Users/sunweiguo/Documents/docker/nginx/conf.d:/etc/nginx/conf.d -
v /Users/sunweiguo/Documents/docker/nginx/log:/var/log/nginx -v
/Users/sunweiguo/Documents/docker/nginx/html:/usr/share/nginx/html
-d nginx
```

测试挂载是否有问题，修改本地的index.html文件：



验证是没有问题的，这样就大功告成了，docker方式一旦熟悉之后，会发现十分方便，以后只需要在控制台点击启动容器即可。