

# 简介

当在宿主机启动较多的容器时候，如果都是手动操作会觉得比较麻烦而且容器出错，这个推荐使用docker单机编排工具docker-compose，docker-compose是docker容器的一种单机编排服务，docker-compose是一个管理多个容器的工具。比如可以解决容器之间的依赖关系，就像启动一个nginx前端服务的时候会调用后端的tomcat，那就得先启动tomcat，但是启动tomcat容器还需要依赖数据库，那就还得先启动数据库，docker-compose就可以解决这样的嵌套依赖关系，其完全可以替代docker run对容器进行创建、启动、和停止。

docker-compose项目是docker官方的开源项目，负责实现对docker容器集群的快速编排，docker-compose将所管理的容器分为三层，分别是工程（project），服务（service），以及容器（container）。

## 基础环境准备

- yum安装docker-compese

```
1 [root@docker-server1 ~]# yum install epel-release.noarch -y
2 [root@docker-server1 ~]# yum install docker-compose.noarch -y
3 [root@docker-server1 ~]# docker-compose version
4 docker-compose version 1.18.0, build 8dd22a9
5 docker-py version: 2.6.1
6 CPython version: 3.6.8
7 OpenSSL version: OpenSSL 1.0.2k-fips 26 Jan 2017
```

- 或者使用二进制安装，进入官网下载对应版本即可

官方下载网址：<https://github.com/docker/compose/releases>

- 相关参数

```
1 docker-compose --help
2 Define and run multi-container applications with Docker.
3 Usage:
4   docker-compose [-f <arg>...] [options] [COMMAND] [ARGS...]
5   docker-compose -h|--help
6
7 #选项说明:
8 -f, --file FILE #指定Compose 模板文件，默认为docker-compose.yml。
9 -p, --project-name NAME #指定项目名称，默认将使用当前所在目录名称作为项目名。
10 --verbose #显示更多输出信息
11 --log-level LEVEL #定义日志级别 (DEBUG, INFO, WARNING, ERROR, CRITICAL)
12 --no-ansi #不显示ANSI 控制字符
13 -v, --version #显示版本
14
15 #以下为命令选项，需要在docker-compose.yml|yaml 文件所在在目录里执行
16 build #构建镜像
17 bundle #从当前docker compose 文件生成一个以<当前目录>为名称的json格式的Docker Bundle 备份文件
18 config -q #查看当前配置，没有错误不输出任何信息
19 create #创建服务，较少使用
20 down #停止和删除所有容器、网络、镜像和卷
21 #events #从容器接收实时事件，可以指定json 日志格式，较少使用
```

```

22 exec #进入指定容器进行操作
23 help #显示帮助信息
24 images #显示镜像信息，较少使用
25 kill #强制终止运行中的容器
26 logs #查看容器的日志
27 pause #暂停服务
28 port #查看端口
29 ps #列出容器，较少使用
30 pull #重新拉取镜像，镜像发生变化后，需要重新拉取镜像，较少使用
31 push #上传镜像
32 restart #重启服务，较少使用
33 rm #删除已经停止的服务
34 run #一次性运行容器
35 scale #设置指定服务运行的容器个数
36 start #启动服务，较少使用
37 stop #停止服务，较少使用
38 top #显示容器运行状态
39 unpause #取消暂定
40 up #创建并启动容器，较少使用

```

## 启动单个容器

- 编写docker-compose文件

```

1 [root@docker-server1 ~]# mkdir /opt/docker
2 [root@docker-server1 ~]# cd /opt/docker/
3 [root@docker-server1 docker]# vim docker-compose.yml
4 service-nginx:
5     image: nginx
6     container_name: nginx_web1
7     ports:
8     - "80:80"

```

- 启动容器

```

1 [root@docker-server1 docker]# docker-compose up -d
2 Creating docker_service-nginx_1 ... done
3 [root@docker-server1 docker]# docker-compose ps
4
5      Name                                Command                                State                                Ports
6 -----
7 nginx_web1    /docker-entrypoint.sh nginx ...    Up                                0.0.0.0:80->80/tcp,:::80->80/tcp

```

## 启动多个容器

- 编辑docker-compose文件

```

1 [root@docker-server1 docker]# cat docker-compose.yml
2 service-nginx:
3     image: nginx
4     container_name: nginx_web1
5     ports:

```

```

6      - "80:80"
7
8  service-tomcat:
9      image: tomcat
10     container_name: tomcat_web1
11     ports:
12     - "8080:8080"
13 [root@docker-server1 docker]# docker-compose up -d
14 nginx_web1    /docker-entrypoint.sh nginx ...    Up      0.0.0.0:80-
>80/tcp,:::80->80/tcp
15 tomcat_web1   catalina.sh run                Up      0.0.0.0:8080-
>8080/tcp,:::8080-
16                                     >8080/tcp

```

## 定义数据卷挂载

- 创建数据卷目录和文件

```

1 [root@docker-server1 docker]# mkdir -p /data/nginx
2 [root@docker-server1 docker]# echo 'docker nginx' > /data/nginx/index.html

```

- 编辑配置文件

```

1 [root@docker-server1 docker]# cat docker-compose.yml
2 service-nginx:
3     image: nginx
4     container_name: nginx_web1
5     volumes:
6     - /data/nginx:/usr/share/nginx/html
7     ports:
8     - "80:80"
9
10  service-tomcat:
11     image: tomcat
12     container_name: tomcat_web1
13     ports:
14     - "8080:8080"

```

- 访问测试

```

1 [root@docker-server1 docker]# curl localhost
2 docker nginx

```

## 相关操作演示

- 重启单个容器

```

1 [root@docker-server1 docker]# docker-compose restart service-nginx

```

- 重启所有容器

```

1 [root@docker-server1 docker]# docker-compose restart

```

- 停止单个容器

```
1 | [root@docker-server1 docker]# docker-compose stop service-nginx
```

- 启动单个容器

```
1 | [root@docker-server1 docker]# docker-compose start service-nginx
```

- 停止所有容器

```
1 | [root@docker-server1 docker]# docker-compose stop
```

- 启动所有容器

```
1 | [root@docker-server1 docker]# docker-compose start
```