

docker 5. docker-compose (容器编排)

什么是docker-compose?

Docker Compose 是 Docker 官方编排 (Orchestration) 项目之一, 负责快速在集群中部署分布式应用。

为什么用docker-compose? (个人认为)

- 1.配置项很直观, 清晰, 方便查看
- 2.同时启动多个服务, 不用一个一个启动, 跟一键式装机一样, 安装好环境, 跑起来即可
- 3.部署方便, 只需要安装好docker-compose, 通过.yml文件启动服务即可

几个术语。

- 服务 (service): 一个应用容器, 实际上可以运行多个相同镜像的实例。
- 项目 (project): 由一组关联的应用容器组成的一个完整业务单元。

可见, 一个项目可以由多个服务 (容器) 关联而成, Compose 面向项目进行管理。

具体通过[*dev_env*](#) 这个仓库的[*docker-compose.yml*](#)讲解

简单例子:

version: "2" #compose的版本, 有1, 2, 3三个版本

services: #服务

mysql: #服务名

image: mysql:5.6.36 #镜像

container_name: mysql #容器名称

restart: always #是否重启

ports: #映射的端口 (可以是多个)

- "3306:3306"

volumes: #映射的目录 (可以多个)

- ~/Desktop/mysql/backup:/mysql/backup

environment: #Dockerfile的环境变量 (在构建镜像的时候指定, 在enterpoint的时候可以用, 也就是第一次启动容器)

MYSQL_ROOT_PASSWORD: root

MYSQL_DATABASE: super_manager_v2

MYSQL_USER: root

MYSQL_PASSWORD: root

docker-compose常用相关命令:

option(参数):

- `-f, --file FILE` 指定使用的 Compose 模板文件，默认为 `docker-compose.yml`，可以多次指定。
- `-p, --project-name NAME` 指定项目名称，默认将使用所在目录名称作为项目名。
- `--x-networking` 使用 Docker 的可拔插网络后端特性
- `--x-network-driver DRIVER` 指定网络后端的驱动，默认为 `bridge`
- `--verbose` 输出更多调试信息。
- `-v, --version` 打印版本并退出。

commands(命令):

up 它将尝试自动完成包括构建镜像，（重新）创建服务，启动服务，并关联服务相关容器的一系列操作。

down 此命令将会停止 `up` 命令所启动的容器，并移除网络等

exec 进入指定容器

config 验证 Compose 文件格式是否正确，若正确则显示配置，若格式错误显示错误原因。

images 列出 Compose 文件中包含的镜像

logs 查看服务容器的输出。默认情况下，`docker-compose` 将对不同的服务输出使用不同的颜色来区分。可以通过 `--no-color` 来关闭颜色（调试问题的时候很有用）

pause 暂停一个服务容器

unpause 恢复处于暂停状态中的服务。

port 打印某个容器端口所映射的公共端口

ps 列出项目中目前的所有容器

pull 拉取服务依赖的镜像

push 推送服务依赖的镜像到 Docker 镜像仓库

restart 重启项目中的服务。

rm 删除所有（停止状态的）服务容器

run 在指定服务上执行一个命令。

start 启动已经存在的服务容器

stop 停止已经处于运行状态的容器，但不删除它。通过 `docker-compose start` 可以再次启动这些容器。

top	查看各个服务容器内运行的进程
scale	设置指定服务运行的容器个数。
build	构建（重新构建）项目中的服务容器
version	查看版本
kill	通过发送 SIGKILL 信号来强制停止服务容器。(docker-compose kill -s SIGINT)

启动一个mysql:

```
docker-compose -f ~/Desktop/docker-share/data/compose/mysql.yml up
```

安装:

linux 安装docker-compose:

- 1.下载(查看最新版本号<https://github.com/docker/compose/releases>) 可以直接下载压缩文件拷贝到/usr/local/bin

```
sudo curl -L https://github.com/docker/compose/releases/download/1.17.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```
- 2.修改文件属性为可执行

```
sudo chmod +x /usr/local/bin/docker-compose
```
- 3.检查是否安装了

```
docker-compose version
```