[Docker Compose](https://links.jianshu.com/go?to=https%3A%2F%2Fgithub.com%2Fdocker%2Fcompose" \t "_blank)是 docker 提供的一个命令行工具，用来定义和运行由多个容器组成的应用。使用 compose，我们可以通过 YAML 文件声明式的定义应用程序的各个服务，并由单个命令完成应用的创建和启动。

🡪  
Compose 中有两个重要的概念：

　　服务(service):一个应用的容器，实际上可以包括若干运行相同镜像的容器实例

　　项目(project):由一组关联的应用容器组成的一个完整业务单元，在docker-compose.yml中定义

🡪

Compose基本原理：

Compose 项目是由Python编写的，实际上就是调用了Docker服务提供的API来对容器进行管理，因此，只要所在的操作系统的平台支持Docker API，就可以在其上利用Compose来进行编排管理.

🡪

大部分的compose命令都需要到docker-compose.yml文件所在的目录下才能执行。

🡪

docker compose命令

docker-compose stop # 停止docker compose服务

docker-compose ps # 列举docker compose服务

docker-compose restart # 重启docker compose服务

docker-compose kill # 杀死docker compose服务

🡪docker-compose.yml文件配置

Services: # 指定服务列表

image: [image名] #指定使用的image

build: [路径] # 指定Dockerfile 所在文件夹的路径。 Compose 将会利用它自动构建这个镜像，然后使用这个镜像。

command: [命令] # 容器启动后默认执行的命令。

links: 指令此服务容器与其他什么容器建立链接

external\_links: 链接到 docker-compose.yml 外部的容器，甚至 并非 Compose 管理的容器。

ports: 暴露端口信息。使用宿主：容器 （HOST:CONTAINER）格式或者仅仅指定容器的端口（宿主将会随机选择端口）都可以。

expose: 暴露端口，但不映射到宿主机，只被连接的服务访问。仅可以指定内部端口为参数

volumes: 卷挂载路径设置。可以设置宿主机路径 （HOST:CONTAINER） 或加上访问模式 （HOST:CONTAINER:ro）。

Volumes\_from: 从另一个服务或容器挂载它的所有卷

environment: 设置环境变量你可以使用数组或字典两种格式。

net: 设置网络模式

dns: 配置 DNS 服务器。可以是一个值，也可以是一个列表。

Depends\_on: 此服务基于某些服务

🡪