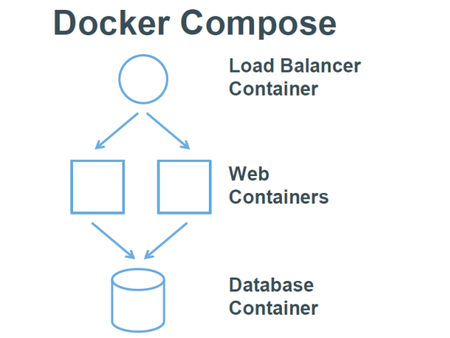
**Docker Compose 项目**

允许用户在一个模板（YAML 格式）中定义一组相关联的应用容器



* 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。

## Compose 命令说明

* --verbose 输出更多调试信息。
* --version 打印版本并退出。
* -f, --file FILE 使用特定的 compose 模板文件，默认为 docker-compose.yml。
* -p, --project-name NAME 指定项目名称，默认使用目录名称。

### build 构建或重新构建服务。

### kill 通过发送 SIGKILL 信号来强制停止服务容器。

### logs 查看服务的输出。

### port 打印绑定的公共端口。

### ps 列出所有容器。

### pull 拉取服务镜像。

### rm 删除停止的服务容器。

### run 在一个服务上执行一个命令。

如果不希望自动启动关联的容器，可以使用 --no-deps 选项

### scale

设置同一个服务运行的容器个数。

通过 service=num 的参数来设置数量。例如：

$ docker-compose scale web=2 worker=3

### 环境变量

## YAML 模板文件

每个服务都必须通过 image 指令指定镜像或 build 指令（需要 Dockerfile）来自动构建。

### command 覆盖容器启动后默认执行的命令。

### links 链接到其它服务中的容器。

links:

- db

- db:database

- redis

使用的别名将会自动在服务容器中的 /etc/hosts 里创建。例如：

172.17.2.186 db

172.17.2.186 database

172.17.2.187 redis

### external\_links

链接到 docker-compose.yml 外部的容器, 参数格式跟 links 类似。

### ports

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。

ports:

- "3000"

- "8000:8000"

- "49100:22"

- "127.0.0.1:8001:8001"

### 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

expose:

- "3000"

- "8000"

### 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:

- /var/lib/mysql

- cache/:/tmp/cache

- ~/configs:/etc/configs/:ro

### volumes\_from

从另一个服务或容器挂载它的所有卷。

volumes\_from:

- service\_name

- container\_name

### environment

只给定名称的变量会自动获取它在 Compose 主机上的值，可以用来防止泄露不必要的数据。

environment:

RACK\_ENV: development

SESSION\_SECRET:

environment:

- RACK\_ENV=development

- SESSION\_SECRET

### env\_file

从文件中获取环境变量，可以为单独的文件路径或列表。