

2.3.3 容器管理

讲师：汪洋





目录

1

Docker 基础概念及命令

2

Docker-Compose



1

Docker 基础概念及命令



Docker 三个重要概念：仓库 (Repository)、镜像 (image) 和 容器 (Container)

```
docker run --name MyWordPress --link db:mysql -p 8080:80 -d wordpress
```

Docker 指令的基本用法：

docker + 命令关键字(COMMAND) + 一系列的参数



<code>docker info</code>	守护进程的系统资源设置
<code>docker search</code>	Docker 仓库的查询
<code>docker pull</code>	Docker 仓库的下载
<code>docker images</code>	Docker 镜像的查询
<code>docker rmi</code>	Docker 镜像的删除
<code>docker ps</code>	容器的查询
<code>docker run</code>	容器的创建启动
<code>docker start/stop</code>	容器启动停止

Docker 指令除了单条使用外，还支持赋值、解析变量、嵌套使用



每个容器被创建后，都会分配一个 CONTAINER ID 作为容器的唯一标示，后续对容器的启动、停止、修改、删除等所有操作，都是通过 CONTAINER ID 来完成，偏向于数据库概念中的主键

```
docker ps --no-trunc
```

查看

```
docker stop/start CONTAINERID
```

停止

```
docker start/stop Mywordpress
```

通过容器别名启动/停止

```
docker inspect Mywordpress
```

查看容器所有基本信息

```
docker logs Mywordpress
```

查看容器日志

```
docker stats Mywordpress
```

查看容器所占用的系统资源

```
docker exec 容器名 容器内执行的命令
```

容器执行命令

```
docker exec -it 容器名 /bin/bash
```

登入容器的bash



`--restart=always`

容器的自动启动

`-h x.xx.xx`

设置容器主机名

`--dns xx.xx.xx.xx`

设置容器使用的 DNS 服务器

`--dns-search`

DNS 搜索设置

`--add-host hostname:IP`

注入 hostname <> IP 解析

`--rm`

服务停止时自动删除



2

Docker-Compose



Docker 提倡理念是 “一个容器一个进程”，假设一个服务需要由多个进程组成，就需要多个容器组成一个系统，相互分工和配合对外提供完整服务

比如：博客系统

组件1: mariadb

组件2: WordPress 的 apache web

在启动容器是，同一台主机下如果两个容器之间需要由数据交流，使用 `--link` 选项建立两个容器之间的互联，前提是建立是 mariadb 已经开启

```
docker start db
```

```
docker start Mywordpress
```

停止:

```
docker stop db Mywordpress 或 docker stop Mywordpress 在 docker top db
```



容器编排工具，允许用户在一个模板（YAML 格式）中定义一组相关联的容器，会根据 `--link` 等参数，对启动的优先级进行排序

```
curl -L https://github.com/docker/compose/releases/download/1.14.0/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose
```

```
docker version
```



<code>-f</code>	指定使用的 yaml 文件位置
<code>ps</code>	显示所有容器信息
<code>restart</code>	重新启动容器
<code>logs</code>	查看日志信息
<code>config -q</code>	验证 yaml 配置文件是否正确
<code>stop</code>	停止容器
<code>start</code>	启动容器

<code>up -d</code>	启动容器项目
<code>pause</code>	暂停容器
<code>unpause</code>	恢复暂停
<code>rm</code>	删除容器



Docker-compose yml 演示案例

[点击跳转](#)



0 v e r