

Docker核心技术之 Docker Compose



Docker核心技术之Docker Compose - 课程概要

- Docker Compose 简介
- 了解 Docker Compose File
- Docker Compose 应用
- 总结

Docker核心技术之Docker Compose

Docker Compose 简介

Docker Compose是什么？

- Docker Compose是一个能一次性定义和管理多个Docker容器的工具。

- 详细地说：

Compose中定义和启动的每一个容器都相当于一个服务(service)

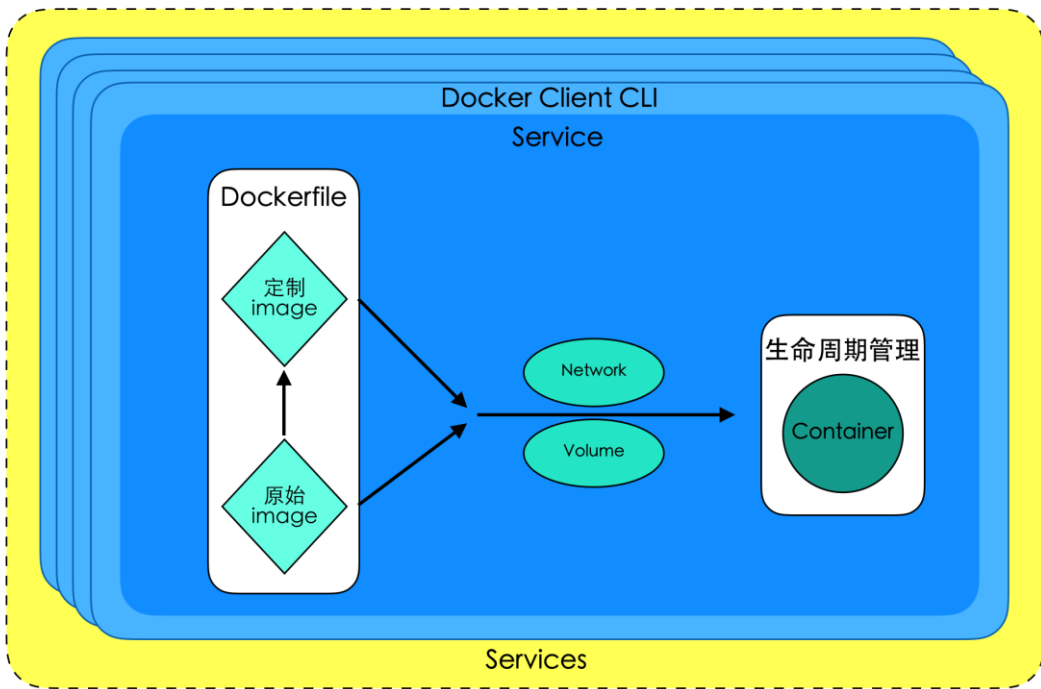
Compose中能定义和启动多个服务，且它们之间通常具有协同关系

- 管理方式：

使用YAML文件来配置我们应用程序的服务。

使用单个命令（docker-compose up），就可以创建并启动配置文件中配置的所有服务。

Docker Compose 工作原理



Docker Compose File + Docker Compose CLI

Docker Compose安装

- Docker for Mac与Docker for Windows自带docker-compose
 - Linux下需要单独安装：
 - 第一步：`sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose`
 - 第二步：`sudo chmod +x /usr/local/bin/docker-compose`
- 终端中使用`docker-compose --version`查看安装的版本
- 这里示例安装版本是1.21.2，很可能您看到这里时，已经出现更新的版本，因此建议换成最新版本。[查看最新版本](#)。
- [其他安装方法查看](#)

Docker Compose CLI

- 利用**docker-compose --help**查看或者[查看官方文档](#)
- 对比后会发现：Docker Compose CLI的很多命令的功能和Docker Client CLI是相似的。最主要的区别就是前者能一次性运行管理多个容器，后者只能一次管理一个。

Docker核心技术之Docker Compose

了解 Docker Compose File

Docker Compose File版本

- [Docker Compose File](#) 有多个版本，基本是向后兼容的，但也有极个别配置项高版本中没有。
- 在docker-compose.yml一开始就需要利用version关键词标明当前file使用的版本

Compose file format	Docker Engine release
3.6	18.02.0+
3.5	17.12.0+
3.4	17.09.0+
3.3	17.06.0+
3.2	17.04.0+
3.1	1.13.1+
3.0	1.13.0+
2.4	17.12.0+
2.3	17.06.0+
2.2	1.13.0+
2.1	1.12.0+
2.0	1.10.0+
1.0	1.9.1.+

Docker Compose File TOP配置参数概览

Docker Compose File 顶级配置项：

- version：指定Docker Compose File版本号
- services：定义多个服务并配置启动参数
- volumes：声明或创建在多个服务中共同使用的数据卷对象
- networks：定义在多个服务中共同使用的网络对象
- configs：声明将在本服务中要使用的一些配置文件
- secrets：声明将在本服务中要使用的一些秘钥、密码文件
- x-***：自定义配置。主要用于复用相同的配置。

[更多详细配置](#)

Docker Compose File 参考示例

```
version: '3.6'
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:6.2.4
    environment:
      - cluster.name=docker-cluster
      - bootstrap.memory_lock=false
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
    volumes:
      - esdata:/usr/share/elasticsearch/data
    ports:
      - 9200:9200 # http端口
      - 9300:9300 # tcp端口
    networks:
      - elk-overlay
    deploy:
      placement:
        constraints:
          - node.role == manager
  logstash:
    image: docker.elastic.co/logstash/logstash:6.2.4
    ports:
      - "5000:5000"
    environment:
      LS_JAVA_OPTS: "-Xmx256m -Xms256m"
    networks:
      - elk-overlay
    deploy:
      replicas: 3
  kibana:
    image: docker.elastic.co/kibana/kibana:6.2.4
    ports:
      - "5601:5601"
    networks:
      - elk-overlay
    deploy:
      placement:
        constraints:
          - node.role == manager
volumes:
  esdata:
    driver: local
networks:
  elk-overlay:
    driver: overlay
```

Docker核心技术之Docker Compose

Docker Compose 应用

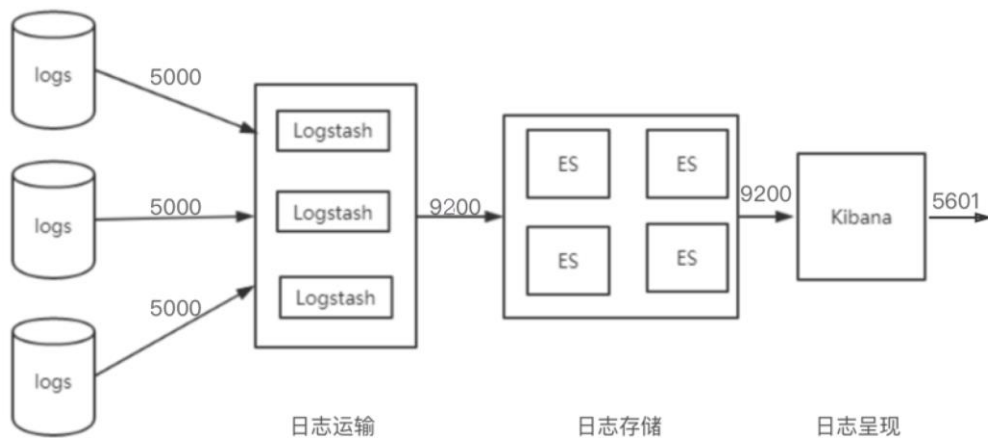
Docker Compose 案例一 小型web服务项目搭建

步骤：

- 搭建一个flask的小型web项目
- 根据项目环境，利用Dockerfile构建镜像
- 撰写docker-compose.yaml配置文件，启动项目

Docker Compose 案例二 单机环境ELK系统搭建（一）

- [ELK](#)工作原理介绍



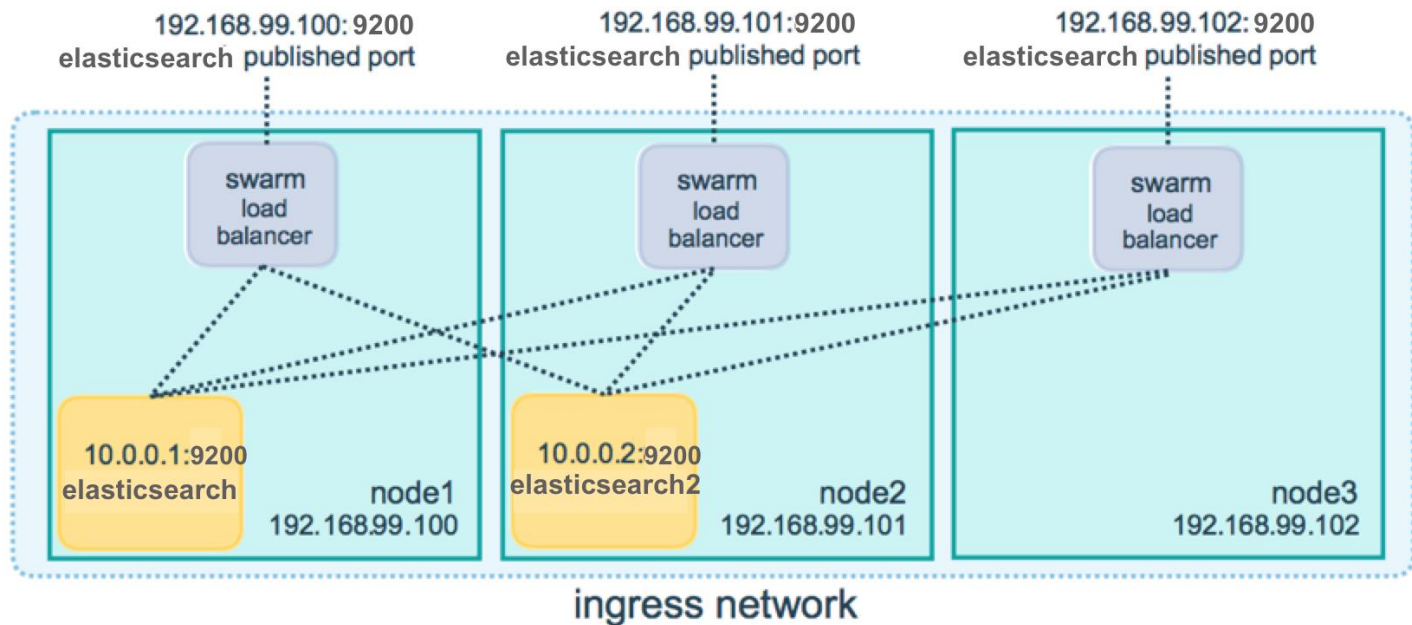
Docker Compose 案例二 单机环境ELK系统搭建（二）

步骤：

- 配置单机版的docker-compose.yaml文件（[ELK镜像地址](#)）
- 利用docker-compose up启动环境

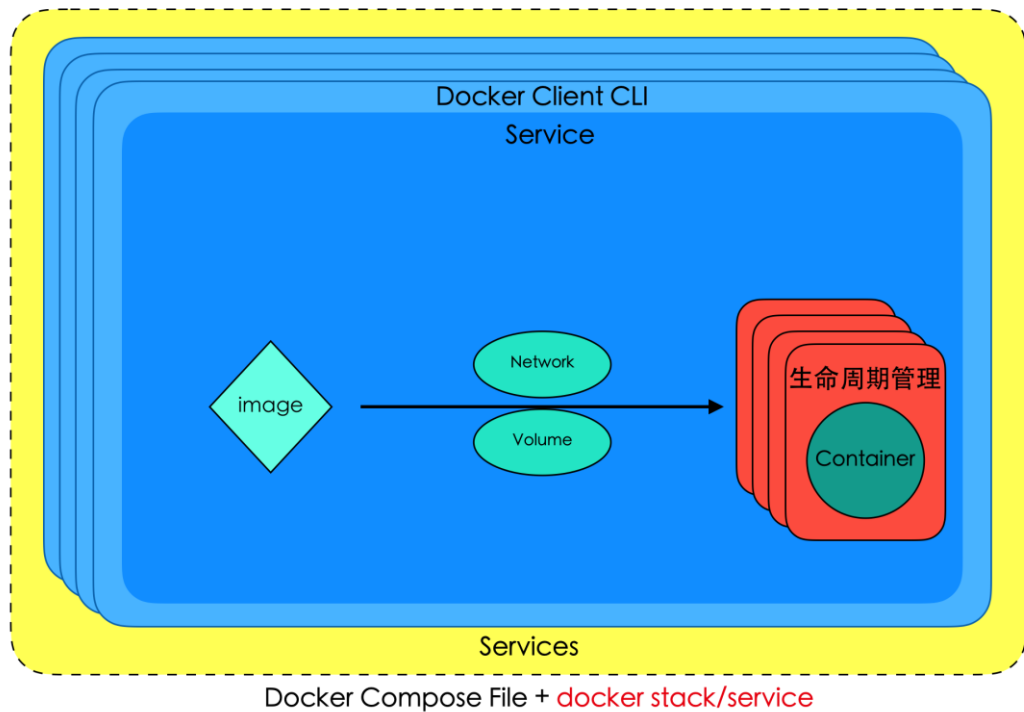
Docker Compose 案例三 多主机环境ELK系统搭建（一）

- Swarm 介绍



Docker Compose 案例三 多主机环境ELK系统搭建（二）

- 集群版Docker Compose工作原理



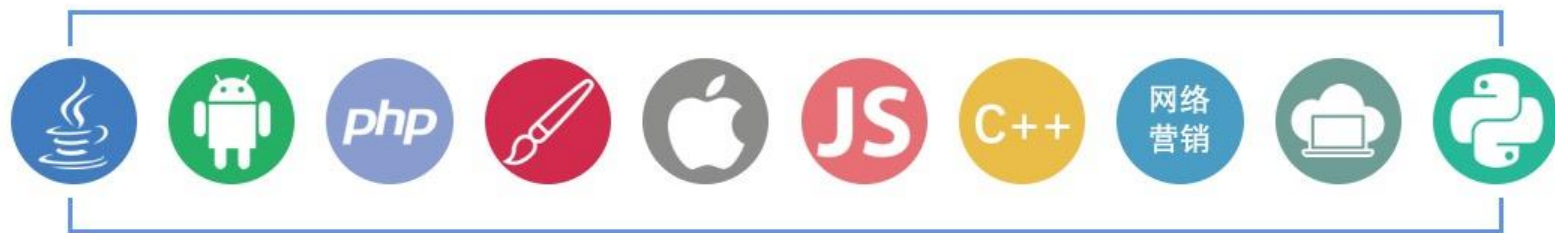
Docker Compose 案例三 多主机环境ELK系统搭建（三）

步骤

- 使用[docker swarm](#)配置多个docker node集群节点
- 配置集群版ELK的docker-compose.yaml文件
- 利用docker stack deploy部署集群版ELK环境

Docker核心技术之Docker Compose

总结



Thank you!

改变中国IT教育，我们正在行动

BOXUEGU.COM