

# D O C K E R 分 享 会

拓展应用部 叶利生

 Docker Hub

# docker

- docker基本概念
- 如何使用docker
- ATM1项目的实践

# docker是什么

Docker 是一个开源的应用容器引擎，基于 Go 语言 并遵从Apache2.0协议开源。可以让开发者打包他们的应用以及依赖包到一个轻量级、可移植的容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化，它将软件组件包装成一个完整的标准化单元，其中包含所有要运行的内容。 无论是代码，运行时环境，工具还是程序库。它保证软件能始终按预期那样运行。容器是完全使用沙箱机制，相互之间不会有任何接口（类似 iPhone 的 app）,更重要的是容器性能开销极低。可以在给定的主机上同时运行多个容器。 它是轻量级的，可立即启动并使用较少的内存。 它是默认安全的，因为每个容器彼此隔离。

**镜像：** 镜像概念类似于虚拟机里的镜像，是一个只读的模板，一个独立的文件系统，包括运行容器所需的数据，可以用来创建新的容器(类)

---

**DOCKERFILE：** 一个包含用于组合映像的命令的文本文档。可以使用在命令行中调用任何命令。 Docker通过读取Dockerfile中的指令自动生成映像。（编写类的实现代码）

---

**容器：** Docker容器是由Docker镜像创建的运行实例（创建类的实例）

---

**仓库：** 保存docker镜像仓库（git仓库）

# docker的特点

1. 持续集成
2. 版本控制
3. 可移植性
4. 隔离性
5. 安全性

## 二、如何使用 D O C K E R

## MongoDB dockerfile



dockerfile

构建镜像 `docker build -t yels/mongo .`

快速环境搭建环境，切入学习重点

部署MongoDB数据库运行环境的例子

```
docker run --rm -p=27017:27017 -v=$(pwd)/db/mongo/data:/data/db --name demo-mongo mongo
```

```
docker exec -it demo-mongo mongo admin
```



## 学习常用的CLI及使用GUI工具辅助操作

- 镜像的指令
- 容器的指令
- portainer

```
docker run -d -p 9000:9000 --name portainer --restart always -v  
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer
```

# 容器编排及部署

# docker compose

使用 Docker 的时候，定义 Dockerfile 文件，然后使用 `docker build`、`docker run` 等命令操作容器。然而微服务架构的应用系统一般包含若干个微服务，每个微服务一般都会部署多个实例，如果每个微服务都要手动启停，那么效率之低，维护量之大可想而知 使用 Docker Compose 可以轻松、高效的管理容器，它是一个用于定义和运行多容器 Docker 的应用程序工具。

# docker machine

Docker 官方提供的一个工具，它可以帮助我们在远程的机器上安装 Docker，或者在虚拟机 host 上直接安装虚拟机并在虚拟机中安装 Docker。我们还可以通过 `docker-machine` 命令来管理这些虚拟机和 Docker。

# docker swarm

Swarm是Docker官方提供的一款集群管理工具，其主要作用是把若干台Docker主机抽象为一个整体，并且通过一个入口统一管理这些Docker主机上的各种Docker资源。Swarm和Kubernetes比较类似，但是更加轻，具有的功能也较kubernetes更少一些

# 三、D O C K E R 实 践

atm1脚本服务的docker应用

Thank you