### DockerCompose

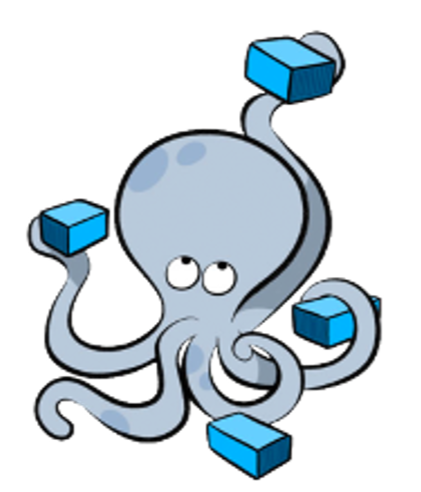
#### 前言：

前面我们已经实现了用Docker来部署像Redis、Mysql这样的中间件，还有利用Dockerfile来实现微服务的自定义镜像构建还有docker部署，但是，所有的这些部署都是我们手动一个个完成的，你想象一下，在实际生产环境下，微服务的数量是非常多的，这么多的微服务我们都一个个去构建部署，这个工作量几乎是不可能的，所以我们一定需要一种集群部署的手段，这就是我们接下来学习的DockerCompose。

#### 初始DockerCompose：

什么是DockerCompose?

请看这个章鱼：



见图知意，章鱼手上拿的是容器，章鱼在帮我们部署容器！DockerCompose可以基于Compose文件帮我们快速部署分布式应用，而无需手动一个个创建和一个个运行容器！即DockerCompose是部署分布式应用的帮手，但是DockerCompose做集群部署，是要基于Compose文件才行。

什么是Compose文件？

Compose文件是一个文本文件！通过指令来定义集群中每个容器如何运行。可以回顾一下，我们以前是通过docker run命令来运行每个容器，而现在是用Compose文件来定义集群中的N个容器如何运行，于是可以认为Compose文件就是N个docker run命令的集合，只不过，Compose文件不是直接用docker run，而是把run里面的东西用另外一种语法：即指令来代替。这个语法长什么样呢？

version: "3.8"

services:

  mysql:

    image: mysql:5.7.25

environment:

MYSQL\_ROOT\_PASSWORD: 123

    volumes:

     - "/tmp/mysql/data:/var/lib/mysql"

     -  "/tmp/mysql/conf/hmy.cnf:/etc/mysql/conf.d/hmy.cnf"

  web:

    build: .

    ports:

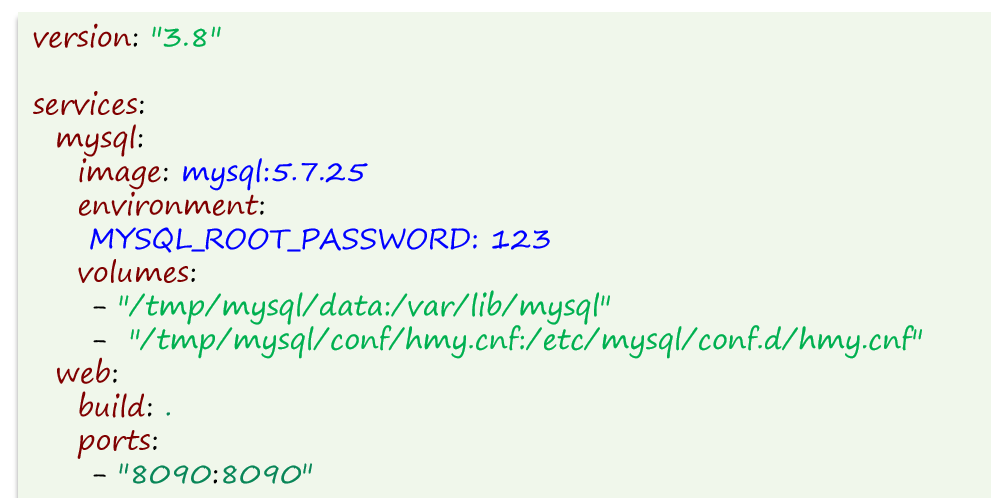
     - "8090:8090"

可以看到，这个语法整体是一个yaml格式：

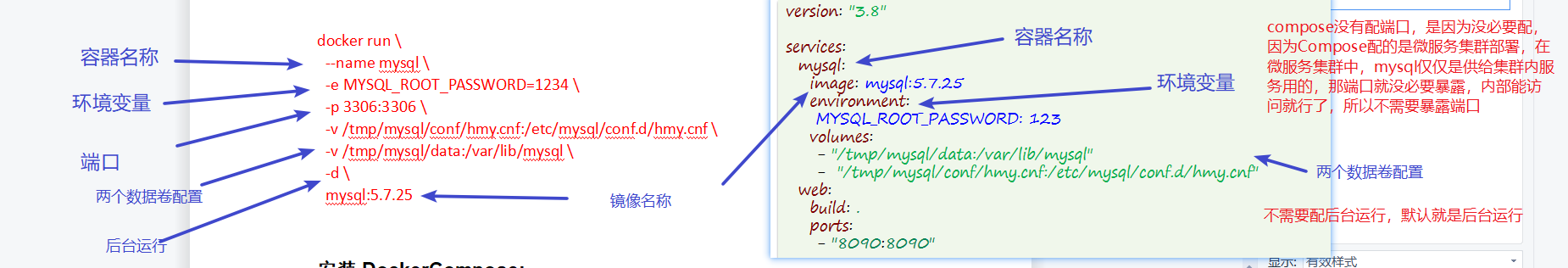
version：是yaml的版本

version往下就是具体的微服务定义，可以看到services有两个二级配置：mysql、web，那么就是两个微服务配置，我们可以看到上面的是mysql的配置；下面的web是web工程的配置。这是Compose文件中比较典型的两种配置方式。**上面是第一种部署服务的方式，就是基于镜像直接部署；下面没有指定镜像，容器部署没有镜像怎么部署？其实下面是把docker build和docker run都包含进来，即临时构建镜像并运行。**

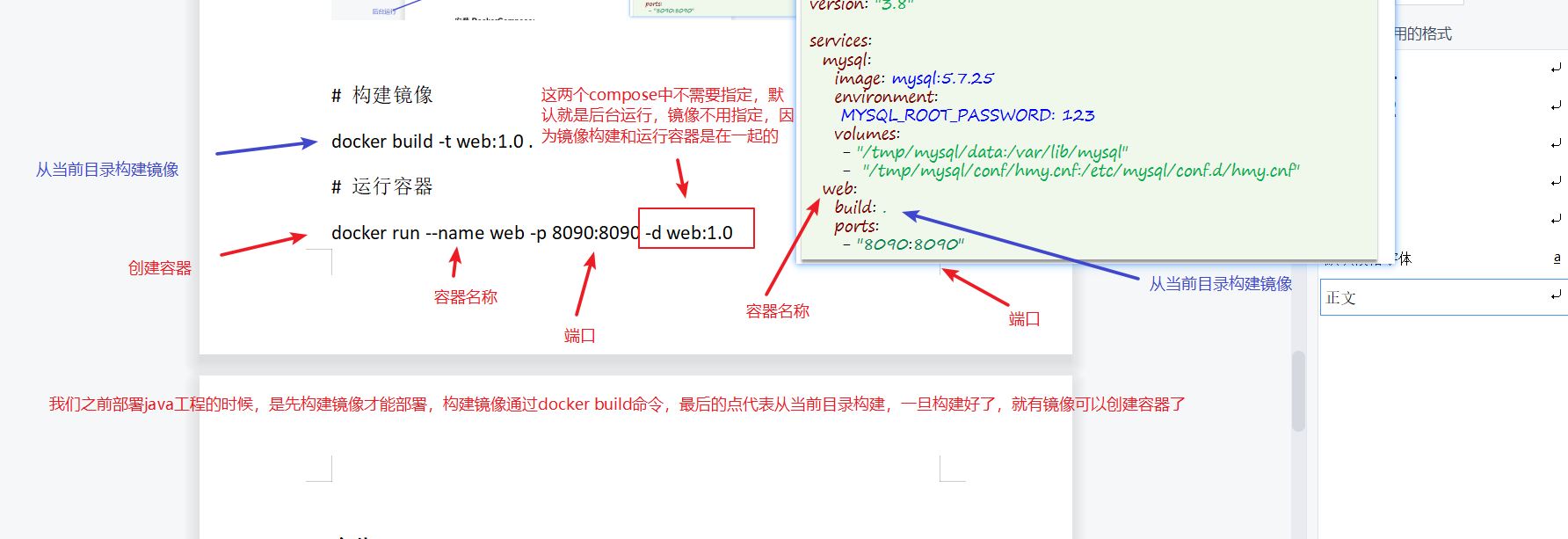
##### 第一种方式：基于镜像直接部署



即然services是在定义容器的部署，我们有说了Compose文件是吧docker run集合进来了，然后把语法转换了一下：我们可以比较一下：见下图，可以发现，确实把docker run中的所有参数都转换成了Compose中的指令了！



##### 第二种方式：**临时构建镜像并运行**



以上两种方式可以发现，DockerCompose文件就是把Docker Run的各种参数转化成指令去定义。DockerCompose的详细语法可以参考官网：

<https://docs.docker.com/compose/compose-file/>

#### 安装DockerCompose:

1：下载DockerCompose

Linux下需要通过命令下载：

curl -L https://github.com/docker/compose/releases/download/1.23.1/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose

如果下载速度较慢，或者下载失败，可以使用课前资料提供的docker-compose文件，上传到/usr/local/bin目录也可以。

2：修改文件权限：chmod +x /usr/local/bin/docker-compose

3：base自动补全命令：

curl -L https://raw.githubusercontent.com/docker/compose/1.29.1/contrib/completion/bash/docker-compose > /etc/bash\_completion.d/docker-compose

4：这里出现错误，需要修改自己的hosts文件后再次执行3：

echo "199.232.68.133 raw.githubusercontent.com" >> /etc/hosts

总结：DockerCompose作用：

基于Compose文件实现集群的快速构建和部署，帮助我们快速部署分布式应用，无需一个个微服务去构建镜像和部署。

#### 利用DockerCompose去部署微服务集群：

eg：将之前学习的alibaba-micro微服务集群利用DcokerCompose部署：

实现思路如下：

1：查看课前资料提供的cloud-demo文件夹，里面已经比那些好了docker-compose文件