# 用docker搭建漏洞靶场 vulhub

# Docker

## 一、Docker是什么

2010年，几个搞IT的年轻人，在美国旧金山成立了一家名叫“dotCloud”的公司

Docker技术诞生之后，并没有引起行业的关注。而dotCloud公司，作为一家小型创业企业，在激烈的竞争之下，也步履维艰。

正当他们快要坚持不下去的时候，脑子里蹦出了“开源”的想法。

什么是“开源”?开源，就是开放源代码。也就是将原来内部保密的程序源代码开放给所有人，然后让大家一起参与进来，贡献代码和意见。

有的软件是一开始就开源的。也有的软件，是混不下去，创造者又不想放弃，所以选择开源。自己养不活，就吃“百家饭”嘛。

2013年3月，dotCloud公司的创始人之一，Docker之父，28岁的Solomon Hykes正式决定，将Docker项目开源。

不开则已，一开惊人。

越来越多的IT工程师发现了Docker的优点，然后蜂拥而至，加入Docker开源社区。

Docker的人气迅速攀升，速度之快，令人瞠目结舌。

开源当月，Docker 0.1版本发布。此后的每一个月，Docker都会发布一个版本。到2014年6月9日，Docker 1.0版本正式发布。

此时的Docker，已经成为行业里人气最火爆的开源技术，没有之一。甚至像Google、微软、Amazon、VMware这样的巨头，都对它青睐有加，表示将全力支持。

Docker火了之后，dotCloud公司干脆把公司名字也改成了Docker Inc.。

Docker和容器技术为什么会这么火爆?说白了，就是因为它“轻”。

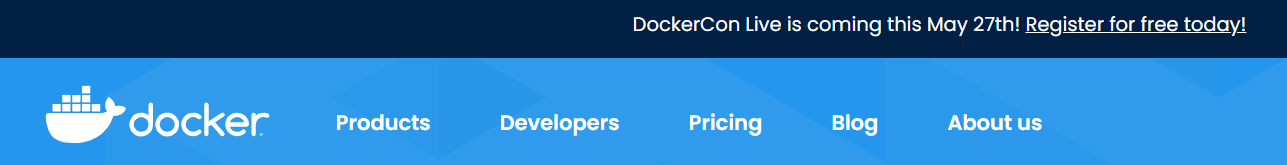
在容器技术之前，业界的网红是虚拟机。虚拟机技术的代表，是VMWare和OpenStack。

虚拟机属于虚拟化技术。而Docker这样的容器技术，也是虚拟化技术，属于轻量级的虚拟化

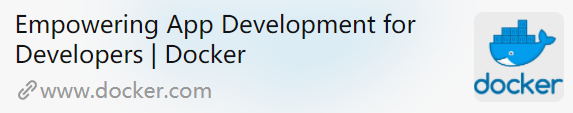
虚拟机虽然可以隔离出很多“子电脑”，但占用空间更大，启动更慢，虚拟机软件可能还要花钱(例如VMWare)。

而容器技术恰好没有这些缺点。它不需要虚拟出整个操作系统，只需要虚拟一个小规模的环境（类似“沙箱”)。









Docker的思想来自于集装箱，集装箱解决了什么问题？在一艘大船上，可以把货物规整的摆放起来。并且各种各样的货物被集装箱标准化了，集装箱和集装箱之间不会互相影响。那么我就不需要专门运送水果的船和专门运送化学品的船了。只要这些货物在集装箱里封装的好好的，那我就可以用一艘大船把他们都运走。

docker就是类似的理念。现在都流行服务器就好比大货轮。docker就是集装箱。

大家需要注意，Docker本身并不是容器，它是创建容器的工具，是应用容器引擎。

想要搞懂Docker，其实看它的两句口号就行。

第一句，是“Build, Ship and Run”。

Docker的第二句口号就是：“Build once，Run anywhere（搭建一次，到处能用）

举个例子：

我来到一片空地，想建个房子，于是我搬石头、砍木头、画图纸，一顿操作，终于把这个房子盖好了。结果，我住了一段时间，想搬到另一片空地去。这时候，按以往的办法，我只能再次搬石头、砍木头、画图纸、盖房子。但是，跑来一个老巫婆，教会我一种魔法。这种魔法，可以把我盖好的房子复制一份，做成“镜像”，放在我的背包里。等我到了另一片空地，就用这个“镜像”，复制一套房子，摆在那边，拎包入住。

Docker技术的三大核心概念，分别是:

·镜像(lmage)

·容器(Container)

·仓库(Repository)

我刚才例子里面，那个放在包里的“镜像”，就是Docker镜像。而我的背包，就是Docker仓库。我在空地上，用魔法造好的房子，就是一个Docker容器。

说白了，这个Docker镜像，是一个特殊的文件系统。它除了提供容器运行时所需的程序、库、资源、配置等文件外，还包含了一些为运行时准备的一些配置参数（例如环境变量)。镜像不包含任何动态数据，其内容在构建之后也不会被改变。

也就是说，每次变出房子，房子是一样的，但生活用品之类的，都是不管的。谁住谁负责添置。

每一个镜像可以变出一种房子。那么，我可以有多个镜像呀!

也就是说，我盖了一个欧式别墅，生成了镜像。另一个哥们可能盖了一个中国四合院，也生成了镜像。还有哥们，盖了一个非洲茅草屋，也生成了镜像。。。

这么一来，我们可以交换镜像，你用我的，我用你的，岂不是很爽?

于是乎，就变成了一个大的公共仓库。

负责对Docker镜像进行管理的，是Docker Registry服务（类似仓库管理员)。

不是任何人建的任何镜像都是合法的。万一有人盖了一个有问题的房子呢?

所以，Docker Registry服务对镜像的管理是非常严格的。

最常使用的Registry公开服务，是官方的Docker Hub，这也是默认的Registry，并拥有大量的高质量的官方镜像。



## 二、Docker的安装

使用下docker才能最明白，docker需要linux内核版本在3.10以上

1. 查看内核的版本

uname -r

1. 升级yum（视情况而定）

yum update -y

1. 重启

reboot

1. 卸载旧版本（如果安装过旧版本的话）

yum remove -y docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-selinux docker-engine-selinux docker-engine

1. 安装需要的软件包

yum install -y yum-utils device-mapper-persistent-data lvm2

1. 由于官网的源太慢了，这里可以使用阿里云Docker Yum源替代：

yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

yum makecache

1. 更新yum包索引

sudo yum makecache fast

1. 安装docker

yum install -y docker

1. 启动docker

systemctl start docker.service

1. Docker设置开机自启动

systemctl enable docker

## 三、Docker的使用

#### 1、docker搭建sqli靶机

练习，用docker搭建sqli靶机

在终端输入 docker search sqli-labs

拉取sqli镜像docker pull docker.io/acgpiano/sqli-labs

运行容器，加载镜像docker run -dit --name sqli-labs -p 8088:80 --rm docker.io/acgpiano/sqli-labs

docker run的参数

OPTIONS说明：

-d: 后台运行容器，并返回容器ID；

-i: 以交互模式运行容器，通常与 -t 同时使用；

-p: 指定端口映射，格式为：主机(宿主)端口:容器端口

-t: 为容器重新分配一个伪输入终端，通常与 -i 同时使用；

--name="nginx-lb": 为容器指定一个名称；

--rm选项，这样在容器退出时就能够自动清理容器内部的文件系统。

4．浏览器输入 [http://IP地址:8088/](http://192.168.137.200:8088/)

容器相关命令：

查看当前正在运行的容器：docker ps -s

查看已经创建的容器：docker ps -a

启动容器名为con\_name的容器  docker start con\_name

停止容器名为con\_name的容器 docker stop con\_name

删除容器名为con\_name的容器docker rm con\_name

重命名一个容器 docker rename old\_name new\_name

进入到docker容器 docker exec -it [容器名/id] bash

-it 交互式终端

bash 运行shell程序

退出容器 exit

Docker rmi 镜像id 删除镜像

# 搭建vulhub

Vulhub是一个面向大众的开源漏洞靶场，无需docker知识，简单执行两条命令即可编译、运行一个完整的漏洞靶场镜像。旨在让漏洞复现变得更加简单，让安全研究者更加专注于漏洞原理本身。

## 下载vulhub 的源码

mkdir /vulhub

#这个git clone 不稳定，建议用下面的zip文件来还原靶场文件

 git clone <https://github.com/vulhub/vulhub.git>

或者下载zip文件

<https://github.com/vulhub/vulhub/archive/master.zip>

#下载unzip工具

yum install -y unzip zip

#解压zip文件

unzip master.zip

#安装到/

mv vulhub-master /

## 2、docker 安装

上方已安装，此处不再赘述

## 3、Dockers compose安装

#设置docker-compose的安装目录

cd /usr/local/bin

#安装docker-compose

curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose && chmod +x /usr/local/bin/docker-compose

#查看docker-compose的版本

docker-compose -version