一、什么是docker

Docker是一个开源的引擎，可以轻松的为任何应用创建一个轻量级的、可移植的、自给自足的容器。开发者在笔记本上编译测试通过的容器可以批量地在生产环境中部署，包括VMs（虚拟机）、bare metal、OpenStack 集群和其他的基础应用平台。

Docker通常用于如下场景：

web应用的自动化打包和发布；

自动化测试和持续集成、发布；

在服务型环境中部署和调整数据库或其他的后台应用；

从头编译或者扩展现有的OpenShift或Cloud Foundry平台来搭建自己的PaaS环境。

入门准备

Docker系统有两个程序：docker服务端和docker客户端。其中docker服务端是一个服务进程，管理着所有的容器。docker客户端则扮演着docker服务端的远程控制器，可以用来控制docker的服务端进程。大部分情况下，docker服务端和客户端运行在一台机器上。

**目标：**

检查docker的版本，这样可以用来确认docker服务在运行并可通过客户端链接。

**提示：**

可以通过在终端输入docker命令来查看所有的参数。

官网的在线模拟器只提供了有限的命令，无法保证所有的命令可以正确执行。

**正确的命令：**

$docker version

搜索可用的镜像

使用docker最简单的方式莫过于从现有的容器镜像开始。Docker官方网站专门有一个页面来存储所有可用的镜像，网址是：index.docker.io。你可以通过浏览这个网页来查找你想要使用的镜像，或者使用命令行的工具来检索。

**目标：**

学会使用命令行的工具来检索名字叫做tutorial的镜像。

**提示：**

命令行的格式为：docker search 镜像名字

**正确的命令：**

$docker search tutorial

使用命令下载镜像

下载镜像的命令非常简单，使用docker pull命令即可。(译者按：docker命令和git有一些类似的地方）。在docker的镜像索引网站上面，镜像都是按照用户名/镜像名的方式来存储的。有一组比较特殊的镜像，比如ubuntu这类基础镜像，经过官方的验证，值得信任，可以直接用镜像名来检索到。

**目标：**

通过docker命令下载tutorial镜像。

**提示：**

执行pull命令的时候要写完整的名字，比如"learn/tutorial"。

**正确的命令：**

$docker pull learn/tutorial

在docker中运行Helloworld

**在docker容器中运行hello world!**

docker容器可以理解为在沙盒中运行的进程。这个沙盒包含了该进程运行所必须的资源，包括文件系统、系统类库、shell 环境等等。但这个沙盒默认是不会运行任何程序的。你需要在沙盒中运行一个进程来启动某一个容器。这个进程是该容器的唯一进程，所以当该进程结束的时候，容器也会完全的停止。

**目标**：

在我们刚刚下载的镜像中输出"hello word"。为了达到这个目的，我们需要在这个容器中运行"echo"命令，输出"hello word"。

**提示：**

docker run命令有两个参数，一个是镜像名，一个是要在镜像中运行的命令。

**正确的命令：**

$docker run learn/tutorial echo "hello word"

在docker中安装程序

下一步我们要做的事情是在容器里面安装一个简单的程序(ping)。我们之前下载的tutorial镜像是基于ubuntu的，所以你可以使用ubuntu的apt-get命令来安装ping程序：apt-get install -y ping。

备注：apt-get 命令执行完毕之后，容器就会停止，但对容器的改动不会丢失。

**目标：**

在learn/tutorial镜像里面安装ping程序。

**提示：**

在执行apt-get 命令的时候，要带上-y参数。如果不指定-y参数的话，apt-get命令会进入交互模式，需要用户输入命令来进行确认，但在docker环境中是无法响应这种交互的。

**正确的命令**：

$docker run learn/tutorial apt-get install -y ping

保存对容器的修改

当你对某一个容器做了修改之后（通过在容器中运行某一个命令），可以把对容器的修改保存下来，这样下次可以从保存后的最新状态运行该容器。docker中保存状态的过程称之为committing，它保存的新旧状态之间的区别，从而产生一个新的版本。

**目标：**

首先使用docker ps -l命令获得安装完ping命令之后容器的id。然后把这个镜像保存为learn/ping。

**提示：**

1. 运行docker commit，可以查看该命令的参数列表。

2. 你需要指定要提交保存容器的ID。(译者按：通过docker ps -l 命令获得)

3. 无需拷贝完整的id，通常来讲最开始的三至四个字母即可区分。（译者按：非常类似git里面的版本号)

**正确的命令：**

$docker commit 698 learn/ping

**运行新的镜像**

ok，到现在为止，你已经建立了一个完整的、自成体系的docker环境，并且安装了ping命令在里面。它可以在任何支持docker环境的系统中运行啦！(译者按：是不是很神奇呢？)让我们来体验一下吧！

**目标：**

在新的镜像中运行ping www.google.com命令。

**提示：**

一定要使用新的镜像名learn/ping来运行ping命令。(译者按：最开始下载的learn/tutorial镜像中是没有ping命令的)

**正确的命令：**

$docker run lean/ping ping www.google.com

检查运行中的镜像

现在你已经运行了一个docker容器，让我们来看下正在运行的容器。

使用docker ps命令可以查看所有正在运行中的容器列表，使用docker inspect命令我们可以查看更详细的关于某一个容器的信息。

**目标：**

查找某一个运行中容器的id，然后使用docker inspect命令查看容器的信息。

**提示：**

可以使用镜像id的前面部分，不需要完整的id。

**正确的命令：**

$ docker inspect efe

发布docker镜像

现在我们已经验证了新镜像可以正常工作，下一步我们可以将其发布到官方的索引网站。还记得我们最开始下载的learn/tutorial镜像吧，我们也可以把我们自己编译的镜像发布到索引页面，一方面可以自己重用，另一方面也可以分享给其他人使用。

**目标**：

把learn/ping镜像发布到docker的index网站。

**提示：**

1. docker images命令可以列出所有安装过的镜像。

2. docker push命令可以将某一个镜像发布到官方网站。

3. 你只能将镜像发布到自己的空间下面。这个模拟器登录的是learn帐号。

**预期的命令：**

$ docker push learn/ping