Docker 入门教程:快速开始

我们在 <u>《一文了解什么是 Docker》</u> 文章中已经介绍了 Docker 是什么,以及为什么需要 Docker 技术。本文将快速介绍一下如何使用 Docker。

安装 Docker

Soarkt技术博客

ttp://www.iteblog.co

Docker 是一个开源的商业产品,支持几乎所有的 Linux 发行版,也支持 Mac 以及 Windows 平台。在各平台上又分为两个版本:免费的社区版(Community Edition,缩写为 CE)和收费的企业版(Enterprise Edition,缩写为 EE),本文下面的介绍都是基于社区版的,并且以 CentOS 为例进行介绍。

这个 文档里面给出了所有支持平台的 Docker

安装方法,已经比较详细了,大家可以根据自己的平台查看相应的安装方法,本文就不再赘述。



或者 \$ docker version

如果输出了服务器端的信息,那么说明 Docker 服务器端正常启动。否则我们可以使用下面命令启动 Docker 服务器:

\$ sudo systemctl start docker

或者 \$ sudo service docker start

当然,我们也可以在开机的时候启动 docker, RHEL, CentOS, Fedora, Ubuntu 16.04 以上版本只需要设置下面命令即可

\$ sudo systemctl enable docker

Ubuntu 14.10 及以下版本使用的是 upstart 来管理重启的服务,而 docker 自带就使用这个命令配置了开机启动 Docker 服务,所以我们不需要额外设置了。更多的说明可以参见 这里。

Docker 核心概念

在介绍下面内容之前,我们先来了解一下 Docker 的核心概念。Docker 有的三大核心概念:

- 镜像(image):镜像是创建 docker 容器的基础, docker 镜像类似于虚拟机镜像,可以将它理解为一个面向 docker 引擎的只读模块,包含文件系统。比如某个镜像包含一个 CentOS 操作系统;或者我们开 发的应用程序等。我们可以从无到有开始创建镜像;也可以基于先有的镜像创建新的镜像 。可以将镜像的内容以及创建的步骤写到一个文本文件里面,这个文件称为 Dockerfile,然后 Docker 可以通过这个文件构建出 Docker 镜像。
- 容器(container):容器就是镜像运行的实例,我们可以使用 Docker 提供的相关命令启动、开始、停止、删除 Docker 容器。
 镜像本身是只读,容器从镜像启动的时候,Docker
 会在镜像的最上层建一个可写层,镜像本身保持不变。可以利用docker
 create命令创建一个容器,创建后的的容器处于停止状态,可以使用docker

start命令来启动它。也可以直接利用docker run命令来直接从镜像启动运行一个容器。docker run = docker creat + docker start。

仓库(repository):存放 docker
镜像的地方称为仓库。仓库分为公有仓库和私有仓库,Docker
Hub(<u>https://hub.docker.com/</u>)是 Docker 公司官方维护的最大公有仓库,Docker 默认的就是这个仓库。当然,出于安全的原因,很多公司自己维护了一个内部的私有仓库,这个仓库只能由公司内部网络下载。Docker 仓库其实和我们熟悉的 Maven 仓库的概念很类似,我们可以将创建好的镜像上传到里面,也可以在里面搜索镜像,下载镜像等。

第一个 Docker 容器

上面我们已经安装好 Docker 了,下面我们来介绍如何使用 Docker。Docker 官方为我们提供了一个非常简单的 Docker 镜像:hello-world(参见 <u>这里</u>), 现在我们来体验一下 Docker 吧。下载 hello-world 镜像:

[iteblog@iteblog.com ~]\$ docker pull library/hello-world Using default tag: latest latest: Pulling from library/hello-world 1b930d010525: Pull complete Digest: sha256:9572f7cdcee8591948c2963463447a53466950b3fc15a247fcad1917ca215a2f Status: Downloaded newer image for hello-world:latest docker.io/library/hello-world:latest

上面命令从默认的 Docker 仓库 (Docker Hub) 下载 library/hello-world 镜像。由于任何人都可以在 Docker Hub 仓库上传镜像,为了区分不同用户的同名镜像,所以下 载镜像的时候需要指定用户名、镜像名称以及 tag,完整格式为 [username]/docker_image_name:tag。由于 Docker 官方提供的镜像文件都放在 library 组里面,它是默认的用户名,可以我们在下载 Docker 官方的镜像时,可以省去 [username],所以上面命令相当于 docker pull hello-world。不指定 tag 的时候将默认使用 latest。

注意:上面命令是从国外的 Docker 的官方仓库下载镜像的,有时候下载速度会很慢,我们往往 需要将默认仓库改成国内的镜像网站,可以参见<u>这篇</u>文章了解如何设置。

接下来我们查看刚刚下载的 Docker 镜像:

[iteblog@iteblog.com ~]\$ docker images hello-world				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	fce289e99eb9	13 months ago	1.84kB

大家应该看到上面命令最后一列的输出,这个 hello-world 镜像大小才 1.84KB!这个镜像的 Dockerfile 其实只有三行代码(参见 这里):

FROM scratch COPY hello / CMD ["/hello"]

下面我们来简单介绍一下这个 Dockerfile 里面的内容:

- FROM scratch:也就是说这个镜像是从零开始构建的,不依赖如何其他镜像;
- COPY hello / : 将 hello 文件拷贝到镜像的根目录,这个 hello 其实是实现编译好的可执行文件;
- CMD ["/hello"]:容器启动时执行 /hello。

好了,我们已经从 Docker 仓库下载了 hello-world 镜像,现在我们来运行这个镜像:

[iteblog@iteblog.com ~]\$ docker run hello-world

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

- 1. The Docker client contacted the Docker daemon.
- 2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)
- 3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
- 4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with: \$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID: https://hub.docker.com/

For more examples and ideas, visit: https://docs.docker.com/get-started/

可以看出,启动 hello-world 镜像会执行 /hello

文件,然后在终端打印出上面的一些信息,最后容器自动终止。

好了,本文就介绍到这里。通过本文我们已经了解到 Docker 最基本的概念和使用,后面的文章我将给大家介绍 Docker 的常用命令以及如何自己制作一个 Docker 镜像。

> 本博客文章除特别声明,全部都是原创! 原创文章版权归过往记忆大数据(<u>过往记忆</u>)所有,未经许可不得转载。 本文链接:【】()