6.运行一个简单的容器

如果要在Kubernetes中运行应用程序，需要先将它们打包到容器镜像中。本文以Docker为例来进行具体讲解。

1.安装Docker

首先我们需要在Linux平台（本文以Centos7为例）上安装Docker软件。

- 关闭SELINUX服务

  SELINUX是CentOS系统捆绑的安全服务程序，因为安全策略太过于严格，所以建议关闭这项服务。

  修改/etc/selinux/config文件，设置SELINUX=disabled

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

- 查看系统内核版本

  uname –a

```
[root@dhr-demo ~]# uname -a
Linux dhr-demo 3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- 更新yum包

  yum update

- 安装依赖软件包

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

- 设置阿里yum源

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

- 安装需要的docker版本

  yum install docker-ce-18.03.1.ce

- 启动并加入开机启动

  systemctl start docker

  systemctl enable docker

2.使用Docker运行hello-world容器

安装完成之后，可以通过Docker客户端运行各种Docker命令。比如，我们可以拉取一个hello-world镜像，然后执行运行。也可以执行通过docker run命令执行hello-world镜像，该命令会自动下载并执行镜像。

```
[root@dhr-demo ~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:8c5aeeb6a5f3ba4883347d3747a7249f491766ca1caa47e5da5dfcf6b9b717c0
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

这看起来可能不太吸引人，但是想象一下，当你把hello-world换成你自己的应用程序时，通过一个简单的命令就能直接下载并执行，完全不需要安装或者其他操作，你还会觉得不够炫酷吗？为了演示，我选择了一个很简单的hello-world应用，但是它也可能是有许多依赖的复杂应用。无论多么复杂，设置和运行应用程序的整个流程是完全一样的。另一个比较重要的点是，在容器中运行的应用与主机上其他进程是完全隔离的。

这背后发生了什么

当我们运行hello-world镜像时，会看到终端显示了一段话，如下图中红框中的内容：

```
[root@dhr-demo ~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:8c5aeeb6a5f3ba4883347d3747a7249f491766ca1caa47e5da5dfcf6b9b717c0
Status: Downloaded newer image for hello-world:latest

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.
```

这段文字其实已经清晰地解释了运行这个镜像背后的原理。

Docker分为Docker Client和Docker Server，在我们启动Docker之前，通过执行docker version命令只能看到Docker Client的信息，如下图：

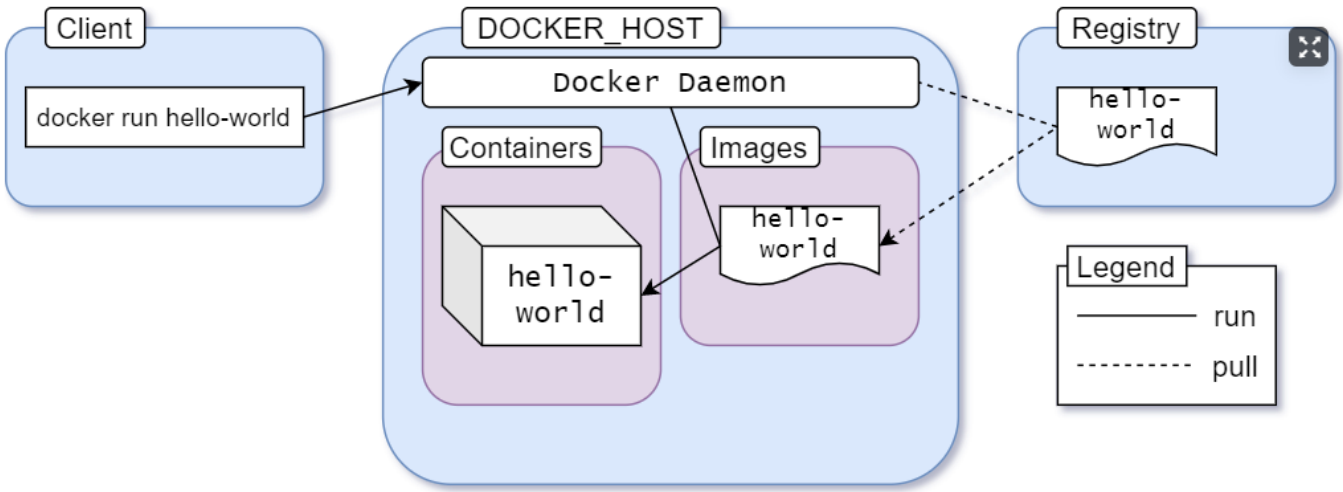
```
[root@dhr-demo ~]# docker version
Client:
 Version:      18.03.1-ce
 API version:  1.37
 Go version:   go1.9.5
 Git commit:   9ee9f40
 Built:        Thu Apr 26 07:20:16 2018
 OS/Arch:      linux/amd64
 Experimental: false
 Orchestrator: swarm
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
```

启动Docker之后，还可以看到Docker Server的信息，如下图：

```
[root@dhr-demo ~]# docker version
Client:
 Version:      18.03.1-ce
 API version:  1.37
 Go version:   go1.9.5
 Git commit:   9ee9f40
 Built:        Thu Apr 26 07:20:16 2018
 OS/Arch:      linux/amd64
 Experimental: false
 Orchestrator: swarm

Server:
 Engine:
  Version:      18.03.1-ce
  API version:  1.37 (minimum version 1.12)
  Go version:   go1.9.5
  Git commit:   9ee9f40
  Built:        Thu Apr 26 07:23:58 2018
  OS/Arch:      linux/amd64
  Experimental: false
```

Docker Client执行docker run命令时，会与Docker daemon通信，该进程会去检查本机是否已经下载了hello-world镜像。如果没有，它会从Docker Hub镜像仓库拉取该镜像。镜像被下载到本机后，Docker daemon会基于这个镜像创建一个容器，然后容器内会运行可执行文件输出内容。Doceker daemon会将输出内容发送给Docker客户端，最终，内容会被显示到终端显示器。



运行容器镜像

运行其他容器镜像跟运行hello-world镜像的方式一样。通常运行镜像的时候我们都不需要指定额外的命令。因为镜像中一般都包含需要执行的命令，但是我们也可以根据情况进行覆盖。可以在http://hub.docker.com/或其他镜像中心上找到官方的或者用户公开的镜像。运行镜像的命令如下：

$ docker run <image>

镜像的版本

基本上所有的软件都会更新，因此一个软件程序一般都有多个版本。Docker支持同一个镜像有个多个版本。每个版本必须有一个唯一的标签（tag）。当使用一个镜像没有显示地指定tag名时，Docker会使用默认的*latest* tag。如果要运行一个不同版本的镜像，可以通过指定tag名的方式，如下：

$ docker run <image>:<tag>

如：docker run ubuntu:14.04