



链滴

# 如何使用 docker 部署一个 beego 项目

作者: [xhaoxiong](#)

原文链接: <https://ld246.com/article/1526210600840>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

#### 理解 Docker

<p>Docker 帮助你为应用程序创建一个单独的可部署单元。这个单元，也被称为容器，包含该应用程序需要的所有东西。它包括代码（或者二进制文件）、runtime（运行环境）、系统工具箱系统库。所有必需的资源打包成一个单元将确保无论应用程序部署到哪里都有完全相同的环境。这也有助于维护一个完全相同的开发和生产配置，这在以前是很难追踪的。</p>

<p>一旦开始，容器的创建和部署将自动完成。它消除了一大类问题。这些问题主要是由于文件没有步或者开发和生产环境之间的差异导致的。Docker 帮助解决了这些问题。</p>

#### 相比于虚拟机的优势

<p>容器提供了与虚拟机相似的资源分配和隔离优势。然而，相同之处仅此而已。</p>

<p>一个虚拟机需要它自己的客户操作系统而容器共享主机操作系统的内核。这意味着容器更加轻量且需要更少的资源。从本质上讲，一个虚拟机是操作系统中的一个操作系统。而另一方面的容器则更是操作系统中的其它应用程序。基本上，容器需要的资源（内存、磁盘空间等等）比虚拟机少很多，且具有比虚拟机快很多的启动时间。</p>

#### Docker 在开发阶段的优势

在开发中使用 Docker 的优势包括：



- 一个用于所有团队成员的标准开发环境

- 更新的依赖性集中化以及在任何地方都能使用相同的容器

- 在开发和生产中完全相同的环境

- 修复了可能只会出现在生产环境中的潜在问题

[准备阶段可参考黑客派中这篇文章](https://ld246.com/forward?goto=https%3A%2F%2Fhacpai.com%2Farticle%2F125961874479)

##### 开始准备:



- 安装好 `go` 环境和 `docker` 环境的 `centos7.0` 服务器一台

- 在 `/root` 目录(或者其他目录)下有 go 工作目录 (我的以 `/root/go/src` 为例)

- 在本机上安装好 `bee` 工具(便于调试)

- 在 `src` 目录下面 `bee new test_docker`

## 配置 Docker 用于开发:



- 创建 Dockerfile

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">在root目录下mkdir docker test 在此目录下创建Dockerfile
```

将Dockerfile放入所对应需要部署的文件夹中以便于将对应的文件加入到docker中)

```
</span></span></code></pre>
```

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">FROM golang:1.9.1
```

</span> </span> <span class="highlight-line"> <span class="highlight-cl">

```
# Install beego on the bee dev tool*
```

```

RUN go get github.com/astaxie/beego && go get github.com/beego/beego

```

</span> </span> <span class="highlight-line"> <span class="highlight-cl">

```
# Expose the application on port 8080
```

EXPOSE 8080

</span> </span> <span class="highlight-line"> <span class="highlight-cl">

```
</span></span><span class="highlight-line"><span class="highlight-cl"># Set the entry po
```

nt of the container to the bee command that runs the

```
</span></span><span class="highlight-line"><span class="highlight-cl"># application and  
watches for changes
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">CMD ["bee", "run"
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span></code></pre>
```

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight  
cl">FROM golang:1.9.1
```

```
</span></span></code></pre>
```

<p>Go 的官方映像文件作为基础映像。该映像文件预安装了 <code>Go 1.9.1</code> 该映像已把 <code>\$GOPATH</code> 的值设置到了 <code>/go</code>。所有安装在 <code>/go/src<code> 中的包将能够被 go 命令访问。</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight  
cl">RUN go get github.com/astaxie/beego &amp;&amp; go get github.com/beego/bee
```

```
</span></span></code></pre>
```

<p>安装 <code>beego</code> 包和 <code>bee</code> 工具。<code>beego</code> 包在应用程序中使用。<code>bee</code> 工具用语在开发中再现地重新加载我们的代码。</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight  
cl">EXPOSE 8080
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span></code></pre>
```

<p>在开发主机上利用容器为应用程序开放 8080 端口</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight  
cl">CMD ["bee", "run"]
```

```
</span></span></code></pre>
```

<p>使用 <code>bee</code> 命令启动应用程序的在线重新加载</p>

```
<ul>
```

```
<li>构建 image</li>
```

```
</ul>
```

<p>进入创建好的 docker\_test 文件夹,运行如下的命令来创建 image(注意后面的 <code>.</code></p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight  
cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">docker build -t tes  
-image .
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">-t 是指tag的意思  
建了一个叫做test-image的镜像
```

```
</span></span></code></pre>
```

<p>为了查看你的系统上的 image 列表, 运行如下的命令: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight  
cl">docker images
```

```
</span></span></code></pre>
```

<p>这行该命令将输出与以下类似的内容: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight  
cl">REPOSITORY TAG IMAGE ID CREATED SIZE
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">test-image latest  
8d53aa0dd0cb 31 seconds ago 784.7 MB
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">golang 1.9.1  
22a6ecf1f7cc 5 days ago 743.9 MB
```

```
</span></span></code></pre>
```

<p>注意 image 的确切名字和编号可能不同, 但是, 你应该至少看到列表中有 <code>golang</code> 和 <code>test-image</code> image</p>

如果发现在尝试过程中 `build` 了多余的无效的镜像，可以尝试使用

```
docker rmi [IMAGE ID](上图中的IMAGE ID)
```

使用 `rmi` 移除对应的 D 的镜像

- 运行容器

一旦 `test-image` 已经完成，你可以使用以下的命令启动一个容器：

```
docker run -it --name test-instance -p 8080:8080 -v /root/go/src/docker_test:/go/src/docker_test -w /go/src/docker_test test-image
```

让我们分析一下上面的命令来看看它做了什么。

- `docker run` 命令用于从一个 image 上启动一个容器
- `-it` 标签以交互的方式启动容器
- `--rm` 标签在容器关闭后将会将其清除
- `--name test-instance` 将容器命名为 `test-instance`
- `-p 8080:8080` 标签允许通过 `8080` 端口访问该容器
- `-v /root/go/src/docker_test:/go/src/docker_test` 更复杂一些。它将主机的 `/root/go/src/docker_test` 映射到容器中的 `/go/src/docker_test` 这将使得开发文件在容器的内部和外部都可以访问。
- `test-image` 部分声明了用于容器的 `image`。

运行完后你会发现，你的 beego 项目运行起来了此时是处于调试 `ctrl+c` 就直停了，重新再执行上述的命令的时候会发现说容器已经存在

解决方法：

```
docker ps -a
```

查看容器运行状态

- `docker rm [容id]` 直接删除容器 然后再执行上述命令运行
- 命令中加 `-` 参数 后台运行 [我没有试成功]
- 或者跑起来 `ctrl+p+q`
- 或者在命令后 空格+`&` 挂起
- 等等

最后我们访问 `ip+端口` 应该是可以访问到我们运行起来的 web 应用

此时我们就用 `docker` 部署了一个开发模式下的 web 应用

<https://ld246.com/forward?goto=http%3A%2F%2Fdockone.io%2Farticle%2F129> 以上参考《docker 构建一个 go web 应用》

由于在 <https://ld246.com/forward?goto=https%3A%2F%2Fsemaphoreci.com%2> Semaphore 一直没有注册成功用户名一直被占用 QAQ 后面的自动化部署就没做了。

于是乎我马上将自己以前写的一个 web 应用用 docker 开始部署。

首先我将 Dockerfile 放在了 beego 项目与 main.go 同级下 由于项目中用了 vendor 于是 Dock

rfile 修改如下

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> FROM golang:1.9
1
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> # Install beego an
the bee dev tool*
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> RUN go get githu
.com/astaxie/beego & & go get github.com/beego/bee
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> RUN go get -u git
ub.com/kardianos/govendor
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ADD vendor /TlfS
aw
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> RUN govendor ge
-v github.com/jinzhu/inflection
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> RUN govendor ge
-v github.com/go-sql-driver/mysql
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> RUN govendor ge
-v github.com/iqysf/gorm
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> RUN govendor ge
-v gopkg.in/chanxuehong/wechat.v2
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> # Expose the appl
ication on port 8080*
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> EXPOSE 8087
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> # Set the entry po
nt of the container to the bee command that runs the*
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> # application and
watches for changes*
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> CMD ["bee", "run"

</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> </code> </pre>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ADD vendor /TlfShaw 这是将主机中vendor文件夹添加到beego项目中
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> </code> </pre>
<p>关于数据库 docker 中连接主机数据库直接用 ip 远程连接的方式了<br>
docker 还有许多学习的地方</p>
```