Docker Compose

安装Docker Compose(ubuntu16.04)

现在你已经安装了Docker，让我们继续安装Docker Compose。首先，安装python-pip为先决条件：

* sudo apt-get -y install python-pip

那么你可以安装Docker Compose：

* sudo pip install docker-compose

验证、查看版本

* docker-compose --verion

步骤3 - 使用Docker组合运行容器

公共Docker注册表Docker Hub包含一个简单的*Hello World*图像。现在我们已经安装了Docker Compose，让我们用这个很简单的例子来测试它。

首先，为我们的YAML文件创建一个目录：

* mkdir hello-world

然后改成目录：

* cd hello-world

现在使用您喜欢的文本编辑器创建YAML文件（我们将使用nano）：

* nano docker-compose.yml

将以下内容放入文件中，保存文件，并退出文本编辑器：

泊坞窗，compose.yml

my-test:

image: hello-world

第一行将用作容器名称的一部分。第二行指定要用于创建容器的图像。该映像将从Docker Hub官方存储库官方下载。

仍然在~/hello-world目录中，执行以下命令创建容器：

* docker-compose up

输出应从以下开始：

Output of docker-compose up

Creating helloworld\_my-test\_1...

Attaching to helloworld\_my-test\_1

my-test\_1 |

my-test\_1 | Hello from Docker.

my-test\_1 | This message shows that your installation appears to be working correctly.

my-test\_1 |

输出然后解释了Docker正在做什么：

1. Docker客户端联系了Docker守护程序。
2. Docker守护进程从Docker Hub中拉出了“hello-world”的图像。
3. Docker守护程序从该映像创建了一个新的容器，该容器运行生成您当前正在读取的输出的可执行文件。
4. Docker守护程序将该输出流传输到Docker客户端，该客户端将其发送到您的终端。

如果过程不自行退出，请按CTRL-C。

这个简单的测试没有显示Docker Compose的主要优点之一 - 能够同时将一组Docker容器上下移动。该[如何与码头工人撰写Ubuntu上安装WordPress和phpMyAdmin的14.04](https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-and-phpmyadmin-with-docker-compose-on-ubuntu-14-04)文章展示了如何使用泊坞窗撰写运行三个容器作为一个应用程序组。

步骤4 - 学习Docker撰写命令

我们来看看docker-compose工具支持的命令。

该docker-compose命令以每个目录为基础。您可以在一台机器上运行多组Docker容器，只需为每个容器创建一个目录，并docker-compose.yml在其目录中为每个容器一个文件。

到目前为止，我们已经docker-compose up自行运行，并将CTRL-C其关闭。这允许调试消息显示在终端窗口中。这不是理想的，但是当你在生产中运行时，你会想要docker-compose更像一个服务。执行此操作的一个简单方法是-d在您up的会话中添加选项：

* docker-compose up -d

docker-compose 现在将分叉到背景。

要显示您的Docker容器组（同时停止并正在运行），请使用以下命令：

* docker-compose ps

例如，以下显示helloworld\_my-test\_1容器已停止：

Output of `docker-compose ps`

Name Command State Ports

-----------------------------------------------

helloworld\_my-test\_1 /hello Exit 0

运行容器将显示Up状态：

Output of `docker-compose ps`

Name Command State Ports

---------------------------------------------------------------

nginx\_nginx\_1 nginx -g daemon off; Up 443/tcp, 80/tcp

要停止应用程序组的所有正在运行的Docker容器，请在与docker-compose.yml用于启动Docker组的文件相同的目录中执行以下命令：

* docker-compose stop

**注意：** docker-compose kill如果您需要更有力地关闭事件，也可以使用。

在某些情况下，Docker容器将其旧信息存储在内部卷中。如果要从头开始，可以使用该rm命令完全删除组成容器组的所有容器：

* docker-compose rm

如果您尝试从包含Docker容器和.yml文件的目录以外的目录中的任何这些命令，它将会抱怨并且不显示您的容器：

Output from wrong directory

Can't find a suitable configuration file in this directory or any parent. Are you in the right directory?

Supported filenames: docker-compose.yml, docker-compose.yaml, fig.yml, fig.yaml

步骤5 - 访问Docker容器文件系统（可选）

如果您需要在容器内的命令提示符下工作，可以使用该 docker exec命令。

在*世界，你好！*示例在运行后退出，因此我们需要启动一个将继续运行的容器，以便我们可以使用docker exec该容器访问该容器的文件系统。我们来看看Docker Hub 的[Nginx镜像](https://hub.docker.com/_/nginx/)。

为它创建一个新的目录并更改：

* mkdir ~/nginx && cd $\_

docker-compose.yml在我们的新目录中创建一个文件：

* nano docker-compose.yml

并粘贴以下内容：

〜/ nginx的/搬运工-compose.yml

nginx:

image: nginx

保存文件并退出。我们只需要使用以下命令启动Nginx容器作为后台进程：

* docker-compose up -d

将下载Nginx映像，然后容器将在后台启动。

现在我们需要CONTAINER ID这个容器。所有正在运行的容器的列表：

* docker ps

您将看到类似于以下内容的内容：

Output of `docker ps`

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

e90e12f70418 nginx "nginx -g 'daemon off" 6 minutes ago Up 5 minutes 80/tcp, 443/tcp nginx\_nginx\_1

**注意：**仅使用命令列出*运行*容器docker ps。

如果我们想要更改这个容器内的文件系统，我们将使用它的ID（在本例中e90e12f70418），并用于docker exec在容器内启动一个shell：

* docker exec -it e90e12f70418 /bin/bash

该-t选项打开一个终端，该-i选项使其成为交互式。这些/bin/bash选项为运行容器打开一个bash shell。确保使用您的容器的ID。

您会看到容器的bash提示符类似于：

root@e90e12f70418:/#

从这里，您可以从命令提示符工作。但请记住，除非您在一个保存为数据卷一部分的目录中，否则，一旦容器重新启动，您的更改就会消失。另一个警告是，大多数Docker图像都是使用非常小的Linux安装创建的，因此您使用的一些命令行实用程序和工具可能不存在。