

基因容器

常见问题

文档版本

01

发布日期

2020-07-08



版权所有 © 华为技术有限公司 2020。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目录

1 如何制作 Docker 镜像?	1
2 如何上传 Docker 镜像至容器镜像服务?	6
3 SDK 中文支持问题如何处理?	9

1 如何制作 Docker 镜像?

自己制作Docker镜像，主要有两种方法：

- 制作快照方式获得镜像（偶尔制作的镜像）：在基础镜像上，比如Ubuntu，先登录镜像系统并安装Docker软件，然后整体制作快照。
- Dockerfile方式构建镜像（经常更新的镜像）：将软件安装的流程写成DockerFile，使用Docker build构建成Docker镜像。

方法一：制作快照方式获得镜像

如果后续镜像没有变化，可采用[方法一](#)制作镜像。



具体操作如下：

1. 找一台主机，安装Docker软件。
2. 启动一个空白的基础容器，并进入容器。
例如：启动一个CentOS的容器。
docker run -it centos
3. 执行安装任务。

```
yum install XXX
git clone https://github.com/lh3/bwa.git
cd bwa;make
```

 说明

请预先安装好Git，并检查本机是否有ssh key设置。

4. 输入exit退出容器。
5. 制作快照。

```
docker commit -m "xx" -a "tsj" container-id tsj/image:tag
```

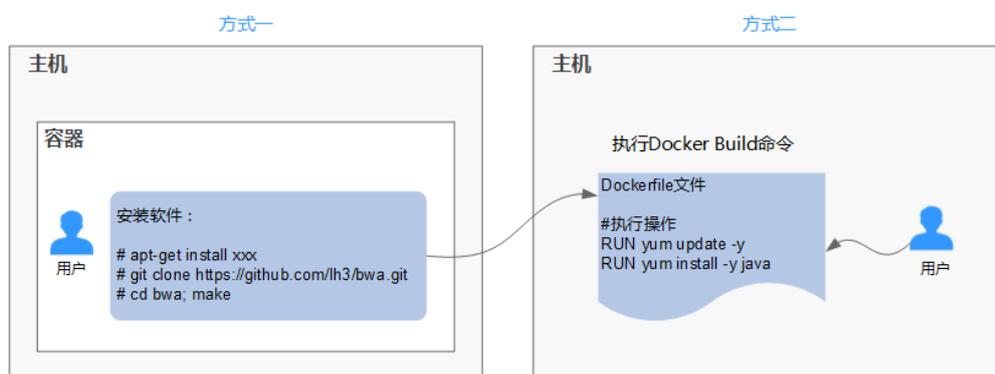
- -a : 提交的镜像作者。
- container-id : [操作步骤2](#)中的容器id。可以使用 `docker ps -a` 查询得到容器id。
- -m : 提交时的说明文字。
- tsj/image:tag : 仓库名/镜像名:TAG名。

6. 执行docker images可以查看到制作完成的Docker镜像。

方法二：使用 Dockerfile 方式构建

如果后续镜像经常变更（例如某个软件更新版本），则需要采用[方法二](#)制作镜像。若仍采用[方法一](#)制作镜像，则每次变更都需要重新执行[方法一](#)的命令，操作过程比较繁琐，所以建议使用自动化制作镜像的方法。

其实就是将[方法一](#)制作镜像的方法，用文件方式写出来（文件名为DockerFile）。然后执行：`docker build -t tsj/image:tag.`命令（命令中“.”表示DockerFile文件的路径），自动完成镜像制作。



简单的DockerFile示例:

 说明

如果依赖外部网络，请搭建网络环境，并保证网络可用。

```
#Version 1.0.1
FROM centos:latest
MAINTAINER ***u "****u@huaweicloud.com"

#设置root用户为后续命令的执行者
USER root

#执行操作
RUN yum update -y
```

```
RUN yum install -y java

#使用&&拼接命令
RUN touch test.txt && echo "abc" >>abc.txt

#对外暴露端口
EXPOSE 80 8080 1038

#添加网络文件
ADD https://www.baidu.com/img/bd_logo1.png /opt/

#设置环境变量
ENV WEBAPP_PORT=9090

#设置工作目录
WORKDIR /opt/

#设置启动命令
ENTRYPOINT ["ls"]

#设置启动参数
CMD ["-a", "-l"]

#设置卷
VOLUME ["/data", "/var/www"]

#设置子镜像的触发操作
ONBUILD ADD . /app/src
ONBUILD RUN echo "on build excuted" >> onbuild.txt
```

详细的操作步骤可以参考：《容器镜像服务 最佳实践》。

DockerFile 基本语法

- FROM：
指定待扩展的父级镜像（基础镜像）。除注释之外，文件开头必须是一个FROM指令，后面的指令便在这个父级镜像的环境中运行，直到遇到下一个FROM指令。通过添加多个FROM命令，可以在同一个Dockerefile文件中创建多个镜像。
- MAINTAINER：
声明创建镜像的作者信息：用户名、邮箱，非必须参数。
- RUN：
修改镜像的命令，常用来安装库、安装程序以及配置程序。一条RUN指令执行完毕后，会在当前镜像上创建一个新的镜像层，接下来对的指令会在新的镜像上继续执行。RUN 语句有两种形式：
 - **RUN yum update**：在/bin/sh路径中执行的指令命令。
 - **RUN ["yum", "update"]**：直接使用系统调用exec来执行。
 - **RUN yum update && yum install nginx**：使用&&符号将多条命令连接在同一条RUN语句中。
- EXPOSE：
指明容器内进程对外开放的端口，多个端口之间使用空格隔开。
运行容器时，通过设置参数-P（大写）即可将EXPOSE里所指定的端口映射到主机上其他的随机端口，其他容器或主机可以通过映射后的端口与此容器通信。
您也可以通过设置参数-p（小写）将Dockerfile中EXPOSE中没有列出的端口设置成公开。
- ADD：
向新镜像中添加文件，这个文件可以是一个主机文件，也可以是一个网络文件，也可以使一个文件夹。

- 第一个参数：源文件（夹）。
 - 如果是相对路径，必须是相对于Dockerfile所在目录的相对路径。
 - 如果是URL，会将文件先下载下来，然后再添加到镜像里。
- 第二个参数：目标路径。
 - 如果源文件是主机上的zip或者tar形式的压缩文件，Docker会先解压缩，然后将文件添加到镜像的指定位置。
 - 如果源文件是一个通过URL指定的网络压缩文件，则不会解压。
- VOLUME：

在镜像里创建一个指定路径(文件或文件夹)的挂载点，这个容器可以来自主机或者其它容器。多个容器可以通过同一个挂载点共享数据，即便其中一个容器已经停止，挂载点也仍可以访问。
- WORKDIR：

为接下来执行的指令指定一个新的工作目录，这个目录可以是绝对目录，也可以是相对目录。根据需要，WORKDIR可以被多次指定。当启动一个容器时，最后一条WORKDIR指令所指的目录将作为容器运行的当前工作目录。
- ENV：

设置容器运行的环境变量。在运行容器的时候，通过设置-e参数可以修改这个环境变量值，也可以添加新的环境变量。

例如：

```
docker run -e WEBAPP_PORT=8000 -e WEBAPP_HOST=www.example.com ...
```
- CMD：

用来设置启动容器时默认运行的命令。
- ENTRYPOINT：

用来指定容器启动时的默认运行的命令。区别在于：运行容器时添加在镜像之后的参数，对ENTRYPOINT是拼接，CMD是覆盖。

 - 若在DockerFile中指定了容器启动时的默认运行命令为ls -l，则运行容器时默认启动命令为 ls -l，例如：
 - **ENTRYPOINT ["ls", "-l"]**：指定容器启动时的程序及参数为 ls -l 。
 - **docker run centos**：当运行centos容器时，默认执行的命令是docker run centos ls -l
 - **docker run centos -a**：当运行centos容器时拼接了-a参数，则默认运行的命令是docker run centos ls -l -a
 - 若在DockerFile中指定了指定了容器启动时的默认运行命令为--entrypoint，则在运行容器时若需要替换默认运行命令，可以通过添加--entrypoint参数来替换Dockerfile中的指定。例如：

```
docker run gutianlangyu/test --entrypoint echo "hello world"
```
- USER：

为容器的运行及RUN、CMD、ENTRYPOINT等指令的运行指定用户或UID。
- ONBUILD：

触发器指令。构建镜像时，Docker的镜像构建器会将所有的ONBUILD指令指定的命令保存到镜像的元数据中，这些命令在当前镜像的构建过程中并不会执行。只有新的镜像使用FROM指令指定父镜像为当前镜像时，才会触发执行。

使用FROM以这个Dockerfile构建出的镜像为父镜像，构建子镜像时：

ONBUILD ADD . /app/src：自动执行**ADD . /app/src**

2 如何上传 Docker 镜像至容器镜像服务?

本文以一个2048应用为例，讲述根据该应用编写Dockerfile文件构建镜像并上传至容器镜像服务的操作。您可以编写一个Dockerfile文件，以[alpine:3.7](#)为基础镜像，来构建一个2048容器镜像。

前提条件

- 已安装Docker，如未安装可参见[安装Docker](#)。
- 已获取2048应用，并将该镜像下载至本地。

创建组织

组织用于隔离镜像，并为租户下用户指定不同的权限（读取、编辑、管理）。

- 读取：只能下载镜像，不能上传。
- 编辑：下载镜像、上传镜像、编辑镜像属性以及创建触发器。
- 管理：下载镜像、上传镜像、删除镜像或版本、编辑镜像属性、添加授权、添加触发器以及共享镜像。

步骤1 登录容器镜像服务控制台。

步骤2 在左侧菜单栏选择“组织管理”，单击右侧“创建组织”，在弹出的页面中填写“组织名称”，然后单击“确定”。

图 2-1 创建组织



----结束

客户端上传镜像

docker客户端上传镜像，是指使用docker命令将镜像上传到容器镜像服务的镜像仓库。

本章节以**2048-demo:v1**镜像为例，介绍如何上传镜像。上传成功后，在“我的镜像”中显示已上传成功的镜像。

须知

- 使用客户端上传镜像，镜像的每个layer大小不能超过10G。
- 上传镜像的Docker客户端版本必须为1.11.2及以上。

步骤1 连接容器镜像服务。

1. 登录容器镜像服务控制台。
2. 在左侧菜单栏选择“我的镜像”，单击右侧“客户端上传”，在弹出的页面中单击“生成临时的docker login指令”，单击  复制docker login指令。docker login指令末尾的域名即为当前镜像仓库地址，记录该地址。

图 2-2 获取 docker login 指令

Step 2. 获取登录Docker访问权限，并复制到节点上执行

请点击 [生成临时docker login指令](#) ([获取长期有效docker login指令](#))

📖 说明

此处获取的docker login指令有效期为16小时，若需要长期有效的docker login指令，请参见[获取长期有效docker login指令](#)。

3. 在安装Docker的机器中执行上一步复制的docker login指令。
登录成功会显示“login succeeded”。

步骤2 在安装docker的机器给2048-demo:v1镜像打标签。

```
docker tag [镜像名称:版本名称] [镜像仓库地址]/[组织名称]/[镜像名称:版本名称]
```

📖 说明

镜像名称不支持多级目录格式，例如：镜像名称可命名为“2048-demo”，不可命名为“2048-demo/test”。

样例如下：

```
docker tag 2048-demo:v1 swr.cn-north-1.myhuaweicloud.com/group/2048-demo:v1
```

其中：

- swr.cn-north-1.myhuaweicloud.com为容器镜像服务的镜像仓库地址。
- group为组织名称，如果该组织还没有创建，容器镜像服务会根据组织名称自动创建一个组织。
- 2048-demo:v1 为镜像名称和版本号。

步骤3 上传镜像至镜像仓库。

```
docker push [镜像仓库地址]/[组织名称]/[镜像名称:版本名称]
```

样例如下：

```
docker push swr.cn-north-1.myhuaweicloud.com/group/2048-demo:v1
```

终端显示如下信息，表明push镜像成功。

```
6d6b9812c8ae: Pushed
695da0025de6: Pushed
fe4c16cbf7a4: Pushed
v1: digest: sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63 size: 948
```

返回系统，在“我的镜像”页面，执行刷新操作后可查看到对应的镜像信息。

----结束

3 SDK 中文支持问题如何处理?

若调用SDK接口时涉及到部分字段包含中文后报错的问题，如description字段，请在python文件头部添加“# -*- coding: utf-8 -*-”，具体做法请参考[Python使用UTF-8编码](#)。