Docker、Docker-Compose使用说明文档

1. Docker安装

环境：CentOS内核版本高于 3.10

1. 通过 **uname -r**命令查看你当前的内核版本

uname -r

1. 使用 root 权限登录 Centos。确保 yum 包更新到最新。

sudo yum update

1. 卸载旧版本(如果安装过旧版本的话)

sudo yum remove docker docker-common docker-selinux docker-engine

1. 安装需要的软件包， yum-util 提供yum-config-manager功能，另外两个是devicemapper驱动依赖的

sudo yum install -y yum-utils device-mapper-persistent-data lvm2

1. 设置yum源

sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

1. 可以查看所有仓库中所有docker版本，并选择特定版本安装

yum list docker-ce --showduplicates | sort –r

1. 安装docker或安装指定版本

sudo yum install docker-ce

1. 启动并加入开机启动

sudo systemctl start docker

sudo systemctl enable docker

1. 验证安装是否成功(有client和service两部分表示docker安装启动都成功了)

docker version

1. 配置镜像加速

在 /etc/docker/daemon.json 中添加一下内容

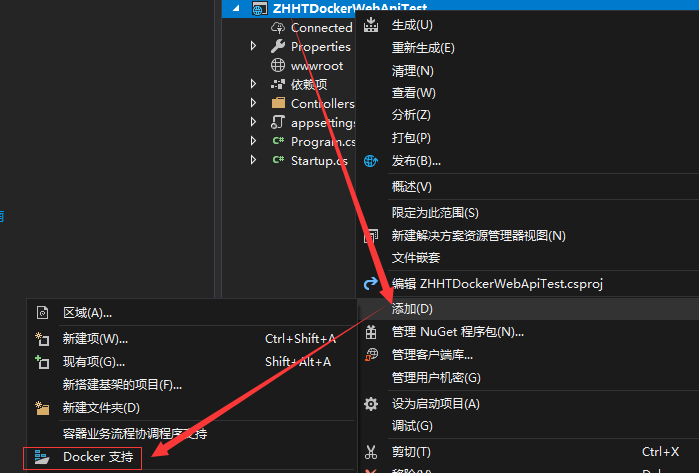
{

"registry-mirrors": ["https://upcvsxht.mirror.aliyuncs.com"],//使用阿里云镜像加速

"live-restore": true//服务器重启，自动运行容器

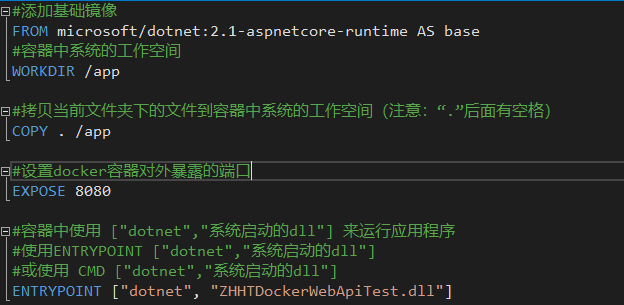
}

1. Webapi项目编写、启用Docker支持、Dockerfile文件说明
2. 新建一个.NetCore WebApi项目：ZHHTDockerWebApiTest
3. 启用Docker支持



启用Docker支持后，会自动生成Dockerfile文件

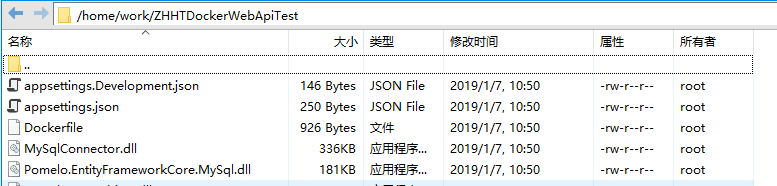
1. Dockerfile文件说明



Console项目的基础镜像为： microsoft/dotnet:2.1-runtime AS base

要暴露几个端口，就再写几个EXPOSE:8081

1. 生成镜像（在CentOS中）
2. 发布项目，并拷贝到CentOS（已经安装了Docker）中，例如：



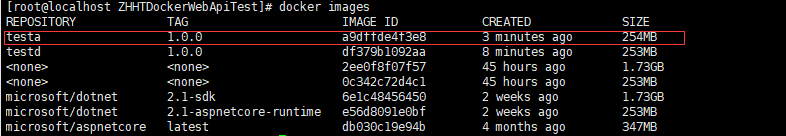
1. 生成镜像

（1）进入程序目录，执行如下命令，生成镜像：

docker build –t testa:1.0.0 .注意镜像名称后边的 .  （testa：1.0.0是标识镜像名称和版本信息）

此时将下载Asp.Net Core 2.1-aspnetcore-runtime 基础镜像，并运行Dockerfile中的后续指令

（2）运行docker images,查看镜像信息：



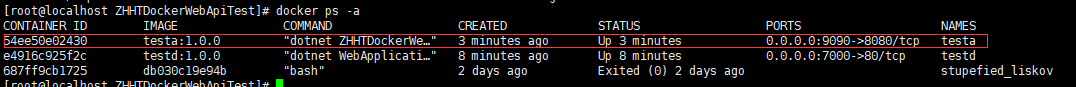
如上，表示镜像创建成功。

1. 创建并启动容器、本机测试
2. 执行创建并运行命令

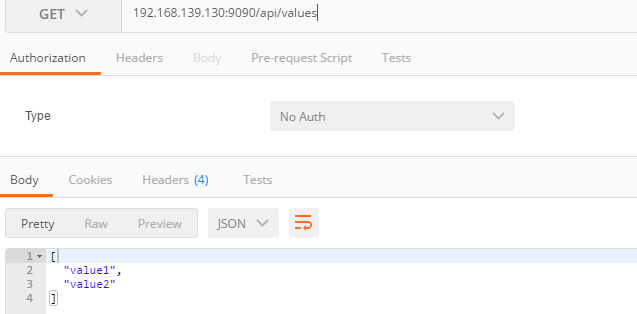
docker run --name testa -d -p 9090:8080 -v --restart=always /home/work/ZHHTDockerWebApiTest:/app testa:1.0.0

注意此处-v，将文件挂载到服务器宿主机上，便于后续更改时，可以直接重启生效。-p 9090:8080 是将容器中的8080端口映射到服务器端口9090

当成功执行之后，会输出testa(容器名称)，这时候可以通过 docker ps -a 查看容器是否启动正常。当容器状态显示为UP时，表示启动成功。



1. 本机测试：





1. 推送镜像到阿里云镜像仓库
2. 阿里云镜像地址: <https://cr.console.aliyun.com/cn-hangzhou/repositories>(这是我测试使用的，到真正商用的时候需要使用公司的)
3. 在阿里云创建一个命名空间：zhhtequipmanagerservices



1. 登录到阿里云：docker login --username=17710560656 registry.cn-hangzhou.aliyuncs.com

Username是阿里云的账户名

1. 标记TAG

docker tag [镜像ID] registry.cn-hangzhou.aliyuncs.com/zhhtequipmanagerservices

/testa:1.0.0

1. 推送到阿里云

docker push registry.cn-hangzhou.aliyuncs.com/zhhtequipmanagerservices

/testa:1.0.0



1. 从阿里云仓库拉取镜像、利用镜像创建启动容器、本地测试
2. 登录到阿里云

docker login --username=17710560656 registry.cn-hangzhou.aliyuncs.com

Username是阿里云的账户名

2、从阿里云拉取镜像

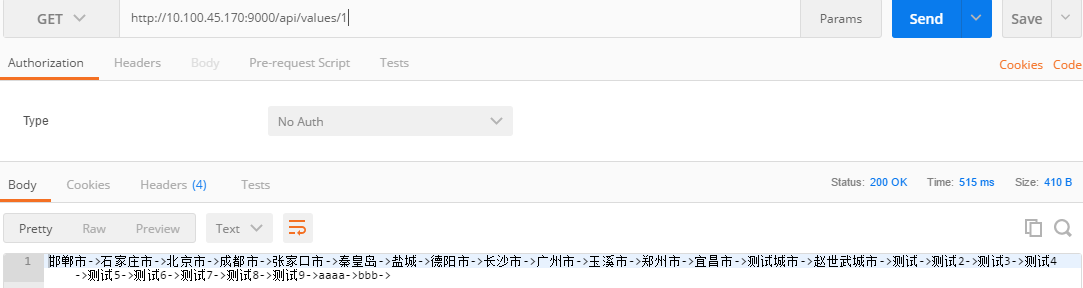
docker pull registry.cn-hangzhou.aliyuncs.com/zhhtequipmanagerservices/testa:1.0.0

1. 创建启动容器

docker run -p 9000:8080 -d --restart=always --name testa registry.cn-hangzhou.aliyuncs.com/zhhtequipmanagerservices/testa:1.0.0

-d表示放在后台运行，--restart=always表示重启自动运行

4、本地测试



1. Docker常用命令

|  |  |  |
| --- | --- | --- |
| 序号 | 命令 | 说明 |
| 1 | docker build –t testa:1.0.0 . | 生成镜像 |
| 2 | docker images | 查看所有镜像 |
| 3 | docker rmi imageID | 删除镜像，删除之前要确认没有使用该镜像的容器 |
| 4 | docker ps –a | 查看所有容器的状态 |
| 5 | docker start containerID | 启动容器 |
| 6 | docker stop containerID | 停止容器 |
| 7 | docker restart containerID | 重启容器 |
| 8 | docker rm containerID | 删除容器，在删除之前该容器没有运行 |
| 9 | docker logs –f containerID | 在外部查看容器的输出 |
| 10 | docker attach --sig-proxy=false containerID | 连接上容器实时查看，--sig-proxy=false保证在退出时，容器继续进行 |
| 11 | docker run –p 9000:8080 –d --restart=always  --name testa registry.cn-hangzhou.aliyuncs.com  /zhhtequipmanagerservices/testa:1.0. | 创建并启动容器 |
| 12 | docker exec -it containerID /bin/bash | 进入一个正在运行的容器（使用exec退出），-it指定以交模式打开 |
| 13 | docker top containerID | 查看容器内进程 |
| 14 | docker inspect containerID | 查看容器详细信息 |

1. 容器下常用命令安装
2. vim编辑器
3. 先进入容器：docker exec -it containerID /bin/bash
4. 安装vim命令，依次执行以下命令：

mv /etc/apt/sources.list /etc/apt/sources.list.bak

echo "deb http://mirrors.163.com/debian/ jessie main non-free contrib" >/etc/apt/sources.list

echo "deb http://mirrors.163.com/debian/ jessie-proposed-updates main non-free contrib" >>/etc/apt/sources.list

echo "deb-src http://mirrors.163.com/debian/ jessie main non-free contrib" >>/etc/apt/sources.list

echo "deb-src http://mirrors.163.com/debian/ jessie-proposed-updates main non-free contrib" >>/etc/apt/sources.list

apt-get update

apt-get install -y vim

1. Docker-Compose 安装

安装Docker-Compose之前，请先安装 python-pip

1. 首先检查Centos有没有安装python-pip，执行命令

pip -V

1. 没有python-pip包就执行命令

yum –y install epel-release

1. 执行成功之后，再次执行命令

yum -y install python-pip

1. 对安装好的pip进行升级，执行命令

pip install –upgrade pip

至此，pip安装完成，再次执行 pip –V 检查匹配环境

安装Docker-Compose

1. 执行安装命令

pip install docker-compose

1. 检查Docker-Compse安装

Docker-compose –version

1. 一个简单的docker-compose.yml示例

version: '3.4'

services:

services1:

image: registry.cn-hangzhou.aliyuncs.com/zhhtequipmanagerservices/testnettydemo:1.0.3 //使用的镜像名称

ports:

- '8007:8007' //端口映射

- '8008:8008'

restart: always //开机自动运行

services2:

image: registry.cn-hangzhou.aliyuncs.com/zhhtequipmanagerservices/testa:1.0.0

ports:

- '8080:8008'

restart: always

十一、常用的docker-compose命令

|  |  |
| --- | --- |
| 命令 | 说明 |
| docker-compose up -d | 创建并启动容器，服务后台运行。如果容器在此之前已经存在，则不创建容器，只后台运行容器 |
| docker-compose up --no-start | 创建但是不启动容器 |
| docker-compose up -d --force-recreate | 强制重新创建容器 |
| docker-compose rm | 删除停止的容器 |
| docker-compose start | 开启容器 |
| docker-compose stop | 停止容器 |
| docker-compose restart | 重启容器 |
| docker-compose down | 停止所有容器，并删除容器 |