微服务部署

建立日期：2020年6月1日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 变更原因 | 变更内容简述 | 编制/修订者 | 适用范围 | 发布日期 |
| V1.0 | 建立 |  | 陈文华 | 开发、运维 | 2020-6-1 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 一、安装docker

1、Docker 要求 CentOS 系统的内核版本高于 3.10 ，查看本页面的前提条件来验证你的CentOS 版本是否支持 Docker 。

通过 **uname -r**命令查看你当前的内核版本

$ uname -r

2、使用 root 权限登录 Centos。确保 yum 包更新到最新。

$ sudo yum update

3、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

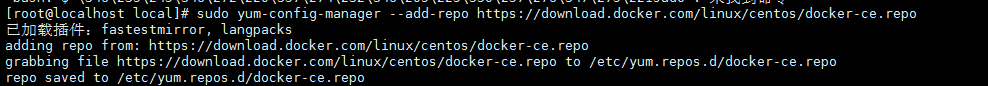
4、安装需要的软件包， yum-util 提供yum-config-manager功能，另外两个是devicemapper驱动依赖的

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

5、设置yum源

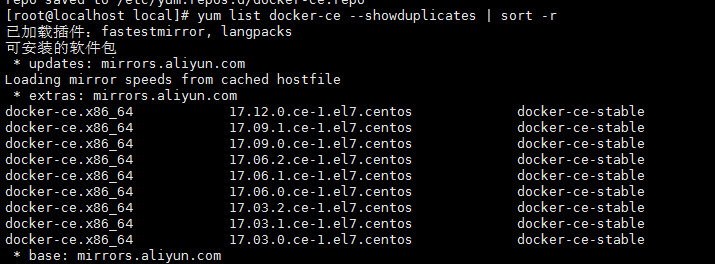
$ yum-config-manager --add-repo

http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo



6、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



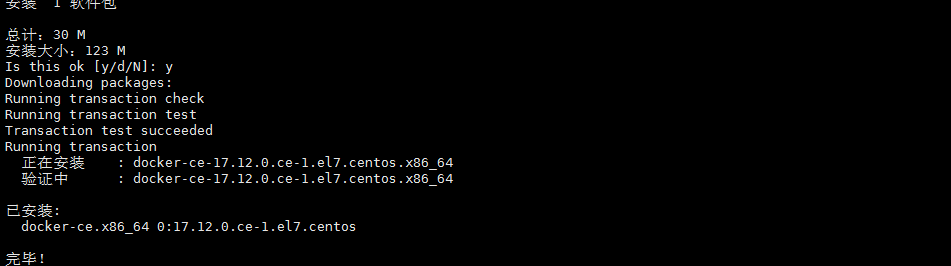
7、安装docker

$ sudo yum install <FQPN> # 例如：

yum install docker-ce-18.09.0 docker-ce-cli-18.09.0 containerd.io

或者直接安装最新版

yum install docker-ce docker-ce-cli containerd.io



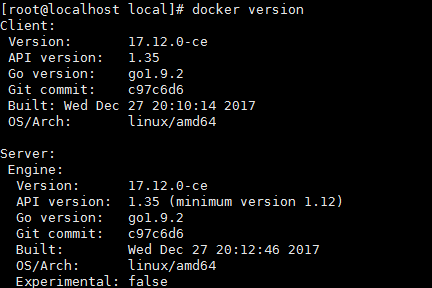
8、启动并加入开机启动

$ sudo systemctl start docker

$ sudo systemctl enable docker

9、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



# 二、配置镜像加速器

1、国内从 DockerHub 拉取镜像有时会遇到困难，此时可以配置镜像加速器。

Docker 官方和国内很多云服务商都提供了国内加速器服务，例如：

* 网易：**https://hub-mirror.c.163.com/**
* 阿里云：**https://<你的ID>.mirror.aliyuncs.com**
* 七牛云加速器：[**https://reg-mirror.qiniu.com**](https://reg-mirror.qiniu.com)

2、阿里云镜像加速器获取地址

<https://cr.console.aliyun.com/cn-hangzhou/instances/mirrors>

登陆后，左侧菜单选中镜像加速器就可以看到你的专属地址



3、按照此页面操作文档配置镜像加速器



# 三、安装Docker Compose

官方安装教程 <https://docs.docker.com/compose/install/>，主要步骤为：

1. 获取Docker Compose的最新稳定版本：

curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

1. 对二进制文件授予可执行权限：

chmod +x /usr/local/bin/docker-compose

1. 创建link：

ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

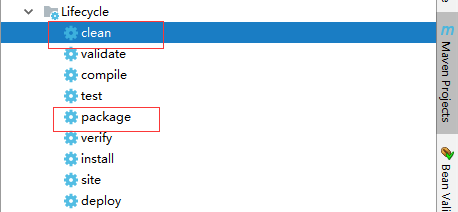
查看是否安装成功：

docker-compose version

# 四、构建后端镜像

1、maven工程打包

点击clean package按钮打包，每个服务模块的target目录下会有个可运行jar包



2、创建服务Dockerfile和jar包相关存放目录

mkdir -p /platform/platform-auth /platform/platform-gateway \

/platform/platform-admin /platform/platform-server-system \

/platform/platform-cloud-manage

说明：

platform-auth（认证服务）

platform-gateway（网关）

platform-admin（spring boot admin监控）

platform-server-system（系统服务）

platform-cloud-manage（前端工程）

3、目录授权

chmod 777 -R /platform

4、将各个模块打包之后target目录下的jar包上传至对应目录，查看是否都已上传成功

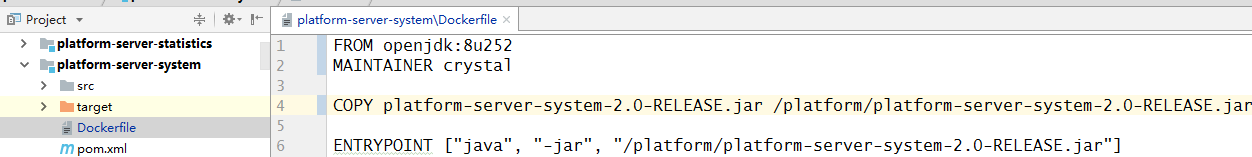
ls /platform/platform-auth /platform/platform-gateway \

/platform/platform-admin /platform/platform-server-system \

/platform/platform-cloud-manage

5、创建每个服务模块的Dockerfile，上传至模块目录

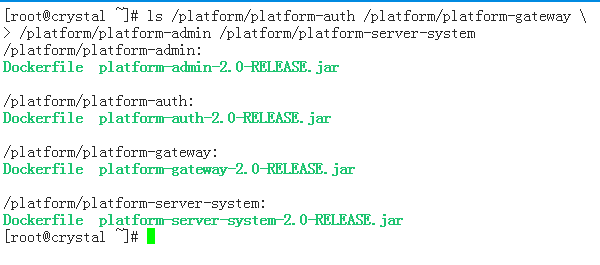
系统服务platform-server-system的Dockerfile内容如下，其他服务照着CV



Dockerfile指令参考

<https://www.runoob.com/docker/docker-dockerfile.html>

6、所有服务的Dockerfile上传至对应目录之后，查看目录是否有文件



7、在/platform/platform-server-system目录下执行

docker build -t platform-server-system .



注意：最后的英文句号 . 不要漏了，代表本次执行的上下文路径

8、其他服务也按照system服务构建好镜像之后，查看所有镜像

docker images

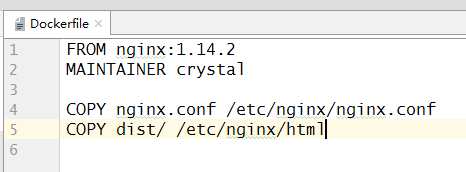
# 五、构建前端镜像

1、前端工程打包，执行

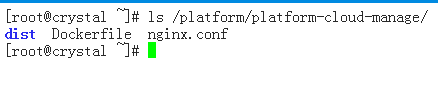
npm run build:prod

2、复制打包后生成的dist目录至 /platform/platform-cloud-manage目录下

上传nginx的配置文件nginx.conf至此目录下，并在此目录下创建Dockerfile，参考如下内容



3、查看/platform/platform-cloud-manage目录下是否已有文件



4、切换到/platform/platform-cloud-manage目录下，执行

docker build -t platform-cloud-manage .

构建镜像

5、查看镜像

docker images

# 六、构建nacos镜像

1、创建nacos文件存放目录

mkdir -p /platform/nacos

2、在此目录下创建docker-compose.yml文件

vim /platform/nacos/docker-compose.yml

内容参照如下：

**version: '3'**

**services:  
 nacos:  
 image:** nacos/nacos-server:1.1.4  
 **container\_name:** nacos-standalone-mysql  
 **environment:** - **"PREFER\_HOST\_MODE=hostname"** - **"MODE=standalone"** - **"SPRING\_DATASOURCE\_PLATFORM=mysql"** - **"MYSQL\_MASTER\_SERVICE\_HOST=127.0.0.1"** - **"MYSQL\_MASTER\_SERVICE\_DB\_NAME=ouma\_nacos\_v1"** - **"MYSQL\_MASTER\_SERVICE\_PORT=3306"** - **"MYSQL\_MASTER\_SERVICE\_USER=root"** - **"MYSQL\_MASTER\_SERVICE\_PASSWORD=123456"** - **"MYSQL\_SLAVE\_SERVICE\_HOST=127.0.0.1"** - **"MYSQL\_SLAVE\_SERVICE\_PORT=3306"  
 volumes:** - /platform/nacos/standalone-logs/:/home/nacos/logs  
 - /platform/nacos/custom.properties:/home/nacos

/init.d/custom.properties  
 **ports:** - 8848:8848  
 **restart:** on-failure

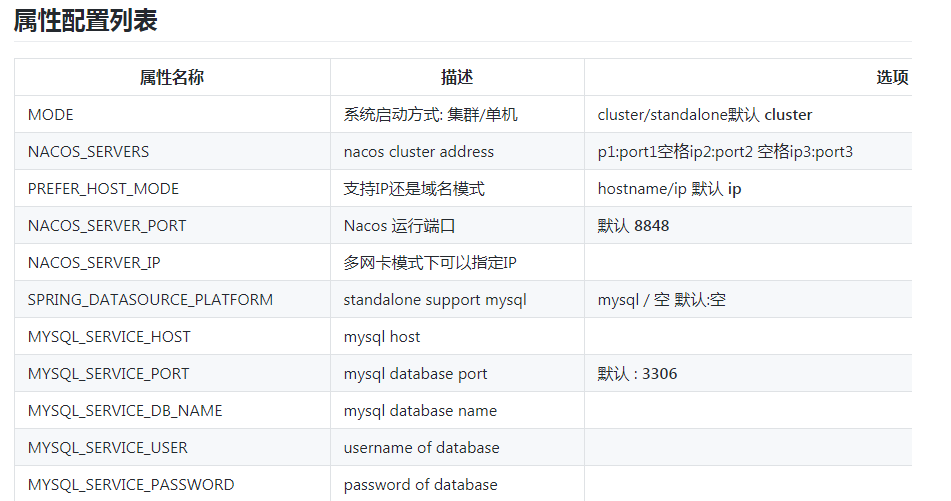
**说明：**

**（1）docker构建nacos参考**

<https://nacos.io/zh-cn/docs/quick-start-docker.html>

**（2）以上配置属于单机版nacos，集群环境参考官方地址**

**（3）以上environment属于1.1.4之前的配置，1.1.4版本以后的配置参照如下**



**（4）属性配置github地址**

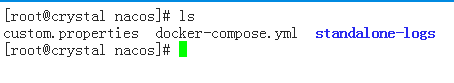
<https://github.com/nacos-group/nacos-docker/blob/master/README_ZH.md>

3、创建挂载点

touch /platform/nacos/custom.properties

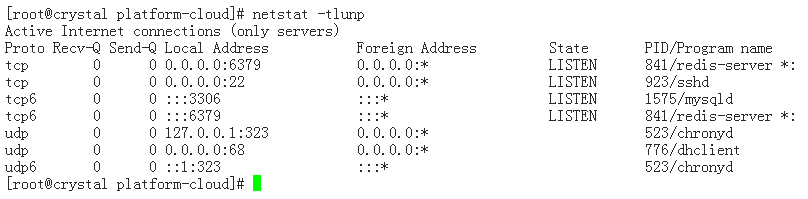
mkdir -p /platform/nacos/standalone-logs

4、查看已创建文件和目录



# 七、docker-compose部署

1、先启动系统所依赖的三方服务，如mysql redis rabbitMQ等，并配置好系统初始化库



如上所示，我已经启动了mysql和redis，这些三方服务也可以使用docker-compose部署，具体步骤和docker-compose.yml参照上一节

2、启动nacos

进入/platform/nacos目录，运行docker-compose up –d 命令，创建镜像，启动容器



访问http://服务器ip:8848/nacos查看是否已运行

3、启动微服务

（1）创建platform-cloud目录，用于存放docker-compose.yml文件：

mkdir -p /platform/platform-cloud

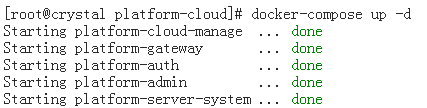
（2）上传docker-compose.yml文件至此目录，文件内容参考如下（修改成你自己的服务镜像配置）

**version: '3'  
  
services:**  
 **platform-auth:  
 image:** platform-auth:latest  
 **container\_name:** platform-auth  
 **volumes:** - **"/platform/log:/log"  
 environment:** - **"TZ=Asia/Shanghai"  
 ports:** - 9001:9001  
 **restart:** on-failure  
 **platform-server-system:  
 image:** platform-server-system:latest  
 **container\_name:** platform-server-system  
 **volumes:** - **"/platform/log:/log"  
 environment:** - **"TZ=Asia/Shanghai"  
 ports:** - 9002:9002  
 **restart:** on-failure

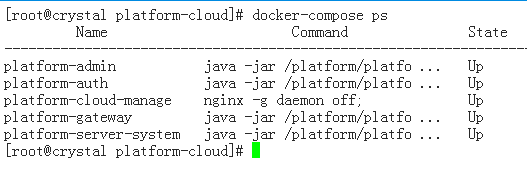
**platform-gateway:  
 image:** platform-gateway:latest  
 **container\_name:** platform-gateway  
 **volumes:** - **"/platform/log:/log"  
 environment:** - **"TZ=Asia/Shanghai"  
 ports:** - 9003:9003  
 **restart:** on-failure

**platform-admin:  
 image:** platform-admin:latest  
 **container\_name:** platform-admin  
 **volumes:** - **"/platform/log:/log"  
 environment:** - **"TZ=Asia/Shanghai"  
 ports:** - 9004:9004  
 **restart:** on-failure  
  
 **platform-cloud-manage:  
 image:** platform-cloud-manage:latest  
 **container\_name:** platform-cloud-manage  
 **volumes:** - **"/platform/log:/log"  
 environment:** - **"TZ=Asia/Shanghai"  
 ports:** - 80:80  
 **restart:** on-failure

（3）进入 /platform/platform-cloud目录运行docker-compose up –d 命令



4、至此，所有服务都已经启动，执行docker-compose ps 查看容器运行状态



State一栏的Up代表已正常启动，Exit表示容器已停止运行。

5、访问sba地址和nacos地址查看微服务运行状态，登录系统查看是否能正常运行。