**IMCloud EP工厂搭建部署文档**

**搭建环境条件：**

1. **单服务器：使用docker + docker compose进行搭建**
2. **多服务器：使用docker + docker compose + swarm 进行搭建**

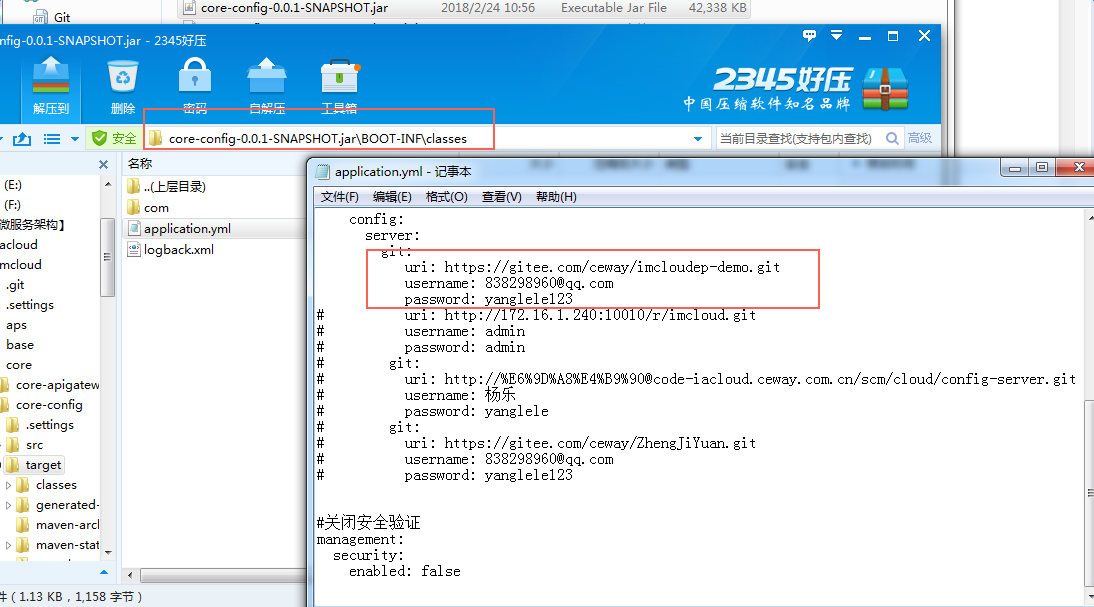
**准备工作：**

1. **Docker的安装**
2. **Docker compose的安装**

[**http://confluence-iacloud.ceway.com.cn/pages/viewpage.action?pageId=6292081**](http://confluence-iacloud.ceway.com.cn/pages/viewpage.action?pageId=6292081)

**第一部分（单服务器）**

1. **创建一个桥接网络，使用 docker network create imcloudep-external，用于单主机docker容器之间进行通信；**
2. **微服务镜像包需要使用到mysql，rabbitmq，redis，gitlab-ce，因此需要先进行这几个镜像的部署，我们可以使用docker-compose来进行一次性部署这些镜像； 需要使用到的mysql配置文件，rabbitmq，redis在baseISO目录下，复制到linux /opt目录下。docker-compose.yml文件在dockercompose-db目录下，拷贝到linux系统目录 /opt/imcloudep下，然后执行docker compose up –d即可**
3. **Git仓库的搭建，条件允许可以使用第三方git仓库，否则搭建gitlab-ce私有git仓库。不管是使用第三方还是自己搭建的，最后都要创建一个git仓库，用于存放微服务的配置文件。最后得到一个git仓库http的地址，这个http地址将在core-config项目代码中使用，所以core-config这个镜像每一家工厂都需要重新编译。可以通过一些压缩软件打开core-config.jar文件，修改application.yml文件中的红色标记代码，然后进行保存，进行重新制作镜像即可。**

****

1. **微服务镜像从阿里云docker仓库下载，阿里云帐号密码（jimaisi2017/Yy123456），注意下载命名空间为imcloudep的为测试包镜像，imcloudep-pro的为正式包镜像。**
2. **基础镜像部署完成，最后使用docker compose部署微服务镜像。拷贝dockercompose-core和dockercompose-function文件夹到linux /opt/imcloudep目录下，先进入dockercompose-core目录，运行docker compose up –d，启动核心的微服务，然后在启动function类型的微服务。**
3. **到这里，微服务就已经全部部署成功了，通过访问 http://{IP}:9000 ,来查看微服务程序是否部署成功。**

**第二部分（多服务器）**

1. **防火墙需要开启，放开几个必须的端口**

**firewall-cmd --permanent --add-port=2376/tcp**

**firewall-cmd --permanent --add-port=2377/tcp**

**firewall-cmd --permanent --add-port=7946/tcp**

**firewall-cmd --permanent --add-port=7946/udp**

**firewall-cmd --permanent --add-port=4789/udp**

**firewall-cmd –reload**

1. **初始化一台服务器为主节点，172.16.1.241为主节点服务器的IP地址**

**docker swarm init --advertise-addr 172.16.1.241**

1. **根据提示，在需要加入swarm的服务器，执行上面产生的结果**

**例如：**

**docker swarm join --token SWMTKN-1-6a2kdr3hx6i1lla8e0f2fmqs5nufbgnw0f2nuq5ksut2yq8rui-46ym3a8niildhwg7hzge1yuj4 172.16.1.241:2377**

1. **这样就建立了主节点和子节点，可以在主节点上使用docker node ls查看子节点信息**
2. **建立overlay网络，用于跨主机的docker容器通信**

**docker network create \**

**--driver overlay \**

**--subnet 152.80.1.0/24 \**

**--opt encrypted \**

**springcloud-overlay**

1. **Mysql，redis，rabbitmq，gitlab-ce同样使用docker-compose进行部署, 单个执行参考下面的代码截图**

****

1. **微服务镜像的部署也同样是这样**
2. **注意最后执行应该是docker stack deploy –c docker-compose-baseISO.yml imcloud/baseiso，而不是docker compose up –d**
3. **这些docker-compose-db.yml，docker-compose-core.yml，docker-compose-base.yml，docker-compose-wms.yml等文件，都在主节点执行**
4. **Swarm会自动查询所有节点，镜像在哪台服务器，则运行在哪台服务器**
5. **到这里，多服务器的部署就完成了，通过 http://{IP}:9000访问，看到页面则说明部署成功。**

version: '3' #表示使用的docker-compose编译的版本，目前为3

services: #微服务列表

db-mysql: #服务的名称，由我们自己取，后面其它微服务可以通过db-mysql连接

image: mysql #镜像名称

volumes: #挂载宿主机目录

- /opt/mysql/data:/var/lib/mysql #mysql数据持久化

- /opt/mysql/conf:/etc/mysql/conf.d #mysql配置文件，解决编码，查询模式等

- /etc/localtime:/etc/localtime #mysql时间同步

environment: #设置环境变量

- MYSQL\_ROOT\_PASSWORD=Hello!!!123 #root用户初始密码设置

ports: #暴露端口

- "3306:3306"

command: mysqld --character-set-server=utf8 --collation-server=utf8\_unicode\_ci

#command为该镜像运行时需要执行的命令，这里的作用是设置编码

cache-redis:

image: redis

ports:

- "6379:6379"

volumes:

- /opt/redis/redis.conf:/usr/local/etc/redis/redis.conf

- /opt/redis/data:/data

command: redis-server /usr/local/etc/redis/redis.conf

imcloudep-rabbitmq:

image: rabbitmq:3-management

environment:

- RABBITMQ\_DEFAULT\_USER=root

- RABBITMQ\_DEFAULT\_PASS=root

volumes:

- /opt/rabbitmq:/var/lib/rabbitmq

ports:

- "15672:15672"

#使用自定义的桥接网络，将所有的服务桥接在一个局域网内，可以通过别名（例如db-mysql）进行通信

networks:

default:

external:

name: imcloudep-external