

一：在准一上部署项目。

项目一：小白准一==>部署RefundStatus从172.16.50.216 ---> 172.16.20.73上跑项目

准一在172.16.50.216上push, 到73上去pull localhost:15000, 听说在10.251.26.11上面跑不通.

1. 创建镜像

```
docker build -t rc-image-guoling-refundstatus-consumer:v20170719_01 .
```

2. 标记镜像

```
docker tag rc-image-guoling-refundstatus-consumer:v20170719_01
```

```
localhost:15000/tyxb/rc-image-guoling-refundstatus-consumer:v20170719_01
```

3. 把被标记的镜像上传仓库((172.16.50.216上面是15000不是5000))

```
docker push localhost:15000/tyxb/rc-image-guoling-refundstatus-consumer:v20170719_01
```

4. 去73服务器把镜像pull下来

```
docker pull 172.16.50.216:15000/tyxb/rc-image-guoling-refundstatus-  
consumer:v20170719_01
```

5. 创建并运行容器

```
docker run -t --name rc-guoling-consumer -d 172.16.50.216:15000/tyxb/rc-image-  
guoling-refundstatus-consumer:v20170719_01 /bin/bash
```

注解：

-t 新开一个控制台

--name 为容器指定一个名称

-d 在后台运行容器并打印容器ID

如果第五步走完了, 没有将容器中的项目启动的话, 说明dockerfile文件中的自启动项目有问题.
就要进行第6步

6. 进入容器

```
docker exec -it rc-guoling-consumer /bin/bash
```

7. 手动启动项目

```
cd /app/python_apps/RefundStatus
```

```
python Refundmain.py
```

8. 查看日志

先进入容器

再 `cd /opt/logs/rc-guoling/python-apps-logs/SendTyxb`

即可查看日志

提示:不知道如何在dockerfile中让挂载日志生效,所以只能进入容器看日志.我这个服务是手动起来的

准一上面的测试数据

非小白的:

```
{"biz": {"order_id": "", "refund_amount": 3, "out_order_id": "", "result": true,
"refund_order_id": "2017071714394900235682", "out_transaction_id":
"2017071721001004860266501478", "transaction_id": ""}}
```

小白的:

```
{"biz": {"order_id": "", "refund_amount": 3, "out_order_id": "", "result": false,
"refund_order_id": "2017071714394900235682", "out_transaction_id":
"2016081600000632018555", "transaction_id": ""}}
{"biz": {"order_id": "", "refund_amount": 3, "out_order_id": "", "result": true,
"refund_order_id": "2017071714394900235682", "out_transaction_id":
"2016081600000632018555", "transaction_id": ""}}
```

二: 在生产上部署项目.

项目二:小白生产==>部署RefundStatus从10.251.26.11->10.128.112.120, 10.128.112.121 上
跑项目 localhost:5000

再到120通过集群的方式一次性开启多个服务器

大致步骤:

1. 在10.251.26.11上面创建镜像

```
docker build -t prod-image-tyxb-refundstatus-guoling:v20170720_02 .
```

2. 标记镜像

```
docker tag prod-image-tyxb-refundstatus-guoling:v20170720_02
```

```
localhost:5000/tyxb/prod-image-tyxb-refundstatus-guoling:v20170720_02
```

3. 把被标记的镜像push到本机的仓库中(10.251.26.11上面是5000不是15000)

```
docker push localhost:5000/tyxb/prod-image-tyxb-refundstatus-guoling:v20170720_02
```

4. 去堡垒机中如:10.128.112.120上面把项目从仓库中pull下来。(注意120和121是关联的,假如你启动了四个服务,他们会内部分发给这两个服务器去跑起来)

4. 如果用docker-compose.yml文件的话,就不用pull项目,直接执行下面的命令就能把服务跑起来.没有第五步了.

```
docker stack deploy -c docker-compose.yml tyxb    # tyxb 只是取一个别名
```

5. 创建服务器集群,实现在一个服务器上的容器中启动项目,同时能启动能把10.128.112.120容器中也能跑一套刚才启动的项目,每个服务器要启动多少个,取决于

docker-compose.yml中replicas参数值;

6. 如果要删除原有集群启动的所有服务

先查看服务

```
docker service ls
```

再找到服务的名字,比如:tyxb_prod-guoling-consumer

```
docker service rm tyxb_prod-guoling-consumer
```

再查看服务,发现真的被删除了,队列中确实没有消费程序了.

```
docker service ls
```

生产上的测试数据:

```
{"biz": {"order_id": "83917071970201000741", "refund_amount": 3000, "out_order_id": "", "result": true, "refund_order_id": "2017071809300985938890", "out_transaction_id": "4009292001201707191564536267", "transaction_id": ""}}
```

查看日志路径:

```
cd /app/logs/prod-guoling/rc-guoling/python-apps-logs/SendTyxb
```

项目三:部署MiscService从172.16.50.216 ---> 172.16.20.73上跑项目

1. 创建镜像

```
docker build -t rc-image-guoling-miscservice:20170721_01 .
```

2. 标记镜像

```
docker tag rc-image-guoling-miscservice:20170721_01 localhost:15000/tyxb/rc-image-guoling-miscservice:20170721_01
```

3. 把被标记的镜像上传仓库

```
docker push localhost:15000/tyxb/rc-image-guoling-miscservice:20170721_01
```

4. 去73服务器把镜像pull下来

```
docker pull 172.16.50.216:15000/tyxb/rc-image-guoling-miscservice:20170721_01
```

5. 创建并运行容器

```
docker run -t --name rc-guoling-miscservice -d 172.16.50.216:15000/tyxb/rc-image-guoling-miscservice:20170721_01 /bin/bash
```

如果第五步走完了, 没有将容器中的项目启动的话, 说明dockerfile文件中的自启动项目有问题. 就要进行第6步

运行容器: (我运行容器. 容器中的代码居然运行起来啦. 哈哈, 可以网页上访问不了这个 MiscService, 发现73上的外网通不了. 于是乎, 就采用了下面的方案2, 结果, 一切问题都解决啦。)

```
docker run -it rc-image-guoling-miscservice:20170721_01
```

6. 进入容器

```
docker exec -it rc-guoling-miscservice /bin/bash
```

7. 手动启动项目

```
cd /app/python_apps/RefundStatus  
python Refundmain.py
```

8. 查看日志

先进入容器

```
再 cd /opt/logs/rc-guoling/python-apps-logs/SendTyxb
```

即可查看日志

提示: 不知道如何在dockerfile中让挂载日志生效, 所以只能进入容器看日志. 我这个服务是手动起来的

或者:

1. 创建镜像

```
docker build -t rc-image-guoling-miscservice:20170721_01 .
```

2. 创建并运行容器

```
docker run -t --name rc-guoling-miscservice -d rc-image-guoling-miscservice:20170721_01 /bin/bash
```

3. 进入容器

```
docker exec -it rc-guoling-miscservice /bin/bash
```

4. 手动启动项目

```
cd /app/python_apps/RefundStatus  
python Main.py
```

方案2(可行):

1. 在172.16.50.216服务器上用dockerfile创建镜像, 再通过执行docker-compose.yml启动项目

```
docker build -t rc-image-guoling-miscservice:20170721_01 .
```

2. 创建并运行容器

```
docker run -t --name rc-guoling-miscservice -d rc-image-guoling-  
miscservice:20170721_01 /bin/bash
```

3. 运行docker-compose.yml文件(端口映射)

```
docker-compose up -d
```

结果:项目就启动起来了...

*****个人信息*****

给古龙涛老师, 让其帮忙绑定队列

准一的:

```
amqp://smallrabbit:123456@172.16.20.73:5672/order|x.order.Feedback|guoling_Routing_Key
```

生产的:

```
amqp://rabbit_order:SJxmvYi5wB@10.128.112.101:5672/order|x.order.Feedback|RefundStatus
```

三: 在生产上部署项目.

小白生产==>部署tyxb-send-sms从10.251.26.11->10.128.112.120, 10.128.112.121 上跑项目

1. 直接进入 10.251.26.11 服务器 走到这个路径下: /home/was/docker_compose/prod-guoling/prod-tyxb-python-job/tyxb-send-sms/build_sh

执行 叫build.sh的脚本 执行命令: `./build.sh` 这个脚本会帮你创建镜像.

2. 进入10.128.112.120 服务器 走到这个路径下: `/home/was/docker_compose/prod-guoling/tyxb-send-sms`

会有一个docker-compose.yml文件, 运行这个文件即可 执行命令: `docker stack deploy -c docker-compose.yml tyxb # tyxb` 就是取一个别名

小白生产服务器: `http://10.128.112.120:19003/` , 可以查进程号, 方便查看日志 19003 端口号是 是通过运行项目所在的服务器上执行:`docker service ls` 看到的;

准一服务器: `http://172.16.20.46:9001/` , 可以查进程号, 方便查看日志