文件名称：Docker分布式部署手册

文件编号：

项目名称：调兵山项目

项目编号：

物料编码：

版 本 号：A

文件密级：秘密

文件状态：CFC

受控标识：受控

|  |  |
| --- | --- |
| 拟制/日期： | 2019年 月 日 |
| 审核： | 2019年 月 日 |
| 会签： |  |
|  |  |
| 批准： | 2019年 月 日 |

**文件发放范围：**

**修订页**

| 序号 | 版本号 | 修订内容简述 | 拟制/日期 | 审核 | 批准 |
| --- | --- | --- | --- | --- | --- |
|  | A | 创建 | 张毅 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **部署概述**

为实现应用服务在Docker环境下的分布式部署，需要以下服务环境：

* Docker：提供服务或应用部署的环境
* Swarm：提供服务或应用的分布式管理、服务的负载均衡、网络网关域管理等
* Ceph：提供服务数据分布式数据同步、配置文件以及数据的一致性的能力
* Rexray：提供swarm与ceph之间消息的交互，从而保证分布式应用服务与数据的正常交互

1. **服务器环境（样例）**



|  |  |  |
| --- | --- | --- |
| **物理服务器** | **虚拟服务器** | **备注** |
| 172.21.32.21 | 172.21.32.106 | 将数据库服务部署此服务器(**MySQL、Canal、Redis、node-red、influxdb、neo4j、mqtt**) |
| 172.21.32.56 | 172.21.32.110 |  |
| 172.21.32.114 |

1. **Docker环境**
2. docker环境搭建

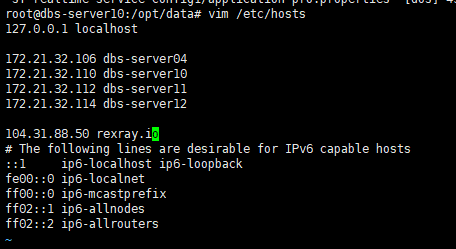
参考地址：<https://www.runoob.com/docker/ubuntu-docker-install.html>

1. docker私服搭建

docker私服为了在分布式环境获取统一的应用镜像以及定制后服务镜像(mysql、xxl-job等)

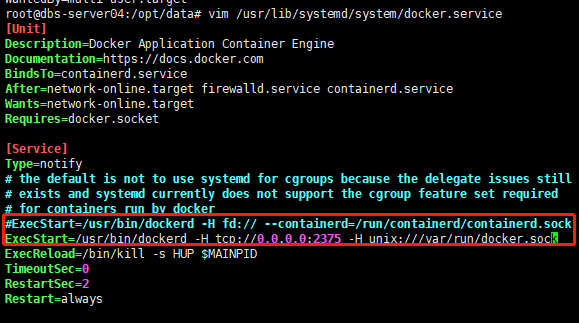
1. **Swarm配置与部署**
2. 修改hosts文件(/etc/hosts)

*在114、110、106服务器添加服务域名与IP的解析关系*



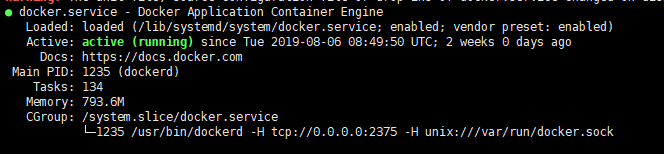
1. Docker开启2375端口(**/usr/lib/systemd/system/docker.service**)

*在114、110、106服务器上分别开启2375端口*



|  |
| --- |
| vim /usr/lib/systemd/system/docker.service  ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock |

*在114、110、106服务器上重启docker并查看docker状态*

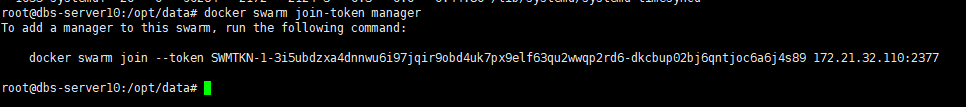


|  |
| --- |
| systemctl daemon-reload  systemctl restart docker  systemctl status docker |

1. Swarm初始化

*在110服务器上初始化swarm管理者*

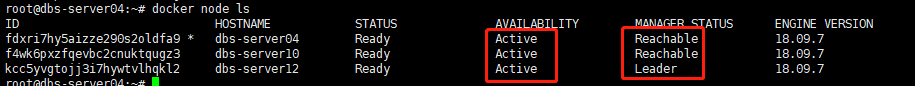
|  |
| --- |
| docker swarm init --advertise-addr 172.21.32.110  docker swarm join-token manager |



*在114、106服务器上加入swarm管理者，执行上述返回结果*

查看swarm集群状态

|  |
| --- |
| docker node ls |



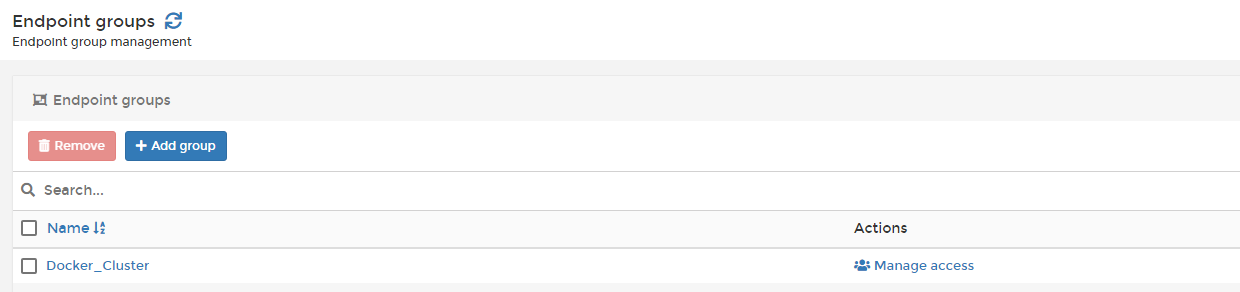
**注：集群可用性为Active以及管理状态为Leader或Reachable即为成功**

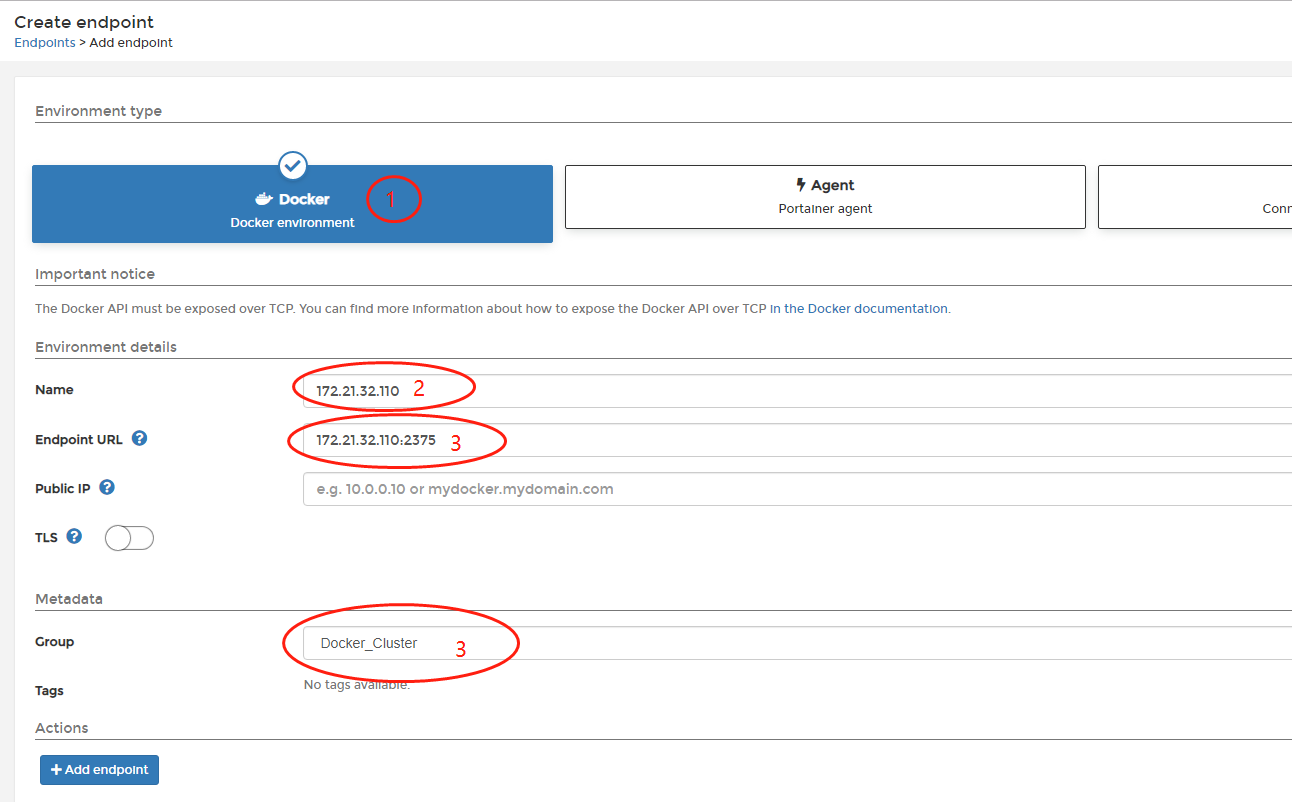
1. **Portainer服务部署**
2. 创建Portainer服务

|  |
| --- |
| docker service create \  --name portainer \  --publish 9000:9000 \  --replicas=1 \  --constraint 'node.role == manager' \  --mount type=bind,src=//var/run/docker.sock,dst=/var/run/docker.sock \  --container-label com.docker.stack.namespace=swarm \  --label com.docker.stack.namespace=swarm \  172.21.32.31:5000/portainer/portainer \  -H unix:///var/run/docker.sock |

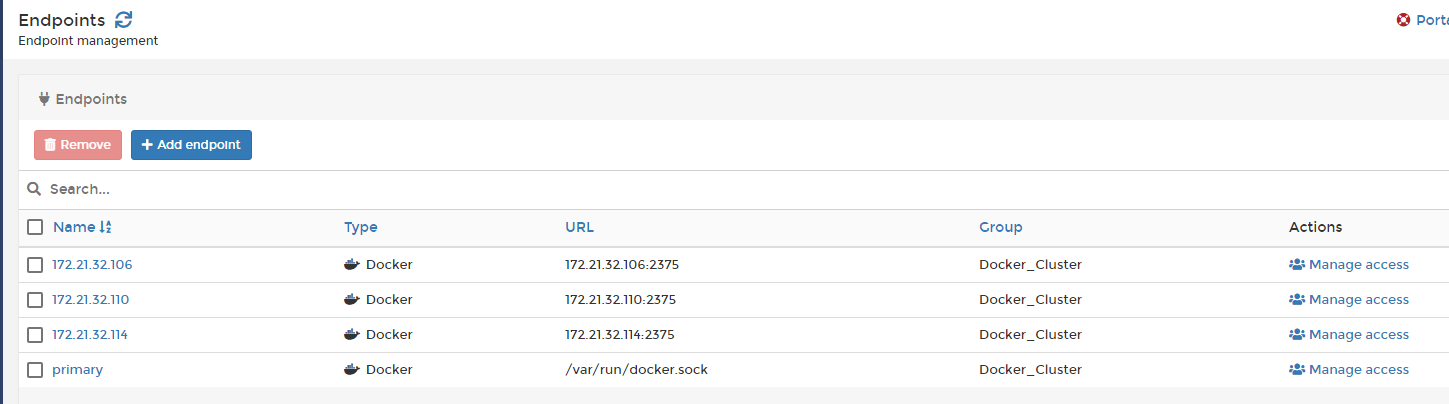
***注：分片数必须是1,多个会出现跳动而无法进入管理界面***

1. 配置swarm节点及分组

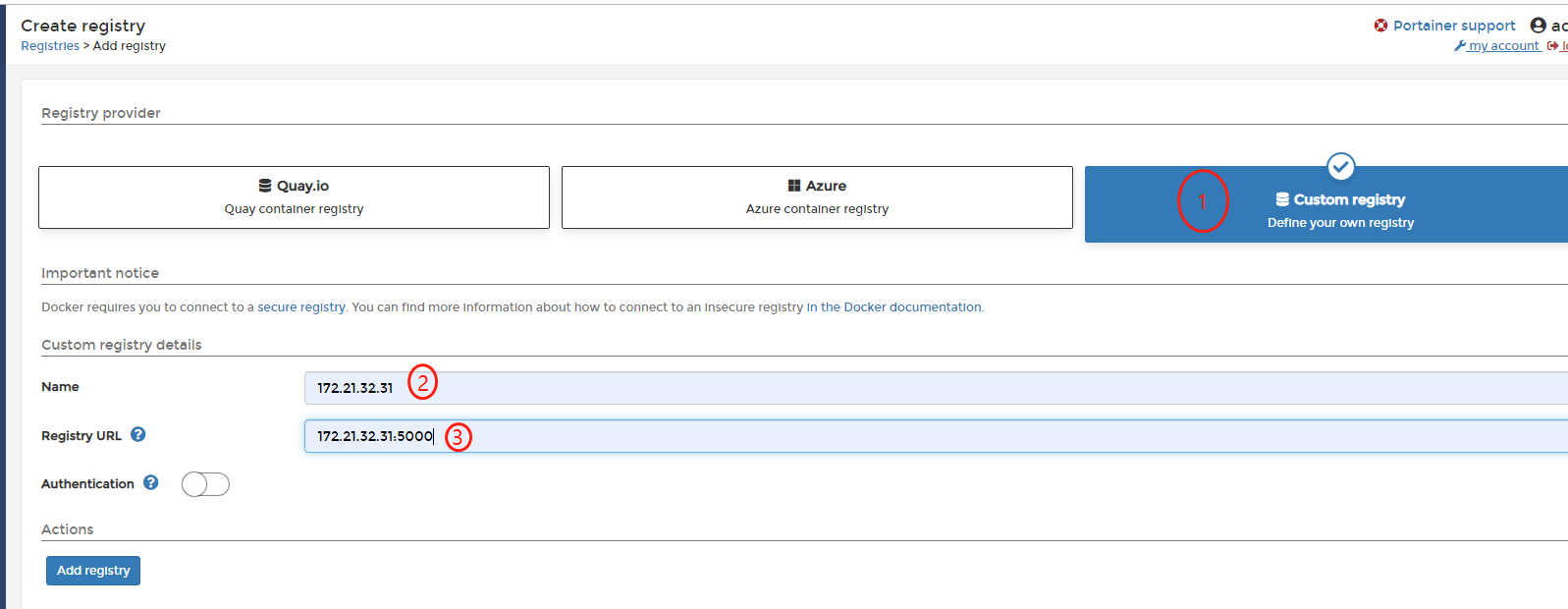




**标号2处就是对应开启的2375端口**



1. 配置docker私库



1. **Ceph配置与部署**
2. 挂载OSD硬盘(如: /dev/sdb)

*在110服务器上操作*

|  |
| --- |
| rm -rf /etc/ceph  rm -rf /var/lib/ceph  rm -rf /opt/ceph/rbd/\*  rm -rf /var/log/ceph  mkdir -p /etc/ceph /var/lib/ceph /var/lib/ceph/osd /opt/ceph/rbd /opt/ceph/data /var/log/ceph  chown -R 167:167 /etc/ceph  chown -R 167:167 /var/lib/ceph  mkfs.xfs /dev/sdb -f  mount /dev/sdb /opt/ceph/rbd |

***注：若持久化挂载硬盘，防止重启后丢失，进入如下命令：(此操作无需重复执行)***

|  |
| --- |
| echo '/dev/sdb /opt/ceph/rbd nfs defaults 0 0' >> /etc/fstab |

1. 创建ceph的docker集群容器(mon、mgr、osd)

*在110服务器上操作*

|  |
| --- |
| docker pull ceph/daemon:latest-devel  docker run -d \  --name mon \  --restart=always \  --mount type=bind,src=//etc/ceph,dst=/etc/ceph \  --mount type=bind,src=//var/lib/ceph,dst=/var/lib/ceph \  --env MON\_IP=172.21.32.110,172.21.32.114,172.21.32.106 \  --env DEBUG=stayalive \  --env CEPH\_PUBLIC\_NETWORK=172.21.32.0/24 \  --network=host \  --label com.docker.stack.namespace=dbs \  ceph/daemon:latest-devel mon  docker run -d \  --name osd \  --privileged=true \  --restart=always \  --mount type=bind,src=//etc/ceph,dst=/etc/ceph \  --mount type=bind,src=//var/lib/ceph,dst=/var/lib/ceph \  --mount type=bind,src=//dev,dst=/dev \  --mount type=bind,src=//opt/ceph/rbd,dst=/var/lib/ceph/osd \  --network=host \  --label com.docker.stack.namespace=dbs \  ceph/daemon:latest-devel osd  docker run -d \  --name mds \  --restart=always \  --mount type=bind,src=//etc/ceph,dst=/etc/ceph \  --mount type=bind,src=//var/lib/ceph,dst=/var/lib/ceph \  --network=host \  --label com.docker.stack.namespace=dbs \  ceph/daemon:latest-devel mds  docker run -d \  --name mgr \  --restart=always \  --mount type=bind,src=//etc/ceph,dst=/etc/ceph \  --mount type=bind,src=//var/lib/ceph,dst=/var/lib/ceph \  --network=host \  --label com.docker.stack.namespace=dbs \  ceph/daemon:latest-devel mgr |

***\*注：mon下MON\_IP配置集群服务器IP，OSD下绑定步骤1中的硬盘***

1. Ceph管理授权

*在110服务器上操作*

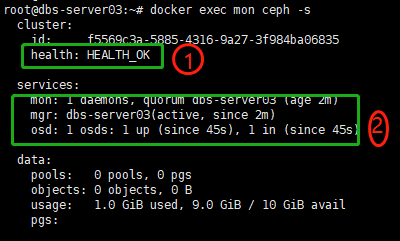
**注：在进行以下操作时需保证docker下mon、mgr、osd的状态为UP**

|  |
| --- |
| docker exec mon ceph auth get client.bootstrap-osd -o /var/lib/ceph/bootstrap-osd/ceph.keyring  docker exec mon ceph auth get client.bootstrap-mds -o /var/lib/ceph/bootstrap-mds/ceph.keyring  docker exec mon ceph auth get client.bootstrap-rgw -o /var/lib/ceph/bootstrap-rgw/ceph.keyring  docker exec mon ceph auth get client.bootstrap-rbd-mirror -o /var/lib/ceph/bootstrap-rbd-mirror/ceph.keyring |

1. 查看集群状态

*在110服务器上操作*

|  |
| --- |
| docker exec mon ceph -s |



**注：请确保标记①为HEALTH\_OK，标记②各节点均加入集群**

1. 拷贝配置服务以及ceph目录

*在110服务器上拷贝到106、114上/home/zhangkun/ceph或/home/zhangkun/ceph-var*

|  |
| --- |
| rsync -a /etc/ceph/\* zhangkun@172.21.32.114:/home/zhangkun/ceph && rsync -a /var/lib/ceph/bootstrap-\* zhangkun@172.21.32.114:/home/zhangkun/ceph-var  rsync -a /etc/ceph/\* zhangkun@172.21.32.106:/home/zhangkun/ceph && rsync -a /var/lib/ceph/bootstrap-\* zhangkun@172.21.32.106:/home/zhangkun/ceph-var |

*在106、114服务器上重复****步骤1****操作后，进行拷贝授权操作*

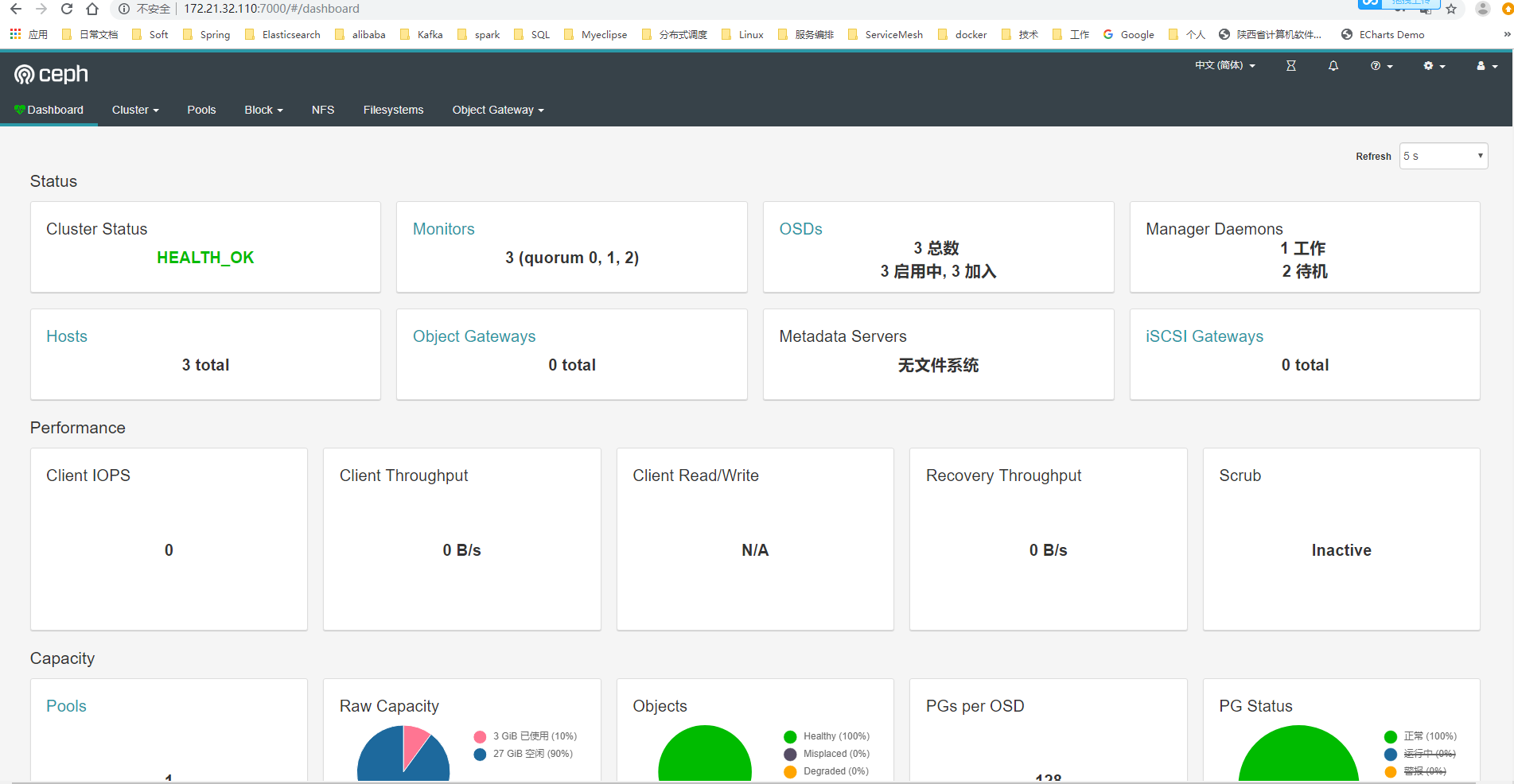
|  |
| --- |
| mv /home/zhangkun/ceph/\* /etc/ceph  mv /home/zhangkun/ceph-var/\* /var/lib/ceph  chown -R 167:167 /etc/ceph  chown -R 167:167 /var/lib/ceph  chown -R root:root /etc/ceph/ceph.conf |

1. 在集群其他服务器(*106、114服务器*)重复2-4步骤
2. 开启管理界面

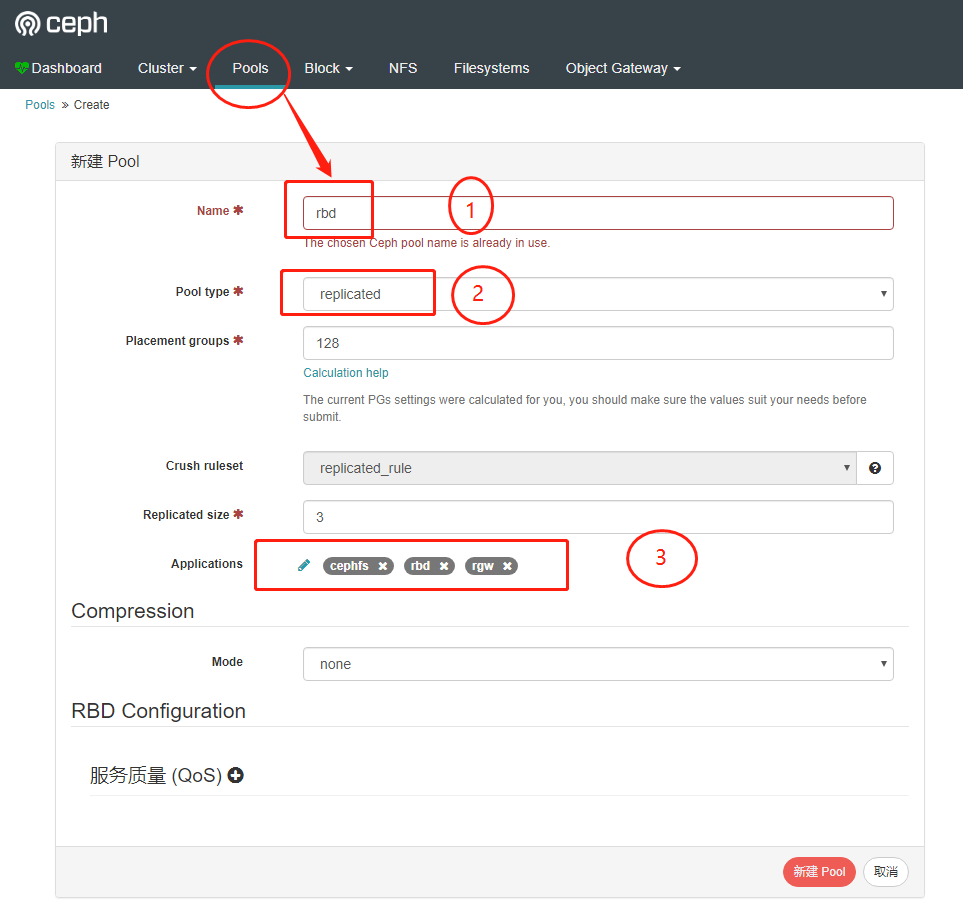
*在110、106、114服务器上均操作*

|  |
| --- |
| docker exec mgr ceph mgr module enable rbd\_support  docker exec mgr ceph mgr module enable dashboard  docker exec mgr ceph dashboard create-self-signed-cert  docker exec mgr ceph dashboard set-login-credentials ceph ceph  docker exec mgr ceph config set mgr mgr/dashboard/server\_port 7000  docker exec mgr ceph config set mgr mgr/dashboard/ssl false  docker restart mgr  docker exec mgr ceph mgr services |

登录<http://172.21.32.110:7000/#/dashboard> 账户密码: ceph/ceph



1. 创建rbd池



**注：标记①必须为：rbd,标记②选replicated，标记③应用标签为：cephfs rbd rgw**



1. 系统内核加载rbd块

*在110、106、114服务器上均操作*

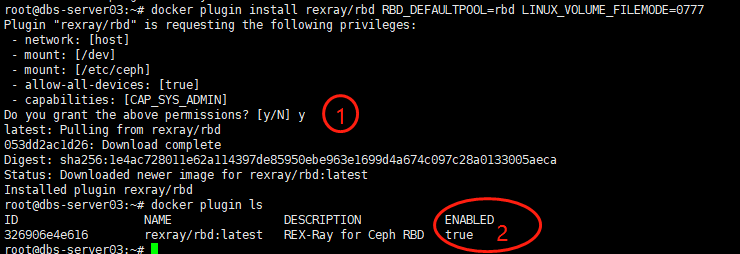
|  |
| --- |
| modprobe rbd  lsmod |grep rbd |



1. **Raxray配置与部署**
2. 安装docker插件rexray/rbd

*在110、106、114服务器上均操作*

|  |
| --- |
| docker plugin disable rexray/rbd  docker plugin rm rexray/rbd  docker plugin install rexray/rbd RBD\_DEFAULTPOOL=rbd LINUX\_VOLUME\_FILEMODE=0777 |



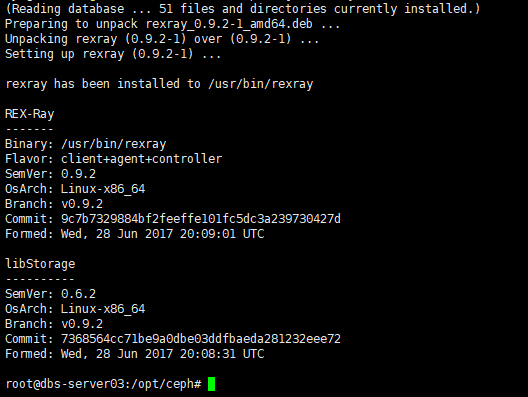
**注：标记①为y，标记②为true**

1. 安装rexray

*在110、106、114服务器上均操作*

|  |
| --- |
| rexray uninstall  rm -rf $(find / -name rexray)  rm -rf $(find / -name libstorage)  curl -sSL https://rexray.io/install | sh -s -- stable 0.9.2 |

**注: raxray版本必须是0.9.2**



1. Raxray配置(/etc/rexray/config.yml)

*在110、106、114服务器上创建config.yml配置内容如下:*

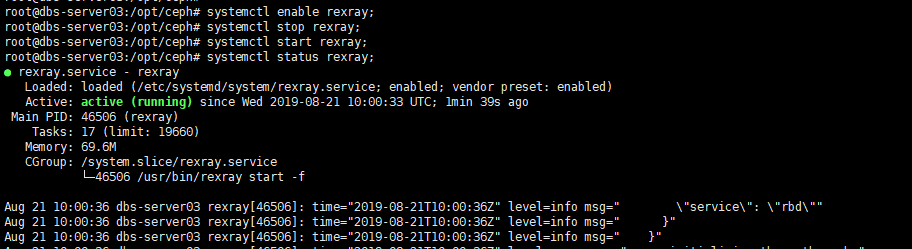
|  |
| --- |
| rexray:  logLevel: info  libstorage:  logging:  level: info  httpRequests: true  httpResponses: true  service: rbd  rbd:  defaultPool: rbd |

配置参考或下载: <https://rexrayconfig.cfapps.io/>

1. 重启Rexray并查看状态

*在110、106、114服务器上均操作*

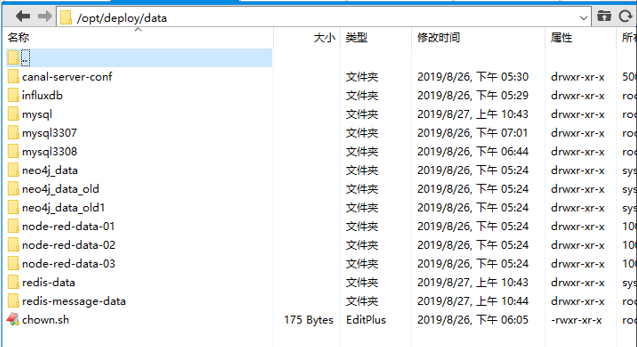
|  |
| --- |
| systemctl enable rexray;  systemctl stop rexray;  systemctl start rexray;  systemctl status rexray; |



**注: 确保指标状态为active(running)**

1. **应用服务配置与部署**
2. 数据库拷贝

将数据一致性的数据库或服务（canal/mysql/neo4j/redis/node-red）拷贝到特定服务器上*(106服务器)，*可以采用(scp –r)/rsync –a)等跨服务器命令或tar包方式拷贝，拷贝目录：/opt/deploy/data

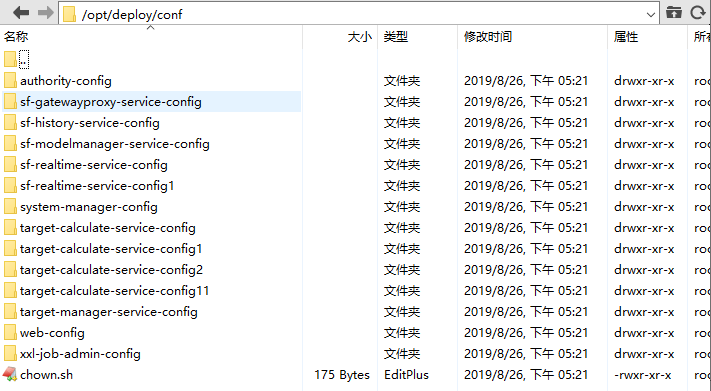


注：数据相关的仅在特定一台服务器上

1. 应用配置拷贝

*在110、106、114服务器上均操作*

将所有应用配置拷贝到集群服务器上，拷贝目录：/opt/deploy/conf

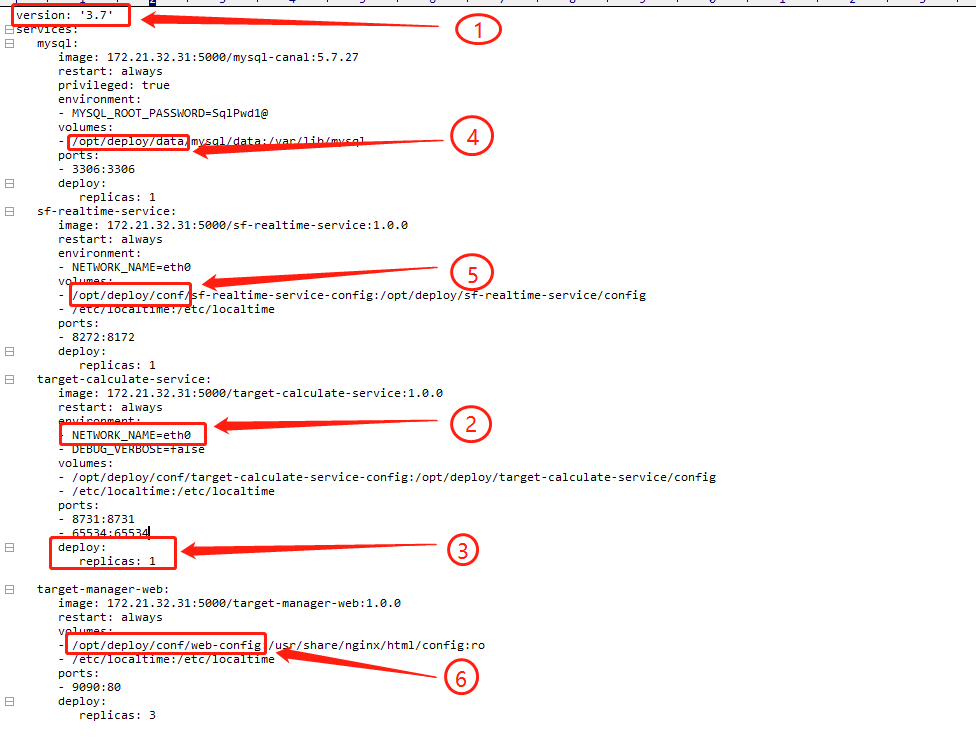


**注：数据服务器*(106服务器)*上既包含数据应用又包含应用服务配置**

1. Docker集群服务配置

Docker集群服务配置见附件





**注：**

**docker的yml文件版本：3.7 （如图①）**

**NETWORK\_NAME：指定网卡名称(实时服务和指标计算服务必须指定网卡如图标注) （如图②）**

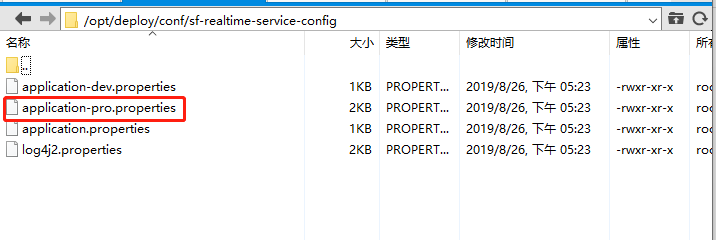
**deploy.replicas：集群分片数即集群服务数（如图③）(MySQL、Canal、Redis、node-red、influxdb、neo4j、mqtt、system-authority-web的分片数必须是1)**

**标记④：数据目录**

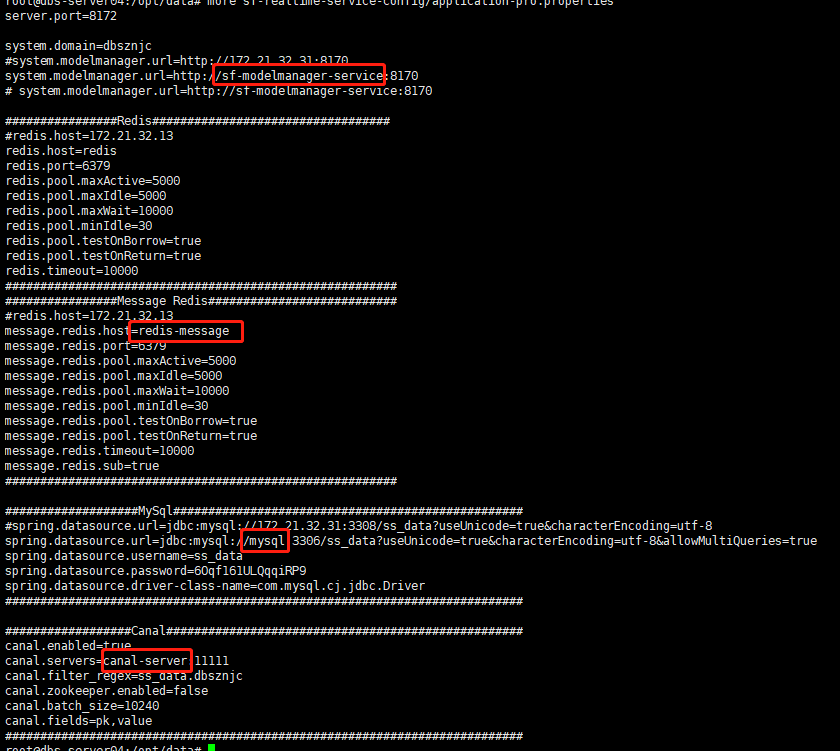
**标记⑤：后台应用目录**

**标记⑥：前台目录(所有前台仅有一个配置文件)**

1. 应用服务配置



各后台应用服务配置修改对应目录下的/opt/deploy/conf/\*-config\*/application-pro.properties内容



**注：所有地方采用域名方式，无需使用IP方式，如上图**

各后台应用服务配置修改对应目录下的/opt/deploy/conf/web-config/config.js内容



**注：**

**标注①：端口为Nginx反向代理地址端口(解决统一登录问题)**

**标注②：修改为集群主机地址**

**标注③：修改诊断服务地址**

**标注④：修改为false**

1. Nginx部署配置

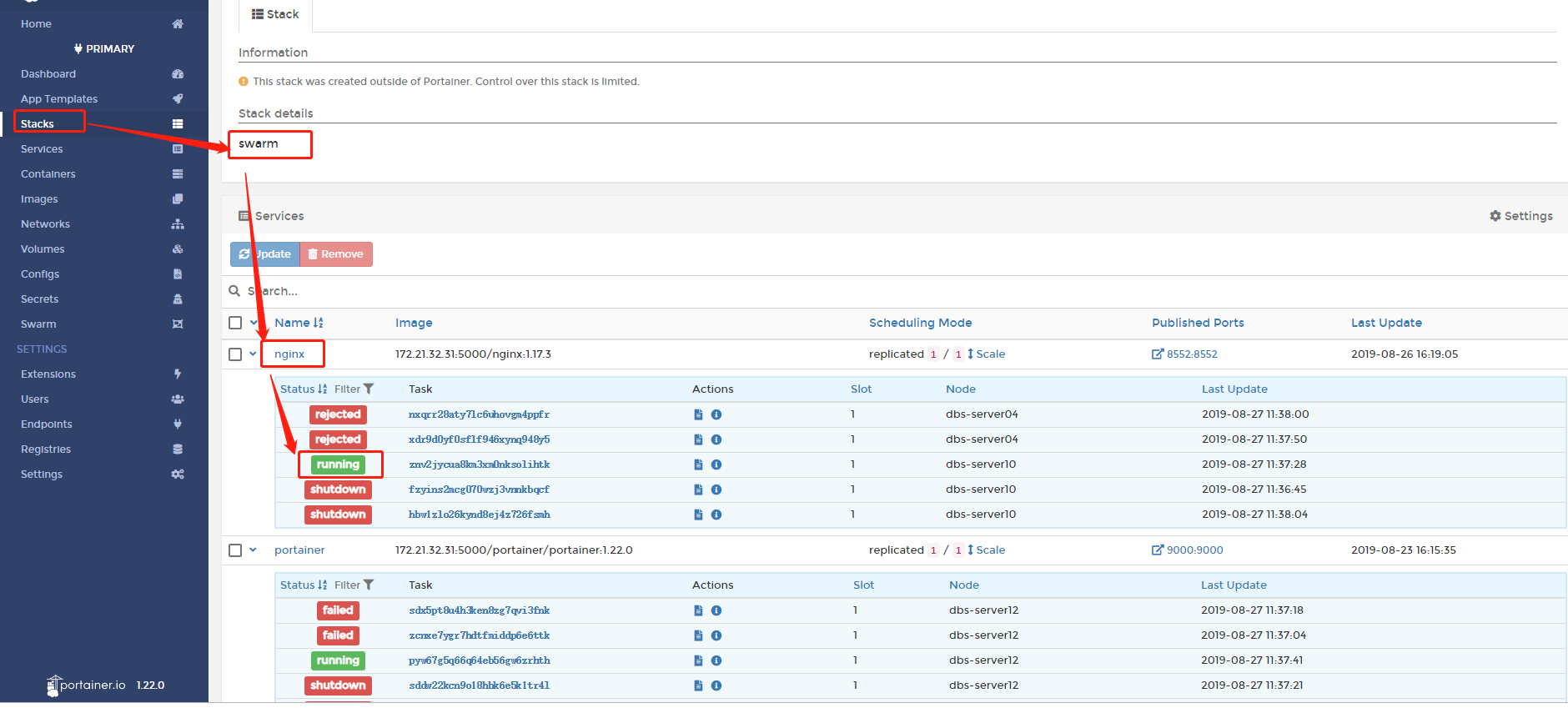
* **配置文件**(路径: /opt/nginx)



注：上图标注部分替换为事件集群IP地址

* **Docker创建代理服务**

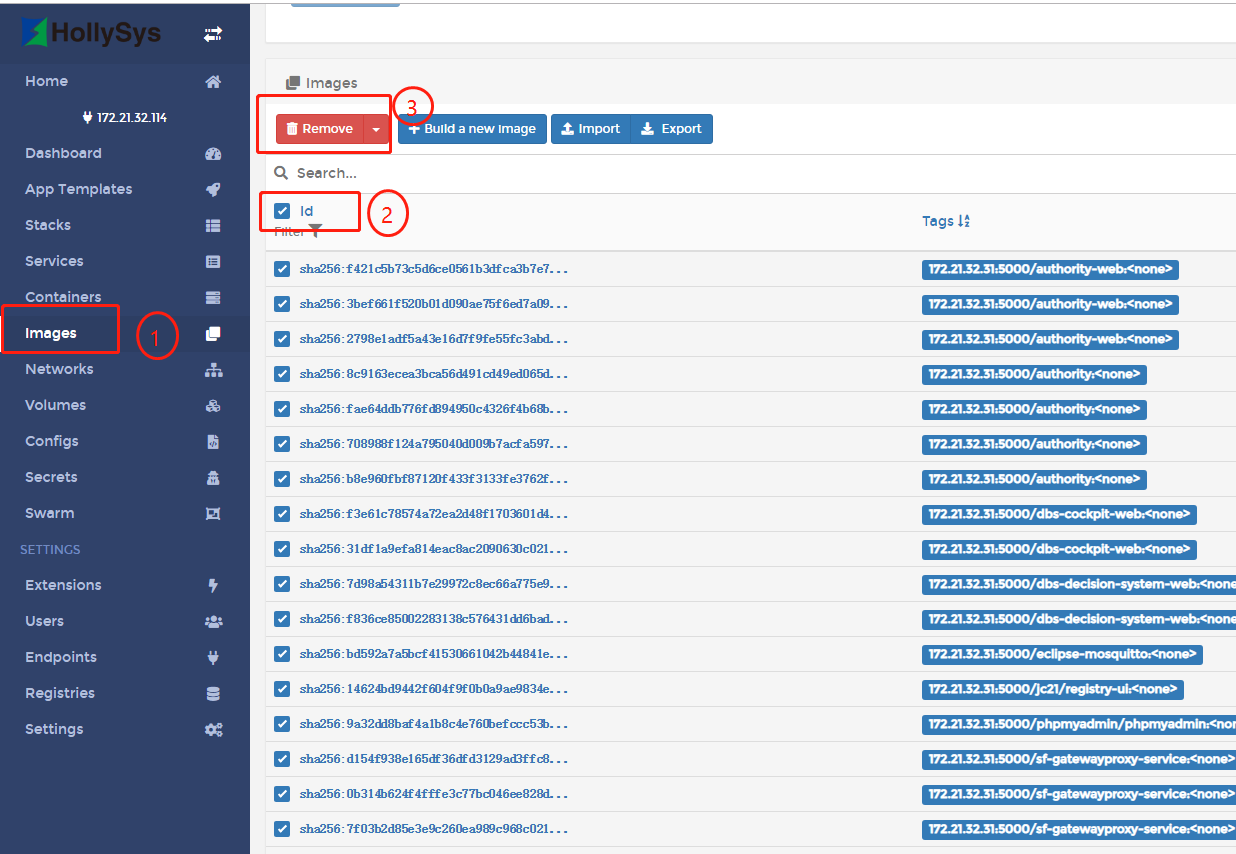
|  |
| --- |
| docker service create \  --name nginx \  --publish 8552:8552 \  --replicas=1 \  --constraint 'node.role == manager' \  --mount type=bind,src=/opt/nginx/nginx.conf,dst=/etc/nginx/nginx.conf \  --container-label com.docker.stack.namespace=swarm \  --label com.docker.stack.namespace=swarm \  nginx:1.17.3 |



**注:查看Nginx状态为running即为成功**

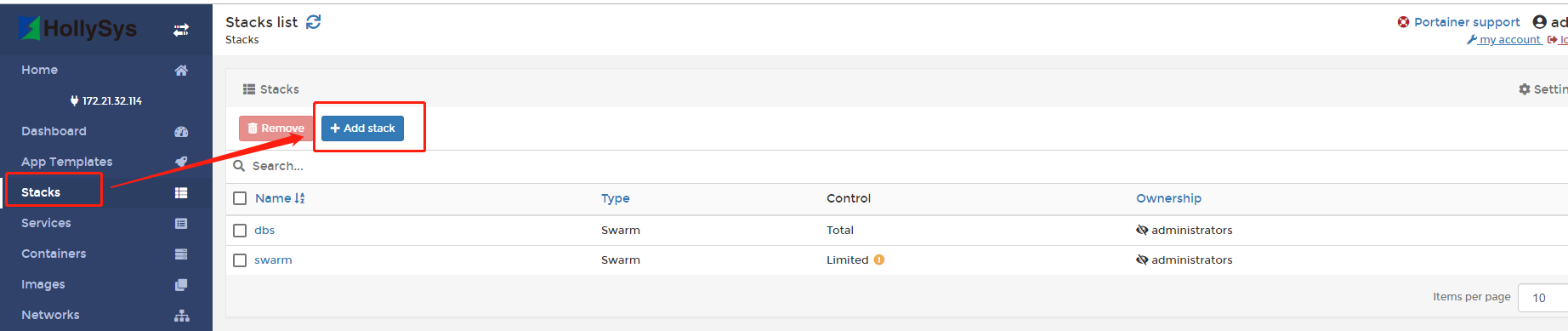
1. Docker集群镜像清理

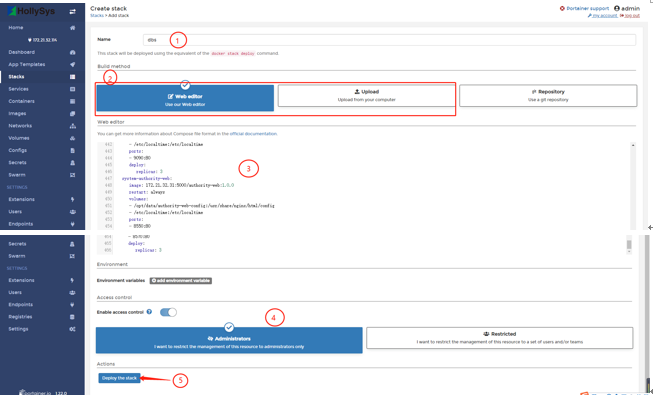
*在110、106、114服务器上均操作*



清理镜像保证从docker私库拉取最新服务镜像

1. Docker集群服务器创建
2. 创建stack





**注：**

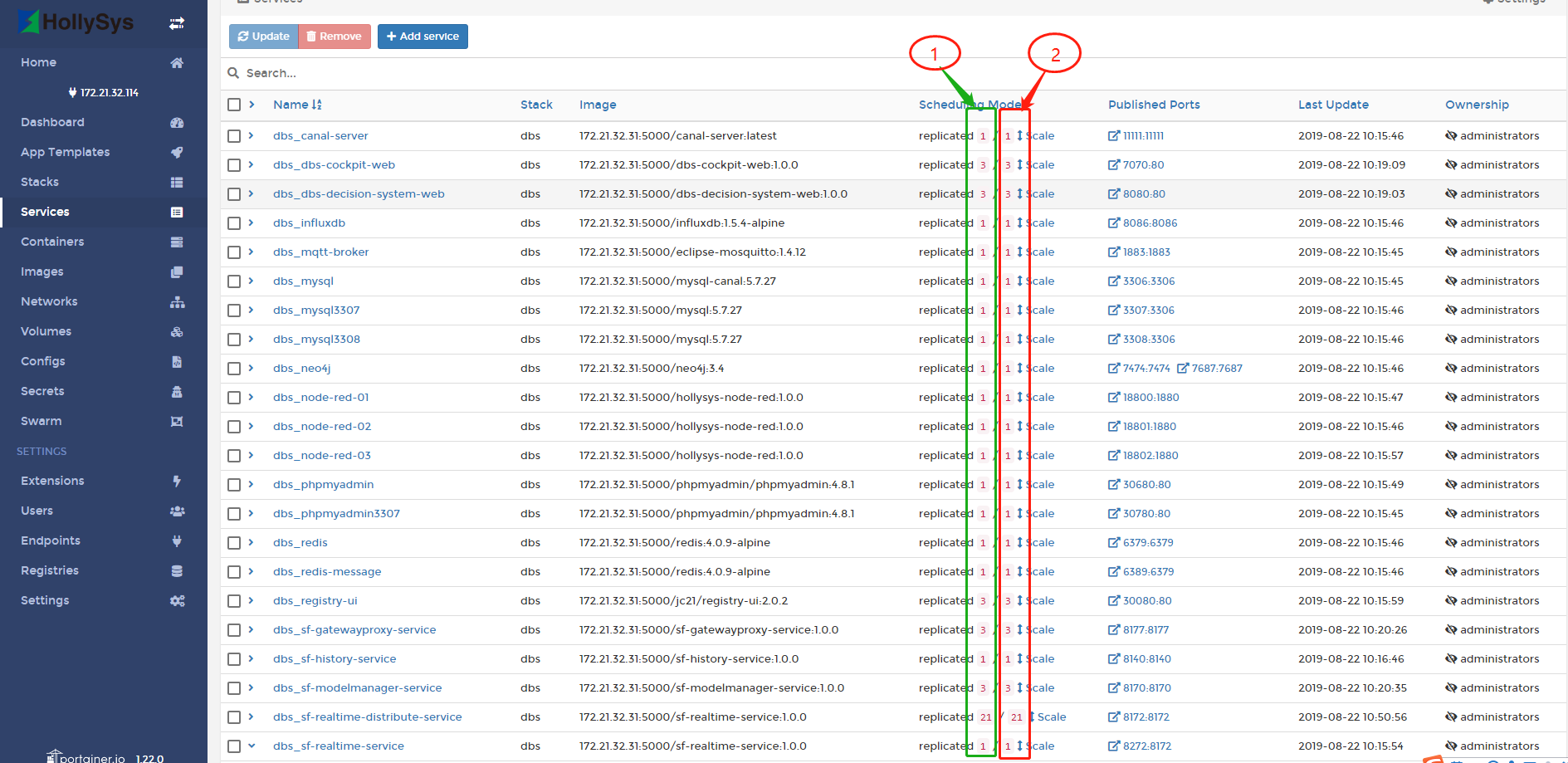
**标记①: 集群域名称，将会产生同一网络域IP地址，保证各服务域名正常通信**

**标记②：选择web编辑则需将步骤3中文件内容拷贝标记③中，选择Upload上传步骤3中文件**

**标记④：选择administrators模式**

**标记⑤：发布服务**

1. 查看应用服务服务

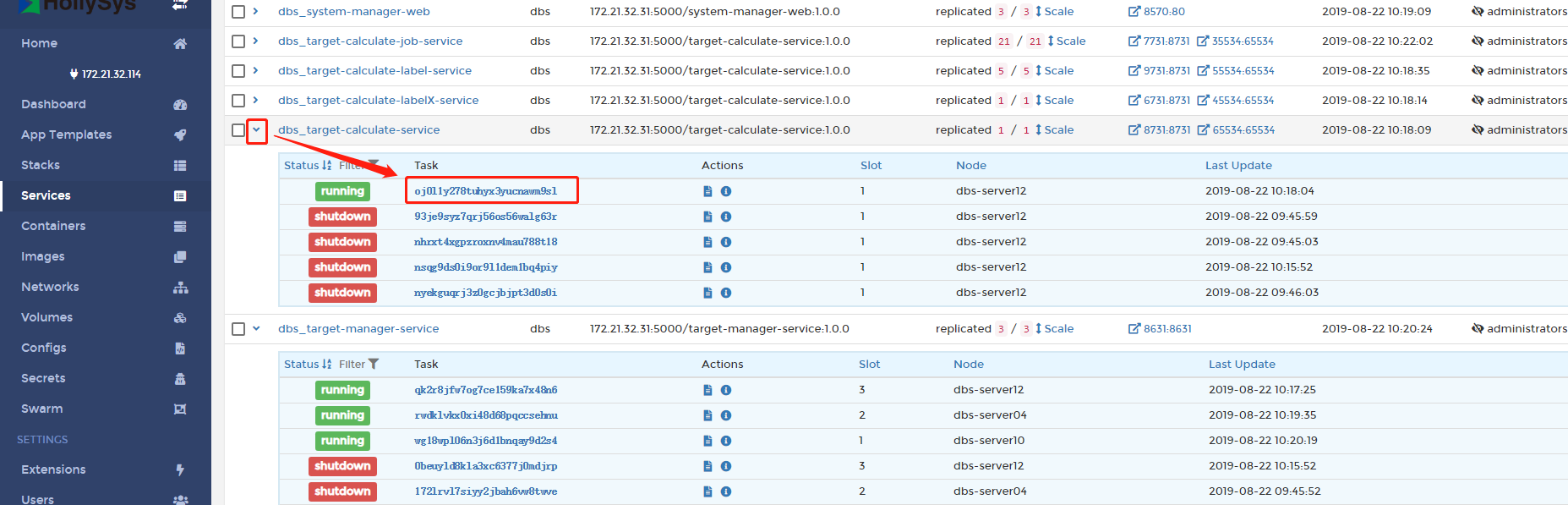


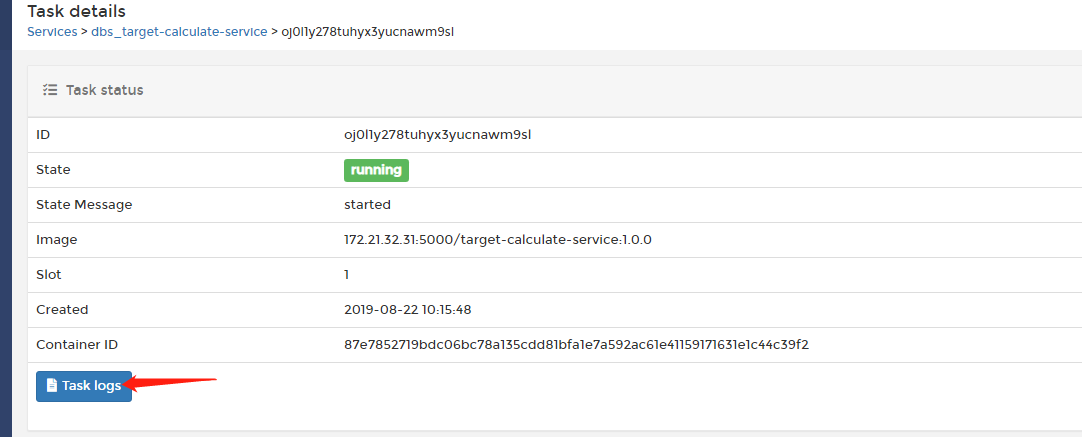
**注：**

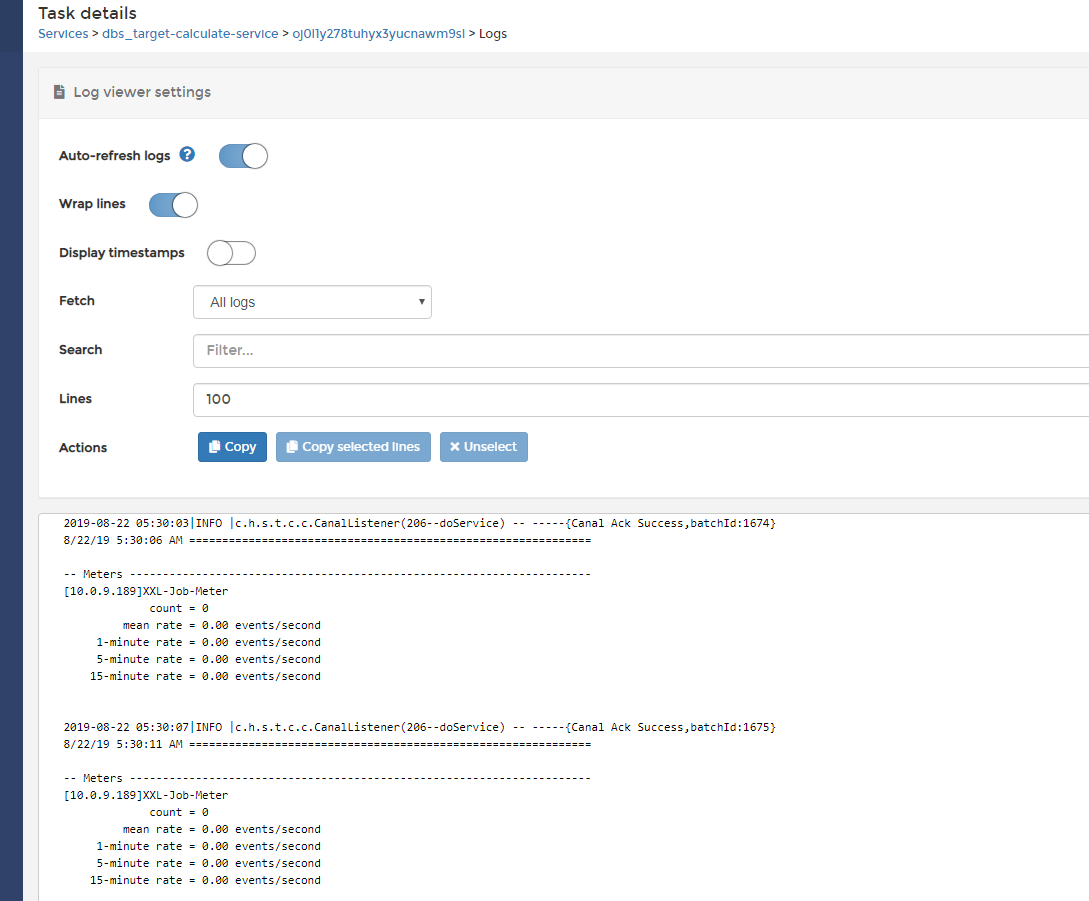
**标记①：正常状态的分片数即集群服务数(running：正常状态)**

**标记②：当前应用的集群总数**

1. 查看各应用节点日志

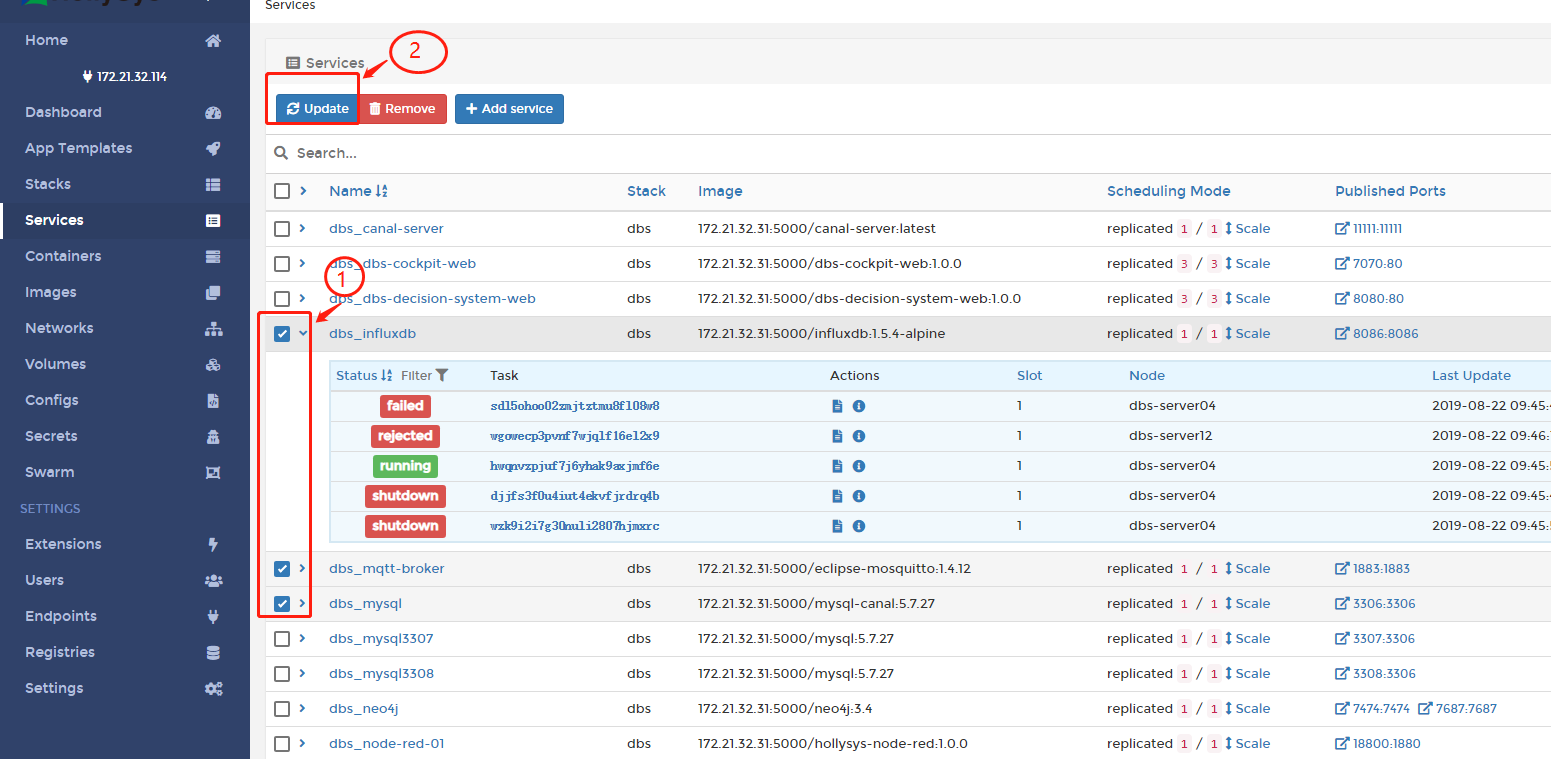






1. 重启服务

* 部分重启



* 全部重启

