

**EDRMS离线部署手册**

**操作系统 V5.18.0.1**

**手册**

**环境要求**

支持CentOS 7系列，CentOS 7.4及以上版本的系统，服务器系统建议选择最小化安装

root用户配有密码，且可以远程ssh连接

生产环境要求: CPU: 16核；内存: 32G；磁盘IO：6k；系统剩余空间25G；docker存储目录剩余空间80G

开启防火墙时，需要开放的端口：

tcp: 2377,7946 ; udp: 7946,4789  此为Docker Swarm所需端口号，必须开放

web端口: 一般为80/443/8083， 根据实际环境来决定开放哪个端口

日常维护(建议开放) tcp 9000, 30001,30002,30003,30004,30005

服务器时间同步

ntpdate ntp.aliyun.com （仅为一次同步，设置ntp参考<[http://v5.edoc2.com/wcm/km#wiki/detail/0bd91937-2233-4c8a-a317-d2eb2701319d](http://v5.edoc2.com/wcm/km" \l "wiki/detail/0bd91937-2233-4c8a-a317-d2eb2701319d)  1.6 配置NTP服务>）

关闭firewalld

systemctl stop firewalld （临时关闭）

systemctl disable firewalld （永久关闭）

关闭selinux

setenforce 0

sed -i 's#SELINUX=enforcing#SELINUX=disabled#' /etc/selinux/config

集群环境中,必须打开CACHEMODULE=redis环境变量

开始安装前务必关闭NetworkManager

systemctl stop NetworkManager

systemctl disable NetworkManager

v5.16.0.0之后的content服务默认打开了资源限制，如果当前系统CPU小于4核，内存小于8G那么content服务将无法启动。**生产环境一定要开启**，测试环境可以去掉compose文件中的资源限制

在安装之前如果服务器有配置主机名的请参照以下规范检查是否符合标准

服务器主机名规范:

1. 不能以数字开头，大小写字母都可以，推荐使用小写字母

2. 不能有除 “-”和“.”之外的标点符号

3. 不能包含空格

注意: 离线部署默认情况下没有开启服务日志收集。

请参考<http://v5.edoc2.com/preview.html?fileid=2950399> <7 配置ECM日志收集>

5.18.0.0版本不再推荐使用主服务器集群中间件单机这种部署架构，后续版本将废弃此架构如果5.18.0.0要使用这种部署架构，请一定确保中间件磁盘的性能。

**已知问题说明**

* v5.18.0.1 不支持低版本升级，需要重新部署
* 配置达梦数据库支持，会出现索引失败问题情况，重启服务可以解决
* OCR部署需要手动部署，参考此文第4节OCR 部署，部署OCR 服务耗时较长

**注意事项**

如果需要配置达梦数据库支持，在 envfile.env 里添加如下配置

离线包中的达梦镜像仅用作测试，如客户使用达梦数据库请安装收费版

DatabaseType=4  
DatabaseServerName=dm  
DatabaseServerPort=5236  
DatabaseUserName=SYSDBA  
DatabasePassword=1qaz2WSX1qaz2WSX

# 单机部署

1.1 下载edrms离线包并上传至系统中

curl -o edrms.tar.xz [http://download.edoc2.com:5999/edrms/v5.18.0.1/edrms.tar.xz](http://download.edoc2.com:5999/edrms/v5.18.0.0/edrms.tar.xz)

1.2 下载docker安装包

curl -o docker-ce.rpm <http://download.edoc2.com:5999/requirements/docker-ce/20.10.5/docker-20.10.5-1.el7.x86_64.rpm>

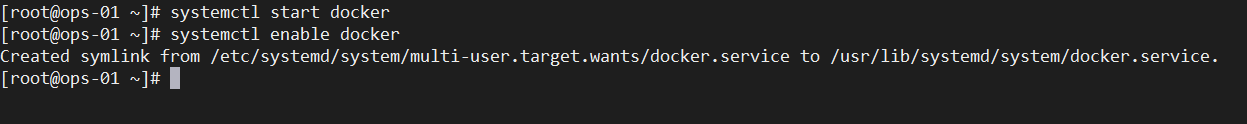
1.3安装docker

rpm -Uvh docker-ce.rpm

1.4启动docker

systemctl start docker

systemctl enable docker



1.5 确定docker是否启动成功

执行systemctl status docker

图片包含 图形用户界面

描述已自动生成

1.6 解压edrms的压缩包

tar xfv edrms.tar.xz

1.7. 进入edrms 目录

执行命令：cd edrms

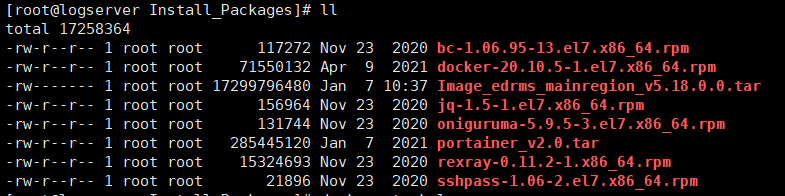
1.8 导入edrms、portainer与APM镜像

执行命令：

cd Install\_Packages

docker image load -i Image\_edrms\_mainregion\_v5.18.0.1.tar

docker image load -i portainer\_v2.0.tar



1.9 安装清理内存缓存依赖包

rpm -Uvh bc-1.06.95-13.el7.x86\_64.rpm

手机屏幕的截图

描述已自动生成

1.10 初始化swarm 环境

执行命令：

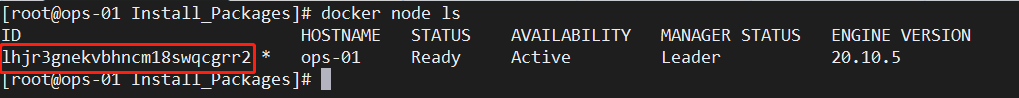
docker swarm init --advertise-addr=本机IP地址

文本

描述已自动生成

1.11 修改节点标签

执行命令: docker node ls 查看nodeid



执行命令：（注意：nodeid根据实际情况更改）

docker node update --label-add nodetype=InDrive  nodeid

docker node update --label-add nodelabels=Middleware nodeid

docker node update --label-add noderedis=true nodeid

docker node update --label-add isapm=true nodeid

docker node update --label-add nodeportainer=Portainer nodeid

文本

中度可信度描述已自动生成

1.12 拷贝compose文件到/opt目录，

进入edrms安装包的Template目录

cd edrms/Template

cp docker-compose-mainregion.yml  /opt/

cp middleware-master-standalone.yml  /opt/

cp portainer-agent-stack.yml  /opt/

cp envfile.env /opt/

注: 5.18.0.1离线安装必须执行以下命令

sed -i -r '/\s{2}redis/,/^$/ s/(.\*)nodelabels == Middleware/\1noderedis == true/g' /opt/middleware-master-standalone.yml

1.13 查看docker-compose-mainregion.yml 与middleware-master-standalone.yml，创建edrms需要的目录，默认为/home/edoc2/macrowing/edoc2v5/data

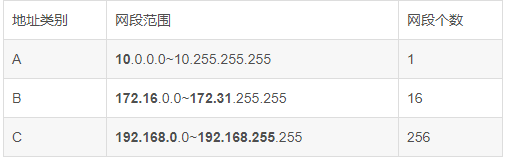
执行命令：

mkdir -p /home/edoc2/macrowing/edoc2v5/data

cd /home/edoc2/macrowing/edoc2v5/data

mkdir edoc2Docs  mysql  rabbitmq  es redis dbbackup edrmscore

1.14 创建docker网络（标红处可以指定docker容器的网段，默认为10.0.1.0段，如果与客户网络不冲突，可以不写，只能使用如图下面的三类私有地址）



docker network create macrowing --driver overlay  --subnet 172.28.0.0/24

1.15 修改系统参数

echo "fs.file-max = 65536"  >> /etc/sysctl.conf

echo "vm.max\_map\_count=262144" >> /etc/sysctl.conf

echo "vm.min\_free\_kbytes=1024000" >> /etc/sysctl.conf

echo "net.core.somaxconn=2048" >> /etc/sysctl.conf

echo "vm.swappiness=0" >> /etc/sysctl.conf

echo "net.ipv4.ip\_local\_port\_range=1024 65300" >> /etc/sysctl.conf

echo "\* - nofile 655300" >> /etc/security/limits.conf

sysctl -p

ulimit -HSn 65536

1.16 拷贝内存清理脚本

mkdir  -p /opt/scripts

进入edrms解压目录中执行:

cp scripts/clean-cache.sh /opt/scripts/

1.17 添加定时任务

echo '1 \*/1 \* \* \*  /bin/bash /opt/scripts/clean-cache.sh &> /dev/null' >> /var/spool/cron/root

1.18 启动中间件

执行命令：

docker stack deploy -c /opt/middleware-master-standalone.yml  middleware

1.19 启动edrms

执行命令：

docker stack deploy -c docker-compose-mainregion.yml edrms

1.20 启动portainer

执行命令：

docker  stack  deploy  -c  portainer-agent-stack.yml   portainer

# 集群部署(应用集群、中间件单机模式)

集群节点要求：集群环境需要至少三台主机，超过三台要保证集群中的主节点为奇数个

集群存储要求：需要挂载nas  nfs等共享存储

注意：本文档采用NFS存储

**集群部署，一定要对存储做IOPS测试，达不到咱们要求，不能部署，参考手册**

<http://v5.edoc2.com/preview.html?fileid=2456930>

2.1 下载edrms离线包并上传至系统中

curl -o edrms.tar.xz [http://download.edoc2.com:5999/edrms/v5.18.0.1/edrms.tar.xz](http://download.edoc2.com:5999/edrms/v5.18.0.0/edrms.tar.xz)

2.2 下载docker安装包(所有节点都要执行)

curl -o docker-ce.rpm <http://download.edoc2.com:5999/requirements/docker-ce/20.10.5/docker-20.10.5-1.el7.x86_64.rpm>

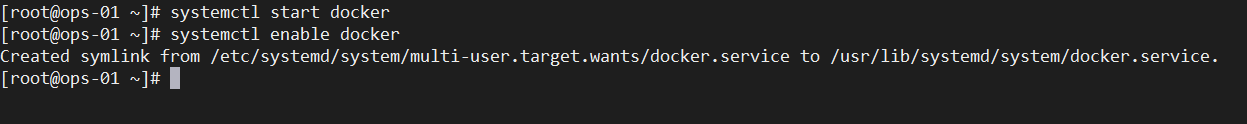
2.3安装docker(所有节点都要执行)

rpm -Uvh docker-ce.rpm

2.4启动docker

systemctl start docker

systemctl enable docker



2.5 确定docker是否启动成功

执行systemctl status docker

图片包含 图形用户界面

描述已自动生成

2.6 解压edrms的压缩包

tar xfv edrms.tar.xz

2.7. 进入edrms目录

执行命令：cd edrms

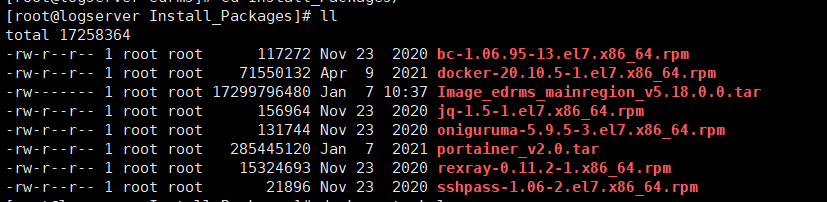
2.8 导入edrms与portainer镜像

执行命令：

cd Install\_Packages

docker image load -i Image\_edrms\_mainregion\_v5.18.0.1.tar

docker image load -i portainer\_v2.0.tar



2.9 初始化swarm环境（集群中任意一节点）

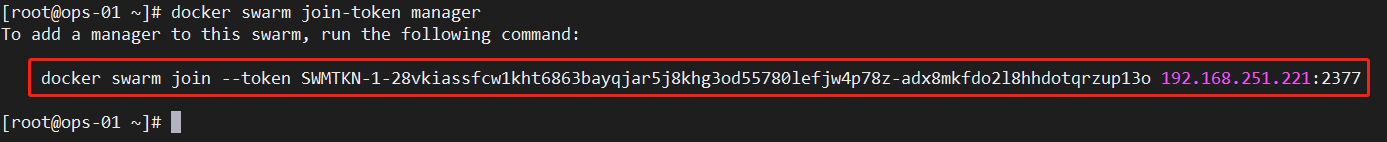
docker swarm init --advertise-addr 192.168.251.221#本机IP地址

文本

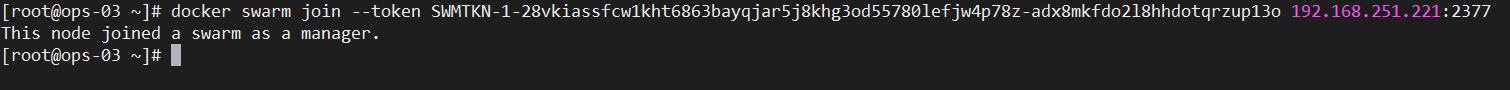
描述已自动生成

2.10 获取swarm的manager token （执行第九步操作节点）

docker swarm join-token manager



2.11 复制上一步红框内的命令至其他两节点执行



2.12 查看swarm节点状态

docker node ls

文本

中度可信度描述已自动生成

2.13 修改节点标签（任意一主节点执行）

nodeid可以从上一步命令获得，依次次替换三个节点的ID，下面每个命令要执行三次

docker  node  update  --label-add  nodetype=InDrive   nodeid

docker  node  update  --label-add  nodelabels=Middleware   nodeid

docker  node  update  --label-add  isapm=true   nodeid

2.14 添加redis服务标签 （任意一主节点执行）只执行一次

将nodeid 替换为运行redis的节点id，这样做可以保证redis的性能，同时也将失去HA功能

docker node update --label-add noderedis=true nodeid

2.15 创建文件夹（共享存储挂载目录）

mkdir /home/edoc2/macrowing/edoc2v5/data

cd /home/edoc2/macrowing/edoc2v5/data  #共享存储挂载目录

mkdir  edoc2Docs  es  rabbitmq  mysql   dbbackup edrmscore

2.16 创建redis数据目录 redis数据目录不要使用共享存储

mkdir /datavol/redis -p

2.17 修改系统参数

echo "fs.file-max = 65536"  >> /etc/sysctl.conf

echo "vm.max\_map\_count=262144" >> /etc/sysctl.conf

echo "vm.min\_free\_kbytes=1024000" >> /etc/sysctl.conf

echo "net.core.somaxconn=2048" >> /etc/sysctl.conf

echo "vm.swappiness=0" >> /etc/sysctl.conf

echo "net.ipv4.ip\_local\_port\_range=1024 65300" >> /etc/sysctl.conf

echo "\* - nofile 655300" >> /etc/security/limits.conf

sysctl -p

ulimit -HSn 65536

2.18 拷贝compose文件到/opt目录，在Template目录下

cp docker-compose-mainregion.yml  /opt/

cp middleware-master-standalone.yml  /opt/

cp docker-compose-inbiz.yml /opt/

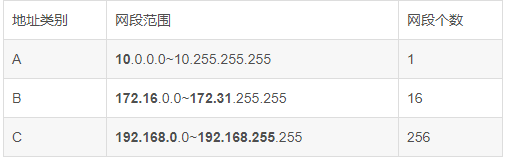
cp portainer-agent-stack.yml  /opt/

cp envfile.env /opt/

注: 5.18.0.1离线安装必须执行以下命令

sed -i -r '/\s{2}redis/,/^$/ s/(.\*)nodelabels == Middleware/\1noderedis == true/g' /opt/middleware-master-standalone.yml

2.19 创建docker网络（标红处可以指定docker容器的网段，默认为10.0.1.0段，如果与客户网络不冲突，可以不写，只能使用如图下面的三类私有地址）



docker  network  create  macrowing  --driver  overlay  --subnet 172.28.0.0/24

2.20 重启docker

systemctl restart docker

2.21 启动中间件

执行命令：

cd /opt

docker stack deploy -c middleware-master-standalone.yml  middleware

2.22 启动edrms

执行命令：

docker stack deploy -c docker-compose-mainregion.yml  edrms

2.23 启动inbiz

docker stack deploy -c docker-compose-inbiz.yml inbiz

2.24 启动portainer

docker  stack  deploy  -c  portainer-agent-stack.yml   portainer

2.25 添加清理内存(所有节点执行)

参考<http://v5.edoc2.com/wcm/km#wiki/detail/b44bc30d-5a20-4d2e-ad5f-a0c90b61c093>

# 集群部署(中间件与应用集群模式)

3.1 下载edrms离线包并上传至系统中(所有节点)

curl -o edrms.tar.xz [http://download.edoc2.com:5999/edrms/v5.18.0.1/edrms.tar.xz](http://download.edoc2.com:5999/edrms/v5.18.0.0/edrms.tar.xz)

3.2 下载docker安装包(所有节点)

curl -o docker-ce.rpm <http://download.edoc2.com:5999/requirements/docker-ce/20.10.5/docker-20.10.5-1.el7.x86_64.rpm>

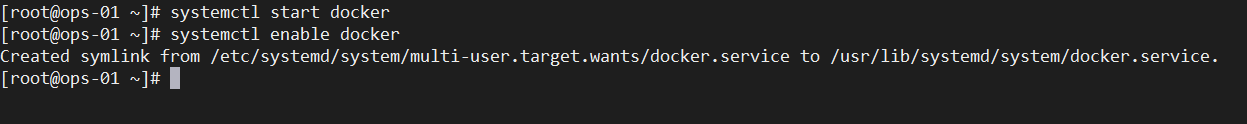
3.3安装docker(所有节点)

rpm -Uvh docker-ce.rpm

3.4启动docker (所有节点)

systemctl start docker

systemctl enable docker



3.5 确定docker是否启动成功 (所有节点)

执行systemctl status docker

图片包含 图形用户界面

描述已自动生成

3.6 解压edrms的压缩包（所有节点）

tar xfv edrms.tar.xz

3.7. 进入edrms目录(所有节点)

执行命令：cd edrms

3.8 导入edrms与portainer镜像(所有节点)

执行命令：

cd Install\_Packages

docker image load -i Image\_edrms\_mainregion\_v5.18.0.1.tar

docker image load -i portainer\_v2.0.tar

3.9 初始化swarm环境（集群中任意一节点）

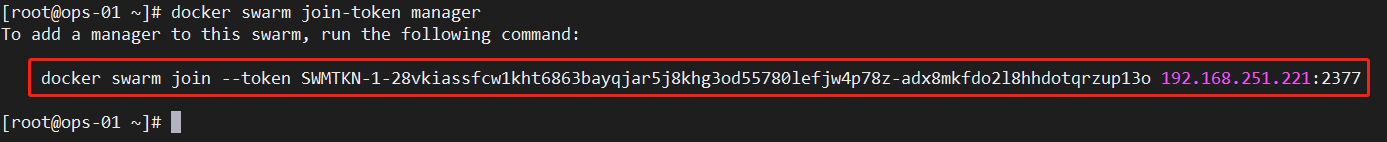
docker swarm init --advertise-addr 192.168.251.221 #本机IP地址

文本

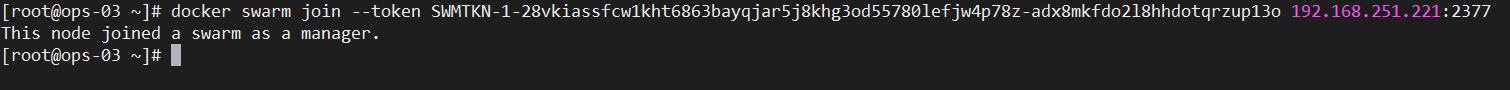
描述已自动生成

3.10 获取swarm的manager token （执行第九步操作节点）

docker swarm join-token manager



3.11 复制上一步红框内的命令至其他两节点执行



3.12 查看swarm节点状态

docker node ls

文本

中度可信度描述已自动生成

3.13 修改节点标签（任意一主节点执行）

nodeid可以从上一步命令获得，依次次替换三个节点的ID，下面每个命令要执行三次

docker  node  update  --label-add  nodetype=InDrive   nodeid

docker node update --label-add isapm=true nodeid

3.14 添加mysql运行的标签（node\_id分别替换为三个节点id）

docker  node  update  --label-add  mysqllabels=mysql0   node\_id1

docker  node  update  --label-add  mysqllabels=mysql1   node\_id2

docker  node  update  --label-add  mysqllabels=mysql2   node\_id3

3.15  添加redis运行的标签

docker  node  update  --label-add  redislabels=redis1  node\_id1

docker  node  update  --label-add  redislabels=redis2  node\_id2

docker  node  update  --label-add  redislabels=redis3  node\_id3

3.16 添加rabbitmq运行的节点标签

docker  node  update  --label-add  rabbitmqlabels=rabbitmq1  node\_id1

docker  node  update  --label-add  rabbitmqlabels=rabbitmq2  node\_id2

docker  node  update  --label-add  rabbitmqlabels=rabbitmq3  node\_id3

3.17添加es运行的节点标签

docker  node  update  --label-add  eslabels=es1  node\_id1

docker  node  update  --label-add  eslabels=es2  node\_id2

docker  node  update  --label-add  eslabels=es3  node\_id3

3.18 创建文件夹edoc2Docs文件夹（共享存储）

mkdir /home/edoc2/macrowing/edoc2v5/data

cd /home/edoc2/macrowing/edoc2v5/data

mkdir  edoc2Docs

3.19 创建中间件文件夹(/data目录根据实际情况修改，非共享存储，所有节点都要执行一遍)

mkdir /data

mkdir  redis  es  rabbitmq  mysql  dbbackup edrmscore

3.20 修改系统参数

echo "fs.file-max = 65536"  >> /etc/sysctl.conf

echo "vm.max\_map\_count=262144" >> /etc/sysctl.conf

echo "vm.min\_free\_kbytes=1024000" >> /etc/sysctl.conf

echo "net.core.somaxconn=2048" >> /etc/sysctl.conf

echo "vm.swappiness=0" >> /etc/sysctl.conf

echo "net.ipv4.ip\_local\_port\_range=1024 65300" >> /etc/sysctl.conf

echo "\* - nofile 655300" >> /etc/security/limits.conf

sysctl -p

ulimit -HSn 65536

3.21 拷贝compose文件到/opt目录，在ecm的Template目录下

cp docker-compose-mainregion.yml  /opt/

cp middleware-master-standalone.yml  /opt/

cp docker-compose-apm.yml /opt/

cp portainer-agent-stack.yml  /opt/

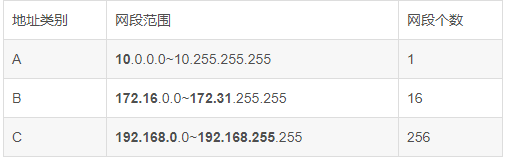
cp envfile.env /opt/

3.22 修改middleware-master-cluster.yml中的存储路径，改为步骤19中创建的路径

文本

描述已自动生成

3.23 创建docker网络（标红处可以指定docker容器的网段，默认为10.0.1.0段，如果与客户网络不冲突，可以不写，只能使用如图下面的三类私有地址）



docker  network  create  macrowing  --driver  overlay  --subnet 172.28.0.0/24

3.24 重启docker

systemctl restart docker

3.25

修改envfile.env

如果redis配了集群，需要添加下面的环境变量

RedisMasterName=mymaster  
RedisClusterModel=SENTINEL  
RedisPort=26379,26379,26379  
RedisHostName=redis1,redis2,redis3

其他中间件配置集群需要添加对应的环境变量，可以参考如下链接内容：

[http://v5.edoc2.com/inbiz/wcm/default#wiki/repositoryId/fac4adbb-69a7-4398-ac11-b5c91d33e961/catalogId/186893f8-603a-48b4-8029-2e2918fdbff7](http://v5.edoc2.com/inbiz/wcm/default" \l "wiki/repositoryId/fac4adbb-69a7-4398-ac11-b5c91d33e961/catalogId/186893f8-603a-48b4-8029-2e2918fdbff7)

3.26 启动中间件

cd /opt

docker stack deploy -c middleware-master-cluster.yml   middleware

3.27 启动edrms

执行命令：

docker stack deploy -c docker-compose-mainregion.yml  edrms

3.28 启动portainer

docker  stack  deploy  -c  portainer-agent-stack.yml   portainer

3.29添加内存清理（所有节点）

参考<http://v5.edoc2.com/wcm/km#wiki/detail/b44bc30d-5a20-4d2e-ad5f-a0c90b61c093>

# OCR 部署

1、安装docker，注意：安装docker版本和之前ecm集群中版本一致

    docker --version

2、将装好的docker以work角色加入到swarm集群中

    docker swarm join-token worker

3、添加节点标签(nodeID为OCR节点id)

    docker node update --label-add nodetype=INWISE nodeID

4、OCR服务器下载inwise镜像包并解压导入镜像

下载镜像包：

curl -o inwise.tar.xz [http://download.edoc2.com:5999/ecm/v5.18.0.1/inwise.tar.xz](http://download.edoc2.com:5999/ecm/v5.18.0.0/inwise.tar.xz)

解压导入镜像：

tar -xvf inwise.tar.xz

docker image load -i inwise.tar

5、创建OCR文件存储目录

    mkdir -p /home/edoc2/macrowing/edoc2v5/data/inwise

6、下载compose服务配置文件

下载配置文件：

curl -o /opt/docker-compose-inwise.yml [http://download.edoc2.com:5999/ecm/v5.18.0.1/docker-compose-inwise.yml](http://download.edoc2.com:5999/ecm/v5.18.0.0/docker-compose-inwise.yml )

注意：OCR服务默认以admin为用户、以edoc2为密码连接edoc2服务，如果需要配置其他用户，需要修改环境变量 EDOC2\_USER和EDOC2\_PASSWORD 这两个变量配置的用户名密码需要在ECM系统组织中存在，并且配置的用户名密码要和ECM系统中一致

7、启动服务、验证功能

docker stack deploy -c docker-compose-inwise.yml inwise