**离线部署手顺**

[安装系统 2](#_Toc521503582)

[修改ip 2](#_Toc521503583)

[开启端口 2](#_Toc521503584)

[安装依赖包（/mnt/packages/） 2](#_Toc521503585)

[pkgs-common 2](#_Toc521503586)

[pkgs-postgres 2](#_Toc521503587)

[pkgs-postgis 2](#_Toc521503588)

[pkgs-pgrouting 4](#_Toc521503589)

[pkgs-rabbitmq 4](#_Toc521503590)

[安装软件（/mnt/app/） 4](#_Toc521503591)

[postgres+配置 4](#_Toc521503592)

[postgis 5](#_Toc521503593)

[pgrouting 5](#_Toc521503594)

[redis 6](#_Toc521503595)

[rabbitmq 6](#_Toc521503596)

[mysql 6](#_Toc521503597)

[mongodb 7](#_Toc521503598)

[创建pg数据库 8](#_Toc521503599)

[tomcat(/opt/java) 8](#_Toc521503600)

[软件部署 8](#_Toc521503601)

[部署软件 8](#_Toc521503602)

[系统服务 8](#_Toc521503603)

[开机启动 8](#_Toc521503604)

[定时任务 9](#_Toc521503605)

## 安装系统

### 修改ip

vi /etc/sysconfig/network-scripts/ifcfg-ens33

BOOTPROTO=static

DEFROUTE=yes

IPADDR=192.168.1.181

GATEWAY=192.168.1.1

NETMASK=255.255.255.0

DNS1=192.168.1.1

ONBOOT=yes

service network restart

echo "nameserver 8.8.8.8" > /etc/resolv.conf

### 开启端口

firewall-cmd --permanent --add-port=5432/tcp(postgresql)

firewall-cmd --permanent --add-port=80/tcp

firewall-cmd --permanent --add-port=15672/tcp (rabbitmq)

firewall-cmd --permanent --add-port=6379/tcp (redis)

firewall-cmd --permanent --add-port=28017/tcp (mongodb)

firewall-cmd --permanent --add-port=8090/tcp(geoserver、LocateServer、map、WXServer)

firewall-cmd --permanent --add-port=8091/tcp(bison、ConfigServer、LocateDCServer)

firewall-cmd --permanent --add-port=8088/tcp(小程序公众号)

firewall-cmd --reload

## 安装依赖包（/mnt/packages/）

### pkgs-common

rpm -i \*.rpm

### pkgs-postgres

rpm -i \*.rpm

### pkgs-postgis

rpm -i \*.rpm

#proj4

./autogen.sh

./configure --prefix=/home/pgsql/plugin/proj

make&make install

echo "/home/pgsql/plugin/proj/lib" > /etc/ld.so.conf.d/proj-4.9.3.conf

ldconfig

#geos（版本3.7+）

./autogen.sh

./configure --prefix=/home/pgsql/plugin/geos

make&make install

echo "/home/pgsql/plugin/geos/lib" > /etc/ld.so.conf.d/geos-3.6.1.conf

ldconfig

#gdal

./configure --prefix=/home/pgsql/plugin/gdal

make&make install

echo "/home/pgsql/plugin/gdal/lib" > /etc/ld.so.conf.d/gdal-2.1.2.conf

ldconfig

#json-c

./configure --prefix=/home/pgsql/plugin/json-c

make&make install

echo "/home/pgsql/plugin/json-c/lib" > /etc/ld.so.conf.d/json-c-0.12.1.conf

ldconfig

#编辑ld.so.conf文件

vim /etc/ld.so.conf

/usr/local/postgresql/lib

/home/pgsql/plugin/geos/lib

/home/pgsql/plugin/proj/lib

/home/pgsql/plugin/gdal/lib

/home/pgsql/plugin/json-c/lib

ldconfig

#配置依赖环境变量

echo "PGHOME=/usr/local/postgresql/" >> /etc/profile

echo "export PGHOME" >> /etc/profile

echo "export PGDATA=/pgdata/data" >> /etc/profile

echo "export LD\_LIBRARY\_PATH=/home/pgsql/plugin/geos/lib:/home/pgsql/plugin/proj4/lib:/home/pgsql/plugin/json-c/lib:/home/pgsql/plugin/gdal/lib:$PGHOME/lib:/lib64:/usr/lib64:/usr/local/lib64:/lib:/usr/lib:/usr/local/lib:$LD\_LIBRARY\_PATH" >> /etc/profile

echo "export PATH=\$PATH:\$HOME/.local/bin:\$HOME/bin:\$PGHOME/bin" >> /etc/profile

echo "export MANPATH=\$PGHOME/share/man:\$MANPATH" >>/etc/profile

echo "export PATH=\$PATH:/mnt/packages/pkgs-pgrouting/cmake-3.11.0-rc4-Linux-x86\_64/bin" >> /etc/profile

source /etc/profile

### pkgs-pgrouting

rpm -i \*.rpm

# cmake

//解压后只需配置环境变量

# gmp

./configure && make && make check && make install

# mpfr

./configure --with-gmp-include=/usr/local/include --with-gmp-lib=/usr/local/lib && make && make check && make install

# mpc

./configure && make && make install

# cgal

cmake . && make && sudo make install

### pkgs-rabbitmq

rpm -i erlang-20.0.1-1.el6.x86\_64.rpm

rpm -i socat-1.7.3.2-2.el7.x86\_64.rpm

## 安装软件（/mnt/app/）

### postgres+配置

tar -zxvf postgresql-9.6.8.tar.gz

cd postgresql-9.6.8

./configure --prefix=/usr/local/postgresql --with-ossp-uuid

make

make install

useradd postgres -m

echo postgres:postgres | chpasswd

mkdir -p /pgdata

chown -R postgres /pgdata

cd /usr/local/postgresql/bin

su postgres

./initdb -D "/pgdata" -U postgres（user：postgres）

sed -i "s/#listen\_addresses = 'localhost'/listen\_addresses = '\*'/g" /pgdata/postgresql.conf

sed -i "s/#port = 5432/port = 5432/g" /pgdata/postgresql.conf

echo "host all all 0.0.0.0/32 md5" >> /pgdata/pg\_hba.conf

cp -pir /mnt/app/postgresql-9.6.8/contrib/start-scripts/linux /etc/init.d/postgresql（开机自启）

chmod +x /etc/init.d/postgresql

sed -i 's/prefix=\/usr\/local\/pgsql/prefix=\/usr\/local\/postgresql/g' /etc/init.d/postgresql

sed -i 's/PGDATA="\/usr\/local\/pgsql\/data"/PGDATA="\/pgdata"/g' /etc/init.d/postgresql

chkconfig --add postgresql

chkconfig postgresql on

service postgresql start

su postgres

psql -f /mnt/shell/changepgpsword.sql

service postgresql restart

### postgis

tar -zxvf postgis-2.4.4dev.tar.gz

cd postgis-2.4.4dev

./configure --prefix=/home/pgsql/plugin/postgis --with-pgconfig=/

usr/local/postgresql/bin/pg\_config --with-geosconfig=/home/pgsql/plugin/geos/bin/geos-config --with-

gdalconfig=/home/pgsql/plugin/gdal/bin/gdal-config --with-projdir=/home/pgsql/plugin/proj

make

make install

cd extensions/postgis

make clean

make

sudo make install

cd ../postgis\_topology

make clean

make

sudo make install

cd ../postgis\_sfcgal

make clean

make

sudo make install

cd ../address\_standardizer

make clean

make

sudo make install

cd ../postgis\_tiger\_geocoder

make clean

make

sudo make install

### pgrouting

tar -zxvf pgrouting-2.4.1.tar.gz

cd pgrouting-2.4.1

mkdir build && cd build && cmake .. && make && sudo make install

cp libpgrouting-2.4.so /mnt/app/pgrouting-2.4.1/build/lib

cp \*.sql /usr/local/postgresql/share/extension/

### redis

tar -zxvf redis-4.0.2.tar.gz

cd redis-4.0.2

make

cd src

make install PREFIX=/usr/local/redis

mkdir /usr/local/redis/etc

cp ../redis.conf /usr/local/redis/etc/

sed -i 's/daemonize no/daemonize yes/g' /usr/local/redis/etc/redis.conf

sed -i 's/# requirepass foobared/requirepass Mote12345/g' /usr/local/redis/etc/redis.conf

sed -i 's/bind 127.0.0.1/# bind 127.0.0.1/g' /usr/local/redis/etc/redis.conf

echo "export PATH=\$PATH:/usr/local/redis/bin" >> /etc/profile

source /etc/profile

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

### rabbitmq

rpm -ivh rabbitmq-server-3.7.5-1.el6.noarch.rpm

cp /usr/share/doc/rabbitmq-server-3.7.5/rabbitmq.config.example /etc/rabbitmq/ rabbitmq.config

service rabbitmq-server start

sed -i 's/\%\% {tcp\_listeners\, \[5672\]}/ {tcp\_listeners\, \[5672\]}/g' /etc/rabbitmq/rabbitmq.config

sed -i 's/\%\% {loopback\_users\, \[\]}\,/ {loopback\_users\, \[\]}\,/g' /etc/rabbitmq/rabbitmq.config

sed -i '/{loopback\_users\, \[\]}\,/a\ {log\_levels\, \[{connection\, warning}\, {channel\, warning}\]}' /etc/rabbitmq/rabbitmq.config

rabbitmq-plugins enable rabbitmq\_management

rabbitmqctl add\_user test 123456

rabbitmqctl set\_permissions -p "/" test '.\*' '.\*' '.\*'

rabbitmqctl set\_user\_tags test administrator

service rabbitmq-server restart

chkconfig rabbitmq-server on

### mysql

tar -zxvf mysql-5.6.33-linux-glibc2.5-x86\_64.tar.gz

cp -pir mysql-5.6.33-linux-glibc2.5-x86\_64 /usr/local/mysql

groupadd mysql

useradd -g mysql mysql

cd /usr/local/mysql/

mkdir data

chown -R mysql:mysql ./

./scripts/mysql\_install\_db --user=mysql --datadir=/usr/local/mysql/data/

cp support-files/mysql.server /etc/init.d/mysqld

chmod 755 /etc/init.d/mysqld

vim /etc/init.d/mysqld

basedir=/usr/local/mysql/

datadir=/usr/local/mysql/data

chkconfig --add mysqld（开机自启）

chkconfig mysqld on

\cp /mnt/conf/my.cnf /etc/my.cnf

echo "export PATH=\$PATH:/usr/local/mysql/bin" >> /etc/profile

source /etc/profile

service mysqld start

mysqladmin -u root password 'Mote12345'

mysql -uroot -pMote12345

use mysql;

grant all privileges on \*.\* to 'root'@'%' identified by 'Mote12345';

flush privileges;

exit

mysql -uroot -pMote12345 bison < /mnt/shell/bison.sql

mysql -uroot -pMote12345 beacon < /mnt/shell/beacon.sql

### mongodb

tar -zxvf mongodb-linux-x86\_64-3.2.6.tgz

mv mongodb-linux-x86\_64-3.2.6 /usr/local/mongodb

echo "export PATH=\$PATH:/usr/local/mongodb/bin" >> /etc/profile

source /etc/profile

cd /usr/local/mongodb

mkdir -p data/db

mkdir log

mkdir conf

\cp /mnt/conf/mongodb.conf /usr/local/mongodb/conf/

mongod -f mongodb.conf

mongo <<EOF

use admin;

db.createUser({user: "admin",pwd: "Mote12345",roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]});

use location;

db.createUser(

{

user: "root",

pwd: "Mote12345",

roles: [

{ role: "readWrite", db: "location" },

{ role: "dbAdmin", db: "location" }

]

}

);

db.person\_collection.ensureIndex({"deviceId":1});

db.person\_collection.ensureIndex({"beginTime":1},{expireAfterSeconds:5184000});

db.asset\_collection.ensureIndex({"deviceId":1});

db.asset\_collection.ensureIndex({"beginTime":1},{expireAfterSeconds:5184000});

use admin;

db.shutdownServer();

exit;

EOF

sed -i "s/#auth/auth/g" conf/mongodb.conf

mongod -f conf/mongodb.conf

### 创建pg数据库&数据导入

su postgres

createdb mote\_basemap

psql -d mote\_basemap -U postgres -f /mnt/shell/insertpgdata.sql

exit

### tomcat(/opt/java)

tar -xf jdk-8u181-linux-x64.tar.gz

mkdir /opt/java

mv jdk1.8.0\_181 /opt/java/

echo "export JAVA\_HOME=/opt/java/jdk1.8.0\_181" >> /etc/profile

echo "export CLASSPATH=.:\$JAVA\_HOME/lib/tools.jar:\$JAVA\_HOME/lib/dt.jar" >> /etc/profile

echo "export PATH=\$JAVA\_HOME/bin:\$PATH" >> /etc/profile

source /etc/profile

tar -zxvf apache-tomcat-7.0.67.tar.gz

mv apache-tomcat-7.0.67 /opt/java/tomcat\_8090

sed -i 's/<Connector port="8080" protocol="HTTP\/1.1"/<Connector port="8090" protocol="HTTP\/1.1"/g' conf/server.xml

./startup.sh

## 软件部署

### 部署软件

### 系统服务&开机启动（/etc/rc.d/init.d）

/etc/rc.d/init.d下加入启动服务文件（/conf/）

postgresql已

redis：

chmod +x redis

chkconfig --add redis（开机自启）

rabbimq已

mongodb：

chmod +x mongodb

chkconfig --add mongodb（开机自启）

mysql已

tomcat：

vim /opt/java/tomcat\_8090/bin/catalina.sh

export CATALINA\_HOME=/opt/java/tomcat\_8090

export CATALINA\_BASE=/opt/java/tomcat\_8090

export CATALINA\_TMPDIR=/opt/java/tomcat\_8090/temp

export JAVA\_HOME=/opt/java/jdk1.8.0\_181

chmod +x tomcat\_8090

chkconfig --add tomcat\_8090（开机自启）

### 定时任务

crontab -e

> 10 0 \* \* \* /mnt/shell/del\_catalina.sh ====》每天凌晨10分执行清理

> 0 \* \* \* \* /mnt/shell/check\_tomcat.sh ====》每小时执行监控

service crond restart