### 私有化步骤：

按f2进入bios，在boot中选择boot首先启动项为U盘启动（generic flash disk 8.07），在save&exit中选择Save Changes and Reset.

<https://blog.csdn.net/weixin_41999594/article/details/112853316>

修该启动盘路径：sdb4，按enter

### 安装linux：

准备centos7启动盘：

<https://jingyan.baidu.com/article/a65957f42b151124e77f9b68.html>

安装centos7:

<https://feiutech.blog.csdn.net/article/details/78684534>

注意：安装好之后，记得再次进入bios,设置启动项为硬盘。

### Centos用户密码：

root用户：

用户：root 密码：nfxy2022

nfxy用户：

用户：nfxy 密码：nfxy2022

### 查看linux内核版本：uname -a

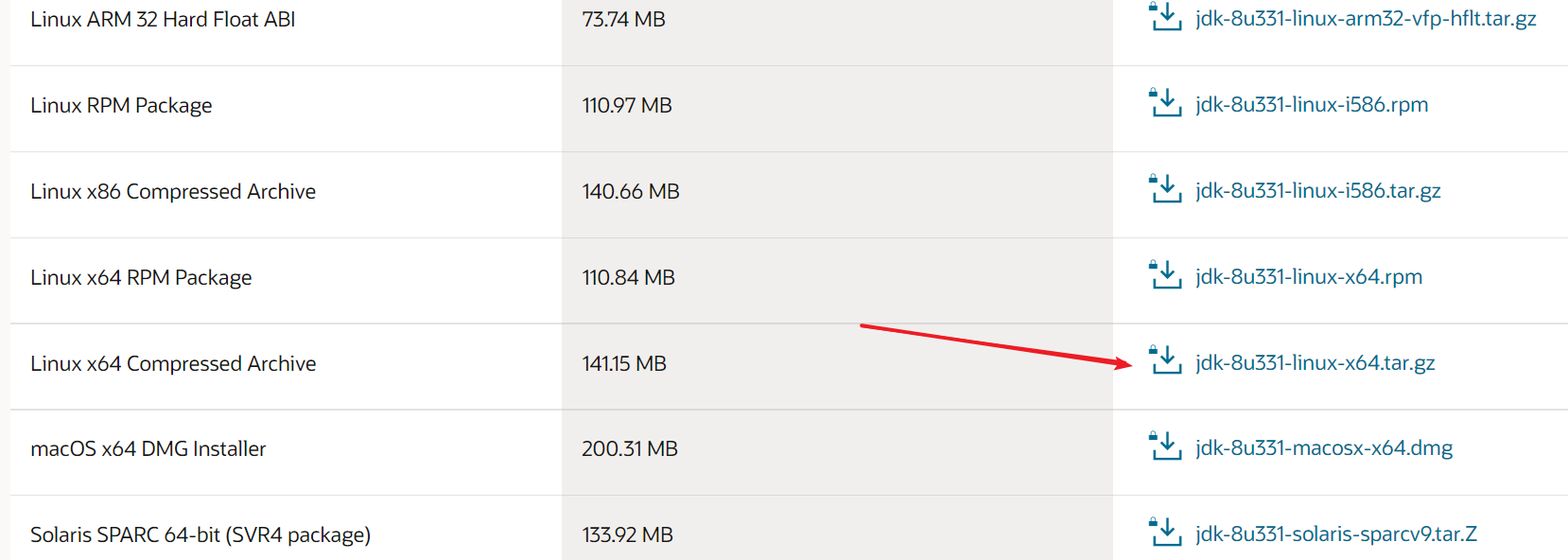
### 安装解压工具：yum install -y unzip zip

### 安装传输工具：yum -y install lrzsz

### Jdk安装：

1：下载jdk

<https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html>



2：上传到：/usr/local

3：解压：tar -zxvf

4：重命名为jdk1.8

5：添加环境变量：vi /etc/profile，末尾加上：

export JAVA\_HOME=/usr/local/jdk1.8

export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin

6：让环境变量生效：source /etc/profile

7：查看是否安装成功：java -version

### 安装nginx：

#### 下载地址：<https://nginx.org/en/download.html>

#### 安装流程

1：安装步骤

<https://blog.csdn.net/qq_37345604/article/details/90034424>

2：检查gcc版本：gcc -v

3：没有安装gcc的话，安装gcc：yum -y install gcc

4：pcre、pcre-devel安装：yum install -y pcre pcre-devel

5：zlib安装：yum install -y zlib zlib-devel

6：安装openssl：yum install -y openssl openssl-devel

7：在/usr/local创建java目录：mkdir java

8：上传nginx到java

9：解压：tar -zxvf

10：进入解压后的nginx目录，依次执行以下三个命令：

./configure

make

make install

11：如需要配置nginx，通过以下命令配置：

vi /usr/local/nginx/conf/nginx.conf

12：启动nginx，进入安装目录/usr/local/nginx，进入sbin：

./nginx

13：查看nginx启动：ps -ef | grep nginx

14：外界访问ngixn

# 查询端口是否开放

firewall-cmd --query-port=80/tcp

# 开放80端口

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

# 修改配置后需要重启防火墙

firewall-cmd --reload

#### 配置nginx开机自启：

1:先停止nginx: ./nginx -s stop

2：配置服务脚本：vi /usr/lib/systemd/system/nginx.service

3：服务脚本内容：

[Unit]

Description=nginx - web server

After=network.target remote-fs.target nss-lookup.target

[Service]

Type=forking

PIDFile=/usr/local/nginx/logs/nginx.pid

ExecStartPre=/usr/local/nginx/sbin/nginx -t -c /usr/local/nginx/conf/nginx.conf

ExecStart=/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf

ExecReload=/usr/local/nginx/sbin/nginx -s reload

ExecStop=/usr/local/nginx/sbin/nginx -s stop

ExecQuit=/usr/local/nginx/sbin/nginx -s quit

PrivateTmp=true

[Install]

WantedBy=multi-user.target

4：重新加载系统服务：systemctl daemon-reload

5：启动服务：systemctl start nginx.service

6：设置开机启动：systemctl enable nginx.service

### 安装reids：

#### 安装流程

安装步骤可参考：

<https://blog.csdn.net/qq_38066290/article/details/119708323>

<https://blog.csdn.net/qq_39715000/article/details/120696149>

1：下载redis：

<https://redis.io/download/#>

2：上传到：/usr/local

3：解压：tar -zxvf

4：重命名为 redis

5：进入/usr/local/reids目录，执行make

6：修改配置：vi /usr/local/redis/redis.conf

#设置后台启动

daemonize yes

#设置无访问限制

bind \* -::\*

#设置访问密码

requirepass 7EwjICun6l

7：进入到/usr/local/redis/src目录启动redis：./redis-server ../redis.conf

8：进入到/usr/local/redis/src目录停止redis：

**./redis-cli -a 7EwjICun6l shutdown**

**或者：kill -9 redis进程id**

**推荐使用：**

**先：./redis-cli -h ip地址 -p 6379**

**后：AUTH密码**

#### 开机自启动redis：

0：进入到/usr/local/redis/src目录停止redis：

**./redis-cli -a 7EwjICun6l shutdown**

1：新建一个系统服务文件：

vi /etc/systemd/system/redis.service

2：服务脚本内容：

[Unit]

Description=redis-server

After=network.target

#Wants=network-online.target

#After=network-online.target

[Service]

Type=forking

ExecStart=/usr/local/redis/src/redis-server /usr/local/redis/redis.conf

PrivateTmp=true

[Install]

WantedBy=multi-user.target

3：重载系统服务：systemctl daemon-reload

4：可以用下面这组命令来操作redis：

# 启动

systemctl start redis.service

# 停止

systemctl stop redis.service

# 重启

systemctl restart redis.service

# 查看状态

systemctl status redis.service

5：让redis开机自启：systemctl enable redis.service

### 安装mysql

#### 安装流程

可参考:

https://blog.csdn.net/weixin\_43451430/article/details/115553108?spm=1001.2101.3001.6650.2&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-2-115553108-blog-121515102.pc\_relevant\_aa&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-2-115553108-blog-121515102.pc\_relevant\_aa&utm\_relevant\_index=5

1：更新yum：yum update -y

2：安装wget工具：yum install -y wget，注意：先进入你想安装的位置，此后全部在此文件夹进行操作。

3：使用wget下载mysql yum源：

wget https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm

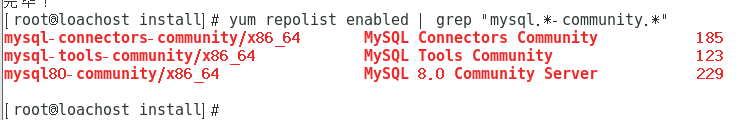
4：添加 mysql yum 源：

yum localinstall mysql80-community-release-el7-3.noarch.rpm -y

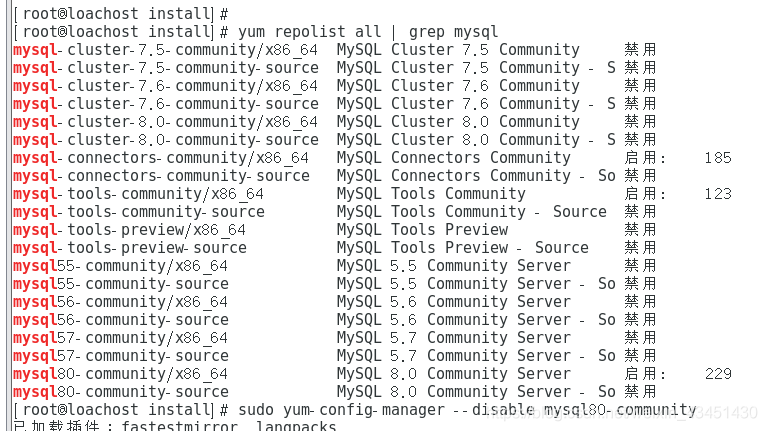
5：安装 yum 工具 yum-utils： yum install -y yum-utils

6：查看可用的mysql：

yum repolist enabled | grep "mysql.\*-community.\*"



7：查看所有的 mysql 版本：yum repolist all | grep mysql



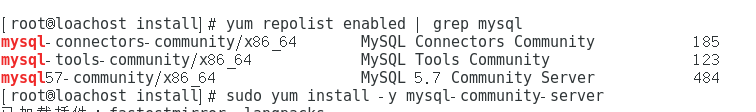
8：使用指定版本MySQL，假如我想使用MySQL5.7，那么我就需要先关闭MySQL8.0，再开启MySQL5.7：

关闭MySQL8.0：yum-config-manager --disable mysql80-community

开启MySQL5.7：yum-config-manager --enable mysql57-community

9：查看当前启用的MySQL版本：

yum repolist enabled | grep mysql



10：安装mysql：yum install -y mysql-community-server

安装过程失败可参考文档：

<https://www.isres.com/linux/4749.html>

10.1：先执行命令：

rm /etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

10.2：再执行命令：

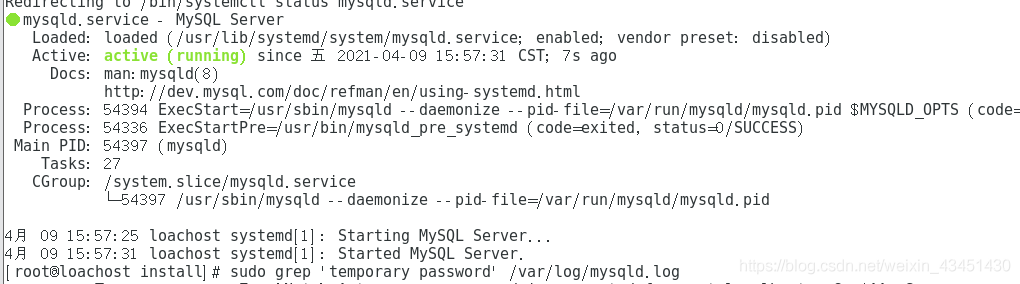
rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022

10.3：再重新安装mysql：

yum install -y mysql-community-server

11：启动MySQL：service mysqld start

12：查看MySQL服务状态：service mysqld status



13：查看Mysql初始化密码：

grep 'temporary password' /var/log/mysqld.log

14：使用初始密码进行登录：mysql -u root -p

15：重新设置root用户密码：

ALTER USER 'root'@'localhost' IDENTIFIED BY '密码';

16：设置MySQL远程连接，在sql里面设置：

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '密码' WITH GRANT OPTION;

17：刷新权限：FLUSH PRIVILEGES;

18：设置 MySQL开机启动：systemctl enable mysqld

注意：在linux里面设置，不在MySql里面配置。

19：开启3306防火墙后测试链接。

#### 另外：为MySql添加用户方法：

1：使用root用户进行登录：mysql -u root -p

2：创建新用户：

CREATE USER '用户'@'LOCALHOST' IDENTIFIED BY '密码';

3：为用户设置MySQL远程连接：

GRANT ALL PRIVILEGES ON \*.\* TO '用户'@'%' IDENTIFIED BY '密码' WITH GRANT OPTION;

4：刷新权限：FLUSH PRIVILEGES;

5：nfxy的服务器mysql用户的账号密码：

5.1：Root用户：

root

IFCYyueanmy123456!!

5.2：Ifcy用户：

Ifcy

Yuean!@#2019

6：部署项目出现失败的时候，可以按以下方式排查重新发布：

6.1：将配置文件的驱动改成：driver-class-name: com.mysql.jdbc.Driver

6.2：同时需要将uri加上：useSSL=false

7：如果出现ip可以连接，但是127.0.0.1或者localhost连接不上的话，可以在数据库为mysql的user表中删除Host为localhost、用户为对应用户的记录！

### 安装kafka：

#### 安装步骤

1：下载kafka：<https://mirrors.aliyun.com/apache/kafka/>

2：上传kafka安装包到/usr/local

3：解压：tar -zxvf

4：进入kafka解压目录，创建kafka-logs目录：mkdir kafka-logs

5：进入config目录，修改kafka的配置文件server.properties：

5.1：为保证可以远程访问Kafka，需修改以下两处配置：

# 去掉注释，允许外部访问

listeners=PLAINTEXT://0.0.0.0:9092

# 去掉注释并修改地址

advertised.listeners=PLAINTEXT://服务器的IP地址:9092

5.2：修改日志目录：log.dirs=/usr/local/Kafka的安装目录/kafka-logs

6：到kafka安装目录，启动Kafka:

6.1：先启动zookeeper：

./bin/zookeeper-server-start.sh -daemon config/zookeeper.properties

查看zookeeper启动状态：ps -ef | grep zookeeper

6.2：再启动Kafka:

./bin/kafka-server-start.sh -daemon config/server.properties

查看zookeeper启动状态：ps -ef | grep kafka

7：在Kafka安装目录下，查看topic列表：

./bin/kafka-topics.sh --list --bootstrap-server localhost:9092

8：在Kafka安装目录下，创建topic:

./bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic test

9：在Kafka安装目录下，发送消息：

./bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test

10：另开窗口，进入kafka目录下，创建消费者consumer消费消息：

./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning

11：在Kafka安装目录下，关闭Kafka:

11.1：先关闭kafka：

./bin/kafka-server-stop.sh config/server.properties

11.2：再关闭zookeeper：

./bin/zookeeper-server-stop.sh config/zookeeper.properties

12：开启9092防火墙后测试链接。

13：注意：部署到新项目时，要修改server.properties文件做如下修改：

**advertised.listeners=PLAINTEXT://服务器的IP地址:9092**

14：Kafka异常停止解决方法：

14.1：先停止Kafka，再meta.properties文件删除或重命名，然后再重启Kafka

14.2：14.1无效的话，先停止Kafka，然后删除日志文件中的\_\_consumer-offsets-\*文件，然后再重启Kafka

#### 设置Kafka开机自启动（zookeeper + kafka）：

**参考：**

**<https://blog.csdn.net/weixin_48484941/article/details/122191933>**

**<https://zhuanlan.zhihu.com/p/373855938>**

##### 一：zookeeper开机启动：

1：在/etc/init.d目录，新建zookeeper文件，注意以下几点：

（1）要设置java环境变量

（2）#chkconfig：2345（level） 20（启动顺序：越小越快） 90（停止顺序）

（3）start、stop、status、restart。都要有对应的命令

2：编辑zookeeper:

#!/bin/bash

export JAVA\_HOME=/usr/local/jdk1.8

export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin

#chkconfig:2345 20 90

#description:zookeeper

#processname:zookeeper

case $1 in

start)

/usr/local/kafka\_2.13-3.2.0/bin/zookeeper-server-start.sh -daemon /usr/local/kafka\_2.13-3.2.0/config/zookeeper.properties

;;

stop)

/usr/local/kafka\_2.13-3.2.0/bin/zookeeper-server-stop.sh /usr/local/kafka\_2.13-3.2.0/config/zookeeper.properties

;;

status)

jps

;;

restart)

/usr/local/kafka\_2.13-3.2.0/bin/zookeeper-server-stop.sh /usr/local/kafka\_2.13-3.2.0/config/zookeeper.properties

/usr/local/kafka\_2.13-3.2.0/bin/zookeeper-server-start.sh -daemon /usr/local/kafka\_2.13-3.2.0/config/zookeeper.properties

;;

\*)

echo "require start|stop|status|restart"

;;

esac

3：授权：chmod 777 /etc/init.d/zookeeper

4：添加到服务：chkconfig --add zookeeper

5：服务开机自动启动：chkconfig zookeeper on

6：查看服务：chkconfig --list

7：测试：service zookeeper start

##### 二：kafka开机自启：

1：在/etc/init.d目录，新建kafka文件，注意以下几点：

（1）要设置java环境变量

（2）#chkconfig：2345（level） 20（启动顺序：越小越快） 80（停止顺序），注意优先级要低于zookeeper；

（3）start、stop、status、restart。都要有对应的命令

2：编辑kafka:

#!/bin/bash

export JAVA\_HOME=/usr/local/jdk1.8

export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin

#chkconfig:2345 21 80

#description:kafka

#processname:kafka

case $1 in

start)

/usr/local/kafka\_2.13-3.2.0/bin/kafka-server-start.sh -daemon /usr/local/kafka\_2.13-3.2.0/config/server.properties

;;

stop)

/usr/local/kafka\_2.13-3.2.0/bin/kafka-server-stop.sh /usr/local/kafka\_2.13-3.2.0/config/server.properties

;;

status)

jps

;;

restart)

/usr/local/kafka\_2.13-3.2.0/bin/kafka-server-stop.sh /usr/local/kafka\_2.13-3.2.0/config/server.properties

/usr/local/kafka\_2.13-3.2.0/bin/kafka-server-start.sh -daemon /usr/local/kafka\_2.13-3.2.0/config/server.properties

;;

\*)

echo "require start|stop|status|restart"

;;

esac

3：授权：chmod 777 /etc/init.d/kafka

4：添加到服务：chkconfig --add kafka

5：服务开机自动启动：chkconfig kafka on

6：查看服务：chkconfig --list

7：测试：service kafka start

### 安装docker：

#### 安装步骤

1：查看linux内核，必须要在3.10以上：uname -a

2：如果之前安装过Docker，先卸载：

yum remove docker \

docker-client \

docker-client-latest \

docker-common \

docker-latest \

docker-latest-logrotate \

docker-logrotate \

docker-selinux \

docker-engine-selinux \

docker-engine \

docker-ce

3：安装yum工具：

yum install -y yum-utils \

device-mapper-persistent-data \

lvm2 --skip-broken

4：更新本地镜像源：

yum-config-manager \

--add-repo \

https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

sed -i 's/download.docker.com/mirrors.aliyun.com\/docker-ce/g' /etc/yum.repos.d/docker-ce.repo

yum makecache fast

5：安装Docker：yum install -y docker-ce

6：docker启动命令、停止命令、重启命令：

systemctl start docker # 启动docker服务

systemctl stop docker # 停止docker服务

systemctl restart docker # 重启docker服务

7：查看Docker运行状态：systemctl status docker

8：查看Docekr版本：docker -v

systemctl start docker # 启动docker服务

systemctl stop docker # 停止docker服务

systemctl restart docker # 重启docker服务

#### 设置docker开机自启动以及容器开机自启动，参考：

<https://blog.csdn.net/qq_37312838/article/details/115710209>

##### systemctl enable docker.service

### 安装mongodb

#### 安装步骤

1：下载mongo：docker pull mongo

2：创建一个文件夹，用作mongodb的数据目录挂载：

mkdir /usr/local/mongodb

3：安装mongo：

docker run -itd --name mongo -v /usr/local/mongodb:/data/db -e MONGO\_INITDB\_ROOT\_USERNAME=gdyaadmin -e MONGO\_INITDB\_ROOT\_PASSWORD=gdya1994 -p 27017:27017 mongo --auth

5：配置外部可以连接操作服务器中的 MongoDB：

5.1：进入mongo：docker exec -it mongo容器id或mongo容器名bash

5.2：更新源：apt-get update

5.3：安装vim：apt-get install vim

5.4：修改mongo的配置文件：vim /etc/mongod.conf.orig

5.4.1：将`bindIp: 127.0.0.1`注释掉；或者改成`bindIp: 0.0.0.0`。即可开启远程连接

5.4.2：开启权限认证

security：

authorization: enabled

6：在5.1执行之后，进入mongo命令行：mongo

7：切换到admin：use admin

8；认证：db.auth(‘gdyaadmin’, ‘gdya1994’)

9：显示数据库：show dbs

10：显示表：show tables

11：在执行6之后，退出mongo：exit

12：在11之后，退出mongo容器：exit

13：开放27017端口

14：使用navicat连接mongo：



15：新建gdya数据库，运行脚本进行数据导入

#### 开机自启动：docker update --restart=always 容器id或name

### 安装rabbitmq

#### 安装步骤

1：下载RabbitMQ：docker pull rabbitmq:3-management

2：安装RabbitMQ：

docker run \

-e RABBITMQ\_DEFAULT\_USER=gdyaadmin \

-e RABBITMQ\_DEFAULT\_PASS=gdya1994 \

--name mq \

--hostname mq1 \

-p 15672:15672 \

-p 5672:5672 \

-d \

rabbitmq:3-management

3：开放5672和15672端口

4：登录管理界面（15672）

#### 开机自启动：docker update --restart=always 容器id或name

### 安装花生壳（用于kafka的内网穿透）：

#### 安装步骤

参考：

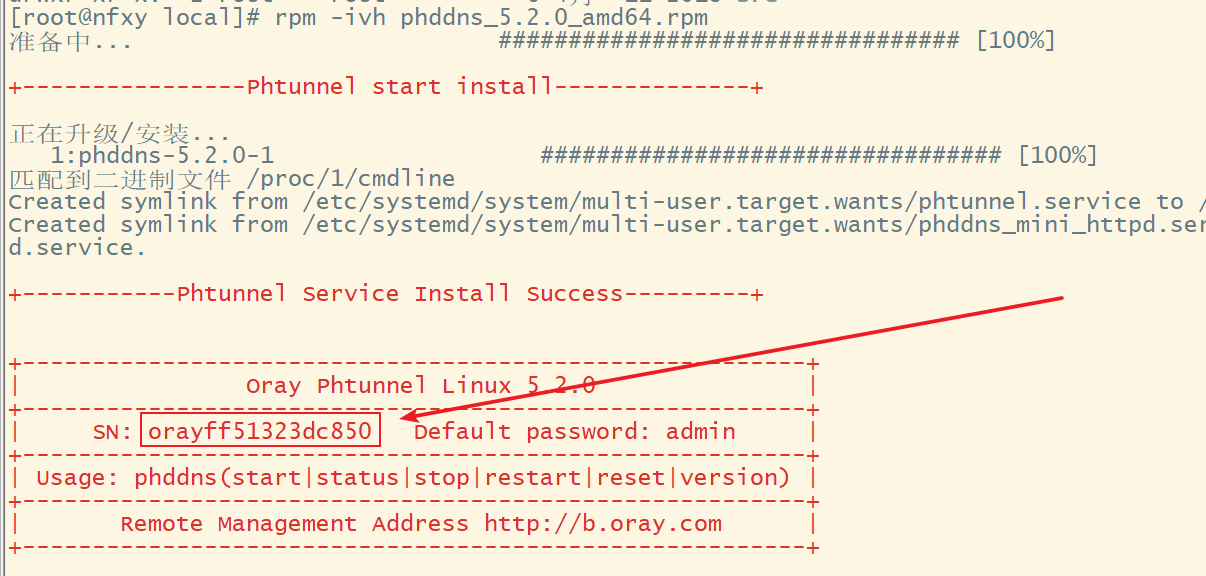
<https://service.oray.com/question/11630.html>

1：进入/usr/local目录：

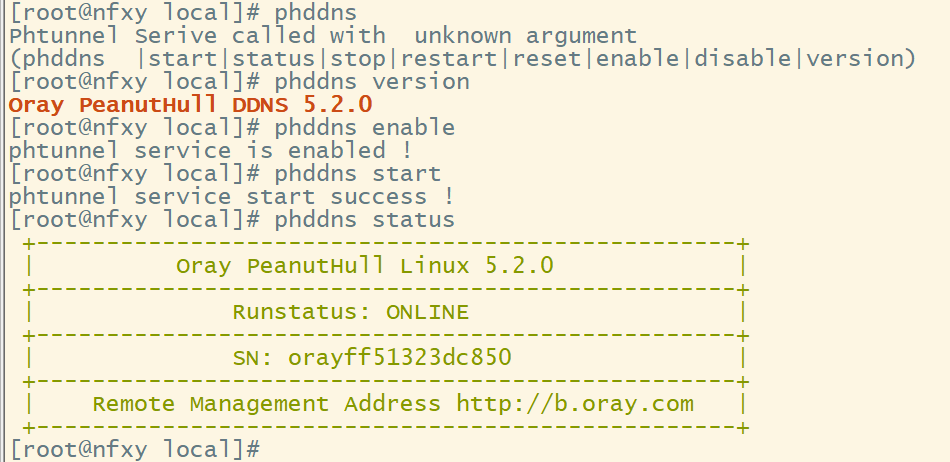
wget "https://down.oray.com/hsk/linux/phddns\_5.2.0\_amd64.rpm" -O phddns\_5.2.0\_amd64.rpm



2：安装：rpm -ivh phddns\_5.2.0\_amd64.rpm



3：设置开机自启动并启动：phddns enable 然后 phddns start



4：在登录管理平台：<http://b.oray.com> 使用sn方式登录

密码：admin

SN码：orayff51323dc850

5：首次登录，需先激活，使用密码激活：

账号：13560155884

密码：1234abcd

### 安装cpolar（app内网穿透用）:

#### Cpolar管理平台

管理后台：<https://dashboard.cpolar.com/login>

账号：[2278181701@qq.com](mailto:2278181701@qq.com) 密码：119920817hyh

#### 安装步骤

1： cpolar 安装（国内使用）：

curl -L https://www.cpolar.com/static/downloads/install-release-cpolar.sh | sudo bash

2：查看版本号：cpolar version

3：token认证，登录后台，查看自己的认证token，之后将token贴在命令行里：

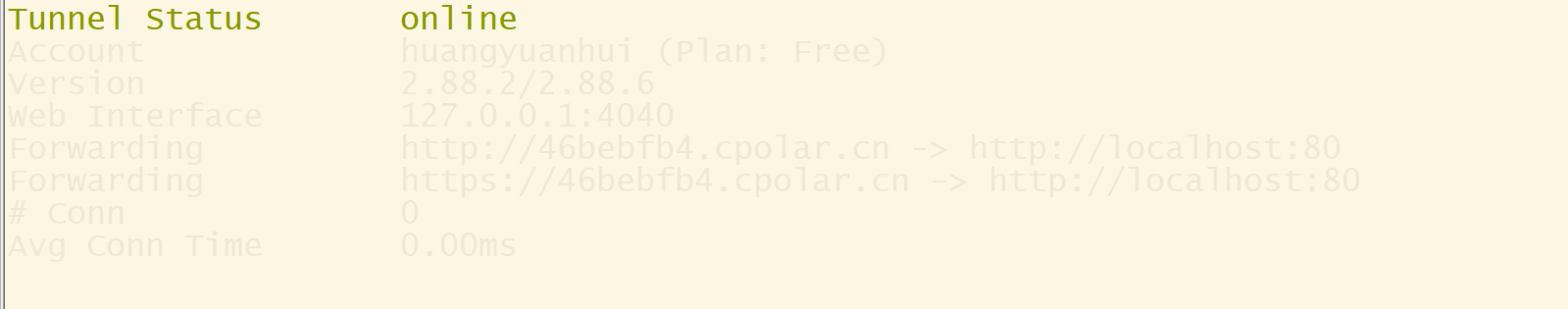
cpolar authtoken MmYwODNlMDAtMGZmNC00ZDk1LWFjODEtNzZhZjc2YjYyZGY4

4：测试内网穿透：进入安装目录：./usr/local/bin，穿透80端口：./cpolar http 80，因为cpolar分配的域名是随机的，可以设置在cploar管理后台一个自己的子域名：gznfxy，设置好后，命令为：

./cpolar http -subdomain=gznfxy 80

5：运行完上面命令之后就会生成下图所示的http域名和http域名以及对应的本地端口：

Account huangyuanhui (Plan: Free) http://gznfxy.cpolar.cn -> http://localhost:80



6：系统添加服务：systemctl enable cpolar

7：启动cpolar服务：systemctl start cpolar

8：查看服务状态：systemctl status cpolar

#### 配置cpolar开机启动：

1：首先在管理后台添加两个保留二级子域名：分别是：

gznfxy（广州南方学院，对应80端口）

gznfxyiot（广州南方学院iot，对应8767端口）

2：进入目录：/usr/local/etc/cpolar/cpolar.yml 修改cpolar配置文件

authtoken: MmYwODNlMDAtMGZmNC00ZDk1LWFjODEtNzZhZjc2YjYyZGY4

tunnels:

gznfxy:

proto: http

addr: "80"

subdomain: "gznfxy"

region: cn

gznfxyiot:

proto: http

addr: "8767"

subdomain: "gznfxyiot"

region: cn

3：测试：cpolar start-all

4：系统添加服务：systemctl enable cpolar

5：启动cpolar服务：systemctl start cpolar

6：查看服务状态：systemctl status cpolar

7：重启机器查看（比较慢，五分钟左右）

### 安装nacos（值守平台用）:

#### 安装步骤

1：上传或者下载nacos压缩包，本例上传至/usr/local/

2：然后解压

3：进入到解压后的目录

4：端口配置：Nacos的默认端口是8848，如果服务器的其它进程占用了8848端口，可以关闭该进程，或者也可以进入nacos的conf目录，修改配置文件application.properties中的端口

5：cd bin目录，运行命令启动Nacos：sh startup.sh -m standalone

#### 访问：http://IP地址:8848/nacos/

### 部署项目：

为了远程控制，需要在一台联网的工控机上安装xshell，以及向日葵，通过向日葵远程连接工控机，再通过工控机上的xshell登录服务器。最好没一台工控机都安装向日葵、xshell，以防万一。

/home/nfxy/java\_jars目录下，例如：

**nohup java -jar ifcy-business-ab-0.0.1-SNAPSHOT.jar >ifcy-business-ab.log 2>&1 &**

#### 开机自启动jar包：

1：进入/home/nfxy/java\_jars

2：创建自启动脚本：touch autoboot.sh

3：编辑自启动脚本：

export JAVA\_HOME=/usr/local/jdk1.8

export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin

cd /home/nfxy/java\_jars/

nohup java -jar ifcy-business-sys-0.0.1-SNAPSHOT.jar >ifcy-business-sys.log 2>&1 &

nohup java -jar wechat-platform-api-0.0.1-SNAPSHOT.jar >wechat.txt 2>&1 &

nohup java -jar ifcy-business-admin-0.0.1-SNAPSHOT.jar >ifcy-business-admin.log 2>&1 &

nohup java -jar ifcy-business-app-0.0.1-SNAPSHOT.jar >ifcy-business-app.log 2>&1 &

nohup java -jar ifcy-business-ab-0.0.1-SNAPSHOT.jar >ifcy-business-ab.log 2>&1 &

nohup java -jar ifcy-business-iot-0.0.1-SNAPSHOT.jar >ifcy-business-iot.log 2>&1 &

4：设置autoboot.sh可执行权限：chmod +x autoboot.sh

5：编辑/etc/rc.local：vi /etc/rc.loacl 末尾加上：

sh /home/nfxy/java\_jars/autoboot.sh

6：设置rc.local可执行权限，chmod +x /etc/rc.d/rc.local

#### 部署值守平台：

1：先在docker安装nginx :

1.1：拉取docker镜像：docker pull nginx

1.2：安装nginx容器：

docker run --name zsptnginx -p 8080:80 -v html:/usr/share/nginx/html -d nginx

1.3：开放8080端口

2：将项目上传至数据卷的html对应目录：

cd /var/lib/docker/volumes/html/\_data

3：解压：unzip html

4：将解压后目录里的文件mv到/var/lib/docker/volumes/html/\_data

5：开放7000端口

6：部署gateway、core\_services项目

#### Iot项目部署：

1：部署iot项目的jar包

2：开放8767端口

3：在cploar管理后台设置子域名：gznfxyiot

3：使用cploar设置子域名gznfxyiot：进入/usr/local/bin目录，执行命令：

./cploar http -subdomain=gznfxy 8767

4：在ctwing中设置对应iot设备的订阅地址

### 查看开机启动的服务：

参考：<https://blog.csdn.net/qq_37335220/article/details/124121398>

#### 查看开机自启动服务：systemctl list-unit-files | grep enable

#### 查看开机自启动服务：chkconfig --list

#### systemctl list-units --type=service

#### 产看运行的jar包详细信息：jps -l

### 要开机自启动的服务：

#### 服务器重启命令：reboot

#### Jdk：java -version

#### Nginx：ps -ef | grep nginx

#### Redis：ps -ef | grep redis

#### Mysql：查看数据库

#### Kafka(其中包括zookeeper + kafka)：jps

#### 花生壳（phddns status）： phddns status

#### Cpolar：systemctl status cpolar **<http://gznfxy.cpolar.cn/>**

#### Docker：systemctl status docker

#### Docker内的容器（Nginx + MongoDB + RabbitMQ）:docker ps

### 生成镜像

参考：

<https://blog.csdn.net/qq_42777313/article/details/125471276>

<https://blog.csdn.net/qq_43412528/article/details/124980392>

注意：直接参考第二篇文章会出现制作镜像闪退问题，参考第一篇文章中的“遇到的问题及闭坑指南”的1，2点解决

1：vi /etc/selinux/config

SELINUX=disabled

**2：cd /etc/yum.repos.d/**

**wget ftp://ftp.mondorescue.org/rhel/7/x86\_64/mondorescue.repo**

**3：yum -y install mondo --nogpgcheck**

**4：vi /etc/mindi/mindi.conf**

**EXTRA\_SPACE = 801520**

**5：执行命令：mondoarchive**

**注意：生成镜像成功后，记得1、4步改回原值**

### Git安装：

参考：<https://blog.csdn.net/u012069313/article/details/123206297>

#### 下载git：<https://github.com/git/git/tags>

#### 上传git压缩包到/usr/local

#### 进入/usr/local目录，解压git压缩包：tar zxvf git.xxxx

#### 进入git解压目录：cd

#### 运行命令：make configure

如果有问题，通过yum provides autoconf查询到没有安装的autoconf，再yum -y install autoconf安装包及对应的依赖！

#### 执行命令：./configure --prefix=/usr/local/git

#### 执行命令：make profix=/usr/local/git

如果出现缺少openssl依赖库问题，运行以下命令：

yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel

#### 安装git：make install

#### 创建配置文件：touch /etc/profile.d/git.sh

#### 编辑vi /etc/profile.d/git.sh，加入export PATH=$PATH:/usr/local/git/bin

#### 设置读写权限：chmod +x /etc/profile.d/git.sh

#### 使其生效：source /etc/profile.d/git.sh

#### 查看git版本：git --version

### 一些问题：

#### 1：平面图加载中断

首先查看nginx错误日志文件，可能是磁盘空间用尽，可以清空nohup.out，清空各项目log日志，最后还是不行，可以服务器重启，服务器重启前先kill掉运行的应用：

// 第一种 ：/dev/nul把它丢弃一切写入其中的数据 意思就是把空的复制粘贴 给nohup.out

**cp /dev/null nohup.out**

// 第二种: 把无底洞定向输出到 nohup.out 起到清空作用

**cat /dev/null > nohup.out**

#### 2：docker start 容器运行不起来

重启服务器

#### 3：nginx、redis启动/停止

Nginx启动: 在 /sdin/下

Redis：到处可以redis-cli，因为安装好redis就加入环境变量了