**一、本地环境配置**

**1、Nginx环境**

]# **wget** [**http://nginx.org/download/nginx-1.15.7.tar.gz**](http://nginx.org/download/nginx-1.15.7.tar.gz)

]# **tar -xf nginx-1.15.7.tar.gz**

# **yum -y install gcc gcc-c++ make openssl-devel pcre-devel zlib-devel**

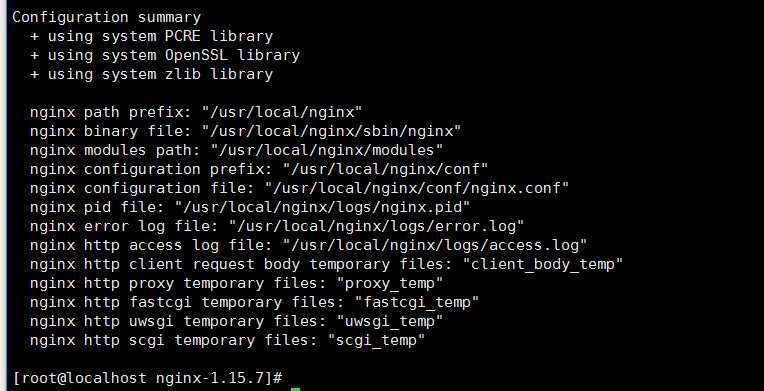
]# **cd nginx-1.15.7/**

]# **useradd -s /sbin/nologin nginx**

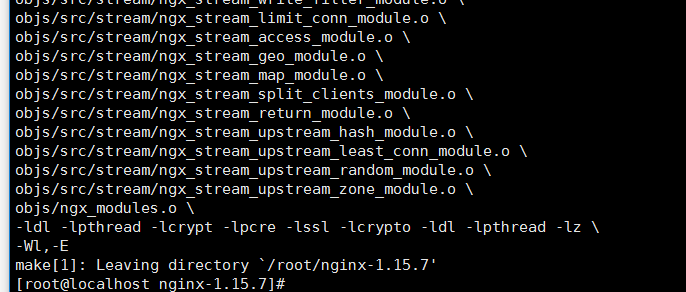
[root@localhost nginx-1.15.7]# **./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --with-stream --with-http\_ssl\_module --with-http\_stub\_status\_module**

**--with-http\_realip\_module**

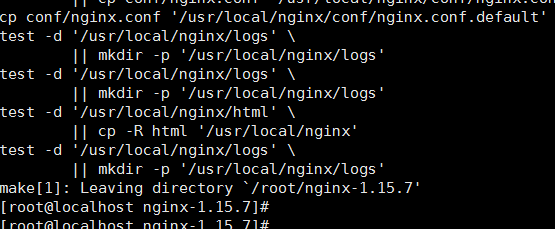
**--without-http\_proxy\_module（加了后proxy\_pass** [**http://127.0.0.1:8078**](http://127.0.0.1:8078)**报proxy\_pass错）**



[root@localhost nginx-1.15.7]# **make -j 4**



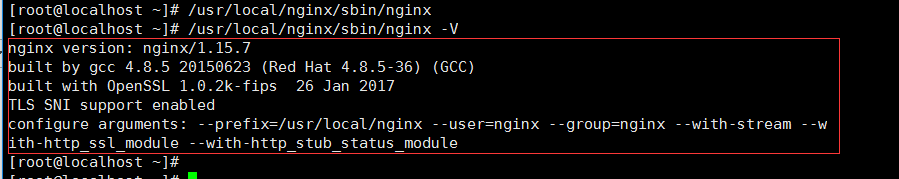
[root@localhost nginx-1.15.7]# **make install**



(上面的make install的提示，能用就行了)

[root@localhost ~]# **/usr/local/nginx/sbin/nginx**

[root@localhost ~]# **/usr/local/nginx/sbin/nginx -V**



创建软链接

[root@localhost nginx-1.15.7]# **ln -s /usr/local/nginx/sbin/nginx /usr/sbin/nginx**

+++++++++++

**2、php环境**

看个人博客，还有剩余的mysqli和mbstring，还有gd的包再补上

**yum -y install** **php72-php-mbstring php72-php-mysqlnd php72-php-gd**

**killall php-fpm** 后重启**systemctl restart php72-php-fpm**服务

(**/opt/remi/php72/root/sbin/php-fpm**绝对路径命令启动也行)

总的安装包为：

yum -y install php php72-php-opcache php72-php-ldap php72-php-odbc php72-php-pear php72-php-xml php72-php-xmlrpc php72-php-soap curl curl-devel php72-php-mbstring php72-php-mysqlnd php72-php-fpm php72-php-gd

+++++++

**3、Redis环境**

**wget http://download.redis.io/releases/redis-4.0.10.tar.gz**

[root@sg ~]# **tar -xf redis-4.0.10.tar.gz**

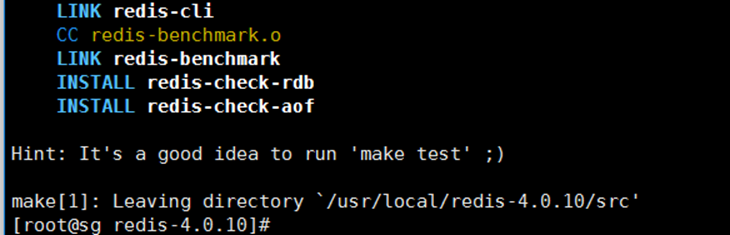
[root@sg ~]# mv redis-4.0.10 /usr/local/ （注意：先移动安装的目录再make，否则会不生效）

[root@sg ~]# pwd

/root

[root@sg ~]# cd /usr/local/redis-4.0.10/

[root@sg redis-4.0.10]# make #直接make安装



[root@localhost redis-4.0.10]# vi redis.conf （当前目录）

修改内容为

**bind 127.0.0.1** (设置只允许本地访问)

**daemonize yes** （no改为yes,目的是为了设置后台运行）

**requirepass** **kU08#y** （设置”kU08#y”的密码）

**protected-mode no** （这里讲原来的yes改为no,目的是为了解决安全模式引起的报错）

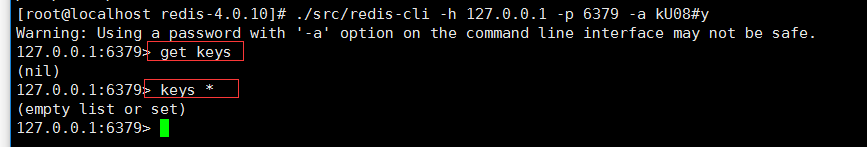
[root@localhost redis-4.0.10]# pwd

/usr/local/redis-4.0.10

[root@localhost redis-4.0.10]# ./src/redis-server redis.conf （注意当前目录的redis.conf）

验证命令,带ip，端口和密码验证

[root@localhost redis-4.0.10]# **./src/redis-cli -h 127.0.0.1 -p 6379 -a kU08#y**

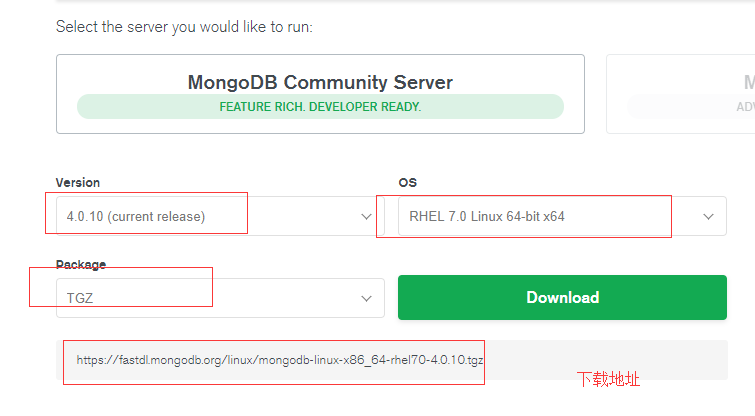


++++++++++++

**4、Mongodb环境**

下载路径百度或查看博客<https://blog.csdn.net/lituxiu/article/details/89416354>

下载地址https://www.mongodb.com/download-center/community



]# wget <https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-4.0.10.tgz> （注意：看有没有链接失效，下载不成功时）

[root@localhost ~]# tar -xf mongodb-linux-x86\_64-rhel70-4.0.10.tgz

[root@localhost ~]# mkdir /usr/local/mongodb4.0

[root@localhost ~]# cp -r mongodb-linux-x86\_64-rhel70-4.0.10/\* /usr/local/mongodb4.0/

**配置环境变量**

vi /etc/profile

export MONGODB\_HOME=/usr/local/mongodb4.0

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

]# source /etc/profile

**# 1.创建数据库文件存放路径**

cd /usr/local/mongodb4.0

mkdir -p /usr/local/mongodb4.0/data/db

chmod -R 777 /usr/local/mongodb4.0/data/db

**# 2.创建日志文件**

mkdir /usr/local/mongodb4.0/logs

touch /usr/local/mongodb4.0/logs/mongodb.log

**# 3.创建启动文件**

vi /usr/local/mongodb4.0/mongodb.conf

vi mongodb.conf (在文件中插入如下内容 )

**#bind\_ip\_all = true**

**#bind\_ip=127.0.0.1** #本机访问 这个0.0.0.0和127.0.0.1没有设置时，默认为127.0.0.1

**#auth=true**  #开启校验用户

**dbpath=/usr/local/mongodb4.0/data/db** #数据文件存放目录

**logpath=/usr/local/mongodb4.0/logs/mongodb.log** #日志存放目录

**port=27017**  #端口

**fork=true**  #以守护程序的方式启用，即在后台运行

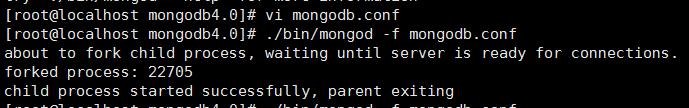
logappend=true  # 启动时 添加还是重写日志文件

**maxConns=5000**

**storageEngine = mmapv1**

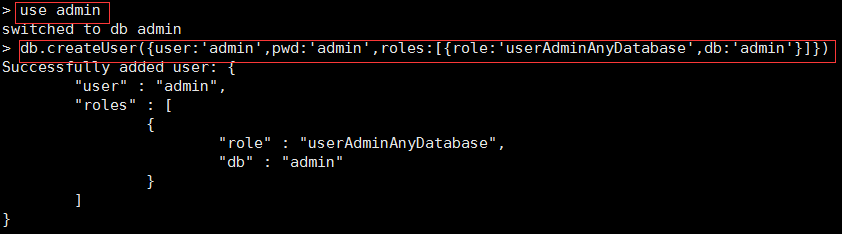
启动服务

[root@localhost mongodb4.0]# **./bin/mongod -f mongodb.conf**



[root@localhost mongodb4.0]# **./bin/mongo**

> use admin

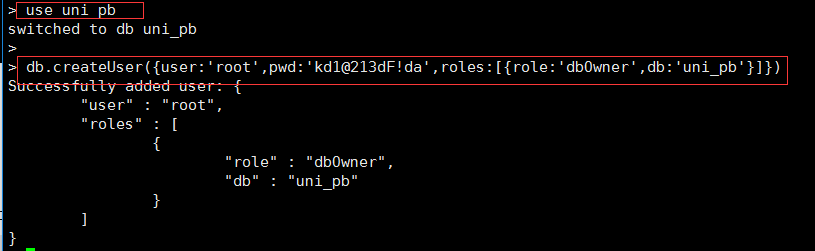
> db.createUser({user:'admin',pwd:'admin',roles:[{role:'userAdminAnyDatabase',db:'admin'}]})

添加相关库并设置了账户和密码

> use uni\_pb

switched to db uni\_pb

> db.createUser({user:'root',pwd:'kd1@213dF!da',roles:[{role:'dbOwner',db:'uni\_pb'}]})



退出再进入验证

> db.auth('root','kd1@213dF!da')

1

***当返回值1是为账户和密码都是正确的***

+++++++++++

**5、Jdk配置**

下载官网https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html （要注册个帐号）

打开下载的链接，用鼠标复制后，在xshell用wget下载

移动命令去改名

[root@localhost ~]# mv jdk-8u211-linux-x64.tar.gz\?AuthParam\=1560243782\_5f9e50f0e0b160d20dd838ed02defc7e jdk-8u211-linux-x64.tar.gz

[root@localhost ~]# tar -xf jdk-8u211-linux-x64.tar.gz

[root@localhost ~]# mv jdk1.8.0\_211/ /usr/local/

vi /etc/profile

#jdk

export JAVA\_HOME=/usr/local/jdk1.8.0\_211

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

source /etc/profile

最后tomcat的文件是在代码/www/java目录里启动的，大致配置如上。

Tomcat是直接拉包上放置，再打命令启动就行了

方法二：直接用yum安装**yum -y install java-1.8.0-openjdk**

+++++++++++++++

**6、Mysql环境**

安装方法博客地址<https://blog.csdn.net/lituxiu/article/details/90239798>

jdbc.user=root

jdbc.pass=uw7!23!Qw2

mysql> set password=password("uw7!23!Qw2");

创建库导入表

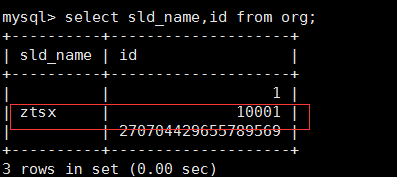
mysql>create database uni\_pb;

mysql>source 路径/表名.sql

在org表里修改域名

use uni\_pb;

select sld\_name,id from org;



以下作修改

mysql> update org set sld\_name = 'docker' where id = 10001;

+++++++

配置完成后，调试

测试tomcat的java接口是否正常

curl <http://127.0.0.1:8078/uni/AccountService/getUserOneById?id=1>



结果显示，中文字样没有乱码，接口信息返回就行。

+++++++++++++++++++++

**二、停止本地环境服务**

1、停止后台tomcat服务

[root@docker-test ~]# cd /www/java/

[root@docker-test java]# ./bin/shutdown.sh

2、停止上传tomcat服务

[root@docker-test fms0]# cd /www/fms/fms0

[root@docker-test fms0]# ./bin/shutdown.sh

3、停止php-fpm和nginx、mysql服务

[root@docker-test fms0]# killall php-fpm

[root@docker-test fms0]# nginx -s stop

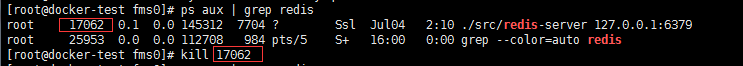
[root@docker-test fms0]# systemctl stop mysqld

[root@docker-test fms0]# systemctl disable mysqld

4、停止redis和mongodb服务

[root@docker-test fms0]# **ps aux | grep redis**

[root@docker-test fms0]# **kill 17062**



[root@docker-test fms0]# **ps aux | grep mongodb**

[root@docker-test fms0]# **kill 20846**

