## 更新apt-get工具

apt-get update

apt-get upgrade

## 升级vi编辑器

apt-get remove vim-common

apt-get install vim

## 安装SSH服务(用于远程连接)

apt-get install openssh-server

apt-get install ssh

启动ssh服务： /etc/init.d/ssh start

在客户端使用ssh工具连接时报错：algorithm negotiation failed

修改sshd的配置文件 /etc/ssh/sshd\_config  
  
在配置文件中添加：  
  
Ciphers aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-ctr,3des-cbc,arcfour128,arcfour256,arcfour,blowfish-cbc,cast128-cbc  
  
MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160,hmac-sha1-96,hmac-md5-96  
  
KexAlgorithms diffie-hellman-group1-sha1,diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha1,diffie-hellman-group-exchange-sha256,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group1-sha1,curve25519-sha256@libssh.org  
  
重启sshd服务后，即可正常连接。  
  
导致此问题的原因是ssh升级后，为了安全，默认不再采用原来一些加密算法，我们手工添加进去即可。

## 安装jdk

从官网下载对应版本的jdk安装包（目前使用的版本是8u121）

解压 tar –zxvf jdk的tar文件

设置环境变量

vim /etc/profile

JAVA\_HOME=

PATH=$PATH:$JAVA\_HOME/bin

export PATH JAVA\_HOME

保存退出

source /etc/profile

## 安装tomcat

从官网下载对应版本的tomcat（目前使用的版本是9.0.0.M17）

解压 tar –zxvf tomcat文件

bin/start.sh启动tomcat服务

## 安装mysql

apt-get install mysql-server

apt-get install mysql-client

apt-get install libmysqlclient-dev

1. 配置mysql可以远程访问

登录mysql，输入

grant all privileges on \*.\* to ‘root’ identified by ‘ut123456’ with grant option;

flush privileges;

修改/etc/mysql/mysql.conf.d/mysqld.cnf

去掉 bind 127.0.0.1

1. 中文乱码问题

修改/etc/mysql/config.d/mysql.cnf (/etc/mysql/conf.d/mysql.cnf)

在mysql下面增加

default-character-set=utf8mb4

修改/etc/mysql/mysql.conf.d/mysqld.cnf

在mysqld下面增加

character-set-server=utf8mb4

修改配置文件之后要重启mysql生效 service mysql restart

## 配置mysql主从服务器

vim /etc/mysql/mysql.conf.d/mysqld.cnf

log-bin=mysql-bin #开启二进制日志

server-id=1 #每台服务器都不同，必须唯一

relay-log=mysql-relay-bin #开启中继日志

log\_slave\_updates = on #从服务器将时间记录到二进制日志中

auto\_increment\_increment=1 #自动增长的步长

auto\_increment\_offset=1 #自动增长的起始数值

主从服务器的mysql配置文件都要改，但是serve-id必须不同

配置完成之后重启mysql服务 service mysql restart

在主服务器的mysql中执行

show master status;

获取到MASTER\_LOG\_FILE 和 MASTER\_LOG\_POS

在从服务器的mysql中执行

change master to master\_host='172.16.63.129',master\_user='slave',master\_password='password',master\_port=3306,MASTER\_LOG\_FILE='mysql-bin.000007', MASTER\_LOG\_POS=1068;

参数修改成自己对应的

在从服务器的mysql中执行

start slave;

启动从服务器的mysql的slave服务同步主从数据库数据

## 安装redis(缓存)

从redis官网下载对应版本的安装包(目前使用的版本是3.2.7)

解压 tar –zxvf redis文件

cd 解压后的目录

make

make install

按照需要修改redis.conf文件(一般也可以不改)

如果要远程可以访问，将修改bind 127.0.0.1 为bind 0.0.0.0

如果要配置登录密码，打开配置 requirepass password

如果要后台启动服务，打开配置 daemonize yes

redis-server 配置文件

## 安装redis集群

安装集群，至少需要6个redis服务

apt-get install ruby

apt-get install rubygems

gem install redis

修改redis.conf文件

##修改配置文件中的下面选项

port 7000

daemonize yes

cluster-enabled yes

cluster-config-file nodes.conf

cluster-node-timeout 5000

appendonly yes

在一台服务器，可以通过不同的配置文件启动不同的端口，启动多个redis server进程

执行redis的创建集群命令创建集群

redis-trib.rb  create --replicas 1 127.0.0.1:7000 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003 127.0.0.1:7004 127.0.0.1:7005

redis-trib.rb 还有很多参数控制集群配置，具体可以网上查询

## 安装Nginx

apt-get update

apt-get install build-essential

apt-get install libtool

apt-get install libpcre3

apt-get install openssl

[apt-get](https://www.baidu.com/s?wd=apt-get&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao) install libssl-dev

apt-get install libpcre3 libpcre3-dev

apt-get install zlib1g-dev

去网站下载pcre库和zlib库

tar解压文件

cd 解压目录

./configure

make

make install

从nginx官网下载对应版本（目前使用的版本是1.11.9）

tar –zxvf nginx文件

cd

./configure --prefix=/usr/local/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module --with-file-aio --with-http\_realip\_module --with-stream

make

make install

根据系统需求配置nginx的配置文件/usr/local/nginx/conf/nginx.conf

启动服务

/usr/local/nginx/sbin/nginx –c /usr/local/nginx/conf/nginx.conf

## 安装fastdfs(分布式文件系统)

apt-get install libevent

cd /usr/local/src/

git clone https://github.com/happyfish100/libfastcommon.git

cd libfastcommon/

./make.sh

./make.sh install

cd /usr/local/src

wget http://nchc.dl.sourceforge.net/project/fastdfs/FastDFS%20Server%20Source%20Code/FastDFS%20Server%20with%20PHP%20Extension%20Source%20Code%20V5.08/FastDFS\_v5.08.tar.gz

tar -zxvf FastDFS\_v5.08.tar.gz

cd FastDFS

./make.sh

./make.sh install

sed -i 's#/usr/local/bin/#/usr/bin/#g' /etc/init.d/fdfs\_trackerd

mkdir -p /data/fastdfs/tracker

**Tracker配置文件修改**

cd /etc/fdfs/

cp tracker.conf.sample tracker.conf

vim tracker.conf

disabled=false                  #启用配置文件

port=22122                      #设置 tracker 的端口号

base\_path=/data/fastdfs/tracker #设置 tracker 的数据文件和日志目录（需预先创建）

http.server\_port=8888           #指的是在tracker服务器上启动http服务进程，如:apache或者nginx 启动时所监听的端口

启动服务：

/etc/init.d/fdfs\_trackerd start

**storage配置文件修改**

sed -i 's#/usr/local/bin/#/usr/bin/#g' /etc/init.d/fdfs\_storaged

mkdir -p /data/fastdfs/base

mkdir -p /data/fastdfs/data

cd /etc/fdfs/

cp storage.conf.sample storage.conf

vim storage.conf

disabled=false                        #启用配置文件

group\_name=group1                     #组名，根据实际情况修改

port=23000                            #设置 storage 的端口号

base\_path=/data/fastdfs/base          #设置 storage 的日志目录（需预先创建）

store\_path\_count=1                    #存储路径个数，需要和 store\_path 个数匹配

store\_path0=/data/fastdfs/date        #存储路径

tracker\_server=10.10.10.81:22122      #tracker 服务器的 IP 地址和端口号

http.server\_port=8888                 #设置storage上启动的http服务的端口号，如安装的nginx的端口号

启动服务：

/etc/init.d/fdfs\_storage start

## 设置服务开机启动

vim /etc/rc.local

将要开机启动的脚本写到文件中

启动redis

redis-server redis的配置文件地址

启动fastdfs

/etc/init.d/fdfs\_trackerd start

/etc/init.d/fdfs\_storaged start

启动tomcat

TomcatHome/bin/startup.sh

启动nginx

/usr/local/nginx/sbin/nginx –c /usr/local/nginx/conf/nginx.conf

如果程序无法启动，请查看文件的权限，用户是否有权限操作