

typora-copy-images-to: 123123

如何配置服务器

一. 准备

1.1 前提工作

- 购买服务器
- 购买云盘
- 获取服务器账号密码
- 远程工具：xshell

1.2 端口配置

1. 实例 -> 选择此实例右侧 更多 -> 网络和安全组 -> 安全组配置 进入安全组 点击 配置规则
2. 点击 快速创建规则 ->
 - 规则方向:入方向
 - 规则策略:允许
 - 常用端口:3306
 - 授权对象:0.0.0.0/0
 - 优先级:1

二. 挂载磁盘

简述:将数据盘挂载到服务器上启动

2.1 查看数据盘

```
fdisk -l
```

1545117723902

查看是否被挂载

```
df -h
```

1545117757546

2.2 分区

对vdb进行分区操作

```
fdisk /dev/vdb
```

1545117893928

- n创建一个新的分区
- p查看当前分区
- w保存修改并退出

1. 依次输入 n p 1 之后 First sector (2048-1048575999, default 2048) 字样 回车 两次然后输出 wq 退出

1545118174243

2. 运行 `fdisk -l` 命令,查看新的分区,新分区 vdb1 已经创建好;

3. 运行 `mkfs.ext3 /dev/vdb1` 对新分区进行格式化。格式化所需时间取决于数据盘大小。您也可自主决定磁盘格式,入ext3等

1545118729014

4. 运行 `mount /dev/vdb1 /home` 挂载新分区

5. 运行 `echo /dev/vdb1 /home ext4 defaults 0 0 >> /etc/fstab` 完成后可以使用 `cat /etc/fstab` 命令查看;其中vdb1和 /home 为 磁盘名称 和 挂载的文件夹名称

6. 执行 `df -h` 查看是否挂载成功

三. 创建文件夹

在home文件中创建下列文件夹

- conf
- data
- install
- lost+found
- script
- server
- software
- update
- webapps

四. 下载安装包

4.1 安装wget工具,然后用wget下载安装包

```
yum install wget
```

下载安装包,install文件夹下下载需要的安装包

wget -P /home/install http://dl.**.com/software/mysql-5.7.20-linux-glibc2.12-x86_64.tar.gz

- nginx-1.12.2
- tomcat
- jdk
- mysql

4.2 解压安装包

cd 进入压缩包的文件夹

- tar -zxvf 解压缩.tar.gz文件,例如:tar -zxvf mysql-5.7.20-linux-glibc2.12-x86_64.tar.gz
- 移动到software文件夹中 (综合命令 tar zxvf nginx-1.12.2.tar.gz -C /home/software)

所有的解压到文件夹 提示:需要一条一条的执行

```
tar zxvf apache-tomcat-8.0.47.tar.gz -C /home/software
tar zxvf jdk-8u152-linux-x64.tar.gz -C /home/software
tar zxvf mysql-5.7.20-linux-glibc2.12-x86_64.tar.gz -C /home/software
tar zxvf nginx-1.12.2.tar.gz -C /home/software
```

五. 安装数据库

5.1 授权

cd 进入压缩包的文件夹

- 添加组,用户等等

```
groupadd mysql
```

```
useradd -g mysql mysql
```

- -r:建立系统账号 -g:指定用户所属的起始群组 第一个 mysql:用户组 第二个mysql:用户名
或 useradd mysql[用户名] -g mysql[用户组]

5.2 授权用户

1. 修改解压后的mysql-5.7.20-linux-glibc2.12-x86_64文件夹改名为mysql

将mysql目录包含所有的子目录和文件,所有者改变为root,所属组改为mysql

进入mysql的父级目录 chown -R root:mysql mysql 或者直接执行命令 chown -R root:mysql /home/software/mysql

2. 创建mysql数据路径 /home/data/mysql

```
chown -R mysql:mysql data 或 chown -R mysql:mysql /home/data
```

5.3 数据库初始化

1. 检查是否有libaio链接库文件 执行

```
rpm -qa|grep libaio
```

如果有会显示该链接库的版本号,如果没有没有则执行,安装链接库

```
yum install libaio
```

2. 进入解压的mysql目录,执行以下语句

```
bin/mysqld --user=mysql --datadir=/home/data/mysql --basedir=/home/software/mysql -  
-initialize
```

执行完成后将密码保存下来

5.3.1 配置文件

建立软连接: `ln -s /etc/my.cnf /home/software/mysql/my.cnf`

修改配置文件

```
[client]  
default-character-set = utf8mb4  
[mysql]  
default-character-set = utf8mb4  
[mysqld]  
skip-name-resolve  
basedir = /home/software/mysql  
datadir = /home/data/mysql  
port = 3306  
server_id = 1  
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES  
character_set_server = utf8mb4  
collation-server = utf8mb4_unicode_ci  
init_connect = 'SET NAMES utf8mb4'  
skip-character-set-client-handshake = true  
max_connections = 1000
```

5.3.2 设置开机自启动

1. 将服务文件拷贝到init.d下,并重命名为mysql

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

2. 赋予可执行权限

```
chmod +x /etc/init.d/mysqld
```

3. 添加服务

```
chkconfig --add mysqld
```

4. 显示服务列表

如果看到mysql的服务 3,4,5都是为on的话则成功,如果为off,则键入

```
chkconfig --level 345 mysqld on
```

5.3.3 启动数据库

```
service mysql start(可能需要重启)
```

service mysqld status查看mysql的状态

5.3.4 启动数据库

首先把mysql客户端放到默认路径,这样mysql就可以通过mysql -u root -p启动

```
ln -s /home/software/mysql/bin/mysql /usr/local/bin/mysql
```

其中 /home/software/mysql/bin/mysql 为mysql路径地址, 之后键入

```
mysql -u root -p
```

根据提示输入密码进入数据库,进行修改密码,之后执行

```
set password=password('111111');
use mysql;
update user set host='%' where user='root';
flush privileges;
```

如果密码忘记,请执行一下步骤 修改my.cnf, 在 [mysqld] 下方加入 `skip-grant-tables` 接下来输入mysql进入数据库 之后进行上方的修改密码操作 删除skip-grant-tables 重启mysql完成 service mysqld restart 使用Navicat测试链接

六. 安装jdk

6.1 设置环境变量

方式一:

在/home/script建立脚本文件env.sh,在profile最后面添加source /home/script/env.sh

注意:env.sh如果在notepad++中打开右下角为应为linux格式的

```
JAVA_HOME=/home/software/jdk1.8.0_152
JRE_HOME=$JAVA_HOME/jre
PATH=$PATH:$JAVA_HOME/bin:/home/software/mysql/bin:/home/software/nginx/sbin
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export JAVA_HOME
export JRE_HOME
export PATH
export CLASSPATH
export XUENW_CFG=file:///home/conf
```

方式二:

编辑profile文件 `vim /etc/profile` 在profile最后面添加

```
export JAVA_HOME=/home/software/jdk1.8.0_152
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

执行profile或者env.sh文件,这样可以使配置不用重启即可立即生效。

```
source /etc/profile
```

或

```
source /home/script/env.sh
```

如果提示 `-bash: /home/script/env.sh: Permission denied` 则执行 `chmod u+x /home/script/*.sh`

检查新安装的jdk

```
java -version
```

```
java version "1.8.0_152"
Java(TM) SE Runtime Environment (build 1.8.0_152-b16)
Java HotSpot(TM) 64-Bit Server VM (build 25.152-b16, mixed mode)
```

或者执行 `echo $JAVA_HOME` 查看环境变量

到此为止,整个安装过程结束

注意 如果方式一需要开机自动配置环境需要在/etc/profile最后面加上 `source /home/script/env.sh`

七. 安装nginx

7.1 安装必要的环境

- gcc安装

安装 nginx 需要先将官网下载的源码进行编译,编译依赖 gcc 环境,如果没有 gcc 环境,则需要安装

```
yum -y install gcc gcc-c++ autoconf automake
```

- zlib安装 PCRE pcre-devel 安装 OpenSSL 安装

zlib 库提供了很多种压缩和解压缩的方式,nginx 使用 zlib 对 http 包的内容进行 gzip,所以需要在 Centos 上安装 zlib 库。

```
yum -y install zlib zlib-devel openssl openssl-devel pcre-devel
```

7.2 配置安装

进入nginx目录,执行以下命令, 注意--prefix等号后面的一串为 /home/software/nginx-1.12.2 nginx目录

```
./configure --prefix=/home/software/nginx-1.12.2 --with-http_stub_status_module --with-http_ssl_module
```

```
[root@iz2zedkdcntqh9bniwae18z nginx-1.12.2]# make && make install
```

查看nginx版本

```
[root@iz2zedkdcntqh9bniwae18z nginx-1.12.2]# /home/software/nginx-1.12.2/sbin/nginx -v
```

到此, nginx安装完成 **启动nginx**

进入nginx目录中的sbin文件夹中执行

```
[root@iz2zedkdcntqh9bniwae18z sbin]# ./nginx
```

查看nginx进程,可以根据IP访问进行测试

```
ps -ef | grep nginx
```

7.3 错误及解决方法

```
nginx: [alert] could not open error log file: open() "/home/software/nginx-1.12.2/logs/error.log" failed (2: No such file or directory)
2018/09/18 18:16:07 [emerg] 7112#0: open() "/home/software/nginx-1.12.2/logs/access.log" failed (2: No such file or directory)
```

解决方法: 在nginx目录中创建logs文件夹

7.4 常用语句

```
//启动nginx
[root@iz2zedkdcntqh9bniwae18z sbin]# ./nginx
//停止
[root@iz2zedkdcntqh9bniwae18z sbin]# ./nginx -s stop
//详情
[root@iz2zedkdcntqh9bniwae18z sbin]# ps -ef | grep nginx
```

7.5 配置文件修改

```
worker_processes 1;

events {
```

```

worker_connections 1024;
}

http {
    include      mime.types;
    default_type application/octet-stream;
    sendfile     on;
    keepalive_timeout 65;
    gzip on;

    server {
        listen      80;
        server_name www.xxx.com;
        #charset koi8-r;

        #access_log logs/host.access.log main;

        location / {
            proxy_pass http://127.0.0.1:8080/xxx-portal/;
            client_max_body_size 1000m;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}

server{
    listen 80;
    server_name service.xxx.com;
    location /xxx-service/ {
        proxy_pass http://127.0.0.1:8080/xxx-service/;
        proxy_set_header request_uri $request_uri;
        proxy_set_header Host $host;
        proxy_set_header Remote_Addr $remote_addr;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        client_max_body_size 1000m;
    }
}

server{
    listen 80;
    server_name manage.xxx.com;
    location / {
        proxy_pass http://127.0.0.1:8080/xxx-manage/;
        proxy_cookie_path /xxx-manage /;
        proxy_set_header request_uri $request_uri;
        proxy_set_header Host $host;
        proxy_set_header Remote_Addr $remote_addr;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        client_max_body_size 1000m;
    }
    location /xxx-resource/ {

```

```

        proxy_pass http://127.0.0.1:8080/xxx-resource/;
        proxy_set_header request_uri $request_uri;
    proxy_set_header    Host      $host;
    proxy_set_header    Remote_Addr    $remote_addr;
    proxy_set_header    X-Real-IP      $remote_addr;
    proxy_set_header    X-Forwarded-For    $proxy_add_x_forwarded_for;
        client_max_body_size 1000m;
    }

}

server{
    listen 80;
    server_name resource.xxx.com;
    location /xxx-resource/ {
        proxy_pass http://127.0.0.1:8080/xxx-resource/;
        proxy_set_header request_uri $request_uri;
    proxy_set_header    Host      $host;
    proxy_set_header    Remote_Addr    $remote_addr;
    proxy_set_header    X-Real-IP      $remote_addr;
    proxy_set_header    X-Forwarded-For    $proxy_add_x_forwarded_for;
        client_max_body_size 1000m;
    }
}

server{
    listen 80;
    server_name r.xxx.com;
    location /{
        root /home/data/file;
        expires 3d;
    }
}
}

```

如果开机自动启动可以放到创建一个脚本start.sh放到/home/script中,注意.sh的文件格式是Linux的,文件内容为

```
/home/software/nginx-1.12.2/sbin/nginx;
```

之后打开 `/etc/rc.d/rc.local` 文件,添加

```
/home/script/start.sh
```

八. 安装tomcat

8.1 配置文件

server.xml

```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
               maxThreads="400"
               acceptCount="500"
               URIEncoding="UTF-8"
           />
```

```
<Host name="localhost"  appBase="/home/webapps"
      unpackWARs="true" autoDeploy="true">
```

8.2 解决Tomcat启动项目慢

yum -y install rng-tools

systemctl start rngd