

开创啦系统安装配置手册

文档修订

版本号	修订内容	修订日期	修订人
1.0	创建文档	2017/04/19	章传胜
2.0	加入 nginx,redis	2017/05/10	章传胜

第 1 章 文档目录

第 2 章 选择安装介质

第 3 章 概要说明

第 4 章 安装准备

在安装部署前，需要为开创啦服务创建执行用户。

以 root 用户登录，执行以下命令创建 oseasy 用户，通过 passwd 命令指定用户登录密码。



```
> adduser oseasy  
> passwd oseasy
```

最好设置一下系统的 ulimit

以 root 用户登录，使用文本编辑器打开 /etc/security/limits.conf

```
> vi /etc/security/limits.conf
```

加入 * - nofile 30000

修改后重启（重启命令:reboot）

```
> ulimit -n
```

查看是否修改成功。（如果是 30000，则修改成功）

查看 linux 版本

```
#cat /etc/redhat-release
```

CentOS Linux release 7.2.1511

查看内核/操作系统/CPU 信息

```
uname -a
```

```
#ifconfig
```

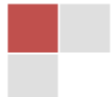
如果不是命令

请安装 ifconfig 命令行

```
# yum install net-tools
```

完成后 #ifconfig

查看网卡是否是 eth0 如果不是，本次版本需要修改 网卡为 eth0



三、修改网卡名称样式为ethx

如果不习惯使用新的命名规则，可以恢复使用传统的方式命名，编辑 grub 文件，增加两个变量，再使用 grub2-mkconfig 重新生成 grub 配置文件即可。

1. 编辑 grub 配置文件

```
vim /etc/sysconfig/grub # 其实是/etc/default/grub的软连接
# 为GRUB_CMDLINE_LINUX变量增加2个参数，具体内容如下(加粗)：
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap net.ifnames=0 biosdevname=0 rhgb quiet"
```

2. 重新生成 grub 配置文件

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

然后重新启动 Linux 操作系统，通过 ip addr 可以看到网卡名称已经变为 eth0。

3. 修改网卡配置文件

原来网卡配置文件名称为 ifcfg-ens33，这里需要修改为 ethx 的格式，并适当调整网卡配置文件。

```
mv /etc/sysconfig/network-scripts/ifcfg-ens33 /etc/sysconfig/network-scripts/ifcfg-eth0
# 修改ifcfg-eth0文件如下内容(其它内容不变)
NAME=eth0
DEVICE=eth0
[root@localhost ~]# systemctl restart network.service # 重启网络服务
```

注意：ifcfg-ens33 文件最好删除掉，否则重启 network 服务时候会报错。

第 5 章 Jdk 的安装

以 root 用户登录. 移除旧版 jdk

```
find / -name javac -print & 查看是否装有 jdk
rpm -qa | grep jdk 查看
rpm -qa | grep gcj 查看
yum -y remove java-1.8.0-* 移除旧版 jdk
```

安装 jdk

```
tar xvf jdk-8u121-linux-x64.tar.gz #解压
```



```
mv jdk1.8.0_121 /usr/local  
cd /usr/local/  
vi + /etc/profile
```

在文件最后插入

```
export JAVA_HOME=/usr/local/jdk1.8.0_121  
export  
CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar  
export PATH=$PATH:$JAVA_HOME/bin
```

保存后

```
source /etc/profile
```

查看 jdk 是否安装成功

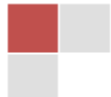
```
java -version
```

```
java version "1.8.0_121"  
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
```

安装成功

第 6 章 ftp 安装(单机)

在线安装



1. 卸载之前安装过的 vsftd

```
yum remove vsftpd
```

2. 关掉 iptables 和 selinux

```
vi /etc/selinux/config  
SELINUX=enforcing 改为 SELINUX=disabled
```

重启机器查看 selinux 状态

```
/usr/sbin/sestatus -v
```

3. 安装 vsftpd

安装 vsftpd 服务器: rpm -ivh vsftpd-*.rpm 或从互联网寻找对应资源直接安装 vsftpd:

```
yum -y install vsftpd
```

```
rpm -ivh vsftpd-3.0.2-10.el7.x86_64.rpm 或  
yum -y install vsftpd
```

装完后，配置文件/etc/vsftpd/vsftpd.conf，去掉注释 chroot_local_user=YES，让这条配置生效，固定用户在主目录下参考提供的 vsftpd.conf 的配置

```
useradd -d /userRemoteFilePath ftponly //创建用户 ftponly 并指定用户主目录为  
/userRemoteFilePath  
usermod -s /sbin/nologin ftponly //设置 ftponly 用户只能 ftp，不能 telnet  
chown -R ftponly.ftponly /userRemoteFilePath //给目录及子目录授予用户 ftponly  
passwd ftponly //设置用户 ftponly 的密码  
chmod a-w /userRemoteFilePath/tool/
```



以上完成，即可用 ftp 客户端测试

service vsftpd start 测试是否安装成功 (或systemctl start vsftpd)

设置开机启动

chkconfig vsftpd on

查看状态

chkconfig --list|grep vsftpd

httpd 的安装

```
yum install httpd -y #安装
chkconfig httpd on #开机启动
vi /etc/httpd/conf/httpd.conf
httpd.conf 配置参考提供的配置
#设置 Apache 的默认文档路径的位置是在/var/www/html
DocumentRoot "/userRemoteFilePath/tool"
service httpd start #开启 apache
```

访问下 <http://192.168.0.105>看能否访问

验证是否可以访问

echo 'test' > /userRemoteFilePath/tool/1.log

访问<http://192.168.0.105/1.log>

如果 You don't have permission to access / on this server. 说明

httpd.conf 没设置正确或者 selinux 没有关闭

httpd.conf 的设置请参考给出的例子

上传 ftp 图片.zip 至/ userRemoteFilePath/tool/oseasy 这个目录。解压

```
unzip -o ftp 图片.zip -d /userRemoteFilePath/tool/oseasy
chown -R ftponly.ftponly /userRemoteFilePath/tool/oseasy
```



如果连接包验证失败：

在配置文件中加入：

allow_writeable_chroot=YES

第 7 章 Mysql 安装(单机)

1. 删除默认 MySQL

```
rpm -qa|grep mysql    //查询 MySQL 是否安装
rpm -e mysql          //普通删除模式
rpm -e --nodeps mysql  //强制删除模式，不删除依赖包
```

2. 安装 mysql

```
tar xvf mysql-5.7.17-linux-glibc2.5-x86_64.tar.gz
mv mysql-5.7.17-linux-glibc2.5-x86_64 /usr/local/
cd /usr/local/
ln -s mysql-5.7.17-linux-glibc2.5-x86_64 mysql
cp my.cnf /etc/ #my.cnf 用已经编辑好的
mkdir -p /usr/local/mysql/data
mkdir -p /usr/local/mysql/log/
echo "> /usr/local/mysql/log/error.log" > /usr/local/mysql/log/error.log
echo "> /usr/local/mysql/log/slow-query.log" > /usr/local/mysql/log/slow-query.log
groupadd oseasy
#####useradd -g oseasy oseasy
```



```
chown -R oseasy:oseasy /usr/local/mysql/
```

初始化数据库 这一步会产生一个随机密码

```
cd /usr/local/mysql
```

```
./bin/mysqld --user=oseasy --basedir=/usr/local/mysql/ --datadir=/usr/local/mysql/data --initialize
```

如果报

error while loading shared libraries: libaio.so

缺少安装包 libaio 和 libaio-devel.

yum install libaio*

成功后查看

```
tail -200f /usr/local/mysql/log/error.log
```

```
:46:49.988555Z 0 [Warning] 'NO_ZERO_DATE', 'NO_ZERO_IN_DATE' and 'ERROR_FOR_DIVISION_BY_ZERO' sql modes should be used with strict mode. They
re release.
:46:49.988563Z 0 [Warning] 'NO_AUTO_CREATE_USER' sql mode was not set.
:46:50.545321Z 0 [Warning] InnoDB: New log files created, LSN=45790
:46:50.706825Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
:46:50.897686Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Gener
fa163e7e06a4.
:46:50.899680Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
:46:50.900492Z 1 [Note] A temporary password is generated for root@localhost: 1R;XLSRYN3ZC
```

会看到随机密码，修改密码时需要用到

3. 启动 mysql 切换到 mysql 用户

```
#su oseasy
```

```
./mysqld_safe --user=oseasy &
```

```
ps -ef|grep mysql 检查是否启动成功
```


连接客户端

```
./mysql -uroot -p  
输入刚才保存的临时密码  
>set password=password('新密码');
```

4. 设置远程访问

```
cd /usr/local/mysql/bin  
mysql -uroot -p
```

如果报错：

ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/lib/mysql/mysql.sock'

权限问题：

```
chown -R oseasy:oseasy /var/lib/mysql/  
chmod 777 -R /var/lib/mysql/
```

设置用户 root 可以在任意 IP 下被访问：

```
grant all privileges on *.* to root@"%" identified by "新密码";
```

刷新权限使之生效

```
flush privileges;
```

```
exit;
```

此时使用远程机器进行访问

5. 添加服务系统服务

```
cp -a ./support-files/mysql.server /etc/init.d/mysqld  
su oseasy
```



```
service mysqld start  
ps -ef|grep mysql
```

如果出现 Starting MySQL...The server quit without updating PID file 这样的问题，
用 `chown -R oseasy:oseasy /usr/local/mysql/`
请切换 oseasy 用户，权限问题引起的

6. 数据的导入导出

方法一

```
cd /usr/local/mysql/bin
```

```
./mysql -uroot -p123456 #连接 mysql
```

```
mysql>create database pro_creativecloud;
```

```
mysql>use pro_creativecloud;
```

```
mysql>source /home/abc/abc.sql;
```

方法二

```
mysql -u 用户名 -p 密码 数据库名 < 数据库名.sql
```

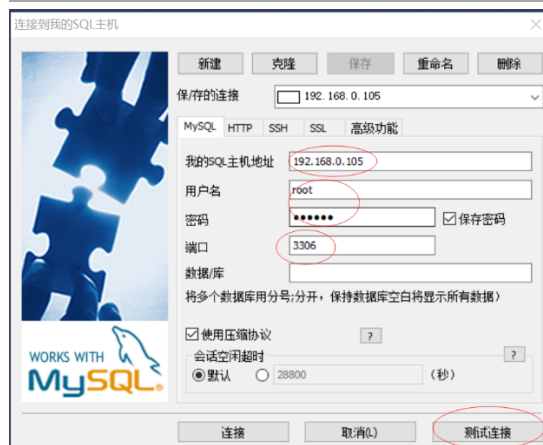
```
# mysqlimport -u root -p123456 < 20170423_3.sql
```

方法三

用可视化工具 SQLyog 导入

安装 sqlyong.连接上 mysql





指定给出的 sql 文件导入。



第 8 章 redis 的安装（单机）

1. 安装 redis

上传redis-3.0.7.tar.gz至服务器,解压至/usr/local/redis-3.0.7

```
tar zxvf redis-3.0.7.tar.gz /usr/local  
ln -s redis-3.0.7.tar.gz redis  
cd /usr/local/redis  
make
```

二进制文件是编译完成后在 src 目录下，通过下面的命令启动 Redis 服务

```
# nohup ./redis-server &    ###启动 redis
```

你可以使用内置的客户端命令 redis-cli 进行使用：

```
# src/redis-cli
```

```
redis> set foo bar
```

```
OK
```

```
redis> get foo
```

```
"bar"
```

如果 make 时出现 zmalloc.h:50:31: 错误: jemalloc/jemalloc.h: 没有那个文件或目录
请

```
make MALLOC=libc
```



第 9 章 tomcat 服务器安装（单机）

1. 安装 tomcat

上传 apache-tomcat-8.5.11.tar.gz 至服务器。解压至/usr/local/ apache-tomcat-8.5.11

```
tar zxvf apache-tomcat-8.5.11.tar.gz /usr/local  
ln -s apache-tomcat-8.5.11 tomcat
```

2. Tomcat 配置

修改/usr/local/tomcat/conf/server.xml



```
-->
<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
maxThreads="50000"
minSpareThreads="1000"
maxSpareThreads="5000"
acceptCount="50000"
enableLookups="false"
socket.appReadBufSize="2048"
socket.appWriteBufSize="2048"
bufferSize="2048"
redirectPort="8443"
URIEncoding="UTF-8" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
-->
```

```
resourceName="userDatabase" />
</Realm>

<Host name="localhost" appBase="/home/www/admin" unpackWARs="true" autoDeploy="true">
  <Context path="/" reloadable="true" docBase="/home/www/admin/ROOT" sessionCookiePath="/" /></Context>
</Host>
</Engine>
</Service>
</Server>
```

详细参考：server.xml 文件，已在目录包中提供



以上为当前配置值。

Jvm 配置: /usr/local/tomcat/bin/catalina.sh

```
# LOGGING_MANAGER=org.apache.juli.logging.manager=org.apache.juli.ClassLoader
# -----
# OS specific support. $var _must_ be set to either true or false.
cygwin=false
os400=false
darwin=false
case "`uname`" in
  CYGWIN*) cygwin=true;;
  OS400*) os400=true;;
  Darwin*) darwin=true;;
  esac

export JAVA_OPTS="-Xms1024m -Xmx8192m -XX:PermSize=512M -XX:MaxPermSize=1024m"

# resolve links - $0 may be a softlink
PRG="$0"

while [ -h "$PRG" ]; do
  ls=`ls -ld "$PRG"`
  link=`expr "$ls" : '.*-> \(.*\) '$`
  if expr "$link" : '/.*' > /dev/null; then
    PRG="$link"
  else
    PRG=`dirname "$PRG"`/"$link"
  fi
done
```

3. 启动和关闭

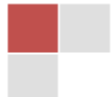
```
cd /usr/local/tomcat/bin
sh startup.sh start; #启动
ps -ef |grep tomcat

sh shutdown.sh ; #关闭
ps -ef |grep tomcat
```

将程序的安装包ROOT.zip 至/home/www/admin/ 解压、最后的目录应该 /home/www/admin/ROOT

解压 unzip -o ROOT.zip -d /home/www/admin/ROOT

cd /home/www/admin/ROOT/WEB-INF/classes #修改数据库 ftp 连接地址



vi initiate.properties

```
jdbc.type=mysql  
jdbc.driver=com.mysql.jdbc.Driver  
jdbc.url=jdbc:mysql://192.168.0.105:3306/pro1_creativecloud?useUnicode=true&characterEncoding=utf-8  
jdbc.username=root  
jdbc.password=123456
```

修改为对应的 ip 和 pro_creativecloud

```
##  
ftp.url=192.168.0.105  
ftp.port=2121  
ftp.username=ftponly  
ftp.passwd=os-easy  
ftp.httpurl=http://192.168.0.105
```

修改 ftp 用户名密码和 ip 地址

修改 redis 的地址

```
#redis settings  
redis.keyPrefix=oseasy  
redis.host=192.168.0.107  
redis.port=6379  
redis.expire=1800  
#####  
##### System settings #####  
#####
```

启动 tomcat, 查看日志

tail -200f logs/catalina.out

看到如下图的内容说明 tomcat 启动成功。

```
2013-1-6 21:31:49 org.apache.catalina.startup.Catalina  
信息: Server startup in 612 ms
```


访问

<http://ip:port/a> 后台

<http://ip:port/f> 前台

第 10 章 导入许可

启动后访问后台<http://ip:port/a> 按要求导入授权，授权成功后。在次访问<http://ip:port/a> 即可。

第 11 章 Tomcat 集群配置

1. nginx 安装

1) 安装之前首先确认系统中是否已安装 gcc、openssl-devel、pcre-devel、zlib-devel

```
rpm -qa pcre pcre-devel
```

```
rpm -qa|grep openssl
```

安装必备工具

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

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



```
Nothing to do
[root@vm-111 ~]# rpm -qa pcre pcre-devel
pcre-devel-8.32-15.el7_2.1.x86_64
pcre-8.32-15.el7_2.1.x86_64
[root@vm-111 ~]# rpm -qa|grep openssl
openssl-devel-1.0.1e-60.el7_3.1.x86_64
openssl-libs-1.0.1e-60.el7_3.1.x86_64
openssl-1.0.1e-60.el7_3.1.x86_64
[root@vm-111 ~]#
```

安装成功。

说明：

pcre: 用来作地址重写的功能。

zlib: nginx 的 gzip 模块, 传输数据打包, 省流量 (但消耗资源)。

openssl: 提供 ssl 加密协议。

2) 新建一个系统级用户组和匿名用户, 以及下面编译时使用

```
groupadd -r nginx
useradd -s /sbin/nologin -g nginx -r nginx-user
```

3) 编译安装 Nginx

解压 nginx 包到 /usr/local 目录 11

```
make clean

./configure \
--prefix=/etc/nginx \
--sbin-path=/usr/sbin/nginx \
--conf-path=/etc/nginx/nginx.conf \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
--pid-path=/var/run/nginx.pid \
--lock-path=/var/run/nginx.lock \
--http-client-temp-path=/var/cache/nginx/client_temp \
--http-proxy-temp-path=/var/cache/nginx/proxy_temp \
--http-fastcgi-temp-path=/var/cache/nginx/fastcgi_temp \
--http-uwsgi-temp-path=/var/cache/nginx/uwsgi_temp \
--http-scgi-temp-path=/var/cache/nginx/scgi_temp \
--user=nginx \
```

```
--group=nginx \  
--with-http_ssl_module \  
--with-http_realip_module \  
--with-http_addition_module \  
--with-http_sub_module \  
--with-http_dav_module \  
--with-http_flv_module \  
--with-http_mp4_module \  
--with-http_gunzip_module \  
--with-http_gzip_static_module \  
--with-http_random_index_module \  
--with-http_secure_link_module \  
--with-http_stub_status_module \  
--with-http_auth_request_module \  
--with-mail \  
--with-mail_ssl_module \  
--with-file-aio \  
--with-ipv6 \  
--with-cc-opt='-O2 -g -pipe -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector --param=ssp-buffer-size=4 -m64 -mtune=generic'
```

```
+ using system zlib library  
nginx path prefix: "/etc/nginx"  
nginx binary file: "/usr/sbin/nginx"  
nginx modules path: "/etc/nginx/modules"  
nginx configuration prefix: "/etc/nginx"  
nginx configuration file: "/etc/nginx/nginx.conf"  
nginx pid file: "/var/run/nginx.pid"  
nginx error log file: "/var/log/nginx/error.log"  
nginx http access log file: "/var/log/nginx/access.log"  
nginx http client request body temporary files: "/var/cache/nginx/client_temp"  
nginx http proxy temporary files: "/var/cache/nginx/proxy_temp"  
nginx http fastcgi temporary files: "/var/cache/nginx/fastcgi_temp"  
nginx http uwsgi temporary files: "/var/cache/nginx/uwsgi_temp"  
nginx http scgi temporary files: "/var/cache/nginx/scgi_temp"
```

完成后

```
make && make install
```



```
cp conf/scgi_params \
/etc/nginx/scgi_params.default'
test -f /etc/nginx/nginx.conf \
|| cp conf/nginx.conf /etc/nginx/nginx.conf'
cp conf/nginx.conf /etc/nginx/nginx.conf.default'
test -d /var/run \
|| mkdir -p /var/run'
test -d /var/log/nginx \
|| mkdir -p /var/log/nginx'
test -d /etc/nginx/html \
|| cp -R html /etc/nginx'
test -d /var/log/nginx \
|| mkdir -p /var/log/nginx'
make[1]: Leaving directory /usr/local/nginx-1.10.3'
[root@vm-111 nginx]#
```

安装成功后没报错，启动 nginx

```
nginx -c /etc/nginx/nginx.conf
```

如果报错

nginx: [emerg] getpwnam("nginx") failed

在 nginx.conf 中 把 user nobody 的注释去掉既可，如果提示

nginx: [emerg] mkdir() "/var/tmp/nginx/client_temp" failed (2: No such file or directory)

则用管理员去 mkdir 创建文件夹后再启动，如果安装成功可以访问<http://ip>

```
mkdir -p /var/tmp/nginx/client_temp
```

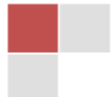
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

安装成功了。



2. nginx 的配置

配置参考 nginx.conf

```
sendfile        on;
#tcp_nopush     on;

#keepalive_timeout 0;
keepalive_timeout 65;

#gzip on;

server {
    listen 80;
    server_name localhost;
    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root html;
        index index.html index.htm;
        proxy_pass http://192.168.0.110:8080;
    }

    #error_page 404            /404.html;
}
```

配置对外域名地址

配置tomcat对应的ip和端口

nginx -c /etc/nginx/nginx.conf -s reload

nginx -t 配置文件是否正确

可能用到的命令：

firewall-cmd --state 查看防火墙是否开启

firewall-cmd --permanent --add-service=ftp

firewall-cmd --reload

systemctl start vsftpd

systemctl enable vsftpd

firewall-cmd --permanent --add-service=httpd

firewall-cmd --permanent --add-port=3306/tcp

```
systemctl stop firewalld
```

