|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 标题 | 版本 | 编写时间 | 更新时间 | 编写作者 | 备注 |
| IT Guide | 2.0 | 2018年4月26日 |  | Oliver |  |

[1. 创建云主机ECS](#_Toc31230)

[2. 安装iptables防火墙](#_Toc22567)

[3. 创建普通用户](#_Toc17403)

[4. 搭建LNMP环境](#_Toc6279)

[4.1 配置mariadb yum源](#_Toc4608)

[4.2 下载一些组件包，如果用源码安装太麻烦，可以4.3的方式yum源安装](#_Toc17066)

[4.3 安装开发工具包](#_Toc4634)

[4.4 安装mariadb数据库](#_Toc5492)

[4.5 启动mariadb并设置mariadb](#_Toc13464)

[4.6 创建nginx和php安装跟目录](#_Toc8451)

[4.7 安装libmcrypt](#_Toc28138)

[5. 源码安装php](#_Toc16816)

[5.1配置php初始配置和启动文件](#_Toc1056)

[5.2 修改php.ini配置文件](#_Toc31734)

[5.3 修改php-fpm.conf文件](#_Toc11697)

[5.4 修改php配置](#_Toc2229)

[5.5 启动php](#_Toc13113)

[6. 源码安装nginx](#_Toc7277)

[6.1 创建nginx运行用户](#_Toc5187)

[6.2编译安装nginx](#_Toc2399)

[6.3 配置nginx启动脚本（6、7都可以用）](#_Toc28515)

[6.4 针对centos7的启动脚本](#_Toc7101)

[7. 安装go环境](#_Toc12334)

[7.1 下载安装源包](#_Toc18510)

[7.2 解压安装](#_Toc22987)

[7.3 设置go环境变量](#_Toc18731)

[7.4 检查go版本](#_Toc15156)

[7.5 设置nginx反向代理go以及SSL证书](#_Toc24971)

[8. 安装redis数据库](#_Toc611)

[8.1 安装redis组件](#_Toc2223)

[8.2 下载redis源码包](#_Toc3582)

[8.3 配置redis文件](#_Toc32110)

[8.4 启动redis](#_Toc6456)

[8.5 修改redis.conf配置文件 vim /etc/redis.conf](#_Toc3832)

[8.6 修改 daemonize yes,以后台运行](#_Toc31758)

[8.7建立用户和日志目录](#_Toc3263)

[9. 搭建supervisor程序管理](#_Toc18945)

[9.1安装](#_Toc7855)

[9.2 创建存放配置文件的目录](#_Toc11254)

[9.3 创建一个golang的web server](#_Toc4123)

[9.4 在/etc/supervisor/conf.d下面创建go-http-server.ini](#_Toc30195)

[9.5 启动supervisor服务，注意：启动服务的时候，不需要启动goweb，不然会报错](#_Toc11008)

[9.6 查看supervisor运行状态](#_Toc16327)

[9.7 修改配置文件重新加载](#_Toc28313)

[10. iptables配置](#_Toc16349)

[11. 安装FTP](#_Toc855)

[11.1 在线yum安装](#_Toc25037)

[11.2 修改ftp配置](#_Toc8110)

[11.3 新建ftp用户](#_Toc7792)

[11.4 需要配置防火墙，开启21端口](#_Toc27751)

[11.5 启动vsftpd服务](#_Toc2425)

[12. 附上Nginx配置](#_Toc11342)

[13. 附上云尚互动服务器操作手册](#_Toc26846)

# 1. 创建云主机ECS

更新yum源，关闭firewall防火墙，采用iptables防火墙，阿里云服务器的selinux默认关闭，不要启动selinux，否则可能会导致云服务器网络问题

# yum update

# systemctl stop firewalld.service

# systemctl disable firewalld.service

# 2. 安装iptables防火墙

# yum install iptables-services

iptables配置文件在 /etc/sysconfig/iptables

开启iptables防火墙并使其开启启动

# systemctl enable iptables.service

# systemctl start iptables.service

# 3. 创建普通用户

# useradd shangtv

# password shangtv

# 4. 搭建LNMP环境

## 4.1 配置mariadb yum源

# echo > /etc/yum.repods.d/mariadb.repo << EOF

[mariadb]

Name = MariaDB

baseurl = <http://yum.mariadb.org/10.1.10/centos7-amd64>

gpgkey = https://yum.mariadb.org/RPM-GPG-KEY-MariaDB

gpgcheck = 1

EOF

备用yum源

#baseurl = http://yum.mariadb.org/10.1.22/centos7-amd64

#baseurl=http://mirrors.ctyun.cn/MariaDB/mariadb-10.1.22/yum/centos7-amd64/rpms/

#gpgkey=http://mirrors.ctyun.cn/MariaDB/yum/RPM-GPG-KEY-MariaDB

#http://ftp.hosteurope.de/mirror/archive.mariadb.org/mariadb-10.1.10/yum/centos7-amd64 无key

#

# yum clean all

# yum makecache

## 4.2 下载一些组件包，如果用源码安装太麻烦，可以4.3的方式yum源安装

软件源代码包存放位置：/usr/local/src

源码包编译安装位置：/usr/local/packegename

# cd /usr/local/src

#wget -ct 5 <http://nchc.dl.sourceforge.net/project/mcrypt/Libmcrypt/2.5.8/libmcrypt-2.5.8.tar.gz>

# wget http://www.tortall.net/projects/yasm/releases/yasm-1.3.0.tar.gz

# wget <http://www.ibiblio.org/pub/Linux/libs/graphics/t1lib-5.1.2.tar.gz>

# wget <https://bitbucket.org/libgd/gd-libgd/downloads/libgd-2.1.0.tar.gz>

# wget <http://download.osgeo.org/libtiff/tiff-4.0.3.tar.gz>

# wget <http://ring.u-toyama.ac.jp/archives/graphics/freetype/freetype2/freetype-2.5.4.tar.gz>

# wget <http://www.ijg.org/files/jpegsrc.v9a.tar.gz>

# wget http://curl.haxx.se/download/curl-7.44.0.tar.gz

# wget https://webm.googlecode.com/files/libvpx-v1.3.0.tar.bz2

# wget -ct 5 <http://www.openssl.org/source/openssl-1.0.1i.tar.gz>

# wget http://www.atomicorp.com/installers/atomic

# wget http://cn2.php.net/distributions/php-7.1.4.tar.gz

# wget http://nginx.org/download/nginx-1.9.15.tar.gz

# git clone https://github.com/miyanaga/nginx-requestkey-module.git

# git clone <https://github.com/arut/nginx-rtmp-module.git>

## 4.3 安装开发工具包

# yum -y groupinstall "Development Tools"

# yum install -y libxml2 libxml2-devel libcrul libcurl-devel gd gd-devel libpng libpng-devel wget apr\* autoconf automake bison bzip2 bzip2\* cloog-ppl compat\* cpp curl curl-devel fontconfig fontconfig-devel freetype freetype\* freetype-devel gcc gcc-c++ gtk+-devel gd gettext gettext-devel glibc kernel kernel-headers keyutils keyutils-libs-devel krb5-devel libcom\_err-devel libpng libpng-devel libjpeg\* libsepol-devel libselinux-devel libstdc++-devel libtool\* libgomp libxml2 libxml2-devel libXpm\* libtiff libtiff\* make mpfr ncurses\* ntp openssl openssl-devel patch pcre pcre-devel perl php-common php-gd policycoreutils telnet t1lib t1lib\* nasm nasm\* zlib-devel gd-devel

## 4.4 安装mariadb数据库

#yum install mariadb\* -y

报错：

libJudy.so.1()(64bit) for package: MariaDB-oqgraph-engine-10.1.22-1.el7.centos.x86\_64 base

需要安装libjudy

#wget <http://ftp.tu-chemnitz.de/pub/linux/dag/redhat/el7/en/x86_64/rpmforge/RPMS/judy-1.0.5-1.el7.rf.x86_64.rpm>

#wget <http://ftp.tu-chemnitz.de/pub/linux/dag/redhat/el7/en/x86_64/rpmforge/RPMS/judy-devel-1.0.5-1.el7.rf.x86_64.rpm>

# yum localinstall judy\*

## 4.5 启动mariadb并设置mariadb

# systemctl start mariadb

#systemctl enable mariadb

# mysql\_secure\_installation #按照指示设置mariadb数据库

Mariadb的安装目录在 /var/lib/mysql，配置文件/etc/my.cnf.d/\*

## 4.6 创建nginx和php安装跟目录

# mkdir /usr/local/php

# mkdir /usr/local/nginx

## 4.7 安装libmcrypt

# cd /usr/local/src

# chmod +x atomic

# ./atomic

# yum install php-mcrypt libmcrypt libmcrypt-devel

# 5. 源码安装php

# groupadd php-fpm

# useradd -g php-fpm php-fpm -s /bin/false

# cd /usr/loca/src

# tar zxvf php-7.1.4.tar.gz

# cd php-7.1.4

./configure --prefix=/usr/local/php \

--with-config-file-path=/usr/local/php/etc \

--enable-mysqlnd \

--with-mysql=shared,mysqlnd \

--with-mysqli=shared,mysqlnd \

--with-pdo-mysql=shared,mysqlnd \

--with-mysql-sock=/var/lib/mysql/mysql.sock \

--with-mysqli=/usr/bin/mysql\_config \

--with-gd \

--with-png-dir \

--with-jpeg-dir \

--with-freetype-dir \

--with-xpm-dir \

--with-zlib-dir \

--with-iconv \

--enable-fpm \

--with-fpm-user=php-fpm \

--with-fpm-group=php-fpm \

--enable-libxml \

--enable-xml \

--enable-bcmath \

--enable-shmop \

--enable-sysvsem \

--enable-inline-optimization \

--enable-opcache \

--enable-mbregex \

--enable-mbstring \

--enable-ftp \

--enable-gd-native-ttf \

--with-openssl \

--enable-pcntl \

--enable-sockets \

--with-xmlrpc \

--enable-zip \

--enable-soap \

--without-pear \

--with-gettext \

--enable-session \

--with-mcrypt \

--with-curl \

--enable-exif \

--with-mhash \

--enable-ctype

# make && make install

## 5.1配置php初始配置和启动文件

# cp /usr/local/src/php-7.1.4/php.ini-production /usr/local/php/etc/php.ini

# cp /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf

# mv /usr/local/php/etc/php-fpm.d/www.conf.default /usr/local/php/etc/php-fpm.d/www.conf

# /usr/local/php/sbin/php-fpm -t

# cp /usr/local/src/php-7.1.4/sapi/fpm/init.d.php-fpm /etc/rc.d/init.d/php-fpm

# chmod +x /etc/rc.d/init.d/php-fpm

# chkconfig php-fpm on

# echo 'export PATH=$PATH:/usr/local/php/bin' >> /etc/profile

# source /etc/profile

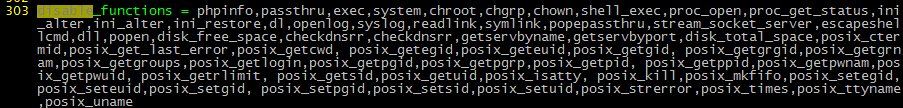
## 5.2 修改php.ini配置文件

# vim /usr/local/php/etc/php.ini

找到"disable\_functions =" （禁用掉某些比较“危险”函数，大概在301行），改为

Disable\_functions = phpinfo,passthru,exec,system,chroot,chgrp,chown,shell\_exec,proc\_open,

proc\_get\_status,ini\_alter,ini\_alter,ini\_restore,dl,openlog,syslog,readlink,symlink,popepassthru,stream\_socket\_server,escapeshelcmd,dll,popen,disk\_free\_space,checkdnsrr,checkdnsrr,getservbyname,getservbyport,disk\_total\_space,posix\_ctermid,posix\_get\_last\_error,posix\_getcwd,posix\_getegid,posix\_geteuid,posix\_getgid,posix\_getgrgid,posix\_getgrnam,posix\_getgroups,posix\_getlogin,posix\_getpgid,posix\_getpgrp,posix\_getpid,posix\_getppid,posix\_getpwnam,posix\_getpwuid,posix\_getrlimit,posix\_getsid,posix\_getuid,posix\_isatty,posix\_kill,posix\_mkfifo,posix\_setegid,posix\_seteuid,posix\_setgid,posix\_setpgid,posix\_setsid,posix\_setuid,posix\_strerror,posix\_times,posix\_ttyname,posix\_uname



找到 ;date.timezone （大概在920行），修改为 date.timezone = Asia/Shanghai

找到 expose\_php = On （禁止显示php版本信息），修改为 expose\_php = Off

找到 short\_open\_tag = Off （支持php短标签），修改为 short\_open\_tag = On

找到 opcache.enable = 0 （支持opcode缓存），修改为 opcache.enable = 1

找到 zend\_extension = “opcache.so”下面添加一下内容，开启opcode缓存功能

zend\_extension = "opcache.so"

opcache.memory\_consumption=128

opcache.interned\_strings\_buffer=8

opcache.max\_accelerated\_files=4000

opcache.revalidate\_freq=60

opcache.fast\_shutdown=1

opcache.enable\_cli=1

支持php\_mysql

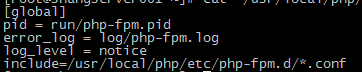
extension=/usr/local/php/lib/php/extensions/no-debug-non-zts-20151012/pdo\_mysql.so

## 5.3 修改php-fpm.conf文件

# vim /usr/local/php/etc/php-fpm.conf

取消pid前面的分号

pid = rum/php-fpm.pid



## 5.4 修改php配置

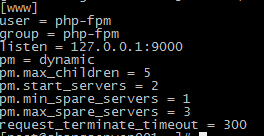
# vim /usr/local/php/etc/php-fpm.d/www.conf

#设置php-fpm运行账号为php-fpm

user = php-fpm

#设置php-fpm运行组为php-fpm

group = php-fpm



## 5.5 启动php

# /etc/init.d/php-fpm restart

# chkconfig php-fpm on

# 6. 源码安装nginx

# yum install -y pcre pcre-devel openssl openssl-devel

## 6.1 创建nginx运行用户

# groupadd nginx

# useradd -g nginx nginx s /bin/false

# mkdir /var/cache/nginx

## 6.2编译安装nginx

# cd /usr/local/src

# tar zxvf nginx-1.9.15.tar.gz

# cd nginx-1.9.15

# ./configure --prefix=/usr/local/nginx \

--http-client-body-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-pcre \

--with-http\_v2\_module \

--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-http\_v2\_module \

--with-threads \

--with-stream \

#添加模块

--add-module=/usr/local/src/nginx-requestkey-module/ \

--add-module=/usr/local/src/nginx-rtmp-module/

# make && make install

## 6.3 配置nginx启动脚本（6、7都可以用）

# cat >> /etc/rd.d/init.d/nginx << EOF

#!/bin/sh

#

# nginx - this script starts and stops the nginx daemon

#

# chkconfig: - 85 15

# description: Nginx is an HTTP(S) server, HTTP(S) reverse \

# proxy and IMAP/POP3 proxy server

# processname: nginx

# config: /usr/local/nginx/conf/nginx.conf

# pidfile: /usr/local/nginx/logs/nginx.pid

# Source function library.

. /etc/rc.d/init.d/functions

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

[ "$NETWORKING" = "no" ] && exit 0

nginx="/usr/local/nginx/sbin/nginx"

prog=$(basename $nginx)

NGINX\_CONF\_FILE="/usr/local/nginx/conf/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/subsys/nginx

make\_dirs() {

# make required directories

user=`$nginx -V 2>&1 | grep "configure arguments:" | sed 's/[^\*]\*--user=\([^ ]\*\).\*/\1/g' -`

if [ -z "`grep $user /etc/passwd`" ]; then

useradd -M -s /bin/nologin $user

fi

options=`$nginx -V 2>&1 | grep 'configure arguments:'`

for opt in $options; do

if [ `echo $opt | grep '.\*-temp-path'` ]; then

value=`echo $opt | cut -d "=" -f 2`

if [ ! -d "$value" ]; then

# echo "creating" $value

mkdir -p $value && chown -R $user $value

fi

fi

done

}

start() {

[ -x $nginx ] || exit 5

[ -f $NGINX\_CONF\_FILE ] || exit 6

make\_dirs

echo -n $"Starting $prog: "

daemon $nginx -c $NGINX\_CONF\_FILE

retval=$?

echo

[ $retval -eq 0 ] && touch $lockfile

return $retval

}

stop() {

echo -n $"Stopping $prog: "

killproc $prog -QUIT

retval=$?

echo

[ $retval -eq 0 ] && rm -f $lockfile

return $retval

}

restart() {

#configtest || return $?

stop

sleep 1

start

}

reload() {

#configtest || return $?

echo -n $"Reloading $prog: "

killproc $nginx -HUP

RETVAL=$?

echo

}

force\_reload() {

restart

}

configtest() {

$nginx -t -c $NGINX\_CONF\_FILE

}

rh\_status() {

status $prog

}

rh\_status\_q() {

rh\_status >/dev/null 2>&1

}

case "$1" in

start)

rh\_status\_q && exit 0

$1

;;

stop)

rh\_status\_q || exit 0

$1

;;

restart|configtest)

$1

;;

reload)

rh\_status\_q || exit 7

$1

;;

force-reload)

force\_reload

;;

status)

rh\_status

;;

condrestart|try-restart)

rh\_status\_q || exit 0

;;

\*)

Echo $"Usage: $0{start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"

exit 2

esac

EOF

# chmod 755 /etc/rc.d/init.d/nginx

# chkconfig nginx on

# /etc/rc.d/init.d/nginx start

# echo 'export PATH=$PATH:/usr/local/nginx/sbin' >> /etc/profile

# source /etc/profile

## 6.4 针对centos7的启动脚本

cat >> /usr/lib/systemd/system/nginx.service << EOF

[Unit]

Description=nginx - high performance web server

Documentation=http://nginx.org/en/docs/

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

[Service]

Type=forking

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

ExecStartPre=/usr/sbin/nginx -t

ExecStart=/usr/sbin/nginx

ExecReload=/usr/sbin/nginx -s reload

ExecStop=/usr/sbin/nginx -s stop

PrivateTmp=true

[Install]

WantedBy=multi-user.target

EOF

systemctl daemon-reload

systemctl start nginx

systemctl enable nginx

systemctl status nginx

# 7. 安装go环境

## 7.1 下载安装源包

# cd /usr/local/src

# wget <http://golangtc.com/static/go/1.8/go1.8.linux-amd64.tar.gz>

## 7.2 解压安装

# cd /usr/local/src

# tar zxvf go1.8.linux-amd64.tar.gz

# cp -a go /usr/local/

# mkdir /data/ #创建go项目工作目录

# setfacl -R -m o::r-- /data

# setfacl -R -m g::rw- /data

# setfacl -R -m u:shangtv:rwx /data

# setfacl -R -d --set o::r-- /data

# setfacl -R -d --set g::rw- /data

# setfacl -R -d --set u:shangtv:rwx /data

# mkdir /data/mygo

## 7.3 设置go环境变量

# echo "export GOROOT=/usr/local/go" >> /etc/profile

# echo "export GOBIN=$GOROOT/bin" >> /etc/profile

# echo "export GOPKG=$GOROOT/pkg/tool/linux\_amd64" >> /etc/profile

# echo "export GOARCH=amd64" >> /etc/profile

# echo "export GOOS=linux" >> /etc/profile

# echo "export GOPATH=/data/mygo/" >> /etc/profile

# echo "export PATH=.:$PATH:$GOBIN:$GOPKG" >> /etc/profile

## 7.4 检查go版本

# go version

go version go1.8 linux/amd64

## 7.5 设置nginx反向代理go以及SSL证书

代理go语言的web架构，需要在nginx做特别配置才能支持

证书存放在/usr/local/nginx/ssl/hdt/Nginx/

map $http\_upgrade $connection\_upgrade {

default upgrade;

'' close;

}

upstream hdtserver {

#server 120.76.40.33:8082 backup;

server 47.75.16.43:3003;

}

server {

listen 443 ssl;

server\_name hdt.shangtv.cn;

server\_tokens off;

access\_log logs/hdt\_https\_access.log main;

error\_log logs/hdt\_https\_error.log error;

add\_header Access-Control-Allow-Origin \*;

ssl\_certificate /usr/local/nginx/ssl/hdt/Nginx/1\_hdt.shangtv.cn\_bundle.crt;

ssl\_certificate\_key /usr/local/nginx/ssl/hdt/Nginx/2\_hdt.shangtv.cn.key;

ssl\_session\_timeout 5m;

ssl\_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;

ssl\_prefer\_server\_ciphers on;

location / {

#limit\_conn addr 5; #设置nginx连接并发

#limit\_req zone=hdt\_req\_zone burst=5;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection $connection\_upgrade;

proxy\_pass\_header Server;

proxy\_pass https://hdtserver;

proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto https;

proxy\_redirect off;

}

}

# 8. 安装redis数据库

## 8.1 安装redis组件

# yum install tcl

## 8.2 下载redis源码包

# cd /usr/local/src

# wget http://download.redis.io/releases/redis-3.0.7.tar.gz

# tar -zvxf redis-3.0.7.tar.gz

# cd redis-3.0.7

# make

# make install #按提示install

# cd src/

# make install

下面提示就说明成功

Hint: To run 'make test' is a good idea ;)

INSTALL install

INSTALL install

INSTALL install

INSTALL install

INSTALL install

以上redis已完成编译安装

在/usr/local/bin生成redis-benchmark、redis-check-aof、redis-check-dump、redis-cli、redis-sentinel 、redis-server六个文件，其中redis-server为启动服务

## 8.3 配置redis文件

# cp /usr/local/src/redis-3.0.7/redis.conf /etc/

# chmod 755 /etc/redis.conf

## 8.4 启动redis

# /usr/local/bin/redis-server /etc/redis.conf

## 8.5 修改redis.conf配置文件 vim /etc/redis.conf

下面是redis.conf的主要配置参数的意义：

daemonize：是否以后台daemon方式运行

pidfile：pid文件位置

port：监听的端口号

timeout：请求超时时间

loglevel：log信息级别

logfile：log文件位置

databases：开启数据库的数量

save \* \*：保存快照的频率，第一个\*表示多长时间，第三个\*表示执行多少次写操作。 在一定时间内执行一定数量的写操作时，自动保存快照。可设置多个条件。

rdbcompression：是否使用压缩

dbfilename：数据快照文件名（只是文件名，不包括目录）

dir：数据快照的保存目录（这个是目录）

appendonly：是否开启appendonlylog，开启的话每次写操作会记一条log，这会提高数 据抗风险能力，但影响效率。

appendfsync：appendonlylog如何同步到磁盘（三个选项，分别是每次写都强制调用fsync、每秒启用一次fsync、不调用fsync等待系统自己同步）

## 8.6 修改 daemonize yes,以后台运行

这时你可以打开一个终端进行测试了，配置文件中默认的监听端口是6379

## 8.7建立用户和日志目录

第一次启动时建议为Redis建立用户和日志目录

# groupadd redis

# useradd -g redis redis -s /bin/false

# mkdir -p /var/lib/redis #db文件放在这里，需要修改redis.conf

# mkdir -p /var/log/redis #日志放在这里

# chown redis.redis /var/lib/redis

# chown redis.redis /var/log/redis

# vim /etc/redis.conf

# The working directory.

#

# The DB will be written inside this directory, with the filename specified

# above using the 'dbfilename' configuration directive.

#

# Also the Append Only File will be created inside this directory.

#

# Note that you must specify a directory here, not a file name.

dir /var/lib/redis

---------------------------------------------------------------------------------------

找到日志logfile

# Specify the log file name. Also 'stdout' can be used to force

# Redis to log on the standard output. Note that if you use standard

# output for logging but daemonize, logs will be sent to /dev/null

logfile /var/log/redis/redislog

或者编写启动脚本，加入到服务

配置Init脚本

Redis管理脚本基于Ubuntu 的发行版上的，在Centos linux 上并不能用，下面有个脚本可以用于CentOS 。

用这个脚本管理之前，需要先配置下面的内核参数，否则Redis脚本在重启或停止redis时，将会报错，并且不能自动在停止服务前同步数据到磁盘上：

# vim /etc/sysctl.conf

vm.overcommit\_memory = 1

然后应用生效：

# sysctl -p

建立redis启动脚本：

# vim /etc/init.d/redis

#!/bin/bash

#

# Init file for redis

#

# chkconfig: - 80 12

# description: redis daemon

#

# processname: redis

# config: /etc/redis.conf

# pidfile: /var/run/redis.pid

source /etc/init.d/functions

#BIN="/usr/local/bin"

BIN="/usr/local/bin"

CONFIG="/etc/redis.conf"

PIDFILE="/var/run/redis.pid"

### Read configuration

[ -r "$SYSCONFIG" ] && source "$SYSCONFIG"

RETVAL=0

prog="redis-server"

desc="Redis Server"

start() {

if [ -e $PIDFILE ];then

echo "$desc already running...."

exit 1

fi

echo -n $"Starting $desc: "

daemon $BIN/$prog $CONFIG

RETVAL=$?

echo

[ $RETVAL -eq 0 ] && touch /var/lock/subsys/$prog

return $RETVAL

}

stop() {

echo -n $"Stop $desc: "

killproc $prog

RETVAL=$?

echo

[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$prog $PIDFILE

return $RETVAL

}

restart() {

stop

start

}

case "$1" in

start)

start

;;

stop)

stop

;;

restart)

restart

;;

condrestart)

[ -e /var/lock/subsys/$prog ] && restart

RETVAL=$?

;;

status)

status $prog

RETVAL=$?

;;

\*)

echo $"Usage: $0 {start|stop|restart|condrestart|status}"

RETVAL=1

esac

exit $RETVAL

然后增加服务并开机自启动：

# chmod 755 /etc/init.d/redis

# chkconfig --add redis

# chkconfig --level 345 redis on

# chkconfig --list redis

# /etc/init.d/redis restart

# 9. 搭建supervisor程序管理

## 9.1安装

系统：centos7，默认在root用户下安装，使用普通用户，可使用sudo

# yum install python-setuptools -y

# easy\_install supervisor #如果没看到报错信息，就说明安装成功了，

# echo\_supervisord\_conf #查看配置详情，而后生成配置文件

# mkdir -m 755 -p /etc/supervisor/

# echo\_supervisord\_conf > /etc/supervisor/supervisord.conf

## 9.2 创建存放配置文件的目录

# mkdir -m 755 /etc/supervisor/conf.d

在主配置文件中引入配置

# vim /etc/supervisor/supervisord.conf

注释去掉

[include]

files = ./conf.d/\*.ini

## 9.3 创建一个golang的web server

程序放在/data/mygo/ 下面

先整一个简单的golang http服务，用于测试，生产环境直接使用服务端的二进制程序

---------------------------------

package main

import (

"fmt"

"log"

"net/http"

)

func main() {

http.HandleFunc("/", func(w http.ResponseWriter, r \*http.Request) {

fmt.Fprintf(w, "Hello world")

})

err := http.ListenAndServe(":9090", nil)

if err != nil {

log.Fatal("ListenAndServe: ", err)

}

}

----------------------------------------

# go build

## 9.4 在/etc/supervisor/conf.d下面创建go-http-server.ini

;--------------------------------------------------------

[program:yshd]

directory = /data/mygo

command= /data/mygo/yshd

user=root

process\_name= yshd\_game

autostart=true

autorestart=true

startretries=3

startsecs=10

stdout\_logfile=/var/log/yshd\_fatmouse-server.log

stdout\_logfile\_maxbytes=1MB

stdout\_logfile\_backups=10

stdout\_capture\_maxbytes=1MB

stderr\_logfile=/var/log/yshd\_fatmouse-server.log

stderr\_logfile\_maxbytes=1MB

stderr\_logfile\_backups=10

stderr\_capture\_maxbytes=1MB

;-----------------------------------------------------------

## 9.5 启动supervisor服务，注意：启动服务的时候，不需要启动goweb，不然会报错

# /usr/bin/supervisord -c /etc/supervisor/supervisord.conf

## 9.6 查看supervisor运行状态

#supervisorctl status



## 9.7 修改配置文件重新加载

# supervisorctl reload #加载新的配置

# supervisorctl shutdown #关闭supervisor

# supervisorctl restart #重启supervisor

# 10. iptables配置

# vim /etc/sysconfig/iptables

# Generated by iptables-save v1.4.21 on Thu Mar 31 11:36:06 2016

\*filter

:INPUT ACCEPT [23077:17524766]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [19534:6305027]

-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 21 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 25 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 8082 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 8083 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 8092 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 1935 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 3306 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 6379 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 9000 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 3000 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 3001 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 3002 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 3003 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 10050 -j ACCEPT

-A INPUT -p tcp -m tcp --dport 10051 -j ACCEPT

-A INPUT -p udp -m udp --dport 10050 -j ACCEPT

-A INPUT -p udp -m udp --dport 10051 -j ACCEPT

COMMIT

# systemctl restart iptables.service

# iptables -L -n

# 11. 安装FTP

## 11.1 在线yum安装

# yum install vsftpd -y

#mkdir /ftp

## 11.2 修改ftp配置

# vim /etc/vsftpd/vsftpd.conf

anonymous\_enable=NO

local\_enable=YES

write\_enable=YES

local\_umask=022

anon\_upload\_enable=YES

anon\_mkdir\_write\_enable=YES

dirmessage\_enable=YES

xferlog\_enable=YES

connect\_from\_port\_20=YES

xferlog\_file=/var/log/vsftpd.log

xferlog\_std\_format=NO

ftpd\_banner=欢迎进入云尚互动FTP服务.

chroot\_local\_user=YES

local\_root=/ftp

allow\_writeable\_chroot=yes

chroot\_list\_file=/etc/vsftpd/chroot\_list

listen=no

listen\_ipv6=YES

pam\_service\_name=vsftpd

userlist\_enable=YES

tcp\_wrappers=YES

## 11.3 新建ftp用户

# useradd -d /home/ftp -g ftp -s /bin/false ftpuser

# passwd ftpuser

## 11.4 需要配置防火墙，开启21端口

## 11.5 启动vsftpd服务

# systemctl start vsftpd

# systemctl enable vsftpd

# 12. 附上Nginx配置

Nginx 配置文件/usr/local/nginx/conf/



Nginx 虚拟站点配置文件 /usr/local/nginx/conf.d/



# 附上云尚互动服务器操作手册

