牡丹江云平台学情分析
系统部署
及服务启动手册

公　司：北京捷成世纪科技股份有限公司

地　址：北京市海淀区知春路 1 号学院国际大厦 7 层

网　址：http://www.jetsen.com.cn

日　期：2016 年 9 月 26

编写日期：2016-09-23
编写作者：马文博
邮　　箱：mawenbo@jetsen.cn

部署环境 （*目前与流媒体安装在同一台机器上）
操作系统: ：　　CentOS Linux release 7.2.1511 (Core)
JDK　版本 ：　　jdk-7u79-linux-x64.gz
Ngnix 版本：　　nginx-1.6.2.tar.gz
Redis 版本：　　redis-3.0.0-rc5.tar.gz
Tomcat 版本：　　apache-tomcat-7.0.70.tar.gz
操作用户：　　root
密　　码：　　MDJEdu!@#
公网 IP：　　218.9.54.44
内外 IP：　　172.15.110.44
公网端口　　8888-》80
公网端口　　9999-》22

1. 基础软件环境的搭建

1.1. JDK 环境变量配置（JDK1.7 以上）

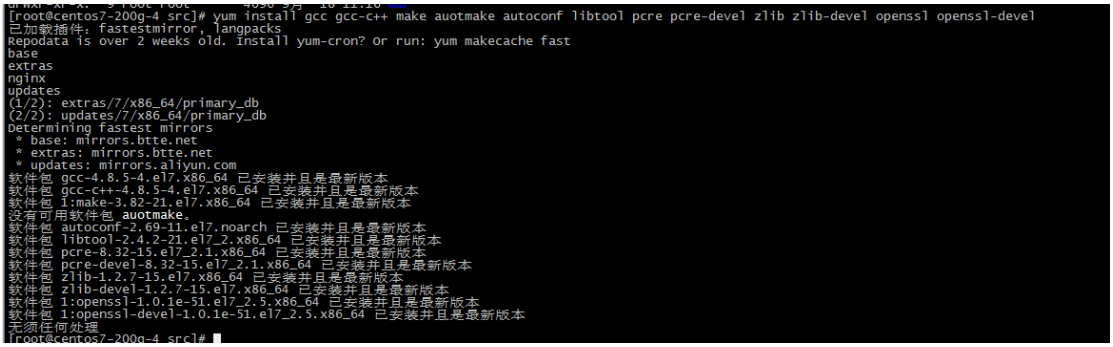
[root@bogon local]# tar -zxvf jdk-7u79-linux-x64.gz

[root@bogon jdk1.7.0_79]# vim /etc/profile
JAVA_HOME=/usr/local/jdk1.7.0_79
PATH=$JAVA_HOME/bin: $PATH
CLASSPATH=$JAVA_HOME/jre/lib/ext:$JAVA_HOME/lib/tools.jar
export PATH JAVA_HOME CLASSPATH

[root@bogon jdk1.7.0_79]# source /etc/profile

1.2. Nginx 环境变量配置（JDK1.7 以上）

安装编译 Nginx 所依赖的包
[root@centos7-200g-4 src]# yum install gcc gcc-c++ make auotmake autoconf libtool pcre pcre-devel zlib zlib-devel openssl openssl-devel



上传 Nginx （nginx-1.6.2.tar.gz）安装包到/usr/local/src 目录，编译安装

[root@centos7-200g-4 src]# cd /usr/local/src/

[root@centos7-200g-4 src]# tar -zxvf nginx-1.6.2.tar.gz

[root@centos7-200g-4 src]# cd nginx-1.6.2/

[root@centos7-200g-4 nginx-1.6.2]# ./configure --prefix=/usr/local/nginx

[root@centos7-200g-4 nginx-1.6.2]# make && make install

配置 Nginx

[root@centos7-200g-4 nginx-1.6.2]# vim /usr/local/nginx/conf/nginx.conf

#user nobody;

worker_processes 4;

#error_log logs/error.log;

#error_log logs/error.log notice;

#error_log logs/error.log info;

#pid logs/nginx.pid;

#一个 nginx 进程打开的最多文件描述符数目，理论值应该是最多打开文件数（系统的值 ulimit -n）与 nginx 进程数相除，但是 nginx 分配请求并不均匀，所以建议与 ulimit -n 的值保持一致。

worker_rlimit_nofile 65535;

#

##工作模式与连接数上限

events

{

##参考事件模型，use [kqueue | rtsig | epoll | /dev/poll | select | poll]; epoll 模型是 Linux 2.6 以上版本内核中的高性能网络 I/O 模型，如果跑在 FreeBSD 上面，就用 kqueue 模型。

use epoll;

##单个进程最大连接数（最大连接数=连接数*进程数）

worker_connections 65535;

}

http {

```
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;
```

#FastCGI 相关参数是为了改善网站的性能：减少资源占用，提高访问速度。下面参数看字面意思都能理解。

```
    fastcgi_connect_timeout 300;
    fastcgi_send_timeout 300;
    fastcgi_read_timeout 300;
    fastcgi_buffer_size 64k;
    fastcgi_buffers 4 64k;
    fastcgi_busy_buffers_size 128k;
    fastcgi_temp_file_write_size 128k;

    #gzip  on;
```

#gzip 模块设置

```
    gzip on; #开启 gzip 压缩输出
    gzip_min_length 1k; #最小压缩文件大小
    gzip_buffers 4 16k; #压缩缓冲区
    gzip_http_version 1.0; #压缩版本（默认 1.1，前端如果是 squid2.5 请使用 1.0）
    gzip_comp_level 2; #压缩等级
    gzip_types text/plain application/x-javascript text/css application/xml;
```

##压缩类型，默认就已经包含 text/html，所以下面就不用再写了，写上去也不会有问题，但是会有一个 warn。

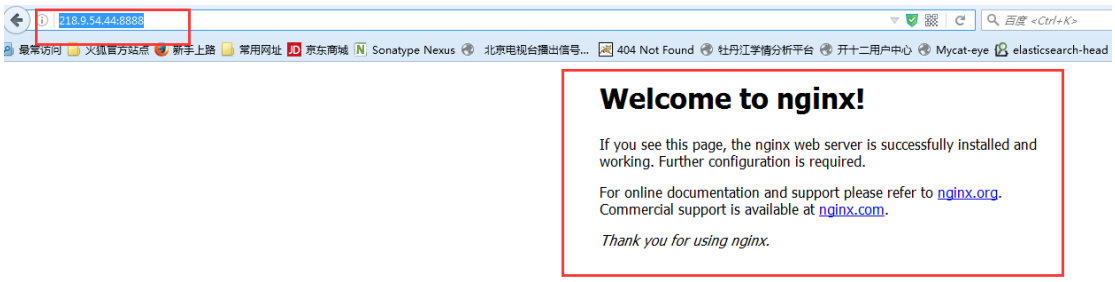
```
    gzip_vary on;

    server {
        listen       80;
        server_name  localhost;

        #charset koi8-r;
```

```
        #access_log  logs/host.access.log  main;

        location / {
            root   html;
            index  index.html index.htm;
        }

        location /analysis {
            proxy_pass http://172.15.110.44:8080/analysis;
            proxy_redirect off;
            proxy_set_header Host $http_host;
            proxy_set_header Cookie $http_cookie;
            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 $scheme;
            client_max_body_size 100m; #允许客户端请求的最大单文件字节数
            client_body_buffer_size 128k; #缓冲区代理缓冲用户端请求的最大字节数
            proxy_connect_timeout  90;     #nginx  跟后端服务器连接超时时间(代理连接超时)
            proxy_read_timeout 90;         #连接成功后，后端服务器响应时间(代理接收超时)
            proxy_buffer_size 4k;          #设置代理服务器（nginx）保存用户头信息的缓冲区大小
            proxy_buffers 6 32k;           #proxy_buffers 缓冲区，网页平均在 32k 以下的话，这样设置
            proxy_busy_buffers_size 64k; #高负荷下缓冲大小（proxy_buffers*2）
            proxy_temp_file_write_size  64k;  #设定缓存文件夹大小，大于这个值，将从 upstream 服务器传
        }

        location
~ .*\.(gif|jpg|jpeg|png|bmp|swf|ioc|rar|zip|txt|flv|mid|doc|ppt|pdf|xls|mp3|wma)$
        {
            root /usr/local/nginx/html;
            expires      30d;
        }

        location ~ .*\.(js|css)?$
        {
            root /usr/local/nginx/html;
            expires          1d;
        }
```

```
    }
}
```

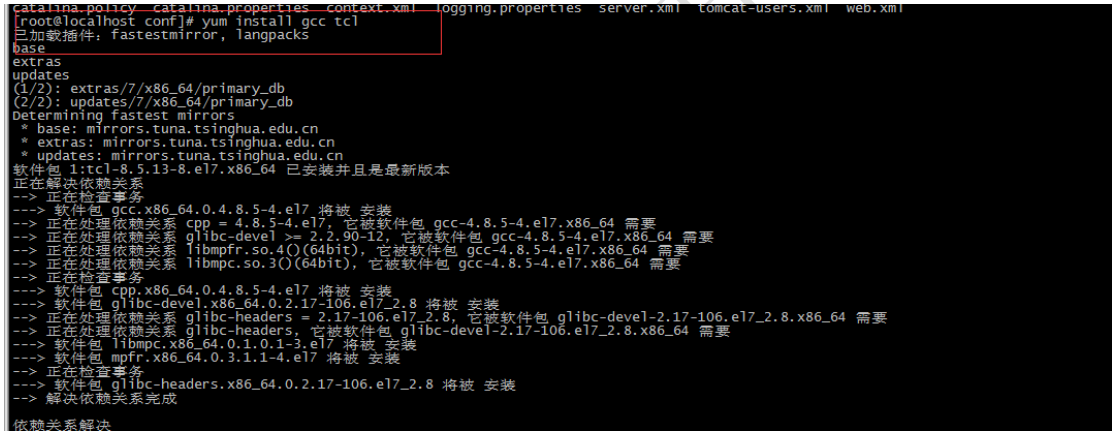
启动 Ngnix

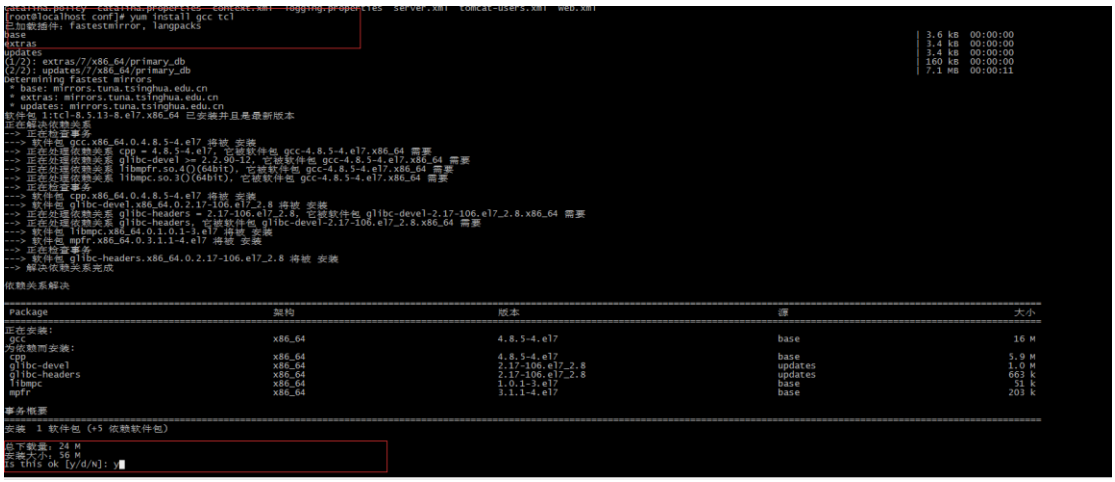
[root@centos7-200g-4 nginx-1.6.2]# /usr/local/nginx/sbin/nginx

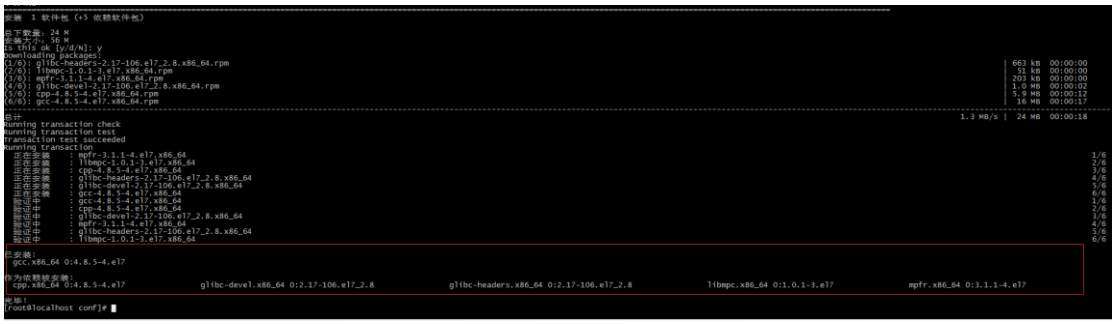


1.3. Redis3.0 单机版安装（目前使用本机的 Redis）

编译安装所需要的包：[root@localhost conf]# yum install gcc tcl



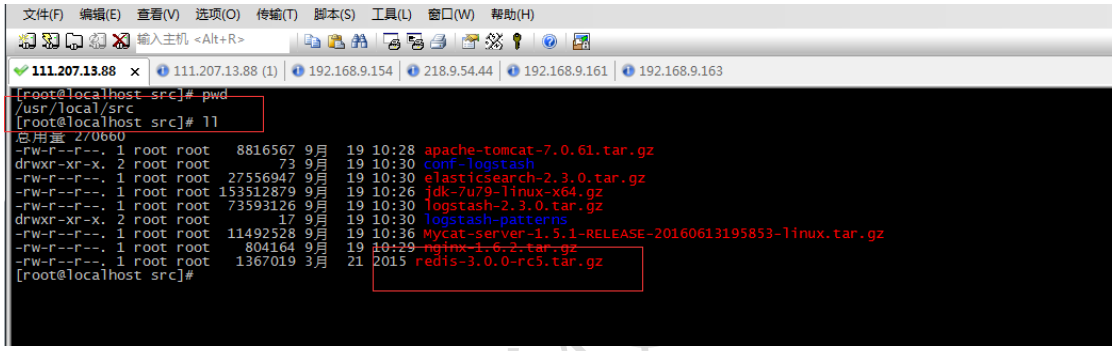




创建安装目录

[root@localhost conf]# mkdir /usr/local/redis

解压 redis-3.0.0-rc5.tar.gz



[root@localhost src]# tar -zxvf redis-3.0.0-rc5.tar.gz

[root@localhost src]# mv redis-3.0.0-rc5 redis3.0

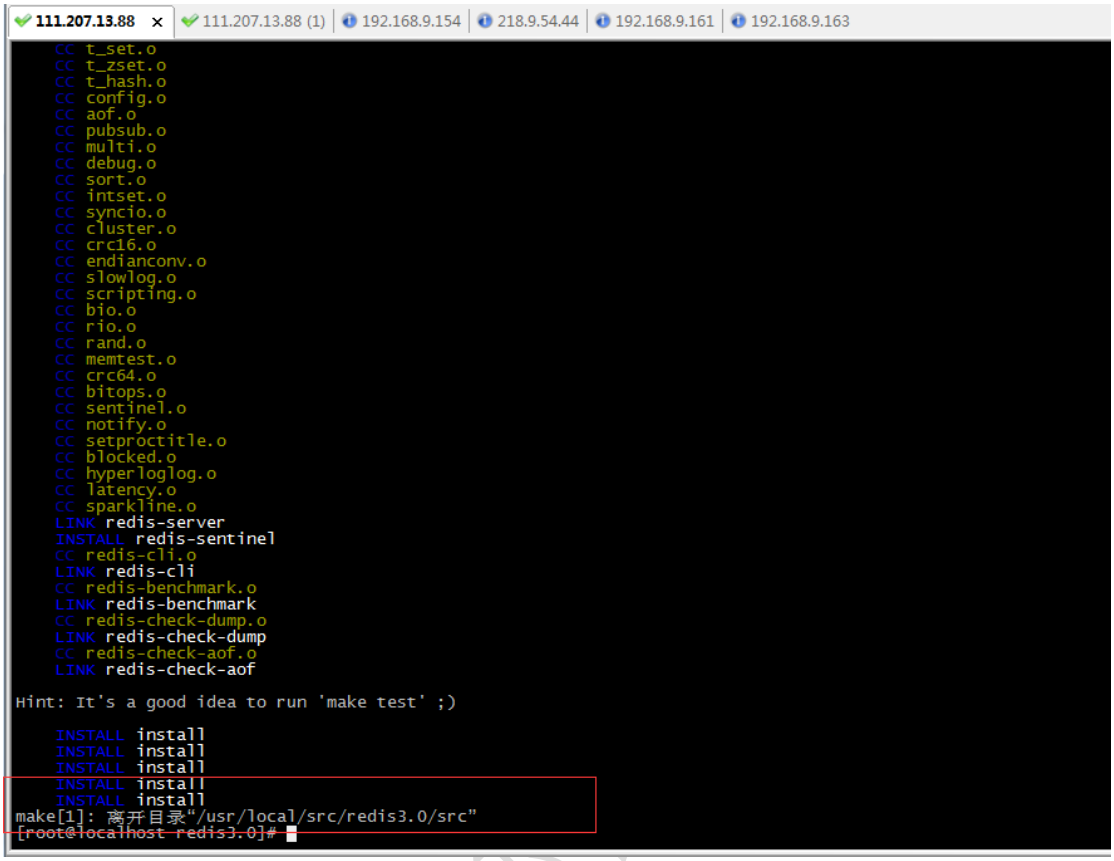
[root@localhost src]# cd redis3.0/

安装（使用 PERFIX 指定安装目录）：

```
[root@localhost src]# mv redis-3.0.0-rc5 redis3.0
[root@localhost src]# cd redis3.0/
[root@localhost redis3.0]# make PREFIX=/usr/local/redis install
cd src && make install
make[1]: 进入目录“/usr/local/src/redis3.0/src”
rm -rf redis-server redis-sentinel redis-cli redis-benchmark redis-check-dump redis-check-aof *.o *.gcda *
(cd ../deps && make distclean)
make[2]: 进入目录“/usr/local/src/redis3.0/deps”
(cd hiredis && make clean) > /dev/null || true
(cd linenoise && make clean) > /dev/null || true
(cd lua && make clean) > /dev/null || true
(cd jemalloc && [ -f Makefile ] && make distclean) > /dev/null || true
(rm -f .make-*)
make[2]: 离开目录“/usr/local/src/redis3.0/deps”
(rm -f .make-*)
echo STD=-std=c99 -pedantic >> .make-settings
echo WARN=-Wall -W >> .make-settings
echo OPT=-O2 >> .make-settings
echo MALLOC=jemalloc >> .make-settings
echo CFLAGS= >> .make-settings
echo LDFLAGS= >> .make-settings
echo REDIS_CFLAGS= >> .make-settings
echo REDIS_LDFLAGS= >> .make-settings
echo PREV_FINAL_CFLAGS=-std=c99 -pedantic -Wall -W -O2 -g -ggdb   -I../deps/hiredis -I../deps/linenoise -I
echo PREV_FINAL_LDFLAGS=  -g -ggdb -rdynamic >> .make-settings
(cd ../deps && make hiredis linenoise lua jemalloc)
make[2]: 进入目录“/usr/local/src/redis3.0/deps”
(cd hiredis && make clean) > /dev/null || true
(cd linenoise && make clean) > /dev/null || true
(cd lua && make clean) > /dev/null || true
(cd jemalloc && [ -f Makefile ] && make distclean) > /dev/null || true
(rm -f .make-*)
(echo "" > .make-cflags)
(echo "" > .make-ldflags)
MAKE hiredis
cd hiredis && make static
make[3]: 进入目录“/usr/local/src/redis3.0/deps/hiredis”
cc -std=c99 -pedantic -c -O3 -fPIC  -Wall -W -Wstrict-prototypes -Wwrite-strings -g -ggdb  net.c
cc -std=c99 -pedantic -c -O3 -fPIC  -Wall -W -Wstrict-prototypes -Wwrite-strings -g -ggdb  hiredis.c
cc -std=c99 -pedantic -c -O3 -fPIC  -Wall -W -Wstrict-prototypes -Wwrite-strings -g -ggdb  sds.c
```

[root@localhost redis3.0]# make PREFIX=/usr/local/redis install

安装完成后，可以看到



在/usr/local/redis/bin 目录下会生成如下文件

-rwxr-xr-x. 1 root root 2075005 9 月 19 11:35 redis-benchmark

-rwxr-xr-x. 1 root root 25165 9 月 19 11:35 redis-check-aof

-rwxr-xr-x. 1 root root 56012 9 月 19 11:35 redis-check-dump

-rwxr-xr-x. 1 root root 2199167 9 月 19 11:35 redis-cli

lrwxrwxrwx. 1 root root 12 9 月 19 11:35 redis-sentinel -> redis-server

-rwxr-xr-x. 1 root root 4328271 9 月 19 11:35 redis-server

将 Redis 配置为服务

将上面的操作步骤，Redis 启动脚本为

/usr/local/src/redis3.0/utils/redis_init_script

将启动脚本复制到/etc/rc.d/init.d/目录下，并命名为 redis:

[root@localhost bin]# cp /usr/local/src/redis3.0/utils/redis_init_script /etc/rc.d/init.d/redis

编辑/etc/redis/init.d/redis 危机，使之成为注册服务（默认的配置如下图：）

```
#!/bin/sh
#
# Simple Redis init.d script conceived to work on Linux systems
# as it does use of the /proc filesystem.

REDISPORT=6379
EXEC=/usr/local/bin/redis-server
CLIEXEC=/usr/local/bin/redis-cli

PIDFILE=/var/run/redis_${REDISPORT}.pid
CONF="/etc/redis/${REDISPORT}.conf"

case "$1" in
    start)
        if [ -f $PIDFILE ]
        then
                echo "$PIDFILE exists, process is already running or crashed"
        else
                echo "Starting Redis server..."
                $EXEC $CONF
        fi
        ;;
    stop)
        if [ ! -f $PIDFILE ]
        then
                echo "$PIDFILE does not exist, process is not running"
        else
                PID=$(cat $PIDFILE)
                echo "Stopping ..."
                $CLIEXEC -p $REDISPORT shutdown
                while [ -x /proc/${PID} ]
                do
                    echo "Waiting for Redis to shutdown ..."
                    sleep 1
                done
                echo "Redis stopped"
        fi
        ;;
    *)
        echo "Please use start or stop as first argument"
        ;;
esac
```

修改为如下标红的

#!/bin/sh

#chkconfig:2345 80 90

Simple Redis init.d script conceived to work on Linux systems

as it does use of the /proc filesystem.

REDISPORT=6379

EXEC=/usr/local/redis/bin/redis-server

CLIEXEC=/usr/local/redis/bin/redis-cli

PIDFILE=/var/run/redis_${REDISPORT}.pid

CONF="/usr/local/redis/conf/${REDISPORT}.conf"

```
case "$1" in
      start)
            if [ -f $PIDFILE ]
            then
                    echo "$PIDFILE exists, process is already running or crashed"
            else
                    echo "Starting Redis server..."
                    $EXEC $CONF &
            fi
```

```
        ;;
    stop)
        if [ ! -f $PIDFILE ]
        then
                echo "$PIDFILE does not exist, process is not running"
        else
                PID=$(cat $PIDFILE)
                echo "Stopping ..."
                $CLIEXEC -p $REDISPORT shutdown
                while [ -x /proc/${PID} ]
                do
                    echo "Waiting for Redis to shutdown ..."
                    sleep 1
                done
                echo "Redis stopped"
        fi
        ;;
    *)
        echo "Please use start or stop as first argument"
        ;;
Esac
```

（1）在脚本第一行添加如下内容**#chkconfig:2345 80 90**

（2）REDISPORT 端口默认为 6379

（3）EXEC=/usr/local/bin/redis-server 修改为 **EXEC=/usr/local/redis/bin/redis-server**

（4）CLIEXEC=/usr/local/bin/redis-cli 修改为 **CLIEXEC=/usr/local/redis/bin/redis-cli**

（5）配置文件设置

创建 redis 配置文件目录

[root@localhost local]# mkdir /usr/local/redis/conf

复制 redis 配置文件/usr/local/src/redis3.0/redis.conf 到/usr/local/redis/conf 目录 并按照端口重命名为 6379.conf

[root@localhost local]# cp /usr/local/src/redis3.0/redis.conf /usr/local/redis/conf/6379.conf

以上操作完成之后，对 conf 属性修改

CONF="/usr/local/redis/conf/${REDISPORT}.conf"

（6）更改 redis 开启的命令，以后台运行$EXEC $CONF **&**

```
#chkconfig:2345 80 90
# Simple Redis init.d script conceived to work on Linux systems
# as it does use of the /proc filesystem.

REDISPORT=6379
EXEC=/usr/local/redis/bin/redis-server
CLIEXEC=/usr/local/redis/bin/redis-cli

PIDFILE=/var/run/redis_${REDISPORT}.pid
CONF="/usr/local/redis/conf/${REDISPORT}.conf"

case "$1" in
    start)
        if [ -f $PIDFILE ]
        then
                echo "$PIDFILE exists, process is already running or crashed"
        else
                echo "Starting Redis server..."
                $EXEC $CONF &
        fi
        ;;
    stop)
        if [ ! -f $PIDFILE ]
        then
                echo "$PIDFILE does not exist, process is not running"
        else
                PID=$(cat $PIDFILE)
                echo "Stopping ..."
                $CLIEXEC -p $REDISPORT shutdown
                while [ -x /proc/${PID} ]
                do
                    echo "Waiting for Redis to shutdown ..."
                    sleep 1
                done
                echo "Redis stopped"
        fi
        ;;
    *)
        echo "Please use start or stop as first argument"
        ;;
esac
~
~
~
~
~
~
"/etc/rc.d/init.d/redis" 42L, 1143C 已写入
[root@localhost bin]#
```

以上配置完成之后，将 Redis 注册为服务

[root@localhost bin]# chkconfig --add redis

修改 redis 配置文件设置

By default Redis does not run as a daemon. Use 'yes' if you need it.

Note that Redis will write a pid file in /var/run/redis.pid when daemonized.

daemonize yes

When running daemonized, Redis writes a pid file in /var/run/redis.pid by

default. You can specify a custom pid file location here.

pidfile /var/run/redis_6379.pid

启动 redis 服务

[root@localhost conf]# service redis start

Starting Redis server...

```
port 6379

# TCP listen() backlog.
#
"6379.conf" 938L, 41409C 已写入
[root@localhost conf]# service redis start
Starting Redis server...
[root@localhost conf]#
[root@localhost conf]#
[root@localhost conf]# ps -ef | grep redis
root     32721     1  0 12:44 ?        00:00:00 /usr/local/redis/bin/redis-server *:6379
root     32725 25436  0 12:44 pts/0    00:00:00 grep --color=auto redis
[root@localhost conf]#
```

将 Redis 添加到环境变量中

JAVA_HOME=/usr/local/jdk1.7.0_79

PATH=$JAVA_HOME/bin:$PATH:**/usr/local/redis/bin**

CLASSPATH=$JAVA_HOME/jre/lib/ext:$JAVA_HOME/lib/tools.jar

export PATH JAVA_HOME CLASSPATH

使得配置生效：

[root@localhost conf]# source /etc/profile

现在就可以直接使用 redis-cli 等 redis 命令了

[root@localhost conf]# redis-cli

127.0.0.1:6379> keys *

(empty list or set)

127.0.0.1:6379>

如想要关闭 redis 服务

[root@localhost conf]# service redis stop

1.4. Tomcat 安装配置

[root@centos7-200g-4 local]# tar -zxvf apache-tomcat-7.0.70.tar.gz

[root@centos7-200g-4 conf]# vim /usr/local/apache-tomcat-7.0.70/conf/server.xml

<Connector port="8080" protocol="HTTP/1.1"

URIEncoding="UTF-8"

minSpareThreads="400"

maxSpareThreads="700"

enableLookups="false"

disableUploadTimeout="true"

connectionTimeout="20000"

acceptCount="700"

maxThreads="700"

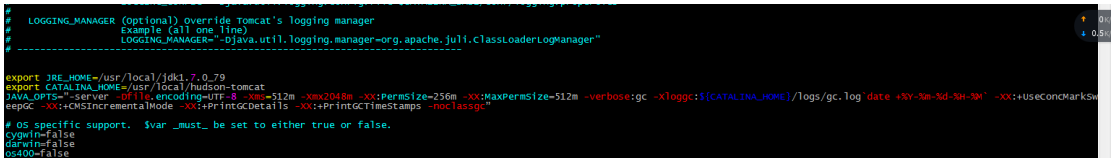
maxProcessors="700"

minProcessors="20"

useURIValidationHack="false"

redirectPort="8443"/>

[root@centos7-200g-4]# vim /usr/local/analysis-tomcat/bin/catalina.sh



export JRE_HOME=/usr/local/jdk1.7.0_79

export CATALINA_HOME=/usr/local/ analysis-tomcat

JAVA_OPTS=”-server -Dfile.encoding=UTF-8 -Xms=512m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=512m -verbose:gc -Xloggc:${CATALINA_HOME}/logs/gc.log`date

+%Y-%m-%d-%H-%M` -XX:+UseConcMarkSw
eepGC -XX:+CMSIncrementalMode -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -noclassgc”

1.5. 数据库安装配置

目前数据库使用平台整体提供的数据库 172.15.110.53，创建数据库 jeesite,将系统脚本导入到数据库中 jeesite.sql

2. 软件服务的启动

2.1. Redis 服务启动与关闭（开机自启动）

启动 redis 服务
[root@localhost conf]# service redis start
停止 redis 服务
[root@localhost conf]# service redis stop
查看是否运行
[root@centos7-200g-4 classes]# ps -ef | grep redis
root 1345 1 0 9 月 22 ? 00:05:22 /usr/local/redis/bin/redis-server *:6379
root 14006 13610 0 10:36 pts/0 00:00:00 grep --color=auto redis
命令行连接 redis 服务
[root@localhost conf]# redis-cli
查看端口占用情况
[root@centos7-200g-4 classes]# netstat -anp | grep 6379
tcp 0 0 0.0.0.0:6379 0.0.0.0:* LISTEN
1345/redis-server *
tcp 0 0 127.0.0.1:6379 127.0.0.1:40096 ESTABLISHED
1345/redis-server *
tcp 0 0 127.0.0.1:40086 127.0.0.1:6379 TIME_WAIT -
tcp6 0 0 :::6379 :::* LISTEN
1345/redis-server *
tcp6 0 0 127.0.0.1:40096 127.0.0.1:6379 ESTABLISHED
3430/java

2.2. Tomcat 服务启动与关闭

Tomcat 服务启动
[root@centos7-200g-4 classes]# /usr/local/apache-tomcat-7.0.70/bin/startup.sh

Tomcat 服务关闭
[root@centos7-200g-4 classes]# /usr/local/apache-tomcat-7.0.70/bin/shutdown.sh

查看 Tomcat 日志
[root@centos7-200g-4 classes]# tail -100f /usr/local/apache-tomcat-7.0.70/logs/catalina.out

2.3. Nginx 服务启动与关闭

Ngnix 服务启动
[root@centos7-200g-4 classes]# /usr/local/nginx/sbin/nginx
Ngnix 服务关闭
[root@centos7-200g-4 classes]# ps -ef | grep nginx
root 3493 1 0 9 月 22 ? 00:00:00 nginx: master process /usr/local/nginx/sbin/nginx
nobody 3494 3493 0 9 月 22 ? 00:00:25 nginx: worker process
nobody 3495 3493 0 9 月 22 ? 00:00:25 nginx: worker process
nobody 3496 3493 0 9 月 22 ? 00:00:22 nginx: worker process
nobody 3497 3493 0 9 月 22 ? 00:00:11 nginx: worker process
root 14042 13610 0 10:43 pts/0 00:00:00 grep --color=auto nginx
[root@centos7-200g-4 classes]# kill -9 3493
Ngnix 服务重新加载
[root@centos7-200g-4 classes]# /usr/local/nginx/sbin/nginx -s reload
Nginx 服务配置文件测试
[root@centos7-200g-4 classes]# /usr/local/nginx/sbin/nginx -t

3. 学情分析系统的部署

3.1. Tomcat 服务关闭

[root@centos7-200g-4 classes]# /usr/local/apache-tomcat-7.0.70/bin/shutdown.sh

3.2. 备份旧版程序 analysis 程序拷贝到备份目录中

[root@centos7-200g-4 war]# mkdir -p /usr/local/src/war/2016-09-26
[root@centos7-200g-4 2016-09-14]# cp -r /usr/local/apache-tomcat-7.0.70/webapps/analysis /usr/local/src/war/2016-09-26/

3.3. 删除旧版程序及 war 包

[root@centos7-200g-4 2016-09-14]# cd /usr/local/apache-tomcat-7.0.70/webapps/
[root@centos7-200g-4 webapps]# rm -rf analysis
[root@centos7-200g-4 webapps]# rm -rf analysis.war

3.4. 上传新版的 war 包

上传新版的 analysis.war 至 Tomcat webapp 目录中/usr/local/apache-tomcat-7.0.70/webapps

3.5. 启动 Tomcat 服务，查看程序运行日志

[root@centos7-200g-4 webapps]# /usr/local/apache-tomcat-7.0.70/bin/startup.sh
[root@centos7-200g-4 webapps]# tail -100f /usr/local/apache-tomcat-7.0.70/logs/catalina.out