**生产环境服务器性能优化指导手册**

**Guide for Production Environment Server Performance Optimization**

版本： 1.0.0

拟制: 蒋 赞

**修改记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **文件编号** | **版本号** | **拟制人/修改人** | **拟制/修改日期** | **更改理由** | **主要更改内容** |
|  | 1.0.0 | 蒋赞 | 2015-09-26 |  | 起草 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 注1：每次更改归档文档时必须填写该表 注2：一次归档时，“更改理由”和“主要更改内容”填无 | | | | | |

**目录**

[1. 系统的优化 4](#_Toc431198152)

[1.1 磁盘分区 4](#_Toc431198153)

[1.2 网络配置 4](#_Toc431198154)

[1.3 SSH服务配置 4](#_Toc431198155)

[1.4 系统配置 5](#_Toc431198156)

[1.4.1安装sudo及配置权限 5](#_Toc431198157)

[1.4.2删除无用的帐号与群组 5](#_Toc431198158)

[1.4.3修改登录欢迎语 6](#_Toc431198159)

[1.4.4创建基本的用户组 6](#_Toc431198160)

[1.4.5修改ulimit限制 7](#_Toc431198161)

[1.4.6修改主机名 7](#_Toc431198162)

[1.4.7禁用SELinux 7](#_Toc431198163)

[1.4.8禁用不常用的服务 8](#_Toc431198164)

[1.5 内核优化 8](#_Toc431198165)

[1.6 操作日志记录 9](#_Toc431198166)

[2. MySQL配置优化 11](#_Toc431198167)

[3. Tomcat配置优化 14](#_Toc431198168)

[3.1 属性文件 14](#_Toc431198169)

[3.2 配置文件 14](#_Toc431198170)

[3.3 应用启动 15](#_Toc431198171)

[3.4 应用停止 15](#_Toc431198172)

[3.5 应用调试 16](#_Toc431198173)

[4. PHP配置优化 17](#_Toc431198174)

[4.1 php.ini优化 17](#_Toc431198175)

[4.2 Opcache优化 17](#_Toc431198176)

[4.3 php-fpmX.conf优化 17](#_Toc431198177)

[4.4 集群服务 17](#_Toc431198178)

[4.5 压测数据 18](#_Toc431198179)

[5. Nginx配置优化 19](#_Toc431198180)

[5.1 主配置 19](#_Toc431198181)

[5.2 针对php-cgi的配置 22](#_Toc431198182)

[5.3 针对php-proxy的配置 24](#_Toc431198183)

[5.4 针对Tomcat的配置 26](#_Toc431198184)

# 系统的优化

服务器操作系统为CentOS 64位服务器版，版本一般选用6.5或7.0，建议采用7.0版本。

CentOS官方网址为：<http://www.centos.org/>

通常下载的CentOS版本如下：

<http://mirrors.aliyun.com/centos/7/isos/x86_64/CentOS-7.0-1406-x86_64-Minimal.iso> (推荐)

<http://mirrors.aliyun.com/centos/7/isos/x86_64/CentOS-7.0-1406-x86_64-Everything.iso>

1. **磁盘分区**

一般CentOS安装时，我们采用CentOS-7.0-1406-x86\_64-Minimal.iso 即可，我们可以将ISO利用UltraISO烧入U盘进行安装，也可以采用VMWare或VirtualBox进行虚拟化安装，在安装过程中我们采用默认方式安装即可，如英文、美式键盘等，需要特别注意的地方是磁盘的分区时，选择Standard Partition分区，分区大小及分配见下表：

|  |  |  |  |
| --- | --- | --- | --- |
| **分区名称** | **分区大小(M)** | **分区类型** | **备注** |
| / | 204800 | xfs | 根分区 |
| /boot | 512 | xfs | 启动分区，必须有，否则无法启动 |
| /swap | 与物理内存相当 | swap | 交换分区，当物理内存不够时，会启用交换分区 |
| /data | 所有剩余空间 | xfs | 独立一个分区做为数据盘，当系统损坏时，确保数据不丢失 |
| /bootbios | 1 |  |  |

对于CentOS6.5，一般采用ext4的分区文件类型，对于CentOS7我们推荐采用xfs分区文件类型。

1. **网络配置**

|  |
| --- |
| #vi /etc/sysconfig/network-scripts/ifcfg-eth0  IPV6INIT="no"  IPV6\_AUTOCONF="no"  IPADDR=192.168.128.222  NETMASK=255.255.255.0  GATEWAY=192.168.128.1  #NETWORK=192.168.128.0  #BROADCAST=192.168.128.255 |

1. **SSH服务配置**

|  |
| --- |
| #vi /etc/ssh/sshd\_config  IgnoreRhosts yes  UseDNS no  //然后重启sshd服务  //for centos6  #/etc/init.d/sshd restart  //for centos7  #systemctl restart sshd |

1. **系统配置**

### 安装sudo及配置权限

|  |
| --- |
| #yum install sudo  #vi /etc/pam.d/su  auth required pam\_wheel.so use\_uid group=wheel  #visudo  %wheel ALL=(ALL) NOPASSWD: ALL  %www ALL=/bin/vi, /usr/bin/vim, /usr/bin/tail, /bin/more, /bin/ls, /bin/rm, /bin/mv, /bin/cp, /bin/cat  %mis ALL=/bin/vi, /usr/bin/vim, /usr/bin/fab, /usr/bin/tail, /bin/cat, /bin/more, /bin/rm, /bin/mv, /bin/ls, /bin/ping, /sbin/service, /bin/kill, /usr/bin/kill, /usr/bin/killall, /bin/chown, /bin/chmod, /bin/chgrp, /usr/bin/yum, /bin/mkdir, /bin/touch, /bin/cp, /bin/netstat, /usr/bin/python, /sbin/iptables |

上面设置了两个组：

* 一个wheel组(管理员)，具备sudo权限，可以提权到root权限
* 一个www组(网站维护组)，只能拥有固定的命令

通常还会设置一个dev组(开发组)，让开发人员只能在自己的home目录进行开发操作

### 删除无用的帐号与群组

为了安全起见，这里把不需要使用的帐号与群组全部删除

|  |
| --- |
| userdel adm  userdel lp  userdel sync  userdel shutdown  userdel halt  userdel news  userdel uucp  userdel operator  userdel games  userdel gopher  userdel ftp  groupdel adm  groupdel lp  groupdel news  groupdel uucp  groupdel games  groupdel dip  groupdel pppusers  groupdel audio  groupdel video |

### 修改登录欢迎语

修改欢迎语，提示登录用户不要乱操作服务器

|  |
| --- |
| #vi /etc/motd    Welcome to Nubia Production Compute Service!  -----------------------------------------------------------  Please read the following words carefully.  1. Don't reboot this machine. (It can't wake up automicly)  2. Don't delete any others file.  3. Please be sure to use your own account.  4. Only members of the wheel group can run sudo.  ----------------------------------------------------------- |

### 创建基本的用户组

创建www组及用户，及常用的用户名

|  |
| --- |
| groupadd www  useradd -g www www -s /sbin/nologin -d /var/www  mkdir -p /data/home  useradd -g wheel nubia -s /bin/bash -d /data/home/nubia  usermod -G wheel,root nubia  useradd -g wheel jiangz -s /bin/bash -d /data/home/jiangz |

上面的命令大致解说如下：

1. 创建一个www用户及www组，同时确保www用户无登录权限，并将www用户锁定在/var/www目录；
2. 创建一个nubia帐号，并加入wheel与root组，默认用户目录为/data/home/nubia；
3. 创建一个jiangz帐号，并加入wheel组，默认用户目录为/data/home/jiangz；

### 修改ulimit限制

|  |
| --- |
| //查看系统最大的文件打开数  #cat /proc/sys/fs/file-max  #vi /etc/security/limits.conf  \* soft nofile 204800  \* hard nofile 204800  \* soft nproc 65535  \* hard nproc 65535 |

或者使用如下命令亦可

|  |
| --- |
| sudo sh -c 'echo "\* soft nofile 204800" >> /etc/security/limits.conf'  sudo sh -c 'echo "\* hard nofile 204800" >> /etc/security/limits.conf'  //for centos6  sudo sed -i 's/4096/65535/' /etc/security/limits.d/20-nproc.conf  //for centos7  sudo sed -i 's/1024/65535/' /etc/security/limits.d/90-nproc.conf |

上面的命令设置系统单进程可以打开的文件数为204800个，系统可以开启65535个进程。

### 修改主机名

1. **CentOS6下修改主机名**

|  |
| --- |
| # vi /etc/hosts  //在127.0.0.1与::1后面增加主机名  # vi /etc/sysconfig/network  //修改主机名(HOSTNAME=)  //输入以下命令：  hostname 新主机名 |

1. **CentOS7下修改主机名**

CentOS 7 里面修改hostname的方式有所改变，修改/etc/hosts和/etc/sysconfig/network两个文件已经不能生效。使用的新命令：

|  |
| --- |
| $ hostnamectl set-hostname <hostname> |

主机名修改之后，然后用ssh客户端重新登录，就会显示新的主机名。

### 禁用SELinux

|  |
| --- |
| #vi /etc/selinux/config  SELINUX=disabled  #setenforce 0 //临时生效  #getenforce //查看selinux状态 |

### 禁用不常用的服务

|  |
| --- |
| 停止postfix服务  systemctl disable postfix  systemctl stop postfix  关闭防火墙  service firewalld stop  禁用防火墙  systemctl disable firewalld |

1. **内核优化**

编辑/etc/sysctl.conf内容如下：

|  |
| --- |
| net.ipv4.ip\_forward = 0  net.ipv4.conf.all.rp\_filter = 0  net.ipv4.conf.default.rp\_filter = 0  net.ipv4.conf.default.accept\_source\_route = 0  kernel.sysrq = 0  kernel.core\_uses\_pid = 1  kernel.msgmnb = 65536  kernel.msgmax = 65536  kernel.shmmax = 68719476736  kernel.shmall = 4294967296  net.ipv4.neigh.default.gc\_stale\_time = 120  net.ipv4.conf.default.arp\_announce = 2  net.ipv4.conf.all.arp\_announce = 2  net.ipv4.conf.lo.arp\_announce = 2  net.core.somaxconn = 81920  net.core.netdev\_max\_backlog = 81920  net.ipv4.ip\_local\_port\_range = 1024 65000  net.ipv4.tcp\_syncookies = 1  net.ipv4.tcp\_keepalive\_time = 300  net.ipv4.tcp\_fin\_timeout = 30  net.ipv4.tcp\_timestamps = 1  net.ipv4.tcp\_tw\_reuse = 1  net.ipv4.tcp\_tw\_recycle = 1  #net.ipv4.tcp\_tw\_timeout = 60  net.ipv4.tcp\_max\_tw\_buckets = 20000  net.ipv4.tcp\_max\_syn\_backlog = 16384  net.ipv4.tcp\_synack\_retries = 3  net.ipv4.tcp\_max\_orphans = 131072  net.ipv4.tcp\_no\_metrics\_save = 1  net.ipv4.tcp\_sack = 1  net.ipv4.tcp\_window\_scaling = 1  #vm.overcommit\_memory = 1  #net.ipv4.tcp\_orphan\_retries = 3 |

使之生效#sysctl -p

#cat /proc/sys/net/ipv4/tcp\_tw\_timeout

#cat /proc/sys/net/ipv4/tcp\_timestamps

1. **操作日志记录**

|  |
| --- |
| #!/bin/sh  #history  #export HISTTIMEFORMAT="[%Y.%m.%d %H:%M:%S]"  export HISTTIMEFORMAT="`whoami` %F %T "  USER\_IP=`who -u am i 2>/dev/null| awk '{print $NF}'|sed -e 's/[()]//g'`  HISTDIR=/var/log/history  if [ -z $USER\_IP ]  then  USER\_IP=`hostname`  fi  if [ ! -d $HISTDIR ]  then  sudo mkdir -p $HISTDIR  sudo chmod 777 $HISTDIR  fi  if [ ! -d $HISTDIR/${LOGNAME} ]  then  mkdir -p $HISTDIR/${LOGNAME}  chmod 300 $HISTDIR/${LOGNAME}  fi  export HISTSIZE=4096  DT=`date +%Y%m%d\_%H%M%S`  #export HISTFILE="$HISTDIR/${LOGNAME}/${USER\_IP}\_history.$DT"  export HISTFILE="$HISTDIR/${LOGNAME}/${USER\_IP}\_history.log"  chmod 600 $HISTDIR/${LOGNAME}/\*\_history\* 2>/dev/null |

# MySQL配置优化

这里的MySQL配置仅适应生产环境，以8核16G为例而优化的。MySQL的优化主要通过MySQL的配置文件来设置。

|  |
| --- |
| [mysqld]  user = mysql  port = 3306  socket = /etc/mysql.sock  basedir = /usr/local/mysql  datadir = /data/database/mysql  slow\_query\_log = 1  slow\_query\_log\_file = /data/database/mysql/mysql\_query\_slow.log  long\_query\_time = 1  server-id = 1  log-bin = master-bin  log-bin-index = master-bin.index  #binlog-do-db=blog  #binlog-do-db=nubia\_bbs  binlog-ignore-db=mysql,test,information\_schema  #sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES  #sql\_mode = NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION  sql\_mode = ANSI  back\_log = 511  sync\_binlog = 1  skip-name-resolve = 1  lower\_case\_table\_names = 1  binlog\_format = mixed  binlog\_cache\_size = 4M  max\_connections = 10000  max\_connect\_errors = 10000  thread\_cache\_size = 256 #1g:16 2g:32 4g:64 16g:256  thread\_concurrency = 8 #2\*cpu cores  key\_buffer\_size = 2048M #1/8 RAM  innodb\_data\_file\_path = ibdata1:1024M:autoexten  innodb\_buffer\_pool\_size = 8G #1/2  innodb\_additional\_mem\_pool\_size = 16M  innodb\_log\_file\_size = 256M  innodb\_log\_buffer\_size = 16M  innodb\_log\_files\_in\_group = 3  innodb\_lock\_wait\_timeout = 60  innodb\_file\_per\_table = 1  innodb\_flush\_log\_at\_trx\_commit = 2  innodb\_file\_io\_threads = 8  innodb\_max\_dirty\_pages\_pct = 90  default-storage-engine = MyISAM  table\_cache = 512  external-locking = FALSE  max\_allowed\_packet = 32M  sort\_buffer\_size = 2M  join\_buffer\_size = 2M  query\_cache\_size = 64M  query\_cache\_limit = 4M  query\_cache\_min\_res\_unit = 2k  thread\_stack = 192K  transaction\_isolation = REPEATABLE-READ  tmp\_table\_size = 256M  max\_heap\_table\_size = 256M  max\_binlog\_cache\_size = 8M  max\_binlog\_size = 512M  expire\_logs\_days = 7  read\_buffer\_size = 1M  read\_rnd\_buffer\_size = 16M  bulk\_insert\_buffer\_size = 64M  myisam\_sort\_buffer\_size = 128M  myisam\_max\_sort\_file\_size = 10G  myisam\_max\_extra\_sort\_file\_size = 10G  myisam\_repair\_threads = 1  myisam\_recover = 1  [mysql]  no-auto-rehash  [mysqldump]  quick  max\_allowed\_packet = 256M  [mysqld\_safe]  open-files-limit = 10240 |

特别注意一下参数innodb\_flush\_log\_at\_trx\_commit对性能的影响比较大，说明如下：

innodb\_flush\_log\_at\_trx\_commit 参数解释：

0（延迟写）： log\_buff  --每隔1秒--> log\_file  —实时—> disk

1（实时写，实时刷）： log\_buff  —实时—>  log\_file  —实时—> disk

2（实时写，延迟刷）： log\_buff  —实时—> log\_file --每隔1秒--> disk

# Tomcat配置优化

生产环境的Tomcat请采用如下两个：

<http://10.206.19.222/down/tomcat/tomcat-7.0.59.compiled.apr-1.1.32_for_centos7.zip>

[http://10.206.19.222/down/tomcat/tomcat-8.5.3.compiled.tcnative-1.2.7\_for\_centos7.7z](http://10.206.19.222/down/tomcat/tomcat-8.0.20.compiled.apr-1.1.32_for_centos7.zip)

这两个Tomcat是经过定制与特定优化的，对于性能的提升非常显著，同时单个应用采用独立JVM，应用之间互相隔离。

1. **属性文件**

|  |
| --- |
| handlers = 1catalina.org.apache.juli.FileHandler, 2localhost.org.apache.juli.FileHandler, java.util.logging.ConsoleHandler  .handlers = 1catalina.org.apache.juli.FileHandler, java.util.logging.ConsoleHandler  1catalina.org.apache.juli.FileHandler.level = FINE  1catalina.org.apache.juli.FileHandler.directory = /data/logs/tomcat-8.5.3  1catalina.org.apache.juli.FileHandler.prefix = demo.catalina.  2localhost.org.apache.juli.FileHandler.level = FINE  2localhost.org.apache.juli.FileHandler.directory = /data/logs/tomcat-8.5.3  2localhost.org.apache.juli.FileHandler.prefix = demo.localhost.  java.util.logging.ConsoleHandler.level = FINE  java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter  org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO  org.apache.catalina.core.ContainerBase.[Catalina].[localhost].handlers = 2localhost.org.apache.juli.FileHandler |

1. **配置文件**

//TODO

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <Server port="9001" shutdown="SHUTDOWN">  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="off" />  <!--Listener className="org.apache.catalina.core.JasperListener" /-->  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />  <Service name="Catalina">  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-" maxThreads="1024" minSpareThreads="64"/>  <Connector executor="tomcatThreadPool" port="8001" protocol="org.apache.coyote.http11.Http11AprProtocol" maxHttpHeaderSize="8192" useBodyEncodingForURI="true" acceptCount="1024" keepAliveTimeout="60000" maxKeepAliveRequests="256" enableLookups="false" connectionTimeout="20000" disableUploadTimeout="true" URIEncoding="UTF-8" address="localhost"/>    <Engine name="Catalina" defaultHost="localhost">  <Host name="localhost" appBase="webapps" unpackWARs="false" autoDeploy="false">  <Context path="/" docBase="/data/wwwroot/demo" reloadable="false" crossContext="false" privileged="true" workDir="work/demo\_8001"/>  <!--Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="10\..\*" /-->  <!--Valve className="org.apache.catalina.valves.AccessLogValve" directory="/data/logs/tomcat-8.8.3/"  prefix="open.nubia.cn\_8001\_access\_log" suffix=".txt"  pattern="%h %l %u %t &quot;%r&quot; %s %b" /-->  </Host>  </Engine>  </Service>  </Server> |

1. **应用启动**

|  |
| --- |
| sudo -E -u www ./demo\_8001.sh start |

1. **应用停止**

|  |
| --- |
| sudo -E -u www ./demo\_8001.sh stop |

1. **应用调试**

|  |
| --- |
| sudo -E -u www ./demo\_8001.sh run |

# PHP配置优化

1. **php.ini优化**

第三方扩展只保留

redis.so

其它mongo.so,memcached.so,gd.so,opcache.so全注释掉

session存储在redis中,减少IO读写

session.save\_handler = redis

session.save\_path = tcp://127.0.0.1:6379

date.timezone = "Asia/Shanghai"

extension\_dir = /data/program/php-5.5.15/lib/php/extensions/no-debug-non-zts-20121212/

extension=gd.so

extension=redis.so

1. **Opcache优化**

注意：保留独立的opcache模块配置

|  |
| --- |
| [opcache]  zend\_extension=/data/program/php-5.5.15/lib/php/extensions/no-debug-non-zts-20121212/opcache.so  opcache.memory\_consumption=128  opcache.optimization\_level=1  opcache.interned\_strings\_buffer=8  opcache.max\_accelerated\_files=4096  opcache.revalidate\_freq=60  opcache.fast\_shutdown=1  opcache.enable=1  opcache.enable\_cli=1 |

1. **php-fpmX.conf优化**

将listen = /tmp/php55-cgi1.socket 替换为 listen = /dev/shm/php55-cgi1.socket

1. **集群服务**

|  |
| --- |
| upstream phpbackend {  server unix:/tmp/php53-cgi1.socket weight=100 max\_fails=3 fail\_timeout=30s;  server unix:/tmp/php53-cgi2.socket weight=100 max\_fails=3 fail\_timeout=30s;  server unix:/tmp/php53-cgi3.socket weight=100 max\_fails=3 fail\_timeout=30s;  server unix:/tmp/php53-cgi4.socket weight=100 max\_fails=3 fail\_timeout=30s;  }    upstream phpbackend55 {  server unix:/dev/shm/php55-cgi1.socket weight=100 max\_fails=3 fail\_timeout=30s;  server unix:/dev/shm/php55-cgi2.socket weight=100 max\_fails=3 fail\_timeout=30s;  server unix:/dev/shm/php55-cgi3.socket weight=100 max\_fails=3 fail\_timeout=30s;  server unix:/dev/shm/php55-cgi4.socket weight=100 max\_fails=3 fail\_timeout=30s;  } |

1. **压测数据**

单压nginx静态页面 1W/s (采用ab -k)

单压nginx+ 4个php主进程 1060/s (采用ab -k，listen=/tmp/php55-cgi1.socket)

单压nginx+ 1个php主进程 1038/s，(采用ab -k，listen=/dev/shm/php55-cgi1.socket)

单压nginx+ 1个php主进程 1369/s，(采用ab，listen=127.0.0.1:9001)

单压nginx+ 1个php主进程 ，(采用ab -k，listen=/dev/shm/php55-cgi1.socket)

单压nginx+ 4个php主进程 1600/s，(采用jmeter 600/50，listen=/dev/shm/php55-cgi1.socket, session=/dev/shm/session)

单压nginx+ 4个php主进程 1200/s，(采用jmeter 600/50，listen=/dev/shm/php55-cgi1.socket, session=tcp.redis)

# Nginx配置优化

这里以4核8G服务器为例，下面的各配置仅做生产环境配置的参考，不特指具体的项目。

1. **主配置**

nginx.conf通用配置

|  |
| --- |
| user www www;  worker\_processes 4;  worker\_cpu\_affinity 0001 0010 0100 1000;  error\_log /data/logs/nginx-1.7.9/error.log error;  #error\_log logs/error.log notice;  #error\_log logs/error.log info;  pid /tmp/nginx.pid;  worker\_rlimit\_nofile 40960;  events {  use epoll;  worker\_connections 10240;  multi\_accept on;  accept\_mutex off;  }  http {  include mime.types;  default\_type application/octet-stream;    log\_format access '$http\_x\_forwarded\_for $remote\_addr [$time\_local] "$request" '  '$status $body\_bytes\_sent "$http\_referer" '  '"$http\_user\_agent"';    sendfile on;    tcp\_nodelay on;  tcp\_nopush on;    server\_tokens off; #关闭版本显示  fastcgi\_connect\_timeout 300;  fastcgi\_send\_timeout 300;  fastcgi\_read\_timeout 300;  fastcgi\_buffer\_size 256k;  fastcgi\_buffers 16 256k;  fastcgi\_busy\_buffers\_size 512k;  fastcgi\_temp\_file\_write\_size 512k;  #设定请求缓冲  server\_names\_hash\_bucket\_size 128;    client\_body\_buffer\_size 128K;  client\_header\_buffer\_size 16k;  large\_client\_header\_buffers 4 64k;  client\_max\_body\_size 8m;    client\_body\_timeout 30;  client\_header\_timeout 10;  keepalive\_timeout 65;  send\_timeout 30;    proxy\_connect\_timeout 90;  proxy\_send\_timeout 90;  proxy\_read\_timeout 90;  proxy\_buffer\_size 128k;  proxy\_buffers 4 256k;  proxy\_busy\_buffers\_size 256k;  proxy\_temp\_file\_write\_size 64m;  proxy\_ignore\_client\_abort on;    open\_file\_cache max=204800 inactive=20s;  open\_file\_cache\_min\_uses 1;  open\_file\_cache\_valid 30s;    #开启gzip模块  gzip on;  gzip\_min\_length 1k;  gzip\_http\_version 1.1;  gzip\_buffers 8 32k;  gzip\_comp\_level 2; #0-9 默认值为1，值越大压缩率越高，消耗的cpu资源越多，传输量减小  gzip\_types text/plain text/css application/json application/x-javascript text/xml application/xml application/xml+rss text/javascript;  gzip\_vary on;    #gzip\_static on;  gzip\_proxied any;  gzip\_disable "MSIE [1-6]\.";    output\_buffers 1 32k;  postpone\_output 1460;    #删除掉kEDH算法  ssl\_ciphers ALL:!kEDH!ADH:RC4+RSA:+HIGH:+EXP;    server {  listen 80 default\_server;  access\_log off;  server\_name \_; #default  return 444;  }      upstream jee\_backend {  hash $http\_x\_forwarded\_for;  server localhost:8001 weight=100 max\_fails=3 fail\_timeout=30s;  server localhost:8002 weight=100 max\_fails=3 fail\_timeout=30s;  server localhost:8003 weight=100 max\_fails=3 fail\_timeout=30s;  server localhost:8004 weight=100 max\_fails=3 fail\_timeout=30s;  #ip\_hash;  keepalive 1024;  }    upstream php\_backend {  #hash $http\_x\_forwarded\_for;  server unix:/dev/shm/php-5.7-dev\_fpm-dev1.socket weight=100 max\_fails=3 fail\_timeout=30s;  server unix:/dev/shm/php-5.7-dev\_fpm-dev2.socket weight=100 max\_fails=3 fail\_timeout=30s;  server unix:/dev/shm/php-5.7-dev\_fpm-dev3.socket weight=100 max\_fails=3 fail\_timeout=30s;  server unix:/dev/shm/php-5.7-dev\_fpm-dev4.socket weight=100 max\_fails=3 fail\_timeout=30s;  #ip\_hash;  keepalive 80;  }      # server {  # listen 6428;  # access\_log off;  #  # location / {  # status 200;  # header Content-Type = text/html;  # body ~ "Welcome to nginx!";  # }  # }  include site/\*.conf;  } |

1. **针对php-cgi的配置**

|  |
| --- |
| server {  listen 80;  server\_name nubia.com www.nubia.com;  access\_log off;  root /nowhere;  rewrite ^ https://nubia.com$request\_uri permanent;  }  server {  listen 443;  server\_name nubia.com;  index index.php;  root /data/wwwroot/virtual\_server;  # include rewrite\_nubia.conf;    rewrite ^(.\*)/([a-z0-9\_]\*)\.html$ $1/?act=$2&%1;  rewrite ^(.\*)/([a-z0-9\_]\*)/([a-z0-9\_\-]+)\.html$ $1/?act=$2!$3&%1;    #access\_log /data/logs/nginx-1.7.9/virtual\_server\_access.log;  access\_log off;    location ~ ^/status/ {  stub\_status on;  }    ssl on;  ssl\_certificate ssl\_key/sso.nubia.com.crt;  #openssl rsa -in sso.nubia.com.key -out sso.nubia.com.unsecure.key  ssl\_certificate\_key ssl\_key/sso.nubia.com.unsecure.key;  ssl\_client\_certificate ssl\_key/ca.pem;    #ssl\_protocols SSLv3 TLSv1;  ##ssl\_ciphers AES:HIGH:!ADH:!MD5;  #ssl\_ciphers ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP;  #ssl\_prefer\_server\_ciphers on;  ##self\_signed\_cert true;    ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;  ssl\_ciphers ECDHE-RSA-AES256-SHA384:AES256-SHA256:RC4:HIGH:!MD5:!aNULL:!eNULL:!NULL:!DH:!EDH:!AESGCM;  #ssl\_ciphers ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:!aNULL:!MD5:!EXP;  #ssl\_ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AES:!ADH:!AECDH:!MD5:!DSS:!3DES;  ssl\_prefer\_server\_ciphers on;  ssl\_session\_cache shared:SSL:10m;  ssl\_session\_timeout 10m;    access\_log off;  error\_log off;    location ~ \.php$ {  fastcgi\_pass php\_backend;  fastcgi\_keep\_conn on;  #fastcgi\_pass unix:/tmp/php-cgi.socket;  fastcgi\_param SCRIPT\_FILENAME /data/wwwroot/virtual\_server$fastcgi\_script\_name;  chunked\_transfer\_encoding off;  include fastcgi\_params;  }  location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$ {  expires 30d;  }  location ~ .\*\.(js|css)?$ {  expires 1h;  }    error\_page 500 502 503 504 /50x.html;  location = /50x.html {  root html;  }    error\_page 404 /404.html;  location = /404.html {  root html;  }  } |

1. **针对php-proxy的配置**

|  |
| --- |
| # GITLAB  # Maintainer: @yin8086  # App Version: 4.1  # Modified from nginx http version  # Modified from http://blog.phusion.nl/2012/04/21/tutorial-setting-up-gitlab-on-debian-6/  # You need from run openssl to generate the ssl certificate.  # $ sudo openssl req -new -x509 -nodes -days 3560 -out gitlab.crt -keyout gitlab.key  # $ sudo chmod o-r gitlab.key    upstream gitlab {  server unix:/home/git/gitlab/tmp/sockets/gitlab.socket;  }  # This is a normal HTTP host which redirects all traffic to the HTTPS host.  server {  listen 80;  server\_name git.nubia.org;  root /nowhere;  rewrite ^ https://git.nubia.org$request\_uri permanent;  }  server {  listen 443;  server\_name git.nubia.org;  root /home/git/gitlab/public;    ssl on;  ssl\_certificate ssl\_key/git.nubia.org.crt;  #openssl rsa -in git.nubia.org.key -out git.nubia.org.unsecure.key  ssl\_certificate\_key ssl\_key/git.nubia.org.unsecure.key;  ssl\_client\_certificate ssl\_key/ca.pem;  #ssl\_protocols SSLv3 TLSv1 TLSv2;  ssl\_protocols SSLv3 TLSv1;  #ssl\_ciphers AES:HIGH:!ADH:!MD5;  ssl\_ciphers ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP;  ssl\_prefer\_server\_ciphers on;  #self\_signed\_cert true;    # individual nginx logs for this gitlab vhost  access\_log /data/logs/nginx-1.7.4/gitlab\_access.log;  error\_log /data/logs/nginx-1.7.4/gitlab\_error.log;    location / {  # serve static files from defined root folder;.  # @gitlab is a named location for the upstream fallback, see below  try\_files $uri $uri/index.html $uri.html @gitlab;  }    # if a file, which is not found in the root folder is requested,  # then the proxy pass the request to the upsteam (gitlab unicorn)  location @gitlab {  proxy\_read\_timeout 300; # https://github.com/gitlabhq/gitlabhq/issues/694  proxy\_connect\_timeout 300; # https://github.com/gitlabhq/gitlabhq/issues/694  proxy\_http\_version 1.1;  proxy\_set\_header Connection "";  proxy\_redirect off;    proxy\_set\_header X-Forwarded-Proto https;  proxy\_set\_header X-Forwarded-Ssl on;  proxy\_set\_header Host $http\_host;  proxy\_set\_header X-Real-IP $remote\_addr;    proxy\_pass http://gitlab;  }  } |

1. **针对Tomcat的配置**

|  |
| --- |
| server {  # listen 80;  server\_name open.nubia.cn;  # access\_log /data/logs/nginx-1.7.4/open.nubia.cn.log access;  access\_log off;  #rewrite ^(.\*)\;JSESSIONID=(.\*)$ $1 break;  location ~ ^/status/ {  stub\_status on;  access\_log off;  allow 10.161.151.167;  allow 218.104.236.138;  deny all;  }    # include site/blacklist.txt;    # if ($host = 'nubia.cn' ) {  # rewrite ^/(.\*)$ http://www.nubia.cn/$1 permanent;  # }  #if ($http\_user\_agent ~ "Chrome/31.0.1650.63 Safari/537.36") {  # return 444;  # }  # if ($http\_user\_agent ~ "Gecko/20120216 Firefox/3.6.27") { return 444; }  #if ($http\_user\_agent ~\* "qihoobot|Baiduspider|Googlebot|Googlebot-Mobile|Googlebot-Image|Mediapartners-Google|Adsbot-Google|Feedfetcher-Google|Yahoo! Slurp|Yahoo! Slurp China|YoudaoBot|Sosospider|Sogou spider|Sogou web spider|MSNBot|ia\_archiver|Tomato Bot") {  # return 444;  #}  # if ($http\_user\_agent ~ ApacheBench|WebBench|Java/|http\_load|wget) {  # return 444;  # }  if ( $http\_user\_agent ~ "^((.\*MIDP.\*)|(.\*WAP.\*)|(.\*UP.Browser.\*)|(.\*Smartphone.\*)|(.\*Obigo.\*)|(.\*Mobile.\*)|(.\*AU.Browser.\*)|(.\*wxd.Mms.\*)|(.\*WxdB.Browser.\*)|(.\*CLDC.\*)|(.\*UP.Link.\*)|(.\*KM.Browser.\*)|(.\*UC.\*)|(.\*SEMC\-Browser.\*)|(.\*Mini.\*)|(.\*Symbian.\*)|(.\*Palm.\*)|(.\*Nokia.\*)|(.\*Panasonic.\*)|(.\*MOT\-.\*)|(.\*SonyEricsson.\*)|(.\*NEC\-.\*)|(.\*Alcatel.\*)|(.\*Ericsson.\*)|(.\*BENQ.\*)|(.\*BenQ.\*)|(.\*Amoisonic.\*)|(.\*Amoi\-.\*)|(.\*Capitel.\*)|(.\*PHILIPS.\*)|(.\*SAMSUNG.\*)|(.\*Lenovo.\*)|(.\*Mitsu.\*)|(.\*Motorola.\*)|(.\*SHARP.\*)|(.\*WAPPER.\*)|(.\*LG\-.\*)|(.\*LG/.\*)|(.\*EG900.\*)|(.\*CECT.\*)|(.\*Compal.\*)|(.\*kejian.\*)|(.\*Bird.\*)|(.\*BIRD.\*)|(.\*G900/V1.0.\*)|(.\*Arima.\*)|(.\*CTL.\*)|(.\*TDG.\*)|(.\*Daxian.\*)|(.\*DAXIAN.\*)|(.\*DBTEL.\*)|(.\*Eastcom.\*)|(.\*EASTCOM.\*)|(.\*PANTECH.\*)|(.\*Dopod.\*)|(.\*Haier.\*)|(.\*HAIER.\*)|(.\*KONKA.\*)|(.\*KEJIAN.\*)|(.\*LENOVO.\*)|(.\*Soutec.\*)|(.\*SOUTEC.\*)|(.\*SAGEM.\*)|(.\*SEC\-.\*)|(.\*SED\-.\*)|(.\*EMOL\-.\*)|(.\*INNO55.\*)|(.\*ZTE.\*)|(.\*iPhone.\*)|(.\*Android.\*)|(.\*Windows CE.\*)|(Wget.\*)|(Java.\*)|(curl.\*)|(Opera.\*))$") { rewrite "^/$" /mobile\_index.html last;}  location / {  index index.html openbuy.html index.jsp;  root /data/wwwroot/nubia\_open;  expires 30d;  }  location /api/uc {  rewrite ^/api/uc$ /api/uc.action last;  }      location ~ \.(jsp|action|do|zte|json|info)$ {  #location / {  proxy\_pass http://jee\_backend;  proxy\_http\_version 1.1;  proxy\_set\_header Connection "";  proxy\_redirect off;  #proxy\_set\_header Accept-Encoding 'gzip';  proxy\_set\_header Host $http\_host;  proxy\_set\_header X-Real-IP $proxy\_add\_x\_forwarded\_for;  #$http\_x\_forwarded\_for;  #$remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header Cookie $http\_cookie;  chunked\_transfer\_encoding off;  }    location ~ ^/(WEB-INF)/ {  deny all;  }    error\_page 403 404 500 502 503 504 /error.html;  location = /error.html {  root /data/wwwroot/nubia\_open;  }  }    # for SLB  server {  listen 6428;  access\_log off;    location / {  index index.html;  root /data/wwwroot/slb;  expires 30d;  }  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  root html;  }  }  } |