**xxx项目**

**部署文档**

**xxx股份有限公司**

**二〇一九年一月**

**修订控制页**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **版本** | **修订人** | **修订日期** | **修订摘要** |
| 1 | V1.0.0 | xxx | 2019/1/1 | 1.初稿 |

**修订说明：为了方便查看对新版本修订的内容，每次以红色标示修订的内容。**

目 录

1 引言 1

1.1 目的 1

1.2 目标范围 1

1.3 读者对象 1

2 部署说明 2

2.1 系统环境要求 2

2.2 软件列表 2

2.3 部署流程 3

2.4 部署架构图 3

3 系统部署 5

3.1 数据库服务器部署 5

3.1.1 安装Oracle 5

3.1.2 安装Redis 5

3.2 数据库生成或更新 7

3.2.1 数据库生成 7

3.2.2 数据库更新 7

3.3 存储器挂载 7

3.4 应用服务器部署 8

3.4.1 备份现有服务 8

3.4.2 第三方服务放行列表 8

3.4.3 应用服务器集群时钟同步 8

3.4.4 安装JDK 10

3.4.5 安装Tomcat并设置自启动 10

3.4.6 API接口服务 10

3.4.7 微信对外号 12

3.4.8 下载页部署 12

3.5 负载均衡服务器部署 12

3.5.1 安装Nginx 12

3.5.2 配置Nginx 16

3.6 客户端软件部署 20

3.6.1 Android客户端 20

3.6.2 iOS客户端 20

4 维护说明 22

4.1 日常维护注意事项 22

4.2 常见故障处理 22

4.2.1 Redis 错误 22

4.2.2 Nginx错误 22

# 引言

## 目的

为更好的指导部署与测试《xxx》系统，包括 API接口、后台管理系统、web客户端、Android、iOS，及依赖中间件如 JDK、Tomcat、Redis和Nginx 等通用依赖服务，特编写编写《xxx部署文档》，以便测试、运维人员更更方便的部署系统。

## 目标范围

本文档仅供内部使用，禁止外传，帮助研发人员，运维人员对系统长期稳定的运行提供技术文档参考。

## 读者对象

1. 项目经理
2. 开发人员
3. 测试人员
4. 运维人员
5. 相关领导

# 部署说明

本部分主要说明项目部署时需要准备的系统环境要求、软件列表、安装流程、部署架构图。

## 系统环境要求

表1 系统环境要求

|  |  |  |  |
| --- | --- | --- | --- |
| **系统/中间件** | **系统要求** | **硬件要求** | **备注** |
| **API接口服务、对外服务系统、后台管理系统、微信对外号** | Centos7.3 64 | 部署3台服务器，每台部署4个服务 | 依赖JDK1.8，Tomcat8.0 |
| **Oracle 11g、Redis** | Centos7.3 64 | 部署1台服务器 |  |
| **Nginx、下载页** | Centos7.3 64 | 部署1台服务器 |  |

## 软件列表

表2 软件列表

|  |  |  |  |
| --- | --- | --- | --- |
| **所属** | **软件** | **版本/命名方式** | **备注** |
| **第三方** | Oracle数据库 | 11g | 表空间、用户名及密码，全部设置为xxx，实例名为orcl。注：Oracle数据库部署信息在API接口服务、对外服务系统、后台管理系统。 |
| Redis | 5.0.x | 缓存数据库；可使用最新稳定版本。 |
| JDK | 1.8 | 产品运行依赖环境Java虚拟机 |
| Tomcat | 8.x | web系统运行容器。注：API接口服务、对外服务系统、后台管理系统需要用到。 |
| Nginx | 1.14.2（20190418） | 高性能的[HTTP](https://baike.baidu.com/item/HTTP%22%20%5Ct%20%22_blank)和[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86/7793488%22%20%5Ct%20%22_blank)服务；可使用最新稳定版本。 |
| **项目软件** | API接口服务 | api-vx.x.x.19xxxx.war |   |
| 对外服务系统web客户端 | public-vx.x.x.19xxxx.war |   |
| 后台管理系统web端 | manager-vx.x.x.19xxxx.war |  |
| Android | Android-vx.x.x.19xxxx.apk |   |
| iOS |  |  |
| 微信对外号 |  |  |
| 下载页 |  |  |

## 部署流程

为避免遇到各类安装失败的问题（如：服务间依赖关系、配置失败等），请按照推荐安装流程进行安装。具体流程如下图所示。



图1 部署流程图

## 部署架构图



图2 平台部署总体架构图

# 系统部署

本部分主要说明项目部署时需要部署的依赖环境及项目工程的步骤及注意事项。下面按照安装流程的先后顺序对系统的安装步骤进行描述。

## 数据库服务器部署

在一台服务器上单独部署Oracle数据库及Redis缓存数据库。

### 安装Oracle

具体请查看官方安装文档，或参考[CentOS 7 安装 Oracle 11g](https://www.jianshu.com/p/2fda26a7edf1)。

### 安装Redis

根据服务器系统版本，从[官网](https://redis.io/download)下载最新Stable版本，目前版本为v5.0.4[20190412]。安装步骤如下，也可参考[网上教程](https://cn.bing.com/search?q=centos+7.3+redis+%E8%87%AA%E5%90%AF%E5%8A%A8%E6%9C%8D%E5%8A%A1&go=%E6%8F%90%E4%BA%A4&qs=n&form=QBLH&sp=-1&pq=%E6%89%8B%E6%9C%BA%E5%8F%B7%E7%94%A8integer%E8%BF%98%E6%98%AFcharact&sc=0-20&sk=&cvid=EEDB5A5E26EF4BD1B1405F2A786497B4)。

*# 下载解压*

tar xzf redis-5.0.4.tar.gz

*# 移动免安装包到对应目录下*

mv /path/to/redis-5.0.4 /usr/local

cd /usr/local/redis-5.0.4

*# 编译*

make

*# 创建存储redis文件目录*

mkdir -p /usr/local/redis

*# 复制redis-server redis-cli到新建立的文件夹*

cp ./redis-server /usr/local/redis/

cp ./redis-cli /usr/local/redis/

*# 复制redis的配置文件*

cd ..

cp redis.conf /usr/local/redis/

*# 编辑配置文件*

cd /usr/local/redis/

vim redis.conf

*# 后台运行*

daemonize yes

添加开机启动服务

vim /etc/systemd/system/redis-server.service

[Unit]

Description=The redis-server Process Manager

After=syslog.target network.target

[Service]

Type=simple

PIDFile=/var/run/redis.pid

ExecStart=/usr/local/redis/redis-server

ExecReload=/bin/kill -USR2 $MAINPID

ExecStop=/bin/kill -SIGINT $MAINPID

[Install]

WantedBy=multi-user.target

 设置开机启动

systemctl daemon-reload

systemctl start redis-server.service

systemctl enable redis-server.service

确认是否启动

ps -ef | grep redis

创建redis命令软连接

ln -s /usr/local/redis/redis-cli /usr/bin/redis

测试redis

redis

Redis安装配置完成后，在API接口服务、对外服务系统、后台管理系统项目中配置文件修改对应的配置为本次Redis部署信息。

## 数据库生成或更新

### 数据库生成

初次部署，在Oracle数据库中新建表空间（xxx）和用户名（xxx）、密码（xxx）。

*-- 1.建表空间*

**CREATE** **TABLESPACE** xxx

**DATAFILE** '/path/to/db.dbf'

**SIZE** 1000M

**AUTOEXTEND** **ON**

**NEXT** 50M **MAXSIZE** 2048M

**EXTENT** **MANAGEMENT** **LOCAL**;

*-- 2.创建临时表空间*

**CREATE** **TEMPORARY** **TABLESPACE** temp

TEMPFILE '/path/to/db.tmp.dbf'

**SIZE** 50m

**AUTOEXTEND** **ON**

**NEXT** 32m **MAXSIZE** 2048m

**EXTENT** **MANAGEMENT** **LOCAL**;

*-- 3.建立用户*

**CREATE** **USER** xxx **IDENTIFIED** **BY** xxx

**DEFAULT** **TABLESPACE** xxx

**TEMPORARY** **TABLESPACE** temp;

### 数据库更新

迭代版本时，使用updateSql-vx.x.x.19xxxx.sql（updateSql-项目名-版本号.sql）文件更新数据库。

## 存储器挂载

采用华为可提供共享服务的磁盘，共10T空间，多台应用服务器平均分配，每台应用服务器挂载2T左右的空间，当作本地磁盘使用。需要修改项目配置文件，将视频、图片、日志、以及其他业务相关，且较大的文件存储地址修改为该磁盘下。

磁盘，挂载路径设置为 /mnt。上报等业务相关的视频、图片存储地址设置为 /mnt/xxx/upload/report，摄像机缩略图地址为 /mnt/xxx/upload/camera，日志地址 /mnt/xxx/logs，备份地址 /mnt/xxx/backup。

## 应用服务器部署

### 备份现有服务

在服务器/mnt/xxx/backup目录下，备份现有系统。

### 第三方服务放行列表

确保应用服务器可以通过互联网访问以下第三方服务，需在应用服务器上设置对应的放行策略。

表3 第三方服务放行地址

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **服务名** | **域名** | **IP** | **端口** | **地址** |
| 认证中心测试服务 |  | **xxx** | **8800** |  |
| 微信API接口 | **api.weixin.qq.com** | 101.227.162.120（可能更改） | 443 | api.weixin.qq.com |

### 应用服务器集群时钟同步

多服务器集群部署时，存在时间不一致的问题，导致程序中代码生成的数据时间对不上，整个业务流程无法正常操作。需要保证所有应用服务器的系统时间完全一致，这里使用一台服务器为同步源，建议用负载均衡服务器为同步源。

同步方法如下：（参考「[centos7.2上搭建ntp服务器，并实现时间同步](https://blog.csdn.net/yuanfang_way/article/details/53959591)」）

#### 确定ntp server及client

|  |  |  |
| --- | --- | --- |
| **IP** | **用处** | **备注** |
| 192.168.1.1 | 单独一台做为ntp server，时间同步源 | Nginx负载均衡服务器 |
| 192.168.1.2 | ntp client，时间同步客户端 | 应用服务器1 |
| 192.168.1.x | xxx | 应用服务器2 |

#### ntp时钟同步配置流程

1. 安装ntp

如果联网了，直接执行yum install ntp安装；离线情况下请参考「[centos7.1-离线-ntpd服务器搭建](https://blog.csdn.net/jpmsdn/article/details/81126340)」；

1. 所有节点（server、client）时区设置为中国区

timedatectl set-timezone Asia/Shanghai

1. server节点启动ntp服务

systemctl start ntpd

systemctl enable ntpd

1. server节点设置现在准确时间

timedatectl set-time HH:MM:%sS

1. server节点ntp.conf设置

设置ntp服务器为其自身；同时设置可以接受连接服务的客户端，其中server设置127.127.1.0为其自身，新增加一个restrict段为可以接受服务的网段。

*# Permit all access over the loopback interface. This could*

*# be tightened as well, but to do so would effect some of*

*# the administrative functions.*

restrict 127.0.0.1

restrict ::1

restrict 192.168.0.0 mask 255.255.0.0

*# Use public servers from the pool.ntp.org project.*

*# Please consider joining the pool(http://www.pool.ntp.org/join.html).*

server 127.127.1.0

1. 重启ntp服务

systemctl restart ntpd

1. client节点ntp.conf设置

*# Use public servers from the pool.ntp.org project.*

*# Please consider joining the pool(http://www.pool.ntp.org/join.html).*

server 192.168.201.5

1. client节点同步server节点时间

ntpdate 192.168.201.5

1. client节点启动ntpd服务

systemctl start ntpd

systemctl enable ntpd

1. 所有节点启动时间同步

timedatectl set-ntp yes

### 安装JDK

在应用服务器上安装JDK。

根据服务器系统版本，从官网上[下载JDK1.8](https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)，下载后安装并配置。具体安装配置过程请参考[CentOS7下安装JDK详细过程](https://blog.csdn.net/qq_25881261/article/details/79698252)。

### 安装Tomcat并设置自启动

根据服务器系统版本，从官网上[下载Tomcat8.x](https://tomcat.apache.org/download-80.cgi)，下载zip版本。具体安装配置过程请参考[centos7下安装tomcat8.0](https://blog.csdn.net/xiaobaoxiaodun/article/details/53539680)。

自启动设置，请参考[在CentOS7中设置Tomcat开机自启动](https://blog.csdn.net/u012351661/article/details/76254268)。

### API接口服务

该服务是提供给移动端及微信对外号使用的接口服务，无需单独测试。

#### 数据库数据源修改

使用解压软件打开war包中的webapps/projectname/WEB\_INF/classes目录下的jdbc.properties文件，修改数据源地址为指定的数据库部署地址，修改数据库用户名和密码为指定的用户名和密码。

# 系统数据库连接

jdbc.driver=oracle.jdbc.OracleDriver

jdbc.url=jdbc:oracle:thin:@ip:1521:orcl

jdbc.username=xxx

jdbc.password=xxx

#### 系统配置文件修改

打开在webapps/projectname/WEB\_INF/classes目录下的system.properties，修改以下配置为对应的信息。

*xxx*

#### Redis配置文件修改

打开在webapps/projectname/WEB\_INF/classes目录下的[spring-redis.xml](http://git.suncreate.com/pahf2/shgz-api/blob/dev/src/main/resources/spring-redis.xml)，修改redis服务器地址及端口为部署的Redis服务配置信息。

<bean id="jedisPool" class="redis.clients.jedis.JedisPool">

<constructor-arg index="0" ref="jedisPoolConfig"/>

*# redis服务器地址*

<constructor-arg index="1" value="127.0.0.1" type="java.lang.String"/>

*# redis服务端口*

<constructor-arg index="2" value="6379" type="int"/>

 <constructor-arg index="3" value="30000" type="int"/>

</bean>

#### 日志配置文件修改

打开在webapps/projectname/WEB\_INF/classes目录下的[log4j.xml](http://git.suncreate.com/pahf2/shgz-api/blob/dev/src/main/resources/log4j.xml)，分别修改debug、info、warn、error、fatal这5个级别的对应路径，对应到系统目录下（路径存在，否则将存在于tomcat/bin下），格式如下。（建议：集中多个系统的日志到一个路径下/home/xxx）

*# 原格式*
<param name="File" value="c:/xxx/logs/api/xxx\_xxx.log"/>

*# 修改为以下形式。/path/to/可修改为系统路径，如c:xxx，或者/home/xxx之类的地址。*

<param name="File" value="/**mnt**/xxx/logs/api/xxx\_xxx.log"/>

*# vim 全局替换命令*

:%s/c:\/xxx\/logs\/api/\/mnt\/xxx\/logs\/api/g

#### 域名配置

待补充

#### war包部署

将需要部署的war包（包括API接口服务、对外服务系统、后台管理系统），复制到Tomcat的webapp下。运行Tomcat的bin目录下的startup.sh，当Tomcat启动完毕后，到日志目录下看启动的日志情况，如果无异常则表明war包已成功地部署到tomcat服务器上。

#### 验证部署

内网打开对应链接地址，出现如下页面，证明服务启动成功。

并点击打开v1/common->v1/common/version，点击“试一下”按钮，查看请求是否成功。成功出现以下页面。

### 微信对外号

部署web 系统，微信对外号后台管理系统配置链接。

具体待补充

注：其他应用服务器部署同上。

### 下载页部署

单独的下载页工程，以静态服务的形式给出。

将静态文件放置在 /usr/tomcat/downloadpage，配置Ngixn 静态服务转发，访问该目录下的index.html。

再公网地址xxx访问查看，并验证下载。

## 负载均衡服务器部署

### 安装Nginx

#### 离线安装

服务器无法访问外网，nginx只能离线装。可先联网下载需要的依赖，需要的依赖包括：gcc、pcre、libstdc++、libstdc++-devel、gcc-c++、zlib，下载地址<https://pkgs.org>，选择对应的系统版本下载。

注意：

1.统一使用rpm格式；

2.各种依赖版本务必统一；

3.服务器如果在安装时没有点选一些模块，可能需要下载的东西较多，在服务器性能允许的情况下，建议全选安装；

4.服务器上已有的一些依赖，版本通常较低，rpm安装时可使用 --force 强制升级安装。

**步骤如下：**

1.安装gcc

rpm -ivh gcc-4.8.5-36.el7.x86\_64.rpm

2.安装pcre

*# 由于机器上已经有低版本的pcre，所以强制安装*

rpm -ivh pcre-8.32-17.el7.x86\_64.rpm --force

rpm -ivh pcre-devel-8.32-17.el7.x86\_64.rpm --force

3.安装libstdc++（gcc-c++依赖）

rpm -ivh libstdc++-4.8.5-36.el7.x86\_64.rpm --forcerpm -ivh libstdc++-devel-4.8.5-36.el7.x86\_64.rpm --force

4.安装gcc-c++

rpm -ivh gcc-c++-4.4.7-4.el6.x86\_64.rpm --force

5.安装zlib

rpm -ivh zlib-1.2.3-29.el6.x86\_64.rpm

rpm -ivh zlib-devel-1.2.3-29.el6.x86\_64.rpm

6.安装Nginx

从[Nginx](https://nginx.org)官网下载最新Stable版本，本次下载安装版本为1.41.2「20190417」。

*# 下载解压*

tar -xzvf nginx-1.14.2.tar.gz

*# 移动源代码到对应目录下*

sudo mv /path/to/nginx-1.14.2 /usr/local/

cd /usr/local/nginx-1.14.2

./configure

make

make install

7.运行Nginx

cd /usr/local/nginx/sbin

./nginx

1. 验证

检查nginx.conf配置文件是否正确。

/usr/local/nginx/sbin/nginx -t

正确之后，浏览器打开[localhost:8080](http://localhost:8080)，如果可访问welcome to nginx页面，表示配置完成。

#### 常用命令

*# 通过help查看常用命令*

/usr/local/nginx/sbin/nginx -h

Options:

 -?,-h : this help

 -v : show version **and** exit

 -V : show version **and** configure options then exit

 -t : test configuration **and** exit

 -T : test configuration, dump it **and** exit

 -q : suppress non-error messages during configuration testing

 -s signal : send signal **to** a master process: stop, quit, reopen, reload

 -p prefix : set prefix path (default: /usr/local/nginx/)

 -c filename : set configuration file (default: conf/nginx.conf)

 -g directives : set global directives out of configuration file

#### 关闭Nginx代理服务器防火墙；

*# 临时关闭*

systemctl stop firewalld.service

*# 禁止开机启动*

systemctl disable firewalld

*# 查看防火墙状态*

systemctl status firewalld

#### 端口验证

确保映射需要的端口到外网，如80、8080、8081。

#### 自启动设置

1. 在系统服务目录里创建nginx.service文件
**vim /usr/lib/systemd/system/nginx.service**
2. 写入内容如下：

[Unit]

Description=nginx

After=network.target

[Service]

Type=forking

ExecStart=/usr/local/nginx/sbin/nginx

ExecReload=/usr/local/nginx/sbin/nginx -s reload

ExecStop=/usr/local/nginx/sbin/nginx -s quit

PrivateTmp=true

[Install]

WantedBy=multi-user.target

3、设置开机自启动
systemctl enable nginx.service

4、查看Nginx状态
systemctl status nginx.service
很奇怪，明明启动成功了，为什么显示Active: inactive (dead)？

5、杀死Nginx重启Nginx

pkill -9 nginx

ps aux | grep nginx

systemctl start nginx

再次查看状态，变成了active。

6、重启服务器测试效果
reboot

7、再次连接后，查看服务状态
systemctl status nginx.service

看到Nginx已经启动，至此，Nginx自启动配置成功。

### 配置Nginx

下面介绍配置Nginx的过程。

在本平台的部署方案中，Nginx部署一台单独的服务器，采取一对多的方式来实现负载均衡。正式上线必须使用不同的域名来映射同一个IP，测试时如果使用同一个IP，必须配置映射不同的端口。当然，测试时可使用假域名的方式，需要配置客户端hosts中的域名与服务器IP的对应关系。

在Nginx主目录下新建conf.d文件夹。我们采用一个服务对应一个配置的原则，所有配置文件都放到该文件夹下。

conf/主目录下nginx.conf为默认配置，conf.d文件夹包括static.conf，api.conf，public.conf，manager.conf，wechat.conf，downloadpage.conf等配置文件，单个配置文件配置多个节点组成服务集群。

#### 主配置文件nginx.conf修改

worker\_processes 4; //与服务器核数相当，可设为自动

events {

 worker\_connections 1024;

}

http {

 include mime.types;

 default\_type application/octet-stream;

 sendfile on;

 keepalive\_timeout 65;

 fastcgi\_connect\_timeout 300;

 fastcgi\_send\_timeout 300;

 fastcgi\_read\_timeout 300;

 fastcgi\_buffer\_size 128k;

 fastcgi\_buffers 2 256k;

 fastcgi\_busy\_buffers\_size 256k;

 fastcgi\_temp\_file\_write\_size 256k;

 gzip on;

 client\_header\_buffer\_size 32k;

 client\_max\_body\_size 20m; //文件传输大小

 **server** {

 listen 80;

 server\_name 192.168.1.1;

 *#charset utf-8;*

 *#access\_log logs/host.access.log main;*

 location / {

 root html;

 index index.html index.htm;

 }

 error\_page 500 502 503 504 /50x.html;

 location = /50x.html {

 root html;

 }

 }

 *# HTTPS server*

 *#*

 *#server {*

 *# listen 443 ssl;*

 *# server\_name localhost;*

 *# ssl\_certificate cert.pem;*

 *# ssl\_certificate\_key cert.key;*

 *# ssl\_session\_cache shared:%sSL:1m;*

 *# ssl\_session\_timeout 5m;*

 *# ssl\_ciphers HIGH:!aNULL:!MD5;*

 *# ssl\_prefer\_server\_ciphers on;*

 *# location / {*

 *# root html;*

 *# index index.html index.htm;*

 *# }*

 *#}*

 include conf.d/\*.conf;

}

#### api.conf配置

upstream apiCluster { *// 集群名称*

 server 192.168.100.2:8081;

 server 192.168.100.3:8081;

}

**server** {

 listen 80;

 server\_name api.xxx.com; *//访问api.xxx.com*

 location / {

 proxy\_pass http://apiCluster*; //必须与upstream配置集群名一致*

}

location ^~ /upload/ {

 *#此处为项目上传文件的路径，如*

 root /mnt/xxx/;

}

}

#### downloadpage.conf配置

upstream downloadCluster { *// 集群名称*

 server 192.168.100.2:8085;

 server 192.168.100.3:8085;

}

server {

 listen 80;

server\_name download.xxx.com; *//访问download.xxx.com*

location / {

 proxy\_pass <http://downloadCluster;> *//必须与upstream配置集群名一致*

}

location ^~ /downloadpage {

 root /mnt/xxx/downloadpage;

 index index.html index.htm;

}

}

修改配置文件后，执行以下命令使之生效。

/usr/local/nginx/sbin/nginx -s reload

## 客户端软件部署

### Android客户端

1. 请运维人员在部署后向研发反馈以下信息：
2. API系统地址

如：

http://ip:8080/api/

1、根据运维人员的反馈，研发人员更换信息，打包APK文件，在全项目提交测试申请；

2、运维人员在后台管理系统中，使用APK管理模块（暂未完成），上传APK文件；

3、测试人员从全项目下载APK文件，安装测试；也可以从提供的下载页「待开发」下载安装测试。

4、系统中某些功能需要注册才能使用，用户体系使用xxx统一认证平台，测试时账户名密码需要自行用本人手机号/身份证申请，注意只能是xxx居民可用。

### iOS客户端

运维人员在部署后向研发反馈以下信息：

1. 请运维人员在部署后向研发反馈以下信息：

a. API系统地址

如：

http://ip:8080/api/

1、根据运维人员反馈的信息，研发人员更改代码，并打包上传Apple Store；

2、等待苹果官方审核通过；

3、测试人员从Apple Store「应用名」下载安装，进行测试。

4、APP中某些功能需要注册才能进入使用，其中用户体系使用xxx统一认证平台，测试时账户名密码需要自行用本人手机号/身份证申请，注意只能是xxx居民可用。

# 维护说明

## 日常维护注意事项

待补充

## 常见故障处理

待补充完整

### Redis 错误

未安装、未启动，或配置错误。

### Nginx错误

#### 404 not found

地址错误，请确保输入正确的链接地址。

#### 504 gateway timeout

1. Nginx原因

配置的缓存空间不足，修改Nginx配置，见「[3.5.2.1主配置文件nginx.conf修改](#_主配置文件nginx.conf修改)」。

1. 第三方服务无法访问

应用服务器放行，具体见「[3.4.2第三方服务放行列表](#_第三方服务放行列表)」。

#### 数据时间错乱

应用服务器集群之间时间不同步，解决方式见「[3.4.3应用服务器集群时钟同步](#_应用服务器集群时钟同步)」。

#### 文件超限：nginx 413 Request Entity Too Large

原因是因为Nginx默认上传限制为1m，解决方式是在nginx的http节点中加入以下内容：

client\_max\_body\_size 20m;