

Linux 系统集群架构线上 项目配置实战



About

主要记录整个线上生产环境从架构设计开始，直到配置完成
满足整个线上运行需求的整个过程

中国霸都

QQ 群：580703149

公众号：扫文末二维码

2017/9/21

目录

前言导读	2
一：项目整体介绍	2
二：服务器逻辑架构图	3
三：服务器基础环境配置	4
四：分发 hosts 文件到其它服务器	6
五：前端 APP 负载服务器安装配置	10
六：管理后台应用工程布署与配置	15
七：官方网站与 FTP 站点配置	17
八：数据库安装与主从同步配置	20
九：Redis 与 zabbix 的安装与配置	25
十：全网数据备份规划	25
十一：数据库备份配置	26
十二：静态资源服务器的数据同步配置	28
十三：同步全网备份数据至备份服务器	30

友侃有笑 | 分享干货的公众号



公众号:youkanyouxiao

长按识别二维码加关注

更多内容更关注民工哥新书《跟菜鸟一起学习 Linux 系统》，公众号后台回复“新书”获取详情

前言导读

目前民工哥的公众号也运营一年多的时间，从刚开始的网络方面的基础知识转到 Linux 系统方面，前后文章也写了许多，前段时间顺便回看了下前面的文章，中小企业架构是完全可以完成的，包括整个的架构设计与规划、配置及后续维护等。

因此，民工哥才有此想法，将目前的线上环境从当时项目开始到最终结束的整个过程详细记录下来，写成文章发布出来，算是共享给各位持续关注公众号的小伙伴，可以做个平时的参考文档，同时也希望各位小伙伴多提意见与建议，一起共同提高、共同成长!!!!

一：项目整体介绍

整个项目包括以下几个应用工程

- 1、用户 APP
- 2、商家 APP
- 3、业务平台管理后台
- 4、商家业务管理后台
- 5、官网

整体项目规划设计如下

- 1、用户 APP 采用 JAVA 环境布署，初始环境两台服务器，实现负载均衡与高可用，前端负载使用 nginx 做负载
- 2、商家 APP 采用 JAVA 环境布署，初始环境两台服务器，实现负载均衡与高可用，前端负载使用 nginx 做负载
- 3、业务平台管理后台采用 JAVA 环境布署，初始环境一台服务器，前端使用 nginx 反向代理，后续业务流量扩大，架构重新扩展
- 4、商家平台管理后台采用 JAVA 环境布署，初始环境一台服务器，前端使用 nginx 反向代理，后续业务流量扩大，架构重新扩展
- 5、官方网站都是一些静态页面，采用 nginx 布署，WEB 服务另安装 FTP 服务器，做为业务管理后台一些静态资源的上传
- 6、数据库使用 Mysql 主从复制架构，程序上实现读写分享，从库开启 binlog 做备份使用
- 7、全网所有备份数据全部同步到专业内网备份服务器，所有业务备份数据保留 30 天
- 8、全网所有服务器监控（基本监控与服务监控）都采用 zabbix 监控

服务器 IP 地址规划如下

序号	服务器名称	外网地址	内网地址	备注
1	nginx-lb	120.0.0.1	10.0.0.1	
2	nginx-proxy	120.0.0.2	10.0.0.2	
3	nginx	120.0.0.3	10.0.0.3	
4	userapp	120.0.0.4	10.0.0.4	
5	sjapp	120.0.0.5	10.0.0.5	
6	admin	120.0.0.6	10.0.0.6	
7	shangjia	120.0.0.7	10.0.0.7	
8	web	120.0.0.8	10.0.0.8	
9	mysql-m	120.0.0.9	10.0.0.9	
10	mysql-s	120.0.0.10	10.0.0.10	
11	ftp-backup	120.0.0.11	10.0.0.11	
12	redis	120.0.0.12	10.0.0.12	
13	backup-server	120.0.0.13	10.0.0.13	
14	zabbix	120.0.0.14	10.0.0.14	

服务器环境与应用软件版本介绍

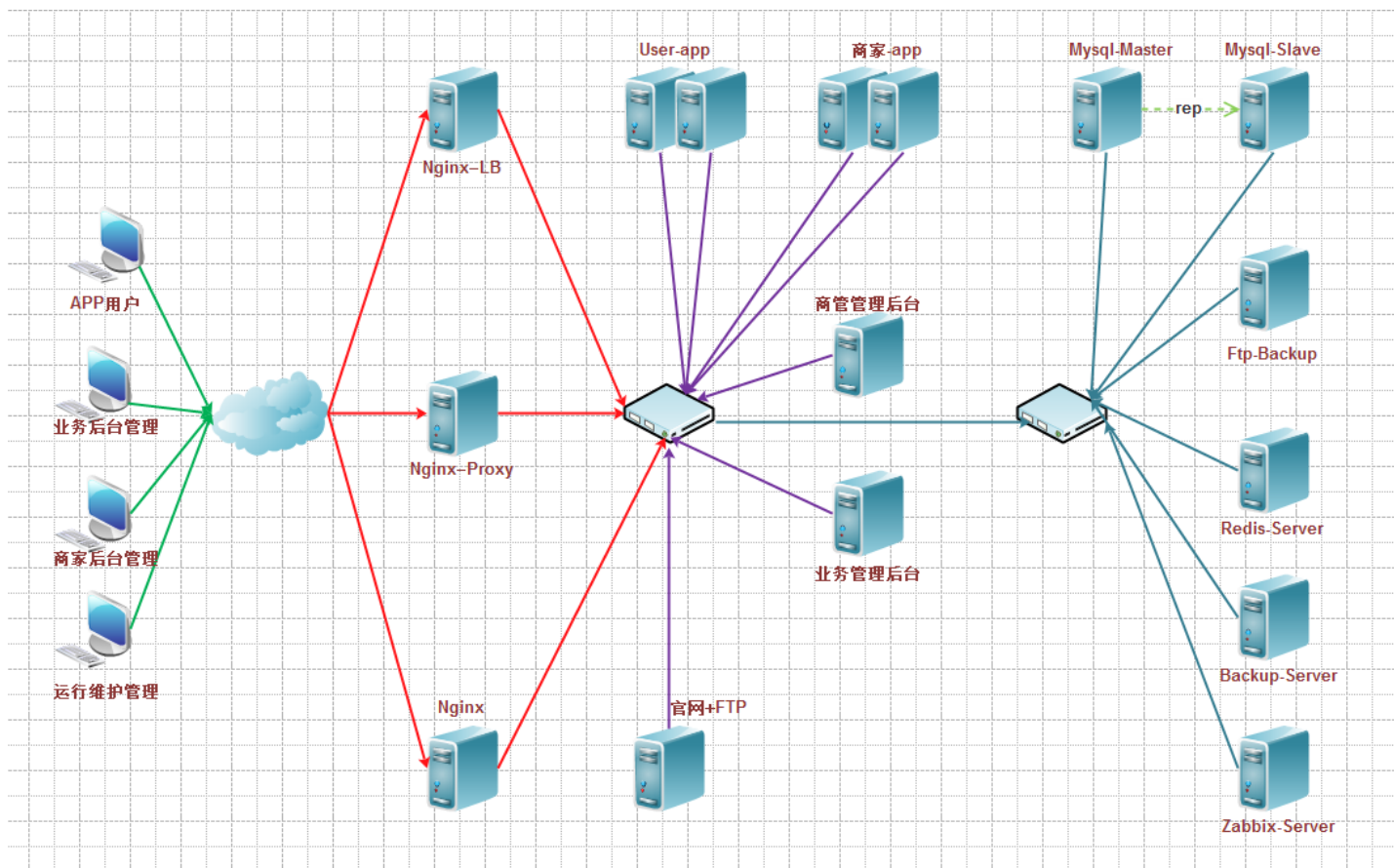
服务器环境

```
[root@centos ~]# cat /etc/redhat-release
CentOS release 6.8 (Final)
[root@centos ~]# uname -r
2.6.32-642.el6.x86_64
```

应用软件版本

```
Nginx 1.12.1
Mysql 5.7.17
Jdk 1.8
Tomcat 8.5.9
Zabbix 3.0
Yum 源使用 163、阿里去、sohu 国内的源都可以，自行选择
Redis 4.0.1
```

二：服务器逻辑架构图



三：服务器基础环境配置

因使用云主机服务器，初始系统安装就略过了，如果需要批量安装的请参考前面的文章
[文章链接如下](#)

配置国内 YUM 源，以 163 为例

```
[root@centos ~]# cd /etc/yum.repos.d/
[root@centos yum.repos.d]# wget http://mirrors.163.com/.help/CentOS6-Base-163.repo
[root@centos yum.repos.d]# mv CentOS-Base.repo CentOS-Base.repo.bak
[root@centos yum.repos.d]# mv CentOS6-Base-163.repo CentOS-Base.repo
```

安装需要的软件包

```
[root@centos ~]# yum install lsof vim telnet dos2unix lrzsz tree -y
```

关闭不需要启动的服务

```
[root@centos ~]# for service in `chkconfig --list|grep 3:on|awk '{print $1}'`;do chkconfig --level 3 $service off;done
[root@centos ~]# for service in crond network sshd rsyslog;do chkconfig --level 3 $service on;done
[root@centos ~]# chkconfig --list|grep 3:on
```

```
[root@centos ~]# chkconfig --list|grep 3:on
crond          0:off  1:off  2:on   3:on   4:on   5:on   6:off
network        0:off  1:off  2:on   3:on   4:on   5:on   6:off
rsyslog         0:off  1:off  2:on   3:on   4:on   5:on   6:off
sshd            0:off  1:off  2:on   3:on   4:on   5:on   6:off
[root@centos ~]#
```

修改 SSH 服务的默认配置

```
cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
sed -i 's#Port 22#Port 2233#g' /etc/ssh/sshd_config
sed -i 's#PermitRootLogin yes#PermitRootLogin no#g' /etc/ssh/sshd_config
sed -i 's#PermitEmptyPasswords no#PermitEmptyPasswords no#g' /etc/ssh/sshd_config
sed -i 's#UseDNS yes#UseDNS no#g' /etc/ssh/sshd_config
egrep "UseDNS|2233|RootLogin|EmptyPass" /etc/ssh/sshd_config
/etc/init.d/sshd reload
```

关闭 selinux

```
配置文件/etc/selinux/config
sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/selinux/config
egrep "SELINUX=" /etc/selinux/config
```

关闭防火墙

如果不是对外网的服务器，直接关闭防火墙，有外网地址的可建议开启，并允许相关的服务端口，并发高的服务器也不需要开启防火墙，可能会影响性能

```
/etc/init.d/pintables stop
/etc/init.d/pintables status
pintables: Firewall is not running.
```

锁定系统关键文件

```
for file in /etc/passwd /etc/shadow /etc/services /etc/initial /etc/rc.local
do
    chattr +i $file
done
```

配置普通操作用户及 sudo 权限

具体操作请参考前面的文章，文章链接如下

配置全网服务器时间同步

```
/usr/sbin/ntpdate time.windows.com
cat>>/var/spool/cron/root<<EOF
#this cron is for ntpdate
*/5 * * * * /usr/sbin/ntpdate time.windows.com >/dev/null 2>&1
#cron config end
EOF
```

配置全网服务器 hosts 文件

```
Cat>>/etc/hosts<<EOF
10.0.0.1      nginx-lb
10.0.0.2      nginx-proxy
10.0.0.3      nginx
10.0.0.4      userapp
10.0.0.5      sjapp
10.0.0.6      admin
10.0.0.7      shangjia
10.0.0.8      web
10.0.0.9      mysql-m
10.0.0.10     mysql-s
10.0.0.11     ftp-backup
10.0.0.12     redis
10.0.0.13     backup-server
10.0.0.14     zabbix
EOF
```

规范所有服务器目录

```
[root@centos ~]# mkdir /download/tools/ -p
[root@centos ~]# mkdir /app/logs -p
[root@centos ~]# mkdir /app -p
[root@centos ~]# mkdir /server/{scripts,backup} -p
```

至此基本的服务器优化配置也就结束了，接下来就是安装各类服务

四：分发 hosts 文件到其它服务器

分发服务器使用全网备份服务器

首先创建分发用户

```
[root@centos ~]# useradd fenfa -u 999
[root@centos ~]# echo "123456"|passwd --stdin fenfa
Changing password for user fenfa.
passwd: all authentication tokens updated successfully.
[root@centos ~]# /bin/cp /etc/sudoers /etc/sudoers.bak
[root@centos ~]# echo "fenfa ALL=(root) NOPASSWD:/usr/bin/rsync" >>/etc/sudoers
[root@centos ~]# tail -1 /etc/sudoers
fenfa ALL=(root) NOPASSWD:/usr/bin/rsync
```

切换到 fenfa 用户创建密钥

```
[root@centos ~]# su - fenfa
[fenfa@centos ~]$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/fenfa/.ssh/id_dsa):
Created directory '/home/fenfa/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/fenfa/.ssh/id_dsa.
Your public key has been saved in /home/fenfa/.ssh/id_dsa.pub.
The key fingerprint is:
3c:5a:ff:a8:f7:02:f9:a1:ab:b7:49:a5:5c:24:18:5d fenfa@centos
The key's randomart image is:
+--[ DSA 1024]-----+
|      .. .E      |
|      o.         |
|      . . .      |
|      . o        |
|      S.o        |
|      +o*        |
|      . ++..     |
|      .o.+o      |
|      .o==.oo    |
+-----+
[fenfa@centos ~]$
```

安装配置 expect 服务

```
[root@centos ~]$ yum install expect* -y
```

编写脚本

```
[fenfa@centos ~]$ vim host.list
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
10.0.0.8
10.0.0.9
10.0.0.10
10.0.0.11
10.0.0.12
10.0.0.13
10.0.0.14
```

```
[fenfa@centos ~]$ vim fenfa.exp
#!/usr/bin/expect
if {$argc != 2} {
send_user "usage: expect xxxx.exp file host\n"
it
}
#define var
set file [lindex $argv 0]
set host [lindex $argv 1]
```



```

set password "123456"

spawn ssh-copy-id -i $file fenfa@$host
expect {
    "yes/no" {send "yes\r";exp_continue}
    "password" {send "$password\r"}
}
expect eof
[fenfa@centos ~]$ vim fenfa_key.sh
#!/bin/sh
. /etc/init.d/functions
for ip in `cat host.list`
do
expect fenfa.exp ~/.ssh/id_dsa.pub $ip >/dev/null 2>&1
if [ $? -eq 0 ];then
    action "$ip" /bin/true
else
    action "$ip" /bin/false
fi
done

```

执行脚本结果如下

```

[fenfa@centos ~]$ sh fenfa_key.sh
10.0.0.1          [ OK ]
10.0.0.2          [ OK ]
10.0.0.3          [ OK ]
10.0.0.4          [ OK ]
10.0.0.5          [ OK ]
10.0.0.6          [ OK ]
10.0.0.7          [ OK ]
10.0.0.8          [ OK ]
10.0.0.9          [ OK ]
10.0.0.10         [ OK ]
10.0.0.11         [ OK ]
10.0.0.12         [ OK ]
10.0.0.13         [ OK ]
10.0.0.14         [ OK ]
[root@centos ~]# su - fenfa
[fenfa@centos ~]$ ll /home/fenfa/.ssh/
total 4
-rw----- 1 fenfa fenfa 602 Sep 21 12:14 authorized_keys

```

编写分发文件脚本

```
[fenfa@centos ~]$ vim fenfa_file.sh
#!/bin/sh
. /etc/init.d/functions
if [ $# -ne 2 ]
then
    echo "USE $0 file dir"
    exit 1
fi

for ip in `cat host.list`
do
    rsync -avzP $1 -e 'ssh -t -p 2233' fenfa@$ip:~ >/dev/null 2>&1
    ssh -t -p 2233 fenfa@$ip sudo rsync -avzP ~/$1 $2 >/dev/null 2>&1
    if [ $? -eq 0 ]
    then
        action "$ip complted" /bin/ture
    else
        action "$ip does't complted" /bin/ture
    fi
done
```

```
[fenfa@centos ~]$ sh fenfa_file.sh
USE fenfa_file.sh file dir
[fenfa@centos ~]$ sh fenfa_file.sh host.list /etc/
10.0.0.1 complted [ OK ]
10.0.0.2 complted [ OK ]
10.0.0.3 complted [ OK ]
10.0.0.4 complted [ OK ]
10.0.0.5 complted [ OK ]
10.0.0.6 complted [ OK ]
10.0.0.7 complted [ OK ]
10.0.0.8 complted [ OK ]
10.0.0.9 complted [ OK ]
10.0.0.10 complted [ OK ]
10.0.0.11 complted [ OK ]
10.0.0.12 complted [ OK ]
10.0.0.13 complted [ OK ]
10.0.0.14 complted [ OK ]
```

```
[fenfa@centos ~]$ ll /etc/host.list
-rw-rw-r-- 1 fenfa fenfa 131 Sep 16 12:45 /etc/host.list
```

```
[fenfa@centos ~]$ ll /etc/host.list
-rw-rw-r-- 1 fenfa fenfa 131 Sep 16 12:45 /etc/host.list
[fenfa@centos ~]$ cat /etc/host.list
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
10.0.0.8
10.0.0.9
10.0.0.10
10.0.0.11
10.0.0.12
10.0.0.13
10.0.0.14
```

五：前端 APP 负载服务器安装配置

安装所需的依赖包

```
[root@centos ~]# yum install pcre-devel zlib-devel openssl-devel gcc-c++ -y
```

编译安装 nginx

```
cd /download/tools/  
wget http://nginx.org/download/nginx-1.12.1.tar.gz  
tar zxf nginx-1.12.1.tar.gz  
cd nginx-1.12.1  
./configure --prefix=/app/nginx-1.12.1  
make && make install
```

友侃有笑 | 分享干货的公众号



公众号:youkanyouxiao



长按识别二维码加关注

```

115 history
[root@centos ~]# tree /app/nginx-1.12.1/ -L 2
/app/nginx-1.12.1/
├── conf
│   ├── fastcgi.conf
│   ├── fastcgi.conf.default
│   ├── fastcgi_params
│   ├── fastcgi_params.default
│   ├── koi-utf
│   ├── koi-win
│   ├── mime.types
│   ├── mime.types.default
│   ├── nginx.conf
│   ├── nginx.conf.default
│   ├── scgi_params
│   ├── scgi_params.default
│   ├── uwsgi_params
│   ├── uwsgi_params.default
│   └── win-utf
├── html
│   ├── 50x.html
│   └── index.html
├── logs
└── sbin
    └── nginx

4 directories, 18 files

```

```

[root@centos ~]# cd /app/
[root@centos app]# ln -s nginx-1.12.1 nginx
[root@centos ~]# cd /app/nginx/conf/
[root@centos conf]# mkdir extra
[root@centos conf]# cp nginx.conf nginx.conf.bak
在 nginx.conf 文件后增加下面的配置
include extra/*.conf;
[root@centos conf]# cd extra/
oot@centos extra]# vim user.app.conf

```

```

#
# HTTPS server configuration
#
upstream userapp {
    server 10.0.0.4:8080;
    server 10.0.0.4:8081;
}
server {
    listen      80;
    server_name app.mingongge.com;
    location / {
        proxy_pass http://userapp;
    }
}

```

```

        proxy_connect_timeout 600;
        proxy_read_timeout 600;
        proxy_send_timeout 600;
    }
[root@centos conf]# cd extra/
[root@centos conf]# vim sj.app.conf
#
# HTTPS server configuration
#
upstream sjapp {
    server 10.0.0.5:8080;
    server 10.0.0.5:8081;
}

server {
    listen      80;
    server_name sjapp.mingongge.com;
    location / {
        proxy_pass http://sjapp;
        proxy_connect_timeout 600;
        proxy_read_timeout 600;
        proxy_send_timeout 600;
    }
}

```

后端 APP 服务器 JAVA 环境安装配置

```

[root@centos tools]# ll
total 181168
-rw-r--r-- 1 root root 185515842 Sep 20 15:52 jdk-8u144-linux-x64.tar.gz
[root@centos tools]# tar xzf jdk-8u144-linux-x64.tar.gz -C /usr/local/
[root@centos tools]# ln -s /usr/local/jdk1.8.0_144 /usr/local/jdk

```

配置 JDK 环境变量

```

[root@centos tools]# cat >>/etc/profile<<EOF
export JAVA_HOME=/usr/local/jdk
export CLASSPATH=. CLASSPATH:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$JAVA_HOME/bin:$PATH
EOF
[root@centos tools]# tail -3 /etc/profile
export JAVA_HOME=/usr/local/jdk
export CLASSPATH=. CLASSPATH:/lib/dt.jar:/lib/tools.jar
export PATH=$JAVA_HOME/bin:$PATH
[root@centos tools]# source /etc/profile

```

```
[root@centos tools]# java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
```

后端 tomcat 配置

wget

<http://mirrors.hust.edu.cn/apache/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz>

```
[root@centos tools]# tar xzf apache-tomcat-8.5.20.tar.gz -C /usr/local/
[root@centos tools]# ln -s /usr/local/apache-tomcat-8.5.20 /usr/local/tomcat
[root@centos tools]# cd /usr/local/apache-tomcat-8.5.20/conf/
```

```
[root@centos conf]# vim server.xml
```

-----此处省略N行-----

```
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />
<Context path="/" docBase="/www/userapp" debug="0" reloadable="true"
crossContext="true" />
```

#增加上述站点目录

```
[root@centos conf]# mkdir /www/userapp -p
[root@centos conf]# echo "this is the userapp server" >/www/userapp/index.html
[root@centos conf]# ../bin/startup.sh
Using CATALINA_BASE:   /usr/local/apache-tomcat-8.5.20
Using CATALINA_HOME:   /usr/local/apache-tomcat-8.5.20
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-8.5.20/temp
Using JRE_HOME:        /usr/local/jdk
Using CLASSPATH:
/usr/local/apache-tomcat-8.5.20/bin/bootstrap.jar:/usr/local/apache-tomcat-8.5.20
/bin/tomcat-juli.jar
Tomcat started.
[root@centos conf]# lsof -i :8080
COMMAND PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
java    1587 root  48u  IPv6 18137      0t0  TCP *:webcache (LISTEN)
```

```

<!-- Access log processes all example.
      Documentation at: /docs/config/valve.html
      Note: The pattern used is equivalent to using pattern="common" -->
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
      prefix="localhost_access_log" suffix=".txt"
      pattern="%h %l %u %t &quot;%r&quot; %s %b" />
<Context path="" docBase="/www/userapp" debug="0" reloadable="true" crossContext="true" />

</Host>
</Engine>
</Service>
"server.xml" 168L, 7611C written
[root@centos conf]# mkdir /www/userapp -p
[root@centos conf]# echo "this is the userapp server" >/www/userapp/index.html
[root@centos conf]# ../bin/startup.sh
Using CATALINA_BASE:   /usr/local/apache-tomcat-8.5.20
Using CATALINA_HOME:   /usr/local/apache-tomcat-8.5.20
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-8.5.20/temp
Using JRE_HOME:        /usr/local/jdk
Using CLASSPATH:       /usr/local/apache-tomcat-8.5.20/bin/bootstrap.jar:/usr/local/apache-tomcat-8.5.20/bin/tomcat-juli.jar
Tomcat started.
[root@centos conf]# lsof -i :8080
COMMAND PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
java    1587 root   48u  IPv6 18137    0t0  TCP *:webcache (LISTEN)
[root@centos conf]# curl http://127.0.0.1:8080
this is the userapp server
[root@centos conf]#

```

另外一台用户 APP 服务器按上述的配置进行

```
[root@centos conf]# mkdir /www/userapp -p
```

```
[root@centos conf]# echo "this is the second userapp" >/www/userapp/index.html
```

前端负载均衡测试

```
[root@centos conf]# curl http://10.0.0.1
```

this is the userapp server

```
[root@centos conf]# curl http://10.0.0.1
```

this is the second userapp

```
[root@centos conf]# curl http://app.mingongge.com
```

this is the userapp server

```
[root@centos conf]# curl http://app.mingongge.com
```

this is the second userapp

```
[root@centos extra]# curl http://sjapp.mingongge.com
```

this is the first sjapp server

```
[root@centos extra]# curl http://sjapp.mingongge.com
```

this is the second sjapp

浏览器访问测试

本地浏览器测试需要配置 hosts 文件

10.0.0.1 app.mingongge.com





六：管理后台应用工程布署与配置

前端 nginx 反向代理服务器配置

安装所需的依赖包

```
[root@centos ~]# yum install pcre-devel zlib-devel openssl-devel gcc-c++ -y
```

编译安装 nginx

```
cd /download/tools/
wget http://nginx.org/download/nginx-1.12.1.tar.gz
tar zxf nginx-1.12.1.tar.gz
cd nginx-1.12.1
./configure --prefix=/app/nginx-1.12.1
make && make install
[root@centos ~]# cd /app/
[root@centos app]# ln -s nginx-1.12.1 nginx
[root@centos ~]# cd /app/nginx/conf/
[root@centos conf]# mkdir extra
[root@centos conf]# cp nginx.conf nginx.conf.bak
在 nginx.conf 文件后增加下面的配置
include extra/*.conf;
[root@centos ~]# cd /app/nginx/conf/extra/
[root@centos extra]# vim admin.mingongge.conf
#
# HTTPS server configuration
#
```

```
server {
    listen      80;
    server_name admin.mingongge.com;
    location / {
        proxy_pass http://10.0.0.6:8080;
        proxy_connect_timeout 600;
        proxy_read_timeout 600;
        proxy_send_timeout 600;
    }
}
```



```
[root@centos extra]# vim shangjia.mingongge.conf
#
# HTTPS server configuration
#
server {
    listen      80;
    server_name shangjia.mingongge.com;
    location / {
        proxy_pass http://10.0.0.7:8080;
        proxy_connect_timeout 600;
        proxy_read_timeout 600;
        proxy_send_timeout 600;
    }
}
```

业务平台管理后台 admin.mingongge.com

JDK Tomcat 环境安装请参考前面的步骤

```
<Context path="" docBase="/www/admin" debug="0" reloadable="true"
crossContext="true" />
```

商家管理后台 shangjia.mingongge.com

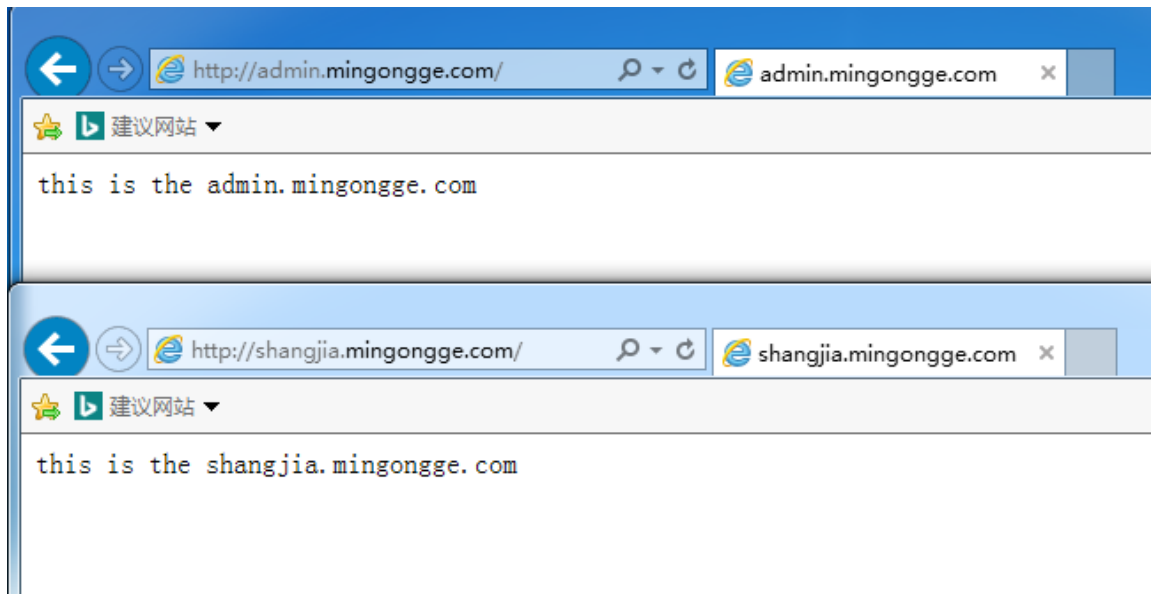
JDK Tomcat 环境安装请参考前面的步骤

```
<Context path="" docBase="/www/shangjia" debug="0" reloadable="true"
crossContext="true" />
```

测试反向代理

```
[root@centos ~]# curl http://admin.mingongge.com
this is the admin.mingongge.com
[root@centos ~]# curl http://shangjia.mingongge.com
this is the shangjia.mingongge.com
```

```
[root@centos ~]# curl http://admin.mingongge.com
this is the admin.mingongge.com
[root@centos ~]# curl http://shangjia.mingongge.com
this is the shangjia.mingongge.com
[root@centos ~]# curl http://admin.mingongge.com
this is the admin.mingongge.com
[root@centos ~]# curl http://shangjia.mingongge.com
this is the shangjia.mingongge.com
[root@centos ~]#
```



七：官方网站与 FTP 站点配置

基本的架构也是一样使用反向代理，为了后期整体架构扩展 Nginx FTP 的安装就不再描述了，太简单了，扯多了累

前面反向代理配置如下

```
[root@centos extra]# vim web.mingongge.conf
```

```
#
```

```
# HTTPS server configuration
```

```
#
```

```
server {  
    listen      80;  
    server_name www.mingongge.com;  
    location / {  
        proxy_pass http://10.0.0.8;  
        proxy_connect_timeout 600;  
        proxy_read_timeout 600;  
        proxy_send_timeout 600;
```

```
}
```

```
}
```

```
[root@centos extra]# vim ftp.mingongge.com
```

```
#
```

```
# HTTPS server configuration
```

```
#
```

```
server {  
    listen      80;  
    server_name ftp.mingongge.com;
```

```

    location / {
        proxy_pass http://10.0.0.8:88;
        proxy_connect_timeout 600;
        proxy_read_timeout 600;
        proxy_send_timeout 600;
    }
}

```

后端 WEB 服务器配置

```

[root@centos html]# vim index.html
welcome to mingongge.s web stie!!!!!!!!!!!!!!
[root@centos ~]# /usr/local/nginx/sbin/nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
[root@centos ~]# /usr/local/nginx/sbin/nginx
[root@centos ~]# curl 10.0.0.8
welcome to mingongge.s web stie!!!!!!!!!!!!!!

```

FTP 这块的配置其实是就是管理后台通过应用程序上传图片，前面用户通过 nginx 能访问到正确的页面即可，由于线下测试环境，不可能拿生产代码来做实验，因此管理后台上传图片无法模拟，因此就配置下访问图片即可

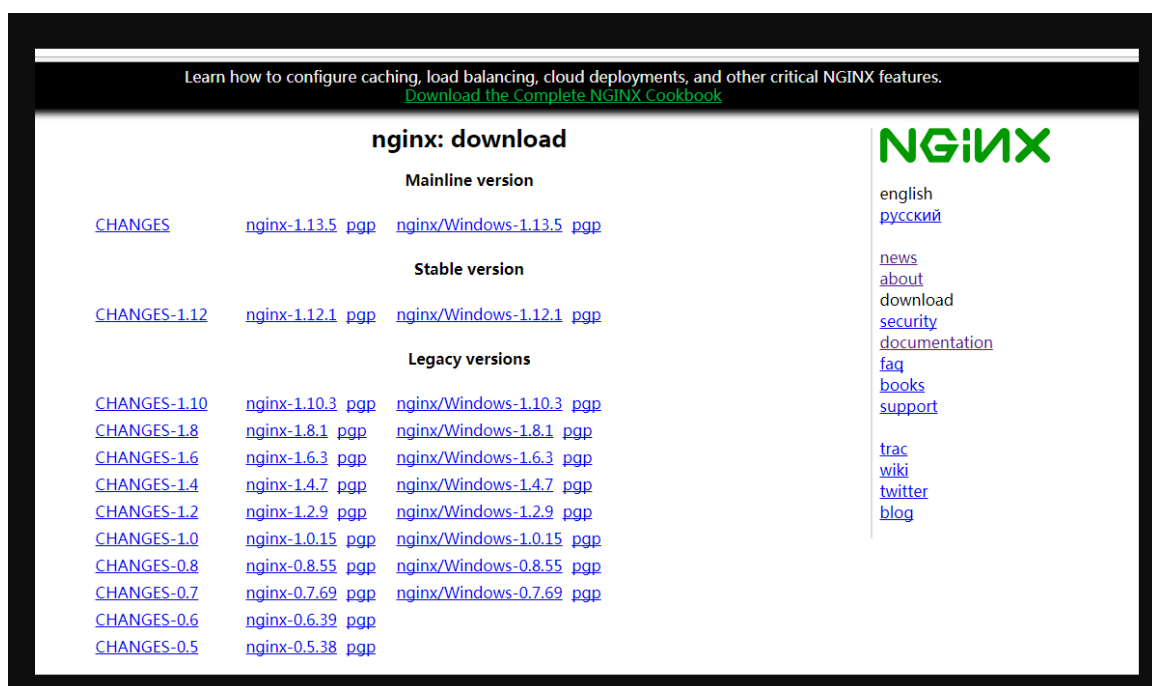
```

[root@centos conf]# cd extra/
[root@centos extra]# vim ftp.mingongge.conf
server {
    listen      88;
    server_name localhost;

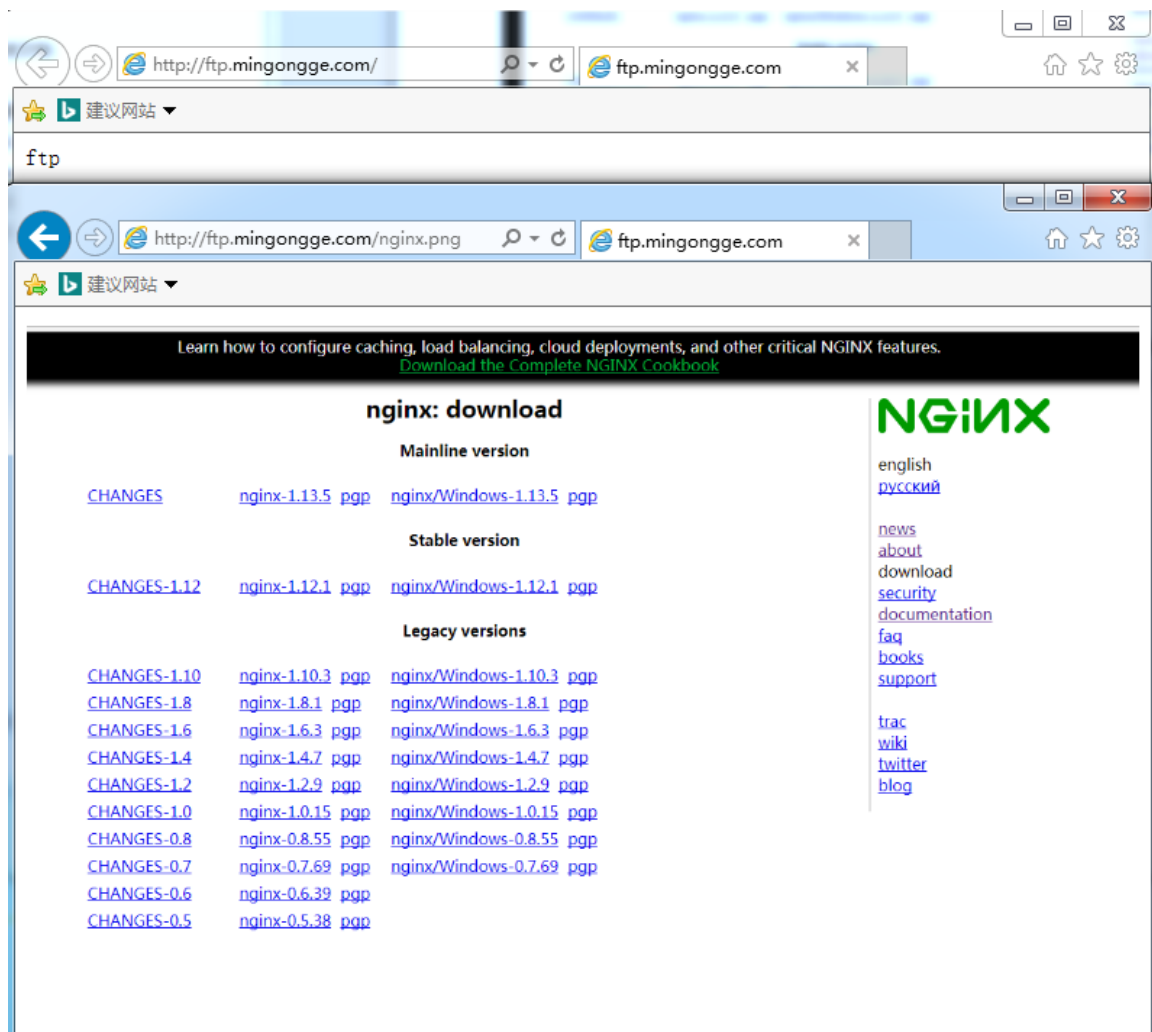
    location / {
        root    /www/ftp;
    }
}

```

上传图片进行测试



通过前端反向代理直接访问域名来读取图，来实现应用程序调用图片的功能在其它前台页面展示的功能



八：数据库安装与主从同步配置

添加用户

```
groupadd mysql
useradd -g mysql mysql -s /sbin/nologin -M
yum -y install libaio libaio-devel
```

下载二进制软件包并解压

```
wget
http://mirrors.sohu.com/mysql/MySQL-5.7/mysql-5.7.17-linux-glibc2.5-x86_64.tar.gz
[root@mysql-m ~]# tar xzf mysql-5.7.17-linux-glibc2.5-x86_64.tar.gz -C /usr/local/
[root@mysql-m ~]# ln -s /usr/local/mysql-5.7.17-linux-glibc2.5-x86_64
/usr/local/mysql
[root@mysql-m mysql]# mkdir /data
[root@mysql-m mysql]# chown -R mysql:mysql /data
```

初始化数据库

```
[root@mysql-m ~]# /usr/local/mysql/bin/mysqld --defaults-file=/etc/my.cnf
--initialize --user=mysql
2017-09-23T09:14:16.724707Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is
deprecated. Please use --explicit_defaults_for_timestamp server option (see
documentation for more details).
2017-09-23T09:14:17.698204Z 0 [Warning] InnoDB: New log files created, LSN=45790
2017-09-23T09:14:17.888059Z 0 [Warning] InnoDB: Creating foreign key constraint
system tables.
2017-09-23T09:14:17.977944Z 0 [Warning] No existing UUID has been found, so we assume
that this is the first time that this server has been started. Generating a new UUID:
93b7f20c-a03f-11e7-b91d-000c29d812ec.
2017-09-23T09:14:17.988894Z 0 [Warning] Gtid table is not ready to be used. Table
'mysql.gtid_executed' cannot be opened.
2017-09-23T09:14:17.995157Z 1 [Note] A temporary password is generated for
root@localhost: csnbztNif0-6
```

```
[root@mysql-m ~]# cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
[root@mysql-m ~]# chmod +x /etc/init.d/mysqld
[root@mysql-m ~]# /etc/init.d/mysqld start
Starting MySQL. SUCCESS!
[root@mysql-m ~]# lsof -i :3306
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
mysqld 1639 mysql 20u IPv6 15683 0t0 TCP *:mysql (LISTEN)
[root@mysql-m ~]# ln -s /data/mysql.sock /tmp/
```

默认启动是通过/tmp/mysql.sock 文件来连接 mysql

```
[root@mysql-m ~]#mysql -uroot -p csnbztNif0-6
```

```
mysql> alter user 'root'@'localhost' identified by '123456';
```

#####以上配置从库同样#####

```
[root@mysql-s ~]# cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
```

```
[root@mysql-s ~]# chmod +x /etc/init.d/mysqld
```

```
[root@mysql-s ~]# /etc/init.d/mysqld start
```

Starting MySQL. SUCCESS!

```
[root@mysql-s ~]# lsof -i :3306
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
---------	-----	------	----	------	--------	----------	------	------

mysqld	1636	mysql	18u	IPv6	15618	0t0	TCP	*:mysql (LISTEN)
--------	------	-------	-----	------	-------	-----	-----	------------------

```
[root@mysql-m ~]# ln -s /data/mysql.sock /tmp/
```

默认启动是通过/tmp/mysql.sock 文件来连接 mysql

```
[root@mysql-m ~]#mysql -uroot -p csbtNf0-6
```

```
mysql> alter user 'root'@'localhost' identified by '123456';
```

1、配置主从同步

```
[root@mysql-m ~]# vim /etc/my.cnf
```

```
[mysqld]
```

```
basedir = /usr/local/mysql/
```

```
datadir = /data
```

```
socket = /data/mysql.sock
```

```
server-id = 1
```

```
log-bin = /data/mysql-bin
```

```
relay-log = /data/master.relay-bin
```

```
relay-log-info-file = /data/master.relay-log.info
```

```
user=mysql
```

```
# Disabling symbolic-links is recommended to prevent assorted security risks
```

```
symbolic-links=0
```

```
replicate-ignore-db=mysql
```

```
[mysqld_safe]
```

```
log-error=/var/log/mysqld.log
```

```
pid-file=/var/run/mysqld/mysqld.pid
```

```
[root@mysql-s ~]# vim /etc/my.cnf
```

```
[mysqld]
```

```
basedir=/usr/local/mysql
```

```
datadir=/data
```

```
socket=/data/mysql.sock
```

```
user=mysql
```

```
# Disabling symbolic-links is recommended to prevent assorted security risks
```

```
symbolic-links=0
```

```
server-id = 2
#log-bin = /data/mysql-bin
relay-log = /data/slave.relay-bin
relay-log-info-file = /data/slave.relay-log.info
```

```
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

登陆主库配置同步权限与用户

```
mysql> grant replication slave on *.* to rep@'10.0.0.0.%' identified by'123456';
Query OK, 0 rows affected, 1 warning (0.04 sec)
mysql> flush privileges;
Query OK, 0 rows affected (0.04 sec)
```

主从服务器重启数据库服务，完全备份主库

```
[root@mysql-m ~]# mysqldump -uroot -p123456 -A -B >/root/master.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
[root@mysql-m ~]# ll /root/master.sql
-rw-r--r-- 1 root root 776313 Sep 23 18:21 /root/master.sql
[root@mysql-m ~]# mysql -uroot -p123456 -e "show master status"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+-----+-----+-----+-----+
| File                | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000001    | 154      |               |                   |                   |
+-----+-----+-----+-----+-----+
[root@mysql-m ~]#
```

将全备文件推送到从库执行全备恢复

```
[root@mysql-m ~]# scp /root/master.sql root@10.0.0.10:/root/
The authenticity of host '10.0.0.10 (10.0.0.10)' can't be established.
RSA key fingerprint is 51:7e:92:bb:01:4c:17:7b:60:c1:56:0b:c7:66:80:0e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.10' (RSA) to the list of known hosts.
root@10.0.0.10's password:
master.sql
100% 758KB 758.1KB/s 00:00
[root@mysql-m ~]#
```

```
[root@mysql-s ~]# mysql -uroot -p123456 </root/master.sql
[root@mysql-s ~]# mysql -uroot -p123456
mysql> change master to
-> master_host='10.0.0.11',
-> master_port=3306,
-> master_user='rep',
-> master_password='123456',
-> master_log_file='mysql-bin.000001',
-> master_log_pos=154;
Query OK, 0 rows affected, 2 warnings (0.04 sec)
```

```
mysql> start slave;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
        Master_Host: 10.0.0.11
        Master_User: rep
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: mysql-bin.000001
    Read_Master_Log_Pos: 313
        Relay_Log_File: slave.000002
        Relay_Log_Pos: 479
    Relay_Master_Log_File: mysql-bin.000001
      Slave_IO_Running: Yes
     Slave_SQL_Running: Yes
        Replicate_Do_DB:
    Replicate_Ignore_DB:
        Replicate_Do_Table:
    Replicate_Ignore_Table:
    Replicate_Wild_Do_Table:
    Replicate_Wild_Ignore_Table:
          Last_Errno: 0
          Last_Error:
        Skip_Counter: 0
     Exec_Master_Log_Pos: 313
        Relay_Log_Space: 676
        Until_Condition: None
        Until_Log_File:
        Until_Log_Pos: 0
    Master_SSL_Allowed: No
    Master_SSL_CA_File:
    Master_SSL_CA_Path:
      Master_SSL_Cert:
    Master_SSL_Cipher:
      Master_SSL_Key:
    Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
          Last_IO_Errno: 0
          Last_IO_Error:
         Last_SQL_Errno: 0
         Last_SQL_Error:
    Replicate_Ignore_Server_Ids:
      Master_Server_Id: 1
```

查看主从同步情况


```
[root@mysql-m ~]# mysql -uroot -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.7.18-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test |
+-----+
5 rows in set (0.02 sec)
```

```
Replicate_Ignore_Server_Ids:
  Master_Server_Id: 1
  Master_UUID: 93b7f20c-a03f-11e7-b91d-000c29d812ec
  Master_Info_File: /data/master.info
  SQL_Delay: 0
  SQL_Remaining_Delay: NULL
  Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
  Master_Retry_Count: 86400
  Master_Bind:
  Last_IO_Error_Timestamp:
  Last_SQL_Error_Timestamp:
  Master_SSL_Crl:
  Master_SSL_Crlpath:
  Retrieved_Gtid_Set:
  Executed_Gtid_Set:
  Auto_Position: 0
  Replicate_Rewrite_DB:
  Channel_Name:
  Master_TLS_Version:
1 row in set (0.00 sec)
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test |
+-----+
5 rows in set (0.02 sec)

mysql> █
```

实际生产环境需要做读写分离，方法很多，可以通过程序指定读写库，也可以通过第三方的工具来实现，看实际生产需求与个人习惯

九：Redis 与 zabbix 的安装与配置

可参考之前的文章



十：全网数据备份规划

数据备份规划如下

数据库

从库开启 binlog，通过全备与增量备份来进行，每天 00:00 全备一次，每天 12:00 增量备份一次，初期数据量不大的情况，后期根据数据量进行调整

本地备份目录

/server/backup/mysql/data

/server/backup/mysql/binlog

FTP 数据

实时同步上传数据，防止服务故障，可以快速切换

/www/ftp--> /www/ftp

其它服务器数据实行实时同步

比如说配置文件备份、代码包备份

本地备份目录

/server/backup/app

/server/backup/sjapp

```
/server/backup/admin  
/server/backup/shangjia  
/server/backup/web
```

所有数据本地保留一周，远端备份服务器保留 30 天

十一：数据库备份配置

```
mkdir /server/backup/mysql/data  
mkdir /server/backup/mysql/binlog  
mkdir /server/scripts/
```

```
[root@mysql-1 ~]# vim fullbak.sh  
#!/bin/sh  
  
#####  
# create by mingongge at hefei 2017-09-29      #  
# this scripts is for backup the data of mysql  #  
# QQ:343728299                                  #  
# 公众号: youkanyouxiao(友侃有笑)              #  
# QQ群:580703149                                #  
#####  
  
USER=root  
PWD="123456"  
MYSQL_CMD=/usr/local/mysql/bin/mysqldump  
BAK_DIR=/server/backup  
DATE=`date +%F`  
ip=`grep 'IPADDR' /etc/sysconfig/network-scripts/ifcfg-eth0|awk -F "=" '{print $2}'`  
  
[ ! -d $BAK_DIR ] && mkdir $BAK_DIR  
cd $BAK_DIR  
DB_NAME=`$MYSQL_CMD -u${USER} -p${PWD} -e "show databases;"|sed '1,5d'`  
for dbname in $DB_NAME  
do  
    $MYSQL_CMD -u$USER -p$PWD -b $dbname |gzip > $dbname_${ip}_${DATE}.sql.gz  
    if [ $# -eq 0 ];then  
        echo "$dbname $DATE backup successful" >>$BAK_DIR/bakmysql.log  
    else  
        echo "$dbname $DATE backup failed" >>$BAK_DIR/bakmysql.log  
    fi  
done  
  
#del the backup file 7 days ago  
find $BAK_DIR -type f -name "*.sql.gz" -mtime +7 |xargs rm -rf  
~
```

```

[root@mysql-1 ~]# vim binlog_bak.sh

#!/bin/sh
#####
# create by mingongge at hefei 2017-09-29      #
# this scripts is for backup the data of mysql  #
# QQ:343728299                                #
# 公众号: youkanyouxiao(友侃有笑)              #
# QQ群:580703149                              #
#####
USER=root
PWD="123456"
MYSQL_CMD=/usr/local/mysql/bin/mysqladmin
BAK_DIR=/server/backup
DATE=`date +%F`
BIN_DIR=/data
BIN_FILE=/data/m1-bin.index
logfile=/server/backup/bak_binlog.log

[ ! -d $BAK_DIR ] && mkdir $BAK_DIR

#生成新的binlog文件
$MYSQL_CMD -u$USER -p$PWD flush-logs

#统计文件数量
count=`wc -l $BIN_FILE|awk '{print $1}'`
NEXT=0

#通过for循环来确定文件是不是最新的，因为最新的文件是正在使用中的
for file in `cat $BIN_FILE`
do
    #将./mysql-bin.000005前面的./去掉
    base=`basename $file`
    NEXT=`expr $NEXT + 1`
    if [ $NEXT -eq $count ]
    then
        echo $base skip!!!! >>$logfile
    else
        #如果不是最新的文件，那么就是我们需要备份的目标文件
        then
            echo $base skip!!!! >>$logfile
        else
            #如果不是最新的文件，那么就是我们需要备份的目标文件
            dfile=$BAK_DIR/$base
            #判断目标文件是否存在于备份目录
            if (test -e $dfile)
            then
                echo $base is exist!!!! >>$logfile
            else
                cp $BIN_DIR/$base $BAK_DIR
                echo $base copying successfully >>$logfile
            fi
        fi
    done
    echo $DATE backup successfully >>$logfile

#del the backup file 7 days ago
find $BAK_DIR -type f -name "*.sql.gz" -mtime +7 |xargs rm -rf

```

脚本写好了，就需要写个定时任务定时执行，然后手工检查
cat>>/var/spool/cron/root<<EOF

```
00 00 * * * /bin/sh /server/scripts/fullbak.sh >/dev/null 2>&1
00 12 * * * /bin/sh /server/scripts/binlog_bak.sh >/dev/null 2>&1
EOF
```

十二：静态资源服务器的数据同步配置

同步服务器创建对等目录

```
mkdir /www/ftp
```

安装 rsync 服务

```
Yum install rsync -y
```

配置 rsync 配置文件

```
vim /etc/rsyncd.conf
##rsync.conf config start
uid = rsync
gid = rsync
use chroot = no
max connections = 200
timeout = 100
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsync.lock
log file = /var/log/rsyncd.log
[backup]
path = /www/ftp
ignore errors
read only = false
list = false
hosts allow = 10.0.0.8
auth users = rsync_backup
secrets file = /etc/rsync.password
##rsync config end

useradd rsync -s /sbin/nologin -M
chown -R rsync:rsync /www/ftp
echo "rsync_backup:rsync.conf">/etc/rsync.password
chmod 600 /etc/rsync.password
rsync -daemon
[root@resources ~]# netstat -lntup|grep rsync
tcp    0 0 0.0.0.0:873    0.0.0.0:*    LISTEN  26280/rsync
```

客户端配置密码

```
echo "rsync.conf">/etc/rsync.password
```

```
chmod 600 /etc/rsync.password
```

测试手工同步数据是否成功

```
[root@resources ~]# rsync -avzP /etc/hosts rsync_backup@10.0.0.11::backup
--password-file=/etc/rsync.password
sending incremental file list
hosts
  473 100%    0.00kB/s    0:00:00 (xfer#1, to-check=0/1)
sent 314 bytes received 27 bytes  682.00 bytes/sec
total size is 473 speedup is 1.39
[root@ftp-bakserver ~]# ll /www/ftp
total 4
-rw-r--r-- 1 rsync rsync 473 May 26 13:24 hosts
```

客户端安装 sersync 服务

```
tar zxvf sersync2.5.4_64bit_binary_stable_final.tar.gz -C /usr/local/
[root@resources ~]# cd /usr/local/
[root@resources local]# mv GNU-Linux-x86 sersync
```

规范文件目录

```
[root@resources sersync]# mkdir -p conf bin logs
```

拷贝配置文件并修改

```
[root@resources sersync]# mv confxml.xml conf/
[root@resources sersync]# mv sersync2 bin/sersync
```

修改配置文件内容 (confxml.xml)

1、修改 24-28 行

```
<localpath watch="/opt/tongbu">
  <remote ip="10.0.0.11" name="tongbu1"/>
  <!--<remote ip="192.168.8.39" name="tongbu"/>-->注释内容
  <!--<remote ip="192.168.8.40" name="tongbu"/>-->注释内容
</localpath>
```

修改后的内容为

```
<localpath watch="/www/ftp"> 本地数据的路径
  <remote ip="10.0.0.11" name="backup"/>远端 IP 与模块名称
</localpath>
<!##### -->注释内容
```

2、修改 31-34 行内容——认证

```
<commonParams params="-artuz"/>
<auth start="false" users="root" passwordfile="/etc/rsync.pass"/>
<userDefinedPort start="false" port="874"/><!-- port=874 -->
<timeout start="false" time="100"/><!-- timeout=100 -->
<ssh start="false"/>
```

修改后的内容为

```
<commonParams params="-aruz"/>
<auth start="true" users="rsync_backup" passwordfile="/etc/rsync.password"/>
<userDefinedPort start="false" port="874"/><!-- port=874 -->
<timeout start="true" time="100"/><!-- timeout=100 -->
<ssh start="false"/>
```

3、修改 36-37 行

```
<failLog path="/tmp/rsync_fail_log.sh" timeToExecute="60"/><!--default
every 60mins execute once-->
```

修改成我们刚刚创建好的 logs 目录

```
<failLog path="/usr/local/sersync/logs/rsync_fail_log.sh" timeToExecut
e="60"/><!--default every 60mins execute once-->
```

配置环境变量

```
echo 'export PATH=$PATH:/usr/local/sersync/bin'>>/etc/profile
[root@resources sersync]# source /etc/profile
[root@resources sersync]# which sersync
/usr/local/sersync/bin/sersync
```

启动服务

```
[root@resources sersync]# sersync -r -d -o /usr/local/sersync/conf/confxml.xml
```

测试实时同步

```
[root@resources opt]# cd backup/
[root@resources backup]# ll
total 0
[root@ftp-bakserver ~]# cd /www/ftp
[root@ftp-bakserver backup]# ll
total 0
[root@resources ftp]# for i in `seq 10`;do mkdir -p /opt/backup/$i;done
[root@resources ftp]# ll
1 5 10 6 2 7 3 8 4 9
[root@ftp-bakserver ftp]# ls
1 5 10 6 2 7 3 8 4 9
可以同步
```

十三：同步全网备份数据至备份服务器

备份服务器配置

```
[user-app]
path = /mnt/backup/user-app
ignore errors
read only = false
```

```
list = false
hosts allow = 10.0.0.4
auth users = rsync_backup
secrets file = /etc/rsync.password
```

```
[sj-app]
path = /mnt/backup/sj-app
ignore errors
read only = false
list = false
hosts allow = 10.0.0.5
auth users = rsync_backup
secrets file = /etc/rsync.password
```

```
[admin]
path = /mnt/backup/admin
ignore errors
read only = false
list = false
hosts allow = 10.0.0.6
auth users = rsync_backup
secrets file = /etc/rsync.password
```

```
[shangjia]
path = /mnt/backup/shangjia
ignore errors
read only = false
list = false
hosts allow = 10.0.0.7
auth users = rsync_backup
secrets file = /etc/rsync.password
```

```
[web]
path = /mnt/backup/web
ignore errors
read only = false
list = false
hosts allow = 10.0.0.8
auth users = rsync_backup
secrets file = /etc/rsync.password
```

其它同步客户端安装与配置 sersync 按上述操作进行
备份脚本基本都是打包备份或直接拷贝文件到本地备份目录

友侃有笑 | 分享干货的公众号



公众号:youkanyouxiao



长按识别二维码加关注

扫码关注公众号