**诺安诺泰新健康云平台部署手册**

**（version 1.0）**

在本文中你将知道

1. 如何部署诺安诺泰新健康云平台

你需要知道

1. 了解linux系统的基本设置，
2. 了解nginx
3. 了解tomcat
4. 了解oracle

**版本控制**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 时间 | 版本 | 撰写人 | 备注 |
| 1 | 2016年10月28日 | 0.1 | 吴强 | 文本创建 |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

目录

[章节1. 平台规划 4](#_Toc433856146)

[1.1网站构架 4](#_Toc433856147)

[1.2 主机地址规划 4](#_Toc433856148)

[1.3 所需软件和功能介绍 4](#_Toc433856149)

[章节2. 初始化系统 4](#_Toc433856150)

[概述 4](#_Toc433856151)

[启用网络并配置网络地址 5](#_Toc433856152)

[关闭不必要的服务 5](#_Toc433856153)

[开放需要的端口 6](#_Toc433856154)

[关闭SElinux的主动模式（可以跳过） 7](#_Toc433856155)

[关闭Ctrl+Alt+Del自动重启的操作关联 7](#_Toc433856156)

[修改主机host文件 8](#_Toc433856157)

[配置ntp自动更新时间 8](#_Toc433856158)

[修改主机内核参数 9](#_Toc433856159)

[修改系统打开文件的最大数 9](#_Toc433856160)

[创建Web站点运行账户 9](#_Toc433856161)

[章节3. 组件 10](#_Toc433856162)

[章节4. 所需依赖 10](#_Toc433856163)

[章节5. 安装运行环境 11](#_Toc433856164)

[安装nginx 11](#_Toc433856165)

[安装python 的web的运行环境和组件 18](#_Toc433856166)

[安装pip 18](#_Toc433856167)

[安装gunicorn 19](#_Toc433856168)

[安装Flask 19](#_Toc433856169)

[安装Redis模块 20](#_Toc433856170)

[设置为开机启动。 20](#_Toc433856171)

[安装redis服务器 20](#_Toc433856172)

[安装jemalloc内存管理库（依赖） 20](#_Toc433856173)

[安装redis服务 20](#_Toc433856174)

[章节6. 部署Web程序 21](#_Toc433856175)

[释放web文件 21](#_Toc433856176)

[启动web服务 22](#_Toc433856177)

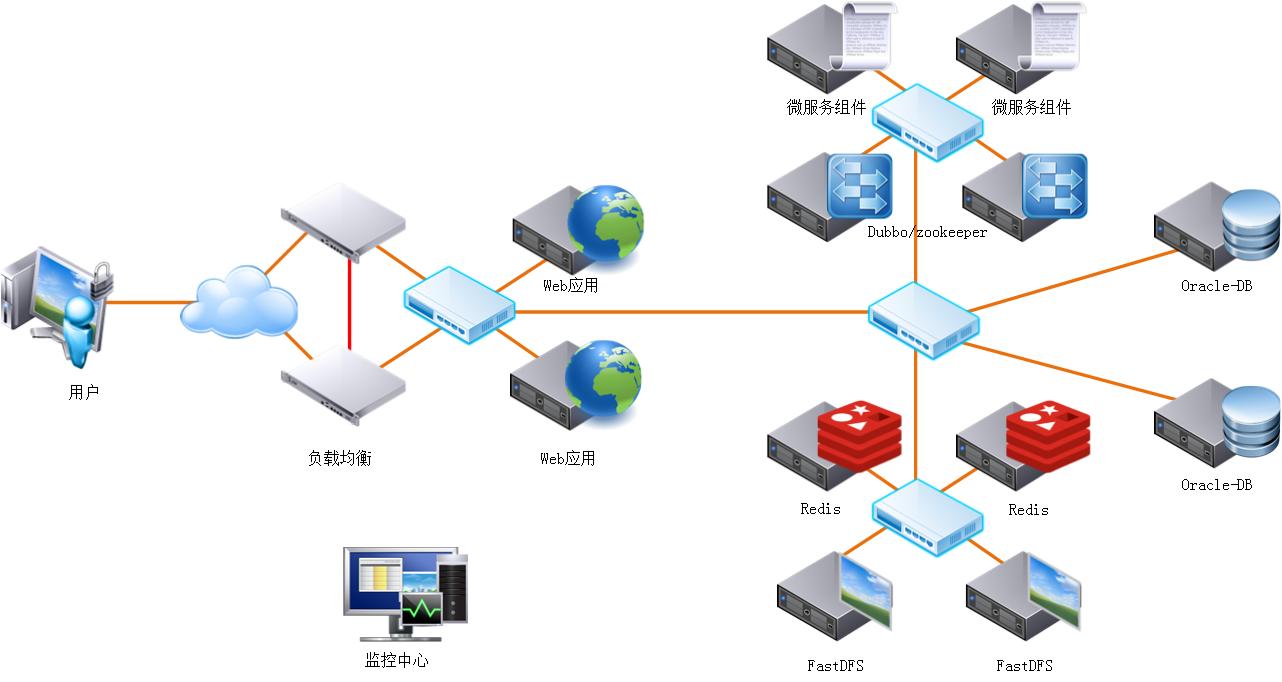
[激活数据库 22](#_Toc433856178)

[章节7. 启动 22](#_Toc433856179)

[章节8. 问与答 23](#_Toc433856180)

# 平台规划

## 网站构架

****

如上图，本系统由两台物理机构建。nginx作为反向代理，将请求分发到后端的tomcat上去，每一台物理机上启用两个Tomcat实例，

## 主机地址规划

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 主机名 | 操作系统 | 功能 | IP地址 | 网关 | 备注 |
| LB0 | CentOS7.1 | 负载均衡（主） | 192.168.88.74 |  |  |
| Web0 | CentOS7.1 |  | 192.168.88.72 |  | service |
| Web1 | CentOS7.1 |  | 192.168.88.73 |  | Zookeeper, service |
| Web0 | CentOS7.1 | web应用 | 192.168.88.77 |  |  |
| Redis0 | CentOS7.1 | Redis服务（主） | 192.168.88.75 |  |  |
| FastDFS0 | CentOS7.1 | FastDFS文件存储0 | 192.168.88.240 |  | Tracker, storage,  nginx |
| DB0 |  | 数据库0 | 192.168.88.71 |  |  |

## 所需软件和功能介绍

|  |  |  |  |
| --- | --- | --- | --- |
| 软件包名称 | 版本 | 功能 | 备注 |
| CentOS | 7.1 | Linux OS | 最小化安装 |
| Nginx | 1.8.1 | Web服务器 |  |
| JDK | 1.7u79 | Java运行环境 | 使用oracle的jdk |
| Tomcat | 7.0.72 | Web中间件 |  |
| Nrpe | 1.21 | 远程监控插件 | Nagios远程监控程序 |
| Keepalived | 1.20 | 高可用组件 |  |
| Oracle | 11gr2 | 数据库软件 | 企业版 |
| fastDFS |  | 文件存储 |  |

## 目录说明

|  |  |  |  |
| --- | --- | --- | --- |
| 主机 | 目录 | 功能 | 备注 |
| WebX/ServerX | /opt | 程序工作根目录 |  |
| WebX | /opt/webapp | 前台页面根目录 | 推荐不使用war包 |
| WebX | /opt/webserver | 后端服务根目录 | 推荐使用war包 |
| ServerX | /opt/Server | 其它组件 |  |
| RedisX | /opt/Server/Redis | Redis数据备份目录 | 文件名为:NewHealth.dump |

## 系统初始帐户密码

|  |  |  |  |
| --- | --- | --- | --- |
| 主机 | 用户 | 密码 | 备注 |
| All | root | oracle |  |
| WebX | root | Admin@123456 |  |
| WebX | Webadmin |  | Web管理员帐户 |
| **88.77** | root | 123456 |  |
|  |  |  |  |

# 始化系统

## 概述

下面所有的系统初始化建议都是针对CentOS7.1最小化系统。如果使用其它相关系统，请适当修改！！

## 启用网络并配置网络地址

Shell># vi /etc/sysconfig/network-scripts/ifcfg-eth0

|  |  |
| --- | --- |
| 原始 | DEVICE=eth0  HWADDR=00:0C:29:C9:F2:34  TYPE=Ethernet  UUID=f7710b52-b28d-4435-8ae3-8c095a094677  ONBOOT=no  NM\_CONTROLLED=yes  BOOTPROTO=dhcp |
| 修改为 | DEVICE=eth0  HWADDR=00:0C:29:C9:F2:34  TYPE=Ethernet  UUID=f7710b52-b28d-4435-8ae3-8c095a094677  **ONBOOT=yes**  **NM\_CONTROLLED=no**  **IPADDR=10.10.10.4**  **NETMASK=255.255.255.0**  **GATEWAY=10.10.10.254**  **DNS1=8.8.8.8** |

重启网络以应用修改

Shell># service network restart

验证

Shell>#ip addr

输出如下

|  |
| --- |
| 1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  inet 127.0.0.1/8 scope host lo  inet6 ::1/128 scope host  valid\_lft forever preferred\_lft forever  2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000  link/ether 00:0c:29:c9:f2:34 brd ff:ff:ff:ff:ff:ff  inet 10.10.10.4/24 brd 10.10.10.255 scope global eth0  inet6 fe80::20c:29ff:fec9:f234/64 scope link  valid\_lft forever preferred\_lft forever |

## 关闭不必要的服务

在系统安装完成后，有很多默认服务我们并不需要，所以我可以使用如下命令关闭

## 开放需要的端口

## 关闭SElinux的主动模式（可以跳过）

直接在系统中关闭

|  |
| --- |
| Shell># setenforce 0 |

保存配置，保证重启后依然是关闭状态的

Shell># vi /etc/selinux/conf

修改如下字段

|  |  |
| --- | --- |
| 原始 | **SELINUX= enforcing**  SELINUXTYPE=targeted |
| 修改为 | **SELINUX=permissive**  SELINUXTYPE=targeted |

验证,输出一下类似内容表示成功

|  |
| --- |
| Shell># getenforce  Permissive |

## 关闭Ctrl+Alt+Del自动重启的操作关联

验证

在CLI中单击Ctrl+Alt+Del，系统没有任何反应，表示成功。

## 修改主机host文件

在hosts文件中添加IP和主机名称（修改位置已经加粗标红）

Shell># vi /etc/host

|  |
| --- |
| 127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  **10.10.10.4web1.test** |

## 配置ntp自动更新时间

安装ntpdate程序

|  |
| --- |
| Shell># yum -y install ntpdate |

使用crontab将系统设定为每天0:00与NTP服务器同步时间

Shell># crontab -u ntp -e

|  |
| --- |
| 0 0 \* \* \* ntpdate your\_ntp\_server\_domain\_name or your\_ntp\_server\_ip |

保存退出，

为了我们步骤继续下去，我们执行一次同步

Shell># ntpdate time.windows.com

如果成功会有如下提示

|  |
| --- |
| Shell># ntpdate time.windows.com  18 Jul 15:34:31 ntpdate[8655]: step time server time.windows.com offset -29.884509 sec |

将系统时间同步到硬件

Shell># hwclock -w

验证

|  |
| --- |
| Shell># date  Sat Jul 18 09:00:47 CST 2015 |

如果输出的时间没有问题，表示成功

## 修改主机内核参数

Shell># vi /etc/sysctl.conf

|  |
| --- |
| net.ipv4.ip\_forward = 0  net.ipv4.conf.default.rp\_filter = 1  net.ipv4.conf.default.accept\_source\_route = 0  kernel.sysrq = 0  kernel.core\_uses\_pid = 1  kernel.printk\_ratelimit = 30  kernel.printk\_ratelimit\_burst = 200  net.ipv4.tcp\_max\_tw\_buckets = 256000  net.ipv4.ip\_local\_port\_range = 1500 65535  net.core.somaxconn = 262144  net.ipv4.tcp\_syncookies = 1  net.ipv4.tcp\_max\_tw\_buckets = 28000  net.ipv4.tcp\_sack = 1  net.ipv4.tcp\_timestamps = 1  net.ipv4.tcp\_tw\_reuse = 1  net.ipv4.tcp\_tw\_recycle = 0  net.ipv4.tcp\_fin\_timeout = 15  net.ipv4.tcp\_synack\_retries = 1  net.ipv4.tcp\_syn\_retries = 1  net.ipv4.tcp\_retries1 = 3  net.ipv4.tcp\_retries2 = 5  net.ipv4.tcp\_keepalive\_time = 1200  net.ipv4.tcp\_max\_syn\_backlog = 10240  net.ipv4.tcp\_rmem = 4096 8192 4194304  net.ipv4.tcp\_wmem = 4096 8192 4194304  net.ipv4.tcp\_max\_orphans = 131070  net.core.optmem\_max = 81920  net.core.wmem\_default = 8388608  net.core.wmem\_max = 16777216  net.core.rmem\_default = 8388608  net.core.rmem\_max = 16777216  net.core.somaxconn = 262144  net.core.netdev\_max\_backlog = 862144  fs.file-max = 65535  fs.aio-max-nr = 65536  kernel.msgmnb = 65536  kernel.msgmax = 65536  kernel.shmmax = 68719476736  kernel.shmall = 4294967296 |

使其修改生效

Shell># sysctl -p

## 修改系统打开文件的最大数

修改系统的最大文件打开数，默认是1024

查看

shell># ulimit -n

1024

但是这个数量级，根本满足不了我们高并发应用对文件打开的需要，

可以使用如下命令临时修改

shell># ulimit -n 65535

通过修改文件固化

shell># vi /etc/security/limits.conf

添加内容如下

|  |
| --- |
| \* soft nofile 65535  \* hard nofile 65535 |

## 创建Web站点运行账户

创建www账户

Shell># useraddd sysadmin

添加密码,成功后提示如下

Shell># passwd sysadmin

|  |
| --- |
| Changing password for user www.  New password:  Retype new password:  passwd: all authentication tokens updated successfully. |

验证

Shell>id Goldfish

|  |
| --- |
| uid=502(sysadmin) gid=502(sysadmin) groups=502(sysadmin) |

## 双网卡绑定

在平台中，我们要考虑到网络适配器或者交换机的硬件损坏，或者线路故障导致的服务器离线。我们可以使用双网卡绑定来实现网络适配器或者线路的冗余，或者提升网络吞吐量。

创建虚拟网卡bond0的配置文件

Shell>#vi /etc/sysconfig/network-scripts/ifcfg-bond0

|  |
| --- |
| DEVICE=bond0  BOOTPROTO=none  ONBOOT=yes  USERCTL=no  IPADDR=192.168.37.110  NETMASK=255.255.255.0  GATEWAY=192.168.37.1  DNS1=127.0.0.1 |

修改eth0/eth1的网卡配置文件

Shell>#vi /etc/sysconfig/network-scripts/ifcfg-eth0

|  |
| --- |
| DEVICE=eth0  TYPE=Ethernet  **ONBOOT=yes**  **NM\_CONTROLLED=no**  **BOOTPROTO=none**  **MASTER=bond0**  **SLAVE=yes** |

Shell>#vi /etc/sysconfig/network-scripts/ifcfg-eth1

|  |
| --- |
| DEVICE=eth1  TYPE=Ethernet  **ONBOOT=yes**  **NM\_CONTROLLED=no**  **BOOTPROTO=none**  **MASTER=bond0**  **SLAVE=yes** |

修改/etc/modprobe.conf 文件

Shell># vi /etc/modprobe.conf

|  |
| --- |
| alias bond0 bonding  options bond0 miimon=100 mode=0 |

这里面的选项说明如下

|  |
| --- |
| miimon 是链路监测的时间间隔单位是毫秒，miimon=100的意思就是，每100毫秒检测网卡和交换机之间是否连通，如不通则使用另外的链路。  mode=0 表示负载均衡方式，两块网卡都工作，需要交换机作支持。  mode=1 表示冗余方式，网卡只有一个工作，一个出问题启用另外的。  mode=6 表示负载均衡方式，两块网卡都工作，不需要交换机作支持。 |

重启网络

Shell># service network restart

验证结果

Shell># ifconfig

|  |
| --- |
| bond0 Link encap:Ethernet HWaddr 00:0C:29:DC:63:C7  inet addr:192.168.37.110 Bcast:192.168.37.255 Mask:255.255.255.0  inet6 addr: fe80::20c:29ff:fedc:63c7/64 Scope:Link  UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1  RX packets:2426 errors:0 dropped:0 overruns:0 frame:0  TX packets:1728 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:0  RX bytes:200041 (195.3 KiB) TX bytes:176604 (172.4 KiB)  eth0 Link encap:Ethernet HWaddr 00:0C:29:DC:63:C7  UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1  RX packets:2407 errors:0 dropped:0 overruns:0 frame:0  TX packets:1268 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:1000  RX bytes:198586 (193.9 KiB) TX bytes:139047 (135.7 KiB)  eth1 Link encap:Ethernet HWaddr 00:0C:29:DC:63:C7  UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1  RX packets:19 errors:0 dropped:0 overruns:0 frame:0  TX packets:460 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:1000  RX bytes:1455 (1.4 KiB) TX bytes:37557 (36.6 KiB)  lo Link encap:Local Loopback  inet addr:127.0.0.1 Mask:255.0.0.0  inet6 addr: ::1/128 Scope:Host  UP LOOPBACK RUNNING MTU:65536 Metric:1  RX packets:4 errors:0 dropped:0 overruns:0 frame:0  TX packets:4 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:0  RX bytes:340 (340.0 b) TX bytes:340 (340.0 b) |

查看bond0的工作状态

Shell>#more /proc/net/bonding/bond0

|  |
| --- |
| Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)  Bonding Mode: load balancing (round-robin)  MII Status: up  MII Polling Interval (ms): 100  Up Delay (ms): 0  Down Delay (ms): 0  Slave Interface: eth0  MII Status: up  Speed: 1000 Mbps  Duplex: full  Link Failure Count: 0  Permanent HW addr: 00:0c:29:dc:63:c7  Slave queue ID: 0  Slave Interface: eth1  MII Status: up  Speed: 1000 Mbps  Duplex: full  Link Failure Count: 0  Permanent HW addr: 00:0c:29:dc:63:d1  Slave queue ID: 0 |

至此我们的网卡绑定就没问题了

# 组件

* nginx
* tomcat
* keepalived
* oracle

# 所需依赖

基于rpm的系统

* 编译源代码需要
  + gcc
  + make
* Nginx依赖
  + pcre-devel
  + openssl-devel
* keepalived
  + net-snmp-devel
  + net-snmp
* 辅助工具
  + wget

# 安装运行环境

## Nginx(LB0、LB1)

目的：nginx承担的角色是负载均衡和分离动态和静态内容。WAF，缓存，后端健康检查

软件官方地址

Nginxhttp://nginx.org

添加用户

Shell>useradd -r nginx

Shell>groupadd nginx**（可选执行）**

验证

|  |  |
| --- | --- |
| Shell >id nginx  出现如下提示，标识用户创建成功   |  | | --- | | uid=500(nginx) gid=500(nginx) group）s=500(nginx) | |

创建文件夹

Shell>mkdir /var/cache/nginx

获取安装包，我们这里使用nginx-1.8的稳定版本

Shell># wget <http://nginx.org/download/nginx-1.8.1.tar.gz>

Shell># wget https://codeload.github.com/yaoweibin/nginx\_upstream\_check\_module/zip/master

解压缩nginx软件包

Shell># tar xf nginx-1.8.1.tar.gz

进入nginx软件目录

Shell># cd nginx-1.8.1

编译软件, 编译nginx，参数如下(按照rpm包的习惯编译)

|  |
| --- |
| ./configure \  --prefix=/etc/nginx \  --sbin-path=/usr/sbin/nginx \  --conf-path=/etc/nginx/nginx.conf \  --error-log-path=/var/log/nginx/error.log \  --http-log-path=/var/log/nginx/access.log \  --pid-path=/var/run/nginx.pid \  --lock-path=/var/run/nginx.lock \  --http-client-body-temp-path=/var/cache/nginx/client\_temp \  --http-proxy-temp-path=/var/cache/nginx/proxy\_temp \  --http-fastcgi-temp-path=/var/cache/nginx/fastcgi\_temp \  --http-uwsgi-temp-path=/var/cache/nginx/uwsgi\_temp \  --http-scgi-temp-path=/var/cache/nginx/scgi\_temp \  --user=nginx \  --group=nginx \  --with-http\_ssl\_module \  --with-http\_realip\_module \  --with-http\_gunzip\_module \  --with-http\_gzip\_static\_module \  --with-http\_stub\_status\_module \  --with-file-aio \  --with-http\_spdy\_module |

安装

Shell >make && make install

修改配置文件（原始文件如下）

Shell > vi /etc/nginx/nginx.conf

|  |
| --- |
| worker\_processes 1;  events {  worker\_connections 1024;  }  http {  include mime.types;  default\_type application/octet-stream;  sendfile on;  keepalive\_timeout 65;  server {  listen 80;  server\_name localhost;  location / {  root html;  index index.html index.htm;  }  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  root html;  }  }  } |

我们为了方便管理进行管理文件分类（主文件）

Shell># vi /etc/nginx/nginx.conf

|  |
| --- |
| user nginx;  worker\_processes 8;  events {  use epoll;  worker\_connections 204800;  multi\_accept on;  }  http {  include mime.types;  default\_type application/octet-stream;  charset UTF-8;  fastcgi\_intercept\_errors on;  server\_names\_hash\_bucket\_size 128;  client\_header\_buffer\_size 4k;  large\_client\_header\_buffers 4 32k;  client\_max\_body\_size 300m;  tcp\_nodelay on;  server\_tokens off;  client\_body\_buffer\_size 512k;  proxy\_ignore\_client\_abort on;  proxy\_connect\_timeout 120;  proxy\_read\_timeout 60;  proxy\_send\_timeout 10;  proxy\_buffer\_size 32k;  proxy\_buffers 4 128k;  proxy\_busy\_buffers\_size 256k;  proxy\_temp\_file\_write\_size 128k;  proxy\_cache\_path /var/www/cache levels=1:2 keys\_zone=webcache:20m inactive=60m max\_size=300m;  proxy\_temp\_path /var/www/cache/tmp;  sendfile on;  tcp\_nopush on;  keepalive\_timeout 60;  gzip on;  gzip\_min\_length 1k;  gzip\_buffers 4 16k;  gzip\_comp\_level 2;  gzip\_types text/plain application/x-javascript text/css application/xml;  gzip\_vary on;  open\_file\_cache max=204800 inactive=20s;  open\_file\_cache\_min\_uses 2;  open\_file\_cache\_valid 30s;  #server\_tokens off  include ./conf.d/upstream.conf;  include ./conf.d/vhost.conf;  } |

修改扩展配置文件

Shell># vi /etc/nginx/conf.d/vhost.conf

|  |
| --- |
| server {  listen 80;  location ~\* /.+\.(gif|jpg|jpeg|png|css|js|ico)$ {  #proxy\_cache webcache;  #proxy\_cache\_valid 200 304 301 302 10d;  #proxy\_cache\_valid any 60m;  #proxy\_cache\_key $host$uri$is\_args$args;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forward-For $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_headers\_hash\_max\_size 51200;  proxy\_headers\_hash\_bucket\_size 6400;  proxy\_set\_header x-for $remote\_addr;  proxy\_set\_header x-server $host;  proxy\_set\_header x-agent $http\_user\_agent;  proxy\_pass http://cluster2;  break;  }    location ~\*/.+\.(html|jsp|do)$ {  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forward-For $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_headers\_hash\_max\_size 51200;  proxy\_headers\_hash\_bucket\_size 6400;  proxy\_set\_header x-for $remote\_addr;  proxy\_set\_header x-server $host;  proxy\_set\_header x-agent $http\_user\_agent;  proxy\_pass http://cluster1;  break;  }  location / {  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forward-For $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_headers\_hash\_max\_size 51200;  proxy\_headers\_hash\_bucket\_size 6400;  proxy\_set\_header x-for $remote\_addr;  proxy\_set\_header x-server $host;  proxy\_set\_header x-agent $http\_user\_agent;  proxy\_pass http://cluster1;  break;  }  } |

验证1

Shell >#netstat -antp

|  |
| --- |
| tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 6245/nginx |

验证2

Shell>#ps -aux | grep nginx

|  |
| --- |
| root 6245 0.0 0.0 45108 1220 ? Ss 23:46 0:00 nginx: master process ./nginx  nginx 6246 0.4 1.5 72248 28620 ? S 23:46 0:00 nginx: worker process  nginx 6247 0.3 1.5 72248 28620 ? S 23:46 0:00 nginx: worker process  nginx 6248 0.5 1.5 72248 28620 ? S 23:46 0:01 nginx: worker process  nginx 6249 0.5 1.5 72248 28600 ? S 23:46 0:01 nginx: worker process |

添加开机自启动，脚本如下

## 安装JDK(Web0、Web1、Server0、Server1)

使用tar包安装

创建目录

Shell># mkdir/usr/local/java

上传tar包，解压

Shell># tar xf jdk-7u79-linux-x64.tar.gz

设置全局变量

Shell># vi /etc/profile

|  |
| --- |
| JAVA\_HOME=/usr/local/java/jdk1.7.0\_79  CLASSPATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib  PATH=$JAVA\_HOME/bin:$PATH:$JAVA\_HOME/jre/bin  export PATH JAVA\_HOME CLASSPATH |

为使其生效我们可以使用如下命令激活

Shell># source /etc/profile

># source /etc/profile

验证

Shell># java -version

|  |
| --- |
| java version "1.7.0\_79"  Java(TM) SE Runtime Environment (build 1.7.0\_79-b15)  Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode) |

## 安装Nginx（Web0、Web1）

目的:处理网页中的静态资源

安装参考前文。

安装编译参数

|  |
| --- |
| ./configure \  --prefix=/etc/nginx \  --sbin-path=/usr/sbin/nginx \  --conf-path=/etc/nginx/nginx.conf \  --error-log-path=/var/log/nginx/error.log \  --http-log-path=/var/log/nginx/access.log \  --pid-path=/var/run/nginx.pid \  --lock-path=/var/run/nginx.lock \  --http-client-body-temp-path=/var/cache/nginx/client\_temp \  --http-fastcgi-temp-path=/var/cache/nginx/fastcgi\_temp \  --http-uwsgi-temp-path=/var/cache/nginx/uwsgi\_temp \  --http-scgi-temp-path=/var/cache/nginx/scgi\_temp \  --user=nginx \  --group=nginx \  --with-http\_realip\_module \  --with-http\_stub\_status\_module \  --with-file-aio |

配置文件

|  |
| --- |
| user nginx;  worker\_processes 2;  events {  worker\_connections 1024;  }  http {  include mime.types;  default\_type application/octet-stream;  sendfile on;  keepalive\_timeout 65;  server {  listen 80;  server\_name localhost;  location / {  root  **/opt/webapp/**;  index index.html index.htm;  }  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  root html;  }  }  } |

## 安装tomcat(Web0、Web1)

目的：处理java程序

下载tomcat

Shell ># wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-7/v7.0.62/bin/apache-tomcat-7.0.72.tar.gz

解压缩

Shell ># tar xf apache-tomcat-7.0.72.tar.gz

修改权限

保证web运行安全，这里示例使用tomcat

Shell>#chown -R tomcat:tomcat apache-tomcat-7.0.62

### 重定向工作目录

Tomcat安装完成后，它使用自己默认的工作路径工作。但是大部分时间我们需要自定义这个路径。

需要在server.xml中进行指定

配置示例如下

|  |
| --- |
| <Engine name="Catalina" defaultHost="localhost">  <Realm className="org.apache.catalina.realm.UserDatabaseRealm"  resourceName="UserDatabase"/>  <Host name="localhost" appBase="webapps"  unpackWARs="true" autoDeploy="true"  xmlValidation="false" xmlNamespaceAware="false">  <！-- 以下内容为制定tomcat的工作目录-->  <Context docBase="/opt/webapp" path="/" reloadable="true" swallowOutput="true"/>  <！-- 以上内容为制定tomcat的工作目录-->  </Host>  </Engine> |

### 改变服务端口

下面的示例就是将8080端口改为了80。

|  |  |
| --- | --- |
| 类型 | 字符段 |
| 原始 | <Service name="Catalina">  <Connector port="**8080**" protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443" />  **……**  </service> |
| 修改后 | <Service name="Catalina">  <Connector port="**80**" protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443" />  **……**  </service> |

## 安装keepalived服务器（LB0、LB1、Redis0、Redis1）

目的：nginx和Redis高可用

安装依赖

Shell>#yum -y install openssl-devel net-snmp-devel

下载软件包

shell># wget http://www.keepalived.org/software/keepalived-1.2.20.tar.gz

解压缩软件

shell># tar xf keepalived-1.2.20.tar.gz

shell># cd keepalived-1.2.20

编译

shell># ./configure --prefix="/usr/" --sysconfdir="/etc/" --enable-snmp

shell># make

针对RHEL系统，启用SNMP时使用如下命令编译

shell># make LDFLAGS="$(net-snmp-config --agent-libs) -lpopt -lssl -lcrypto"

针对其它类Unix系统

Shell># make

安装软件

shell># make install

添加keepalived到系统服务列表

|  |  |
| --- | --- |
| RHEL6 | shell># chkconfig --add keepalived |

设置keepalived在运行3级别时开机启动

|  |  |
| --- | --- |
| RHEL6 | shell># chkconfig --level 3 keepalived on |

配置

192.168.1.215

|  |
| --- |
| ! Configuration File for keepalived  global\_defs {  notification\_email {  dblvs@jn.idc  }  notification\_email\_from dblvs@jn.idc  smtp\_server 127.0.0.1  smtp\_connect\_timeout 30  router\_id LVS\_REDIS  }  vrrp\_script checkredis {  script "/etc/keepalived/check.sh"  interval 3  weight -20  }  vrrp\_instance VI\_1 {  state BACKUP  interface em2  virtual\_router\_id 61  priority 110  advert\_int 1  authentication {  auth\_type PASS  auth\_pass REDIS@122333  }  track\_script {  checkredis  }  virtual\_ipaddress {  192.168.1.220/24  }  }  virtual\_server 192.168.1.220 12002 {  delay\_loop 2  lb\_algo rr  lb\_kind DR  persistence\_timeout 50  protocol TCP  real\_server 192.168.1.215 12002 {  weight 20  }  } |

192.168.1.216

|  |
| --- |
| ! Configuration File for keepalived  global\_defs {  notification\_email {  dblvs@jn.idc  }  notification\_email\_from dblvs@jn.idc  smtp\_server 127.0.0.1  smtp\_connect\_timeout 30  router\_id LVS\_REDIS  }  vrrp\_script checkredis {  script "/etc/keepalived/check.sh"  interval 3  weight -20  }  vrrp\_instance VI\_1 {  state BACKUP  interface em2  virtual\_router\_id 61  priority 100  advert\_int 1  authentication {  auth\_type PASS  auth\_pass REDIS@122333  }  track\_script {  checkredis  }  virtual\_ipaddress {  192.168.1.220/24  }  }  virtual\_server 192.168.1.220 12002 {  delay\_loop 2  lb\_algo rr  lb\_kind DR  persistence\_timeout 50  protocol TCP  real\_server 192.168.1.216 12002 {  weight 20  }  } |

Check.sh脚本内容

|  |
| --- |
| #!/bin/sh  #mymater is localhost ip addr。eg : host ip is 192.168.1.215 mymaster='192.168.1.215'  mymaster='192.168.1.215'  PIDFILE=/var/run/redis-sentinel.pid  if [ -f $PIDFILE ];then  master=`/usr/local/bin/redis-cli -p 26379 info | grep 'mymaster' | awk -F 'address=' '{print $2}' | awk -F ':12002' '{print $1}'`  if [ $mymaster == $master ];then  exit 0  else  exit 1  fi  else  exit 2  fi |

然后我们启动keepalived就可以了。

## 安装Redis(Redis0、Redis1、Redis2)

目的：安装redis sentinel群集，redis0 和redis1 为业务提供节点，redis2 为sentinel仲裁节点。使用keepalived或者pacemaker实现高可用

## 安装FastDFS(FastDFS0、FastDFS1)

目的：存储文件

## 安装oracle

# 部署Web程序

Web前端

解包后放到Web0、Web1 的/opt/webapp下

Web后端

安放于Web0、Web1 的 /opt/webserver 下

使用shell># nohup java -jar web\_server.jar &启动应用

# 章节7. 启动与关闭

## 开启Redis

Shell># redis-server /etc/redis.conf

## 开启Tomcat

在/opt目录下，

shell># ./tomcat.sh start

可以使用shell># ./tomcat.sh status 查看tomcat的基本运行状态

## 开启服务

Shell># nohup java -jar <包文件名>.jar &

## zookeeper

/opt/zookeeper-3.4.9/bin/zkServer.sh start

/opt/zookeeper-3.4.9/bin/zkServer.sh stop

## 关闭Redis

Shell># kill -9 `ps aux | grep redis | grep -v grep |awk -F ' ' '{print $2}'`

## 关闭tomcat

./tomcat stop

如不成功，则可用如下命令

Shell># kill -9 `ps aux | grep ‘/opt/apache-tomcat/’ | grep -v grep |awk -F ' ' '{print $2}'`

## 关闭服务

Shell># kill -9 `ps aux | grep ‘user-provider.jar’ | grep -v grep |awk -F ' ' '{print $2}'`

或者使用

Shell># killall -9 java

启动fastdfs:

88.240：

* 启动fastdfs tracker

/opt/FastDFS/tracker/fdfs\_trackerd /etc/fdfs/tracker.conf

* 启动fastdfs storage

/opt/FastDFS/storage/fdfs\_storaged /etc/fdfs/storage.conf

* 启动nginx:

./nginx -c /usr/local/nginx/conf/nginx.conf

# 章节8. 问与答