

1 系统用户的初始化

先查看磁盘，home 目录下是否空间足够，如果不够则将用户目录挂载到其他磁盘

- 创建用户

切换到root用户，执行如下命令

```
useradd aifgw
# 密码默认 1qaz!QAZ
passwd aifgw
```

- 给用户aifgw赋权 sudo

```
vim /etc/sudoers
# 在root ALL=(ALL) ALL下 增加
aifgw ALL=(ALL) ALL
```

2 mysql数据库的安装

- 解压并复制

```
tar -zxvf mysql-5.6.41-linux-glibc2.12-x86_64.tar.gz
# 重命名
mv mysql-5.6.41-linux-glibc2.12-x86_64 mysql
cd mysql
scripts/mysql_install_db --basedir=/home/aifgw/mysql --datadir=/home/aifgw/mysql/data
sudo cp support-files/mysql.server /etc/init.d/mysqld
sudo chmod 755 /etc/init.d/mysqld
sudo cp support-files/my-default.cnf /etc/my.cnf
```

- 修改配置文件 /etc/init.d/mysqld

```
sudo vim /etc/init.d/mysqld
修改 basedir 和 datadir
basedir=/home/aifgw/mysql
datadir=/home/aifgw/mysql/data

修改启动命令
$bindir/mysqld_safe --datadir="$datadir" --pid-file="$mysqld_pid_file_path" $other_args
>/dev/null &
改为
```

```
$bindir/mysql_safe --user=root --datadir="$datadir" --pid-file="$mysqld_pid_file_path"
$other_args >/dev/null &
```

- **启动并登录mysql**

```
sudo service mysqld start
#提示输入密码, 直接回车
~/mysql/bin/mysql -uroot -p
```

- **登陆后修改root密码并支持远程登陆**

```
use mysql;
update user set host = '%' where user = 'root' and host = 'localhost';
update user set password=password("123456") where user="root";
delete from user where user='';
flush privileges;
```

3 mysql 主从的配置

重复2 的步骤, 安装两台mysql, 并分配主从

3.1 主机操作

- **创建同步账号**

```
CREATE USER 'sync'@'%' IDENTIFIED BY 'sync';
grant replication slave on *.* to 'sync'@'172.25.45.220' identified by 'sync';
flush privileges;
```

- **修改 /etc/my.cnf**

```
server-id=10
log-bin=mysql-bin
lower_case_table_names=1(数据表不区分大小写)
```

- **重启服务, root登录并查看状态**

```
#重启服务
sudo service mysqld restart
```

```
# root登录
~/mysql/bin/mysql -uroot -p
#查看状态并截图
show master status \G
```

```
all rights reserved. Other names may be trademarks of their respective
owners.

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

mysql> show master status\G
***** 1. row *****
      File: mysql-bin.000001
      Position: 120
      Binlog_Do_DB:
      Binlog_Ignore_DB:
      Executed_Gtid_Set:
1 row in set (0.00 sec)

mysql>
```

3.2 备机操作

- 修改 /etc/my.cn

```
server-id=14
lower_case_table_names=1(数据表不区分大小写)
```

- 配置同步账号

```
CHANGE MASTER TO
MASTER_HOST='10.228.204.38',MASTER_PORT=3306,MASTER_USER='sync',MASTER_PASSWORD='sync',MASTER_L
OG_FILE='mysql-bin.000001',MASTER_LOG_POS=1146;

SET GLOBAL SQL_SLAVE_SKIP_COUNTER=1;
```

- 启动和关闭同步

```
start slave;
stop slave;
```

- 检查状态并截图

```
#查看状态并截图
show slave status \G
```

```
-----
1 row in set (0.00 sec)

mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Connecting to master
      Master_Host: 10.228.204.38
      Master_User: sync
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000001
      Read_Master_Log_Pos: 120
      Relay_Log_File: localhost-relay-bin.000001
      Relay_Log_Pos: 4
      Relay_Master_Log_File: mysql-bin.000001
      Slave_IO_Running: Connecting
      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: 1
      Exec_Master_Log_Pos: 120
      Relay_Log_Space: 120
      Until_Condition: None
      Until_Log_File:
      Until_Log_Pos: 0
```

4 jdk 和 nginx的安装

- 准备包，并复制到对应目录

nginx相关包

[installEngine.sh](#)

[openssl-1.1.1d.tar.gz](#)

[tengine-2.3.2.tar.gz](#)

[zlib-1.2.11.tar.gz](#) [pcre-8.43.tar.gz](#) [nginx.conf](#)

jdk相关包

[deploy_jdk.sh](#)

[jdk-8u172-linux-x64.tar.gz](#)

- 执行安装

```
1安装jdk
./deploy_jdk.sh
2 安装 nginx
./installEngine.sh
```

```
# 检查是否安装ok
java -version
```

- 结合到当前网关的 nginx 升级，可以如下操作

```
# 杀死nginx进程
pkill nginx
# 查看原先的nginx 版本
nginx -V (大写的v)
# 备份 nginx
mv /home/aifgw/nginx /home/aifgw/nginx-old
# 将新的 tengine-2.3.3.tar.gz 包, 以及 openssl pcre zlib 放同一个目录下, 并修改 installTengine.sh
中的 ng
# 执行 installTengine.sh
./installTengine.sh
# 将升级前的 html 目录, config 配置文件还原
cp /home/aifgw/nginx-old/conf/nginx.conf /home/aifgw/nginx/conf/nginx.conf
cp -r /home/aifgw/nginx-old/html /home/aifgw/nginx/ht
# 测试加载配置文件
/home/aifgw/nginx/sbin/nginx -t -c /home/aifgw/nginx/conf/nginx.conf
# 查看版本是否正确
nginx -V (大写的v)
```

- 关于nginx 的升级和安装

configure 命令的参数含义

<https://nginx.org/en/docs/configure.html>

如何进行nginx升级

<https://www.cnblogs.com/sblack/p/12809034.html>

configure 和 原先的保持一致，参数 (通过 nginx -V 来查看编译参数)

只 make 不安装

将make后的 nginx 文件替换原先的文件

如何nginx 平滑升级

<https://www.cnblogs.com/bixiaoyu/p/6937453.html>

如何在nginx 升级的时候，加入模块

<https://blog.csdn.net/pcn01/article/details/105182600/>

http://tengine.taobao.org/document_cn/http_upstream_check_cn.html

- 有可能遇到的问题

1 安装失败，需要 缺少C 编译器

<https://www.cnblogs.com/moonhmily/p/11455814.html>

```
sudo yum -y install gcc-c++
```

2 安装失败，缺少 perl

<https://blog.csdn.net/pyd1040201698/article/details/98488982>

```
wget https://www.cpan.org/src/5.0/perl-5.28.0.tar.gz
tar -zxvf perl-5.28.0.tar.gz
cd perl-5.28.0
./Configure -des -Dprefix=$HOME/localperl
make && make install
```

5 redis 集群的安装

假设有三台机器，每台机器安装两个节点，一共6个节点 3主 3从
端口分布为 7001~7006

- 编译安装

```
tar -zxvf redis-5.0.8.tar.gz
#进入压缩后的文件夹
cd redis-5.0.8
#编译
make
#安装redis
cd src
make install
```

- 在某个机器上创建两个节点，并配置

```
# 循环创建目录
mkdir -p ~/redis-cluster/node1
mkdir -p ~/redis-cluster/node2
# 复制配置文件
cp ~/redis-5.0.8/redis.conf ~/redis-cluster/node1
cp ~/redis-5.0.8/src/{redis-server,redis-cli} ~/redis-cluster/node1
cp ~/redis-5.0.8/redis.conf ~/redis-cluster/node2
cp ~/redis-5.0.8/src/{redis-server,redis-cli} ~/redis-cluster/node2
```

- 修改配置redis.conf

```
端口配置 7001 ~ 7006
修改守护进程 daemonize no改为yes
开启远程访问 注释掉 bind 127.0.0.1
修改 protected-mode no
开启集群 cluster-enabled yes
```

- 复制到其他机器并修改端口

```
scp -r ~/redis-cluster aifgw@ip:/home/aifgw
```

- 启动所有redis节点

```
cd ~/redis-cluster/node1  
redis-server redis.conf
```

```
cd ~/redis-cluster/node2  
redis-server redis.conf
```

- 创建redis 集群 (注意修改ip)

```
redis-cli --cluster create 10.228.204.244:7001 10.228.204.244:7002 10.228.204.245:7003  
10.228.204.245:7004 10.228.204.246:7005 10.228.204.246:7006 --cluster-replicas 1
```

- 查看集群状态

```
# 登录redis集群  
redis-cli -h 10.228.204.244 -p 7001 -c  
# 命令查看  
cluster nodes  
cluster info
```

- 重置redis集群

```
# 分别在多个节点执行  
2 ps -x , 通过kill -9 杀死进程  
3  
4 rm -rf ~/redis-cluster/node1/nodes.conf  
5 rm -rf ~/redis-cluster/node2/nodes.conf  
6  
7 cd ~/redis-cluster/node1  
8 redis-server redis.conf  
9 cd ~/redis-cluster/node2  
10 redis-server redis.conf
```

- 安装问题处理

如果控制台一直卡住，则需要检查防火墙是否屏蔽了 17001~17006 端口

```
>>> Nodes configuration updated
>>> Assign a different config epoch to each node
>>> Sending CLUSTER MEET messages to join the cluster
Waiting for the cluster to join
.....
.....
```

6 安装网关视图前后台

6.1 安装网关视图后台

- 新建库 aifgw,并将 aifgw.sql 刷进去
- 解压aifgw-web-ui.zip

```
# 在home目录下创建目录 console
mkdir console
mv aifgw-web-ui.zip console/
# 解压
unzip aifgw-web-ui.zip
```

- 修改配置文件 application.yml

1. 修改启动端口 为9200

```
# 开发环境配置
server:
  # 服务器的HTTP端口，默认为8080
  port: 9200
  servlet:
    # 应用的访问路径
    context-path: /
tomcat:
  # tomcat的URI编码
```

- 修改配置文件 application-local.yml

1. 修改数据库配置

```
# 数据源配置
spring:
  datasource:
    type: com.alibaba.druid.pool.DruidDataSource
    driverClassName: com.mysql.cj.jdbc.Driver
    druid:
      # 主库数据源
      master:
        url: jdbc:mysql://10.19.90.22:30305/aifgw-guangzhou?useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertNullZero
        username: aifgw
        password: 1qaz!QAZ
      # 从库数据源
      slave:
        # 从数据源开关/默认关闭
        enabled: false
```

2. 修改redis 配置

注意:

redis 配置有两处需要 全部修改

```
# redis 配置
redis:
  # 地址
  cluster:
    nodes: 10.19.83.179:6001,10.19.83.179:6002,10.19.83.179:6003,10.19.83.179:6004,10.19.83.179:6005,10.19.83.179:6006
  # 端口, 默认为6379
  #port: 6379
  # 数据库索引
  database: 0
  # 密码
  password: foobared
```

```
sync:
  redis:
    pub-channel: CacheUpdateMessage
    lettuce:
      pool:
        max-active: 8
        max-wait: -1
        min-idle: 0
        max-idle: 8
    areas:
      default:
        # host: 127.0.0.1
        # port: 6379
        password: foobared
        cluster:
          nodes: 10.19.83.179:6001,10.19.83.179:6002,10.19.83.179:6003,10.19.83.179:6004,10.19.83.179:6005,10.19.83.179:6006
```

- 启动网关视图后台

```
./ry.sh
```

6.2 安装网关视图前台

- 解压并修改配置

```
# 复制aifgw-web-ui.zip 到 console 目录下并解压
mv aifgw-web-ui.zip console
unzip aifgw-web-ui.zip
```

- **nginx 配置参考**

```
worker_processes 1;
error_log "pipe:rollback logs/error_log interval=1d baknum=7 maxsize=2G";
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    access_log "pipe:rollback logs/access_log interval=1d baknum=7 maxsize=2G";
    client_body_timeout 600s;
    client_header_timeout 600s;
    send_timeout 600s;
    server_tokens off;
    sendfile on;
    keepalive_timeout 600s;
    gzip on;
    upstream gateway_console_cluster {
        ip_hash;
        server 10.101.65.83:9200;
        server 10.101.65.84:9200;
    }
    server {
        listen 9300;
        server_name localhost;
        location / {
            root /home/aifgw/console/aifgw-web-ui/dist;
            index index.html index.htm;
            try_files $uri $uri/ /index.html;
        }
        location /prod-api/ {
            proxy_pass http://gateway_console_cluster;
            proxy_redirect off;
            proxy_set_header REMOTE-HOST $remote_addr;
            proxy_set_header Cookie $http_cookie;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_next_upstream error timeout invalid_header http_500 http_502
            http_503 http_504;
        }
    }
    upstream gw {
        server 10.101.65.74:21100;
        server 10.101.65.75:21100;
        server 10.101.65.76:21100;
        server 10.101.65.77:21100;
    }
    server {
        listen 21100;
```

```
server_name localhost;
location / {
    proxy_pass http://gw;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_next_upstream error timeout invalid_header http_500 http_502
http_503 http_504;
}
}
}
```

- 启动nginx

```
# 启动nginx
nginx/sbin/nginx -c nginx/conf/nginx.conf
```

- 查看是否启动成功

<http://adress:9300/>

管理员:manager/stH@Oct27

开发者:dev/stH@Oct27

运营者:opr/stH@Oct27

7 网关后台 backend 的安装

- 解压

```
# 在home目录下创建目录 backend
mkdir backend
mv aifgw-backend-1.1.0.tar.gz backend/
# 解压
tar -zxvf aifgw-backend-1.1.0.tar.gz
```

- 配置redis

修改 aifgw.yml, 下图是redis集群的配置方式, 如果是单节点redis的配置, 将 isCluster 改为no

```
#redis 客户端配置
redisConfig:
  parameter:
    maxTotal: 200
    maxIdle: 10
    minIdle: 1
    testOnBorrow: false
    needWriteSyn: true
    connectionTimeout: 30000
  routers:
    - dataType: OprMenuInfo
      group: REDIS_CLUSTER
  serverGroups:
    - code: REDIS_CLUSTER
      isCluster: yes
      requirePass:
        addresses:
          - 10.228.204.244:7001
          - 10.228.204.244:7002
          - 10.228.204.245:7003
          - 10.228.204.245:7004
          - 10.228.204.246:7005
          - 10.228.204.246:7006
# - 10.10.83.179:6002
```

- 配置服务编排功能的数据库地址

查看 aifgw.yml 中有没有这个配置项

```
# 请求报文控制, 防SQL注入、XSS攻击
illegalCharacterCheck:
  enable: false
  regexpr: document.cookie|href|script|//|select |select/|select\(|select\*|insert |in
checkedProtocols:
  - http
  - https

# 熔断时间窗口配置
# 注意由于统计时间窗口是分钟级别的, 若 timeWindow 小于 60s, 则结束熔断状态后仍可能再进入熔断状态
circuitBreakingTimeWindowSeconds: 60

# 服务编排开关
arrangeEnabled: true
# 网关前缀开关
# 开启后将入网关的请求头改为gw开头
# apiCode -> gwApiCode
headerPrefixEnabled: false
```

如果这个配置项为 false,则不需要配置 服务编排的数据库地址。

如果 配置为 true , 或者不配置该配置项 (鲍哥加的配置) 则需要增加数据库配置信息

像某些局点已经把 default.xml 放到 网关config目录下了, 有些没有。 需要同时查看两个地方
最好是删除 aifgw-static-config 目录下的 default.xml 配置, 再打包。

修改 lib 中 aifgw-static-config.jar 中的配置文件 defaults.xml, 修改一个数据库地址
并删除 config 目录下的 /conf/system/service/default.xml 否则会影响

- 启动backend

```
./bin/startup daemon
```

8 安全网关 security 的安装

(待补充)

9 常见的安装问题

1 mysql 主从同步问题

2 端口防火墙屏蔽问题

可以通过 firewall-cmd 来放开口

3 console 配置的 redis area 和 console 前台不一致的问题

4 log4x 埋点不生效的问题

10 常用命令

1 查看端口进程

```
netstat -tunlp | grep 8092 # 查出进程号  
ps -x | grep 24814 # 利用进程号 查出某个进程的具体命令
```



```
develop@ecs-tousu-app1-0001 bin]$ netstat -tunlp | grep 8092  
tcp6        0      0  :::8092          :::*               LISTEN      24814/java  
develop@ecs-tousu-app1-0001 bin]$ ps -x | grep 24814  
24814 ?      Sl      22   18 java -jar zskserver-2.0.1.jar  
develop@ecs-tousu-app1-0001 bin]$
```

2 查看防火墙

```
#查看开放的端口  
firewall-cmd --list-ports  
#查询6379端口是否开放  
firewall-cmd --query-port=6379/tcp  
#移除6379端口  
firewall-cmd --permanent --remove-port=6379/tcp  
# 开放6379端口  
firewall-cmd --add-port=6379/tcp --permanent  
firewall-cmd --reload
```

参考链接 <https://www.cnblogs.com/wps54213/p/12571188.html>

3 文件复制

```
#跨主机的整个目录复制  
scp -r ${sourceDir}/ aigfw@ip:${destDir}
```

