Apusic 金蝶天燕

生成时间：2026-05-19

安装部署

金蝶Apusic服务治理平台

版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

Apusic是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

目录

1 ASGP V1.0-产品安装手册

1.1 安装要求

金蝶Apusic服务治理平台V1.0需要具备以下基本要求：

1. 网络

   由于nginx与tomcat里面各组件需要进行网络通信，所以需要保证各个服务之间的网络端口通信正常。

2. JDK

   ASGP V1.0运行依赖JDK1.8版本，可通过运行 `java -version` 命令查看是否安装了JDK以及安装版本；

3. 硬件

组件	要求
操作系统	Centos7
物理内存	32G或以上
硬盘	可用空间100G或以上
Java环境	JDK1.8
浏览器	常用浏览器较高版本

1.2 插件安装

1.2.1 Redis安装

1、将redis的安装包上传到服务器，如：`redis-6.2.6.tar.gz`

2、解压redes安装包：`tar -zvxf redis-6.2.6.tar.gz`

3、移动redis目录：`mv redis-6.2.6 /usr/local/redis`

4、进入到/usr/local/redis目录下：`cd /usr/local/redis`

5、编译：`make`

6、安装：`make PREFIX=/usr/local/redis install`

7、编辑配置文件，使启动redis时为后台启动：`vim redis.conf`

```
################################## GENERAL #####################################

# By default Redis does not run as a daemon. Use 'yes' if you need it.
# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.
# When Redis is supervised by upstart or systemd, this parameter has no impact.
daemonize yes
```

如上，将`no`改为`yes`，保存退出。

8、进入bin目录：`cd bin`

9、启动：`./redis-server ../redis.conf`

1.2.2 nginx安装

1. 安装gcc-c++编译器：`yum install gcc-c++`
2. 安装openssl库：`yum install -y openssl openssl-devel`
3. 安装pcre库：`yum install -y pcre pcre-devel`
4. 安装zlib库：`yum install -y zlib zlib-devel`
5. 在/usr/local/下创建文件nginx文件：`mkdir /usr/local/nginx`
6. 在网上下nginx包上传至Linux或者自己上传，如：

   `wget https://nginx.org/download/nginx-1.9.9.tar.gz`
7. 解压：`tar -zxvf nginx-1.9.9.tar.gz`
8. 进入解压之后的目录：`cd nginx-1.9.9/`
9. 使用nginx默认配置：`./configure`
10. 编译：`make`
11. 安装：`make install`
12. 编辑配置文件：`vim /usr/local/nginx/conf/nginx.conf`

```
worker_processes  1;
```

```
events {
    worker_connections  1024;
}


http {
    include       mime.types;
    default_type  application/octet-stream;

    sendfile        on;
    keepalive_timeout  65;

   upstream backend {
  ip_hash;
        server localhost:8765;
    }

  underscores_in_headers on;

    server {
        listen       8080;
        server_name  localhost;
  server_tokens off;  #关闭版本号

    location / {

      root  /home/asgp/AsgpMonitor/nginx/static; #asgp前端页面

            index  index.html index.htm;
        }

    location ^~/gateway/ {
```

```
        proxy_set_header X-Nginx-Proxy true;
        proxy_redirect off;
        proxy_set_header Host $host:$server_port;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header  X-Forwarded-For
$proxy_add_x_forwarded_for;
        client_max_body_size   500m;
        client_body_buffer_size  128k;
        proxy_connect_timeout  100;
        proxy_send_timeout   100;
        proxy_read_timeout 100;
        proxy_buffer_size 4k;
        proxy_buffers  4 32k;
        proxy_busy_buffers_size 64k;
        proxy_temp_file_write_size  64k;
        proxy_pass_request_headers on;
        proxy_pass http://backend/;
          }

      location /images {

        root  /home;

          }

        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }

    }
```

```
}
```

13. 进入nginx的sbin目录：cd /usr/local/nginx/sbin

14. 启动nginx：./nginx

1.2.3 mysql安装

1、 查看系统是否自带mariadb数据库：`rpm -qa|grep mariadb`

```
[root@localhost etc]# rpm -qa|grep mariadb
mariadb-libs-5.5.56-2.el7.x86_64
```

2、卸载卸载CentOS7系统自带mariadb：

`rpm -e --nodeps mariadb-libs-5.5.56-2.el7.x86_64`

3、删除etc目录下的my.cnf：`rm -rf /etc/my.cnf`

4、上传mysql安装包`mysql-5.7.33-linux-glibc2.12-x86_64.tar.gz`

5、解压安装包：`tar -zxvf mysql-5.7.33-linux-glibc2.12-x86_64.tar.gz`

6、重命名解压完的安装包为mysql:

`mv mysql-5.7.33-linux-glibc2.12-x86_64 /usr/local/mysql/`

7、创建组：`groupadd mysql`

8、创建用户：`useradd -r -g mysql mysql`

9、分配目录权限：`chown -R mysql:mysql /usr/local/mysql`

10、进入mysql的bin目录：`cd /usr/local/mysql/bin`

11、安装mysql：

`./mysql_install_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data`

12、移动文件：`cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql`

13、在etc下新建my.cnf，并进行编辑：`vim /etc/my.cnf`

```
[client]
port=3306
[mysqld]
port=3306
user=mysql
basedir=/usr/local/mysql
datadir=/usr/local/mysql/data
log-error=error.log
max_connections = 300
transaction_isolation = READ-COMMITTED
character-set-server = utf8mb4
collation-server = utf8mb4_general_ci
lower_case_table_names = 1
sql_mode=STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_
```

14、启动mysql：`service mysql start`

15、设置开机启动：`chkconfig --add mysql`

16、添加环境变量：`vim /etc/profile`

在文件最后添加：`export PATH=$PATH:/usr/local/mysql/bin`

保存退出后执行：`source /etc/profile`

17、停止mysql：`service mysql stop`

18、重置mysql密码：

进入mysql的bin目录执行：`./mysqld --skip-grant-tables &`

无密码登陆mysql：`mysql -u root`

执行：`update mysql.user set authentication_string=password('123456') where user='root' ;`

执行：`quit`

19、杀掉mysql进程：

`ps -ef | grep mysql`

`kill -9 ‘进程号’`

20、启动mysql：`service mysql start`

21、登陆mysql：`mysql -u root -p`

22、修改连接密码：`alter user 'root'@'localhost' identified by '123456';`

23、允许远程访问：

```
update user set host='%' where user='root';
```

```
grant all privileges on *.* to 'root'@'%' identified by 'pwd' with grant option;
```

24、刷新：`flush privileges;`

1.2.4 MongoDB安装

1. 上传安装包：`mongodb-linux-x86_64-3.4.18.tgz`
2. 解压安装包：`tar -zxvf mongodb-linux-x86_64-3.4.18.tgz`
3. 移动并重命名：`mv mongodb-linux-x86_64-3.4.18 /usr/local/mongodb`
4. 修改环境变量：`vim /etc/profile`

   在文件末尾添加：

   ```
   export MONGODB_HOME=/usr/local/mongodb
   ```

   ```
   export PATH=$PATH:$MONGODB_HOME/bin
   ```

   保存退出，执行：`source /etc/profile`
5. 进入mongo的bin目录：`cd /usr/local/mongodb/bin/`
6. 创建mongodb.conf文件：`vim mongodb.conf`

   添加以下内容：

   ```
   dbpath = /usr/local/mongodb/data/db
   logpath = /usr/local/mongodb/logs/mongodb.log
   port = 27017
   fork = true
   nohttpinterface = true
   ```

7. 根据配置文件创建相应的文件夹：

   `mkdir -p /usr/local/mongodb/data/db`

   `mkdir -p /usr/local/mongodb/logs`

   `touch /usr/local/mongodb/logs/mongodb.log`

8. 启动：`./mongod -f mongodb.conf`

9. 创建数据库：

   执行mongo命令进入命令列：`mongo`

   创建数据库：`use asgp`

```
[root@localhost bin]# mongo
MongoDB shell version v3.4.18
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.18
Server has startup warnings:
2022-05-19T17:40:30.038+0800 I CONTROL  [initandlisten]
2022-05-19T17:40:30.050+0800 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-05-19T17:40:30.050+0800 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrest
2022-05-19T17:40:30.050+0800 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is n
2022-05-19T17:40:30.050+0800 I CONTROL  [initandlisten]
2022-05-19T17:40:30.050+0800 I CONTROL  [initandlisten]
2022-05-19T17:40:30.050+0800 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2022-05-19T17:40:30.050+0800 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2022-05-19T17:40:30.051+0800 I CONTROL  [initandlisten]
2022-05-19T17:40:30.051+0800 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2022-05-19T17:40:30.051+0800 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2022-05-19T17:40:30.051+0800 I CONTROL  [initandlisten]
> show dbs
admin  0.000GB
local  0.000GB
> use asgp
switched to db asgp
```

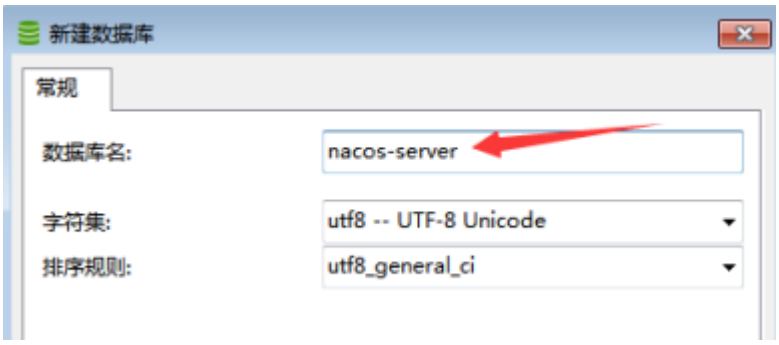
1.2.5 nacos安装

1. 上传安装包到服务器：`nacos-server-1.4.1.zip`

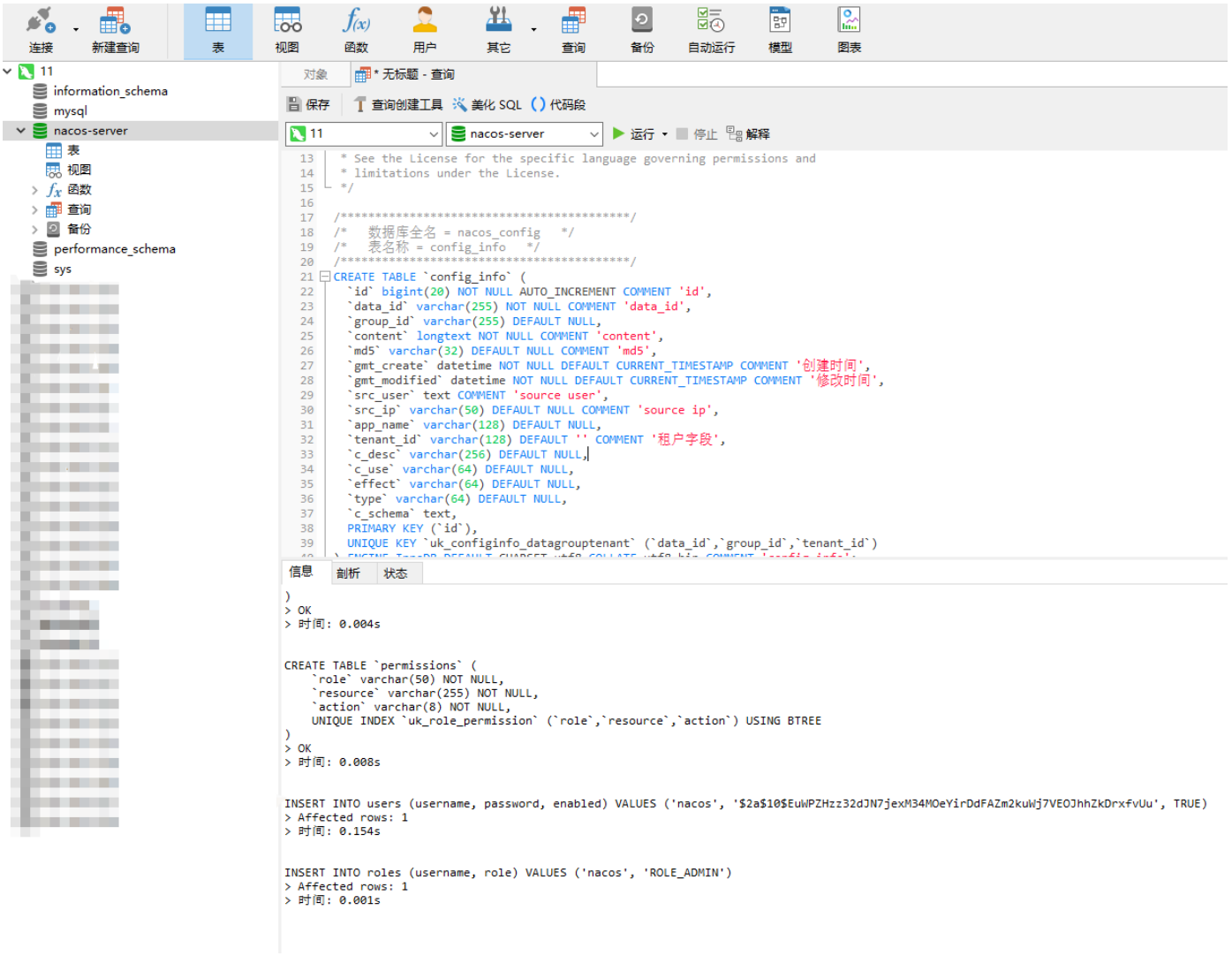
2. 解压安装包：`unzip nacos-server-1.4.1.zip`

3. 移动安装包：`mv nacos /usr/local`

4. 创建nacos-server数据库：



5. 执行nacos-server数据库初始化脚本，初始化脚本在nacos-server解压后的conf子目录下：`/usr/local/nacos/conf/nacos-mysql.sql`



6. 查找服务器中java的安装目录：`echo $JAVA_HOME`

   结果为空的话说明未配置环境变量，需要配置环境变量：

7. 终端输入：`which java`

```
[root@localhost conf]# which java
/usr/bin/java
```

8. 终端输入：`ls -lr /usr/bin/java`

```
[root@localhost conf]# ls -lr /usr/bin/java
lrwxrwxrwx. 1 root root 22 5月  18 02:15 /usr/bin/java -> /etc/alternatives/java
```

9. 终端输入：`ls -lrt /etc/alternatives/java`

```
[root@localhost conf]# ls -lrt /etc/alternatives/java
lrwxrwxrwx. 1 root root 72 5月  18 02:15 /etc/alternatives/java -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64/jre/bin/java
```

10. 至此，我们确定java的安装目录为：

    /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64

11. 配置环境变量：`vim /etc/profile`

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
```

    保存退出，并执行：`source /etc/profile`

12. 测试配置结果：`echo $JAVA_HOME`

13. 修改 `startup.sh` 脚本中 `JAVA_HOME` 的内容，替换成服务器中java安装目录的绝对路径：`vim /usr/local/nacos/bin/startup.sh`，注释掉上面的四行，添加第五行内容

```
#[ ! -e "$JAVA_HOME/bin/java" ] && JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64
#[ ! -e "$JAVA_HOME/bin/java" ] && JAVA_HOME=/usr/java
#[ ! -e "$JAVA_HOME/bin/java" ] && JAVA_HOME=/opt/taobao/java
#[ ! -e "$JAVA_HOME/bin/java" ] && unset JAVA_HOME
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64/jre
if [ -z "$JAVA_HOME" ]; then
  if $darwin; then
```

14. 修改 `application.properties` 中有关数据库的内容：

```
### Connect URL of DB:
db.url.0=jdbc:mysql://127.0.0.1:3306/nacos-server?characterEncoding=utf8&connectTimeout=1000&socketTimeout=3000&autoReconnect=true&useUnicode=true&useSSL=false&serverTimezone=UTC
db.user.0=root
db.password.0=123456
```

15. 启动nacos：`./startup.sh -m standalone`

16. 浏览器输入：http://ip:8848/nacos

    用户名：nacos

    密码：nacos

1.2.6 rocketmq安装

1. 上传安装包到服务器：`rocketmq-all-4.9.2-bin-release.zip`

2. 解压安装包：`unzip rocketmq-all-4.9.2-bin-release.zip`

3. 移动目录：`mv rocketmq-4.9.2/ /usr/local/rocketmq`

4. 调整内存大小：

   进入 `/usr/local/rocketmq/bin` 目录下找到 `runbroker.sh` 和 `runserver.sh` 和 `tools.sh` 然后vim 编辑 把所有8G,4G,2G 的参数 改为256m 128m（根据实际情况调整）

5. 启动：

   `nohup sh bin/mqnamesrv > mqsrv.log &`

   `nohup sh bin/mqbroker -n 服务器ip:9876 autoCreateTopicEnable=true -c ./conf/broker.conf > mqbroker.log &`

6. 配置管理端：

   上传 `rocketmq-console-ng-1.0.1.jar` 包到rocketmq安装路径下，然后执行：`nohup java -jar rocketmq-console-ng-1.0.1.jar > ng.log &`

7. 通过浏览器验证：

   `http://服务器ip:8888/`

   账号：admin

密码：admin

1.3 ASGP安装

1. 上传安装包到服务器：`ASGP-1.0-linux64-tomcat-R202205180931.zip`

2. 解压安装包：`unzip ASGP-1.0-linux64-tomcat-R202205180931.zip`

```
-rw-r--r--. 1 root root 669117085 5月  20 15:57 ASGP-1.0-linux64-tomcat-R202205180931.zip
-rw-r--r--. 1 root root     12626 5月  18 09:31 build.xml
drwxr-xr-x. 2 root root       118 5月  18 09:31 config-detail
-rw-r--r--. 1 root root       130 5月  18 09:31 config.properties
drwxr-xr-x. 8 root root       100 5月  18 09:31 product-config
-rw-r--r--. 1 root root       111 5月  18 09:31 startant.cmd
-rw-r--r--. 1 root root        18 5月  18 09:31 start_deploy_tool.cmd
-rwxr-xr-x. 1 root root       189 5月  18 09:31 start_deploy_tool.sh
```

3. 修改引擎配置文件：`vim config-detail/AsgpEngine.properties`

```
#etcd 配置
ETCD_URL=192.168.1.15:2379
#引擎端口
ENGINE_PORT=9080
ENGINE_SSL_PORT=9443

#javaRunner 配置
#JAVA_RUNNER_LOCAL=/opt/asgp/ASGP-Runner/asgp/javaRunner
JAVA_RUNNER_MEMERY=512m
JAVA_RUNNER_LOG=info
```

4. 修改管控端配置文件：`vim config-detail/AsgpMonitor.properties`

```
# 数据库类型:
# 达梦数据库: dm7; mysql数据库: mysql; 人大金仓数据库: kingbase8; 南大通用数据库: gbase
DB.TYPE=mysql
DB.IP=192.168.1.15
DB.PORT=3306
DB.USERNAME=root
DB.PASSWORD=123456

#application-dev.yml
#数据库连接信息
spring.datasource.url=jdbc:mysql://localhost:3306/apusic_asgp?useUnicode=true&characterEncoding=UTF
ToNull&serverTimezone=Asia/Shanghai
spring.datasource.username=root
spring.datasource.password=123456
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

#redis配置信息   application-dev.yml
spring.redis.host=192.168.1.15
spring.redis.port=6379

# mongodb配置信息 application-dev.yml
spring.data.mongodb.host=192.168.1.15
spring.data.mongodb.port=27017


# 配置rocketmq的生产者  application-dev.yml
#open/no
rocketmq.producer.isOpen=open
```

```
# mq的nameserver地址
rocketmq.producer.namesrvAddr=192.168.1.15:9876


#配置rocketmq的消费者  application-admin.yml  application-asgp.yml
#open/no
rocketmq.cousumer.isOpen=yes
rocketmq.cousumer.namesrvAddr=192.168.1.15:9876


#统一变量配置
#nacos服务注册中心   application-dev.yml
apusic.nacos.server-addr=localhost:8848
#数据库类型   application-dev.yml
apusic.dbType=mysql
#文件上传路径 application-admin.yml   application-resources.yml
upload.file.path=/home/uploadFile
#application-admin.yml
userIcon.file.path=/home/icon
```

5. 执行source命令：`source start_deploy_tool.sh`

```
-rw-r--r--. 1 root root 669117085 5月  20 15:57 ASGP-1.0-linux64-tomcat-R202205180931.zip
-rw-r--r--. 1 root root     12626 5月  18 09:31 build.xml
drwxr-xr-x. 2 root root        65 5月  20 16:23 config-detail
-rw-r--r--. 1 root root       130 5月  18 09:31 config.properties
drwxr-xr-x. 8 root root       100 5月  18 09:31 product-config
-rw-r--r--. 1 root root       111 5月  18 09:31 startant.cmd
-rw-r--r--. 1 root root        18 5月  18 09:31 start_deploy_tool.cmd
-rwxr-xr-x. 1 root root       189 5月  18 09:31 start_deploy_tool.sh
[root@localhost asgp]# source start_deploy_tool.sh
Unable to locate tools.jar. Expected to find it in /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64/lib/tools.jar
Buildfile: /home/asgp/build.xml

SHOWMESSAGE:
     [echo] 部署工具的配置信息在config.properties文件中，启用工具前，请提前做好配置
     [echo] 可用命令如下：
     [echo] ant reallAsgpMonitor       :清理管理平台并重新生成
     [echo] ant reallAsgpEngine        :[Linux]清理引擎并重新生成
     [echo] ant reall              :[Linux]清理所有并重新生成
     [echo] ant initdb              :[慎用]重新创建ADXP数据库并初始化，数据库需要启动
     [echo] ant reconfig               :重新配置所有配置文件
     [echo] 请使用ant -p 命令查看所有可用任务
     [echo] 使用ant 命令1 命令2 命令3 可自动顺序执行多个命令

BUILD SUCCESSFUL
Total time: 3 seconds
[root@localhost asgp]#
```

6. 清理管理平台并重新生成：`ant reallAsgpMonitor`

```
[root@localhost asgp]# ant reallAsgpMonitor
Unable to locate tools.jar. Expected to find it in /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64/lib/tools.jar
Buildfile: /home/asgp/build.xml

clearAsgpMonitor:

getAsgpMonitor:
    [untar] Expanding: /home/asgp/product-config/product/asgp-monitor.tar.gz into /home/asgp/product-config/product-temp/AsgpMonitor
    [mkdir] Created dir: /home/asgp/product-config/product-temp/AsgpMonitor/asgp-monitor/log

configAsgpMonitor:
   [rename] DEPRECATED - The rename task is deprecated.  Use move instead.

clearTempt:
   [delete] Deleting directory /home/asgp/product-config/product-temp
    [mkdir] Created dir: /home/asgp/product-config/product-temp

reallAsgpMonitor:

BUILD SUCCESSFUL
Total time: 25 seconds
```

7. 清理引擎并重新生成：`ant reallAsgpEngine`

```
[root@localhost asgp]# ant reallAsgpEngine
Unable to locate tools.jar. Expected to find it in /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64/lib/tools.jar
Buildfile: /home/asgp/build.xml

clearAsgpEngine:

getAsgpEngine:
    [mkdir] Created dir: /home/asgp/AsgpEngine
    [untar] Expanding: /home/asgp/product-config/product/asgp-engine.tar.gz into /home/asgp/product-config/product-temp/AsgpEngi
     [exec] 未找到目录: /home/asgp/product-config/product-temp/AsgpEngine/asgp_install_dir, 新建目录: /home/asgp/product-config/p
ll_dir
     [exec] /home/asgp/product-config/product-temp/AsgpEngine/asgp_packages/librsa.so

configAsgpEngine:
   [rename] DEPRECATED - The rename task is deprecated.  Use move instead.
   [rename] DEPRECATED - The rename task is deprecated.  Use move instead.
   [rename] DEPRECATED - The rename task is deprecated.  Use move instead.
   [rename] DEPRECATED - The rename task is deprecated.  Use move instead.

clearTempt:
   [delete] Deleting directory /home/asgp/product-config/product-temp
    [mkdir] Created dir: /home/asgp/product-config/product-temp

reallAsgpEngine:

BUILD SUCCESSFUL
```

8. 重新创建ASGP数据库并初始化，数据库需要启动：`ant initdb`

```
[root@localhost asgp]# ant initdb
Unable to locate tools.jar. Expected to find it in /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64/lib/tools.jar
Buildfile: /home/asgp/build.xml

initdb:

sub_initdb:
     [echo] 执行此任务前，请确认数据库服务已开启！
      [sql] Executing resource: /home/asgp/product-config/dbScripts/apusic_asgp.sql
      [sql] 401 of 401 SQL statements executed successfully

BUILD SUCCESSFUL
Total time: 3 seconds
```

9. 修改 `AsgpEngine/asgp/javaRunner/config` 的 `application.yaml`：

```yaml
logging:
  level:
    root: info

spring:
    data:
        mongodb:
          uri: mongodb://192.168.1.15:27017/asgp

##日志开启写入mq后，需要配置rocketmq的produer配置
rocketmq:
  producer:
    isOpen: open     #open/no
    # 发送同一类消息设置为同一个group，保证唯一默认不需要设置，rocketmq会使用ip@pid（pid代表jvm名字）作为唯一标识
    groupName: asgp-group
    # mq的nameserver地址
    namesrvAddr: 192.168.1.15:9876
    # 消息最大长度 默认 1024 * 4 (4M)
    maxMessageSize: 4096
    # 发送消息超时时间，默认 3000
    sendMsgTimeOut: 3000
    # 发送消息失败重试次数，默认2
    retryTimesWhenSendFailed: 2

asgp:
  cache:
    expired: ${APISIX_CONF_EXPIRE_TIME}
    capacity: 1000
  db:
    cache:
      expired: 60
      capacity: 1000
```

```
    testWait: 1000
    proxyNode: 127.0.0.1:9080
  etcd:
    uri: 192.168.1.15:2379
#  plugin:
#    path: I:\apusic\asgp-alb-java-runner\plugin
  socket:
    file: ${APISIX_LISTEN_ADDRESS}
  handler:
    logging:
      pluginName: serverdefalut
      logClass: com.apusic.asgp.runner.plugin.server.defalut.impl.MongoLoggingDaoImpl
  java_runner:
    engine_id: {JAVA_RUNNER_ID}
  xss:
    writeList:
      - /asgp/service/add
      - /asgp/ssl/add
```

10. 进入 `AsgpMonitor/bin` 目录启动管控端：

    进入目录：`cd AsgpMonitor/bin`

    授权：`chmod 755 *.sh`

    启动：`./asgpMonitor.sh start`

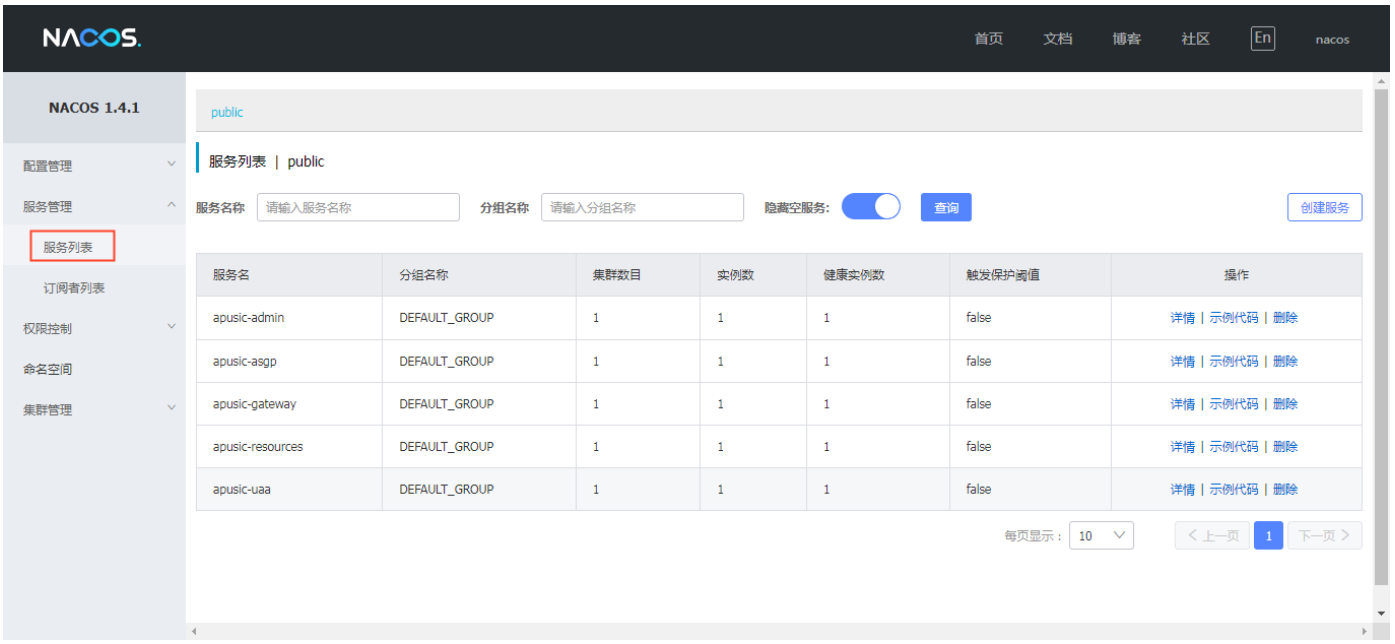
```
[root@localhost bin]# ./asgpMonitor.sh start
当前目录:/home/asgp/AsgpMonitor/bin
----------开始启动apusic-admin----------
APUSIC_ADMIN pid is 45337
----------apusic-admin启动成功----------
----------开始启动apusic-uaa----------
APUSIC_UAA pid is 53300
----------apusic-uaa启动成功----------
----------开始启动apusic-resources----------
APUSIC_RESOURCES pid is 67310
----------apusic-resources启动成功----------
----------开始启动apusic-asgp----------
APUSIC_ASGP pid is 73410
----------apusic-asgp启动成功----------
----------开始启动apusic-gateway----------
APUSIC_GATEWAY pid is 78939
----------apusic-gateway启动成功----------
```

11. 验证服务是否注册到nacos：

浏览器输入：http://192.168.1.15:8848/nacos/#/login

账号：nacos

密码：nacos



12. 验证是否正常进入平台：

浏览器输入：http://192.168.1.15:8080/#/login

用户名：sysadmin

密码：123456

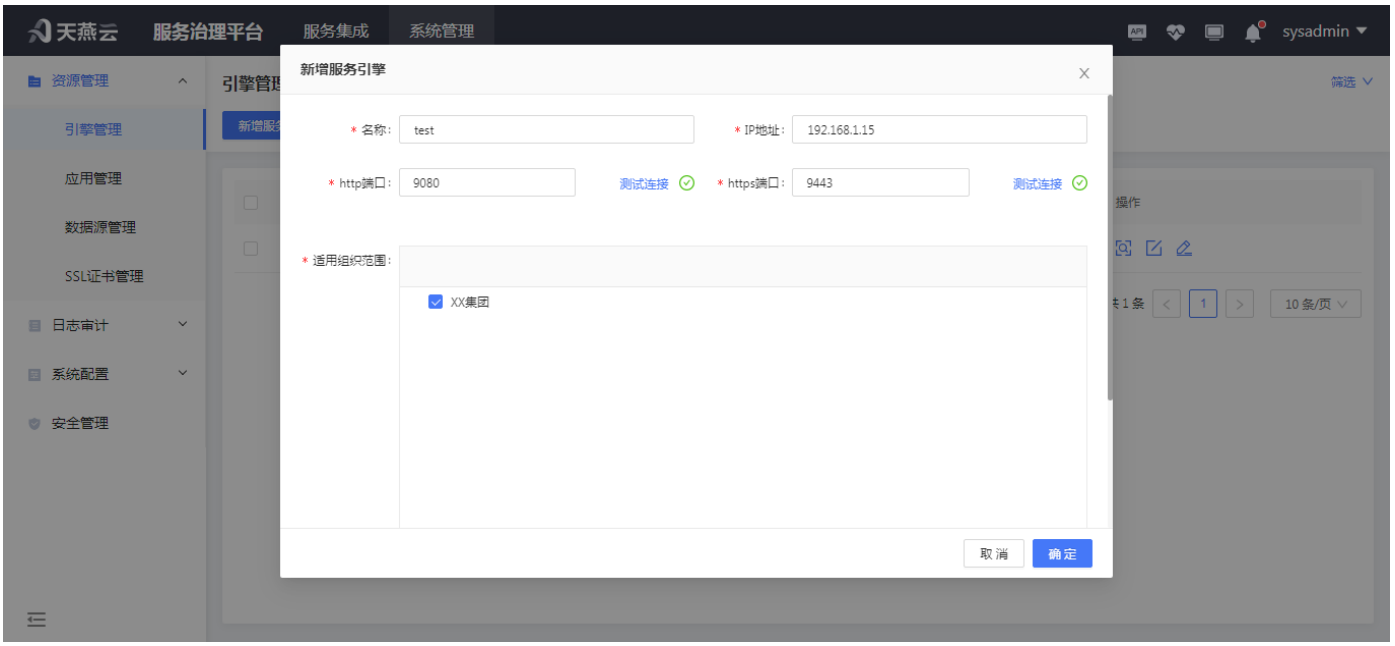
1. 进入AsgpEngine目录启动引擎：`./start_asgp.sh`

```
[root@localhost AsgpEngine]# ./start_asgp.sh
检测到系统为centos
检测到硬件架构为x86
已经安装：/home/asgp/AsgpEngine/
/usr/local/openresty/luajit/bin/luajit ALB stop
nginx: [alert] could not open error log file: open() "/home/asgp/AsgpEngine/asgp/logs/error.log" failed (2: No such file or directory)
2022/05/20 17:33:10 [notice] 104273#0: signal process started
2022/05/20 17:33:10 [error] 104273#0: open() "/home/asgp/AsgpEngine/asgp/logs/nginx.pid" failed (2: No such file or directory)
/usr/local/openresty/luajit/bin/luajit ALB start
Warning! Current maximum number of open file descriptors [1024] is too small, please increase user limits by execute 'ulimit -n <new use
 performance is low.
```

2. 验证是否启动成功：`ps -ef | grep nginx`

```
[root@localhost sbin]# ps -ef | grep nginx
nobody     5476  72866  0 17:42 ?        00:00:00 nginx: worker process
root       6452 120775  0 17:42 pts/1    00:00:00 grep --color=auto nginx
root      72866      1  0 5月19 ?       00:00:00 nginx: master process ./nginx
root     104694      1  0 17:33 ?        00:00:00 nginx: master process openresty -p /home/asgp/AsgpEngine/asgp
nobody   104695 104694  0 17:33 ?        00:00:01 nginx: worker process
nobody   104696 104694  0 17:33 ?        00:00:01 nginx: worker process
nobody   104697 104694  0 17:33 ?        00:00:02 nginx: worker process
nobody   104698 104694  0 17:33 ?        00:00:03 nginx: worker process
nobody   104699 104694  0 17:33 ?        00:00:01 nginx: worker process
nobody   104700 104694  0 17:33 ?        00:00:05 nginx: worker process
nobody   104701 104694  0 17:33 ?        00:00:00 nginx: cache manager process
root     104703 104694  0 17:33 ?        00:00:01 nginx: privileged agent process
```

3. 进入系统测试引擎是否可连接：


天燕云
服务治理平台
服务集成
系统管理
sysadmin
资源管理
引擎管理
应用管理
数据源管理
SSL证书管理
日志审计
系统配置
安全管理
新增服务引擎
* 名称: test
* IP地址: 192.168.1.15
* http端口: 9080
测试连接
* https端口: 9443
测试连接
* 适用组织范围:
XX集团
取 消
确 定

全国统一服务热线

4008-555-800

金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年，前身为“金蝶中间件公司”，是金蝶集团旗下新一代软件基础云平台服务商，云计算国家标准制定企业，国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业，也是“两网一站四库十二金”国家重点工程的基础平台提供商，产品广泛应用于政府、军工、金融、能源等关键行业，累计服务客户总数超过10万家。

Apusic 金蝶天燕

云计算国家标准制定企业

金蝶集团旗下基础软件企业

信息技术应用创新核心企业

官网：www.apusic.com