



APUSIC
固若长城
睿比世界

安装手册

金蝶Apusic中间件云平台V8.0

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

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

 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 前言
 - 1.1 适用读者
 - 1.2 相关文档
- 2 系统环境要求
 - 2.1 环境要求
 - 2.2 配置要求
- 3 安装前准备工作
 - 3.1 安装JDK
 - 3.2 安装数据库
 - 3.3 安装缓存
 - 3.4 产品安装包
 - 3.4.1 获取安装包
 - 3.4.2 安装介质说明
 - 3.4.3 服务端口说明
- 4 产品安装:
 - 4.1 单机部署
 - 4.1.1 安装存储服务
 - 4.1.2 安装日志服务
 - 4.1.3 安装中间件云平台
 - 4.1.4 安装指标监控服务
 - 4.1.5 安装管控代理
 - 4.2 集群部署
 - 4.2.1 安装存储服务集群
 - 4.2.2 安装日志服务集群
 - 4.2.3 安装中间件云平台
 - 4.2.3.1 安装maas
 - 4.2.3.2 安装高可用keepalived
 - 4.2.4 安装监控服务集群
- 5 产品卸载
 - 5.1 卸载指标监控
 - 5.2 卸载日志服务
 - 5.3 卸载存储服务

- 5.4 卸载中间件云平台
- 5.5 卸载管控代理
- 6 其他操作
 - 6.1 指标监控
 - 6.1.1 启动
 - 6.1.2 停止
 - 6.1.3 查看状态
 - 6.2 日志服务
 - 6.2.1 启动
 - 6.2.2 停止
 - 6.2.3 查看状态
 - 6.3 存储服务
 - 6.3.1 启动
 - 6.3.2 停止
 - 6.3.3 查看状态
 - 6.4 中间件云平台
 - 6.4.1 启动
 - 6.4.2 停止
 - 6.5 管控代理
 - 6.5.1 启动
 - 6.5.2 停止
 - 6.5.3 查看状态

1 前言

本文档介绍了金蝶Apusic中间件云平台（简称“ACP”）V8.0在各种计算环境、单机及高可用集群部署架构的安装部署说明以及常见问题。

1.1 适用读者

本文档主要面向金蝶Apusic中间件云平台的管理运维人员，应用程序开发人员。

1.2 相关文档

有关金蝶Apusic中间件云平台产品更多信息，请参考如下文档资源：

- [金蝶Apusic中间件云平台产品简介](#)
- [金蝶Apusic中间件云平台功能清单](#)
- [金蝶Apusic中间件云平台用户手册](#)
- [金蝶Apusic中间件云平台管理员手册](#)

2 系统环境要求

ACP产品支持Linux（包括采用龙芯和飞腾等国产芯片的Linux服务器）等多个操作系统平台的安装部署。

2.1 环境要求

软件及操作系统环境要求，如下表：

组件	要求
操作系统	Linux Red Hat 5.2或以上； 国产操作系统如银河麒麟系列、中标麒麟系列、普华、中科红旗、深度等。
CPU	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz或以上； HUAWEI,Kunpeng 920； phytium FT1500a等。
浏览器	FireFox 70及以上、Chrome 60及以上、IE 11及以上

2.2 配置要求

平台提供单机、集群两种安装模式，对应的配置要求，如下表：

部署模式	操作系统	硬件规格 (CPU/内存/硬盘)	服务器台数
单机	Linux	8核/16G/500G	1
集群	Linux	8核/16G/1T	4

3 安装前准备工作

3.1 安装JDK

ACP部分组件运行环境依赖于JAVA运行环境，在运行这些组件之前，需要具备。

目前支持的JDK厂商版本包括：Oracle JDK 17+、Open JDK 17+、IBM JDK 17+

3.2 安装数据库

ACP的主要业务数据存储的关系型数据库中。在这种情况下，ACP运行环境需要安装数据库服务，来实时记录授权产品的使用情况，并将数据同步到数据库中。

当前支持MySQL、达梦、金仓等多种类型的关系数据库。

3.3 安装缓存

ACP的热点数据缓存在缓存数据库中。推荐使用金蝶Apusic分布式缓存，也可使用Redis。

3.4 产品安装包

3.4.1 获取安装包

从[金蝶天燕官方网站](#)下载金蝶 Apusic 中间件云平台软件安装包，或从金蝶 Apusic 中间件云平台软件产品光盘中获得相应的安装包文件。

3.4.2 安装介质说明

ACP V8.0完整的产品包括如下安装程序文件，不同CPU架构平台请使用对应的产品安装包。若产品介质名称中不包含x86_64、arm64平台架构的字样，则适合跨平台部署，无需区分。

组件名称	文件名	说明
中间件云平台	acp-maas-v8.0.tar.gz	Web管控台，包括云门户、管控中心、监控中心、日志中心等
指标监控	acp-metrics-server-v2.37-linux.amd64.tar.gz	指标监控服务，用于平台、服务器、中间件的指标数据的采集和存储
日志服务	acp-log-server-v7.5-linux.amd64.tar.gz	日志服务，用于平台、服务器、中间件的日志数据的采集和存储

存储服务	acp-storage-server-v1.0-linux.amd64.tar.gz	分布式存储服务，用于软件仓库，数据备份等数据存储
管控代理	acp-maas-agent-v8.0-linux.amd64.tar.gz	用于中间件全生命周期管理、暴露指标、上报日志等

3.4.3 服务端口说明

端口	对应组件	作用	访问范围
80	中间件云平台	反向代理	外部访问
8000	中间件云平台	统一认证服务	内部访问
9999	中间件云平台	云门户	内部访问
9997	中间件云平台	监控中心	内部访问
9666	中间件云平台	管控中心	内部访问
9016	中间件云平台	告警中心	内部访问
9014	中间件云平台	工单系统	内部访问
6888	中间件云平台	日志中心	内部访问
9090	指标监控	指标查询API	内部访问
8080	指标监控	指标监控配置管理	内部访问
9093	指标监控	告警通知	内部访问
9094	指标监控	告警通知集群通信	内部访问
9001	存储服务	web端口	内部访问
9010	存储服务	管控端口	内部访问
5044	日志服务	日志采集	内部访问
9200	日志服务	客户端通信	内部访问
9300	日志服务	集群内部通信	内部访问

4 产品安装:

ACP支持单机、集群两种部署方式。单机模式仅限于用于测试、验证,生产环境推荐使用集群模式。

4.1 单机部署

4.1.1 安装存储服务

1.解压安装包

```
mkdir -p /opt/acp
tar -xzvf acp-storage-server-v1.0.linux-amd64.tar.gz -C /opt/acp
```

2.执行安装脚本

```
cd /opt/acp/storage-server/bin
./install.sh --data.path /home
```

安装参数说明:

- --data.path: 数据存储路径。通过 df -h命令获取剩余空间充足且非root的分区。如下图, data.path为/home

```
[root@linux-4-195 bin]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        7.8G   0 7.8G   0% /dev
tmpfs           7.8G   0 7.8G   0% /dev/shm
tmpfs           7.8G 753M 7.1G  10% /run
tmpfs           7.8G   0 7.8G   0% /sys/fs/cgroup
/dev/mapper/centos-root 445G  21G 425G   5% /
/dev/mapper/centos-home 50G 173M  50G   1% /home
/dev/sda1       1014M  194M  821M  20% /boot
shm             64M    0  64M   0% /run/containerd/io.containerd.grpc.v1.cri/sandbox-c4c46558db911ea5c81/shm
overlay         445G  21G 425G   5% /run/containerd/io.containerd.runtime.v2.task/k86c4c46558db911ea5c81/rootfs
overlay         445G  21G 425G   5% /run/containerd/io.containerd.runtime.v2.task/k80291debb9181550d846d/rootfs
tmpfs           1.6G   0  1.6G   0% /run/user/0
```

出现如下结果, 表示安装成功:

```
[root@linux-4-162 bin]# ./install.sh --data.path /home
Archive:  /opt/acp/storage-server/pkggs/minio-amd64.zip
  creating: /opt/acp/storage-server/runtime/minio/
  inflating: /opt/acp/storage-server/runtime/minio/initial_aump_package.sh
  inflating: /opt/acp/storage-server/runtime/minio/mc
  inflating: /opt/acp/storage-server/runtime/minio/minio
  inflating: /opt/acp/storage-server/runtime/minio/start.sh
  inflating: /opt/acp/storage-server/runtime/minio/startstandalone.sh
  inflating: /opt/acp/storage-server/runtime/minio/stop.sh
-----
start minio
start minio success!
-----
```

4.1.2 安装日志服务

(本服务依赖Java环境, 请先安装JDK17)

1. 解压安装包

```
tar -xzvf acp-log-server-v7.12.linux-amd64.tar.gz -C /opt/acp
```

2. 执行安装脚本

```
cd /opt/acp/log-server/bin
./install.sh
```

出现如下结果, 表示安装成功:

```

[root@linux-4-162 bin]# ./install.sh
-----
install LogServer start
installing, please wait
install_path: /opt/acp/log-server/runtime
-----
install ailp-es start
installing, please wait
es es_pkg_path: /opt/acp/log-server/pkg/elasticsearch-7.5.0-linux-x86_64.tar.gz
es es_install_path: /opt/acp/log-server/runtime

es JAVA_HOME path: /opt/acp/log-server/runtime/elasticsearch/jdk
es JAVA_HOME: /opt/acp/log-server/runtime/elasticsearch/jdk
install
modify config
update /etc/sysctl.conf
register ailp-es service
Created symlink from /etc/systemd/system/multi-user.target.wants/ailp-es.service to /usr/lib/systemd
create crontab auto clear es data
check es start success
ailp-es start is success!
ailp-es PID 29589
ailp-es http_port is 9200, transport_port is 9300
install ailp-es end
-----
install ailp-logstash start
installing, please wait
logstash JAVA_HOME path: /opt/acp/log-server/runtime/logstash/jdk
logstash JAVA_HOME: /opt/acp/log-server/runtime/logstash/jdk
Created symlink from /etc/systemd/system/multi-user.target.wants/ailp-logstash.service to /usr/lib/s
start ailp-logstash success!
ailp-logstash port is 9600
ailp-logstash PID 30060
install ailp-logstash end
-----
install LogServer end
-----

```

4.1.3 安装中间件云平台

(本服务依赖Java环境, 请先安装JDK17)

1. 解压安装包

```
tar -xzf acp-maas-v8.0.linux-amd64.tar.gz -C /opt/acp
```

2. 初始化数据 (仅需执行一次)

创建两个数据库: maas_manager、maas_core

分别执行初始化数据脚本: maas-manager.sql, maas-core.sql

3. 使用加密工具获取数据库密码密文:

```
cd /opt/acp/maas
./bin/maas.sh generate <password>
```

从返回结果中复制密文：

```
14:57:55.41 java ops ---> -Xms512m -Xmx2048m -XX:MetaspaceSize=512m
-XX:MaxMetaspaceSize=1024m --add-opens java.base/java.lang=ALL-
UNNAMED --add-opens java.base/java.lang.reflect=ALL-UNNAMED -
XX:ParallelGCThreads=4 -XX:+HeapDumpOnOutOfMemoryError -
Dfile.encoding=utf-8
14:57:55.43 INFO ==> 加密密码为： 1
K777f1b3GSiiJZ/Qw0isYzRZzfAza7601o7I3Aic2gILj23j7DI7v8fk9EdNFaJcSg32Zst
```

4.修改maas-manager配置文件

```
vi /opt/acp/maas/conf/application-prod.properties
```

修改数据库连接配置（以MySQL为例）：

```
apusic.datasource.host=localhost
apusic.datasource.port=3306
apusic.datasource.password=S+Bq/bvr23MecW6BhiyDkPW21F8JiIyirrdNBsoefvM
apusic.datasource.username=root
apusic.datasource.database=maas_manager
```

5.修改maas-core配置文件

```
vi /opt/acp/maas/conf/maas-core/application-prod.properties
```

修改数据库连接配置（以MySQL为例）：

```
spring.datasource.url=jdbc:mysql://localhost:3306/maas_core?
useUnicode=true&characterEncoding=utf8&useSSL=false&useLegacyDatettimeCo
```

```
spring.datasource.username=root  
spring.datasource.password=S+Bq/bvr23MecW6BhiyDkPW21F8JiIyirrdNBsoefvMv
```

修改缓存连接配置：

```
# redis  
spring.redis.timeout=3600  
spring.redis.host=172.24.4.162  
spring.redis.port=6379  
spring.redis.password=
```

修改存储服务地址，改为实际ip

```
package.url=http://172.24.4.162:9010
```

6.执行启动脚本

```
cd /opt/acp/maas/bin  
./maas.sh start all -d
```

出现如下结果，所有模块均为success，表示安装成功：

```
[root@linux-4-162 bin]# ./maas.sh start all -d
16:17:38.48 java ops --> -Xms512m -Xmx512m -XX:MetaspaceSize=512m -XX:MaxMetaspaceSize=1024m -XX:
ryError -Dfile.encoding=utf-8 --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java
16:17:38.48 后台启动 ...
16:17:38.50 nginx pid
16:17:38.50 INFO ==> nginx start success ...
16:17:38.50 jar name alarm-manager-1.2-SNAPSHOT.jar
16:17:38.51 start maas application alarm-manager-1.2-SNAPSHOT.jar
16:17:38.51 INFO ==> 不包含maas
16:17:38.52 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:38.52 start alarm-manager-1.2-SNAPSHOT.jar success ...
16:17:38.53 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.53 jar name aump-admin-2.0.jar
16:17:38.53 start maas application aump-admin-2.0.jar
16:17:38.54 INFO ==> 不包含maas
16:17:38.55 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:38.55 start aump-admin-2.0.jar success ...
16:17:38.56 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.56 jar name cloudlog-0.0.1-SNAPSHOT.jar
16:17:38.57 start maas application cloudlog-0.0.1-SNAPSHOT.jar
16:17:38.57 INFO ==> 不包含maas
16:17:38.58 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:38.58 start cloudlog-0.0.1-SNAPSHOT.jar success ...
16:17:38.59 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.59 jar name maas-authentication-0.0.2-SNAPSHOT.jar
16:17:38.60 start maas application maas-authentication-0.0.2-SNAPSHOT.jar
16:17:38.61 INFO ==> 包含maas
16:17:38.61 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/application-prod.properties
16:17:38.62 start maas-authentication-0.0.2-SNAPSHOT.jar success ...
16:17:38.63 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.63 jar name maas-manager-0.0.2-SNAPSHOT.jar
16:17:38.64 start maas application maas-manager-0.0.2-SNAPSHOT.jar
16:17:38.65 INFO ==> 包含maas
16:17:38.65 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/application-prod.properties
16:17:38.66 start maas-manager-0.0.2-SNAPSHOT.jar success ...
16:17:38.67 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.69 jar name maas-monitor-0.0.2-SNAPSHOT.jar
16:17:38.70 start maas application maas-monitor-0.0.2-SNAPSHOT.jar
16:17:38.72 INFO ==> 包含maas
16:17:38.74 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/application-prod.properties
16:17:38.76 start maas-monitor-0.0.2-SNAPSHOT.jar success ...
16:17:38.78 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.80 jar name workorder-0.0.1-SNAPSHOT.jar
16:17:38.85 start maas application workorder-0.0.1-SNAPSHOT.jar
16:17:38.93 INFO ==> 不包含maas
16:17:39.00 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:39.06 start workorder-0.0.1-SNAPSHOT.jar success ...
16:17:39.08 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
```

7.验证

访问ACP云门户http://192.168.0.1/* (ip为实际部署服务器ip) *, 显示如下页面则表示安装成功。



4.1.4 安装指标监控服务

1. 解压安装包

```
tar -xzvf acp-metrics-server-v2.37.linux-amd64.tar.gz -C /opt/acp
```

2. 执行安装脚本

```
cd /opt/acp/metrics-server/bin
./install.sh --acp.monitor.url http://127.0.0.1:9997 --acp.alarm.url
http://127.0.0.1:9016
```

安装参数说明:

- --acp.monitor.url: 监控中心地址
- --acp.alarm.url: 告警中心地址

出现如下结果, 表示安装成功:

```

[root@linux-4-162 bin]# ./install.sh --acp.monitor.url http://127.0.0.1:9997 --acp.alarm.url http://127.0.0.1:9016
Not specified node! Default:127.0.0.1

install prometheus

install alertmanager

install guance manager

-----
start guance-manager
start guance-manager success!
guance-manager port is 8080
guance-manager PID 31186
start guance-manager end
-----
start prometheus
start prometheus success!
prometheus port is 9090
prometheus PID 31210
start prometheus end
-----
start alertmanager
start alertmanager success!
alertmanager port is 9093
alertmanager PID 31236
start alertmanager end
-----

```

4.1.5 安装管控代理

(本步骤仅在被管控服务器上安装)

1.解压安装包

```

mkdir -p /opt/acp
tar -xzf acp-maas-agent-v8.0.linux-amd64.tar.gz -C /opt/acp

```

2.执行安装脚本

```

cd /opt/acp/maas-agent/bin
./install.sh --acp.logstash 192.168.0.1:5044

```

安装参数说明:

- --acp.logstash: 日志服务节点, 端口默认未5044

出现如下结果, 所有模块均为success, 表示安装成功:

```
#logging:
logging.level: warning
logging.to_files: true
logging.to_syslog: false
logging.files:
  path: /opt/acp/maas-agent/logs
  name: filebeat.log
  keepfiles: 7
install aump-agent

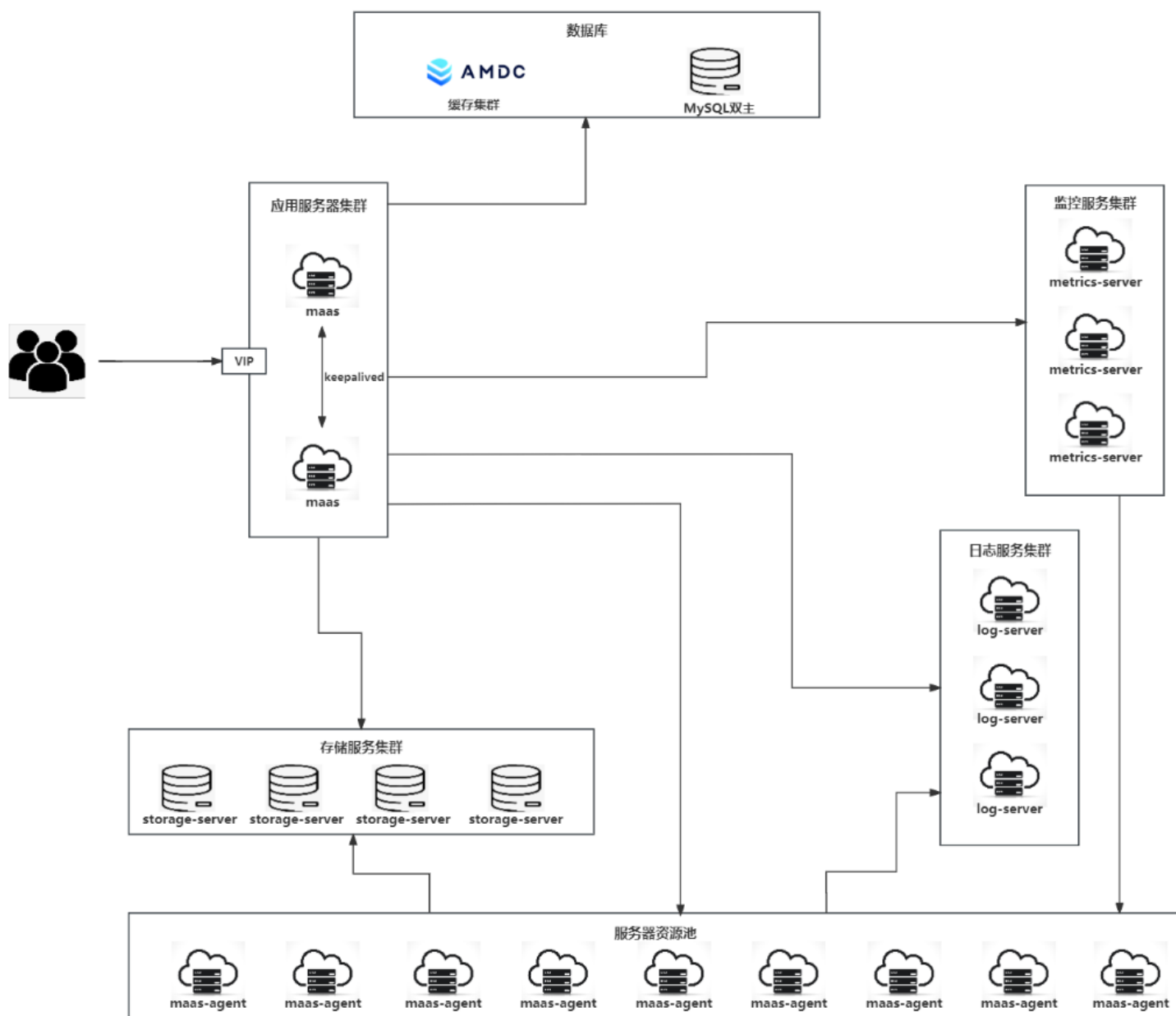
Created symlink from /etc/systemd/system/multi-user.target.wants/aump-agent.service to /etc/systemd/
install mc

-----
start aump-agent
start aump-agent success!
aump-agent PID 3147
-----

start filebeat
start filebeat success!
filebeat PID 3171
-----
```

4.2 集群部署

ACP提供集群模式，保障服务和数据的可用性。ACP集群包含应用服务器集群（maas）、监控服务集群、日志服务集群、存储服务集群。同时需要依赖反向代理集群、注册中心集群来支撑内外部服务访问的可用性。



4.2.1 安装存储服务集群

(以下操作在每个节点上都要执行)

1. 解压安装包

```
mkdir -p /opt/acp
tar -xzvf acp-storage-server-v1.0.linux-amd64.tar.gz -C /opt/acp
```

2. 执行安装脚本

```
cd /opt/acp/storage-server/bin
./install.sh --data.path /home --nodes
192.168.0.1,192.168.0.2,192.168.0.3,191.168.0.4
```

安装参数说明:

- --data.path: 数据存储路径。通过 `df -h`命令获取剩余空间充足且非root的分区。如下图, data.path为/home

```
[root@linux-4-162 bin]# ./install.sh --data.path /home
Archive: /opt/acp/storage-server/pkgs/minio-amd64.zip
  creating: /opt/acp/storage-server/runtime/minio/
  inflating: /opt/acp/storage-server/runtime/minio/initial_aump_package.sh
  inflating: /opt/acp/storage-server/runtime/minio/mc
  inflating: /opt/acp/storage-server/runtime/minio/minio
  inflating: /opt/acp/storage-server/runtime/minio/start.sh
  inflating: /opt/acp/storage-server/runtime/minio/startstandalone.sh
  inflating: /opt/acp/storage-server/runtime/minio/stop.sh
-----
start minio
start minio success!
-----
```

- --nodes: 集群节点列表

3.检查安装结果

```
./status.sh
```

出现如下结果, 表示安装成功:

```
[root@linux-4-162 bin]# ./status.sh
-----
check minio
minio is running!
minio PID 25278
-----
```

4.2.2 安装日志服务集群

(本服务依赖Java环境, 请先安装JDK17; 以下操作在每个节点上都要执行)

1.解压安装包

```
mkdir -p /opt/acp
tar -xzvf acp-log-server-v7.12.linux-amd64.tar.gz -C /opt/acp
```

2. 执行安装脚本

```
cd /opt/acp/log-server/bin
./install.sh
```

3. 检查安装结果

```
./check.sh
```

出现如下结果，表示安装成功：

```
[root@linux-4-162 bin]# ./check.sh
-----
check LogServer status start
please wait
ailp-es status is active!
ailp-es is running!
-----
ailp-logstash status is active!
ailp-logstash is running!
-----
check LogServer status end
-----
```

4.2.3 安装中间件云平台

4.2.3.1 安装maas

(本服务依赖Java环境，请先安装JDK17)

(以下操作在两个节点上都要执行)

1. 解压安装包

```
mkdir -p /opt/acp
tar -xzvf acp-maas-v8.0.linux-amd64.tar.gz -C /opt/acp
```

2. 初始化数据 (仅需执行一次)

创建两个数据库：maas-manager、maas-core

分别执行初始化数据脚本：maas-manager.sql, maas-core.sql

3. 使用加密工具获取密码密文：

```
cd /opt/acp/maas
./bin/maas.sh generate {password}
```

从返回结果中复制密文：

```
14:57:55.41 java ops --> -Xms512m -Xmx2048m -XX:MetaspaceSize=512m
-XX:MaxMetaspaceSize=1024m --add-opens java.base/java.lang=ALL-
UNNAMED --add-opens java.base/java.lang.reflect=ALL-UNNAMED -
XX:ParallelGCThreads=4 -XX:+HeapDumpOnOutOfMemoryError -
Dfile.encoding=utf-8
14:57:55.43 INFO ==> 加密密码为： 1
K777f1b3GSiiJZ/Qw0isYzRZzfAza7601o7I3Aic2gILj23j7DI7v8fk9EdNFaJcSg32Zst
```

4.修改maas-manager配置文件

```
vi /opt/acp/maas/conf/application-prod.properties
```

修改数据库连接配置（以MySQL为例）：

```
apusic.datasource.host=localhost
apusic.datasource.port=3306
apusic.datasource.password=S+Bq/bvr23MecW6BhiyDkPW21F8JiIyirrdNBsoefvM
apusic.datasource.username=root
apusic.datasource.database=maas_manager
```

5.修改maas-core配置文件

```
vi /opt/acp/maas/conf/maas-core/application-prod.properties
```

修改数据库连接配置（以MySQL为例）：

```
spring.datasource.url=jdbc:mysql://localhost:3306/maas_core?
useUnicode=true&characterEncoding=utf8&useSSL=false&useLegacyDatettimeCo
```

```
spring.datasource.username=root  
spring.datasource.password=S+Bq/bvr23MecW6BhiyDkPW21F8JiIyirrdNBsoefvMv
```

修改缓存连接配置：

```
# redis  
spring.redis.timeout=3600  
spring.redis.host=172.24.4.162  
spring.redis.port=6379
```

修改存储服务地址，改为实际ip

```
package.url=http://172.24.4.162:9010
```

6.执行启动脚本

```
cd /opt/acp/maas/bin  
./maas.sh start all -d
```

出现如下结果，所有模块均为success，表示安装成功：

```

[root@linux-4-162 bin]# ./maas.sh start all -d
16:17:38.48 java ops ---> -Xms512m -Xmx512m -XX:MetaspaceSize=512m -XX:MaxMetaspaceSize=1024m -XX:
ryError -Dfile.encoding=utf-8 --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java
16:17:38.48 后台启动 ...
16:17:38.50 nginx pid
16:17:38.50 INFO ==> nginx start success ...
16:17:38.50 jar name alarm-manager-1.2-SNAPSHOT.jar
16:17:38.51 start maas application alarm-manager-1.2-SNAPSHOT.jar
16:17:38.51 INFO ==> 不包含maas
16:17:38.52 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:38.52 start alarm-manager-1.2-SNAPSHOT.jar success ...
16:17:38.53 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.53 jar name aump-admin-2.0.jar
16:17:38.53 start maas application aump-admin-2.0.jar
16:17:38.54 INFO ==> 不包含maas
16:17:38.55 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:38.55 start aump-admin-2.0.jar success ...
16:17:38.56 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.56 jar name cloudlog-0.0.1-SNAPSHOT.jar
16:17:38.57 start maas application cloudlog-0.0.1-SNAPSHOT.jar
16:17:38.57 INFO ==> 不包含maas
16:17:38.58 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:38.58 start cloudlog-0.0.1-SNAPSHOT.jar success ...
16:17:38.59 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.59 jar name maas-authentication-0.0.2-SNAPSHOT.jar
16:17:38.60 start maas application maas-authentication-0.0.2-SNAPSHOT.jar
16:17:38.61 INFO ==> 包含maas
16:17:38.61 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/application-prod.properties
16:17:38.62 start maas-authentication-0.0.2-SNAPSHOT.jar success ...
16:17:38.63 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.63 jar name maas-manager-0.0.2-SNAPSHOT.jar
16:17:38.64 start maas application maas-manager-0.0.2-SNAPSHOT.jar
16:17:38.65 INFO ==> 包含maas
16:17:38.65 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/application-prod.properties
16:17:38.66 start maas-manager-0.0.2-SNAPSHOT.jar success ...
16:17:38.67 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.69 jar name maas-monitor-0.0.2-SNAPSHOT.jar
16:17:38.70 start maas application maas-monitor-0.0.2-SNAPSHOT.jar
16:17:38.72 INFO ==> 包含maas
16:17:38.74 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/application-prod.properties
16:17:38.76 start maas-monitor-0.0.2-SNAPSHOT.jar success ...
16:17:38.78 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.80 jar name workorder-0.0.1-SNAPSHOT.jar
16:17:38.85 start maas application workorder-0.0.1-SNAPSHOT.jar
16:17:38.93 INFO ==> 不包含maas
16:17:39.00 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:39.06 start workorder-0.0.1-SNAPSHOT.jar success ...
16:17:39.08 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid

```

通过keepalived实现maas的高可用，提供外部虚拟ip进行访问，若主maas宕机失败后，将切换到从maas，从而实现maas的高可用

4.2.3.2 安装高可用keepalived

(ACP不提供keepalived安装包，请自行获取；下面步骤以yum方式安装，供参考)

1.安装Keepalived

```
yum install -y keepalived
```

2.新建检查maas状态的脚本

```
vi /opt/acp/check_maas.sh
```

```
if [ $(ps -C nginx --no-header |wc -l) -eq 0 ];then
    /usr/local/nginx/sbin/nginx #启动nginx
    sleep 2 #等待nginx完全启动
fi
if [ $(ps -C nginx --no-header |wc -l) -eq 0 ];then
    killall keepalived
fi
```

修改检查文本为可执行文件。

```
chmod +x /opt/acp/check_maas.sh
```

3.编辑keepalived的配置文件，在/etc/keepalived目录下添加keepalived.conf，若存在，直接进行修改。

```
vi /etc/keepalived/keepalived.conf
```

添加主节点的keepalived配置

```
global_defs {
    router_id maas_01 #名称
}

vrrp_script check_maas {
    script "/opt/acp/check_maas.sh"
    interval 2 #每2秒检测
    一次nginx的运行状态
    weight -20 #每失败一次，优
    优先级减少20
}
```

```

vrp_instance vrrptest {
    state MASTER
    interface ens160                #选择使用的网卡, 保持一致
    virtual_router_id 100           #分组标记
    mcast_src_ip 172.24.4.110      #本地实际的ip
    priority 150                    #优先级, 主节点比从
    节点的值大一些
    advert_int 1                    #两台服务器的心跳间
    隔
    authentication {
        auth_type PASS
        auth_pass 1111
    }

    track_script {                  #检查nginx的脚本,
    和上面的script check_maas脚本结合使用
        check_maas
    }

    virtual_ipaddress {             #两台服务器共用的虚拟vip
        172.24.4.166
    }
}

```

4.从节点的keepalived的配置基本不变,修改router_id名称, mcast_src_ip本地实际的ip, priority比master小一些, state修改为BACKUP。

```

global_defs {
    router_id maas_01              #名称
}

vrrp_script check_maas {
    script "/opt/acp/check_maas.sh"
    interval 2                      #每2秒检测
    一次nginx的运行状态
    weight -20                       #每失败一次, 优
    先级减少20
}

```

```

vrrp_instance vrrptest {
    state BACKUP
    interface ens160                #选择使用的网卡, 保持一致
    virtual_router_id 100           #分组标记
    mcast_src_ip 172.24.4.110      #本地实际的ip
    priority 100                    #优先级, 主节点比从
    advert_int 1                    #两台服务器的心跳间
    authentication {
        auth_type PASS
        auth_pass 1111
    }

    track_script {                  #检查nginx的脚本,
    和上面的script check_maas脚本结合使用
        check_maas
    }
    virtual_ipaddress {             #两台服务器共用的虚拟vip
        172.24.4.166
    }
}

```

5.关闭防火墙, 从主节点启动keepalived

```
service keepalived start
```

6.验证

访问ACP云门户http://192.168.0.1/* (ip为实际vip) *, 显示如下页面则表示安装成功。



4.2.4 安装监控服务集群

(以下操作在每个节点上都要执行)

1. 解压安装包

```
mkdir -p /opt/acp
tar -xzvf acp-metrics-server-v2.37.linux-amd64.tar.gz -C /opt/acp
```

2. 执行安装脚本

```
cd /opt/acp/metrics-server/bin
./install.sh --acp.monitor.url http://localhost:9997 --acp.alarm.url
http://localhost:9016 --cluster.nodes
192.168.0.1,192.168.0.2,192.168.0.3
```

安装参数说明:

- --acp.monitor.url: 监控中心地址
- --acp.alarm.url: 告警中心地址
- --cluster.nodes: 集群节点列表

出现如下结果，表示安装成功：

```
[root@linux-4-162 bin]# ./install.sh --acp.monitor.url http://127.0.0.1:9997 --acp.alarm.url http://127.0.0.1:9016
Not specified node! Default:127.0.0.1

install prometheus

install alertmanager

install guance manager

-----
start guance-manager
start guance-manager success!
guance-manager port is 8080
guance-manager PID 31186
start guance-manager end
-----
start prometheus
start prometheus success!
prometheus port is 9090
prometheus PID 31210
start prometheus end
-----
start alertmanager
start alertmanager success!
alertmanager port is 9093
alertmanager PID 31236
start alertmanager end
-----
```

5 产品卸载

5.1 卸载指标监控

1. 执行卸载脚本

```
cd /opt/acp/metrics-server/bin
./uninstall.sh
```

出现如下结果，每个模块卸载都是success，则表示卸载成功：

```
[root@linux-4-162 bin]# ./uninstall.sh
-----
stop guance-manager
guance-manager PID 6449
kill -9 6449
stop guance-manager success!
stop guance-manager end
-----
stop prometheus
prometheus PID 6473
kill -9 6473
stop prometheus success!
stop prometheus end
-----
stop alertmanager
alertmanager PID 6499
kill -9 6499
stop alertmanager success!
stop alertmanager end
-----
remove install files
```

2. 移除安装目录

```
rm -rf /opt/acp/metrics-server
```

5.2 卸载日志服务

1. 执行卸载脚本

```
cd /opt/acp/log-server/bin
./uninstall.sh
```

出现如下结果，每个模块卸载都是success，则表示卸载成功：

```
[root@linux-4-162 bin]# ./uninstall.sh
-----
uninstall LogServer start
uninstalling, please wait
install_path: /opt/acp/log-server/runtime
-----
uninstall ailp-es start
uninstalling, please wait
ailp-es is running!
Removed symlink /etc/systemd/system/multi-user.target.wants/ailp-es.service.
Failed to reset failed state of unit ailp-es.service: Unit ailp-es.service is not loaded.
stop ailp-es success!
delete ailp-es is success!
uninstall ailp-es is success!
uninstall ailp-es end
-----
uninstall ailp-logstash start
uninstalling, please wait
ailp-logstash is running!
Removed symlink /etc/systemd/system/multi-user.target.wants/ailp-logstash.service.
Failed to reset failed state of unit ailp-logstash.service: Unit ailp-logstash.service is not loaded.
stop ailp-logstash success!
delete ailp-logstash is success!
uninstall ailp-logstash is success!
uninstall ailp-logstash end
-----
uninstall LogServer end
-----
```

2. 移除安装目录

```
rm -rf /opt/acp/log-server
```

5.3 卸载存储服务

1. 执行卸载脚本

```
cd /opt/acp/storage-server/bin
./uninstall.sh
```

出现如下结果，每个模块卸载都是success，则表示卸载成功：

```
[root@linux-4-162 bin]# ./uninstall.sh
-----
stop minio
minio PID 9586
kill -9 9586
stop minio success!
-----
```

2. 移除安装目录

```
rm -rf /opt/storage-server
```

5.4 卸载中间件云平台

1. 停止中间件云平台

```
cd /opt/acp/maas/bin
./maas.sh stop all
```

```
[root@linux-4-162 bin]# ./maas.sh stop all
16:19:56.51 java ops ---> -Xms512m -Xmx512m -XX:MetaspaceSize=512m -XX:MaxMetaspaceSize=1024m -XX:
ryError -Dfile.encoding=utf-8 --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java
16:19:56.59 nginx pid
16:19:56.63 INFO ==> nginx stop success ...
16:19:56.71 close alarm-manager-1.2-SNAPSHOT.jar success
16:19:56.76 close aump-admin-2.0.jar success
16:19:56.79 close cloudlog-0.0.1-SNAPSHOT.jar success
16:19:56.81 close maas-authentication-0.0.2-SNAPSHOT.jar success
16:19:56.85 close maas-manager-0.0.2-SNAPSHOT.jar success
16:19:56.85 close maas-monitor-0.0.2-SNAPSHOT.jar success
16:19:56.86 close workorder-0.0.1-SNAPSHOT.jar success
```

2. 移除安装目录

```
rm -rf /opt/acp/maas
```

5.5 卸载管控代理

1. 执行卸载脚本

```
cd /opt/acp/maas-agent/bin
./uninstall.sh
```

出现如下结果，每个模块的启动都是success，则表示启动成功：

```
[root@linux-4-162 bin]# ./uninstall.sh
-----
stop aump-agent
stop aump-agent success!
-----
-----
stop filebeat
stop filebeat success!
-----
uninstall filebeat
Removed symlink /etc/systemd/system/multi-user.target.wants/filebeat.service.
warning: /etc/filebeat/filebeat.yml saved as /etc/filebeat/filebeat.yml.rpmsave
uninstall aump-agent
Removed symlink /etc/systemd/system/multi-user.target.wants/aump-agent.service.
```

2. 移除安装目录

```
rm -rf /opt/acp/maas-agent
```

6 其他操作

6.1 指标监控

指标监控目前提供启动、停止、查看状态等操作。

6.1.1 启动

```
cd /opt/acp/metrics-server/bin
./start.sh
```

出现如下结果，每个模块的启动都是success，则表示启动成功：

```
[root@linux-4-162 bin]# ./start.sh
-----
start guance-manager
start guance-manager success!
guance-manager port is 8080
guance-manager PID 6449
start guance-manager end
-----
start prometheus
start prometheus success!
prometheus port is 9090
prometheus PID 6473
start prometheus end
-----
start alertmanager
start alertmanager success!
alertmanager port is 9093
alertmanager PID 6499
start alertmanager end
-----
```

6.1.2 停止

```
cd /opt/acp/metrics-server/bin
./stop.sh
```

返回如下结果，每个模块的停止都是success，则表示停止成功

```
[root@linux-4-162 bin]# ./stop.sh
-----
stop guance-manager
guance-manager PID 3238
kill -9 3238
stop guance-manager success!
stop guance-manager end
-----
stop prometheus
prometheus PID 3267
kill -9 3267
stop prometheus success!
stop prometheus end
-----
stop alertmanager
alertmanager PID 3291
kill -9 3291
stop alertmanager success!
stop alertmanager end
-----
```

6.1.3 查看状态

```
cd /opt/acp/metrics-server/bin
./status.sh
```

返回结果如下：

```
-----
check guance-manager running status
guance-manager is running!
guance-manager PID 3238
-----
check prometheus running status
prometheus is running!
prometheus PID 3267
-----
check alertmanager running status
alertmanager is running!
alertmanager PID 3291
-----
```

6.2 日志服务

日志服务目前提供启动、停止、查看状态等操作。

6.2.1 启动

```
cd /opt/acp/log-server/bin
./start.sh
```

出现如下结果，每个模块的启动都是success，则表示启动成功：

```
[root@linux-4-162 bin]# ./start.sh
-----
start LogServer start
starting, please wait
start ailp-es success
start ailp-logstash success
start LogServer end
-----
```

6.2.2 停止

```
cd /opt/acp/log-server/bin
./stop.sh
```

返回如下结果，每个模块的停止都是success，则表示停止成功

```
[root@linux-4-162 bin]# ./stop.sh
-----
stop LogServer start
stopping, please wait
stop ailp-es success
stop ailp-logstash success
stop LogServer end
-----
```

6.2.3 查看状态

```
cd /opt/acp/log-server/bin
./check.sh
```

返回结果如下：

```
[root@linux-4-162 bin]# ./check.sh
-----
check LogServer status start
please wait
ailp-es status is active!
ailp-es is running!
-----
ailp-logstash status is active!
ailp-logstash is running!
-----
check LogServer status end
-----
```

6.3 存储服务

存储服务目前提供启动、停止、查看状态等操作。

6.3.1 启动

```
cd /opt/acp/storage-server/bin
./start.sh
```

出现如下结果，每个模块的启动都是success，则表示启动成功：

```
[root@linux-4-162 bin]# ./start.sh
-----
start minio
start minio success!
-----
```

6.3.2 停止

```
cd /opt/acp/storage-server/bin
./stop.sh
```

返回如下结果，每个模块的停止都是success，则表示停止成功

```
[root@linux-4-162 bin]# ./stop.sh
-----
stop minio
minio PID 25278
kill -9 25278
stop minio success!
-----
```

6.3.3 查看状态

```
cd /opt/acp/storage-server/bin
./status.sh
```

返回结果如下:

```
[root@linux-4-162 bin]# ./status.sh
-----
check minio
minio is running!
minio PID 25278
-----
```

6.4 中间件云平台

中间件云平台目前提供启动、停止等操作。

6.4.1 启动

```
cd /opt/acp/maas/bin
./maas.sh start all -d
```

出现如下结果，每个模块的启动都是success，则表示启动成功：

```

[root@linux-4-162 bin]# ./maas.sh start all -d
16:17:38.48 java ops ---> -Xms512m -Xmx512m -XX:MetaspaceSize=512m -XX:MaxMetaspaceSize=1024m -XX:
ryError -Dfile.encoding=utf-8 --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java
16:17:38.48 后台启动 ...
16:17:38.50 nginx pid
16:17:38.50 INFO ==> nginx start success ...
16:17:38.50 jar name alarm-manager-1.2-SNAPSHOT.jar
16:17:38.51 start maas application alarm-manager-1.2-SNAPSHOT.jar
16:17:38.51 INFO ==> 不包含maas
16:17:38.52 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:38.52 start alarm-manager-1.2-SNAPSHOT.jar success ...
16:17:38.53 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.53 jar name aump-admin-2.0.jar
16:17:38.53 start maas application aump-admin-2.0.jar
16:17:38.54 INFO ==> 不包含maas
16:17:38.55 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:38.55 start aump-admin-2.0.jar success ...
16:17:38.56 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.56 jar name cloudlog-0.0.1-SNAPSHOT.jar
16:17:38.57 start maas application cloudlog-0.0.1-SNAPSHOT.jar
16:17:38.57 INFO ==> 不包含maas
16:17:38.58 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:38.58 start cloudlog-0.0.1-SNAPSHOT.jar success ...
16:17:38.59 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.59 jar name maas-authentication-0.0.2-SNAPSHOT.jar
16:17:38.60 start maas application maas-authentication-0.0.2-SNAPSHOT.jar
16:17:38.61 INFO ==> 包含maas
16:17:38.61 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/application-prod.properties
16:17:38.62 start maas-authentication-0.0.2-SNAPSHOT.jar success ...
16:17:38.63 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.63 jar name maas-manager-0.0.2-SNAPSHOT.jar
16:17:38.64 start maas application maas-manager-0.0.2-SNAPSHOT.jar
16:17:38.65 INFO ==> 包含maas
16:17:38.65 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/application-prod.properties
16:17:38.66 start maas-manager-0.0.2-SNAPSHOT.jar success ...
16:17:38.67 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.69 jar name maas-monitor-0.0.2-SNAPSHOT.jar
16:17:38.70 start maas application maas-monitor-0.0.2-SNAPSHOT.jar
16:17:38.72 INFO ==> 包含maas
16:17:38.74 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/application-prod.properties
16:17:38.76 start maas-monitor-0.0.2-SNAPSHOT.jar success ...
16:17:38.78 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid
16:17:38.80 jar name workorder-0.0.1-SNAPSHOT.jar
16:17:38.85 start maas application workorder-0.0.1-SNAPSHOT.jar
16:17:38.93 INFO ==> 不包含maas
16:17:39.00 启动配置文件为: /opt/acp/maas-0.0.2-SNAPSHOT/maas/conf/amp/application-prod.properties
16:17:39.06 start workorder-0.0.1-SNAPSHOT.jar success ...
16:17:39.08 /opt/acp/maas-0.0.2-SNAPSHOT/maas/.pid

```

6.4.2 停止

```

cd /opt/acp/maas/bin
./maas.sh stop all

```

返回如下结果，每个模块的停止都是success，则表示停止成功

```
[root@linux-4-162 bin]# ./maas.sh stop all
16:19:56.51 java ops ---> -Xms512m -Xmx512m -XX:MetaspaceSize=512m -XX:MaxMetaspaceSize=1024m -XX:
ryError -Dfile.encoding=utf-8 --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java
16:19:56.59 nginx pid
16:19:56.63 INFO ==> nginx stop success ...
16:19:56.71 close alarm-manager-1.2-SNAPSHOT.jar success
16:19:56.76 close aump-admin-2.0.jar success
16:19:56.79 close cloudlog-0.0.1-SNAPSHOT.jar success
16:19:56.81 close maas-authentication-0.0.2-SNAPSHOT.jar success
16:19:56.85 close maas-manager-0.0.2-SNAPSHOT.jar success
16:19:56.85 close maas-monitor-0.0.2-SNAPSHOT.jar success
16:19:56.86 close workorder-0.0.1-SNAPSHOT.jar success
```

6.5 管控代理

中间件云平台目前提供启动、停止等操作。

6.5.1 启动

```
cd /opt/acp/maas-agent/bin
./start.sh
```

出现如下结果，所有模块启动都是success，则表示启动成功：

```
[root@linux-4-162 bin]# ./start.sh
-----
start aump-agent
start aump-agent success!
aump-agent PID 16585
-----
start filebeat
start filebeat success!
filebeat PID 16609
-----
```

6.5.2 停止

```
cd /opt/acp/maas-agent/bin
./stop.sh
```

返回如下结果，所有模块停止都是success，则表示停止成功

```
[root@linux-4-162 bin]# ./stop.sh
-----
stop aump-agent
stop aump-agent success!
-----
-----
stop filebeat
stop filebeat success!
-----
```

6.5.3 查看状态

```
cd /opt/acp/maas-agent/bin
./status.sh
```

返回结果如下:

```
[root@linux-4-162 bin]# ./status.sh
-----
check aump-agent
aump-agent is running!
aump-agent PID 16585
-----
-----
check filebeat
filebeat is running!
filebeat PID 16609
-----
```

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

