Apusic 金蝶天燕

生成时间：2026-05-19

安装手册

金蝶Apusic智能日志平台v2.0

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 前言

1.1 产品简介

金蝶Apusic智能日志平台(简称：AILP)是一个通用的日志大数据平台，可以使用各种开源的日志收集工具将日志统一上传，并根据预先定义的解析规则将日志数据结构化存储，提供准实时的搜索和仪表盘对日志进行后续的分析处理。

典型的日志数据包括：

- Linux系统日志
- Apache Web服务器日志
- Nginx Web服务器日志
- 中间件日志
- 数据库日志
- JSON日志
- 其他任意文件型日志

1.2 范围和读者

本手册介绍AILPV2.0使用详细说明，适用于该产品的使用用户，产品技术顾问，产品维护人员，以及希望学习了解AILP日志平台的相关人员。

1.3 约定与术语

一些约定的缩略词诠释：

- AMP

金蝶Apusic监控平台

- ElasticSearch

ElasticSearch是一个基于Lucene的搜索服务器

- ZooKeeper

ZooKeeper是一个分布式的，开放源码的分布式应用程序协调服务。

- Kafka

Kafka是由Apache软件基金会开发的一个开源流处理平台

- Logstash

Logstash是一个开源数据收集引擎

- Filebeat

Filebeat是用于转发和集中日志数据的轻量级传送工具

2 安装环境要求

2.1 配置要求

安装日志服务产品的最低配置要求见下表

表2-1 软件及操作系统环境要求

资源环境	要求
操作系统	Linux Red Hat 5.2或以上(及其他Kernel 2.25或以上linux版本)
CPU	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz或以上
MySQL	5.6或以上
Redis	3.0或以上
内存	16G或以上
硬盘	可用空间1T或以上
浏览器	FireFox 21及以上、Chrome 23及以上、IE 10及以上

2.2 推荐配置

安装日志服务产品的推荐的配置见下表。

表2-2 软件及操作系统环境推荐配置

资源环境	要求
操作系统	Linux
CPU	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz或以上
MySQL	5.6或以上
Redis	5.0或以上
内存	16G或以上
硬盘	可用空间1T或以上
浏览器	Chrome 60及以上、FireFox 21及以上

3 安装Web控制台

如果您已经安装AMP平台产品，则已经默认包含了Web控制台，您无需重复安装。请跳过本章节，直接参照第4章的说明，安装日志服务并注册到平台，即可使用AILP产品。

如果您需要部署独立的AILP产品，请按照本章安装指引说明进行Web控制台的安装。

3.1 产品介质说明

Web控制台组件相关安装介质如下

表3-1 Web控制台产品介质

| 组件名称 | 文件名 | 说明 | | --- | --- | --- | | Web控制台 | amp-console-prod-xxx.tar.gz | Web统一控制台，SpringBoot应用 |

3.2 安装Web控制台

3.2.1 安装准备

- JDK

Web控制台应用运行需要JDK8 环境，参考附录6.1节安装说明

- Redis

Web控制台运行需要Redis缓存服务，参考附录6.2节安装说明。

- MySQL数据库

系统默认推荐使用MySQL，参考附录6.3节MySQL的安装说明，如已经安装请跳过。如果你使用其他类型的数据库，请参考对应厂商说明帮助手册进行安装。

3.2.2 安装说明

创建AILP产品安装根目录，指定${PATH} 为实际路径，将amp-console-prod-xxx.tar.gz解压到对应目录及完成产品包安装，/${PATH}/AMP/amp-console为产品解压后的目录。

```
# mkdir -p /${PATH}/AMP
# tar -zxvf amp-console-prod-xxx.tar.gz -C /${PATH}/AMP
```

如上，即完成Web控制台应用的解压工作，接下来修改相关参数配置。

3.2.3 配置参数

1. 修改amp-console/conf/application.yml文件，该配置文件为SpringBoot应用的默认配置文件，active的值为prod，其对应生效的文件是application-prod.yml,采用的是MySQL数据库连接配置。

系统中提供如下可选的配置文件

表3-2 Web控制台应用配置文件

文件名	说明
application-dev.yml	H2数据库作为持数据持久化存储的配置，开发环境阶段使用，生产环境不建议使用
application-prod.yml	MySQL数据库作为数据持久化存储的配置，默认使用用该文件
application-samle-dm.yml	达梦数据库作为数据持久化存储的配置
application-sample-gbase8s.yml	南大通用Gbase8s作为数据持久化存储的配置
application-sample-kingbasees.yml	人大进仓KingbaseES作为数据持久化存储的配置
application-sample-shentong.yml	神舟通用数据库作为数据持久化存储的配置

2. 用户可根据实际部署环境修改application.yml文件中的active值为prod、sample-kingbasees、sample-dm、sample-shentong、sample-gbase-8s来切换不同环境的配置。

下面以采用MySQL配置的application-prod.yml文件为例，说明相关主要参数配置。

```
vi /${PATH}/AMP/amp-console/conf/application-prod.yml
```

- server.port 参数指定了该web应用的默认端口，默认值为9000
- spring.redis 指定了应用连接Redis相关配置，需要根据实际部署环境进行修改。
    - timeout 为超时时间，默认3600s
    - host为redis的ip地址，默认值localhost
    - port为redis端口，默认值6379
    - password 为redis连接密码
- spring.datasource 为数据库连接配置， 需要根据实际部署环境进行修改。
    - url为数据库JDBC连接配置，包含数据库地址、端口、数据库名称等参数
    - Username 指定数据库连接用户名
    - Password 指定数据库连接密码

配置参考样例如下：

```
server:
  port: 9000
  servlet:
context-path: /
  redis:
    timeout: 3600
    host: localhost
    port: 6379
    password: root
datasource:
type: com.zaxxer.hikari.HikariDataSource
    url: jdbc:mysql://localhost:3306/amp_console?
useUnicode=true&characterEncoding=utf8&useSSL=false&useLegacyDatetimeCode=false&serverTimezone=UTC
    username: root
    password: root
  <省略其他
```

3.2.4 初始化数据库

登录mysql数据库。

```
mysql -u username -p password
```

执行amp-console/sql/mysql目录下的create.sql数据库创建脚本文件，initial.sql数据库初始化脚本文件，初始化日志服务数据库。

```
create database amp_console;
use amp_console;
```

```
source /${PATH}/AMP/amp_console/sql/mysql/create.sql;
source /${PATH}/AMP/amp_console/sql/mysql/initial.sql;
exit
```

3.3 安装后工作

3.3.1 了解产品目录结构

表3-3 amp-console控制台目录结构

目录	包含
bin	控制台组件的启动脚本。
boot	控制台程序的jar文件。
conf	一些配置文件。
lib	应用程序依赖的一些jar包。
sql	控制台对应的amp_console数据库多种版本的sql创建及初始化脚本文件。
HELP.md	帮助文档，对控制台项目的补充说明。

3.3.2 启动Web控制台

1. 修改完amp-console的配置文件后，后台启动Web控制台。

```
nohup /${PATH}/AMP/amp-console/bin/startup.sh &
```

2. 查看Web控制台运行状态，若端口9000存在，表示启动成功。

```
netstat -lntp | grep 9000
```

3. 访问浏览器验证：http://amp-console_ip:9000,出现如下图登录页面，输入用户名：admin和密码：admin，登录成功，则表明部署成功。

图3-1控制台登录页

3.3.3 停止运行服务

目前可以根据端口号查找出该应用程序的进程，使用kill命令终止Web控制台进程。

```
# netstat -lntp | grep 9000
tcp6       0      0 :::9000       :::*            LISTEN      19358/java
# kill -9 19358
```

3.3.4 卸载服务

删除AILP安装部署目录，即可完成卸载控制台操作。

```
rm -rf /${PATH}/AMP/amp-console
```

4 安装日志服务

4.1 产品介质说明

日志服务产品相关安装介质文件

表4-1 日志服务相关产品组件

| 组件名称 | 文件名 | 说明 | | --- | --- | --- | | 日志服务 | tar -zxvf ailp-v2.0-amd64.tar.gz | 日志服务组件，包含cloudlog服务、elasticsearch、jdk、kafka以及logstash_backend |

4.2 安装前准备

安装前需要为cloudlog创建用户、清除防火墙规则、修改系统文件使用限制以及修改内存与网络设置，这些都需切换root用户操作。

4.2.1 创建用户

```
useradd   adp   ##创建用户adp
passwd   adp   ##设置密码
groupadd  adp   ##创建用户组
```

4.2.2 清除默认的防火墙规则

如果是Centos/RedHat，需要清除默认的防火墙规则

```
# iptables -P INPUT ACCEPT
# iptables -F
# iptables -X
# iptables -Z
##检查确认
# iptables -L -n
```

4.2.3 修改文件使用限制

下面操作硬件与软件的数量，如果有设置，则不进行修改

```
# vim  /etc/security/limits.conf
```

```
* soft nofile 65536
* hard nofile 65536
hadoop soft nofile 65536
hadoop hard nofile 131072
hadoop soft nproc 2048
hadoop hard nproc 4096
adp soft nproc 10240
adp hard nproc 10240
```

4.2.4 修改虚拟内存和网络设置

增加如下配置（如果以前有相关的配置，则进行修改）

```
# vim  /etc/sysctl.conf
```

```
vm.max_map_count = 655360
vm.swappiness = 1
vm.dirty_backgroud_ratio = 5
vm.dirty_ratio = 60
net.core.wmem_default = 131072
net.core.rmem_default = 131072
net.core.wmem_max = 2097152
net.core.rmem_max = 2097152
net.ipv4.tcp_wmem = 4096 65536 2048000
net.ipv4.tcp_rmem = 4096 65536 2048000
net.ipv4.tcp_window_scaling = 1
net.ipv4.tcp_max_syn_backlog = 10240
net.core.somaxconn = 10240
net.core.netdev_max_backlog = 10240
```

使系统控制权限配置生效，执行下面命令

```
#/sbin/sysctl -p
```

4.2.5 配置环境变量

编辑 ~/.bashrc,在文件末尾增加:

```
export JAVA_HOME=/home/adp/cloudlog/jdk
export PATH=$PATH:$JAVA_HOME/bin
export LANG=en_US.UTF-8
export LANGUAGE= en_US.UTF-8
export LC_ALL=en_US.UTF-8
####/home/adp/cloudlog将是后面云日志的安装位置
```

4.3 安装日志服务

4.3.1 安装包说明

本文档以在Linux x86_64环境下为例进行安装过程说明，其他aarch64、mips64等平台安装过程一致，安装过程中选择对应平台的产品包介质进行安装即可，不做重复说明。日志服务安装操作需要用4.1.1创建的用户adp（重新打开一个窗口使用adp用户登录）来进行接下来的产品安装操作。

adp用户上传产品包到/home/adp目录后，解压安装包

```
# tar -zxvf tar -zxvf ailp-v2.0-amd64.tar.gz
```

日志服务产品组件目录为/home/adp/cloudlog，解压后的目录结构如下

```
[adp@cloudlog cloudlog]$ cd  /home/adp/cloudlog
[adp@cloudlog cloudlog]$ ll
drwxrwxrwx  2 adp adp 264 Apr 23 16:17 bin
drwxr-xr-x  2 adp adp  73 Apr 23 16:02 config
drwxr-xr-x  8 adp adp 143 Apr 23 16:11 elasticsearch
drwxr-xr-x  8 adp adp 255 Apr 23 14:38 jdk
drwxr-xr-x  7 adp adp 101 Apr 23 16:12 kafka
drwxr-xr-x  2 adp adp  26 Apr 23 14:38 lib
drwxr-xr-x  2 adp adp  44 Apr 23 16:17 logs
```

```
drwxr-xr-x 12 adp adp 308 Apr 23 16:15 logstash_backend
drwxr-xr-x  4 adp adp  44 Apr 26 16:20 public
```

如果上传软件包是root用户上传，则需要给安装目录赋权

```
# chown -R adp:adp .
```

4.3.2 安装Elasticsearch

- 普通方式安装

  一般使用这种方式安装，es访问时不需要用户名，密码，不开启xpack ssl证书认证。

修改配置文件/home/adp/cloudlog/elasticsearch/config/elasticsearch.yml

```
$ vim elasticsearch/config/elasticsearch.yml
```

需要修改的配置示例

```
cluster.name: elasticsearch
path.data: /home/adp/datastore/es/data
path.logs: /home/adp/datastore/es/logs
network.host: 172.0.0.1   ###修改为elasticsearch实际部ip
```

创建elasticsearch数据，日志文件保存目录，并授权给adp用户

```
$ mkdir -p /home/adp/datastore/es/data
$ mkdir -p /home/adp/datastore/es/logs
$ chown -R adp:adp /home/adp/datastore/es
```

一般使用不需要开启es的xpack认证, 则es安装完毕，不需要进行下面的操作。

- 开启xpack security认证

如果es需要开启xpack认证，则需要对es默认的用户设置密码，在上一步普通方式安装的基础上进行操作。

修改elasticsearch/config/elasticsearch.yml配置文件

增加如下配置

```
xpack.security.enabled: true
```

在启动elasticsearch的情况下，去执行下面命令生成用户名，密码。

进入es的安装目录，生成elasticsearch的用户密码，两种方式，使用命令自动生成密码，或者分别对不同用户设置密码，完成后保存相关密码，这里我们最终统一使用elastic用户。

执行下面命令，生成用户密码。

```
$ cd /home/adp/cloudlog/elasticsearch
$./bin/elasticsearch-setup-passwords auto
```

或者可以自定义用户密码，使用下面命令，一步步对不同的用户设置密码。

```
$ cd /home/adp/cloudlog/elasticsearch
$./bin/elasticsearch-setup-passwords interactive
```

该种方式es安装完毕。

- 开启xpack ssl security证书认证

  在上一步开启用户名，密码的前提下，增加了es的transport ssl访问认证，需要先完成前面普通安装，生成es用户密码步骤，再执行下面操作，生成es的ssl证书。

在启动elasticsearch的情况下，去执行下面命令生成ssl证书。

执行下面命令，生成ssl证书，并移到到es的config目录下, 并将新生成的证书文件权限授权给adp用户，这里直接将es整个目录授权给adp用户。

```
$ ./bin/elasticsearch-certutil ca
$./bin/elasticsearch-certutil cert --ca elastic-stack-ca.p12
$ mv elastic-certificates.p12 config/
$ mv elastic-stack-ca.p12 config/
$ chown -R adp:adp /home/adp/cloudlog/elasticsearch
```

根据上面操作生成ssl证书后，将生成的elastic-certificates.p12文件需要进行保存，后续安装cloudlog服务时，需要刚才生成的ssl证书文件。

修改elasticsearch/config/elasticsearch.yml配置文件，增加下列配置

```
#开启安全特性
xpack.security.enabled: true
#开启transport tls
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.keystore.path: elastic-certificates.p12
xpack.security.transport.ssl.truststore.path: elastic-certificates.p12
```

Ssl证书方式安装完毕。

4.3.3 安装Zookeeper

使用kafka本身自带的zookeeper，单机版本不需要再进行安装，zookeeper使用默认配置。

4.3.4 安装Kafka

修改配置文件/home/adp/cloudlog/kafka/config/server.properties

```
# vim server.properties
```

需要修改的配置示例

```
###修改为kafka实际部署IP
listeners=PLAINTEXT://0.0.0.0:9092
advertised.listeners=PLAINTEXT://127.0.0.1:9092
```

4.3.5 安装Logstash

修改配置文件/home/adp/cloudlog/logstash_backend/config/logstash.yml

```
# vim logstash.yml
```

需要修改的配置示例

```
###配置外部可以访问(这一步可以略过，即启动成功后，不能通过外部访问验证，只能通过查看端口或本地 curl
localhost:9600 方式查看，建议修改为logstash实际部署IP)
http.host: "127.0.0.1"
```

4.3.6 安装Cloudlog

修改配置文件/home/adp/cloudlog/config/application.properties

```
# vim application.properties
```

需要修改的配置示例

```
##修改cloudlog服务host与port
server.port=6888
server.host=127.0.0.1
#修改elasticsearch服务host与port
elasticsearch.host=127.0.0.1
elasticsearch.port=9300
elasticsearch.http.port=9200
# 修改kafkaServer
recvlog.server=127.0.0.1:9092
# 修改控制台配置
amp.console.url= http://uat-test.com:9000
#修改redis配置
spring.redis.host= localhost
spring.redis.password=
spring.redis.port= 6379
```

Es普通方式安装，则cloudlog安装完毕。

如果es开启xpack security认证，application.properties文件则需要增加下面配置。

```
# xpack security 认证配置，是否开启，用户名username，密码password
xpack.security.enabled=true
elasticsearch.username=elastic
elasticsearch.password=elastic
```

如果es开启了xpack ssl 证书认证，application.properties文件则需要增加下面配置，并需要将安装es时生成的ssl证书文件复制上传到cloudlog服务相关目录，如/home/adp/cloudlog/config目录下。

```
#xpack security 认证配置，是否开启，用户名username，密码password
xpack.security.enabled=true
elasticsearch.username=elastic
elasticsearch.password=elastic

# ssl认证配置
# 如果开启ssl认证，智能日志服务则需要elasticsearch生成的elastic-certificates.p12证书文件
xpack.security.transport.ssl.enabled=true
xpack.security.transport.ssl.verification_mode=certificate
xpack.security.transport.ssl.keystore.path=/home/adp/cloudlog/config/elastic-certificates.p12
xpack.security.transport.ssl.truststore.path=/home/adp/cloudlog/config/elastic-certificates.p12
```

4.4 安装后的工作

以下几个部分描述日志服务产品安装后的工作：

4.4.1 了解产品目录结构

在执行完日志服务产品的安装工作之后，日志服务器等目录结构

表 4-2 日志服务平台目录结构

目录	说明
bin	监控服务器组件的启动脚本。
config	Cloudlog的配置文件。
elasticsearch	elasticsearch的配置文件。
jdk	jdk的配置文件。
kafka	kafka的配置文件。
lib	日志服务应用程序jar包。
logs	Cloudlog日志文件。
logstash_backend	logstash_backend的配置文件。
public	保存logstash与logagent数据文件

4.4.2 启动日志服务

启动日志服务必需按顺序启动：启动elasticsearch 启动zookeeper, kafka 启动cloudlog 启动logstash。

日志服务所有启动脚本都存放在/home/adp/cloudlog/bin目录下，所有组件启动必须切换到 adp（非root用户）用户执行。

4.4.2.1 启动Elasticsearch

1. adp用户在/home/adp/cloudlog/bin目录执行Elasticsearch启动命令

```
# ./start-elastic.sh
```

2. 查看elasticsearch控制台运行状态，若端口9200与9300存在，表示启动成功。

```
netstat -ntpl|grep 9200
netstat -ntpl|grep 9300
```

3. 访问 http://localhost: 9200端口验证， 访问到页面则成功

4.4.2.2 启动Zookeeper

1. adp用户在/home/adp/cloudlog/bin目录执行zookeeper启动命令

```
# ./ start-zookeeper.sh
```

2. 查看zookeeper控制台运行状态，若端口2181存在，表示启动成功。

```
netstat -ntpl|grep 2181
```

4.4.2.3 启动Kafka

1. adp用户在/home/adp/cloudlog/bin目录执行Kafka启动命令

```
# ./ start-Kafka.sh
```

2. 查看Kafka控制台运行状态，若端口9092存在，表示启动成功。

```
netstat -ntpl|grep 9092
```

4.4.2.4 启动Cloudlog

1. adp用户在/home/adp/cloudlog/bin目录执行cloudlog启动命令

```
# ./ start-cloudlog.sh
```

2. 查看cloudlog控制台运行状态，若端口6888功。

```
netstat -ntpl|grep 6888
```

3. 访问 http://localhost:6888端口验证， 访问到页面则成功

4.4.2.5 启动Logstash

1. adp用户在/home/adp/cloudlog/bin目录执行logstash启动命令

```
#./ start-logstash-backend.sh
```

2. 更新logstash配置，使用一个进程定时更新配置,修改ip地址为logstash实际部署的ip地址

```
#./start-load-logstash-conf.sh  http://127.0.0.1:6888/logstashconfig/logstash.backend.conf
```

3. 查看logstash控制台运行状态，若端口5601示启动成功。

```
netstat -ntpl|grep 9600
```

4. 访问 http://localhost: 9600端口验证， 访问到页面则成功

4.4.3 注册为平台服务

日志服务部署完成并注册为平台服务后，用户才可以访问使用。

登录访问Web控制台，选择【平台管理】-\【服务管理】。编辑【日志服务】服务URL址。修改完成，点击服务上线，完成日志服务的注册和上线。

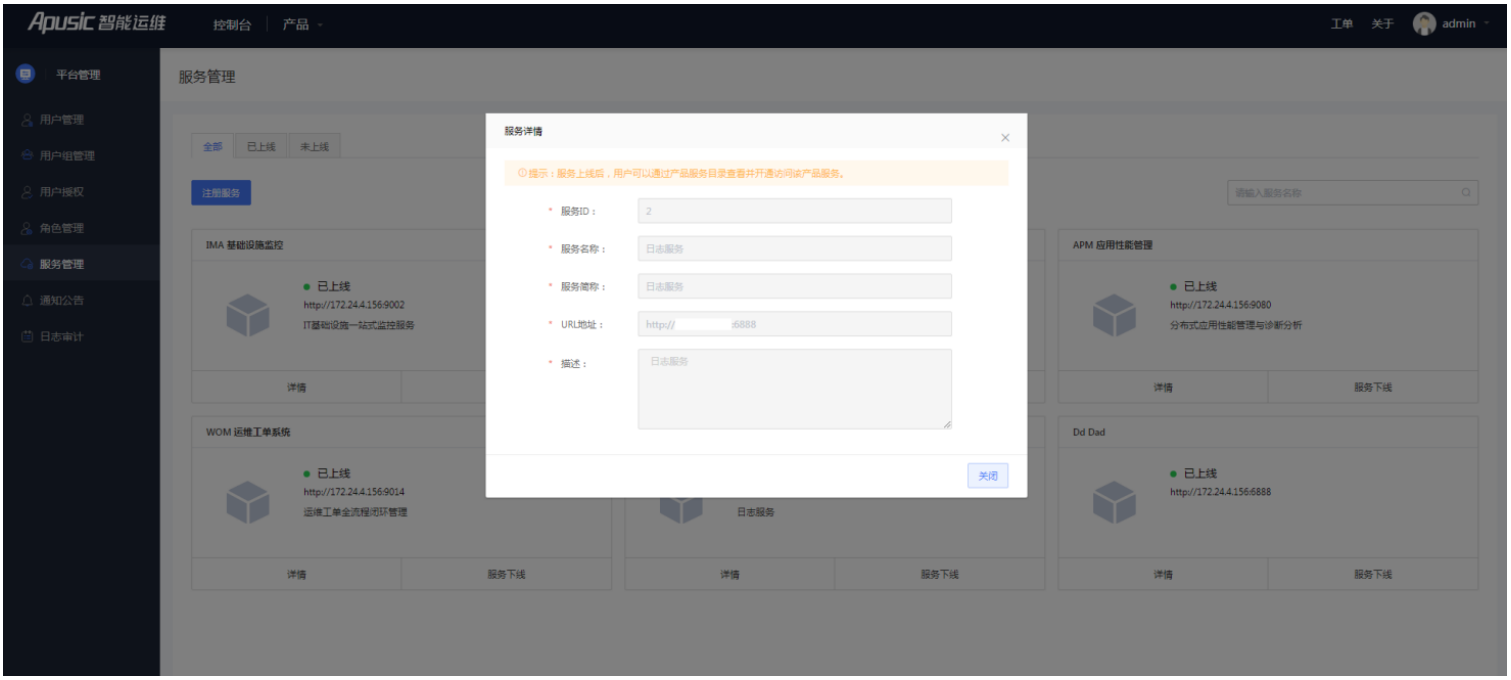


图4-1注册日志服务

控制台首页产品下拉列表中切换至日志服务，出现如下图则表明部署成功 。

图4-2日志服务首页

4.4.4 停止运行日志服务

4.4.4.1 停止运行Elasticsearch

查看其端口9200所运行的进程的pid，使用kill命令终止监控服务器进程。

```
# netstat -lntp | grep 9200
tcp6       0      0 :::9200       :::*            LISTEN      19358 /./jdk/bin/jav
# kill -9 19358
```

查看其端口9300所运行的进程的pid，使用kill命令终止监控服务器进程。

```
# netstat -lntp | grep 9300
tcp6       0      0 :::9300       :::*            LISTEN      19358 /./jdk/bin/jav
# kill -9 19358
```

4.4.4.2 停止运行Zookeeper

查看其端口2181所运行的进程的pid，使用kill命令终止监控服务器进程

```
# netstat -lntp | grep 2181
tcp6       0      0 :::2181       :::*            LISTEN      19358 /./jdk/bin/jav
# kill -9 19358
```

4.4.4.3 停止运行Kafka

查看其端口9092所运行的进程的pid，使用kill命令终止监控服务器进程。

```
# netstat -lntp | grep 9092
tcp6       0      0 :::9092       :::*            LISTEN      19358 /./jdk/bin/jav
# kill -9 19358
```

4.4.4.4 停止运行Cloudlog

查看其端口6888进程的pid，使用kill命令终止监控服务器进程。

```
# netstat -lntp | grep 6888
tcp6       0      0 :::6888::*            LISTEN      19358 /java
# kill -9 19358
```

4.4.4.5 停止运行Logstash

查看其端口5601进程的pid，使用kill命令终止监控服务器进程。

```
# netstat -lntp | grep 9600
tcp6       0      0 ::: 9600   :::*            LISTEN      19358 /node
# kill -9 19358
```

4.4.5 卸载产品

停止日志产品所有服务，执行下列命令，删除日志服务安装部署目录与数据存储目录即可完成卸载操作。

```
rm -rf /home/adp/cloudlog
rm -rf /home/adp/datastore
```

5 附录：集群高可用安装

5.1 部署规划

AILP日志服务高可用规划

IP	主机名	CPU	内存	磁盘	用途
172.20 .140.101	linux-140-101	8	16G	500G	kafka+zookeeper、es-master1、logstash、 cloudlog
172.20.140.102	linux-140-102	8	16G	500G	kafka+zookeeper、es-master2、logstash
172.20.140.103	linux-140-103	8	16G	500G	kafka+zookeeper、es-master3

各服务器节点需要安装JDK，如果未安装，可参考附录第六章第一节进行安装JDK。

5.2 集群安装部署

5.2.1 Zookeeper集群安装

安装搭建zookeeper集群，我们使用kafka自带的zookeeper进行部署zookeeper集群，也可以根据需求单独使用zookeeper进行部署zookeeper集群。

获取kafka产品包kafka_2.12-2.7.0.tgz，或者从日志服务产品包中获取kafka安装包：cloudlog/kafka。

如果使用日志产品包中的cloudlog/kafka，则不需要解压产品包，直接复制拷贝到对应目录，可以跳过第一步解压产品包，进行后续的操作。

1.解压kafka产品包

在172.20.140.101服务器进行操作

这里统一将kafka解压到/usr/local/kafka目录。

```
$ cd /usr/local
$ tar -zxvf kafka_2.12-2.7.0.tgz

$ mv kafka_2.12-2.7.0  kafka
$ cd kafka
```

2.修改zookeeper配置文件

在172.20.140.101服务器进行操作

可以使用默认配置，如果在同一台服务器部署多个节点需要修改存储地址，端口。

```
$  vim config/zookeeper.properties

#zookeeper数据保存目录(该目录需要手动创建)
dataDir=/usr/local/kafka/zookeeper
dataLogDir=/usr/local/kafka/logs/zookeeper
maxClientCnxns=100
tickTime=2000
initLimit=10
syncLimit=5
admin.enableServer=false

#zookeeper端口
```

```
clientPort=2181

server.1=172.20.140.101:2888:3888
server.2=172.20.140.102:2888:3888
server.3=172.20.140.103:2888:3888
```

在kafka的zookeeper数据目录添加myid文件，写入broker.id属性值。注意myid文件必须在zookeeper的数据目录下添加，在上面配置中即dataDir的目录/usr/local/kafka/zookeeper目录下进行添加。

```
#创建zookeeper数据目录并进入该目录
$ mkdir -p /usr/local/kafka/zookeeper
$ cd /usr/local/kafka/zookeeper

# 创建myid文件，写入brokerid，该值不能相同，其他服务器分别是2,3
$ echo 1  myid
```

同时将kafka文件夹复制到172.20.140.102， 172.20.140.103两台服务器并修改kafka组件中zookeeper的配置文件zookeeper.properties（可以不修改，三台服务器数据目录保持相同）， 其他服务器分别修改myid文件的值，分别写入2，3。

使用下面命令并输入密码，将kafka复制到其他服务器。

```
#复制安装包到其他两台服务器
$ scp -r /usr/local/kafka  root@172.20.140.102:/usr/local/
$ scp -r /usr/local/kafka  root@172.20.140.102:/usr/local/
```

3.启动zookeeper

启动三个zookeeper节点,进入kafka安装目录，启动zookeeper

```
#进入kafka目录
$ cd /usr/local/kafka

$ nohup ./bin/zookeeper-server-start.sh  config/zookeeper.properties &
```

分别启动三台服务器上的zookeeper。

先启动101服务器zookeeper时，成功启动zookeeper服务后，可以看到该zookeeper连接102,103服务器的zookeeper失败，这是正常的，因为其他两台服务器zookeeper还未启动。再启动102服务器zookeeper，成功启动该zookeeper后，可以看到该zookeeper已连接101服务器zookeeper，连接102服务器zookeeper失败。最后启动103服务器zookeeper,可以看到该zookeeper已连接101,102服务器zookeeper,此时已经完成了zookeeper集群搭建。

注意

第一次启动服务时， 可以先使用下面命令进行启动，确认配置正常，zookeeper服务可以正常启动时，关掉该进程，再通过后台方式nohup方式进行启动。

```
#第一次进行启动zookeeper，确定zookeeper是否启动成功
$ ./bin/zookeeper-server-start.sh  config/zookeeper.properties
```

4.检查zookeeper是否正常启动

分别检查三台服务器zookeeper端口2181是否正常。

```
$  netstat -ntlp |grep 2181
```

5.2.2 Kafka集群安装

上一步在Zookeeper集群安装时，已经成功拷贝并解压kafka的安装包，下面进行修改kafka配置文件，并进行安装kafka集群。

1.修改kafka配置文件

在172.20.140.101服务器，进入kafka安装目录， 修改kafka配置文件sever.properties。

```
#进入kafka目录
$ cd /usr/local/kafka

$  vim config/server.properties

#修改id为1
broker.id=1

#kafka数据保存目录
log.dirs=/usr/local/kafka/kafka-logs

# 下面的ip分别为对应服务器的ip
listeners=PLAINTEXT://172.20.140.101:9092
advertised.listeners=PLAINTEXT://172.20.140.101:9092

#zookeeper地址
zookeeper.connect=172.20.140.101:2181,172.20.140.102:2181,172.20.140.103:2181
```

同时修改172.20.140.102， 172.20.140.103其他两台服务器上的kafka配置文件，分别修改broker.id为2,3,修改listeners监听地址为对应服务器的ip地址，修改zookeeper连接地址。

2.启动kafka

分别进入各服务器kafka安装目录，启动kafka。

```
#进入kafka目录
$ cd /usr/local/kafka

#启动kafka
$ nohup ./bin/kafka-server-start.sh  config/server.properties  &
```

分别启动三台服务器的kafka组件。

注意

第一次启动服务时， 可以先使用下面命令进行启动，确认配置正常，kafka服务可以正常启动时，关掉该进程，再通过后台方式nohup方式进行启动。

```
#第一次进行启动kafka，确定kafka是否启动成功
$ ./bin/kafkar-server-start.sh  config/server.properties
```

3.检查kafka是否正常启动

判断kafka是否正常启动，判断kafka服务端口9092是否正常启动

```
#判断端口是否正常启动
$ netstat -ntlp|grep 9092
```

分别检查三台服务器kafka是否正常启动。

5.2.3 Elasticsearch集群安装

获取elasticsearch产品包elasticsearch-7.5.0.tar.gz，或者从日志服务产品包中获取elasticsearch安装包：cloudlog/elasticsearch。

在三台服务器部署elasticsearch, 三个es节点都作为master主节点方式，并且都保存数据。

如果使用日志产品包中的cloudlog/elasticsearch，则不需要解压产品包，直接复制拷贝到对应目录，可以跳过第一步解压产品包，进行后续的操作。

1.解压产品包

解压安装elasticsearch, 在三台服务于器101~103各节点都要执行，我们统一将elasticsearch安装到/opt/data目录下

在172.20.140.101服务器进行操作

```
# 创建elasticsearch安装目录
$ mkdir -p /opt/data/elasticsearch
#进入目录
$ cd /opt/data

# 拷贝产品包到/opt/data目录，解压产品包
$ tar -zxvf elasticsearch-7.5.0.tar.gz
$ mv elasticsearch-7.5.0  elasticsearch
$ cd elasticsearch
```

2.修改配置文件

在172.20.140.101服务器进行操作，修改elasticsearch.yml配置文件。

```
$  vim config/elasticsearch.yml

# 定义集群名称，多个节点集群名称保持一致
cluster.name: elasticsearch
# 可以使用本机的hostname，多台服务器的名称不能相同
node.name: linux-140-101
node.master: true
node.data: true
# data和logs目录
path.data: /opt/data/elasticsearch/app/data
path.logs: /opt/data/elasticsearch/app/logs

network.host: 0.0.0.0
http.port: 9200
transport.tcp.port: 9300

# 集群配置
discovery.zen.ping.unicast.hosts: ["172.20.140.101:9300", "172.20.140.102:9300",
"172.20.140.103:9300"]
discovery.zen.minimum_master_nodes: 2
discovery.zen.ping_timeout: 3s
```

3.创建启动用户，并授权

Elasticsearch需要使用非root用户启动，创建elasticsearch用户，并创建数据，日志保存目录。

```
# 创建用户
$ sudo useradd elasticsearch

# 创建elasticsearch的data和log目录
$ mkdir -p  /opt/data/elasticsearch/app/data
$ mkdir -p  /opt/data/elasticsearch/app/logs

$ chown -R elasticsearch:elasticsearch  /opt/data/elasticsearch
```

拷贝产品包到其他两台服务器172.20.140.102， 172.20.140.103，修改配置文件elasticsearch.yml，node.name在每个节点的名称不相同。

4.启动es服务

```
# 进入elasticsearch目录安装目录
$ cd /opt/data/elasticsearch
# 切换到elasticsearch用户
$ su elasticsearch

$ ./bin/elasticsearch
# 确认无误后，使用后台方式启动
$ sudo  ./bin/elasticsearch -d
```

其他两台服务器，也进行启动。

启动后查看各服务器9200,9300是否正常启动。

```
# 查看elasticsearch是否正常启动
$ netstat -ntlp|grep 9200
```

5.查看集群状态

查看es集群状态，*号代表的是该节点为es当前的master节点。

5.2.4 Cloudlog安装

选择在 172.20.140.101上部署一个云日志服务，云日志服务是用于配置和查看日志，部署单节点即可。

1.修改配置文件/home/adp/cloudlog/config/application.properties

```
$  vim  application.properties
```

修改配置文件中的下面内容

elasticsearch使用集群方式存储，修改为多个elasticsearch节点地址。

Kafka使用集群方式存储，修改为多个kafka节点地址。

控制台及cloulog应用的redis存储如果使用哨兵集群方式进行存储，修改对应redis相关的配置。

如果redis仍是单节点部署，可以使用单节点部署cloudlog的redis配置。

```
##修改cloudlog服务host与port
server.port=6888
server.host=172.20.140.101

# elasticsearch集群配置 elasticsearch节点，多个节点使用英文逗号分隔
```

```
elasticsearch.host=172.20.140.101,172.20.140.102,172.20.140.103
# elasticsearch内部tcp通信端口，多个节点的端口使用英文逗号分隔
elasticsearch.port=9300,9300,9300
# elasticsearch外部http通信端口，多个节点的端口使用英文逗号分隔
elasticsearch.http.port=9200,9200,9200

# kafka集群配置，修改kafkaServer
recvlog.server=172.20.140.101:9092,172.20.140.102:9092,172.20.140.103:9092

# 修改控制台配置
amp.console.url= http://console.com:9000

# redis哨兵集群配置，修改redis
spring.data.redis.repositories.enabled=false
spring.redis.password=
spring.redis.sentinel.nodes=172.24.4.110:26379,172.24.4.111:26379,172.24.4.112:26379
spring.redis.sentinel.master=mymaster
spring.redis.timeout= 3600
```

2.启动cloudlog服务

```
$  cd  /home/adp/cloudlog/bin
$  ./start-cloudlog.sh
```

3.验证

查看端口

```
$  netstat -ntlp |grep 6888
```

5.2.5 Logstash安装

安装两个Logstash节点，利用kafka的groupId实现logstash的高可用。在172.20.140.101，172.20.140.102服务器上面安装logstash。

根据服务器操作系统，获取x86环境下安装包logstash-7.12.0-linux-x86_64.tar.gz或者arm64环境下安装包logstash-7.12.0-linux-aarch64.tar.gz，也可以从日志服务产品包中获取logstash安装包：cloudlog/logstash_backend。

部署logstash时，需要先成功部署cloudlog服务，并启动。

如果使用日志产品包中的cloudlog/logstash_backend，则不需要解压产品包，直接复制拷贝到对应目录，可以跳过第一步解压产品包，进行后续的操作。

1.解压产品包

我们将logstasth安装在/opt/data目录下，这里以x86安装包进行说明。

```
$ cd /opt/data
$ tar -zxvf logstash-7.12.0-linux-x86_64.tar.gz

$ mv logstash-7.12.0  logstash
```

修改配置文件

```
# 进入logstash安装目录
$ cd  logstash
```

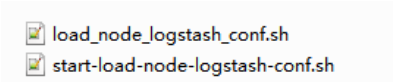
```
$ vim logstash.yml

# 配置外部可以访问(这一步可以略过，即启动成功后，不能通过外部访问验证，只能通过查看端口或本地 curl localhost:9600 方式查看，建议修改为logstash实际部署IP)

http.host: "172.20.140.101"
```

2.拉去最新配置文件信息

在日志服务cloudlog/bin目录下。拷贝下面文件load_node_logstash_conf.sh, start-load-node-logstash-conf.sh到当前服务器logstash的安装目录下/opt/data/logstash



更新logstash配置，使用一个进程定时更新配置,修改ip地址为cloudlog实际部署的ip地址

下面内容在一行中执行。

```
$ ./start-load-node-logstash-conf.sh
http://172.20.140.101:6888/logstashconfig/logstash.backend.conf
```

查看该进程是否启动成功

```
$ ps -aux|grep logstash
```

3.启动logstash

进入logstash安装目录，执行logstash启动命令

```
$ ./bin/logstash -f ./logstash.backend.conf --config.reload.automatic

# 判断无误后，可以使用后台方式启动
nohup ./bin/logstash -f ./logstash.backend.conf --config.reload.automatic logstash.log &
```

4.验证是否启动成功

查看logstash控制台运行状态，若端口9600正常则启动成功。

```
netstat -ntpl|grep 9600
```

或者访问 http://172.20.140.101: 9600端口验证， 访问到页面则成功。

如果多台服务器需要部署，可以重复以上操作进行部署。

6 附录：环境组件安装

6.1 安装JDK

进入Oracle官网(<https://www.oracle.com/technetwork/java/javase/downloads/index.html)，下载对应的JDK版本包进行安装，这里以jdk-8u181-linux-x64.tar.gz版本为例介绍JDK安装流程。

1.创建存放java的目录，将jdk安装包解压到特定目录下。

```
mkdir /usr/local/java
tar -zxvf jdk-8u181-linux-x64.tar.gz  -C /usr/local/java
```

2.配置java环境变量。

```
vi /etc/profile
```

3.在/etc/profile里面添加如下内容，修改完成后，wq保存并退出（先按Esc，接着输入 :wq）。

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_181
export JAVA_BIN=/usr/local/java/jdk1.8.0_181/bin
export PATH=$PATH:$JAVA_HOME/bin
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export JAVA_HOME JAVA_BIN PATH CLASSPATH
```

4.配置完成后，输入source profile，再输入java -version命令查看是否配置成功，如果显示java version " jdk1.8.0_181"信息，则表示已经配置成功。

```
source profile
java -version
```

6.2 安装Redis

Web控制台运行需要Redis缓存服务，以下是Redis的简要安装步骤。

1.下载5.05版本在 /usr/local/ 下新建一个 redis 文件夹。

```
wget http://download.redis.io/releases/redis-5.0.4.tar.gz
```

2.在 /usr/local/ 下新建一个 redis 文件夹。

```
cd /usr/local
mkdir redis
```

3.解压redis-5.0.5.tar.gz安装包。

```
tar -zxvf redis-5.0.5.tar.gz
```

4.安装 gcc 环境。

```
yum ./../assets/images/install gcc-c++
```

5.进入解压后的 redis-5.0.5 目录，执行 make 命令。

```
cd redis-5.0.5
make
```

6.进入 redis-5.0.5的src 目录后执行 make ./assets/images/install命令。

```
cd src/
make ./assets/images/install
```

7.在 redis 目录下创建 bin 和 etc 两个文件夹。

```
mkdir -p /usr/local/redis/bin
mkdir -p /usr/local/redis/etc
```

8.redis-5.0.5 里的主配置文件 redis.conf 移动到刚创建的 etc 文件夹。

```
cd redis-5.0.5
mv redis.conf /usr/local/redis/etc/
```

9.将 src 目录里带有绿色标识的文件全都移动到刚创建的 bin 文件夹。

```
cd src/
mv mkreleasehdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-sentinel
redis-server redis-trib.rb /usr/local/redis/bin/
```

10.进入 etc 目录，修改 redis.conf 文件。

```
cd /usr/local/redis/etc/
vi redis.conf
```

11.注释掉 bind 127.0.0.1 这一行。

```
#bind 127.0.0.1
```

12.将 protected-mode 属性改为 no （关闭保护模式，不然会阻止远程访问；同上，正式服务器项目上线可不修改）。

```
protected-mode no
```

13.将 daemonize 属性改为 yes （这样启动时就在后台启动）。

```
daemonize yes
```

14.设置密码（可选，建议还是设个密码），修改完成后，保存并退出。

```
requirepass redispassword
```

15.在 redis 目录下执行,启动redis，查看redis是否成功启动。

```
cd /usr/local/redis/
./bin/redis-server /usr/local/redis/etc/redis.conf
ps -ef | grep redis
```

6.3 安装MySQL

AILP的监控平台的运行依赖数据库服务，当前支持MySQL，人大金仓等多种类型的关系数据库部署。此处以MySQL为例介绍数据库的安装过程，其他类型数据库请参考数据库厂商产品安装指南进行。

1.首先关闭linux的防火墙，执行命令。

```
chkconfig iptables off
```

2.从mysql官网上下载自己适合的mysql版本"https://dev.mysql.com/downloads/mysql/5.6.html#downloads"，进入mysql官网，进行下载，以下载mysql-5.6.46-linux-glibc2.12-x86_64.tar.gz为例。

3.将下载好的mysql压缩文件放置在linux的/usr/local文件夹下，解压mysql安装包。

```
tar zxvf mysql-5.6.46-linux-glibc2.12-x86_64.tar.gz
```

4.将解压后的文件重命名为mysql。

```
mv mysql-5.6.46-linux-glibc2.12-x86_64 mysql
```

5.创建mysql用户组及用户。

```
groupadd mysql
useradd -r -g mysql mysql
```

6.进入到mysql目录，执行添加MySQL配置的操作。

```
cp support-files/my-medium.cnf /etc/my.cnf
或：cp support-files/my-default.cnf /etc/my.cnf
```

是否覆盖？按y 回车

7.编辑/etc/my.cnf文件。

```
vi /etc/my.cnf
```

8.在my.cnf文件中添加或者修改相关配置，更改完成后保存退出。

```
#These are commonly set, remove the # and set as required.
basedir = /usr/local/mysql
datadir = /usr/local/mysql/data
port = 3306
# server_id = .....
socket = /tmp/mysql.sock
character-set-server = utf8
skip-name-resolve
log-err = /usr/local/mysql/data/error.log
pid-file = /usr/local/mysql/data/mysql.pid
```

9.在mysql当前目录下设定目录的访问权限（注意后面的小点，表示当前目录）。

```
chown -R mysql .
chgrp -R mysql .
scripts/mysql_./assets/images/install_db --user=mysql
chown -R root .
chown -R mysql data
```

10.面第三步执行可能会出现下面的错误。

```
[root@localhost mysql-mult]# ./scripts/mysql_./assets/images/install_db  --defaults-
file=conf/3306my.cnf
FATAL ERROR: please ./assets/images/install the following Perl modules before executing
./scripts/mysql_./assets/images/install_db:
```

11.解决方法： 安装autoconf库。

```
yum -y ./assets/images/install autoconf
```

12.初始化数据（在mysql/bin或者mysql/scripts下有个 mysql_./assets/images/install_db 可执行文件初始化数据库），进入mysql/bin或者mysql/scripts目录下，执行下面命令。

```
./mysql_./assets/images/install_db --verbose --user=root --defaults-file=/etc/my.cnf --
datadir=/usr/local/mysql/data --basedir=/usr/local/mysql
```

13.启动mysql，进入/usr/local/mysql/bin目录，执行下面命令。

```
./mysqld_safe --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=root &
```

注意，如果光标停留在屏幕上，表示启动成功，需要我们先关闭shell终端，再开启一个新的shell终端，不要执行退出操作。如果出现 mysql ended这样的语句，表示Mysql没有正常启动，你可以到log中查找问题。

14.设置开机启动，新开启shell中断后，进入mysql目录，执行下面命令。

```
cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
cp /usr/local/mysql/support-files/mysql.server /etc/rc.d/init.d/mysql
chmod 700 /etc/init.d/mysql
chkconfig --add mysqld
chkconfig --level 2345 mysqld on
chown mysql:mysql -R /usr/local/mysql/
```

15.重启linux。

```
reboot
```

16.查看mysql状态。

```
service mysqld status
```

17.添加远程访问权限

1)添加mysql命令。

```
ln  -s /usr/local/mysql/bin/mysql  /usr/bin
```

2)登录mysql，更改访问权限。

```
mysql -uroot -p  #密码为空直接回车,运行以下三条命令。
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'yourpassword' with grant option;
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY 'yourpassword' with grant option;
```

3)退出mysql。

```
exit
```

18.mysql安装完毕。

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