



APUSIC
固若长城
睿比世界

快速入门

金蝶Apusic智能告警v1.0

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 基础介绍
- .2 快速安装
 - .2.1 产品介绍说明
 - .2.2 产品组件关系
 - .2.3 安装控制台
 - .2.4 安装监控服务
 - .2.4.1 监控服务器
 - .2.4.2 网络拓扑组件
 - .2.4.3 站点监控组件
 - .2.4.4 Web监控平台
 - .2.5 安装告警服务
 - .2.5.1 env
 - .2.5.2 Flink
 - .2.5.3 安装kafka
 - .2.5.4 安装Alarm-manager
 - .2.6
 - .2.7 环境组件安装
 - .2.7.1 JDK
 - .2.7.2 Redis
 - .2.7.3 MySQL数据库

1 基础介绍

金蝶Apm智能告警平台（简称 AAlarm）是一款可视化工作流的一站式的运维告警平台，实现多源告警事件的统一接入和管理，通过接入各监控系统和事件源的原始告警事件，做到集中处理和分析并统一通知。同时通过引入AI、大数据、流式计算等技术，高效实现海量数据的分析和挖掘，快速定位故障根因、并对潜在系统运行风险进行预警、实现IT和业务的持续洞察和改进。智能告警平台通过“大数据”+“AI智能”的模式，致力于为IT运维人员，技术工程师打造一款具备全栈数据处理。

2 快速安装

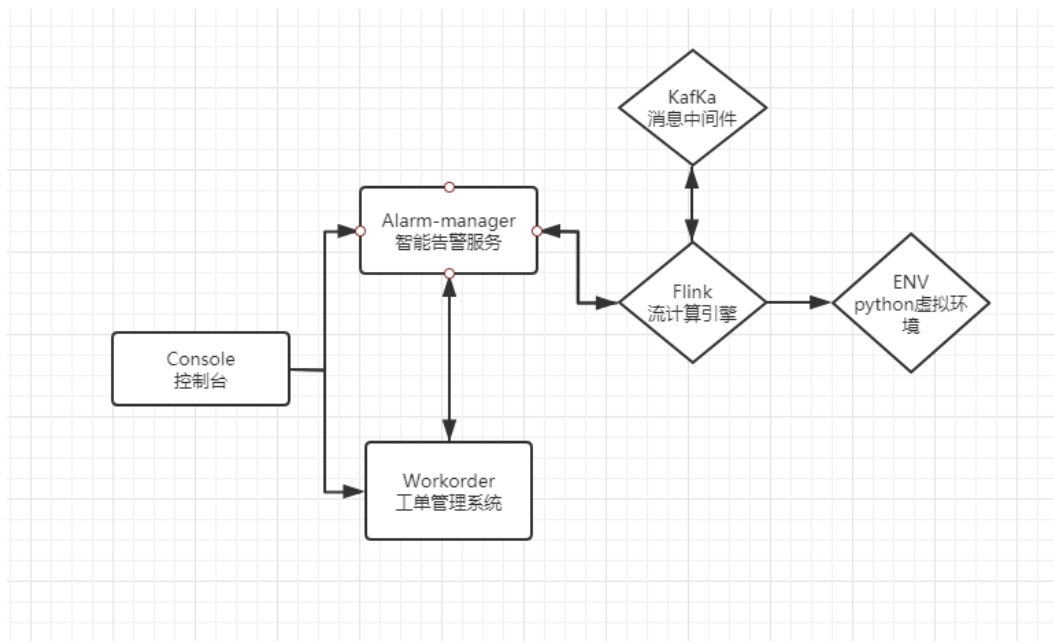
本文档对安装只做简单介绍, 详细安装步骤请查阅《安装部署》文档

2.1 产品介质说明

安装Alarm平台最低配置要求见下表

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
Python	3.7.10 (无法提前安装python, Alarm产品包里包含定制Python安装包)
Java	java1.8 (并且配置JAVA_HOME)
内存	4核8G或以上
硬盘	可用空间1T或以上
浏览器	FireFox 21及以上、Chrome 32及以上、IE 10及以上

2.2 产品组件关系



2.3 安装控制台

创建Alarm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。

2.4 安装监控服务

2.4.1 监控服务器

创建AMP产品安装根目录，将amp-prom-agent-v1.1.0.linux-amd64.tar.gz解压到对应目录及完成产品包安装。注意：\${PATH} 为部署环境实际目录

```
# tar -zxvf amp-prom-agent-v1.1.0.linux-amd64.tar.gz -C /${PATH}/AMP
```

监控服务器prom-agent安装后的目录为/\${PATH}/AMP/prom-agent。解压即完成监控服务器的安装，接下来进行相关参数配置。

2.4.2 网络拓扑组件

1. 解压amp-nettopo-amd64.tar.gz安装介质包到AMP目录：

```
tar -zxvf amp-nettopo-amd64.tar.gz -C /${PATH}/AMP
```

2. 压缩包解压后目录为amp-nettopo-amd64，编辑config.yaml。

```
vi /${PATH}/AMP/amp-nettopo-amd64/config.yaml
```

2.4.3 站点监控组件

如果需要监控web站点，则需要启动部署web站点监控采集器程序，上传amp-monitor-blackbox-v1.1.0-linux-amd64.tar.gz安装介质包并解压到AMP目录：

```
tar -zxvf amp-balckbox-exporter-v3.3.0.linux-amd64.tar.gz -C /${PATH}/AMP
```

2.4.4 Web监控平台

创建AMP产品安装根目录，指定\${PATH} 为实际路径，将amp-infra-monitor-v3.3.tar.gz解压到对应目录及完成产品包安装，/\${PATH}/AMP/amp-infra-monitor为产品解压后的目录。

```
# mkdir -p /${PATH}/AMP
# tar -zxvf amp-infra-monitor-v3.3.tar.gz -C /${PATH}/AMP
```

将license.xml授权文件拷贝到/\${PATH}/AMP/amp-infra-monitor根目录，否则监控平台web应用将无法启动。如上即完成Web监控平台应用的解压工作，接下来修改相关参数配置。5.2.6.3 配置参数 修改amp-infra-monitor/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：

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

表5-1 基础设施监控服务web监控平台应用配置文件

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-infra-monitor/conf/application-prod.yml
```

□ server.port 参数指定了该Web监控平台的默认端口，默认值为9002 □ 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 □ amp为相关连接配置信息，需要根据实际部署环境进行修改。
- amp.console.url 为web控制台 amp-console应用的部署地址，默认<http://localhost:9000>
- amp.agent.url 为监控服务器agent的部署地址，默认值<http://localhost:8100>，修改为对应的地址。
- amp.alarm.url 为告警服务器代理alarm-gateway的地址，默认值<http://localhost:8200>，修改为实际部署地址。
- amp.net_topo.url为网络拓组件amp-nettopo的地址，默认值为<http://localhost:9808>，修改为实际部署地址。
- amp.workorder.url为运维工单系统amp-workorder的地址，默认值为<http://localhost:9014>，修改为实际部署地址。 配置参考样例如下：

```

redis:
  timeout: 3600
  host: localhost
  port: 6379
  password: root
datasource:
  type: com.zaxxer.hikari.HikariDataSource
  url: jdbc:mysql://localhost:3306/amp_monitoring
  ?
useUnicode=true&characterEncoding=utf8&useSSL=false&useLegacyDatetimeCode=false&serverTimezone=UTC
  username: root
  password: root
amp:
  console:
    url: http://localhost:9000
  agent:
    url: http://localhost:8100
  alarm:
    url: http://localhost:8200
  net_topo:
    url: http://localhost:9808
<省略其他>

```

注意：没有授权文件监控平台web应用将无法启动，请联系您的销售或者技术顾问获取授权文件。

2.5 安装告警服务

本手册以在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。

根据业务需要监控应用较多，建议用两台以上部署

创建Alarm产品安装根目录，指定\${PATH} 为实际路径。Alarm产品包里的所有组件建议都安装在/\${PATH}/AOPS目录下

```
mkdir -p /${PATH}/AOPS
```

2.5.1 env

1. 解压安装包到/\${PATH}/AOPS/目录下

```
tar -zxvf py37_x86_env.tar.tar.gz -C /${PATH}/AOPS
```

2. 修改环境变量，将env/bin至环境变量

```
vim /etc/profile
#若环境变量中存在Anaconda、miniconda或其他python环境，需将其删除
export ENV_HOME=/${PATH}/AOPS/env
export PATH=$ENV_HOME/bin:$PATH
#立即生效
source /etc/profile
```

3. 修改python软连接

```
#若已存在/usr/bin/python，则先删除
rm /usr/bin/python
#修改 python软链接
ln -s env/bin/python3.7 /usr/bin/python
```

4. 命令行执行python，显示Python3.7.10，则表示安装成功（env无需启动）。

```
[xxx@xxxx AOPS]# python
Python 3.7.10 (default, Feb 26 2021, 18:47:35)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

2.5.2 Flink

安装Flink至少准备2台服务器，一个节点部署JobManager，其他节点部署TaskManager。安装目录要保持一致，建议统一安装在/\${PATH}/AOPS目录下。

1. 各节点统一解压flink至/\${PATH}/AOPS目录下

```
tar zxvf flink-1.12.2-bin-scala_2.11.tgz -C /${PATH}/AOPS
```

2. 修改配置参数

进入Flink配置文件目录

```
cd /${PATH}/AOPS/flink-1.12.2/con
```

修改flink-conf.yaml

```
vim flink-conf.yaml##修改jobmanager实际部署IP##jobmanager.rpc.address:实际部署IP##修改TaskManager个数，
不能超过服务器内核数。例：内存为8核16G，设置TaskManager个数不要超过8个taskmanager.numberOfTaskSlots: 8##修改
zookeeper实际部署IP##high-availability.zookeeper.quorum: localhost2181
```

修改masters文件，设定master节点

```
vim masters##修改为jobmanager实际部署IP##localhost8081
```

修改workers文件，设定work节点

```
vim workers##修改为TaskManager实际部署IP##localhost
```

2.5.3 安装kafka

1. 解压安装包到/\${PATH}/AOPS/目录下

```
tar -zxvf kafka_2.12-2.7.0.tgz -C /${PATH}/AOPS
```

2. 创建kafka日志目录与zookeeper存储快照的目录

```
#创建kafka日志目录mkdir /${PATH}/AOPS/kafka_2.12-2.7.0/log #创建zookeeper存储快照的目录mkdir /${PATH}/AOPS/kafka_2.12-2.7.0/zookeeper
```

3. 修改配置参数

- 进入kafka配置文件目录

```
cd /${PATH}/AOPS/kafka_2.12-2.7.0/config
```

- 修改 server.properties

```
vim server.properties###修改为kafka实际部署
IPlisteners=PLAINTEXT://localhost9092advertised.listeners=PLAINTEXT://localhost9092
zookeeper.connect=localhost2181###修改为kafka日志文件到对应目录
log.dirs=${PATH}/AOPS/kafka_2.12-2.7.0/log
```

- 修改producer.properties

```
vim producer.properties###修改为kafka实际部署IPbootstrap.servers=localhost9093
```

- 修改zookeeper.properties

```
vim zookeeper.properties###修改为zookeeper对应目录dataDir=${PATH}/AOPS/kafka_2.12-2.7.0/zookeeper
```

2.5.4 安装Alarm-manager

1. 解压安装包到/\${PATH}/AOPS/目录下

```
tar -zxvf aops-alarm-manager-prod-v1.1.tar.gz -C /${PATH}/AOPS
```

2. 修改配置参数

- 修改aops-alarm-manager/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:

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

表格4- 2Alarm-manager应用配置文件

- 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}/AOPS/aops-alarm-manager/conf/application-prod.yml
```

l server.port 参数指定了该web应用的默认端口，默认值为9000

l 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

l 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

配置关键参数参考样例如下：

```
server: port: 9016 servlet: context-path: / redis: timeout: 3600 host:
10.0.0.146 port: 6379 ##redis设置访问密码时，要添加password，如果没有则不用# password:
apusic$2020 datasource: type: com.zaxxer.hikari.HikariDataSource url:
jdbc:mysql://localhost:3306/aops_alarm_manager?
useUnicode=true&characterEncoding=utf8&useSSL=false&useLegacyDatetimeCode=false&serverTimezone=
username: root password: root kafka: bootstrap-servers: localhost:9092 amp: #
管控制台配置 console: url: http://localhost:9000 #工单系统配置 workorder: url:
http://localhost:9014aops: #流处理模块配置 flink: jobManager: localhost:8081 <省略其他>
```

2.6

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

```
tar -zxvf amp-workorder-prod-xxx.tar.gz -C /${PATH}/AMP
```

如上，即完成运维工单系统的解压工作，接下来修改相关参数配置。

2.7 环境组件安装

2.7.1 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x86_64架构下的jdk-8u181-linux-x64.tar.gz版本为例介绍安装流程。

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 /etc/profile
java -version
```

2.7.2 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 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 install命令。

```
cd src/
make 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
```

2.7.3 MySQL数据库

AMP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。是否覆盖？按y 回车

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

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_install_db --user=mysql
chown -R root .
chown -R mysql data
```

10. 上面第三步执行可能会出现下面的错误。

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

11. 解决方法：安装autoconf库。

```
yum -y install autoconf
```

12. 初始化数据（在mysql/bin或者mysql/scripts下有个mysql_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_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，您可以到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. 重启操作系统。

```
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

