



APUSIC  
固若长城  
睿比世界

# 快速入门

金蝶Apusic监控平台V3.4

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。

本档如有更新，不另行通知。对本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 简介
- 2 配置要求
- 3 快速安装
  - 3.1 产品介绍说明
  - 3.2 产品组件关系
  - 3.3 安装控制台
  - 3.4 安装监控服务
    - 3.4.1 监控服务器
    - 3.4.2 网络拓扑组件
    - 3.4.3 站点监控组件
    - 3.4.4 Web监控平台
  - 3.5 安装告警服务
    - 3.5.1 安装kafka
  - 3.6 安装工单
  - 3.7 相关组件安装
    - 3.7.1 安装JDK
    - 3.7.2 安装Redis
    - 3.7.3 安装MySQL
    - 3.7.4 安装Nginx

# 1 简介

金蝶Apmic监控软件（简称 AMP）能够实现对服务器、网络设备、存储、数据库、中间件、基础云服务、业务应用系统进行一体化、自动化、智能化的全面监控。保障IT基础设施的高可用和业务系统正常稳定可靠运行，极大提高信息中心IT运维的效率，使得对IT基础架构管理从被动分散的维护转变为主动集中的控制和自动化，智能化的管理。

AMP使IT基础架构真正成为保障业务服务水平的、可管理、可控制的业务平台，构建业务人员和IT管理运维人员的联系纽带，帮助组织快速应对外部变化给IT基础设施和业务应用带来的冲击和挑战。本章通过简单的监控示例来说明如何快速使用AMP产品实现对目标资源设施的监控。

## 2 配置要求

安装AMP V3.4产品的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), 支持x86_64、arm64、鲲鹏、飞腾、龙芯等架构
CPU	8vCPU以上
内存	16GiB 以上
硬盘	500GiB以上
MySQL	v5.6及以上版本
Redis	v3.0及以上版本

## 3 快速安装

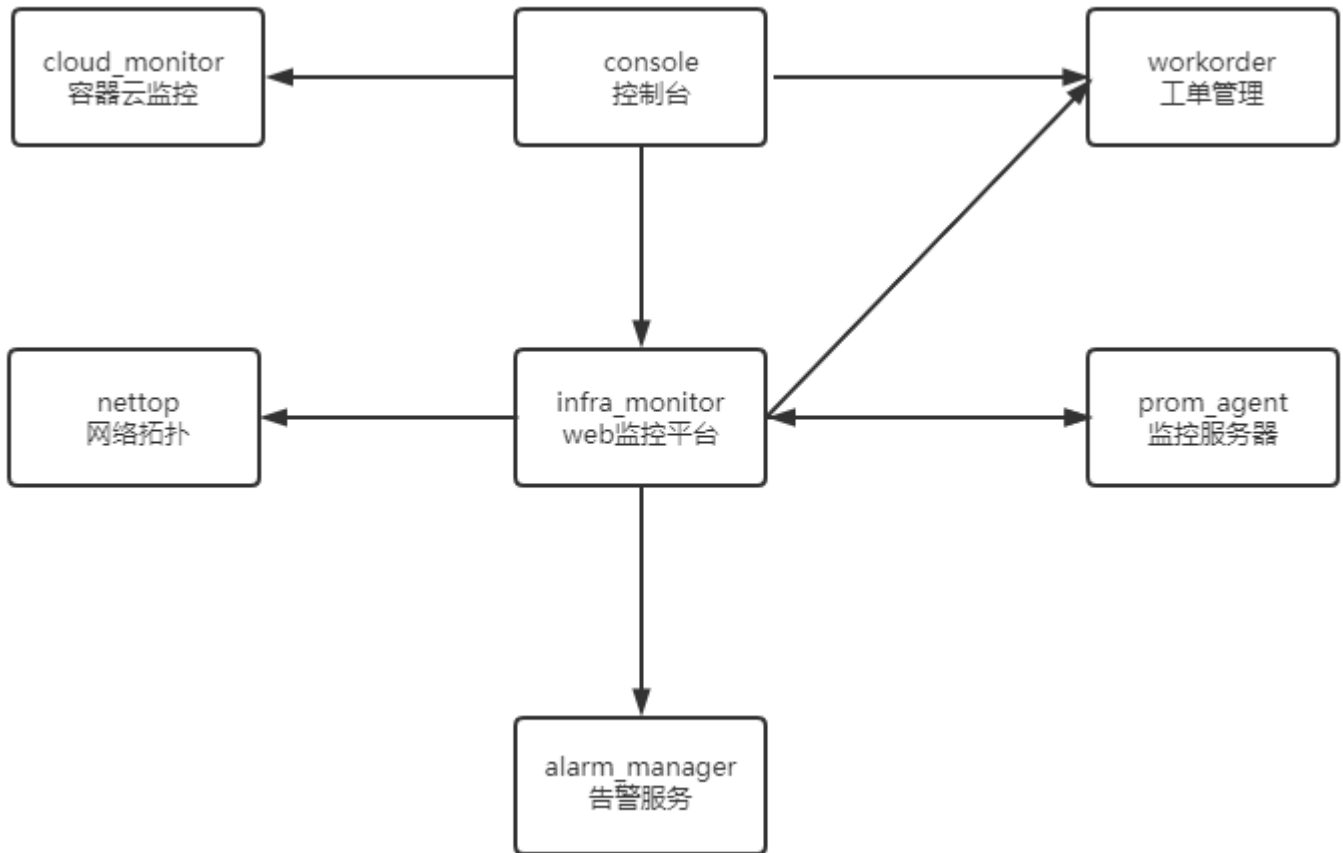
本文档对安装只做简单介绍、详细安装步骤请查阅安装手册文档

### 3.1 产品介质说明

AMP V3.4产品包括如下10个文件，请在安装前逐一检查。不同平台请使用对应用的产品安装包，若产品介质名称中不包含平台架构的字样，则适用于所有平台部署。以下以x86\_64的产品包介质为例说明：

组件名称	文件	说明
控制台	amp-console-v3.4.tar.gz	统一web控制台应用
Web监控平台	amp-infra-monitor-v3.4.tar.gz	web监控平台应用
监控服务器	amp-prom-agent-v3.4-linux-amd64.tar.gz	监控服务，包括监控引擎和prom-agent监控服务器代理
告警服务	aalarm-manager_SE-v1.2.tar.gz	告警服务，用于监控报警信息的通知
网络拓扑	amp-nettopo-v3.4-amd64.tar.gz	监控服务的网络拓组件，用户网络设备发现及拓扑图自动生成
站点监控	amp-balckbox-exporter-v3.4.0.linux-amd64.tar.gz	监控服务器的拨测组件，用于web站点、API等对象的可用性探测
工单管理系统	amp-workorder-v3.4.tar.gz	运维工单 web系统，运维工单的流程闭环管理
容器云监控	amp-cloud-monitor-v3.4.tar.gz	容器云监控服务，用于监控kubernetes集群
容器云监控插件	amp-monitor-kubernetes.linux-amd64.tar.gz	容器云监控服务的kubernetes插件
推送网关	amp-pushgateway-amd64.tar.gz	监控服务推送网关服务，用于推送监控指标数据给监控服务

### 3.2 产品组件关系



AMP V3.4各组件的作用如下所示。

- Web控制台：用户，权限，角色，项目，服务，日志审计管理等功能。
- Web监控平台：实现具体监控功能的管理，如创建监控任务，报警策略等。
- 监控服务器：包括监控agent服务，prometheus监控引擎，实现具体的监控功能。
- 告警服务器：对告警信息进行分组处理，产生告警通知。
- 网络拓扑：用于展示网络中各网络设备之间的关系。
- 运维工单系统：提交表单，跟进问题，反馈处理结果。
- 容器云监控：用于监控k8s集群。

### 3.3 安装控制台

创建AMP产品安装根目录，指定 $\{PATH\}$ 为实际路径，将amp-console-v3.3.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 /${PATH}/AMP
tar -zxvf amp-console-v3.3.tar.gz -C /${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。

## 3.4 安装监控服务

### 3.4.1 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。

### 3.4.2 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并进行配置。

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

### 3.4.3 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
```

### 3.4.4 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：

配置文件	说明
application-dev.yml	H2数据库作为持久化存储的配置，开发测试阶段使用，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-sample-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

用户可根据实际部署环境修改application.yml文件中的active值为采用不同环境的配置。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。

- 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
- 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.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>
- amp.alarm.url 为告警服务器代理alarm-gateway的地址，默认值<http://localhost:8200>
- amp.net\_topo.url为网络拓扑组件amp-nettopo的地址，默认值为<http://localhost:9808>
- amp.workorder.url为运维工单系统amp-workorder的地址，默认值为<http://localhost:9014>

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&useLegacyDatetimeCo
  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：没有授权许可文件，AMP监控平台将无法启动。

### 3.5 安装告警服务

根据业务需要监控应用较多，建议用两台以上服务器进行部署。

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

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

注意：AAlarm智能告警产品的安装过程可参考AAlarm产品的《安装手册》文档。

### 3.5.1 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

```
mkdir /${PATH}/AOPS/kafka_2.12-2.7.0/log
mkdir /${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 文件

```
###修改为kafka实际部署IP

listeners=PLAINTEXT://:9092
advertised.listeners=PLAINTEXT://localhost:9092
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 文件，修改为kafka实际部署IP

```
bootstrap.servers=localhost:9092
```

- 修改 `zookeeper.properties` 文件, 修改为zookeeper对应目录

```
dataDir=/${PATH}/AOPS/kafka_2.12-2.7.0/zookeeper
```

## 3.6 安装工单

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

```
tar -zxvf amp-workorder-v3.3.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。

## 3.7 相关组件安装

### 3.7.1 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, 在 `/etc/profile` 里面修改环境变量, 并保存退出。

```
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
```

3. 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
```

### 3.7.2 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 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 目录里下述可执行文件全都移动到创建的 bin 目录。

```
cd src/
mv mkreleashdr.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
```

### 3.7.3 安装MySQL

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官网上下载自己适合的mysql 5.5及以上的版本

<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
```

7. 编辑/etc/my.cnf文件, 在my.cnf文件中添加或者修改如下相关配置, 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
```

8. 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
```

9. 上面第三步执行可能会出现下面的错误。

```
FATAL ERROR: please install the following Perl modules before  
executing ./scripts/mysql_install_db:
```

解决方法: 安装autoconf库。

```
yum -y install autoconf
```

10. 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
```

11. 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文件中排查问题。

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/
```

然后执行 `reboot` 命令重启操作系统。

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。

```
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;
```

执行完毕后退出, 至此MySQL安装完毕。

### 3.7.4 安装Nginx

1. 到Nginx官网<http://nginx.org/en/download.html>下载nginx-1.16.1.tar.gz安装包
2. 安装gcc, gcc-c++, pcre, pcre-devel, zlib, zlib-devel, openssl, openssl-devel等nginx依赖的插件。

```
yum install -y gcc gcc-c++ pcre pcre-devel zlib zlib-devel  
openssl openssl-devel
```

3. 进入放源码的目录下,将下载的nginx-1.16.1.tar.gz安装包放在该目录下

```
cd /usr/local/src
```

4. 解压nginx-1.16.1.tar.gz

```
tar -xvf nginx-1.16.1.tar.gz
```

5. 将解压包移动到/usr/local/nginx目录

```
mv nginx-1.16.1 /usr/local/nginx
```

6. 编译安装Nginx

```
cd /usr/local/nginx
./configure --prefix=/usr/local/nginx #将nginx所有的资源文件放置
在/usr/local/nginx目录下
make && make install
```

7. 若报错在/usr/local/nginx下找不到错误日志logs/err.log, 则创建该文件。

```
mkdir -p /usr/local/nginx/logs
touch logs/error.log #创建错误日志文件
```

8. 确定nginx的80端口没有被其他程序占用, 启动nginx程序。

```
/usr/local/nginx/sbin/nginx
```

9. nginx的常用命令

```
./nginx -v #查看nginx版本
./nginx #启动nginx
ps -ef|grep nginx #查看nginx是否启动成功
./nginx -s stop #停止nginx
./nginx -s reload #重新加载nginx
```

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](http://www.apusic.com)

