



APUSIC
固若长城
睿比世界

安装手册

金蝶Apusic应用性能管理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 前言
 - .1.1 产品简介
 - .1.2 范围和读者
 - .1.3 文档导航
 - .1.4 约定与术语
- .2 系统环境要求
 - .2.1 环境要求
 - .2.2 推荐配置
- .3 安装Web控制台
 - .3.1 产品介质说明
- .4 安装Web控制台
 - .4.1 安装准备
 - .4.2 安装说明
 - .4.3 配置参数
 - .4.4 初始化数据库
- .5 安装后工作
 - .5.1 了解产品目录结构
 - .5.2 启动Web控制台
 - .5.3 停止运行服务
 - .5.4 卸载服务
- .6 安装应用性能管理服务
- .7 产品安装介质
- .8 安装应用性能管理
 - .8.1 安装准备
 - .8.2 安装说明
 - .8.3 配置参数
 - .8.4 初始化数据库
- .9 安装后工作
 - .9.1 了解产品目录结构
 - .9.2 启动应用性能管理
 - .9.3 注册为平台服务
 - .9.4 停止运行服务
 - .9.5 卸载服务
- .10 附录: 环境组件安装
- .11 安装JDK
- .12 安装Redis
- .13 安装MySQL
- .14 安装ElasticSearch
- .15 附录: 集群高可用安装
- .16 集群环境准备
 - .16.1 服务器规划
 - .16.2 安装JDK1.8
 - .16.3 安装路径
 - .16.4 关闭防火墙

- 17 Collector集群安装
 - 17.1 集群依赖安装
 - 17.1.1 ZooKeeper集群安装
 - 17.1.2 Elasticsearch集群安装
 - 17.2 collector集群安装
- 18 UI单机安装
 - 18.1 UI依赖安装
 - 18.1.1 MySQL
 - 18.1.2 Redis
 - 18.1.3 Console单机部署
 - 18.2 UI 安装
- 19 验证集群安装

1 前言

1.1 产品简介

金蝶Apusic应用性能管理平台（Apusic Application Performance Management，以下简称 AAPM）是金蝶天燕云计算股份有限公司经过多年经验积累，维护实践、自主研发和技术创新的一体化云原生监控平台产品。

AAPM 从业务视角对业务应用进行性能监测、分析、诊断，提高企业应用的可靠性和质量，保证用户得到良好的服务，以便发生故障的时候，能够快速定位和解决问题，降低IT总拥有成本。

1.2 范围和读者

本手册介绍AAPM产品安装相关的内容，主要适用于实施人员，维护人员等。

1.3 文档导航

| 章节 | 内容概述 |
|-------------|----------------|
| 1. 前言 | 文档范围，约定内容 |
| 2. 系统环境要求 | 安装产品所需的软硬件环境 |
| 3. 安装准备 | 产品安装前部署软件 |
| 4. 安装Web控制台 | Web控制台安装过程说明 |
| 5. 启动应用性能管理 | 应用性能管理服务安装过程说明 |

1.4 约定与术语

一些约定的缩略词诠释：

- AAPM

金蝶Apusic应用性能管理（Apusic Application Performance Management）

2 系统环境要求

AAPM产品支持Windows Server、Linux（包括采用龙芯和飞腾国产芯片的Linux服务器）、Unix等多个操作系统平台的安装部署，本章以Linux X86-64 环境为例介绍ApusicAPM产品安装过程。

2.1 环境要求

表格 2-1 软件及操作系统环境要求

| 组件 | 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 70及以上、Chrome 60及以上、IE 11及以上 |

2.2 推荐配置

安装ApusicAPM产品的推荐配置见下表：

表格 2-2 推荐配置

| 组件 | 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 70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，安装应用性能管理服务并注册到平台，即可使用ApusicAPM产品。

如果您需要部署独立的ApusicAPM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：

| 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 |

表格4- 1Web控制台产品介质

4 安装Web控制台

4.1 安装准备

- JDK

Web控制台应用运行需要JDK8 环境，参考附录5.1节安装说明

- Redis

Web控制台运行需要Redis缓存服务，参考附录5.2节安装说明。

- MySQL数据库

系统默认推荐使用MySQL，参考附录5.3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。

4.2 安装说明

创建ApusicAPM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}/ApusicAPM/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。

4.3 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：

| 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-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 |

表格4- 2Web控制台应用配置文件

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>
```

4.4 初始化数据库

1.登录mysql数据库。

```
mysql -uusername -ppassword
```

2.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，初始化应用性能管理数据库。

```
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
```

5 安装后工作

5.1 了解产品目录结构

| 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。 |

表格4- 3amp-console控制台目录结构

5.2 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4-1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。

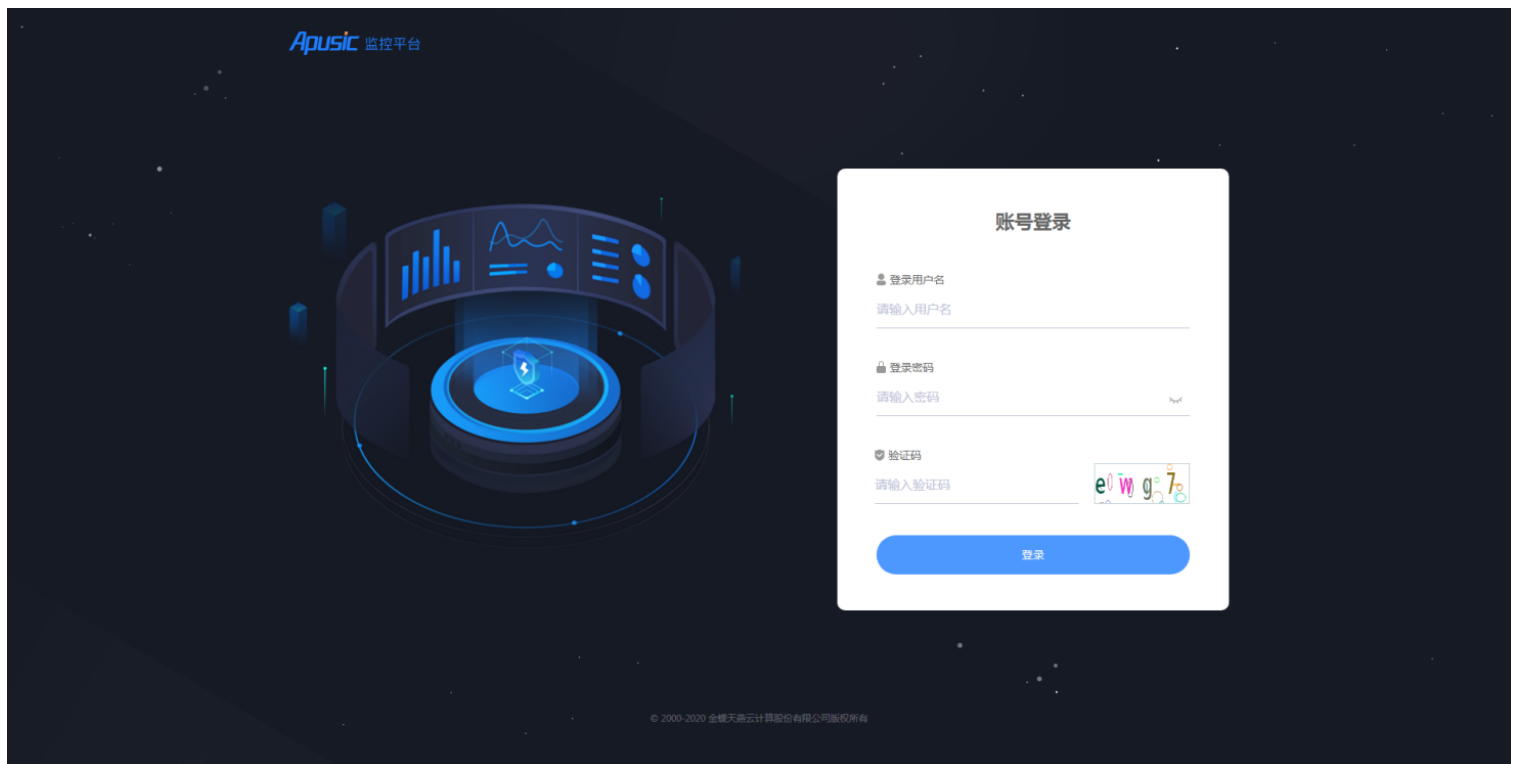


图4- 1登录页

5.3 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
```

5.4 卸载服务

删除ApusicAPM安装部署目录，即可完成卸载控制台操作。

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

6 安装应用性能管理服务

7 产品安装介质

应用性能管理相关安装介质如下：

| 组件名称 | 文件名 | 说明 |
|--------|----------------------------------|------------------------------|
| 应用性能管理 | apusic-skywalking-apm-bin.tar.gz | 应用性能管理组件，WEB 应用及APM的后端OAP 服务 |

图4- 1应用性能管理产品介质

8 安装应用性能管理

8.1 安装准备

- JDK

APM服务端和Web管理端应用运行需要JDK8 环境，参考附录5.1节安装说明。

- ElasticSearch

APM服务端存储采用ElasticSearch，参考附录5.4节安装说明。

- MySQL数据库

系统默认推荐使用MySQL，参考附录5.3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。

8.2 安装说明

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

```
mkdir -p /${PATH}/AMP
tar -zxvf apusic-skywalking-apm-bin.tar.gz -C /${PATH}/AMP
```

如上，即完成应用性能管理的解压工作，接下来修改相关参数配置。

8.3 配置参数

修改apusic-skywalking-apm-bin/config/application.yml配置文件。

```
vim /${PATH}/AMP/apusic-skywalking-apm-bin/config/application.yml
```

修改clusterNodes: \${SW_STORAGE_ES_CLUSTER_NODES:localhost:9200}的localhost为Elasticsearch存储所在的ip，多个以逗号隔开，修改完成后，wq保存并退出（先按Esc，接着输入:wq）。

```
storage:
  selector: ${SW_STORAGE:elasticsearch}
  elasticsearch:
    namespace: ${SW_NAMESPACE:""}
    clusterNodes: ${SW_STORAGE_ES_CLUSTER_NODES:localhost:9200}
    protocol: ${SW_STORAGE_ES_HTTP_PROTOCOL:"http"}
    trustStorePath: ${SW_STORAGE_ES_SSL_JKS_PATH:""}
    trustStorePass: ${SW_STORAGE_ES_SSL_JKS_PASS:""}
    user: ${SW_ES_USER:""}
    password: ${SW_ES_PASSWORD:""}
    secretsManagementFile: ${SW_ES_SECRETS_MANAGEMENT_FILE:""}
```

1. 编辑apusic-skywalking-apm-bin/webapp/webapp.yml配置文件。

```
vim /${PATH}/AMP/apusic-skywalking-apm-bin/webapp/webapp.yml
```

- server.port 参数指定了该web应用的默认端口，默认值为9080。
- spring.listOfServers ApusicAPM平台后端地址，需要根据实际部署环境进行修改。

- 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>
 - 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
2. 修改listOfServers: 127.0.0.1:12800的127.0.0.1为SkyWalking Server所在的ip，多个以逗号隔开，修改控制台URL: <http://localhost:9000>，配置参考样例如下:

```
server:
  port: 9080
collector:
  path: /graphql
  ribbon:
    ReadTimeout: 10000
    # Point to all backend's restHost:restPort, split by ,
    listOfServers: 127.0.0.1:12800
amp:
  console:
    url: http://localhost:9000
```

3. 如果选择MySQL数据库存储，则修改下面的MySQL的数据库配置，配置参考样例如下:

```
spring:
  datasource:
    type: com.zaxxer.hikari.HikariDataSource
    url: jdbc:mysql://localhost:3306/amp_apm_webapp?
useUnicode=true&characterEncoding=utf8&useSSL=false
    username: root
    password: root
    hikari:
      data-source-properties:
        cachePrepStmts: true
        prepStmtCacheSize: 250
        prepStmtCacheSqlLimit: 2048
        useServerPrepStmts: true
```

4. 如果选择KingBase数据库存储，注释上一步MySQL数据库数据源配置，取消注释KingBase数据库数据源配置，修改KingBase数据库配置内容，配置参考样例如下:

```
# KingBase数据库数据源配置
datasource:
  type: com.zaxxer.hikari.HikariDataSource
  driver-class-name: com.kingbase8.Driver
  url: jdbc:kingbase8://127.0.0.1:54321/amp_apm_webapp
  username: system
  password: system
```

```
hikari:
  poolName: Hikari
  auto-commit: false
  useServerPrepStmts: true
```

<省略其他>

如果选择其他国产数据库存储（达梦，神通，GBase-8s），和KingBase数据库存储一样，注释MySQL数据库配置信息，修改数据库配置信息。修改数据库连接地址，数据库名，账户，密码信息。

8.4 初始化数据库

ApmicAPM采用Elasticsearch存储数据，需要提前安装好Elasticsearch。数据库安装部署请参考第3.3章MySQL安装步骤说明，其他类型数据库请参考厂商安装部署文档。下面以mysql数据库的初始化为例，其他厂商数据库初始化类似。

1. 登录mysql数据库。

```
mysql -username -ppassword
```

2. 执行apmic-skywalking-apm-bin/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，初始化应用性能管理数据库。

```
create database amp_apm_webapp;
use amp_apm_webapp;
source /${PATH}/AMP/apmic-skywalking-apm-bin/sql/mysql/create.sql;
source /${PATH}/AMP/apmic-skywalking-apm-bin/sql/mysql/initial.sql;
```

9 安装后工作

9.1 了解产品目录结构

应用性能管理用于实时监控应用，安装后的产品安装包目录结构如下

| 目录 | 包含 |
|--------------|-------------------------------|
| bin | 应用性能管理平台组件的启动执行脚本 |
| agent | 存放APM探针 |
| licenses | 存放Licenses的txt文本信息 |
| config | 一些配置文件 |
| oap-libs | 应用程序依赖的一些jar包 |
| sql | 平台对应的apm多种数据库版本的sql创建及初始化脚本文件 |
| webapp | 前端UI包 |
| share_config | 存放探针下载包 |
| tool | 存放一些工具 |
| LICENSE | 授权文件信息 |
| NOTICE | 注意事项文档，对配置管理平台项目的注意事项说明 |
| HELP.md | 帮助文档，对配置管理平台项目的补充说明 |

表格4-1应用性能管理目录结构

9.2 启动应用性能管理

1. 修改完apusic-skywalking-apm-bin的配置文件后，后台启动应用性能管理。

```
/${PATH}/AMP/apusic-skywalking-apm-bin/bin/startup.sh
```

2. 检查apusic-skywalking-apm-bin是否正常启动，输入查看web界面端口9080和后台端口11800命令，若端口存在，表示启动成功。

```
netstat -lntp | grep 9080
netstat -lntp | grep 11800
```

9.3 注册为平台服务

应用性能管理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【应用性能管理】服务URL址。修改完成，点击服务上线，完成应用性能管理服务的注册和上线。

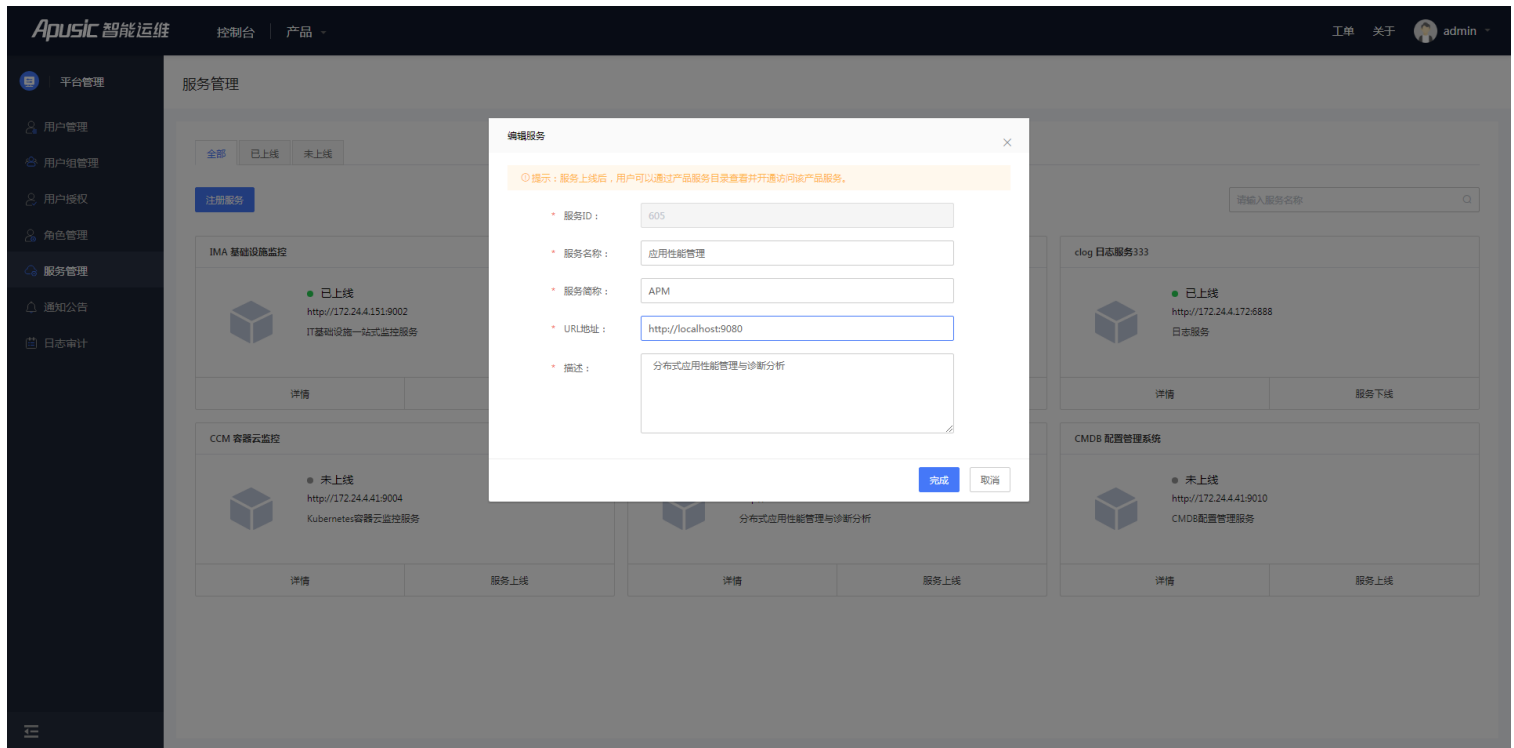


图4- 2注册应用性能管理服务

控制台首页产品下拉列表中切换至应用性能管理，出现如下图则表明部署成功。

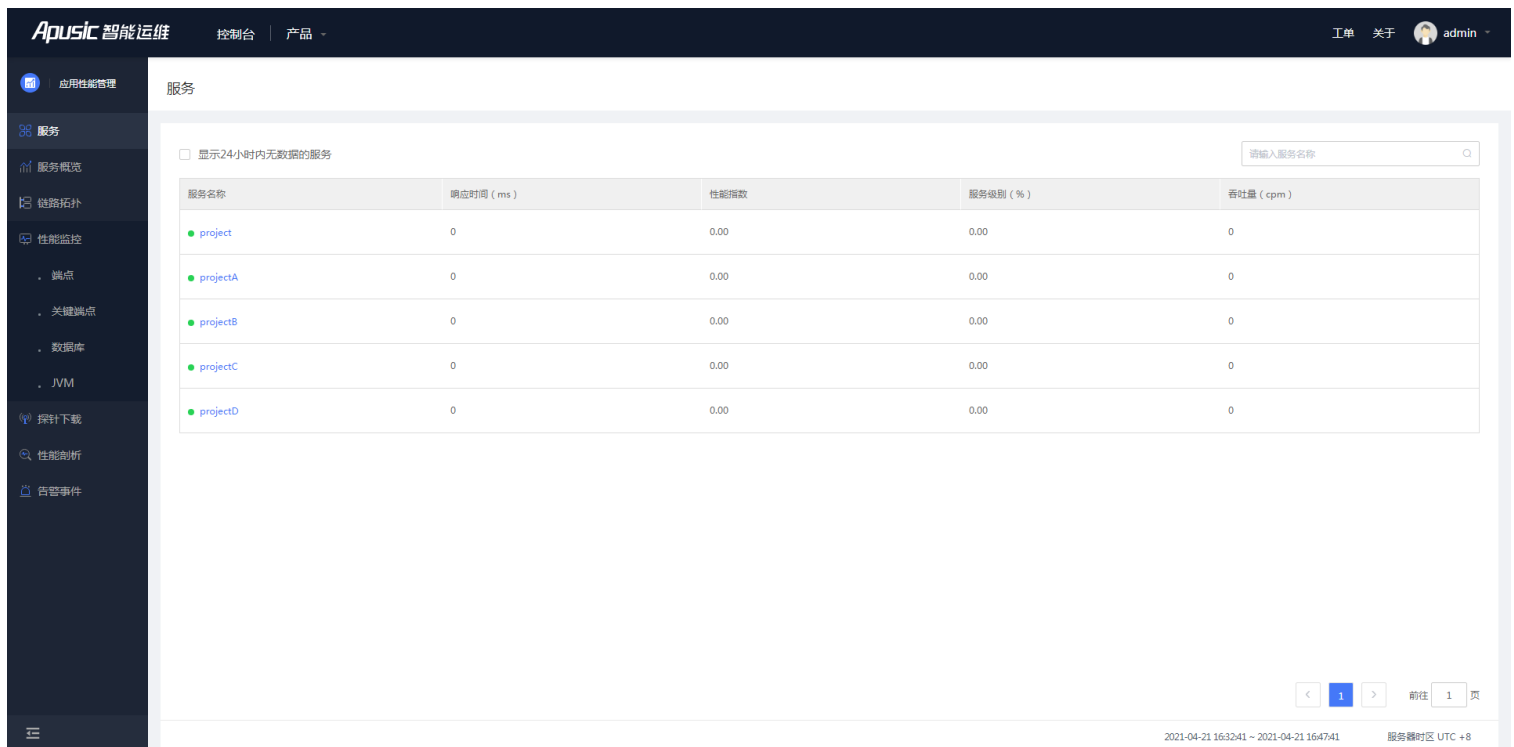


图4- 3应用性能管理服务首页

9.4 停止运行服务

1. 关闭应用性能管理web界面，使用kill命令终止应用性能管理前端进程。

```
# netstat -lntp | grep 9080
tcp6          0          0 :::9080          :::*             LISTEN        19348/java
```

2. 关闭应用性能管理后台，使用kill命令终止应用性能管理后台进程。

```
# netstat -lntp | grep 11800
tcp6          0          0 :::11800         :::*             LISTEN        19323/java
```

9.5 卸载服务

1. 删除ApusicAPM安装部署目录，即可完成卸载操作。

```
rm -rf /${PATH}/AMP/apusic-skywalking-apm-bin*
```

10 附录: 环境组件安装

11 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
```

12 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 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
```

13 安装MySQL

ApusicAPM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_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，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/mysqld
chmod 700 /etc/init.d/mysqld
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。

14 安装ElasticSearch

ApusicAPM平台的Tracing 数据存储依赖Elasticsearch存储器，需要搭建一个 Elasticsearch 服务。

1. 修改Linux系统的限制配置，打开文件vim /etc/security/limits.conf。

```
* soft nofile 65536
* hard nofile 65536
* soft nproc 65536
* hard nproc 65536
es - memlock unlimited
es - nofile 65536
```

2. 修改系统控制权限，更改当前用户拥有的内存权限太小，打开文件vim /etc/sysctl.conf。

```
vm.max_map_count=262144
```

3. 使系统控制权限配置生效。

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

4. 从elasticsearch官网上<https://www.elastic.co/cn/downloads/elasticsearch>，下载适合的elasticsearch6或7的版本，进行下载。
5. 以的elasticsearch-6.8.6.tar.gz安装包为例，将下载好的elasticsearch压缩文件放置在linux的/usr/local文件夹下，解压elasticsearch压缩包。

```
tar -zxvf elasticsearch-6.8.6.tar.gz
```

6. 将解压后的文件重命名为elasticsearch。

```
mv elasticsearch-6.8.6.tar.gz elasticsearch
```

7. 添加es用户，将文件夹elasticsearch赋予es权限。

```
useradd es
chown -R es:es /usr/local/elasticsearch
```

8. 创建elasticsearch数据文件的保存地址，并将文件夹权限赋值给es用户。

```
mkdir -p /data/elasticsearch
chown -R es:es /data/elasticsearch
```

9. 打开elasticsearch.yml文件vim /usr/local/elasticsearch/config/elasticsearch.yml，文件内增加如下代码。

```
discovery.type: single-node
cluster.name: es #集群名称，相同名称为一个集群
node.name: node-1 #节点名称，集群模式下每个节点名称唯一
node.master: true #当前节点是否可以被选举为master节点，是: true、否: false
node.data: true #当前节点是否用于存储数据，是: true、否: false
network.host: 0.0.0.0 #设置可访问的客户端ip。0.0.0.0代表任意客户端访问
#cluster.initial_master_nodes: ["node-1"] #之后新增的配置，写入候选主节点的设备地址，在开启服务后可以被选为主节点
```

```

discovery.zen.ping.unicast.hosts: ["localhost"]#新节点通过广播的形式加入集群
path.data: /data/elasticsearch/data #索引数据存放的位置
path.logs: /data/elasticsearch/logs #日志文件存放的位置
bootstrap.memory_lock: false #需求锁住物理内存, 是: true、否: false
bootstrap.system_call_filter: false #SecComp检测, 是: true、否: false
http.port: 9200 #对外提供的http端口, 默认 9200
http.cors.enabled: true #是否支持跨域, 是: true, 在使用插件时需要此配置
http.cors.allow-origin: "*" #表示支持所有域名

```

10. 打开jvm.options文件, vim /usr/local/elasticsearch/config/jvm.options, 修改jvm.options文件如下。

```

-Xms2g
-Xmx2g

```

11. 切换为es用户,使用es用户启动Elasticsearch。

```

es us
/usr/local/elasticsearch/bin/elasticsearch -d

```

12. 查看elasticsearch运行状态, 或者在浏览器访问[错误!超链接引用无效。](#)可以成功访问, elasticsearch安装完毕。

```

netstat -lntp | grep 9200

```

15 附录: 集群高可用安装

16 集群环境准备

16.1 服务器规划

| IP | 主机名 | CPU | 内存 | 磁盘 | 用途 |
|----------------|---------------|-----|-----|------|-----------------------------------|
| 172.18.100.168 | linux-100-168 | 8 | 16G | 500G | collector、zookeeper、elasticsearch |
| 172.18.100.169 | linux-100-169 | 8 | 16G | 500G | collector、zookeeper、elasticsearch |
| 172.18.100.170 | linux-100-170 | 8 | 16G | 500G | collector、zookeeper、elasticsearch |
| 172.18.100.181 | linux-100-181 | 8 | 16G | 500G | ui、console、mysql、redis、live-demo |

16.2 安装JDK1.8

各节点都要安装，具体步骤请参照第5章附录：环境组件安装----->5.1安装JDK。

16.3 安装路径

```
# 各节点都要执行
$ sudo mkdir /applications
```

16.4 关闭防火墙

```
# 各节点都要执行
$ sudo systemctl stop firewalld
$ sudo systemctl disable firewalld
```

17 Collector集群安装

17.1 集群依赖安装

17.1.1 ZooKeeper集群安装

1. 解压zookeeper到/applications目录

```
$ sudo tar zxvf apache-zookeeper-3.5.8-bin.tar.gz -C /applications
$ cd /applications
$ sudo mv apache-zookeeper-3.5.8-bin zookeeper
$ cd zookeeper
```

2. 配置zookeeper

```
# 备份默认配置
$ sudo cp conf/zoo_sample.cfg conf/zoo_sample.cfg.bak
$ sudo mv conf/zoo_sample.cfg conf/zoo.cfg
$ sudo vim conf/zoo.cfg
修改如下:
...
dataDir=/applications/zookeeper/data
dataLogDir=/applications/zookeeper/logs
...
autopurge.snapRetainCount=500
autopurge.purgeInterval=24
server.1=172.18.100.168:2888:3888
server.2=172.18.100.169:2888:3888
server.3=172.18.100.170:2888:3888
```

3. 生成数据和日志目录

```
$ sudo mkdir -p /applications/zookeeper/data
$ sudo mkdir -p /applications/zookeeper/logs
```

4. 远程拷贝zookeeper到各个节点

```
$ cd ../../
# 拷贝到172.18.100.169, 输入root账户的密码完成传输
$ sudo scp -r /applications/zookeeper root@172.18.100.169:/applications
#拷贝到172.18.100.170
$ sudo scp -r /applications/zookeeper root@172.18.100.170:/applications
```

5. 给zookeeper各节点分配server-id 此步骤需要远程登录到各个节点进行手动分配。

```
# 172.18.100.168节点的server-id分配指定为1;
$ sudo sh -c "echo '1'"> /applications/zookeeper/data/myid"
# 172.18.100.169节点的server-id分配指定为2;
$ sudo sh -c 'echo "2"'> /applications/zookeeper/data/myid'
```

```
# 172.18.100.170节点的server-id分配指定为3;
$ sudo sh -c 'echo "3"> /applications/zookeeper/data/myid'
```

6. 启动zk服务

```
# 分别进入到各节点的/applications/zookeeper/bin/下启动 zookeeper
$ cd /applications/zookeeper/bin/
$ sudo ./zkServer.sh start
```

7. 测试 在以上三个节点任意一个测试即可

```
# 测试集群的效果:
$ ./zkCli.sh -server 172.18.100.168:2181,172.18.100.169:2181,172.18.100.170:2181
# 查看zk集群的日志
$ ./zkServer.sh status
```

8. 编写 ZooKeeper 启动文件

```
$ sudo vim /usr/lib/systemd/system/zookeeper.service
[Unit]
Description=zookeeper.service
After=network.target
[Service]
Type=forking
Environment=JAVA_HOME=/usr/java/jdk1.8.0_291-amd64
User=root
Group=root
ExecStart=/applications/zookeeper/bin/zkServer.sh start
ExecStop=/applications/zookeeper/bin/zkServer.sh stop
ExecReload=/applications/zookeeper/bin/zkServer.sh restart
[Install]
WantedBy=multi-user.target
```

9. 启动服务

```
# 步骤6有启动, 则需关闭, 否则会端口冲突
$ ./zkServer.sh stop
# 分别进入到各节点执行
$ sudo systemctl daemon-reload
$ sudo systemctl start zookeeper
$ sudo systemctl enable zookeeper
```

17.1.2 Elasticsearch集群安装

1. 安装elasticsearch

```
# 168~170各节点都要执行, 这里选版本7.7.1是因为官方有arm64的安装包, 不用手动自行构建
$ sudo yum install -y elasticsearch-7.7.1-x86_64.rpm
```

2. 配置elasticsearch集群

```

$ sudo cp /etc/elasticsearch/elasticsearch.yml /etc/elasticsearch/elasticsearch.yml.bak
$ sudo vim /etc/elasticsearch/elasticsearch.yml
修改如下:
# 定义集群名称
cluster.name: skywalking
# 本机的hostname, hostname指令查看本机的主机名
node.name: linux-100-168
node.master: true
node.data: true
# data和logs目录, 应该放在空间较大的数据盘, 避免放入系统盘
path.data: /appdata/elasticsearch/data
path.logs: /appdata/elasticsearch/logs
...
network.host: 0.0.0.0
http.port: 9200
...
# 集群的host
discovery.zen.ping.unicast.hosts: ["172.18.100.168", "172.18.100.169", "172.18.100.170"]
discovery.zen.minimum_master_nodes: 2
# 集群的hostname
cluster.initial_master_nodes: ["linux-100-168", "linux-100-169", "linux-100-170"]

```

3. 配置JVM信息

```

# 备份默认配置
$ sudo cp /etc/elasticsearch/jvm.options /etc/elasticsearch/jvm.options.bak
# 修改jvm.options配置
$ sudo vim /etc/elasticsearch/jvm.options
修改如下:
-Xms4g
-Xmx4g
...
-XX:+UseConcMarkSweepGC
-XX:CMSInitiatingOccupancyFraction=75
-XX:+UseCMSInitiatingOccupancyOnly
...
-Djava.io.tmpdir=${ES_TMPDIR}
-XX:+HeapDumpOnOutOfMemoryError
# /appdata为数据盘路径
-XX:HeapDumpPath=/appdata/elasticsearch
-XX:ErrorFile=/appdata/elasticsearch/logs/hs_err_pid%p.log

```

4. 创建相关目录

```

# elasticsearch的data和log目录创建在数据盘目录/appdata/
$ sudo mkdir -p /appdata/elasticsearch/data
$ sudo mkdir -p /appdata/elasticsearch/logs
$ sudo chown -R elasticsearch:elasticsearch /appdata/elasticsearch

```

5. 启动服务

```
# 查看进程是否启动, 可以用这个命令ps -ax | grep elasticsearch
$ sudo systemctl start elasticsearch
$ sudo systemctl enable elasticsearch
# 关闭指令:
$ sudo systemctl stop elasticsearch
```

6. 验证服务

```
$ curl 172.18.100.168:9200/_cluster/stats?pretty
# 查看节点状态:
$ curl 172.18.100.168:9200/_nodes/process?pretty
```

17.2 collector集群安装

1. 解压collector到/applications目录

```
$ sudo tar zxvf apusic-skywalking-apm-bin-es7.tar.gz -C /applications
```

2. 配置collector连接zookeeper

```
$ cd /applications/apusic-skywalking-apm-bin-es7
# 备份默认配置
$ sudo cp config/application.yml config/application.yml.bak
$ sudo vimconfig/application.yml
修改如下:
cluster:
  selector: ${SW_CLUSTER:zookeeper}
  ...
  zookeeper:
    nameSpace: ${SW_NAMESPACE:"SW_CLUSTER_ZK_NAMESPACE"}
    hostPort:
${SW_CLUSTER_ZK_HOST_PORT:172.18.100.168:2181,172.18.100.169:2181,172.18.100.170:2181}
  ...
```

3. 配置collector连接elasticsearch

```
$ sudo vim config/application.yml
修改如下:
storage:
  selector: ${SW_STORAGE:elasticsearch7}
  ...
  elasticsearch7:
    ...
    clusterNodes:
${SW_STORAGE_ES_CLUSTER_NODES:172.18.100.168:9200,172.18.100.169:9200,172.18.100.170:9200}
    ...
    bulkActions: ${SW_STORAGE_ES_BULK_ACTIONS:2000} # Execute the bulk every 2000 requests
    ...
```

4. 修改collector启动脚本

修改初始化内存2G， 最大可用6G

```
$ sudo vim bin/oapService.sh
JAVA_OPTS="-Xms2G -Xmx6G"
```

5. 远程拷贝程序到其它节点

```
# 拷贝到169
$ sudo scp -r /applications/apusic-skywalking-apm-bin-es7 root@172.18.100.169:/applications
# 拷贝到170
$ sudo scp -r /applications/apusic-skywalking-apm-bin-es7 root@172.18.100.170:/applications
```

6. 启动

在三台节点上分别进行启动操作。

启动指令：

```
$ sudo bin/oapService.sh
```

7. 验证

```
# 查看进程：
$ ps -ax | grep skywalking
```

8. 编写 collector 启动文件

```
$ sudo vim /usr/lib/systemd/system/oap-server.service
[Unit]
Description=oap-server.service
After=network.target
[Service]
Type=forking
Environment=JAVA_HOME=/usr/java/jdk1.8.0_291-amd64
User=root
Group=root
ExecStart=/applications/apusic-skywalking-apm-bin-es7/bin/oapService.sh
WorkingDirectory=/applications/apusic-skywalking-apm-bin-es7
[Install]
WantedBy=multi-user.target
$ sudo cp bin/oapService.sh bin/oapService.sh.bak
$ sudo vim bin/oapService.sh
删除行首License相关注释
```

9. 启动服务

```
# 如步骤6有启动oapService进程，用kill -9 杀死，以防端口冲突
$ sudo systemctl daemon-reload
$ sudo systemctl start oap-server
$ sudo systemctl enable oap-server
```

18 UI单机安装

这里为了简化部署，将ui及其依赖console采用单机部署，如需集群部署请参照《金蝶Apm监控平台V3.0产品安装手册》--->第8章 附录：集群高可用安装--->8.4AMP组件高可用部署。

18.1 UI依赖安装

18.1.1 MySQL

参照第5章 附录：环境组件安装--->安装MySQL。

18.1.2 Redis

参照第5章 附录：环境组件安装--->安装Redis。

18.1.3 Console单机部署

照《金蝶Apm监控平台V3.0产品安装手册》--->第4章安装Web控制台。

18.2 UI 安装

1. 解压ui到/applications目录

```
sudo tar zxvf apusic-skywalking-apm-bin-es7.tar.gz -C /applications
```

2. 配置UI

```
$ cd /applications/apusic-skywalking-apm-bin-es7
$ sudo vim webapp/webapp.yml
# 修改collector listOfServers选项
...
collector:
  ...
  ribbon:
    ...
    listOfServers: 172.18.100.168:12800,172.18.100.169:12800,172.18.100.170:12800
    ...
  ...
# 修改数据源
spring:
  ...
  datasource:
    ...
    url: jdbc:mysql://172.18.100.181:3306/amp_apm_webapp?
useUnicode=true&characterEncoding=utf8&useSSL=false
    ...
  ...
# 修改控制台
amp:
  console:
    url: http://172.18.100.181:9000
  ...
```

3. 启动UI

```
$ sudo bin/webappService.sh
```

4. 验证

```
# 查看UI启动日志  
$ tail -f logs/webapp.log
```

5. 编写 collector 启动文件

```
$ sudo vim /usr/lib/systemd/system/apm-webapp.service  
[Unit]  
Description=apm-webapp.service  
After=network.target  
[Service]  
Type=forking  
Environment=JAVA_HOME=/usr/java/jdk1.8.0_291-amd64  
User=root  
Group=root  
ExecStart=/applications/apusic-skywalking-apm-bin-es7/bin/webappService.sh  
[Install]  
WantedBy=multi-user.target  
$ sudo cp bin/webappService.sh bin/webappService.sh.bak  
$ sudo vim bin/webappService.sh  
删除行首License相关注释
```

6. 启动服务

```
# 如步骤3有启动webappService进程, 用kill -9 杀死, 以防端口冲突  
$ sudo systemctl daemon-reload  
$ sudo systemctl start apm-webapp  
$ sudo systemctl enable apm-webapp
```

19 验证集群安装

1. 解压live-demo到/applications目录

```
$ sudo tar zxvf live-demo-assembly.tar.gz -C /applications
```

2. 启动服务

```
$ export COLLECTOR_SERVER_LIST=172.18.100.168:11800,172.18.100.169:11800,172.18.100.170:11800
$ cd live-demo/bin
$ ./startup.sh
```

3. 请求服务

```
$ curl http://localhost:8764/projectA/test
```

4. UI查看结果

登录控制台，进入应用性能管理，查看UI拓扑图和调用链

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

