



APUSIC
固若长城
睿比世界

安装手册

金蝶Apusic中间件云平台v8.0.5

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 修订说明
- 2 前言
 - 2.1 适用读者
 - 2.2 相关文档
- 3 平台架构
 - 3.1 部署架构
 - 3.2 技术架构
- 4 产品安装
 - 4.1 支持的环境
 - 4.2 推荐硬件配置
 - 4.3 部署前准备
 - 4.3.1 环境要求
 - 4.3.2 软件要求
 - 4.3.3 关于License
 - 4.3.4 获取安装包
 - 4.3.5 安装介质说明
 - 4.3.6 依赖详情
 - 4.3.7 服务端口说明
 - 4.4 环境检查
 - 4.4.1 关闭防火墙和 SELinux
 - 4.4.2 JDK
 - 4.4.3 安装依赖
 - 4.4.4 内核参数
 - 4.5 安装前准备
 - 4.6 开始安装
 - 4.6.1 安装平台组件
 - 4.6.2 安装管控台
 - 4.6.3 授权激活
 - 4.6.4 注册系统节点
 - 4.7 高可用部署
 - 4.8 常见问题与排错
 - 4.8.1 1. 端口冲突（启动失败提示 `BindException: Address already in use`）

- 4.8.2 2. 服务启动失败 (`status` 显示非 `running`)
- 4.8.3 3. JVM 内存不足 (`OutOfMemoryError` / 卡顿)
- 4.8.4 4. Elasticsearch 集群状态异常
- 4.8.5 5. PostgreSQL 连接失败 (`FATAL: password authentication failed`)
- 4.8.6 6. 高可用集群 Eureka 节点未互相注册
- 4.8.7 7. 授权激活失败 (授权校验失败)
- 4.8.8 8. 浏览器访问管控台 404 / 无法访问

1 修订说明

本文根据实际情况进行更新，最新版本包含历史修改记录。

日期	手册版本	适用产品	更新说明
2024年12月	V8.0.3	ACP v8.0.3	调整格式，细化手册内容
2025年12月	V8.0.4	ACP v8.0.4	内容调整
2026年4月	V8.0.4	ACP v8.0.4-20260427	简化安装步骤
2026年6月	V8.0.5	ACP v8.0.5	更新版本

2 前言

本文档介绍了金蝶Apusic中间件云平台（简称“ACP”） V8.0.5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。

2.1 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。

2.2 相关文档

- [金蝶Apusic中间件云平台功能清单](#)
- [产品常见问题](#)
- [附录](#)

3 平台架构

3.1 部署架构

组件说明：

1. maas-aump

- 功能：
 - 自动化安装：通过脚本或配置管理工具（如 Ansible、Chef、Puppet）实现中间件的自动化安装，减少人为操作，确保一致性。
 - 依赖关系管理：管理中间件所依赖的其他组件和服务，确保安装过程中所需的资源和软件包都已正确安装和配置。
 - 环境配置：根据部署环境（如开发、测试、生产等）配置中间件的参数，如端口号、内存限制、日志存储路径等。
 - 容器化部署：使用容器化技术（如 Docker）来部署中间件，确保在不同环境下的一致性和易于扩展。

2. maas-gateway

- 功能：
 1. 请求路由
 - 路由转发：根据请求的 URL 路径、请求头、参数等信息将请求转发到不同的微服务或后端系统。网关能够根据配置的路由规则，将不同类型的请求引导到不同的服务实例。
 - 动态路由：支持动态调整路由规则，例如根据负载情况、服务健康状态等自动选择最佳的服务实例进行路由。
 - 路径重写与转发：网关可以根据配置对请求路径进行重写或修改，例如将 `/api/v1/*` 转发为 `/service-v1/*`。
 2. 负载均衡
 - 流量分配：根据设定的负载均衡策略（如轮询、最少连接、加权等），在多个后端实例之间分配客户端请求的流量。
 - 自动伸缩：结合容器编排平台（如 Kubernetes）和云服务的能力，网关可以根据后端服务的负载和健康状态进行自动扩展和缩减。
 - 高可用性：通过对多个实例进行负载均衡，确保系统的高可用性，避免单点故障。

3. maas-manager

- 功能：

1. 业务流程管理 (BPM)

- 流程设计与配置：提供可视化的流程设计工具，使得业务流程可以通过配置来设计与定制，包括审批流程、任务流转、通知机制等。
- 流程监控与优化：实时监控业务流程的执行状态，分析瓶颈，优化流程，确保业务流畅运行。
- 自动化与协同：自动化处理一些重复性任务，减少人工干预，提高业务处理效率。同时，支持团队协作，多个角色间的任务分配和协同工作。

2. 任务与工作流管理

- 任务分配与跟踪：根据角色、权限和业务规则将任务分配给相应人员，并跟踪任务的完成情况。
- 工作流审批：管理和处理需要审批的业务流程，如请假申请、合同审批、资金审批等，支持多层级、多角色的审批流程。
- 任务提醒与通知：系统能够根据任务进度和到期时间自动发出提醒，减少遗漏和延迟。

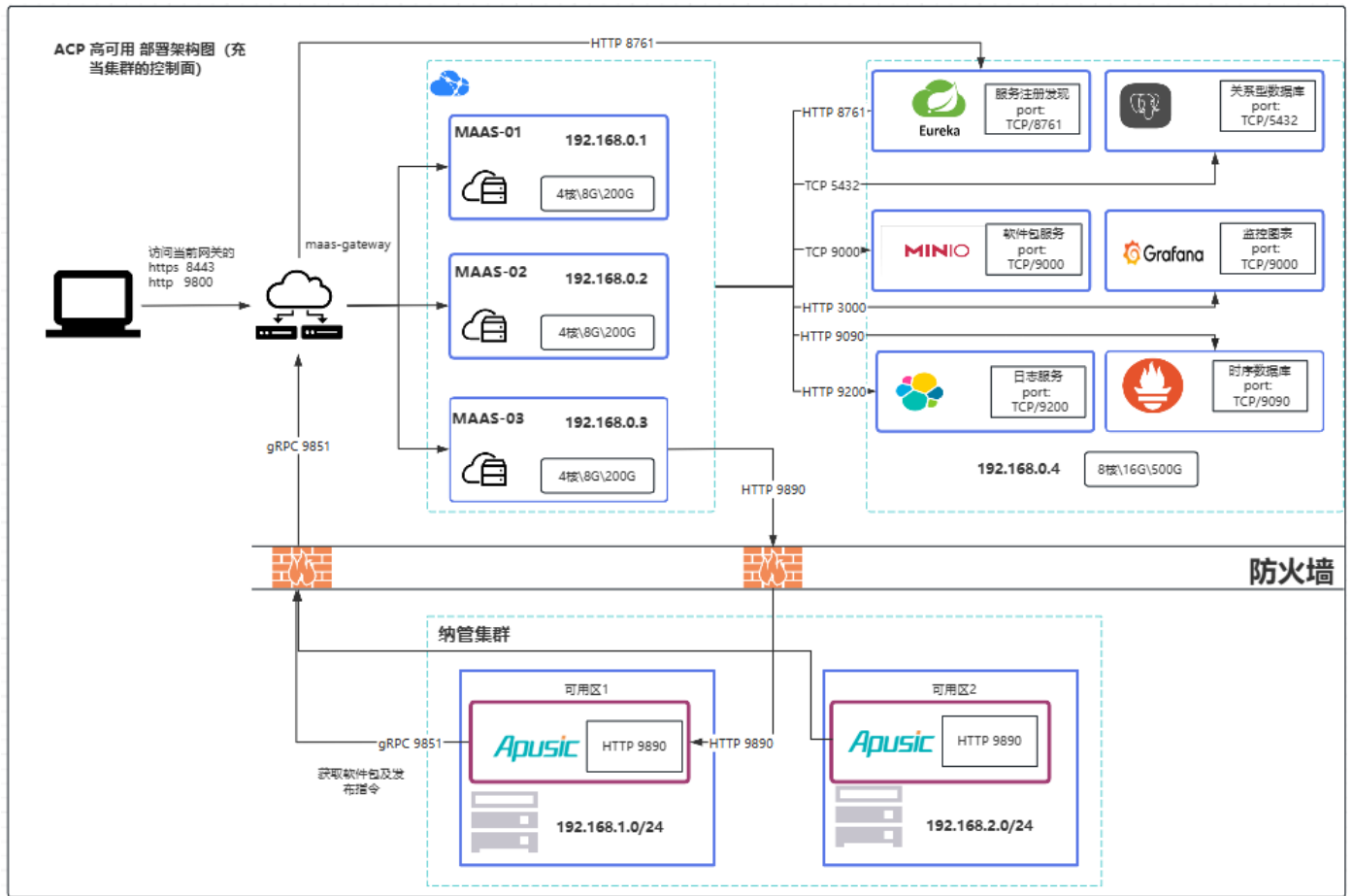
3. 数据与报表管理

- 业务数据收集与存储：收集并存储业务过程中的相关数据，例如销售数据、客户信息、订单详情等。
- 自定义报表生成：根据用户需求，提供自定义的业务报表，支持数据可视化展示，帮助决策者快速了解业务状况。
- 数据分析与决策支持：分析历史业务数据，帮助公司识别趋势、预测未来发展，并为管理层决策提供依据。

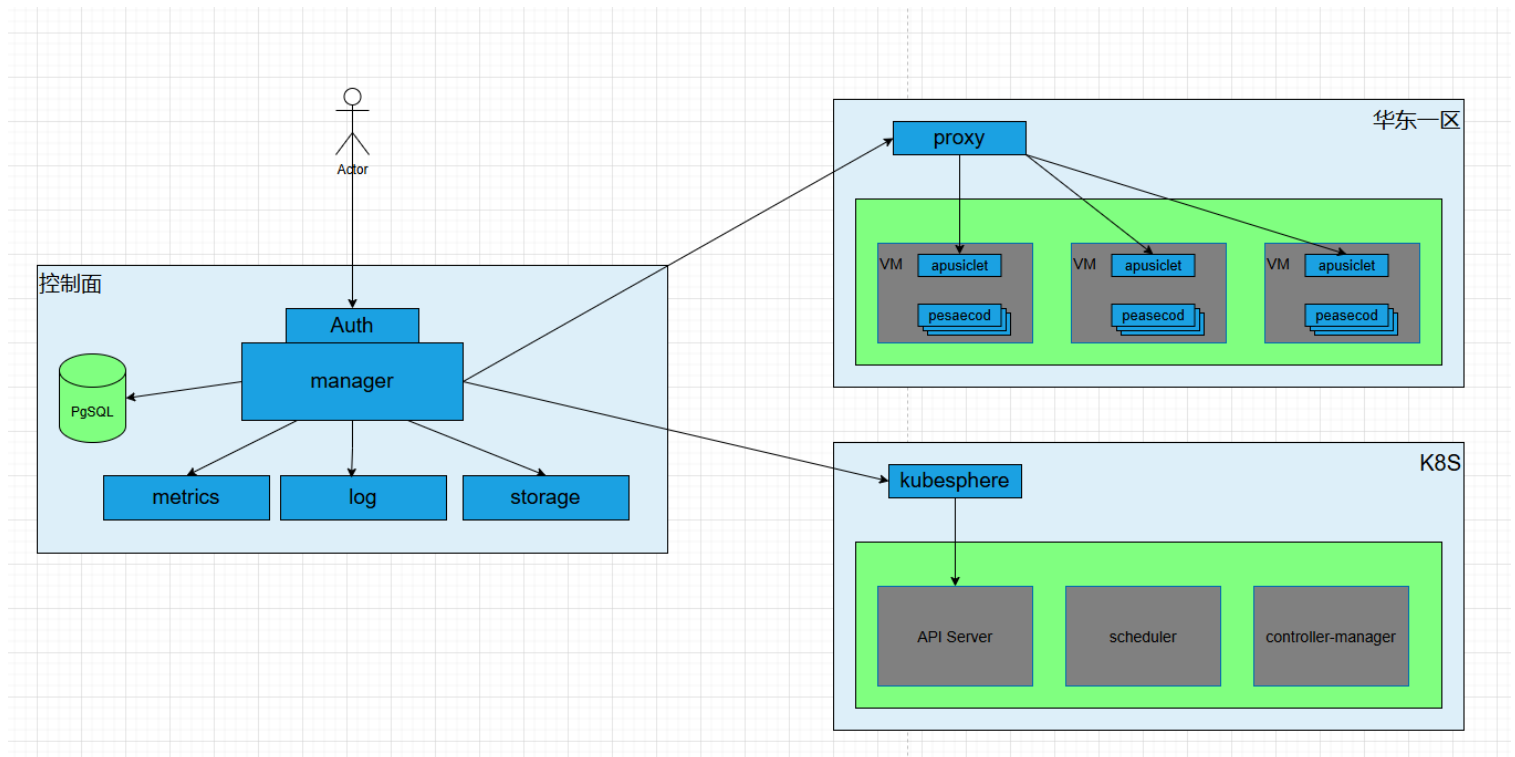
4. maas-resource

- 功能:
 - 云计算平台：选择合适的云计算平台（如 阿里云、腾讯云等）作为云资源提供方，确保与现有架构的兼容性和适配。
 - 虚拟机与容器管理：在云端创建虚拟机实例或容器化的服务来承载应用，支持弹性扩展和负载均衡。
 - 云存储与数据库：对接云端存储服务（如 AWS S3、阿里云 OSS）和数据库服务（如 AWS RDS、Google Cloud SQL），使数据可以轻松迁移、存储并高效访问。
 - 容器服务与编排：使用云平台的容器服务（如 Kubernetes、Docker Swarm 等）进行资源管理，确保云资源的自动化部署和扩展。

高可用部署架构图：



3.2 技术架构



4 产品安装

ACP 支持单机、 高可用模式 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 高可用模式 。

高可用部署方案，需要开放的防火墙策略（表中 IP 地址为示例网段，请替换为实际规划的地址）：

序号	源 IP 地址(发起请求的 IP)	发起请求的系统名称	目的 IP 地址(提供服务的 IP)	目的端口	服务协议	提供服务的系统名称	用途
1	192.168.1.0/24	apusiclet	192.168.0.1-3	9800	HTTP	maas-gateway -> maas-aump	获取软件包下载地址
2	192.168.1.0/24	apusiclet	192.168.0.1-3	9851	GRPC	maas-aump	中间件安装及安装事件处理
3	192.168.0.1-3	maas	192.168.0.1-3	8761	HTTP	maas-register	服务注册与发现
4	192.168.0.4-6	maas	192.168.0.1-3	5432	TCP	postgres	关系型数据库
5	192.168.0.4-6	maas	192.168.0.1-3	9000	HTTP	minio	软件包存放目录
6	192.168.0.4-6	maas	192.168.0.1-3	9200	TCP	elastic	搜索引擎
7	192.168.0.4-6	maas	192.168.0.1-3	9090	HTTP	prometheus	指标监控时序数据库

注： 192.168.0.1-3 分别为 192.168.0.1, 192.168.0.2, 192.168.0.3 三个 IP，主要用于部署 MAAS 高可用集群和 MAAS 所需中间件。

4.1 支持的环境

平台类型	系统类型
芯片类型	华为鲲鹏、海思、飞腾、兆芯等 x86/ARM 架构芯片
操作系统	银河麒麟系列、统信UOS、中标麒麟等
其他 Linux 系列	RedHat 系列、CentOS 系列、Ubuntu 系列等

4.2 推荐硬件配置

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

4.3 部署前准备

4.3.1 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； 华为 Kunpeng 920； 飞腾 FT1500a 等
浏览器	FireFox 70及以上、Chrome 60及以上、IE 11及以上
JDK	ACP 平台默认自带 JDK(17.0.10+11-LTS-240)，若虚拟机中默认设置了 JAVA_HOME 以系统环境变量优先，unset JAVA_HOME 可以跳过系统 JDK

4.3.2 软件要求

数据库

ACP的主要业务数据存储在关系型数据库中。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、PostgreSQL、瀚高、达梦、金仓等多种类型的关系数据库。

ACP 默认使用 PostgreSQL，PostgreSQL 安装可参考附录。

提示：生产环境推荐使用商业数据库。

4.3.3 关于License

1. 使用官方授权的有效期内license文件。
2. 使用金蝶天燕官方授权中心。
3. 使用KBC统一授权中心（Kingdee Authorization Center，金蝶授权中心）。

4.3.4 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。

4.3.5 安装介质说明

ACP V8.0.5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-8.0.5-20260630-Linux-amd64.tar.gz	Web管控台，包括云门户、管控中心、监控中心、日志中心等
管控代理	apusiclet-8.0.5.23-Linux-amd64.rpm	用于中间件全生命周期管理、暴露指标、上报日志等

4.3.6 依赖详情

组件名称	默认端口	版本	备注
PostgreSQL	5432	15.6	关系型数据库
alertmanager	9093	0.26.0	监控告警
elasticsearch	9200 9300	7.17.24	日志存储
logstash	5044	8.16.1	日志收集
minio	9000	20210422	对象存储
prometheus	9090	2.37.9	监控服务
grafana	3000	9.5.0	监控图表

4.3.7 服务端口说明

中间件云平台

端口	对应组件	作用	访问范围	服务名称
8443	中间件云平台	反向代理 HTTPS	外部访问	maas-gateway
9800	中间件云平台	反向代理HTTP	外部访问	maas-gateway
9820	中间件云平台	云门户	内部访问	maas-manager
9830	中间件云平台	监控中心	内部访问	maas-monitor
9840	中间件云平台	资源中心-主机管理	内部访问	maas-resource
9870	中间件云平台	工单系统	内部访问	maas-workorder
9850	中间件云平台	中间件管理	内部访问	maas-aump
9851	中间件云平台	中间件管理	外部访问	maas-aump
9801	中间件云平台	前端服务 管控台	内部访问	maas-fe
9890	中间件云平台	agent	内部访问	apusiclet

- 指标监控

端口	对应组件	作用	访问范围	服务名称
9090	指标监控	指标后台管理	内部访问	prometheus
9093	指标监控	告警后台管理	内部访问	alertmanager
9094	指标监控	告警集群间通讯	内部访问	alertmanager

- 日志服务

端口	对应组件	作用	访问范围	服务名称
9200	日志服务	日志引擎 API 接口	内部访问	elasticsearch
9300	日志服务	日志集群间通讯	内部访问	elasticsearch
5044	日志服务	日志解析	内部访问	logstash

- 存储服务

端口	对应组件	作用	访问范围	服务名称
----	------	----	------	------

9000	存储服务	后台管理页	外部访问	minio
9001	存储服务	API 端口	外部访问	minio

- 注册中心

端口	对应组件	作用	访问范围	服务名称
8761	注册中心	服务注册发现	内部访问	maas-register

4.4 环境检查

4.4.1 关闭防火墙和 SELinux

为确保 ACP 各组件之间网络通信正常，建议安装前根据部署环境调整防火墙和 SELinux 配置：**测试/演示环境可直接关闭防火墙与 SELinux；生产环境推荐开启防火墙，并按本文档「服务端口说明」章节开放对应端口（仅关闭 SELinux 或按需配置为 `permissive`）。**

1. 关闭防火墙

```
systemctl stop firewalld
systemctl disable firewalld
```

2. 关闭 SELinux

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
```

提示：生产环境如需开启防火墙，请根据本文档中的服务端口说明开放相应端口。

安全提示：ACP 管控台的出厂默认账号为 `acpadmin`，默认密码为 `admin`。请勿在生产环境使用默认密码，首次登录后请立即修改。

4.4.2 JDK

避免 JDK 冲突，建议卸载服务器上已经安装的 JDK。

4.4.3 安装依赖

安装之前需要安装以下软件：

```
yum install -y wget netstat fontconfig
```

4.4.4 内核参数

按以下步骤修改系统内核参数及用户资源限制，确保配置永久生效。

步骤 1: 修改用户资源限制配置

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

文件末尾添加以下内容：

```
# 方式1：对所有普通用户生效
* soft nofile 65535
* hard nofile 65535

# 方式2：单独给指定用户生效（推荐，更安全）
# appuser soft nofile 65535
# appuser hard nofile 65535

# root用户也配置上避免遗漏
root soft nofile 65535
root hard nofile 65535
```

参数说明：

- `soft`：软限制，程序默认生效值，可在硬限制内随意调整
- `hard`：硬上限，普通用户最大只能设置到这个数值，只有 `root` 能调高

步骤 2: 确保 PAM 模块加载限制（CentOS7 必做，否则不生效）

```
vim /etc/pam.d/sshd
```

确认存在这一行（没有就手动添加）：

```
session required pam_limits.so
```

步骤 3：调高系统内核全局最大文件句柄（兜底限制）

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

添加：

```
vm.max_map_count=262144  
fs.file-max = 655350  
fs.nr_open = 1048576
```

生效内核配置：

```
sysctl -p
```

步骤 4：解决 CentOS7 systemd 后台服务不生效问题（Java / 中间件常用）

如果程序通过 systemd 托管启动，需要额外修改 systemd 全局限制：

```
vim /etc/systemd/system.conf  
vim /etc/systemd/user.conf
```

两个文件都添加：

```
DefaultLimitNOFILE=65535
```

重新加载 systemd 配置：

```
systemctl daemon-reload
```

验证是否生效

提示：上述修改需要退出当前 SSH 会话并重新登录后才会加载。

重新登录后执行验证：

```
ulimit -Sn  
ulimit -Hn
```

4.5 安装前准备

1. 创建非 Root 用户（如 apusic），并拥有其账户权限

```
sudo groupadd apusic  
sudo useradd -m -g apusic apusic  
# 切换至 apusic 用户  
su apusic
```

2. 将安装包传输到用户目录（如/home/apusic）
3. 解压安装包

```
tar -xzvf ACP-MAAS-8.0.5-20260630-Linux-amd64.tar.gz
```

4.6 开始安装

4.6.1 安装平台组件

ACP 平台依赖数据库、监控、存储等组件，支持一键安装平台的所有组件

注意：

- ACP 提供的 PostgreSQL 数据库仅供测试使用，我们建议生产环境使用商业数据库。
- 目前，我们支持的 PostgreSQL 仅适用于 amd64 架构 Ubuntu、Kylin、Centos、OpenEuler、Redhat，aarch64 架构支持 Kylin、OpenEuler、Redhat、Ubuntu 操作系统，其他自行安装 PostgreSQL

1. 安装平台组件

```
cd /home/apusic/maas-8.0.5  
# 安装所有组件
```

```
./bin/maas.sh middleware install
```

安装成功后，控制台会输出类似如下日志：

```
[INFO] Starting middleware components installation...
[INFO] Postgresql installation completed.
[INFO] Minio installation completed.
[INFO] Elasticsearch installation completed.
[INFO] Logstash installation completed.
[INFO] Prometheus installation completed.
[INFO] Alertmanager installation completed.
[INFO] Grafana installation completed.
[INFO] All middleware components installation completed.
```

提示：当出现“All middleware components installation completed.”，则表示安装成功

2. 启动平台组件

启动所有组件

```
./bin/maas.sh middleware start
```

提示：当出现“All middleware components launched successfully.”，则表示启动成功。

```
```text
14:40:38.68 INFO ==> Start launching all middleware components...
14:40:38.68 INFO ==> Starting postgresql...
14:40:38.81 INFO ==> start postgresql success!
14:40:38.81 INFO ==> postgresql started successfully.
14:40:38.81 INFO ==> Starting minio...
14:40:38.82 INFO ==> start minio success!
14:40:38.83 INFO ==> minio started successfully.
14:40:38.83 INFO ==> Starting elasticsearch...
14:40:38.83 INFO ==> start elasticsearch success!
14:40:38.84 INFO ==> elasticsearch started successfully.
14:40:38.84 INFO ==> Starting logstash...
14:40:38.84 INFO ==> start logstash success!
```

```

14:40:38.85 INFO ==> logstash started successfully.
14:40:38.85 INFO ==> Starting prometheus...
14:40:38.85 INFO ==> start prometheus success!
14:40:38.85 INFO ==> prometheus started successfully.
14:40:38.86 INFO ==> Starting grafana...
14:40:38.86 INFO ==> start grafana success!
14:40:38.87 INFO ==> grafana started successfully.
14:40:38.87 INFO ==> Starting alertmanager...
14:40:38.87 INFO ==> start alertmanager success!
14:40:38.87 INFO ==> alertmanager started successfully.
14:40:38.87 INFO ==> Starting license...
14:40:38.88 INFO ==> start license server success!
14:40:38.88 INFO ==> license started successfully.
14:40:38.88 INFO ==> All middleware components launched.
...

```

### 3. 查看组件状态

#### # 查看所有组件

```
./bin/maas.sh middleware status
```

提示：当组件状态不是 *running* 时，可以在 *logs* 目录下查看相关日志

提示：单机部署时 Elasticsearch 显示 `WARNING state (YELLOW)` 属于正常现象。这是因为单节点 ES 默认 `YELLOW` 状态（集群缺少副本分片），不影响日志收集与查询功能；仅在多节点高可用集群部署下才会自动转为 `GREEN` 状态。若需消除告警日志，可通过调整 ES 副本分片数为 0 来实现（生产高可用集群不建议此操作）。

```

14:39:48.55 INFO ==> Checking status of all middleware
components...
 14:39:48.55 INFO ==> Checking postgresql status...
pg_ctl: server is running (PID: 1796)
/home/apusic/maas-8.0.5/service/postgresql/bin/postgres "-D"
"/home/apusic/maas-8.0.5/service/postgresql/data"
 14:39:48.56 INFO ==> Checking minio status...
MinIO is running successfully!
 14:39:48.59 INFO ==> Checking elasticsearch status...

```

```

14:39:48.62 INFO ==> ⚠ Elasticsearch is in WARNING state (YELLOW)
14:39:48.62 INFO ==> Checking logstash status...
Logstash is healthy (GREEN)
14:39:48.71 INFO ==> Checking prometheus status...
14:39:48.73 INFO ==> Prometheus is healthy
14:39:48.73 INFO ==> Checking grafana status...
GRAFANA is healthy (GREEN)
14:39:48.74 INFO ==> Checking alertmanager status...
14:39:48.77 INFO ==> Alertmanager is running successfully!
14:39:48.77 INFO ==> Checking license status...
14:39:48.79 INFO ==> license-server is running successfully!

+-----+-----+-----+-----+
| Middleware | Health | Status | Description |
+-----+-----+-----+-----+
| postgresql | healthy | running | Service is running |
| minio | healthy | running | Service is running |
| elasticsearch | healthy | running | Service is running |
|
| logstash | healthy | running | Service is running |
| prometheus | healthy | running | Service is running |
| grafana | healthy | running | Service is running |
| alertmanager | healthy | running | Service is running |
|
| license | healthy | running | Service is running |
+-----+-----+-----+-----+

14:39:48.80 INFO ==> All middleware components status checked.

```

#### 4. 创建数据库及用户

```

初始化postgresql
./bin/maas.sh middleware init postgresql

```

提示：记住这里填入的数据库名、用户名、密码，后续安装步骤会使用

## 4.6.2 安装管控台

### 1. 修改配置文件

- 修改网关地址

```
cd /home/apusic/maas-8.0.5
vim conf/application-prod.properties

如下 localhost 修改为真实的 IP。
apusic.gateway.addr=http://localhost:9800
```

- 修改数据库连接配置（以 PostgreSQL 为例）：

```
vim conf/application-postgresql.properties
数据库名称、用户、密码请与安装数据库时输入的参数保持一致。
apusic.datasource.host=localhost
apusic.datasource.port=5432
apusic.datasource.username=root
apusic.datasource.password=[YOUR_PASSWORD]
apusic.datasource.database=maas-release_v805
```

提示：数据库密码如需使用密文方式，需要将 `apusic.datasource.encrypt.enabled` 设置为 `true`。同时通过如下命令 `./bin/maas.sh generate {password}` 获取密码密文，并进行配置。

### 2. 执行启动脚本

```
./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：

```
14:44:06.67 后台启动 ...
3087
14:44:06.71 INFO ==> start console success ...
```

```
14:44:06.72 start maas application maas-register-8.0.5.jar
14:44:06.72 启动配置文件为: /home/apusic/maas-
8.0.5/conf/application-prod.properties
14:44:06.75 INFO ==> start maas-register-8.0.5.jar success
...
14:44:06.76 start maas application maas-resource-8.0.5.jar
14:44:06.77 启动配置文件为: /home/apusic/maas-
8.0.5/conf/application-prod.properties
14:44:06.77 INFO ==> start maas-resource-8.0.5.jar success
...
14:44:06.78 start maas application maas-manager-8.0.5.jar
14:44:06.79 启动配置文件为: /home/apusic/maas-
8.0.5/conf/application-prod.properties
14:44:06.79 INFO ==> start maas-manager-8.0.5.jar success ...
14:44:06.80 start maas application maas-gateway-8.0.5.jar
14:44:06.80 启动配置文件为: /home/apusic/maas-
8.0.5/conf/application-prod.properties
14:44:06.81 INFO ==> start maas-gateway-8.0.5.jar success ...
14:44:06.82 start maas application maas-aump-8.0.5.jar
14:44:06.82 启动配置文件为: /home/apusic/maas-
8.0.5/conf/application-prod.properties
14:44:06.83 INFO ==> start maas-aump-8.0.5.jar success ...
14:44:06.83 start maas application maas-cloudlog-8.0.5.jar
14:44:06.83 启动配置文件为: /home/apusic/maas-
8.0.5/conf/application-prod.properties
14:44:06.85 INFO ==> start maas-cloudlog-8.0.5.jar success
...
```

### 3. 查看管控状态

```
./bin/maas.sh status
```

若所有组件均UP，则说明安装成功。

注册中心与服务发现服务 (maas-register) :运行中, PID: 3103

资源服务 (maas-resource) :运行中, PID: 3116

管理服务 (maas-manager) :运行中, PID: 3133

网关服务 (maas-gateway) :运行中, PID: 3162

中间件服务 (maas-aump) :运行中, PID: 3180

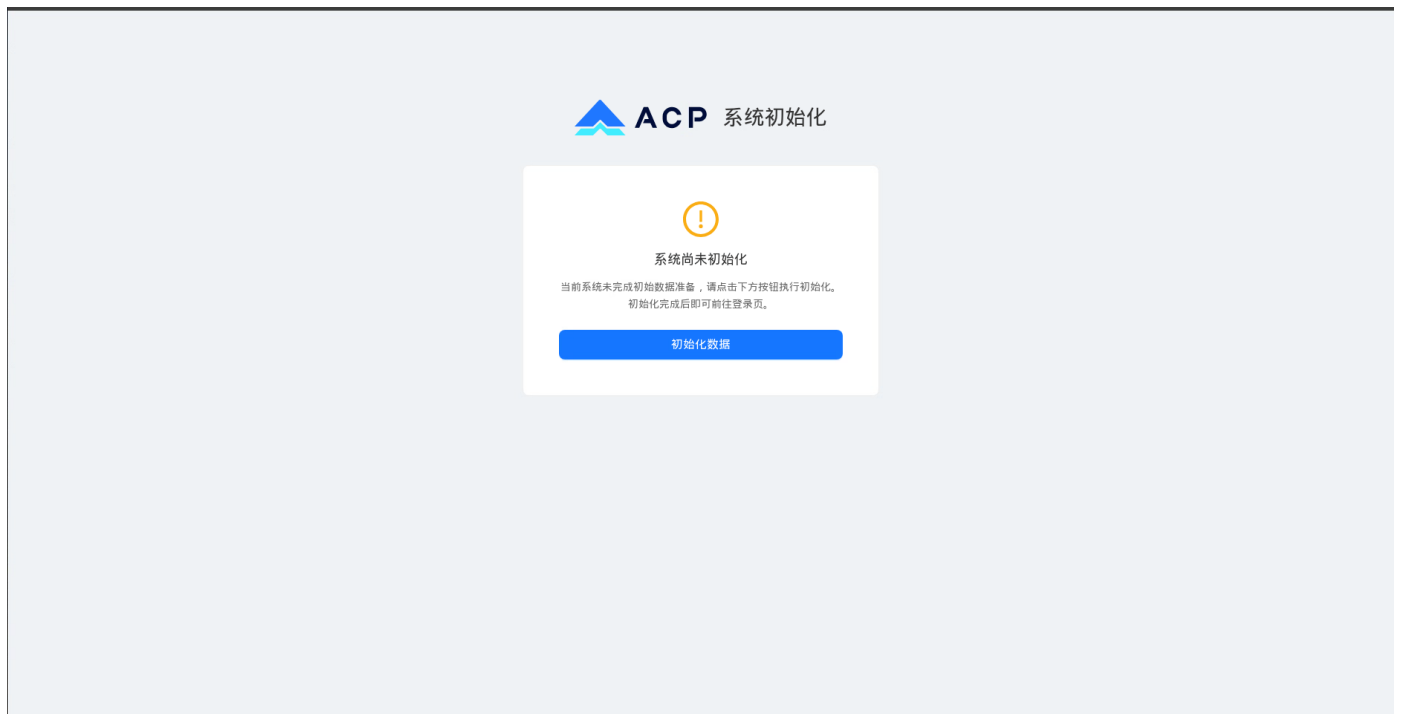
日志服务 (maas-cloudlog) :运行中, PID: 3217

#### 4. 访问

浏览器访问 <http://IP:9800>，使用出厂默认账号登录（默认凭据及修改要求见「环境检查」章节的安全提示），\*\*首次登录后请立即修改密码\*\*

#### 5. 数据初始化

首次访问 ACP 管控台时，若系统检测到数据尚未初始化，页面会自动跳转到数据初始化页面，如下图所示：



请按照页面提示完成数据初始化操作。初始化完成后，系统将自动跳转到管控台登录页面，使用账号登录即可正常使用。

**提示：**数据初始化仅需在首次部署时执行一次，后续重启或升级无需再次操作。

## 4.6.3 授权激活

### 获取授权

首次访问 ACP 平台需完成授权操作，具体步骤如下：

1. 首次登录 ACP 后，页面弹出“授权校验失败”提醒，点击【跳转授权】按钮，进入平台授权管理页面。



2. 在【平台运营】-【授权管理】-【集中授权】页面中，点击【软件特征码】选项，获取当前平台的特征码。

ACP 平台首页

文件授权 集中授权 在线实例

产品名称 发行版本

导入

软件特征码

命名空间	产品名称	发行版本	授权类型	总数	已用	开始时间	结束时间	签发时间	操作
深圳市金蝶天策云计算股份有限公司	Apusic In-Memory Data Cache	Standard Edition	实例授权	100	0	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看
深圳市金蝶天策云计算股份有限公司	Apusic Distributed Message Queue	Standard Edition	实例授权	200	0	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看
深圳市金蝶天策云计算股份有限公司	Apusic Distributed Message Queue	Agile Edition	实例授权	100	0	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看
深圳市金蝶天策云计算股份有限公司	Apusic Load Balancer	Standard Edition	实例授权	100	0	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看
深圳市金蝶天策云计算股份有限公司	Apusic Load Balancer	Agile Edition	实例授权	100	0	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看
深圳市金蝶天策云计算股份有限公司	Apusic Copilot	Standard Edition	实例授权	1	0	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看
深圳市金蝶天策云计算股份有限公司	Apusic Distributed Config Center	Standard Edition	实例授权	100	0	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看
深圳市金蝶天策云计算股份有限公司	Apusic Search Engine	Standard Edition	实例授权	200	0	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看
深圳市金蝶天策云计算股份有限公司	Apusic Application Server	Enterprise Edition	实例授权	100	1	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看
深圳市金蝶天策云计算股份有限公司	Apusic Application Server	Agile Edition	实例授权	100	0	2025-12-01 11:00:00	2026-03-02 10:59:59	2025-12-02 02:04:09	查看

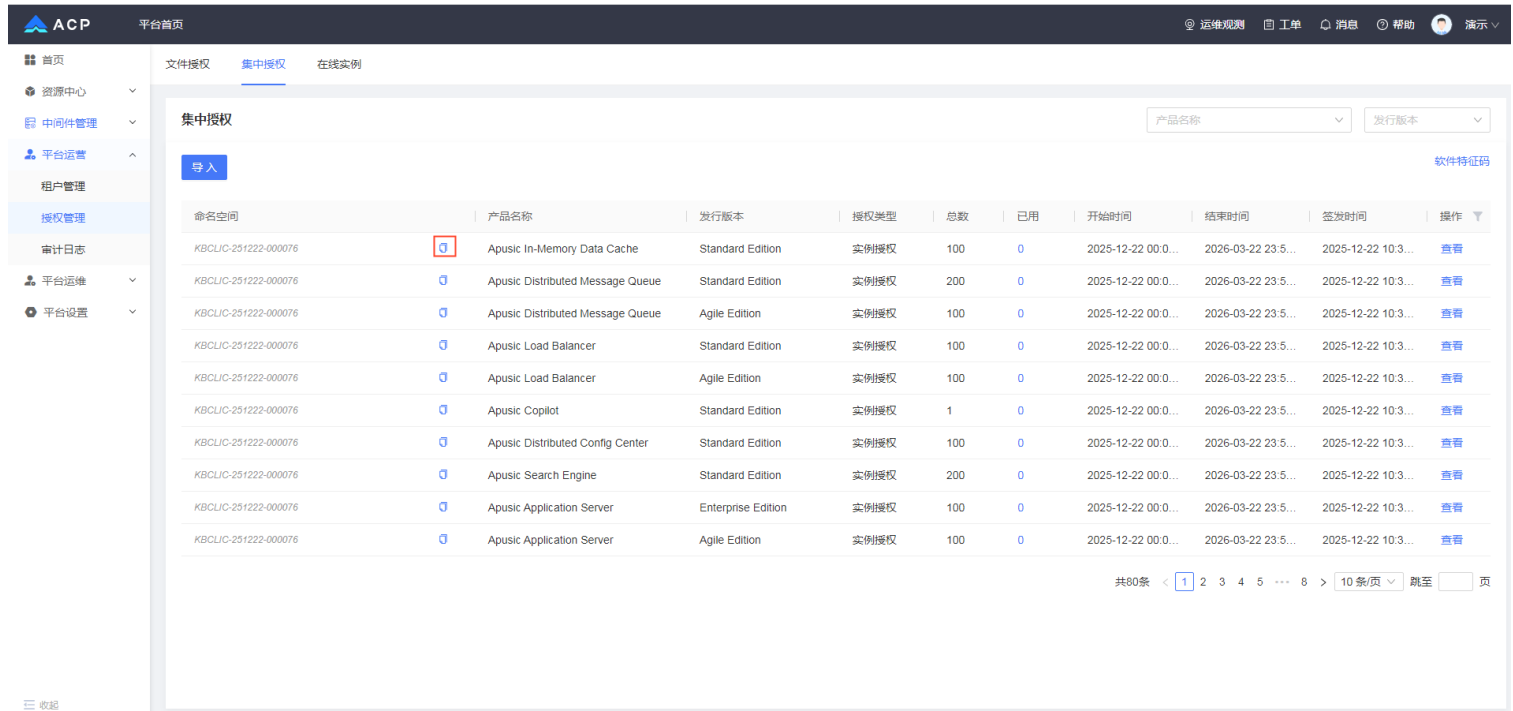
共14条 < 1 2 > 10 条页 跳至 页

3. 联系技术支持人员，提供上述特征码完成平台激活，并获取对应授权。

### 导入授权

获取授权文件（.lic文件）后，需要完成如下操作：

1. 在【平台运营】-【授权管理】-【集中授权】页面中，点击【导入】按钮，导入授权文件后，点击“命名空间”复制按钮获取命名空间



2. 修改 ACP 安装目录下的授权配置acls.properties

```
cd /home/apusic/maas-8.0.5
vim acls.properties
...
修改下述配置为上一步骤授权列表中的命名空间（下值为示例，请替换为实际命名空间）
apusic_acls_ns=KBCLIC-XXXXXX-XXXXXX
修改 localhost 为具体 ip
apusic_acls_authUrls=192.0.2.217:6869
```

提示：本步骤是配置 ACP 自身授权

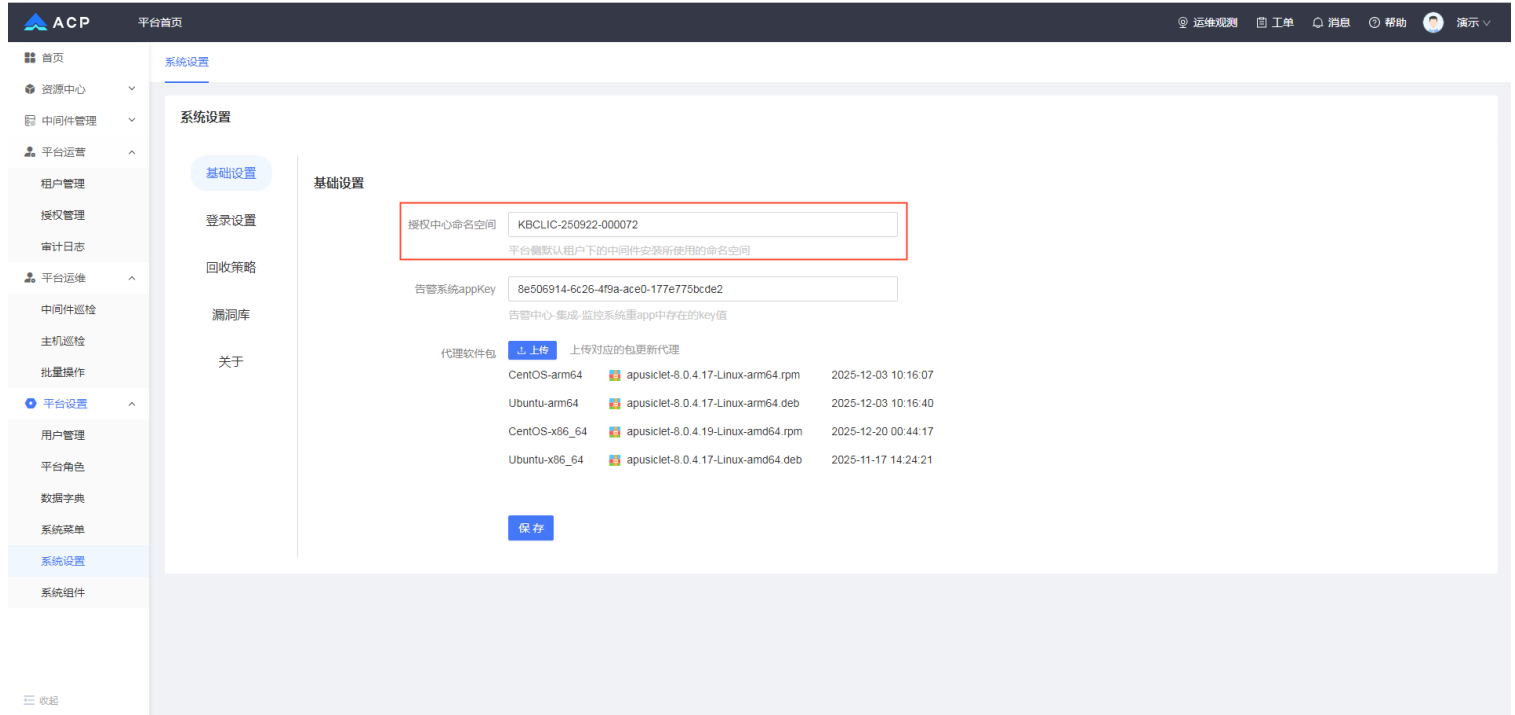
3. 重新启动 ACP

```
cd /home/apusic/maas-8.0.5
./bin/maas.sh stop all
./bin/maas.sh start all -d
```

最后，浏览器访问 <http://IP:9800>，使用默认账号登录（\*\*首次登录后请立即修改密码\*\*）。

## 修改授权中心命名空间

在【平台设置】-【基础设置】页面，修改授权中心命名空间。

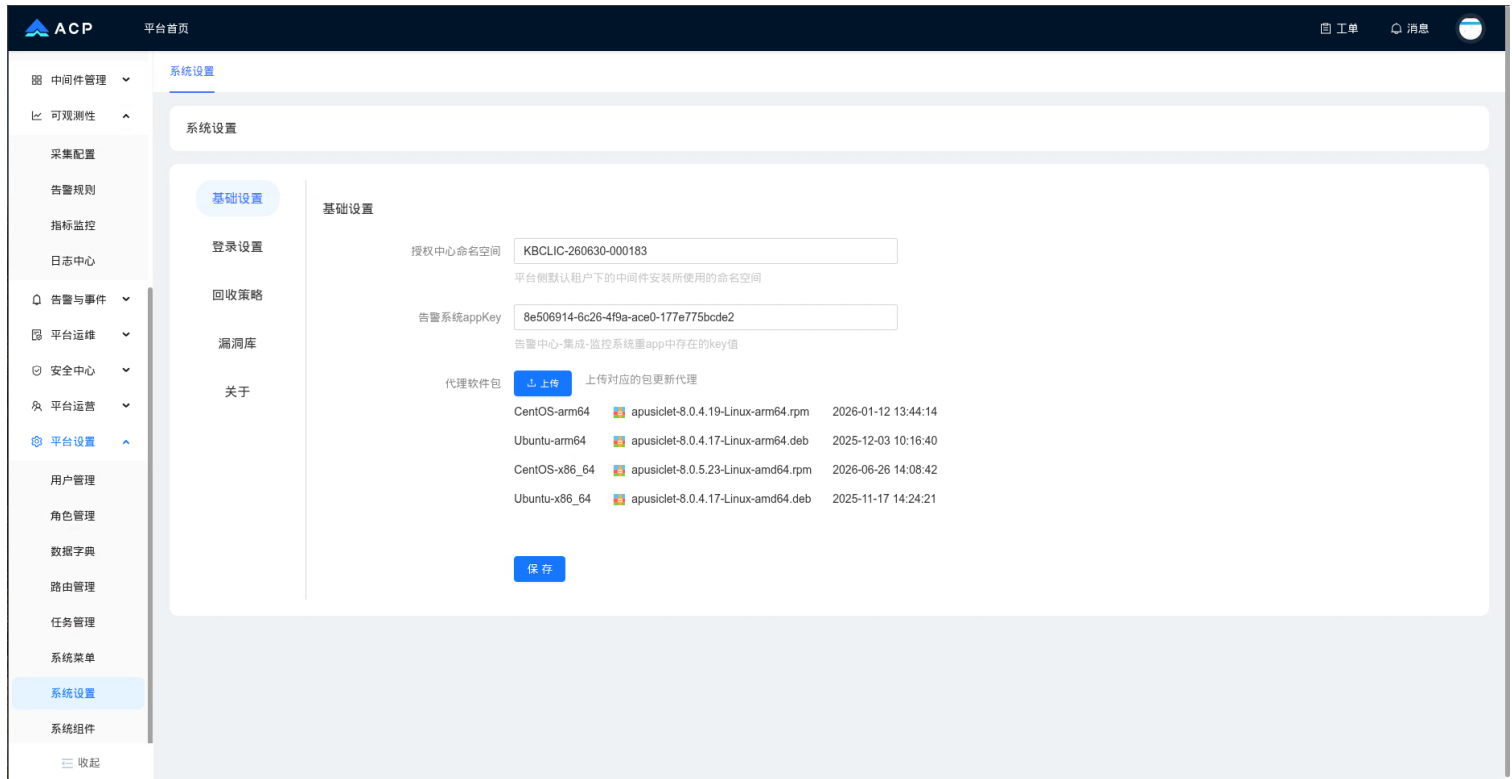


提示：通过 ACP 安装的 Apusic 中间件默认采用授权中心模式，本步骤是配置中间件使用的授权中心地址。

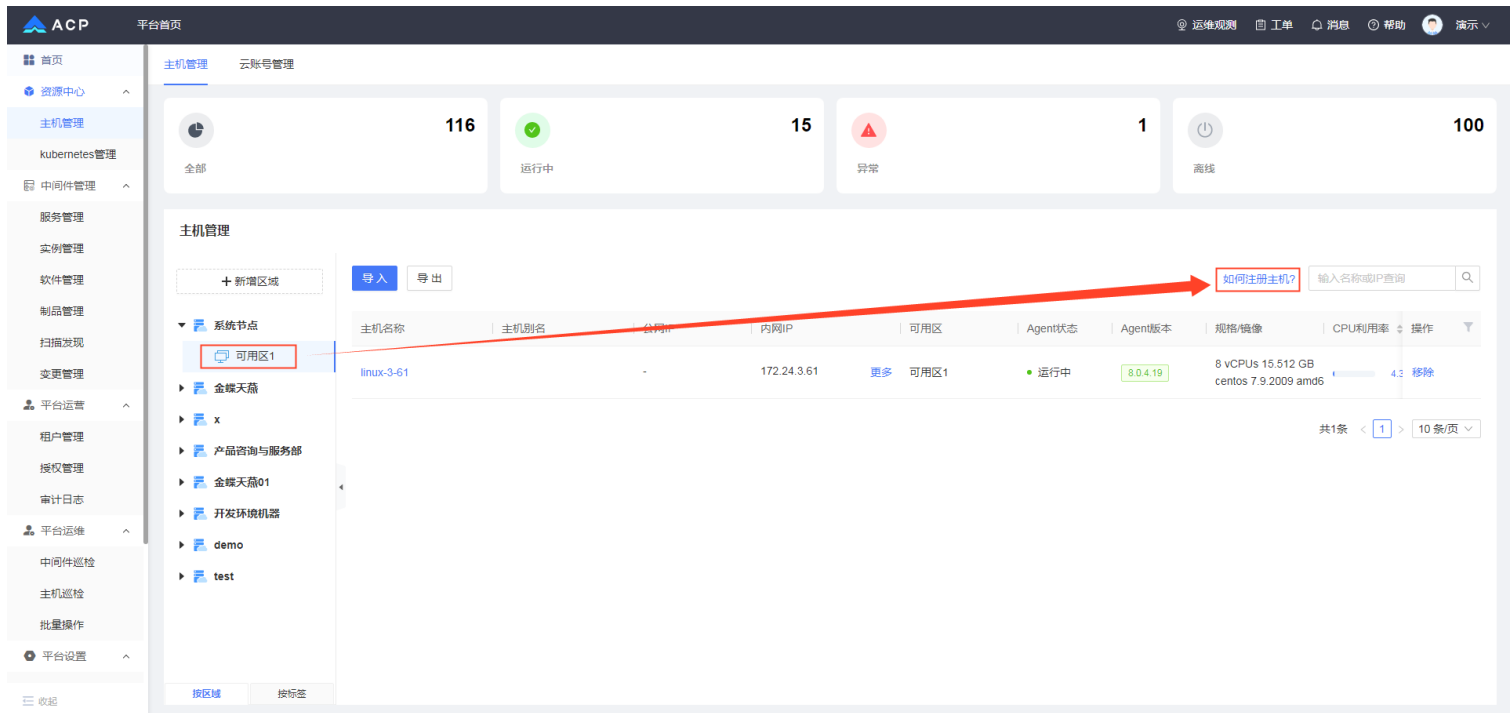
## 4.6.4 注册系统节点

1. 【平台设置】-【系统设置】-【基础设置】中上传代理软件包

提示：上传完成后记得点击保存



2. 在【资源中心】-【主机管理】页面，选择系统节点，点击打开页面上的“如何注册主机”，按指导在部署 ACP 的服务器上完成注册操作



## 4.7 高可用部署

安装高可用的方式是以单机的方式在三台机器上面安装好 MAAS 服务之后将 MAAS 组成 高可用模式。

提示: Eureka 服务为 MAAS 服务中的 maas-register 服务, 默认端口为 8761, 默认在安装 MAAS 服务的时候已经安装, 不需要另外安装 Eureka 集群。

修改所有节点 maas 的配置文件。

```
vim conf/application-prod.properties
```

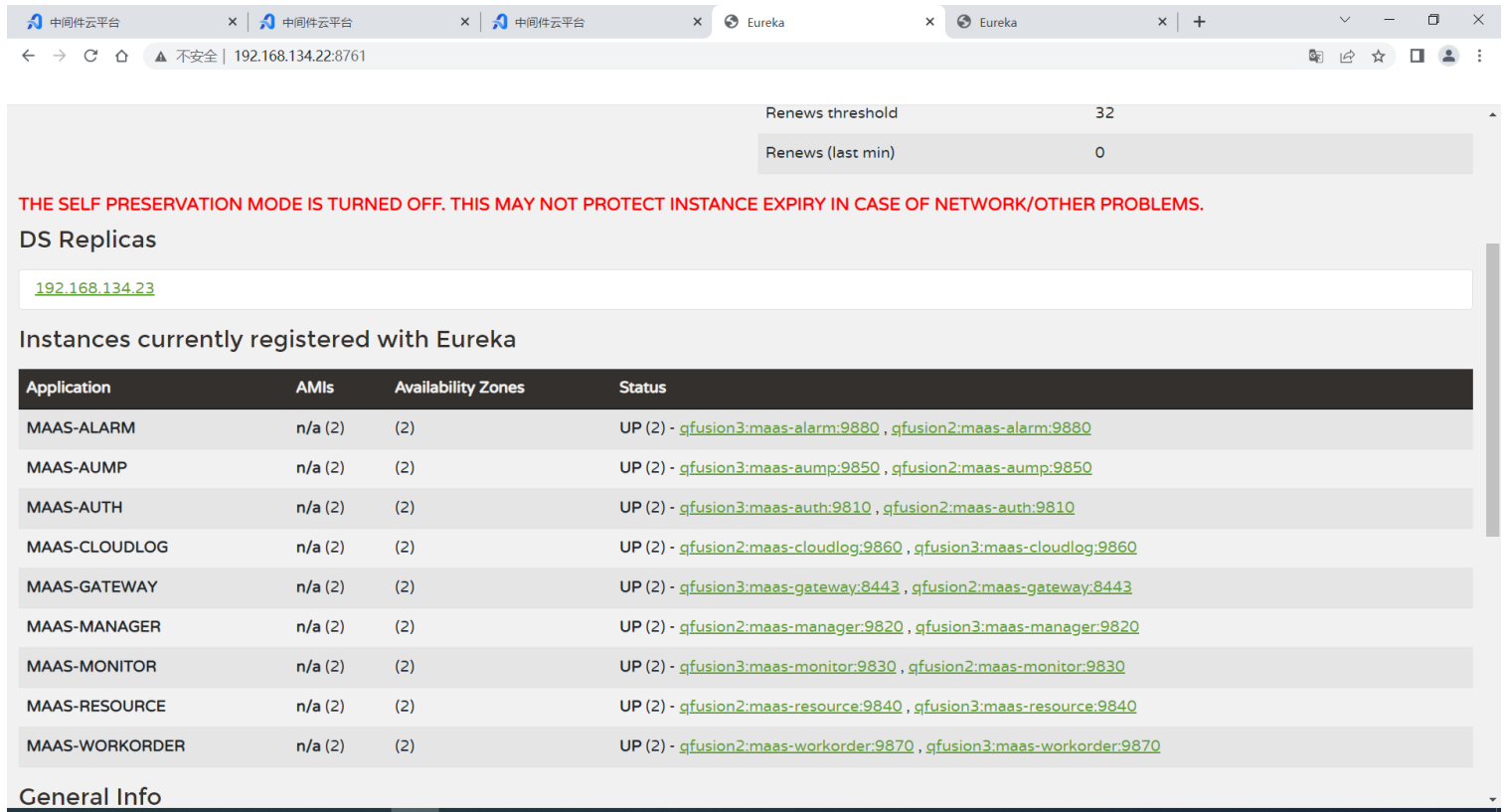
```
当前节点的主机名或 IP, 集群模式的时候需要开启
eureka.instance.hostname=localhost
--- Eureka ---
此处是注册中心的地址, 部署单个节点时此处不需要修改; 部署高可用集群时
此处需要配置其他两个节点的 Eureka 节点地址, 默认其他节点使用 8761 端口
部署高可用时此处需要配置其他节点服务的监听地址 (示例 IP, 请替换为实际地址)
例如: server1 http://198.51.100.61:8761/eureka
server2 http://198.51.100.62:8761/eureka
server3 http://198.51.100.63:8761/eureka
server1 配置文件如下:
apusic.registry.server.address=http://198.51.100.62:8761/eureka,http://
server2 配置文件如下:
apusic.registry.server.address=http://198.51.100.61:8761/eureka,http://
server3 配置文件如下:
apusic.registry.server.address=http://198.51.100.61:8761/eureka,http://
如果当前 Eureka 配置了用户名密码的话, 按照上面的配置添加用户密码, 例如:
#
apusic.registry.server.address=http://${apusic.registry.user.name}:${ap
#
http://${apusic.registry.user.name}:${apusic.registry.user.password}@19
以此类推
eureka 使用的用户名密码 (示例值, 生产环境请修改)
apusic.registry.user.name=apusic
apusic.registry.user.password=[YOUR_PASSWORD]
apusic.registry.server.address=http://${apusic.registry.user.name}:${ap
默认不需要修改
apusic.registry.server.port=8761
当前节点的主机名或 IP, 集群模式的时候需要开启
eureka.instance.hostname=localhost
```

```

当前节点的主机名或 IP, 集群模式的时候需要开启
eureka.instance.hostname=localhost
--- Eureka 客户端地址 ---
如果是单机部署的话只用连接本地的 Eureka 地址, 此处只需要填写
http://localhost:8761/eureka 地址就好了
如果部署的是一个高可用的 Eureka 集群, 这个地址我们只需要改成所有的 Eureka 地
址。
例如: server1 http://198.51.100.61:8761/eureka
server2 http://198.51.100.62:8761/eureka
server3 http://198.51.100.63:8761/eureka
所以这个配置如下: eureka.client.service-
url.defaultZone=http://198.51.100.61:8761/eureka,http://198.51.100.62:8
eureka.client.service-
url.defaultZone=http://${apusic.registry.user.name}:${apusic.registry.t
eureka.instance.metadata-map.zone=local
心跳间隔, 10秒发送一次
eureka.instance.lease-renewal-interval-in-seconds=10
心跳过期时间, 90秒内未收到心跳则认为服务不可用
eureka.instance.lease-expiration-duration-in-seconds=90
启用注册表拉取 客户端会定期向 Eureka Server 拉取服务注册表, 确保获取最新的服
务信息。如果在拉取时连接失败, 客户端会进行重试。
eureka.client.fetch-registry=true
注册表拉取超时10秒->设置客户端获取注册表时的超时时间。如果超过这个时间还未能获
取到注册表, 客户端会失败并重新尝试。
eureka.client.registry-fetch-timeout-seconds=10
eureka.client.register-with-eureka=true

```

修改完成后, 启动所有节点的 MAAS 服务, 并等待各节点上的 MAAS 服务完全启动和运行正常。接着, 依次访问每个节点的 8761 端口。当显示的服务状态数量与当前集群的节点数量一致时, 说明集群已成功组建, 且各节点已互为高可用集群。



提示：安装完成后需要激活平台，才能正常使用

## 4.8 常见问题与排错

本章节汇总安装部署过程中的高频故障及排查方法，建议优先按本节定位，再结合 09.faq.md 查阅更多案例。

### 4.8.1 1. 端口冲突（启动失败提示 `BindException: Address already in use`）

现象：执行 `maas.sh middleware start` 或 `maas.sh start all -d` 后，控制台输出 `BindException` 或 `port already in use`。

排查步骤：

```
1. 查询占用目标端口的进程（如 9800、8761、5432）
netstat -tlnp | grep <端口号>

2. 确认是否为 ACP 组件残留进程
ps -ef | grep -E
'maas|apusic|postgres|minio|elasticsearch|logstash|prometheus|grafana|a
```

解决方案：

- 若为 ACP 组件残留，使用 `./bin/maas.sh stop all` 停止后重新启动。
- 若为其他业务进程占用，需更换 ACP 部署端口或在对方服务中释放端口（生产环境推荐前者）。
- 关键端口参考：`maas-gateway:9800/8443`、`maas-register:8761`、`postgresql:5432`、`minio:9000/9001`、`elasticsearch:9200/9300`、`prometheus:9090`、`alertmanager:9093/9094`、`logstash:5044`、`grafana:3000`。

#### 4.8.2 2. 服务启动失败（`status` 显示非 `running`）

现象：`./bin/maas.sh middleware status` 或 `./bin/maas.sh status` 中部分组件显示 `stopped` / `failed` / `unhealthy`。

排查步骤：

```
进入 ACP 安装目录
cd /home/apusic/maas-8.0.5

查看对应组件日志（以 elasticsearch 为例）
ls logs/
tail -n 200 logs/elasticsearch/elasticsearch.log
```

常见原因及处置：

现象	原因	处置
<code>max file descriptors</code> 错误	文件句柄上限不足	按「环境检查 → 内核参数」章节调高 <code>nofile</code> 与 <code>vm.max_map_count</code> ，退出 SSH 重新登录
<code>JAVA_HOME not set</code> / <code>JDK version mismatch</code>	系统 JDK 与平台内置 JDK 冲突	执行 <code>unset JAVA_HOME</code> 后重试，或按「环境检查 → JDK」章节卸载系统 JDK
<code>Permission denied</code>	apusic 用户无目录读写权限	<code>chown -R apusic:apusic /home/apusic/maas-8.0.5</code>
<code>OutOfMemoryError</code>	JVM 内存不足	编辑 <code>bin/maas.sh</code> 或对应组件启动脚本中的 <code>-Xmx</code> 参数，重启组件
端口被防火墙拦截	<code>iptables/firewalld</code> 未放行	参见「服务端口说明」章节放行端口，或临时 <code>systemctl stop firewalld</code> 验证

#### 4.8.3 3. JVM 内存不足（`OutOfMemoryError` / 卡顿）

**现象：**管控台响应缓慢、组件日志频繁报 `OutOfMemoryError: Java heap space` 或 `GC overhead limit exceeded`。

**解决方案：**

1. 编辑 `conf/application-prod.properties` 或各组件独立配置文件，调整 JVM 堆内存（示例）：

```
单机部署推荐配置
JAVA_OPTS=-Xms4g -Xmx8g -XX:MetaspaceSize=512m -
XX:MaxMetaspaceSize=1g
```

2. 集群部署按节点规格线性扩展，建议每节点预留 `50%~70%` 物理内存给 ACP 进程。
3. 重启服务：`./bin/maas.sh restart all`。

#### 4.8.4 4. Elasticsearch 集群状态异常

**现象：**`status` 输出 `Elasticsearch is in WARNING state (YELLOW)` 或 `RED`。

**原因与处置：**

状态	含义	处置建议
GREEN	全部主分片与副本分片可用	正常，无需处理
YELLOW	主分片可用，副本分片未分配（单节点 ES 默认状态）	单机部署属正常；高可用集群需检查节点是否全部在线
RED	主分片不可用	立即排查 ES 节点存活状态、磁盘空间、JVM 内存

**补充排查命令：**

```
curl -X GET "http://<ES_IP>:9200/_cluster/health?pretty"
curl -X GET "http://<ES_IP>:9200/_cat/nodes?v"
```

#### 4.8.5 5. PostgreSQL 连接失败（`FATAL: password authentication failed`）

**现象：**管控台启动后访问报数据库连接失败，或 `init postgresql` 提示密码错误。

**排查步骤：**

1. 确认 `conf/application-postgresql.properties` 中的 `apusic.datasource.username` / `apusic.datasource.password` 与 `init postgresql` 步骤输入的参数一致。
2. 若使用密文密码，确认 `apusic.datasource.encrypt.enabled=true` 并使用 `./bin/maas.sh generate {password}` 重新生成密文。
3. 测试数据库连通性：

```
psql -h <host> -p 5432 -U <username> -d <database>
```

#### 4.8.6 6. 高可用集群 Eureka 节点未互相注册

**现象：**访问各节点 `:8761` 时，`Instances currently registered with Eureka` 中未包含其他节点。

**排查步骤：**

1. 确认所有节点的 `conf/application-prod.properties` 中 `apusic.registry.server.address` 已按 `server1 / server2 / server3` 示例配置成对端地址（含用户名密码占位符）。
2. 确认 `/etc/hosts` 或 DNS 中各节点主机名可互相解析。
3. 确认防火墙放行了 8761 端口（参见「服务端口说明」）。
4. 重启各节点 MAAS 服务后等待 30~60 秒，Eureka 心跳同步存在延迟。

#### 4.8.7 7. 授权激活失败（授权校验失败）

**现象：**导入 `.lic` 文件后仍提示授权校验失败，或 `acls.properties` 修改后未生效。

**排查步骤：**

1. 确认 `acls.properties` 中 `apusic_acls_ns` 与授权管理页面复制的命名空间完全一致（无多余空格/换行）。
2. 确认 `apusic_acls_authUrls` 中 IP:PORT 可达，防火墙已放行对应端口。
3. 执行 `./bin/maas.sh stop all && ./bin/maas.sh start all -d` 重启服务。
4. 若仍失败，导出 `logs/` 下 `license` 相关日志联系金蝶技术支持。

#### 4.8.8 8. 浏览器访问管控台 404 / 无法访问

**现象：**浏览器访问 `http://IP:9800` 报 404、空白页或拒绝连接。

**排查步骤：**

1. 确认服务已启动：`./bin/maas.sh status`（应全部 UP）。
2. 确认 `conf/application-prod.properties` 中 `apusic.gateway.addr` 已改为 `http://<本机 IP>:9800`（非 `localhost`）。
3. 确认防火墙已放行 9800 端口，或临时 `systemctl stop firewalld` 验证。

4. 清除浏览器缓存后重试（推荐 Chrome / Firefox 最新版）。

提示：更多 FAQ 请参考 [产品常见问题](#)；日志统一存放于 `<ACP安装目录>/logs/` 下，按组件分子目录归档。

---

更多内容请参考：

- [产品常见问题](#)
- [附录](#)

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。



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)

