Apusic 金蝶天燕

生成时间：2026-07-13

安装手册

金蝶Apusic分布式消息队列V2.0.6_for_kafka

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

11 技术支持

1 前言

本文档为金蝶Apusic分布式消息队列for Kafka（Apusic Distributed Message Queue，简称：ADMQ for Kafka）的安装手册，提供产品各种部署模式（含单机、集群）在各操作系统、容器环境的安装部署过程指导以及相关配置说明。

1.1 适用对象

本文档适用于IT信息化业务负责人、研发经理、软件项目经理、软件架构师、运维工程师。

1.2 相关文档

了解更多ADMQ for Kafka产品相关的信息，请参阅以下ADMQ for Kafka产品手册文档集：

序号	手册文档	说明
1	金蝶Apusic分布式消息队列for Kafka 快速使用手册	简单介绍了如何快速上手使用ADMQ for Kafka 。
2	金蝶Apusic分布式消息队列for Kafka 安装手册	详细介绍如何在各操作系统上安装ADMQ for Kafka，以及ADMQ for Kafka服务启停等操作。
3	金蝶Apusic分布式消息队列for Kafka 消息引擎用户手册	详细介绍 ADMQ for Kafka 消息引擎相关功能的使用、配置、管理及配套工具的使用方法。
4	金蝶Apusic分布式消息队列for Kafka 管控台用户手册	详细介绍ADMQ for Kafka管控台相关功能的使用和操作说明。
5	金蝶Apusic分布式消息队列for Kafka 开发手册	详细介绍基于各开发语言进行ADMQ for Kafka客户端应用开发的说明。
6	金蝶Apusic分布式消息队列for Kafka 迁移手册	详细介绍从Kafka迁移到ADMQ for Kafka的说明。
7	金蝶Apusic分布式消息队列for Kafka 运维手册	详细介绍ADMQ for Kafka的监控、运维、安全加固等运维说明。
8	金蝶Apusic分布式消息队列for Kafka 性能优化手册	详细介绍ADMQ for Kafka性能调优的说明。

1.3 技术支持

ADMQ for Kafka产品提供全面的技术支持服务，您可以通过以下方式获得技术支持：

- 网址：www.apusic.com
- 电话：400-855-5800
- 邮箱：support@apusic.com

- 金蝶云社区：https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN

您在取得技术支持时，请提供如下信息：

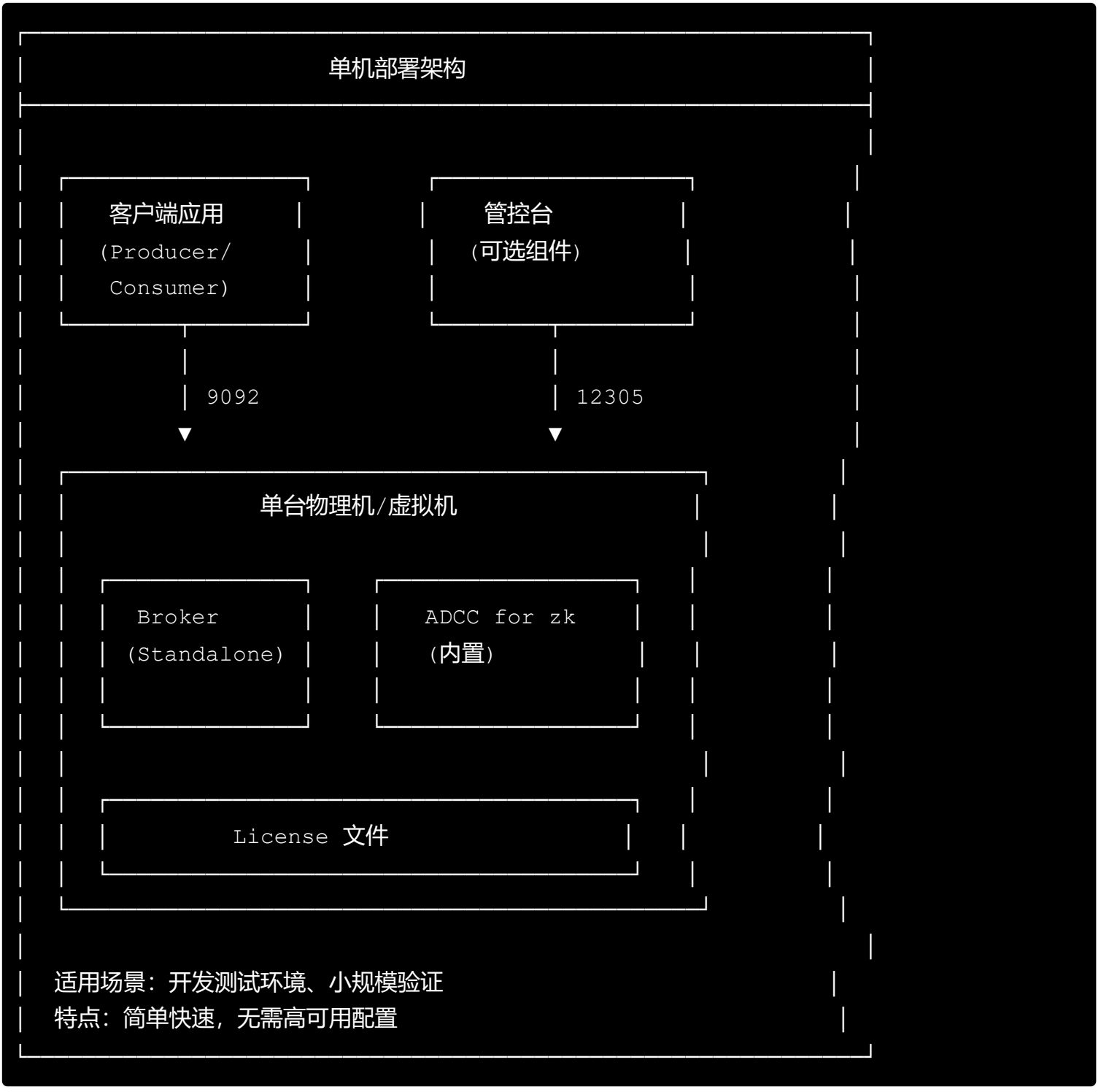
1. 您的姓名
2. 公司信息与联系方式
3. 操作系统及其版本
4. 产品版本号
5. 出现异常及错误的日志、截图等详细信息

2 概述

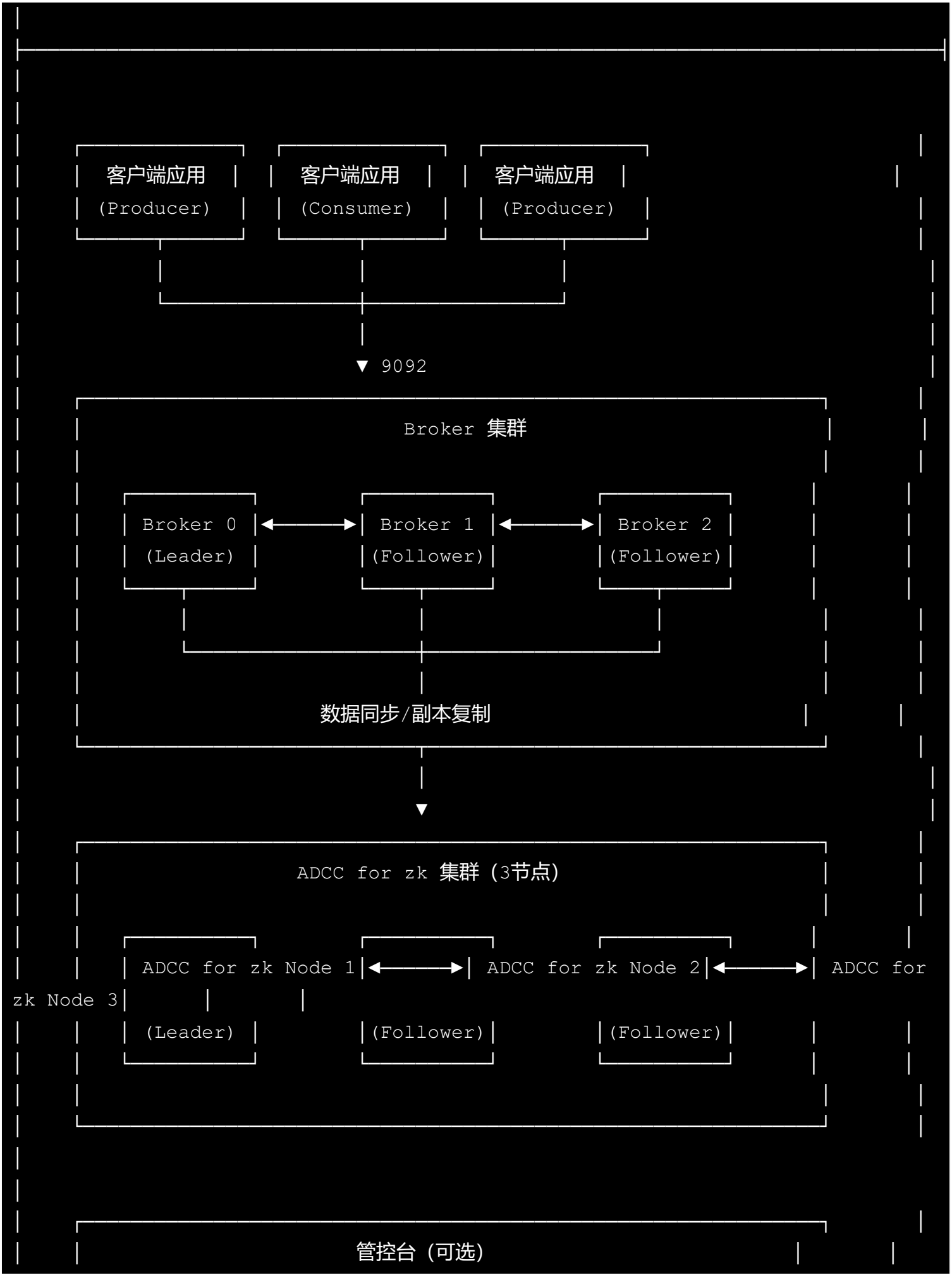
ADMQ for Kafka包含核心引擎、分布式协调、管控台三个部分，其中核心引擎受license控制，需要在安装时申请并导入license。管控台为可选部件，如需要可视化管控时，可单独安装管控台。

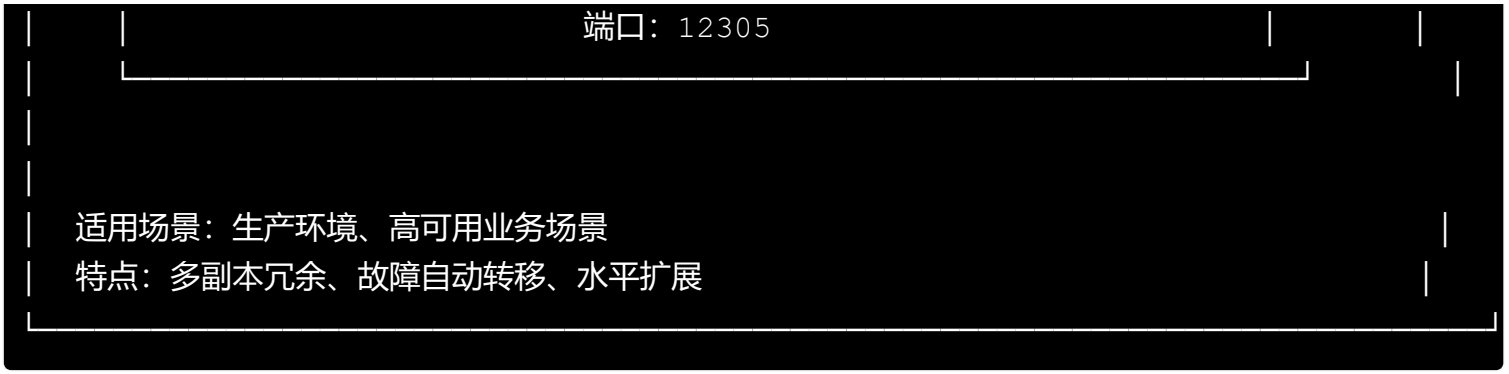
3 部署架构图

3.1 单机部署架构

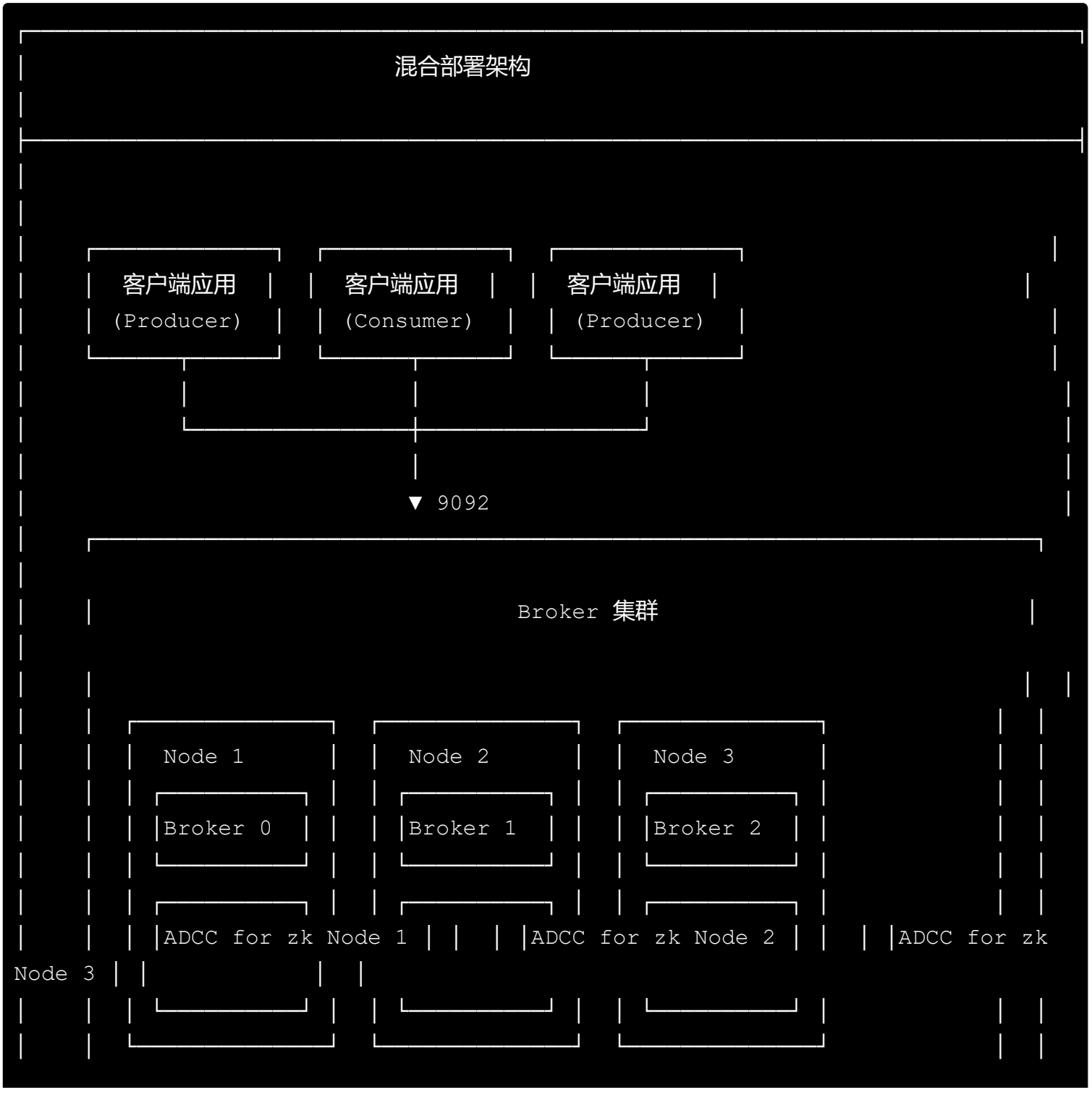


3.2 集群部署架构


客户端应用
(Producer)
客户端应用
(Consumer)
客户端应用
(Producer)
9092
Broker 集群
Broker 0
(Leader)
Broker 1
(Follower)
Broker 2
(Follower)
数据同步/副本复制
ADCC for zk 集群 (3节点)
ADCC for zk Node 1
(Leader)
ADCC for zk Node 2
(Follower)
ADCC for zk Node 3
(Follower)
管控台（可选）


端口：12305
适用场景：生产环境、高可用业务场景
特点：多副本冗余、故障自动转移、水平扩展

3.3 混合部署架构


混合部署架构
客户端应用 (Producer)
客户端应用 (Consumer)
客户端应用 (Producer)
▼ 9092
Broker 集群
Node 1
Broker 0
ADCC for zk Node 1
Node 2
Broker 1
ADCC for zk Node 2
Node 3
Broker 2
ADCC for zk Node 3


管控台（可选，独立部署）
端口：12305
适用场景：资源受限的生产环境、中小型集群
特点：节省服务器资源，ZK与Broker共用节点，部署成本更低
注意：需确保节点资源充足（CPU/内存/磁盘），避免资源竞争

4 安装前准备

4.1 系统要求

组件	最低要求	推荐配置
操作系统	Linux (CentOS 7+, Ubuntu 18.04+, 麒麟 V10, 统信 UOS)	-
CPU	4 核	8 核及以上
内存	8 GB	16 GB 及以上
磁盘空间	100 GB	500 GB 及以上 SSD
JDK	JDK 1.8+ (内置 OpenJDK 11)	JDK 11/17

4.2 端口检查

确保以下端口可用：

端口	服务	说明
9092	Broker	生产者和消费者连接
2181	ADCC for zk	ADCC for zk客户端连接
2888	ADCC for zk	Leader 和 Follower 通信
3888	ADCC for zk	选举通信
12305	管控台	Web 界面（HTTP）

4.3 关闭防火墙（测试环境）

```
# CentOS/RHEL
systemctl stop firewalld
systemctl disable firewalld

# Ubuntu
ufw disable
```

5 单机部署

5.1 1. 创建部署目录

首先创建应用根目录并进入：

```
mkdir -p /apusic
cd /apusic
```

5.2 2. 下载并解压软件包

可从产品光盘或联系销售获取安装包：

```
# 假设安装包已拷贝到服务器
tar zxvf ADMQ-V2.0.6.391-Kafka-20260319.tar.gz
```

5.3 3. 目录结构说明

解压完成后，当前目录下将生成 `admq-kafka` 。标准目录结构如下：

目录名	说明
`bin`	存放启动、停止及管理脚本
`config`	存放配置文件 (如 `kafka-standalone.conf`)
`data`	存放消息数据及ADCC for zk数据
`jdk`	内置的 Java 运行环境，无需额外配置系统 JDK
`logs`	存放运行日志 (Broker 日志、ADCC for zk日志等)
`service`	系统服务注册相关文件 (可选)

5.4 4. 配置修改

编辑单机模式配置文件，绑定指定的 IP 地址和端口：

```
vi config/kafka-standalone.conf
```

修改 `listeners` 配置项：

```
listeners=PLAINTEXT://192.168.1.10:9092
```

5.5 5. 授权许可

ADMQ for Kafka 产品需要 License 文件才能正常运行：

1. **申请授权**：联系管理员或通过授权平台申请对应版本的 `license.xml` 文件
2. **放置文件**：将获取到的 `license.xml` 文件上传至部署目录根路径

```
# 确认文件位置
ls -l /apusic/admq-kafka/license.xml
```

5.6 6. 启动服务

ADMQ for Kafka 单机模式依赖 ADCC for zk，需按顺序启动。

5.6.1 6.1 启动 ADCC for zk

```
bin/admq-daemon start kafka zk
```

5.6.2 6.2 启动 Broker

待 ADCC for zk 启动成功后（通常等待 3-5 秒），启动 Kafka 主服务：

```
bin/admq-daemon start kafka standalone
```

5.7 7. 服务验证

5.7.1 7.1 检查进程状态

```
ps -ef | grep admq-kafka
```

正常状态下应能看到两个主要 Java 进程（一个 ADCC for zk，一个 Broker）。

5.7.2 7.2 检查端口监听

```
netstat -nltup | grep 9092
```

确认 9092 端口处于 LISTEN 状态。

5.7.3 7.3 功能测试

```
# 创建测试 Topic
kafka/bin/kafka-topics.sh --create \
  --bootstrap-server 192.168.1.10:9092 \
  --topic test-topic \
  --partitions 3 \
  --replication-factor 1

# 生产消息
kafka/bin/kafka-console-producer.sh \
  --bootstrap-server 192.168.1.10:9092 \
  --topic test-topic

# 消费消息
kafka/bin/kafka-console-consumer.sh \
  --bootstrap-server 192.168.1.10:9092 \
  --topic test-topic \
  --from-beginning
```

5.8 8. 停止服务

操作	命令
停止 ADCC for zk	`bin/admq-daemon stop kafka zk`
停止 Broker	`bin/admq-daemon stop kafka standalone`

6 集群部署

6.1 1. 部署规划

假设部署 3 节点高可用集群（ADCC for zk与 Broker 混合部署）：

节点	IP 地址	ADCC for zk MyID	Broker ID	ADCC for zk 端口	Broker 端口
Node 1	192.168.1.10	1	0	2181,2888,3888	9092
Node 2	192.168.1.11	2	1	2181,2888,3888	9092
Node 3	192.168.1.12	3	2	2181,2888,3888	9092

前置要求：

1. 确保三台机器时间同步（NTP）
2. 关闭防火墙或开放端口：2181 , 2888, 3888, 9092

6.2 2. 所有节点执行：基础安装

在 Node 1, Node 2, Node 3 上分别执行：

```
mkdir -p /apusic
cd /apusic
tar zxvf ADMQ-V2.0.6.391-Kafka-20260319.tar.gz
cd admq-kafka
```

6.3 3. 放置授权文件

将 `license.xml` 上传至 `/apusic/admq-kafka/` 目录（所有节点）。

6.4 4. 配置 ADCC for zk 集群

6.4.1 4.1 修改 ADCC for zk 配置文件

编辑 `config/kafka-zk.conf`。在所有三个节点上，配置内容完全一致：

```
server.1=192.168.1.10:2888:3888:participant;2181
server.2=192.168.1.11:2888:3888:participant;2181
server.3=192.168.1.12:2888:3888:participant;2181
```

6.4.2 4.2 创建 MyID 文件

Node 1 (192.168.1.10):

```
mkdir -p /apusic/admq-kafka/data/zk/data
mkdir -p /apusic/admq-kafka/data/zk/logs
echo "1" > /apusic/admq-kafka/data/zk/data/myid
```

Node 2 (192.168.1.11):

```
mkdir -p /apusic/admq-kafka/data/zk/data
mkdir -p /apusic/admq-kafka/data/zk/logs
echo "2" > /apusic/admq-kafka/data/zk/data/myid
```

Node 3 (192.168.1.12):

```
mkdir -p /apusic/admq-kafka/data/zk/data
mkdir -p /apusic/admq-kafka/data/zk/logs
echo "3" > /apusic/admq-kafka/data/zk/data/myid
```

6.5 5. 配置 Broker 集群

编辑 `config/kafka-broker.conf` 。

6.5.1 5.1 Node 1 配置 (192.168.1.10)

```
broker.id=0
listeners=PLAINTEXT://192.168.1.10:9092
zookeeper.connect=192.168.1.10:2181,192.168.1.11:2181,192.168.1.12:2181
zookeeper.connection.timeout.ms=18000
```

6.5.2 5.2 Node 2 配置 (192.168.1.11)

```
broker.id=1
listeners=PLAINTEXT://192.168.1.11:9092
zookeeper.connect=192.168.1.10:2181,192.168.1.11:2181,192.168.1.12:2181
```

6.5.3 5.3 Node 3 配置 (192.168.1.12)

```
broker.id=2
listeners=PLAINTEXT://192.168.1.12:9092
zookeeper.connect=192.168.1.10:2181,192.168.1.11:2181,192.168.1.12:2181
```

6.6 6. 启动集群

严格遵循启动顺序：先 ADCC for zk 集群，后 Broker 集群。

6.6.1 6.1 第一步：启动所有节点的 ADCC for zk

依次登录 Node 1, 2, 3 执行：

```
cd /apusic/admq-kafka
bin/admq-daemon start kafka zk
```

6.6.2 6.2 第二步：启动所有节点的 Broker

依次登录 Node 1, 2, 3 执行：

```
cd /apusic/admq-kafka
bin/admq-daemon start kafka broker
```

6.7 7. 验证与测试

6.7.1 7.1 验证 ADCC for zk 集群状态

```
echo ruok | nc 192.168.1.10 2181
echo ruok | nc 192.168.1.11 2181
echo ruok | nc 192.168.1.12 2181
```

预期返回 `imok`。

6.7.2 7.2 验证 Broker 集群元数据

```
kafka/bin/kafka-broker-api-versions.sh --bootstrap-server 192.168.1.10:9092
```

6.7.3 7.3 功能测试

1. 创建 Topic（3 副本）：

```
kafka/bin/kafka-topics.sh --create \
  --topic test-cluster \
  --bootstrap-server 192.168.1.10:9092 \
  --partitions 3 \
  --replication-factor 3
```

2. 查看 Topic 详情:

```
kafka/bin/kafka-topics.sh --describe \
  --topic test-cluster \
  --bootstrap-server 192.168.1.10:9092
```

3. 生产与消费测试:

```
# 生产者
kafka/bin/kafka-console-producer.sh \
  --topic test-cluster \
  --bootstrap-server 192.168.1.10:9092

# 消费者
kafka/bin/kafka-console-consumer.sh \
  --topic test-cluster \
  --from-beginning \
  --bootstrap-server 192.168.1.10:9092
```

6.8 8. 停止服务

```
# 先停 Kafka
bin/admq-daemon stop kafka broker
# 后停 ADCC for zk
bin/admq-daemon stop kafka zk
```

7 服务化部署（systemd）

7.1 注册系统服务

```
# 复制服务文件
cp service/admq-kafka.service /etc/systemd/system/
cp service/admq-zookeeper.service /etc/systemd/system/

# 重新加载
systemctl daemon-reload

# 启动服务
systemctl start admq-zookeeper
systemctl start admq-kafka

# 查看状态
systemctl status admq-kafka
```

7.2 服务管理命令

操作	命令
启动 ADCC for zk	`systemctl start admq-zookeeper`
启动 Broker	`systemctl start admq-kafka`
停止	`systemctl stop admq-kafka`
重启	`systemctl restart admq-kafka`
状态	`systemctl status admq-kafka`
开机自启	`systemctl enable admq-kafka`

8 常见问题

8.1 安装问题

问题	原因	解决方法
端口被占用	9092 或 2181 端口已被占用	检查端口并释放或修改配置端口
内存不足	服务器内存低于最低要求	增加服务器内存
磁盘空间不足	磁盘空间不足	清理磁盘或增加存储
License 无效	License 文件不正确或过期	申请有效 License 并重新放置

8.2 启动问题

问题	原因	解决方法
ADCC for zk 启动失败	myid 文件配置错误	检查 myid 文件内容
Broker 无法连接 ADCC for zk	ADCC for zk 地址配置错误	检查 zookeeper.connect 配置
Broker ID 冲突	broker.id 重复	检查并修改 broker.id

9 回滚方案

9.1 单机回滚

如需回滚到安装前状态：

```
# 1. 停止服务
bin/admq-daemon stop kafka standalone
bin/admq-daemon stop kafka zk

# 2. 备份并删除数据目录（谨慎操作）
mv /apusic/admq-kafka /apusic/admq-kafka.bak.$(date +%Y%m%d)

# 3. 如需恢复，重新解压安装包并配置
```

9.2 集群回滚

```
# 1. 按顺序停止所有节点服务
bin/admq-daemon stop kafka broker
bin/admq-daemon stop kafka zk

# 2. 备份各节点数据
mv /apusic/admq-kafka /apusic/admq-kafka.bak.$(date +%Y%m%d)

# 3. 清理 ADCC for zk 数据
cd /apusic/admq-kafka.bak.*/data/zk
cp -r data /backup/zk-data-$(date +%Y%m%d)

# 4. 重新部署并恢复配置
```

9.3 配置回滚

```
# 备份当前配置
cp config/kafka-broker.conf config/kafka-broker.conf.bak
cp config/kafka-zk.conf config/kafka-zk.conf.bak

# 恢复配置
```

```
cp config/kafka-broker.conf.bak config/kafka-broker.conf
cp config/kafka-zk.conf.bak config/kafka-zk.conf

# 重启服务生效
bin/admq-daemon restart kafka broker
```

10 安全加固

10.1 1. 访问控制

10.1.1 1.1 防火墙配置

```
# 仅开放必要端口
iptables -A INPUT -p tcp --dport 9092 -s 192.168.1.0/24 -j ACCEPT
iptables -A INPUT -p tcp --dport 2181 -s 192.168.1.0/24 -j ACCEPT
iptables -A INPUT -p tcp --dport 12305 -s 192.168.1.0/24 -j ACCEPT
iptables -A INPUT -p tcp --dport 9092 -j DROP
iptables -A INPUT -p tcp --dport 2181 -j DROP
```

10.1.2 1.2 禁用不必要的协议

```
# 仅启用安全协议
listeners=SASL_SSL://0.0.0.0:9092
security.inter.broker.protocol=SASL_SSL
```

10.2 2. 认证配置

10.2.1 2.1 SASL/SCRAM 认证

```
# 启用 SCRAM 认证
sasl.enabled.mechanisms=SCRAM-SHA-256,SCRAM-SHA-512
sasl.mechanism.inter.broker.protocol=SCRAM-SHA-512
```

10.2.2 2.2 创建安全用户

```
# 创建管理员用户
kafka-configs.sh --zookeeper localhost:2181 \
  --alter --add-config 'SCRAM-SHA-256=[password=admin-password]' \
  --entity-type users --entity-name admin

# 创建应用用户
kafka-configs.sh --zookeeper localhost:2181 \
```

```
  --alter --add-config 'SCRAM-SHA-256=[password=app-password]' \
  --entity-type users --entity-name app-user
```

10.3 3. 授权配置

10.3.1 3.1 配置 ACL

```
# 授予生产者权限
kafka-acls.sh --authorizer-properties zookeeper.connect=localhost:2181 \
  --add --allow-principal User:producer \
  --operation Write --topic my-topic

# 授予消费者权限
kafka-acls.sh --authorizer-properties zookeeper.connect=localhost:2181 \
  --add --allow-principal User:consumer \
  --operation Read --topic my-topic \
  --group my-consumer-group

# 拒绝非法访问
kafka-acls.sh --authorizer-properties zookeeper.connect=localhost:2181 \
  --add --deny-principal User:Anonymous \
  --operation All --topic '*'
```

10.4 4. SSL/TLS 加密

10.4.1 4.1 生成证书

```
# 生成 CA 证书
keytool -keystore kafka.server.keystore.jks \
  -alias ca-cert -import -file ca-cert.crt

# 生成 Broker 证书
keytool -keystore kafka.server.keystore.jks \
  -alias broker-cert -validity 365 -genkey -keyalg RSA

# 导入信任证书
keytool -keystore kafka.server.truststore.jks \
  -alias ca-cert -import -file ca-cert.crt
```

10.4.2 4.2 配置 SSL

```
# SSL 配置
listeners=SSL://0.0.0.0:9093
ssl.keystore.location=/path/to/kafka.server.keystore.jks
ssl.keystore.password=keystore-password
ssl.key.password=key-password
ssl.truststore.location=/path/to/kafka.server.truststore.jks
ssl.truststore.password=truststore-password
ssl.client.auth=required
ssl.enabled.protocols=TLSv1.2,TLSv1.3
```

10.5 5. 审计日志

```
# 启用审计日志
log4j.logger.kafka.authorizer.logger=WARN, authorizerAppender
log4j.appender.authorizerAppender=org.apache.log4j.DailyRollingFileAppender
log4j.appender.authorizerAppender.File=${log.dir}/kafka-authorizer.log
log4j.appender.authorizerAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.authorizerAppender.layout.ConversionPattern=[%d] %p %m (%c)%n
```

10.6 6. 操作系统安全

10.6.1 6.1 文件权限

```
# 设置目录权限
chmod 750 /apusic/admq-kafka
chmod 600 /apusic/admq-kafka/config/*.conf
chmod 600 /apusic/admq-kafka/license.xml

# 限制日志访问
chmod 640 /apusic/admq-kafka/logs/*.log
```

10.6.2 6.2 禁用 root 运行

```
# 创建专用用户
useradd -r -s /bin/false admq-kafka
```

```
chown -R admq-kafka:admq-kafka /apusic/admq-kafka

# 使用 systemd 限制权限
# 在 admq-kafka.service 中添加:
User=admq-kafka
Group=admq-kafka
NoNewPrivileges=true
ProtectSystem=strict
ProtectHome=true
```

11 技术支持

如遇到无法解决的问题，请联系技术支持：

- 电话：400-855-5800
- 邮箱：support@apusic.com

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