



APUSIC
固若长城
睿比世界

安装手册

金蝶Apusic分布式缓存 V2.0.0

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 安装部署
 - 1.1 系统环境要求
 - 1.2 安装
 - 1.2.1 管控台安装
 - 1.2.2 核心引擎安装
 - 1.2.3 国密代理客户端安装
 - 1.2.4 其他方式安装核心引擎
 - 1.3 核心引擎配置项说明

1 安装部署

1.1 系统环境要求

无特殊要求，选择系统相应的安装包即可。

1.2 安装

1.2.1 管控台安装

1. 将系统对应的管控台安装包放置到自定义的安装目录中。
2. 解压并进入 `amdc-console-pak`
3. 将 许可证 `license` 放入根目录中。
4. 执行 `./amdc-console`。

1.2.2 核心引擎安装

1. 将软件包放置到自定义的安装目录中。
2. 解压并进入 `amdc` 目录。
3. 将 许可证 `license` 放入根目录中。
4. 执行 `./amdc-server -conf conf.yaml`。

1.2.3 国密代理客户端安装

1. 将软件包放置到自定义的安装目录中。
2. 解压并进入 `amdc_proxy` 目录。
3. 根据配置项配置需要 被代理的客户端 的ip以及端口和 AMDC的国密接收ip以及端口。
4. 执行 `./amdc-proxy -conf conf.yaml`。

1.2.4 其他方式安装核心引擎

通过管控台进行自动部署安装：

1. 登录管控台。
2. 进入机器管理页面并添加目标服务器信息。
3. 进入自动部署页面，填写相应的部署信息，点击安装等待完成即可。

可参考用户手册了解更多手动部署模式。

1.3 核心引擎配置项说明

Network:

绑定的ip地址, 可以绑定多个ip地址, 支持ipv4/ipv6地址, eg:

Bind:

- "127.0.0.1"

- "::1"

Bind:

- "127.0.0.1"

端口号, 如果只启用tls, 则当前端口设置为0, 则只监听SSL的端口

Port: 6359

最大连接数, 超过后服务端拒绝接收新的连接, 当设置为0时, 不启限制最大连接数

MaxClients: 0

连接在空闲超过设置的时间后, 服务器会主动关闭连接, 当设置为0时, 不启用超时机制, timeout单位为秒

Timeout: 0

TCP keepalive, 单位为秒, 当为0时不设置tcp keepalive

TcpKeepAlive: 300

IO协程数量, 主协程也是IO协程, 默认配置1表示不启动多IO协程处理, 大于1表示启动多IO协程处理

IOGroutineNum: 1

IO协程是否进行客户端读请求和解析请求, yes/no

IOGroutineDoReads: "yes"

General:

db数量

Databases: 16

日志等级, 用于过滤输出日志, 包括debug, info, warn, error, fatal五个等级

LogLevel: "info"

日志文件输出目录, 当设置为空字符串时, 日志文件不会写入磁盘, eg: LogFile: "/tmp/server.log"

LogFile: ""

license文件位置

LicensePath: "./license.xml"

MemoryManagement:

最大内存限制, 如果maxmemory值为0, 表示不做限制, 如果数字后没有单位, 则默认

单位为字节，单位不区分大小写

```
# eg: "1gb", "1GB", "1000mb", "1000m", "1000000KB", "1000000kb",
"1000000000B", "1000000000b", "1000000000"
```

```
Maxmemory: "0"
```

```
# 缓存淘汰策略, noeviction, volatile-lru, allkeys-lru, volatile-
random, allkeys-random, volatile-ttl
```

```
MaxmemoryPolicy: "noeviction"
```

```
# 每次缓存淘汰时的采样数量
```

```
MaxmemorySamples: 5
```

```
# lfu-log-factor可以调整计数器counter的增长概率, lfu-log-factor越大,
counter增长概率越小
```

```
# 计算公式为: 1 / (old_value * lfu_log_factor + 1)
```

```
LFULogFactor: 10.000000
```

```
# lfu-decay-time是一个以分钟为单位的数值, 可以调整counter的减少速度
```

```
LFUDecayTime: 1.000000
```

```
# 从节点是否忽略maxmemory检查
```

```
ReplSlaveIgnoreMaxmemory: "yes"
```

```
#pool paramter,pool slice num, 0: not use pool
```

```
PoolSliceNum: 400
```

```
#pool paramter,pool slice num, 0: not use pool
```

```
PoolSliceSize: 256
```

```
#pool paramter,new memory goroutine num
```

```
PoolNewWorkerNum: 4
```

Snapshotting:

```
# save <seconds> <changes>, 指定在多长时间, 有多少次更新操作后, server
会将rdb文件写入重新写入磁盘, eg:
```

```
# Save:
```

```
# - "900 1"
```

```
# - "300 10"
```

```
# - "60 10000"
```

```
# 当 Save: ""为空时, 不启用RDB自动保存
```

```
Save: ""
```

```
# bgsave保存失败后, 服务器是否停止接受写入
```

```
StopWritesOnBgsaveError: "yes"
```

```
# 是否开启对string对象开启lzf压缩, yes/no
```

```
RdbCompression: "yes"  
# 是否开启CRC64校验, yes/no  
RdbChecksum: "yes"  
# rdb文件名, 不包括路径  
DbFileName: "dump.rdb"  
# aof文件名称, 不包含路径  
AppendFileName: "appendonly.aof"  
# 工作目录, rdb和aof文件会存储在Dir路径下  
Dir: "./"
```

Security:

```
# auth密码, 在users.acl存在的情况下, server优先使用users.acl中的密码  
RequirePass: ""  
# ACL权限控制文件保存位置  
ACLFile: "./users.acl"  
# ACL channel的默认权限, allchannels / resetchannels  
ACLPubsubDefault: "allchannels"  
# ACL log日志保存的最大数量  
ACLLogMaxLen: 128
```

SlowLog:

```
# 指定执行时间超过多少微秒的命令被记录到日志上  
SlowLogSlowerThan: 10000  
# slowlog日志最大长度  
SlowLogMaxLen: 128
```

Script:

```
# lua脚本的最大执行时间, 单位为毫秒, 设置为0时不对最大执行时间作限制  
LuaTimeLimit: 5000  
# lua脚本的最大参数数量  
LuaMaxLocalVarNum: 600
```

LazyFree:

```
# 在缓存淘汰键时, 是否采用lazy free机制, yes / no  
LazyEviction: "no"  
# 针对设置有TTL的键, 达到过期后, 被清理删除时是否采用lazy free机制, yes /
```

```

no
  LazyExpire: "no"
  # 针对有些指令在处理已存在的键时，会带有一个隐式的DEL键的操作。如rename命令，
  是否采用lazy free机制， yes / no
  LazyServerDel: "no"
  # 针对slave进行全量数据同步，slave在加载master的RDB文件前，会执行flushall
  来清理自己的数据，参数设置决定是否采用异步flush机制， yes / no
  ReplicaLazyFlush: "no"
  # 修改del相关命令的默认行为，是否使用unlink异步删除， yes / no
  LazyUserDel: "no"

Replication:
  # 设置启动服务器为指定服务器的从节点，eg: Replicaof: "127.0.0.1 6378"
  Replicaof: ""
  # 用于主从节点认证的密码
  MasterAuth: ""
  # 主从节点连接超时时间，单位为秒
  ReplTimeout: 60
  # 在主从节点断开或同步阶段，是否允许从节点继续处理读请求
  ReplServeStaleData: "yes"
  # 如果正常从节点数小于该配置，则主节点拒绝执行命令
  MinReplicasToWrite: 0
  # 如果从节点超过该配置时间没有返回ACK信息，则判定从节点不正常，单位为秒
  MinReplicasMaxLag: 10
  # 从节点是否只处理读请求，不能修改数据
  ReplicaReadOnly: "yes"
  # 若启用了端口转发或者NAT, ReplicaAnnounceIp设置的值会覆盖从节点的默认IP值
  ReplicaAnnounceIp: ""
  # 若启用了端口转发或者NAT, ReplicaAnnouncePort设置的值会覆盖从节点的默认
  Port值
  ReplicaAnnouncePort: 0
  # 是否在SYNC后关闭 TCP_NODELAY
  ReplDisableTcpNoDelay: "no"
  # 主节点向从节点发送PING命令的时间间隔，单位为秒
  ReplPingSlavePeriod: 10
  # 复制积压缓冲区大小

```

```
ReplBacklogSize: "1mb"
```

```
# 主节点处于无从节点状态的时间, 超过该配置将释放复制积压区, 单位为秒
```

```
ReplBacklogTTL: 3600
```

```
# 当主节点无法正常工作后Sentinel通过这个值来决定将哪个从节点提升为主节点
```

```
# 值越小越优先进行提升, 这个值为0表示从节点永远不能被提升为主节点
```

```
ReplicaPriority: 100
```

EventNotification:

```
# 键通知支持类型
```

```
# K      Keyspace events, published with __keyspace@<db>__ prefix.
```

```
# E      Keyevent events, published with __keyevent@<db>__ prefix.
```

```
# g      Generic commands (non-type specific) like DEL, EXPIRE,
```

```
RENAME, ...
```

```
# $      String commands
```

```
# l      List commands
```

```
# s      Set commands
```

```
# h      Hash commands
```

```
# z      Sorted set commands
```

```
# x      Expired events (events generated every time a key expires)
```

```
# e      Evicted events (events generated when a key is evicted for  
maxmemory)
```

```
# A      Alias for g$lshzxe, so that the "AKE" string means all the  
events.
```

```
NotifyKeyspaceEvents: ""
```

Cluster:

```
# 是否开启集群模式, yes / no
```

```
ClusterEnabled: "no"
```

```
# 每个集群节点配置文件名称, 不可重名, 节点自动生成与更新, 不可手动编辑
```

```
ClusterConfigFile: "./node.conf"
```

```
# 集群节点超时时间, 单位ms
```

```
ClusterNodeTimeout: 15000
```

```
# 与主节点断开
```

```
(ReplPingSlavePeriod+ClusterReplicaValidityFactor*ClusterNodeTimeout)  
秒的从节点不参与故障转移
```

```
ClusterReplicaValidityFactor: 10
```

```

# 只有当一个主节点至少拥有ClusterMigrationBarrier个处于正常工作中的从节点的时候，才会分配从节点给集群中孤立的主节点
ClusterMigrationBarrier: 1
# 集群中16384个slots是否需要完全分配
# yes : 如果slots没有被完全分配，那么整个集群将处于不可用状态，直到所有slots被分配，集群会自动变得可用
# no : 当slots没有被完全分配时，集群的部分节点仍然是可用的
ClusterRequireFullCoverage: "yes"
# 此集群节点是否参与自动故障转移
# yes: 此集群从节点不会参与自动故障转移过程，但是可以手动强制执行故障转移
# no : 此集群从节点参与自动故障转移过程
ClusterReplicaNoFailover: "no"
# 为了使集群在启用了端口转发或者NAT的环境中工作，需要静态配置ClusterAnnounceIp，使集群中每个节点都知道其公开的ip地址
ClusterAnnounceIp: ""
# 为了使集群在启用了端口转发或者NAT的环境中工作，需要静态配置ClusterAnnouncePort，使集群中每个节点都知道其公开的端口号
ClusterAnnouncePort: 0
# 为了使集群在启用了端口转发或者NAT的环境中工作，需要静态配置ClusterAnnounceBusPort，使集群中每个节点都知道其公开的集群消息广播端口
ClusterAnnounceBusPort: 0

Advanced:
# 请求输入缓冲区最大值，当超过缓冲区最大值时，服务器会直接关闭连接
ClientQueryBufferLimit: "1gb"
# 对于RESP协议的Bulk多行请求，每行字符串的最大长度
ProtoMaxBulkLen: "512mb"

Proxy:
#是否启动代理
Enabled: "no"
# proxy ip地址
Bind: "127.0.0.1"
# 端口号
Port: 8002
# 绑定的ip地址

```

```
Proxy2IP: "127.0.0.1"
# 端口号
Proxy2Port: 6359
#是否加密
CriptEnabled: "yes"
#是否国密
GMflag: 1
#加解密认证文件所在路径
CertPath: "./certs/gm_cert"

Prometheus:
# 是否开启Prometheus监控指标数据
Enabled: false # 请选择 true 或 false
# prometheus HTTP服务的绑定的地址, 默认127.0.0.1本机访问,
# 如果需要暴露给局域网或指定IP, 请填写当前环境的局域网ip段
# 所有流量均可访问可以设置为: 0.0.0.0
Bind: "127.0.0.1"
# prometheus HTTP metrics的访问端口
Port: 8004
# prometheus metrics的每个指标的前缀, 如默认为amdc:
amdc_command_total, 设置为redis: redis_command_total
NameSpace: "amdc"
# prometheus metrics指标的http访问URL地址, 如结合上面的bind + port为:
http://127.0.0.0:8004/metrics
MetricsPath: "/metrics"
# 客户端与amdc连接的超时时间。
ConnectionTimeOut: "15s"
# 是否展示amdc连接的客户端信息
export-client-list: true
# 使用HTTPS访问, 默认false使用http, 需要使用HTTPS则需要改为true
EnableHTTPS: false
# HTTPS访问能力需要提供SSL证书, 确保文件夹里面的证书的命名为: server.pem和
server.key
CertPath: "./certs/tls_cert"

IpTable:
```

```
# 将某个Ip地址添加到白名单
```

```
# Ip:
```

```
# - "127.0.0.1"
```

```
# - "192.168.116.1"
```

```
Ip: ""
```

```
# 将某一网段添加到白名单
```

```
# Segment:
```

```
# - "127.0.0.1/24"
```

```
Segment: ""
```

```
SSL:
```

```
# 是否开启ssl, 如果开启则使用true, 否则是false
```

```
Enable: false
```

```
# SSL监听端口, 如果仅开启SSL监听, 需要把NetWord: Port端口设置为0
```

```
Port: 6369
```

```
# 服务端SSL证书文件
```

```
TlsCertFile: "./certs/ssl_tls_cert/server.crt"
```

```
# 服务端SSL证书的密钥
```

```
TlsKeyFile: "./certs/ssl_tls_cert/server.key"
```

```
# 证书签发机构的证书文件, 即可信从根证书
```

```
TlsCaCertFile: "./certs/ssl_tls_cert/ca.crt"
```

```
# 证书签发机构的证书文件夹, 即多个可信的根证书, 可放在一个文件夹中
```

```
TlsCaCertDir: ""
```

```
# 客户端SSL证书文件, 为集群/主从模式使用
```

```
TlsClientCertFile: "./certs/ssl_tls_cert/client.crt"
```

```
# 客户端SSL证书的密钥, 为集群/主从模式使用
```

```
TlsClientKeyFile: "./certs/ssl_tls_cert/client.key"
```

```
# 服务器是否验证客户端证书, 默认为空需要客户端提供证书, 并且服务端作验证。 如果不需要客户端可以配置为"no",
```

```
# 设置为optional, 即客户端可以提供证书也可以不提供证书, 但是如果客户端提供证书, 那么amdc服务器将验证证书的合法性
```

```
TlsAuthClients: ""
```

```
# 主从模式是否使用TLS通信, 当master开启了tls模式, 那么从节点必须开启, 否则无法连接master节点
```

```
TlsReplication: false
```

集群之间是否采用tls进行通讯, 设置为true表示集群使用tls, 否则不使用。

```
TlsCluster: false
```

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

