Apusic 金蝶天燕

生成时间：2026-05-07

Installation Manual

Kingdee Apusic In-Memory Data Cache v2.0.1

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

1 System Environment Requirements

There are no special requirements; choose the installation package corresponding to your system.

2 Recommended Configuration

Deployment Mode	Operating System	Installation Content	Hardware Specifications (CPU/Memory/Disk)	Number of Servers
Standalone	Linux	AMDC Console, AMDC Service	8 cores/16GB/100GB	1
Master-Slave	Linux	AMDC Console, AMDC Service	8 cores/16GB/100GB	2
Sentinel	Linux	AMDC Console, AMDC Service	8 cores/16GB/100GB	3
Cluster	Linux	AMDC Console, AMDC Service	8 cores/16GB/100GB	3

3 Supported Environments

Platform Type	System Type
Chip Type	Huawei Kunpeng, Hisilicon, Phytium, Zhaoxin and other X86/ARM architecture chips
Operating Systems	NeoKylin series, UOS by UnionTech, NeoKylin and others
Other Linux Series	RedHat series, CentOS series, Ubuntu series and others

4 Mode Selection

- Standalone：Use in scenarios where dependence on cache is low and only serves to enhance system response speed.
- Master-Slave：Use in scenarios where dependence on cache is low, enhances system response speed, but data is important or requires quick replacement.
- Sentinel：Use in scenarios with high dependence on cache and needs rapid failover.
- Cluster：Use in scenarios with high dependence on cache, needs rapid failover, and provides elastic scaling capabilities (expansion or reduction of capacity).

5 Installation

5.1 AMDC Deployment and Start/Stop

AMDC is divided into two parts: "Cache Service" and "Console". The AMDC cache service primarily covers Redis features and supports national encryption standards, data persistence, etc. The AMDC management console supports cache monitoring, automatic deployment, cluster/node management, automatic alerts, real-time configuration, data operations, and permission control functions.

5.1.1 Preparations Before Deployment

The AMDC cache service and console can be deployed separately or through the console. This section will introduce the installation and deployment of the AMDC cache service and AMDC console respectively.

5.1.2 Obtaining the Installation Package

Download the AMDC software installation package from the Kingdee Apusic official website, or obtain the corresponding installation package files from the AMDC product CD-ROM. Note the distinction between the AMDC cache service and AMDC console installation packages.

5.1.3 Regarding License

1. Use an officially authorized valid license file within its validity period.
2. Use the official Kingdee Apusic authorization center.
3. Use the KBC unified authorization center.

Obtaining the License File

The license file is generated using a feature code. Different machines have different feature codes, so they are not universal. When starting via terminal command, the feature code information will be printed in the terminal, starting with SZTY as shown in the figure below, the content highlighted in red is the feature code:

5.1.4 Deploying the AMDC Console

The main steps for installation are:

1. Upload the AMDC console installation package(amdc_console_release_[version]_[chip_type].tar.gz, hereafter omitting_[chip_type]) to the installation directory of the target server(e.g., /opt directory).
2. Unpack the installation package: `tar -zxf amdc_console_release_v2.tar.gz`

3. Enter the unpacked folder: `cd amdc-console`
4. Execute the start command: `nohup ./amdc-console >nohup.out 2>&1 &` (backend startup)
5. After starting, access the AMDC console through a browser: use Google Chrome or Firefox to visit `http://ip:port` (where ip represents the IP address of the server where the AMDC console is deployed， portis the console port). e.g. `192.168.0.129:9001`

After starting, you can access the console through a browser.

5.1.5 Stopping the AMDC Console

To stop the AMDC console service, forcibly terminate the AMDC console service process.

Steps:

1. Find the console process PID via the port process command: `netstat -nltp | grep 9001`
2. Execute the kill -9 command to forcefully kill the process: `kill -9 Process ID`

5.1.6 Deploying the AMDC Cache Service

The AMDC caching service can be deployed with a single click through the AMDC Console's graphical user interface, or it can be deployed via the command line. Below, we will describe how to deploy the AMDC caching service through the command line method, as well as how to deploy it through the AMDC Console.

The main steps for deploying the AMDC cache service via the command line are:

5.1.6.1 Standalone Mode

Steps:

1. Upload the AMDC cache service installation package(amdc_amd64.tar.gz)to the installation directory of the target server (e.g., /opt directory)
2. Unpack the installation package: `tar -zxf amdc-core-[version]-linux-[arch]-[date].tar.gz`
3. Enter the unpacked folder: `cd amdc`
4. Execute the start command to launch the AMDC cache service: : `./amdc-server` (foreground start) / `./amdc-server --daemonize yes` (background start)

Relevant configuration items:

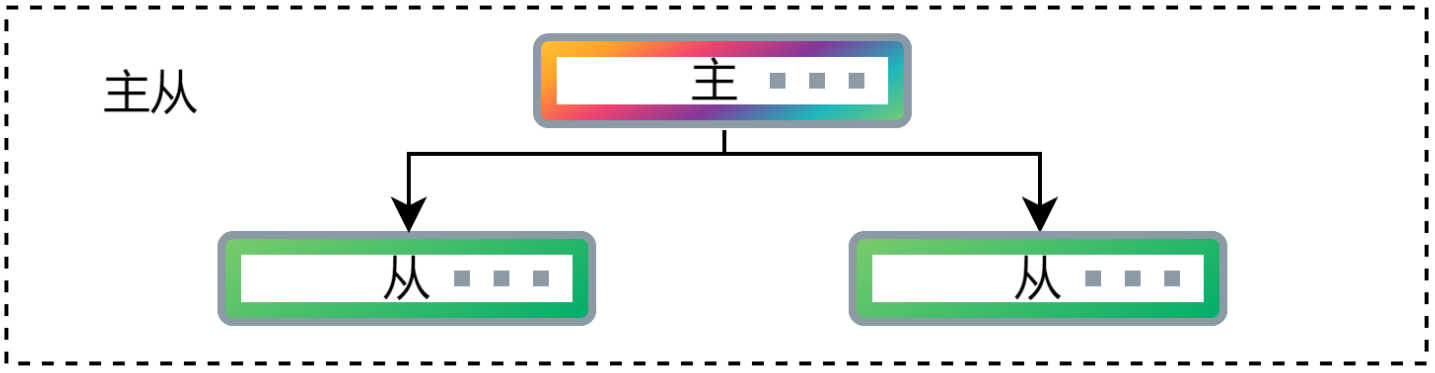
Parameter Name	Explanation	Usage
Bind	Specifies the IP address that the client uses to access the cache. Apart from the bound address, connections cannot be made. For example: 127.0.0.1, accessible only by the local	Listening IP address, when the default value is 0.0.0.0, no other address can be bound; additionally multiple addresses can be bound, it is recommended to add the

	machine; 0.0.0.0 indicates all IP addresses of the local machine can access it	local access IP and remote access IP, such as: Bind: - "127.0.0.1" - "192.168.0.190"
Port	Port number of the cache service	Port: 6359
RequirePass	Set password	Authentication password, if the users.acl file exists, the server prioritizes the password in users.acl

5.1.6.2 Master-Slave Mode

1. Upload the AMDC cache service installation package(amdc_amd64.tar.gz)to the installation directory of the target server (e.g., /opt directory).At least two nodes are required to form one master and one slave, more slave nodes can also be deployed to form one master and n slaves.
2. Unpack the installation package: `tar -zxf amdc-core-[version]-linux-[arch]-[date].tar.gz`
3. Enter the unpacked folder: `cd amdc`
4. If this node is the master node, proceed to step (5); if this node is a slave node, configure the "ip port" of the master node in the Replicaof configuration item under the Replication section of the conf.yaml file.
5. Execute the start command to launch the AMDC cache service: `./amdc-server` (foreground start) / `./amdc-server --daemonize yes` (background start)

Master-Slave Deployment Diagram



Relevant configuration items:

Parameter Name	Explanation	Usage

Bind	Specifies the IP address that the client uses to access the cache. Apart from the bound address, connections cannot be made. For example: 127.0.0.1, accessible only by the local machine; 0.0.0.0 indicates all IP addresses of the local machine can access it	Listening IP address, when the default value is 0.0.0.0, no other address can be bound; additionally multiple addresses can be bound, it is recommended to add the local access IP and remote access IP, such as: Bind: - "127.0.0.1" - "192.168.0.190"
Port	Port number of the cache service	Port: 6359
RequirePass	Set password, can be left empty. It is recommended that the password of the slave node matches the password of the master node to ensure normal connection and usage after a master-slave switch.	Authentication password, if the users.acl file exists, the server prioritizes the password in users.acl, eg: RequirePass: "123456"
Replicaof	Specifies the IP and port of the master node	Set the server to start as a slave node of the specified server, eg: Replicaof: "127.0.0.1 6378"
MasterAuth	Password of the master node, not required if the master node has no password	MasterAuth: "123456"

5.1.6.3 Sentinel Mode

1. Set up an operational master-slave cluster model and place the sentinel files amdc-sentinel and sentinel.yaml under the target server directory;
2. Configure the sentinel.yaml file for the sentinel. Bind specifies the IP for accessing the sentinel, and port is the port for the sentinel application. Among the SentinelItems, the sentinel monitor entry must be designated for the master node.

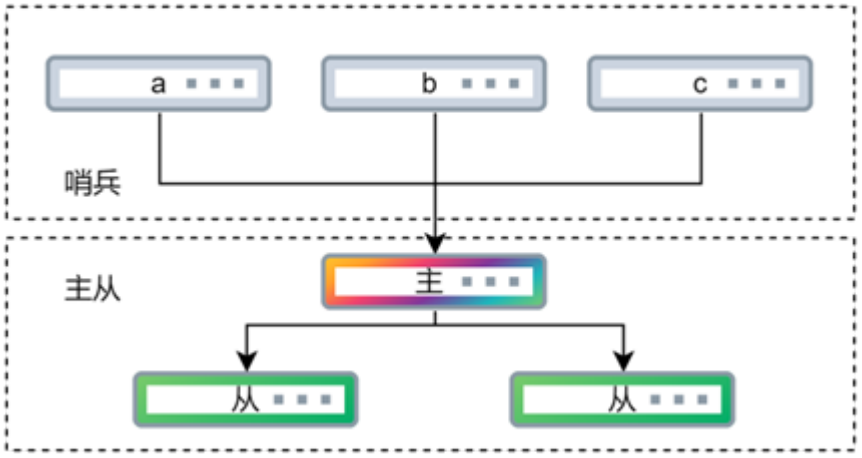
- sentinel monitor mymaster: Master service IP, Master service port, Quorum (The quorum value should be less than the total number of nodes but more than half the number of nodes).

```
Sentinel:
  # 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
  Port: 26359
  # 日志等级，用于过滤输出日志，包括debug, info, warn, error, fatal五个等级
  LogLevel: "debug"
  # 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
  LogFile: ""
  # license文件位置
  LicensePath: "./license.xml"
  # sentinel相关配置项
  SentinelItems:
    - "sentinel monitor mymaster 127.0.0.1 6379 1"       # 设置监听的主节点信息，IP地址/端口/判断为主观下线的哨兵数
    - "sentinel down-after-milliseconds mymaster 30000"  # 设置主节点主观下线的超时时间（单位毫秒）
    - "sentinel failover-timeout mymaster 180000"
    - "sentinel parallel-syncs mymaster 1"
    # - "sentinel auth-pass mymaster 123456"              # 设置哨兵访问主节点的密码，如果主节点有设置密码，请求123456为目标密码并取消注释
    # - "sentinel config-epoch mymaster 0"                # 设置节点的选举纪元
    # - "sentinel leader-epoch mymaster 0"
    # - "sentinel known-replica mymaster 127.0.0.1 6380"  # 设置其他已知的从节点信息
    # - "sentinel current-epoch 0"                        # 设置当前哨兵选举纪元
```

添加机器对外的ip

3. Execute the startup command to initiate the AMDC Sentinel: `nohup ./amdc-sentinel >sentinel.log 2>&1 &`

Sentinel Deployment Diagram



Relevant Configuration Items:

Parameter Name	Explanation	Usage
Bind	Specifies the IP address for client access to the sentinel. Connections cannot be made from outside the bound address. Default is 127.0.0.1, accessible only by the local machine.	Listening IP address, multiple addresses can be bound. It is recommended to add both local access IP and remote access IP, for example:

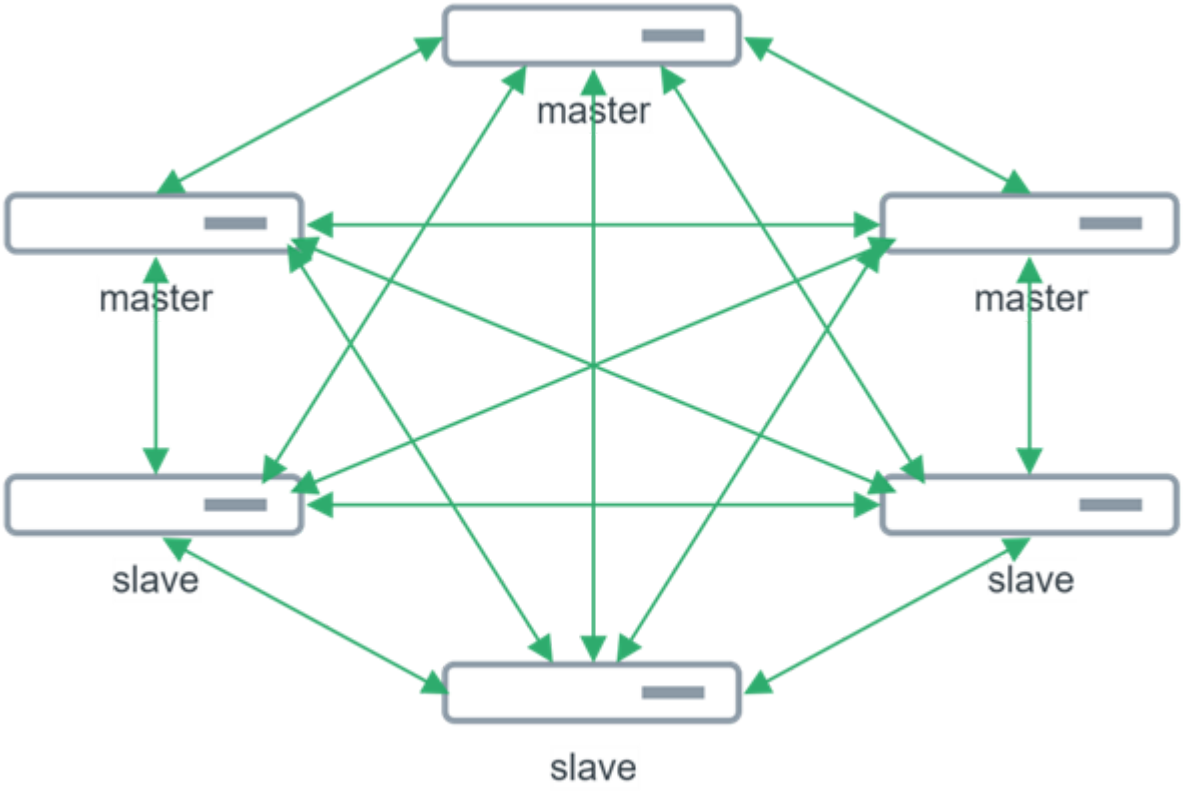
		Bind: - "127.0.0.1" - "192.168.0.190"
Port	The port number for the caching service.	Port: 6359
SentinelItems	Detailed configuration for the sentinel, reference can be made to Redis sentinel configurations.	Other configurations can be modified according to needs, however, monitor must be set to the IP and port of the master node. - "sentinel monitor mymaster 127.0.0.1 6379 1" If the master node has a password, include the password validation parameter. - "sentinel auth-pass mymaster 123456"

5.1.6.4 Cluster Mode

Steps:

1. Upload the AMDC cache service installation package(amdc_amd64.tar.gz)to the installation directory on the target server (e.g., /opt), requiring at least 3 master nodes to form a cluster, typically using 3 masters and 3 slaves.
2. Extract the installation package: `tar -zxf amdc-core-[version]-linux-[arch]-[date].tar.gz`
3. Enter the unpacked folder: `cd amdc`
4. Configure parameters such as Bind, ClusterEnabled, etc., for all nodes, refer to the relevant configurations below.
5. Execute the startup command to start the AMDC cache service: `./amdc-server` (foreground startup) / `./amdc-server --daemonize yes` (background startup)
6. On any one of the nodes, send a command using amdc-cli: `./amdc-cli --cluster create [ip:port ...(multiple node information separated by spaces)] --cluster-replicas [number(number of slave nodes per master node)]`

Cluster Deployment Diagram



Relevant Configuration Items:

Parameter Name	Explanation	Usage
Bind	Specifies the IP address for clients to access the cache service, unable to connect from outside the bound address. Default is 127.0.0.1, accessible only by the local machine.	Listening IP address, multiple addresses can be bound. It is recommended to add both local access IP and remote access IP, for example: Bind: - "127.0.0.1" - "192.168.0.190"
Port	The port number for the cache service.	Port: 6359
RequirePass	Set a password, the password must be consistent across all nodes, can be left blank.	Authentication password, in the presence of users.acl, the server prioritizes the password in users.acl, e.g.: RequirePass: "123456"
MasterAuth	Password for the master node, if the master node does not have a password, it doesn't need to be filled in. If there is a password, all	MasterAuth: "123456"

	nodes regardless of being master or slave need to specify this parameter.	
ClusterEnabled	Whether to enable cluster mode.	yes / no，eg: ClusterEnabled: "yes"
ClusterConfigFile	Name of each cluster node's configuration file, automatically generated and updated by the node, cannot be renamed after generation, and cannot be manually edited.	ClusterConfigFile: "./node.conf"

5.1.7 Launching the AMDC Cache Service

There are two ways to launch the AMDC cache service: command-line startup and console startup (effective for services deployed through the console). When the console automatically deploys, the AMDC cache service is launched by default.

5.1.7.1 Command Line Startup of the AMDC Cache Service

The command line has both front-end and back-end startup modes. In front-end startup mode, after starting, no other command-line operations can be performed in the terminal. If you want to operate, you must use Ctrl+C, which simultaneously stops the AMDC cache service. Therefore, when launching the AMDC cache service via the command line, it is recommended to use back-end startup mode.

- Front-end startup command for the AMDC cache service: `./amdc-server [-conf conf.yaml]`
- Back-end startup command for the AMDC cache service: `./amdc-server [-conf conf.yaml] --daemonize yes`

At this point, the installation and launch of the AMDC cache service are complete.

Front-end startup screenshot:

```
./amdc-server
```

Successful front-end startup:

```
[WARN][Wed, 11 Oct 2023 10:25:22 CST] The Maxmemory option in the configuration file is biger than those in the license file.
[WARN][Wed, 11 Oct 2023 10:25:22 CST] The Maxmemory option is reset to 5.59GB
                                      Apusic In-Memory Data Cache
     ___                        _
    /   |  ____  __  _______  (_)____
   / /| | / __ \/ / / / ___/ / / ___/   License start: 2023-06-28 00:00:00
  / ___ |/ /_/ / /_/ (__  ) / /__      License end: 2023-12-31 00:00:00
 /_/  |_/ .___/\__,_/____/_/\___/      IP granted: 255.255.255.255
       /_/                             Licensee: 金蝶天燕测试用户
                                       Max clients: unlimited
  version: v2.0     PID:  3937         Max memory: 5.59GB
[INFO][Wed, 11 Oct 2023 10:25:22 CST] ReadonlyProGoNum=4
[INFO][Wed, 11 Oct 2023 10:25:22 CST] IOReadGroutineNum=12
[INFO][Wed, 11 Oct 2023 10:25:22 CST] IOWriteGroutineNum=15
[INFO][Wed, 11 Oct 2023 10:25:22 CST] IOGroutineDoReads=true
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_14 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_7 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_8 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_9 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_3 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_10 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_11 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_1 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_2 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_13 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_12 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_6 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_5 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] io_goroutine_4 start
[INFO][Wed, 11 Oct 2023 10:25:22 CST] The server is running with conf.yaml config file.
[INFO][Wed, 11 Oct 2023 10:25:22 CST] Bind: 0.0.0.0:6359, start listening...
[INFO][Wed, 11 Oct 2023 10:25:22 CST] Ready to accept connections.
```

Back-end startup screenshot:

```
./amdc-server --daemonize yes
```

Upon successful back-end startup, the logs will point to the nohup.out file, as shown in the following figure:

```
[WARN][Wed, 11 Oct 2023 10:34:02 CST] The Maxmemory option in the configuration file is biger than those in the license file.
[WARN][Wed, 11 Oct 2023 10:34:02 CST] The Maxmemory option is reset to 5.59GB
[INFO][Wed, 11 Oct 2023 10:34:02 CST]
                                      Apusic In-Memory Data Cache
     ___                        _
    /   |  ____  __  _______  (_)____
   / /| | / __ \/ / / / ___/ / / ___/   License start: 2023-06-28 00:00:00
  / ___ |/ /_/ / /_/ (__  ) / /__      License end: 2023-12-31 00:00:00
 /_/  |_/ .___/\__,_/____/_/\___/      IP granted: 255.255.255.255
       /_/                             Licensee: 金蝶天燕测试用户
                                       Max clients: unlimited
  version: v2.0     PID:  5100         Max memory: 5.59GB
./amdc-server [PID] 5108 running...
```

5.1.7.2 One-Click Start, Stop, and Restart of AMDC Cache Service via Console

For clusters automatically deployed on the console, the console offers one-click access to start, restart, or stop functionalities, achieving one-button start, stop, and restart. Log into the console, proceed to the [Service List]/[Tenant Details], click the [Settings] button under [Services], then on the cluster settings page, click the [Start], [Stop], [Restart] buttons under the nodes to accomplish one-click start, stop, and restart of the AMDC cache service.

5.1.8 Stopping the AMDC Cache Service

There are three primary methods to halt the AMDC cache service:

1. Client Command Method: Send a shutdown command to the AMDC from the client.
2. Forced Process Termination: Terminate the AMDC cache service process forcibly through terminal commands.
3. One-Click Stop via Console: Effective through the console for AMDC cache services automatically deployed.

Given that the AMDC cache service may be in the process of synchronizing data from memory to disk, forcefully terminating the AMDC cache service process might result in data loss.The proper method to stop the AMDC cache service is by sending a shutdown command from the client to the AMDC cache service, leading to its shutdown.

- Method 1: Stopping the AMDC Cache Service via Client: Send a shutdown command to the AMDC cache service from the client.
- Method 2: Under normal client connection conditions, input the shutdown command; once the AMDC cache service receives the shutdown command, it will sever all client connections, then execute persistence based on its configuration, and ultimately exit.
- Method 3: Forcibly End the AMDC Cache Service Program: Utilize kill -9 on the process's PID to forcibly terminate the AMDC cache service process.
    - Step 1: Identify the process PID by examining the port process command: `netstat -nltp |grep 6359`
    - Step 2: Execute the kill -9 command to forcefully kill the process: `kill -9 Process ID`
- Method 4: One-Click Stop of the AMDC Cache Service via the AMDC Console (valid for clusters automatically deployed on the console) — log in with an administrator account, navigate to [Tenant List] > [Tenant Details] > [Settings], select the [Node], then click the [Stop] button on the toolbar to achieve one-click stop.

5.1.9 Deploying the AMDC National Cryptographic Proxy Service

The AMDC national cryptographic proxy service comprises server-side and client-side components, deployed through command-line methods. There is no requirement for a specific order in deploying the server and client sides. Below is an introduction to deploying the AMDC national cryptographic proxy service via the command line.

5.1.9.1 Configuration Items

AMDC encryption-enabled configuration in conf.yaml

Configuration Domain	Configuration Item	Description
Proxy	Enable	Whether to start encryption
	Bind	Bind to the IP in the License

	Port	Proxy server port
	Proxy2IP	Bind the IP to be proxied
	Proxy2Port	Bind the port to be proxied
	CriptEnabled	Whether to encrypt
	GMflag	Whether to enable national cryptography
	CertPath	Location of the encryption certificate

AMDC-proxy-client client-side configuration

Configuration Domain	Configuration Item	Description
Network	Bind	Bind IP
	Port	Proxy server port
	Proxy2IP	Bind the IP to be proxied
	Proxy2Port	Bind the port to be proxied
Cript	GMflag	Whether to enable national cryptography

Proxy2IP和Proxy2Port的指向关系

Example:

- AMDC's IP is 192.168.0.1 with a port of 6359, the cryptographic proxy service port is 7000
- The client-proxy's IP is 192.168.0.2, the client-proxy port is 8000
- The customer's server IP is 192.168.0.3, the server port is 8080

Modify the AMDC configuration as follows (only showing the configuration items that need modification):

```
Network:
  Bind:
    - "192.168.0.1"
```

```
  Port: 6359
  Proxy:
  Enabled: "yes"
  Bind: "192.168.0.1"
  Port: 7000
  #Due to the ability to bind multiple IPs, the service IP still
needs to be specified manually
  Proxy2IP: "192.168.0.1"
  Proxy2Port: 6359
```

Modify the client-proxy configuration as follows:

```
Network:
  Bind: "192.168.0.2"
  Port: 8000
  Proxy2IP: "192.168.0.1"
  Proxy2Port: 7000
```

After configuring, start the service. The business system should configure the access port for AMDC as the IP 192.168.0.2 and port 7000 of the client-proxy.

5.1.9.2 Usage

The steps to deploy the AMDC national cryptographic proxy service via command-line are mainly:

1. Edit the AMDC configuration file based on the description of the configuration items above.
2. Start AMDC.
3. Upload the AMDC national cryptographic proxy service client installation package (amdc_proxy_client.tar.gz) to the target server (does not need to be on the same server as the client).
4. Unpack the installation package: `tar -zxf amdc_proxy_client.tar.gz`
5. Enter the unpacked folder: `cd amdc-proxy-client`
6. Edit the configuration file based on the description of the configuration items above.
7. Start the proxy service with the command `nohup ./amdc-proxy-client -conf conf.yaml >nohup.out 2>&1 &`

At this point, the national cryptographic proxy service will be fully set up, but using the national cryptographic proxy will reduce the overall performance of AMDC, with a performance loss of around 30%. Note: Encryption currently does not support cluster mode, which will be upgraded in future versions.

5.1.10 Uninstallation of the AMDC Console and Cache Service

Before uninstalling the AMDC console and cache service, please confirm that the service has stopped. After the service has stopped, enter the installation directory and execute the deletion operation, deleting all files.

1. Enter the AMDC installation directory: `cd /opt/`
2. Execute the command to delete the cache service: `rm -r amdc`
3. Execute the command to delete the management console: `rm -r amdc-console/`
4. Delete the installation package: `rm -r [Package Name]`

5.2 Using the Redis Configuration File

Starting from AMDC v2.1, compatibility with Redis configuration files is provided, where AMDC automatically translates the contents of Redis.conf into AMDC configuration content.

Usage: `./amdc-server -conf redis.conf (specify the Redis configuration file)`

6 How AMDC Smoothly Replaces Redis

AMDC is compatible with the Redis protocol and various language clients. There are two approaches to smoothly replacing Redis with AMDC: promotion of a slave node and inheritance of RDB/AOF files.

6.1 Promotion of a Slave Node

Utilize the data synchronization in the master-slave mode to obtain data from Redis. Once the synchronization is successfully completed, shut down the Redis node and manually or wait for the sentinel or cluster's automatic failover feature to switch AMDC to the master node.

To manually promote a slave node, choose one of the following methods:

1. Use `amdc-cli -h <ip> -p <port> slaveof no one`
2. Connect to the slave node and use the command `replicaof no one`

Promotion of a slave node by the sentinel

1. Modify the AMDC configuration file to add the Redis master node to the AMDC configuration file
   `vim /installation_directory/amdc/conf.yaml`
   Modify the Replicaof field to: `Replicaof: [Redis master service IP] [Master service port]`
2. After successful data synchronization, set AMDC as the sentinel master node and start the AMDC sentinel. Once the sentinel is started, the Redis node can be shut down. Successful synchronization:

Sentinel configuration:

Configuration Item	Meaning
Bind	IP address bound to the sentinel, multiple IP addresses can be bound
Port	Port listened to by the sentinel
Sentinel monitor	Master service IP, port, quorum number (less than the number of slave nodes, more than half the number of slave nodes)

6.2 Inheritance of RDB/AOF Files

Step 1: Obtain the original Redis data persistence files RDB/AOF, place the persistence file in the location specified in the AMDC configuration file: `cat /installation_directory/amdc/conf.yaml`, determine the file location by checking the DbFileName, Dir fields.

Configuration Item	Default Value	Meaning
DbFileName	"dump.rdb"	RDB file name, excluding path
Dir	"./"	RDB and AOF files stored in the installation directory

Step 2: Restart the AMDC node.

6.3 Replacement in Cluster Mode

In cluster mode, assistance from the RDB data migration tool is required. Refer to the section on the 《RDB Cluster Data Migration Tool》.

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