



Migration Guide

Apusic Load Balancer

Copyright Notice

All intellectual property rights including software copyrights involved in this document have been legally registered and duly owned by Kingdee Apusic Cloud Computing Co., Ltd. Such rights are protected by Copyright Law of the People's Republic of China, Regulations on Protection of Computer Software, Regulations on Intellectual Property Protection, relevant international copyright conventions, laws, regulations and other intellectual property legislations. Unauthorized use is strictly prohibited.

Disclaimer

All copyright information contained herein is legally owned by Kingdee Apusic Cloud Computing Co., Ltd. and protected by applicable laws. The Company shall bear no liability for any information unrelated to Kingdee Apusic contained in this document. You may view, copy and print this document only within the scope permitted by Copyright Law of the People's Republic of China. No entity or individual may use, modify or republish any part or content of this document without written authorization from Kingdee Apusic Cloud Computing Co., Ltd. Any violation shall constitute infringement, and the Company reserves the right to pursue legal liabilities accordingly.

Updates to this document shall not be notified separately. You may consult the Company regarding issues related to this document. All rights not expressly granted by the Company are reserved.

Trademark Notice

 is a registered trademark filed with the National Intellectual Property Administration of PRC by Kingdee Apusic Cloud Computing Co., Ltd. Shenzhen. Exclusive rights to the registered trademark are legally owned by Kingdee Apusic and protected by law. Without written consent of Kingdee Apusic, no entity or individual shall use, copy, modify, disseminate, transcribe, or bundle and sell any part of this trademark with other products in any manner or for any reason. Kingdee Apusic shall pursue legal liabilities against any infringement of its trademark rights. All other trademarks or registered trademarks mentioned in this document belong to their respective owners.

Contents

- 1 Introduction
 - 1.1 Target Audience
 - 1.2 Related Documents
 - 1.3 Technical Support
- 2 Migration Overview
- 3 Supported Migration Versions
- 4 Migrating from V2.0.5 to V2.0.6
- 5 Migrating from V2.0.2 Agile Edition to V2.0.6
- 6 Migrating from Nginx to ALB (Using the ALB Migration Tool)
 - 6.1 Features
 - 6.1.1 Configuration Migration
 - 6.1.2 SSL Certificate Migration
 - 6.1.3 Static Resource Migration
 - 6.1.4 Log Path Migration
 - 6.1.5 Other Features
 - 6.2 Usage
 - 6.3 Command-Line Options
 - 6.4 Usage Examples
 - 6.5 Directory Structure After Migration
 - 6.6 Notes
 - 6.6.1 Path Issues
 - 6.6.2 Backup
 - 6.6.3 Permissions
 - 6.6.4 Verification
- 7 Migrating from Nginx to ALB (Manual Migration)
 - 7.1 Simple Direct Use of the nginx.conf File
- 8 FAQ
 - 8.1 Unable to Log In to the Console After Migration
 - 8.2 License Verification Failure After Migration

1 Introduction

This document is the migration guide for Kingdee Apusic Load Balancer software V2.0.6 Standard Edition. It describes how to migrate and upgrade from historical ALB versions to V2.0.6 Standard Edition, helping users complete the version upgrade smoothly.

1.1 Target Audience

This document is intended for IT business leads, R&D managers, software project managers, software architects, and operations engineers.

1.2 Related Documents

For more information about ALB V2.0.6 products, please refer to the following ALB V2.0.6 product manual document set:

No.	Manual Document	Description
1	ALB V2.0.6 Release Notes	Introduces the version update contents of the ALB product.
2	ALB V2.0.6 Quick Start Guide	Briefly introduces how to quickly get started with ALB.
3	ALB V2.0.6 Installation Guide	Describes in detail how to install ALB on various operating systems, how to start and stop the ALB service, and the product registration process.
4	ALB V2.0.6 Proxy Function Manual	Describes in detail the usage, configuration, management, and supporting tools of ALB-related functions.
5	ALB V2.0.6 Console User Manual	Describes in detail the usage and operation of ALB Console functions.
6	ALB V2.0.6 Developer Manual	Describes in detail the external API of ALB, the alb-cli command-line tool, and the AI Agent integration methods.
7	ALB V2.0.6 Migration Guide	Describes in detail how to migrate and upgrade from historical ALB versions to V2.0.6.
8	ALB V2.0.6 Operations Manual	Describes in detail the monitoring, operations, security hardening, and other operations of ALB.
9	ALB V2.0.6 Performance Optimization Guide	Describes in detail the performance tuning of ALB.

1.3 Technical Support

ALB products provide comprehensive technical support services. You can obtain technical support through the following channels:

- Website: www.apusic.com
- Phone: 400-855-5800
- Email: support@apusic.com
- Kingdee Cloud Community: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

When requesting technical support, please provide the following information:

1. Your name
2. Company information and contact details
3. Operating system and its version
4. Product version number
5. Detailed information about the exception and errors, such as logs and screenshots

2 Migration Overview

ALB V2.0.6 Standard Edition has been comprehensively upgraded based on V2.0.5 Standard Edition, mainly including:

- The console is fully upgraded, supporting finer-grained three-role permission management and auditing functions.
- The installation package directory structure is unified as `ALB-V2.0.6-SE-amd64` / `ALB-V2.0.6-SE-arm64` .
- The console binary is unified as `console/alb-console` , and a new user database structure is adopted.
- The configuration file remains **Nginx** compatible, and the `alb.conf` from historical versions can be reused directly.
- License files are compatible: `license.lic` , `license.xml` , and the unified license `acls.properties` can all continue to be used.

3 Supported Migration Versions

Source Version	Target Version	Description
ALB V2.0.5 Standard Edition	ALB V2.0.6 Standard Edition	Recommended migration path, highly compatible configuration files
ALB V2.0.2 Standard Edition	Compatible	If the standard Nginx configuration files are used, an overall upgrade to V2.0.6 is possible
ALB V2.0.2 Agile Edition	ALB V2.0.6 Standard Edition	Large functional differences; it is recommended to deploy as a new installation and then migrate business configurations

4 Migrating from V2.0.5 to V2.0.6

V2.0.5 can be upgraded to V2.0.6 without downtime. The operations are as follows:

1. Back up the existing environment

- Back up the old version installation directory, for example:

```
cp -r /opt/ALB-V2.0.5-SE-amd64 /opt/ALB-V2.0.5-SE-  
amd64.bak
```

2. Upload the V2.0.6 installation package to the /opt directory and extract it

- Extract the installation package

```
tar -zxvf ALB-V2.0.6-SE-*.tar.gz -C /opt/
```

3. Replace the lualib directory

- This migration uses an in-place replacement strategy: while preserving the original installation directory structure, the new V2.0.6 files are gradually replaced into the V2.0.5 directory. Replace the `lualib` directory:

```
mv /opt/ALB-V2.0.5-SE-amd64/lualib /opt/ALB-V2.0.5-SE-  
amd64/lualib.bak  
cp -r /opt/ALB-V2.0.6-SE-amd64/lualib /opt/ALB-V2.0.5-SE-  
amd64/lualib
```

4. Replace the alb executable

- Replace `sbin/alb` :

```
mv opt/ALB-V2.0.5-SE-amd64/sbin/alb opt/ALB-V2.0.5-SE-  
amd64/sbin/alb.bak
```

```
cp /opt/ALB-V2.0.6-SE-amd64/sbin/alb /opt/ALB-V2.0.5-SE-
amd64/sbin/alb
```

5. Replace the startup scripts

- Replace the startup scripts in the `bin` directory:

```
cp /opt/ALB-V2.0.6-SE-amd64/bin/* /opt/ALB-V2.0.5-SE-
amd64/bin/
```

6. Send the restart signal

- Send the `USR2` signal to the `alb` master process to reload the configuration file:

```
kill -USR2 $(cat /opt/ALB-V2.0.5-SE-amd64/logs/alb.pid)
# Check whether both old and new processes exist, that
is, whether there are two alb master processes
ps -ef | grep alb | grep -v grep
# Check whether the alb.pid.oldbin file exists in the
logs directory. If it exists, both old and new processes
are running
```

7. Gracefully stop the old worker and master processes

```
# Gracefully stop the old worker process
kill -WINCH $(cat /opt/ALB-V2.0.5-SE-amd64/logs/alb.pid.oldbin)

# Gracefully stop the old master process
kill -QUIT `cat /opt/ALB-V2.0.5-SE-amd64/logs/alb.pid.oldbin`
```

8. Check whether only the new process remains

- If only one alb master process exists, the old version has been stopped

9. Check the version

```
./sbin/alb -v  
# The output is:  
alb version: alb/2.0.6-260615
```

10. Console offline upgrade

```
# Find the console process and kill it  
ps aux | grep alb-console | grep -v grep | awk '{print $2}' |  
xargs kill -9  
# Back up the old console directory  
mv /opt/ALB-V2.0.5-SE-amd64/console /opt/ALB-V2.0.5-SE-  
amd64/console.bak  
# Replace the console directory  
cp -r /opt/ALB-V2.0.6-SE-amd64/console /opt/ALB-V2.0.5-SE-  
amd64/console  
# Start the console (use the absolute path to avoid directory  
errors)  
cd /opt/ALB-V2.0.5-SE-amd64/console && nohup ./alb-console >  
nohup.log 2>&1 &
```

5 Migrating from V2.0.2 Agile Edition to V2.0.6

V2.0.2 Agile Edition can be migrated to V2.0.6. The operations are as follows:

1. Back up the existing environment

- Back up the old version installation directory, for example:

```
cp -r /opt/ALB-V2.0.2-AE-amd64 /opt/ALB-V2.0.2-AE-  
amd64.bak
```

2. Upload the V2.0.6 Standard Edition installation package to the /opt directory and extract it

- Extract the installation package (the Standard Edition installation package can be used to migrate and replace binaries and scripts)

```
tar -zxvf ALB-V2.0.6-SE-*.tar.gz -C /opt/
```

3. Copy the lualib directory

- First back up the old lualib, then copy the new directory:

```
mv /opt/ALB-V2.0.2-AE-amd64/lualib /opt/ALB-V2.0.2-AE-  
amd64/lualib.bak  
cp -r /opt/ALB-V2.0.6-SE-amd64/lualib /opt/ALB-V2.0.2-AE-  
amd64/lualib
```

4. Replace the alb executable

- Replace `sbin/alb` :

```
mv /opt/ALB-V2.0.2-AE-amd64/sbin/alb /opt/ALB-V2.0.2-AE-  
amd64/sbin/alb.bak
```

```
cp /opt/ALB-V2.0.6-SE-amd64/sbin/alb /opt/ALB-V2.0.2-AE-
amd64/sbin/alb
```

5. Replace the startup scripts

- Replace the startup scripts in the `bin` directory:

```
cp /opt/ALB-V2.0.6-SE-amd64/bin/* /opt/ALB-V2.0.2-AE-
amd64/bin/
```

6. Send the restart signal

- Send the `USR2` signal to the `alb` master process to reload the configuration file:

```
kill -USR2 $(cat /opt/ALB-V2.0.2-AE-amd64/logs/alb.pid)
# Check whether both old and new processes exist, that is,
whether there are two alb master processes
ps -ef | grep alb | grep -v grep
# Check whether the alb.pid.oldbin file exists in the logs
directory. If it exists, both old and new processes are running
```

7. Gracefully stop the old worker and master processes

```
# Gracefully stop the old worker process
kill -WINCH $(cat /opt/ALB-V2.0.2-AE-amd64/logs/alb.pid.oldbin)

# Gracefully stop the old master process
kill -QUIT `cat /opt/ALB-V2.0.2-AE-amd64/logs/alb.pid.oldbin`
```

8. Check whether only the new process remains

- If only one alb master process exists, the old version has been stopped

9. Check the version

```
./sbin/alb -v
# The output is:
alb version: alb/2.0.6-260615
```

10. Console offline upgrade

```
# Find the console process and kill it
ps aux | grep alb-2.0-light-console | grep -v grep | awk
'{print $2}' | xargs kill -9
# Back up the old console directory
mv /opt/ALB-V2.0.2-AE-amd64/console /opt/ALB-V2.0.2-AE-
amd64/console.bak
# Replace the console directory
cp -r /opt/ALB-V2.0.6-SE-amd64/console /opt/ALB-V2.0.2-AE-
amd64/console
# Start the console (use the absolute path to avoid directory
errors)
cd /opt/ALB-V2.0.2-AE-amd64/console && nohup ./alb-console >
nohup.log 2>&1 &
```

6 Migrating from Nginx to ALB (Using the ALB Migration Tool)

ALB comes with a nginx configuration file migration tool, which is used to migrate nginx configurations to ALB configurations.

ALB-Nginx migration tool: `[ALB installation path]/utils/nginx-migrate`

6.1 Features

6.1.1 Configuration Migration

- Scan the nginx main configuration file (nginx.conf)
- Parse and migrate all include sub-configuration files
- Sub-configurations are migrated to the `conf/conf.d/` directory
- Automatically update the include paths in the main configuration to `conf.d/*.conf`

6.1.2 SSL Certificate Migration

- Automatically scan `ssl_certificate`, `ssl_certificate_key`, and `ssl_trusted_certificate`
- Copy certificate files to the `conf/certs/` directory
- Update the certificate paths in the configuration to the new paths

6.1.3 Static Resource Migration

- Automatically scan `root` and `alias` directives
- Copy static resources to the `conf/html/` directory
- Update the static resource paths in the configuration to absolute paths

6.1.4 Log Path Migration

- Automatically migrate `error_log`, `access_log`, and `pid` paths
- Unified migration to the `{ALBPath}/logs/` directory

6.1.5 Other Features

- Automatically back up the original configuration before migration
- Interactive confirmation (can be skipped)
- Verbose log mode
- Configuration validation

6.2 Usage

Prerequisites:

1. ALB has already been deployed and installed on the same machine.
2. The directory containing the nginx configuration files and the ALB installation directory already exist.

```
# Automatically detect the nginx configuration and ALB path
./nginx-migrate

# Specify the nginx configuration and ALB directory
./nginx-migrate --nginx-path /path/to/nginx.conf --alb-path
/path/to/ALB
```

6.3 Command-Line Options

Option	Short Form	Description	Default
--nginx-path	-	nginx configuration file path	Auto-detect
--alb-path	-	ALB installation directory	Inferred from the tool location
--from-running	-	Obtain the configuration path from the running nginx process	false
--dry-run	-	Only scan, do not perform migration	false
-y / --yes	-	Automatically confirm all prompts	false
-v / --verbose	-	Show verbose logs	false
--backup	-	Back up the original configuration before migration	true

6.4 Usage Examples

```
# Obtain the configuration from the running nginx process and
migrate
./nginx-migrate --from-running --alb-path /path/to/ALB
```

```
# Full mode (auto-confirm + verbose logs)
./nginx-migrate --nginx-path /etc/nginx/nginx.conf --alb-path
/opt/ALB -y -v

# Preview mode (scan only)
./nginx-migrate --nginx-path /etc/nginx/nginx.conf --alb-path
/opt/ALB --dry-run

# Disable backup
./nginx-migrate --nginx-path /etc/nginx/nginx.conf --alb-path
/opt/ALB --backup=false
```

6.5 Directory Structure After Migration

```
ALB/
├─ conf/
│   ├── alb.conf          # Main configuration file
│   ├── certs/           # SSL certificates
│   │   ├── server.crt
│   │   ├── server.key
│   │   └─ ca.crt
│   ├── conf.d/          # Sub-configuration files
│   │   ├── default.conf
│   │   ├── ssl.conf
│   │   └─ upstream.conf
│   └─ html/             # Static resources
│       ├── index.html
│       ├── css/
│       └─ js/
└─ logs/                 # Log directory
    ├── access.log
    ├── error.log
    └─ nginx.pid
```

6.6 Notes

6.6.1 Path Issues

- Ensure that the paths in the nginx configuration file are **relative paths** or **absolute paths**
- Relative paths will be resolved based on the directory where the configuration file is located
- It is recommended to use `--dry-run` to preview in the test environment first

6.6.2 Backup

- Before migration, the original configuration is automatically backed up to `conf.backup_YYYYMMDD_HHMMSS`
- To restore, delete the new configuration and rename the backup directory to `conf`

6.6.3 Permissions

- Ensure that you have permission to read the nginx configuration and certificate files
- Ensure that you have permission to write to the ALB directory

6.6.4 Verification

- After the migration is complete, the configuration syntax is automatically verified
- It is recommended to manually check the configuration before restarting ALB

7 Migrating from Nginx to ALB (Manual Migration)

7.1 Simple Direct Use of the nginx.conf File

1. Back up the old configuration

- Back up the old ALB configuration, and replace nginx.conf with alb.conf:

```
cp /opt/ALB-V2.0.6-SE-amd64/conf/alb.conf /opt/ALB-V2.0.6-SE-amd64/conf/alb.conf.bak
cp /path/to/nginx.conf /opt/ALB-V2.0.6-SE-amd64/conf/alb.conf
# Modify the source path according to the actual situation, and copy the nginx.conf file to the target path
chmod 644 /opt/ALB-V2.0.6-SE-amd64/conf/alb.conf
```

2. Adjust include paths

- If the original nginx.conf uses the `include` directive to reference sub-configuration files (such as `conf.d/*.conf`), make sure these paths are valid in the ALB installation directory. It is recommended to copy the sub-configuration files to the `/opt/ALB-V2.0.6-SE-amd64/conf/conf.d/` directory, and modify the include paths to relative or absolute paths:

```
mkdir -p /opt/ALB-V2.0.6-SE-amd64/conf/conf.d
cp /path/to/nginx/conf.d/*.conf /opt/ALB-V2.0.6-SE-amd64/conf/conf.d/
```

3. Migrate SSL certificate files

- Copy the SSL certificate files used by the original nginx to the ALB directory, and update the certificate paths in alb.conf:

```
mkdir -p /opt/ALB-V2.0.6-SE-amd64/conf/certs
cp /path/to/ssl/server.crt /opt/ALB-V2.0.6-SE-amd64/conf/certs/
cp /path/to/ssl/server.key /opt/ALB-V2.0.6-SE-amd64/conf/certs/
```

- Edit alb.conf, and update the `ssl_certificate` and `ssl_certificate_key` paths to the new absolute paths.

4. Adjust log paths

- Modify the `error_log`, `access_log`, and `pid` paths in alb.conf to the ALB log directory:

```
error_log /opt/ALB-V2.0.6-SE-amd64/logs/error.log;
access_log /opt/ALB-V2.0.6-SE-amd64/logs/access.log;
pid /opt/ALB-V2.0.6-SE-amd64/logs/alb.pid;
```

5. Verify the configuration and start ALB

- Use the built-in configuration check command of ALB to verify the syntax:

```
/opt/ALB-V2.0.6-SE-amd64/sbin/alb -t -c /opt/ALB-V2.0.6-SE-
amd64/conf/alb.conf
```

- Stop Nginx, and start ALB:

```
/path/to/sbin/nginx -s stop    # Modify the Nginx installation
path according to the actual situation
/opt/ALB-V2.0.6-SE-amd64/bin/start-alb.sh
```

8 FAQ

8.1 Unable to Log In to the Console After Migration

- Cause: V2.0.6 Console uses a new user database, and the old version user data is incompatible.
- Solution: When the V2.0.6 Console is started for the first time, the system automatically generates an initial random password and outputs it to the console log. Please check the log to obtain the default user account and password for login. To reset the password, please operate through the console interface or contact the system administrator. Do not modify system files manually.

8.2 License Verification Failure After Migration

- Cause: The license file is not correctly copied to the new version installation directory, or the license file has expired.
- Solution: Check whether `license.lic` , `license.xml` , or `acls.properties` exists and is valid.

UNIFIED SERVICE HOTLINE

4008-555-800

Kingdee Apusic Cloud Computing Co., Ltd. (referred to as “Kingdee Apusic”) was founded in 2000. Its predecessor was “Kingdee Middleware Company.” It is a basic software platform provider under Kingdee Group, a national standards-setting enterprise for cloud computing, and a core software enterprise in the national IT application innovation industry. Kingdee Apusic is an enterprise undertaking the National 863 Key R&D Program and the National Major Science and Technology Project for Core Electronic Components, High-end General-purpose Chips and Fundamental Software, and is also a basic platform provider for the national key projects of “Two Networks, One Website, Four Databases, and Twelve Golden Projects.” Its products are widely used in key industries such as government, military, finance, and energy, and it has cumulatively served more than 100,000 customers.



National standard setter for cloud computing
Basic software enterprise under Kingdee Group
Core enterprise for IT application innovation
Website: www.apusic.com

