



Nginx 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 Preface
 - 1.1 Target Audience
 - 1.2 Related Documents
 - 1.3 Technical Support
- 1.3.1 Nginx Compatibility
- 2 Overall Compatibility Statement
- 3 Configuration Compatibility Classification
- 4 Built-in Module List
- 5 Product Differences Compared with Nginx
- 6 Typical Use Scenario Compatibility Quick Reference
- 6.0.1 Migration Guide
 - 7 Migration Process Overview
 - 8 Pre-Migration Evaluation and Inventory
 - 9 Install ALB and Apply for License
 - 10 Migration Method 1: Console "Migrate from Nginx" Wizard (Recommended)
 - 11 Migration Method 2: nginx-migrate Command-Line Tool
 - 11.1 Features
 - 11.2 Usage Method
 - 11.3 Command-Line Options
 - 11.4 Usage Examples
 - 11.5 Post-Migration Directory Structure
 - 11.6 Notes
 - 12 Migration Method 3: Manual Migration
 - 12.1 Simple Direct Use of nginx.conf File
 - 13 Post-Migration Adaptation and Adjustment
 - 13.1 Ports and Listening
 - 13.2 Users and Permissions
 - 13.3 Path Verification Checklist
 - 13.4 National Cryptographic Certificates
 - 13.5 Third-Party Modules
 - 13.6 Service Start/Stop and Boot Auto-Start
 - 13.7 Log Rotation

- 13.8 Monitoring Integration
 - 13.9 Optional Enhancements (Enable as Needed After Migration)
 - 14 Verification and Cutover
 - 14.1 Configuration Verification
 - 14.2 Parallel Verification (Recommended)
 - 14.3 Business Cutover
 - 15 Rollback Plan
- 15.0.1 FAQ
- 15.1 After Migration, Executing `alb -t` Reports unknown directive Error
 - 15.2 After Migration, `load_module` Loading Failed
 - 15.3 Startup Failed: `getgrnam("nobody")` failed
 - 15.4 After Migration, Accessing the Page Returns a 403 Error
 - 15.5 502 Error: no live upstreams while connecting to upstream
 - 15.6 Regular User Starting ALB Cannot Listen on Port 80
 - 15.7 After Migration, Security Scanning or Front-End Validation Anomalies
 - 15.8 Configuration Imported by the Console Migration Wizard Does Not Take Effect
- 15.8.1 Appendix: Nginx → ALB Common Operations Quick Reference

1 Preface

This document is the Nginx Adaptation and Migration Guide for the Apusic Application Server Load Balancer (ALB) V2.0.6 Standard Edition. It is intended for users currently using Nginx and introduces ALB V2.0.6's Nginx compatibility, as well as the complete methodology for smoothly migrating from Nginx to ALB V2.0.6, helping users successfully complete the replacement and upgrade.

1.1 Target Audience

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

1.2 Related Documents

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

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

10	ALB V2.0.6 Nginx Adaptation and Migration Guide	Details ALB's Nginx compatibility and the methodology for migrating from Nginx to ALB (this document).
----	---	--

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 seeking 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 such as logs and screenshots of the anomalies and errors that occurred

1.3.1 Nginx Compatibility

2 Overall Compatibility Statement

The ALB V2.0.6 core is compatible with the configuration syntax and directive system of **Nginx 1.10 ~ Nginx 1.30** versions. A standard `nginx.conf` configuration file can be directly reused as the ALB main configuration `alb.conf`, and in most scenarios the replacement can be completed without rewriting the configuration.

On top of Nginx compatibility, ALB V2.0.6 also provides the following enhanced capabilities:

- **National Cryptographic SSL:** Supports the SM2 dual-certificate (signing certificate + encryption certificate) system.
- **Active Health Check:** Built-in `alb_healthcheck_module`, supports layer-4 (TCP/UDP) and layer-7 (HTTP) active probing.
- **Session Persistence:** Built-in sticky module, supports cookie, route, and learn working modes.
- **Traffic Monitoring:** Built-in vts/sts status statistics module, provides page, JSON, and Prometheus monitoring data output.
- **Protocol Capabilities:** Supports HTTP/2, HTTP/3 (QUIC), gRPC, WebSocket, RTMP live streaming, HTTP CONNECT forward proxy.
- **Extensibility:** Built-in Lua scripting engine, Brotli compression, headers-more, geo2 geolocation module.
- **Management Capabilities:** Provides Web Console (site management, configuration management, certificate management, high availability, backup/restore, third-party module management), alb-cli command line, and AI Agent integration.
- **High Availability:** Supports active/standby HA (including unicast mode), configuration synchronization, smooth upgrade and rollback.
- **TCP Acceleration:** The ebpf feature installation package supports kernel eBPF accelerated TCP forwarding.

3 Configuration Compatibility Classification

The directives in existing Nginx configurations are classified into the following four levels based on their compatibility with ALB V2.0.6:

Compatibility Level	Description	Typical Directives/Configuration
Fully compatible, directly reusable	Nginx standard directives, syntax and behavior consistent, no modification needed	<code>worker_processes</code> , <code>events</code> , <code>http/server/location</code> , <code>upstream</code> (round-robin, <code>weight</code> , <code>ip_hash</code> , <code>least_conn</code> , <code>hash ... consistent</code>), <code>proxy_pass</code> and <code>proxy_*</code> series, <code>listen</code> , <code>server_name</code> , <code>root/alias</code> , <code>index</code> , <code>try_files</code> , <code>rewrite/return</code> , <code>map/geo</code> , <code>gzip</code> , <code>proxy_cache</code> , <code>limit_req/limit_conn</code> , <code>allow/deny</code> , <code>auth_basic</code> , <code>error_page</code> , <code>add_header</code> , <code>ssl_*</code> (RSA/ECC certificates), <code>stub_status</code> , <code>stream</code> layer-4 proxy, <code>include</code> , etc.
Compatible, but requires adjustment	Syntax compatible, but due to differences in product directory structure and default values, paths or parameters need adjustment	Log paths (<code>access_log</code> , <code>error_log</code>), <code>pid</code> path, certificate file paths, static resource <code>root/alias</code> paths, <code>include</code> relative paths, default listening port (ALB factory default is 8080), <code>user</code> directive (default <code>nobody</code>)
ALB extensions, enable as needed	Capabilities not present in Nginx open source version, available after migration to enhance configuration	<code>check</code> (active health check), <code>sticky</code> (session persistence), <code>vhost_traffic_status_*</code> (vts monitoring), <code>stream_server_traffic_status_*</code> (layer-4 monitoring), national cryptographic dual-certificate configuration (<code>ssl_certificate</code> configures sig + enc dual certificates), <code>brotli_*</code> , Lua directives such as <code>access_by_lua_*</code> , <code>more_set_headers</code> , <code>proxy_connect</code> , <code>geo2</code> , eBPF acceleration
Incompatible, manual handling required	Not built into ALB or conflicts with product mechanism, needs to be replaced or removed during migration	Third-party dynamic modules (<code>.so</code>) compiled for Nginx cannot be directly loaded in ALB; modules built for ALB need to be used and uploaded/enabled through the Console "Third-party Module Management"; Nginx commercial version (Nginx Plus) exclusive directives are not in the compatibility scope; module directives not in the ALB built-in module list will report <code>unknown directive</code> errors during <code>alb -t</code> validation

Note: For Nginx standard directives not listed in the table above, please refer to the actual validation result of

```
./sbin/alb -t .
```

4 Built-in Module List

ALB V2.0.6 is factory-compiled with the following modules built-in, corresponding to Nginx same-name compilation options or modules:

Module	Function Description
<code>--with-http_stub_status_module</code>	Enables the <code>stub_status</code> interface to provide runtime status statistics.
<code>--with-http_ssl_module</code>	Enables HTTPS/SSL support (including national cryptographic SSL).
<code>--with-ipv6</code>	Enables IPv6 listening support.
<code>--with-http_mp4_module</code>	Supports MP4 streaming pseudo-streaming.
<code>--with-http_auth_request_module</code>	Implements sub-request authentication (<code>auth_request</code>), facilitating integration with external authentication services.
<code>--with-http_gzip_static_module</code>	Supports directly sending <code>.gz</code> pre-compressed files, saving CPU.
<code>--with-http_realip_module</code>	Obtains the real client IP from X-Forwarded-For/X-Real-IP and other headers.
<code>--with-http_gunzip_module</code>	Automatically decompresses response content for clients that do not support gzip.
<code>--with-http_sub_module</code>	Response body string replacement (<code>sub_filter</code>).
<code>--with-stream</code>	Enables layer-4 TCP/UDP proxy (stream subsystem).
<code>--with-stream_ssl_module</code>	Provides SSL/TLS support for the stream subsystem.
<code>--with-threads</code>	Enables thread pool support to improve file I/O performance.
<code>--with-file-aio</code>	Enables asynchronous file I/O (Linux AIO).
<code>--with-http_v2_module</code>	Enables HTTP/2 support.
<code>--with-http_v3_module</code>	Enables HTTP/3 (QUIC) support.
<code>--with-http_addition_module</code>	Appends content before and after response (<code>add_before_body</code> / <code>add_after_body</code>).
<code>--with-http_dav_module</code>	Supports WebDAV protocol (PUT, DELETE, MKCOL, etc.).
<code>--with-http_random_index_module</code>	Randomly selects the default file when listing directories.

<code>--with-http_secure_link_module</code>	Generates and validates "secure links" with time-based anti-tampering.
<code>--with-mail</code>	Enables mail proxy (IMAP/POP3/SMTP) subsystem.
<code>--with-mail_ssl_module</code>	Provides SSL/TLS support for mail proxy.
<code>--with-http_slice_module</code>	Supports large file slice download (Range Slice).
<code>--with-stream_ssl_preread_module</code>	Pre-reads SNI/ALPN in layer-4 proxy to enable domain-based routing.
<code>alb_healthcheck_module</code>	ALB active health check module, supports layer-4 and layer-7 proxy node active probing.
<code>alb-module-vts</code>	Provides detailed traffic / status statistics interface (<code>/status</code>).
<code>alb-module-stream-sts</code>	Provides real-time status statistics for the stream subsystem.
<code>alb-module-sts</code>	Provides real-time status statistics for the http subsystem.
<code>alb_http_proxy_connect_module</code>	Supports HTTP CONNECT method to forward proxy HTTPS traffic.
<code>alb-rtmp-module</code>	Introduces RTMP/HLS/DASH live streaming and streaming media functions.
<code>alb_brotli</code>	Supports Brotli compression algorithm to reduce transmission volume.
<code>alb-module-lua</code>	Introduces Lua scripting functionality, extensible ALB functions.
<code>alb-headers-more</code>	Sets and clears request and response headers.

If your existing Nginx uses modules outside the above table (third-party modules compiled via `--add-module` or dynamic modules loaded via `load_module`), please investigate and confirm individually before migration, and contact technical support to obtain ALB-version module files if necessary.

5 Product Differences Compared with Nginx

After migrating to ALB, the objects and methods of daily operations differ from Nginx in the following aspects:

Item	Nginx	ALB V2.0.6	Migration Impact
Binary Program	<code>sbin/nginx</code>	<code>sbin/alb</code>	Program names in scripts and operations commands need to be replaced
Main Configuration File	<code>conf/nginx.conf</code>	<code>conf/alb.conf</code>	The original <code>nginx.conf</code> can be directly copied as <code>alb.conf</code>
Factory Default HTTP Port	80	8080	When migrating the configuration, follow the <code>listen</code> from the original <code>nginx</code> configuration
pid File	<code>logs/nginx.pid</code>	<code>logs/alb.pid</code>	The pid path in operations scripts needs adjustment
Configuration Syntax Check	<code>nginx -t</code>	<code>./sbin/alb -t</code>	Same usage
Smooth reload	<code>nginx -s reload</code>	<code>bin/reload-alb.sh</code> (OR <code>kill -HUP</code>)	Recommended to use the scripts provided with the installation package
Start/Stop Method	<code>nginx / nginx -s quit</code>	<code>bin/start-alb.sh</code> , <code>bin/stop-alb.sh</code> , <code>bin/start-alb-and-console.sh</code> , <code>bin/stop-</code>	Refer to the Quick Start

		<code>alb-and-console.sh</code>	Guide
Process Signals	HUP/USR2/WINCH/QUIT	Same-name signals fully compatible, supports USR2 smooth upgrade	Original signal-based operation habits can be preserved
systemd Service	Self-written nginx.service	Installation package <code>utils/systemd/</code> provides <code>alb.service</code> and <code>enable_startup.sh</code> one-click registration	Need to replace the original <code>nginx.service</code> and disable it
Response Server Header	<code>Server: nginx</code>	<code>Server: alb</code>	If there are front-end validation or security scanning rules that depend on this string, adjustments are needed
Product Licensing	None	Need to place license file: <code>license.lic</code> , <code>license.xml</code> or unified license <code>acls.properties</code>	Apply for license before migration, refer to the Installation Guide
Visual Management	Not available in open source version	Console (default port 8887), supports site/configuration/certificate/HA/backup management	Can be migrated to Console management after migration
Monitoring Interface	<code>stub_status</code>	Compatible with <code>stub_status</code> , additionally provides <code>vts /status</code> (page/JSON/Prometheus)	Monitoring collection can be smoothly upgraded

6 Typical Use Scenario Compatibility Quick Reference

Nginx Typical Scenario	ALB V2.0.6 Compatibility	Reference Chapter (Proxy Function Manual)
Static resource hosting / SPA single-page application	Fully supported	User Scenario Configuration Examples - Scenario One
Reverse proxy and load balancing	Fully supported, supports round-robin/weight/IP hash/least connections/consistent hash	Load Balancing Algorithms
HTTPS / SSL Termination	Fully supported, and supports national cryptographic dual certificates	HTTPS and National Cryptographic
HTTP/2	Fully supported	HTTPS and National Cryptographic
HTTP/3 (QUIC)	Supported (from V2.0.6)	HTTPS and National Cryptographic - HTTP/3
WebSocket Proxy	Fully supported	WebSocket Proxy
gRPC Proxy	Fully supported	gRPC Proxy
TCP/UDP Layer-4 Proxy	Fully supported (stream)	TCP Proxy
Forward Proxy (HTTP CONNECT)	Supported	Forward Proxy
Reverse Proxy Cache (proxy_cache)	Fully supported	Cache Configuration
Rate Limiting (limit_req/limit_conn)	Fully supported	Traffic Control
URL Rewriting and Redirection	Fully supported	URL Rewriting and Redirection
Cross-Origin (CORS)	Fully supported	Cross-Origin Configuration (CORS)
Failover and Retry	Fully supported (proxy_next_upstream)	Failover and Retry
Session Persistence	Supports ip_hash / Cookie consistent hash	Session Persistence

	/ sticky module / Lua+Redis	
Active Health Check	Supported (<code>check</code> directive, not available in Nginx open source version)	Health Check
Canary Release	Supported (by IP, ratio, Cookie, Lua dynamic routing)	Canary Release and Canary Testing
Compression	Supports Gzip, Brotli, pre-compressed static files	Compression
Mail Proxy (IMAP/POP3/SMTP)	Supported	Built-in Module List
RTMP/HLS Live Streaming	Supported	Built-in Module List
Lua Extension Development	Built-in Lua engine and lua-resty ecosystem libraries	Session Persistence - Method Three
Dynamic DNS Resolution (resolver)	Fully supported	Dynamic DNS Resolution

6.0.1 Migration Guide

7 Migration Process Overview

The overall process for migrating from Nginx to ALB V2.0.6 is as follows:

Migration Evaluation and Inventory → Install ALB and Apply for License
 → Configuration Migration (choose one of three methods)
 → Adaptation and Adjustment → Verification and Testing → Business
 Cutover → Observation Period (Keep Rollback Capability)

Comparison of three configuration migration methods:

Migration Method	Applicable Scenarios	Features
Console "Migrate from Nginx" Wizard	All scenarios, recommended	Visual operation, automatically scans and imports sites and configurations, supports append/overwrite modes
nginx-migrate command-line tool	No browser access to the console on the server, accustomed to command-line operations	Automatically migrates configuration, certificates, static resources, log paths, supports preview mode (some installation packages do not include this tool, refer to the actual installation package <code>utils/</code> directory)
Manual Migration	Simple configuration, or need fine-grained control over migration content	Directly copy <code>nginx.conf</code> as <code>alb.conf</code> , adjust paths item by item

8 Pre-Migration Evaluation and Inventory

Please complete the following inventory work before migration and output an inventory list:

1. Confirm Nginx Version and Compilation Parameters

```
nginx -v      # View version, confirm within the 1.10 ~ 1.30
               compatibility range
nginx -V      # View compilation options and built-in/third-
               party modules
```

2. Export Complete Effective Configuration

```
nginx -T > nginx-full-config-backup.conf    # Export main
configuration and all include sub-configurations
```

3. Inventory Configuration Dependencies, record item by item:

Inventory Item	Inventory Content	Investigation Command/Method Example
Configuration Files	Location of main configuration and all include sub-configuration files	<code>nginx -T</code> , check <code>include</code> directive
Third-Party Modules	<code>load_module</code> dynamic modules, <code>--add-module</code> compiled modules	<code>nginx -V</code> , <code>nginx -T grep load_module</code>
SSL Certificates	Certificate/private key file paths, expiration time, whether national cryptographic	<code>grep -r ssl_certificate</code>
Static Resources	Directories pointed to by <code>root</code> , <code>alias</code> and disk usage	<code>grep -E 'root alias'</code>
Listening Ports	All <code>listen</code> ports and firewall allowance	<code>grep -r listen</code> , check firewall rules

Backend Services	upstream backend node list, current health check method	<code>grep -A5 upstream</code>
Logs	access_log/error_log paths, logrotate configuration	<code>grep -E '_log' , /etc/logrotate.d/</code>
Auto-start	systemd unit, init scripts, rc.local	<code>systemctl status nginx</code>
Scheduled Tasks	Certificate renewal, log rotation, etc. crontab	<code>crontab -l , /etc/cron.d/</code>
Monitoring Integration	stub_status collection, Prometheus exporter	Check monitoring system configuration

4. **Special Directive Investigation:** If the configuration includes `load_module` , Perl/njs scripts, Nginx Plus commercial version directives, etc., please record them separately and focus on handling them during migration.

9 Install ALB and Apply for License

1. Install ALB V2.0.6 on the target server (supports tar.gz extraction, RPM, Docker, K8s multiple deployment methods), refer to the Installation Guide for detailed steps.

```
tar -zxvf ALB-V2.0.6-SE-amd64.tar.gz -C /opt/
```

2. Obtain the local machine authorization code and apply for the license file, place the license file (`license.lic` / `license.xml` / `acls.properties`) in the installation directory:

```
cd /opt/ALB-V2.0.6-SE-amd64
./bin/alb-authcode          # Obtain the authorization code,
                             apply for the license file with the authorization code
```

3. Verify ALB starts normally:

```
./bin/start-alb.sh
./sbin/alb -v
# Output: alb version: alb/2.0.6-xxxxxxx
```

Migration period recommendation: After installing ALB, keep the default port (8080) running first, parallel with the existing Nginx, and cutover the port after the configuration migration verification is complete.

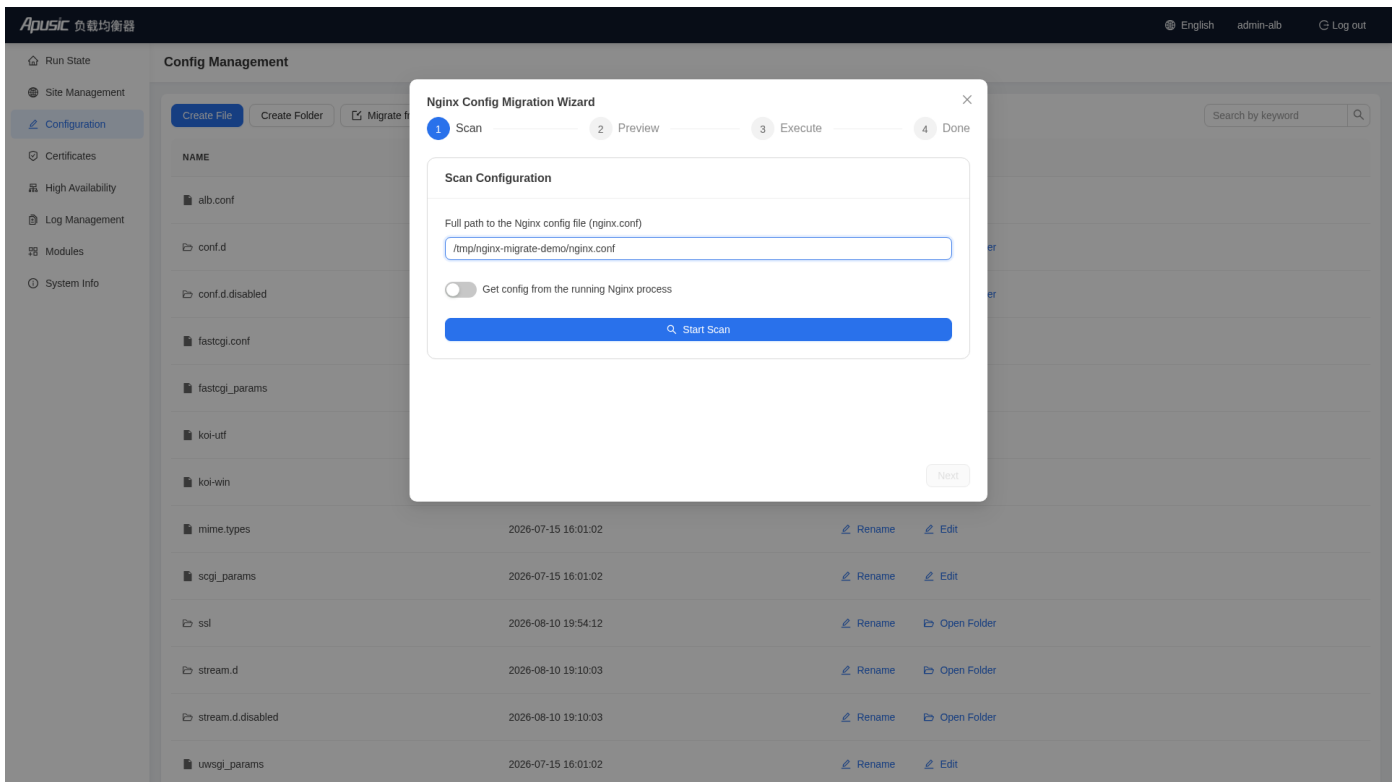
10 Migration Method 1: Console "Migrate from Nginx" Wizard (Recommended)

ALB V2.0.6 Console has a built-in Nginx migration wizard that can import existing nginx configuration files as ALB sites and configurations.

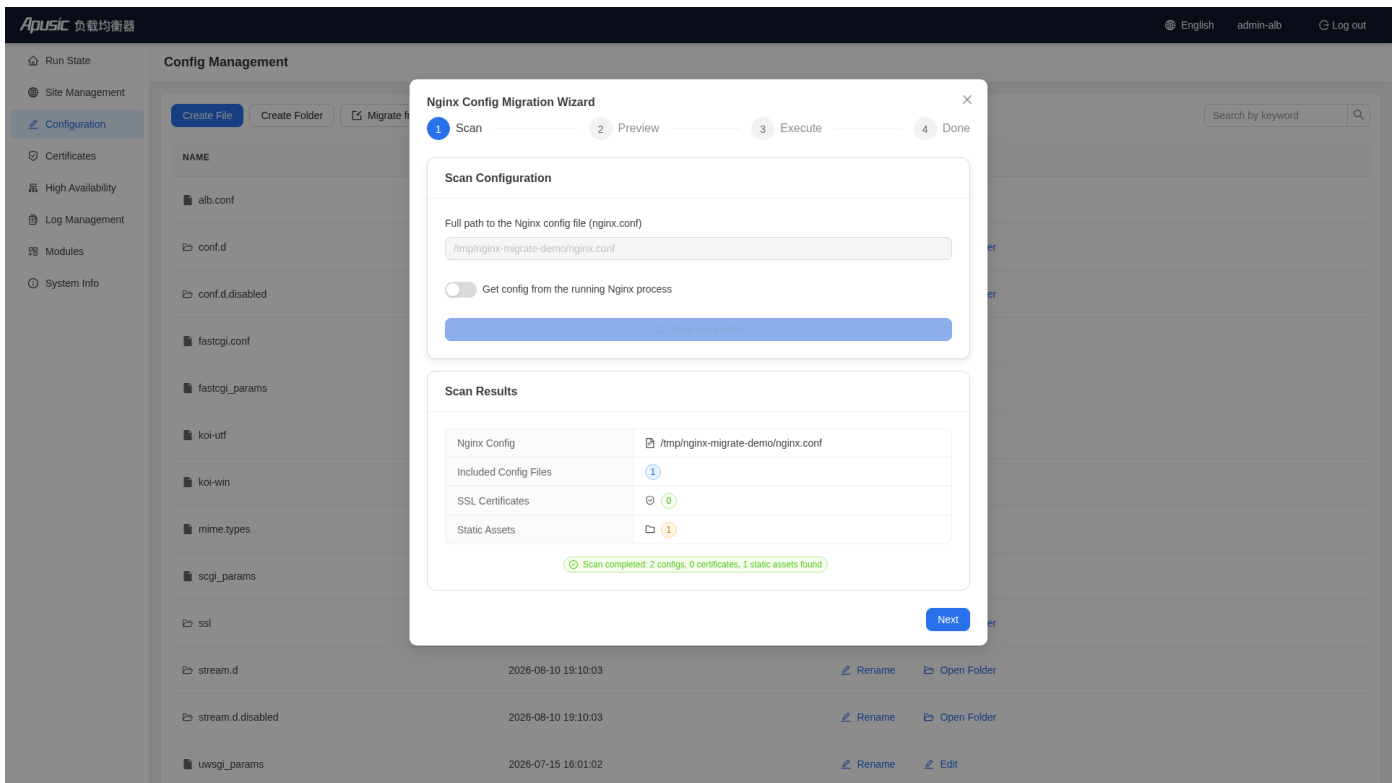
Prerequisites: ALB and Console have been started, and the Nginx configuration files are located on the ALB server (or the configuration files have been copied to that server).

Operation Steps:

1. Log in to the Console (default `http://server IP:8887`), enter the Configuration Management page, click "Migrate from Nginx", fill in the Nginx configuration file path in the dialog box, and click Scan:

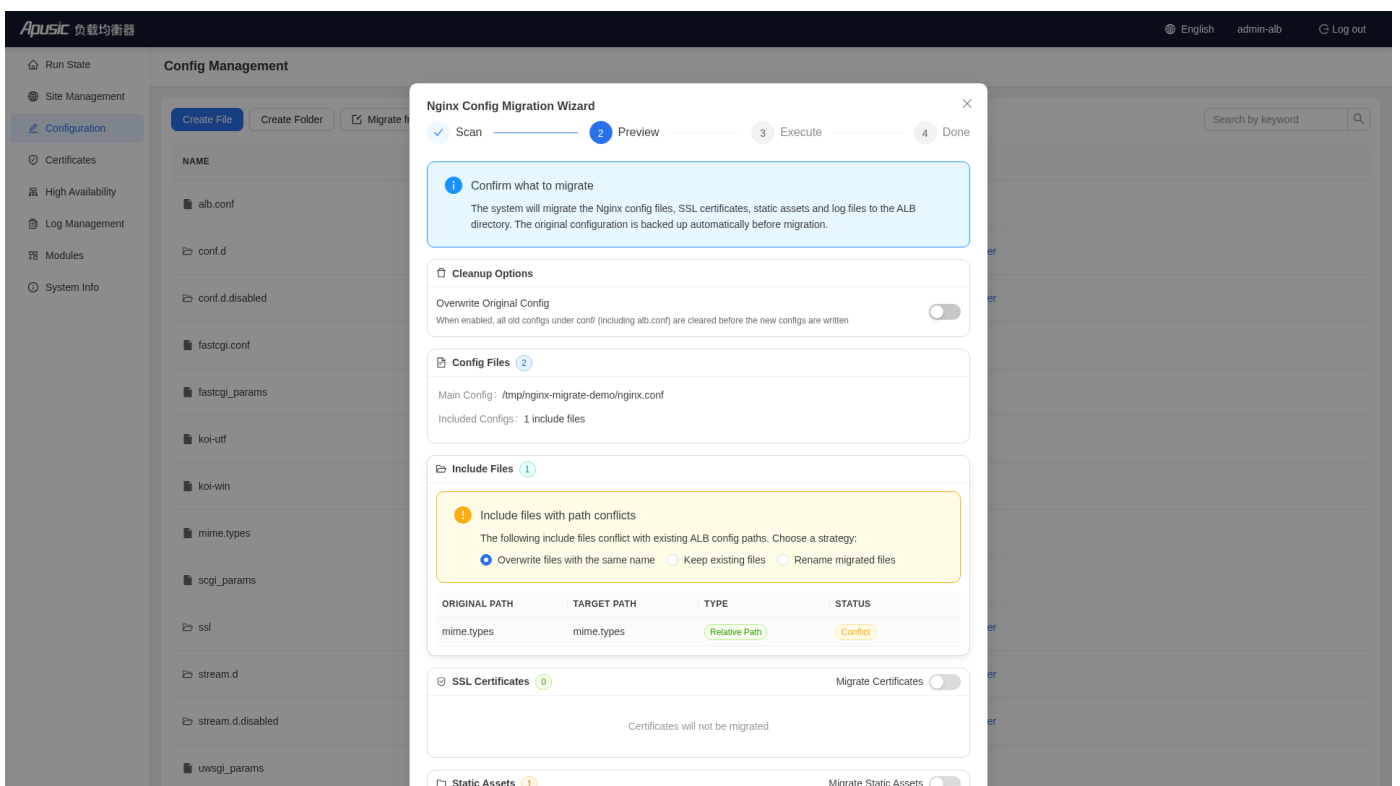


2. After the scan is successful, you can view the scan results, confirm and proceed with migration:

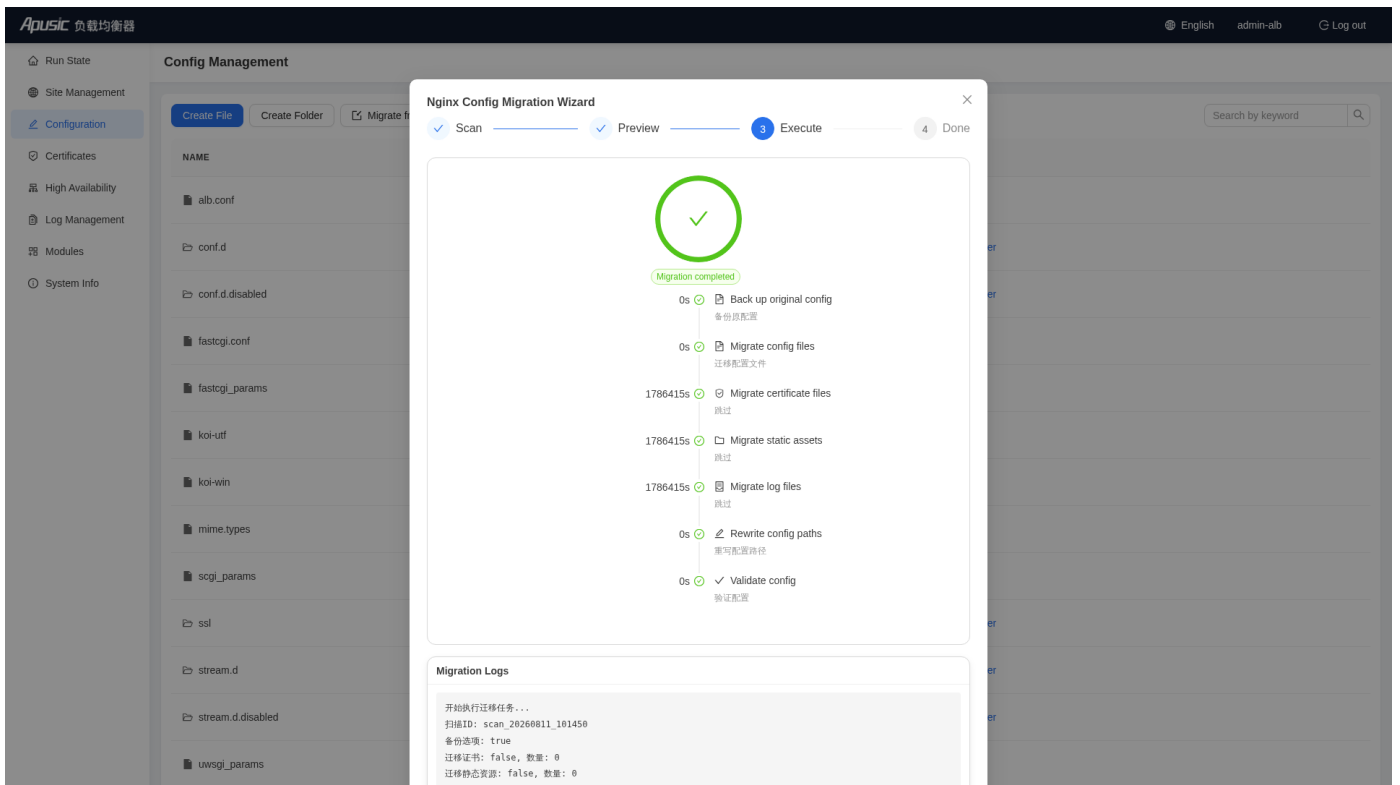


3. View the migration results after migration is complete.

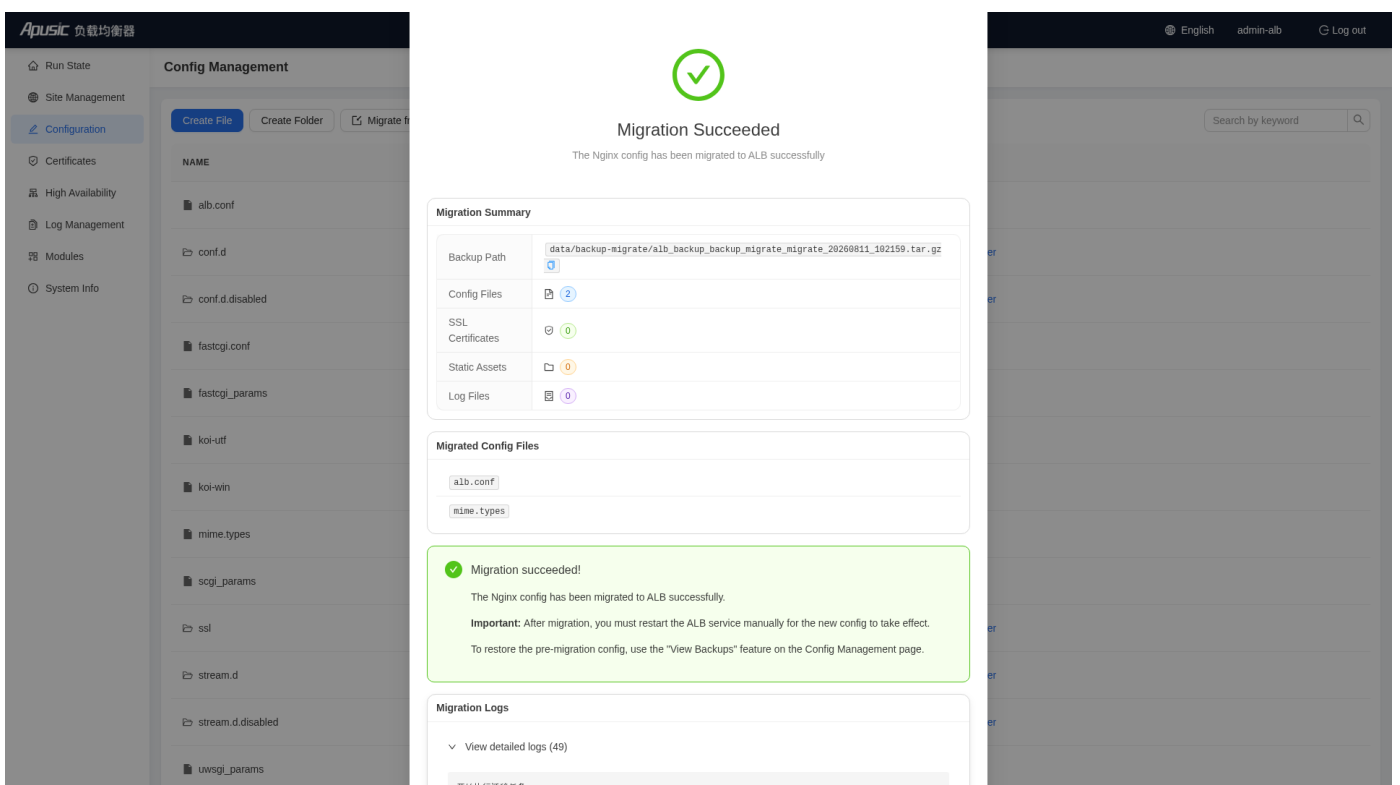
*Note: Enabling "Overwrite Original Configuration" will clear the original configuration; the default behavior is to **append** the Nginx configuration rather than overwrite it.*



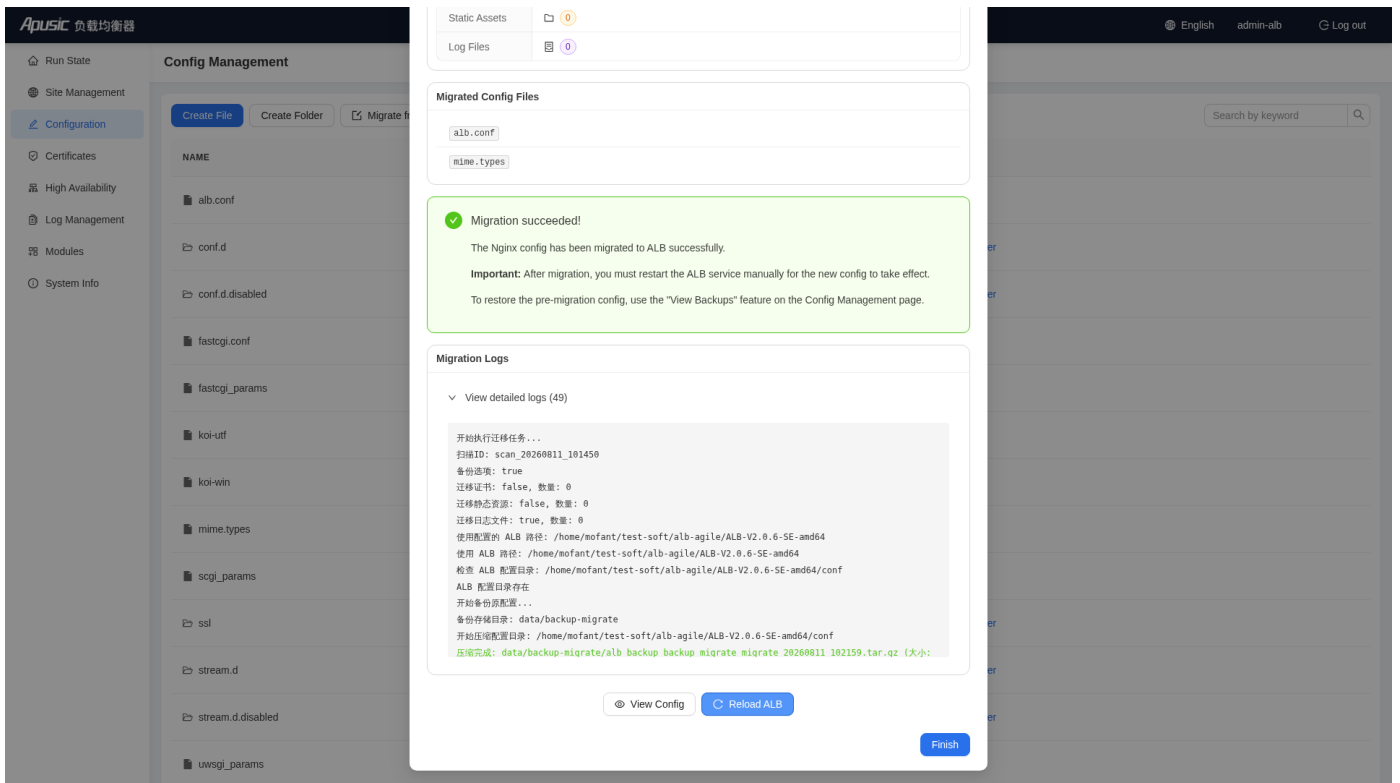
4. View migration result details:



5. View the final result:



6. After migration is complete, reload ALB to apply the configuration:



Notes:

- The site configuration imported by migration is written to the `conf/conf.d/` directory, please confirm that the main configuration `conf/alb.conf` contains `include conf.d/*.conf;`, otherwise the Console will show a yellow warning and the site configuration will not take effect.
- After migration is complete, it is recommended to use the Console "Configuration Management" backup function to create a complete backup.
- If the configuration contains directives not built into ALB, a configuration error will be prompted during reload, please handle it according to the "Post-Migration Adaptation and Adjustment" chapter.

11 Migration Method 2: nginx-migrate Command-Line Tool

Some installation packages provide a Nginx configuration migration command-line tool, located at `[ALB installation path]/utils/nginx-migrate`, used to migrate nginx configuration to ALB configuration with one click.

Note: Please refer to the actual installation package `utils/` directory. If this tool is not provided in the installation package, please use the Console migration wizard or manual migration.

11.1 Features

- **Configuration Migration:** Scans nginx main configuration file (`nginx.conf`), parses and migrates all include sub-configuration files to the `conf/conf.d/` directory, automatically updates the main configuration's include path to `conf.d/*.conf`.
- **SSL Certificate Migration:** Automatically scans `ssl_certificate`, `ssl_certificate_key`, `ssl_trusted_certificate`, copies certificate files to the `conf/certs/` directory and updates the certificate paths in the configuration.
- **Static Resource Migration:** Automatically scans `root` and `alias` directives, copies static resources to the `conf/html/` directory and updates the static resource paths in the configuration to absolute paths.
- **Log Path Migration:** Automatically migrates `error_log`, `access_log`, `pid` paths, unified into the `{ALBPath}/logs/` directory.
- **Others:** Automatically backs up original configuration before migration, interactive confirmation (can be skipped), verbose logging mode, configuration validation.

11.2 Usage Method

Prerequisites:

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

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

# Specify nginx configuration and ALB directory
```

```
./nginx-migrate --nginx-path /path/to/nginx.conf --alb-path /path/to/ALB
```

11.3 Command-Line Options

Option	Short	Description	Default Value
--nginx-path	-	nginx configuration file path	Auto-detect
--alb-path	-	ALB installation directory	Inferred from tool location
--from-running	-	Get configuration path from 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 original configuration before migration	true

11.4 Usage Examples

```
# Get configuration from 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
```

11.5 Post-Migration Directory Structure

```
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
    └── alb.pid
```

11.6 Notes

- **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` for preview first.
- **Backup:** The original configuration will be automatically backed up to `conf.backup_YYYYMMDD_HHMMSS` before migration; to restore, delete the new configuration and rename the backup directory to `conf`.
- **Permissions:** Ensure you have permission to read nginx configuration and certificate files, and permission to write to the ALB directory.
- **Verification:** After migration is complete, the tool will automatically validate the configuration syntax. It is recommended to manually run `./sbin/alb -t` once more before restarting ALB.

12 Migration Method 3: Manual Migration

12.1 Simple Direct Use of nginx.conf File

1. Back up old configuration, and copy nginx.conf as 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 actual situation
chmod 644 /opt/ALB-V2.0.6-SE-amd64/conf/alb.conf
```

2. Adjust include path: If the original nginx.conf uses the `include` directive to reference sub-configuration files (such as `conf.d/*.conf`), you need to ensure these paths are valid in the ALB installation directory. It is recommended to copy the sub-configuration files to the `conf/conf.d/` directory:

```
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, update the `ssl_certificate` and `ssl_certificate_key` paths to the new absolute paths.

4. Adjust log and pid paths:

```
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. Validate configuration and start ALB:

```
/opt/ALB-V2.0.6-SE-amd64/sbin/alb -t -c /opt/ALB-V2.0.6-SE-  
amd64/conf/alb.conf  
  
# Validation success prompts: syntax is ok and test is  
successful  
  
# Stop Nginx (modify Nginx installation path according to  
actual situation)  
/path/to/sbin/nginx -s stop  
  
# Start ALB  
/opt/ALB-V2.0.6-SE-amd64/bin/start-alb.sh
```

13 Post-Migration Adaptation and Adjustment

After configuration migration is complete, please check the following adaptation items one by one:

13.1 Ports and Listening

- The ALB factory default configuration listens on port 8080. When migrating, follow the `listen` from the original nginx configuration.
- If you need to run ALB as a regular user and listen on privileged ports such as 80/443, you need to execute first (root user):

```
setcap cap_net_bind_service=+eip ./sbin/alb
```

- Confirm that the firewall/security group has allowed the ALB listening port and the Console port (default 8887).

13.2 Users and Permissions

- Confirm that the user/group specified by the `user` directive in the configuration exists on the system (default `nobody`). If startup reports `getgrnam("nobody") failed`, please create the corresponding user group or modify the `user` directive, refer to the Operations Manual FAQ.

13.3 Path Verification Checklist

- Whether the `access_log`, `error_log`, `pid` paths point to the ALB installation directory `logs/`.
- Whether the `ssl_certificate`, `ssl_certificate_key`, `ssl_trusted_certificate` certificate paths are valid.
- Whether the `root`, `alias` static resource paths are valid, and whether the directory permissions are correct.
- Whether cache paths such as `proxy_cache_path` are valid.
- Whether all sub-configuration files referenced by `include` are in place (relative paths are resolved based on the ALB installation directory).

13.4 National Cryptographic Certificates

- If national cryptographic SSL is enabled after migration, both the signing certificate (sig) and the encryption certificate (enc) need to be configured, and the **sig certificate must be configured before the**

enc certificate:

```
ssl_certificate      /path/to/your_domain_name.sig.crt.pem;    #
Signing certificate (configure first)
ssl_certificate_key  /path/to/your_domain_name.sig.key.pem;
ssl_certificate      /path/to/your_domain_name.enc.crt.pem;    #
Encryption certificate (configure after)
ssl_certificate_key  /path/to/your_domain_name.enc.key.pem;
```

- You can also upload national cryptographic certificates through the Console "Certificate Management" and bind them to sites.

13.5 Third-Party Modules

- The `.so` dynamic modules loaded by the original Nginx via `load_module` cannot be used directly in ALB. You need to use module files built for ALB, and upload and enable them through the Console "Third-party Module Management".
- The `load_module` lines remaining in the configuration that do not correspond to a valid module need to be commented out first, otherwise the `alb -t` validation will fail.

13.6 Service Start/Stop and Boot Auto-Start

- Disable and remove the original nginx auto-start:

```
systemctl stop nginx && systemctl disable nginx
```

- Register ALB boot auto-start:

```
cd /opt/ALB-V2.0.6-SE-amd64
./utils/systemd/enable_startup.sh    # Automatically set the
boot startup item (usually requires root permissions)
systemctl start alb
```

13.7 Log Rotation

- The original nginx logrotate configuration does not apply to ALB log paths, please use ALB's built-in log rotation tool (`utils/logrotate`), which supports scheduled rotation and backup, refer to the Log Rotation chapter in the Operations Manual. The Console "Log Management" also supports scheduled rotation and backup of log files according to the configuration.

13.8 Monitoring Integration

- The original `stub_status` -based collection can be directly continued.
- It is recommended to upgrade to vts monitoring for more comprehensive traffic and status indicators, and supports direct Prometheus integration:

```
http {
    vhost_traffic_status_zone;                # Enable vts module
    server {
        listen 8080;
        location /status {                  # Configure the URI
            address for vts viewing
            vhost_traffic_status_display;
            vhost_traffic_status_display_format html;
        }
    }
}
```

- Monitoring page: `http://IP:Port/status`
- JSON data: `http://IP:Port/status/format/json`
- Prometheus: `http://IP:Port/status/format/prometheus`

13.9 Optional Enhancements (Enable as Needed After Migration)

- Active Health Check: Upgrade from the original passive `max_fails/fail_timeout` to `check` active probing:

```
upstream backend {
    server 192.168.0.1:80;
    server 192.168.0.2:80;
```

```

    check interval=3000 rise=2 fall=3 timeout=2000 type=http;
    check_http_send "GET /healthcheck HTTP/1.0\r\nHost:
www.example.com\r\n\r\n";
    check_http_expect_alive http_2xx http_3xx;
}

```

- **Enhanced Session Persistence:** Use the built-in sticky module instead of ip_hash to avoid uneven sessions in NAT scenarios:

```

upstream backend {
    sticky cookie srv_id expires=1h domain=.example.com path=/
    httponly secure;
    server 10.0.0.10:8080;
    server 10.0.0.11:8080;
}

```

- **Console Management:** Manage sites, certificates, high availability and configuration backups through the Console to improve operations efficiency.

14 Verification and Cutover

14.1 Configuration Verification

```
cd /opt/ALB-V2.0.6-SE-amd64
./sbin/alb -t          # Validate whether the conf/alb.conf
configuration has syntax errors
```

14.2 Parallel Verification (Recommended)

Before cutover, let ALB and Nginx run in parallel (ALB uses a temporary port), compare and verify one by one:

```
# Compare response status codes, response bodies, key response
headers
curl -s -o /dev/null -w "%{http_code}\n" http://nginx-
address/business-path
curl -s -o /dev/null -w "%{http_code}\n" -H "Host: business-domain"
http://alb-temporary-port/business-path
```

Functional Verification Checklist:

- Static resource access (pages, images, CSS/JS)
- Reverse proxy backend business (including POST requests, large file uploads)
- HTTPS sites (certificates, protocol versions, HTTP/2)
- WebSocket long connections (if any)
- Load balancing and session persistence (if any)
- Security policies such as rate limiting and access control (if any)
- Logs are output normally, and the format meets expectations

If you need performance comparison verification, refer to the Performance Optimization Guide.

14.3 Business Cutover

It is recommended to perform cutover during business low-peak hours:

1. Modify the `listen` in the ALB configuration to the formal business port (80/443, etc.), and execute `./sbin/alb -t` to verify.

2. Stop the Nginx service: `systemctl stop nginx` (Or `nginx -s quit`).
3. Reload or start ALB: `./bin/reload-alb.sh` (Or `./bin/start-alb.sh`).
4. Immediately verify the core business access, and observe whether there are any anomalies in `logs/error.log`.
5. Enter the observation period (recommended not less than one business peak cycle), during which the Nginx environment and configuration are kept to ensure that you can roll back at any time.

If there are traffic entrances such as DNS, LVS, and cloud load balancing at the front end, you can also adopt the canary cutover method of "add ALB nodes → gradually switch traffic → take Nginx nodes offline" to further reduce risk.

15 Rollback Plan

- **Configuration-Level Rollback:** For ALB configuration issues, you can roll back to any historical backup through the Console "Configuration Management → View Backup" (backups are stored in `[ALB installation directory]/backup/`, default retains the most recent 30), rollback does not interrupt service, only triggers configuration reload; command-line method can restore the `conf.backup_YYYYMMDD_HHMMSS` directory automatically backed up by the migration tool.
- **Service-Level Rollback:** If you need to roll back to Nginx entirely during the observation period, stop ALB and directly restart the original Nginx (the original Nginx binary and configuration have not been modified):

```
cd /opt/ALB-V2.0.6-SE-amd64 && ./bin/stop-alb.sh
systemctl start nginx      # or /path/to/sbin/nginx
```

15.0.1 FAQ

15.1 After Migration, Executing `alb -t` Reports unknown directive Error

- Cause: The configuration contains module directives not built into ALB (such as third-party module directives from the original Nginx).
- Solution: Compare with the "Built-in Module List" in this document to confirm module support; third-party modules need to use the version built for ALB and upload and enable them through the Console "Third-party Module Management"; for temporary handling, you can first comment out the relevant configuration lines. If you cannot confirm, please contact technical support.

15.2 After Migration, `load_module` Loading Failed

- Cause: The `.so` dynamic modules compiled for Nginx are incompatible with the ALB binary.
- Solution: Remove or comment out the `load_module` configuration line, and contact technical support to obtain the ALB-version module files.

15.3 Startup Failed: `getgrnam("nobody")` failed

- Cause: The user group specified by the `user` directive in the configuration does not exist in the system.

- Solution: Create the corresponding user/group, or modify the `user` directive to an existing system user, see the Operations Manual FAQ for details.

15.4 After Migration, Accessing the Page Returns a 403 Error

- Cause: Insufficient permissions on the static resource directory, missing index files in the directory, or the running user specified by the `user` directive does not have permission to read the resources.
- Solution: Check the directory permissions and the `root/alias` paths, see the Operations Manual FAQ for details.

15.5 502 Error: no live upstreams while connecting to upstream

- Cause: All upstream backend nodes are unavailable, or the health check removes all nodes.
- Solution: Confirm that the backend services are reachable; if the `check` active health check is configured, confirm that the health check parameters (URL, expected status code) match the backend reality, see the Operations Manual FAQ for details.

15.6 Regular User Starting ALB Cannot Listen on Port 80

- Cause: Privileged ports such as 80/443 can only be bound by root by default.
- Solution: After the root user executes `setcap cap_net_bind_service=+eip ./sbin/alb`, regular users can listen on port 80.

15.7 After Migration, Security Scanning or Front-End Validation Anomalies

- Cause: The `server` identifier in the ALB response header is `alb`, which is different from Nginx's `nginx`.
- Solution: Adjust the validation/scanning rules that depend on this string; if you need to hide version information, you can configure `server_tokens off;`.

15.8 Configuration Imported by the Console Migration Wizard Does Not Take Effect

- Cause: The main configuration `alb.conf` does not contain `include conf.d/*.conf;`.
- Solution: Add this include directive to the `http` block in `alb.conf` and reload; the Console page will display a yellow warning prompt for this situation.

15.8.1 Appendix: Nginx → ALB Common Operations Quick Reference

Operation	Nginx	ALB V2.0.6
View Version	<code>nginx -v</code>	<code>./sbin/alb -v</code>
View Compilation Parameters	<code>nginx -V</code>	<code>./sbin/alb -V</code>
Validate Configuration	<code>nginx -t</code>	<code>./sbin/alb -t</code>
Validate Specified Configuration	<code>nginx -t -c /path/nginx.conf</code>	<code>./sbin/alb -t -c conf/alb.conf</code>
Start	<code>nginx</code> Or <code>systemctl start nginx</code>	<code>./bin/start-alb.sh</code> (with Console: <code>./bin/start-alb-and-console.sh</code>)
Stop	<code>nginx -s quit</code> Or <code>systemctl stop nginx</code>	<code>./bin/stop-alb.sh</code> (with Console: <code>./bin/stop-alb-and-console.sh</code>)
Smooth Reload	<code>nginx -s reload</code>	<code>./bin/reload-alb.sh</code> (Or <code>kill -HUP \$(cat logs/alb.pid)</code>)
Smooth Binary Upgrade	<code>kill -USR2 + kill -WINCH/QUIT</code>	Same-name signals fully compatible, refer to the Migration Guide
Main Configuration File	<code>conf/nginx.conf</code>	<code>conf/alb.conf</code>
pid File	<code>logs/nginx.pid</code>	<code>logs/alb.pid</code>
Boot Auto-Start	Self-written systemd unit	<code>./utils/systemd/enable_startup.sh</code>
Log Rotation	<code>logrotate</code>	<code>utils/logrotate</code> tool or Console log management
Runtime Status	<code>stub_status</code>	<code>stub_status</code> compatible, also supports <code>vtls /status</code> (page/JSON/Prometheus)
Visual Management	None	Console <code>http://IP:8887</code>

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

