



Operations Manual

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 Log Rotation
 - 2.1 Tool Structure Description
 - 2.2 Example Configuration
 - 2.3 Usage
- 3 Startup Items and System Services
 - 3.1 Remove System Startup Item
- 4 Starting ALB as a Regular User and Listening on Port 80
- 5 ALB Proxy Security Configuration
 - 5.1 General Security Configuration
 - 5.1.1 Hide ALB Version Information
 - 5.1.2 Restrict Request Methods
 - 5.1.3 Limit Client Request Size
 - 5.1.4 Set Reasonable Timeout Values
 - 5.1.5 Enable HTTPS and Force Redirection
 - 5.2 Reverse Proxy Security Configuration
 - 5.2.1 Clean/Rewrite Request Headers
 - 5.2.2 Restrict Backend Response Headers
 - 5.2.3 Restrict Proxy Request Rate
 - 5.3 Security Configuration for Static Resource Service
 - 5.3.1 Disable Directory Listing
 - 5.3.2 Prohibit Access to Dangerous File Types
 - 5.3.3 Prevent Sensitive File Leakage
 - 5.3.4 Set Secure Content Security Policy (CSP) and Other Security Headers
- 6 Product Frequently Asked Questions
 - 6.1 Startup Failure: getgrnam("nobody") failed
 - 6.2 Docker Container Startup Failure
 - 6.3 ALB Startup Hangs After Displaying License Information
 - 6.4 403 Error When Accessing Pages

- 6.5 AAS Login Failure
- 6.6 When a Node of the Backend Reverse Proxy Service Goes Down, the Service Becomes Inaccessible While Other Nodes Are Normal
- 6.7 Startup Failure Using Unified Authorization Center
- 6.8 502 Error: no live upstreams while connecting to upstream
- 6.9 Resetting Three-Role Account Passwords in the Console

1 Introduction

This document is the Operations Manual for Apusic Load Balancer (ALB) software V2.0.6 Standard Edition. It provides users with monitoring, security hardening, troubleshooting of common issues, and routine operations guidance during ALB runtime, helping users ensure the stable operation of ALB services.

1.1 Target Audience

This document is intended for IT business owners, 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 set:

No.	Manual 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 product registration.
4	ALB V2.0.6 Proxy Function Manual	Details the use, configuration, management, and supporting tools for ALB-related functions.
5	ALB V2.0.6 Console User Manual	Details the use and operation instructions for ALB console-related functions.
6	ALB V2.0.6 Developer Manual	Details ALB external API, alb-cli command-line tool, and AI Agent integration methods.
7	ALB V2.0.6 Migration Guide	Details instructions for migrating and upgrading from historical ALB versions to V2.0.6.
8	ALB V2.0.6 Operations Manual	Details ALB monitoring, operations, and security hardening instructions.
9	ALB V2.0.6 Performance Optimization Guide	Details ALB performance tuning instructions.

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 including logs and screenshots of the exception or error

2 Log Rotation

Since log files keep growing, regularly cleaning and splitting log files is very important. For log rotation, ALB supports:

- Log rotation configuration for a specific log file in the console log management
- Using the log rotation tool to manually perform the rotation operation

The following is an introduction to the ALB log rotation tool. For console-based rotation, refer to the corresponding console manual.

ALB provides a default log rotation tool that automatically generates log rotation scripts (Shell scripts or logrotate configurations) from a JSON configuration file, and automatically registers them in the system crontab to automate log rotation.

The following logs are supported:

-  Single log files (e.g., /var/log/app.log)
-  Wildcard log paths (e.g., /var/log/alb/*.log)
-  Two log rotation modes: shell or logrotate
-  Automatic compression, retention day control, process signal notification (USR1)
-  Full lifecycle management of install / list / remove tasks
-  Safe crontab auto-registration and cleanup

2.1 Tool Structure Description

Tool directory: `ALB installation directory/Utils/logrotate`

```
.
├─ logrotate-config.json      ← Configuration file (default name)
├─ shell_scripts/            ← Generated Shell script directory
│   └─ <task-name>.sh
├─ logrotate_configs/        ← Generated logrotate configuration
directory
│   └─ <task-name>
├─ installed_tasks.json      ← Metadata for installed tasks
(used for list/remove)
```

Field	Type	Required	Description
-------	------	----------	-------------

name	string	✓	Task name (used for file names and comments)
log_file	string	✓	Log path, supports wildcards (e.g., *.log); if a relative path, automatically concatenated to albBaseDir
mode	string	✓	"shell" or "logrotate"
retain_days	int	✗	Retention days, default 7 days
compress	bool	✗	Whether to compress old logs, default false
cron_schedule	string	✗	Cron expression (e.g., "0 2 * * *"), default "0 1 * * *" (every day at 1 AM)

2.2 Example Configuration

```
[
  {
    "name": "alb-access",
    "log_file": "logs/access.log",
    "mode": "shell",
    "retain_days": 14,
    "compress": true,
    "cron_schedule": "0 2 * * *"
  },
  {
    "name": "alb-error",
    "log_file": "logs/error.log",
    "mode": "logrotate",
    "retain_days": 30,
    "compress": true
  },
  {
    "name": "all-app-logs",
    "log_file": "logs/app/*.log",
    "mode": "logrotate",
    "retain_days": 7,
    "compress": false
  }
]
```

```
}
]
```

✓ *Relative path logs/access.log is automatically converted to an absolute path: /path/to/alb/logs/access.log*

✓ *Wildcard logs/app/*.log is correctly handled in both logrotate and shell modes*

2.3 Usage

0. Permission Requirements

- Requires read/write permission to the log directory
- Must be able to execute `crontab -l` and `crontab -e` (typically any regular user is fine)
- If logs are written by root, it is recommended to run this tool as the same user
- If executable permission cannot be guaranteed, switch to the root user to run this tool
- Switch to the utils/logrotate directory before executing

1. Install Task (Generate Script + Register crontab)

```
# Switch to the utils/logrotate directory (required)
cd utils/logrotate
# Install log rotation tasks using the default configuration file
./logrotate-gen --install
```

After execution:

- Shell scripts are generated to `./shell_scripts/`
- logrotate configurations are generated to `./logrotate_configs/`
- Tasks are automatically added to the current user's crontab
- Installation records are saved to `installed_tasks.json`

2. Preview Generated Content (Do Not Write Files)

```
# Preview all generated content
./logrotate-gen --dry-run

# Preview a specific configuration, provided test.json exists in the
current directory
./logrotate-gen --dry-run --config ./test.json
```

Example Output

```

=== SHELL SCRIPT: ./shell_scripts/alb-access.sh ===
#!/bin/bash
set -euo pipefail
shopt -s nullglob
...

=== LOGROTATE CONFIG: ./logrotate_configs/alb-error ===
# Auto-generated for task: alb-error
/path/to/alb/logs/error.log {
    daily
    rotate 30
    compress
    ...
}

```

3. List Installed Tasks

```
./logrotate-gen -list
```

Output example:

```

 Installed tasks (installed at: Mon Oct 27 10:00:00 CST 2025):
1. Name: alb-access
   Mode: shell
   Log File: /path/to/alb/logs/access.log
   Schedule: 0 2 * * *
   Retain Days: 14
   Compress: true

2. Name: alb-error
   Mode: logrotate
   Log File: /path/to/alb/logs/error.log
   Schedule: 0 1 * * * (default)

```

```
Retain Days: 30
```

```
Compress: true
```

 Current crontab entries for logrotate:

```
0 2 * * * /path/to/tool/shell_scripts/alb-access.sh # alb-access
0 1 * * * logrotate /path/to/tool/logrotate_configs/alb-error # alb-
error
```

4. Uninstall All Tasks

```
./logrotate-gen -remove-all
```

After execution:

- Remove all entries added by this tool from crontab
- Delete the shell_scripts/ and logrotate_configs/ directories
- Delete installed_tasks.json

 *This operation is irreversible. Please use with caution!*

3 Startup Items and System Services

ALB does not start automatically on boot by default. To enable auto-start on boot, switch to the installation directory and execute (the following commands usually require root privileges)

```
cd /opt/ALB-V2.0.6-SE-amd64          # Enter the installation
directory (use the actual installation path)
./utils/systemd/enable_startup.sh    # Automatically set the
startup item
systemctl start alb                  # Start ALB
```

3.1 Remove System Startup Item

```
cd /opt/ALB-V2.0.6-SE-amd64          # Enter the installation
directory (use the actual installation path)
./utils/systemd/remove_startup.sh    # Remove the startup item
```

4 Starting ALB as a Regular User and Listening on Port 80

ALB supports modifying program capabilities so that a regular user can listen on port 80. The specific operations are as follows:

1. Switch to the **root user** and navigate to the ALB installation directory.
2. Grant permission to listen on port 80: `setcap cap_net_bind_service=+ep ./sbin/alb`
3. Switch back to the **regular user**
4. Start ALB: `./sbin/alb` (or `./bin/start-alb.sh` . If ALB is already running, execute `./bin/stop-alb.sh` first to stop it before starting)

5 ALB Proxy Security Configuration

5.1 General Security Configuration

5.1.1 Hide ALB Version Information

```
http {
    server_tokens off; # Add this line in the http block
}
```

5.1.2 Restrict Request Methods

Only allow necessary HTTP methods (such as GET, POST, HEAD, PUT):

```
http {
    server {
        location / {
            if ($request_method !~ ^(GET|HEAD|POST|PUT)$) {
                return 405;
            }
        }
    }
}
```

5.1.3 Limit Client Request Size

Prevent DoS attacks (e.g., large file uploads, overly long URLs):

```
http {
    client_max_body_size 1M; # Limit request body, adjust
according to actual business
    client_header_buffer_size 1k; # Request header buffer
    large_client_header_buffers 4 8k;
}
```

5.1.4 Set Reasonable Timeout Values

Prevent slow attacks (such as Slowloris)

```
http {
    client_body_timeout 10s;
    client_header_timeout 10s;
    keepalive_timeout 60s;
    send_timeout 10s;
    lingering_timeout 5s;
    lingering_close on;
}
```

5.1.5 Enable HTTPS and Force Redirection

```
server {
    listen 80;
    server_name example.com;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl http2;
    ssl_certificate /path/to/cert.pem;
    ssl_certificate_key /path/to/privkey.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers
ECDHE+AESGCM:DHE+AESGCM:AES256+EECDH:AES256+EDH:!aNULL:!MD5:!DSS;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 1d;
    ssl_session_tickets off;

    # HSTS configuration (recommended to test first on initial
    deployment, then enable after confirming correctness)
```

```
add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains" always;
}
```

5.2 Reverse Proxy Security Configuration

5.2.1 Clean/Rewrite Request Headers

Prevent header injection or information leakage:

```
http {
    server {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # Clear internal headers that the client may forge (setting
        to empty string does not fully prevent header forwarding; it is
        recommended to add verification on the backend)
        proxy_set_header X-Original-URI "";
        proxy_set_header X-Internal-Secret "";
    }
}
```

5.2.2 Restrict Backend Response Headers

Avoid leaking sensitive information from the backend (such as Server, X-Powered-By): Can be hidden via proxy_hide_header:

```
http {
    server {
        proxy_hide_header Server;
        proxy_hide_header X-Powered-By;
        proxy_hide_header X-AspNet-Version;
```

```
}
}
```

5.2.3 Restrict Proxy Request Rate

Defend against CC attacks

```
http {
    limit_req_zone $binary_remote_addr zone=api:10m rate=10r/s;

    server {
        location /api/ {
            limit_req zone=api burst=20 nodelay;
            proxy_pass http://backend;
        }
    }
}
```

5.3 Security Configuration for Static Resource Service

5.3.1 Disable Directory Listing

Ensure `autoindex on;` is not configured; it is off by default.

5.3.2 Prohibit Access to Dangerous File Types

Configure according to the static resource type, prohibit dangerous extensions (e.g., .php, .pl, .py, .jsp, .asp, .sh, .cgi):

```
location ~ \.(php|pl|py|jsp|asp|sh|cgi)$ {
    return 403;
}
```

5.3.3 Prevent Sensitive File Leakage

```
http {
    server {
```

```

        # Prohibit access to hidden directories starting with a dot
        (e.g., .git, .svn)
        location ~ /\. {
            deny all;
        }

        # Prohibit access to .env files
        location ~ /\.env {
            deny all;
        }
    }
}

```

5.3.4 Set Secure Content Security Policy (CSP) and Other Security Headers

```

server {
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-Frame-Options "DENY" always;
    add_header Referrer-Policy "strict-origin-when-cross-origin"
always;
    add_header Permissions-Policy "geolocation=(), microphone=(),
camera=()" always;

    # HSTS configuration (recommended to test first on initial
deployment, then enable after confirming correctness)
    add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains" always;

    # CSP example (adjust according to actual situation; production
environments should avoid using 'unsafe-inline')
    add_header Content-Security-Policy "default-src 'self'; script-
src 'self' 'unsafe-inline' https://trusted.cdn.com; style-src 'self'
'unsafe-inline'; img-src 'self' data;; font-src 'self'; object-src
'none'; base-uri 'self'; form-action 'self';" always;
}

```

6 Product Frequently Asked Questions

Note: For performance-related errors, refer to the [Performance Optimization] manual.

6.1 Startup Failure: getgrnam("nobody") failed

The following error occurs during startup:

```
root@t16:/home/lyan/桌面/temp/alb/alb-agile-2.0-x86/bin# ./start-alb.sh
Starting ALB...
alb: [emerg] : getgrnam("nobody") failed
Start ALB failed, please check logs/error.log
root@t16:/home/lyan/桌面/temp/alb/alb-agile-2.0-x86/bin# ls
reload-alb.sh start-alb.sh start-all.sh stop-alb.sh stop-all.sh
```

Solution:

Modify the user directive in the conf/alb.conf file, change `user nobody;` to the username that has access permission to the corresponding static resources (recommended to create a dedicated user such as `alb` and grant permissions, rather than directly using root):

```
user alb;
```

6.2 Docker Container Startup Failure

After Docker container startup fails, use `docker logs <container_id>` to view the container logs.

1. Get the ALB container ID

```
docker ps -a
```

CONTAINER ID	IMAGE	COMMAND
xxxxxxxxxxxx	harbor.apusic.com/apusic/alb:V2.0.6-ae.20260421-	
kylin-amd64	"bash /opt/ALB-V2...."	17 seconds ago Exited

```
(1) 16 seconds ago
alb-1
```

```
alb-v206-se-docker-amd64-
```

The ALB container ID above is `xxxxxxxxxxxx`

2. View container logs

Use docker logs to view logs, as shown in the following example:

```
# docker logs 48c60b6fced6
Starting ALB...
alb: [emerg] : open() "/opt/ALB-V2.0.6-SE-amd64/conf/alb.conf"
failed (13: Permission denied)
Start ALB failed, please check logs/error.log
```

3. Docker Startup Permission Denied Error

Cause: The process inside the container does not have sufficient permissions to access files and directories in the mounted volume.

Solution:

- Modify `alb-standard-dockercompose.yaml`, mark the mount paths with `:z` (the `:z` flag only applies to SELinux environments; non-SELinux systems do not need to add it)

```
services:
  alb:
    image: harbor.apusic.com/apusic/alb:V2.0.6-ae.20260421-kylin-
amd64
    ports:
      - 8080:8080
      - 8887:8887
    volumes:
      - ./alb/alb.conf:/opt/ALB-V2.0.6-SE/conf/alb.conf:z
      - ./alb/license.xml:/opt/ALB-V2.0.6-SE/license.xml:z
      - ./alb/logs:/opt/ALB-V2.0.6-SE/logs:z
    command: bash /opt/ALB-V2.0.6-SE/bin/start-alb-and-console.sh
```

- ```
docker compose -f alb-standard-dockercompose.yaml up -d
```

After executing ALB startup, the terminal displays the license information and then hangs without automatically exiting the startup script, as shown below:

```
Starting ALB...
```

```
version: V2.0.6
```

- Symptom: After executing the startup script, the script does not exit, and no prompt such as `ALB is running now!` is output
- Cause Analysis: ALB is started in the foreground instead of the background
- Solution: In the `conf/alb.conf` configuration file, check whether the `daemon off;` configuration exists. If so, comment it out ( `#daemon off;` ), then restart ALB.

```
./bin/stop-alb.sh && ./bin/start-alb.sh
```

## 19

After ALB is configured, accessing pages shows a 403 error page, as shown below:

# 403 Forbidden

---

alb/2.0

- Symptom: Any page (static resource) access returns a 403 error.
- Cause Analysis: The user starting ALB does not have read permission for the corresponding static resources, resulting in a 403 error.
- Solution: The preferred solution is to modify the static resource directory permissions ( `chown -R nobody:nobody /path/to/static` ) so that the ALB startup user can access the static resources. The alternative solution is to modify the `user` directive in the `conf/alb.conf` configuration file to the username that has access permission to the corresponding static resources (recommended to create a dedicated user such as `alb` ), then restart ALB.

## 6.5 AAS Login Failure

- Symptom: When using ALB to proxy multiple AAS services, after entering the username and password and clicking the login page, login fails or redirects to the login page.
- Cause Analysis:
  - Inconsistent sessions: When the browser accesses, different requests are sent to different AAS nodes, while the login session only exists in one of the AAS servers.
  - Request headers or cookies lost: The services in the AAS server rely on request headers or cookies to verify whether the client is logged in. If the request headers or cookies are lost, it cannot verify whether the client is logged in.
  - WebSocket connection failure: Some services establish WebSocket connections after login. If the WebSocket connection fails, real-time functionality will not be available.
- Solution:
  - Inconsistent sessions: Use the `ip_hash` or sticky load balancing algorithm in ALB.
  - Request headers or cookies lost: Configure to forward all request headers and cookies

```
location / {

 proxy_pass http://backend/;

 # The following content is the newly added
 configuration

 proxy_cookie_path / /; # Ensure Cookie path is correct
 proxy_pass_header Set-Cookie;

 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto $scheme;

}
```

- WebSocket connection failure: Write a separate location configuration for the WebSocket connection address

```
location /websocket_url {
 proxy_pass http://backend/websocket_url;

 # WebSocket required configuration
 proxy_http_version 1.1;
 proxy_set_header Upgrade $http_upgrade;
 proxy_set_header Connection "upgrade";

 # Set WebSocket connection read timeout to 10
 minutes, modify time according to business
 proxy_read_timeout 600s;
```

```
Set WebSocket connection write timeout to 10
minutes, modify time according to business

proxy_send_timeout 600s;

proxy_cookie_path / /; # Ensure Cookie path is
correct

proxy_pass_header Set-Cookie;

proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
}
```

## 6.6 When a Node of the Backend Reverse Proxy Service Goes Down, the Service Becomes Inaccessible While Other Nodes Are Normal

- Symptom: For ALB reverse proxy services A, B, C, where A is down or crashed but B and C services are normal, accessing ALB at this time results in short-term inaccessibility.
- Cause Analysis: ALB's connection to the already-down A service times out, causing the request to wait too long. ALB's default passive health check mechanism marks a node as unavailable after `max_fails` failures, but the default timeout is relatively long, during which new requests may still be forwarded to the failed node. It is recommended to enable active health checks so that ALB can sense the node status in advance and stop forwarding requests to the down service.
- Solution: Modify the timeout for ALB to connect to upstream services, default is 60 seconds, change to 10 seconds or less, and enable active health checks so that ALB no longer forwards requests to down services. The following is a reference example

```
http {
 upstream backend {
 sticky cookie srv_id expires=1h domain=.example.com path=/
 httponly secure; # Use session affinity load balancing algorithm
 (choose one with ip_hash, cannot be used together)
 }
}
```

```
server 192.168.1.100:8080 max_fails=3 fail_timeout=10s;
server 192.168.1.2:8080 max_fails=3 fail_timeout=10s;
server 192.168.1.3:8080 max_fails=3 fail_timeout=10s;

Configure active health check (if ALB does not have this
module built-in, please refer to official documentation to confirm)
check timeout=1000 fall=3 rise=2 type=http interval=3000;
check_http_expect_alive http_2xx http_3xx;
}

server {
 location / {
 proxy_pass http://backend/;
 # Core configuration, shorten service connection timeout
 (recommended at least 5s, adjust according to actual high-
 concurrency scenarios)
 proxy_connect_timeout 5s;
 }
}
```

## 6.7 Startup Failure Using Unified Authorization Center

- Symptom: Startup fails using the unified authorization center method, with the following error message:

```
alb: [emerg]: center auth: request to center fault, err: send auth
info to authserver error
```

- Cause Analysis: Authorization center update causes ALB request parsing to fail
- Solution: Update to the ALB released in 2025.

## 6.8 502 Error: no live upstreams while connecting to upstream

- Symptom: The frontend receives 502 errors, and the ALB error log contains a large number of `no live upstreams while connecting to upstream`

- Cause Analysis: In the ALB reverse proxy, if the upstream node does not respond within the time specified by `proxy_connect_timeout`, `proxy_send_timeout`, `proxy_read_timeout`, the node will be marked as unavailable. The default value of `max_fails` is 1, used together with `fail_timeout`. After the threshold is reached, ALB will temporarily mark the node as unavailable, resulting in the above error.
- Solution: Modify `proxy_connect_timeout`, `proxy_send_timeout`, `proxy_read_timeout` to a longer time (over 60s), or set `max_fails` to 3, which means the node will not be marked as unavailable.
- Configuration Example:

```
upstream gateway_api {
 # Add max_fails and fail_timeout parameters
 server 127.0.0.1:9099 max_fails=3 fail_timeout=15s;
 server 127.0.0.2:9099 max_fails=3 fail_timeout=15s;
}

server {
 location /api/ {
 proxy_pass http://gateway_api/;

 # Add timeout parameters
 proxy_connect_timeout 15s;
 proxy_send_timeout 60s;
 proxy_read_timeout 60s;
 }
}
```

## 6.9 Resetting Three-Role Account Passwords in the Console

- Symptom: All three-role accounts in the system (admin-system, admin-security, admin-audit) are unavailable, passwords forgotten or locked, unable to log in through the console interface.
- Cause Analysis: Passwords have not been changed for a long time, initial random passwords were not saved, or accounts were locked by security policies.
- Solution: Reset the password of the specified account via the server-side command-line tool. After resetting, a new random password will be output. The system will force a password change after login.

```
Switch to the console installation directory (use the actual
installation path)
```

```
cd /opt/ALB-V2.0.6-SE-amd64/console

Reset the specified account password (can reset admin-system,
admin-security, admin-audit, admin-alb, user-alb)
[root@localhost console]# ./alb-console --reinit admin-system
[alb-agile-console]2026/06/24 - 04:24:04.462 info
.../sqlite.go:64 register table success
[alb-agile-console]2026/06/24 - 04:24:04.463 info
.../sqlite.go:95 数据库中已存在用户, 跳过默认用户初始化
[alb-agile-console]2026/06/24 - 04:24:04.520 info
.../sqlite.go:204 用户密码重置成功 {"username": "admin-
system", "new_password": "%B4H%VKDf#yLj8oG"}
用户 admin-system 密码已重置为新密码: %B4H%VKDf#yLj8oG
```

- Operation Notes:
  - The `--reinit` parameter is followed by the username to be reset; any existing console account can be reset.
  - After the command is executed successfully, the terminal will output the newly generated random password. Please record it immediately and force a change after login.
  - This command only resets the password and does not reset other configuration information of the account (such as role, status).
  - To batch reset multiple accounts, execute the above command separately for each; only one account can be reset at a time.
- Notes: The new password output by this command is a system-generated strong password. Please keep it safe and immediately change it to a custom password after logging in to the console. If you still cannot log in after multiple resets, please contact Apusic technical support.

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](http://www.apusic.com)

