



維運手冊

金蝶Apusic負載均衡器

版權所有 © 深圳市金蝶天燕雲計算股份有限公司2026。保留所有權利。

版權聲明

本文件所涉及的軟件著作權、版權等知識產權已依法進行了註冊，由金蝶天燕雲計算股份有限公司合法擁有。受《中華人民共和國著作權法》《計算機軟件保護條例》《知識產權保護條例》和相關國際版權條約、法律、法規以及其它知識產權法律和條約的保護。未經授權許可，不得非法使用。

免責聲明

本文件包含的版權資訊由金蝶天燕雲計算股份有限公司合法擁有，受法律的保護，金蝶天燕雲計算股份有限公司對本文件可能涉及到的非金蝶天燕雲計算股份有限公司的資訊不承擔任何責任。在法律允許的範圍內，您可以查閱並僅能夠在《中華人民共和國著作權法》規定的合法範圍內複製和打印本文件。任何單位和個人未經金蝶天燕雲計算股份有限公司書面授權許可，不得使用、修改、再發佈本文件的任何部分和內容，否則將被視為侵權，金蝶天燕雲計算股份有限公司有依法追究其責任的權利。

本文件如有更新，不另行通知。對本文件中的問題您可向金蝶天燕雲計算股份有限公司告知或查詢。未經本公司明確授予的任何權利均予保留。

商標聲明

 是深圳市金蝶天燕雲計算股份有限公司向中華人民共和國國家商標局申請註冊的註冊商標，註冊商標專用權由金蝶天燕合法擁有，受法律保護。未經金蝶天燕的書面許可，任何單位及個人不得以任何方式或理由對該商標的任何部分進行使用、複製、修改、傳播、抄錄或與其它產品捆綁使用銷售。凡侵犯金蝶天燕商標權的，金蝶天燕將依法追究其法律責任。本文件提及的其他所有商標或註冊商標，由各自的所有人擁有。

目錄

- 1 前言
 - 1.1 適用對象
 - 1.2 相關文件
 - 1.3 技術支援
- 2 日誌切割
 - 2.1 工具結構說明
 - 2.2 範例設定
 - 2.3 使用方式
- 3 開機啟動項與系統服務
 - 3.1 移除系統啟動項
- 4 普通使用者啟動ALB並且開啟監聽80埠
- 5 ALB代理安全設定
 - 5.1 通用安全設定
 - 5.1.1 隱藏ALB版本資訊
 - 5.1.2 限制請求方法
 - 5.1.3 限制用戶端請求大小
 - 5.1.4 設定合理的逾時時間
 - 5.1.5 啟用 HTTPS 並強制跳轉
 - 5.2 反向代理安全設定
 - 5.2.1 清理/重寫請求標頭
 - 5.2.2 限制後端回應標頭
 - 5.2.3 限制代理請求速率
 - 5.3 靜態資源服務專用安全設定
 - 5.3.1 禁止目錄列表
 - 5.3.2 禁止危險檔案類型存取
 - 5.3.3 防止敏感檔案洩露
 - 5.3.4 設定安全的 Content Security Policy (CSP) 和其他安全標頭
- 6 產品常見問題
 - 6.1 啟動失敗: getgrnam("nobody") failed
 - 6.2 Docker容器啟動失敗
 - 6.3 啟動ALB時, 顯示授權資訊後, 停住了
 - 6.4 存取頁面出現403錯誤

- 6.5 AAS登入失敗
- 6.6 後端反向代理服務的一個節點當機後，服務出現無法存取，但其他節點正常
- 6.7 使用統一授權中心授權方式啟動失敗
- 6.8 502錯誤： no live upstreams while connecting to upstream
- 6.9 管控台三員帳號密碼重設

1 前言

本文件為金蝶Apusic負載均衡器軟體V2.0.6標準版的維運手冊，為使用者提供ALB執行期間的監控、安全加固、常見問題排查及日常維運操作說明，協助使用者保障ALB服務的穩定運作。

1.1 適用對象

本文件適用於IT資訊化業務負責人、研發經理、軟體專案經理、軟體架構師、維運工程師。

1.2 相關文件

了解更多ALB V2.0.6產品相關的資訊，請參閱以下ALB V2.0.6產品手冊文件集：

序號	手冊文件	說明
1	金蝶Apusic負載均衡器軟體 V2.0.6 發版說明	介紹了ALB 產品版本更新內容。
2	金蝶Apusic負載均衡器軟體 V2.0.6 快速使用手冊	簡單介紹了如何快速上手使用ALB。
3	金蝶Apusic負載均衡器軟體 V2.0.6 安裝手冊	詳細介紹如何在各作業系統上安裝ALB，以及ALB服務啟停操作，產品的註冊過程。
4	金蝶Apusic負載均衡器軟體 V2.0.6 代理功能手冊	詳細介紹 ALB 相關功能的使用、設定、管理及配套工具的使用方法。
5	金蝶Apusic負載均衡器軟體 V2.0.6 管控台使用者手冊	詳細介紹ALB管控台相關功能的使用和操作說明。
6	金蝶Apusic負載均衡器軟體 V2.0.6 開發手冊	詳細介紹ALB外部API、alb-cli命令列工具及AI Agent整合方法。
7	金蝶Apusic負載均衡器軟體 V2.0.6 遷移手冊	詳細介紹ALB歷史版本遷移升級到V2.0.6版本的說明。
8	金蝶Apusic負載均衡器軟體 V2.0.6 維運手冊	詳細介紹ALB的監控、維運、安全加固等維運說明。
9	金蝶Apusic負載均衡器軟體 V2.0.6 效能最佳化手冊	詳細介紹ALB效能調校的說明。

1.3 技術支援

ALB產品提供全面的技術支援服務，您可以透過以下方式獲得技術支援：

- 網址：www.apusic.com
- 電話：400-855-5800
- 電子郵件：support@apusic.com
- 金蝶雲社區：<https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技術支援時，請提供如下資訊：

1. 您的姓名
2. 公司資訊與聯絡方式
3. 作業系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細資訊

2 日誌切割

由於日誌檔案會不斷增長，定期清理和分割日誌檔案是非常重要的，針對日誌切割，ALB支援：

- 管控台日誌管理中，針對某個日誌檔案設定日誌切割
- 使用日誌切割工具，手工操作產生切割功能

下面是ALB日誌切割工具的介紹，如需透過管控台，可查看對應管控台的手冊。

ALB提供了預設的日誌切割工具，根據 JSON 設定檔自動產生 日誌切割腳本（Shell 腳本或 logrotate 設定），並自動註冊到系統 crontab，實現日誌輪轉自動化。

支援以下日誌：

- ☒ 單個日誌檔案（如 /var/log/app.log）
- ☒ 通配符日誌路徑（如 /var/log/alb/*.log）
- ☒ 兩種日誌輪轉模式：shell 或 logrotate
- ☒ 自動壓縮、保留天數控制、程序訊號通知（USR1）
- ☒ 安裝 / 列出 / 卸載任務全生命週期管理
- ☒ 安全的 crontab 自動註冊與清理

2.1 工具結構說明

工具目錄：`ALB安裝目錄/Utils/logrotate`

```

.
├── logrotate-config.json      ← 設定檔（預設名）
├── shell_scripts/           ← 產生的 Shell 腳本目錄
│   └── <task-name>.sh
├── logrotate_configs/       ← 產生的 logrotate 設定目錄
│   └── <task-name>
└── installed_tasks.json      ← 已安裝任務的元資料（用於 list/remove）
  
```

欄位	類型	必填	說明
name	string	☒	任務名稱（用於檔名和註解）
log_file	string	☒	日誌路徑，支援通配符（如 *.log）；若為相對路徑，會自動拼接到 albBaseDir

mode	string	✓	"shell" 或 "logrotate"
retain_days	int	✗	保留天數，預設 7 天
compress	bool	✗	是否壓縮舊日誌，預設 false
cron_schedule	string	✗	cron 運算式（如 "0 2 * * *"），預設 "0 1 * * *"（每天凌晨 1 點）

2.2 範例設定

```
[
  {
    "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
  }
]
```

✓ 相對路徑 `logs/access.log` 會被自動轉為絕對路徑： `/path/to/alb/logs/access.log` ✓ 通配符 `logs/app/*.log` 在 `logrotate` 和 `shell` 模式下均能正確處理

2.3 使用方式

0. 權限要求

- 需要對日誌目錄有讀寫權限
- 需要能執行 `crontab -l` 和 `crontab -e` (通常普通使用者即可)
- 若日誌由 root 寫入，建議以相同使用者身分執行本工具
- 若無法保證可執行權限，則切換至 root 使用者執行此工具
- 切換至 `utils/logrotate` 目錄下執行

1. 安裝任務 (產生腳本 + 註冊 crontab)

```
# 切換至utils/logrotate目錄下 (必須)
cd utils/logrotate
# 使用預設設定檔安裝日誌切割任務
./logrotate-gen --install
```

執行後：

- 產生 Shell 腳本到 `./shell_scripts/`
- 產生 logrotate 設定到 `./logrotate_configs/`
- 自動將任務加入當前使用者的 crontab
- 儲存安裝記錄到 `installed_tasks.json`

2. 預覽產生內容 (不寫檔案)

```
# 預覽所有產生內容
./logrotate-gen --dry-run

# 預覽指定設定，前提是設定檔test.json在當前目錄
./logrotate-gen --dry-run --config ./test.json
```

範例輸出

```
=== 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. 列出已安裝的任務

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

輸出範例:

```
 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. 卸載所有任務

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

執行後：

- 從 crontab 中移除所有本工具新增的條目
- 刪除 shell_scripts/ 和 logrotate_configs/ 目錄
- 刪除 installed_tasks.json

 此操作不可逆，請謹慎使用！

3 開機啟動項與系統服務

ALB預設不開啟開機自動啟動，如需開機啟動，切換到安裝目錄，執行（以下命令通常需要 root 權限）

```
cd /opt/ALB-V2.0.6-SE-amd64          # 進入安裝目錄（以實際安裝路徑為準）
./utils/systemd/enable_startup.sh     # 自動設定開機啟動項
systemctl start alb                  # 啟動ALB
```

3.1 移除系統啟動項

```
cd /opt/ALB-V2.0.6-SE-amd64          # 進入安裝目錄（以實際安裝路徑為準）
./utils/systemd/remove_startup.sh     # 移除開機啟動項
```

4 普通使用者啟動ALB並且開啟監聽80埠

ALB支援透過修改程式權限，使得在普通使用者下可以監聽80埠。具體操作如下：

1. 切換到root使用者，並切換到ALB安裝目錄。
2. 授權監聽80埠：`setcap cap_net_bind_service=+ep ./sbin/alb`
3. 切換回普通使用者
4. 啟動ALB：`./sbin/alb`（或 `./bin/start-alb.sh`，若ALB已在執行，請先執行 `./bin/stop-alb.sh` 停止後再啟動）

5 ALB代理安全設定

5.1 通用安全設定

5.1.1 隱藏ALB版本資訊

```
http {  
    server_tokens off; # 在http塊中加入這行  
}
```

5.1.2 限制請求方法

只允許必要的 HTTP 方法（如 GET、POST、HEAD、PUT）：

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

5.1.3 限制用戶端請求大小

防止 DoS 攻擊（如大檔案上傳、超長 URL）：

```
http {  
    client_max_body_size 1M; # 限制請求體，根據實際業務調整  
    client_header_buffer_size 1k; # 請求標頭緩衝區  
    large_client_header_buffers 4 8k;  
}
```

5.1.4 設定合理的逾時時間

防止慢速攻擊 (Slowloris 等)

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

5.1.5 啟用 HTTPS 並強制跳轉

```
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 設定 (首次部署建議先測試, 確認無誤後再啟用)
    add_header Strict-Transport-Security "max-age=31536000;
```

```
includeSubDomains" always;

}
```

5.2 反向代理安全設定

5.2.1 清理/重寫請求標頭

防止標頭注入或資訊洩露：

```
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;

        # 清除用戶端可能偽造的內部標頭（設定為空字串並不能完全阻止標頭傳遞，建議
        後端增加驗證）
        proxy_set_header X-Original-URI "";
        proxy_set_header X-Internal-Secret "";
    }
}
```

5.2.2 限制後端回應標頭

避免後端洩露敏感資訊（如 Server、X-Powered-By）：可透過 proxy_hide_header 隱藏：

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

5.2.3 限制代理請求速率

防CC攻擊

```
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 靜態資源服務專用安全設定

5.3.1 禁止目錄列表

確保未設定 `autoindex on;`，預設關閉即可。

5.3.2 禁止危險檔案類型存取

根據靜態資源類型設定，禁止危險副檔名（如 .php, .pl, .py, .jsp, .asp, .sh, .cgi）：

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

5.3.3 防止敏感檔案洩露

```
http {
    server {
        # 禁止存取以點開頭的隱藏目錄（如 .git、.svn）
        location ~ /\. {
            deny all;
        }
    }
}
```

```

# 禁止存取 .env 檔案
location ~ /\.env {
    deny all;
}
}
}

```

5.3.4 設定安全的 Content Security Policy (CSP) 和其他安全標頭

```

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 設定 (首次部署建議先測試, 確認無誤後再啟用)
    add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains" always;

    # CSP 範例 (根據實際調整, 生產環境應避免使用 '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 產品常見問題

註：效能相關錯誤參考【效能最佳化】手冊。

6.1 啟動失敗：getgrnam("nobody") failed

啟動中出現下面錯誤：

```
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
```

修復方式：

修改conf/alb.conf檔案中的user指令，把 `user nobody;` 改為對應靜態資源有存取權限的使用者名稱（建議建立專用使用者如 `alb` 並賦予權限，而非直接使用root）：

```
user alb;
```

6.2 Docker容器啟動失敗

Docker 容器啟動失敗後，可以使用 `docker logs <container_id>` 查看容器日誌。

1. 取得ALB容器ID

```
docker ps -a
CONTAINER ID          IMAGE                                COMMAND
CREATED              STATUS
PORTS                NAMES
xxxxxxxxxxxxx        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-v206-se-docker-amd64-
alb-1
```

如上ALB容器的ID是 xxxxxxxxxxxx

2. 查看容器日誌

使用docker logs 查看日誌，如下範例：

```
# 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啟動Permission denied錯誤

錯誤原因：容器內部的程序沒有足夠的權限來存取掛載的磁碟區中的檔案和目錄。

解決方案：

- 修改 alb-standard-dockercompose.yaml，標記使用 :z 標記掛載路徑（:z 標記僅適用於 SELinux 環境，非 SELinux 系統無需新增）

```
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
```

如上，每個掛載路徑結尾新增 :z 標記。

- 重啟容器

在設定好ALB後，存取頁面出現403錯誤頁面，如下所示：

403 Forbidden

alb/2.0

- 現象：存取任意的頁面（靜態資源）均出現403錯誤。
- 原因分析：ALB啟動的使用者，沒有對應靜態資源的讀權限，導致403錯誤。
- 問題解決：首選方案是修改靜態資源目錄權限（`chown -R nobody:nobody /path/to/static`），使ALB啟動使用者能夠存取靜態資源。次選方案是在`conf/alb.conf`設定檔案中修改 `user` 指令為對應靜態資源有存取權限的使用者名稱（建議建立專用使用者如 `alb`），最後重啟ALB即可。

6.5 AAS登入失敗

- 現象：使用ALB代理多個AAS服務，在輸入使用者名稱和密碼等點擊登入頁面後，然後出現登入失敗或者跳轉到登入頁面。
- 原因分析：
 - 工作階段不一致：瀏覽器存取的時候，不同請求發送到不同的AAS節點上，而登入的工作階段只在其中一個AAS伺服器中。
 - 請求標頭或cookie遺失：AAS伺服器中的服務依賴請求標頭或者cookie驗證用戶端是否登入，如果請求標頭或者cookie遺失，則無法驗證用戶端是否登入。
 - WebSocket連線失敗：某些服務在登入後，會建立WebSocket連線，如果WebSocket連線失敗，則導致即時功能不可用。
- 問題解決：
 - 工作階段不一致：ALB使用`ip_hash`或者`sticky`負載均衡演算法。
 - 請求標頭或cookie遺失：設定轉發全部請求標頭和cookie

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

```
# 以下內容為新增的設定

proxy_cookie_path / /; # 確保 Cookie 路徑正確
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連線失敗：單獨為WebSocket連線位址編寫location設定代理

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


    # WebSocket 必要設定
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";


    # 設定WebSocket連線讀逾時時間為10分鐘，根據業務修改時間
    proxy_read_timeout 600s;
    # 設定WebSocket連線寫逾時時間為10分鐘，根據業務修改時間
    proxy_send_timeout 600s;


    proxy_cookie_path / /; # 確保 Cookie 路徑正確
    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 後端反向代理服務的一個節點當機後，服務出現無法存取，但其他節點正常

- 現象：ALB反向代理服務的ABC三個服務節點，其中A當機或掛了，但BC服務正常，此時存取ALB出現短時間無法存取。
- 原因分析：ALB連線已經當機的A服務逾時導致請求等待時間過長。ALB預設的被動健康檢查機制會在 `max_fails` 次失敗後標記節點不可用，但預設逾時時間較長，期間新的請求仍可能轉發到故障節點。建議啟用主動健康檢查，讓ALB提前感知節點狀態，不再把請求轉發給當機服務。
- 問題解決：修改ALB連線上游服務的逾時時間，預設為60秒，修改為10秒或者更少，同時啟用主動健康檢查，讓ALB不再把請求轉發給當機服務，下面是參考範例

```

http {
    upstream backend {
        sticky cookie srv_id expires=1h domain=.example.com path=/
httponly secure; # 使用工作階段親和負載均衡演算法（與ip_hash二選一，不可同時
使用）

        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;

        # 設定主動健康檢查（如ALB未內建該模組，請參考官方文件確認）
        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/;
            # 核心設定，縮短服務連線逾時（建議至少5s，高併發場景根據實際調整）
            proxy_connect_timeout 5s;

```

```

    }
}
}

```

6.7 使用統一授權中心授權方式啟動失敗

- 現象：使用統一授權中心授權方式啟動失敗，失敗錯誤提示如下：

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

- 原因分析：授權中心更新導致ALB解析請求失敗
- 問題解決：更新至2025年發版的ALB即可。

6.8 502錯誤：no live upstreams while connecting to upstream

- 現象：前端存取出現502錯誤，並且ALB錯誤日誌出現大量 `no live upstreams while connecting to upstream`
- 原因分析：在ALB反向代理中，如果上游節點在 `proxy_connect_timeout`、`proxy_send_timeout`、`proxy_read_timeout` 指定的時間內未回應，那麼將會標記節點不可用。`max_fails` 預設值為1，配合 `fail_timeout` 使用，達到閾值後ALB會暫時將該節點標記為不可用，導致出現上述錯誤。
- 問題解決：修改 `proxy_connect_timeout`、`proxy_send_timeout`、`proxy_read_timeout` 為更長的時間（超過60s），或者將 `max_fails` 設定為3，表示不標記節點不可用。
- 設定範例：

```

upstream gateway_api {
    # 增加 max_fails和fail_timeout參數
    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/;

        # 增加逾時時間參數

```

```

    proxy_connect_timeout 15s;
    proxy_send_timeout 60s;
    proxy_read_timeout 60s;
}
}

```

6.9 管控台三員帳號密碼重設

- 現象：系統中所有三員帳號（admin-system、admin-security、admin-audit）均不可用，密碼遺忘或被鎖定，無法透過管控台介面登入。
- 原因分析：密碼長期未修改、初始隨機密碼未儲存，或帳號被安全策略鎖定。
- 問題解決：透過伺服器端命令列工具重設指定帳號的密碼，重設後會輸出新的隨機密碼，登入後系統會強制要求修改密碼。

切換到管控台安裝目錄（以實際安裝路徑為準）

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

重設指定帳號密碼（可重設 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      資料庫中已存在使用者,跳過預設使用者初始化
```

```
[alb-agile-console]2026/06/24 - 04:24:04.520      info
```

```
.../sqlite.go:204      使用者密碼重設成功          {"username": "admin-
system", "new_password": "%B4H%VKDf#yLj8oG"}
```

使用者 admin-system 密碼已重設為新密碼： %B4H%VKDf#yLj8oG

- 操作說明：
 - `--reinit` 參數後接需要重設的使用者名稱，可重設任意已存在的管控台帳號。
 - 命令執行成功後，終端會輸出新產生的隨機密碼，請立即記錄並在登入後強制修改。
 - 該命令僅重設密碼，不會重設帳號的其他設定資訊（如角色、狀態）。
 - 如需批次重設多個帳號，請分別執行上述命令，每次僅能重設一個帳號。
- 注意事項：該命令輸出的新密碼為系統自動產生的強密碼，請妥善保管並立即在管控台登入後修改為自訂密碼。如多次重設後仍無法登入，請聯絡金蝶天燕技術支援。

全國統一服務熱線
4008-555-800

金蝶天燕雲計算股份有限公司（簡稱“金蝶天燕”）成立於2000年，前身為“金蝶中間件公司”，是金蝶集團旗下基礎軟件平台提供商，雲計算國家標準制定企業，國家信創產業核心軟件企業。金蝶天燕是國家863重點研發計劃與核高基重大專項承接企業，也是“兩網一站四庫十二金”國家重點工程的基礎平台提供商，產品廣泛應用於政府、軍工、金融、能源等關鍵行業，累計服務客戶總數超過10萬家。



雲計算國家標準制定企業
金蝶集團旗下基礎軟件企業
資訊技術應用創新核心企業
官網：www.apusic.com

