



APUSIC
固若长城
睿比世界

Nginx 遷移手冊

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
- 1.3.1 Nginx 適配情況
- 2 總體適配聲明
- 3 設定相容性分級
- 4 內建功能模組清單
- 5 與 Nginx 的產品差異對照
- 6 典型使用場景適配性速查
- 6.0.1 遷移指南
 - 7 遷移流程總覽
 - 8 遷移前評估與盤點
 - 9 安裝 ALB 並申請授權
 - 10 遷移方式一：管控台「從 Nginx 遷移」精靈（推薦）
 - 11 遷移方式二：nginx-migrate 命令列工具
 - 11.1 功能特性
 - 11.2 使用方法
 - 11.3 命令列選項
 - 11.4 使用範例
 - 11.5 遷移後的目錄結構
 - 11.6 注意事項
 - 12 遷移方式三：手工遷移
 - 12.1 簡單直接使用 nginx.conf 檔案
 - 13 遷移後適配調整
 - 13.1 埠與監聽
 - 13.2 使用者與權限
 - 13.3 路徑核對清單
 - 13.4 國密憑證
 - 13.5 第三方模組
 - 13.6 服務啟停與開機啟動
 - 13.7 日誌切割

- 13.8 監控對接
 - 13.9 可選增強（遷移完成後按需啟用）
 - 14 驗證與割接
 - 14.1 設定驗證
 - 14.2 並行驗證（推薦）
 - 14.3 業務割接
 - 15 回滾方案
- 15.0.1 常見問題
- 15.1 遷移後執行 `alb -t` 報 `unknown directive` 錯誤
 - 15.2 遷移後 `load_module` 載入失敗
 - 15.3 啟動失敗：`getgrnam("nobody") failed`
 - 15.4 遷移後存取頁面出現 403 錯誤
 - 15.5 502 錯誤：`no live upstreams while connecting to upstream`
 - 15.6 普通使用者啟動 ALB 無法監聽 80 埠
 - 15.7 遷移後安全掃描或前端校驗異常
 - 15.8 管控台遷移精靈匯入的設定未生效
- 15.8.1 附錄：Nginx → ALB 常用操作速查

1 前言

本文件為金蝶Apusic負載均衡器軟體（ALB）V2.0.6標準版的Nginx適配與遷移手冊，針對正在使用Nginx的使用者，介紹ALB V2.0.6對Nginx的適配（相容）情況，以及從Nginx平滑遷移到ALB V2.0.6的完整方法，協助使用者順利完成替換升級。

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。

10	金蝶Apusic負載均衡器軟體 V2.0.6 Nginx 適配與遷移手冊	詳細介紹ALB對Nginx的適配情況，以及從Nginx遷移到ALB的方法（本文件）。
----	--------------------------------------	--

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
- 郵件：support@apusic.com
- 金蝶雲社群：<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

1.3.1 Nginx 適配情況

2 總體適配聲明

ALB V2.0.6 核心相容 Nginx 1.10 ~ Nginx 1.30 版本的設定語法與指令體系，標準的 `nginx.conf` 設定檔可直接作為 ALB 的主設定 `alb.conf` 複用，絕大多數場景無需改寫設定即可完成替換。

在相容 Nginx 的基礎上，ALB V2.0.6 還提供了以下增強能力：

- **國密 SSL**：支援 SM2 雙憑證（簽名憑證 + 加密憑證）體系。
- **主動健康檢查**：內建 `alb_healthcheck_module`，支援四層（TCP/UDP）和七層（HTTP）主動探活。
- **會話保持**：內建 sticky 模組，支援 cookie、route、learn 三種工作模式。
- **流量監控**：內建 vts/sts 狀態統計模組，提供頁面、JSON、Prometheus 三種監控資料輸出。
- **協議能力**：支援 HTTP/2、HTTP/3（QUIC）、gRPC、WebSocket、RTMP 直播、HTTP CONNECT 正向代理。
- **擴充能力**：內建 Lua 腳本引擎、Brotli 壓縮、headers-more、geo2 地理位置模組。
- **管理能力**：提供 Web 管控台（站台管理、設定管理、憑證管理、高可用、備份還原、第三方模組管理）、alb-cli 命令列與 AI Agent 整合。
- **高可用**：支援主備高可用（含單播模式）、設定同步、平滑升級與回滾。
- **TCP 加速**：ebpf 特性安裝包支援核心 eBPF 加速 TCP 轉發。

3 設定相容性分級

現有 Nginx 設定中的指令，按在 ALB V2.0.6 中的適配情況分為以下四級：

相容級別	說明	典型指令/設定
完全相容，直接複用	Nginx 標準指令，語法與行為一致，無需修改	worker_processes、events、http/server/location、upstream（輪詢、權重、ip_hash、least_conn、hash ... consistent）、proxy_pass 及 proxy_* 系列、listen、server_name、root/alias、index、try_files、rewrite/return、map/geo、gzip、proxy_cache、limit_req/limit_conn、allow/deny、auth_basic、error_page、add_header、ssl_*（RSA/ECC 憑證）、stub_status、stream 四層代理、include 等
相容，但需調整	語法相容，因產品目錄結構、預設值不同，需調整路徑或參數	日誌路徑（access_log、error_log）、pid 路徑、憑證檔案路徑、靜態資源 root/alias 路徑、include 相對路徑、預設監聽埠（ALB 出廠預設為 8080）、user 指令（預設 nobody）
ALB 擴充，按需啟用	Nginx 開源版不具備的能力，遷移後可選用以增強設定	check（主動健康檢查）、sticky（會話保持）、vhost_traffic_status_*（vts 監控）、stream_server_traffic_status_*（四層監控）、國密雙憑證設定（ssl_certificate 設定 sig + enc 雙憑證）、brotli_*、access_by_lua_* 等 Lua 指令、more_set_headers、proxy_connect、geo2、eBPF 加速
不相容，需人工處理	ALB 未內建或與產品機制衝突，遷移時需替換或移除	為 Nginx 編譯的第三方動態模組（.so）無法直接在 ALB 中載入，需使用為 ALB 構建的模組並透過管控台「第三方模組管理」上傳啟用；Nginx 商業版（Nginx Plus）專屬指令不在相容範圍內；未在 ALB 內建模組清單中的模組指令，alb -t 校驗時將報 unknown directive 錯誤

注：上表未列出的 Nginx 標準指令，以 `./sbin/alb -t` 實際校驗結果為準。

4 內建功能模組清單

ALB V2.0.6 出廠編譯內建以下模組，對應 Nginx 同名編譯選項或模組：

模組	功能說明
<code>--with-http_stub_status_module</code>	啟用 stub_status 介面，提供執行狀態統計資訊。
<code>--with-http_ssl_module</code>	啟用 HTTPS/SSL 支援（含國密 SSL）。
<code>--with-ipv6</code>	啟用 IPv6 監聽支援。
<code>--with-http_mp4_module</code>	支援 MP4 串流媒體虛擬串流（pseudo-streaming）。
<code>--with-http_auth_request_module</code>	實作子請求認證（auth_request），方便與外部認證服務整合。
<code>--with-http_gzip_static_module</code>	支援直接發送 .gz 預壓縮檔案，節省 CPU。
<code>--with-http_realip_module</code>	從 X-Forwarded-For/X-Real-IP 等標頭中取得真實用戶端 IP。
<code>--with-http_gunzip_module</code>	對不支援 gzip 的用戶端自動解壓縮回應內容。
<code>--with-http_sub_module</code>	回應主體字串替換（sub_filter）。
<code>--with-stream</code>	啟用 4 層 TCP/UDP 代理（stream 子系統）。
<code>--with-stream_ssl_module</code>	為 stream 子系統提供 SSL/TLS 支援。
<code>--with-threads</code>	開啟執行緒池支援，提升檔案 I/O 效能。
<code>--with-file-aio</code>	啟用非同步檔案 I/O（Linux AIO）。
<code>--with-http_v2_module</code>	啟用 HTTP/2 支援。
<code>--with-http_v3_module</code>	啟用 HTTP/3（QUIC）支援。
<code>--with-http_addition_module</code>	在回應前後追加內容（add_before_body / add_after_body）。
<code>--with-http_dav_module</code>	支援 WebDAV 協議（PUT、DELETE、MKCOL 等）。
<code>--with-http_random_index_module</code>	目錄索引時隨機選擇預設檔案。
<code>--with-http_secure_link_module</code>	產生和校驗帶時效、防竄改的「安全連結」。
<code>--with-mail</code>	啟用郵件代理（IMAP/POP3/SMTP）子系統。
<code>--with-mail_ssl_module</code>	為郵件代理提供 SSL/TLS 支援。

<code>--with-http_slice_module</code>	支援大檔案分片下載 (Range Slice) 。
<code>--with-stream_ssl_preread_module</code>	在 4 層代理裡預讀 SNI/ALPN，實現按域名路由。
<code>alb_healthcheck_module</code>	ALB主動健康檢查模組，支援四層和七層代理節點主動探活。
<code>alb-module-vts</code>	提供詳細流量 / 狀態統計介面 (<code>/status</code>) 。
<code>alb-module-stream-sts</code>	為 stream 子系統提供即時狀態統計。
<code>alb-module-sts</code>	為 http 子系統提供即時狀態統計。
<code>alb_http_proxy_connect_module</code>	支援 HTTP CONNECT 方法，正向代理 HTTPS 流量。
<code>alb-rtmp-module</code>	引入 RTMP/HLS/DASH 直播與串流媒體功能。
<code>alb_brotli</code>	支援 Brotli 壓縮演算法，減少傳輸體積。
<code>alb-module-lua</code>	引入 Lua 腳本功能，可擴充 ALB 功能。
<code>alb-headers-more</code>	設定和清除請求和回應標頭。

如現有 Nginx 使用了上表之外的模組（透過 `--add-module` 編譯的第三方模組或 `load_module` 載入的動態模組），請在遷移前單獨排查確認，必要時聯絡技術支援取得 ALB 版本的模組檔案。

5 與 Nginx 的產品差異對照

遷移到 ALB 後，日常操作的物件和方式與 Nginx 存在以下差異：

項目	Nginx	ALB V2.0.6	遷移影響
二進位程式	<code>sbin/nginx</code>	<code>sbin/alb</code>	腳本、維運命令中的程式名需替換
主設定檔	<code>conf/nginx.conf</code>	<code>conf/alb.conf</code>	原 <code>nginx.conf</code> 可直接複製為 <code>alb.conf</code> 使用
出廠預設 HTTP 埠	80	8080	遷移設定時以原 <code>nginx</code> 設定的 <code>listen</code> 為準
pid 檔案	<code>logs/nginx.pid</code>	<code>logs/alb.pid</code>	維運腳本中的 <code>pid</code> 路徑需調整
設定語法檢查	<code>nginx -t</code>	<code>./sbin/alb -t</code>	用法一致
平滑 reload	<code>nginx -s reload</code>	<code>bin/reload-alb.sh</code> (或 <code>kill -HUP</code>)	推薦使用安裝包自帶腳本
啟停方式	<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-alb-and-console.sh</code>	參考《快速使用手冊》
行程訊號	HUP/USR2/WINCH/QUIT	同名訊號完全相容，支援 USR2 平滑升級	原有基於訊號的操作習慣可保留
systemd 服務	自行編寫 <code>nginx.service</code>	安裝包 <code>utils/systemd/</code> 提供 <code>alb.service</code> 與 <code>enable_startup.sh</code> 一鍵註冊	需替換原 <code>nginx.service</code> 並停用
回應 Server 標頭	<code>Server: nginx</code>	<code>Server: alb</code>	若有依賴該字樣的前端校驗或安全掃描規則需調整
產品授權	無	需放置授權檔案： <code>license.lic</code> 、 <code>license.xml</code> 或統一授權 <code>acls.properties</code>	遷移前先申請授權，參考《安裝手冊》
視覺化管理	開源版無	管控台（預設埠 8887），支援站台/設定/憑證/高可用/備份管理	遷移後可轉為管控台納管

監控介面	<code>stub_status</code>	相容 <code>stub_status</code> , 另提供 <code>vtls /status</code> (頁面/JSON/Prometheus)	監控蒐集可平滑升級
------	--------------------------	---	-----------

6 典型使用場景適配性速查

Nginx 典型場景	ALB V2.0.6 適配情況	參考章節（《代理功能手冊》）
靜態資源代管 / SPA 單頁應用	完全支援	使用者場景設定案例·場景一
反向代理與負載均衡	完全支援，支援輪詢/權重/IP雜湊/最少連線/一致性雜湊	負載均衡演算法
HTTPS / SSL 終止	完全支援，並支援國密雙憑證	HTTPS與國密
HTTP/2	完全支援	HTTPS與國密
HTTP/3 (QUIC)	支援 (V2.0.6 起)	HTTPS與國密·HTTP/3
WebSocket 代理	完全支援	WebSocket 代理
gRPC 代理	完全支援	gRPC 代理
TCP/UDP 四層代理	完全支援 (stream)	TCP代理
正向代理 (HTTP CONNECT)	支援	正向代理
反向代理快取 (proxy_cache)	完全支援	快取設定
限流限速 (limit_req/limit_conn)	完全支援	流量控制
URL 重寫與重導向	完全支援	URL 重寫與重導向
跨域 (CORS)	完全支援	跨域設定 (CORS)
故障轉移與重試	完全支援 (proxy_next_upstream)	故障轉移與重試
會話保持	支援 ip_hash / Cookie 一致性雜湊 / sticky 模組 / Lua+Redis	會話保持
主動健康檢查	支援 (check 指令, Nginx 開源版不具備)	健康檢查
灰度發布	支援 (按 IP、比例、Cookie、Lua 動態分流)	灰度發布和灰度測試
壓縮	支援 Gzip、Brotli、預壓縮靜態檔案	壓縮

郵件代理 (IMAP/POP3/SMTP)	支援	內建模組清單
RTMP/HLS 直播	支援	內建模組清單
Lua 擴充開發	內建 Lua 引擎與 lua-resty 生態函式庫	會話保持 方式三
動態 DNS 解析 (resolver)	完全支援	動態DNS解析

6.0.1 遷移指南

7 遷移流程總覽

從 Nginx 遷移到 ALB V2.0.6 的整體流程如下：

遷移評估與盤點 → 安裝 ALB 並申請授權 → 設定遷移（三種方式任選）
→ 適配調整 → 驗證測試 → 業務割接 → 觀察期（保留回滾能力）

三種設定遷移方式對比：

遷移方式	適用場景	特點
管控台「從 Nginx 遷移」精靈	所有場景，推薦使用	視覺化操作，自動掃描並匯入站台與設定，支援追加/覆蓋兩種模式
nginx-migrate 命令列工具	伺服器上無瀏覽器存取 管控台、習慣命令列操作	自動遷移設定、憑證、靜態資源、日誌路徑，支援預覽模式（部分安裝包未內建該工具，以實際安裝包 <code>utils/</code> 目錄為準）
手工遷移	設定簡單、或需精細化 控制遷移內容	直接複製 <code>nginx.conf</code> 為 <code>alb.conf</code> ，逐項調整路徑

8 遷移前評估與盤點

遷移前請先完成以下盤點工作，並輸出盤點清單：

1. 確認 Nginx 版本與編譯參數

```
nginx -v      # 查看版本，確認在 1.10 ~ 1.30 相容範圍內
nginx -V      # 查看編譯選項與內建/第三方模組
```

2. 匯出完整生效設定

```
nginx -T > nginx-full-config-backup.conf # 匯出主設定與所有
include 子設定
```

3. 盤點設定依賴項，逐項記錄：

盤點項	盤點內容	排查命令/方法範例
設定檔	主設定與所有 include 子設定檔的位置	nginx -T、檢查 include 指令
第三方 模組	load_module 動態模組、--add-module 編譯模組	nginx -V、nginx -T grep load_module
SSL 憑 證	憑證/私密金鑰檔案路徑、到期時間、是否國密	grep -r ssl_certificate
靜態資 源	root、alias 指向的目錄及磁碟佔用	grep -E 'root alias'
監聽埠	所有 listen 埠及防火牆放行情況	grep -r listen、檢查防火牆規則
後端服 務	upstream 後端節點清單、目前健康檢查方式	grep -A5 upstream
日誌	access_log/error_log 路徑、logrotate 設定	grep -E '_log'、/etc/logrotate.d/
自動啟 動	systemd unit、init 腳本、rc.local	systemctl status nginx

排程任務	憑證續期、日誌切割等 crontab	<code>crontab -l</code> 、 <code>/etc/cron.d/</code>
監控對接	stub_status 蒐集、Prometheus exporter	檢查監控系統設定

4. **特殊指令排查**：如設定中包含 `load_module`、Perl/njs 腳本、Nginx Plus 商業版指令等，請單獨記錄並在遷移時重點處理。

9 安裝 ALB 並申請授權

1. 在目標伺服器上安裝 ALB V2.0.6（支援 tar.gz 解壓、RPM、Docker、K8s 多種部署方式），詳細步驟參考《安裝手冊》。

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

2. 取得本機授權碼並申請授權檔案，將授權檔案（`license.lic` / `license.xml` / `acls.properties`）放置在安裝目錄下：

```
cd /opt/ALB-V2.0.6-SE-amd64
./bin/alb-authcode          # 取得授權碼，憑授權碼申請授權檔案
```

3. 驗證 ALB 可正常啟動：

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

遷移期建議：安裝 ALB 後先保持預設埠（8080）執行，與現有 Nginx 並行，待設定遷移驗證完成後再割接埠。

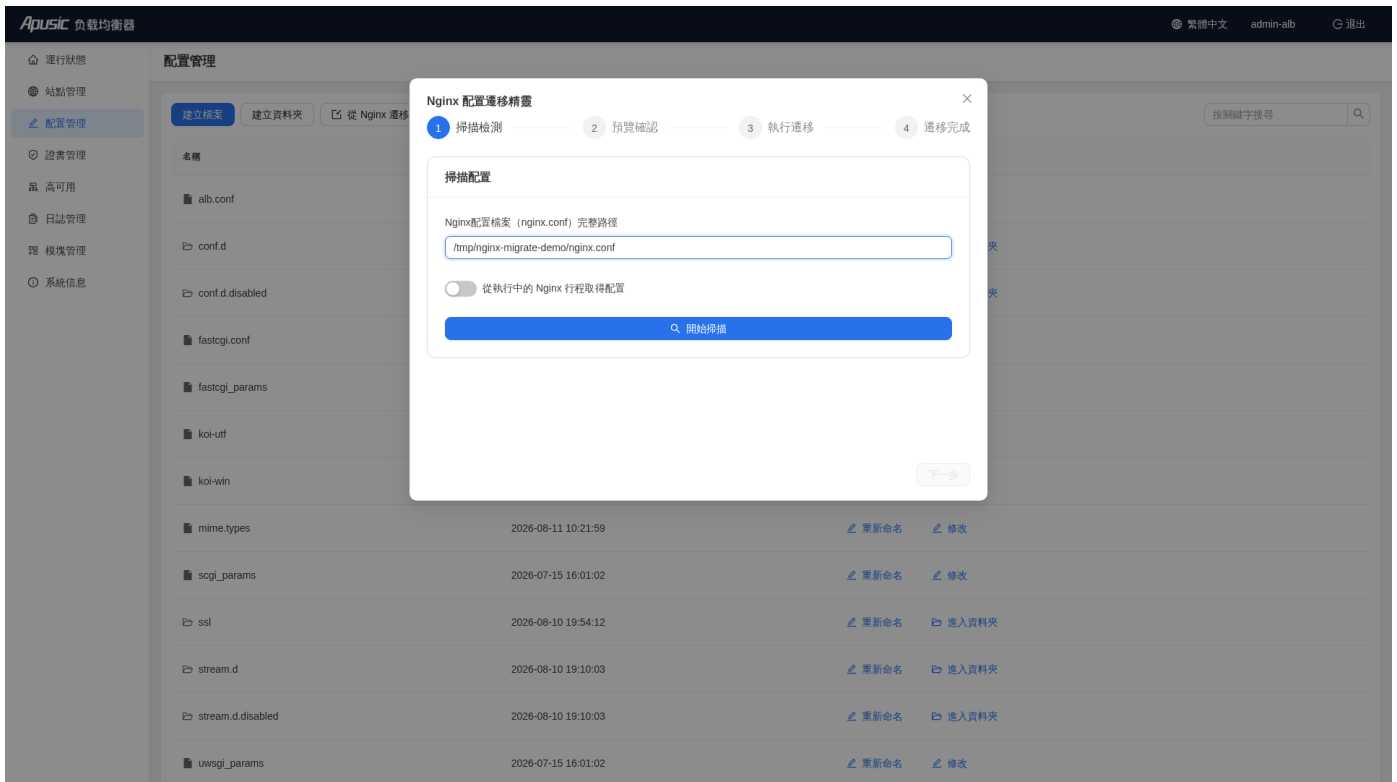
10 遷移方式一：管控台「從 Nginx 遷移」精靈（推薦）

ALB V2.0.6 管控台內建 Nginx 遷移精靈，可將現有 nginx 設定檔匯入為 ALB 站台與設定。

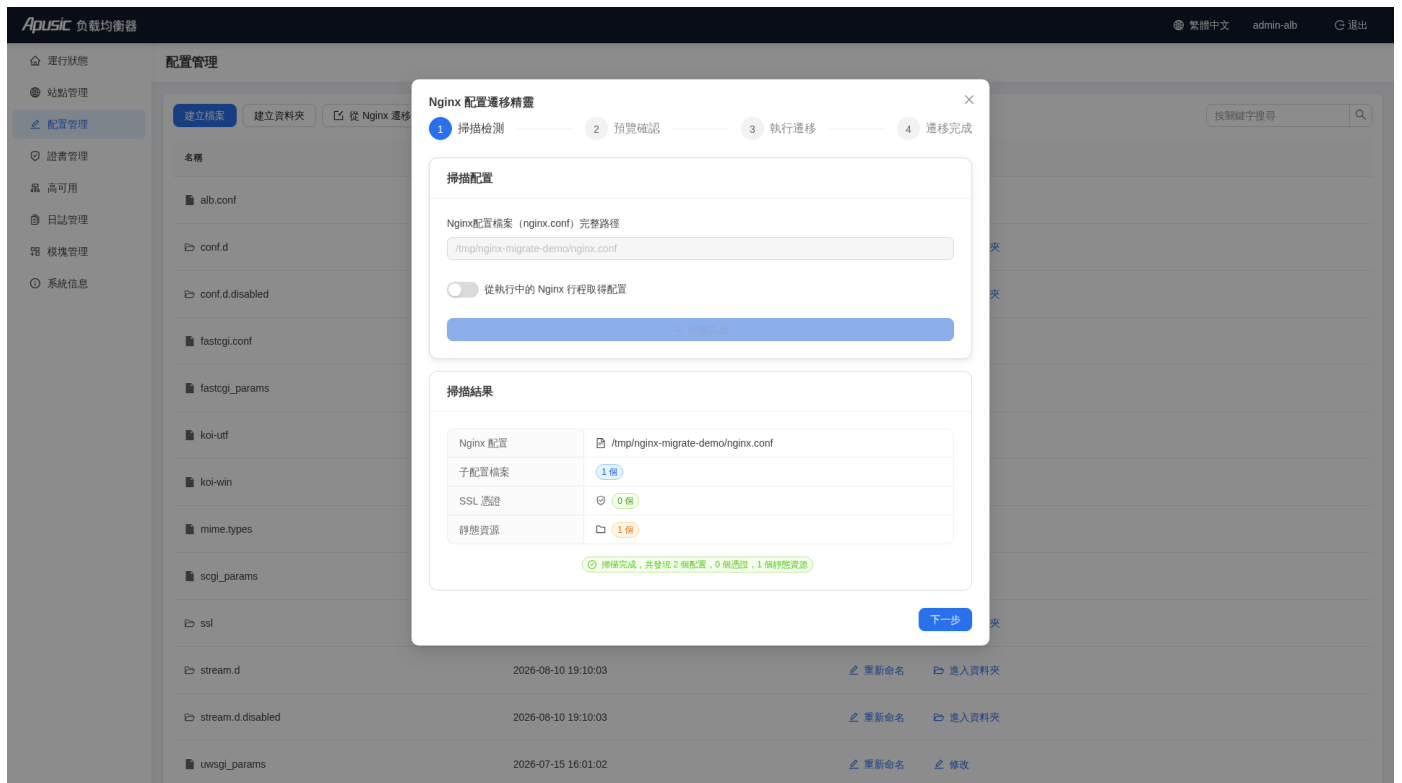
前置條件：ALB 與管控台已啟動，且 Nginx 設定檔位於 ALB 所在伺服器上（或已將設定檔拷貝至該伺服器）。

操作步驟：

1. 登入管控台（預設 `http://伺服器IP:8887`），進入設定管理頁面，點擊「從 Nginx 遷移」，在對話框中填入 Nginx 的設定檔路徑，點擊掃描：

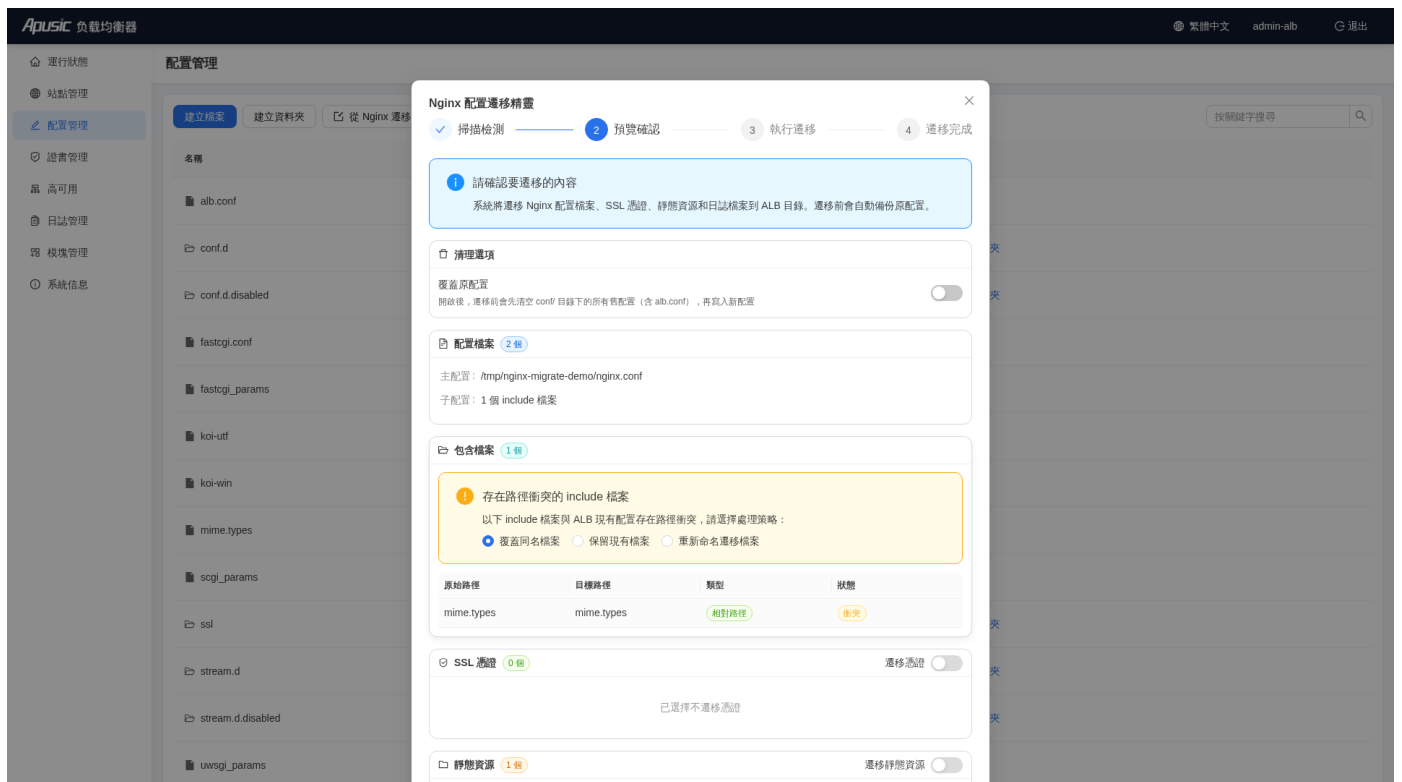


2. 掃描成功後，可查看掃描結果，確認無誤後進行遷移：

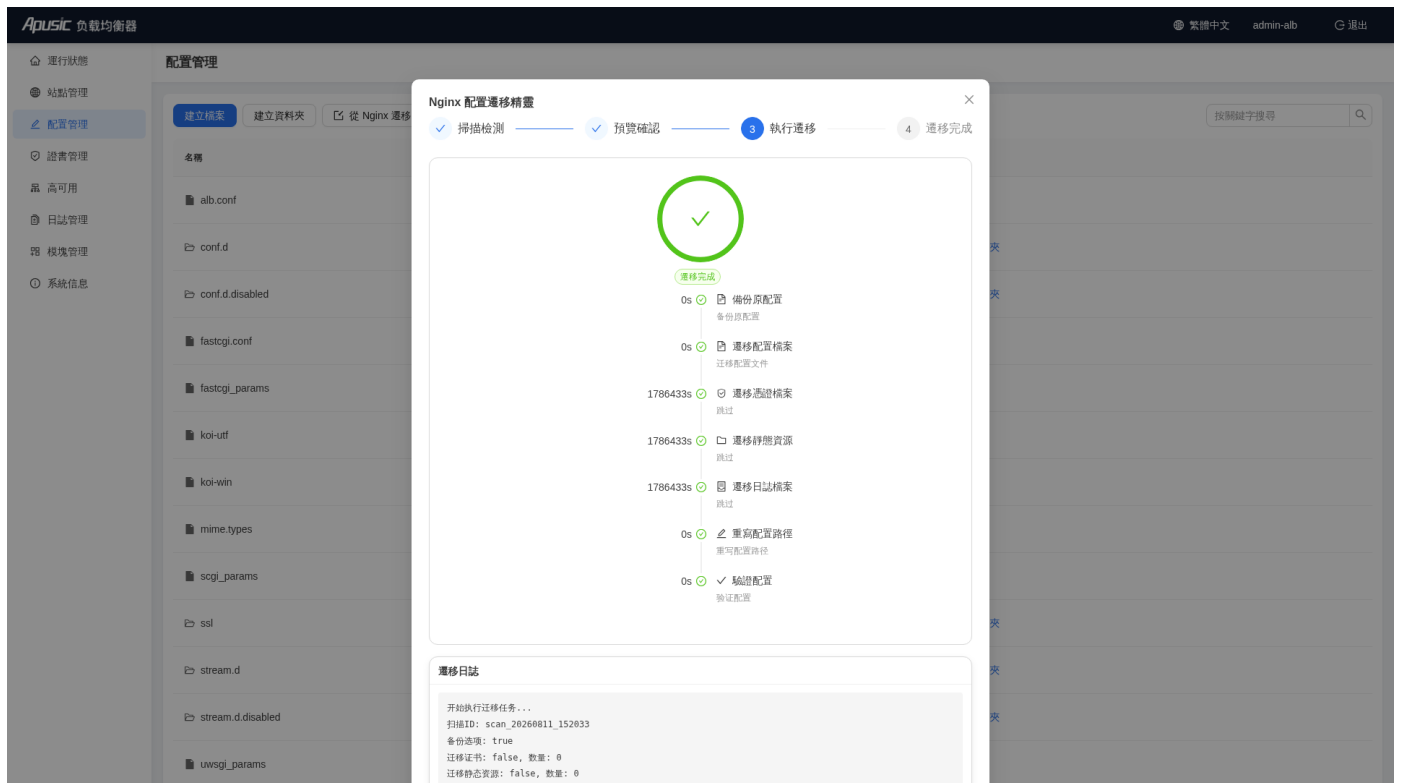


3. 遷移完成後可查看遷移結果。

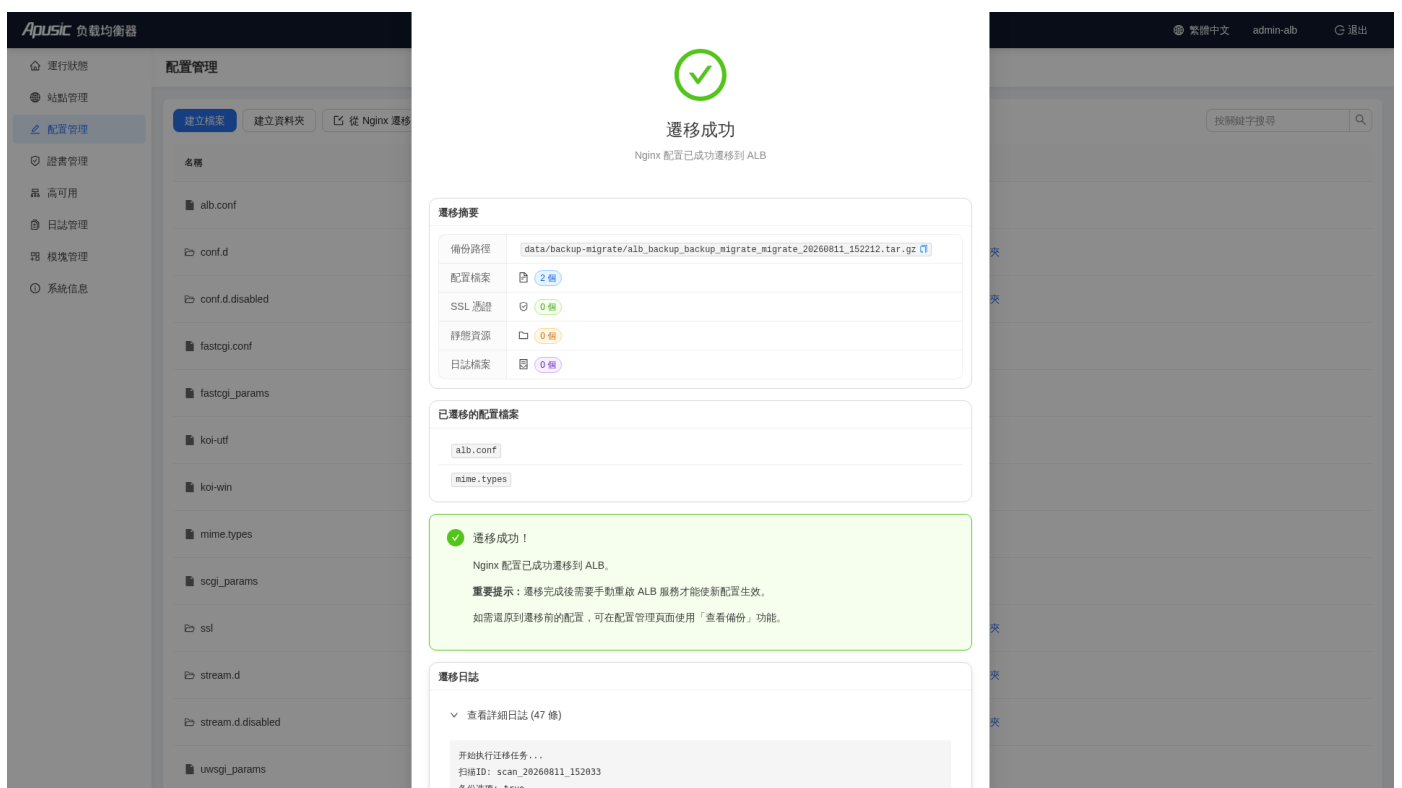
注：開啟「覆蓋原設定」將清空原有設定；預設行為是追加 Ngix 設定而非覆蓋。



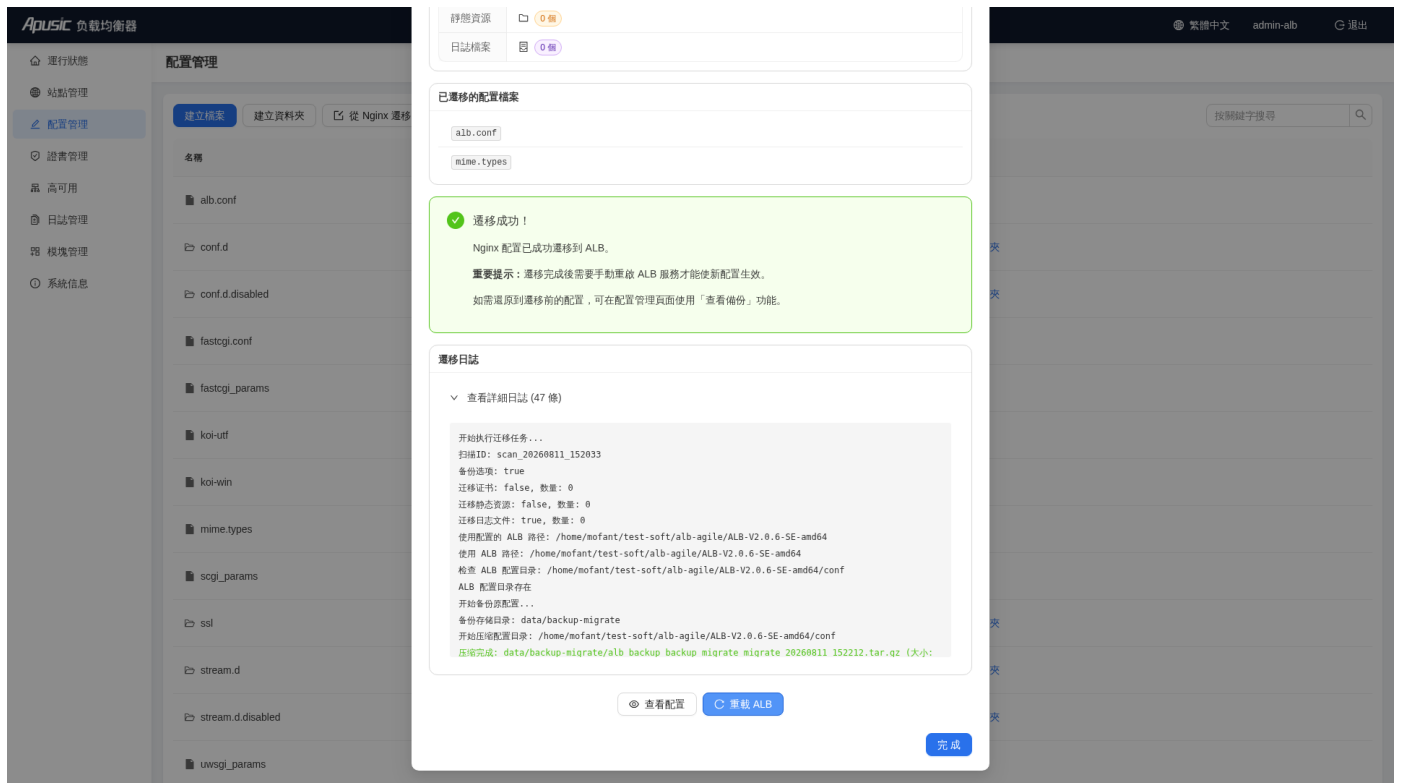
4. 查看遷移結果明細：



5. 查看最終結果：



6. 遷移完成後，重新載入 ALB 使設定生效：



注意事項：

- 遷移匯入的站台設定寫入 `conf/conf.d/` 目錄，請確認主設定 `conf/alb.conf` 中包含 `include conf.d/*.conf;`，否則管控制台會顯示黃色警告，站台設定不會生效。
- 遷移完成後建議使用管控制台「設定管理」的備份功能建立一次完整備份。
- 若設定中包含 ALB 未內建的指令，重載時會提示設定錯誤，請按「遷移後適配調整」章節處理。

11 遷移方式二：nginx-migrate 命令列工具

部分安裝包提供了 Nginx 設定遷移命令列工具，位於 `[ALB安裝路徑]/utils/nginx-migrate`，用於將 nginx 設定一鍵遷移為 ALB 設定。

注：請以實際安裝包 `utils/` 目錄為準。若安裝包中未提供該工具，請使用管控台遷移精靈或手工遷移。

11.1 功能特性

- **設定遷移**：掃描 nginx 主設定檔（`nginx.conf`），解析並遷移所有 `include` 子設定檔到 `conf/conf.d/` 目錄，自動更新主設定的 `include` 路徑為 `conf.d/*.conf`。
- **SSL 憑證遷移**：自動掃描 `ssl_certificate`、`ssl_certificate_key`、`ssl_trusted_certificate`，複製憑證檔案到 `conf/certs/` 目錄並更新設定中的憑證路徑。
- **靜態資源遷移**：自動掃描 `root` 和 `alias` 指令，複製靜態資源到 `conf/html/` 目錄並更新設定中的靜態資源路徑為絕對路徑。
- **日誌路徑遷移**：自動遷移 `error_log`、`access_log`、`pid` 路徑，統一到 `{ALBPath}/logs/` 目錄。
- **其他**：遷移前自動備份原設定、互動式確認（可跳過）、詳細日誌模式、設定驗證。

11.2 使用方法

使用前提：

1. 同一台機器已經部署安裝好 ALB。
2. nginx 設定檔所在目錄和 ALB 安裝目錄已存在。

```
# 自動偵測 nginx 設定和 ALB 路徑
./nginx-migrate

# 指定 nginx 設定和 ALB 目錄
./nginx-migrate --nginx-path /path/to/nginx.conf --alb-path /path/to/ALB
```

11.3 命令列選項

選項	簡寫	說明	預設值
<code>--nginx-path</code>	-	nginx 設定檔路徑	自動偵測

<code>--alb-path</code>	-	ALB 安裝目錄	從工具位置推斷
<code>--from-running</code>	-	從執行中的 nginx 行程取得設定路徑	false
<code>--dry-run</code>	-	僅掃描，不執行遷移	false
<code>-y / --yes</code>	-	自動確認所有提示	false
<code>-v / --verbose</code>	-	顯示詳細日誌	false
<code>--backup</code>	-	遷移前備份原設定	true

11.4 使用範例

```
# 從執行中的 nginx 行程取得設定並遷移
./nginx-migrate --from-running --alb-path /path/to/ALB

# 完整模式 (自動確認 + 詳細日誌)
./nginx-migrate --nginx-path /etc/nginx/nginx.conf --alb-path
/opt/ALB -y -v

# 預覽模式 (僅掃描)
./nginx-migrate --nginx-path /etc/nginx/nginx.conf --alb-path
/opt/ALB --dry-run

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

11.5 遷移後的目錄結構

```
ALB/
├─ conf/
│   └─ alb.conf           # 主設定檔
│   └─ certs/            # SSL 憑證
│       └─ server.crt
│       └─ server.key
```

```

├── ca.crt
├── conf.d/                # 子設定檔
│   ├── default.conf
│   ├── ssl.conf
│   └── upstream.conf
├── html/                 # 靜態資源
│   ├── index.html
│   ├── css/
│   └── js/
└── logs/                 # 日誌目錄
    ├── access.log
    ├── error.log
    └── alb.pid

```

11.6 注意事項

- **路徑問題**：確保 nginx 設定檔中的路徑為**相對路徑**或**絕對路徑**，相對路徑會基於設定檔所在目錄解析。建議先使用 `--dry-run` 預覽。
- **備份**：遷移前會自動備份原設定到 `conf.backup_YYYYMMDD_HHMMSS`；如需還原，刪除新設定並將備份目錄重新命名為 `conf`。
- **權限**：確保有讀取 nginx 設定和憑證檔案的權限，以及寫入 ALB 目錄的權限。
- **驗證**：遷移完成後工具會自動驗證設定語法，建議重啟 ALB 前再手動執行一次 `./sbin/alb -t` 檢查。

12 遷移方式三：手工遷移

12.1 簡單直接使用 nginx.conf 檔案

1. 備份舊設定，並將 nginx.conf 複製為 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
# 根據實際情況修改來源路徑
chmod 644 /opt/ALB-V2.0.6-SE-amd64/conf/alb.conf
```

2. 調整 include 路徑：如果原 nginx.conf 中使用了 `include` 指令引用子設定檔（如 `conf.d/*.conf`），需確保這些路徑在 ALB 安裝目錄中有效。建議將子設定檔複製到 `conf/conf.d/` 目錄：

```
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. 遷移 SSL 憑證檔案：將原 nginx 使用的 SSL 憑證檔案複製到 ALB 目錄，並更新 alb.conf 中的憑證路徑：

```
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/
```

編輯 alb.conf，將 `ssl_certificate` 和 `ssl_certificate_key` 路徑更新為新的絕對路徑。

4. 調整日誌與 pid 路徑：

```
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. 驗證設定並啟動 ALB:

```
/opt/ALB-V2.0.6-SE-amd64/sbin/alb -t -c /opt/ALB-V2.0.6-SE-  
amd64/conf/alb.conf  
# 驗證成功提示: syntax is ok 和 test is successful  
  
# 停止 Nginx (根據實際情況修改 Nginx 安裝路徑)  
/path/to/sbin/nginx -s stop  
  
# 啟動 ALB  
/opt/ALB-V2.0.6-SE-amd64/bin/start-alb.sh
```

13 遷移後適配調整

設定遷移完成後，請逐項核查以下適配事項：

13.1 埠與監聽

- ALB 出廠預設設定監聽 8080 埠，遷移時應以原 nginx 設定中的 `listen` 為準。
- 如需以普通使用者執行 ALB 並監聽 80/443 等特殊權限埠，需先執行（root 使用者）：

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

- 確認防火牆/安全群組已放行 ALB 監聽埠及管控台埠（預設 8887）。

13.2 使用者與權限

- 確認設定中 `user` 指令指定的使用者/群組在系統上存在（預設 `nobody`）。若啟動報 `getgrnam("nobody") failed`，請建立對應使用者群組或修改 `user` 指令，參考《維運手冊》常見問題。

13.3 路徑核查清單

- `access_log`、`error_log`、`pid` 路徑是否指向 ALB 安裝目錄 `logs/`。
- `ssl_certificate`、`ssl_certificate_key`、`ssl_trusted_certificate` 憑證路徑是否有效。
- `root`、`alias` 靜態資源路徑是否有效，目錄權限是否正確。
- `proxy_cache_path` 等快取路徑是否有效。
- `include` 引用的子設定檔是否全部就位（相對路徑基於 ALB 安裝目錄解析）。

13.4 國密憑證

- 如遷移後啟用國密 SSL，需同時設定簽名憑證（sig）和加密憑證（enc），且 sig 憑證必須設定在 enc 憑證之前：

```
ssl_certificate      /path/to/your_domain_name.sig.crt.pem;    #
簽名憑證（先設定）
ssl_certificate_key  /path/to/your_domain_name.sig.key.pem;
ssl_certificate      /path/to/your_domain_name.enc.crt.pem;    #
```

加密憑證（後設定）

```
ssl_certificate_key /path/to/your_domain_name.enc.key.pem;
```

- 也可透過管控台「憑證管理」上傳國密憑證並綁定站台。

13.5 第三方模組

- 原 Nginx 透過 `load_module` 載入的 `.so` 動態模組**不能直接在 ALB 中使用**，需使用為 ALB 構建的模組檔案，透過管控台「第三方模組管理」上傳並啟用。
- 設定中殘留的 `load_module` 行如未對應有效模組，需先註解，否則 `alb -t` 校驗失敗。

13.6 服務啟停與開機啟動

- 停用並移除原 nginx 自動啟動：

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

- 註冊 ALB 開機自動啟動：

```
cd /opt/ALB-V2.0.6-SE-amd64
./utils/systemd/enable_startup.sh      # 自動設定開機啟動項（通常需要
root 權限）
systemctl start alb
```

13.7 日誌切割

- 原 nginx 的 logrotate 設定不適用於 ALB 日誌路徑，請使用 ALB 自帶的日誌切割工具（`utils/logrotate`），支援排程切割與備份，參考《維運手冊》日誌切割章節。管控台「日誌管理」也支援按設定排程切割和備份日誌檔案。

13.8 監控對接

- 原有基於 `stub_status` 的蒐集可直接沿用。
- 建議升級為 vts 監控，獲得更全面的流量與狀態指標，支援 Prometheus 直接整合：

```

http {
    vhost_traffic_status_zone;                # 啟用vts模組

    server {
        listen 8080;

        location /status {                  # 設定vts查看的uri地址
            vhost_traffic_status_display;
            vhost_traffic_status_display_format html;
        }
    }
}

```

- 監控頁面: `http://IP:Port/status`
- JSON 資料: `http://IP:Port/status/format/json`
- Prometheus: `http://IP:Port/status/format/prometheus`

13.9 可選增強（遷移完成後按需啟用）

- 主動健康檢查: 將原被動的 `max_fails/fail_timeout` 升級為 `check` 主動探活:

```

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;
}

```

- 會話保持增強: 使用內建 `sticky` 模組替代 `ip_hash`, 避免 NAT 場景下會話不均:

```

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;  
}
```

- **管控台納管：**透過管控台管理站台、憑證、高可用與設定備份，提升維運效率。

14 驗證與割接

14.1 設定驗證

```
cd /opt/ALB-V2.0.6-SE-amd64
./sbin/alb -t          # 驗證 conf/alb.conf 設定是否存在語法錯誤
```

14.2 並行驗證（推薦）

割接前讓 ALB 與 Nginx 並行執行（ALB 使用臨時埠），逐一對比驗證：

```
# 對比回應狀態碼、回應主體、關鍵回應標頭
curl -s -o /dev/null -w "%{http_code}\n" http://nginx地址/業務路徑
curl -s -o /dev/null -w "%{http_code}\n" -H "Host: 業務域名"
http://alb臨時埠/業務路徑
```

功能驗證清單：

- 靜態資源存取（頁面、圖片、CSS/JS）
- 反向代理後端業務（含 POST 請求、大檔案上傳）
- HTTPS 站台（憑證、協議版本、HTTP/2）
- WebSocket 長連線（如有）
- 負載均衡與會話保持（如有）
- 限流、存取控制等安全策略（如有）
- 日誌正常輸出，格式符合預期

如需效能對比驗證，參考《效能最佳化手冊》。

14.3 業務割接

建議在業務低峰期執行割接：

1. 修改 ALB 設定中的 `listen` 為正式業務埠（80/443 等），執行 `./sbin/alb -t` 驗證。
2. 停止 Nginx 服務：`systemctl stop nginx`（或 `nginx -s quit`）。
3. 重載或啟動 ALB：`./bin/reload-alb.sh`（或 `./bin/start-alb.sh`）。
4. 立即驗證核心業務存取，觀察 `logs/error.log` 有無異常。

5. 進入觀察期（建議不少於一個業務高峰週期），期間保留 Nginx 環境與設定，確保可隨時回退。

若前端有 DNS、LVS、雲負載均衡等流量入口，也可採用「新增 ALB 節點→逐步切流→下線 Nginx 節點」的灰度割接方式，進一步降低風險。

15 回滾方案

- **設定級回滾**：ALB 側設定問題，可透過管控台「設定管理→查看備份」回滾到任一歷史備份（備份儲存於【ALB安裝目錄】/backup/，預設保留最近 30 個），回滾不中斷服務、僅觸發設定重載；命令列方式可還原遷移工具自動備份的 `conf.backup_YYYYMMDD_HHMMSS` 目錄。
- **服務級回滾**：觀察期內如需整體回退到 Nginx，停止 ALB 後直接重新啟動原 Nginx 即可（原 Nginx 二進位與設定未做改動）：

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

15.0.1 常見問題

15.1 遷移後執行 `alb -t` 報 `unknown directive` 錯誤

- 原因：設定中包含了 ALB 未內建的模組指令（如原 Nginx 的第三方模組指令）。
- 解決：對照本文件「內建功能模組清單」確認模組支援情況；第三方模組需使用為 ALB 構建的版本並透過管控台「第三方模組管理」上傳啟用；臨時處置可先註解相關設定行。如無法確認，請聯絡技術支援。

15.2 遷移後 `load_module` 載入失敗

- 原因：為 Nginx 編譯的 `.so` 動態模組與 ALB 二進位不相容。
- 解決：移除或註解 `load_module` 設定行，聯絡技術支援取得 ALB 版本的模組檔案。

15.3 啟動失敗：`getgrnam("nobody") failed`

- 原因：系統中不存在設定中 `user` 指令指定的使用者群組。
- 解決：建立對應使用者/群組，或修改 `user` 指令為系統已有使用者，詳見《維運手冊》常見問題。

15.4 遷移後存取頁面出現 403 錯誤

- 原因：靜態資源目錄權限不足、目錄中缺少索引檔案、或 `user` 指令指定的執行使用者無權讀取資源。
- 解決：檢查目錄權限與 `root/alias` 路徑，詳見《維運手冊》常見問題。

15.5 502 錯誤：`no live upstreams while connecting to upstream`

- 原因：upstream 後端節點全部不可用，或健康檢查將節點全部剔除。

- 解決：確認後端服務可達；若設定了 `check` 主動健康檢查，確認健康檢查參數（URL、期望狀態碼）與後端實際匹配，詳見《維運手冊》常見問題。

15.6 普通使用者啟動 ALB 無法監聽 80 埠

- 原因：80/443 等特殊權限埠預設僅 root 可綁定。
- 解決：root 使用者執行 `setcap cap_net_bind_service=+eip ./sbin/alb` 後，普通使用者即可監聽 80 埠。

15.7 遷移後安全掃描或前端校驗異常

- 原因：ALB 回應標頭 `Server` 標識為 `alb`，與 Nginx 的 `nginx` 不同。
- 解決：調整依賴該字樣的校驗/掃描規則；如需隱藏版本資訊，可設定 `server_tokens off;`。

15.8 管控台遷移精靈匯入的設定未生效

- 原因：主設定 `alb.conf` 未包含 `include conf.d/*.conf;`。
- 解決：在 `alb.conf` 的 `http` 區塊中加入該 `include` 指令後重載；管控台頁面會對此情況顯示黃色警告提示。

15.8.1 附錄：Nginx → ALB 常用操作速查

操作	Nginx	ALB V2.0.6
查看版本	<code>nginx -v</code>	<code>./sbin/alb -v</code>
查看編譯參數	<code>nginx -V</code>	<code>./sbin/alb -V</code>
校驗設定	<code>nginx -t</code>	<code>./sbin/alb -t</code>
校驗指定設定	<code>nginx -t -c /path/nginx.conf</code>	<code>./sbin/alb -t -c conf/alb.conf</code>
啟動	<code>nginx</code> 或 <code>systemctl start nginx</code>	<code>./bin/start-alb.sh</code> （含管控台： <code>./bin/start-alb-and-console.sh</code> ）
停止	<code>nginx -s quit</code> 或 <code>systemctl stop nginx</code>	<code>./bin/stop-alb.sh</code> （含管控台： <code>./bin/stop-alb-and-console.sh</code> ）
平滑重載	<code>nginx -s reload</code>	<code>./bin/reload-alb.sh</code> （或 <code>kill -HUP \$(cat logs/alb.pid)</code> ）

平滑升級二進位	<code>kill -USR2 + kill -WINCH/QUIT</code>	同名訊號完全相容，參考《遷移手冊》
主設定檔	<code>conf/nginx.conf</code>	<code>conf/alb.conf</code>
pid 檔案	<code>logs/nginx.pid</code>	<code>logs/alb.pid</code>
開機自動啟動	自編 systemd unit	<code>./utils/systemd/enable_startup.sh</code>
日誌切割	logrotate	<code>utils/logrotate</code> 工具或管控台日誌管理
執行狀態	<code>stub_status</code>	<code>stub_status</code> 相容，另支援 <code>vtS /status</code> （頁面/JSON/Prometheus）
視覺化管理	無	管控台 <code>http://IP:8887</code>

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

