



APUSIC
固若长城
睿比世界

安裝手冊

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 產品安裝
 - 2.1 安裝包說明
 - 2.1.1 安裝媒體
 - 2.2 單節點部署
 - 2.3 叢集部署
 - 2.4 Docker 部署
 - 2.4.1 安裝包說明
 - 2.4.2 安裝步驟
 - 2.5 K8s 部署
 - 2.5.1 準備工作
 - 2.5.2 配置檔案與授權管理
 - 2.5.3 部署 ALB 服務
 - 2.5.4 部署驗證
 - 2.5.5 停止與清理
- 3 高可用叢集搭建
 - 3.1 前提準備
 - 3.2 使用管控台配置高可用
 - 3.3 手工配置 ALB 高可用服務
 - 3.3.1 查看實體網卡名稱
 - 3.3.2 配置 aha
 - 3.3.2.1 主節點
 - 3.3.2.2 備節點
 - 3.3.3 註冊系統服務並啟動系統服務
 - 3.3.4 驗證 VIP 是否自動漂移測試步驟
 - 3.3.5 其他配置說明：
 - 3.4 使用 keepalived 實現高可用
 - 3.4.1 安裝 keepalived
 - 3.4.2 查看實體網卡名稱

- 3.4.3 配置 keepalived
- 3.4.4 主備節點均啟動 keepalived 服務
- 3.4.5 驗證 VIP 是否指向主節點
- 3.4.6 停止主節點 alb，驗證 VIP 是否漂移到備節點
- 4 ALB 授權與服務配置
 - 4.1 license 使用說明
 - 4.1.1 本地授權
 - 4.1.2 統一授權
 - 4.2 產品端口說明
 - 4.3 開機啟動項與系統服務
 - 4.3.1 移除系統啟動項
 - 4.4 普通使用者啟動 ALB 並且開啟監聽 80 端口

1 前言

本文檔為金蝶Apusic負載均衡器軟體 V2.0.6 標準版的安裝手冊，詳細介紹如何在各作業系統上安裝、部署和授權 ALB，以及 ALB 服務的啟停操作。

1.1 適用對象

本文檔適用於 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、配置、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 管控制台使用者手冊	詳細介紹 ALB 管控制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 效能調優的說明。

1.3 技術支援

ALB 產品提供全面的技術支援服務，您可以通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

2 產品安裝

2.1 安裝包說明

ALB 標準版安裝包名稱結構是：

ALB-版本號-SE-建構日期-CPU 架構.tar.gz 或 ALB-版本號-SE-建構日期-CPU 架構-ebpf.tar.gz

目前僅支援 Linux 平台的 arm64 與 amd64 架構：

- Linux-arm64 架構安裝包：ALB-V2.0.6-SE-建構日期-arm64.tar.gz
- Linux-amd64 (x86_64) 架構安裝包：ALB-V2.0.6-SE-建構日期-amd64.tar.gz
- Linux-arm64 架構 ebpf 特性安裝包：ALB-V2.0.6-SE-建構日期-arm64-ebpf.tar.gz
- Linux-amd64 (x86_64) 架構 ebpf 特性安裝包：ALB-V2.0.6-SE-建構日期-amd64-ebpf.tar.gz

註：建構日期為 ALB 產品包具體建構日期，同一個版本可能存在多個建構日期，取得安裝包以最新日期為準。本手冊全部示範內容中的安裝包名稱移除了建構日期。

註：ebpf 特性安裝包，Linux 核心版本不低於 4.19。

2.1.1 安裝媒體

- tar.gz 壓縮包：ALB-版本號-SE-建構日期-CPU 架構.tar.gz 解壓到目標機器任意路徑即可使用。
- Docker 映像檔：ALB-版本號-SE-Docker-建構日期-CPU 架構-docker.tar.gz 或 ALB-版本號-SE-Docker-建構日期-CPU 架構-基礎作業系統.tar.gz（基礎作業系統如 kylin、ubuntu 等，請以實際安裝包為準）。
- RPM 包：ALB-版本號-SE-建構日期-CPU 架構.rpm，使用 rpm -ivh 安裝包名 安裝，安裝完成後，其安裝路徑為 /opt/ALB-V2.0.6-SE-amd64（路徑中的版本號請以實際安裝包為準）。

2.2 單節點部署

- 取得軟體安裝包 ALB 安裝媒體為：ALB-V2.0.6-SE-amd64.tar.gz 或 ALB-V2.0.6-SE-arm64.tar.gz
- 複製安裝包到目標主機
- 解壓：ALB-V2.0.6-SE-amd64.tar.gz 到目標安裝路徑（範例為 /opt 目錄）

tar -zxvf ALB-V2.0.6-SE-amd64.tar.gz -C /opt/ 4. 將申請到的授權檔案放置在安裝目錄下，可參考本文檔後續章節：《ALB 授權與服務配置》。 5. ALB 使用參考《快速使用手冊》。

註：如果沒有權限，請使用 root 帳戶或 sudo。

2.3 叢集部署

在主備節點上分別執行單節點安裝步驟安裝完畢後，參考本文檔中的「高可用叢集搭建」章節進行配置即可。

2.4 Docker 部署

ALB 標準版提供 amd64 和 arm64 架構的 Docker 系統安裝包，可通過 ALB Docker 安裝包直接部署 ALB。

2.4.1 安裝包說明

1. ALB 壓縮安裝包名稱

- ALB-V2.0.6-SE-docker-amd64.tar.gz 或 ALB-V2.0.6-SE-Docker-20260421-amd64-kylin.tar.gz
- ALB-V2.0.6-SE-docker-arm64.tar.gz 或 ALB-V2.0.6-SE-Docker-20260421-arm64-kylin.tar.gz

註：安裝包日期可能不同，請根據實際情況選擇。

2. 安裝包目錄結構

```
tree -L 1 ALB-V2.0.6-SE-Docker-20260421-amd64-kylin
ALB-V2.0.6-SE-Docker-20260421-amd64-kylin      # docker 安裝包解壓後目錄
|-- ALB-V2.0.6-SE-Docker-20260421-amd64-kylin.tar.gz # docker 映像檔
|-- alb                                             # ALB 啟動配置檔案、授權檔案、日誌等掛載目錄
|   |-- alb.conf                                  # ALB 配置檔案
|   |-- license.xml                              # 授權檔案
-- alb-standard-dockercompose.yaml                # docker compose 配置檔案
```

2.4.2 安裝步驟

下列操作以 ALB-V2.0.6-SE-Docker-20260421-amd64-kylin.tar.gz 為例

1. 解壓和匯入映像檔

- 解壓 `tar -zxvf ALB-V2.0.6-SE-Docker-20260421-amd64-kylin.tar.gz`
- 進入解壓後的資料夾 `cd ALB-V2.0.6-SE-Docker-20260421-amd64-kylin`

- 載入映像檔：`docker load < ALB-V2.0.6-SE-Docker-20260421-amd64-kylin.tar.gz`

2. 修改授權檔案

ALB 授權檔案需要掛載到容器內部安裝目錄，目前範例是 `alb/license.xml`，在啟動容器前，請更換授權檔案。

授權檔案掛載說明

- `license.lic`: 本地舊版授權檔案
- `license.xml`: 本地授權檔案，支援舊授權和 KBC 授權（如果是 KBC 授權，把 KBC 授權內容複製寫入 `license.xml` 即可）
- `acls.properties`: 統一授權配置檔案，需要掛載到容器路徑：`/opt/ALB-V2.0.6-SE/acls.properties`

3. 映像檔啟動與停止

- 啟動命令：`docker compose -f alb-standard-dockercompose.yaml up -d`
- 停止命令：`docker compose -f alb-standard-dockercompose.yaml down`

4. `alb-standard-docker-compose.yaml` 配置說明

```
version: '3'
services:
  alb:
    image: harbor.apusic.com/apusic/alb:V2.0.6-se.20260421-kylin-
amd64 # ALB 映像檔
    ports:
      - 8080:8080 # ALB http 對外端口
      - 8887:8887 # ALB 管控台端口
    volumes:
      - ./alb/alb.conf:/opt/ALB-V2.0.6-SE/conf/alb.conf # alb 配
置檔案
      - ./alb/license.xml:/opt/ALB-V2.0.6-SE/license.xml # alb 授
權檔案
      - ./alb/logs:/opt/ALB-V2.0.6-SE/logs # alb 訪
問和錯誤日誌儲存目錄
    command: bash /opt/ALB-V2.0.6-SE/bin/start-alb-and-console.sh
```

2.5 K8s 部署

ALB 標準版支援透過 Kubernetes 進行部署，以下是詳細的部署步驟。

2.5.1 準備工作

- 環境要求
 - 已安裝 kubectl 並設定好 Kubernetes 叢集存取權限。
 - 若使用私有映像檔倉庫（如 Harbor），確保叢集有權限拉取映像檔。
- 映像檔準備
 - 將 Docker 映像檔上傳至映像檔倉庫（以 AMD64 映像檔為例）

解壓安裝包

```
tar -zxvf ALB-V2.0.6-SE-Docker-20260421-amd64-kylin.tar.gz
```

切換到解壓後的安裝包

```
cd ALB-V2.0.6-SE-Docker-20260421-amd64-kylin
```

匯入 ALB 映像檔

```
docker load < ALB-V2.0.6-SE-Docker-20260421-amd64-kylin.tar.gz
```

重新標記映像檔，請根據自己映像檔倉庫地址進行修改

```
docker tag [映像檔 ID] harbor.apusic.com/apusic/alb:V2.0.6-se.20260421-kylin-amd64
```

推送映像檔至映像檔倉庫

```
docker push harbor.apusic.com/apusic/alb:V2.0.6-se.20260421-kylin-amd64
```

2.5.2 配置檔案與授權管理

- 建立 ConfigMap 將 `alb.conf` 配置檔案掛載至容器：

```
kubectl create configmap alb-config --from-file=./alb/alb.conf
```

- 建立 Secret 將 `license.xml` 授權檔案以 Secret 形式儲存（敏感資訊建議使用 Secret）：

```
kubectl create secret generic alb-license --from-file=./alb/license.xml
```

2.5.3 部署 ALB 服務

1. 建立 Deployment 編寫 alb-deployment.yaml 檔案：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: alb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: alb
  template:
    metadata:
      labels:
        app: alb
    spec:
      containers:
        - name: alb
          image: harbor.apusic.com/apusic/alb:V2.0.6-se.20260421-kylin-amd64 # 請根據倉庫地址進行修改
          command: ["bash", "/opt/ALB-V2.0.6-SE/bin/start-alb-and-console.sh"]
          ports:
            - containerPort: 8080
            - containerPort: 8887
          volumeMounts:
            - name: alb-config
              mountPath: "/opt/ALB-V2.0.6-SE/conf/alb.conf"
              subPath: "alb.conf"
            - name: alb-license
              mountPath: "/opt/ALB-V2.0.6-SE/license.xml"
              subPath: "license.xml"
            - name: alb-logs
              mountPath: "/opt/ALB-V2.0.6-SE/logs"
```

```

volumes:
- name: alb-config
  configMap:
    name: alb-config
- name: alb-license
  secret:
    secretName: alb-license
- name: alb-logs
  hostPath:
    path: "/var/alb/logs"
    type: DirectoryOrCreate

```

參數說明

- image: 替換為實際映像檔地址。
- hostPath: 日誌目錄掛載路徑，可按需改為 PVC（如使用雲端儲存）。

2. 建立 Service

```

apiVersion: v1
kind: Service
metadata:
  name: alb-service
spec:
  type: NodePort
  selector:
    app: alb
  ports:
    - name: http
      port: 8080
      targetPort: 8080
      nodePort: 30080
    - name: console
      port: 8887
      targetPort: 8887
      nodePort: 30887

```

參數說明

- `nodePort` 預設範圍為 30000-32767，若省略端口則由系統自動分配。
- 若在雲端環境，可將 `type` 改為 `LoadBalancer` 直接取得外部 IP。

3. 套用配置

```
kubectl apply -f alb-deployment.yaml
kubectl apply -f alb-service.yaml
```

2.5.4 部署驗證

1. 檢查 Pod 狀態

```
kubectl get pods -l app=alb
```

2. 存取服務

- HTTP 服務: `http://<節點IP>:30080`
- 管控台: `http://<節點IP>:30887`

2.5.5 停止與清理

1. 刪除部署

```
kubectl delete -f alb-deployment.yaml -f alb-service.yaml
```

2. 清理配置

```
kubectl delete configmap alb-config
kubectl delete secret alb-license
```

3 高可用叢集搭建

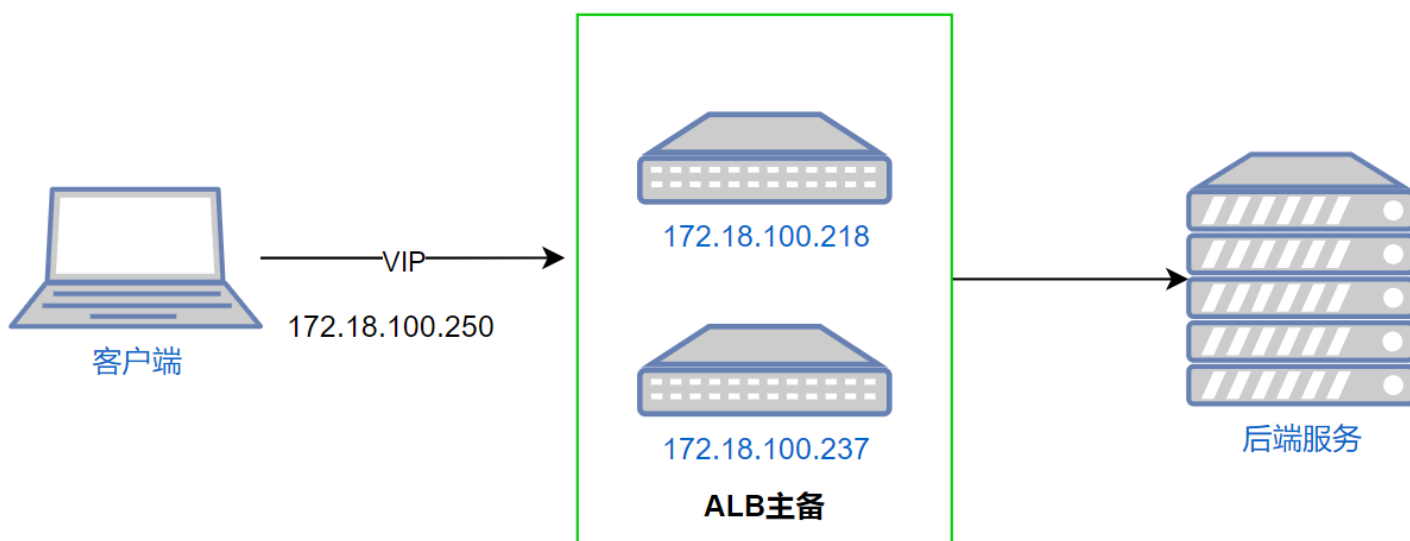
ALB 支援原生高可用和 keepalived 兩種高可用方式，搭建高可用叢集，當叢集節點發生故障時，自動將流量切換至備節點。其中原生高可用可透過 ALB 管控台配置，也可以透過命令列進行配置。

在 ALB 的管控台中一鍵配置和啟動高可用，下面介紹在命令列中的安裝配置方式。

3.1 前提準備

1. 兩台伺服器，一台作為主節點（Master），一台作為備用節點（Backup）。
2. 在兩台伺服器上部署 ALB 服務。（安裝過程忽略）
3. 在兩台伺服器的區域網路中，準備一個未使用的 IP 地址作為虛擬 IP（VIP）。
4. 確保兩台伺服器之間的網路通訊正常。
5. 確保兩台伺服器上的防火牆允許 keepalived 和 ALB 服務的通訊。

下面以兩台伺服器（192.168.2.10 和 192.168.2.11）為例，選擇 VIP 為 192.168.2.30。

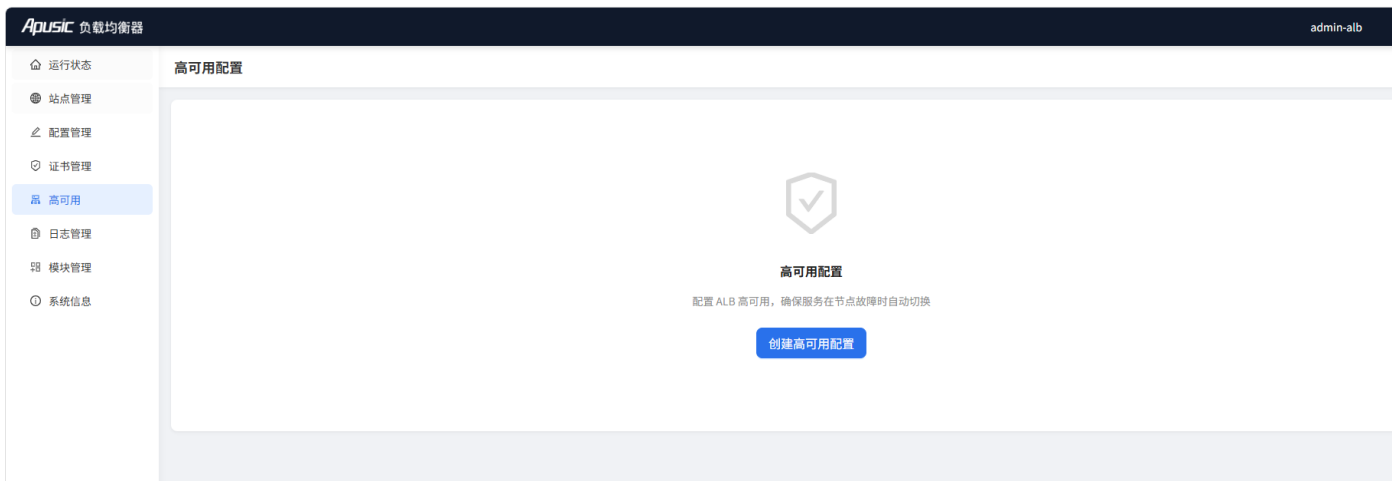


透過搭建兩套 ALB，並配置 VIP，實現用戶端透過 VIP 存取 ALB，當主節點宕機時，VIP 自動切換至備節點。

3.2 使用管控台配置高可用

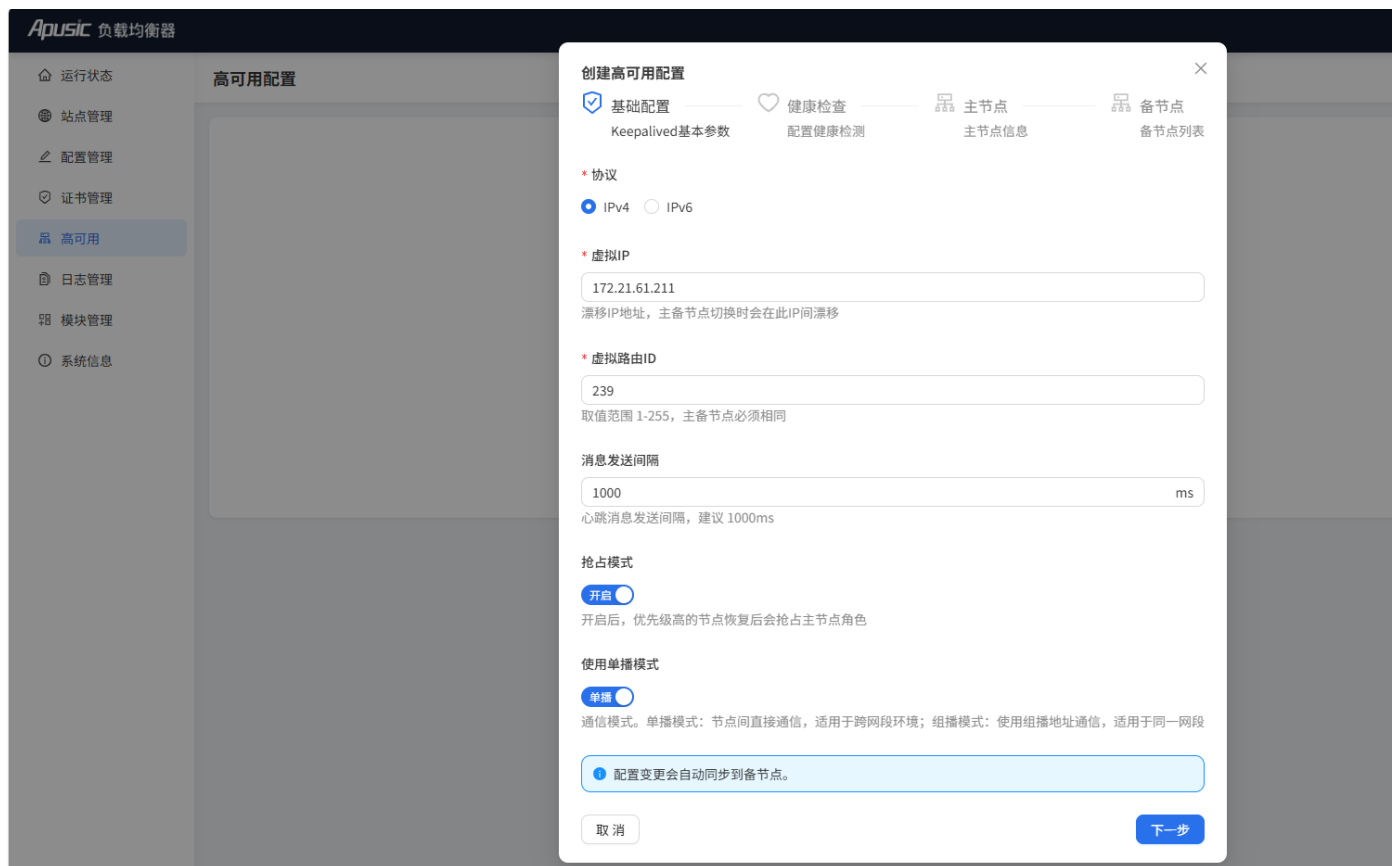
1. 登入 ALB 管控台，進入高可用配置頁面。

2. 點擊「高可用」

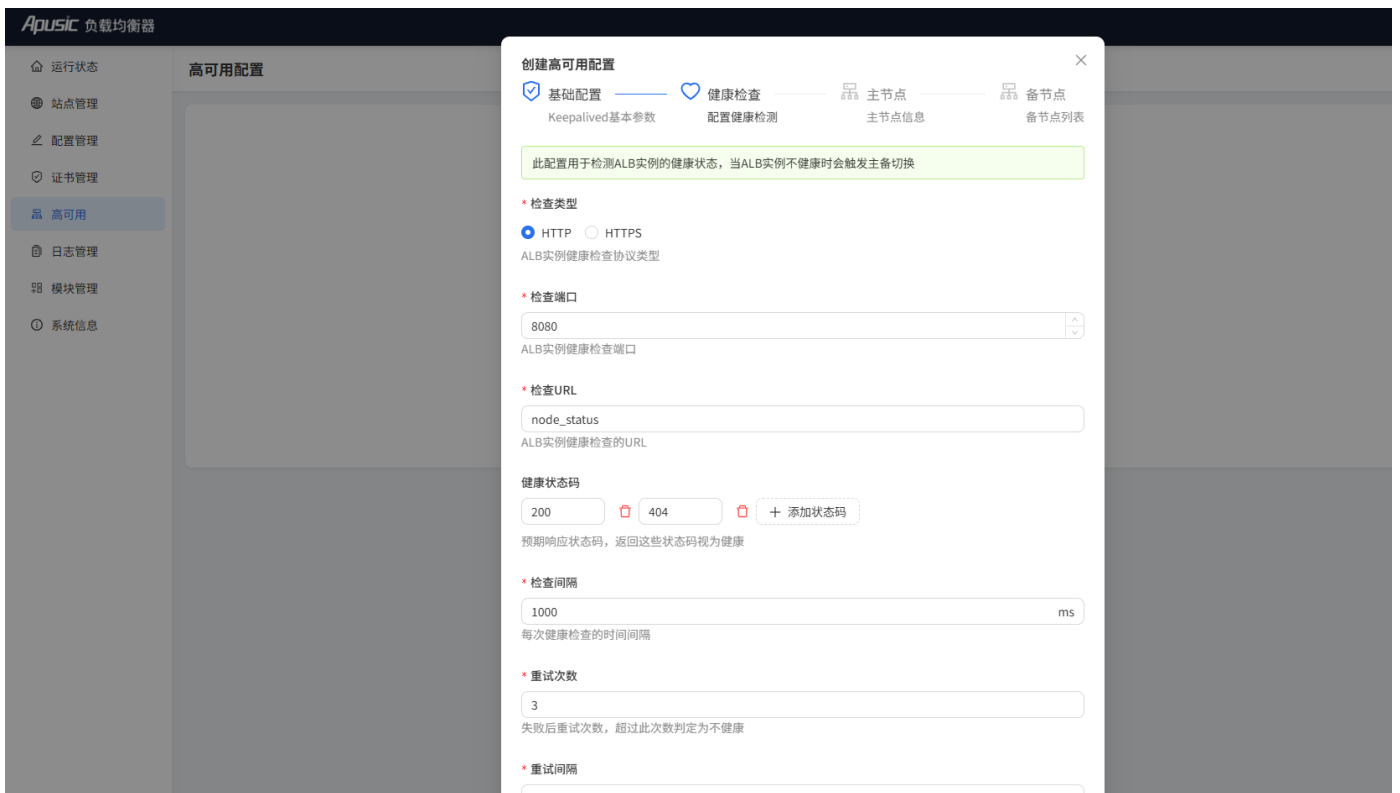


3. 點擊「建立高可用」

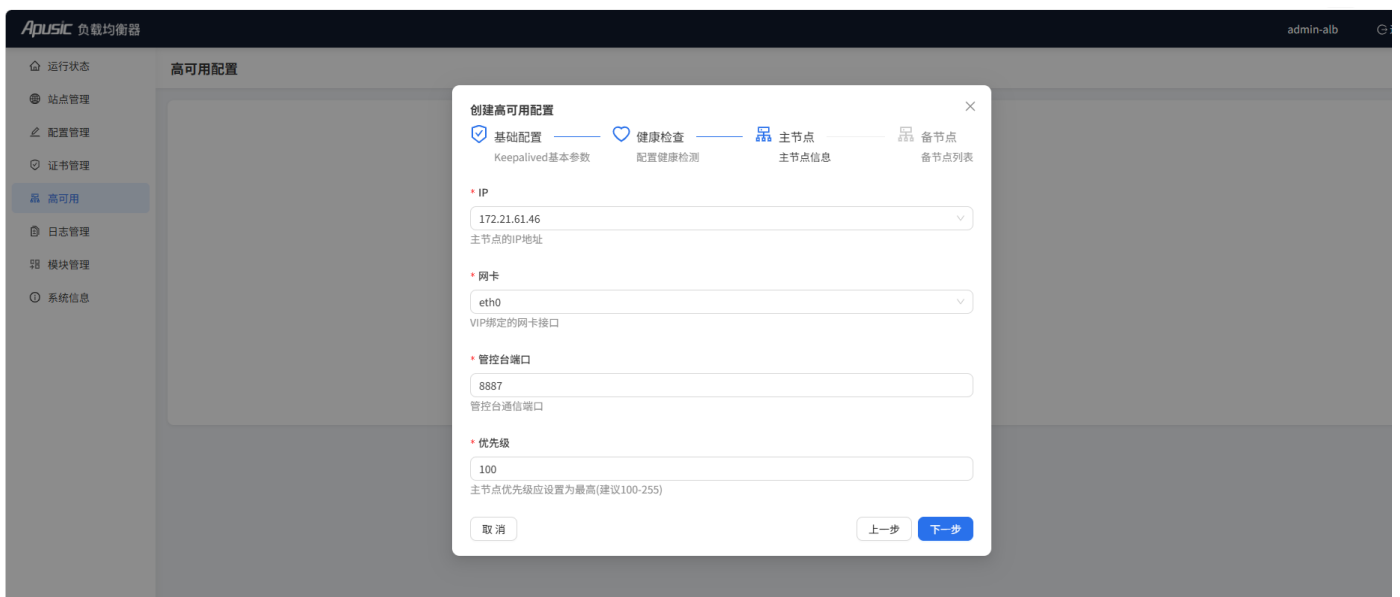
4. 填寫 VIP 地址，點擊下一步



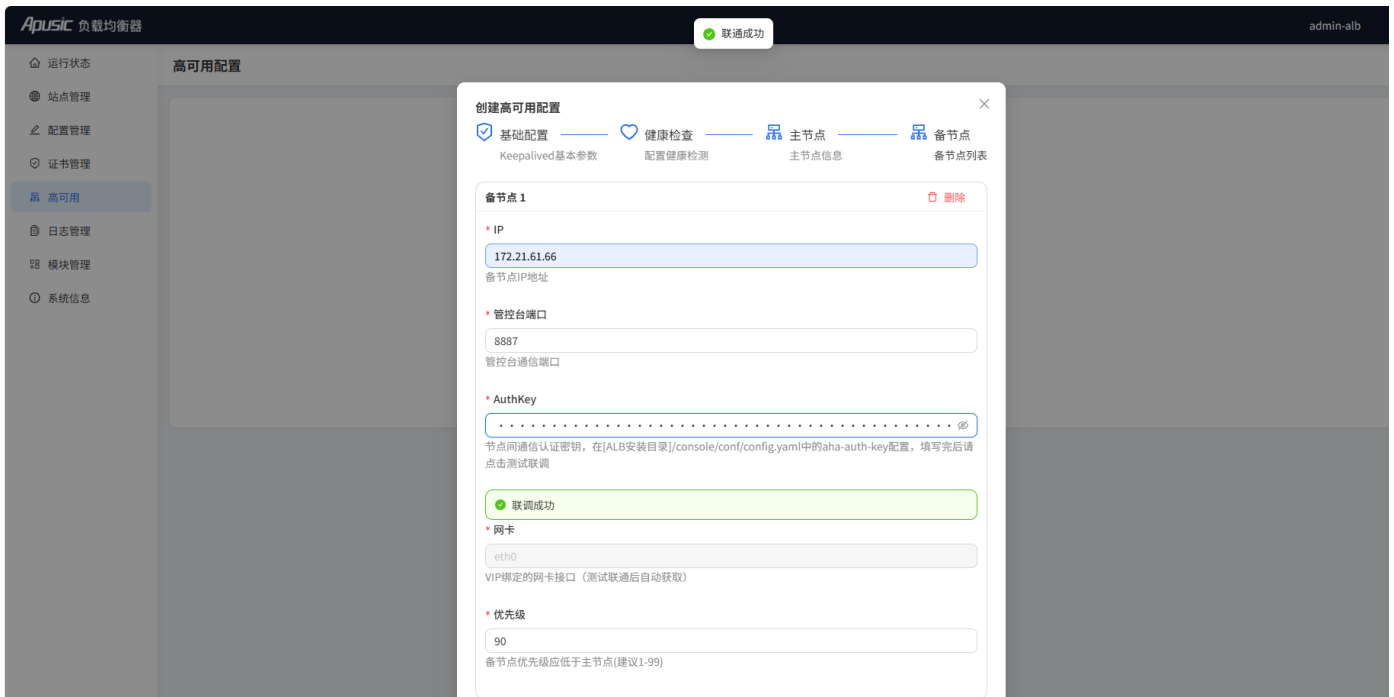
5. 填寫健康檢查內容（預設無需更改），點擊下一步



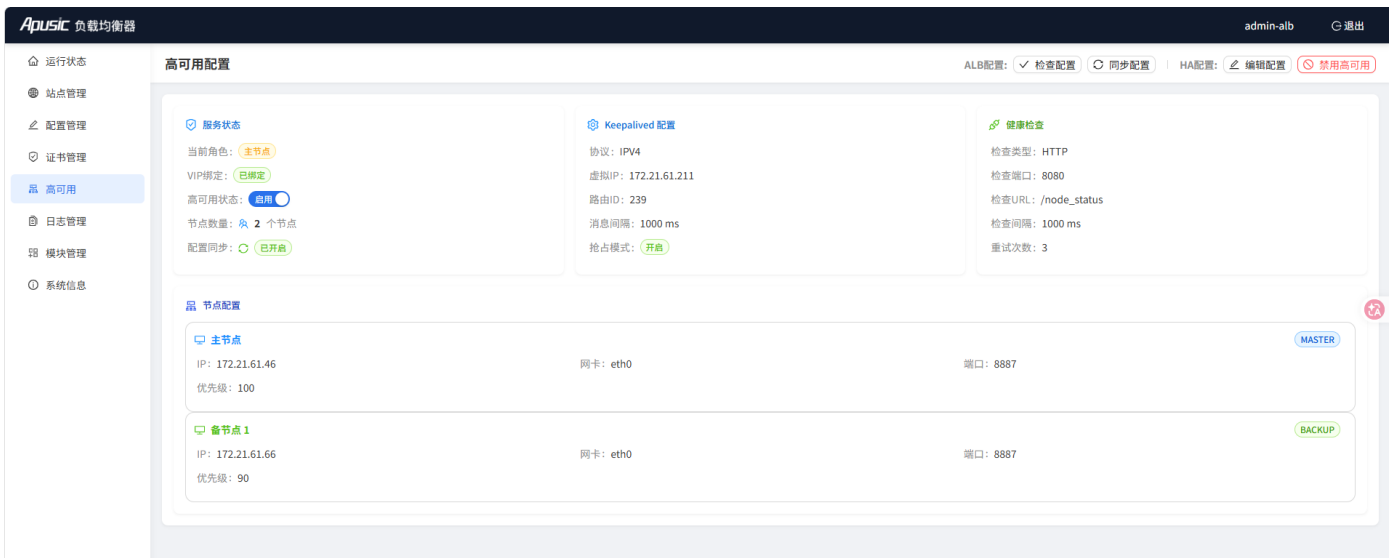
6. 選擇主節點 IP 和網卡，點擊下一步



7. 點擊「新增備節點」，輸入備節點的 IP、端口和 AuthKey，點擊「測試聯調」通過後點擊「建立」



8. 建立成功



3.3 手工配置 ALB 高可用服務

ALB 自帶高可用元件，可一鍵部署和啟動高可用服務。在 ALB 管控台中配置高可用時，預設使用自帶高可用元件。

3.3.1 查看實體網卡名稱

1. 在主節點和備用節點上執行以下命令，查看當前機器 IP 的實體網卡名稱：

```
ip addr show
```

2. 記錄實體網卡名稱，例如：eth0。

3.3.2 配置 aha

主節點

1. 角色配置為：master
2. 修改 VIP 地址
3. 修改網卡名稱

備節點

1. 角色配置為：backup
2. 修改 VIP 地址
3. 修改網卡名稱

3.3.2.1 主節點

1. 切換至 utils/aha 目錄，編輯 config.yaml 檔案，如下列範例

```
keepalived:
  # 虛擬路由 ID
  vroute_id: 239

  # 虛擬 IP 綁定網卡名稱
  interface: eth0

  # 虛擬 IP 地址
  vip: 192.168.2.30

  # 協議類型，可選值為 ipv4 和 ipv6
  protocol: ipv4

  # 是否搶佔模式
  # 如果啟用了搶佔模式 (Preempt Mode)，當一個更高優先級的路由器加入到 VRRP 組
  # 中時，它可以取代目前的活動路由器成為新的活動路由器。
  # 如果未啟用搶佔模式，則即使有更高優先級的路由器加入，目前的活動路由器也不會被
  # 取代
```

```
preempt: true

# 發送訊息間隔 (毫秒)
msg-send-interval: 800

# 優先級, 0-255
priority: 100

# 角色, 可選值為 master 和 backup
role: master

checker:

# 檢查服務的狀態時間間隔 (毫秒)
check-interval: 1000

# 重試次數
retry-count: 3

# 重試間隔 (毫秒)
retry-interval: 2000

# 健康檢查方式, 可以 url 或 shell, url 為 http 或 https, shell 為 shell
命令
health-type: url

# 目前目標服務狀態檢查 url, 用於檢查目標服務是否健康
health-url: http://localhost:8080/node_status

# 自訂健康檢查返回的 url 狀態碼
health-codes:
  - 200
  - 404

# shell 健康檢查命令, shell 命令執行返回 0 表示健康, 其他值表示異常
health-cmd:
```

```

# 當節點故障時候，是否嘗試重啟目標服務（預設為 false）
restart-on-failure: false
# 重啟目標服務命令
restart-cmd:

aha:
# 日誌檔案路徑
log-file: aha.log

# 日誌級別，可選值為 debug, info, warn, error
log-level: info

```

3.3.2.2 備節點

```

keepalived:
# 虛擬路由 ID
vroute_id: 239

# 虛擬 IP 綁定網卡名稱
interface: eth0

# 虛擬 IP 地址
vip: 192.168.2.30

# 協議類型，可選值為 ipv4 和 ipv6
protocol: ipv4

# 是否搶佔模式
# 如果啟用了搶佔模式（Preempt Mode），當一個更高優先級的路由器加入到 VRRP 組
# 中時，它可以取代目前的活動路由器成為新的活動路由器。
# 如果未啟用搶佔模式，則即使有更高優先級的路由器加入，目前的活動路由器也不會被
# 取代
preempt: true

# 發送訊息間隔（毫秒）

```

```
msg-send-interval: 800

# 優先級, 0-255 (備節點需低於主節點)
priority: 90

# 角色, 可選值為 master 和 backup
role: backup

checker:

# 檢查服務的狀態時間間隔 (毫秒)
check-interval: 1000

# 重試次數
retry-count: 3

# 重試間隔 (毫秒)
retry-interval: 2000

# 健康檢查方式, 可以 url 或 shell, url 為 http 或 https, shell 為 shell
命令
health-type: url

# 目前目標服務狀態檢查 url, 用於檢查目標服務是否健康
health-url: http://localhost:8080/node_status

# 自訂健康檢查返回的 url 狀態碼
health-codes:
  - 200
  - 404

# shell 健康檢查命令, shell 命令執行返回 0 表示健康, 其他值表示異常
health-cmd:

# 當節點故障時候, 是否嘗試重啟目標服務 (預設為 false)
restart-on-failure: false
```

```
# 重啟目標服務命令
restart-cmd:

aha:
# 日誌檔案路徑
log-file: aha.log

# 日誌級別, 可選值為 debug, info, warn, error
log-level: info
```

3.3.3 註冊系統服務並啟動系統服務

主備節點均需要執行以下操作：

1. 切換到安裝目錄下的 `utils/aha/sys-service` 目錄
2. 切換到 root 使用者, 或使用 `sudo` 執行下面命令
3. 執行: `./setup-aha-service.sh` 自動註冊系統服務
4. 啟動: `systemctl start aha.service`

3.3.4 驗證 VIP 是否自動漂移測試步驟

1. 使用 VIP 192.168.2.30 存取 ALB 正常。
2. 停掉主節點所在的 ALB, 繼續使用 VIP 存取 ALB, 驗證是否正常。

3.3.5 其他配置說明:

1. 自訂健康檢查 URL: 修改配置檔案中的 `health-url`: http://localhost:8080/node_status
2. 自訂 URL 的健康 HTTP 狀態碼: 修改配置中的: `health-codes`: 200, 404
3. 配置 `aha.log` 日誌檔案路徑: 修改配置檔案中的 `log-file`: `aha.log`
4. 透過自訂 shell 命令或腳本實現健康檢查, 要求返回 0 表示健康, 其餘值在 ALB 中標記不健康, 如下列範例:

```
# 使用 shell 命令進行健康檢查
health-type: shell
# 使用 shell 命令檢查 alb 進程存在則健康
health-cmd: ps aux | grep alb | grep -v grep
```

3.4 使用 keepalived 實現高可用

3.4.1 安裝 keepalived

```
# kylin、openEuler、CentOS 作業系統
yum install -y keepalived

# UOS、Ubuntu 作業系統
apt-get install -y keepalived
```

3.4.2 查看實體網卡名稱

1. 在主節點和備用節點上執行以下命令，查看實體網卡名稱：

```
ip addr show
```

2. 記錄實體網卡名稱，例如：eth0。

3.4.3 配置 keepalived

1. 編輯主節點（Master）的 keepalived.conf 檔案：

```
vi /etc/keepalived/keepalived.conf
```

在主節點 /etc/keepalived.conf 檔案中加入以下內容：

註：

- 把 `interface eth0` 替換為主節點的實體網卡名稱。
- 把 `192.168.2.30` 替換為準備使用的虛擬 IP 地址。

```
! Configuration File for keepalived

global_defs {
    router_id alb
}

vrrp_script chk_http_port {
    script "/etc/keepalived/checkalb.sh"
    interval 2 # (偵測腳本執行的間隔)
    weight 2
```

```

}

vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    # 主、備機的 virtual_router_id 必須相同
    priority 100
    # 主、備機取不同的優先級，主機值較大，備份機值較小
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111 # 範例值，生產環境請更換為強密碼
    }
    virtual_ipaddress {
        # 請根據需要，修改這個虛擬 IP 地址
        192.168.2.30
    }
    track_script {
        chk_http_port
    }
}

```

2. 編輯備用節點 (Backup) 的 keepalived.conf 檔案：

```
vi /etc/keepalived/keepalived.conf
```

在備用節點的 keepalived.conf 檔案中加入以下內容：

註：

- 把 `interface eth0` 替換為備用節點的實體網卡名稱。
- 把 `192.168.2.30` 替換為準備使用的虛擬 IP 地址。

```

! Configuration File for keepalived
global_defs {

```

```

    router_id alb
}
vrrp_script chk_http_port {
    script "/etc/keepalived/checkalb.sh"
    interval 2 # (偵測腳本執行的間隔)
    weight 2
}
vrrp_instance VI_1 {
    state BACKUP
    # state MASTER
    # 備份伺服器上將 MASTER 改為 BACKUP
    interface eth0
    virtual_router_id 51
    # 主、備機的 virtual_router_id 必須相同
    priority 90
    # 主、備機取不同的優先級，主機值較大，備份機值較小
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111 # 範例值，生產環境請更換為強密碼
    }
    virtual_ipaddress {
        192.168.2.30 # 請根據需要，修改這個虛擬 IP 地址
    }
    track_script {
        chk_http_port
    }
}

```

3. 在 兩個節點 建立偵測腳本 checkalb.sh:

```
vi /etc/keepalived/checkalb.sh
```

在 checkalb.sh 檔案中加入以下內容：

```
#!/bin/bash
# 偵測 ALB 主進程是否存活 (根據實際安裝路徑調整 PID 檔案路徑)
alb_pid=$(cat /opt/ALB-V2.0.6-SE-amd64/logs/alb.pid 2>/dev/null)
alb_count=0
if [ -n "$alb_pid" ]; then
    alb_count=$(ps -p "$alb_pid" --no-headers | wc -l)
fi
#1.判斷 ALB 是否存活, 如果不存活停止 keepalived
if [ $alb_count -eq 0 ];then
    sleep 3
    #2.等待 3 秒後再次獲取一次 alb 狀態
    alb_pid=$(cat /opt/ALB-V2.0.6-SE-amd64/logs/alb.pid 2>/dev/null)
    alb_count=0
    if [ -n "$alb_pid" ]; then
        alb_count=$(ps -p "$alb_pid" --no-headers | wc -l)
    fi
    #3.再次進行判斷, 如 alb 還不存活則停止 Keepalived, 讓地址進行漂移, 並退出
    腳本
    if [ $alb_count -eq 0 ];then
        echo "alb is down"
        systemctl stop keepalived
        exit 1
    fi
fi
```

在建立偵測腳本後, 給 checkalb.sh 加入執行權限:

```
chmod +x /etc/keepalived/checkalb.sh
```

3.4.4 主備節點均啟動 keepalived 服務

```
systemctl start keepalived.service
```

3.4.5 驗證 VIP 是否指向主節點

在主節點上查看虛擬 IP 地址（把 eth0 改為主節點網卡名稱，192.168.2.30 替換為實際 VIP）：

```
ip addr show eth0 | grep '192.168.2.30'
```

3.4.6 停止主節點 alb，驗證 VIP 是否漂移到備節點

```
# 切換到 alb 安裝目錄
./bin/stop-alb.sh
```

在備節點上查看虛擬 IP 地址（把 eth0 改為備節點網卡名稱）：

```
ip addr show eth0 |grep inet|grep -v inet6|awk '{print $2}'
```

如果 VIP 漂移到備節點，則說明 keepalived 配置成功。

4 ALB 授權與服務配置

4.1 license 使用說明

ALB 標準版支援本地授權（`license.lic` 或 `license.xml`）和統一授權（`acls.properties`），授權檔案需要放到產品安裝目錄下。

- `license.lic`：本地授權檔案（舊版授權格式）。
- `license.xml`：本地授權檔案（支援舊授權和 KBC 授權；如果是 KBC 授權，把 KBC 授權內容複製寫入 `license.xml` 即可）。
- `acls.properties`：統一授權配置檔案，需要放到 ALB 安裝目錄下。

4.1.1 本地授權

1. 取得本機特徵碼

```
# 取得本機特徵碼
./bin/alb-authcode

# 指定網卡取得特徵碼
./bin/alb-authcode -ac eth0

# 指定 IP 地址取得特徵碼
./bin/alb-authcode -ac 192.168.1.100
```

註：-ac 參數僅支援 alb 標準版建構日期大於等於 2025 年 6 月及以後版本。2. 取得特徵碼後，請與金蝶天燕聯繫，將特徵碼發回金蝶天燕進行授權檔案取得，或者登入金蝶天燕授權平台取得授權檔案。3. 將取得的授權檔案修改為檔名 `license.lic` 或 `license.xml`，然後放到 ALB 安裝目錄下。若使用統一授權，請將 `acls.properties` 放到 ALB 安裝目錄下。

4.1.2 統一授權

1. 統一授權中心必須包含 ALB 標準版的有效授權資訊。（請登入授權中心進行查看，如果沒有有效的產品授權資訊，請參考授權中心手冊，匯入有效的產品授權資訊。）
2. 產品配置連線授權中心的連線方式：
 - `acls.properties` 配置檔案：`acls.properties` 配置檔案放置在 ALB 安裝目錄下。

```

apusic_acls_enable=true # 開啟變數統一授權
apusic_acls_authUrls=192.168.4.10:6869 # 範例統一授權中心地址,
請替換為實際地址
apusic_acls_ns=public # 命名空間
apusic_acls_tenant=public # 租戶名稱

```

- 系統環境變數配置

```

export apusic_acls_enable=true # 開啟變數統一授權
export apusic_acls_authUrls=192.168.5.10:6869 # 範例統一授權中心
地址, 請替換為實際地址
export apusic_acls_ns=後付費 # 命名空間
export apusic_acls_tenant=user_env # 租戶名稱

```

註：2 種授權方式如果都進行了配置，則優先級為系統環境變數配置 > acls.properties 檔案配置；如果同時配置了本地授權方式和統一授權方式，則使用統一授權方式，若授權驗證失敗，將不會執行後續的授權驗證。

4.2 產品端口說明

ALB 預設端口如下：

- 8080：http 服務端口。
- 8887：ALB 管控台端口。

如需修改預設 http 代理端口，可以編輯 `ALB 安裝目錄/conf/alb.conf` 檔案中的 `listen 8080` 行。

如需修改管控台的端口，可以編輯 `ALB 安裝目錄/console/conf/config.yaml` 檔案中 system 中的 addr 欄位端口。

4.3 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

```

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（通常需要 root

```

權限)

```
systemctl start alb # 啟動 ALB
```

4.3.1 移除系統啟動項

```
cd /opt/ALB-V2.0.6-SE-amd64 # 進入安裝目錄
./utils/systemd/remove_startup.sh # 移除開機啟動項
```

4.4 普通使用者啟動 ALB 並且開啟監聽 80 端口

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 端口。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 端口：`setcap cap_net_bind_service=+eip ./sbin/alb`
3. 切換回 普通使用者

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

