



APUSIC
固若长城
睿比世界

ALB 极简运维手册

金蝶Apusic负载均衡器V2.0.5标准版

版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

目录

- 1 概述
- 2 产品简介
 - 2.1 什么是 ALB
 - 2.2 核心特性
 - 2.3 适用场景
 - 2.4 产品组成
- 3 部署架构
 - 3.1 单节点部署
 - 3.2 主备高可用部署（推荐生产环境）
- 4 快速安装
 - 4.1 系统要求
 - 4.2 端口规划
 - 4.3 单节点安装（5 分钟完成）
 - 4.4 Docker 快速部署
 - 4.5 主备高可用部署（2 节点）
- 5 常用命令速查
 - 5.1 进程管理
 - 5.2 授权相关
 - 5.3 配置反向代理（HTTP 七层）
 - 5.4 TCP/UDP 四层代理
 - 5.5 负载均衡算法
 - 5.6 HTTPS 配置
 - 5.7 限流限速配置
 - 5.8 健康检查
 - 5.9 灰度发布（按比例分流）
 - 5.10 静态资源服务
 - 5.11 WebSocket 代理
- 6 日常运维
 - 6.1 巡检清单
 - 6.2 核心配置文件
 - 6.3 故障处理
- 7 性能调优
 - 7.1 系统级优化
 - 7.2 ALB 高并发配置
 - 7.3 SSL 性能优化

- 8 监控指标
 - 8.1 关键告警指标
 - 8.2 监控方式
- 9 目录结构
- 10 技术支持

1 概述

本文档为金蝶Apusic负载均衡器（ALB） V2.0.5 标准版极简运维手册，涵盖部署架构、快速安装、常用命令、日常运维、性能调优及监控告警核心内容。

2 产品简介

2.1 什么是 ALB

金蝶天燕负载均衡器软件标准版（Apusic Load Balancer，ALB）是一款兼具轻量级、高性能及高可用性的负载均衡器软件，核心兼容 Nginx 配置语法。ALB 能有效应对大规模服务集群在面向客户端时的请求与流量管理挑战，承担请求安全控制、验证、过滤以及负载均衡和反向代理的重任。

2.2 核心特性

特性	说明
协议完全兼容	支持 HTTP/HTTPS/HTTP2/HTTP3、TCP、UDP、WebSocket、gRPC、FastCGI、邮件协议等
Nginx 配置兼容	可直接使用 Nginx 配置方式及内容进行配置，降低学习成本
信创适配	适配麒麟、统信等信创操作系统，兼容鲲鹏、龙芯、兆芯、海光等国产芯片
国密 SSL	支持国密 SSL 双向认证通信，满足政企合规要求
高可用架构	支持主备高可用集群部署，通过 VIP 实现故障自动转移
可视化管控	提供 Web 管控台，支持配置管理、证书管理、流量监控、高可用配置
Lua 扩展	支持 Lua 语言编写扩展插件，实现自定义流量处理逻辑

2.3 适用场景

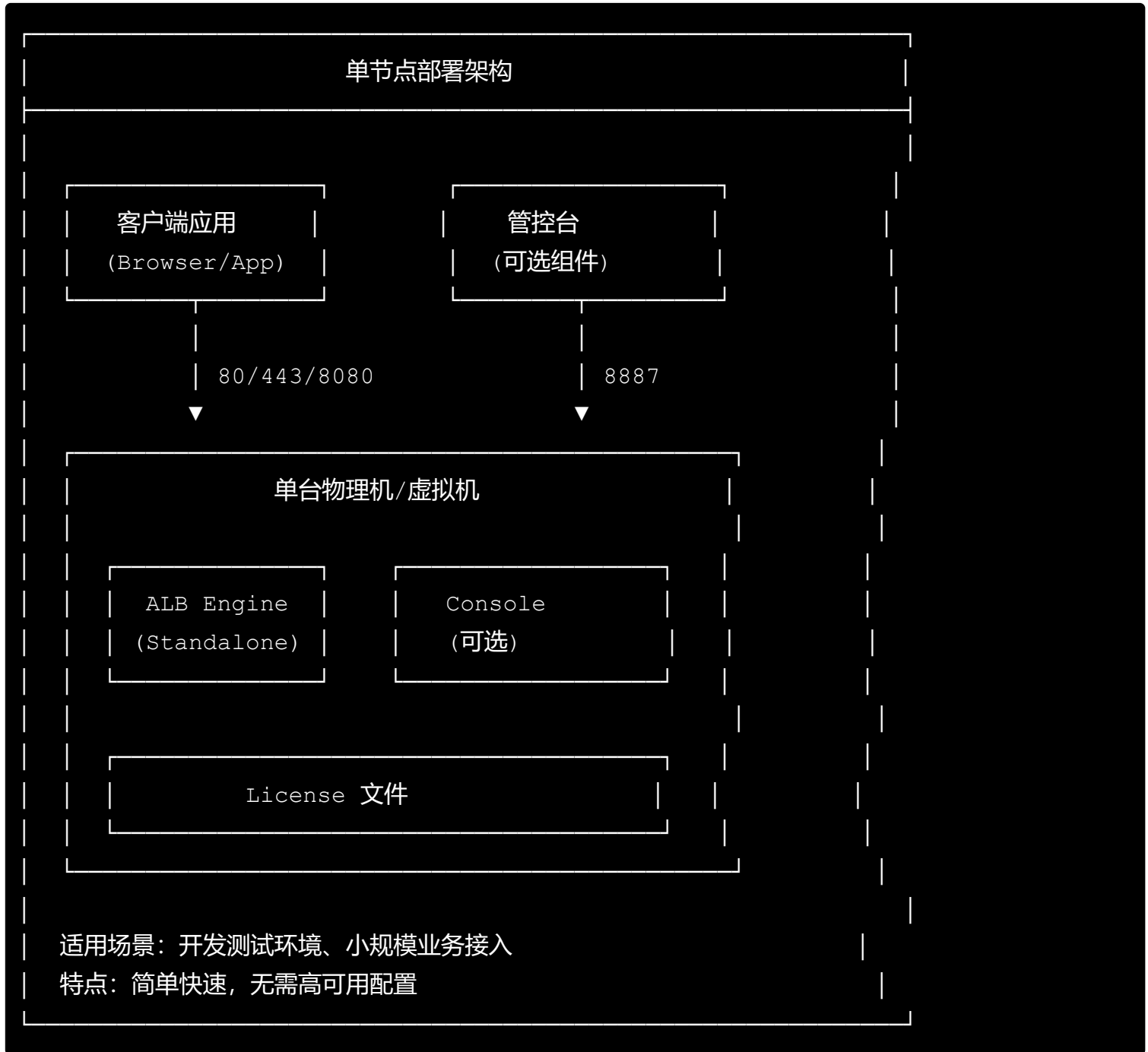
- **企业入口网关**：统一 API 入口、SSL 终止、安全控制、流量治理
- **静态资源发布**：静态文件缓存、压缩、SPA 路由支持
- **负载均衡**：四层/七层负载均衡，支持轮询、加权轮询、IP Hash、最小连接数、Sticky 会话粘滞
- **高可用代理**：微服务反向代理、WebSocket/AI 流式响应代理、TCP/UDP 代理
- **安全防护**：IP 黑白名单、限流限速、防盗链、敏感路径拦截

2.4 产品组成

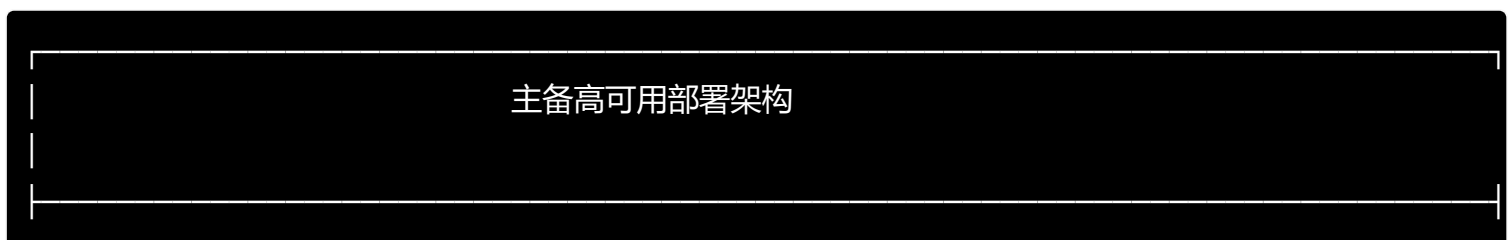
组件	说明	必需
ALB 核心引擎	请求处理与代理转发核心，受 License 控制	是
管控台	Web 可视化运维管理界面，支持三员权限	否（可选）
AHA 高可用组件	原生高可用组件，支持 VIP 漂移（可选）	否（可选）

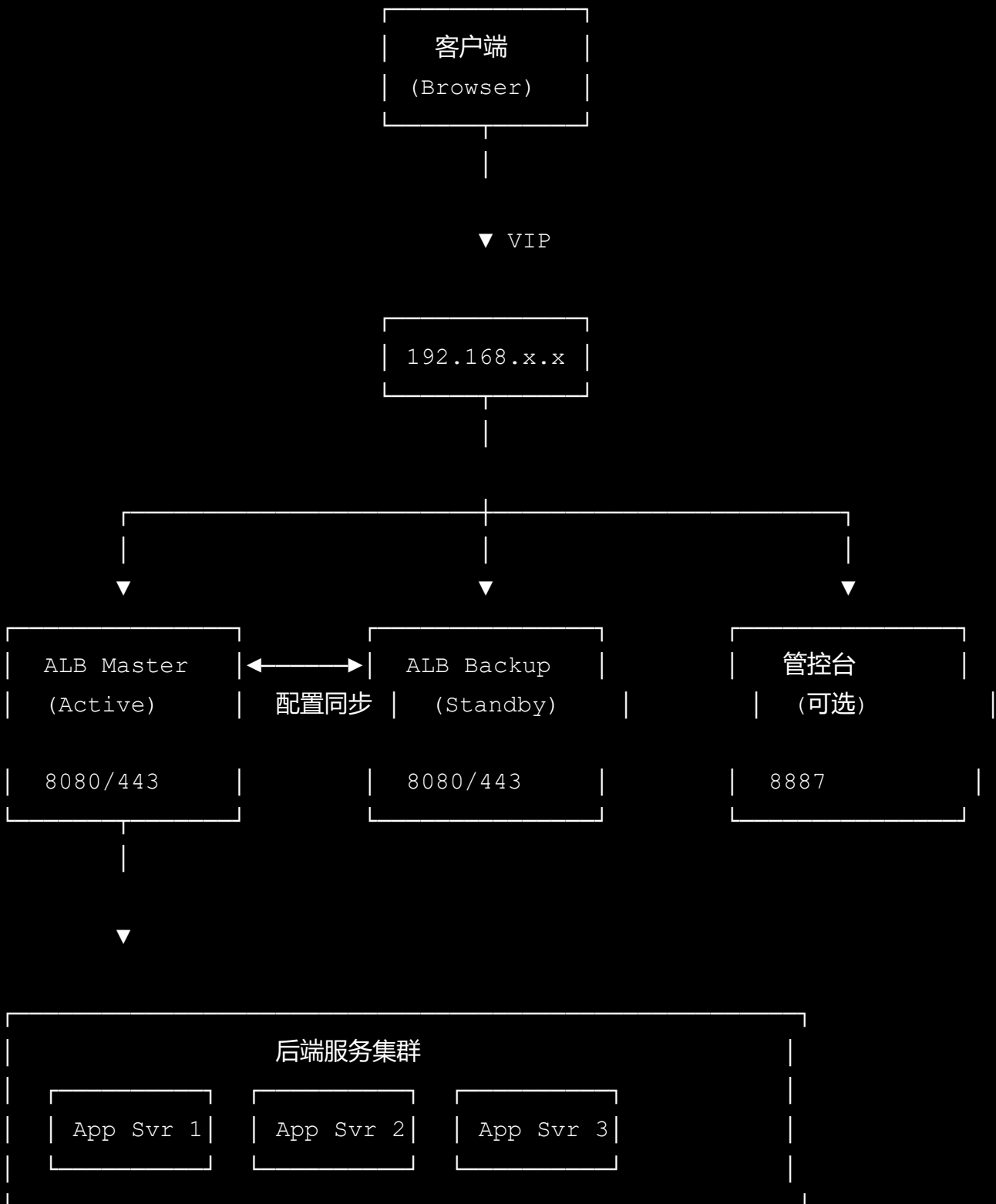
3 部署架构

3.1 单节点部署



3.2 主备高可用部署（推荐生产环境）





适用场景：生产环境、高可用业务场景

特点：VIP 自动漂移、故障秒级切换、配置主备同步

4 快速安装

4.1 系统要求

组件	最低要求	推荐配置
操作系统	Linux (CentOS 7+, Ubuntu 18.04+, 麒麟 V10, 统信 UOS)	-
CPU 架构	amd64 (x86_64) 或 arm64	-
CPU	2 核	4 核及以上
内存	4 GB	8 GB 及以上
磁盘空间	50 GB	200 GB 及以上 SSD

4.2 端口规划

端口	服务	说明
8080	ALB HTTP 服务	默认业务代理入口
443	ALB HTTPS 服务	SSL 业务入口
8887	管控台	Web 管理界面 (HTTP)
53/任意	TCP/UDP 代理	根据实际代理业务而定

4.3 单节点安装 (5 分钟完成)

```
# 1. 解压
cd /opt
tar -zxvf ALB-V2.0.5-SE-amd64.tar.gz
cd ALB-V2.0.5-SE-amd64

# 2. 放置 License
cp /path/to/license.xml ./

# 3. 验证配置 (可选)
./sbin/alb -t

# 4. 启动 ALB
./bin/start-alb.sh
```

5. 验证

```
curl http://localhost:8080
```

4.4 Docker 快速部署

1. 解压

```
tar -zxvf ALB-V2.0.5-SE-Docker-20250225-amd64-kylin.tar.gz
cd ALB-V2.0.5-SE-Docker-20250225-amd64-kylin
```

2. 替换授权文件

```
# 将 license.xml 放入 ./alb/license.xml
```

3. 启动

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

4. 访问

```
# ALB: http://宿主机IP:8080
```

```
# 管控台: http://宿主机IP:8887
```

4.5 主备高可用部署 (2 节点)

节点规划:

节点	IP	角色	优先级
Node 1	192.168.2.10	Master	100
Node 2	192.168.2.11	Backup	90
VIP	192.168.2.30	-	-

各节点单节点安装完成后, 编辑 AHA 配置:

```
cd /opt/ALB-V2.0.5-SE-amd64/utils/aha
vi config.yaml
```

主节点配置:

```
keepalived:
  vroute_id: 239
```

```
interface: eth0
vip: 192.168.2.30
protocol: ipv4
preempt: true
msg-send-interval: 800
priority: 100
role: master
checker:
  check-interval: 1000
  retry-count: 3
  retry-interval: 2000
  health-type: url
  health-url: http://localhost:8080/node_status
  health-codes:
    - 200
    - 404
  restart-on-failure: false
```

备节点配置：仅修改 `role: backup` 和 `priority: 90`。

注册并启动 AHA 服务（主备均执行）：

```
cd /opt/ALB-V2.0.5-SE-amd64/utils/sys-service
./setup-aha-service.sh
systemctl start aha.service
```

5 常用命令速查

5.1 进程管理

```
# 启动 ALB + 管控台
./bin/start-alb-and-console.sh

# 仅启动 ALB
./bin/start-alb.sh

# 停止 ALB + 管控台
./bin/stop-alb-and-console.sh

# 仅停止 ALB
./bin/stop-alb.sh

# 热重载配置（不中断连接）
./bin/reload-alb.sh

# 验证配置文件
./sbin/alb -t
./sbin/alb -t -c conf/alb.conf
```

5.2 授权相关

```
# 获取本机授权码
./bin/alb-authcode

# 指定网卡
./bin/alb-authcode -ac eth0

# 指定 IP
./bin/alb-authcode -ac 192.168.1.100
```

5.3 配置反向代理（HTTP 七层）

```
http {  
    upstream backend {  
        server 192.168.1.100:8080 weight=5;  
        server 192.168.1.101:8080 weight=5;  
    }  
  
    server {  
        listen 80;  
        server_name example.com;  
  
        location / {  
            proxy_pass http://backend/;  
            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;  
        }  
    }  
}
```

5.4 TCP/UDP 四层代理

```
stream {  
    upstream tcp_backend {  
        server 192.168.1.100:3306;  
        server 192.168.1.101:3306;  
    }  
  
    server {  
        listen 3306;  
        proxy_pass tcp_backend;  
    }  
  
    upstream udp_backend {  
        server 8.8.8.8:53;  
        server 114.114.114.114:53;  
    }  
}
```

```
server {
    listen 53 udp;
    proxy_pass udp_backend;
}
}
```

5.5 负载均衡算法

```
upstream backend {
    # 轮询（默认）
    server 192.168.1.100:8080;
    server 192.168.1.101:8080;

    # 加权轮询
    # server 192.168.1.100:8080 weight=10;
    # server 192.168.1.101:8080 weight=20;

    # IP Hash
    # ip_hash;

    # 最少连接
    # least_conn;

    # Sticky 会话粘滞
    # sticky;
}
```

5.6 HTTPS 配置

```
server {
    listen 443 ssl http2;
    server_name example.com;

    ssl_certificate      /opt/ALB-V2.0.5-SE-amd64/conf/cert/fullchain.pem;
    ssl_certificate_key  /opt/ALB-V2.0.5-SE-amd64/conf/cert/privkey.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
```

```

ssl_ciphers HIGH:!aNULL:!MD5;
ssl_prefer_server_ciphers on;
ssl_session_cache shared:SSL:10m;
ssl_session_timeout 10m;

location / {
    proxy_pass http://backend;
}

# HTTP 强制跳转 HTTPS
server {
    listen 80;
    server_name example.com;
    return 301 https://$host$request_uri;
}

```

5.7 限流限速配置

```

http {
    # 限流: 每个 IP 每秒 10 个请求
    limit_req_zone $binary_remote_addr zone=api:10m rate=10r/s;

    # 限并发: 每个 IP 最多 20 个连接
    limit_conn_zone $binary_remote_addr zone=addr:10m;

    server {
        listen 80;

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

        location /download/ {
            limit_rate 100k;
            proxy_pass http://backend;
        }
    }
}

```

```

    }
}
}

```

5.8 健康检查

```

upstream backend {
    server 192.168.1.100:8080;
    server 192.168.1.101:8080;

    # 主动健康检查
    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;
}

```

5.9 灰度发布（按比例分流）

```

split_clients "${remote_addr}AAA" $is_canary {
    5%      1;
    *       0;
}

map $is_canary $backend {
    1      canary;
    0      stable;
}

upstream stable { server 10.0.0.100:8080; }
upstream canary { server 10.0.0.200:8080; }

server {
    listen 80;
    location / {
        proxy_pass http://$backend;
        access_log logs/canary.log combined if=$is_canary;
    }
}

```

```
}
}
```

5.10 静态资源服务

```
server {
    listen 80;
    server_name static.example.com;
    root /data/www;
    index index.html;

    location / {
        try_files $uri $uri/ /index.html;
    }

    location ~* \.(js|css|png|jpg|svg|woff2)$ {
        expires 1y;
        add_header Cache-Control "public, immutable";
    }

    gzip on;
    gzip_vary on;
    gzip_min_length 1024;
    gzip_comp_level 6;
    gzip_types text/plain text/css application/json application/javascript;
}
```

5.11 WebSocket 代理

```
location /ws/ {
    proxy_pass http://websocket_backend;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_read_timeout 86400s;
}
```

6 日常运维

6.1 巡检清单

检查项	命令	正常指标
ALB 主进程	<code>ps -ef grep "alb: main process"</code>	进程存在
端口监听	<code>netstat -nltup grep -E "8080 443 8887"</code>	LISTEN 状态
配置文件语法	<code>./sbin/alb -t</code>	syntax is ok / test is successful
磁盘使用率	<code>df -h</code>	< 70%
访问日志	<code>tail -f logs/access.log</code>	无异常激增
错误日志	<code>tail -f logs/error.log</code>	warn/error 级别不持续增长
后端健康	<code>curl http://localhost:8080/status</code>	upstream 节点状态为 up
VIP 绑定	<code>ip addr show eth0</code>	VIP 在主节点

6.2 核心配置文件

主配置文件 (`conf/alb.conf`):

```
# 运行用户
user root;

# worker 进程数, 建议等于 CPU 核心数
worker_processes auto;

error_log logs/error.log warn;
pid logs/alb.pid;

events {
    worker_connections 10240;
    use epoll;
    multi_accept on;
}

http {
    include mime.types;
    default_type application/octet-stream;
```

```
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';

access_log logs/access.log main;

sendfile on;
tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 65;

# 性能优化
gzip on;
gzip_vary on;
gzip_min_length 1k;
gzip_comp_level 6;
gzip_types text/plain text/css application/json application/javascript;

server {
    listen 8080;
    server_name localhost;

    location / {
        root html;
        index index.html index.htm;
    }

    # 状态监控
    location /node_status {
        stub_status on;
        access_log off;
        allow 127.0.0.1;
    }
}
```

6.3 故障处理

故障现象	排查命令	解决方案
ALB 启动失败	<code>./sbin/alb -t</code> 查看错误日志	修正配置文件语法错误；检查 <code>user</code> 配置
<code>getgrnam("nobody") failed</code>	查看 <code>logs/error.log</code>	<code>conf/alb.conf</code> 中改为 <code>user root</code> ;
启动后终端卡住	检查配置	删除 <code>daemon off</code> ; 配置
访问 403	检查目录权限	将 <code>user</code> 改为有静态资源读取权限的用户
管控台非法 Host 请求	查看管控台日志	<code>console/conf/config.yaml</code> 中设置 <code>docker: true</code>
<code>no live upstreams</code>	查看 <code>logs/error.log</code>	增加 <code>proxy_connect/read/send_timeout</code> 或调整 <code>max_fails</code>
会话登录失败	检查负载均衡算法	使用 <code>ip_hash</code> 或 <code>sticky</code> 保持会话一致性
WebSocket 连接断开	检查 <code>location</code> 配置	配置 <code>Upgrade/Connection</code> 头并延长 <code>proxy_read_timeout</code>
AAS 登录状态丢失	抓包检查 Cookie	配置 <code>proxy_cookie_path / /;</code> 和 <code>proxy_pass_header Set-Cookie</code> ;

7 性能调优

7.1 系统级优化

```
# 文件描述符
ulimit -n 65535

# 网络参数
sysctl -w net.core.somaxconn=65535
sysctl -w net.ipv4.tcp_max_syn_backlog=65535
sysctl -w net.ipv4.ip_local_port_range="1024 65535"
sysctl -w net.ipv4.tcp_tw_reuse=1
sysctl -w net.ipv4.tcp_fin_timeout=30

# 禁用 swap
sysctl -w vm.swappiness=0
```

7.2 ALB 高并发配置

```
user root;
worker_processes auto;
worker_rlimit_nofile 100000;

events {
    worker_connections 10240;
    use epoll;
    multi_accept on;
}

http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;

    keepalive_timeout 60;
    keepalive_requests 10000;
```

```

# 压缩
gzip on;
gzip_vary on;
gzip_min_length 1k;
gzip_comp_level 5;
gzip_types text/plain text/css application/json application/javascript;

# 连接池
proxy_http_version 1.1;
proxy_set_header Connection "";

# 缓冲
proxy_buffering on;
proxy_buffer_size 4k;
proxy_buffers 8 4k;
proxy_busy_buffers_size 8k;
}

```

7.3 SSL 性能优化

```

server {
    listen 443 ssl http2;

    ssl_certificate /path/to/cert.pem;
    ssl_certificate_key /path/to/key.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers
ECDHE+AESGCM:DHE+AESGCM:AES256+EECDH:AES256+EDH:!aNULL:!MD5:!DSS;
    ssl_prefer_server_ciphers off;

    # 会话复用
    ssl_session_cache shared:SSL:50m;
    ssl_session_timeout 1d;
    ssl_session_tickets off;
}

```

8 监控指标

8.1 关键告警指标

指标	告警阈值	说明
ALB 进程存活	= 0	主进程不存在
5xx 错误率	> 1%	服务端错误比例
请求延迟 (P99)	> 500ms	高延迟告警
后端节点 down	> 0	upstream 中不健康节点
活跃连接数	> worker_connections * 80%	连接数接近上限
磁盘使用率	> 80%	日志盘或缓存盘不足
VIP 漂移异常	VIP 不在期望主节点	高可用异常

8.2 监控方式

1. 内置 stub_status（简单流量统计）

```
server {
    listen 8080;
    location /node_status {
        stub_status on;
        access_log off;
        allow 127.0.0.1;
    }
}
```

访问 `http://IP:8080/node_status` 可获取：

- Active connections
- accepts / handled / requests
- Reading / Writing / Waiting

2. VTS 精细化监控

```
http {
    vhost_traffic_status_zone;
```

```
server {  
    location /status {  
        vhost_traffic_status_display;  
        vhost_traffic_status_display_format html;  
    }  
}
```

访问地址：

- HTML: `http://IP:Port/status`
- JSON: `http://IP:Port/status/format/json`
- Prometheus: `http://IP:Port/status/format/prometheus`

3. Prometheus Exporter

```
cd utils/exporter  
./start-exporter.sh  
# 输入 ALB node_status 端口 (默认 8080)  
# 输入 exporter 监听端口 (默认 9113)
```

采集地址: `http://IP:9113/metrics`

Grafana 面板模板: `utils/grafana/alb-grafana.json`

9 目录结构

```

ALB-V2.0.5-SE-amd64/
├── bin/                                # 启动、停止脚本
│   ├── start-alb.sh
│   ├── start-alb-and-console.sh
│   ├── stop-alb.sh
│   ├── stop-alb-and-console.sh
│   ├── reload-alb.sh
│   └── alb-authcode
├── conf/                              # 配置文件
│   ├── alb.conf                      # 主配置文件
│   ├── mime.types
│   └── cert/                         # SSL 证书目录
├── console/                          # 管控台
│   ├── conf/config.yaml             # 管控台配置
│   ├── data/alb.db                 # 管控台数据库
│   └── cert/                       # 管控台证书
├── html/                             # 默认静态页面
├── logs/                             # 访问日志 / 错误日志
│   ├── access.log
│   ├── error.log
│   └── alb.pid
├── sbin/                             # ALB 二进制
│   └── alb
├── utils/                            # 工具脚本
│   ├── systemd/                   # 开机启动
│   ├── aha/                      # 高可用组件
│   ├── exporter/                 # Prometheus exporter
│   ├── grafana/                 # Grafana 面板
│   └── logrotate/               # 日志切割工具
└── license.xml                     # 授权文件

```

10 技术支持

- 电话：400-855-5800
- 邮箱：support@apusic.com
- 网址：www.apusic.com

全国统一服务热线
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年，前身为“金蝶中间件公司”，是金蝶集团旗下新一代软件基础云平台服务商，云计算国家标准制定企业，国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业，也是“两网一站四库十二金”国家重点工程的基础平台提供商，产品广泛应用于政府、军工、金融、能源等关键行业，累计服务客户总数超过10万家。



云计算国家标准制定企业
金蝶集团旗下基础软件企业
信息技术应用创新核心企业
官网：www.apusic.com

