

503錯誤處理方法



使用的是 K8s

vminsert有掛VirtualService、Service，vmalert的-remoteWrite.url是vminsert的VirtualService

這是出現在vmalert寫入alert到vminsert時出現的log，503，可是vminsert 有把alert寫進vmstorage

vmalert log出現這個，我找不到處理方法：

2025-08-01T16:56:03.749+0800

warn

VictoriaMetrics/app/ vmalert/remotewrite/client.go:256

attempt 1 to send

request failed: unexpected response code 503 for http://vm-vminsert.ifond.edg.f18.tsmc

.com/insert/0/prometheus/api/v1 /write. Response body "upstream connect error or disconnect/reset before headers. reset reason: connection termination"

(retriable: true)2025-08-01



已思考 1m 2s

- Istio 通常只有在進行 ALPN 協商時才會使用 HTTP/2，但 Pod 之間的連接可能會強制使用 mTLS 並依舊使用 HTTP/2。VictoriaMetrics 支援 HTTP/1.1，似乎也能處理 HTTP/2。如果 Envoy 終止 mTLS 並代理 HTTP/1.1，應該不會有問題，但錯誤提示表明 Envoy 可能無法建立或維持連線。這可能與 upstream 的閒置超時或 Pod 重啟等因素有關。
- vmalert 配置成推送警報規則的時間序列到 /write，若 vminsert 將資料寫入 vmstorage 時即使遇到 503 錯誤，最終可能會因重試成功。確認 vmalert 的 remoteWrite.url 是否指向穩定的 vminsert 集群 DNS，並建議使用 K8s ClusterIP 服務而非 VirtualService 主機。
- 在 VirtualService 中設置重新嘗試和超時很重要，例如將重試次數設為 3 次，每次嘗試時間限制為 30 秒，並設置對 5xx 錯誤等進行重試。此外，檢



查 vmalet 的 remoteWrite 設定，特別是傳輸大小和超時設置，可能需要進一步調整。

- 檢查 vminsert 的 Pod 中斷預算，確保 HPA 沒有終止 Pod。此外，對於 Istio sidecar 資源，設定 `proxy.istio.io/config` 來延長 `holdApplicationUntilProxyStarts` 和 `terminationDrainDuration`，並確定 livenessProbe 有啟用 grpc 健康檢查。檢查 API 路徑並確認端口與協議。
- 檢查連線問題，確認是否為 Envoy 端的 503 錯誤，而且了解是否與 mTLS、健康檢查或網路超時有關。建議繞過 Istio 使用 ClusterIP 服務，或配置正確的 VirtualService 及 DestinationRule 來解決過多重試、超時或連線池相關問題。

🕒 完成

先幫你拆解這個 503 訊息的意義，再給一套「確認 → 立即止血 → 穩定化」的做法與可直接套用的 YAML。

發生了什麼？

```
`upstream connect error or disconnect/reset before headers. reset reason: connection termination`
```

這是 **Envoy/Istio** 在還沒拿到上游（vminsert）的 **HTTP header** 前就被關連線的典型訊息。常見原因：

- vminsert Pod 短暫重啟 / OOM / readiness 轉不過來 → 連線被關。
- 透過 **VirtualService** 走 sidecar 時的 **mTLS/協定 (HTTP/1.1 vs h2c) /timeout/連線池** 設定不合或太保守。
- 連線空間被回收（idle timeout / keepalive 不對齊）或 rolling restart/drain 期切到關閉中的 Pod。
- URL 或 DNS 解析偶發問題（含 **URL 末端的空白字元** 等）。

“為什麼資料仍寫進 vmstorage？”

vmalet 會重試 (retriable: true)，所以有些請求失敗、但後續重試成功，就會看到「有寫進去，但仍有 503 warn」。

先做 6 項快速檢查（5 分鐘可抓到 80% 問題）

1. 確認 URL 是否乾淨

你的 log 斷行看起來像：`.../api/v1 /write`（中間多了空白）。請確定 `remoteWrite.url` 沒有多餘空白/換行。

小技巧（在 vmaalert Pod 內）：

```
echo -n "$REMOTEWRITE_URL" | hexdump -C
```

結尾若不是 `... 2f 77 72 69 74 65`（/write）就要清掉奇怪字元。

2. 就近用 ClusterIP 直連測試（先暫避 VirtualService）

從 vmaalert Pod 內做：

```
nc -vz vminsert.<ns>.svc.cluster.local 8480
curl -sS -v http://vminsert.<ns>.svc.cluster.local:8480/health
```

- 成功：表示 vminsert 活著，先把 `remoteWrite.url` 改走 ClusterIP（止血法，見下）。
- 失敗：優先排查 vminsert 的 Pod/Service/Endpoints。

3. 看 vminsert 是否在重啟或被 OOMKill

```
kubectl -n <ns> get pod -l app=vminsert -o wide
kubectl -n <ns> describe pod -l app=vminsert | egrep -i "Restart|OOM"
```

4. 確定 Service port 命名正確

- Istio 需靠 **port name** 判斷協定。vminsert HTTP 入口的 Service port 請命名成 `http`（例如 `name: http-8480`）。命名成 `tcp-*` 會讓 VS 規則吃不到或走錯 filter chain。

5. 檢查 Istio 端點視圖（看是否有壞 endpoint 或連線失敗）

```
istioctl pc endpoints <vmalert-pod> -n <ns> --cluster
"outbound|8480||vminsert.<ns>.svc.cluster.local"
istioctl pc clusters <vmalert-pod> -n <ns> | grep vminsert
istioctl pc stats <vmalert-pod> -n <ns> --pattern ".*vminsert.*
(upstream_cx_connect_fail|upstream_rq_5xx|upstream_cx_destroy_remote)"
```

6. mTLS 模式一致性

- vminsert 若 在 **mesh**：DestinationRule 要 `tls.mode: ISTIO_MUTUAL`。
- vminsert 若 不在 **mesh**：請 `tls.mode: DISABLE`（避免加密到一個不會講 mTLS 的端點）。

立即止血（最簡單、最穩定的路徑）

把 **vmalert** 直接打到 **vminsert** 的 **ClusterIP Service**，先繞過 VirtualService / Gateway 的不確定因素：

```
-remoteWrite.url=http://vminsert.
<ns>.svc.cluster.local:8480/insert/0/prometheus/api/v1/write
```

“兩個重點：”

1. “**Service 名稱 + .svc.cluster.local + port 8480**（依你的部署調整）”
2. “**確認 URL 沒有多餘空白**（真的很常中）”

這樣通常就會把 503 降到幾乎沒有。

若你必須經過 Istio (VirtualService/DestinationRule) — 推薦設定

“目標：放寬 *timeout*、避免 *h2* 升級問題、加強連線池與重試，並避免把流量打到正在熔斷/重啟中的 *Pod*。”

VirtualService (重試與超時)

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: vminsert
  namespace: <ns>
spec:
  hosts:
    - vm-vminsert.ifond.edg.f18.tsmc.com # 你的 FQDN
  http:
    - route:
        - destination:
            host: vminsert.<ns>.svc.cluster.local
            port:
                number: 8480
      retries:
        attempts: 3
        perTryTimeout: 10s
        retryOn: 5xx,connect-failure,refused-stream,reset,gateway-error,stream-
reset
      timeout: 30s
```

DestinationRule (連線池、outlier、TLS、禁升級 h2)

```

apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  name: vminsert
  namespace: <ns>
spec:
  host: vminsert.<ns>.svc.cluster.local
  trafficPolicy:
    # 若 vminsert 在 mesh : 用 ISTIO_MUTUAL ; 不在 mesh : 改 DISABLE
    tls:
      mode: ISTIO_MUTUAL
    connectionPool:
      tcp:
        maxConnections: 2000
        connectTimeout: 5s
        tcpKeepalive:
          time: 7200s
          interval: 75s
      http:
        # 避免升級到 HTTP/2 (部分後端/中間層對 h2c 行為不一致)
        h2UpgradePolicy: DO_NOT_UPGRADE
        http1MaxPendingRequests: 10000
        maxRequestsPerConnection: 0      # 0 = 不限制, 避免頻繁關連線
        idleTimeout: 120s                # 放寬空閒連線時間
    outlierDetection:
      consecutive5xx: 20
      interval: 5s
      baseEjectionTime: 30s
    loadBalancer:
      simple: ROUND_ROBIN

```

“另外：**Service port** 命名請用 `name: http-8480` (或至少 `http` 開頭)。”

vmalert 端可以再加的穩定化（可選）



- **送出逾時**（例如 30s）與**佇列大小**（避免尖峰時丟包/大量重試）適度放寬。
（參數名稱依你使用的 vmaalert 版本，通常有 send timeout / queue / batch 類型設定。原則：稍微放寬，但別無限大）
- 若你看到大量 intermittent 503，但最終都有寫入，可先把 log level 維持預設，只要頻率不高就算「可接受抖動」。

故障排除決策樹（簡版）

1. **ClusterIP 直連 OK** → 問題多半在 Istio 路由/策略 → 套用上面 VS/DR。
2. **ClusterIP 直連也偶發失敗** → 看 vminsert Pod（重啟/OOM/CPU 飆高）、節點壓力、PDB/HPA。
3. **只在滾動更新或證書輪替時發生** → 增加 `terminationDrainDuration``、調整 PDB，並避免把流量打到 draining 中的 Pod。
4. **只在閒置後第一發失敗** → 提高 `idleTimeout``、啟用 `tcpKeepalive``、增大 `connectTimeout``。

小結

- 你的訊息是 **Envoy 在還沒拿到 header 前連線就被關**，與 vminsert 寫入成功不矛盾，因為重試後成功。
- 最快的止血是 **把 vmaalert 直接指到 vminsert 的 ClusterIP**（確認 URL 乾淨、port 名稱為 http）。
- 若一定要走 Istio，按上面的 **VS/DR 範本** 去放寬 timeout、禁 h2 升級、加大連線池與 outlier 設定，通常可以把這種 503 幾乎消掉。

如果你願意，把目前的 **VirtualService / DestinationRule / Service YAML**（以及 `remoteWrite.url`` 的實際值）貼上來，我可以直接幫你改成可用的版本。