



Docker Compose 建立集群



以繁體中文詳細說明用docker-compose建立cluster VictoriaMetrics



在使用 Docker Compose 建立一個 VictoriaMetrics 的 Cluster 時，您可以利用官方文件或自定義配置來實現。以下是詳細的步驟說明：

1. 準備環境

- 安裝 **Docker** 與 **Docker Compose** :
 - 安裝 Docker : [官方安裝指引](#)
 - 安裝 Docker Compose : [官方安裝指引](#)
- 建立專案目錄：

```
mkdir victoria-metrics-cluster
cd victoria-metrics-cluster
```

2. 撰寫 `docker-compose.yml` 檔案

在專案目錄下建立 `docker-compose.yml`，以下是一個 VictoriaMetrics Cluster 的範例配置。

```
version: '3.8'
services:
  vmselect:
    image: victoriametrics/vmselect:latest
    container_name: vmselect
    ports:
      - "8481:8481" # 對外服務的 HTTP API
    command:
      - '-storageNode=vmstorage:8401' # 指定 vmstorage 節點
    depends_on:
      - vmstorage
    networks:
      - victoria-network

  vminsert:
    image: victoriametrics/vminsert:latest
    container_name: vminsert
    ports:
      - "8480:8480" # 對外服務的 HTTP API
    command:
      - '-storageNode=vmstorage:8400' # 指定 vmstorage 節點
    depends_on:
      - vmstorage
```





```
networks:
  - victoria-network

vmstorage:
  image: victoriametrics/vmstorage:latest
  container_name: vmstorage
  ports:
    - "8400:8400" # 接收 vminsert 的數據
    - "8401:8401" # 提供 vmselect 查詢服務
  volumes:
    - ./data:/storage # 儲存數據到本地
  networks:
    - victoria-network

networks:
  victoria-network:
    driver: bridge
```

3. 配置文件詳解

- `\vmselect`` :
 - 提供數據查詢功能。
 - `\-storageNode`` : 指向 `\vmstorage`` 服務的地址和埠 (`\vmstorage:8401``)。
 - 映射本地埠 `\8481``，用於對外提供 HTTP API。
- `\vminsert`` :
 - 負責接收數據並寫入到 `\vmstorage``。
 - `\-storageNode`` : 指向 `\vmstorage`` 服務的地址和埠 (`\vmstorage:8400``)。
 - 映射本地埠 `\8480``，用於對外提供 HTTP API。
- `\vmstorage`` :
 - 負責數據存儲和處理。
 - 本地掛載資料夾 `\./data`` 用於持久化數據。
 - 提供 `\8400`` 和 `\8401`` 兩個埠，用於數據寫入和查詢。

4. 啟動 VictoriaMetrics Cluster

執行以下指令以啟動服務：

```
docker-compose up -d
```

檢查服務是否成功啟動：

```
docker-compose ps
```



5. 驗證與測試

- **驗證 vminsert (寫入數據)**：使用 `curl` 將測試數據寫入：

```
curl -X POST -d 'example_metric{label="test"} 123' http://localhost:8480/vminsert/0/prometheus/api/v1/write
```

- **驗證 vmselect (查詢數據)**：使用 `curl` 查詢數據：

```
curl 'http://localhost:8481/select/0/prometheus/api/v1/query_range?query=example_metric&start=1672531200&end=1672617600'
```

6. 停止與清理

- 停止服務：

```
docker-compose down
```

- 停止並刪除數據：

```
docker-compose down -v
```

7. 擴展與監控

- **節點擴展**：

- 增加 `vmselect` 或 `vminsert` 節點以應對更多負載。
- 修改配置文件，添加新的服務並設定對應的埠。

- **監控**：

- VictoriaMetrics 提供內建的 `/metrics` 頁面，可通過 `curl` 或 Prometheus 拉取監控數據。

以上配置可幫助您快速建立一個 VictoriaMetrics Cluster。如果有其他需求（例如高可用性、備份等），可以根據官方文檔進一步調整。



以繁體中文詳細說明作法



以下是以繁體中文詳細說明如何使用 Docker Compose 建立一個 VictoriaMetrics Cluster 的完整過程，包括每一步的操作和解釋。





1. 安裝必要工具

安裝 Docker

1. 開啟終端機 (Terminal)。
2. 執行以下指令以安裝 Docker：

- 對於 Ubuntu：

```
sudo apt update
sudo apt install -y docker.io
```

- 對於 macOS 或 Windows，請參考官方文檔：[Docker Desktop](#)。

3. 確保 Docker 已正確安裝：

```
docker --version
```

安裝 Docker Compose

1. 執行以下指令以安裝 Docker Compose：

- 對於 Linux：

```
sudo curl -L "https://github.com/docker/compose/releases/download/v2.2.0/docker-compose-$(uname -s)-$(uname -m)" -i
sudo chmod +x /usr/local/bin/docker-compose
```

- 對於 macOS 或 Windows，Docker Compose 通常與 Docker Desktop 一起安裝。

2. 確保 Docker Compose 已正確安裝：

```
docker-compose --version
```

2. 建立專案目錄

1. 在終端機中建立一個目錄來存放專案檔案，例如：

```
mkdir victoria-metrics-cluster
cd victoria-metrics-cluster
```

2. 在該目錄中，創建一個 Docker Compose 檔案：

```
touch docker-compose.yml
```

3. 撰寫 `docker-compose.yml` 檔案



打開您喜歡的編輯器（例如 `vim`、`nano` 或 Visual Studio Code），編輯 `docker-compose.yml` 檔案，並貼上以下內容：

```
version: '3.8'
services:
  vmselect:
    image: victoriametrics/vmselect:latest
    container_name: vmselect
    ports:
      - "8481:8481" # 提供 HTTP API 查詢接口
    command:
      - '-storageNode=vmstorage:8401' # 指定存儲節點
    depends_on:
      - vmstorage
    networks:
      - victoria-network

  vminsert:
    image: victoriametrics/vminsert:latest
    container_name: vminsert
    ports:
      - "8480:8480" # 提供 HTTP API 寫入接口
    command:
      - '-storageNode=vmstorage:8400' # 指定存儲節點
    depends_on:
      - vmstorage
    networks:
      - victoria-network

  vmstorage:
    image: victoriametrics/vmstorage:latest
    container_name: vmstorage
    ports:
      - "8400:8400" # 接收寫入數據
      - "8401:8401" # 提供查詢服務
    volumes:
      - ./data:/storage # 本地掛載目錄，用於數據持久化
    networks:
      - victoria-network

networks:
  victoria-network:
    driver: bridge
```

配置解釋

1. **vmselect**：

- 處理查詢的節點，從 `vmstorage` 中檢索數據。
- 埠 `8481` 用於接收查詢請求。

2. **vminsert**：

- 處理數據寫入的節點，將數據存入 `vmstorage`。
- 埠 `8480` 用於接收寫入請求。

3. **vmstorage**：

- 實際存儲數據的節點。





- 持久化掛載目錄 `./data:/storage`，確保重啟後數據不會丟失。

4. 啟動服務

在專案目錄下執行以下指令啟動服務：

```
docker-compose up -d
```

- `-d` 表示以背景模式運行。

您可以執行以下指令檢查服務狀態：

```
docker-compose ps
```

5. 測試服務

寫入測試數據

使用 `curl` 發送測試數據到 `vminsert`：

```
curl -X POST -d 'example_metric{label="test"} 123' http://localhost:8480/insert/0/prometheus/api/v1/write
```

查詢測試數據

使用 `curl` 從 `vmselect` 查詢數據：

```
curl 'http://localhost:8481/select/0/prometheus/api/v1/query_range?query=example_metric&start=1672531200&end=1672617600&ste
```

“注意：`start` 和 `end` 是時間戳（UNIX 時間），請根據當前時間替換。”

6. 停止與清理服務

停止服務

執行以下指令停止所有容器：

```
docker-compose down
```

清理數據

如需清理數據，刪除本地存儲目錄：

```
rm -rf ./data
```



7. 擴展與高可用性

- **擴展節點數量：**
 - 您可以通過複製 `vmselect` 或 `vminsert` 配置並分配不同的埠來增加節點。`
- **監控服務：**
 - 可通過 `http://localhost:8481/metrics` 查看 vmselect` 的內部監控數據。`
 - 使用 Prometheus 或 Grafana 獲取更多可視化數據。

至此，您已成功使用 Docker Compose 部署一個完整的 VictoriaMetrics Cluster！如有其他需求（例如 SSL/TLS 支持），可以進一步調整配置文件。



以繁體中文詳細說明如何debug



在使用 Docker Compose 部署的 VictoriaMetrics Cluster 中進行 **debug** 時，可以針對不同層面逐步排查問題。以下是詳細的 debug 方法說明：

1. 基本問題排查

查看容器狀態

使用以下指令檢查容器的執行狀態：

```
docker-compose ps
```

- **正常狀態：**
 - `State` 應顯示為 Up`。`
 - 如果容器未啟動，可能是配置問題或依賴未正確啟動。

查看容器日誌

檢查容器的輸出日誌，以找出異常信息：

```
docker logs <容器名稱>
```

範例：

```
docker logs vmstorage
docker logs vmselect
```





```
docker logs vminsert
```

- 日誌中常見問題：
 - 配置錯誤，例如 `-storageNode`` 指向的地址錯誤。
 - 無法連接其他節點，通常是網路配置問題。

2. 排查網路問題

檢查網路是否正確連接

列出 Docker Compose 中的網路：

```
docker network ls
```

檢查指定網路（例如 ``victoria-network``）中的容器：

```
docker network inspect victoria-network
```

- 確保 ``vminsert``、``vmselect`` 和 ``vmstorage`` 都屬於同一網路。

測試節點之間的連接

進入容器內部，測試連接：

```
docker exec -it vminsert sh
```

在容器內使用 ``curl`` 測試連接 ``vmstorage``：

```
curl vmstorage:8400/metrics
```

- 如果無法連接，可能是網路配置有誤，請檢查 ``docker-compose.yml`` 的 ``networks`` 配置。

3. 排查 VictoriaMetrics 配置問題

確認配置是否正確

- ``vmselect`` 和 ``vminsert`` 的 ``-storageNode`` 是否指向正確的 ``vmstorage`` 埠。
- 埠號是否對應：
 - ``vminsert`` 寫入：指向 ``vmstorage:8400``
 - ``vmselect`` 查詢：指向 ``vmstorage:8401``

檢查配置文件是否有誤：





```
cat docker-compose.yml
```

檢查埠是否被占用

使用以下指令檢查埠是否已被其他應用占用：

```
sudo netstat -tuln | grep <埠號>
```

4. 測試寫入與查詢

測試數據寫入

通過 `curl` 測試寫入數據到 `vminsert`：

```
curl -X POST -d 'example_metric{label="debug"} 123' http://localhost:8480/insert/0/prometheus/api/v1/write
```

- **成功情況：**返回 HTTP 狀態碼 `200`。
- **失敗情況：**返回錯誤碼（如 `400` 或 `500`），請檢查 `vminsert` 容器的日誌。

測試數據查詢

通過 `curl` 測試查詢數據從 `vmselect`：

```
curl 'http://localhost:8481/select/0/prometheus/api/v1/query_range?query=example_metric&start=1672531200&end=1672617600&ste
```

- **成功情況：**返回 JSON 格式的查詢結果。
- **失敗情況：**返回錯誤，請檢查 `vmselect` 的日誌是否有錯誤信息。

5. 使用監控接口

VictoriaMetrics 提供內建的 `/metrics` 和 `/health` 接口以供 debug。

檢查健康狀態

訪問以下 URL 確認節點健康狀態：

- `vminsert`：

```
curl http://localhost:8480/health
```

- `vmselect`：

```
curl http://localhost:8481/health
```

- `vmstorage`：



```
curl http://localhost:8401/health
```

- 返回 **"OK"**：表示節點運行正常。
- 返回其他內容或無響應：檢查容器日誌以了解問題。

檢查性能與內部指標

使用 ``/metrics`` 獲取內部指標：

- ``vminsert``：

```
curl http://localhost:8480/metrics
```

- ``vmselect``：

```
curl http://localhost:8481/metrics
```

- ``vmstorage``：

```
curl http://localhost:8401/metrics
```

這些指標可以用來分析吞吐量、內存使用情況等問題。

6. 獲取更詳細的日誌

啟用更多日誌輸出

在 ``docker-compose.yml`` 中修改服務命令，啟用調試模式：

```
command:  
- '-loggerLevel=DEBUG'
```

重新啟動服務以應用更改：

```
docker-compose down  
docker-compose up -d
```

- 查看詳細日誌輸出以定位問題：

```
docker logs <容器名稱>
```

7. 常見問題與解決方法



問題	可能原因	解決方法
無法連接 `vmstorage`	`-storageNode` 配置錯誤，或網路未正確連接	確認 `docker-compose.yml` 中的配置正確，並檢查容器之間的網路連接。
數據查詢無結果	無數據寫入或查詢語句有誤	測試寫入數據並檢查查詢參數（時間範圍和查詢指標）。
容器無法啟動	配置文件有誤或埠被占用	檢查 `docker-compose.yml` 並確認埠未被占用。
日誌中出現 "connection reset"	節點之間的連接被拒絕	檢查節點之間的網路，確保防火牆或 Docker 配置未阻止流量。

以上方法涵蓋了 VictoriaMetrics Cluster 部署的 debug 過程。如果問題仍無法解決，可提供具體的日誌或錯誤信息以便進一步協助。



 **Cluster version.pdf**
PDF

參考"Cluster version.pdf"，以繁體中文詳細說明用docker-compose建立cluster VictoriaMetrics



以下是參考 "**Cluster version.pdf**" 的內容，詳細說明如何使用 Docker Compose 建立 VictoriaMetrics Cluster 的步驟與說明。

1. 環境準備

安裝必要工具

- 安裝 Docker：[官方指引](#)
- 安裝 Docker Compose：[官方指引](#)

2. 建立目錄與設定檔

1. 建立專案目錄：

```
mkdir victoria-metrics-cluster
cd victoria-metrics-cluster
```

2. 建立 `docker-compose.yml`：

```
touch docker-compose.yml
```





3. 編寫 `docker-compose.yml` 配置檔

根據文件中的建議，撰寫以下配置，包含三個核心服務（`vminsert`、`vmselect` 和 `vmstorage`）：

```
version: '3.8'
services:
  vmselect:
    image: victoriametrics/vmselect:latest
    container_name: vmselect
    ports:
      - "8481:8481"
    command:
      - '-storageNode=vmstorage:8401' # 指定 vmstorage 節點
    depends_on:
      - vmstorage
    networks:
      - victoria-network

  vminsert:
    image: victoriametrics/vminsert:latest
    container_name: vminsert
    ports:
      - "8480:8480"
    command:
      - '-storageNode=vmstorage:8400' # 指定 vmstorage 節點
    depends_on:
      - vmstorage
    networks:
      - victoria-network

  vmstorage:
    image: victoriametrics/vmstorage:latest
    container_name: vmstorage
    ports:
      - "8400:8400" # 用於接收數據寫入
      - "8401:8401" # 提供數據查詢
    volumes:
      - ./data:/storage # 本地持久化存儲
    networks:
      - victoria-network

networks:
  victoria-network:
    driver: bridge
```

配置詳解

- `vmselect`：
 - 負責數據查詢。
 - `'-storageNode`：指定 `vmstorage` 的查詢接口埠。
- `vminsert`：
 - 負責數據寫入。
 - `'-storageNode`：指定 `vmstorage` 的寫入接口埠。



- `vmstorage`` :
 - 實際數據存儲。
 - 本地目錄 `./data`` 用於數據持久化，防止容器重啟時數據丟失。

4. 啟動服務

執行以下指令以啟動服務：

```
docker-compose up -d
```

檢查容器是否正常運行：

```
docker-compose ps
```

5. 測試服務

測試數據寫入

向 `vminsert`` 發送測試數據：

```
curl -X POST -d 'test_metric{label="debug"} 123' http://localhost:8480/insert/0/prometheus/api/v1/write
```

測試數據查詢

從 `vmselect`` 查詢測試數據：

```
curl 'http://localhost:8481/select/0/prometheus/api/v1/query?query=test_metric'
```

6. 停止與清理

- 停止所有服務：

```
docker-compose down
```

- 停止並刪除數據：

```
docker-compose down -v
```

7. 高可用與擴展

高可用配置



- 將每個服務（`vminsert`、`vmselect`、`vmstorage`）部署多個副本，並使用負載均衡器（如 `nginx`）分配流量。
- 在 `docker-compose.yml` 中增加新的副本並配置埠。

橫向擴展

- 增加新的 `vmstorage` 節點：
 1. 啟動新的 `vmstorage` 容器。
 2. 更新 `vminsert` 和 `vmselect` 的配置，將新的 `vmstorage` 節點加入 `storageNode` 列表。
 3. 重啟所有服務以應用更改。

8. 監控與維護

- 監控數據：
 - 每個服務提供 `/metrics` 頁面，可通過 Prometheus 或 Grafana 獲取。
- 故障排查：
 - 查看容器日誌：

```
docker logs <container_name>
```

透過以上步驟，您可以快速搭建一個可用的 VictoriaMetrics Cluster。如有其他需求（如多租戶、多層架構），可根據需求進一步配置。



參考"Cluster version.pdf"，以繁體中文詳細說明如何debug此cluster



以下是根據 "**Cluster version.pdf**" 的內容，詳細說明如何對 VictoriaMetrics Cluster 進行 debug 的方法：

1. 基本排查步驟

檢查容器狀態

使用以下指令檢查容器是否正常運行：

```
docker-compose ps
```

- 期望結果：
 - 所有服務（`vminsert`、`vmselect`、`vmstorage`）的狀態應為 `Up`。





- **問題排查：**
 - 如果狀態為 `Exited`，請檢查服務日誌（見第 3 步）。

查看容器日誌

使用以下指令檢查每個服務的日誌，查看是否有錯誤信息：

```
docker logs <container_name>
```

範例：

```
docker logs vmselect
docker logs vminsert
docker logs vmstorage
```

- **常見錯誤：**
 - **連線問題：**檢查 `-storageNode` 配置是否正確，是否正確指向 `vmstorage` 的埠。
 - **資源不足：**檢查是否存在 `Out of Memory` 或 `Disk Full` 的提示。

2. 網路連線排查

檢查網路配置

列出 Docker Compose 的網路：

```
docker network ls
```

檢查網路內的容器是否正確連接：

```
docker network inspect victoria-network
```

- **問題排查：**
 - 確保所有服務（`vminsert`、`vmselect`、`vmstorage`）都在同一個網路內。

測試節點連線

進入容器內，測試連線：

```
docker exec -it vminsert sh
```

在容器內使用 `curl` 測試與其他服務的連接：

```
curl vmstorage:8400/metrics
```



- 如果無法連接，可能是網路配置問題或服務未啟動。

3. 測試數據寫入與查詢

測試數據寫入

使用以下指令將測試數據寫入 `\vminsert``：

```
curl -X POST -d 'test_metric{label="debug"} 123' http://localhost:8480/insert/0/prometheus/api/v1/write
```

- **成功**：返回 HTTP 狀態碼 ``200``。
- **失敗**：
 - 返回 ``400``：檢查數據格式是否正確。
 - 返回 ``500``：檢查 `\vminsert`` 的日誌。

測試數據查詢

從 `\vmselect`` 查詢數據：

```
curl 'http://localhost:8481/select/0/prometheus/api/v1/query?query=test_metric'
```

- **成功**：返回 JSON 格式的數據結果。
- **失敗**：
 - 返回錯誤代碼或空結果，請檢查：
 1. 是否正確指向 `\vmstorage``。
 2. 是否有數據成功寫入。

4. 查看健康檢查與內部指標

健康檢查

每個服務都提供 ``/health`` 接口檢查健康狀態：

- `\vminsert``：

```
curl http://localhost:8480/health
```

- `\vmselect``：

```
curl http://localhost:8481/health
```

- `\vmstorage``：

```
curl http://localhost:8401/health
```




- 返回 "OK"：表示服務健康。
- 未返回或報錯：請檢查相應的日誌。

內部指標

訪問 `/metrics`` 查看內部指標：

- ``vminsert``：

```
curl http://localhost:8480/metrics
```

- ``vmselect``：

```
curl http://localhost:8481/metrics
```

- ``vmstorage``：

```
curl http://localhost:8401/metrics
```

這些指標可以用於監控資源使用情況，例如 CPU、內存和磁碟使用量。

5. 特定問題的排查

問題：數據寫入緩慢

- 檢查 ``vminsert`` 的 CPU 和內存使用情況：

```
docker stats vminsert
```

- 查看 ``vmstorage`` 是否有足夠的磁碟空間：

```
curl http://localhost:8401/metrics | grep 'vm_storage_free_space_bytes'
```

問題：查詢延遲高

- 查看 ``vmselect`` 的日誌，檢查是否存在查詢超時：

```
docker logs vmselect
```

- 檢查 CPU 使用情況，確保 ``vmselect`` 節點有足夠資源。

問題：磁碟空間不足

- 查看 ``vmstorage`` 的剩餘空間：

```
df -h
```



- 解決方法：
 1. 增加磁碟空間。
 2. 調整數據保留時間（`-retentionPeriod`）。

6. 啟用更多日誌輸出

在 `docker-compose.yml` 中修改命令，增加日誌級別：

```
command:
  - '-loggerLevel=DEBUG'
```

重新啟動服務：

```
docker-compose down
docker-compose up -d
```

檢查日誌輸出是否有更多詳細信息：

```
docker logs <container_name>
```

7. 高可用與故障轉移

高可用配置

確保每個服務有多個副本：

- 增加多個 `vminsert``、`vmselect`` 和 `vmstorage`` 節點。
- 使用負載均衡器（如 Nginx）分配請求。

vmstorage 故障排查

如果某個 `vmstorage`` 節點不可用，`vminsert`` 會將數據重新路由到其他節點。

- 檢查剩餘節點的資源使用情況，確保它們能夠承受額外負載。

透過以上方法，您可以有效地 debug VictoriaMetrics Cluster，快速找出問題根源並解決。
若需進一步協助，可參考官方文件或提供具體日誌進行更深入的排查。



參考"Cluster version.pdf"，以繁體中文詳細說明若此cluster VictoriaMetric失效





以下是根據 "**Cluster version.pdf**"，詳細說明當 **VictoriaMetrics Cluster** 發生失效時，如何分析原因並恢復系統的完整策略。

1. 確認失效範圍與影響

失效可能涉及數據寫入（`vminsert`）、數據查詢（`vmselect`）或存儲層（`vmstorage`）。需要快速判斷影響範圍。

檢查服務健康狀態

通過 `/health` 接口檢查各服務的健康狀態：

- `vminsert`：

```
curl http://<vminsert_host>:8480/health
```

- `vmselect`：

```
curl http://<vmselect_host>:8481/health
```

- `vmstorage`：

```
curl http://<vmstorage_host>:8491/health
```

分析結果：

1. **返回 "OK"**：表示服務仍在運行，可能存在連接或資源問題。
2. **無響應或錯誤**：表示服務失效，需檢查相應的容器和日誌。

2. 確認具體失效原因

以下是常見的失效類型與對應的排查方法：

(A) 數據寫入中斷（`vminsert`）

- 可能原因：

1. 無法連接到 `vmstorage`。
2. `vmstorage` 資源不足或進入只讀模式。
3. 寫入量超過 `vminsert` 處理能力。

- 排查步驟：

1. 查看 `vminsert` 日誌：

```
docker logs vminsert
```

2. 確認連接：



```
curl http://<vmstorage_host>:8400/health
```

3. 檢查 `vmstorage` 剩餘磁碟空間：

```
curl http://<vmstorage_host>:8401/metrics | grep 'vm_storage_free_space_bytes'
```

(B) 查詢無法執行 (`vmselect`)

- 可能原因：

1. `vmselect` 無法連接到 `vmstorage`。
2. 查詢量過大，資源耗盡。

- 排查步驟：

1. 查看 `vmselect` 日誌：

```
docker logs vmselect
```

2. 測試連接：

```
curl http://<vmstorage_host>:8401/metrics
```

3. 確認查詢指令的正確性，排查是否為過重的查詢。

(C) 存儲層無法提供服務 (`vmstorage`)

- 可能原因：

1. 磁碟空間不足。
2. 高負載導致資源枯竭。
3. 节点故障。

- 排查步驟：

1. 查看 `vmstorage` 日誌：

```
docker logs vmstorage
```

2. 確認磁碟空間是否足夠：

```
df -h
```

3. 測試健康狀態：

```
curl http://<vmstorage_host>:8401/health
```

3. 恢復策略

(A) 恢復數據寫入



- 檢查 `vminsert` 到 `vmstorage` 的連接：
 - 確保 `docker-compose.yml` 中 `vminsert` 的 `-storageNode` 指向正確的 `vmstorage` 節點。
- 增加 `vminsert` 容量：
 - 水平擴展部署多個 `vminsert` 節點，並使用負載均衡器分配流量。

(B) 恢復數據查詢

- 修復 `vmselect` 連接：
 - 檢查 `-storageNode` 配置，確保 `vmselect` 能正確訪問所有 `vmstorage` 節點。
- 優化查詢性能：
 - 增加 `vmselect` 節點的 CPU 和內存。
 - 分析日誌中的查詢模式，避免執行過重查詢。

(C) 恢復存儲層

- 釋放磁碟空間：
 1. 刪除過期數據或減少保留時間（`-retentionPeriod`）。
 2. 檢查並清理舊的快照。
- 水平擴展 `vmstorage` 節點：
 1. 新增 `vmstorage` 節點。
 2. 更新 `vminsert` 和 `vmselect` 的配置，將新節點加入集群。
- 檢查只讀模式：
 - 如果 `vmstorage` 進入只讀模式（`vm_storage_is_read_only`），應立即增加磁碟空間。

4. 高可用配置建議

為避免未來的失效，可採取以下高可用措施：

1. 多副本部署：
 - 每個服務（`vminsert`、`vmselect`、`vmstorage`）至少部署兩個以上的副本。
2. 負載均衡：
 - 使用負載均衡器（如 `nginx`）分配請求。
3. 資源監控與自動擴展：
 - 使用 Prometheus 或 Grafana 監控資源使用情況，配置自動擴展策略。
4. 定期備份：
 - 使用 `/snapshot/create` 接口定期創建快照，確保數據安全。



5. 特殊情境應對策略

(A) 單節點故障

- 症狀：
 - 集群可用性下降，部分數據不可用。
- 解決方法：
 - 確保至少兩個節點在線以維持高可用。
 - 修復或替換故障節點。

(B) 整體集群故障

- 症狀：
 - 所有數據無法訪問。
- 解決方法：
 1. 確保所有容器正在運行：

```
docker-compose up -d
```
 2. 檢查負載均衡配置，避免將請求發送到故障節點。
 3. 如果無法恢復，使用備份快照進行還原。

透過上述方法，您可以迅速判斷並解決 VictoriaMetrics Cluster 的失效問題，同時採取高可用配置以避免未來發生類似情況。若仍有問題，可提供具體日誌進行更進一步的排查。