



Docker VictoriaMetrics Cluster 設定



Cluster version.pdf
PDF

參考此pdf，以繁體中文詳細說明用docker image 而不使用k8s，建立cluster VictoriaMetrics，vmstorage、vminsert、vmselect都各2個節點



根據提供的PDF內容與目標需求，以下是使用 Docker image（不使用 Kubernetes）部署包含 `vmstorage`、`vminsert` 和 `vmselect` 各 2 個節點的 VictoriaMetrics Cluster 的詳細說明：

部署架構概述

VictoriaMetrics Cluster 包括以下核心組件：

1. **vmstorage**：存儲原始數據，負責根據查詢條件返回相關數據。
2. **vminsert**：接收數據寫入請求，並根據一致性哈希分配到不同的 vmstorage 節點。
3. **vmselect**：處理查詢請求，從 vmstorage 節點獲取數據並返回結果。

準備環境

1. 安裝 Docker

確保已在主機上安裝 Docker，並能正常使用。





2. 獲取 Docker Image

從 Docker Hub 下載所需的 VictoriaMetrics image：

```
docker pull victoriametrics/vmstorage
docker pull victoriametrics/vminsert
docker pull victoriametrics/vmselect
```

3. 準備存儲路徑

為每個 vmstorage 節點準備獨立的存儲目錄。例如：

```
/data/vmstorage1
/data/vmstorage2
```

部署步驟

1. 部署 vmstorage 節點

啟動兩個 vmstorage 節點，分別使用不同的存儲路徑和端口：

```
docker run -d --name vmstorage1 \
-v /data/vmstorage1:/storage \
-p 8400:8400 \
-p 8482:8482 \
victoriametrics/vmstorage \
-retentionPeriod=30 \
-storageDataPath=/storage

docker run -d --name vmstorage2 \
-v /data/vmstorage2:/storage \
-p 8401:8400 \
-p 8483:8482 \
victoriametrics/vmstorage \
-retentionPeriod=30 \
-storageDataPath=/storage
```



2. 部署 vminsert 節點

配置 vminsert 與 vmstorage 節點連接，啟動兩個 vminsert 節點：

```
docker run -d --name vminsert1 \  
-p 8480:8480 \  
victoriametrics/vminsert \  
-storageNode=vmstorage1:8400,vmstorage2:8400  
  
docker run -d --name vminsert2 \  
-p 8484:8480 \  
victoriametrics/vminsert \  
-storageNode=vmstorage1:8400,vmstorage2:8400
```

3. 部署 vmselect 節點

配置 vmselect 與 vmstorage 節點連接，啟動兩個 vmselect 節點：

```
docker run -d --name vmselect1 \  
-p 8481:8481 \  
victoriametrics/vmselect \  
-storageNode=vmstorage1:8400,vmstorage2:8400  
  
docker run -d --name vmselect2 \  
-p 8485:8481 \  
victoriametrics/vmselect \  
-storageNode=vmstorage1:8400,vmstorage2:8400
```

配置負載均衡

可以使用 Nginx 或其他反向代理進行負載均衡，將數據寫入請求轉發至 `vminsert` 節點，查詢請求轉發至 `vmselect` 節點。

Nginx 配置範例



```
http {  
    upstream vminsert {  
        server localhost:8480;  
        server localhost:8484;  
    }  
  
    upstream vmselect {  
        server localhost:8481;  
        server localhost:8485;  
    }  
  
    server {  
        listen 8080;  
  
        location /insert/ {  
            proxy_pass http://vminsert;  
        }  
  
        location /select/ {  
            proxy_pass http://vmselect;  
        }  
    }  
}
```

啟動 Nginx：

```
sudo systemctl restart nginx
```

測試與驗證

1. **數據寫入測試** 使用以下命令模擬數據寫入：

```
curl -X POST  
"http://<nginx_host>:8080/insert/0/prometheus/api/v1/write" \
```



```
--data-binary @data.prom
```

2. 查詢測試 發送查詢請求：

```
curl "http://<nginx_host>:8080/select/0/prometheus/api/v1/query?  
query=up"
```

以上配置完成後，即可實現基於 Docker 的高可用 VictoriaMetrics 集群部署。若有其他需求或問題，請隨時告知！



參考"Cluster version.pdf"，以繁體中文詳細說明配置負載均衡，可以不配置負載均衡嗎



在部署 **VictoriaMetrics Cluster** 時，負載均衡並非必須，但推薦配置，以實現高效流量分發與高可用性。在以下詳細說明中，我將解釋負載均衡的必要性以及不配置負載均衡的替代方案。

為什麼需要負載均衡？

VictoriaMetrics Cluster 通常包括多個節點（`vmstorage`、`vminsert` 和 `vmselect`），每個節點可以獨立擴展。配置負載均衡主要有以下好處：

1. **流量分發**：自動將數據寫入請求分發到多個 `vminsert` 節點，查詢請求分發到多個 `vmselect` 節點，避免單點壓力過大。



2. **高可用性**：當某些節點故障時，負載均衡可以自動將請求路由到其他可用節點，減少中斷風險。
3. **簡化客戶端配置**：使用統一入口點，客戶端不需要關心內部節點數量或地址。

如何配置負載均衡？

以下是使用 Nginx 配置負載均衡的詳細步驟。

1. 安裝 Nginx

在負載均衡器主機上安裝 Nginx：

```
sudo apt update
sudo apt install nginx
```

2. 配置 Nginx 負載均衡

編輯 Nginx 配置文件，例如 ``/etc/nginx/nginx.conf``：

```
http {
    upstream vminsert {
        server <vminsert1_ip>:8480;
        server <vminsert2_ip>:8480;
    }

    upstream vmselect {
        server <vmselect1_ip>:8481;
        server <vmselect2_ip>:8481;
    }

    server {
        listen 8080;

        # 用於數據寫入的請求路由
        location /insert/ {
            proxy_pass http://vminsert;
        }
    }
}
```



```
    }

    # 用於查詢的請求路由
    location /select/ {
        proxy_pass http://vmselect;
    }
}
}
```

“其中 `<vminsert1_ip>`、`<vminsert2_ip>` 分別是兩個 `vminsert` 節點的 IP 地址；
`<vmselect1_ip>`、`<vmselect2_ip>` 是兩個 `vmselect` 節點的 IP 地址。”

3. 測試與啟用配置

測試 Nginx 配置是否正確：

```
sudo nginx -t
```

重啟 Nginx 以應用新配置：

```
sudo systemctl restart nginx
```

4. 客戶端訪問

使用以下統一地址訪問集群：

- 數據寫入：`http://<nginx_host>:8080/insert/`
- 數據查詢：`http://<nginx_host>:8080/select/`

不配置負載均衡的替代方案

在不使用負載均衡的情況下，您需要手動將請求路由到相應的節點。以下是操作細節與限制。

直接使用節點地址



1. 數據寫入

客戶端需要明確指定 `vminsert` 節點地址，例如：

```
curl -X POST
"http://<vminsert1_ip>:8480/insert/0/prometheus/api/v1/write" \
--data-binary @data.prom
```

2. 數據查詢

客戶端需要明確指定 `vmselect` 節點地址，例如：

```
curl "http://<vmselect1_ip>:8481/select/0/prometheus/api/v1/query?
query=up"
```

優點

- 配置簡單，無需額外的負載均衡器。
- 減少了中間代理層的潛在性能瓶頸。

限制

1. **缺乏高可用性**：如果某個節點故障，客戶端必須手動切換到其他可用節點。
2. **運維成本高**：客戶端需要了解所有節點的地址，且請求分配需要人工管理。
3. **均衡性不足**：可能導致某些節點過載，而其他節點閒置。

是否需要配置負載均衡的建議

- **推薦配置**：如果您的集群規模較大，且需要高可靠性與性能，建議使用負載均衡。
- **可以不配置**：如果集群只有少量節點，並且對高可用性要求不高，可以選擇不配置負載均衡，但需要制定清晰的節點管理策略。

如果有進一步問題，請隨時告知！



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



若負載均衡的 Nginx 節點失效，可能會對 VictoriaMetrics Cluster 的整體運作造成影響，具體後果和解決方法如下：

Nginx 節點失效的後果

1. **數據寫入中斷**：所有指向 Nginx 的數據寫入請求將無法送達 `vminsert` 節點，導致新數據無法存儲。
2. **查詢請求中斷**：所有查詢流量將無法送達 `vmselect` 節點，導致用戶無法執行查詢。
3. **系統中斷風險**：如果 Nginx 是唯一的入口點，失效後將無法訪問整個 VictoriaMetrics Cluster，對系統可用性影響重大。

Nginx 節點失效的解決方案

1. 增設備援 Nginx 節點

配置多個 Nginx 節點作為負載均衡器，並通過高可用性（HA）技術保證服務連續性。

步驟：

1. 部署多個 Nginx 節點

在其他伺服器上部署額外的 Nginx 節點，使用相同的配置文件，確保配置一致性。

```
sudo apt update
sudo apt install nginx
```



然後將 `/etc/nginx/nginx.conf` 複製到新的 Nginx 節點。

2. **配置 Keepalived 實現 VIP（虛擬 IP）** 使用 Keepalived 為多個 Nginx 節點提供一個虛擬 IP 地址（VIP），客戶端只需訪問該 VIP。

Keepalived 配置範例：

```
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 100
    authentication {
        auth_type PASS
        auth_pass 12345
    }
    virtual_ipaddress {
        192.168.1.100
    }
}
```

3. **啟用 Keepalived**

```
sudo systemctl restart keepalived
```

4. **測試 VIP 切換** 模擬主 Nginx 節點故障，檢查 VIP 是否自動切換到備援節點。

2. 客戶端直接訪問後端節點

若 Nginx 節點故障，可將流量直接導向後端的 `vminsert` 和 `vmselect` 節點。

操作方式：

1. 修改客戶端配置，直接使用 `vminsert` 和 `vmselect` 節點的地址進行數據寫入和查詢：



- **數據寫入：**

```
curl -X POST
"http://<vminsert1_ip>:8480/insert/0/prometheus/api/v1/write" \
--data-binary @data.prom
```

- **數據查詢：**

```
curl "http://<vmselect1_ip>:8481/select/0/prometheus/api/v1/query?
query=up"
```

限制：

- 此方法為臨時應急方案，對於大型系統或頻繁訪問的應用場景，操作繁瑣且易出錯。
- 客戶端需明確掌握所有後端節點的地址，增加配置複雜性。

3. 使用雲端或硬體負載均衡服務

可選擇部署高可用的負載均衡服務，避免單一 Nginx 節點成為系統瓶頸。

- **雲端服務：**AWS ELB、Google Cloud Load Balancer、Azure Load Balancer。
- **硬體設備：**如 F5、Citrix NetScaler。

優點：

- 提供內建高可用性，故障轉移快速且無縫。
- 減少內部運維負擔。

缺點：

- 增加了運行成本。
- 需要額外學習和配置特定負載均衡服務。



預防措施

1. 監控 Nginx 狀態

配置監控工具（如 Prometheus + Grafana）監控 Nginx 節點的健康狀態。當 Nginx 節點失效時，及時觸發告警。

- 常用的監控指標包括：
 - `nginx_http_requests_total`（請求數量）
 - `nginx_up`（可用性）

2. 定期備份配置文件

定期備份 Nginx 配置文件，快速恢復故障。

3. 設置自動重啟機制

配置 Nginx 在故障後自動重啟：

```
sudo systemctl enable nginx
sudo systemctl restart nginx
```

總結

1. 若 Nginx 節點失效，會影響數據寫入和查詢，但可以通過多種方式恢復或減少影響：

- 增設備援 Nginx 節點，配合 VIP 實現高可用。
- 客戶端直接訪問後端節點作為應急方案。
- 使用雲端或硬體負載均衡服務作為替代。

2. 預防措施包括：

- 配置監控和告警。
- 定期備份配置。
- 配置自動重啟機制。



若需更多技術細節或協助，請隨時告知！



參考"Cluster version.pdf"，以繁體中文詳細說明若沒有 Nginx，怎麼做到vminsert ha



如果不使用 **Nginx** 或其他負載均衡器，依然可以實現 **vminsert** 的高可用性 (HA)，但需要根據 VictoriaMetrics Cluster 的特性進行配置與調整。以下是詳細的操作與說明：

vminsert 高可用性需求分析

1. 目標：

- 保證數據寫入不中斷，即使某個 `vminsert` 節點失效，其他節點仍可正常運行。
- 實現負載分擔，避免單個節點過載。

2. 挑戰：

- 缺乏負載均衡器的自動流量分發功能，客戶端需要管理寫入流量分配。
- 節點故障時，需要手動或程序化切換寫入路徑。

方案一：客戶端輪詢實現 HA

1. 客戶端直接管理節點路由

讓客戶端應用程序負責將數據寫入請求輪詢分發到多個 `vminsert` 節點。

步驟：

1. 在客戶端設置一個節點列表：



```
vminsert_nodes = [  
    "http://vminsert1:8480",  
    "http://vminsert2:8480"  
]
```

2. 將數據寫入請求以輪詢方式發送到這些節點：

```
import requests  
from itertools import cycle  
  
# 建立輪詢機制  
nodes = cycle(vminsert_nodes)  
  
def send_data(data):  
    node = next(nodes)  
    url = f"{node}/insert/0/prometheus/api/v1/write"  
    try:  
        response = requests.post(url, data=data)  
        response.raise_for_status()  
        print(f"Data sent to {node}")  
    except requests.exceptions.RequestException as e:  
        print(f"Failed to send data to {node}: {e}")  
        # 可選：嘗試下一個節點
```

優點：

- 實現簡單，客戶端可控性強。
- 無需額外基礎設施。

缺點：

- 需要在客戶端實現錯誤處理和重試機制。
- 當節點數量增加時，管理複雜性提高。

方案二：配置多個 DNS A 記錄



利用 DNS A 記錄將多個 `vminsert` 節點的 IP 地址綁定到同一個域名，讓客戶端通過域名進行寫入。

1. 配置 DNS

將多個 `vminsert` 節點的 IP 地址添加到同一域名，例如 `vminsert.example.com`：

```
vminsert.example.com. 300 IN A 192.168.1.101
vminsert.example.com. 300 IN A 192.168.1.102
```

2. 客戶端使用域名進行寫入

客戶端直接向該域名發送數據寫入請求，DNS 將自動隨機返回一個節點 IP：

```
curl -X POST
"http://vminsert.example.com:8480/insert/0/prometheus/api/v1/write" \
--data-binary @data.prom
```

優點：

- 簡化了客戶端的節點管理。
- 當節點數量變化時，只需更新 DNS 記錄。

缺點：

- DNS 的負載分配為隨機，無法根據負載動態調整。
- DNS 生效可能有延遲（受 TTL 設置影響）。

方案三：使用 vmauth 實現 HA

vmauth 是 VictoriaMetrics 提供的認證和代理工具，可以用於管理多個 `vminsert` 節點的流量。

1. 部署 vmauth

從 Docker Hub 獲取 vmauth：



```
docker pull victoriametrics/vmauth
```

啟動 vmauth，配置後端 `vminsert` 節點：

```
docker run -d --name vmauth \  
-p 8427:8427 \  
-v /path/to/vmauth.yml:/vmauth.yml \  
victoriametrics/vmauth -config=/vmauth.yml
```

2. 配置文件範例

`vmauth.yml` 文件：

```
users:  
- url_prefix: "http://vminsert"  
  urls:  
    - "http://vminsert1:8480"  
    - "http://vminsert2:8480"  
  password: "optional_password"
```

3. 客戶端訪問

客戶端通過 vmauth 的統一入口進行數據寫入：

```
curl -X POST "http://vmauth:8427/insert/0/prometheus/api/v1/write" \  
--data-binary @data.prom
```

優點：

- 提供統一入口，簡化客戶端配置。
- 支援認證功能，增強安全性。

缺點：

- 增加了運維工作量。
- vmauth 本身可能成為單點故障。



方案四：利用客戶端庫的內建 HA 功能

某些數據收集器或應用程序庫（如 Prometheus 的遠端寫入功能）支持自動配置多個目標地址，並內建故障切換和負載均衡功能。

1. 配置 Prometheus

在 Prometheus 配置文件中設置多個 `vminsert` 節點：

```
remote_write:
  - url: "http://vminsert1:8480/insert/0/prometheus/api/v1/write"
  - url: "http://vminsert2:8480/insert/0/prometheus/api/v1/write"
```

Prometheus 將自動嘗試寫入所有配置的地址，並在某個節點不可用時切換到其他節點。

優點：

- 自動故障切換，無需額外開發。
- 方便整合現有的監控系統。

缺點：

- 需要特定的應用程序支持。

比較與建議

方案	優點	缺點	適用場景
客戶端輪詢	簡單靈活，無需額外工具	增加客戶端開發和管理負擔	小型集群，簡單應用場景
多個 DNS A 記錄	配置簡單，無需修改客戶端	隨機分配，無法動態調整負載	節點數量固定，負載較輕



方案	優點	缺點	適用場景
使用 vmauth	提供統一入口，支持認證	vmauth 本身可能成為單點故障	需要統一入口和安全控制
客戶端庫內建 HA 功能	自動負載均衡與故障切換	依賴特定庫支持	整合現有數據收集工具

推薦方案：根據您的具體需求選擇：

- **快速實現：**選擇 DNS A 記錄方案。
- **統一入口與認證：**使用 vmauth。
- **應急方案：**客戶端輪詢。

若需進一步協助，請隨時告知！