



安裝VictoriaMetrics單節點



教我怎麼在ubuntu 22.04建立單節點VictoriaMetrics



以下是步驟來在 **Ubuntu 22.04** 中安裝和設置單節點版本的 **VictoriaMetrics**，這是一個高性能的時序數據庫：

1. 安裝必要工具

確保系統已更新並安裝必要的工具。

```
sudo apt update
sudo apt upgrade -y
sudo apt install wget curl -y
```

2. 下載 VictoriaMetrics 單節點執行檔

從官方網站下載最新的單節點 VictoriaMetrics 執行檔。

```
wget https://github.com/VictoriaMetrics/VictoriaMetrics/releases/latest/download/victoria-metrics-linux-amd64.tar.gz
```

解壓下載的檔案：

```
tar -xvf victoria-metrics-linux-amd64.tar.gz
```

進入解壓的目錄：

```
cd victoria-metrics-linux-amd64
```

3. 移動執行檔到系統目錄

將執行檔移動到 `/usr/local/bin` 以便全系統使用：

```
sudo mv victoria-metrics-prod /usr/local/bin/victoria-metrics
```



4. 建立資料目錄

為 VictoriaMetrics 資料創建專用目錄：

```
sudo mkdir -p /var/lib/victoriametrics
sudo chown $(whoami):$(whoami) /var/lib/victoriametrics
```

5. 啟動 VictoriaMetrics

以最簡單的方式啟動 VictoriaMetrics：

```
victoria-metrics -storageDataPath=/var/lib/victoriametrics
```

如果執行成功，你會看到類似以下的日誌輸出，表示服務正在運行：

```
2025-01-08T10:00:00.000Z info VictoriaMetrics v1.x.x started ...
```

VictoriaMetrics 的默認 HTTP 介面運行在 **8428** 埠，你可以訪問以下網址確認服務是否啟動：

```
http://<你的IP>:8428
```

6. 設置 Systemd 服務（可選，推薦）

為 VictoriaMetrics 創建一個 systemd 服務文件，讓它可以作為系統服務運行：

```
sudo nano /etc/systemd/system/victoria-metrics.service
```

內容如下：

```
[Unit]
Description=VictoriaMetrics Single Node
After=network.target

[Service]
ExecStart=/usr/local/bin/victoria-metrics -storageDataPath=/var/lib/victoriametrics
Restart=always
User=nobody
Group=nogroup

[Install]
WantedBy=multi-user.target
```

儲存並退出（`Ctrl+O`，`Enter`，`Ctrl+X`）。



7. 啟用並啟動服務

啟用服務使其開機自啟：

```
sudo systemctl enable victoria-metrics
```

立即啟動服務：

```
sudo systemctl start victoria-metrics
```

檢查服務狀態：

```
sudo systemctl status victoria-metrics
```

如果一切正常，你應該會看到 `Active: active (running)`。

8. 測試 VictoriaMetrics

打開瀏覽器，訪問以下網址：

```
http://<你的伺服器IP>:8428
```

9. 進一步配置

你可以根據需求配置：

- **Prometheus** 作為數據源
- **Grafana** 用於數據可視化
- 使用 CLI 或 HTTP API 提交和查詢數據。

需要更多幫助可以隨時告訴我！



用docker呢



使用 Docker 安裝和運行單節點版本的 VictoriaMetrics 非常簡單。以下是具體步驟：



1. 安裝 Docker

如果你的系統尚未安裝 Docker，請執行以下命令進行安裝：

```
sudo apt update
sudo apt install -y docker.io
sudo systemctl start docker
sudo systemctl enable docker
```

驗證 Docker 是否成功安裝：

```
docker --version
```

2. 拉取 VictoriaMetrics Docker 映像

從 Docker Hub 拉取最新的單節點 VictoriaMetrics 映像：

```
docker pull victoriametrics/victoria-metrics
```

3. 啟動 VictoriaMetrics

使用 Docker 啟動 VictoriaMetrics：

```
docker run -d --name victoria-metrics \
-p 8428:8428 \
-v /path/to/data:/var/lib/victoria-metrics \
victoriametrics/victoria-metrics
```

參數說明：

- `-d`：以後台模式運行容器。
- `--name victoria-metrics`：指定容器名稱。
- `-p 8428:8428`：將容器內部的 8428 埠映射到主機。
- `-v /path/to/data:/var/lib/victoria-metrics`：將主機的資料目錄掛載到容器內部，確保數據持久化。

例如：如果你的數據目錄位於 `/home/user/victoriametrics-data`，命令應改為：

```
docker run -d --name victoria-metrics \
-p 8428:8428 \
-v /home/user/victoriametrics-data:/var/lib/victoria-metrics \
victoriametrics/victoria-metrics
```



4. 驗證服務是否啟動

檢查 Docker 容器是否正在運行：

```
docker ps
```

訪問以下網址，確認 VictoriaMetrics 是否正常運行：

```
http://<你的伺服器IP>:8428
```

5. 測試和提交數據

VictoriaMetrics 提供 HTTP API 供測試，你可以使用 `curl` 提交測試數據。

提交測試數據：

```
curl -d 'metric_name{label="value"} 123' http://<你的伺服器IP>:8428/api/v1/import
```

查詢數據：

```
http://<你的伺服器IP>:8428/api/v1/query?query=metric_name
```

6. 使用 Docker Compose（可選）

如果你更喜歡使用 `docker-compose`，可以創建一個 `docker-compose.yml` 文件：

```
version: '3.7'

services:
  victoria-metrics:
    image: victoriametrics/victoria-metrics
    container_name: victoria-metrics
    ports:
      - "8428:8428"
    volumes:
      - /path/to/data:/var/lib/victoria-metrics
```

啟動服務：

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



7. 停止與刪除容器

停止容器：

```
docker stop victoria-metrics
```

刪除容器：

```
docker rm victoria-metrics
```

VictoriaMetrics 現在已通過 Docker 成功部署。若有其他問題，請隨時詢問！



這2種方式是單節點還是多節點



上述的兩種方式（直接執行檔和 Docker 部署）都是 **單節點** 部署的方式，因為它們啟動的是 **VictoriaMetrics 的單節點版本**（``victoria-metrics``），該版本專為簡單部署、單機運行而設計。

單節點版本特性

- **適用場景**：單機運行，適合小型應用、測試環境或數據量較小的場景。
- **無分布式功能**：所有的數據存儲、查詢和處理都在一個節點內完成。
- **性能**：即使是單節點，VictoriaMetrics 依然可以處理大量的時序數據。
- **數據存儲路徑**：所有數據都存儲在單個節點的磁碟上（例如 ``/var/lib/victoria-metrics`` 或 Docker 的掛載目錄）。

如果需要多節點部署

VictoriaMetrics 也支持多節點模式，這通常用於高可用性或需要水平擴展的場景。多節點模式主要包括以下組件：

1. **vmstorage**：負責數據存儲。
2. **vminsert**：負責接收並寫入數據。
3. **vmselect**：負責查詢數據。





多節點部署需要額外的配置，並且每個組件需要分開啟動。例如，可以使用 Docker Compose 或 Kubernetes 來實現分布式部署。

如果你的需求涉及到更大的數據量或需要高可用性，可以考慮切換到多節點模式。我可以為你提供相關的部署指南！



詳細說明docker單節點，配置Prometheus 作為數據源、Grafana 用於數據可視化



以下是使用 Docker 配置 **VictoriaMetrics** 單節點，並將 **Prometheus** 作為數據源，以及通過 **Grafana** 用於數據可視化的詳細步驟：

1. 安裝 Docker 和 Docker Compose

確保系統已安裝 Docker 和 Docker Compose。

如果尚未安裝，可以使用以下命令進行安裝：

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

驗證安裝：

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

2. 創建工作目錄

創建一個目錄來存放所有相關文件：

```
mkdir -p ~/victoria-stack
cd ~/victoria-stack
```

3. 創建 `docker-compose.yml` 文件

在工作目錄中創建 `docker-compose.yml` 文件：





```
nano docker-compose.yml
```

填入以下內容：

```
version: '3.7'

services:
  victoria-metrics:
    image: victoriametrics/victoria-metrics
    container_name: victoria-metrics
    ports:
      - "8428:8428"
    volumes:
      - ./victoria-metrics-data:/var/lib/victoria-metrics
    restart: unless-stopped

  prometheus:
    image: prom/prometheus
    container_name: prometheus
    ports:
      - "9090:9090"
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
    command:
      - "--storage.tsdb.retention.time=15d"
    restart: unless-stopped

  grafana:
    image: grafana/grafana
    container_name: grafana
    ports:
      - "3000:3000"
    environment:
      - GF_SECURITY_ADMIN_USER=admin
      - GF_SECURITY_ADMIN_PASSWORD=admin
    volumes:
      - grafana-data:/var/lib/grafana
    restart: unless-stopped

volumes:
  grafana-data:
```

4. 配置 Prometheus

在工作目錄中創建 `prometheus.yml` 文件：

```
nano prometheus.yml
```

填入以下內容：



```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: "victoriametrics"
    static_configs:
      - targets: ['victoria-metrics:8428']
```

說明：

- `scrape_interval: 15s`：定義 Prometheus 的數據抓取頻率。
- `targets`：指定要監控的目標（VictoriaMetrics 的 HTTP 接口）。

5. 啟動所有服務

在工作目錄中運行以下命令：

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

檢查容器是否正常運行：

```
docker ps
```

應該看到以下三個容器正在運行：

- `victoria-metrics`
- `prometheus`
- `grafana`

6. 配置 Grafana

(1) 訪問 Grafana

在瀏覽器中打開以下地址：

```
http://<你的伺服器IP>:3000
```

默認用戶名和密碼：

- 用戶名：`admin`
- 密碼：`admin`

(2) 添加數據源

1. 登錄 Grafana，點擊左側菜單的「齒輪圖標 (Settings)」->「Data Sources」->「Add Data Source」。



2. 選擇 **Prometheus**。

3. 配置 URL：

```
http://prometheus:9090
```

4. 點擊「Save & Test」，確認數據源正常連接。

7. 創建 Dashboard

(1) 添加圖表

1. 點擊左側菜單「+」->「Dashboard」->「New Panel」。
2. 在「Query」部分選擇 Prometheus 作為數據源。
3. 輸入 PromQL 查詢語句，例如：
 - `up{job="victoriametrics"}`：查看 VictoriaMetrics 的運行狀態。
 - `scrape_samples_scraped`：查看 Prometheus 抓取的樣本數。

(2) 配置圖表樣式

1. 配置圖表標題、類型（如時間序列圖）。
2. 點擊「Apply」，保存圖表。

(3) 保存 Dashboard

1. 點擊右上角「Save Dashboard」。
2. 命名並保存你的 Dashboard。

8. 驗證系統是否運行正常

- **VictoriaMetrics**：瀏覽器訪問 `http://<你的伺服器IP>:8428`，確認界面是否正常。
- **Prometheus**：瀏覽器訪問 `http://<你的伺服器IP>:9090`，測試 PromQL 查詢是否返回數據。
- **Grafana**：瀏覽器訪問 `http://<你的伺服器IP>:3000`，確認 Dashboard 是否顯示數據。

9. 其他注意事項

1. **數據持久化**：`victoria-metrics-data` 和 `grafana-data` 目錄確保數據不會因容器重啟丟失。
2. **Prometheus 配置**：可根據需求添加更多監控目標。
3. **安全性**：將 Grafana 的默認密碼修改為更安全的值。



若需要進一步優化或擴展部署，請隨時告訴我！



分別詳細說明VictoriaMetrics，怎麼配置Prometheus 作為數據源、怎麼Grafana 用於數據可視化



以下是詳細說明 **VictoriaMetrics** 如何配置 **Prometheus** 作為數據源，以及 **Grafana** 用於數據可視化 的步驟：

1. 配置 VictoriaMetrics

VictoriaMetrics 是高性能的時序數據庫，可以直接作為 Prometheus 的遠程存儲。

(1) 啟動 VictoriaMetrics（單節點）

如果使用 Docker，運行以下命令啟動 VictoriaMetrics：

```
docker run -d --name victoria-metrics \
  -p 8428:8428 \
  -v /path/to/data:/var/lib/victoria-metrics \
  victoriametrics/victoria-metrics
```

參數解釋：

- `-p 8428:8428``：將 VictoriaMetrics 的 HTTP 端口暴露到主機的 8428。
- `-v /path/to/data:/var/lib/victoria-metrics``：將數據持久化到主機。

瀏覽器訪問以下地址確認是否正常運行：

```
http://<你的伺服器IP>:8428
```

2. 配置 Prometheus 作為數據源

Prometheus 的作用是抓取目標數據，並將其寫入 VictoriaMetrics。

(1) 創建 Prometheus 配置文件

在 Prometheus 的目錄下創建 `prometheus.yml` 文件：

```
global:
  scrape_interval: 15s # 設置抓取數據的頻率
```





```
scrape_configs:
  - job_name: "victoriametrics"
    static_configs:
      - targets: ['victoria-metrics:8428']

remote_write:
  - url: "http://victoria-metrics:8428/api/v1/write"
```

配置說明：

- `scrape_configs`：定義 Prometheus 需要抓取的目標數據。
 - `targets`：這裡可以是其他監控目標，例如應用程序的 `/metrics` 接口。
- `remote_write`：設置 Prometheus 將抓取的數據寫入到 VictoriaMetrics 中。

(2) 啟動 Prometheus

如果使用 Docker，啟動命令如下：

```
docker run -d --name prometheus \
  -p 9090:9090 \
  -v /path/to/prometheus.yml:/etc/prometheus/prometheus.yml \
  prom/prometheus
```

確認 Prometheus 運行正常：

```
http://<你的伺服器IP>:9090
```

3. 配置 Grafana 用於數據可視化

Grafana 用於從 VictoriaMetrics 中讀取數據並進行可視化。

(1) 啟動 Grafana

如果使用 Docker，運行以下命令：

```
docker run -d --name grafana \
  -p 3000:3000 \
  -e "GF_SECURITY_ADMIN_USER=admin" \
  -e "GF_SECURITY_ADMIN_PASSWORD=admin" \
  grafana/grafana
```

瀏覽器訪問以下地址：

```
http://<你的伺服器IP>:3000
```

登錄默認用戶名和密碼：



- 用戶名：`admin`
- 密碼：`admin`

(2) 配置 VictoriaMetrics 作為 Grafana 的數據源

1. 登錄 Grafana。
2. 點擊左側菜單的 齒輪圖標 (Settings) -> Data Sources -> Add data source。
3. 選擇 **Prometheus** (VictoriaMetrics 兼容 Prometheus 協議)。
4. 配置以下內容：
 - **URL**：`http://victoria-metrics:8428`
 - **HTTP Method**：GET
 - **Scrape Interval**：15s
5. 點擊 **Save & Test**，確認數據源正常連接。

(3) 創建 Grafana Dashboard

1. 點擊左側菜單的「+」->「Dashboard」->「New Panel」。
2. 在 Query 部分：
 - 選擇剛剛配置的數據源。
 - 編寫 PromQL 查詢語句，例如：
 - `up`：查看 Prometheus 目標是否處於運行狀態。
 - `scrape_samples_scraped`：查看 Prometheus 抓取的樣本數量。
3. 配置圖表類型和標題。
4. 點擊 **Apply** 保存圖表。

(4) 保存 Dashboard

1. 點擊右上角 **Save Dashboard**。
2. 命名並保存你的 Dashboard。

4. 總結

完成上述配置後，整個系統的工作流程如下：

1. **Prometheus** 抓取目標數據，並通過 `remote_write` 寫入到 **VictoriaMetrics**。
2. **VictoriaMetrics** 存儲這些時序數據。



3. **Grafana** 通過 **Prometheus API** 從 VictoriaMetrics 中讀取數據並展示圖表。

這樣，你就可以實現 VictoriaMetrics 的數據存儲和可視化分析。若有任何問題，請隨時告訴我！



將上述目標，5天的worker list



以下是實現配置 **VictoriaMetrics**、**Prometheus** 作為數據源 和 **Grafana** 用於數據可視化的 5 天工作清單 (Worker List)：

Day 1: 環境準備

目標：準備部署環境並確保基礎工具可用

1. 安裝基礎工具：

- 更新系統：`sudo apt update && sudo apt upgrade -y`
- 安裝 Docker：`sudo apt install -y docker.io`
- 安裝 Docker Compose：`sudo apt install -y docker-compose`

2. 驗證 Docker 和 Docker Compose 安裝：

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

3. 创建工作目錄：

```
mkdir -p ~/victoria-stack && cd ~/victoria-stack
```

Day 2: 部署 VictoriaMetrics 和 Prometheus

目標：完成 VictoriaMetrics 和 Prometheus 部署，並配置 Prometheus 寫入數據到 VictoriaMetrics

1. 部署 VictoriaMetrics：

- 使用 Docker 啟動單節點 VictoriaMetrics：

```
docker run -d --name victoria-metrics \
  -p 8428:8428 \
  -v ~/victoria-stack/victoria-metrics-data:/var/lib/victoria-metrics \
  victoriametrics/victoria-metrics
```





- 驗證是否運行成功：瀏覽器訪問 `http://<伺服器IP>:8428`，應顯示 VictoriaMetrics 的 Web UI。

2. 配置 Prometheus 作為數據抓取工具：

- 在工作目錄中創建 `prometheus.yml`：

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: "victoriametrics"
    static_configs:
      - targets: ['victoria-metrics:8428']

remote_write:
  - url: "http://victoria-metrics:8428/api/v1/write"
```

- 使用 Docker 啟動 Prometheus：

```
docker run -d --name prometheus \
  -p 9090:9090 \
  -v ~/victoria-stack/prometheus.yml:/etc/prometheus/prometheus.yml \
  prom/prometheus
```

- 驗證是否運行成功：瀏覽器訪問 `http://<伺服器IP>:9090`，確認 Prometheus 是否正常運行。

Day 3: 部署 Grafana

目標：完成 Grafana 部署並連接 VictoriaMetrics

1. 部署 Grafana：

- 使用 Docker 啟動 Grafana：

```
docker run -d --name grafana \
  -p 3000:3000 \
  -e "GF_SECURITY_ADMIN_USER=admin" \
  -e "GF_SECURITY_ADMIN_PASSWORD=admin" \
  grafana/grafana
```

- 驗證是否運行成功：瀏覽器訪問 `http://<伺服器IP>:3000`，登錄默認用戶名和密碼（`admin/admin`）。

2. 連接 Grafana 和 VictoriaMetrics：

- 在 Grafana 界面中，添加 **Prometheus** 作為數據源：
 - URL：`http://victoria-metrics:8428`



- 測試連接，確保數據源可用。

Day 4: 創建可視化圖表

目標：建立基本的 **Grafana Dashboard**，顯示 **Prometheus** 和 **VictoriaMetrics** 的數據

1. 創建 Dashboard：

- 點擊左側菜單的 **+** -> **Dashboard** -> **New Panel**。

2. 配置圖表：

- 選擇數據源 (VictoriaMetrics)。
- 使用以下 PromQL 查詢數據：
 - `up{job="victoriametrics"}`：查看 VictoriaMetrics 的目標狀態。
 - `scrape_samples_scraped`：查看 Prometheus 抓取的樣本數量。

3. 保存圖表：

- 點擊右上角 **Save Dashboard**。
- 為 Dashboard 命名，保存。

Day 5: 測試與優化

目標：確保系統穩定運行，進行性能測試與調整

1. 測試數據流是否正常：

- 在 Prometheus 界面查詢：

```
up
```

確認 Prometheus 是否成功抓取數據。

- 在 VictoriaMetrics 界面查詢：

```
http://<伺服器IP>:8428/metrics
```

確認數據是否存儲成功。

2. 性能調整：

- 增加 Prometheus 的抓取目標：在 `prometheus.yml` 文件中添加其他目標服務。
- 測試 Grafana 的可視化能力，添加更多圖表。

3. 優化配置：

- 設置 `Prometheus` 的數據保留時間（例如 15 天）。
- 為 VictoriaMetrics 優化磁盤存儲路徑，確保數據持久性。



完成後的成果

- **VictoriaMetrics** 成功存儲時序數據。
- **Prometheus** 持續抓取數據並寫入 VictoriaMetrics。
- **Grafana** 提供實時的數據可視化 Dashboard。

如果你需要進一步的詳細指導或遇到任何問題，可以隨時告訴我！