

# **Prometheus & VictoriaMetrics intro.**

# Intrduction

- Prometheus Remote Write & Remote Write

- 簡介
- Prometheus Remote Write
- Prometheus Federation
- Remote Write 與 Federation 的比較表
- 結論

- **VictoriaMetrics**

- 介绍
- 为什么选择 VictoriaMetrics?
- VictoriaMetrics 的功能亮点
- 单节点部署
- 集群模式的架构
- 部署步骤和配置
- 优势与限制
- 使用场景
- 配置与运维建议
- 与其他解决方案的比较
- 混合使用案例
- 总结与建议

## **Prometheus Remote Write & Remote Write**

# 簡介

- Prometheus 是一款開源監控系統，提供高效的時間序列數據存儲與查詢能力。
- Remote Write 和 Federation 各具特色。

# Prometheus Remote Write

- **長期存儲:** Remote Write 允許將數據持久化到外部存儲系統，適合需要長期保存的情況。
- **多集群數據匯聚:** 在多個 Prometheus 實例的架構中，將所有數據發送到一個集中式存儲，實現統一管理和分析。
- **災難恢復場景:** 將數據同步到遠程存儲系統，作為異地備份的一部分。

# Prometheus Remote Write

- **Prometheus Remote Write** 是一種數據推送機制，允許 **Prometheus** 將收集到的時間序列數據發送到遠端存儲系統，例如：
- **優點**
  - 長期存儲：將短期存儲的局限性交由外部存儲系統解決。
  - 集中式分析：支持將多個集群的數據匯聚到同一存儲系統。
  - 與大規模分析工具集成：可配合外部工具（如 **Grafana**）進行更強大的分析。
- **缺點**
  - 缺乏即時性：傳輸延遲可能導致數據的即時性下降。
  - 依賴外部存儲：需要安裝和配置額外的存儲系統，增加運維成本。
  - 不可直接查詢：**Prometheus** 無法直接查詢遠端存儲數據，查詢依賴於存儲後端的支持。

# Prometheus Federation

- Prometheus Federation 是一種數據共享機制，允許一個 Prometheus 實例（上層實例）通過 HTTP 抓取（scrape）另一個 Prometheus 實例（下層實例）的數據。這種方式適用於層次化監控架構。
- 例如：
  - 上層實例聚合來自多個下層實例的數據。
  - 僅抓取關鍵指標，減少上層負載。

# Prometheus Federation

- 優點

- 內建支持：不需要額外的存儲系統或服務。
- 數據即時性高：與 Prometheus 本地抓取數據的即時性相當。
- 數據過濾：支持僅抓取部分關鍵指標，減少不必要的負載。

- 缺點

- 適用於短期數據：不適合用於長期存儲。
- 負載影響：如果上層實例需要抓取多個下層實例，可能會對下層造成性能壓力。
- 有限數據處理能力：只適用於 Prometheus 的短期數據存儲窗口。



## Remote Write 與 Federation 的比較表

功能	Remote Write	Federation
用途	長期存儲和集中式數據分析	層次化監控和指標共享
數據傳輸方向	推送（Push）	拉取（Pull）
即時性	視網路和存儲延遲	高即時性
依賴性	依賴外部存儲系統	完全內建，無需外部依賴
數據篩選	通過 relabel 配置進行篩選	通過 match[] 配置抓取所需指標
適合場景	長期存儲、大規模數據匯聚	分層監控、即時數據共享

## 結論

- **使用情境總結:** Remote Write 更適合需要長期存儲的場景，Federation 則便於數據整合分析。
- **技術選擇建議:** 根據系統需求，若需集成多數據源優先考慮 Federation，否則選擇 Remote Write。
- **性能與成本平衡:** 在面對資源限制時，應考慮權衡數據存儲需求與系統性能的相對關係。

**VictoriaMetrics**

# VictoriaMetrics 简介与应用

- 介绍
- 为什么选择 VictoriaMetrics ?
- VictoriaMetrics 的功能亮点
- 单节点部署
- 集群模式的架构
- 部署步骤和配置
- 优势与限制
- 使用场景
- 配置与运维建议
- 与其他解决方案的比较
- 混合使用案例
- 总结与建议

# 介绍

- **数据存储必要性:** 在大规模监控场景中，强大的数据存储能力是确保系统可靠性的关键因素。
- **性能优势:** VictoriaMetrics 提供极高的查询性能和资源利用率，支持高并发的数据读写操作。
- **应用场景:** 适用于微服务架构、物联网等场景中，能够处理海量时间序列数据的需求。

# 為什麼選擇 VictoriaMetrics

- 存儲效率：其高效數據壓縮技術大幅降低存儲空間需求，提供卓越的存儲效能和數據管理能力。
- 查詢速度：使用優化查詢引擎，極大地縮短了查詢響應時間，支持即時數據分析。
- 多租戶支持：系統原生支持多租戶架構，便於不同業務單位獨立管理和控制其數據。

## VictoriaMetrics 的功能亮點

- Prometheus API 完全兼容：支持 Prometheus 的 API，方便現有系統的無縫遷移與集成。
- 高壓縮比：壓縮比可達 70 倍，有效減少存儲需求，提升資源利用率。
- 靈活部署模式：支持單節點和集群部署，可根據負載動態擴展，提供更高的可用性。

## 部署模式-單節點部署

- 適用於小規模環境，可快速部署並開始監控數據存儲。
- 配置示例：
  - 適用於每秒數據點少於 **100** 萬的場景。
  - 方法簡單，適合低並發需求的應用。



# 部署模式-集群模式架構

- **vmstorage**：存儲原始數據，提供高效的查詢能力並支持持久化存儲。
- **vminsert**：接收數據並存儲到 **vmstorage**。
- **vmselect**：負責處理查詢請求，從存儲中檢索數據。
- 架構特點：
  - 各組件協同工作，實現高效數據處理與存儲。
  - 適用於高並發、大規模數據場景。

## 部署步驟和配置

- Helm 部署：使用 Helm 簡化部署流程，執行命令即可自動管理配置與資源。
- 手動部署：使用 YAML 文件配置各組件，靈活性高但需要謹慎管理。
- 選擇建議
  - 單節點部署適合小規模場景。
  - 集群部署更適合高並發需求場景。

# 優勢與限制

- 高基數支持：可有效處理高基數數據，使監控系統更靈活。
- 低資源需求：存儲和計算資源消耗顯著低於其他解決方案。
- **限制**
- 基礎設施限制：  
對存儲節點數量有依賴，在設計基礎設施時需注意。

## 使用場景

- 大規模監控數據存儲：通過高效數據壓縮與管理支持大規模數據存儲與分析。
- 多租戶監控能力：支持多租戶架構，保障數據的獨立性與安全性。
- SLA 報告與合規性：提供靈活的 SLA 報告功能，優化系統可靠性與性能。

## 配置與運維建議

- 數據保留策略：使用 `-retentionPeriod` 參數設定數據存儲時間，避免浪費存儲資源。
- 監控建議：使用 **Grafana** 儀表板監控集群性能，實時了解系統狀態。
- 備份與恢復：定期執行數據備份，防止意外數據丟失。

## 與其他解決方案的比較

- 存儲效率：相較於 Prometheus 和 InfluxDB，VictoriaMetrics 提供更高效的壓縮和存儲。
- 查詢性能：在查詢響應時間上優於 Thanos 和 InfluxDB，更適合大規模數據場景。
- 運維簡化：相比 Prometheus 和 Thanos，VictoriaMetrics 部署與運維更加簡單。

## 混合使用案例

- 與 Prometheus 集成：結合使用可優化數據存儲與分析能力，提升系統整體性能。
- 應用場景：在微服務架構中，兩者配合實現高效監控與數據處理。
- 效益：提高數據處理速度與擴展性，降低運維成本。

## 總結與建議

- 高效與可擴展性：VictoriaMetrics 強調高效數據存儲與處理，特別適合大規模監控需求。
- 推薦場景：處理高基數、大規模時間序列數據的場景。
- 無縫集成：完全支持 Prometheus API，便於與現有監控架構集成。