



Tieni sotto controllo le tue applicazioni Delphi con Prometheus

Marco Breveglieri

Marco Breveglieri

Software Developer 🧑💻 | Trainer and Consultant 🧑🎓 | Tech Content Creator 🎥 | Community lover 💜



Homepage
<https://www.breveglieri.it>



Blog tecnico
<https://www.compilaquindiva.com>



Delphi Podcast
<https://www.delhipodcast.com>



Canale Twitch
<https://twitch.tv/compilaquindiva>

Tutti gli altri link:
👉 <https://linktr.ee/marco.breveglieri>

Iniziamo! 🙌

C'era una volta il Monitoring...

...da cui tutto ebbe inizio



Monitoring: cos'è? come funziona?

- Significa avere visibilità sullo stato di salute di applicazioni, servizi e architetture
 - Comprendere e diagnosticare cosa succede nei nostri sistemi software è fondamentale
- Basato sulla conoscenza di ciò che si vuole cercare
 - Approccio «check-based»
 - Responsi semplici: 🙌 OK / KO 🙄
- Adatto a infrastrutture relativamente statiche
 - Verifica dello stato tramite script dedicati
 - Utilizzo di tool dedicati «maturi» (es. *Zabbix, Nagios, ...*)

I limiti del monitoring

- Difficoltà nella gestione di sistemi complessi e distribuiti
 - Molti tool non supportano infrastrutture cloud
- Incapacità di rispondere a domande «impreviste»
 - Avvisi solo su cause riscontrate per problemi noti
- Capacità di prevenzione dei problemi limitata
 - Analisi post-incidente e risoluzione in emergenza
 - Utilizzo prevalente dei soli log per la diagnostica
- Mancanza di una visione «olistica» del sistema
 - Comunque, meglio averlo che non averlo 😊

Benvenuta Observability!

...e l'evoluzione del Monitoring



Observability vs Monitoring

MONITORING

- Ti dice se il sistema è sostanzialmente «rotto» (analisi statica)
- Si deve sapere in anticipo cosa potrebbe rompersi
- Si basa su semplici dashboard che mostrano gli stati noti
- Disponibili solo informazioni di base (es. file di log, registro eventi)
- Ambienti statici (asset poco variabili)

REATTIVITÀ

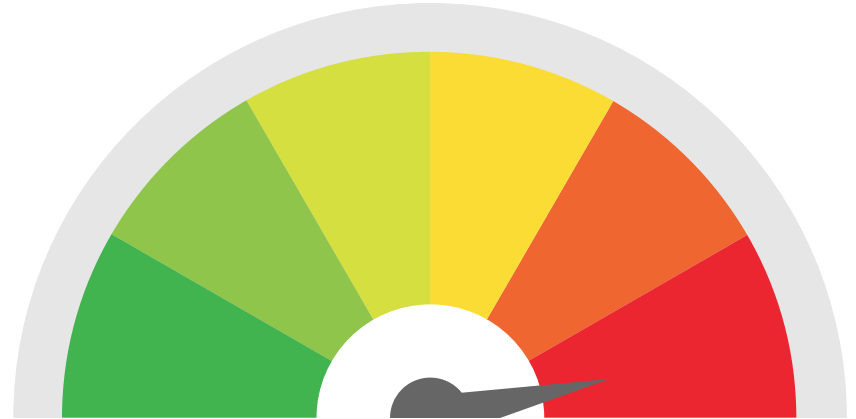
OBSERVABILITY

- Espone lo stato interno di un sistema complesso (analisi di trend)
- Potrebbe rispondere anche a domande non anticipate
- Utilizza diverse misurazioni e tool per diversi aspetti (metriche, log, eventi...)
- Raccolta e analisi di segnali di telemetria «ricchi e variegati»
- Asset dinamici (es. cloud, container, ...)

PROATTIVITÀ

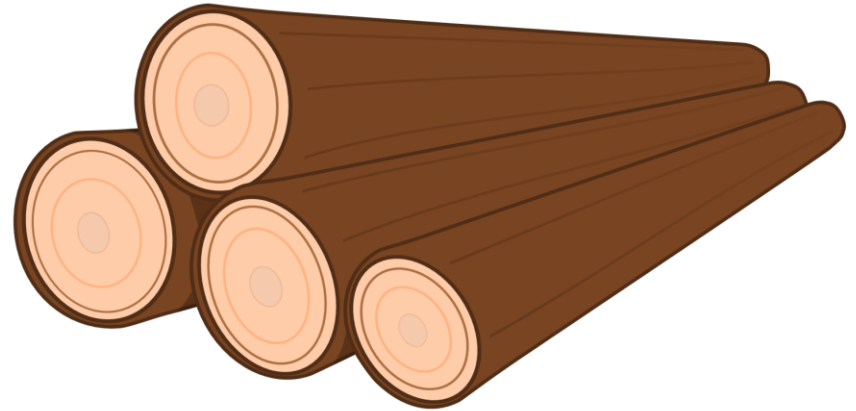
Metriche

- Sono rappresentazioni numeriche
 - Uso di CPU, RAM, I/O disco, banda di rete, ...
- Sono «time based» e «scoped»
 - Relazionate al momento in cui vengono raccolte
 - Decorate da informazioni di contesto
- Sono semplici ed economiche
 - Prometheus alloca 2 byte medi per data-point
- Si possono elaborare a fini statistici
 - KPI nel business sono ottenuti da metriche



Log

- Esistono sin dall'alba dei tempi
 - I computer hanno sempre scritto log a video
- Sono facili da implementare
 - Qualunque linguaggio è in grado di scriverli (a video e/o su file)
 - Delphi ha diverse librerie: *LoggerPro, QuickLogger, ...*
- Non sono generalmente strutturati
 - Righe di testo lunghe, formato JSON (e affini) solo in rari casi
 - Fatti per essere letti da umani



Trace

- Sono i «signal» meno osservabili
 - Non sono ottimali per risalire a tendenze e contesti
 - Tracce di funzioni o step nel codice, di interesse allo sviluppatore
- Utili ma onerosi da diversi punti di vista
 - Overhead per la generazione
 - Costo di storage e recupero dati
 - Sampling necessariamente circostanziato
- Possono essere anche distribuiti
 - Coinvolgono potenzialmente più sistemi (ad esempio, condividendo un «Trace ID»)



Prometheus

Conosciamolo!



Prometheus

Tool di monitoraggio di sistemi software e alerting

- Originariamente nato in SoundCloud
- Prodotto completamente open source
- Adottato e supportato da grandi aziende
- Progetto indipendente e autonomo
- Community di sviluppatori e utenti molto attiva
- Programmato in linguaggio Go
- Eseguibili nativi e cross-platform



Sito ufficiale (con risorse e documentazione)

👉 <https://prometheus.io>

Caratteristiche principali

- Prometheus è un tool molto utilizzato in ambito Observability
 - Fa una sola cosa e la fa bene: è focalizzato sulle metriche
 - Può interoperare con altri tool (es. Grafana)
- Progettato per memorizzare in modo efficiente «time series»
 - Usa valori di tipo numerico, espressi in formato semplice
 - Si tratta dei valori di più alto livello che puoi collezionare
 - ① Log e trace hanno più dettagli, ma occorre rimuoverli e/o aggregarli per ottenere le info di alto livello
- E' il tool a cui affidarti per poter rispondere alla domanda:
«Ehi, come stanno andando le cose?»

Che cosa fa?

- Recupera le metriche con un metodo «pull based»
 - Estrae i dati dai sistemi che tiene sotto controllo
 - Lavora in modo diverso rispetto ai tool di monitoring
 - Usano metodi «push based»
 - Usano tecniche di «check retries» degli stati
- Memorizza le metriche nel proprio TSDB specializzato



Come funziona?

- Contatta i servizi tramite endpoint per estrarre le metriche
- Offre una dashboard minimale per visualizzare i dati archiviati
 - Forma grafica
 - Forma tabellare
- Consente di interrogare il TSDB tramite query scritte in un semplice linguaggio: PromQL



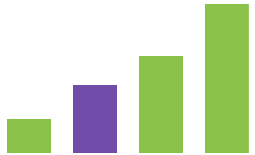
Le metriche



COUNTER: rappresenta un contatore sempre crescente
Esempi: numero di chiamate HTTP, conteggio errori



GAUGE: rappresenta un valore che può salire o scendere
Esempi: uso CPU, RAM allocata, temperatura



HISTOGRAM: campiona valori suddividendoli in range
Esempi: durate round-trip HTTP, tempi di risposta



SUMMARY: simile a Histogram, ma suddivide in «quartili»
Esempi: gli stessi di Histogram con range non noti

Come espongo le metriche?

➤ Esistono le Client Libraries!

- Semplificano l'instrumenting del proprio codice
- Consentono la raccolta dei dati per produrre le metriche
- Espongono a Prometheus i dati nel modo formalmente valido e atteso
- Si integrano con le architetture e i framework esistenti
- I linguaggi più diffusi sono supportati... anche Delphi!





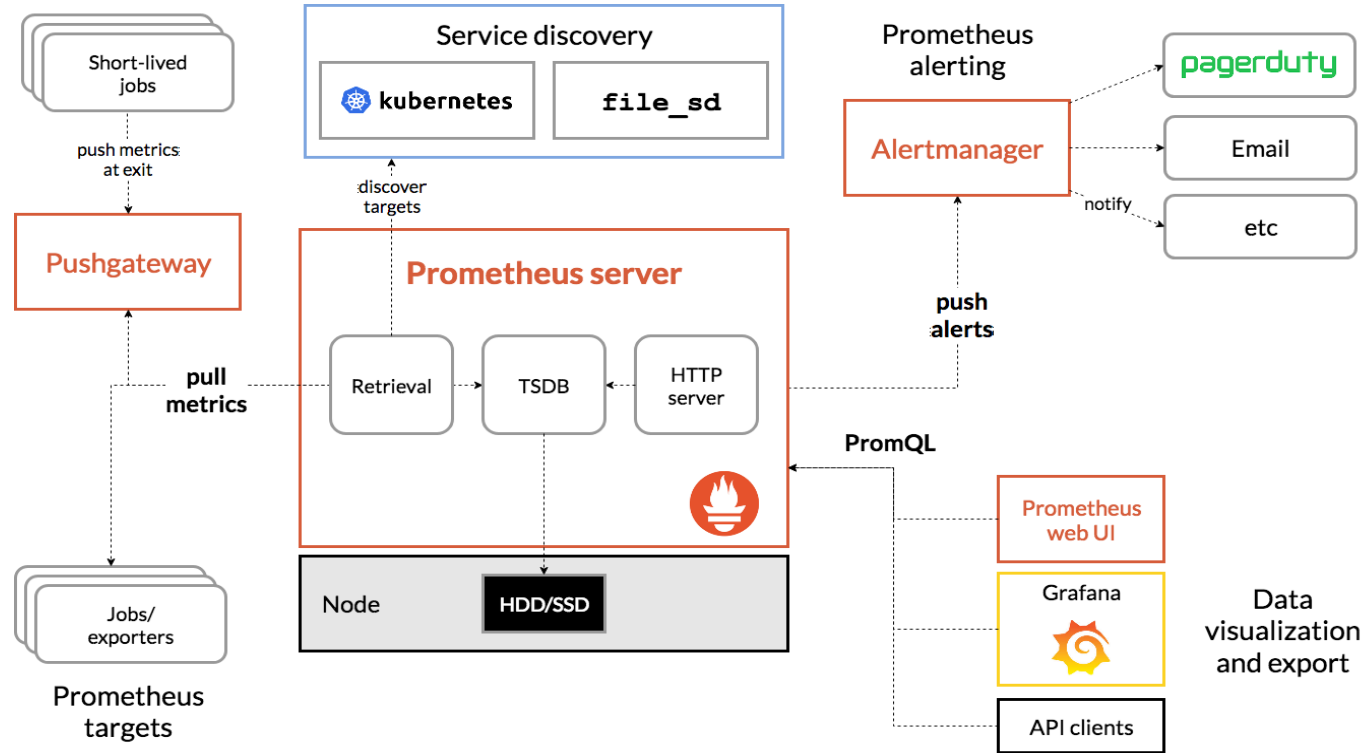
Demo time! 👍

Let's go Full Stack!

I tool «amici» di Prometheus



Panoramica dello stack



Alertmanager

- Supporta un sistema di alerting (opzionale): una vera killer feature!
 - Può mandare avvisi relativi a trend che persistono nelle metriche
 - Riduce al minimo il fenomeno delle cosiddette «silent failures»
 - Supporta un concetto di durata (for)



Prometheus



AlertManager



Demo time! 👍

Pushgateway

- Problema da risolvere
 - Job brevi o schedulati (es. script, batch, cron)
 - Non vivono abbastanza a lungo per essere «scrapati»
 - Serve un modo per l'invio delle metriche
- Soluzione: Pushgateway
 - I job inviano metriche al Pushgateway
 - Prometheus interroga il Pushgateway come fosse un target HTTP



Pushgateway Prometheus



Demo time! 👍

Grafana

Visualizziamo metriche e alert
in modo elegante!



Che cos'è Grafana?

- Web UI (anche) open source per creare dashboard visuali e interattive
 - Supporta molteplici data source (tra cui – indovinate un po' – c'è Prometheus)
 - Consente di creare grafici, tabelle, allarmi e alert visivi
 - Permette il monitoraggio in tempo reale
 - Visualizza grafici storici ed evidenzia trend
 - Ideale per monitoring, incident response, inspection, ...
 - È apprezzato sia dai «tecnici», ma anche dal «management» e dal «business»



Sito ufficiale (con risorse e documentazione)

👉 <https://grafana.com/>




Demo time! 👍

Conclusioni

Cosa ci portiamo a casa?



Cosa ci portiamo a casa?

- Lo stato di salute del vostro software è importantissimo!
 - Il monitoring non è sufficiente al giorno d'oggi
 - È necessario fare un passo avanti con l'Observability
- Prometheus + Pushgateway + Alertmanager + Grafana
 - Raccolta metriche, generazione e invio avvisi, visualizzazione
 - Soluzioni open source, adatte a ogni tipo di sistema
- Inizia ad evolvere già da domani le tue applicazioni Delphi!
 - Scarica la client library dal sito GitHub!
 <https://github.com/marcobreveglieri/prometheus-client-delphi>



Domande? 🖐️



Grazie!

