Università Politecnica delle Marche

Facoltà di Ingegneria
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica e dell'Automazione

Realizzazione di un sistema di monitoraggio per infrastrutture e sistemi IT

Relatore:
Prof. Alessandro Cucchiarelli

Candidato:
Mario Consorti

Anno accademico 2019/2020

Indice

Capitolo 1- Introduzione

Il progetto TD Monitor nasce dall'esigenza di tenere sotto controllo tutti i parametri e le performance del sistema Informatico di un Azienda o di un Organizzazione, mettendo a disposizione dell'amministratore di sistema una dashboard aggiornata, che, in tempo reale, sia in grado di fornire, in modo veloce e sintetico, tutte le informazioni necessarie alla risoluzione di problematiche e all'ottimizzazione delle risorse dell'infrastruttura.

In una infrastruttura IT complessa è necessario infatti avere sotto controllo tutti gli aspetti e tutti i componenti che la compongono, sia Hardware che Software.

Partendo dal presupposto che non è possibile controllare qualcosa che non conosciamo, abbiamo scelto di sviluppare una soluzione che, grazie a strumenti opportuni e avvalendosi delle tecnologie necessarie, sia in grado di raccogliere, analizzare e rappresentare nel modo più efficiente possibile tutti i dati necessari a monitorare un sistema informatico.

Al contempo la soluzione dovrà essere in grado di rappresentare tutte le informazioni raccolte in un'interfaccia semplice e di facile lettura, che permetta all'amministratore di sistema di avere il controllo completo dell'intero sistema riducendo così eventuali tempi di intervento e ottimizzando le risorse.

La soluzione consentirà di raccogliere in una o più dashboard tutti i controlli richiesti dall'utente.

Partendo dallo strato Hardware ad esempio sarà possibile avere in tempo reale lo stato delle porte di uno switch/router o di un qualsiasi apparato di rete (errori, velocità di connessione, throughput, temperatura, velocità delle ventole, stato del software, alimentatori, etc..). Si potrà inoltre conoscere lo stato dell'hardware e delle risorse di un PC o di un Server (CPU, RAM, spazio occupato, stato SMART del disco, memoria in uso, raggiungibilità, processi attivi etc..).

Per quanto riguarda la parte Software e applicativa invece sarà possibile monitorare tutti i parametri utili, dall'occupazione di banda o delle risorse allo stato di un processo, dai tempi di un'elaborazione all'esito di un backup e così via.

L'applicazione permetterà anche di raccogliere tutti i log necessari e di effettuarne un'analisi in tempo reale (ad esempio log di accesso di un webserver, di un OS o di un firewall).

I controlli ovviamente saranno customizzati in base alle esigenze dell'utente e alla struttura del sistema da controllare.

Nel sistema di monitoraggio sarà inoltre possibile configurare diverse tipologie di avvisi da inoltrare all'utente in caso di necessità, come ad esempio un utilizzo troppo elevato di una risorsa o l'avvenuto esito di un processo.

Scopo del lavoro illustrato in questa tesi è quello di descrivere in modo completo tutti i passaggi e le fasi che si sono susseguite nella realizzazione della soluzione di monitoring descritta.

Capitolo 2- Contesto Applicativo

2.1 Cos'è un sistema informatico?

Quando parliamo di "sistema informativo aziendale" parliamo di un sistema complesso ed eterogeneo progettato per organizzare ed elaborare grandi quantità di informazioni.

Parte integrante di un sistema Informativo è il "sistema informatico", quella parte cioè del sistema informativo costituita da ciò che viene comunemente definita infrastruttura IT, ossia un insieme di componenti hardware e software che, organizzati in architetture opportune, hanno il compito di elaborare dati e informazioni.

Il sistema informatico dunque ricopre all'interno dell'azienda un ruolo particolarmente importante, in quanto deve garantire affidabilità, efficienza, disponibilità e sicurezza dei dati e delle informazioni gestite.

L'amministratore di sistema è la figura professionale che si occupa della gestione e della manutenzione di tutti gli aspetti tecnici dell'infrastruttura IT e che quindi deve dotarsi di tutti gli strumenti necessari a garantire il corretto funzionamento di tutti i servizi e di tutti i componenti dell'infrastruttura.

Il progetto si svilupperà attorno all'esigenza di implementare un sistema in grado di fornire all'amministratore di sistema un software in grado di monitorare lo stato di tutte le risorse, di tutti i componenti e di tutti i processi di un sistema informatico aziendale.

È in questo contesto che l'azienda Tecnodata SRL svolge la sua attività dal 1987, progettando soluzioni gestionali ERP per medio-grandi aziende e gestendo in outsourcing architetture e soluzioni per la gestione delle infrastrutture IT.

Nei prossimi capitoli approfondiremo le attività e i contesti per i quali l'azienda ha scelto di sviluppare la soluzione di monitoring descritta in questa tesi.

2.2 Tecnodata SRL

Tecnodata[1] utilizza strumenti e standard di riferimento nel contesto tecnologico, orientandosi negli ultimi anni verso le tecnologie di punta del panorama “open source”.

Dal 2001 l’azienda ha acquisito il mandato di Concessione Zucchetti per i prodotti gestionali amministrativo – contabili di fascia alta.

Il sistema aziendale di gestione per la qualità, a partire dal 2004, è regolarmente certificato dal RINA in conformità alla ISO 9001.

Dal 2013 l’azienda opera esclusivamente in tecnologia web sia per i portali collaborativi che per il prodotto gestionale amministrativo – contabili Zucchetti ADHOC Infinity, in virtù delle competenze acquisite in progetti di grandi dimensioni.

Dal 2017 l’azienda è stata inclusa nel ristretto network Zucchetti dei Centri di Competenza Infinity.

Nel 2018 Tecnodata ha aderito al Consorzio IUSTEC, associazione di professionisti e aziende altamente qualificati, il cui scopo è di supportare le organizzazioni aziendali sanitarie ed industriali nel percorso di accountability e monitoraggio imposto dal GDPR, General Data Protection Regulation.

Tecnodata è inoltre partner certificato Sophos per tutto ciò che riguarda soluzioni relative alla Cybersecurity .

Andremo ora ad approfondire gli aspetti tecnici e organizzativi che l’azienda svolge per gestire in outsourcing architetture e soluzioni per le infrastrutture IT.

2.3 Soluzioni per l'infrastruttura IT di Tecnodata

Le soluzioni per l'infrastruttura IT offerte da Tecnodata sono rivolte a clienti che hanno necessità di dotarsi di sistemi capaci di adattarsi alle esigenze di ogni settore industriale e di supportare diversi modelli di business.

Ecco i servizi proposti dall'azienda per soddisfare le esigenze dei clienti che hanno scelto di gestire in outsourcing la propria infrastruttura IT :

- NOC – Network Operation Center, servizi di gestione dell'infrastruttura IT centralizzati : Mette a disposizione del committente tutta l'esperienza Tecnodata in termini di Networking, System Administration e Application Management

- SOC – Security Operation Center, servizi di gestione della sicurezza centralizzati : Fornisce supporto organizzativo per tutte le tematiche di protezione dell'infrastruttura IT, dotando il cliente di tutti gli strumenti necessari alla gestione a al controllo della stessa. Il team IT di Tecnodata fornisce supporto nell'attivazione di servizi proattivi e nell'attività di incident response.

- Inventory : è il servizio che si occupa di gestire tutta la documentazione relativa all'infrastruttura di rete e per la catalogazione e il controllo completo di tutto l'hardware e il software presente all'interno della struttura ICT

- Supporto Tecnico specializzato : Consulenza sistemistica per le quotidiane attività di monitoraggio, configurazione, controllo e amministrazione di apparati, applicativi di rete e sistemi operativi, grazie al portale di gestione delle segnalazioni disponibile 24 ore su 24, 7 giorni su 7.

2.4 Servizi NOC per l'infrastruttura IT

Approfondiremo ora la soluzione NOC (Network Operation Center) proposta da Tecnodata, ovvero il servizio che mette a disposizione delle aziende un centro operativo in grado di controllare in tempo reale tutte le funzionalità del proprio sistema informatico.

Il focus principale del servizio NOC è ovviamente quello di Risolvere tutti i problemi che possono presentarsi in ambito Networking, che siano essi rilevati da sistemi di controllo automatizzati predisposti in fase di installazione che segnalati manualmente dal personale dell'azienda che sceglie di dotarsi di un tale servizio.

Per ciò che riguarda i sistemi di controllo gli avvisi vengono captati e gestiti direttamente dal personale tecnico specializzato di Tecnodata, per quanto riguarda invece le segnalazioni aperte dagli utenti il tutto viene gestito in un portale di ticketing dedicato, ogni ticket sarà quindi smistato al centro servizi di competenza il quale, grazie a standard elevati e a best practices procederà alla risoluzione.

Un altro importante aspetto di una soluzione simile gestita in outsourcing consiste nel garantire al cliente una business continuity 24 ore su 24 e 365 giorni l'anno, proprio per questo il centro NOC esegue periodicamente sui sistemi gestiti una serie di operazioni programmate per garantire la massima affidabilità quali :

- Controllo delle patch di sicurezza dei sistemi e patching programmato
- Backup delle configurazioni dei sistemi e dei device di rete
- Restore programmato e test di quest'ultimi
- Reportistica su tutte le attività svolte

2.5 servizi SOC per le infrastrutture IT

Un altro importante servizio fornito da Tecnodata è il quello di Security Operation Center (SOC).

Il servizio consiste nel mettere a disposizione dei clienti un insieme di servizi finalizzati a monitorare e gestire tutti gli aspetti relativi alla sicurezza del sistema informatico.

Alla base dell'erogazione di un servizio SOC c'è quello che viene chiamato Security Assessment o Valutazione della sicurezza che si compone generalmente di due fasi:

- Valutazione della vulnerabilità: in questa fase vengono esaminati tutti i componenti sia Hardware che software che compongono l'infrastruttura IT per valutare se sono presenti già nel sistema vulnerabilità conosciute
- Penetration test: in questa fase, sfruttando vulnerabilità più o meno note e grazie a tecniche e tecnologie specifiche per ogni componente del sistema informatico, si cerca di mettere a nudo le suddette vulnerabilità

Una volta effettuate le procedure necessarie, si affianca il cliente nella messa a punto della soluzione migliore per mettere in sicurezza il proprio sistema ottimizzando la configurazione degli strumenti di cui è già in possesso oppure, nel caso in cui gli strumenti siano insufficienti, progettando e fornendo le soluzioni per la sicurezza più adatte per risolvere le vulnerabilità rilevate.

In questa fase si procede anche a quella che viene definita gestione della configurazione, cioè, mettendo a punto le policy e le regole relative al filtraggio o l'autorizzazione di connessioni attraverso uno o più firewall si regolano gli accessi in entrata e in uscita alla rete informatica aziendale.

Una volta completate tutte le fasi il sistema sarà pronto ad essere monitorato 24 ore su 24 dal team di Security management di Tecnodata attraverso strumenti opportuni.

2.6 Inventory

Un altro importante servizio che Tecnodata offre ai propri clienti è quello della gestione delle risorse IT aziendali.

Viene messo a disposizione del cliente uno strumento in grado di :

- Individuare e scansionare tutte le risorse collegate alla rete grazie a scansioni di tutte le diverse tipologie di reti (LAN, WAN, VPN, etc..)
- Gestire tutti gli asset sia hardware che software, permettendo di creare un inventario
- Gestire il ciclo di vita delle risorse
- Verificare validità e scadenze delle licenze software, garantendone conformità
- Monitorare i costi della gestione totale degli asset

Tale soluzione si compone di una suite di strumenti e servizi software che viene definito ITAM (IT Asset Management).

L'importanza per l'azienda di dotarsi di uno strumento del genere sta diventando negli anni sempre più significativa in quanto, essendo l'insieme delle risorse IT aziendali sempre più eterogeneo, è importante tenere sotto controllo ogni singolo aspetto di ogni singolo asset.

I vantaggi che una soluzione software ITAM porta con se sono ovviamente molti, oltre infatti a creare un inventario completo ed esaustivo dell'intero sistema IT, c'è il vantaggio di poter definire una corretta politica di gestione delle risorse, garantendo così un'ottimizzazione dei costi e, grazie al monitoraggio di licenze software e delle relative conformità, si ha la possibilità di ridurre potenziali rischi relativi a compliance GDPR e sicurezza .

Capitolo 3 – Obiettivi del Progetto

Alla luce di quanto descritto nel precedente capitolo è stato quindi necessario per l'azienda dotarsi di uno strumento modulare e scalabile in grado di integrare, automatizzare e ottimizzare tutte le attività sopra descritte.

La rapida e continua evoluzione dei sistemi informatici all'interno delle aziende infatti ha reso negli anni sempre più complesso monitorare e controllare tutte le risorse di cui un sistema informatico è composto.

L'idea del progetto denominato "TD Monitor" nasce allora dall'esigenza di tenere sotto controllo tutti i parametri e le performance di una Infrastruttura IT mettendo a disposizione dell'utente una dashboard aggiornata in tempo reale che fornisca, in modo veloce e sintetico, tutte le informazioni necessarie alla risoluzione di problematiche e all'ottimizzazione delle risorse dell'infrastruttura.

In questo capitolo andremo a mettere in evidenza quelli che sono stati gli obiettivi del progetto, analizzando tutti i requisiti funzionali che questo ha dovuto soddisfare per poter assolvere al meglio ai compiti richiesti.

3.1 Monitoring e linee guida

Abbiamo visto nel capitolo precedente (2.4) come conoscere e monitorare lo stato della rete, degli apparati hardware e delle risorse software all'interno di un'azienda è quindi necessario non solo per risolvere eventuali problemi che possono manifestarsi ma soprattutto è cruciale nel prevenirli garantendo così la continuità operativa dei sistemi informatici che si andranno a monitorare.

Il monitoraggio delle risorse IT o "IT monitoring" può essere realizzato attraverso metodiche diverse.

Le principali sono definite nel framework ITIL, un insieme di linee guida e di regole che forniscono strategie efficaci per l'erogazione dei servizi IT definendo in modo chiaro i processi e le responsabilità in modo da poter garantire la migliore operatività possibile nell'erogazione dei servizi IT, ed è proprio a queste linee guida che si è scelto di ispirarsi per poter realizzare un sistema di monitoraggio il più efficiente possibile.

Le tipologie di monitoring che il nostro sistema implementa sono sostanzialmente due, un monitoring di carattere osservativo e uno di carattere analitico.

Per quanto riguarda il primo si tratta di un sistema che si limita ad osservare le risorse, siano esse sia hardware che software, del sistema esaminato e a fornire in output le indicazioni sull'operatività dei singoli componenti del sistema, lanciando opportuni allarmi in caso di malfunzionamenti o guasti.

Il secondo invece esegue anche un'analisi approfondita dei comportamenti del sistema monitorato, cercando di determinare quale è stata la causa di eventuali problemi o disservizi e fornendo all'utilizzatore informazioni più complete per la risoluzione di eventuali guasti.

Quest'ultima tipologia in alcuni ambiti viene spesso affiancata a sistemi AIOps (Artificial Intelligence for IT Operations), ossia a sistemi di intelligenza artificiale in grado anche di prevedere l'insorgere di eventuali problemi sulla base dei dati analizzati.

Tuttavia, in questa tesi non prenderemo ancora in considerazione quest'ultima tipologia di analisi dei sistemi IT.

3.2 Privacy by design

Un altro obiettivo fondamentale del progetto è stato quello di tenere conto, sia nella scelta degli strumenti per la raccolta dati, che nella scelta degli strumenti per la loro elaborazione e visualizzazione, del principio di “privacy by design”.

Il regolamento generale sulla protezione dei dati, noto anche come GDPR (Regolamento UE 679/2016 [2]) introduce infatti nell’articolo 25 l’obbligo di applicazione di quello che viene definito “principio di protezione dei dati fin dalla progettazione” (o “privacy by design”), secondo la normativa infatti : “…si deve tener conto dello stato dell’arte, dei costi di attuazione, nonché della natura, dell’ambito di applicazione, del contesto e della finalità del trattamento, sia al momento di determinare i mezzi del trattamento sia all’atto del trattamento stesso il titolare del trattamento mette in atto misure tecniche organizzative adeguate , volte ad attuare in modo efficace i principi di protezione dei dati…”.

Gli aspetti più importanti per la fase di progettazione, riportati nell’articolo 25, nel nostro caso sono ovviamente lo stato dell’arte e le misure di sicurezza da implementare per garantire la protezione dei dati.

Quello che si definisce stato dell’arte nel nostro caso non è infatti un qualcosa di statico: se infatti è necessario che le corrette misure di sicurezza siano adottate oggi in fase di progettazione è vero allo stesso modo che queste debbano garantire un adeguamento continuo allo stato dell’arte nel tempo.

Sarà quindi importante in fase di progettazione scegliere tutti i componenti in modo che garantiscano l’implementazione del principio di privacy by design sia al momento della realizzazione, sia durante tutta la vita futura del progetto, perseguendo in modo reattivo il principio di adeguamento allo stato dell’arte.

3.3 Raccolta dei dati

Il primo passo nella progettazione di un sistema di monitoraggio è quello di definire la migliore strategia per la raccolta dei dati.

In questa fase è stato necessario definire quali dati raccogliere affinché questi soddisfino tutti i requisiti elencati nei capitoli precedenti.

Poiché il nostro sistema dovrà monitorare un insieme complesso ed eterogeneo di risorse sia Hardware che Software è opportuno ora definire due tipologie di strategie per la raccolta dati, una prima basata su un software che definiremo “agent” che si occuperà di collezionare dati e metriche che definiremo “agent based monitoring”; e la seconda che non richiederà l’installazione di nessun software per la raccolta dati che definiremo “agentless monitoring”.

Analizziamo ora nel dettaglio questi due approcci alla raccolta dei dati, mettendo in luce vantaggi e svantaggi di ognuno.

Come già accennato l’agent based monitoring necessita di con componente software (l’agent appunto) che ha il compito di raccogliere i dati necessari (log, metriche, ecc..) a effettuare le operazioni di monitoraggio richieste e presenta i seguenti vantaggi :

- Miglior affidabilità: l’agent infatti può disporre di una cache in grado di memorizzare per il tempo necessario i dati monitorati garantendo così, anche in caso di disservizi relativi alla connetività, di non perdere informazioni
- Miglior Sicurezza: poiché è l’agent ad occuparsi dell’invio dei dati del nodo/device monitorato al server centrale di monitoraggio, ciò evita che il nodo debba essere monitorato in modo diretto, riducendo così eventuali superfici di attacco
- Più Completo: l’agent based monitoring permette di raccogliere una più grande varietà di informazioni

- Più efficiente: poiché i dati vengono raccolti e filtrati già localmente nel nodo monitorato.
  Questo ottimizza di molto l'occupazione di banda necessaria ad inviarli al server centrale di monitoraggio

Di contro analizziamo invece ora i vantaggi che può portare un sistema di monitoraggio "agentless", che non sfrutta cioè nessun software per la raccolta dati ma si occupa di andare a reperire direttamente le informazioni nel nodo monitorato:

- Meno oneroso da mantenere: nell'agent based monitoring sarà infatti necessario tenere aggiornato il software che si occuperà di raccogliere i dati, ciò non è necessario invece quando parliamo di agentless monitoring
- Più "leggero" per il nodo da monitorare: un altro vantaggio dell'agentless monitoring è quello di "pesare" meno sulle risorse del nodo locale (come RAM e CPU) visto che non dovrà essere installato nessun agente di monitoraggio
- Meno invadente: non richiedendo l'installazione di un agent infatti non sarà necessaria nessuna fase di distribuzione dell'agent sui nodi
- Più facile e veloce da implementare

In conclusione, quindi possiamo affermare che un monitoraggio senza agenti software distribuiti e installati sui nodi da monitorare è sì più facile e veloce da implementare, tuttavia presenta delle carenze in termini di varietà di informazioni che sarà possibile invece collezionare con un sistema basato su agenti, non tutti i sistemi e non tutte le applicazioni che si andranno a monitorare infatti sono in grado di fornire in modo autonomo informazioni utili sul loro stato.

Nell' agentless monitoring inoltre si dovrà prevedere il fatto che il nodo monitorato dovrà comunicare con il server centrale di monitoraggio utilizzando porte diverse, ciò aumenterà l'area d'attacco a disposizione di eventuali malintenzionati, e implicherà

quindi la creazione di opportune regole sui firewall del sistema, per non parlare del fatto che per raccogliere certi dati potrebbe essere necessario effettuare l'accesso con un utente che abbia privilegi di amministratore al nodo monitorato da remoto.

Tutto ciò non avviene invece per il monitoring basato sull'utilizzo di agent, poiché, come già detto sopra, le comunicazioni sono gestite internamente all'agent installato nel nodo monitorato, riducendo di molto la superficie di attacco e garantendo oltre ad una sicurezza migliore anche una sensibile riduzione del traffico di rete necessario per trasmettere i dati monitorati, ottimizzando la banda di rete necessaria, cosa che invece non avviene nell'agentless monitoring dove viene introdotto necessariamente traffico aggiuntivo.

Abbiamo visto quindi come entrambe i metodi di raccolta dati per il nostro sistema di monitoraggio presentano dei pro e dei contro, concludendo possiamo affermare che un sistema di monitoraggio agentless può prestarsi meglio a monitorare risorse come dispositivi di rete o dispositivi di archiviazione, mentre se consideriamo risorse come Server, PC o applicativi è molto meglio affidarsi ad un sistema orientato all'utilizzo di un agent.

Poiché nella fase di progettazione del sistema di monitoraggio realizzato sono state prese in esame risorse eterogenee, che andavano appunto da uno switch di rete a un application server, si è scelto di adottare in una soluzione "ibrida" entrambe le tipologie di raccolta dati, sfruttando i pro di entrambe a seconda del tipo di infrastruttura da monitorare.

3.4 Dashboard e Alerting

Lo step finale nella realizzazione di un sistema di monitoring come quello descritto in questo elaborato può essere suddiviso in due fasi, una prima fase è quella di filtrare tutti i dati, i parametri misurati (che chiameremo da ora in poi metriche) e i log collezionati dal sistema di raccolta dati e rappresentarli al meglio in un cruscotto (o dashboard) fruibile in modo semplice e immediato dall'utilizzatore finale.

La seconda fase invece consiste nel fornire all'utente finale la possibilità di ricevere allarmi o notifiche (alert) nel caso in cui vengano rilevati malfunzionamenti o irregolarità nei dati raccolti.

Analizzeremo ora in modo più dettagliato i requisiti da soddisfare affinché sia il sistema di rappresentazione che quello di alerting soddisfino a pieno le specifiche del progetto.

Iniziamo con l'analizzare i vantaggi che si possono trarre dalla rappresentazione attraverso una dashboard dei dati raccolti.

Con il termine dashboard andiamo a definire un insieme di oggetti o più in generale di rappresentazioni grafiche che, con una sola occhiata, permettono di visualizzare in modo veloce e immediato una certa quantità e una certa varietà di informazioni in tempo reale.

Tra i principali vantaggi della visualizzazione dei dati attraverso una dashboard abbiamo:

- Visibilità immediata dei dati raccolti
- Interfaccia grafica personalizzabile e facilmente comprensibile
- Visualizzazione contemporanea e nella stessa dashboard di dati raccolti da fonti di diversa natura
- Conoscenza, in tempo reale, dello stato delle risorse che sto monitorando

Poiché il nostro sistema di monitoring, come già accennato, dovrà essere accessibile da remoto, la nostra dashboard sarà fruibile da una pagina web messa a disposizione dell'utente che non dovrà fare altro che effettuare un login e visualizzare le informazioni.

All'interno della dashboard dovrà essere possibile inoltre interagire con i grafici e gli oggetti che la compongono, dando così all'utente finale la possibilità di filtrare i dati rappresentati.

Il sistema di alerting invece dovrà occuparsi di inviare all'utente finale delle notifiche in caso di disservizi, malfunzionamenti o superamento di certe soglie preimpostate.

Questo ovviamente comporta il fatto che esso debba essere perfettamente integrato nella nostra dashboard, l'utente infatti dovrà poter effettuare le seguenti operazioni:

- Impostare soglie nei grafici per poter indicare oltre quale valore far scattare l'alert.
- Scegliere il canale di comunicazione con cui inviare gli alert, come ad esempio SMS, email, canale Telegram o più semplicemente un messaggio pop-up mostrato sulla dashboard o la riproduzione di un suono.

3.5 Monitoraggio Personalizzato

Concludiamo questo capitolo dedicato alla descrizione degli obiettivi parlando della customizzazione del nostro sistema di monitoraggio IT e degli step necessari affinché venga realizzato un software di monitoraggio adatto alle esigenze del cliente.

Come già introdotto in precedenza infatti il progetto nasce dall'esigenza di fornire all'utente finale un prodotto cucito su misura per il proprio sistema informatico, che sia quindi in grado di raccogliere, rappresentare e monitorare efficacemente e in tempo reale i dati vitali del sistema IT aziendale.

Per fare ciò si dovrà necessariamente eseguire un'analisi di tutto il sistema IT da monitorare: questo permetterà di avere una completa visuale sui sistemi utilizzati. Come già descritto nel secondo capitolo infatti Tecnodata è in grado di fornire consulenze specialistiche anche in questo senso, una volta completata la fase di analisi si procederà quindi alla valutazione delle risorse in modo da poter identificare quelle da monitorare.

Completata la fase di identificazione delle risorse da monitorare si procederà infine alla realizzazione del sistema di monitoraggio seguendo le regole e gli obiettivi sopra descritti.

Capitolo 4 – strumenti utilizzati

In questo capitolo andremo ad elencare ed analizzare i componenti e gli strumenti utilizzati per la realizzazione del sistema di monitoring.

Tutti gli strumenti che vedremo sono stati selezionati dopo vari test e rispettando le regole, le esigenze e gli obiettivi descritti nei capitoli precedenti.

4.1 – configurazione del server centrale

Per la realizzazione del nostro sistema di monitoring si è optato per adottare una architettura di tipo centralizzato, composta sostanzialmente da un server centrale che si occuperà di ricevere, analizzare e rappresentare i dati collezionati dagli agenti e da cui partiranno tutte le chiamate per analizzare le risorse “agentless”.

Fulcro del nostro sistema sarà quindi il server di monitoraggio, la macchina cioè sulla quale verrà installata la nostra dashboard.

Il server ovviamente potrà essere un hardware fisico ma anche una macchina virtuale o in un container.

4.1.1 dimensionamento del server e scelta dell’OS

Se parliamo di server dovremmo fare i conti con il dimensionamento delle risorse di sistema.

Affinché vengano garantite prestazioni ottimali per tutto il sistema infatti le risorse come CPU, RAM e disco dovranno essere dimensionate in modo opportuno.

A tal proposito faremo riferimento più avanti ai requisiti minimi di sistema in base ai componenti software che adotteremo e che andremo ad analizzare in questo capitolo a partire dal sistema operativo.

Il sistema operativo scelto per ospitare il nostro sistema è CentOS [9], un sistema operativo derivato da Red Hat enterprise Linux, CentOS è un sistema gratuito e open source.

Oltre al vantaggio prettamente economico tuttavia CentOS presenta altri vantaggi, tra i quali:

- Security By Design: il sistema viene fornito con una serie di componenti come SELinux (layer di sicurezza che definisce i domini di accesso di una applicazione alle risorse di sistema) e FirewallD (attraverso il quale è possibile configurare le regole del firewall di sistema, il port forwarding, etc..)
- Stabilità: adottando componenti stabili CentOS riesce a ridurre sensibilmente problemi dovuti alla presenza di eventuali bug
- Supporto: seguendo la stessa timeline del supporto di Red Hat enterprise Linux, CentOS può godere di un supporto di circa 10 anni, di cui 5 di supporto completo e 5 di supporto di mantenimento
- Containerizzabile: CentOS è disponibile anche in versione container, ciò permette di sfruttare tutte le sue potenzialità anche in ambienti come Docker o Kubernetes.

Tornando alla questione del dimensionamento delle risorse, la versione di CentOS che ospiterà il nostro software di monitoring è la versione 7 che in termini di risorse richiederà un minimo di 1Gb di RAM e un minimo di 10Gb di spazio disco.

Nel nostro caso tuttavia forniremo alla macchina/VM che farà da server 2Gb di RAM e almeno 20Gb di spazio disco.

4.2 I Time series data

Introduciamo ora il concetto di time series, per time series si intende una collezione di informazioni temporalmente etichettate, che si usa caratterizzare processi tempo-dipendenti.

Ogni time series conterrà una certa quantità di dati, che chiameremo “time series data”.

Ogni time series quindi potrà essere analizzata per fornire informazioni non solo sullo stato attuale del sistema monitorato, ma anche su quello precedente e per effettuare proiezioni future.

Negli ultimi dieci anni, l’utilizzo delle time series è cresciuto notevolmente soprattutto nei settori finanziari, gestionali e IT, questo grazie ai grandi passi in avanti che sono stati fatti in termini di innovazione tecnologica relativamente alla loro memorizzazione, alla loro gestione, al tempo di vita delle time series e alle modalità con cui vengono raccolti i dati.

Definiamo “periodo”, il tempo che passa tra due punti di una time series.

Il periodo è solitamente legato alla frequenza con cui i dati vengono raccolti e potrà variare da secondi fino a minuti o ore ma anche mesi o anni e questo grazie a strumenti in grado di analizzare le informazioni e gestirle in modo efficiente.

Concludiamo dicendo che una time series può essere generata dai dati raccolti da un sensore, costruita in base a eventi o comportamenti o, come nel nostro caso fornita da un agent di monitoraggio o da un device.

4.2.1 il Time series Database

Per time series database (TSDB) [3] intendiamo una struttura software implementata per contenere e gestire i time series data: è quindi un database che ha lo scopo di memorizzare, identificare univocamente e gestire misurazioni e metriche contenute nei time series data.

Sebbene siano nati per la ricerca e l'analisi di dati in ambito finanziario, i time series database con il tempo si sono evoluti per contenere e trattare anche altre tipologie di dato.

I TSDB utilizzano generalmente algoritmi di compressione che consentono di ottenere un alto grado di efficienza anche nei casi in cui la frequenza di raccolta delle time series è molto elevata.

La grande diffusione dei time series database ha contribuito indubbiamente a superare quella che è stata per lungo tempo la suddivisione tradizionale tra database relazionali e non relazionali, in questo caso infatti i concetti di chiavi e tabelle connesse tra loro sono superati e tutto l'interesse viene concentrato sul fattore tempo.

Ciò permette indubbiamente di gestire anche una grande mole di dati in tempo reale, analizzarne le informazioni che più ci interessano ma con un approccio decisamente innovativo e più efficiente, caratteristica che ha permesso ai time series database di trovare largo impiego anche nella raccolta di dati real-time per monitorare ad esempio le risorse computazionali utilizzate anche in sistemi cloud, nonché le performance e la stabilità di apparati di networking o di database e applicativi, che sono proprio i casi che analizzeremo in questo elaborato.

4.2.2 InfluxDB

La scelta sul miglior time series database da adottare per realizzare il nostro progetto e raccogliere in modo efficiente le metriche dai diversi device e sistemi è ricaduta su InfluxDB [4], un TSDB sviluppato dalla InfuxData in grado di eseguire query in real time su enormi quantità di dati, è quindi il database ideale da utilizzare in una soluzione di monitoring come quella che andremo a realizzare.

È oltretutto semplice da installare (come vedremo più avanti) e utilizza un linguaggio chiamato InfluQL molto simile a SQL.

Essendo un time series database quello che in InfluxDB viene definito "measurement" (misura) corrisponde al concetto di tabella nel modello dei database relazionali, mentre la chiave primaria è sempre definita dal "timestamp" che può anche essere generato automaticamente dal database stesso.

InfluxDB è il componente centrale dello stack TICK, composto da Telegraf, Chronograf e Kapacitor.

Telegraf è un software client che dispone di diversi plugin e consente di raccogliere dati da fonti di natura diversa, Chronograf è un interfaccia web che mette a disposizione degli utenti una dashboard per mostrare i dati raccolti e permette all'admin di configurare tramite interfaccia grafica i parametri necessari al funzionamento dello stack, Kapacitor invece è un componente in grado di elaborare e trasmettere dati da InfluxDB.

In figura è riportato lo schema di funzionamento dello stack TICK:

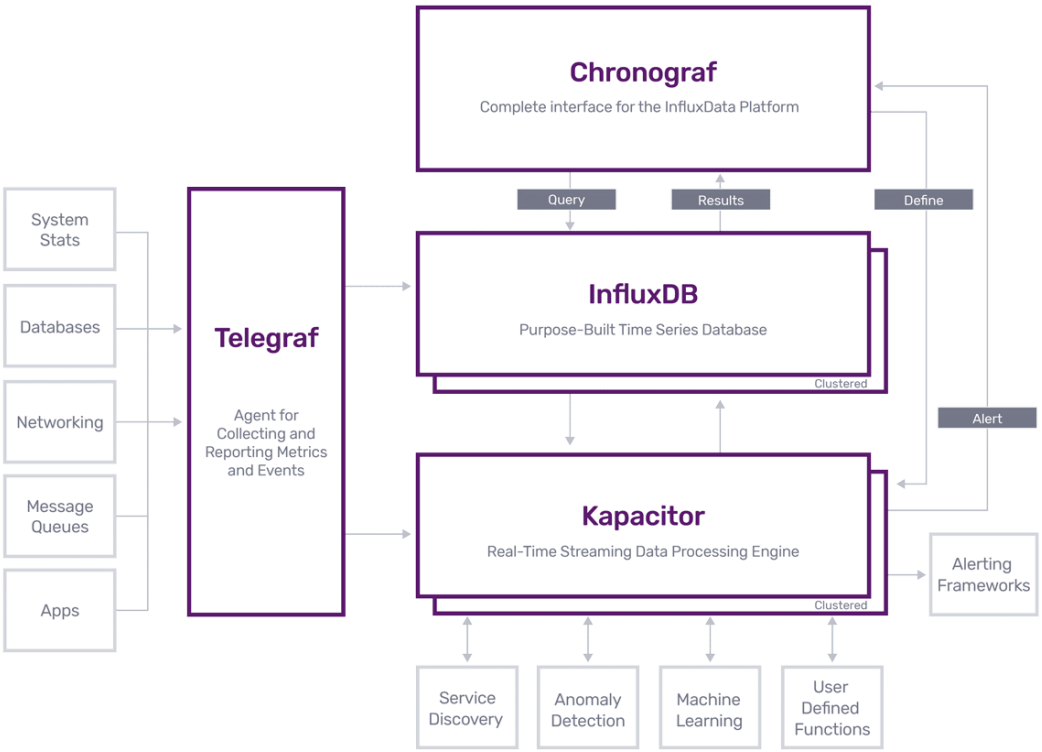

Figura 1 – schema di funzionamento stack TICK

Tuttavia, per il nostro progetto ci limiteremo a sfruttare solamente due dei componenti dello stack TICK, InfluxDB, che sarà ovviamente utilizzato come database centrale del nostro sistema di monitoraggio e telegraf che sarà il nostro agent per la raccolta dei dati.

Come vedremo nei successivi capitoli infatti per quanto riguarda la dashboard adotteremo uno strumento molto più completo e versatile di Chronograf chiamato Grafana.

4.3 Telegraf per la raccolta delle metriche

Abbiamo approfondito nel terzo capitolo la differenza tra monitoring con agent e agentless, come già detto nel nostro caso abbiamo deciso di implementare una soluzione ibrida che sfruttasse i vantaggi presenti in entrambe le modalità.

Ecco che allora è stato necessario scegliere una soluzione che permettesse di implementare efficacemente una soluzione di monitoraggio che faccia uso di un software per la raccolta dei dati e delle metriche necessarie (l'agent appunto).

La scelta, come già accennato sopra, è ricaduta su Telegraf [5], un agente opensource che si basa sull'utilizzo di appositi plug-in per la raccolta e l'invio di metriche da: sistemi, sensori IoT o database.

Telegraf è scritto in GO, ciò significa che è un file binario compilato e autonomo in grado quindi di essere eseguito su qualsiasi sistema senza bisogno di far riferimento a dipendenze esterne.

La community di Telegraf con il tempo è riuscita a mettere a disposizione degli utenti oltre duecento plug-in per la raccolta dei dati da diverse tipologie di endpoint, rendendo davvero molto semplice il monitoraggio e la raccolta di metriche da fonti eterogenee.

Il sistema di plug-in di Telegraf permette di aggiungere facilmente, come vedremo più avanti, nuovi input e output.

Ciò rende quindi questo strumento l'agent ideale da adottare nella nostra soluzione di monitoring.

4.4 La Log Analysis

Un altro aspetto fondamentale che il nostro sistema di monitoring dovrà gestire è quello dell'analisi dei log provenienti dai server, dai database e da altre tipologie di endpoint. a tale scopo è stato necessario individuare lo strumento più adatto e meglio ottimizzato per raccogliere e analizzare i log.

Se infatti, come già visto, InfluxDB è un database ottimizzato per collezionare e gestire serie temporali ciò non è altrettanto vero per quanto riguarda l'analisi dei log.

Consideriamo infatti che un file di log può essere anche composto da migliaia di righe, inoltre applicazioni e sistemi possono arrivare a scrivere sul file svariati record nell'arco di pochi secondi.

Queste scritture possono essere molto complesse da analizzare e interpretare ma contengono informazioni preziosissime per l'amministratore di sistema che si troverà ad effettuare eventuali troubleshooting.

Poiché quando parliamo di log parliamo essenzialmente di testo, ecco che allora sarebbe opportuno trovare uno strumento in grado di memorizzare tutte le informazioni all'interno del log ed effettuare una ricerca che permetta di individuare anche termini specifici nel minor tempo possibile.

A tal scopo è stato individuato in Elasticsearch lo strumento ideale.

Nel capitolo successivo andremo ad analizzare in modo più approfondito questo strumento e descriveremo le ragioni che ci hanno portato a sceglierlo per l'analisi dei log.

4.4.1 Elasticsearch e lo stack ELK

Elasticsearch [6] è un database NoSQL nato nel 2010, costruito sulla base di uno dei più popolari sistemi di indicizzazione, Apache Lucene. Con il tempo si è affermato per la sua capacità di effettuare ricerche full-text e per essere altamente scalabile.

Il processo di indicizzazione di Elasticsearch si basa sugli indici, che sono equivalenti a livello logico, ai database nell'ambito relazionale.

Possiamo suddividere gli indici a loro volta in frammenti o "shard" distribuiti su tutte le macchine che compongono il cluster di ricerca.

È proprio la capacità di distribuire gli shard e di ottimizzare lo spazio occupato che rende Elasticsearch molto performante in termini di distribuzione e parallelizzazione dei dati.

L'aspetto più interessante tuttavia nel nostro caso è quello che viene definito indice invertito.

Per indice invertito si intende una mappa in cui da un lato abbiamo tutti i termini memorizzati e dall'altra i documenti in cui questi si trovano, ciò consente di effettuare query molto performanti.

Elasticsearch è parte integrante di quello che viene chiamato stack ELK [6] che comprende:

- Elasticsearch: come già detto, database NoSQL basato sul motore di ricerca Apache Lucene
- Logstash: un potente strumento in grado di collezionare, esaminare ed elaborare log provenienti dalle fonti più disparate e in grado di esportare i dati verso Elasticsearch, nel nostro caso
- Kibana: una dashboard di visualizzazione

Oltre a questo che possiamo definire il cuore dello stack, dalle ultime release è anche disponibile una famiglia di “log shipper” o “log forwarder” chiamata Beats [6], in grado di raccogliere, oltre ai file di log (Filebeat), anche altre tipologie di dati come ad esempio metriche provenienti da svariati tipologie di device e sensori (Metricbeat) o dati relativi ai pacchetti che viaggiano sulla rete (Packetbeat).

Tuttavia come già detto, nel nostro sistema di monitoring adotteremo per la raccolta e l’analisi delle metriche Telegraf e InfluxDB: nel prossimo capitolo spiegheremo il perché.

L’architettura di base dello stack ELK è molto semplice ed è riassunte in figura 2:

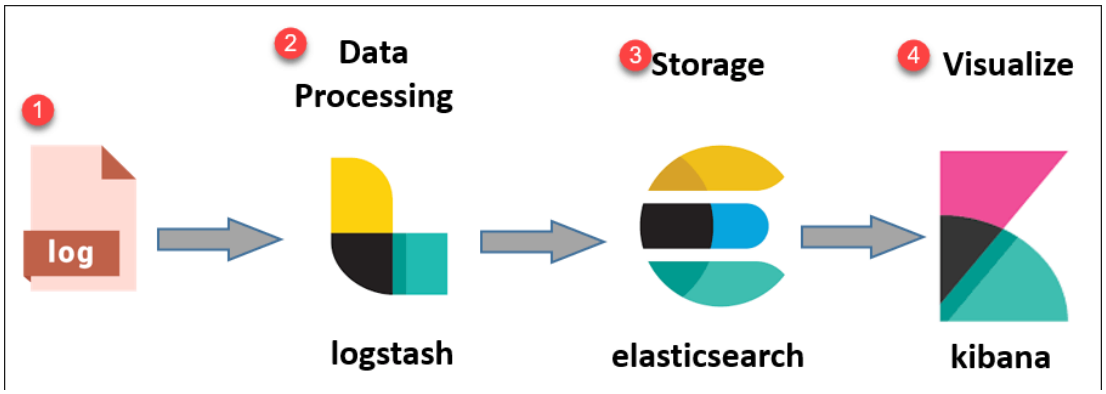

Figura 2 – architettura stack ELK

4.5 Confronto InfluxDB ed Elasticsearch per l'analisi delle metriche

Per quanto concerne la raccolta delle metriche, abbiamo visto come InfluxDB si riveli un eccellente strumento per archiviare ed elaborare le serie temporali.

D'altra parte abbiamo anche visto che, anche se ad una prima analisi Elasticsearch non sembra essere uno strumento nato per l'analisi delle time series, grazie alla sua scalabilità e alla velocità di memorizzazione potrebbe comunque essere un tool decisamente valido per assolvere a questo compito.

Ci troviamo quindi a dover scegliere tra, uno strumento nato e ottimizzato per l'analisi delle time series e un motore di ricerca progettato per immagazzinare anche grandi quantità di dati ed altamente scalabile.

Proviamo allora a ragionare su un caso pratico, consideriamo un sensore con una capacità di campionamento nell'ordine di pochi millisecondi.

In questo caso si dovranno gestire una grande mole di dati in un lasso di tempo relativamente breve.

In un caso come questo InfluxDB si è rivelato lo strumento migliore, infatti da alcuni benchmark eseguiti InfluxDB ha superato Elasticsearch in due test in particolare, quello dello spazio su disco e quello sul throughput di scrittura, dove InfluxDB ha occupato nove volte meno spazio su disco rispetto ad Elasticsearch e con throughput di scrittura maggiore di quasi quattro volte.

Le query testate su InfluxDB inoltre sono stati circa otto volte più veloci rispetto alle query memorizzate in cache da Elasticsearch.

In figura 3, 4 e 5 i grafici [7] che riportano quanto detto:

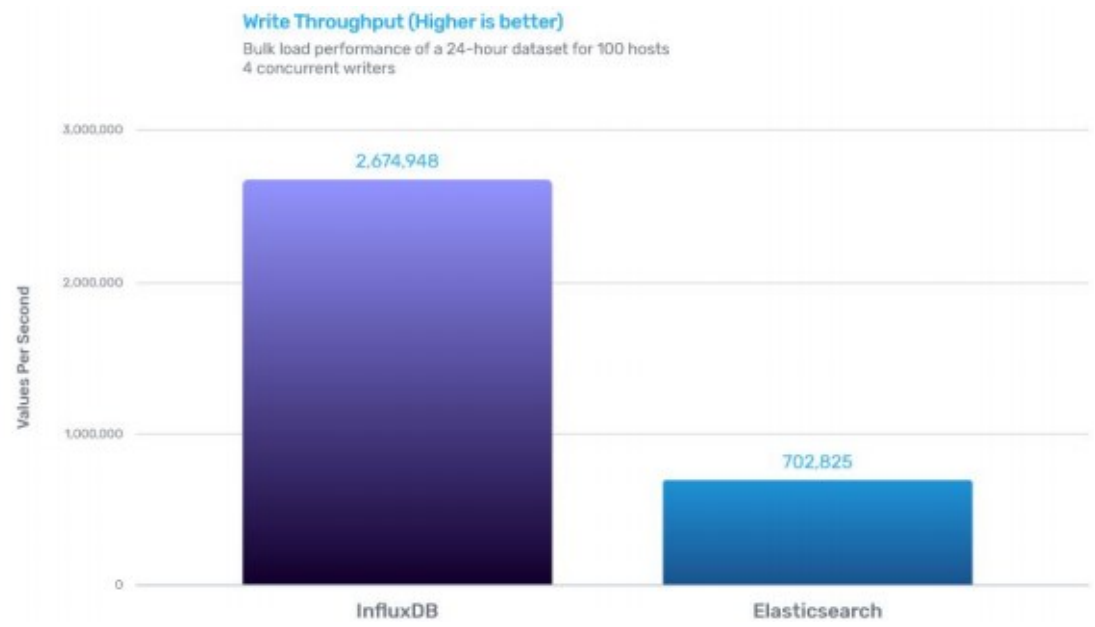

Figura 3 – confronto Write throughput

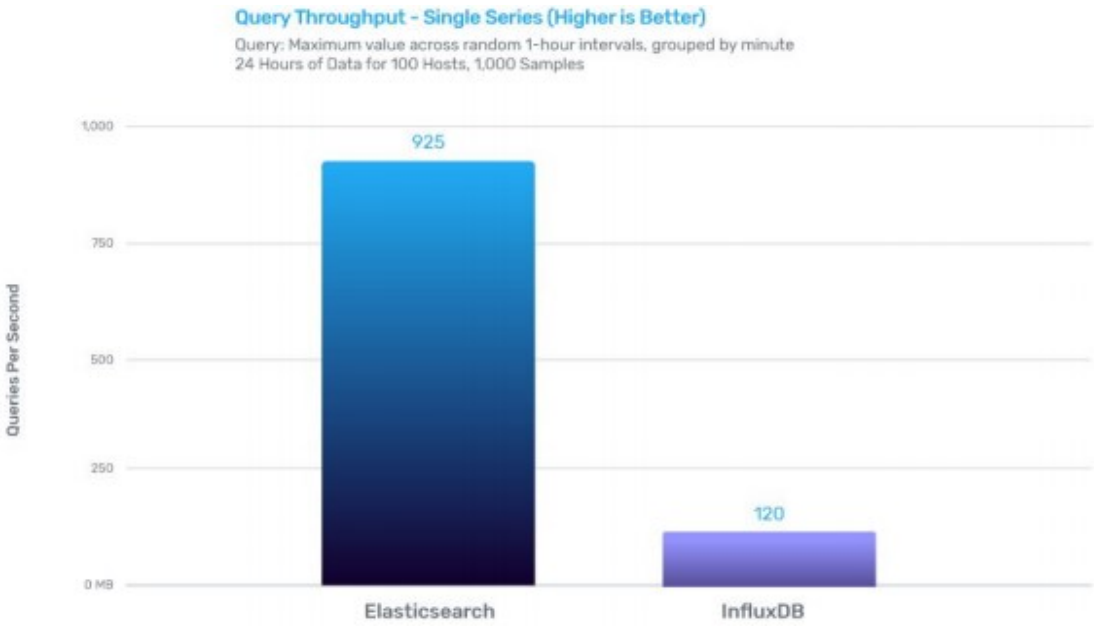

Figura 4 – confronto Query Throughput

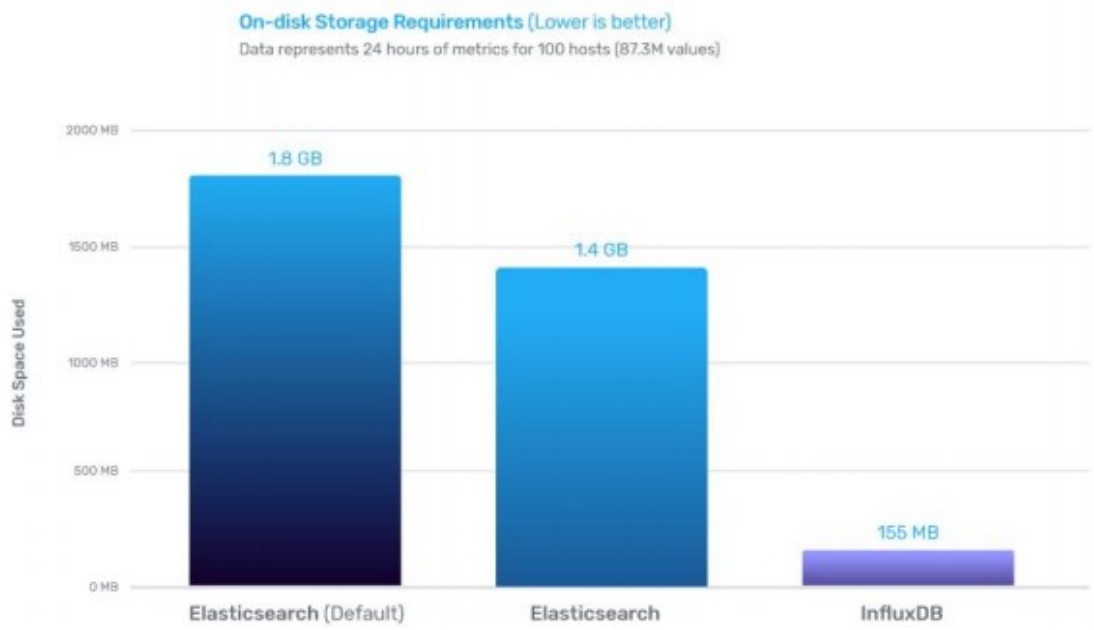

Figura 5 – confronto richiesta storage

Oltretutto dobbiamo anche considerare che Elasticsearch, per l'analisi delle time series, necessita di una serie di configurazioni non banali come la definizione degli indici e il dimensionamento della memoria che andrà allocata in modo dinamico, mentre InfluxDB è già pronto all'uso, senza bisogno di configurazioni iniziali. Un altro punto a favore di Influx per concludere è senza dubbio l'utilizzo di un linguaggio di query progettato e ottimizzato per le serie temporali e ciò si può vedere in questo esempio, dove si va a ricavare l'utilizzo di CPU da una time series:

Query con Elasticsearch:

```
{ "size" : 0,
"aggs": {
  "result": {
     "filter": {
        "range": {
            "timestamp": {
                                "gte": "2018-01-12T04:29:14-08:00",
                                "lt": "2018-01-13T04:29:14-08:00" }
              }
            },
```

```
"aggs": {
            "result2": {
                "date_histogram": {
                                    "field": "timestamp",
                                    "interval": "1h",
                                    "format": "yyyy-MM-dd-HH" },

"aggs": {
"avg_of_field": {
"avg": {
"field": "usage_user" } } } } } } }
```

Query con InfluxDB:

```
SELECT mean(usage_user) from cpu where time >= '2018-01-12T04:29:14-08:00'
and time < '2018-01-13T04:29:14-08:00' group by time(1h)
```

Ecco quindi spiegato il perché la nostra scelta per l'analisi delle metriche è ricaduta su InfluxDB e non su Elasticsearch.

Il discorso ovviamente cambia per l'analisi dei Log come già spiegato nel capitolo precedente.

4.6 La Dashboard

Per dashboard (o cruscotto) si intende uno strumento in grado di racchiudere al suo interno una serie di informazioni relative ad un ambiente o un ambito applicativo.

Abbiamo visto nel capitolo dedicato alle specifiche di sistema quanto sia importante adottare una dashboard in grado di:

- Rappresentare tutto quello che il nostro sistema di monitoring colleziona in modo interattivo, che si tratti di metriche o di dati estratti dall'analisi dei log
- Comunicare attraverso alert opportuni eventuali stati di malfunzionamento o superamento di soglie

Analizzando inoltre quanto detto nel capitolo precedente, vediamo come sia InfluxDB che Elasticsearch dispongono, all'interno del loro ambiente, di strumenti in grado di rappresentare graficamente i dati raccolti: abbiamo infatti rispettivamente Kibana [6] per Elasticsearch e Chronograf [4] per InfluxDB.

Tuttavia, nel nostro caso è di cruciale importanza avere a disposizione uno strumento unico in grado di raccogliere e rappresentare attraverso diverse tipologie di grafici o tabelle i dati collezionati.

Per assolvere a questo compito quindi si è scelto di adottare Grafana, una soluzione open-source in grado di visualizzare graficamente dati da svariate sorgenti.

4.6.1 Grafana

Grafana [8] è un software web open-source e multipiattaforma che, come detto precedentemente, è in grado di analizzare e visualizzare in modo interattivo grafici e avvisi da diversi data source.

Il progetto Grafana è stato rilasciato per la prima volta nel 2014 da Torkel Ödegaard, nato per supportare database di tipo time series come InfluxDB, OpenTSDB e Prometheus, con il tempo si è evoluta fino a supportare fonti relazionali come MySQL, PostgreSQL o Microsoft SQL Server.

La versione open-source di Grafana mette a disposizione degli utenti, grazie ad una community di sviluppatori, una grande varietà di plug-in in grado di estendere e personalizzare le funzioni di base delle dashboard.

I plugin che Grafana ci mette a disposizione sono di tre tipi:

- Plugin relativi ai pannelli: sono plugin in gradi di aggiungere nuove modalità di visualizzazione dei data in Grafana, sia per dati di tipo time series che non.
- Plugin di tipo Datasource, consentono a Grafana di connettersi a svariati tipi di database e fonti esterne.
- App, in questo caso i plugin contengono una sorta di bundle che comprende pannelli, Datasource e dashboard già pronte all'uso.

Quando in Grafana si parla di "pannello" si intende l'elemento base che compone la visualizzazione della dashboard.

Ogni pannello in Grafana è infatti dotato di un editor di query specifico per la sorgente dati che abbiamo precedentemente selezionato: ciò consentirà di estrarre in ogni pannello la visualizzazione desiderata e di rappresentarla con il grafico più idoneo alla nostra esigenza.

Esistono infatti svariate opzioni di stile e di formattazione per ciascun pannello, è possibile inoltre organizzare e ridimensionare i pannelli nella dashboard con delle

semplici operazioni di trascinamento, raggruppare tipologie di pannelli, e molte altre tipologie di personalizzazione che vedremo più avanti nella descrizione operativa del nostro progetto.

Proseguiamo la descrizione di Grafana parlando del sistema di alerting di Grafana, in ogni pannello infatti sarà anche possibile definire gli avvisi che consentiranno di identificare le eventuali problematiche rilevate.

Possiamo suddividere gli alert in due fasi:

- Le regole di avviso: nella configurazione del pannello vengono definite da una o più condizioni le regole che Grafana va a valutare per attivare l'eventuale avviso da inoltrare.
- Il canale di notifica: qui definiamo sostanzialmente attraverso quale canale vogliamo che Grafana invii gli avvisi.

Grafana ci permette inoltre di inserire negli alert qualsiasi tipo di informazione, da informazioni sulla risoluzione del problema rilevato a link esterni e così via.

Concludiamo riassumendo quindi i motivi principali che hanno portato alla scelta di Grafana come dashboard centrale del nostro sistema di monitoraggio:

- Supporta diversi data source
- Semplice da configurare ed utilizzare
- Open-source, stabile e multipiattaforma
- Semplicità di gestione dei pannelli per la visualizzazione dei grafici
- Grafici e visualizzazioni gradevoli

Capitolo 5 – Applicazione sviluppata

In questo capitolo descriveremo la soluzione software realizzata, partendo dalla configurazione dell'ambiente di sviluppo e di raccolta dati, procedendo poi con l'analisi e l'illustrazione di tutte le fasi che sono state necessarie all'implementazione del nostro software di monitoraggio e concludendo infine con delle riflessioni sui risultati ottenuti e su possibili futuri sviluppi.

5.1 Installazione dell'OS e dell'ambiente di Sviluppo

Descriveremo ora gli step e le procedure eseguite per l'installazione e la configurazione del nostro Server:

- Creazione e dimensionamento del Server
- Installazione e configurazione del Sistema Operativo
- Installazione e configurazione dei Data Source
- Configurazione dashboard Grafana e collegamento ai data source
- Configurazione e gestione delle policy di accesso degli utenti alla dashboard

5.1.1 creazione VM

Per ospitare il nostro server si è optato per una soluzione virtualizzata, nello specifico per virtualizzazione intendiamo quella tecnica che consente di utilizzare un software per simulare delle funzionalità hardware.

Questo permette ad organizzazioni IT di poter eseguire più di un sistema virtuale su di un unico server fisico.

Pensiamo quindi a una macchina virtuale o VM, come ad un container software altamente isolato che includerà nel nostro caso il sistema applicativo e tutti gli strumenti per la raccolta e l'elaborazione di metriche e dati di cui abbiamo discusso in precedenza.

Chiameremo "Hypervisor" il layer software che separa le macchine virtuali dall'host fisico e si occupa di allocare dinamicamente le risorse (CPU, RAM, Spazio Disco, etc..) a ciascuna macchina virtuale in base alle esigenze.

Elenchiamo quindi in breve una serie di vantaggi che l'utilizzo di VM offre:

- Esecuzione di più sistemi operativi su un'unica macchina fisica
- Partizionamento delle risorse tra le VM
- Isolamento di problemi relativi alla sicurezza o a guasti a livello hardware
- Ottimizzazione delle prestazioni grazie al controllo avanzato delle risorse
- Possibilità di salvare su un file l'intero stato della VM
- Facilità di esportazione e copia delle VM
- Grazie all'indipendenza dall'hardware, possibilità di spostare le VM su qualsiasi server fisico

Fatta questa premessa procediamo ora alla descrizione delle varie fasi che hanno portato alla configurazione della VM contenente il nostro server di monitoraggio.

Come primo passo procediamo effettuando il login al nostro hypervisor (nel nostro caso si è optato per una soluzione VMware).

Una volta caricata l'interfaccia di amministrazione ci viene data subito la possibilità di aggiungere una nuova macchina virtuale: selezionando la voce "Create/Register VM" apparirà la schermata sottostante:

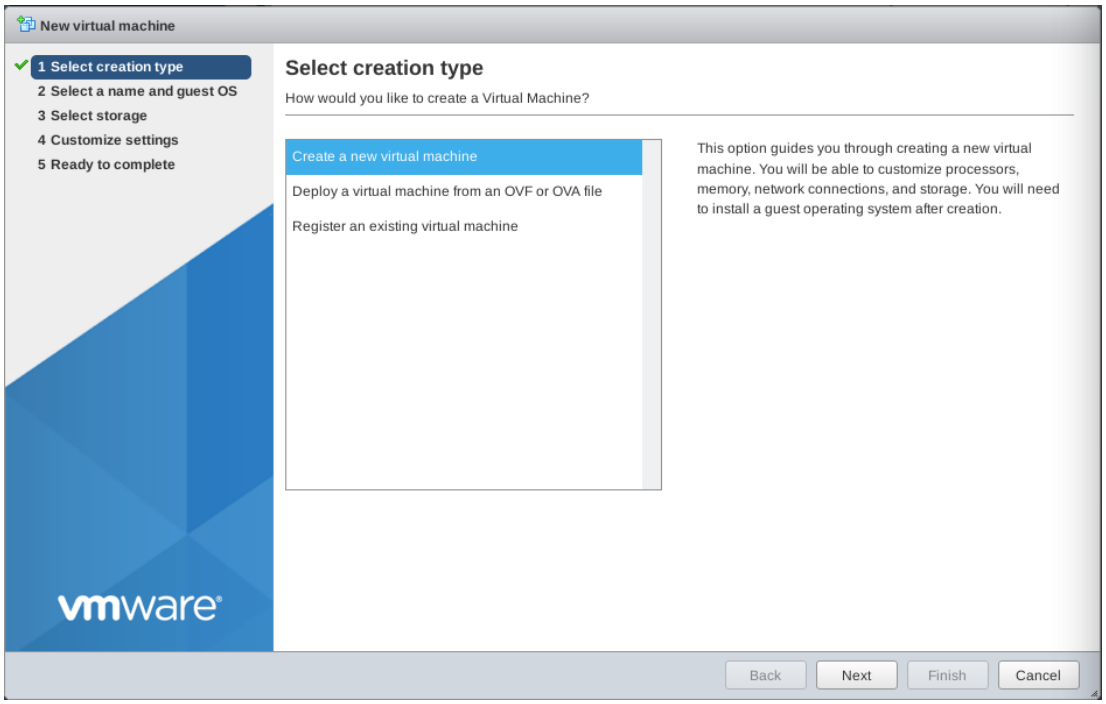

Figura 6 – Nuova VM

Qui sarà possibile scegliere se importare una VM preesistente o procedere alla creazione di una nuova, nel nostro caso lasciamo selezionata la prima opzione e procediamo con le operazioni seguenti.

Nello step successivo inseriamo il nome che avrà il nostro server e il tipo di sistema operativo che la nostra VM dovrà ospitare, nel nostro caso selezioneremo quindi "Linux" alla voce "Guest OS family" e come versione selezioneremo "CentOS 7 (64-bit)"

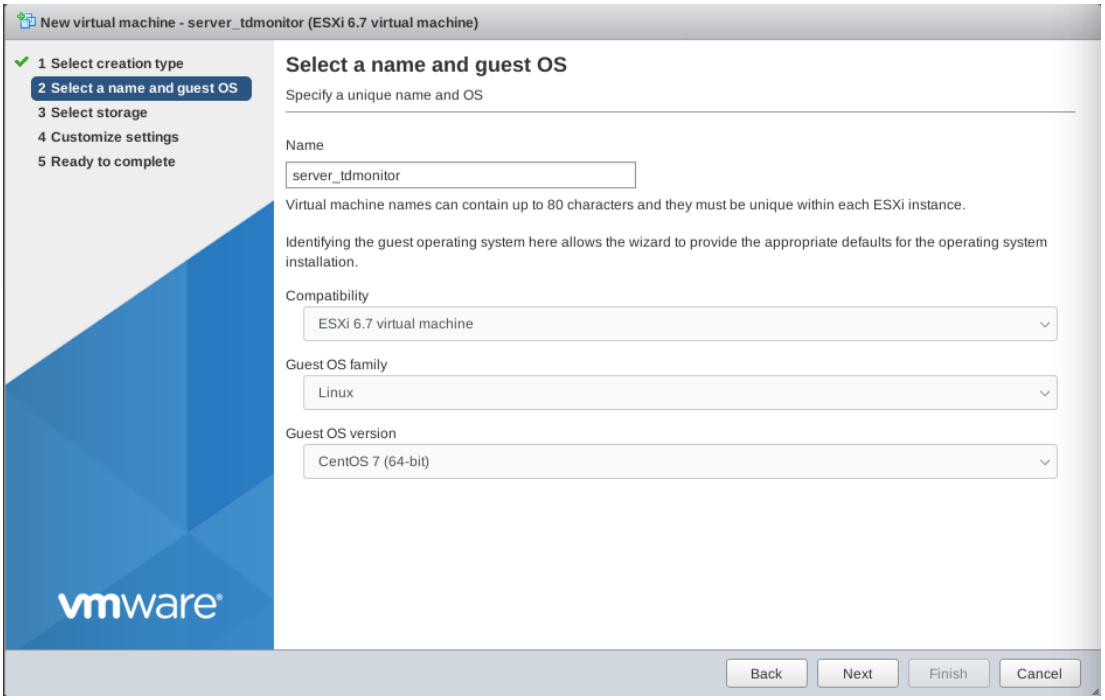

Figura 7 – Selezione nome host

Procedendo con i passi successivi andiamo a selezionare lo storage fisico che ospiterà la nostra macchina virtuale e le varie risorse hardware.

Nel nostro caso, come già accennato nelle specifiche partiremo dalla seguente configurazione di base:

- 1 CPU
- RAM 4Gb
- Spazio disco 40Gb (LVM e ridimensionabile)
- Una sola scheda di rete

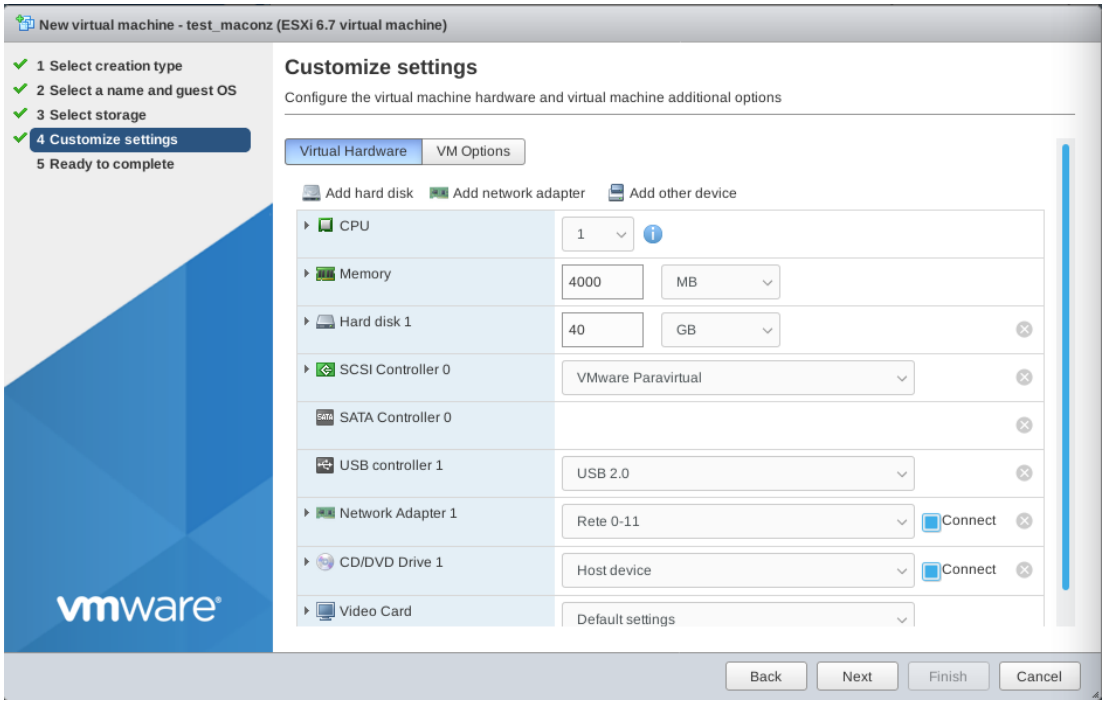

Figura 8 – Settaggio Risorse

Una volta completato il dimensionamento della macchina saremo pronti a caricare la ISO del sistema operativo prescelto, nel nostro caso CentOS 7 (64-bit), impostare la periferica virtuale contenente la ISO come prima periferica su cui avviare il boot di sistema e avviare la nostra macchina virtuale.

5.1.2 Installazione CentOS 7

Come già anticipato nel quarto capitolo la scelta sul sistema operativo che ospiterà il nostro sistema di monitoraggio è ricaduta sulla versione più stabile di CentOS [9], la versione 7, rilasciata per la prima volta nel 2014 e ad oggi ancora supportata.

Vediamo ora le fasi cruciali dell'installazione e della configurazione del nostro OS. Nella prima schermata che ci si presenta dopo il boot ci verrà chiesto di scegliere se Installare CentOS o effettuare un test preliminare: procediamo con la prima opzione.

Una volta che il sistema avrà completato il caricamento del programma di installazione ci verrà chiesto di scegliere la lingua da utilizzare durante il processo di installazione, nel nostro caso optiamo per "English".

Nella schermata successiva ci viene proposto il sommario di installazione:

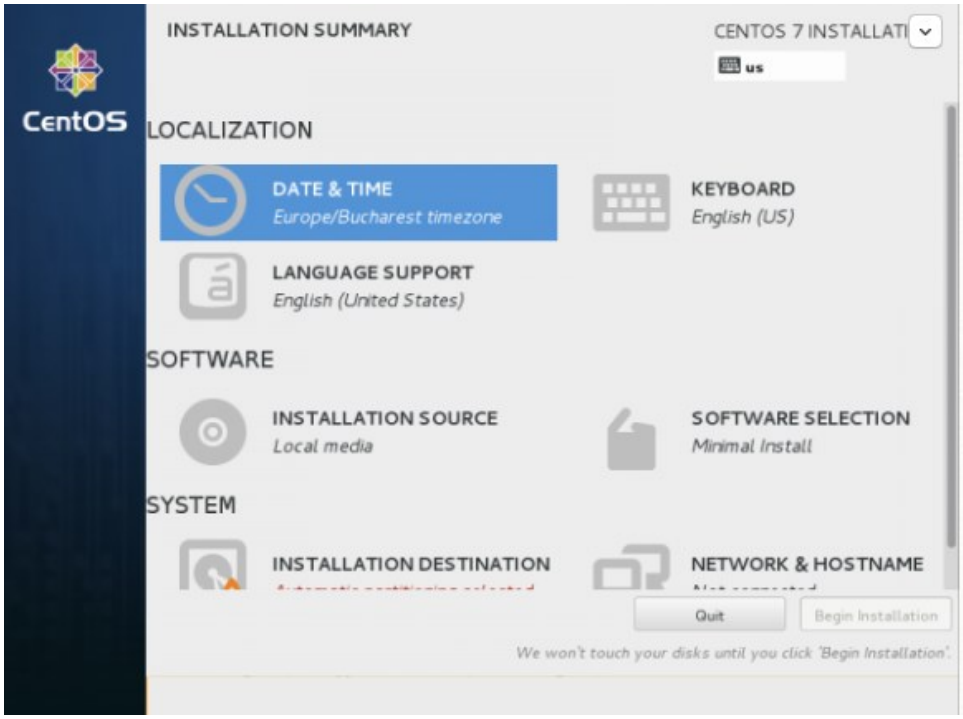

Figura 9 – Sommario di Installazione CentOS

In figura 9 vediamo che ci vengono proposte una serie di opzioni di configurazione; per procedere seguiremo in ordine i seguenti passaggi:

- Scelta del layout della tastiera, selezionando nel nostro caso un layout per tastiera italiana standard
- Impostazione della data e dell'ora di sistema selezionando "Date & Time"
- Impostare sotto la voce "Installation Source" le sorgenti software preselezionate
- Nella finestra che si aprirà selezionando "Software Selection" possiamo scegliere l'opzione "minimal install" in quanto nel nostro caso servirà il sistema base e procederemo succesivamente all'installazione dei componenti necessari
- Selezionando "Installation Destination" avremo accesso alla schermata che consentirà di selezionare lo storage di sistema e di partizionarlo a nostro piacimento.

  Selezionando il partizionamento manuale quindi andremo a definire le tre partizioni come segue:

  - Partizione di Boot identificata con "/boot" non LVM
  - Partizione di root identificata con "/root" - LVM
  - Partizione di Swap – LVM

  LVM (Logical Volume Manager) è uno schema di partizionamento che ci consentirà di allocare dinamicamente lo spazio disponibile; ciò ci permetterà ad esempio, nel caso in cui in futuro sia necessario più spazio disco, di effettuare un ridimensionamento dello storage senza dover inizializzare le partizioni.

  Il risultato ottenuto sarà simile alla figura sottostante:

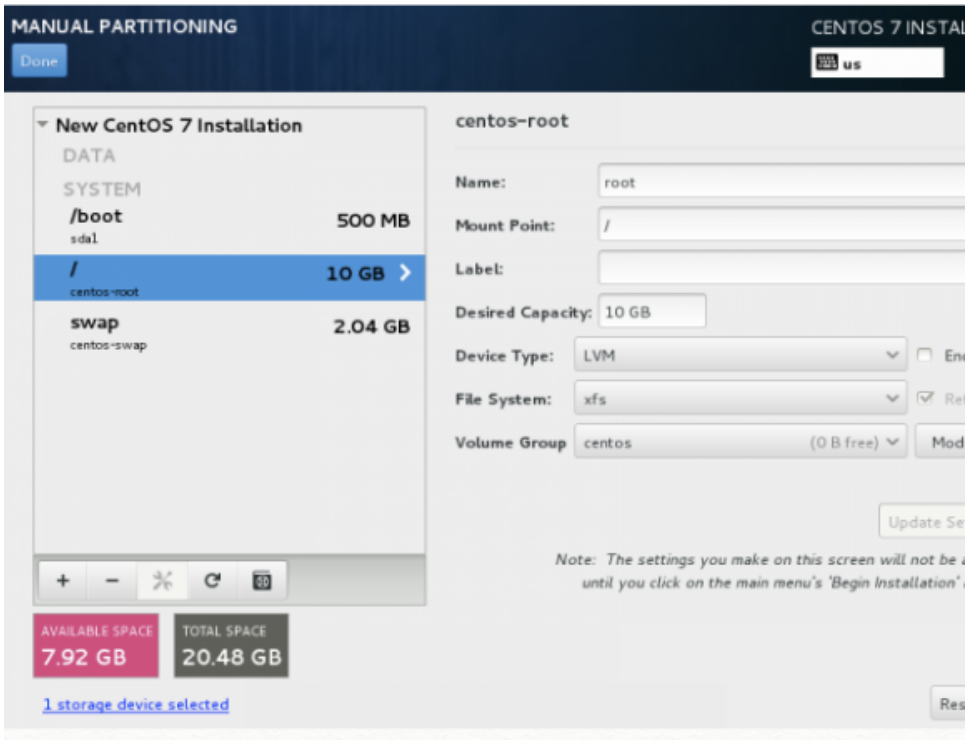

Figura 10 – Partizionamento disco

- Selezionando infine la voce “Network & Hostname” procediamo alla configurazione della parte Networking e alla scelta del nome host del nostro sistema.

  Come prima cosa sarà necessario abilitare l’interfaccia di rete desiderata (nel nostro caso ne avremo solo una) e scegliere il nome host, nel nostro caso inseriremo come nome host “tdmonitor.tecno.intra”. Successivamente, selezionando il tasto “Configure” ci sarà possibile, sotto la scheda “IPv4 Settings”, configurare in modalità manuale i parametri di rete necessari affinché il nostro server sia raggiungibile in rete:

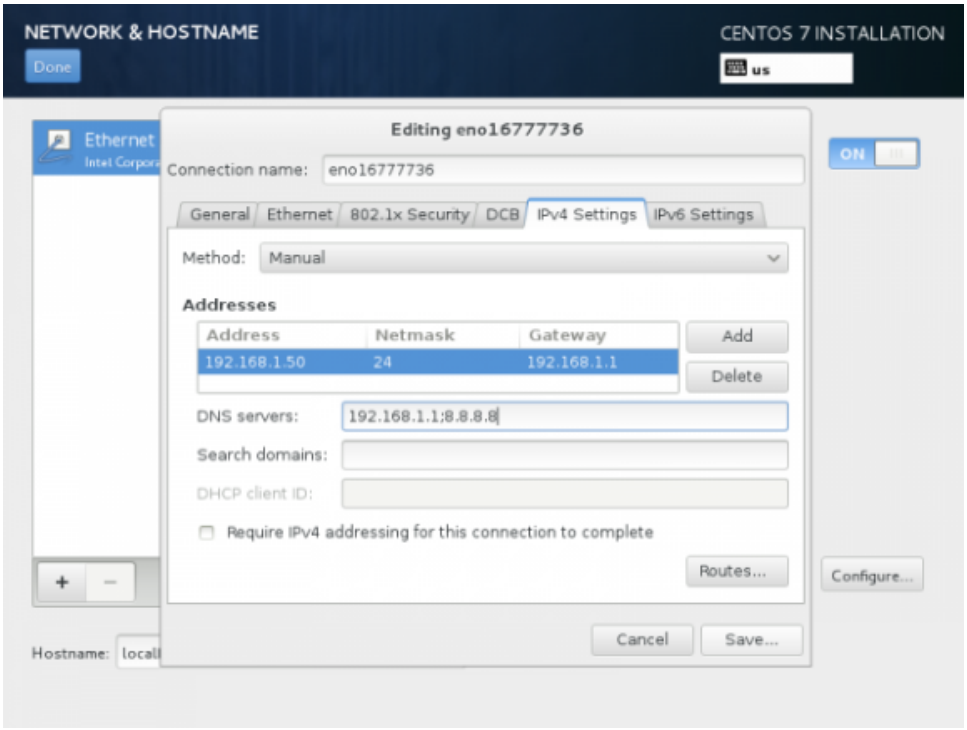

Figura 11 – configurazione parametri di rete

Optiamo per la configurazione manuale in modo tale da poter fornire al nostro server un indirizzo IPv4 statico, visto che attraverso quello dovrà essere raggiungibile da tutto il network.

- Una volta completati questi passaggi clicchiamo sul pulsante “Begin Installation” e ci verrà proposta la schermata mostrata in figura 12:

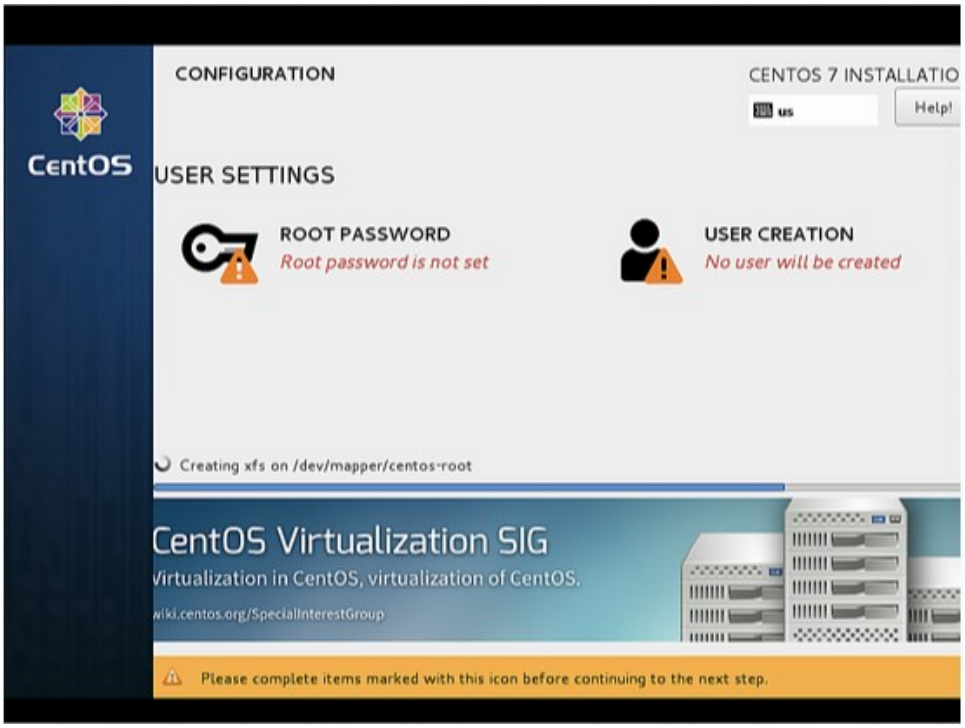

Figura 12 – configurazione password di root

Qui andremo a inserire la password di root del sistema (con root nei sistemi Linux si identifica l'utente Amministratore di sistema) e ignoriamo, per ora l'opzione per la creazione di un utente diverso da root. Una volta completata l'installazione ci verrà chiesto di riavviare il sistema.

Terminata l'installazione il nostro sistema sarà pronto, effettuiamo quindi il login come utente root e come prima cosa procediamo ad eseguire l'aggiornamento del sistema per allineare tutti i suoi componenti alle versioni più recenti, per fare ciò lanciamo il comando:

```
yum update
```

Poiché la versione "minimal" di CentOS 7 non dispone del comando "ifconfig", che potrà esserci utile in seguito per la configurazione e le impostazioni di rete, procediamo alla sua installazione lanciando il comando:

```
yum install net-tools
```

5.2 Installazione e configurazione InfluxDB

In questa sezione illustriamo la prima parte di quella che abbiamo definito come installazione dell'ambiente di sviluppo, andremo ora a descrivere le fasi cruciali dell'installazione e della configurazione di InfluxDB [4].

Come prima cosa sarà necessario aggiungere il repository necessario all'installazione di InfluxDB al nostro sistema; questo ci consentirà anche in futuro di aggiornare in modo semplice InfluxDB.

Per farlo lanciamo i comandi:

```
cat <<EOF | sudo tee /etc/yum.repos.d/influxdb.repo
[influxdb]
name = InfluxDB Repository - RHEL \$releasever
baseurl = https://repos.influxdata.com/rhel/\$releasever/\$basearch/stable
enabled = 1
gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdb.key
EOF
```

Grazie a questa aggiunta e una volta aggiornata la cache di sistema potremmo procedere all'installazione vera propria di InfluxDB lanciando il comando:

```
sudo yum -y install influxdb
```

Una volta terminata l'installazione possiamo procedere ad avviare il servizio e ad abilitarlo in modo che all'avvio del sistema venga attivato in automatico; per fare ciò lanciamo il comando:

```
sudo systemctl start influxdb && sudo systemctl enable influxdb
```

Come abbiamo visto tutti i comandi di setup lanciati sinora sono eseguiti come utente root, questo perché InfluxDB richiede come requisito di essere installato con privilegi di amministratore.

Un altro requisito prevede che il firewall di sistema di CentOS 7 renda disponibile l'accesso alle porte TCP 8086 e 8088, la prima deve essere disponibile per la comunicazione client-server mentre la seconda per rendere disponibile il servizio RPC che consentirà di eseguire eventuali operazioni di Backup e ripristino.

Per aprire le porte possiamo lanciare i seguenti comandi:

```
sudo firewall-cmd --add-port=8086/tcp –permanent

sudo firewall-cmd --add-port=8088/tcp --permanent

sudo firewall-cmd --reload
```

Ricordiamo inoltre che InfluxDB mette a disposizione diversi plug-in che potrebbero richiedere l'apertura di altre porte; la mappatura delle porte potrà comunque sempre essere cambiata nel file di configurazione che si trova in "/etc/influxdb/influxdb.conf", come vedremo più avanti.

Un'altra impostazione che abilitiamo nel file di configurazione di InfluxDB è l'abilitazione alla porta http, disabilitata di default.

Per fare questo eseguiamo il comando:

```
sudo vim /etc/influxdb/influxdb.conf
```

E andiamo ad abilitare l'autenticazione http sotto la sezione opportuna come segue:

```
[http]
auth-enabled = true
```

Fatto ciò, andiamo a creare un utente con password i di autenticazione:

```
curl -XPOST "http://localhost:8086/query" --data-urlencode \

"q=CREATE USER username WITH PASSWORD 'password' WITH ALL PRIVILEGES"
```

Dove al posto di “username” e “password” inseriremo rispettivamente il nome utente e la password scelte per l’utente che dovrà eseguire qualsiasi comando influxdb da terminale.

Ogni volta che andrà lanciato un comando infatti sarà necessario specificare il nome utente utilizzando *-username* e la password utilizzando *-password* come segue:

```
influx -username "nome utente" -password "password"
```

5.3 Installazione e configurazione base di Grafana

Procediamo ora all’installazione e configurazione di Grafana, nello specifico installeremo la versione 7.1.1; come già descritto precedentemente Grafana ci consentirà di organizzare e visualizzare i dati collezionati.

In CentOS è possibile installare Grafana in modo manuale, scaricando un file “.tar.gz” e installandolo localmente, oppure , come già visto per InfluxDB, aggiungere il repository opportuno così da rendere più semplice e agile la procedura di aggiornamento.

Procediamo quindi come per InfluxDB e aggiungiamo il repository di Grafana:

```
cat <<EOF | sudo tee /etc/yum.repos.d/grafana.repo
[grafana]
name=grafana
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
EOF
```

Una volta aggiunto il repository, procediamo all'installazione sempre con privilegi di amministratore lanciando il comando:

```
sudo yum -y install grafana
```

Il sistema chiederà di accettare la chiave gpg; possiamo accettare e proseguire.

Fatto ciò, avviamo grafana e abilitiamolo all'avvio lanciando:

```
sudo systemctl start grafana-server

sudo systemctl enable grafana-server
```

Durante il processo di installazione viene creato automaticamente l'utente "grafana".

Il processo "grafana-server" avviato con i comandi descritti precedentemente viene lanciato di default proprio con questo utente.

Per raggiungere l'interfaccia di Grafana la porta predefinita è la 3000, come già visto per InfluxDB sarà necessario rendere raggiungibile Grafana aprendo queste porte sul firewall di sistema.

Lanciamo quindi il comando:

```
sudo firewall-cmd --add-port = 3000 / tcp --permanent
sudo firewall-cmd --reload
```

Una volta che il processo "grafana-server" è avviato e raggiungibile, basterà digitare da browser l'indirizzo http://indirzzo_ip_server:3000 ed effettuare login con la password e lo username di default che per il primo accesso sono "admin" e "admin", ci troveremo davanti la schermata mostrata in figura 13:

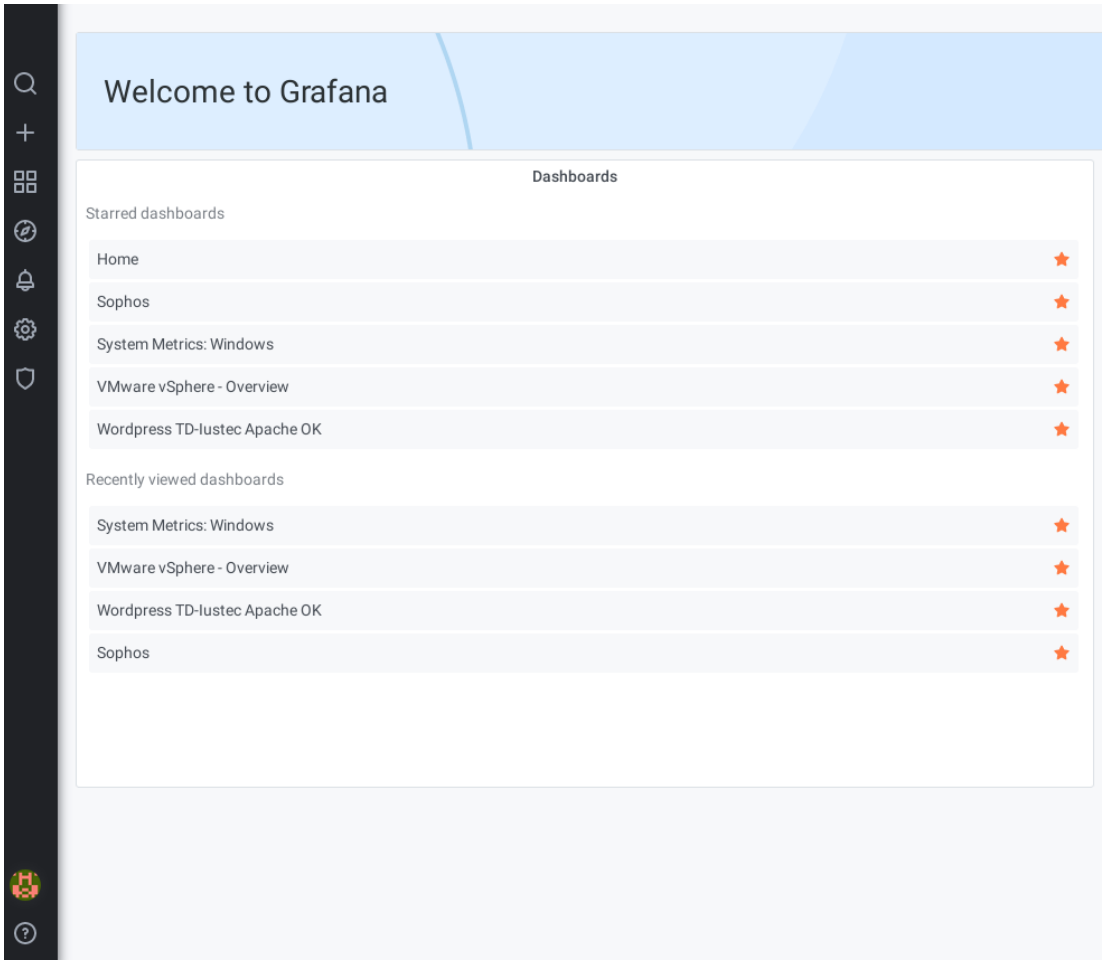

Figura 13 – Home Grafana

Come prima cosa, una volta effettuato il login, sarà necessario modificare la password di default dell'utente admin; per farlo basta cliccare sull'immagine del profilo "admin" situata in basso a sinistra e selezionare la voce "change password".

Il file di configurazione predefinito di Grafana si trova sotto la directory "/etc/grafana/grafana.in", mentre possiamo trovare tutte le informazioni sul suo funzionamento nei log allocati in "/var/log/grana".

Nei prossimi capitoli torneremo sulla configurazione di Grafana per vedere come aggiungere i parametri di connessione ai diversi datasource.

5.4 Installazione e configurazione Elasticsearch

Installiamo ora Elasticsearch [6], il componente del nostro sistema di monitoraggio che dovrà occuparsi di cercare e analizzare informazioni all'interno dei log collezionati.

Per quanto riguarda questo componente i requisiti minimi di utilizzo potrebbero essere più alti (solitamente RAM e spazio disco più elevati), nel caso descritto in questo elaborato tuttavia saranno più che sufficienti le risorse della macchina descritta in precedenza.

Anche in questo caso è possibile scaricare il pacchetto RPM da installare su CentOS dal sito web ufficiale di Elasticsearch ma, come già visto per i precedenti componenti, percorriamo la via dell'installazione da repository ufficiale con il comando:

```
cat <<EOF | sudo tee /etc/yum.repos.d/elasticsearch.repo
[elasticsearch]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=0
autorefresh=1
type=rpm-md
EOF
```

Fatto ciò, procediamo all'installazione del pacchetto sempre come utente root lanciando:

```
sudo yum install - enablerepo = elasticsearch elasticsearch
```

Poiché Elasticsearch è costruito sulla base di uno dei più popolari sistemi di indicizzazione, Apache Lucene, nelle precedenti versioni era necessario installare come prerequisito la versione consigliata sul sito delle OpenJDK, tuttavia nel nostro

caso adotteremo l'ultima versione disponibile di Elasticsearch, che include già una versione in bundle di OpenJDK nel pacchetto di installazione che abbiamo utilizzato.

Come abbiamo già visto nel quarto capitolo, Il processo di indicizzazione di Elasticsearch si basa sugli indici che sono equivalenti, a livello logico, ai database nell'ambito relazionale.

Per impostazione predefinita in fase di installazione Elasticsearch viene configurato per consentire automaticamente la creazione degli indici e di solito non è necessario effettuare altre operazioni.

Avviamo Elasticsearch con il comando:

```
sudo systemctl start elasticsearch

sudo systemctl enable elastisearch
```

Per quanto riguarda la configurazione, Elasticsearch mette a disposizione tre file di configurazione nella cartella predefinita di installazione collocata in "/etc/elastcsearch".

I tre file sono:

- Elasticsearch.yml: file dedicato alla configurazione di Elasticsearch
- Jvm.options: file dedicato alla configurazione e alle impostazioni JVM (Java Virtual Machine) di Elasticsearch
- Log4j2.properties: file dedicato alla configurazione dei log generati da Elasticsearch

5.5 installazione e configurazione Telegraf per la raccolta delle metriche

In questo capitolo inizieremo ad esaminare l'installazione e la configurazione di Telegraf [10].

Come già visto nel quarto capitolo Telegraf sarà l'agent principale del nostro sistema di monitoraggio che si occuperà della raccolta delle metriche e del loro inoltro ad InfluxDB.

Poiché Telegraf è un agent multipiattaforma, in grado cioè di essere compatibile e installabile su sistemi operativi diversi, analizzeremo nel dettaglio la sua configurazione e il suo funzionamento sia in ambiente Linux che in ambiente Windows.

Iniziamo ora analizzando la procedura di installazione in ambiente Linux.

5.5.1 Installazione e configurazione Telegraf in ambiente Linux

Come primo passo andiamo ad aggiungere alla configurazione il repository con il solito comando:

```
cat <<EOF | sudo tee /etc/yum.repos.d/influxdb.repo
[influxdb]
name = InfluxDB Repository - RHEL \$releasever
baseurl = https://repos.influxdata.com/rhel/\$releasever/\$basearch/stable
enabled = 1
gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdb.key
EOF
```

Una volta aggiunto il repository alla configurazione del gestore dei pacchetti di CentOS (yum), procediamo all'installazione lanciando:

```
sudo yum install telegraf
```

E lanciamo l'agent con il comando:

```
sudo systemctl start telegraf
```

Terminata la fase di installazione procediamo ora ad analizzare la configurazione di Telegraf in ambiente Linux.

Il file di configurazione di Telegraf è denominato "telegraf.conf" ed è raggiungibile al percorso "/etc/telegraf/telefraf.conf".

È anche possibile se necessario definire la directory"/etc/telegraf/telegraf.d" che servirà, a seconda dei casi, a contenere altri file di configurazione nel caso in cui non si voglia scrivere l'intera configurazione di Telegraf nel singolo file "telegraf.conf".

Come già detto in precedenza uno dei punti di forza di Telegraf è proprio quello di metterci a disposizione un gran numero di plug-in di input per consentirci di raccogliere svariate tipologie di dati. Dovremo tenere conto allora del fatto che questi ultimi possano richiedere accesso a porte personalizzate.

Vedremo più avanti come configurare la mappatura delle porte nel file "telegraf.conf".

Per creare il file telegraf.conf contenente già una collezione di plug-in predefiniti va lanciato il comando:

```
telegraf config > telegraf.conf
```

Il file di configurazione predefinito ha un aspetto simile:

```
[global_tags]
[agent]
  interval = "10s"
  round_interval = true
  metric_batch_size = 1000
  metric_buffer_limit = 10000
```

```
  collection_jitter = "0s"
  flush_interval = "10s"
  flush_jitter = "0s"
  precision = ""
  hostname = ""
  omit_hostname = false
[[outputs.influxdb]]
[[inputs.cpu]]
  percpu = true
  totalcpu = true
  collect_cpu_time = false
  report_active = false
[[inputs.disk]]
  ignore_fs = ["tmpfs", "devtmpfs", "devfs", "iso9660", "overlay", "aufs",
"squashfs"]
[[inputs.diskio]]
[[inputs.kernel]]
[[inputs.mem]]
[[inputs.processes]]
[[inputs.swap]]
[[inputs.system]]
```

In alternativa possiamo anche creare un file di configurazione contenente parametri di input e output specifici utilizzando i flag “--input-filter" e “--output-filter" come segue:

```
telegraf --input-filter cpu:mem:net:swap --output-filter influxdb:kafka config
```

Come possiamo già intuire da una prima analisi dell’esempio sopra riportato, i plug-in all’interno del file di configurazione di Telegraf possono essere di diverse tipologie:

- Plug-in di Output che hanno il compito di scrivere le metriche raccolte su una destinazione (InfluxDB nel nostro caso)
- Plug-in di Input che raccolgono le metriche dal sistema, da servizi o da applicazioni
- Plug-in che hanno il compito di aggregare le metriche raccolte

- Processor Plug-in, sono plug-in che possono trasformare, modificare o filtrare le metriche raccolte

Analizziamo più da vicino un plugin di Output.

Nel nostro caso il flusso delle metriche collezionate sarà inviato all'istanza di InfluxDB precedentemente installata sul server di monitoraggio.

Per fare ciò basterà individuare la sezione con il plugin di output per InfluxDB, identificata con "[[outputs.influxdb]]", togliere il commento dalla riga contenente l'indirizzo del server del plugin di output e inserire l'indirizzo del nostro server.

In alternativa nulla ci vieta di inserire manualmente la definizione del plugin di output.

il risultato sarà simile a questo:

```
[[outputs.influxdb]]

  ## The full HTTP or UDP URL for your InfluxDB instance.
  ##
  ## Multiple URLs can be specified for a single cluster, only ONE of the
  ## urls will be written to each interval.
  # urls = ["unix:///var/run/influxdb.sock"]
  # urls = ["udp://127.0.0.1:8089"]
    urls = ["http://indirizzo_ip_server:8086"]
```

Per quanto riguarda invece i plug-in di input, come abbiamo già accennato in precedenza, possono essere sia definiti durante la creazione del file di configurazione in modo predefinito che scaricati manualmente da sorgenti come github e definiti o caricati all'interno della configurazione di Telegraf.

In questo caso sono identificati all'interno del file di configurazione di telegraf con le sezioni "[[inputs.<risorsa_da_monitorare>]]", in ogni sezione saranno contenuti i parametri di cui raccogliere le metriche.

Vediamo nel codice qui sotto l'esempio di un plug-in di tipo Input per monitorare lo stato della risorse CPU:

```
[[inputs.cpu]]

## se true raggruppa le statistiche per ogni CPU
percpu = true

## se true riporta tutte le statistiche relative  alla CPU
totalcpu = true

## se true raccogliele metriche relative ai tempi di utilizzo della CPU
collect_cpu_time = false

## se true calcola e riporta lostato delle CPU attive
report_active = false
```

Come si può osservare nell'esempio, abbiamo commentato ogni parametro di configurazione per spiegarne il funzionamento.

Per avere il dettaglio relativo al funzionamento di ogni parametro di configurazione di ogni plug-in di input è sempre possibile fare riferimento al dettaglio del repository Github di Telegraf raggiungibile al link https://github.com/influxdata /telegraf/tree/master/plugins/inputs.

Ci sono tuttavia dei parametri che sono disponibili per tutti i plug-in di Input e sono i seguenti:

- interval: si tratta della frequenza con cui vogliamo che la metrica venga collezionata.

  Di norma i plugin fanno tutti riferimento ad un intervallo definito globalmente ma nel caso in cui un particolare parametro di input dovesse essere raccolto meno o più spesso, potremo definirlo variando questo parametro
- name_override: serve a sovrascrivere il nome di default della misurazione

- name_prefix: serve a specificare un prefisso da inserire nel nome della misurazione
- name_suffix: serve a specificare un suffisso da inserire nel nome della misurazione
- tags: consente di inserire dei tag da applicare alle misurazioni di uno specifico input

anche nei plug-in di aggregazione troviamo dei parametri comuni :

- period: tutte le metriche inviate con timestamp al di fuori di questo periodo verrano ignorate
- delay: indica il ritardo prima che ogni aggregatore venga scaricato.
  Ciò serve a controllare quanto tempo i plug-in di aggregazione devono attendere prima prima di ricevere le metriche dai plug-in di input.
- drop_original: se impostata su true, la metrica originale viene eliminata dall'aggregatore e non verrà inoltrata in output
- name_override: sovrascrive il nome di base della misurazione, andando a sovrascrivere il nome dell'ingresso che di default dà il nome alla misurazione
- name_prefix: serve a specificare un prefisso da inserire nel nome della misurazione
- name_suffix: serve a specificare un suffisso da inserire nel nome della misurazione
- tags: campo per inserire eventuali tag da assegnare alla misura

Per quanto riguarda invece i Processor plug-in il parametro che sarà sempre disponibile è "order" che definisce l'ordine in cui vengono eseguiti i processi; nel caso non venga specificato l'ordine di processazione sarà casuale.

5.5.2 Installazione e configurazione Telegraf in ambiente Windows

In questo capitolo vedremo come installare e configurare Telegraf [5] per raccogliere le metriche necessarie a monitorare risorse basate su sistemi operativi Microsoft Windows.

Come primo passo sarà necessario effettuare il download della più recente versione stabile di Telegraf dal sito ufficiale https://portal.influxdata.com/downloads/.

Una volta completato il download, procediamo all'estrazione del contenuto del file compresso ottenuto nella directory "C:\Program Files\Telegraf".

La directory, come per linux conterrà il file di configurazione e il file eseguibile (.exe).

Per garantire che il processo telegraf abbia accesso a tutte le risorse del sistema, sarà necessario installarlo come servizio di sistema di Windows utilizzando privilegi amministrativi; in questo modo sarà anche possibile gestirlo dal pannello di gestione dei servizi di Windows o da riga di comando.

Per fare ciò avviamo PowerShell di Windows utilizzando l'utente amministratore e lanciamo il comando:

```
> C:\"Program Files"\Telegraf\telegraf.exe --service install
```

A questo punto possiamo procedere a configurare telegraf per Windows nello stesso modo in cui abbiamo configurato la versione per Linux.

Il file ovviamente è molto simile anche se presenterà delle differenze sostanziali in termini di configurazione dei parametri di input.

Per quanto riguarda invece i plugin di output, anche qui andremo ad inserire, come analogamente fatto in ambiente Linux, l'indirizzo del server dove è installata l'istanza di InfluxDB come segue:

```
[[outputs.influxdb]]
  ## The full HTTP or UDP URL for your InfluxDB instance.
  ##
  ##
  # urls = ["unix:///var/run/influxdb.sock"]
  # urls = ["udp://127.0.0.1:8089"]
    urls = ["http://indirizzo_server:8086"]
```

Una volta completata la configurazione (più avanti vedremo esempi pratici), possiamo salvarla e testare il corretto funzionamento del nostro agent lanciando sempre da PowerShell i comandi:

```
>C:\"Program Files"\Telegraf\telegraf.exe --config

>C:\"Program Files"\Telegraf\telegraf.conf --test
```

Per iniziare infine a raccogliere le metriche dalla nostra macchina Windows lanciamo:

```
> net start telegraf
```

Concludiamo dicendo che avendo avviato Telegraf come servizio Windows, potrebbe essere opportuno proteggere il file di configurazione di Telegraf per impedire ad utenti non autorizzati di recuperare informazioni potenzialmente sensibili, per fare questo possiamo utilizzare il comando “icacls”, si tratta di uno strumento per la gestione delle access list (ACLs) per oggetti in Windows.

Lanciamo allora sempre da PowerShell il comando:

```
PS> icacls telegraf.conf /reset
```

Con questo comando vengono rimosse dal file di configurazione tutte le ACL; l’oggetto erediterà solo le autorizzazioni dell’oggetto padre, che nel nostro caso è la directory “C:\Program Files\telegraf”.

Con il comando successivo invece andiamo ad eliminare anche l'eridetarietà e qualsiasi altra ACL: a questo punto nessun utente potrà accedere al file.

Inseriamo allora il flag "/grant system:r" che consentirà all'account di sistema locale di leggere il file, con il comando:

```
PS> icacls outputs.conf /inheritance:r /grant system:r
```

5.6 Installazione e configurazione di Filebeat per l'invio dei log ad Elasticsearch in CentOS

Nel quarto capitolo abbiamo descritto il perché uno strumento come Elasticsearch sia la soluzione ottimale per raccogliere ed analizzare i log provenienti dalle risorse della nostra rete monitorata.

Tuttavia, per poter raccogliere ed inviare ad Elasticsearch i file di log non possiamo affidarci come per le metriche a Telegraf, ma è necessario adottare una soluzione performante e perfettamente integrata con Elasticsearch.

Avevamo già parlato, sempre nel precedente capitolo, dell'utilizzo dei "beats", una famiglia di agenti o "shipper" facente parte dello stack ELK [6] che hanno il compito di inviare dati.

Il "beat" di cui ora descriveremo l'installazione e la configurazione è Filebeat.

Filebeat ha il ruolo di inviare i file di log generati dalla macchina dove è installato ad Elasticsearch, è scritto in GO, ha un basso impatto sulle risorse, supporta connessioni criptate ed è in grado di gestire grandi quantità di dati in modo efficiente.

Per installare Filebeat in ambiente Linux CentOS procediamo in modo analogo a quanto visto in precedenza.

Visto che i repository sono gli stessi già utilizzati per Elasticsearch ci basterà eseguire l'installazione lanciando:

```
sudo yum install filebeat
```

Per configurare Filebeat possiamo far riferimento al file di configurazione "filebeat.yml" che troviamo sotto la directory "/etc/filebeat".

All'interno del file sarà possibile definire nella sezione input il percorso o i percorsi dei file di log che voglio raccogliere, ad esempio:

```
filebeat.inputs:

- type: log
  enabled: true
  paths:
    - /var/log/*.log
```

Per inoltrare i log ad Elasticsearch invece sarà necessario editare la sezione output relativa ad Elasticsearch del file di configurazione specificando l'indirizzo del server di destinazione e, se necessario, le credenziali di accesso come segue:

```
  output.elasticsearch:

  hosts: ["indirizzo_ip_server_elasticsearch:9200"]
  username: "user"
  password: "password"
```

Ovviamente all'interno del file di configurazione di Filebeat sarà possibile configurare diverse tipologie di input e di output a seconda delle nostre esigenze.

Anche Filebeat infatti, oltre alla configurazione di base per i log più comuni come ad esempio Apache o MySQL, dispone di diversi moduli aggiuntivi abilitabili per estendere le funzioni di Filebeat.

Una volta aggiunti i parametri di input e output necessari alla configurazione possiamo iniziare a collezionare i log avviando Filebeat con la seguente istruzione:

```
sudo service filebeat start
```

5.7 Installazione e configurazione di Winlogbeat per l'invio dei log ad Elasticsearch in CentOS

Analogamente a quanto detto in precedenza per Filebeat, anche in ambiente Windows abbiamo l'esigenza di inviare verso Elasticsearch i log estratti dal sistema.

Per assolvere a questo compito possiamo utilizzare un altro "beat" facente parte dello stack ELK, Winlogbeat.

Per installare Winlogbeat in ambiente Windows possiamo scaricarlo dal sito ufficiale https://www.elastic.co/downloads/beats/winlogbeat; una volta scaricato il file compresso procediamo ad estrarlo nella directory "C:\ProgramData\winlogbeat".

Per quanto riguarda la configurazione faremo riferimento al file "winlogbeat.yml" contenuto nella directory sopracitata, il cui contenuto avrà un aspetto simile:

```
winlogbeat.event_logs:
  - name: Application
  - name: Security
  - name: System

output.elasticsearch:
  hosts:
    - indirizzo_ip_server:9200

logging.to_files: true
logging.files:
  path: C:\ProgramData\winlogbeat\Logs
logging.level: info
```

Nella sezione "event_logs" andremo a specificare gli eventi che vogliamo monitorare, di default Winlogbeat è configurato per monitorare i log di registro di Application, Security e System.

La sezione "output.elasticsearch" conterrà, come già visto per Filebeat, l'indirizzo del server a cui saranno inviati i log.

Nell'ultima parte del file di configurazione invece possiamo specificare altre opzioni come ad esempio la cartella dove Winlogbeat organizzerà i file di log prima di inviarli (nel nostro caso "C:\ProgramData\winlogbeat\Logs").

Completata la fase di configurazione possiamo testare il corretto funzionamento di Winlogbeat lanciando da PowerShell il comando:

```
PS C:\Program Files\Winlogbeat> .\winlogbeat.exe test config -c .\winlogbeat.yml -e
```

ed infine iniziare a collezionare log dalla macchina Windows/Windows Server con l'istruzione:

```
winlogbeat.exe -c winlogbeat.yml
```

5.8 Collegamento di Grafana ai Datasource

In questa sezione dimostreremo come collegare la nostra installazione di Grafana ai Datasource precedentemente configurati, InfluxDB ed Elasticsearch.

Come già detto Grafana supporta una grande varietà di sorgenti di dati: per ogni Datasource Grafana mette a disposizione un editor di query specifico.

Per connettere correttamente a Grafana i Datasource, basterà effettuare il login con l'utente amministratore alla nostra installazione di Grafana e selezionare Datasource all'interno del pannello di configurazione accessibile dal menu laterale.

Una volta all'interno della configurazione, cliccando sul pulsante "Add data source" dovremmo visualizzare una schermata che conterrà i principali Datasource disponibili in Grafana (vedi figura 14):

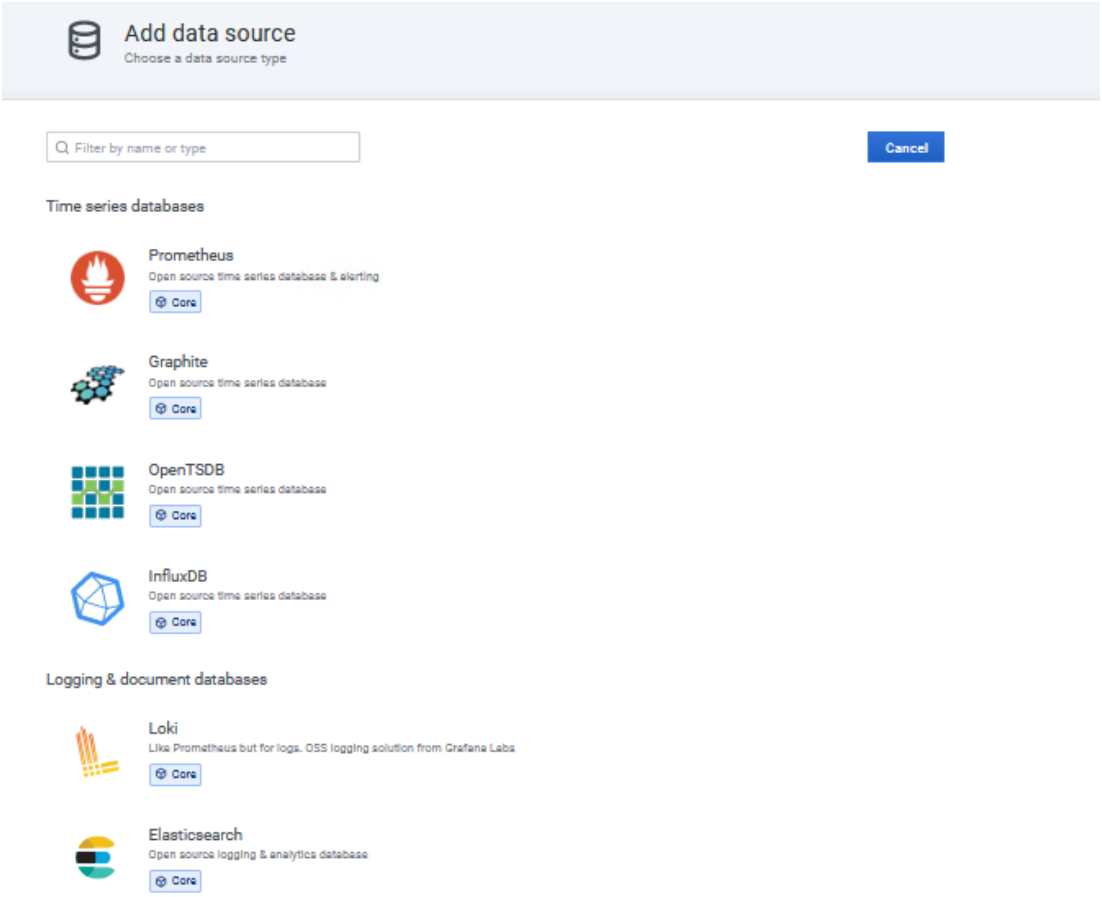

Figura 14 – Datasource in Grafana

Selezioniamo quindi nel nostro caso InfluxDB, trovandoci davanti alla schermata mostrata in figura 15:

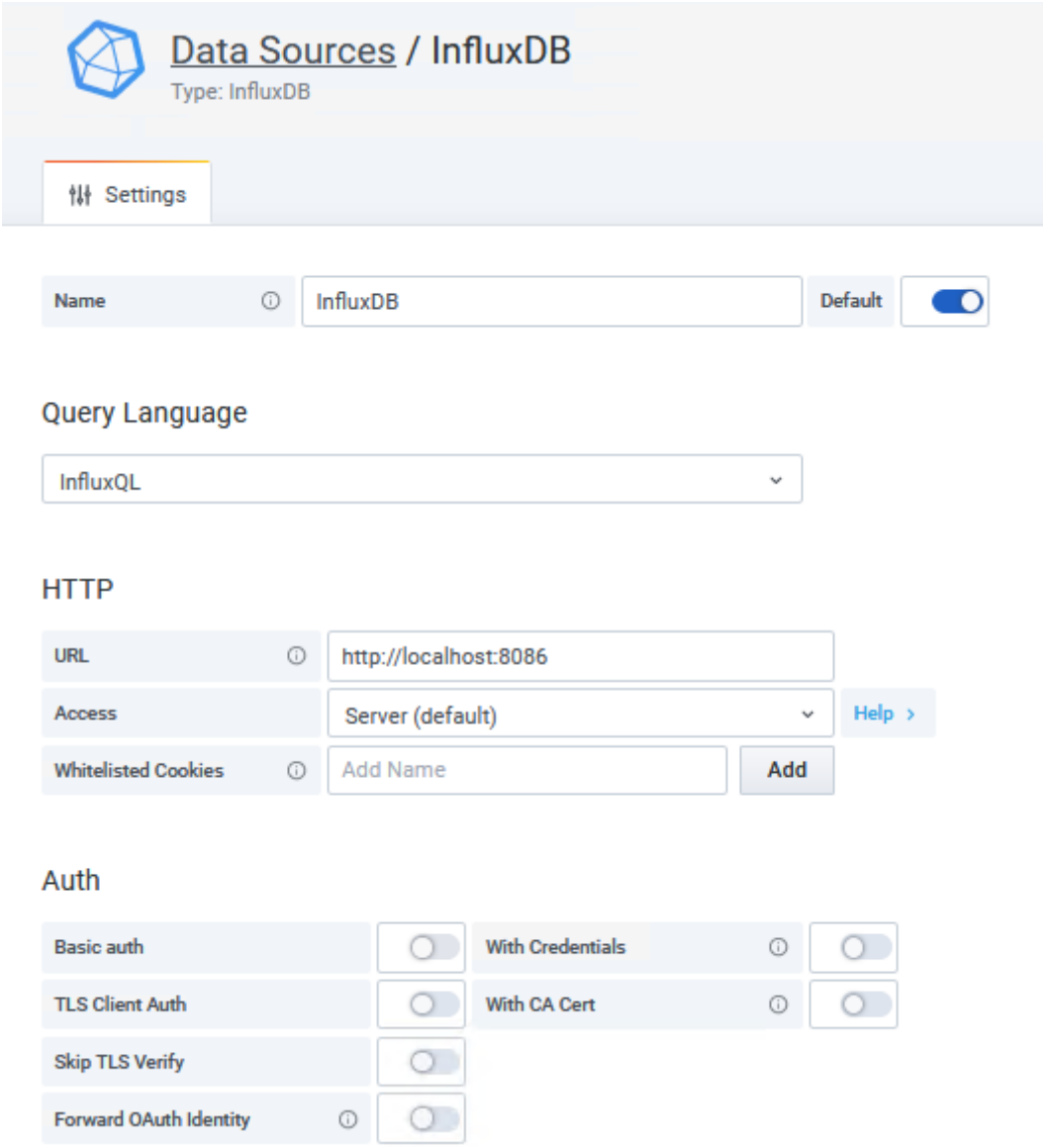

Figura 15 – datasource InfluxDB in Grafana

Andremo a compilare il campo URL della sezione http inserendo l'indirizzo IP della macchina dove risiede l'installazione di InfluxDB e la relativa porta in ascolto messa a disposizione per ricevere metriche dalle varie istanze di Telegraf, utilizzando la sintassi http://indirizzo_ip_server:8086, e selezioneremo, a seconda delle esigenze, la tipologia e la modalità di connessione.

Nel caso in cui l'installazione di InfluxDB risieda nella stessa macchina di Grafana invece basterà lasciare l'URL: http://localhost:8086.

Nel nostro caso InfluxDB è stato selezionato come Datasource di default del nostro sistema di monitoraggio: per fare ciò basta attivare il flag in alto a destra accanto alla scritta “Default”.

Stessa procedura di configurazione andrà applicata ad Elasticsearch; in questo caso l’indirizzo da inserire nella sezione URL sarà seguito dalla porta precedentemente messa a disposizione sulla macchina dove risiede l’installazione di Elasticsearch, la 9200, la porta cioè in ascolto per ricevere il flusso di log inviato dalle istanze di Filebeat e Winlogbeat.

Il risultato sarà di questo processo di configurazione è mostrato in figura 16:

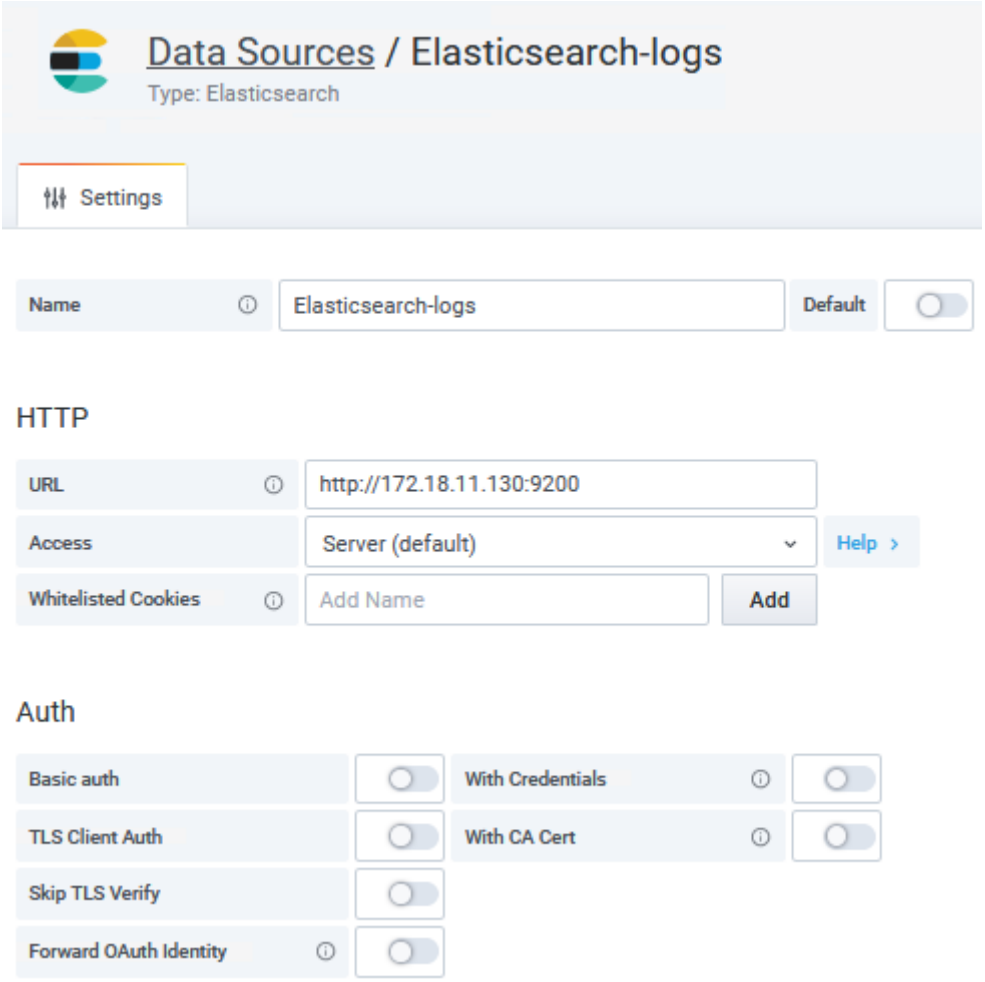

Figura 16 – Datasource Elasticsearch in Grafana

Terminato l'inserimento di ognuno dei nostri Datasource possiamo salvare e testare la configurazione selezionando il pulsante "Save & Test".

5.9 Lo stack TIG

A partire da questa sezione analizzeremo più nel dettaglio il funzionamento del nostro sistema di monitoraggio.

Iniziamo approfondendo il funzionamento dello stack TIG, ossia della componente riservata alla raccolta e all'analisi delle metriche prodotte dalle risorse IT del sistema monitorato composta da Telegraf, InfluxDB e Grafana.

Abbiamo già analizzato nei capitoli precedenti tutti gli step necessari all'installazione dei componenti dello stack TIG e alla loro configurazione, il risultato è riassumibile nello schema seguente:

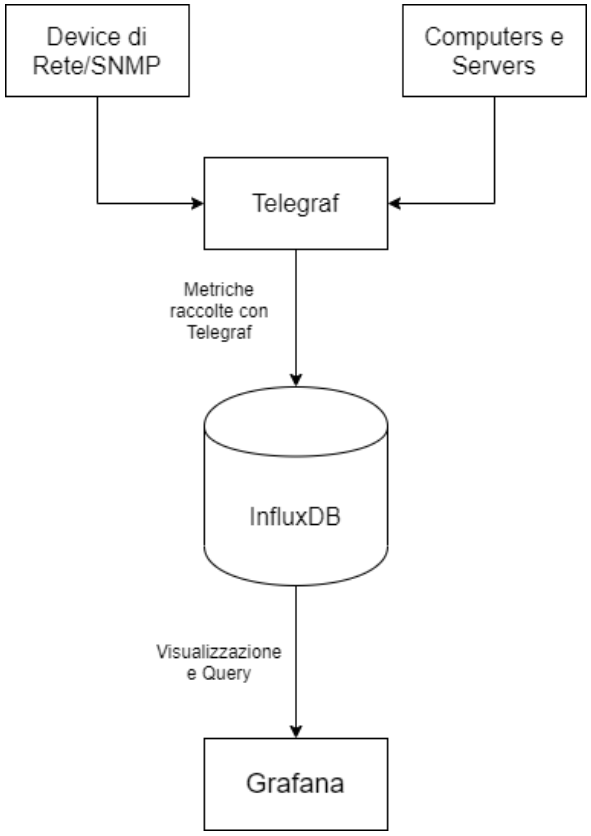

Figura 17 – stack TIG

Come abbiamo già detto, lo stack TIG è in sostanza la struttura principale del nostro sistema di monitoring in quanto risulta altamente scalabile, flessibile e in grado di supportare sia il monitoring con agente (telegraf) che senza agente.

A tal proposito approfondiamo ora il monitoraggio di device attraverso protocollo SNMP sfruttando lo stack TIG.

5.9.1 SNMP monitoring con stack TIG

SNMP sta per Simple Network Management Protocol, si tratta di un protocollo di rete che appartiene alla suite di protocolli internet definito dalla IETF (Internet Engineering Task Force) che consente di semplificare la gestione e il monitoraggio di dispositivi collegati ad una rete.

Elenchiamo una serie di vantaggi che il monitoraggio attraverso protocollo SNMP comporta:

- Supportato da una vasta gamma di device inclusi router, switch, access point wireless, stampanti, scanner e anche device IoT
- Supportato anche da servizi come DHCP
- Adatto ad essere utilizzato sia in rete di piccole che di grandi dimensioni grazie all'ottimizzazione delle risorse necessarie ad effettuare il monitoraggio

I dispositivi compatibili con il protocollo SNMP dispongono già al loro interno di agenti abilitati di default in grado di monitorare costantemente lo stato del dispositivo.

Le informazioni che l'agent SNMP all'interno del device raccoglie vengono scritti all'interno di un database definito Management Information Base o MIB contenuto all'interno del dispositivo stesso.

Quello che il nostro sistema di monitoraggio dovrà fare quindi altro non è che comunicare con il MIB all'interno dei device e leggere gli oggetti al loro interno che sono definiti da identificatori di oggetto o OID.

I messaggi SNMP, noti anche come PDU (Protocol Data Units), che il sistema scambierà con i device possono essere di sette tipi:

- Trap: messaggio di avviso come ad esempio la notifica da parte del device SNMP di un guasto improvviso.

Il monitoraggio delle Trap è fondamentale per rilevare ed affrontare in modo proattivo eventuali guasti o disservizi.

- Get: semplice richiesta di informazioni al dispositivo; è il modo principale per raccogliere informazioni dai device
- GetNext: richiesta di informazioni per i successivo segmento di informazioni
- GetBulk: richiesta di informazioni attraverso una sequenza di GetNext
- Set: invio al dispositivo delle istruzioni per modificare impostazioni o comportamenti
- Response: la risposta fornita dall'agente a una nostra richiesta
- Inform: conferma della ricezione di una Trap

Il monitoraggio SNMP potrà quindi essere implementato in due modalità PULL o PUSH (Trap):

- Modalità PULL: in questa modalità un agent invia richieste SNMP all'agente SNMP interno al dispositivo monitorato a intervalli di tempo regolari.
  L'agente sul dispositivo risponderà con la metrica richiesta
- Modalità PUSH o SNMP Trap: al contrario della modalità PULL il dispositivo monitorato invia il messaggio di stato (Trap) al nostro sistema di raccolta dati. Non c'è nessun tipo di pianificazione e i messaggi vengono inviati nello stesso momento in cui vengono generati fornendoci in tempo reale le informazioni necessarie.

InfluxDB supporta entrambe le modalità di monitoraggio SNMP grazie a due plugin di Telegraf:

- Telegraf SNMP Input Plugin (inputs.snmp)
- Telegraf SNMP Trap Input Plugin (inputs.snmp_trap)

Questo ci consente di avere a disposizione una soluzione completa per il monitoring di tipo SNMP, sia agent che agentless.

Per abilitare gli input attraverso protocollo SNMP per Telegraf installeremo e abiliteremo come prima cosa i pacchetti “net-snmp” e “net-snmp-utils”:

```
yum install net-snmp net-snmp-utils

systemctl enable snmpd
```

Una volta terminate le configurazioni necessarie vediamo nell’esempio seguente come configurare Telegraf per far sì che raccolga informazioni da una sorgente SNMP in modalità PULL:

```
#Recupera i dati dalle sorgenti SNMP remote

[[inputs.snmp]]
  agents = [ "indirizzo_device_1:161", "indirizzo_device_2:161" ]
  timeout = "500s"
  interval = "300s"
  version = 2
  retries = 3
  community = "snmp-test"
  max_repetitions = 100
  name = "snmp-test"
```

Una volta che Telegraf avrà raccolto le metriche necessarie queste verranno inviate al database di InfluxDB.

Lo step successivo sarà rappresentare in una dashboard di Grafana le metriche di cui abbiamo bisogno e settare eventuali alert.

Nel prossimo capitolo vedremo in un esempio dettagliato come creare una dashboard a cui aggiungere i pannelli necessari a rappresentare le metriche raccolte.

5.9.2 Rappresentazione e Alerting delle Metriche in Grafana

Analizziamo ora tutti gli step necessari alla rappresentazione delle metriche raccolte attraverso Telegraf e InfluxDB.

Accediamo quindi alla nostra installazione di Grafana digitando da un browser a nostra scelta l'URL http://indirizzo_ip_grafana:3000 ed effettuando il login utilizzando l'utente "admin".

Nei capitoli precedenti abbiamo già visto come configurare InfluxDB come Datasource predefinito in Grafana: possiamo quindi procedere con la creazione della Dashboard.

Clicchiamo allora sul pulsante Create "+" e selezioniamo Dashboard.

Ci verrà proposta la schermata mostrata in figura 18:

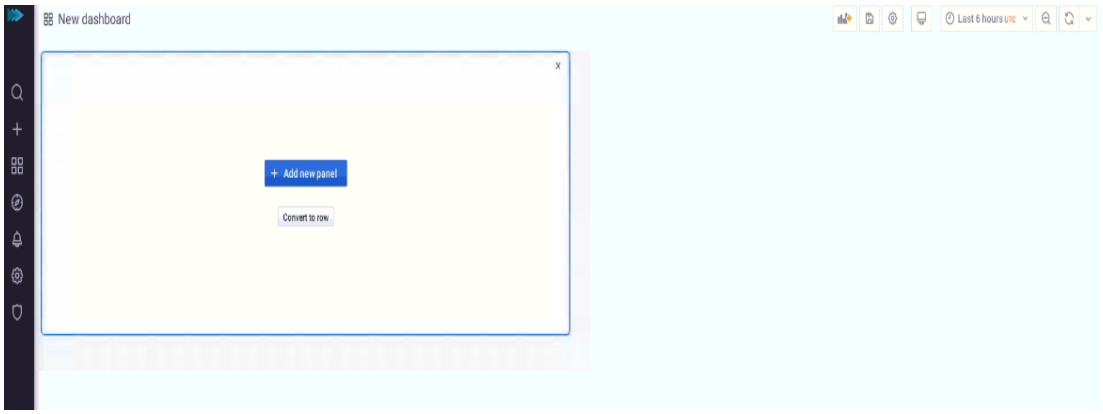

Figura 18 – Aggiunta pannello a Dashboard

Clicchiamo sul pulsante "Add new Panel" e aggiungiamo un nuovo pannello alla nostra Dashboard, otterremo la schermata di figura 19:

Figura 19 – pannello Grafana

Ci troveremo così davanti all'interfaccia di editing di Grafana dove sarà possibile effettuare le query verso il nostro database InfluxDB.

Supponiamo ora di voler visualizzare semplicemente l'utilizzo di CPU di uno dei nodi monitorati: procediamo attraverso i seguenti step:

- nella sezione "Query" verifichiamo di aver selezionato InfluxDB (nel caso non fosse stato impostato come Datasource predefinito)
- Grafana ci mette a disposizione un editor grafico per creare la query che estrarrà i dati necessari organizzata come in figura 20:

Figura 20 – query editor

Nella sezione "FROM" andremo a definire da quale fonte desideriamo ottenere, se "default" nel nostro caso si tratterà di InfluxDB.

“WHERE” ci permetterà di selezionare da quale nodo vogliamo reperire le metriche specificando ad esempio il nome host.

“SELECT” ci permetterà invece di selezionare i tipi di metriche raccolte; come formato ovviamente sceglieremo “Time Series” e li raggrupperemo in base all’intervallo temporale.
Ovviamente sarà possibile anche inserire una query in modalità testuale: basterà infatti cliccare sul simbolo della matita nella finestra della query per poter accedere alla modalità di editing manuale (vedi figura 21):

Figura 21 – query manuale

- Nella parte destra dell’interfaccia di editing potremmo andare a selezionare il tipo di visualizzazione più indicata per rappresentare al meglio i dati collezionati (figura 22):

Figura 22 – visualizzazione pannello

- Selezioniamo ad esempio la visualizzazione “Graph” per creare un grafico che rappresenti l’utilizzo della CPU nel tempo.
  Al di sotto della colonna dei grafici possiamo accedere alla sezione “Display” dove ci sarà possibile selezionare varie modalità di rappresentazione del grafico scelto, come ad esempio la rappresentazione con linee, barre o punti.
- a questo punto avremmo già la possibilità di vedere in anteprima il grafico ricavato grazie alla query impostata nello step precedente (figura 23):

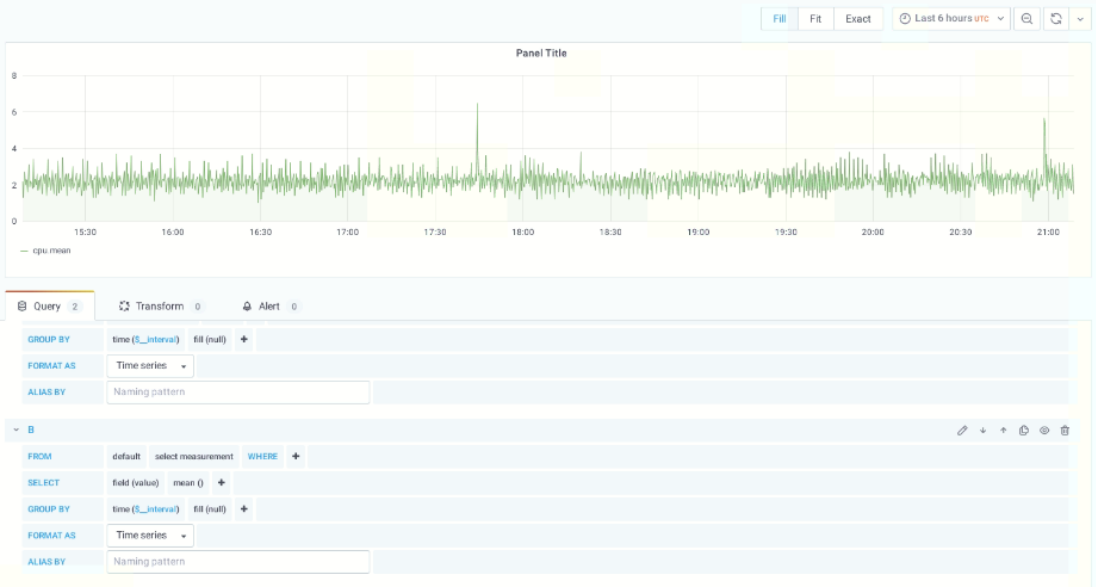

Figura 23 – risultato visualizzazione

- una volta completata la fase di rappresentazione possiamo passare al pannello “Alert” dove ci sarà possibile inserire una condizione al verificarsi della quale Grafana invierà un allarme. L’interfaccia di Grafana per la configurazione di questa componente è mostrata in figura 24:

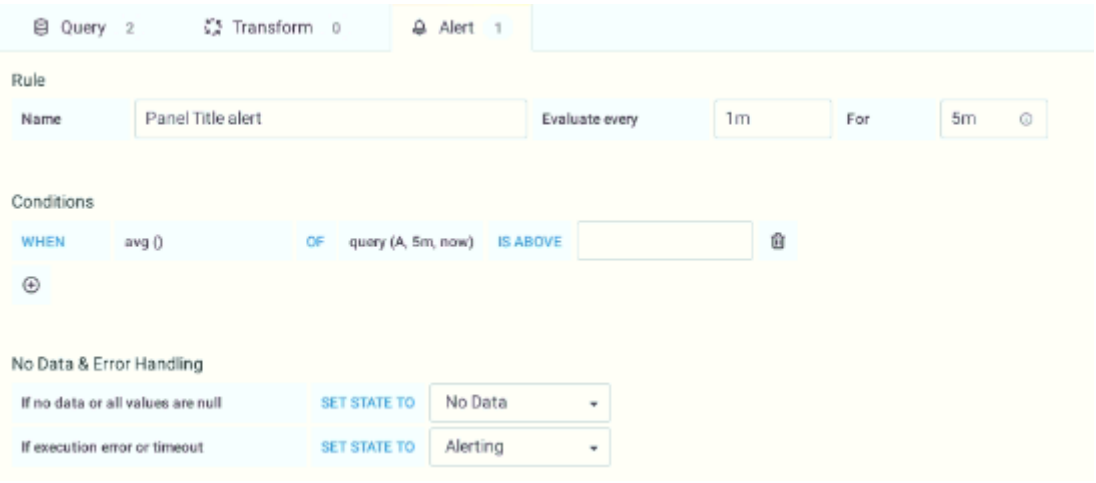

Figura 24 – scheda Alert

Nel campo “Name” andrà inserito il nome del controllo mentre nel campo “Evaluate every” indicheremo il tempo che intercorrerà tra un controllo e

l'altro, nella sezione "Conditions" invece ci sarà possibile settare la condizione da verificare affinché venga lanciato l'alert.

- Nella sezione "Notifications" infine potremo inserire una o più tipologie di notifiche tra quelle supportate da Grafana già elencate in precedenza con il relativo messaggio che sceglieremo di inviare (figura 25):

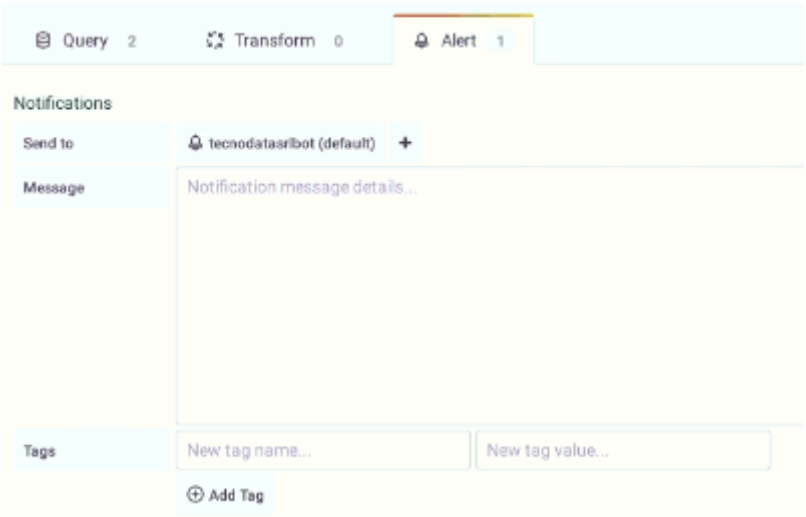

Figura 25 – Notifications in Grafana

5.10 Rappresentazione dei log inviati a Elasticsearch in Grafana

Esaminata la parte relativa all'elaborazione e alla rappresentazione delle metriche, passiamo ora a quella relativa alla rappresentazione e analisi dei log di sistema in Grafana attraverso Elasticearch.

In questo caso il sistema di raccolta e analisi dei log di cui abbiamo già mostrato nel dettaglio installazione e configurazione è ben rappresentato dal diagramma di figura 26:

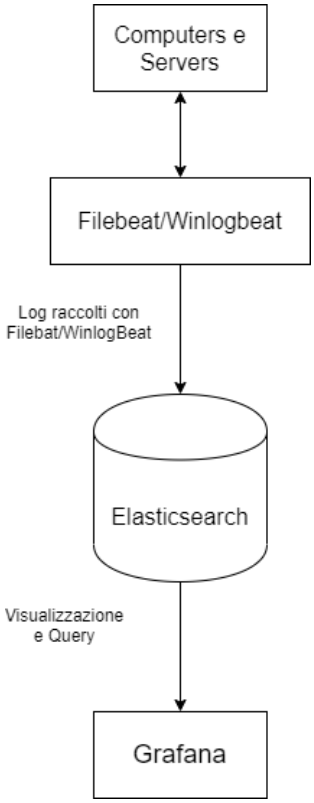

Figura 26 – architettura stack Beat+Elasticsearch+Grafana

La procedura iniziale sarà analoga a quella vista per le metriche ma questa volta dovremmo selezionare come Datasource nel nostro pannello “Elasticsearch-logs” (figura 27):

Figura 27 – Elasticsearch query editor

Nel campo query andremo ad inserire la query relativa all’indice da interrogare in Elasticsearch.

Per quanto riguarda la sezione visualizzazione e la sezione notifiche valgono le stesse considerazioni già fatte per le visualizzazioni delle metriche.

5.11 Personalizzazione di Grafana

Abbiamo visto tutti gli step necessari a creare il nostro sistema di monitoraggio, dalla configurazione degli agent e dei datasource alla creazione della dashboard per visualizzare log e metriche.
Come descritto all'inizio dell'elaborato, il progetto TD Monitor si configura come una soluzione nata per fornire ad aziende o organizzazioni una dashboard per monitorare proattivamente le risorse dei propri sistemi.
Ecco che allora potrebbe essere necessario personalizzare anche graficamente l'aspetto dell'interfaccia riservata agli utenti che dovranno consultare TD Monitor.

La versione Open-source di Grafana infatti ci dà la possibilità di sostituire i file .ico, .svg e .png all'interno della directory /usr/share/grafana/public/img, in modo da poter fornire ad ogni cliente una dashboard personalizzata graficamente.

Contestualmente alla personalizzazione dell'interfaccia Grafana ci da anche la possibilità di creare Dashboard, Pannelli, Utenti e gruppi di utenti (o Team) con autorizzazioni varie.
I permessi degli utenti in Grafana possono essere determinati da:

- Organization Role: posso creare all'interno di Grafana diverse Organizzazioni, ognuna di queste potrà avere tre livelli di amministrazione, admin, editor e viewer. Gli utenti appartenenti ad un'organizzazione con diritti di "admin" potranno effettuare tutte le operazioni, dalla creazione di una dashboard alla modifica di una esistente, gli utenti appartenenti ad un'organizzazione di tipo "Editor" potranno modificare solo le dashboard a loro assegnate e per finire gli utenti facenti parte di un organizzazione con privilegi "viewer" potranno solamente visualizzare le dashboard o i pannelli a loro assegnati.

- Team: ogni utente potrà far parte di diversi Team, i Team potranno contenere anche utenti appartenenti a organizzazioni diverse ma con privilegi assegnati diversi da quelli della loro organizzazione
- Assegnazione diretta dei permessi al singolo utente

Grafana inoltre supporta una grande varietà di modalità di autenticazione facilmente configurabili come ad esempio quella basata su LDAP.

Per quanto riguarda invece le autorizzazioni di dashboard e cartelle (oggetti che contengono dashboard), possiamo seguire lo stesso ragionamento fatto per gli utenti, con la differenza che queste consentono di rimuovere o aggiungere autorizzazioni solo per organizzazioni o a team specifici.

Capitolo 6 - Conclusioni e Sviluppi Futuri

L'analisi, l'organizzazione e lo sviluppo di tutti i punti espressi nei capitoli precedenti sono state le attività sviluppate dallo scrivente durante il percorso di tesi descritto in questa relazione.

Nel capitolo dedicato agli obiettivi di progetto è stato descritto quali sono le esigenze e le funzioni che un sistema di monitoraggio IT deve soddisfare affinché sia in grado di fornire all'utente finale una soluzione efficiente ed efficace in grado di monitorare in modo proattivo un Network IT di un Azienda o di un Organizzazione.

Nelle fasi successive sono stati analizzati tutti i componenti della soluzione TD Monitor, spiegando come questi siano stati scelti in base a linee guida dettate da performance ottimali, stabilità e security by design.

Infine, sono state illustrate le fasi di installazione e configurazione della soluzione proposta, approfondendo di volta in volta le scelte e le tecnologie adottate.

È stato inoltre approfondito nello specifico il cuore del progetto, ossia la raccolta, l'analisi e il monitoraggio delle metriche attraverso time series.

Tuttavia, è facilmente intuibile come i servizi di monitoring erogati da TD Monitor possano essere estesi anche ad altre sorgenti dati come ad esempio l'interrogazione di database di vario genere, come PostgreSQL e MySQL, o anche alla raccolta di dati provenienti da sorgenti non propriamente IT come sensori di macchinari in ambienti di produzione o dispositivi IoT.

La soluzione TD Monitor si presta inoltre a eventuali integrazioni e sviluppi futuri.

È possibile infatti integrare TD Monitor con algoritmi di tipo machine learning o deep learning in grado di potenziare le funzioni della soluzione illustrata.

Con un algoritmo di intelligenza artificiale integrato, infatti, la nostra soluzione di monitoring farebbe un ulteriore salto in avanti, riuscendo non solo a monitorare proattivamente il sistema inviando opportune notifiche, ma addirittura potrebbe interpretare i comportamenti delle risorse monitorate e predire il manifestarsi di eventuali problematiche, proponendo così soluzioni preventive con indubbi vantaggi per l'utenza finale, sia in termini di performance del sistema che in termini economici.

Bibliografia e Sitografia

[1] https://www.tecnodatasrl.com - Pagina Web Tecnodata SRL – consultato a Luglio 2020

[2] https://www.garanteprivacy.it – sezione relativa a Regolamento generale protezione dei dati - consultato a Luglio 2020

[3] Paul Dix – "1000x better performance from Time-Series DBs compared to others" Whitepaper https://www.influxdata.com/resources/why-time-series-matters-for-metrics-real-time-and-sensor-data/ - consultato a Maggio 2020

[4] https://docs.influxdata.com – Documentazione Ufficiale InfluxDB - consultato a Maggio 2020

[5] https://docs.influxdata.com/telegraf/v1.15/ - Documentazione Ufficiale Telegraf relativa alla versione 1.15 - consultato a Giugno 2020

[6] Pranav Shukla e Sharath Kumar - "Learning Elastic Stack 7.0 – Second edition" – Packt 2019

[7] Chris Churilo – "InfluxDB vs. Elasticsearch for Time Series Data & Metrics Benchmark" - (https://influxdata.com/blog) - consultato a Luglio 2020

[8] https://grafana.com/docs/ - Documentazione Ufficiale Grafana relativa alla Versione 7.0 - consultato a Maggio 2020

[9] https://docs.centos.org/ - Documentazione Ufficiale CentOS relativa alla versione 7 - consultato a Luglio 2020