Università degli Studi di Camerino

SCUOLA DI SCIENZE E TECNOLOGIE

ELK Stack e Pentaho per l'analisi di dati in ambito universitario

Laureandi

Samuel Piatanesi

Luca Ruschioni

Relatori

Prof. Roberto Gagliardi

Prof. Fausto Marcantoni

Correlatore

Prof. Francesco De Angelis

Corso di Laurea in Informatica

A.A. 2018/2019

Indice

Abstract

L’informatizzazione ha ormai cambiato in modo rapido quanto radicale ogni aspetto della nostra vita, dall’ambito lavorativo a quello sociale, acquisendo sempre più centralità ed importanza.

Questo cambiamento ha portato alla produzione di grandissime quantità di dati informatici – big data – che si sono ben presto rilevati risorse di inestimabile valore soprattutto per le grandi aziende. Di conseguenza è cresciuta anche la necessità di gestire queste preziose informazioni. Analizzandole in modo adeguato infatti, è possibile derivare, attraverso strumenti evoluti, statistiche utili sotto molti aspetti, tra cui lato marketing e finanziario.

Nello stesso modo l’interconnessione attraverso le reti aumenta il grado di rischio dei sistemi, perciò controllare e capire quello che sta succedendo all’interno di specifiche reti di calcolatori, in tempo reale, è indispensabile per salvaguardare dati sensibili. Per fare ciò, vengono utilizzati sistemi centralizzati dove si registrano e si analizzano in modo sequenziale tutte quelle informazioni che identificano le attività di dispositivi connessi alla rete, le applicazioni in esecuzione e gli eventi collegati agli utenti del sistema.

1. Introduzione

1.1. Scenario iniziale

Il progetto commissionato come ricerca da svolgere durante lo Stage++, sotto la supervisione dei professori *Fausto Marcantoni*, *Roberto Gagliardi* e *Francesco De Angelis*, prevedeva inizialmente la realizzazione di un sistema centralizzato, un server, predisposto ad analizzare i log provenienti da sistemi e dispositivi differenti all'interno della rete universitaria di Unicam.

In seguito, lo scenario è stato ampliato, per includere la possibilità di immagazzinare record provenienti dal database accademico, con lo scopo di consentirne lo studio attraverso grafici statistici da mettere a disposizione del personale docente ed amministrativo. Questi, fruibili attraverso un'applicazione accessibile previa autenticazione, e personalizzata in base all'utente che ha effettuato il login nella piattaforma.

1.2. Obiettivi

La sviluppo del sistema, nella pratica due applicazioni web, una per i log e una per il database, prevede due principali focus riguardanti i dati, dai quali derivano differenti obiettivi, fondamentali da raggiungere:

- *Raccolta***,** i log generati all'interno della rete devono essere filtrati e memorizzati in real time, per consentirne la visualizzazione in tempo reale nel breve periodo; differente è invece la gestione dei big data: trattandosi di informazioni conservate nel tempo necessitano solamente di essere raccolti in modo statico e permanente, oltre che sicuro
- *Analisi***,** tutti i dati memorizzati nel sistema devono poi essere analizzati in modo coerente per creare statistiche ed avere informazioni aggiuntive utili in un quadro generale più ampio; i log sono utili ad amministratori di sistema per monitorare le attività all'interno della rete, mentre le query verso il database sono interrogabili dal personale docente per creare viste personalizzate

1.3. Log di sistema

Nel campo informatico i *log* sono identificati come una lista di informazioni derivanti da attività svolte dal sistema preso in considerazione, che ne descrivono il comportamento. Si possono riconoscere ed elencare cinque principali tipologie di log:

- *Log di Sistema*, che tracciano tutti gli eventi relativi al funzionamento dei sistemi operativi degli apparati infrastrutturali
- *Log Applicativi*, rappresentanti gli eventi dei software in esecuzione, tra cui eventuali errori o anomalie
- *Log di Sicurezza*, che servono a tenere sotto controllo tutte le operazioni degli utenti connessi, che potrebbero compromettere l'integrità della piattaforma
- *Log di Database*, che salvano su un registro tutte le operazioni riguardanti la base di dati, come l'inserimento, la modifica o l'eliminazione di record
- *Log di Accesso*, che registrano i tentativi di riconoscimento di un utente, quali login e logout

Queste informazioni hanno assunto un valore non indifferente nel mondo lavorativo, rappresentando un'importante fonte di investigazione, e con esse l'attività di analisi che ne scaturisce, composta da diverse fasi operative attraverso le quali estrapolare dati significativi:

1. Innanzitutto bisogna conoscere quali sono le applicazioni/sistemi target da analizzare e come i log vengano da essi generati
2. Tutti i log sono memorizzati in modo temporaneo in una locazione precisa del sistema, venendo sovrascritti dai nuovi in maniera ciclica, per non creare una collezione di dati troppo grande
3. I log devono essere gestiti per essere inviati ad un sistema di memorizzazione "permanente", solitamente in real-time per avere un riscontro immediato
4. I log ricevuti dal lato server sono memorizzati da un software che si occuperà di tradurli con l'utilizzo di pattern adatti
5. I log catalogati sono poi oggetto di analisi per la creazione di statistiche personalizzate

Syslog è un protocollo di rete ormai riconosciuto come uno standard per trasmettere attraverso una rete semplici informazioni di log. Ogni messaggio è etichettato da un *facility code*, che indica il tipo di software che genera il messaggio, e da un *severity code*, che indica il tipo di messaggio (da 0 a 7) (Figura 1) [1].

Facility code	Keyword	Description
0	kern	Kernel messages
1	user	User-level messages
2	mail	Mail system
3	daemon	System daemons
4	auth	Security/authentication messages
5	syslog	Messages generated internally by syslogd
6	lpr	Line printer subsystem
7	news	Network news subsystem
8	uucp	UUCP subsystem
9	cron	Clock daemon
10	authpriv	Security/authentication messages
11	ftp	FTP daemon
12	ntp	NTP subsystem
13	security	Log audit
14	console	Log alert
15	solaris-cron	Scheduling daemon
16–23	local0 – local7	Locally used facilities

Value	Severity	Keyword	Deprecated keywords	Description
0	Emergency	`emerg`	`panic` [7]	System is unusable
1	Alert	`alert`		Action must be taken immediately
2	Critical	`crit`		Critical conditions
3	Error	`err`	`error` [7]	Error conditions
4	Warning	`warning`	`warn` [7]	Warning conditions
5	Notice	`notice`		Normal but significant conditions
6	Informational	`info`		Informational messages
7	Debug	`debug`		Debug-level messages

Figura 1 - Tabelle codici syslog [fonte Wikipedia]

1.4. Database

Il *database*, chiamato anche base di dati, consiste in un insieme di dati raccolti all'interno di un calcolatore, per permettere lo svolgimento di ricerche automatizzate, sempre più complesse con l'aumentare della quantità di informazioni.

I database possono essere di differente tipologia:

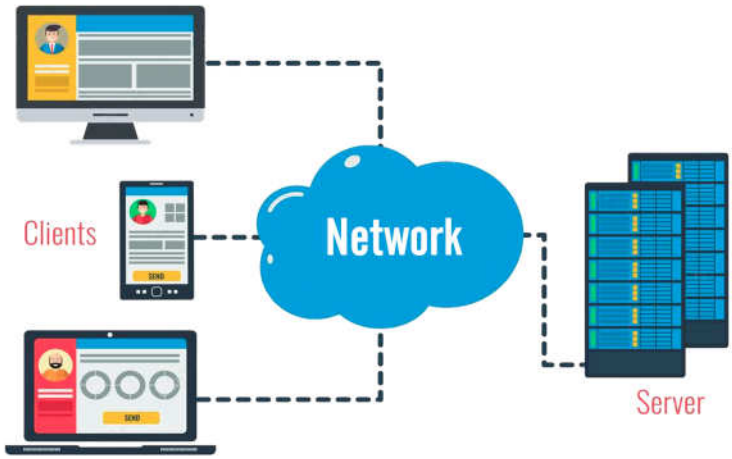
- *Relazionali*, nei quali per ogni record viene definita una chiave, ovvero un identificatore univoco. Tra una relazione e l'altra, è proprio essa che va a definire il legame tra due o più elementi. Il linguaggio più utilizzato per gestire questo tipo di database è l'SQL
- *Navigazionali*, non sono previste tabelle e relazioni e ci si muove attraversando l'intero database per trovare il dato interessato
- *Multidimensionali*, vanno a compensare le scarse performance offerte dai database relazionali nel caso di utilizzo delle basi di dati stesse per processi di analisi
- *A oggetti*, consentono agli utenti di memorizzare veri e propri oggetti all'interno del database, utilizzando gli stessi principi della programmazione a oggetti

Il server si occupa di fornire i servizi di utilizzo della base di dati ad altri programmi e ad altri dispositivi secondo la modalità client/server. Memorizza i dati, riceve le richieste dei client ed elabora le risposte appropriate.

Tra i più diffusi sistemi open source DBMS - Database Management System - disponibili troviamo *MySQL*, *MariaDB* e *PostgreSQL*. In questo specifico progetto è stato utilizzato il primo, in quanto offre un sistema di gestione di database, presente ormai da oltre 20 anni, perfettamente compatibile con le piattaforme utilizzate e largamente documentato [2].

1.5. Server

Concettualmente, il modello logico associabile ed in effetti applicato in questo tipo di architettura è l'ormai noto *Client-Server*: lo sviluppo dei due sistemi di gestione prevede infatti l'utilizzo di altrettante macchine lato server che avrebbero centralizzato la raccolta dei dati provenienti da svariati hosts presenti nella rete.



Focalizzando ora l'attenzione sulla parte server, la scelta per il sistema operativo da installare per questo tipo di applicazione è ricaduta su *CentOS* (Versione 7*)*, virtualizzato tramite il software *VirtualBox*. Questa distribuzione *Linux* da anni presente sul mercato, può essere considerata a tutti gli effetti un sistema Enterprise appropriato per utilizzo aziendale, in particolare per attività di tipo server. Altre caratteristiche molto importanti che hanno influenzato la scelta sono state:

- Stabilità, in quanto il livello di integrazione ed affidabilità è garantito da anni di sviluppo e versioni consolidate
- Grado di sicurezza elevato (i software sono testati e resi sicuri, non includono mai, o molto raramente, versioni beta)
- Supporto fornito da una grande community e documentazione molto ampia

2. Centralizzazione dei log

La gestione dei log (*log management*) comprende tutte quelle attività riguardanti la raccolta e l'analisi degli eventi che accadono in qualunque contesto di carattere informatico, all'interno di un sistema complesso. Queste operazioni possono e devono essere facilitate da un'efficace centralizzazione delle informazioni.

Questo è di vitale importanza, nello specifico per le aziende che sono sempre più soggette a subire attacchi informatici. Con una corretta centralizzazione dei log è possibile prevenirli, comprendendo cosa è successo o cosa sta succedendo in tempo reale, grazie a tool grafici che permettono di effettuare statistiche o di inviare messaggi di allarme non appena vengono identificate attività sospette. Un "cracker" va a colpire la rete aziendale in punti differenti, ma un attaccante lascia sempre qualche traccia delle proprie azioni.

2.1. Scelta del software

La scelta del software da utilizzare comprende la comparazione analitica preliminare di tecnologie adatte allo scopo, con un'adeguata ricerca di informazioni utili a determinare il miglior sistema che soddisfi i requisiti sopra descritti. Inizialmente sono state confrontate tre tecnologie idonee per l'analisi dei log in real time, simili sotto molti aspetti ma con caratteristiche discostanti, aprendo la possibilità a soluzioni differenti.

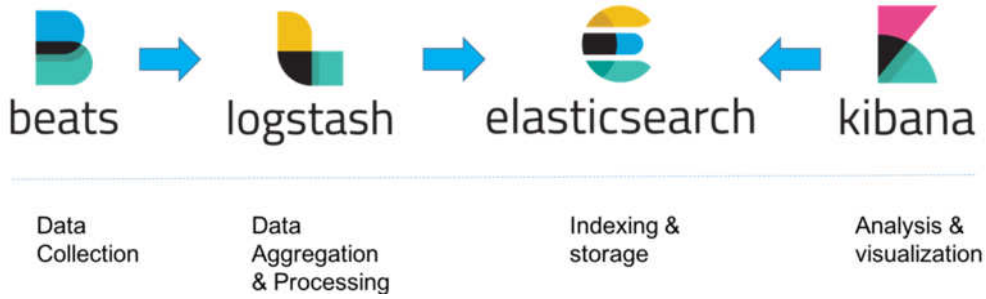
Rsyslog è il più veloce trasmettitore di dati tra le opzioni considerate, molto leggero e flessibile, è in grado di svolgere il proprio compito anche con risorse limitate. Nonostante ciò non ha molta documentazione e supporto, e in più non è adatto per gestire grosse quantità di informazioni.

Syslog-ng è anch'esso veloce e leggero, ma sebbene sia supportato e abbia un'ottima documentazione, la versione open source ha molte limitazioni rispetto alla versione Premium, e non offre un'esperienza soddisfacente [3].

L'ultimo sistema testato è quello che alla fine è stato poi designato come il più adatto allo scopo. *ELK Stack* è un acronimo che identifica un'architettura formata da tre software open source complementari, ognuno dei quali ricopre un ruolo all'interno dell'ambiente globale:

- *Elasticsearch* rappresenta il database dove i dati vengono immagazzinati
- *Logstash* processa, filtra ed invia i dati
- *Kibana* è l'interfaccia grafica che permette all'utente di visualizzare queste informazioni con tabelle e grafici, prelevandole dal database

Per inviare i dati dai dispositivi di rete o dai client si utilizzano invece i prodotti del pacchetto *Beats*, che include una vasta gamma di software, ognuno designato per un caso d'uso differente.



2.2. Installazione ELK Stack

Consultando come riferimento la guida presente su Digital Ocean [4], nota community per sviluppatori, sono stati seguiti gli step per l'installazione dell'intera architettura *ELK* sulla prima macchina virtuale CentOS, rispettando i requisiti minimi.

Tutti e tre i software sono stati inizialmente installati nella loro versione 6.6.0, in quanto al momento la 6.7.0 risultava in fase beta e causava alcuni problemi di compatibilità con il sistema. Successivamente sono stati tutti aggiornati alla versione 7.4.0.

Di default *Elasticsearch* non è presente nella repository standard di CentOS, quindi occorre installare il tutto tramite *yum.* Con il seguente comando si importa la chiave pubblica GPG:

```
sudo rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

In seguito si aggiunge manualmente la repository di *Elasticsearch*, per poter poi procedere con l'installazione:

/etc/yum.repos.d/elasticsearch.repo

```
[elasticsearch-7.x]
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=1
autorefresh=1
type=rpm-md
```

```
sudo yum install elasticsearch
```

Fondamentale è la modifica del file di configurazione, dove è necessario impostare il campo network.host con l'indirizzo sul quale si vuole far eseguire il software. Inizialmente è possibile impostare questo come localhost (indirizzo dell'interfaccia di loopback), ma qualora si volesse raggiungere *Elasticsearch* dall'esterno, questa impostazione dovrà essere modificata.

/etc/elasticsearch/elasticsearch.yml

```
. . .
network.host: localhost
. . .
```

Tutti, o la maggior parte, dei file di configurazione sono nel formato YAML, nel quale l'indentazione corretta è obbligatorio. Anche un carattere di spazio posto in una riga sbagliata, causa errore di parsing.

Completata l'installazione e la configurazione del pacchetto è possibile avviarlo e impostarlo per procedere all'accensione del sistema:

```
systemctl start elasticsearch
systemctl enable elasticsearch
```

Per quanto concerne *Kibana*, la configurazione in un primo momento può essere lasciata quella predefinita e sono quindi sufficienti i tre comandi:

```
yum install kibana
systemctl start kibana
systemctl enable kibana
```

L'ultimo componente rimasto è infine *Logstash*:

```
yum install logstash
```

Per eseguire però questo software è consigliato utilizzare al posto di *systemctl*, il seguente comando (soluzione consigliata per operazioni di testing, in quanto viene stampato automaticamente l'output, ed è quindi una sorta di modalità "verbose" molto comoda), all'interno del path `/usr/share/logstash` (Figura 2):

```
bin/logstash -f log.conf
```

L'opzione -f permette di impostare il programma adottando una configurazione personalizzata contenuta in uno specifico file con estensione *.conf*, in questo caso *log.conf*.

Nel capitolo successivo verrà affrontato in modo dettagliato, come settare correttamente il file di configurazione di Logstash, in relazione allo scopo prefissato.

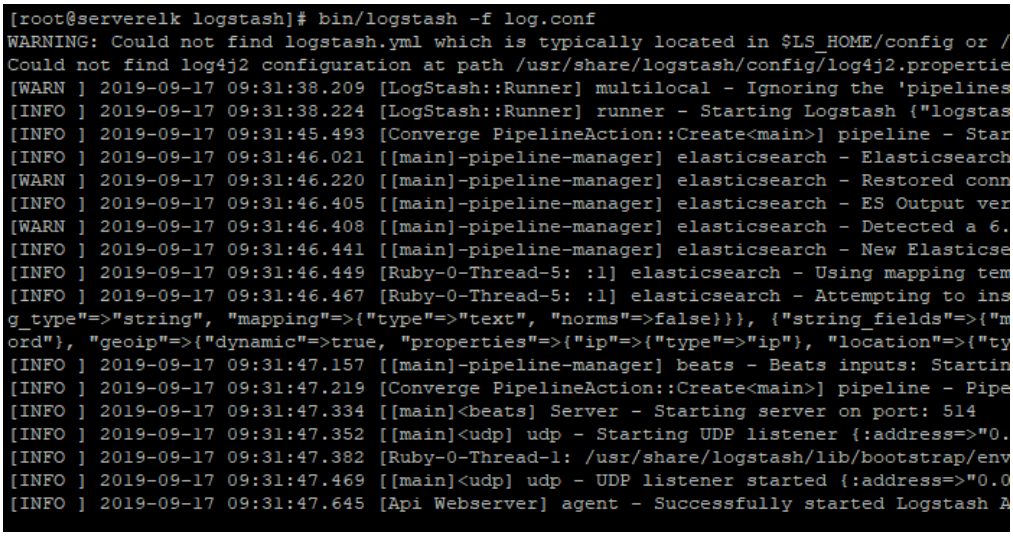


Figura 2 - Avvio Logstash da terminale

Quando la configurazione è divenuta stabile e definitiva la si aggiunge dentro al percorso principale `/etc/logstash/conf.d`, in modo tale che una volta avviato con i comandi descritti in precedenza, *Logstash* la riconosca in modo automatico, adottandola come di default.

Completate le configurazioni e avviati tutti i software, è possibile accedere alla console di *Kibana* all'indirizzo *http://localhost:5601*

2.3. Parsing dei log

All'interno dell'architettura *ELK Stack*, *Logstash* è il componente che si occupa di ricevere log grezzi, elaborarli seguendo pattern e filtri per poi inviarli ad *Elasticsearch*, che si occupa di immagazzinarli.

Creando come esempio iniziale la seguente configurazione [5] si mette in ascolto il servizio per ricevere dati tramite protocollo udp, in ascolto nella porta specificata, inviandoli in output ad *Elasticsearch*:

```
input {
  udp {
    port => 514
    type => syslog
  }
}


output {
  elasticsearch { hosts => ["localhost:9200"] }
}
```

Essendo configurato uno switch di prova da parte del sistemista dell'università, è iniziato un primo test per la raccolta dei log, per comprendere il funzionamento generale dell'architettura. Ovviamente, per verificare la corretta ricezione dei dati si è utilizzato *tcpdump* (sull'interfaccia di rete ens30 e sulla porta 514):

```
tcpdump -i ens30 port 514
```

Nello specifico, questo tool consente l'intercettazione (in termini tecnici lo "sniffing") di pacchetti in entrata e in uscita dall'interfaccia di rete e sulla porta interessata, e permette di visionare se e quali dati arrivano. Ovviamente le informazioni raccolte erano semplici messaggi di log, non utili singolarmente in quanto non ancora divisi in campi adatti a creare statistiche e grafici di analisi. Arricchendo la configurazione precedente con filtri e pattern personalizzabili si ha invece un riscontro più preciso e accurato dei dati ricevuti.

Per effettuare la necessaria “traduzione” è disponibile online un applicativo chiamato *Grok Debugger* [6] (è presente anche all’interno di *Kibana*, ma alcune app online presentano più funzionalità utili). Questa tipologia di programma aiuta a capire come dividere il messaggio in più campi (Figura 3). Vengono utilizzati dei pattern specifici per suddividere in maniera personalizzata l’intero dato, in modo tale da avere per ogni log, il campo del protocollo, dell’host, dell’orario, ecc…

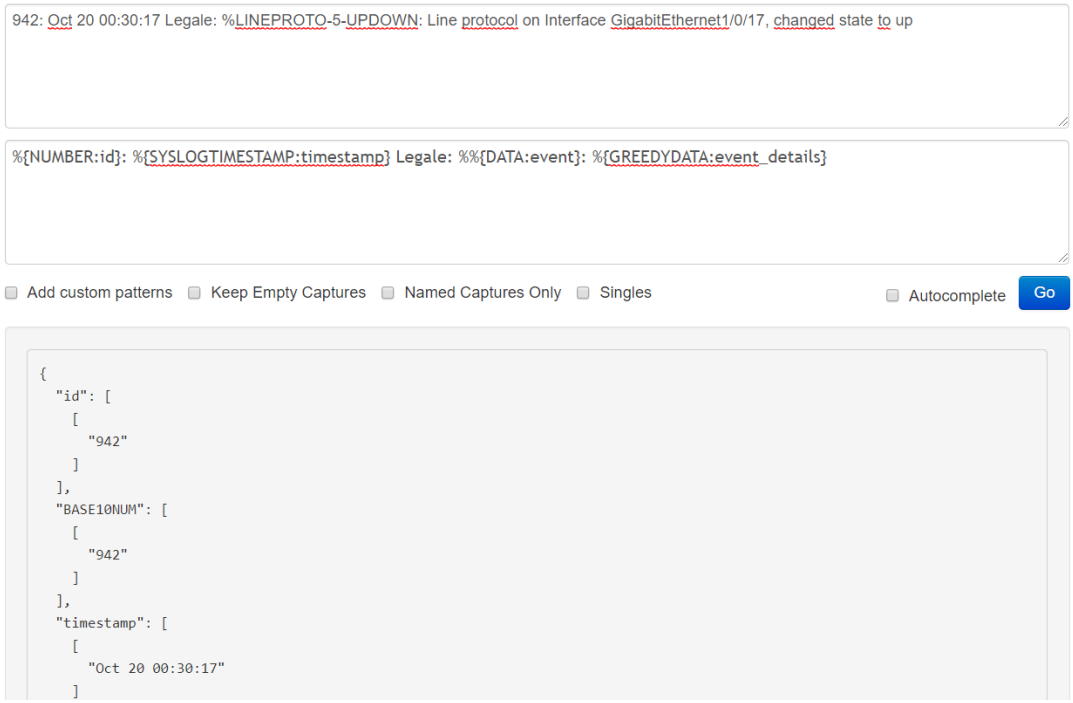


Figura 3 – Grok Debugger

Una volta ottenuto il pattern corretto, va aggiunto alla configurazione *Logstash* utilizzata, attraverso il plugin Grok [7].

```
filter {
   grok {
      match => {
         “message” => “%{NUMBER:id}:
%{SYSLOGTIMESTAMP:timestamp} Legale: %%{DATA:event}:
%{GREEDYDATA:event_details}”
      }
   }
}
```

Per fare il parsing si trovano in rete pattern grok già preimpostati, come ad esempio i pattern per *PfSense* [8], *Syslog* [9], *Cisco* [13] e *Nginx* [11]. A volte può capitare usando questi pattern che uno stesso campo viene estratto in maiuscolo o in minuscolo e quindi c’è un mismatch da due termini simili (ad esempio “udp” e “UDP”), in questo caso è utile il filtro di *Logstash*, mutate [12].

```
mutate {
   lowercase => [ 'protocol' ]
}
```

Per gestire dati provenienti da diverse fonti, occorre creare più indici su *Elasticsearch*, così da poter successivamente separare le informazioni su *Kibana*.

```
input {
   beats {
      port => 5044
      type => winlog
   }
   udp {
      port => 514
      type => syslog
   }
}
```

In questo caso specifico vengono creati due index differenti, uno “winlog” e uno “syslog”, e di conseguenza su *Kibana* potranno essere suddivisi in gruppi distinti.

2.4 Invio dati da Windows e Linux

L'invio di dati da parte di apparati Windows e Linux può essere gestito sia attraverso *Logstash*, sia mediante l'utilizzo di appositi software dell'architettura *ELK*, già citati in precedenza, appartenenti al pacchetto *Beats*. Nello specifico, si è ritenuto opportuno adoperare i seguenti tre componenti (fra i molti presenti):

- *Filebeat*, adatto per inviare grandi quantità di log provenienti da software differenti – fra tutti Nginx, MySQL e log di sistema (guida all'installazione [13])
- *Winlogbeat*, sviluppato per l'invio di log provenienti da ambienti Windows Server (guida all'installazione [14])
- *Auditbeat*, utilizzabile all'interno di sistemi Linux per ogni tipo di evento riguardante la sicurezza (guida all'installazione [15])

Tutti i *Beats* operano tramite protocollo TCP (affidabile e sicuro) e le informazioni vengono preliminarmente parzializzate seguendo pattern di default. Perciò, configurando correttamente l'indirizzo del server, si possono inviare dati già elaborati direttamente ad *Elasticsearch*. Questo non esclude la possibilità di eseguire un'ulteriore elaborazione tramite *Logstash*, che può in alternativa essere completamente tralasciato.

Nella configurazione di uno di questi software, è possibile infatti selezionare l'output desiderato verso il quale inviare i log. È sufficiente commentare e/o de commentare le righe all'interno della configurazione in base alle preferenze dell'amministratore.

```
output.elasticsearch:
  hosts: ["https://localhost:9200"]

#output.logstash:
  #hosts: ["127.0.0.1:5044"]
```

Un'ulteriore caratteristica che rende questi componenti ancor più utili e completi è rappresentata dalle dashboard standard già presenti nel software. Possono essere inviate direttamente a *Kibana* modificando come segue la configurazione, così da collegare automaticamente gli index ricevuti con i grafici di default [16].

```
setup.kibana.host: "http://localhost:5601"
setup.dashboards.enabled: true
```

2.5 Gestione e analisi dei dati

La gestione dei dati è affidata completamente ad *Elasticsearch*, che è a tutti gli effetti un database completamente testuale, distribuito e non relazionale (*NoSQL*).

L'interrogazione dei dati memorizzati avviene attraverso l'utilizzo di *Kibana*. Se sono stati settati correttamente tutti e tre i software, all'interno della sezione Management / Index Management saranno visibili gli indici all'interno del database *Elasticsearch*, ricevuti da *Logstash* o direttamente dai *Beats*.

Sarà possibile quindi andare a creare dei pattern per questi index, step obbligatorio per procedere con l'utilizzo della piattaforma (Figura 4).

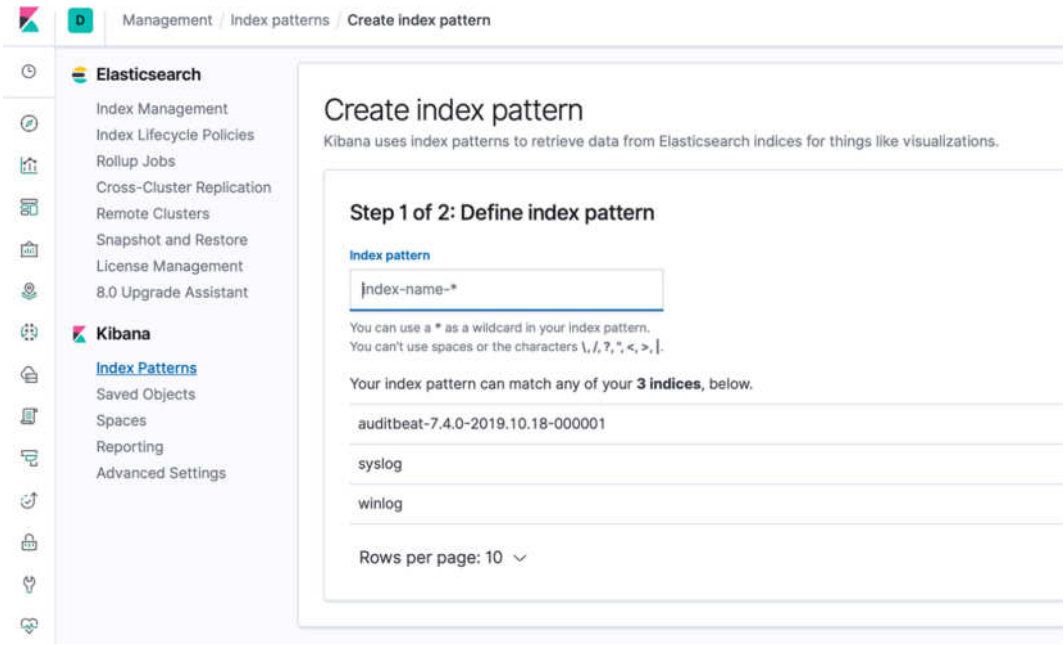


Figura 4 – Creazione di un index pattern

Le informazioni all'interno dell'indice creato possono provenire da differenti host del sistema, ma devono essere ordinate seguendo un vincolo temporale che può essere rappresentato dalla data in cui il log è stato ricevuto o dalla tempistica durante la quale l'evento si è verificato.

Alla creazione dell'indice segue una visualizzazione preliminare dalla sezione *Discover* (Figura 5), utile soprattutto per esaminare i dettagli presenti all'interno dei log (formato *raw*) e quali di essi potranno poi essere analizzati per creare statistiche e grafici adeguati.

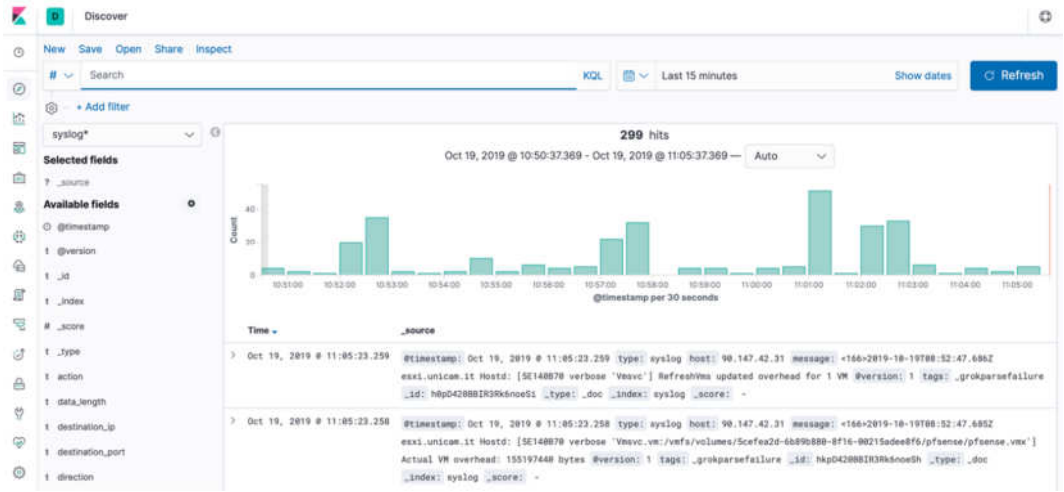
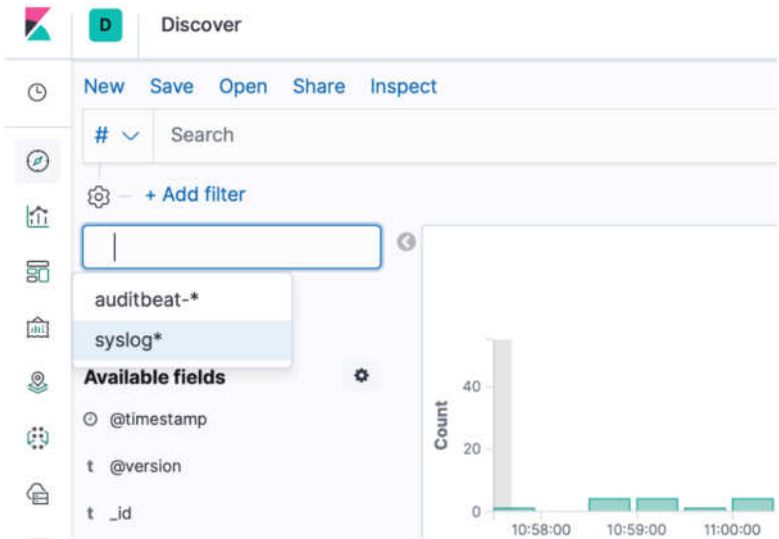


Figura 5 - Sezione Discover

Dalla stessa sezione è possibile cambiare l'indice da visualizzare attraverso il menu a tendina posto in alto a sinistra.



Va tenuta in considerazione la possibilità che, soprattutto se vengono usati pattern personalizzati in *Logstash*, potrebbe verificarsi un errato parsing dei log ricevuti. Il sistema rileva automaticamente quest'eventualità e indica l'errore attraverso il tag `_grokparsefailure`, che sta ad evidenziare la mancata o non corretta elaborazione del messaggio.

Come si può vedere nell'immagine (Figura 6) i campi del messaggio non vengono divisi, ma sono tutti accorpati in un unico tag "message", perciò sarà necessario rivedere il pattern costruito precedentemente in *Logstash*.

⊙	@timestamp	Oct 19, 2019 @ 11:08:37.735
t	@version	1
t	_id	WEpG420BBIR3Rk6nmeVN
t	_index	syslog
#	_score	-
t	_type	_doc
t	host	90.147.42.31
t	message	<30>2019-10-19T08:56:02Z esxi.unicam.it root: vsanObserver.sh: vsantraced is not started - can't start vsanObserver
t	tags	_grokparsefailure
t	type	syslog

Figura 6 - Esempio di grokparsefailure

2.6 Creazione di una dashboard su Kibana

Una delle operazioni più importanti per l'utente di *Kibana* è quella di poter creare e gestire dashboard personalizzate, per controllare il continuo flusso di dati o per esportare grafici su sistemi esterni [17].

Alla dashboard è possibile inserire più di una singola *Visualization* – grafico personalizzato – così da avere una vista più ampia della situazione. La comodità dell'utilizzo delle dashboard sta proprio nel legame che si crea tra i componenti che analizzano gli stessi dati, perché applicando dei filtri o più semplicemente interagendo con una delle visualizzazioni, tutte le informazioni presenti nell'interfaccia cambieranno di conseguenza.

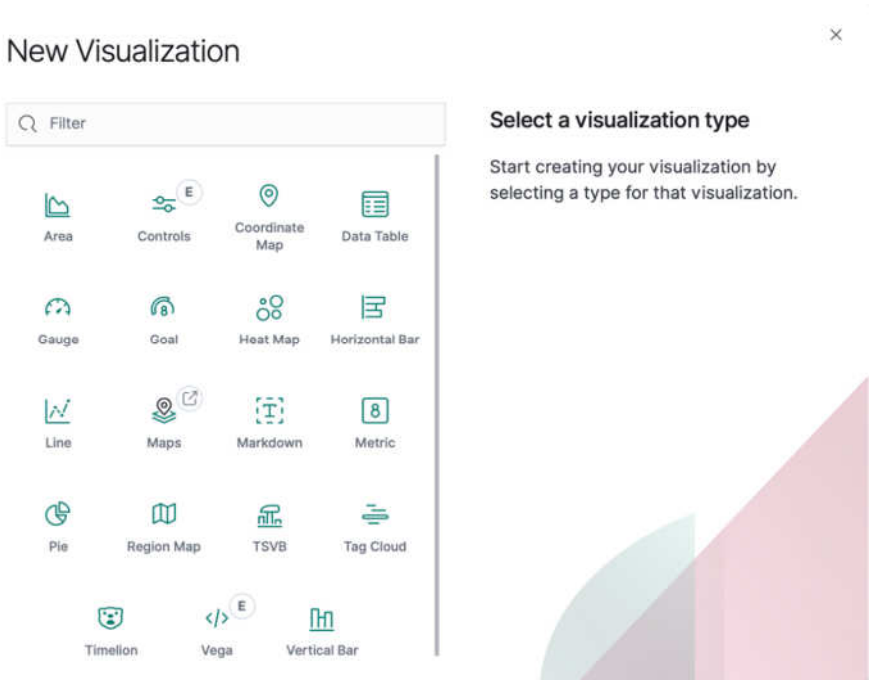


Figura 7 – Creazione di una nuova visualizzazione

Una volta selezionato il tipo di grafico tra quelli disponibili e l'indice sul quale operare, lo step successivo è quello di creare ed applicare filtri appositi sui dati. Si può ad esempio dividere il grafico selezionato in più grafici, a seconda del tipo di filtro inserito, o suddividere lo stesso grafico in più righe nel caso di una tabella, in più fette nel caso di un grafico a torta, e via dicendo.

Queste funzioni possono essere utilizzate selezionando il tutto da un menu a tendina, oppure scrivendo con l'apposito linguaggio per rendere il sistema di filtraggio più customizzabile (Apache Lucene) (Figura 8) [18].

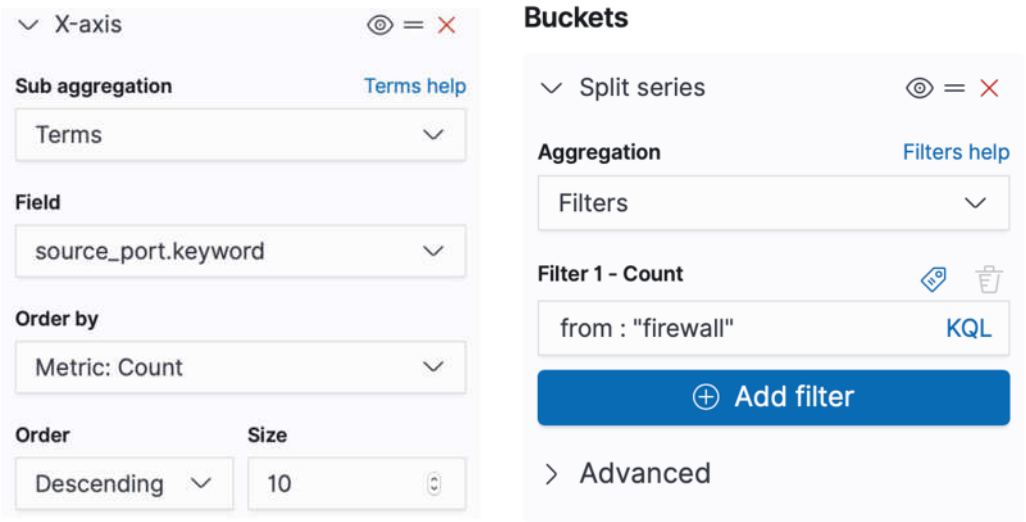


Figura 8 – Applicazione dei filtri nelle visualizzazioni

La visualizzazione, una volta salvata, verrà automaticamente aggiornata anche nella dashboard nella quale è contenuta.

Nella Figura 9, è possibile visionare un esempio di dashboard riguardante gli indirizzi e le porte di sorgente e destinazione provenienti dai log di uno dei firewall all'interno dell'università.

Figura 9 – Esempio dashboard

2.7 Elastic SIEM

Di recente, precisamente dalla versione 7.2.0, è stata introdotta una nuova funzionalità all'interno di *Kibana*, denominata *SIEM*, tuttora in fase beta.

SIEM è un acronimo che sta per Security Information and Event Management, e sempre più software sul mercato si stanno evolvendo in questa direzione.

A differenza di un semplice software di log management, attraverso un *SIEM* vi è la possibilità aggiuntiva di:

- unificare dati provenienti da sistemi diversi e verificare se sono presenti correlazioni tra di essi
- notificare l'amministratore di rete con dei messaggi di *alert* nel caso in cui ci siano attività sospette
- analizzare le eventuali minacce in maniera interattiva, attraverso una timeline dedicata (Figura 10)

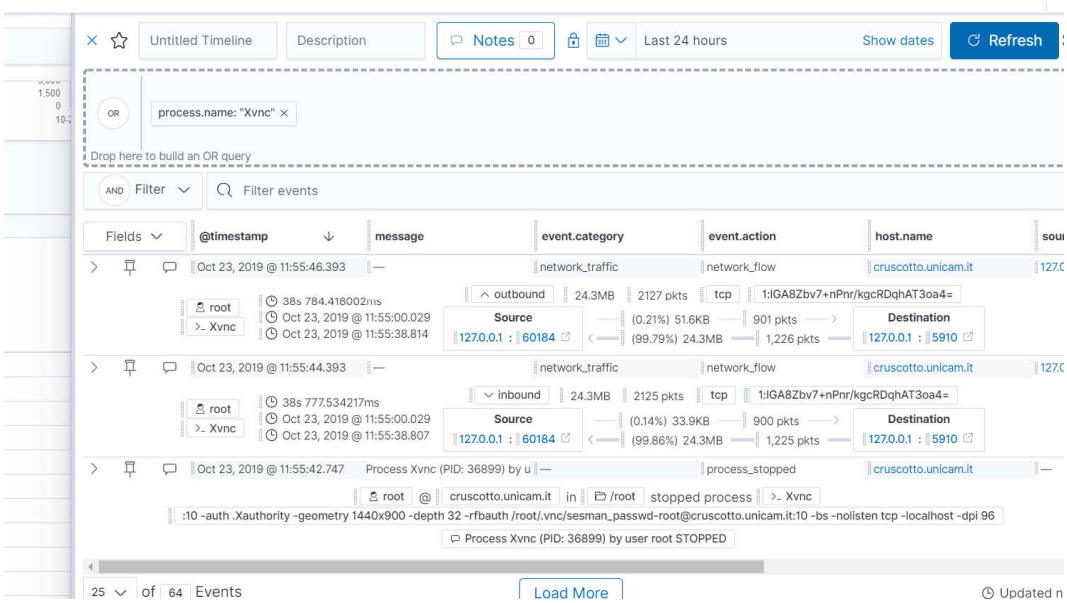


Figura 10 – Timeline interattiva SIEM

Per utilizzare questa nuova feature, occorre inviare i log tramite i *Beats* supportati (*Auditbeat* da Linux, *Winlogbeat* da Windows Server…), impostando l'output sia su *Elasticsearch* (non è utile inviarli su Logstash in quanto sono già di default filtrati e suddivisi in campi, dunque non c'è necessità di ulteriori parsing) sia su *Kibana*, in quanto devono essere inviate le dashboard preimpostate. Una volta effettuato ciò, si può disabilitare l'output su *Kibana* e lasciare soltanto quello relativo ad *Elasticsearch*.

Rispetto al concetto di dashboard, esplicato nel capitolo precedente, *SIEM* permette una maggiore interazione con i dati e una comprensione "automatica" della natura dei log.

Attualmente, le dashboard sono però uno strumento decisamente più stabile e ottimizzato, in quanto è possibile incorporarle in risorse web esterne e personalizzare i grafici in base alla preferenza dell'utente, cosa non possibile per la sezione *SIEM*, che rimane a disposizione solamente all'interno del sistema.

3. Analisi del database

L'utilizzo di un database non si limita alla conservazione e alla gestione di grandi quantità di dati, ma comprende anche tutte le operazioni di analisi sulle informazioni presenti. Quest'ultime possono essere collegate tra loro attraverso modelli logici differenti, scelti in base ad esigenze specifiche.

Le operazioni svolte mediante l'utilizzo di linguaggi *query*, utilizzati all'interno di software *DBMS* dedicati, sono denominate interrogazioni. Il loro scopo principale è fornire una vista più ristretta e specifica dei dati, filtrando le informazioni d'interesse e collegandole tra loro in grafici o tabelle.

3.1. Scelta del software

Per analizzare una grossa mole di dati come il database universitario, contenente tutti i dati, sensibili e non, relativi a: anagrafiche di studenti e docenti, esami sostenuti con votazioni positive o negative, immatricolazioni, corsi di studio… la scelta si è indirizzata verso il software *Pentaho* (Community Edition), che appoggiandosi su un server Apache Tomcat, è in grado di amministrare questi dati con differenti strumenti presenti al suo interno.

Inizialmente è stato provato lo stesso *ELK Stack* per gestire anche il database, creando un nuovo set di dati provenienti dal server. Successivamente paragonandolo a *Pentaho* è stato ritenuto più adeguato quest'ultimo come sistema, soprattutto per il fatto che le query sono più personalizzabili.

Riassumendo, *Pentaho* offre soluzioni certamente migliori e soprattutto più adatte per l'analisi di dati statici, mentre *ELK* è ottimizzato per analizzare dati in tempo reale, quindi adatto alla centralizzazione di flussi di dati continui.

Nello specifico su *Logstash* era stato inserito il plugin Jdbc [19] per consentire la connessione al database esterno. La seguente è la configurazione utilizzata.

```
input {
  jdbc {
    jdbc_driver_library => "/etc/mysqlconnector/mysql-
connector-java-8.0.16.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string =>
"jdbc:mysql://INDIRIZZO:3306/DATABASE"
    jdbc_user => "USERNAME"
    jdbc_password => "PASSWORD"
    statement => "QUERY"
  }
}

output {
    elasticsearch {
        index => "jdbc "
        hosts => ["serverelk.unicam.it:9200"]
    }
}
```

3.2. Installazione Pentaho

Pentaho Community Edition è disponibile gratuitamente sul sito SourceForge [20]. Una volta scaricato, è tutto già preimpostato e al suo interno è presente e settato il server sul quale si appoggia, *Apache Tomcat*. È stato dunque sufficiente scaricare il pacchetto (sulla seconda macchina virtuale) per poter iniziare a lavorare sulla piattaforma.

Per avviare il programma è sufficiente lanciare lo `startup.bat` presente dentro la cartella di Apache e poi accedere dal browser all'indirizzo *http://localhost:8080* [21]. Al primo avvio si accede con le credenziali di default (username: Admin e password: password), una volta all'interno del portale potranno poi essere modificate.

3.3. Installazione mySQL

Inizialmente la visualizzazione dei dati reali provenienti dal database accademico non era permessa. Per questo è stato importato il database in locale, popolato da dati fittizi generati in maniera randomica dal prof. *Francesco De Angelis*, con lo scopo di oscurare le informazioni sensibili. Soltanto dopo aver appurato il funzionamento del sistema, è stato possibile accedere al "vero" database, con la firma di una liberatoria concedente i permessi per visionarlo, per scopi di ricerca all'interno dell'università.

Dunque, è stato installato sulla macchina CentOS, mySQL Server, con il seguente comando:

```
yum install mysql-community-server
```

Una volta impostata la password di root con il comando

```
mysql -u root -p
```

si è proceduto con l'importazione del file dump SQL.

```
mysql -u root -p nomedatabase < file.sql
```

3.4. Creazione di una dashboard su Pentaho

Come visto precedentemente su *Kibana*, per andare a eseguire filtri e operazioni sui dati è necessario anche in questo caso creare una nuova dashboard personalizzata [22].

Il primo passo è quello di creare il layout della dashboard nel *Layout Panel*, ovvero indicare il posizionamento dei grafici, il numero di righe e colonne (Figura 11) ecc.

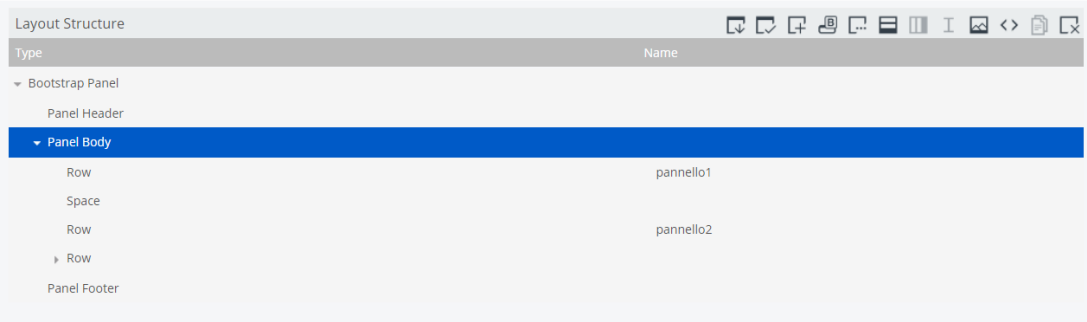


Figura 11 – Layout Panel

Per personalizzare ancora di più il tutto dal punto di vista grafico, è possibile inserire anche risorse esterne, come file .css e .js, per poi relazionarli agli altri elementi (Figura 12).

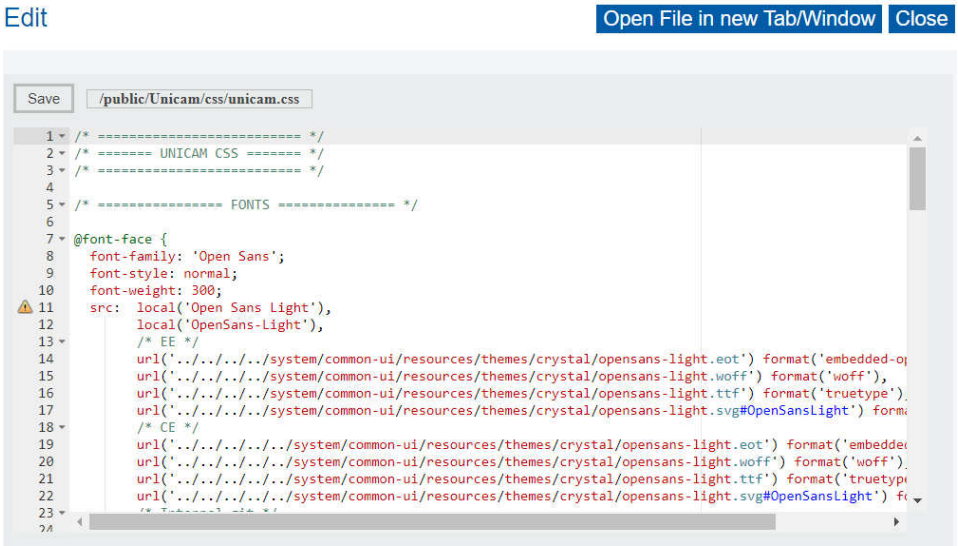


Figura 12 – Modifica del foglio di stile unicam.css

Per effettuate le query occorre spostarsi nel *Datasources Panel*, e ne va creata una per ogni grafico. Per configurare la connessione Jdbc verso il database bisogna includere il driver `com.mysql.jdbc.driver` nell'apposito campo, e assicurarsi che esso sia presente nella macchina. In caso negativo, va installato come da documentazione [23].

Nella Figura 13, vi è un esempio di configurazione per la connessione verso il database.

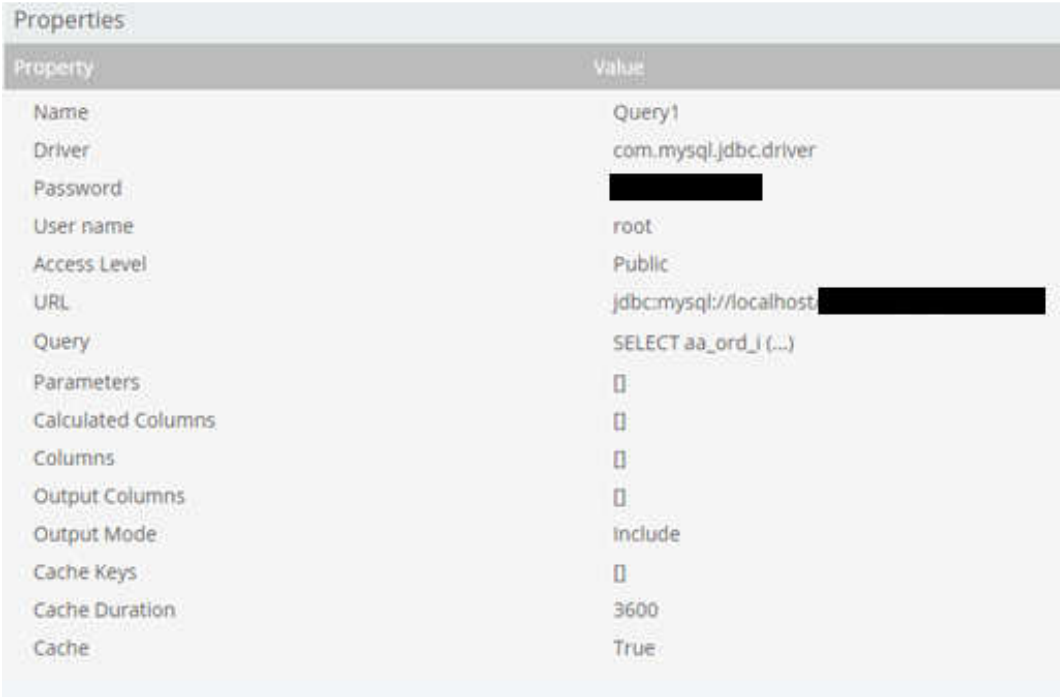


Figura 13 – Creazione nuova query

In SQL, in questa query di esempio vengono divisi i dati relativi agli esami mancanti per coorte di iscrizione.

```
SELECT
   CONCAT(vista.aa_iscr_id,'/',vista.aa_iscr_id+1) AS
   'Coorte di iscrizione',
   SUM(des = ${SelectorAD}) AS 'Studenti mancanti',
   ROUND(SUM(des != ${SelectorAD})/(COUNT(DISTINCT cod) -
1),1) AS 'Valore medio degli altri insegnamenti'
FROM
   database.esami AS esami,
   database.matricole AS vista
WHERE
   vista.aa_iscr_id < 2019
   AND esami.stu_id = vista.stu_id
   AND esami.cds_id = ${SelectorCDS}
   AND voto is null
   AND aa_sup_id is null
GROUP BY vista.aa_iscr_id
ORDER BY vista.aa_iscr_id DESC
LIMIT 5
```

L'ultimo step è quello di creare nel *Components Panel* (Figura 14) un oggetto grafico per ogni query, ovvero scegliere come mostrare i dati ottenuti (mediante grafici, tabelle o istogrammi).

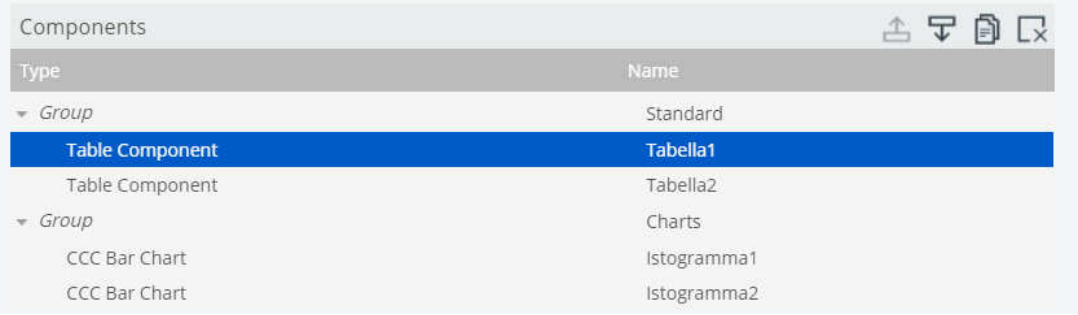


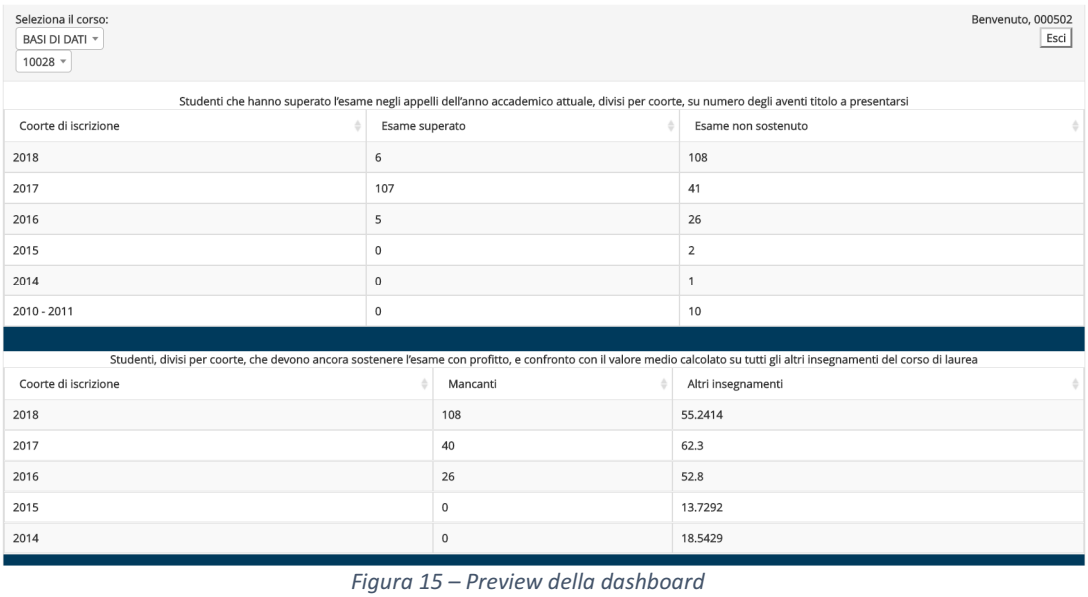
Figura 14 – Components Panel

Per utilizzare query parametrizzate che richiamano variabili esterne (ad esempio WHERE aa_freq_id=${annoScelto}), è da inserire il componente *Simple Parameter*, al quale è possibile dare un valore (2018, 2019…). Nella query interessata occorre quindi inserire nel campo *Parameters*, il nome del parametro inserito.

Se si volesse effettuare il tutto in maniera dinamica, in modo da far poter cambiare questo parametro all'utente, allora in questo caso c'è da aggiungere un terzo elemento, come ad esempio un menu a tendina, *Selector Component*. All'interno di quest'ultimo va inserito nei campi *Listeners* e *Parameters* il nome del parametro, e nella query questa volta va messo direttamente il nome del Selector e non del parametro, in quanto il Selector fa da tramite.

Inoltre, anche nei grafici che utilizzano la query, va inserito il nome del selettore su Listeners e Parameters, in modo tale che variano non appena c'è un evento modificatore sul menu da parte dell'utente.

Per verificare la corretta impostazione della dashboard, passo dopo passo, è conveniente sfruttare il bottone in alto a destra, che permette di visualizzare una preview aggiornata del lavoro svolto (Figura 15).


Seleziona il corso:
BASI DI DATI
10028
Benvenuto, 000502
Esci

Studenti che hanno superato l'esame negli appelli dell'anno accademico attuale, divisi per coorte, su numero degli aventi titolo a presentarsi

Coorte di iscrizione	Esame superato	Esame non sostenuto
2018	6	108
2017	107	41
2016	5	26
2015	0	2
2014	0	1
2010 - 2011	0	10

Studenti, divisi per coorte, che devono ancora sostenere l'esame con profitto, e confronto con il valore medio calcolato su tutti gli altri insegnamenti del corso di laurea

Coorte di iscrizione	Mancanti	Altri insegnamenti
2018	108	55.2414
2017	40	62.3
2016	26	52.8
2015	0	13.7292
2014	0	18.5429

Figura 15 – Preview della dashboard

4. Creazione applicazioni web

Le due macchine virtuali fino ad ora create come soluzioni temporanee, sono in seguito state spostate su due server definitivi gestiti da Unicam, uno per la piattaforma *ELK Stack* ed uno per *Pentaho*, rispettivamente agli indirizzi:

http://serverelk.unicam.it

http://cruscotto.unicam.it

Su gli stessi sono state create due differenti pagine web con lo scopo di rendere disponibili le dashboard create a docenti o dipendenti dell'Università di Camerino.

4.1. Installazione Nginx

Kibana e *Pentaho* operano rispettivamente sulle porte 5601 e 8080. Chiunque conosce l'indirizzo IP della macchina sulla quale questi servizi sono in esecuzione, possono di conseguenza accedervi. In un ambiente professionale, ciò può portare a gravi problemi di sicurezza.

Per risolvere la questione è stato usato come reverse proxy su entrambe le macchine virtuali *Nginx*, attraverso la procedura di installazione standard:

```
yum install nginx
yum start nginx
yum enable nginx
```

Per configurare correttamente Nginx così da poter visualizzare il portale web, occorre creare un file di configurazione all'interno del percorso `/etc/nginx/conf.d`. Le impostazioni iniziali sono le seguenti [24]:

```
server {
    listen 80;

    server_name INDIRIZZO IP;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.users;

    location / {
        proxy_pass http://localhost:PORTA;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Il *server_name* servirà da indirizzo di ricerca all'interno de browser, la porta sulla quale verranno inoltrate le richieste sarà la 80 (se il protocollo è http) e le credenziali di accesso verranno prese dal file che segue il campo *auth_basic_user_file*.

Per impostare quest'ultime il comando è il seguente:

```
echo "admin:`openssl passwd -apr1`" | sudo tee -a
/etc/nginx/htpasswd.users
```

Il prompt dei comandi chiederà di inserire la password relativa al nome dell'user specificato (admin) e di riconfermarla.

Sono applicabili differenti *location* in modo da configurare il sito nel modo più adeguate alle esigenze: la principale è la /, all'interno della quale sono inseriti il percorso della cartella contenente i file e l'*index.html* per l'homepage del portale (in questo caso /var/www).

4.2. Struttura delle piattaforme

Entrambe le piattaforme hanno la medesima strutturazione con linguaggio HTML e lo stesso linguaggio di programmazione (Javascript), differendo soltanto nel contenuto.

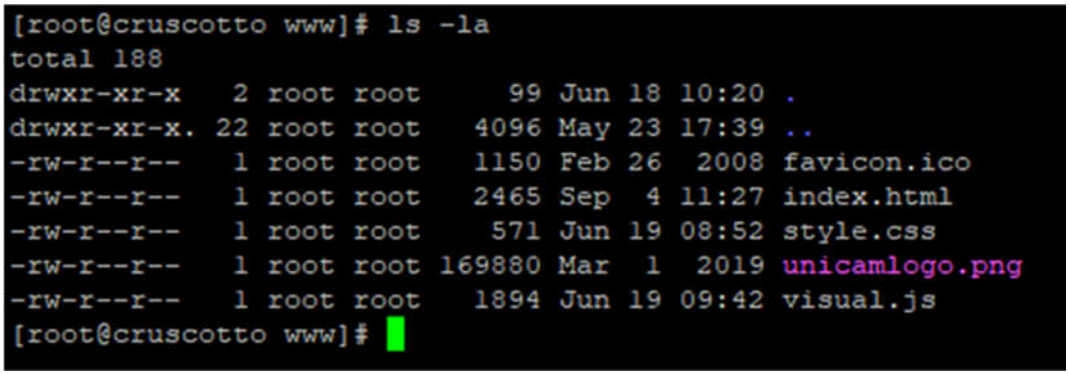


Figura 16 – Struttura della directory dell'applicazione web

Come visionabile nella Figura 16 è presente un *index.html*, ovvero la pagina principale, lo *style.css* in cui sono racchiuse le proprietà grafiche, e il file *visual.js* che gestisce le funzionalità. Nello stesso percorso possono essere inoltre inseriti anche file contenenti immagini e altre risorse.

Per impostare la favicon del sito (l'icona presente nella barra superiore del browser), è sufficiente posizionarla nello stesso path dell'index.

Il tutto è reso responsive grazie all'utilizzo del framework *Bootstrap*, che rende il sistema ottimizzato per tutti i tipi di dispositivi aventi differente risoluzione (da ambienti desktop a mobile).

4.3. Esportazione dashboard Kibana

Esportare una dashboard di *Kibana* verso un sistema esterno è un'operazione davvero semplice, resa possibile dal tasto Share all'interno della sezione. Attraverso questa funzione potranno essere generati degli Embed Code relativi alla dashboard da esportare. Sono disponibili due scelte (Figura 17):

- *Saved Object*, per avere un iframe auto-aggiornante in caso di modifiche frequenti alla dashboard, senza dover aggiornare ogni qual volta vi siano cambiamenti il link relativo alla pagina
- *Snapshot*, una semplice vista della dashboard al momento dell'esportazione, insensibile ai cambiamenti strutturali dei componenti al suo interno

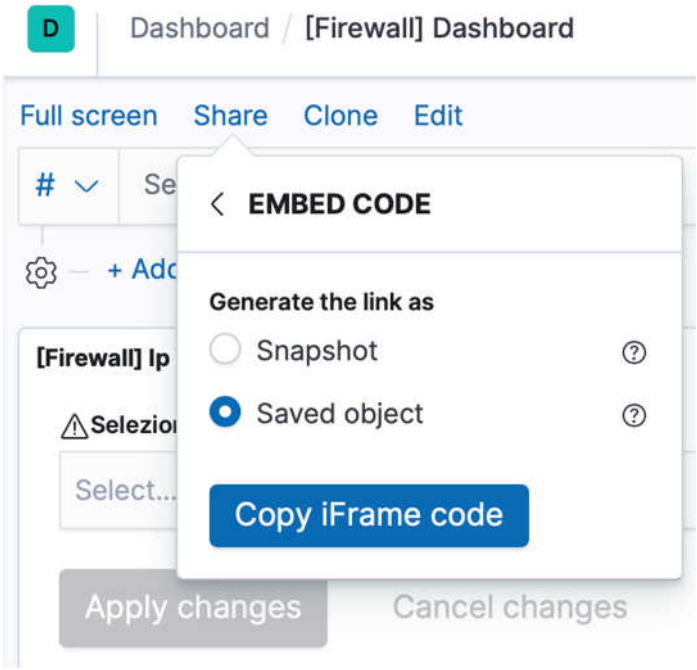


Figura 17 – Funzione di esportazione della dashboard

4.4. Esportazione dashboard Pentaho

Per esportare invece una dashboard di *Pentaho*, è sufficiente completare il seguente indirizzo con il percorso dove essa è stata salvata più eventuali parametri aggiuntivi [25].

```
http://localhost:8080/pentaho/api/repos/<path>/generatedContent?<parameters>
```

A differenza di quelle realizzabili con *Kibana*, le dashboard create all'interno del sistema *Pentaho* sono maggiormente personalizzabili modificando lo stile di tabelle, grafici o addirittura creare intere pagine con menu e funzioni.

4.5. Incorporamento delle dashboard

Le dashboard in precedenza esportate vanno poi inserite all'interno del file html, nel quale è sufficiente posizionare il codice all'interno del blocco desiderato.

```
<div class="col-sm-12">
   <iframe src="CODICE ESPORTATO" height="600" width="800">
   </iframe>
</div>
```

4.6. Autenticazione Kibana

Per quanto riguarda la versione gratuita di *Kibana*, non è incluso un sistema di autenticazione e gestione degli utenti, che è invece disponibile con le soluzioni a pagamento oppure attivando le features riguardanti la sicurezza della piattaforma, con la prova gratuita di 30 giorni (Figura 18).

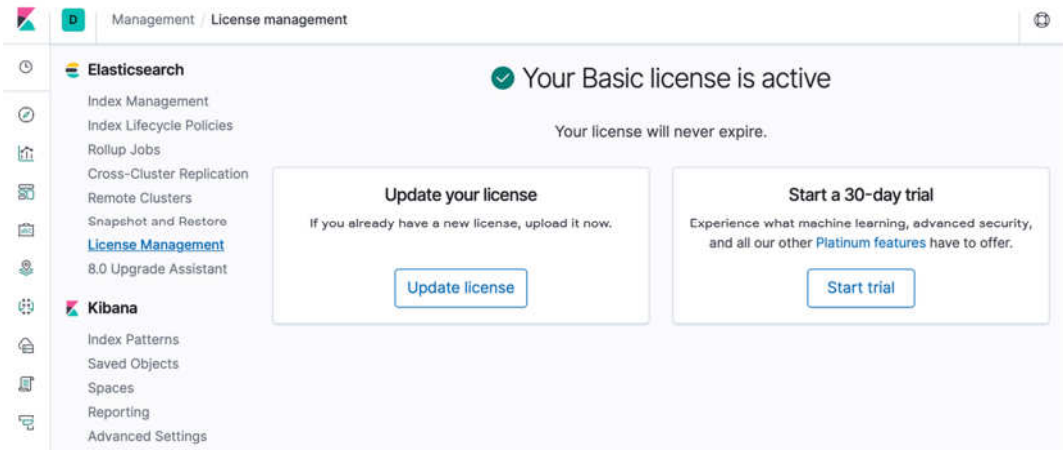


Figura 18 – Gestione delle license

Una volta attivata, compariranno sulla sezione Management / Security, i menù Users and Roles, dal quale poi si possono aggiungere utenti e ruoli ad essi associati (Figura 19). Ogni utente avrà così permessi specifici all'interno del sistema, potendo creare e modificare dashboard o semplicemente visionare le stesse.

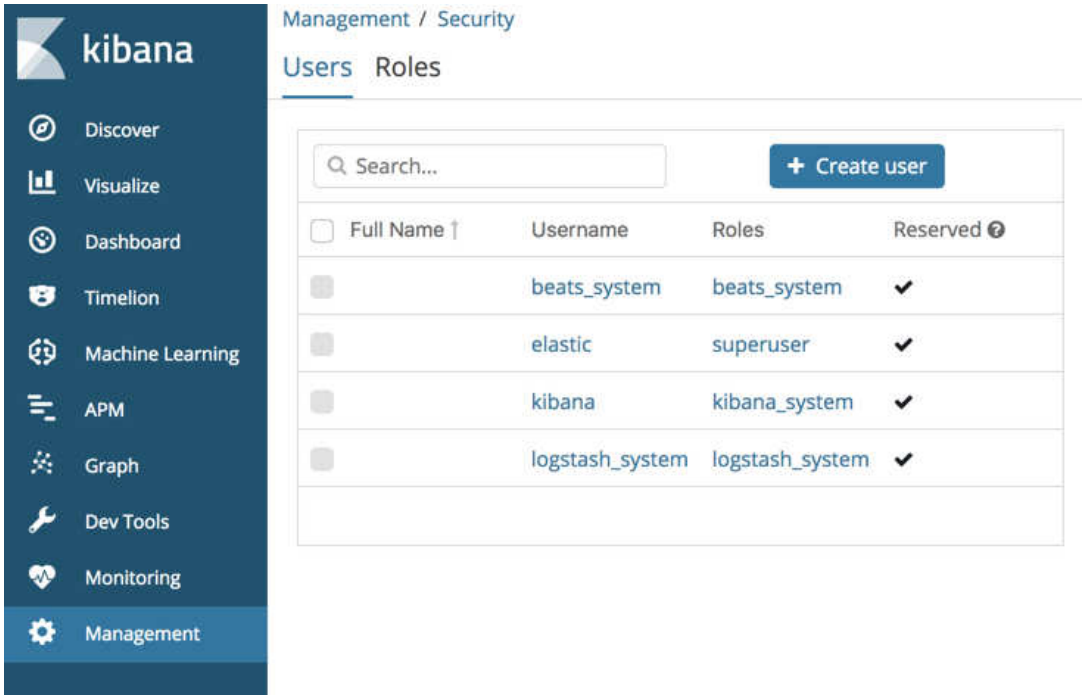


Figura 19 – Sezione Security di Kibana

Per accedere in maniera sicura alla piattaforma senza eseguire le precedenti operazioni, è possibile utilizzare alternativamente l'autenticazione di *Nginx*, sul percorso su cui opera il software (vedi capitolo "Installazione Nginx").

Occorre impostare un path alternativo al server tramite proxy, modificando la configurazione di *Kibana* come segue [26]:

```
server.port : 8888
server.host: 127.0.0.1
server.basePath: "/kibana"
```

Con questa soluzione il server è raggiungibile solamente da locale attraverso la porta impostata, oppure da remoto mediante il percorso proxy specificato nel *server.basePath*.

Riconfigurando ora *Nginx* con un nuovo percorso disponibile che reinderizzi al server in locale, sarà possibile accedere con credenziali anche all'interno del server *Kibana*.

```
upstream kibana {
    server 127.0.0.1:8888;
    keepalive 4096;
}

server {
    ...

    location /kibana/ {
        proxy_pass http://kibana/;
        proxy_redirect off;
        proxy_buffering off;
        proxy_http_version 1.1;
        proxy_set_header Connection "Keep-Alive
        proxy_set_header Proxy-Connection "Keep-Alive";
    }
}
```

4.7. Autenticazione Pentaho

Pentaho invece, è provvisto di un sistema di autenticazione e gestione degli utenti anche nella versione gratuita, ed è possibile impostare permessi differenti sia per i semplici utilizzatori del portale sia per gli amministratori (Figura 20).

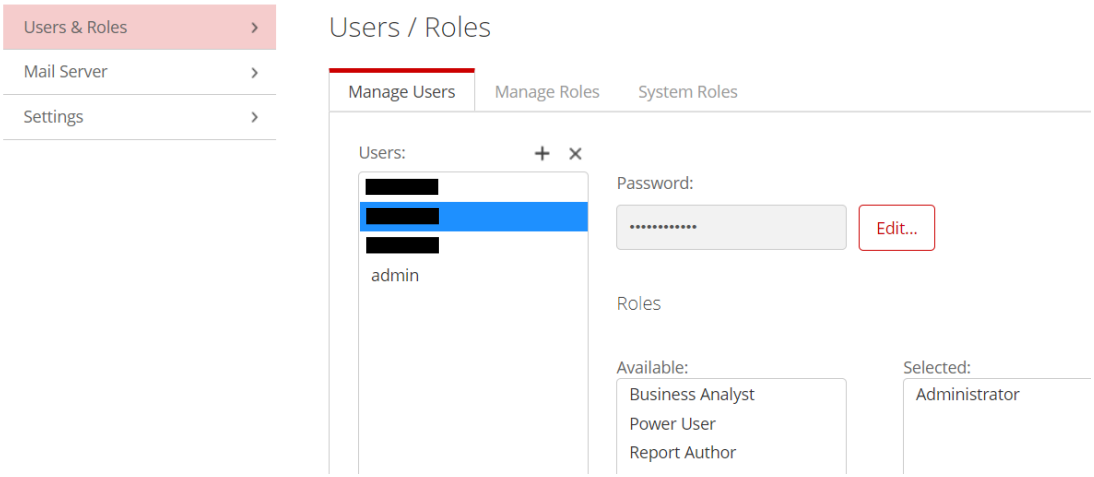


Figura 20 – Users e Roles su Pentaho

Permette inoltre di aggiungere un sistema di autenticazione personalizzato tramite LDAP (Lightweight Directory Access Protocol), protocollo per la gestione centralizzata delle utenze, con l'opzione di importare utenti già usati in altre piattaforme (Figura 21). L'interfaccia grafica non è presente nella versione gratuita, perciò occorre eseguire le operazione da terminale riportate nella guida [27].

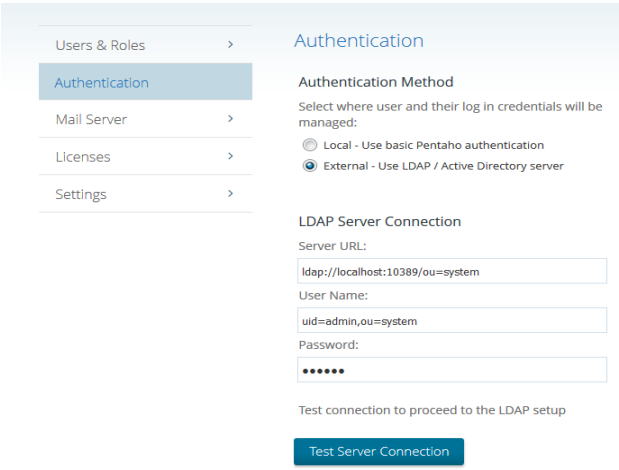


Figura 21 – Sezione Authentication di Pentaho

5. Confronto sistemi di analisi

La differenza fra le due sorgenti informative presentate, ha permesso un confronto bilaterale dei due sistemi distinti presi in analisi, che ha poi determinato l'utilizzo di entrambi per gli scopi differenti.

5.1. Tempi e modalità di installazione

Per quanto concerne l'installazione e la configurazione dei due sistemi, i tempi e le modalità sono alquanto differenti.

L'intera architettura *ELK* richiede un'installazione completamente manuale da riga di comando, ed è una fase piuttosto delicata, in quanto nel momento in cui uno solo dei tre componenti (*Elastic, Logstash o Kibana*) non funziona correttamente, il sistema risulta corrotto e non in grado di ricevere e operare sui dati.

Pentaho invece, con la sua versione Community, è disponibile come già citato nell'apposito capitolo, in free download, ed è tutto pre-configurato e adatto anche per utenti poco esperti.

5.2. Differenza tra le dashboard

La caratteristica che li accomuna più fortemente, oltre ad essere parte centrale dello studio di questi due sistemi, è la possibilità di creare, partendo dai dati ricevuti, grafici e dashboard personalizzabili con lo scopo di visualizzare in modo interattivo concetti di interesse per l'utilizzatore. Le informazioni restituite sono frutto di un'analisi approfondita e specifica all'interno del sistema, che permette di filtrare e correlare elementi della stessa natura.

Le dashboard di *Kibana* sono ad un primo impatto più semplici e intuitive da configurare. D'altra parte, *Pentaho*, con la sua dashboard CDE, inizialmente è più complesso da inizializzare, ma permette una personalizzazione maggiore, con la possibilità di incorporare non solo la stessa dashboard, ma costruirci attorno un vero e proprio sito web.

5.3. Difficoltà riscontrate

Durante l'aggiornamento software dell'*ELK Stack* dalla versione 6.6.0 alla 7.4.0, sono stati riscontrati alcuni problemi. È stato effettuato infatti un upgrade diretto e questo ha corrotto il cluster di *Elasticsearch*, non riuscendo a far connettere quest'ultimo con *Kibana* (`Kibana server is not ready`).

In seguito, è stato fatto un downgrade ritornando alla versione precedente, e poi aggiornato dalla 6.6.0 all'ultima major release del pacchetto 6.x. Dopo questo passaggio, è stato nuovamente lanciato l'update alla 7.4.0. Con questa modalità di aggiornamento il sistema ha ripreso a funzionare correttamente ma sono stati persi i dati di *Kibana*, quali le visualizzazioni e le dashboard salvate. Molti utenti lamentano sulle community online questo genere di problematiche, e solo una piccola parte è riuscito a risolvere la questione, attraverso dei backup, ma non è garantito che ciò funzioni.

Per quanto concerne invece *Pentaho*, va segnalato un bug nel sistema di autenticazione principale. Nel caso in cui vengono inseriti username e password infatti, e si clicca sul pulsante di Login non attraverso il mouse ma attraverso la pressione del tasto Enter (dalla keyboard), è impossibile accedere al sistema e la pagina viene ricaricata, resettando i campi.

5.4. Conclusioni

Dopo averne studiato il funzionamento, e avendo presentato in maniera approfondita entrambi i software, è possibile constatare che sono soluzioni molto valide nel mondo business.

Se si vuole tuttavia avere un'esperienza utente completa, per quanto riguarda *ELK* è consigliata la soluzione a pagamento che comprende X-Pack, una singola estensione che aggiunge funzionalità importanti come quelle di alerting, monitoring, reporting e Machine Learning. Dall'altra parte, la versione free di *Pentaho* permette di lavorare senza troppi problemi con il software, non obbligando l'utente a fare l'upgrade alla versione Enterprise: sono presenti solo alcune piccole limitazioni (certe impostazioni non possono essere impostate tramite un'interfaccia ma soltanto da riga di comando) e sono assenti alcuni plugin per personalizzazioni aggiuntive, mancanze comunque non significative come quelle di *Elasticsearch*.

Concludendo, i due sistemi possono essere utilizzati separatamente o in maniera complementare. Ad esempio, nel caso di una grande azienda, è raccomandato utilizzare *ELK* per analizzare log di sistema in tempo reale e monitorare le attività della rete, mentre è opportuno indirizzarsi verso *Pentaho* per visionare statistiche sui database interni, con lo scopo di effettuare valutazioni utili in termini economici e finanziari.

6. Sitografia

[1] «Syslog,» in *https://en.wikipedia.org/wiki/Syslog*.

[2] «DBMS differences,» in *https://db-engines.com/en/system/MariaDB%3BMySQL%3BPostgreSQL*.

[3] «Logstash Alternatives,» in *https://sematext.com/blog/logstash-alternatives/*.

[4] «ELK Stack Installation,» in *https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-centos-7*.

[5] «Logstash Configuration,» in *https://www.elastic.co/guide/en/logstash/6.6/config-examples.html*.

[6] «Grok Debugger,» in *http://grokdebug.herokuapp.com/*.

[7] «Grok Filter,» in *https://www.elastic.co/guide/en/logstash/6.6/plugins-filters-grok.html*.

[8] «PfSense Grok Patterns,» in *http://pfelk.3ilson.com/2017/10/pfsense-v24xkibanaelasticsearchlogstash.html?fbclid=IwAR1_S197pNDY5iuqYeJdkrnYvtmpLcY7g2rASwFekjKAYWrb98Hv8rDeVWo & https://github.com/patrickjennings/logstash-pfsense/blob/master/patterns/pfsense2-4.grok*.

[9] «Syslog Grok Patterns,» in *https://stackoverflow.com/questions/48526210/syslog-to-logstash-grokparsefailure & https://medium.com/@oguzhan00/configure-logstash-with-syslog-and-grok-for-geopoint-logging-to-elasticsearch-bd8365d114f3*.

[10] «Cisco Grok Patterns,» in *https://gist.github.com/clay584/5a75009ad571af3d0648*.

[11] «Nginx Grok Patterns,» in *https://gist.github.com/kmassada/e7f472a8d4ae5884d321*.

[12] «Mutate Filter,» in *https://www.elastic.co/guide/en/logstash/6.6/plugins-filters-mutate.html*.

[13] «Filebeat Installation,» in *https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-installation.html*.

[14] «Winlogbeat Installation,» in *https://www.elastic.co/guide/en/beats/winlogbeat/current/winlogbeat-installation.html*.

[15] «Auditbeat Installation,» in *https://www.elastic.co/guide/en/beats/auditbeat/current/auditbeat-installation.html*.

[16] «Auditbeat dashboards,» in *https://www.elastic.co/guide/en/beats/auditbeat/current/setup-kibana-endpoint.html*.

[17] «Dashboard Kibana,» in *https://www.elastic.co/guide/en/kibana/6.6/dashboard-getting-started.html*.

[18] «Kibana Language,» in *https://www.elastic.co/guide/en/kibana/current/search.html*.

[19] «Jdbc Plugin,» in *https://www.elastic.co/guide/en/logstash/6.6/plugins-inputs-jdbc.html*.

[20] «Download Pentaho,» in *https://sourceforge.net/projects/pentaho/*.

[21] «Start Pentaho,» in *https://help.pentaho.com/Documentation/7.1/Installation/Manual/070_Starting_pentaho_server*.

[22] «Create Dashboard Pentaho,» in *https://help.pentaho.com/Documentation/7.0/0R0/CTools/CDE_Dashboard_Overview*.

[23] «Jdbc Driver Pentaho,» in *https://help.pentaho.com/Documentation/8.1/Setup/JDBC_Drivers_Reference*.

[24] «Nginx Setup,» in *http://nginx.org/en/docs/beginners_guide.html*.

[25] «Embed Pentaho Dashboard,» in *https://help.pentaho.com/Documentation/8.1/Developer_Center/Embed_Pentaho_Server*.

[26] «Kibana proxy,» in *https://discuss.elastic.co/t/nginx-reverse-proxy-setup-for-kibana/167327*.

[27] «LDAP Pentaho,» in *https://help.pentaho.com/Documentation/8.2/Setup/Administration/User_Security/LDAP*.