ELK Beats

Istruzioni di Setup

Ewan Borsa – I3BB

CPT Trevano  - SAM Informatica

Sommario

[Setup rete 2](#_Toc124752876)

[Proxy gateway 2](#_Toc124752877)

[Setup proxy Ubuntu 2](#_Toc124752878)

[Per connessioni generiche 2](#_Toc124752879)

[Proxy per apt 3](#_Toc124752880)

[Setup ELK 4](#_Toc124752881)

[Setup Beats 5](#_Toc124752882)

[ELK server 5](#_Toc124752883)

[Metricbeat 5](#_Toc124752884)

[Windows DC 6](#_Toc124752885)

[Metricbeat 6](#_Toc124752886)

[Ubuntu OwnCloud 7](#_Toc124752887)

[Configurazione Kibana 8](#_Toc124752888)

[Setup Kibana 8](#_Toc124752889)

[Configurazione Metricbeat 10](#_Toc124752890)

[Setup metricbeat 10](#_Toc124752891)

[SSL 10](#_Toc124752892)

[API Key 10](#_Toc124752893)

[Chiave per "Beats setup" 11](#_Toc124752894)

[Aggiornare il file di configurazione di metricbeat 11](#_Toc124752895)

[Test della configurazione 12](#_Toc124752896)

[Setup indici e dashboard 12](#_Toc124752897)

[Chiave per "Publisher" 13](#_Toc124752898)

# Setup rete

## Proxy Gateway

Per l’autenticazione usiamo un proxy aggiuntivo che si interpone tra l'host e il proxy cpt "iniettando" l'autenticazione ntlm (credenziali di dominio) in ogni richiesta.  
  
Il proxy px è da avviare usando il file px.bat e la configurazione fa in modo che risponda alla porta 0.0.0.0:5865.  
Le VM in NAT su interfaccia di rete 1 possono usare il server DHCP incorporato in VirtualBox che si identifica in maniera trasparente con l'ip della macchina host.  
DHCP/GW NAT 10.0.2.2 = ip host

## Setup proxy Ubuntu

### Per connessioni generiche

test del proxy => impostare le variabili d'ambiente (temporanee)

export http\_proxy="http://10.0.2.2:5865"

export https\_proxy="http://10.0.2.2:5865"

export HTTP\_PROXY="http://10.0.2.2:5865"

export HTTPS\_PROXY="http://10.0.2.2:5865"

export no\_proxy=localhost,127.0.0.1

test connessione

curl https://www.google.com

deve scaricare l'html della pagina indicata

Se funziona impostare le variabili in maniera permanente. Questo può essere fatto in ogni profilo utente oppure per tutta la VM.

sudo nano /etc/environment

http\_proxy="http://10.0.2.2:5865"

https\_proxy="http://10.0.2.2:5865"

HTTP\_PROXY="http://10.0.2.2:5865"

HTTPS\_PROXY="http://10.0.2.2:5865"

no\_proxy=localhost,127.0.0.1

sudo reboot

Testare nuovamente.

### Proxy per apt

Creare un file di configurazione in /etc/apt/apt.conf.d (caricamento sequenziale dei file di configurazione)

sudo nano /etc/apt/apt.conf.d/proxy.conf

Acquire::http::Proxy "http://10.0.2.2:5865";

Acquire::https::Proxy "http://10.0.2.2:5865";

Test con

sudo apt update

Variabile no\_proxy per intnet

no\_proxy=localhost,127.0.0.1,10.20.0.\*

# Installazione ELK

È possibile installare ELK in vari modi, tutti elencati in questa guida:  
<https://www.elastic.co/guide/en/elasticsearch/reference/current/install-elasticsearch.html>

Noi usando un sistema Ubuntu andremo ad installarlo usando la distribuzione tramite repository:  
<https://www.elastic.co/guide/en/elasticsearch/reference/current/deb.html>

## Importare la chiave PGP di Elasticsearch

Firmiamo tutti i nostri pacchetti con la chiave di firma Elasticsearch (chiave PGP D88E42B4, Disponibile dal https://pgp.mit.edu) con impronta digitale:

4609 5ACC 8548 582C 1A26 99A9 D27D 666C D88E 42B4

Scarica e installa la chiave di firma pubblica:

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg

## Avvia Elasticsearch con la sicurezza abilitata

Quando si installa Elasticsearch, le funzionalità di sicurezza sono abilitate e configurate per impostazione predefinita. Quando installi Elasticsearch, la seguente configurazione di sicurezza si verifica automaticamente:

L'autenticazione e l'autorizzazione sono abilitate e viene generata una password per Superuser integrato. elastic

I certificati e le chiavi per TLS vengono generati per il livello di trasporto e HTTP, e TLS è abilitato e configurato con queste chiavi e certificati.

La password, il certificato e le chiavi vengono inviati al terminale. Per esempio:

## Esecuzione di Elasticsearch con systemd

Per configurare Elasticsearch in modo che si avvii automaticamente all'avvio del sistema, Eseguire i seguenti comandi:

sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable elasticsearch.service

Elasticsearch può essere avviato e arrestato come segue:

sudo systemctl start elasticsearch.service

sudo systemctl stop elasticsearch.service

Questi comandi non forniscono alcun feedback sull'avvio di Elasticsearch con successo o meno. Invece, queste informazioni verranno scritte nel registro file che si trovano in ./var/log/elasticsearch/

Se hai protetto con password il tuo keystore Elasticsearch, dovrai fornire la password del keystore utilizzando un file locale e un ambiente systemd Variabili. Questo file locale deve essere protetto finché esiste e può essere eliminato in modo sicuro una volta che Elasticsearch è attivo e funzionante. systemd

echo "keystore\_password" > /path/to/my\_pwd\_file.tmp

chmod 600 /path/to/my\_pwd\_file.tmp

sudo systemctl set-environment ES\_KEYSTORE\_PASSPHRASE\_FILE=/path/to/my\_pwd\_file.tmp

sudo systemctl start elasticsearch.service

Per impostazione predefinita, il servizio Elasticsearch non registra le informazioni nel journal. Per abilitare la registrazione, l'opzione deve essere rimossa dalla riga di comando nel file.  
systemd journalctl --quietExecStartelasticsearch.service

Quando la registrazione è abilitata, le informazioni di registrazione sono disponibili utilizzando I comandi:  
systemd journalctl

Per completare il diario:  
sudo journalctl -f

Per elencare le voci del journal per il servizio elasticsearch:  
sudo journalctl --unit elasticsearch

Per elencare le voci del journal per il servizio elasticsearch a partire da un determinato momento:  
sudo journalctl --unit elasticsearch --since "2016-10-30 18:17:16"

## Verifica che Elasticsearch sia in esecuzione

Puoi verificare che il nodo Elasticsearch sia in esecuzione inviando una richiesta HTTPS per il porting.

curl --cacert /etc/elasticsearch/certs/http\_ca.crt -u elastic https://localhost:9200

Assicurarsi di utilizzare nella chiamata, altrimenti la richiesta avrà esito negativo.

--cacert

Percorso del certificato generato per il livello HTTP(http\_ca.crt).

Immettere la password per l'utente generata durante, che dovrebbe restituire una risposta simile alla seguente:

{

"name" : "Cp8oag6",

"cluster\_name" : "elasticsearch",

"cluster\_uuid" : "AT69\_T\_DTp-1qgIJlatQqA",

"version" : {

"number" : "8.6.0",

"build\_type" : "tar",

"build\_hash" : "f27399d",

"build\_flavor" : "default",

"build\_date" : "2016-03-30T09:51:41.449Z",

"build\_snapshot" : false,

"lucene\_version" : "9.4.2",

"minimum\_wire\_compatibility\_version" : "1.2.3",

"minimum\_index\_compatibility\_version" : "1.2.3"

},

"tagline" : "You Know, for Search"

}

## Connettere i client a Elasticsearch

Quando avvii Elasticsearch per la prima volta, TLS viene configurato automaticamente per il Livello HTTP. Un certificato CA viene generato e archiviato su disco in:  
/etc/elasticsearch/certs/http\_ca.crt

L'impronta digitale SHA-256 con codifica esadecimale di questo Il certificato viene anche inviato al terminale. Tutti i client che si connettono a Elasticsearch, come i client Elasticsearch, Beats, agenti elastici autonomi e Logstash devono verificare di considerare attendibile il certificato utilizzato da Elasticsearch per HTTPS. Server della flotta e gestione della flotta Gli Elastic Agent vengono configurati automaticamente per considerare attendibile il certificato CA. Altri client possono stabilire l'attendibilità utilizzando l'impronta digitale della CA o il certificato CA stesso.

Se il processo di configurazione automatica è già stato completato, è comunque possibile ottenere l’impronta digitale del certificato di sicurezza. È inoltre possibile copiare il certificato CA alla tua macchina e configura il tuo client per usarlo.

### Utilizzare l'impronta digitale CA

Copia il valore dell'impronta digitale generato sul terminale all'avvio di Elasticsearch e Configura il tuo client per utilizzare questa impronta digitale per stabilire l'attendibilità quando si connette a Elasticsearch.

Se il processo di configurazione automatica è già stato completato, è comunque possibile ottenere l’impronta digitale del certificato di protezione eseguendo il comando riportato di seguito. Le path corrispondono al certificato CA generato automaticamente per il livello HTTP.  
openssl x509 -fingerprint -sha256 -in config/certs/http\_ca.crt

Il comando restituisce il certificato di protezione, inclusa l'impronta digitale. Il dovrebbe essere .issuerElasticsearch security auto-configuration HTTP CA

issuer= /CN=Elasticsearch security auto-configuration HTTP CA

SHA256 Fingerprint=<fingerprint>

### Utilizzare il certificato CA

Se la libreria non supporta un metodo di convalida dell'impronta digitale, il parametro Il certificato CA generato automaticamente viene creato nella seguente directory su ogni Elasticsearch nodo:  
/etc/elasticsearch/certs/http\_ca.crt

Copiare il file sul computer e configurare il client per l'utilizzo di questo per stabilire l'attendibilità quando si connette a Elasticsearch.http\_ca.crt

## Configurazioni aggiuntive

sudo nano /etc/elasticsearch/elasticsearch.yml  
cercare e scommentare la linea con la variabile "host" e ip 0.0.0.0  
in questo modo Elastic aprirà la porta di ascolto su tutti gli ip della macchina

# Configurazione ELK

## Setup Proxy

modificare la configurazione delle variabili d'ambiente del proxy in modo da escludere il server elk

cambiare

no\_proxy=localhost,127.0.0.1

in

no\_proxy=localhost,127.0.0.1,10.20.0.50

## Configurazione del demone elasticsearch

Al momento il servizio di elasticsearch risponde solo all'indirizzo localhost:9200 perciò risulta impossibile accedere in lettura o scrittura da altri server, cosa che ci è indispensabile per poter inviare dati di monitoraggio da altre macchine.

È quindi necessario modificare la configurazione del demone.

Aprire il file

/etc/elasticsearch/elasticsearch.yml

Cercare la sezione Network e impostare l'ip di ascolto a 0.0.0.0

network.host: 0.0.0.0

Se non ancora fatto impostare il demone, abilitare l'avvio con il sistema e riavviarlo

systemctl daemon-reload  
systemctl enable elasticsearch.service  
systemctl restart elasticsearch.service

A questo punto sarà possibile accedere all'API dal browser del proprio pc usando l'indirizzo sull'interfaccia host only e fornendo le credenziali (elastic:password\_generata)

[https://192.168.56.50:9200](https://192.168.56.50:9200/)

### Uso dell'API con tool esterni

Per lavorare al meglio con l'API consiglio di installare l'estensione firefox RESTClient o usare Insomnia in modo da poter sottomettere richieste POST e formattare i JSON prodotti.

Lavorando con l'API, invece di inserire ogni volta utente e password, è possibile impostare un header di authorization, in questo modo.

1. Generare una stringa base64 contenente utente e password

echo -n 'elastic:PASSWORD' | base64

Da notare il ":" tra elastic e la password, ma senza mettere spazi o altri caratteri

1. Aggiungere un header di questo tipo

'Authorization: Basic CREDENZIALI\_B64'

esempio

'Authorization: Basic ZWxhc3RpYzpobWxhY3JsODI4MmRVSHY2dFFmOA=='

Questo header può essere usato sia nei tool indicati

sia con curl, esempio

curl -X GET -k -H 'Authorization: Basic ZWxhc3RpYzpobWxhY3JsODI4MmRVSHY2dFFmOA==' -i 'https://192.168.56.50:9200'

# Kibana

## Installare Kibana

sudo apt install kibana  
modificare la configurazione  
sudo nano /etc/kibana/kibana.yml  
server.port: 5601  
server.host: 0.0.0.0

Configurare kibana per l'avvio automatico => abilitare il demone  
systemctl daemon-reload  
systemctl enable kibana

Configurare l'ip di ascolto nel file /etc/kibana/kibana.yml  
server.host: 0.0.0.0

Avviare il servizio  
sudo systemctl start kibana

Per poter usare kibana come utente di elasticsearch bisogna associarlo con i permessi corretti. Questo viene fatto in due modi

## Setup Kibana

### Enrolment key Modo facile

Accedere alla pagina <http://192.168.56.50:5601/>   
Incollare il token di enrolment generato tramite il comando

/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana

Generare un codice di verifica con il comando

/usr/share/kibana/bin/kibana-verification-code

Da notare che questi token hanno una scadenza

##### Config manuale **Modo facile ma lungo**

##### Accedere alla pagina <http://192.168.56.50:5601/>

Scegliere configurazione manuale

Eseguire il reset della password per l'utente kibana\_system

/usr/share/elasticsearch/bin/elasticsearch-reset-password --username kibana\_system

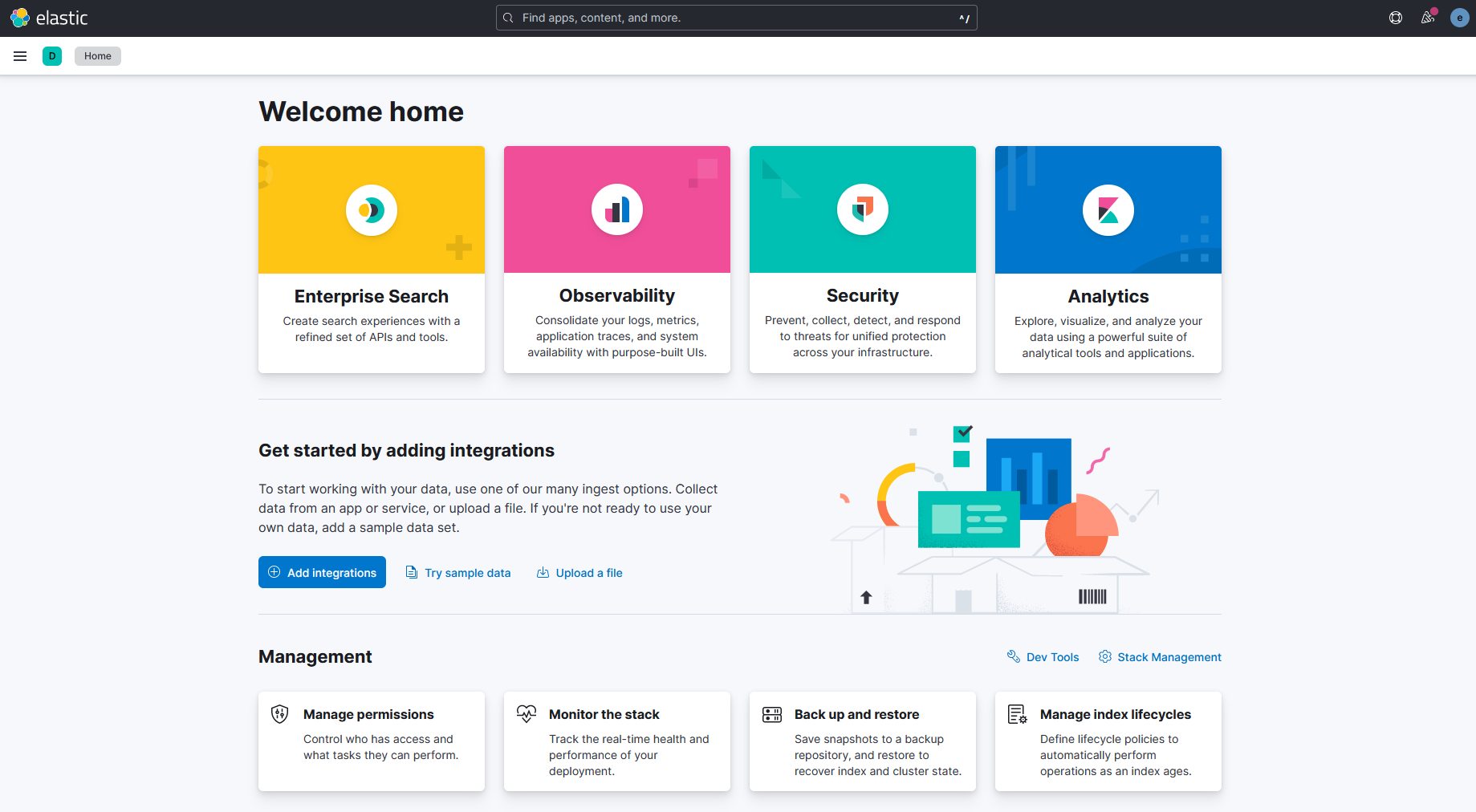
inserire i dati di configurazione come segue

address: https://192.168.56.50:9200  
username: kibana\_system  
password: PASSWORD\_GENERATA  
trust certificate: yes

### Login

A questo punto Kibana è configurato ed è possibile accedere con login, per comodità usare l'utente elastic.

Ignorate l'aggiunta di integratori e dovreste arrivare alla pagina home:



# Configurazione Metricbeat

## Setup metricbeat

Installazione e abilitazione del demone

sudo apt install metricbeat  
sudo systemctl daemon-reload  
sudo systemctl enable metricbeat

Questa era la parte facile. Il "difficile" arriva ora con la configurazione dell'utenza.

## Configurazione

Siccome abbiamo abilitato la sicurezza, tramite ssl e utenze, dobbiamo configurare anche metricbeat per poter inviare i dati ad elastic in maniera sicura.

Le configurazioni del demone sono da impostare nel file

/etc/metricbeat/metricbeat.yml

### SSL

Uno dei parametri di configurazione è il trust del certificato usato dal demone elasticsearch. Nel nostro caso stiamo usando un certificato auto generato del quale dobbiamo un un qualche modo fidarci.

Questo lo possiamo fare in due modi, aggiungerlo al keystore di metricbeat oppure nel file di configurazione (più raipido)

Per prima cosa determiniamo l'impronta digitale del cerificato, chiamata "fingerprint"

Questa possiamo leggerla dal certificato stesso, con un browser oppure con il comando

openssl x509 -in /etc/elasticsearch/certs/http\_ca.crt -text -fingerprint -sha256

siccome l'putput è molto lungo si può usare grep per filtrarlo

Il valore cercato è qualcosa di simile a questo

sha256 Fingerprint=AB:20:0D:92:6B:B4:DE:FE:68:52:47:A3:45:DC:16:B5:D1:B0:88:6C:27:EA:09:79:AB:1F:20:6D:36:98:EE:09

Tenetelo da parte togliendo tutti i ":" ottenendo una stringa unita come questa "AB200D926BB4DEFE685247A345DC16B5D1B0886C27EA0979AB1F206D3698EE09"

### API Key

Per autenticarsi si possono usare nome utente e password (basic auth) che sono salvati nel file di config o nel keystore, oppure una chiave api.

Ne useremo due, create in modi diversi, per scopi diversi.

### Chiave per "Beats setup"

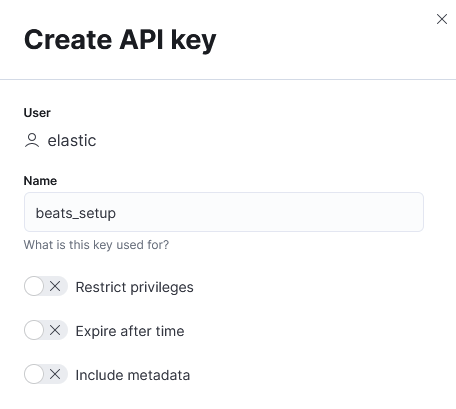
Questa chiave servirà una sola volta per configurare kibana ed elastic. Infatti al momento in elastic non è presente un "DB" adatto a contenere i dati rilevati da metricbeat, e in Kibana non sono presenti delle dashboard per visualizzarne i dati.

Questo viene fatto con un comando apposito, ma richiede dei permessi specifici.

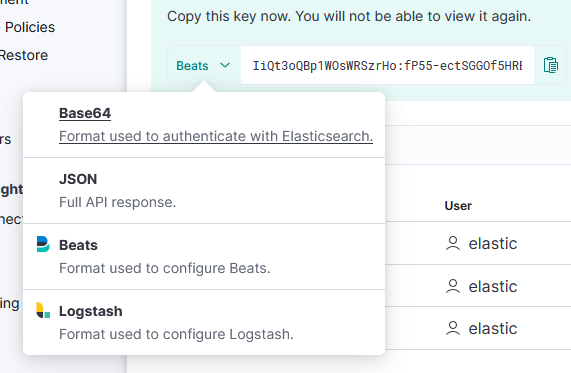
La guida https://www.elastic.co/guide/en/beats/metricbeat/8.5/privileges-to-setup-beats.html spiega quali sono i permessi necessari, ma in modo inutilmente complicato.

Per semplificare le cose usiamo Kibana

Accedere al menu di Management > Security > API keys



Nel formato di output scegliere JSON oppure Beats (o entrambi) e salvarsela. Se non la si salva non sarà più possibile avere questi dati e sarà necessario ricrearla.



### Aggiornare il file di configurazione di metricbeat

Aprire il file:

/etc/metricbeat/metricbeat.yml

nella sezione "Elasticsearch Output" commentare quanto presente e inserire i dati seguenti:

output.elasticsearch:  
  hosts: ["https://10.20.0.50:9200"]  
  api\_key: ID:CHIAVE  
  ssl:  
    enabled: true  
    ca\_trusted\_fingerprint: "FINGERPRINT\_CERTIFICATO"

Esempio:

output.elasticsearch:  
  hosts: ["https://10.20.0.50:9200"]  
  api\_key: IyTd3IQBp1WOsWRSFqwQ:Tb7BkKMySqqufn-eof5uYQ  
  ssl:  
    enabled: true  
    ca\_trusted\_fingerprint: "AB200D926BB4DEFE685247A345DC16B5D1B0886C27EA0979AB1F206D3698EE09"

In fondo al file oppure nella sezione "Kibana" aggiungere

setup.ilm.overwrite : true

Questo permetterà di eseguire correttamente il setup.

### Test della configurazione

Per verificare se la configurazione è corretta e la connettività ad elasticsearch funziona basta eseguire i comandi seguenti

metricbeat test config

Output previsto:

Config ok

metricbeat test output

Output previsto:

elasticsearch: https://10.20.0.50:9200...  
  parse url... OK  
  connection...  
    parse host... OK  
    dns lookup... OK  
    addresses: 10.20.0.50  
    dial up... OK  
  TLS...  
    security: server's certificate chain verification is enabled  
    handshake... OK  
    TLS version: TLSv1.3  
    dial up... OK  
  talk to server... OK  
  version: 8.5.2

## Setup indici e dashboard

Creare le dashboard (il comando impiega qualche secondo, attendere con pazienza)

metricbeat setup --dashboards  
  
Loading dashboards (Kibana must be running and reachable)  
Loaded dashboards

Creare gli indici (questo è veloce)

metricbeat setup --index-management

Index setup finished.

Per verificare che ci siano davvero basta andare in Kibana

Nella sezione Discover, nel filtro degli indici deve comparire metricbeat-\*

Nella dezione Dashboard devono essere elencate numerose voci che cominciano con "[Metricbeat "

### Chiave per "Publisher"

In gergo, un publisher è un beat che invia dati ad elasticsearch. Anche in questo caso andiamo a creare una chiave, ma questa volta lo facciamo tramite API REST.

Come indicato nella guida https://www.elastic.co/guide/en/beats/metricbeat/8.5/privileges-to-publish-events.html possiamo eseguire il tutto tramite curl.

Per`è meglio farlo tramite GUI.

# Setup Beats

## ELK server

### Metricbeat

Per installare metricbeat.

sudo apt install metricbeat

Visualizzare quali moduli sono presenti in metricbeat e quali sono attivi o meno (.disabled)…

…come files:

ls /etc/metricbeat/modules.d/

…come comando:

sudo metricbeat modules list

Per disattivare tutto:

sudo metricbeat modules disable

Per attivare moduli singoli (di default system dovrebbe essere attivo, a noi serve quindi lo attiviamo).

sudo metricbeat modules enable system

Creare le dashboard e configurazioni relative in Kibana (da fare una sola volta).

sudo metricbeat setup -e

sudo service metricbeat start

Da questo momento in poi in Kibana compaiono dati.

Impostare start automatico…

…con systemd:

sudo systemctl enable metricbeat.service

…con initd:

sudo update-rc.d metricbeat defaults

## Windows DC

### Metricbeat

se serve installare telnet per effettuare test di connessione  
dism /online /Enable-Feature /FeatureName:TelnetClient  
telnet 10.20.0.50 9200

**Attenzione, usare la versione corrispondente al proprio stack ELK!**

download:   
https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.8.14-windows-x86\_64.zip  
  
decomprimere in C:\Program Files\metricbeat  
  
verificare la versione metricbeat.exe version  
metricbeat version 6.8.14 (amd64), libbeat 6.8.14 [5cd281153df1eb5e95a4a31994a7846d2c9493e8 built 2021-02-02 18:56:41 +0000 UTC]  
  
PS C:\Program Files\metricbeat> .\install-service-metricbeat.ps1

Status   Name               DisplayName  
------   ----               -----------  
Stopped  metricbeat         metricbeat  
  
modificare "C:\Program Files\metricbeat\metricbeat.yml"  
  
output.elasticsearch:  
  # Array of hosts to connect to.  
  hosts: ["localhost:9200"]  
  
setup.kibana:  
  
  # Kibana Host  
  # Scheme and port can be left out and will be set to the default (http and 5601)  
  # In case you specify and additional path, the scheme is required: http://localhost:5601/path  
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601  
  host: "localhost:5601"  
   
in

output.elasticsearch:  
  # Array of hosts to connect to.  
  hosts: ["10.20.0.50:9200"]    
  
setup.kibana:  
    
# Kibana Host  
  # Scheme and port can be left out and will be set to the default (http and 5601)  
  # In case you specify and additional path, the scheme is required: http://localhost:5601/path  
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601  
  host: "10.20.0.50:5601"  
  
metricbeat.exe modules enable windows  
  
metricbeat.exe test config  
  
Service-Start metricbeat

## Ubuntu OwnCloud

Eseguire il setup di Metricbeat in autonomia abilitando i moduli necessari.

Installare anche filebeat in modo da aggiungere altre metriche.