document technique

Malika NEDJAR

7 juillet 2015

1 Introduction

L'installation et le paramétrage de ELK (Elasticsearch, Logstash et Kibana) sur un serveur "loghost-ng" et sans installation de Logstash Forwarder (appelé auparavant Lumberjack) sur les machines clientes dont nous souhaitons analyser les logs car les logs sont centralisé au niveau de serveur. Le serveur Web sera le serveur apache.

La récupération des logs est centralisée avec les outils rsyslog et Logstash qui est un excellent produit de tri, traitement et collecte des logs.

Le logiciel logstash possède de nombreux avantages, entres autres :

— Il est compatible avec les protocoles courants comme syslog, idéal pour s'insérer dans des architectures existantes.

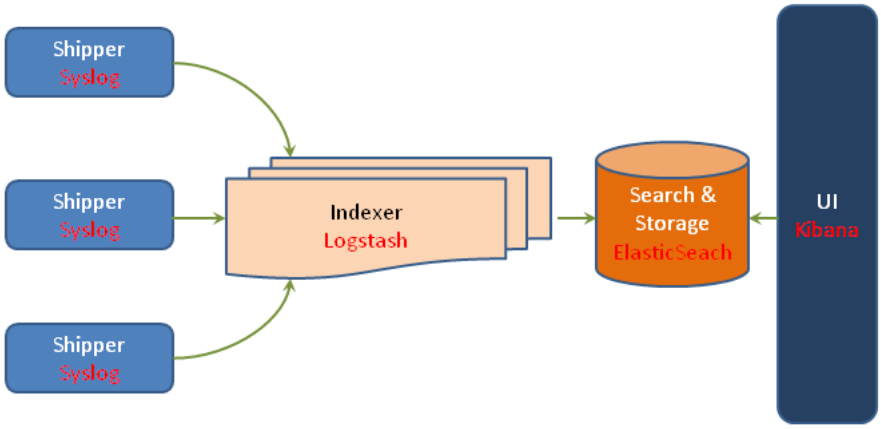
— Il est extensible dans le sens où tout est configuré en utilisant des expressions régulières, dont beaucoup sont déjà fournies.

— Il est très scalable car il repose sur une base de données elasticsearch (qui peut fonctionner en cluster), des services de "cache" (broker).

Pour la mise en oeuvre au sein d'une architecture simple, logstash se positionne sur un serveur central faisant office de collecteur de logs. Sur ce serveur logstash est configuré avec une entrée de type syslog sur le port 5000 (par ex).

2 Installation de ELK

ELK : Il s'agit de la combinaison de trois logiciels : **Elasticsearch** en tant que moteur d'indexation, **Logstash** pour la collecte, l'analyse et le stockage de logs, et **Kibana** pour construire des graphiques et requêter et visualiser n'importe quelles données. L'ensemble étant assez complet.



2.1 Installation de Elasticsearch

Elasticsearch est le moteur de recherche open source rendu populaire avec l'arrivée du big data. Il propose des fonctionnalités avancées, facilement configurable, rivalisant avec les meilleurs logiciels propriétaires. Un moteur de recherche, c'est une brique technologique capable d'ingérer un nombre

massif de données, et de les mettre à disposition d'un utilisateur de façon intelligente, en quelques millisecondes.

```
$wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
$echo "deb http://packages.elastic.co/elasticsearch/1.5/debian stable main" |
 sudo tee -a /etc/apt/sources.list
$sudo apt-get update && sudo apt-get install elasticsearch
```

Après on modifie quelques champs dans le fichier **/etc/default/elasticsearch** et en redémarre Elasticsearch.

```
# Elasticsearch log directory
LOG_DIR=/var/log/elasticsearch

# Elasticsearch data directory
DATA_DIR=/var/lib/elasticsearch

# Elasticsearch work directory
WORK_DIR=/tmp/elasticsearch
https://www.youtube.com/watch?v=Rbsg5QWqvPM
# Elasticsearch configuration directory
CONF_DIR=/etc/elasticsearch

# Elasticsearch configuration file (elasticsearch.yml)
CONF_FILE=/etc/elasticsearch/elasticsearch.yml
```

ElasticSearch est maintenant installé. on passe à la configuration d'Elasticsearch, en éditant le fichier de configuration, /etc/elasticsearch/elasticsearch.yml, et modifier les éléments suivants :

- Remplacer la ligne #**network.host : 192.168.0.1** par **network.host : loghost-ng**
- Décommenter la ligne #**cluster.name : elasticsearch**
- Remplacer la ligne #**node.name : "Franz Kafka"** par **node.name : cremi**
- Décommenter la ligne #**node.master : true**
- Décommenter la ligne #**node.data : true**
- Décommenter la ligne #**discovery.zen.ping.multicast.enabled : false**
- Remplacer la ligne #**index.number_of_shards : 5** par **index.number_of_shards : 1**
- Remplacer la ligne #**index.number_of_replicas : 1** par **index.number_of_replicas : 0**

Après avoir configurer Elasticsearch, on active le service elasticsearch :

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

Ensuite, on exécute la commande suivante pour démarrer Elasticsearch au démarrage :

```
$sudo update-rc.d elasticsearch defaults 95 10
```

On vérifie qu'ElasticSearch s'est correctement lancé consiste à ouvrir l'URL ***http ://loghost-ng :9200/*** dans un navigateur Web. En effet, ElasticSearch écoute par défaut sur le port 9200 pour répondre aux éventuelles requêtes HTTP REST qui lui sont faites. La réponse du serveur confirme le bon fonctionnement d'ElasticSearch et rappelle le numéro de la version installée :

```
# curl 'http://loghost-ng:9200/_search?pretty'
ou
# curl 'http://loghost-ng:9200/_nodes/?pretty'
```

2.2 Installation de logstash

Logstash est un outil de collecte, analyse et stockage de logs. Pour la collecte, il sait gérer plus d'une trentaine d'événements. Un événement peut être un message syslog, un mail via le protocole IMAP, un tweet ou encore une commande IRC.

Il va ensuite analyser ces événements, et les mettre en forme à l'aide de filtres. Il existe également une vingtaine de filtres : standardisation de la date, découpage du message, structuration du message via grok. Après filtrage, on obtient un message relativement clair, avec des couples clé-valeur que l'on pourra exploiter plus tard.

Enfin, il exporte ces données, traitées ou non, sous divers formats : email, sortie standard, fichier texte, alarme Nagios, entrée en base de données ElasticSearch.

Ainsi, on peut très bien traiter plusieurs flux de logs différents, appliquer un filtre à chacun pour harmoniser le tout, et les stocker correctement.

Logstash utilise Java, il est donc nécessaire qu'un JDK soit installé sur le "serveur". On peut utiliser au choix OpenJDK ou Oracle JDK.

```
$wget -qO - http://packages.elasticsearch.org/GPG-KEY-elasticsearch | sudo apt-key add -
$deb http://packages.elasticsearch.org/logstash/1.4/debian stable main
$echo 'deb http://packages.elasticsearch.org/logstash/1.4/debian stable main' |
 sudo tee /etc/apt/sources.list.d/logstash.list
$apt-get update
$apt-get install logstash

$curl 'http://loghost-ng:9200/'
$service logstash start
```

Dans mon installation, les fichiers de configuration se trouvent dans le dossier **/etc/logstash/conf.d**.

Dans ma configuration j'utilise deux types d'entrées :

— Des entrées de types fichiers, créés par le rsyslog serveur.

— Des entrées de type syslog qui agissent comme un serveur syslog qui écoute.

Pour la partie "input" du serveur, on crée le fichier **input.conf** ou j'ai choisis les entrées de type syslog qui agissent comme un serveur syslog qui écoute sur le port 5000 et pour celà, j'ai configuré **rsyslog** pour qu'il envois tous les logs ver le port 5000 en créant le fichier **/etc/rsyslog.d/logstash.conf** :

```
$template LOGSTASH,"<%PRI%>%TIMESTAMP% %HOSTNAME% %syslogtag%%msg%"
$ActionForwardDefaultTemplate LOGSTASH
*.* @@10.0.220.22:5000
```

voici le fichier **input.conf**.

```
input {
        syslog {
                port => 5000
                host => "0.0.0.0"
                type => "syslog"
        }

#       file {
#               path => ["/srv/log/*.log", "/srv/log/syslog"]
#               exclude => "*.gz"
#                type => "syslog"
#       }
}
```

Pour la partie "input" du serveur, on crée le fichier **filter.conf**.

```
filter {

if [type] == "syslog" {
        grok {
                match => { "message" => "<%{POSINT:syslog_pri}>%{SYSLOGTIMESTAMP:timestamp} %{SY
                overwrite => [ "%{[logsource]}" ]
                add_field => [ "received_at", "%{@timestamp}" ]
                add_field => [ "received_from","%{host}" ]
        }
        syslog_pri { }

        date {
                locale => "fr"
                match => [ "timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
        }
                mutate {
                        replace=> ["received_at", "%{timestamp}"]
                        convert => [ "received_at" , "string" ]
                        gsub => [
                                # replace all : with -
                                "received_at", ":", "."
                        ]
                }
}

if [program] == "pperso-access" or [program] == "apache2" {
        grok {
                # logs HTTPS
                match => { "message" => "%{COMBINEDAPACHELOG}  %{IPORHOST:host} %{GREEDYDATA:ssl
                # logs HTTP
                match => { "message" => "%{COMBINEDAPACHELOG}  %{IPORHOST:host}" }
        }
        mutate {
                replace => ["type","apache"]
        }
}
if [type] == "apache" {
        useragent {
                prefix => "agent."
                source => "agent"
        }
# Add geolocalization attributes based on ip.
        geoip {
                source => "clientip"
                target => "geoip"
                database => "/usr/share/GeoIP/GeoLiteCity.dat"
                add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
                add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}"  ]
        }
        mutate {
                convert => [ "[geoip][coordinates]", "float" ]
                gsub => [
                        "referrer", "^\"", "",
```

```
                        "referrer", "\"$", "",
                        "agent", "^\"", "",
                        "agent", "\"$", ""
                ]
        }
}
#On reforme le message avec les infos sélectionnées
#       mutate {
#               replace => ["host","%{logsource}"]
#       }
}
```

Pour la partie “output” du serveur, on crée le fichier **output.conf**.

```
output {
  elasticsearch {
    protocol => "http"
    flush_size => '100'
    host => 'loghost-ng'\maketitle
    index => 'logstash-%{+YYYY.MM.dd}'
    port => '9200'
  }
# just pour débuger /srv/log/logstash/logstash.stdout
    stdout {
       codec => rubydebug
    }
}
```

2.3 Installation de Kibana

Kibana est une interface web permettant de rechercher des informations stockées par Logstash dans ElasticSearch. Techniquement, Kibana est une application web développée en Javascript pur et donc déployable sur un simple serveur HTTP du type Apache HTTP Server.

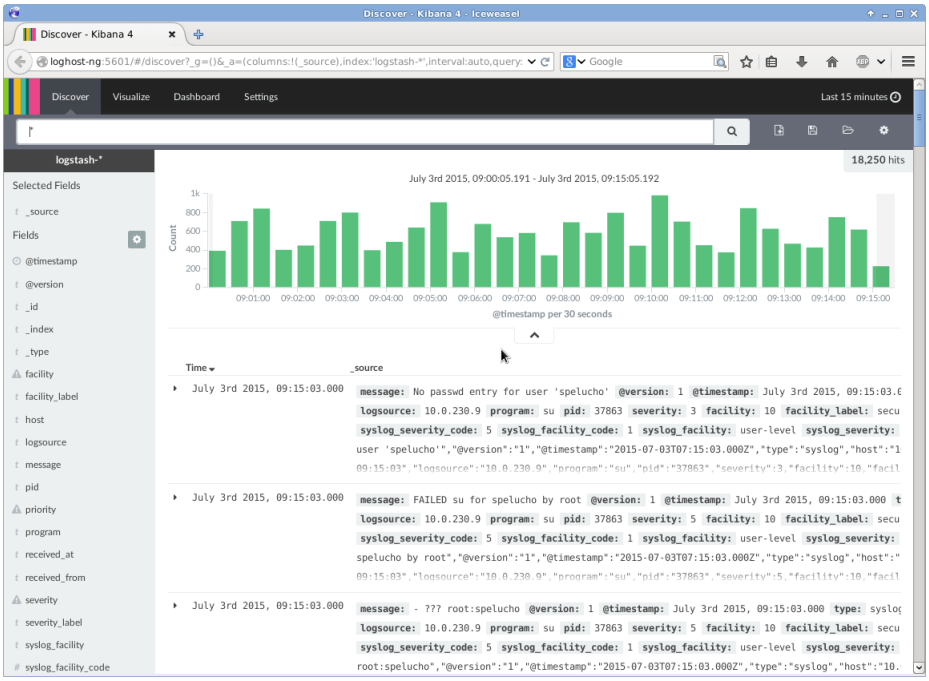
```
$cd /opt
$wget https://download.elasticsearch.org/kibana/kibana/kibana-4.0.1-linux-x64.tar.gz

$tar xvf kibana-*.tar.gz
$vi /opt/kibana/config/kibana.yml
$cd /etc/init.d && sudo wget https://gist.githubusercontent.com/thisismitch/
 8b15ac909aed214ad04a/raw/bce61d85643c2dcdfbc2728c55a41dab444dca20/kibana4

$sudo chmod +x /etc/init.d/kibana4
$sudo update-rc.d kibana4 defaults 96 9
$sudo service kibana4 start

$systemctl enable kibana
$systemctl start kibana
```

après l’installation on passe à la configuration de kinana, on lance le navigateur par l’adresse : *http ://loghost-ng :5601*

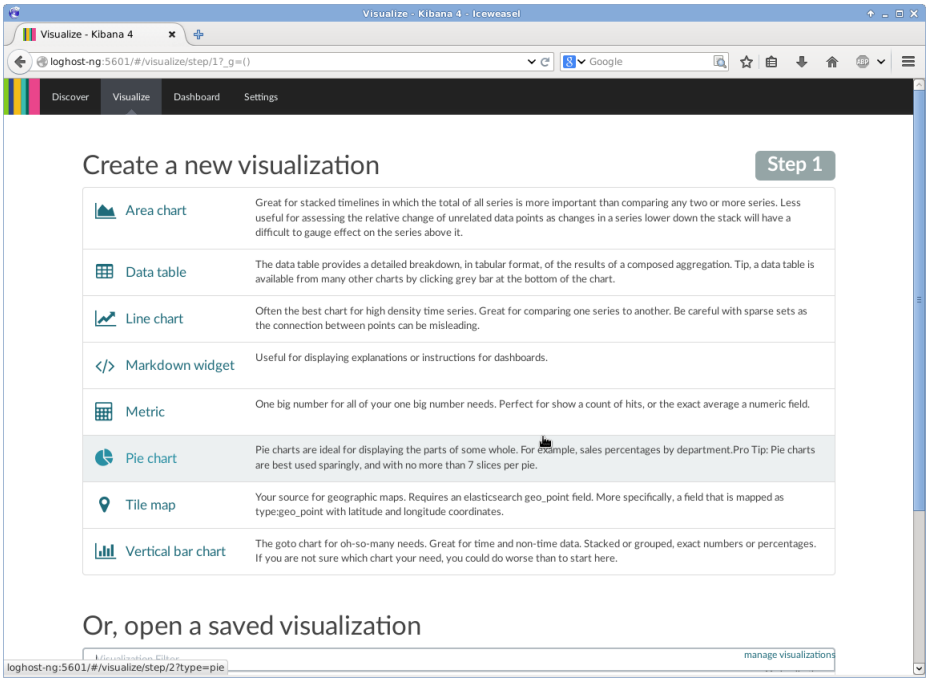
L’interface Kibana est divisé en quatre sections principales :

— Discover
— Visualize
— Dashboard
— Settings

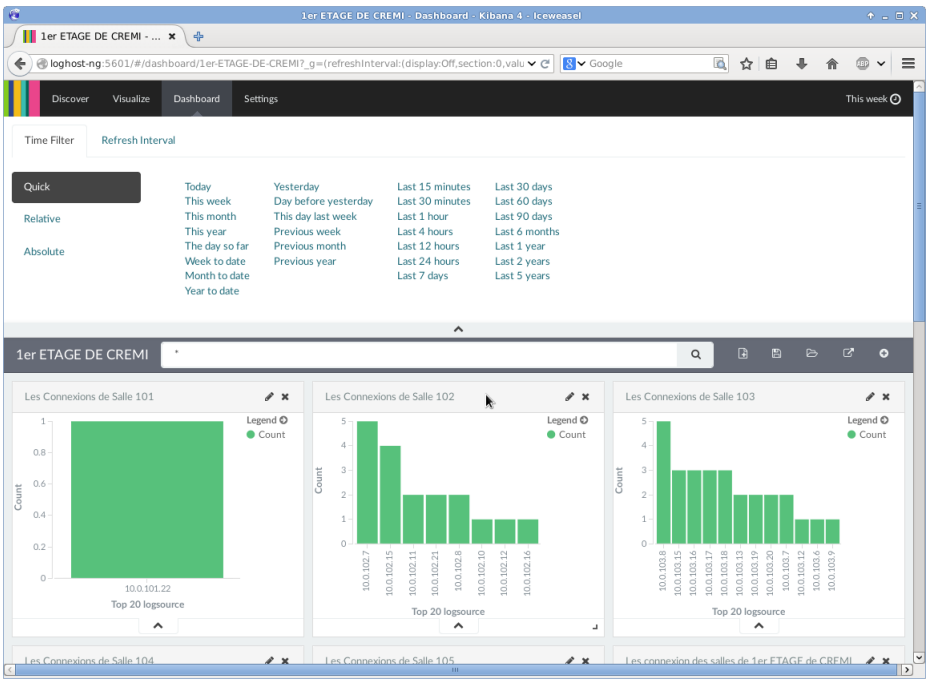
Discover : Présente toutes les données de votre index comme une table des documents.



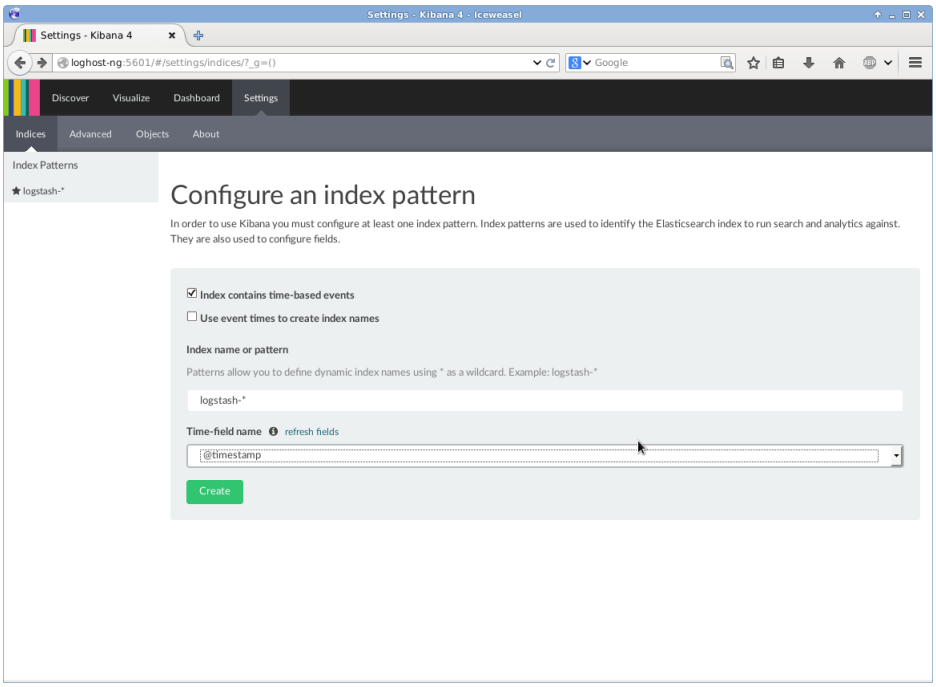
Visualize : Cette page est l'endroit où on peut créer, modifier et consulter nos propres visualisations personnalisées . Il existe plusieurs types de visualisations.



Dashboard : Une page ou on peut créer , modifier et afficher nos propres tableaux de bord personnalisés . Avec un tableau de bord , nous pouvons combiner plusieurs visualisations sur une seule page, puis les filtrer en fournissant une requête de recherche ou en sélectionnant des filtres en cliquant sur les éléments de la visualisation. Les tableaux de bord sont utiles lorsque nous souhaitons obtenir un aperçu de nos journaux , et établir des corrélations entre les différentes visualisations et les journaux.



Settings : Cette page nous permet de modifier une variété de choses comme valeurs par défaut ou des modèles d'index.



En effectuant des requêtes qui permettent de retrouver parmi tous les logs les informations qu'on voulait dans une barre de recherche :

— La requête pour visualiser toutes les connexions(les authentifications) :
`"(kdm:session)" AND "session opened for user"`
— La requête pour visualiser tous les connexions de la salle 001 & 004 :
`"(kdm:session)" AND "session opened for user" AND 10.0.4.*`
— La requête pour visualiser tous les connexions de la salle 101 :
`"(kdm:session)" AND "session opened for user" AND 10.0.101.*`
— La requête pour visualiser tous les connexions x2go :
— La requête pour visualiser tous les connexions ssh :
`"pam_unix(sshd:session): session opened for user" NOT fberarde NOT root`

3 Création des vues et des dashboards Kibana

pour plus d'information d'utilisation de kibana 4 consulter :
https ://www.elastic.co/blog/kibana-4-video-tutorials-part-1