Graylog : Centralisateur de log

Introduction :

Dans ce tuto, nous allons voir comment installer un centralisateur de journaux d'événements windows, graylog, sur une machine redhat.

Graylog est un outil très puissant de centralisation, d'analyse et d'extraction des logs, pour ne garder que l'essentiel de ses fonctionnalités et de son côté très évolutif. Graylog remplace idéalement le serveur syslog-ng qui collecte les logs de tous vos serveurs/machines.

Vouloir collecter les logs c'est très bien, mais pouvoir les centraliser, les traiter et les analyser c'est encore mieux! Dans l'installation du serveur graylog, on va utiliser le service Rsyslog, Nxlog qui permettra de collecter les journaux de systèmes Windows, Unix, Linux, BSD, etc … Et en outre, le service Rsyslog aura pour rôle la collecte des logs du serveur graylog.

I-Installation des prérequis

-Installation des utilitaires « mlocate», «updatedb» et «net-tools»

Les utilitaires «mlocate», «updatedb» et «net-tools» font partis du hit parade d'outils utiles dans un environnement Linux pour rechercher notamment des fichiers, nous allons donc les installer:

yum install mlocate –y

updatedb

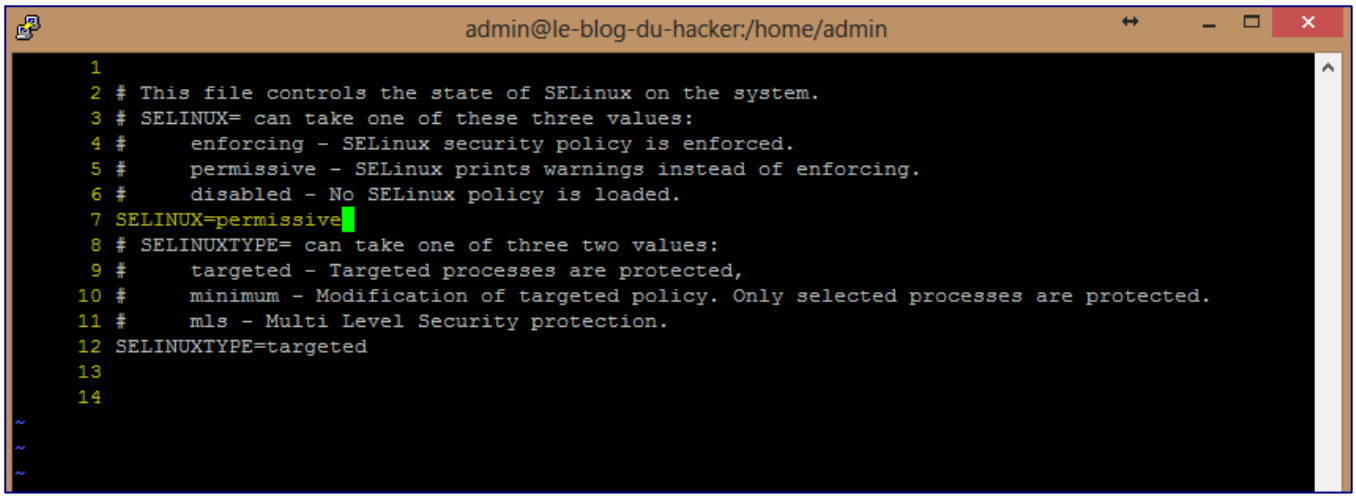
yum install net-tools -y

-Configuration de «Selinux»:

SELinux (*Security Enhanced Linux*) est un système de contrôle d'accès obligatoire (*Mandatory Access Control*) qui s'appuie sur l'interface *Linux Security Modules* fournie par le noyau Linux. Concrètement, le noyau interroge SELinux avant chaque appel système pour savoir si le processus est autorisé à effectuer l'opération concernée. Pour cette installation nous avons décidé de modifier Selinux. Pour des raisons de simplification de cette installation, nous avons choisi de le rendre «*permissive*» au lieu de «*enforcing*»(par défaut). Dans certains cas il sera nécessaire de le désactiver (*disabled).*

Nous allons éditer le fichier de configuration SeLinux, puis rendre « permissive » Selinux:

vi /etc/selinux/config



-Installation de Java

L'installation du paquet Java fait partie des prérequis nécessaires à l'installation du serveur GRAYLOG. **Il faudra s'assurer que nous avons installé la dernière version de Java**. Il est à noter que Java est un paquet délicat à prendre en compte dès maintenant, car il conditionne le bon déroulement de la suite des opérations d'installation (serveur GRAYLOG et ses fonctionnalités ainsi que tout l'environnement MongoDB, Elasticsearch etc.). **La version de Java devra être identique et la plus récente, dans le cas où plusieurs serveur GRAYLOG devaient coexister au sein d'une même infrastructure réseau, ceci afin d'éviter un dysfonctionnement des centralisateurs GRAYLOG, voir le «plantage total»!** Lançons au préalable une mise à jour de yum puis installons le paquet Java:

yum update

yum install java-1.6.0-openjdk*

-Installation de EPEL et de dépôts complémentaires pour CentOS

Le dépôt EPEL propose des pack d'extensions complémentaires utiles non disponibles depuis les dépôts officiels de Red Hat Enterprise Linux. Les instructions son également inclus pour installer d'autres dépôts secondaires, tels que le Projet Communautaire IUS ou le dépôt Remi RPM. Bien que EPEL propose uniquement des logiciels qui ne sont pas inclus dans les dépôts officiels des éditions Linux CentOS et Red Hat Enterprise, IUS et Remi proposent de nouvelles versions des logiciels (tels que MySQL et PHP) qui sont déjà présents dans les dépôts officiels.

Il est conseillé de réaliser une installation manuelle du dépôt comme suit :

yum install wget -y

wget http://dl.fedoraproject.org/pub/epel/5/x86_64/epel-release-7-5.noarch.rpm

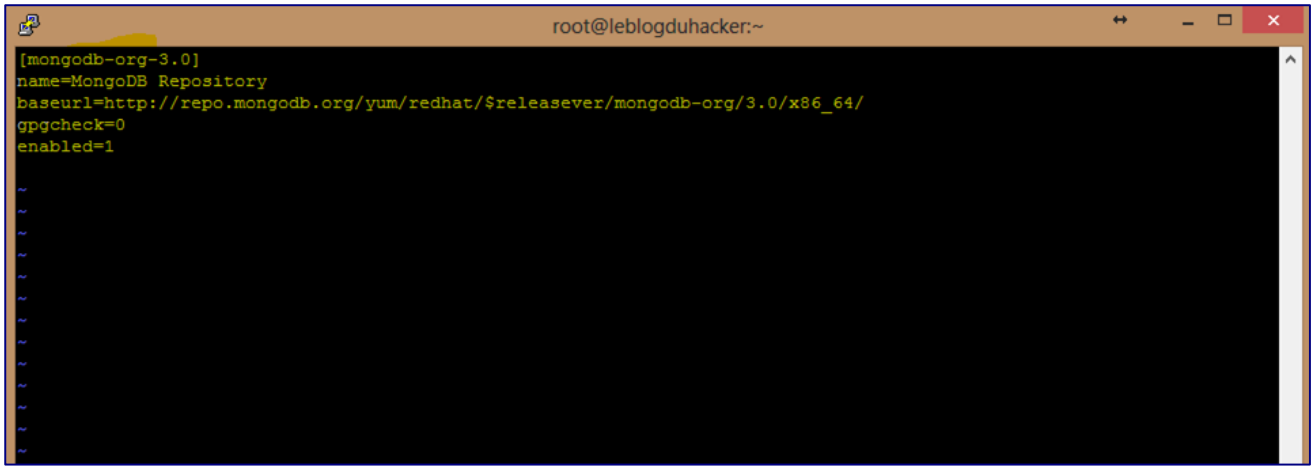
rpm -Uvh epel-release-7*.rpm

II- Installation et configuration du mongodb

Mongodb est un système de gestion de base de données ou SGBD, comme MySql ou PostgreSql, mais dont le mécanisme est complètement différent. Fini le temps ou il fallait créer un schéma de tables relationnelles et créer des requêtes Sql complexes. Grâce à MongoDB vous allez pouvoir stocker vos données un peu comme vous le feriez dans un fichier Json. C'est à dire, une sorte de dictionnaire géant composé de clés et de valeurs. Ces données peuvent ensuite être exploitées par du javascript, directement intégré dans MongoDB, mais peuvent également être exploitées par d'autre langage comme python.

Création du fichier de dépôt:

vi /etc/yum.repos.d/mongodb-org-3.0.repo



yum install -y mongodb-org

chkconfig mongod on

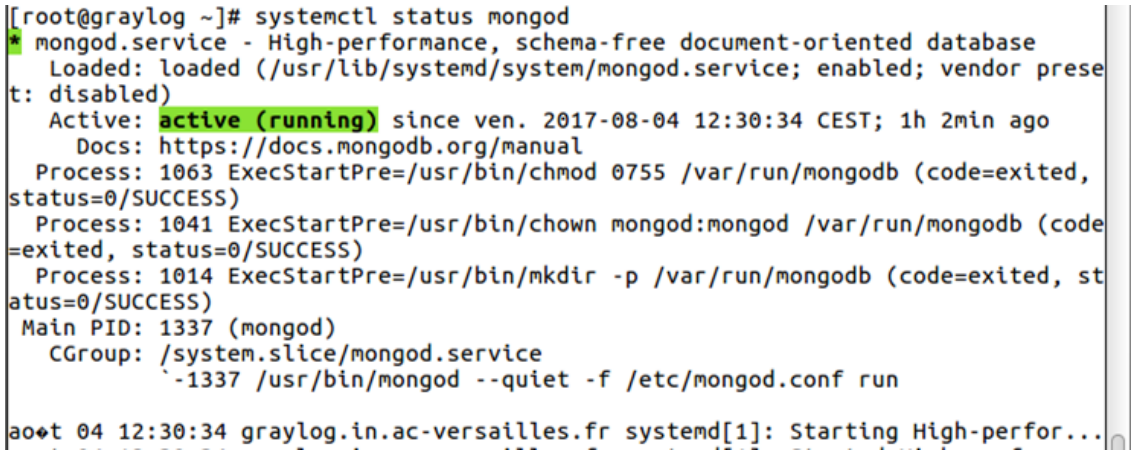
systemctl start mongod

systemctl deamon reload

Voici le résultat de sortie du statut de « **mongod.service**« :

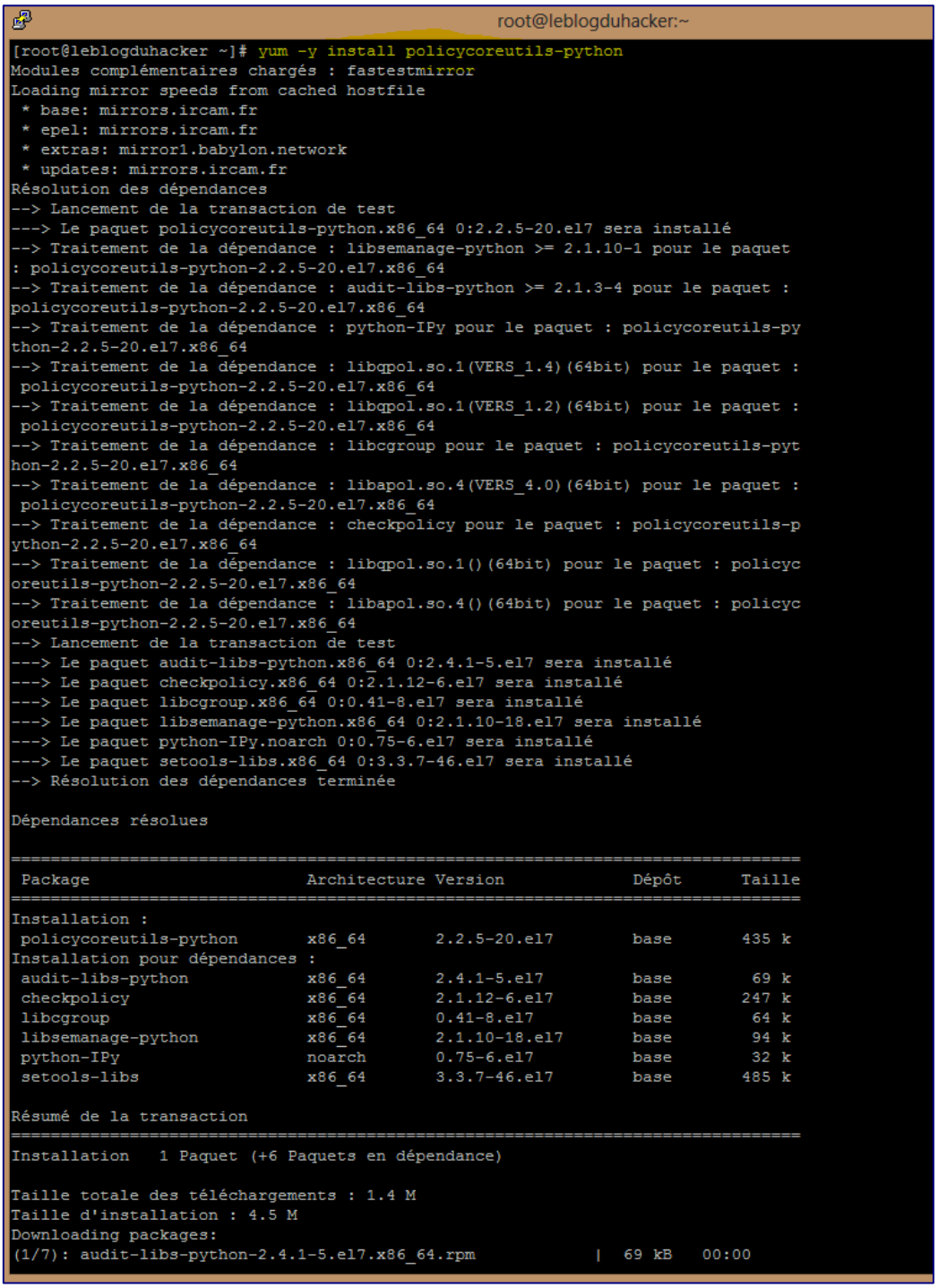
systemctl status mongod

-



Installation des utilitaires Python

yum -y install policycoreutils-python



-Configuration de SeLinux pour autoriser MongoDB à démarrer:

semanage port -a -t mongod_port_t -p tcp 27017

service mongod start

chkconfig mongod on

Aide commande :

chkconfig : activer les services au démarrage sous RedHat / CentOS
semanage port -l |grep -w mongod_port_t //permet de voir le port utilisé par le service mongod
netstat -a | grep – mongod

III- Installation du moteur de recherche et d'indexation 'ELASTICSEARCH'

Importer la clé et créer le fichier de dépôt Elasticsearch:

rpm –import **https://packages.elastic.co/GPG-KEY-elasticsearch**

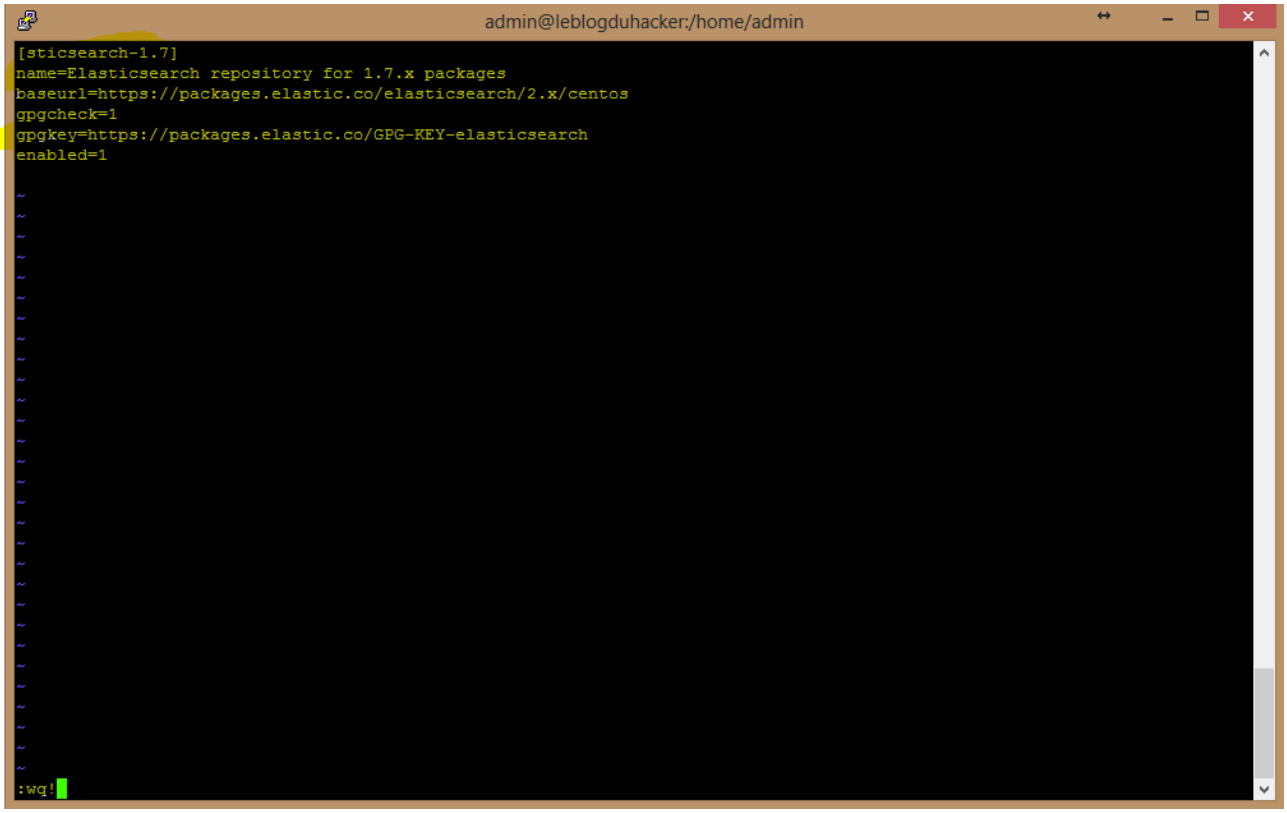
Ici nous allons créer le fichier de dépôt Elasticsearch:

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

et y insérer les lignes suivantes:

```
[elasticsearch-1.7]
name=Elasticsearch repository for 1.7.x packages
baseurl=http://packages.elastic.co/elasticsearch/1.7/centos
gpgcheck=1
gpgkey=http://packages.elastic.co/GPG-KEY-elasticsearch
enabled=1
```

Voici le résultat:



-Installation de la dernière version d'Elasticsearch:

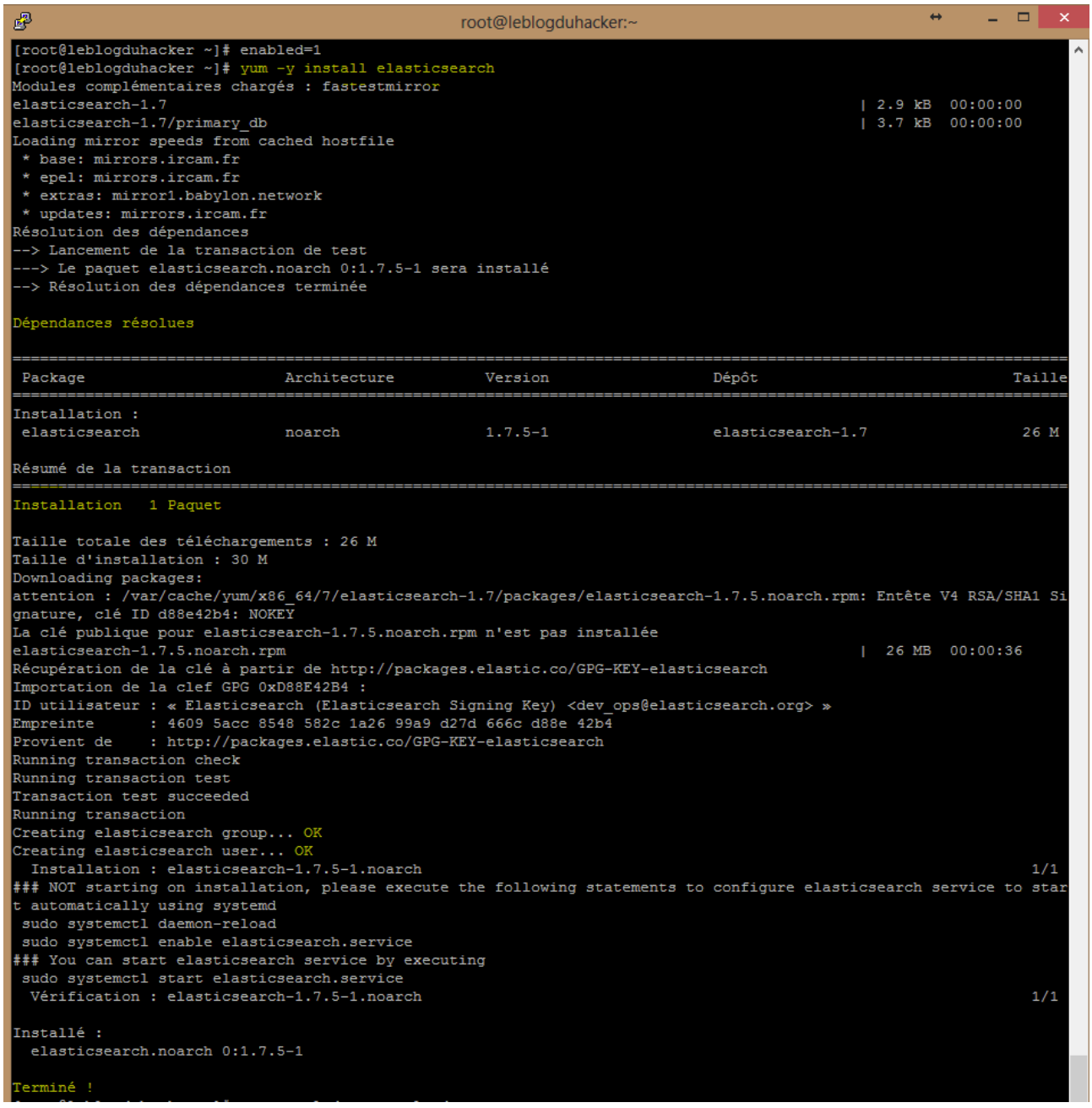
yum –y install elasticsearch*

Note importante:

Les fichiers de configuration d'Elasticsearch ont été installé ici: **/etc/elasticsearch/**

Les fichiers d'installation d'Elasticsearch ont été installé ici: **/usr/share/elasticsearch/**

Voici le résultat au terme de l'installation d'Elasticsearch:



-*Vérification des dépôts disponibles:*

Vous pouvez voir si les dépôts dont vous avez besoin sont installés et activés en exécutant la commande suivante :

yum repolist

Le résultat de sortie devrait ressembler à ceci:

```
[root@le-blog-du-hacker admin]# yum repolist
Modules complémentaires chargés : fastestmirror
Loading mirror speeds from cached hostfile
 * base: centos.crazyfrogs.org
 * epel: epel.mirrors.ovh.net
 * extras: centos.quelquesmots.fr
 * updates: centos.crazyfrogs.org
id du dépôt                          nom du dépôt
base/7/x86_64                        CentOS-7 - Base
elasticsearch-2.x                    Elasticsearch repository for 2.x packages
*epel/x86_64                         Extra Packages for Enterprise Linux 7 - x86_64
extras/7/x86_64                      CentOS-7 - Extras
updates/7/x86_64                     CentOS-7 - Updates
repolist: 19 929
[root@le-blog-du-hacker admin]#
```

Maintenant rechargeons le «**démon**», activons '**elasticsearch.service**', redémarrons **Elasticsearch** et vérifions son **Statut**:

systemctl daemon-reload

systemctl enable elasticsearch.service

systemctl start elasticsearch.service

systemctl status elasticsearch.service

```
root@graylog:~
)
 Main PID: 1337 (mongod)
   CGroup: /system.slice/mongod.service
           `-1337 /usr/bin/mongod --quiet -f /etc/mongod.conf run

ao♦t 04 12:30:34 graylog.in.ac-versailles.fr systemd[1]: Starting High-perfor...
ao♦t 04 12:30:34 graylog.in.ac-versailles.fr systemd[1]: Started High-perform...
ao♦t 04 12:30:36 graylog.in.ac-versailles.fr mongod[1075]: about to fork chil...
ao♦t 04 12:30:36 graylog.in.ac-versailles.fr mongod[1075]: forked process: 1337
Hint: Some lines were ellipsized, use -l to show in full.
[root@graylog ~]# systemctl status elasticsearch.service
* elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset: disa
bled)
   Active: active (running) since ven. 2017-08-04 12:30:34 CEST; 1h 16min ago
     Docs: http://www.elastic.co
 Main PID: 1033 (java)
   CGroup: /system.slice/elasticsearch.service
           `-1033 /bin/java -Xms256m -Xmx1g -Djava.awt.headless=true -XX:+Use...

ao♦t 04 12:30:34 graylog.in.ac-versailles.fr systemd[1]: Started Elasticsearch.
ao♦t 04 12:30:34 graylog.in.ac-versailles.fr systemd[1]: Starting Elasticsear...
Hint: Some lines were ellipsized, use -l to show in full.
[root@graylog ~]#
```

Nous allons éditer le fichier de configuration Elasticsearch c'est à dire ici '**/etc/elasticsearch/elasticsearch.yml**':

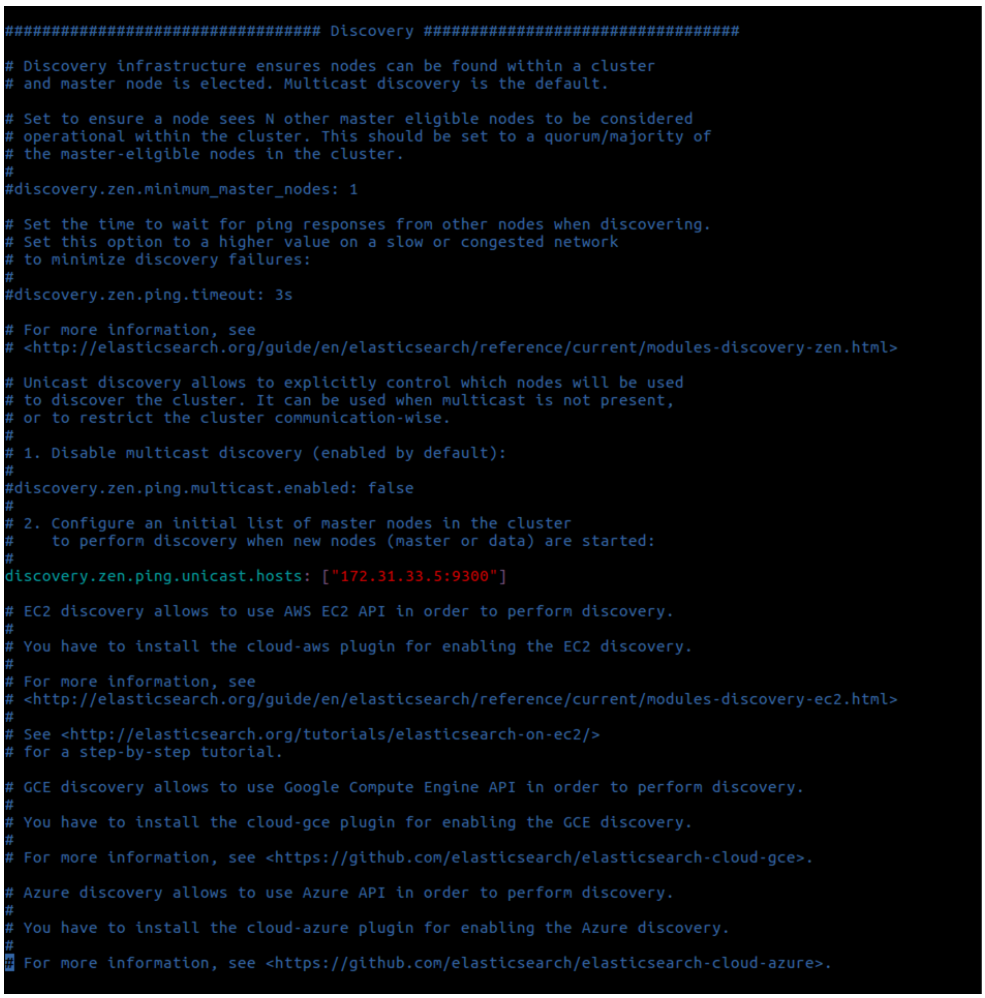
vi /etc/elasticsearch/elasticsearch.yml

Remarque: Dans le cadre de cette installation, nous utiliserons le fichier de configuration installé par Elasticsearch dans '**/etc/elasticsearch/elasticsearch.yml '** et non celui généré par l'environnement de travail dans ' /etc/elasticsearch/elasticsearch-1.7.5/config/elasticsearch.yml '

Voici un extrait du fichier de configuration Elasticsearch de cette installation:

```
# to disable it, set the following:
#node.max_local_storage_nodes: 1


################################### Index ###################################

# You can set a number of options (such as shard/replica options, mapping
# or analyzer definitions, translog settings, ...) for indices globally,
# in this file.
#
# Note, that it makes more sense to configure index settings specifically for
# a certain index, either when creating it or by using the index templates API.
#
# See <http://elasticsearch.org/guide/en/elasticsearch/reference/current/index-modules.html> and
# <http://elasticsearch.org/guide/en/elasticsearch/reference/current/indices-create-index.html>
# for more information.

# Set the number of shards (splits) of an index (5 by default):
#
#index.number_of_shards: 5

# Set the number of replicas (additional copies) of an index (1 by default):
#
#index.number_of_replicas: 1

# Note, that for development on a local machine, with small indices, it usually
# makes sense to "disable" the distributed features:
#
index.number_of_shards: 1
index.number_of_replicas: 0

# These settings directly affect the performance of index and search operations
# in your cluster. Assuming you have enough machines to hold shards and
# replicas, the rule of thumb is:
#
# 1. Having more *shards* enhances the _indexing_ performance and allows to
#    _distribute_ a big index across machines.
# 2. Having more *replicas* enhances the _search_ performance and improves the
#    cluster _availability_.
#
# The "number_of_shards" is a one-time setting for an index.
#
# The "number_of_replicas" can be increased or decreased anytime,
# by using the Index Update Settings API.
#
# Elasticsearch takes care about load balancing, relocating, gathering the
# results from nodes, etc. Experiment with different settings to fine-tune
# your setup.

# Use the Index Status API (<http://localhost:9200/A/_status>) to inspect
# the index status.


################################### Paths ###################################

# Path to directory containing configuration (this file and logging.yml):
#
#path.conf: /path/to/conf
```

```
# You should also make sure that the Elasticsearch process is allowed to lock
# the memory, eg. by using `ulimit -l unlimited`.


############################## Network And HTTP ##############################

# Elasticsearch, by default, binds itself to the 0.0.0.0 address, and listens
# on port [9200-9300] for HTTP traffic and on port [9300-9400] for node-to-node
# communication. (the range means that if the port is busy, it will automatically
# try the next port).

# Set the bind address specifically (IPv4 or IPv6):
#

#network.bind_host: 127.0.0.1
network.bind_host: 0.0.0.0

# Set the address other nodes will use to communicate with this node. If not
# set, it is automatically derived. It must point to an actual IP address.
#
#network.publish_host: 192.168.0.1

# Set both 'bind_host' and 'publish_host':
#
network.host: 172.31.33.5

# Set a custom port for the node to node communication (9300 by default):
#
#transport.tcp.port: 9300

# Enable compression for all communication between nodes (disabled by default):
#
#transport.tcp.compress: true

# Set a custom port to listen for HTTP traffic:
#
http.port: 9200

# Set a custom allowed content length:
#
#http.max_content_length: 100mb

# Disable HTTP completely:
#
#http.enabled: false


################################### Gateway ###################################

# The gateway allows for persisting the cluster state between full cluster
# restarts. Every change to the state (such as adding an index) will be stored
# in the gateway, and when the cluster starts up for the first time,
# it will read its state from the gateway.

# There are several types of gateway implementations. For more information, see
# <http://elasticsearch.org/guide/en/elasticsearch/reference/current/modules-gateway.html>.

# The default gateway type is the "local" gateway (recommended):
#
```



```
################################### Discovery ###################################

# Discovery infrastructure ensures nodes can be found within a cluster
# and master node is elected. Multicast discovery is the default.

# Set to ensure a node sees N other master eligible nodes to be considered
# operational within the cluster. This should be set to a quorum/majority of
# the master-eligible nodes in the cluster.
#
#discovery.zen.minimum_master_nodes: 1

# Set the time to wait for ping responses from other nodes when discovering.
# Set this option to a higher value on a slow or congested network
# to minimize discovery failures:
#
#discovery.zen.ping.timeout: 3s

# For more information, see
# <http://elasticsearch.org/guide/en/elasticsearch/reference/current/modules-discovery-zen.html>

# Unicast discovery allows to explicitly control which nodes will be used
# to discover the cluster. It can be used when multicast is not present,
# or to restrict the cluster communication-wise.
#
# 1. Disable multicast discovery (enabled by default):
#
#discovery.zen.ping.multicast.enabled: false
#
# 2. Configure an initial list of master nodes in the cluster
#    to perform discovery when new nodes (master or data) are started:
#
discovery.zen.ping.unicast.hosts: ["172.31.33.5:9300"]

# EC2 discovery allows to use AWS EC2 API in order to perform discovery.
#
# You have to install the cloud-aws plugin for enabling the EC2 discovery.
#
# For more information, see
# <http://elasticsearch.org/guide/en/elasticsearch/reference/current/modules-discovery-ec2.html>
#
# See <http://elasticsearch.org/tutorials/elasticsearch-on-ec2/>
# for a step-by-step tutorial.

# GCE discovery allows to use Google Compute Engine API in order to perform discovery.
#
# You have to install the cloud-gce plugin for enabling the GCE discovery.
#
# For more information, see <https://github.com/elasticsearch/elasticsearch-cloud-gce>.

# Azure discovery allows to use Azure API in order to perform discovery.
#
# You have to install the cloud-azure plugin for enabling the Azure discovery.
#
# For more information, see <https://github.com/elasticsearch/elasticsearch-cloud-azure>.
```

-Testons l'accès en localhost sur le port 9200 avec CURL :

Voici le résultat de sortie de la commande:

curl -X GET http://localhost:9200

{
« status » : 200,
« name » : « hacker Diki »,
« cluster_name » : « leblogduhacker »,
« version » : {
« number » : « 1.7.5 »,
« build_hash » : « xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx »,
« build_timestamp » : « xxxxxxxxxxxxxxxxxx »,
« build_snapshot » : false,
« lucene_version » : « 4.10.4 »
},
« tagline » : « You Know, for Search »
}

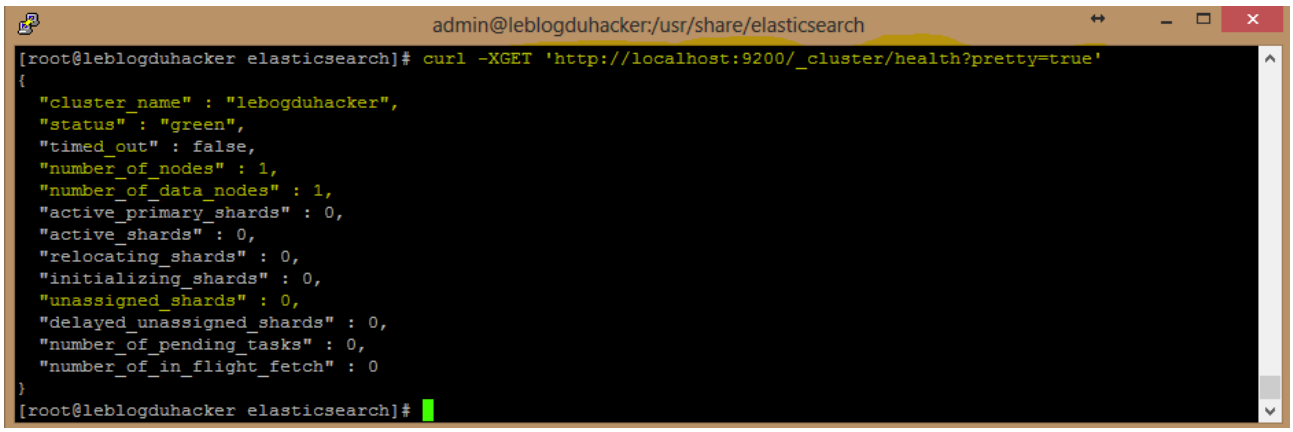
```
################################## Slow Log ##################################

# Shard level query and fetch threshold logging.

#index.search.slowlog.threshold.query.warn: 10s
#index.search.slowlog.threshold.query.info: 5s
#index.search.slowlog.threshold.query.debug: 2s
#index.search.slowlog.threshold.query.trace: 500ms

#index.search.slowlog.threshold.fetch.warn: 1s
#index.search.slowlog.threshold.fetch.info: 800ms
#index.search.slowlog.threshold.fetch.debug: 500ms
#index.search.slowlog.threshold.fetch.trace: 200ms

#index.indexing.slowlog.threshold.index.warn: 10s
#index.indexing.slowlog.threshold.index.info: 5s
#index.indexing.slowlog.threshold.index.debug: 2s
#index.indexing.slowlog.threshold.index.trace: 500ms

################################## GC Logging ################################

#monitor.jvm.gc.young.warn: 1000ms
#monitor.jvm.gc.young.info: 700ms
#monitor.jvm.gc.young.debug: 400ms

#monitor.jvm.gc.old.warn: 10s
#monitor.jvm.gc.old.info: 5s
#monitor.jvm.gc.old.debug: 2s

################################## Security ##################################

# Uncomment if you want to enable JSONP as a valid return transport on the
# http server. With this enabled, it may pose a security risk, so disabling
# it unless you need it is recommended (it is disabled by default).
#
#http.jsonp.enable: true
script.disable_dynamic: true
```

-Testons la santé du 'cluster' (instance) Elasticsearch:

curl -XGET 'http://localhost:9200/_cluster/health?pretty=true'

Le résultat de santé du **'cluster'** Elasticsearch doit être normalement à l'état « **green** » (vert):

```
root@graylog:~
[root@graylog ~]# nano /etc/elasticsearch/elasticsearch.yml
[root@graylog ~]# vim /etc/elasticsearch/elasticsearch.yml
[root@graylog ~]# curl -X GET http://localhost:9200
{
  "status" : 200,
  "name" : "Graylog",
  "cluster_name" : "elasticsearch",
  "version" : {
    "number" : "1.7.6",
    "build_hash" : "c730b59357f8ebc555286794dcd90b3411f517c9",
    "build_timestamp" : "2016-11-18T15:21:16Z",
    "build_snapshot" : false,
    "lucene_version" : "4.10.4"
  },
  "tagline" : "You Know, for Search"
}
[root@graylog ~]#
```



IV- Installation des plugins HEAD et MARVEL

Avant d'installer ces deux plugins, rendons-nous dans le bon répertoire des fichiers de configuration Elasticsearch:

cd /usr/share/elasticsearch/

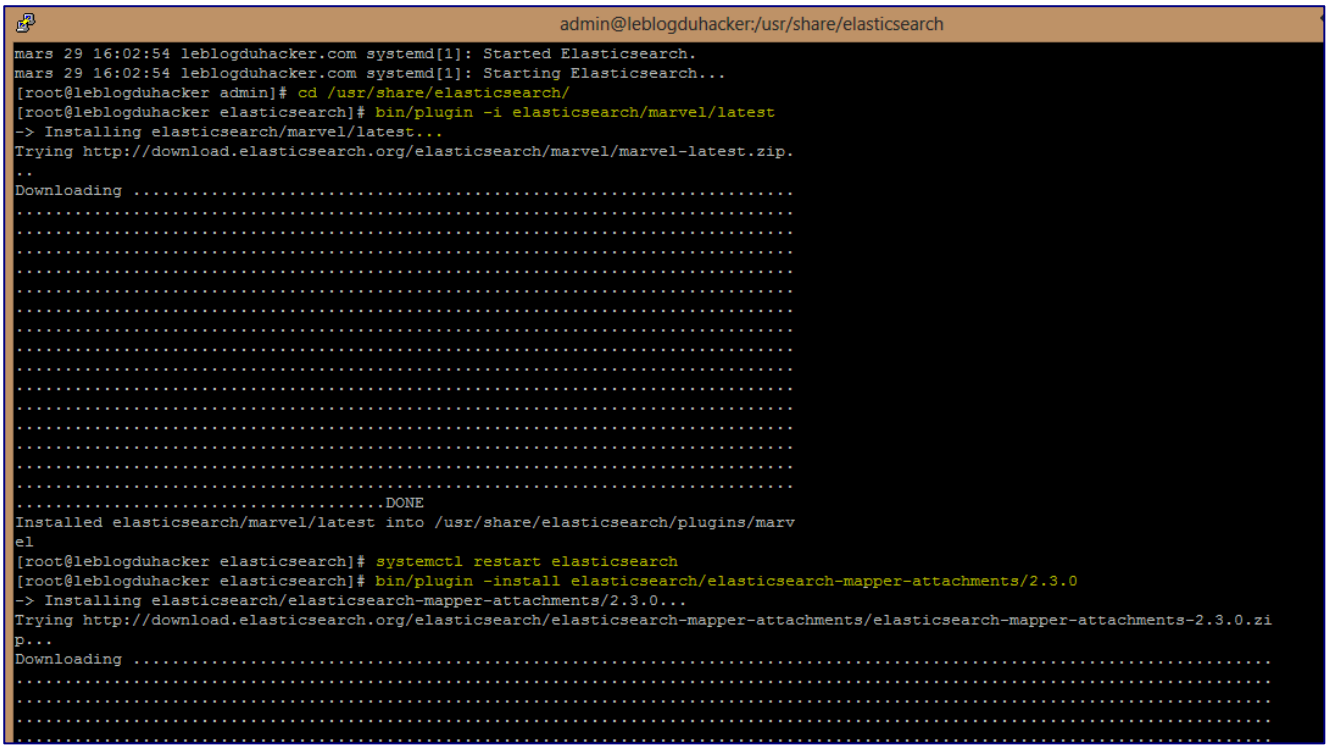
Ces Plugins vont nous permettre d'accéder et de communiquer avec Elasticsearch via un navigateur internet à l'adresse : *http://adresse_IP_Serveur:9200/_plugin/head* et **http://adresse_IP_Serveur:9200/_plugin/marvel** à l'aide d'un navigateur internet:

./bin/plugin -install mobz/elasticsearch-head

./bin/plugin -i elasticsearch/marvel/latest

systemctl restart elasticsearch

./bin/elasticsearch -f



Deux autres plugins sont nécessaires. Installons-les toujours au même endroit**:**

cd /usr/share/elasticsearch/

«Mapper Attachment Type» pour ElasticSearch =>
https://github.com/elasticsearch/elasticsearch-mapper-attachments

«ICU Analysis pour Elasticsearch» =>**https://github.com/elasticsearch/elasticsearch-analysis-icu**

Vous pouvez les installer en utilisant ces deux commandes :

bin/plugin -install elasticsearch/elasticsearch-mapper-attachments/2.5.0
bin/plugin -install elasticsearch/elasticsearch-analysis-icu/2.5.0

Remarque: Ici j'ai souhaité vous faire découvrir ces deux plugins, mais un seul suffit. Généralement le plugin 'HEAD' est le plus adapté pour ce genre d'installation.

Avant d'ouvrir votre navigateur internet pour se connecter à l'**API HTTP** '**HEAD**' et «**MARVEL**» d'Elasticsearch, **je vous propose maintenant de redémarrer le serveur graylog pour prendre en compte l'installation des plugins** (Normalement nous aurions du arrêter le « oeud» pour installer les plugins. Je préfère simplifier):

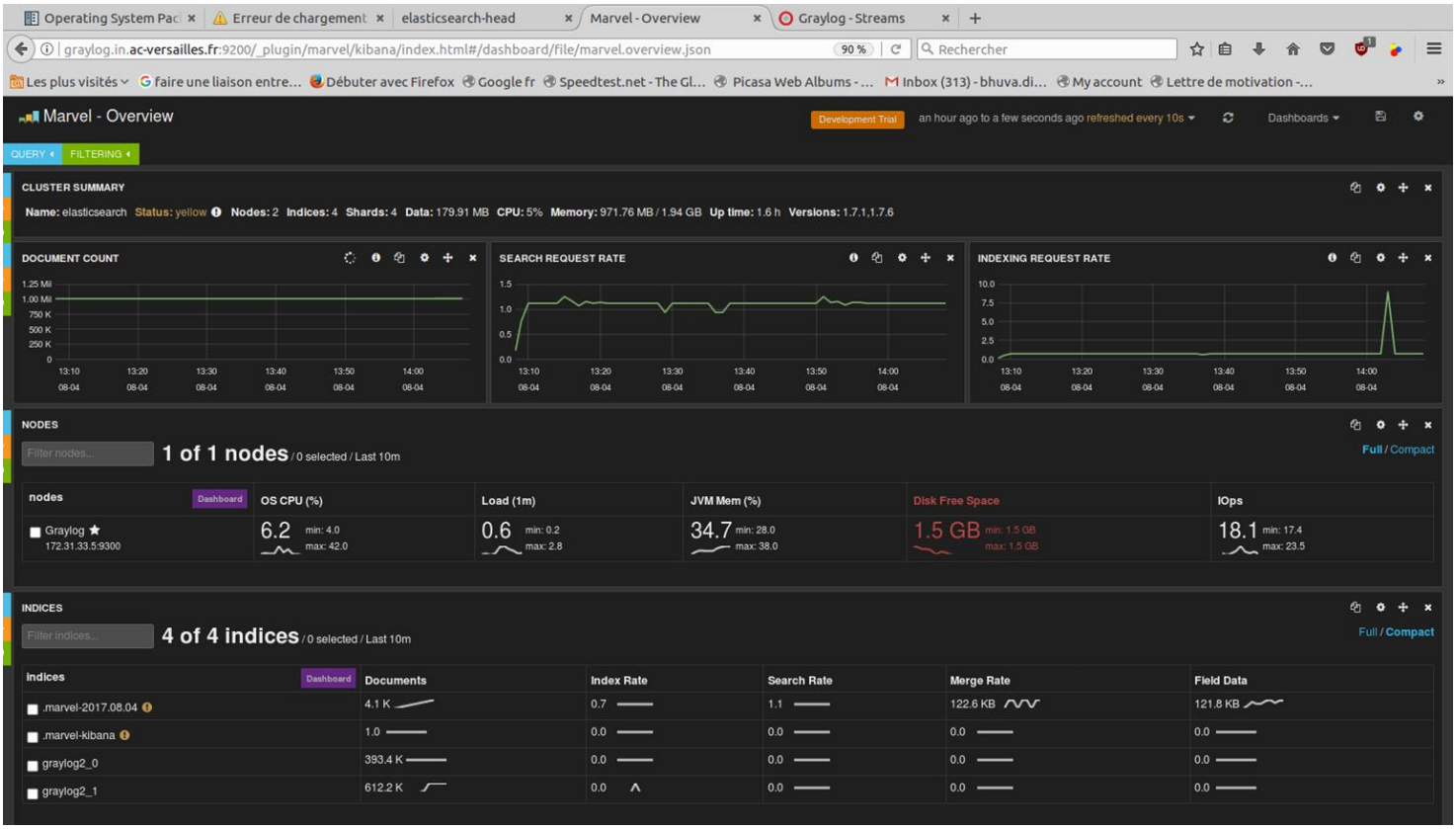
reboot

Pour accéder à Elasticsearch via le navigateur internet, c'est très simple, saisissez l'URL suivante:

http://'adresse-Ip-server':9200/_plugin/marvel

http://'adresse-Ip-server':9200/_plugin/head

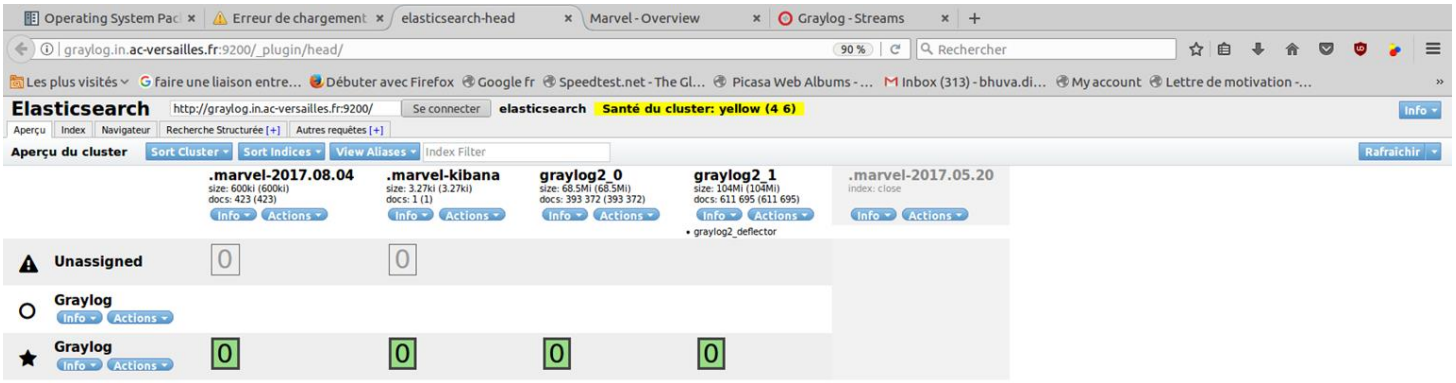
Connexion via l'API REST HTTP 'MARVEL' Elasticsearch



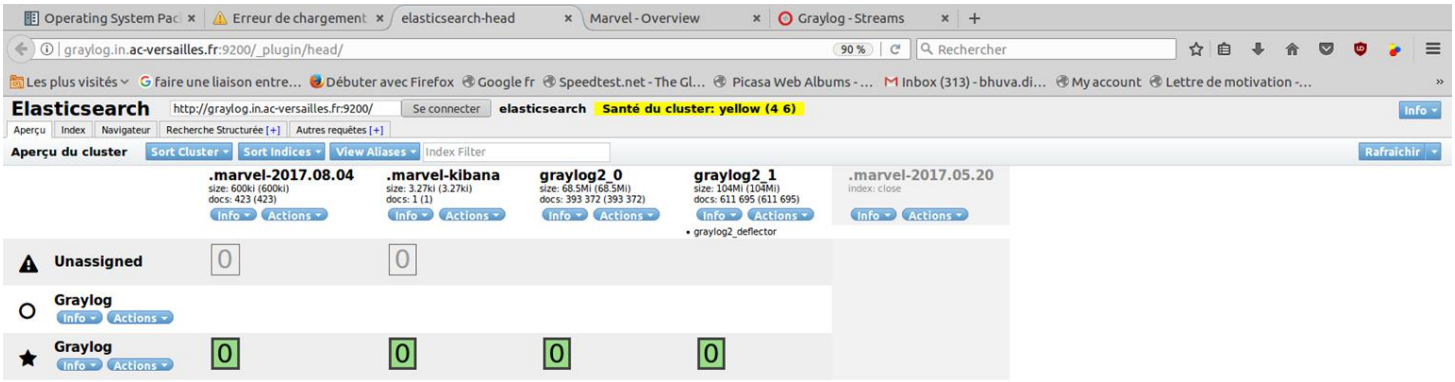
Connexion via l'api rest http 'HEAD' Elasticsearch

-PROBLÈME DE 'CLUSTER': état de santé au «jaune»:

En cas de problème dans votre 'cluster' il se pourrait, que pour différentes raisons «obscures» que seule la raison pourrait expliquer, l'état de santé de votre 'cluster» soit «jaune» par exemple. **Sans rentrer dans les détails, si votre 'cluster» a la «jaunisse» ce n'est pas normal. Voici la protocole médical à suivre, c'est très simple:**



La première commande va demander via le port d'écoute 9200 d'Elasticsearch de rechercher



dans le dossier 'shards' tous les 'shards' et 'index' non assignés, et de nous les retourner:

curl -s http://'adresse_IP_serveur':9200/_cat/shards | grep UNASS

```
[root@leblogduhacker elasticsearch]# curl -s http://192.168.1.55:9200/_cat/shards | grep UNASS

.marvel-2016.04.04 0 r UNASSIGNED
.marvel-2016.04.03 0 r UNASSIGNED
.marvel-2016.04.02 0 r UNASSIGNED
.marvel-2016.04.01 0 r UNASSIGNED
.marvel-2016.04.08 0 r UNASSIGNED
.marvel-2016.03.29 0 r UNASSIGNED
.marvel-2016.04.05 0 r UNASSIGNED
.marvel-kibana     0 r UNASSIGNED
.marvel-2016.03.31 0 r UNASSIGNED
.marvel-2016.03.30 0 r UNASSIGNED
```

La deuxième commande (pour faire simple) va supprimer les ‘shards non assignés (à effectuer un par un pour chaque ‘shards’). La suivante va modifier le nombre de ‘replicas’ à la valeur ‘0’ (zéro). Pourquoi ? C’est du au fait qu’un ‘shard’ a besoin d’un autre ‘node’ (noeud) disponible pour lui tout seul, afin qu’il puisse se positionner dans ce ‘node’, pour ensuite appliquer un ‘réplicas’ à ce ‘shard’. Un ‘shard’ et un ‘replicas’ ne peuvent pas cohabiter dans le même «node». En d’autre terme, si l’état de santé de votre ‘cluster’ est «jaune», cela signifie que vous avez probablement mal configuré le fichier de configuration /etc/elasticsearch/elasticsearch.yml’ et accidentellement créé plusieurs «nodes». **Je vous rappelle au passage que cette installation évolue autour d’un seul «noeud maître» (master node), un seul «Shard» et aucun «replicat». Pourquoi ? Simplement parce que notre unique «node» est conçu nativement pour effectuer à lui seul toutes les opérations nécessaires (du Node, du Shard et du replicat etc. à travers une Instance unique, le «Cluster»)**

(commande à faire pour chaque ‘shards’ non assigné, ici: ‘ .marvel-1016.04.04, 03, 02, 01 ‘ etc)

curl -XDELETE http://’adresse_IP-server’:9200/’non_du_shard’

curl -XPUT http://192.168.1.55:9200/_settings -d ‘{«number_of_replicas» :0}’

La troisième commande va vérifier qu’il n’y ait plus de «shards’ non assignés (la même que la première):

curl -s http://’adresse_IP_serveur’:9200/_cat/shards | grep UNASS

Les deux dernières commandes vont tout simplement redémarrer le service ‘**elasticsearch**‘ et ‘**graylog-server**‘:

systemctl restart elasticsearch

systemctl restart graylog-server

V- Installation du serveur GRAYLOG

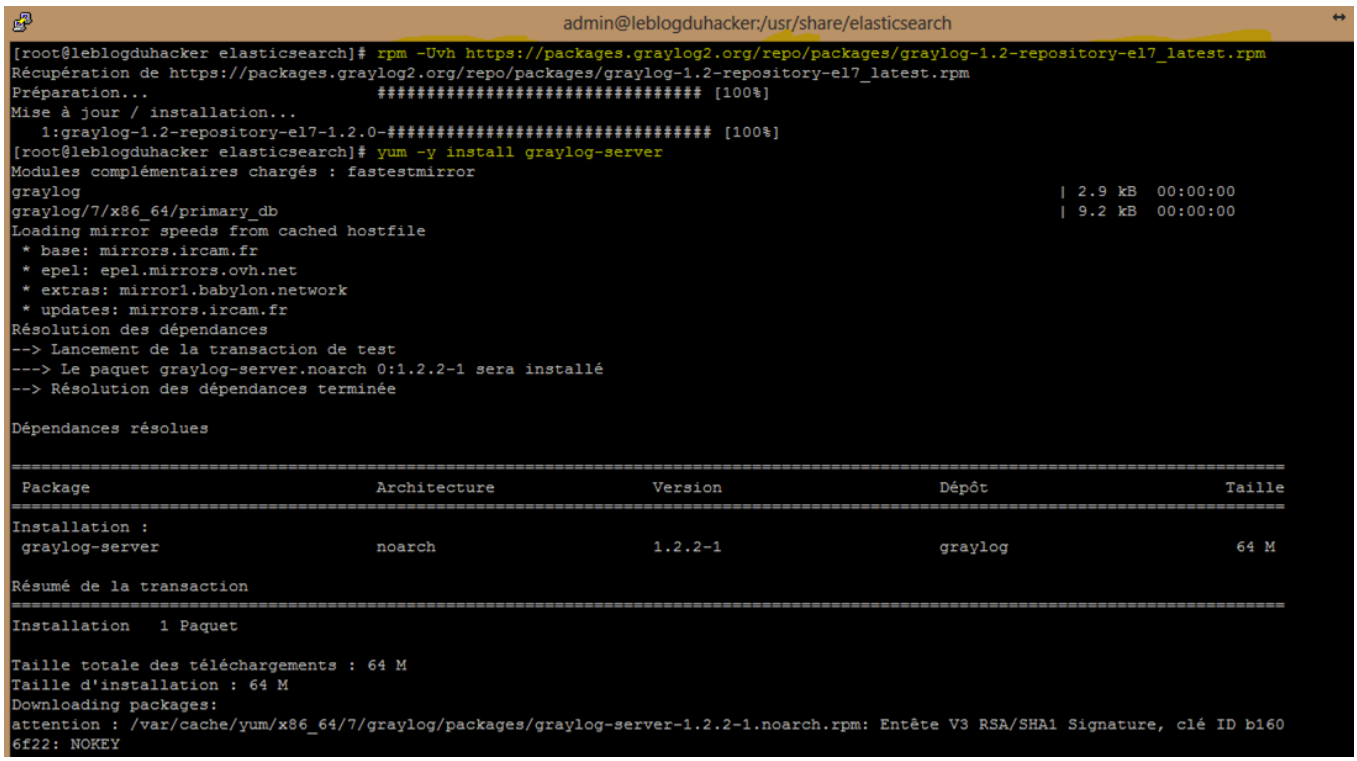
-Création du dépôt:

rpm -Uvh https://packages.graylog2.org/repo/packages/graylog-1.2-repository-el7_latest.rpm

Installation de la dernière version du serveur GRAYLOG:

yum -y install graylog-server*

Voici le résultat que vous devriez obtenir:



Installation du générateur de mot de passe utilisateurs Graylog et création du mot de passe utilisateur Graylog: '/etc/graylog/server/server.conf'

yum install -y pwgen

Génération d'un mot de passe sécurisé pour Graylog à copier ensuite dans son fichier de configuration '

pwgen -N 1 -s 96

```
[root@leblogduhacker elasticsearch]# yum -y install pwgen
Modules complémentaires chargés : fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.ircam.fr
 * epel: mirrors.ircam.fr
 * extras: mirror1.babylon.network
 * updates: mirrors.ircam.fr
Résolution des dépendances
--> Lancement de la transaction de test
---> Le paquet pwgen.x86_64 0:2.07-1.el7 sera installé
--> Résolution des dépendances terminée

Dépendances résolues

================================================================================================================================================
 Package                        Architecture                    Version                              Dépôt                            Taille
================================================================================================================================================
Installation :
 pwgen                          x86_64                          2.07-1.el7                           epel                             24 k

Résumé de la transaction
================================================================================================================================================
Installation   1 Paquet

Taille totale des téléchargements : 24 k
Taille d'installation : 37 k
Downloading packages:
attention : /var/cache/yum/x86_64/7/epel/packages/pwgen-2.07-1.el7.x86_64.rpm: Entête V3 RSA/SHA256 Signature, clé ID 352c64e5: NOKEYB  --:--:-- ETA
La clé publique pour pwgen-2.07-1.el7.x86_64.rpm n'est pas installée
pwgen-2.07-1.el7.x86_64.rpm                                                                                 |  24 kB  00:00:00
Récupération de la clé à partir de file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Importation de la clef GPG 0x352C64E5 :
ID utilisateur : « Fedora EPEL (7) <epel@fedoraproject.org> »
Empreinte      : 91e9 7d7c 4a5e 96f1 7f3e 888f 6a2f aea2 352c 64e5
Paquet         : epel-release-7-5.noarch (installed)
Provient de    : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installation : pwgen-2.07-1.el7.x86_64                                                                                        1/1
  Vérification : pwgen-2.07-1.el7.x86_64                                                                                        1/1

Installé :
  pwgen.x86_64 0:2.07-1.el7

Terminé !
[root@leblogduhacker elasticsearch]# pwgen -N 1 -s 96
iizpFmidD7MrsGUHZFWICxE3GxO80filaaAZBwZ0mE9UHHBPSp55LhbIBAEj9XmOzJ6UIRX7b1iBrEgjJmm5rbuxcEUubygY
[root@leblogduhacker elasticsearch]#
```

Toujours dans le fichier de configuration ‘vi /etc/graylog/server/server.conf ‘ créons notre mot de passe utilisateur (ici ‘123456‘) Graylog

```
[root@leblogduhacker elasticsearch]# pwgen -N 1 -s 96
iizpFmidD7MrsGUHZFWICxE3GxO80filaaAZBwZ0mE9UHHBPSp55LhbIBAEj9XmOzJ6UIRX7b1iBrEgjJmm5rbuxcEUubygY
[root@leblogduhacker elasticsearch]# vi /etc/graylog/server/server.conf
[root@leblogduhacker elasticsearch]# echo -n 123456 | sha256sum
8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92  -
[root@leblogduhacker elasticsearch]# vi /etc/graylog/server/server.conf
[root@leblogduhacker elasticsearch]#
```

Editer le fichier de configuration Graylog et copier les deux mots de passe:

vi /etc/graylog/server/server.conf

Voici le résultat que vous devrez obtenir dans votre fichier de configuration **‘server.conf’**:

```
# this must be the same as for your Elasticsearch cluster
elasticsearch_cluster_name = elasticsearch

# you could also leave this out, but makes it easier to identify the graylog2 client instance
elasticsearch_node_name = Graylog

# we don't want the graylog2 server to store any data, or be master node
#elasticsearch_node_master = false
#elasticsearch_node_data = false

# use a different port if you run multiple Elasticsearch nodes on one machine
#elasticsearch_transport_tcp_port = 9350

# we don't need to run the embedded HTTP server here
elasticsearch_http_enabled = false

#elasticsearch_discovery_zen_ping_multicast_enabled = false
elasticsearch_discovery_zen_ping_unicast_hosts = 172.31.33.5:9300

# Change the following setting if you are running into problems with timeouts during Elasticsearch cluster discovery.
# The setting is specified in milliseconds, the default is 5000ms (5 seconds).
elasticsearch_cluster_discovery_timeout = 15000

# the following settings allow to change the bind addresses for the Elasticsearch client in graylog2
# these settings are empty by default, letting Elasticsearch choose automatically,
# override them here or in the 'elasticsearch_config_file' if you need to bind to a special address
# refer to http://www.elasticsearch.org/guide/en/elasticsearch/reference/0.90/modules-network.html
# for special values here
#elasticsearch_network_host =
#elasticsearch_network_bind_host =
#elasticsearch_network_publish_host =

# The total amount of time discovery will look for other Elasticsearch nodes in the cluster
# before giving up and declaring the current node master.
#elasticsearch_discovery_initial_state_timeout = 3s

# Analyzer (tokenizer) to use for message and full_message field. The "standard" filter usually is a good idea.
# All supported analyzers are: standard, simple, whitespace, stop, keyword, pattern, language, snowball, custom
# Elasticsearch documentation: http://www.elasticsearch.org/guide/reference/index-modules/analysis/
# Note that this setting only takes effect on newly created indices.
elasticsearch_analyzer = standard

# Global request timeout for Elasticsearch requests (e. g. during search, index creation, or index time-range
# calculations) based on a best-effort to restrict the runtime of Elasticsearch operations.
# Default: 1m
#elasticsearch_request_timeout = 1m

# Batch size for the Elasticsearch output. This is the maximum (!) number of messages the Elasticsearch output
# module will get at once and write to Elasticsearch in a batch call. If the configured batch size has not been
# reached within output_flush_interval seconds, everything that is available will be flushed at once. Remember
# that every outputbuffer processor manages its own batch and performs its own batch write calls.
# ("outputbuffer_processors" variable)
output_batch_size = 500

# Flush if this many seconds pass since the last flush, even if output_batch_size has not been reached yet.
output_flush_interval = 1

# The number of parallel running processors.
# Raise this number if your buffers are filling up.
processbuffer_processors = 5
outputbuffer_processors = 3

#outputbuffer_processor_keep_alive_time = 5000
#outputbuffer_processor_threads_core_pool_size = 3
#outputbuffer_processor_threads_max_pool_size = 30

# UDP receive buffer size for all message inputs (e. g. SyslogUDPInput).
#udp_recvbuffer_sizes = 1048576

# Wait strategy describing how buffer processors wait on a cursor sequence. (default: sleeping)
# Possible types:
#  - yielding
#     Compromise between performance and CPU usage.
#  - sleeping
#     Compromise between performance and CPU usage. Latency spikes can occur after quiet periods.
#  - blocking
#     High throughput, low latency, higher CPU usage.
#  - busy_spinning
#     Avoids syscalls which could introduce latency jitter. Best when threads can be bound to specific CPU cores.
processor_wait_strategy = blocking

# Size of internal ring buffers. Raise this if raising outputbuffer_processors does not help anymore.
# For optimum performance your LogMessage objects in the ring buffer should fit in your CPU L3 cache.
# Start server with --statistics flag to see buffer utilization.
# Must be a power of 2. (512, 1024, 2048, ...)
ring_size = 65536

inputbuffer_ring_size = 65536
inputbuffer_processors = 2
inputbuffer_wait_strategy = blocking

# Enable the disk based message journal.
message_journal_enabled = true

# The directory which will be used to store the message journal. The directory must me exclusively used for Graylog and
# must not contain any other files than the ones created by Graylog itself.
message_journal_dir = /var/lib/graylog-server/journal

# Journal hold messages before they could be written to Elasticsearch.
# For a maximum of 12 hours or 5 GB whichever happens first.
# During normal operation the journal will be smaller.
#message_journal_max_age = 12h
#message_journal_max_size = 5gb
#message_journal_flush_age = 1m
#message_journal_flush_interval = 1000000
#message_journal_segment_age = 1h
#message_journal_segment_size = 100mb

# Number of threads used exclusively for dispatching internal events. Default is 2.
#async_eventbus_processors = 2
```

```
# releases.
dead_letters_enabled = false

# How many seconds to wait between marking node as DEAD for possible load balancers and starting the actual
# shutdown process. Set to 0 if you have no status checking load balancers in front.
lb_recognition_period_seconds = 3

# Every message is matched against the configured streams and it can happen that a stream contains rules which
# take an unusual amount of time to run, for example if its using regular expressions that perform excessive backtracking.
# This will impact the processing of the entire server. To keep such misbehaving stream rules from impacting other
# streams, Graylog limits the execution time for each stream.
# The default values are noted below, the timeout is in milliseconds.
# If the stream matching for one stream took longer than the timeout value, and this happened more than "max_faults" times
# that stream is disabled and a notification is shown in the web interface.
#stream_processing_timeout = 2000
#stream_processing_max_faults = 3

# Length of the interval in seconds in which the alert conditions for all streams should be checked
# and alarms are being sent.
#alert_check_interval = 60

# Since 0.21 the graylog2 server supports pluggable output modules. This means a single message can be written to multiple
# outputs. The next setting defines the timeout for a single output module, including the default output module where all
# messages end up.
#
# Time in milliseconds to wait for all message outputs to finish writing a single message.
#output_module_timeout = 10000

# Time in milliseconds after which a detected stale master node is being rechecked on startup.
#stale_master_timeout = 2000

# Time in milliseconds which Graylog is waiting for all threads to stop on shutdown.
#shutdown_timeout = 30000

mongodb_useauth = false
# MongoDB connection string
# See http://docs.mongodb.org/manual/reference/connection-string/ for details
mongodb_uri = mongodb://127.0.0.1:27017/graylog2

# Authenticate against the MongoDB server
#mongodb_uri = mongodb://grayloguser:secret@localhost:27017/graylog2

# Use a replica set instead of a single host
#mongodb_uri = mongodb://grayloguser:secret@localhost:27017,localhost:27018,localhost:27019/graylog2

# Increase this value according to the maximum connections your MongoDB server can handle from a single client
# if you encounter MongoDB connection problems.
mongodb_max_connections = 100

# Number of threads allowed to be blocked by MongoDB connections multiplier. Default: 5
# If mongodb_max_connections is 100, and mongodb_threads_allowed_to_block_multiplier is 5,
# then 500 threads can block. More than that and an exception will be thrown.
# http://api.mongodb.org/java/current/com/mongodb/MongoOptions.html#threadsAllowedToBlockForConnectionMultiplier
mongodb_threads_allowed_to_block_multiplier = 5
#stale_master_timeout = 2000

# Time in milliseconds which Graylog is waiting for all threads to stop on shutdown.
#shutdown_timeout = 30000

mongodb_useauth = false
# MongoDB connection string
# See http://docs.mongodb.org/manual/reference/connection-string/ for details
mongodb_uri = mongodb://127.0.0.1:27017/graylog2

# Authenticate against the MongoDB server
#mongodb_uri = mongodb://grayloguser:secret@localhost:27017/graylog2

# Use a replica set instead of a single host
#mongodb_uri = mongodb://grayloguser:secret@localhost:27017,localhost:27018,localhost:27019/graylog2

# Increase this value according to the maximum connections your MongoDB server can handle from a single client
# if you encounter MongoDB connection problems.
mongodb_max_connections = 100

# Number of threads allowed to be blocked by MongoDB connections multiplier. Default: 5
# If mongodb_max_connections is 100, and mongodb_threads_allowed_to_block_multiplier is 5,
# then 500 threads can block. More than that and an exception will be thrown.
# http://api.mongodb.org/java/current/com/mongodb/MongoOptions.html#threadsAllowedToBlockForConnectionMultiplier
mongodb_threads_allowed_to_block_multiplier = 5

# Drools Rule File (Use to rewrite incoming log messages)
# See: https://www.graylog.org/documentation/general/rewriting/
#rules_file = /etc/graylog/server/rules.drl

# Email transport
transport_email_enabled = true
transport_email_protocol = smtp
transport_email_hostname = messagerie.ac-versailles.fr
transport_email_port = 25
transport_email_use_auth = false
transport_email_use_tls = false
transport_email_use_ssl = false
#transport_email_auth_username = you@example.com
#transport_email_auth_password = secret
transport_email_subject_prefix = ac-versailles.fr
transport_email_from_email = dsi-systeme@ac-versailles.fr

# Specify and uncomment this if you want to include links to the stream in your stream alert mails.
# This should define the fully qualified base url to your web interface exactly the same way as it is accessed by your users.
transport_email_web_interface_url = http://graylog.in.ac-versailles.fr:9000

# The default connect timeout for outgoing HTTP connections.
```

VI- Installation de l'interface web graylog

yum -y install graylog-web

Edition du fichier de configuration '**/etc/graylog/web/web.conf** ':

vi /etc/graylog/web/web.conf

```
# graylog2-server REST URIs (one or more, comma separated) For example: "http://127.0.0.1:12900/,http://127.0.0.1:12910/,"
graylog2-server.uris="http://172.31.33.5:12900/,"

# Learn how to configure custom logging in the documentation:
#    http://docs.graylog.org/en/latest/pages/installation.html#manual-setup-graylog-web-interface-on-linux

# Secret key
# ~~~~~
# The secret key is used to secure cryptographics functions. Set this to a long and randomly generated string.
# If you deploy your application to several instances be sure to use the same key!
# Generate for example with: pwgen -N 1 -s 96
application.secret="4tZlNfZ3jhtxlDuqZc9wZYwAs4EsTzCo9yzA7YK0KjQ78F4EafCejSSldEjL2ADmoZOwzHajJzxGjL7Ts1Dp7bfGxwg8RN92"

# Web interface timezone
# Graylog stores all timestamps in UTC. To properly display times, set the default timezone of the interface.
# If you leave this out, Graylog will pick your system default as the timezone. Usually you will want to configure it explicitly.
# timezone="Europe/Berlin"

# Message field limit
# Your web interface can cause high load in your browser when you have a lot of different message fields. The default
# limit of message fields is 100. Set it to 0 if you always want to get all fields. They are for example used in the
# search result sidebar or for autocompletion of field names.
field_list_limit=100

# Use this to run Graylog with a path prefix
#application.context=/graylog2

# You usually do not want to change this.
application.global=lib.Global

# Global timeout for communication with Graylog server nodes; default: 5s
#timeout.DEFAULT=5s

# Accept any server certificate without checking for validity; required if using self-signed certificates.
# Default: true
# graylog2.client.accept-any-certificate=true
~
~
~
```

Relancer l'interface Web Graylog:

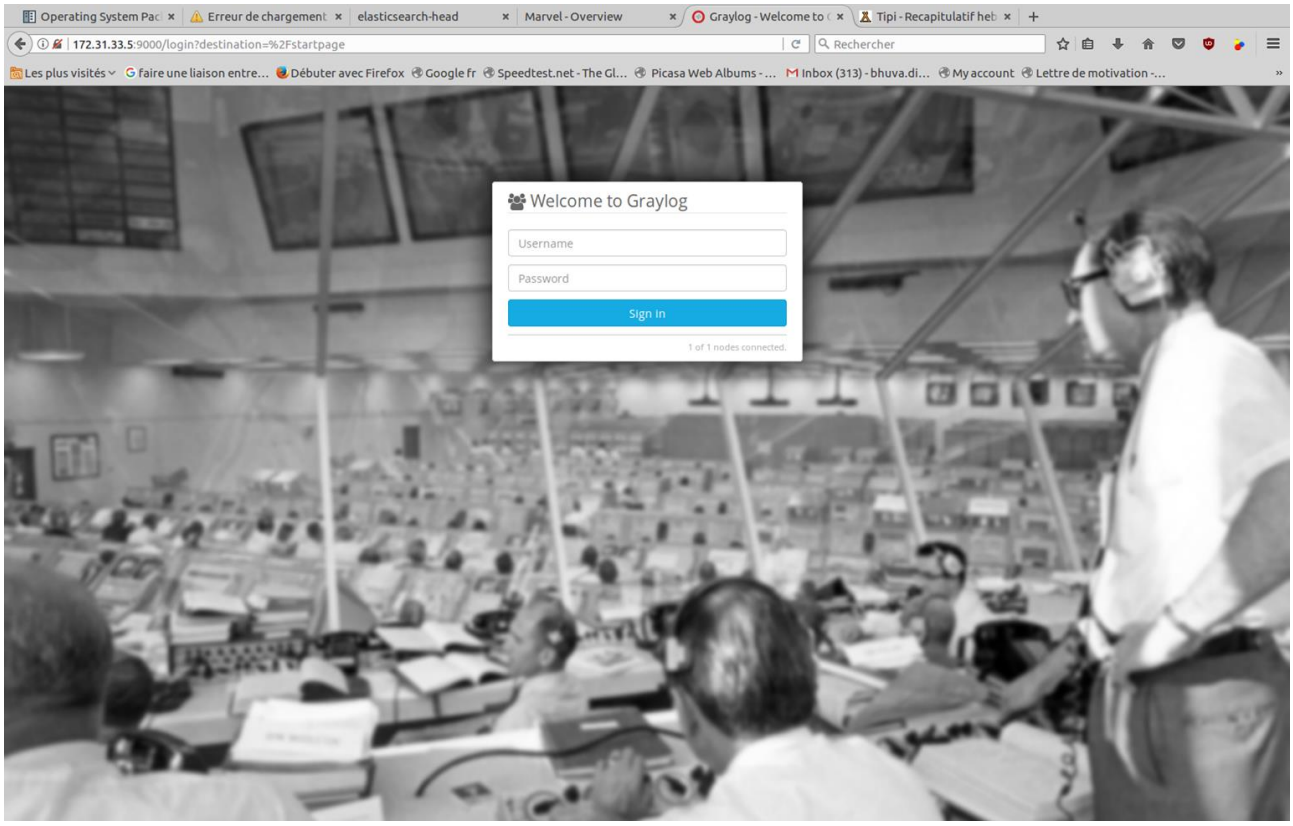
systemctl restart graylog-web

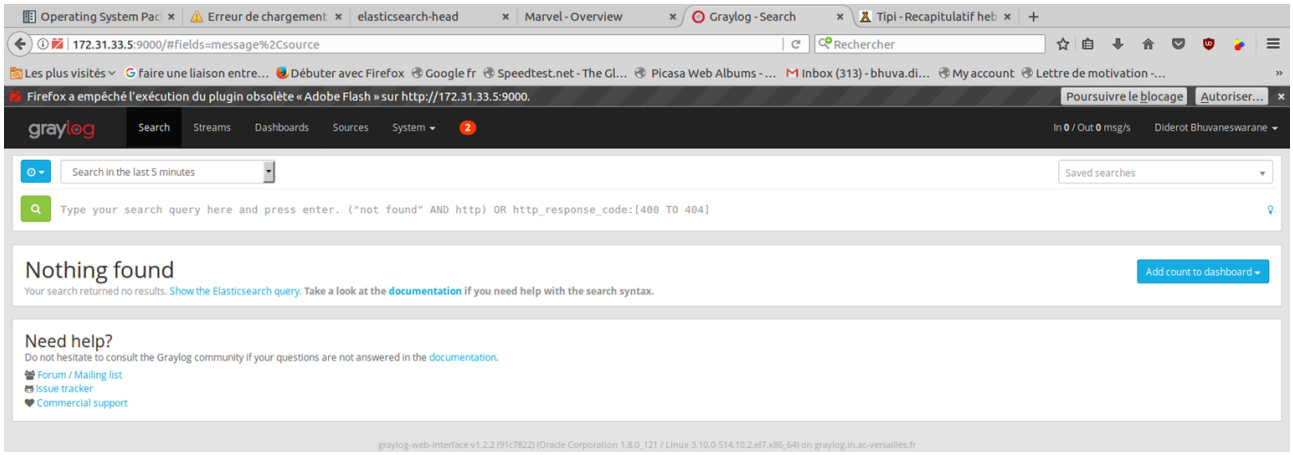
L'interface est configurée pour écouter le port 9000, il faut donc paramétrer le Firewall pour autoriser le trafic sur ce port (9000):

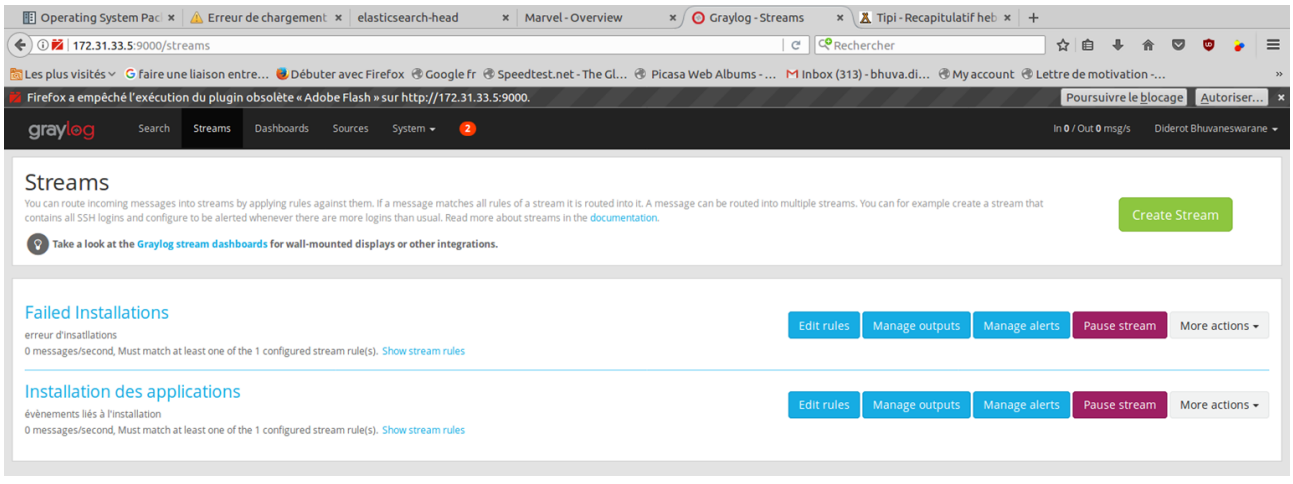
firewall-cmd –permanent –zone=public –add-port=9000/tcp

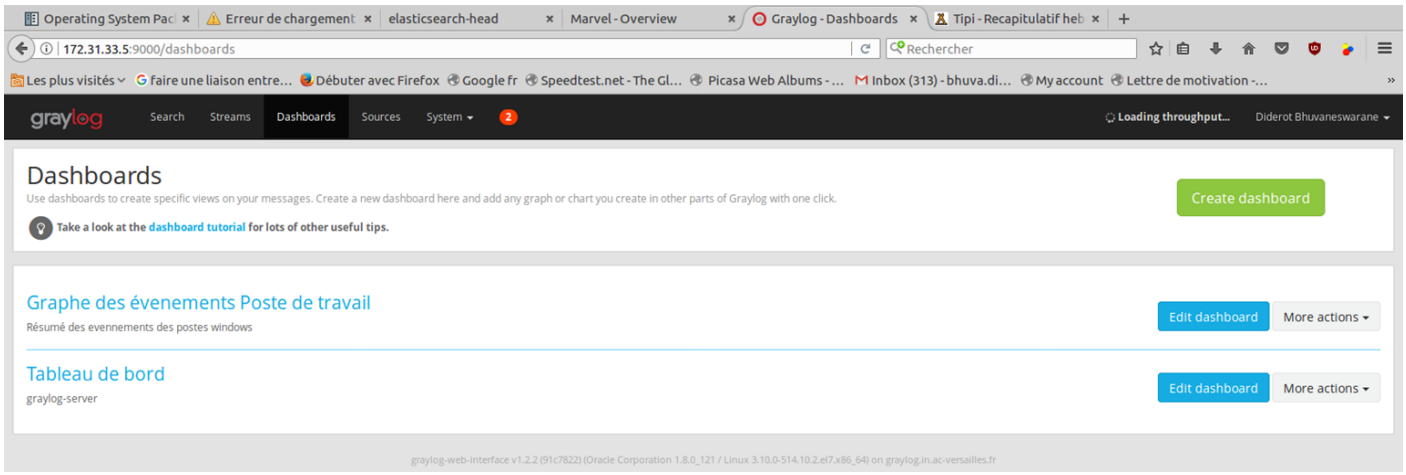
firewall-cmd –reload

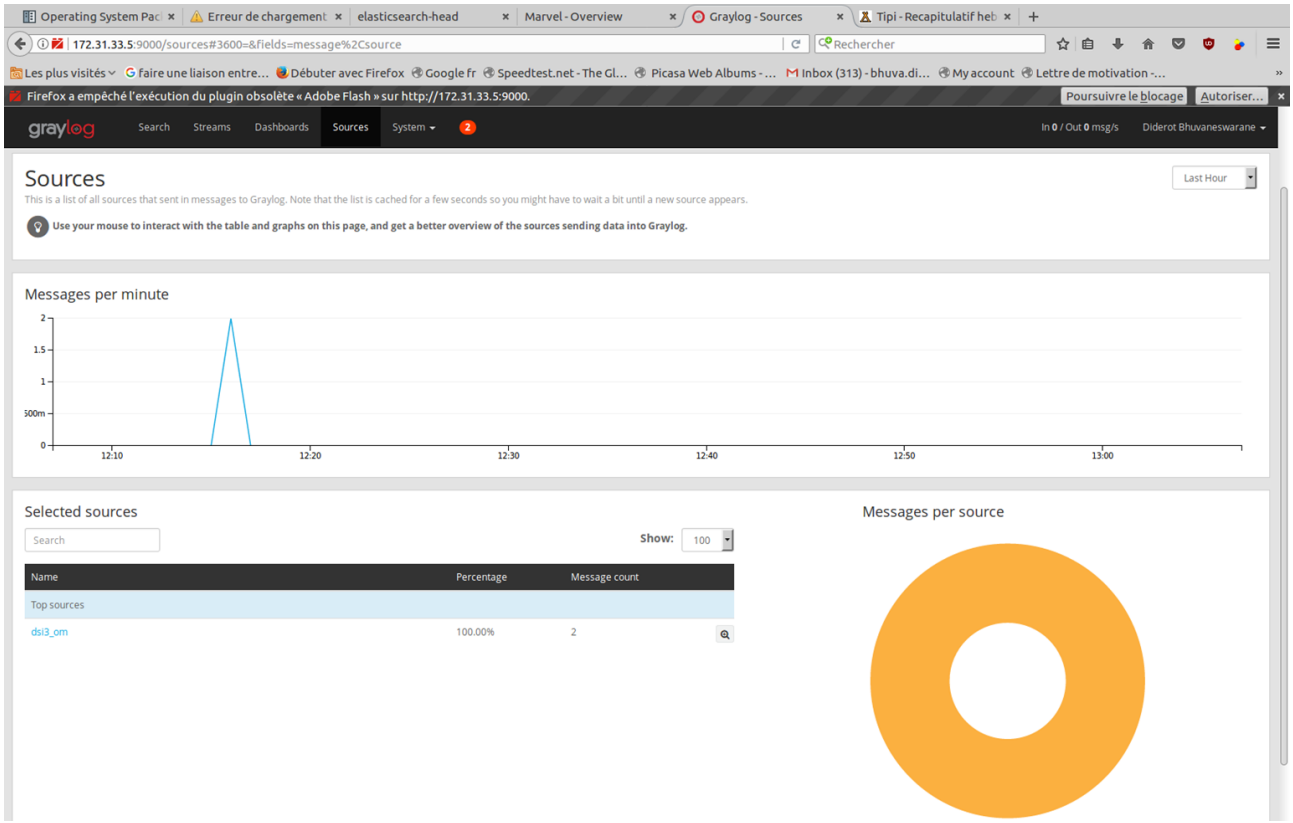
Ouverture du navigateur pour se connecter à l'adresse IP sur server Graylog en utilisant l'identifiant par défaut «**admin**» et notre mot de passe (ici **123456**) configuré dans le fichier de configuration.

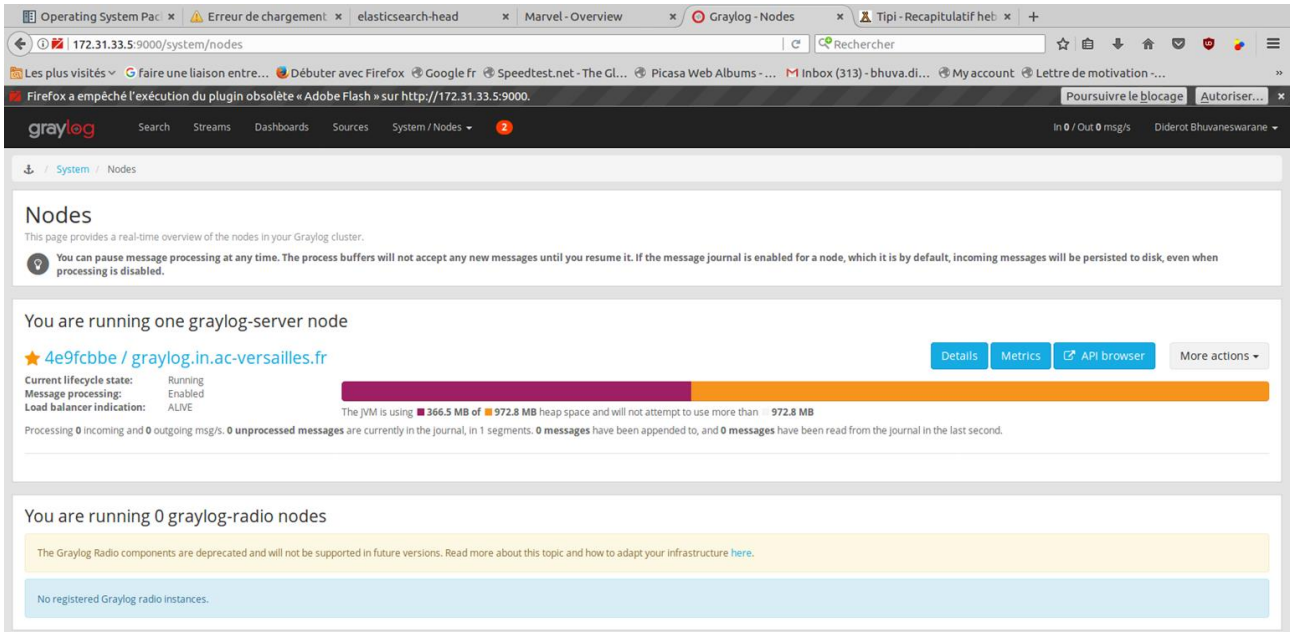

Operating System Pac
Erreur de chargement
elasticsearch-head
Marvel - Overview
Graylog - Welcome to
Tipi - Recapitulatif heb
172.31.33.5:9000/login?destination=%2Fstartpage
Rechercher
Les plus visités
faire une liaison entre...
Débuter avec Firefox
Google fr
Speedtest.net - The Gl...
Picasa Web Albums - ...
Inbox (313) - bhuva.di...
My account
Lettre de motivation -...
Welcome to Graylog
Username
Password
Sign in
1 of 1 nodes connected.


Operating System Pac
Erreur de chargement
elasticsearch-head
Marvel - Overview
Graylog - Search
Tipi - Recapitulatif heb
172.31.33.5:9000/#fields=message%2Csource
Rechercher
Les plus visités
faire une liaison entre...
Débuter avec Firefox
Google fr
Speedtest.net - The Gl...
Picasa Web Albums - ...
Inbox (313) - bhuva.di...
My account
Lettre de motivation -...
Firefox a empêché l'exécution du plugin obsolète « Adobe Flash » sur http://172.31.33.5:9000.
Poursuivre le blocage
Autoriser...
graylog
Search
Streams
Dashboards
Sources
System
2
In 0 / Out 0 msg/s
Diderot Bhuvaneswarane
Search in the last 5 minutes
Saved searches
Type your search query here and press enter. ("not found" AND http) OR http_response_code:[400 TO 404]
Nothing found
Your search returned no results. Show the Elasticsearch query. Take a look at the documentation if you need help with the search syntax.
Add count to dashboard
Need help?
Do not hesitate to consult the Graylog community if your questions are not answered in the documentation.
Forum / Mailing list
Issue tracker
Commercial support
graylog-web-interface v1.2.2 (91c7822) (Oracle Corporation 1.8.0_121 / Linux 3.10.0-514.10.2.el7.x86_64) on graylog.in.ac-versailles.fr

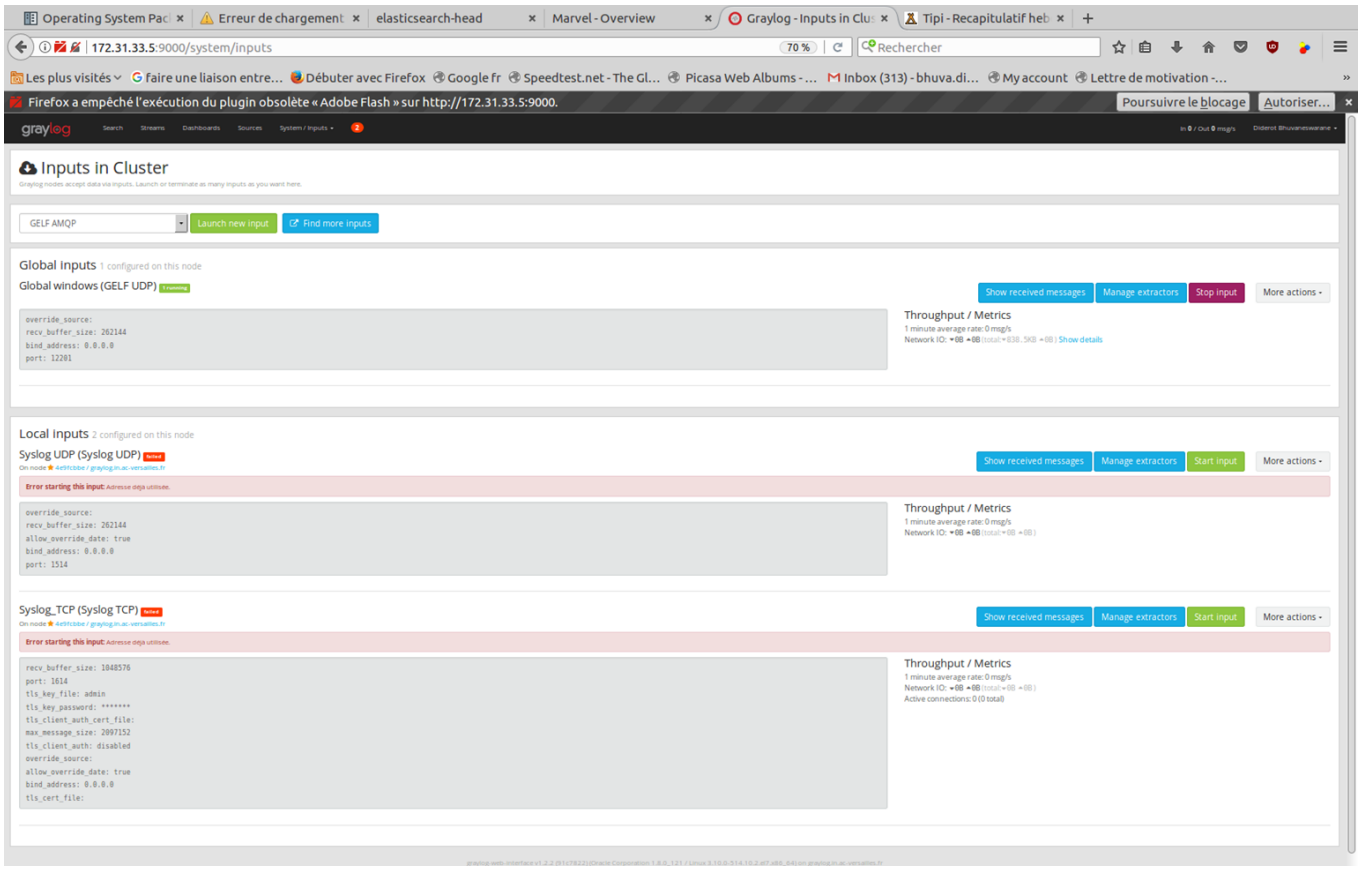

Operating System Pac
Erreur de chargement
elasticsearch-head
Marvel - Overview
Graylog - Streams
Tipi - Recapitulatif heb
172.31.33.5:9000/streams
Rechercher
Les plus visités
faire une liaison entre...
Débuter avec Firefox
Google fr
Speedtest.net - The Gl...
Picasa Web Albums - ...
Inbox (313) - bhuva.di...
My account
Lettre de motivation -...
Firefox a empêché l'exécution du plugin obsolète « Adobe Flash » sur http://172.31.33.5:9000.
Poursuivre le blocage
Autoriser...
graylog
Search
Streams
Dashboards
Sources
System
2
In 0 / Out 0 msg/s
Diderot Bhuvaneswarane
Streams
You can route incoming messages into streams by applying rules against them. If a message matches all rules of a stream it is routed into it. A message can be routed into multiple streams. You can for example create a stream that contains all SSH logins and configure to be alerted whenever there are more logins than usual. Read more about streams in the documentation.
Create Stream
Take a look at the Graylog stream dashboards for wall-mounted displays or other integrations.
Failed Installations
erreur d'insatllations
0 messages/second, Must match at least one of the 1 configured stream rule(s). Show stream rules
Edit rules
Manage outputs
Manage alerts
Pause stream
More actions
Installation des applications
évènements liés à l'installation
0 messages/second, Must match at least one of the 1 configured stream rule(s). Show stream rules
Edit rules
Manage outputs
Manage alerts
Pause stream
More actions


Operating System Pac
Erreur de chargement
elasticsearch-head
Marvel - Overview
Graylog - Dashboards
Tipi - Recapitulatif heb
172.31.33.5:9000/dashboards
Rechercher
Les plus visités
faire une liaison entre...
Débuter avec Firefox
Google fr
Speedtest.net - The Gl...
Picasa Web Albums - ...
Inbox (313) - bhuva.di...
My account
Lettre de motivation -...
graylog
Search
Streams
Dashboards
Sources
System
2
Loading throughput...
Diderot Bhuvaneswarane
Dashboards
Use dashboards to create specific views on your messages. Create a new dashboard here and add any graph or chart you create in other parts of Graylog with one click.
Take a look at the dashboard tutorial for lots of other useful tips.
Create dashboard
Graphe des évenements Poste de travail
Résumé des evennements des postes windows
Edit dashboard
More actions
Tableau de bord
graylog-server
Edit dashboard
More actions
graylog-web-interface v1.2.2 (91c7822) (Oracle Corporation 1.8.0_121 / Linux 3.10.0-514.10.2.el7.x86_64) on graylog.in.ac-versailles.fr

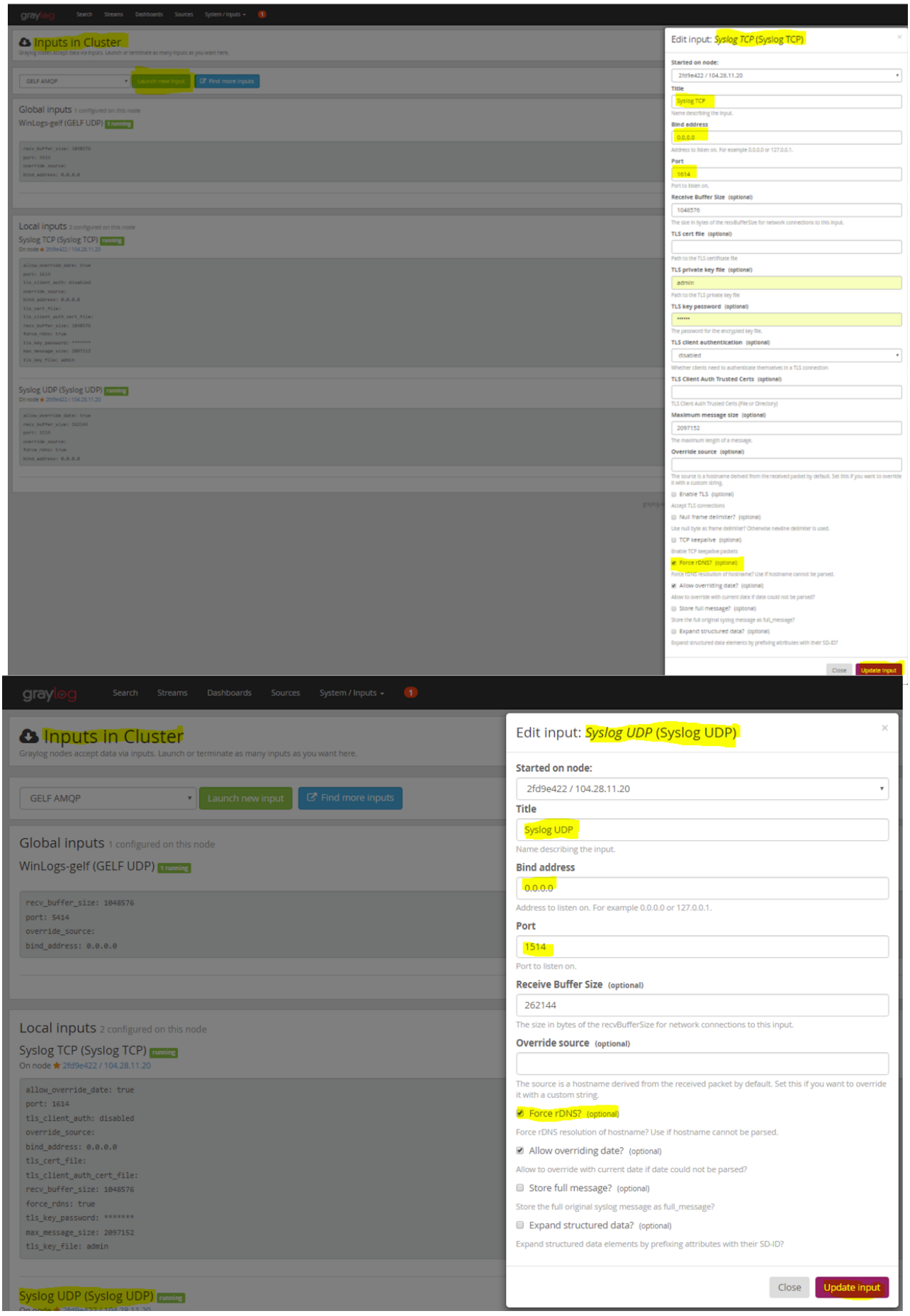

Operating System Pac
Erreur de chargement
elasticsearch-head
Marvel - Overview
Graylog - Sources
Tipi - Recapitulatif heb
172.31.33.5:9000/sources#3600=&fields=message%2Csource
Rechercher
Les plus visités
faire une liaison entre...
Débuter avec Firefox
Google fr
Speedtest.net - The Gl...
Picasa Web Albums - ...
Inbox (313) - bhuva.di...
My account
Lettre de motivation -...
Firefox a empêché l'exécution du plugin obsolète « Adobe Flash » sur http://172.31.33.5:9000.
Poursuivre le blocage
Autoriser...
graylog
Search
Streams
Dashboards
Sources
System
2
In 0 / Out 0 msg/s
Diderot Bhuvaneswarane
Sources
This is a list of all sources that sent in messages to Graylog. Note that the list is cached for a few seconds so you might have to wait a bit until a new source appears.
Use your mouse to interact with the table and graphs on this page, and get a better overview of the sources sending data into Graylog.
Last Hour
Messages per minute
2
1.5
1
500m
0
12:10
12:20
12:30
12:40
12:50
13:00
Selected sources
Search
Show:
100
Name
Percentage
Message count
Top sources
dsi3_om
100.00%
2
Messages per source



VII- Paramétrage d'un écouteur (Input) sur Graylog



Configuration d’un écouteur (Input Syslog UDP port 1514 et TCP port 1614) via Graylog-Web:



VIII- Fichier de configuration Nxlog

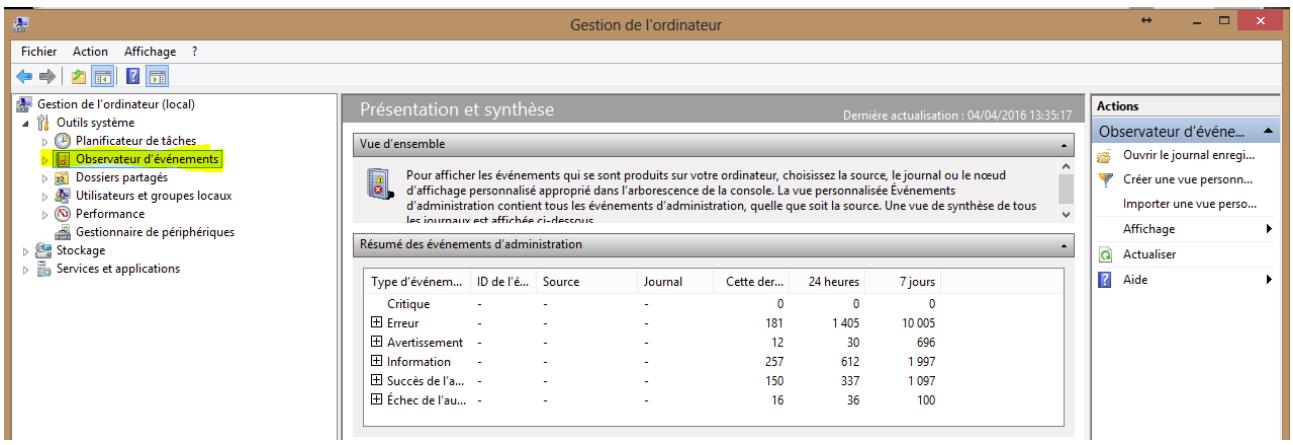
Maintenant nous allons aborder les interfaces Nxlog (Windows) et Rsyslog (graylog). Effectivement, pour que le Centralisateur Graylog puisse centraliser les logs en provenance du Serveur «graylog» et des Eventlogs de Windows (machine physique), il nous faut configurer quelques éléments d'information sur leur fichier de configuration.

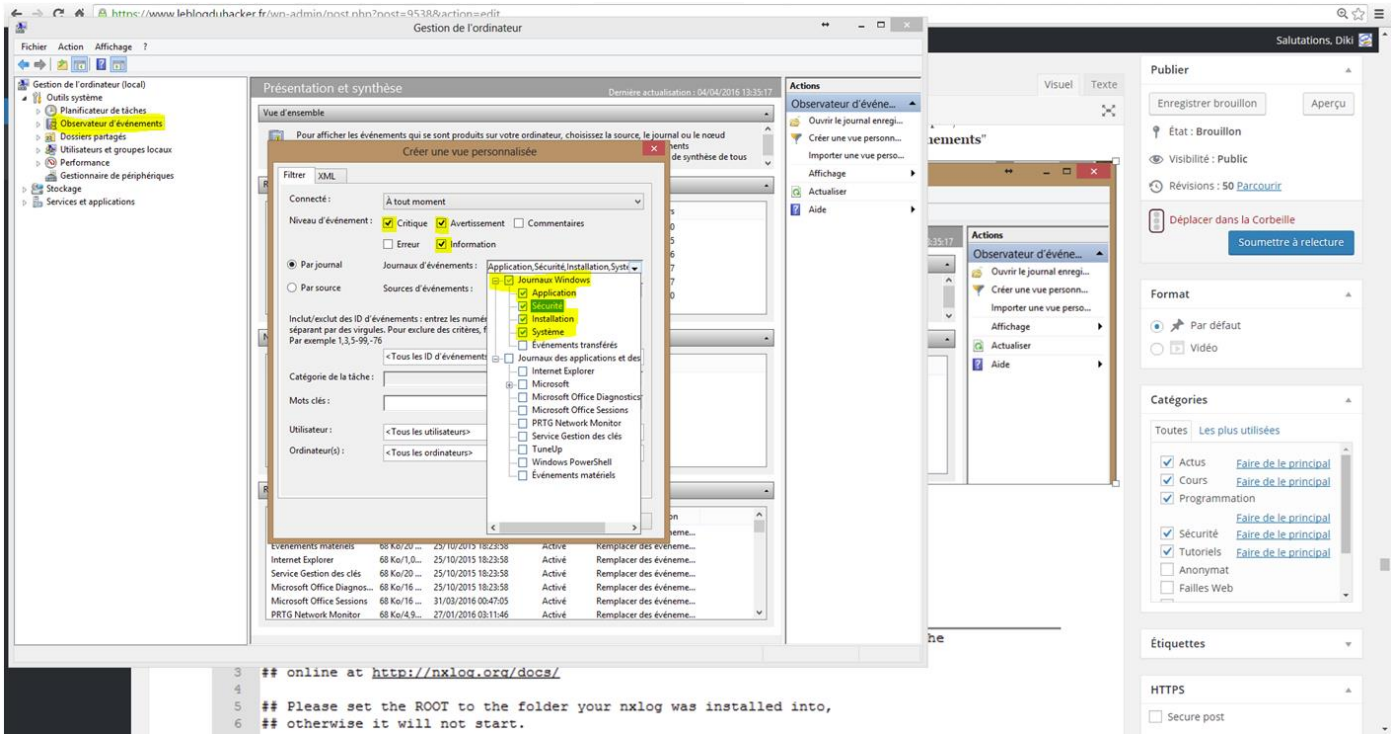
Débutons par Nxlog

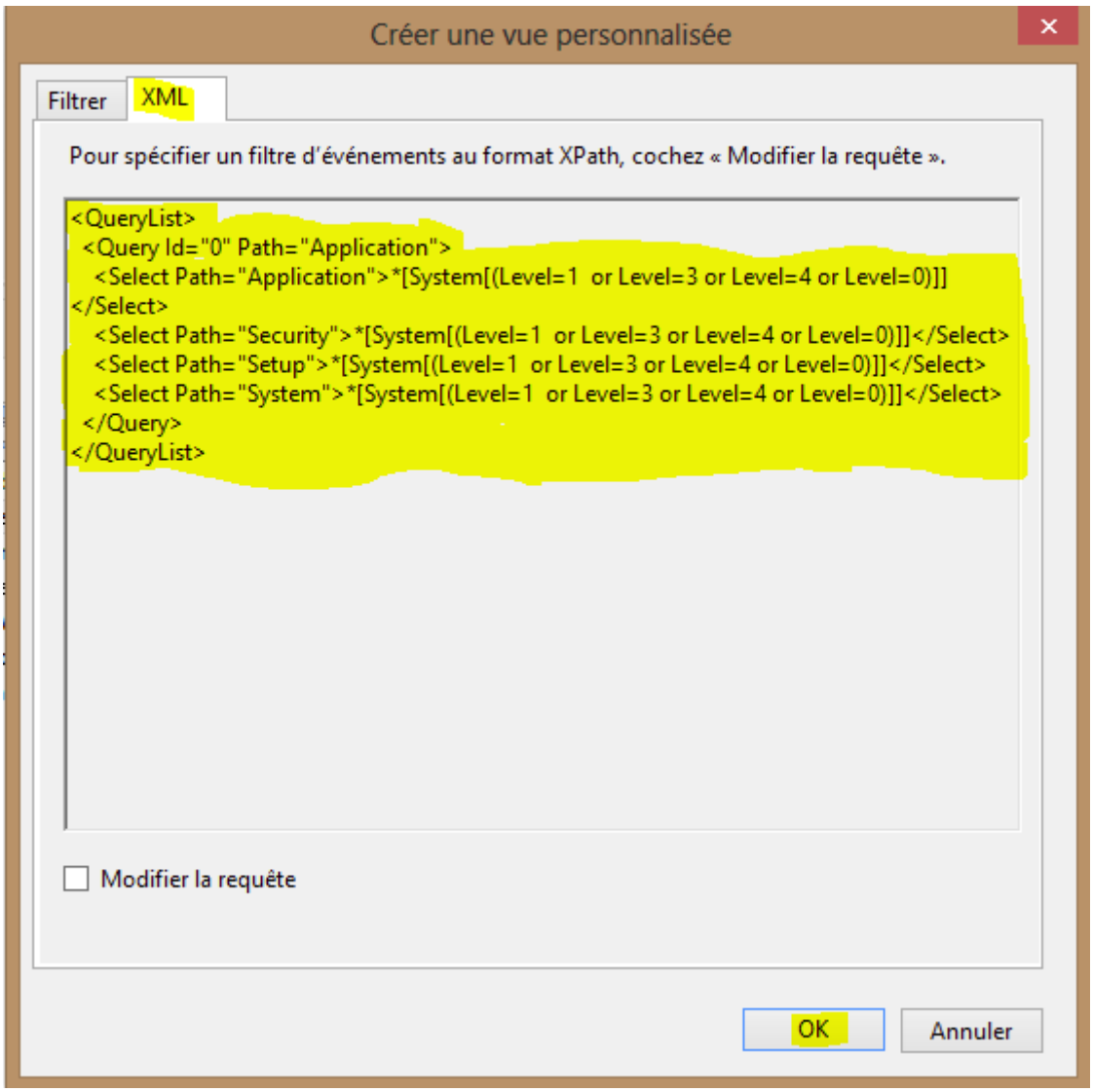
Nous n'allons pas rentrer dans le détail du fichier de configuration Nxlog , mais sachez que c'est ici que nous définissons le comportement du Client Nxlog (Windows) pour collecter les Eventlogs Windows qui seront ensuite convertis au format de communication «Syslog» (Standard) et «GELF» (Graylog) et finalement renvoyés vers le Centralisateur Graylog. Nxlog va simplement utiliser des modules prédéfinis, en l'occurrence «xm_syslog», «im_msvistalog» et «xm_gelf» qui vont permettre aux entrées «input» de traiter les logs (journaux d'événements Windows) en provenance de la machine physique HOST (ici: 192.168.1.31) au bon format de communication qui seront ensuite convertis (logs) via les sorties («out») qui indiqueront quelle «route» (chemin) suivre pour être finalement centralisés par Graylog . Un processus en informatique suit toujours la même séquence de traitement suivante: «Entrée» => «Processus» => «Sortie».

De quels événements Windows avons-nous besoin ?

L'idée ici est de générer le bon script d'événements Windows grâce à l'observateur d'événements Windows. Nous n'aurons ainsi plus qu'à copier/coller ce script dans le fichier de configuration Nxlog. Pour ce faire, rendez-vous dans «**Panneau de Configuration / Outils d'administration / Observateur d'événements»**, cliquez droit sur «**Observateur d'événements / Créer une vue personnalisée»**, dans l'onglet «**Filtrer**» section «**Journal**» sélectionnez les critères des journaux d'événements Windows que vous souhaitez générer (ici: Application, Sécurité, Installation, Système). Validez le nom de votre nouvelle vue puis rendez-vous dans l'onglet «**XML**» pour copier et coller le script (filtre) dont vous avez besoin dans le fichier de configuration '**Nxlog.conf**'.




Gestion de l'ordinateur
Fichier Action Affichage ?
Gestion de l'ordinateur (local)
Outils système
Planificateur de tâches
Observateur d'événements
Dossiers partagés
Utilisateurs et groupes locaux
Performance
Gestionnaire de périphériques
Stockage
Services et applications
Présentation et synthèse
Dernière actualisation : 04/04/2016 13:35:17
Vue d'ensemble
Créer une vue personnalisée
Filtrer
XML
Connecté :
À tout moment
Niveau d'événement :
Critique
Avertissement
Commentaires
Erreur
Information
Par journal
Journaux d'événements :
Par source
Sources d'événements :
Catégorie de la tâche :
Mots clés :
Utilisateur :
<Tous les utilisateurs>
Ordinateur(s) :
<Tous les ordinateurs>
Journaux Windows
Application
Sécurité
Installation
Système
Événements transférés
Journaux des applications et des
Internet Explorer
Microsoft
Microsoft Office Diagnostics
Microsoft Office Sessions
PRTG Network Monitor
Service Gestion des clés
TuneUp
Windows PowerShell
Événements matériels
Actions
Observateur d'événe...
Ouvrir le journal enregi...
Créer une vue personn...
Importer une vue perso...
Affichage
Actualiser
Aide
Salutations, Diki
Visuel
Texte
Publier
Enregistrer brouillon
Aperçu
État : Brouillon
Visibilité : Public
Révisions : 50 Parcourir
Déplacer dans la Corbeille
Soumettre à relecture
Format
Par défaut
Vidéo
Catégories
Toutes
Les plus utilisées
Actus
Cours
Programmation
Sécurité
Tutoriels
Anonymat
Failles Web
Faire de le principal
Étiquettes
HTTPS
Secure post
online at http://nxlog.org/docs/
Please set the ROOT to the folder your nxlog was installed into,
otherwise it will not start.



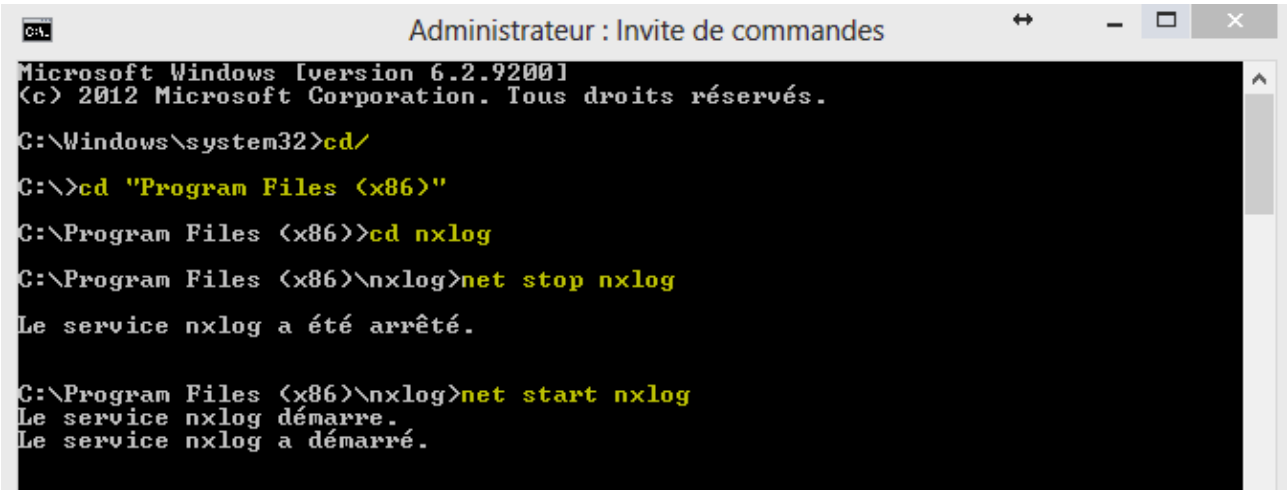
Note importante: avant d'apporter toute modification sur le fichier de configuration ' nxlog.conf ', il est nécessaire de stopper préalablement le service 'nxlog'. Pour ce faire ouvrir un invite de commande Windows (touches clavier: «WINDOWS» + «R», tapez ' cmd ' puis «OK») et saisissez les commandes suivantes:

cd/

cd «Program Files (x86)»

cd nxlog

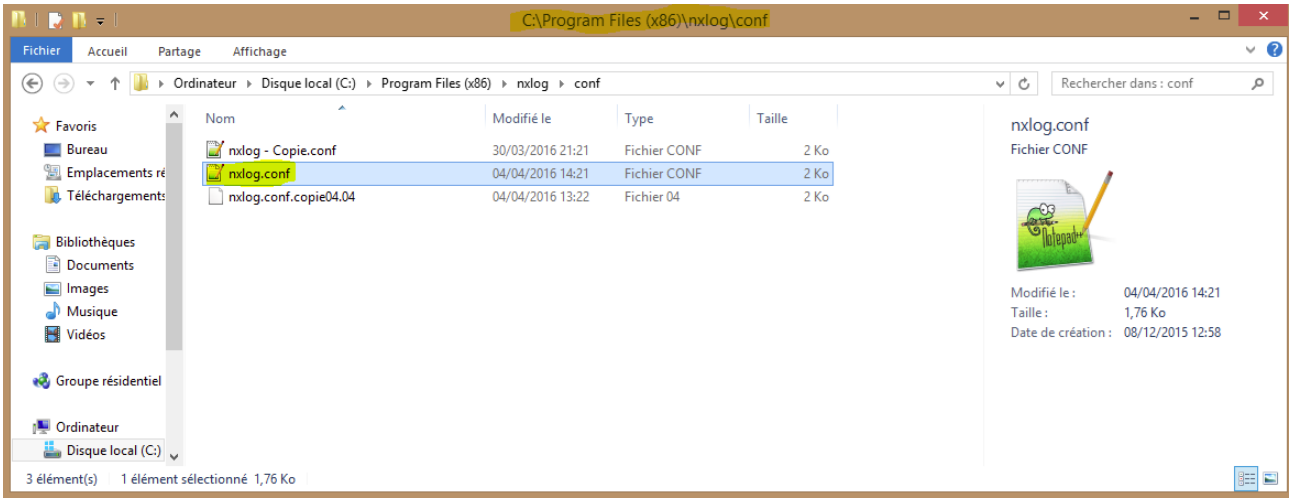
net stop nxlog



Il ne vous reste plus qu'à répercuter le script généré ci-dessus (le votre!) dans le fichier de configuration Nxlog (ci-dessous), plus exactement dans l'entrée intitulée ' in ' par défaut dédiée au format «Syslog» tout en préservant la syntaxe et la structure initiale du fichier 'nxlog.conf'. L'entrée intitulée ici ' ingelf ' sera dédiée au format «GELF» de Graylog. Les sorties (' out ' par défaut) seront définies dans cet exemple avec ' out1', ' out2' 'outgelf' pour les besoins de cet article uniquement. La ' route 'sera donc pour cet extrait:

in => **out1**, **out2**, **outgelf**'

Localisation du fichier de configuration ' **nxlog.conf** ': **C:Program Files (x86)nxlogconfnxlog.conf**



Nxlog traitera donc les «EventLogs Windows» au format «Syslog» (UDP et/ou TCP) et au format «GELF» de Graylog (UDP et/ou TCP).

```
## This is a sample configuration file. See the nxlog reference manual about the
## configuration options. It should be installed locally and is also available
## online at http://nxlog.org/docs/

## Please set the ROOT to the folder your nxlog was installed into,
## otherwise it will not start.

#define ROOT C:\Program Files\nxlog
define ROOT C:\Program Files (x86)\nxlog


Moduledir %ROOT%\modules
CacheDir %ROOT%\data
Pidfile %ROOT%\data\nxlog.pid
SpoolDir %ROOT%\data
LogFile %ROOT%\data\nxlog.log

<Extension _syslog>
    Module      xm_syslog
</Extension>

<Extension gelf>
    Module      xm_gelf
</Extension>


<Input in>
# Pour version OS Windows 2008 et ultérieure:
   Module  im_msvistalog
 Query <QueryList>\
   <Query Id="0">\
            <Select Path="Security">*</Select>\
            <Select Path="System">*[System/Level=4]</Select>\
            <Select Path="Application">*[Application/Level=4]</Select>\
            <Select Path="Setup">*[System/Level=3]</Select>\
            <Select Path="Microsoft-Windows-GroupPolicy/Operational">*</Select>\
            <Select Path='Windows PowerShell'>*</Select>\
   </Query>\
 </QueryList>

 </Input>

#<Input ingelf>
#   Module      im_msvistalog
# Pour version OS Windows 2008 et ultérieure:
#   Module      im_mseventlog
#</Input>


<Output out1>
    Module      om_tcp
    Host        192.168.1.55
    Port        1614
    Exec        to_syslog_snare();
</Output>

<Output out2>
    Module      om_udp
    Host        192.168.1.55
    Port        1514
    Exec        to_syslog_snare();
</Output>

<Output outgelf>
    Module      om_udp
    Host        192.168.1.55
    Port        12201
    OutputType  GELF
</Output>

<Route 1>
    Path        in => out1, out2, outgelf
</Route>

```

IX- Fichier de configuration /etc/rsyslog/ de graylog (redhat)

Le principe de fonctionnement du fichier de configuration Rsyslog est le même que celui de Nxlog. Celui que je vous propose de découvrir ci-dessous est un générique qui pourra bien évidemment s'adapter en fonction de votre besoin, objectif et environnement. Les lignes 9,10, 11, 12, 15 et 19 chargent les modules nécessaires à Rsyslog pour traiter les logs (messages). Il existe d'autres modules que nous évoquerons pas dans cet article. L'idée ici est de demander à Rsyslog d'activer son serveur pour recevoir tout le trafic UDP sur le port 514 et tout le trafic TCP sur le même port 514. Généralement, l'Administrateur choisira un seul protocole de communication soit UDP ou soit TCP. Moi j'ai décidé de faire autrement pour les besoins de cet article. Pourquoi ? Je vais tout simplement demander à Rsyslog de SEGMENTER le trafic UDP et TCP et de le REDIRIGER vers deux autres ports d'écoute du serveur GRAYLOG (plus bas).Les lignes 95 et 96 sont IMPORTANTES puisqu'elles vont permettre de REDIRIGER tous logs du serveur graylog au format UDP et TCP à partir du serveur Rsyslog vers le serveur Graylog (Centralisateur de Logs) comme suit:

Trafic UDP: **@172,31,33,5:1514** (vers le port 1514 écouté par Graylog – un arobas '@' indique que c'est UDP)

Trafic TCP: **@@172,31,33,5:1614** (vers le port 1614 écouté par Graylog – deux arobas '@@' indique que c'est TCP)

Rsyslog traitera donc les logs en provenance du serveur CentOS 7 au format de communication « Syslog » avec soit le protocole TCP et/ou UDP. Dans l'exemple de mon installation, j'ai volontairement appliqué TCP et UDP pour vous montrer le principe de configuration. Généralement nous n'avons besoin que d'un seul protocole de communication soit TCP ou bien UDP. Sachez que les logs d'équipements réseau notamment sont transmis en UDP qui est un format approprié car leurs logs (routeur, commutateur, contrôleur Wi-Fi…) génèrent peu d'information au contraire du format de communication GELF de Graylog qui est beaucoup plus riche et mieux adapté au traitement des Eventlogs de Windows.

vi /etc/rsyslog.conf

```
# For more information see /usr/share/doc/rsyslog-*/rsyslog_conf.html
# If you experience problems, see http://www.rsyslog.com/doc/troubleshoot.html

#### MODULES ####

# The imjournal module bellow is now used as a message source instead of imuxsock.
$ModLoad imuxsock # provides support for local system logging (e.g. via logger command)
$ModLoad imjournal # provides access to the systemd journal
$ModLoad imklog # reads kernel messages (the same are read from journald)
$ModLoad immark  # provides --MARK-- message capability

# Provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 1514

# Provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 1614


#### GLOBAL DIRECTIVES ####

# Where to place auxiliary files
$WorkDirectory /var/lib/rsyslog

# Use default timestamp format
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

# File syncing capability is disabled by default. This feature is usually not required,
# not useful and an extreme performance hit
#$ActionFileEnableSync on

# Include all config files in /etc/rsyslog.d/
$IncludeConfig /etc/rsyslog.d/*.conf

# Turn off message reception via local log socket;
# local messages are retrieved through imjournal now.
$OmitLocalLogging on

# File to store the position in the journal
$IMJournalStateFile imjournal.state


#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog
```

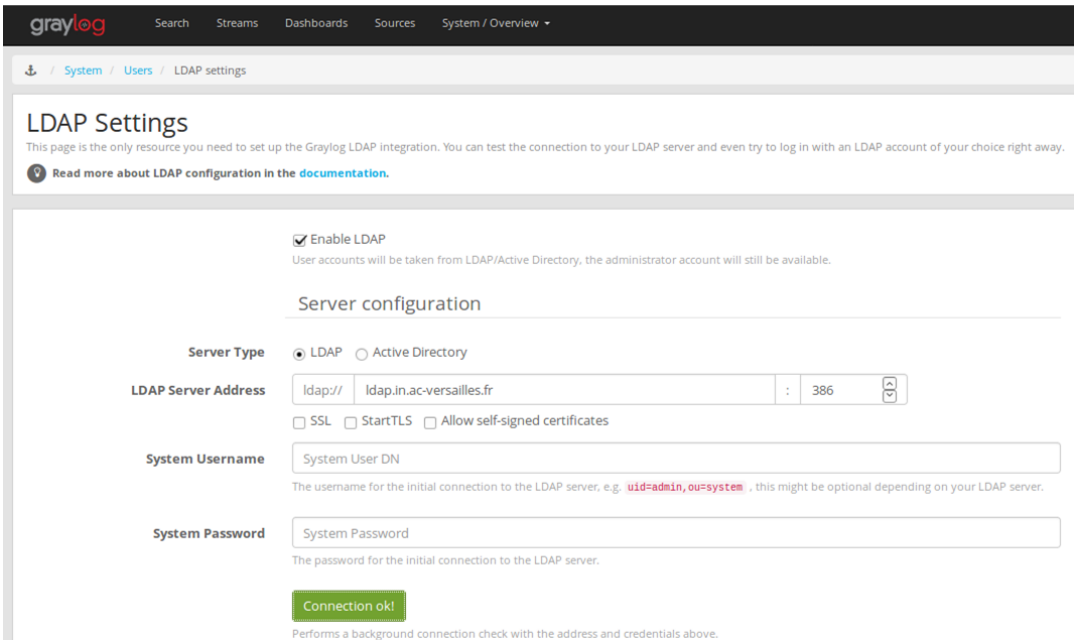
```
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 :omusrmsg:*

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log


# ### begin forwarding rule ###
# The statement between the begin ... end define a SINGLE forwarding
# rule. They belong together, do NOT split them. If you create multiple
# forwarding rules, duplicate the whole block!
# Remote Logging (we use TCP for reliable delivery)
#
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.
#$ActionQueueFileName fwdRule1 # unique name prefix for spool files
#$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as possible)
#$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
#$ActionQueueType LinkedList   # run asynchronously
#$ActionResumeRetryCount -1    # infinite retries if host is down
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
#*.* @@remote-host:514
*.* @@172.31.33.5:1614
*.* @172.31.33.5:1514
# ### end of the forwarding rule ###
#*.*    @172.31.131.4
*.*     @172.31.131.25
*.*     @172.31.131.26
kern.*                                                  /dev/tty6
*.* @172.31.131.25
```

X- Authentification externe avec LDAP

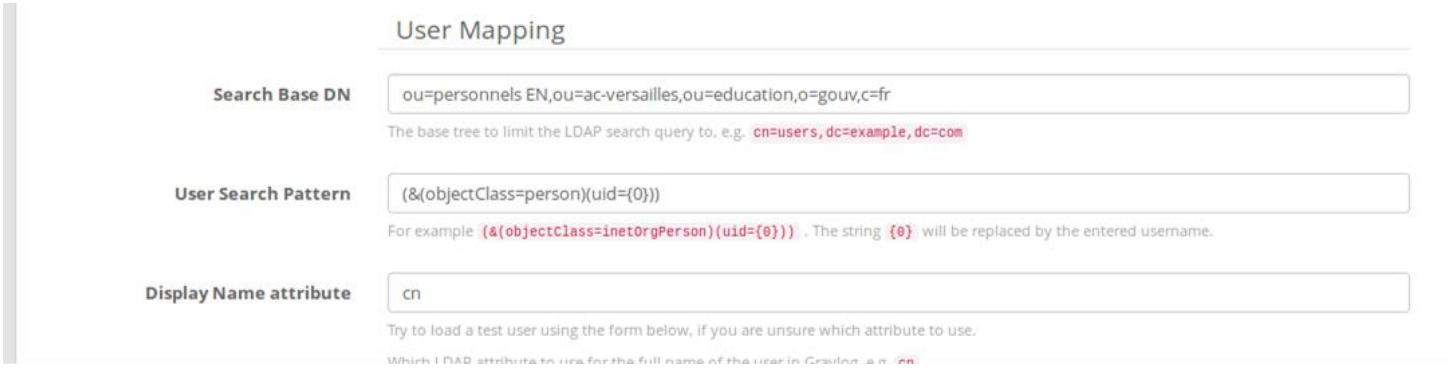
Avec graylog, il est possible de déclarer la connexion à un AD/LDAP pour authentifier les utilisateurs (certains utilisateurs, pas tous).

Pour configurer le LDAP/AD, cliquez sur le bouton « System » → « Users » → « Configure Ldap ». Puis entrez les informations du serveur telle ci-dessous (bien sûr en adaptant les informations). Puis déclarez l'annuaire LDAP sur graylog et vérifiez la connexion du serveur LDAP.

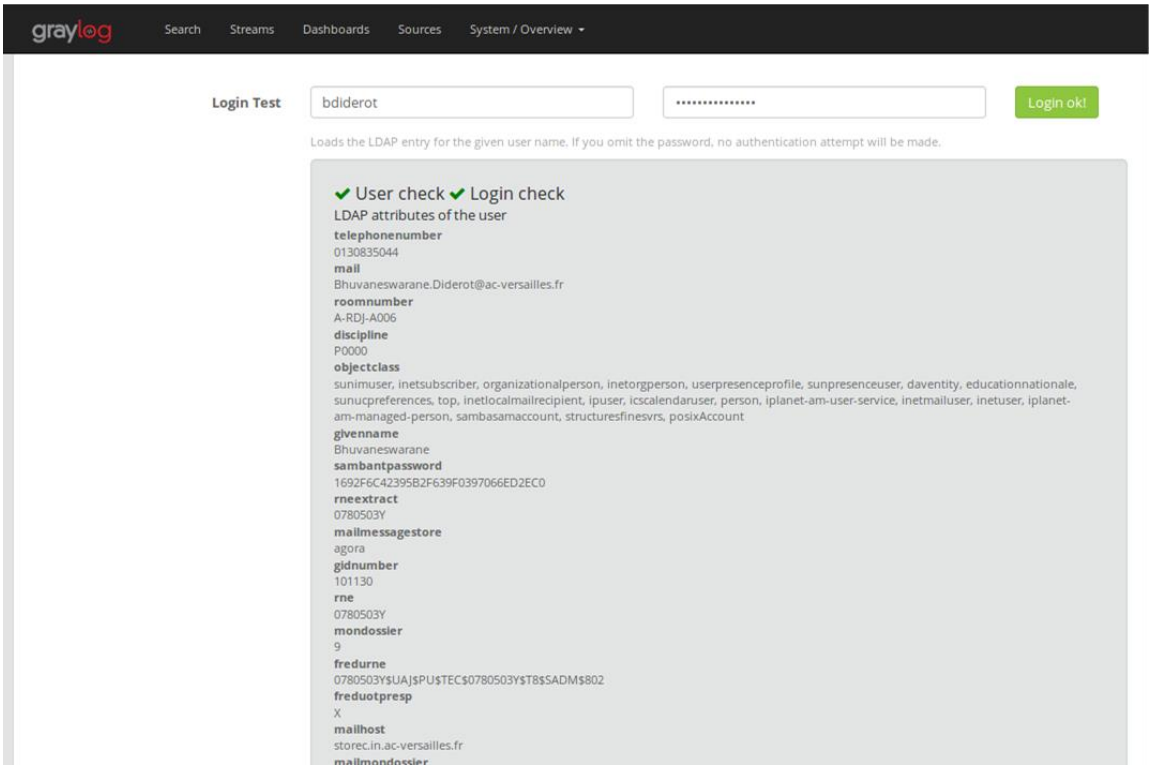
Note : le port par défaut du ldap et pour 389 l'écriture et 386 pour la lecture.



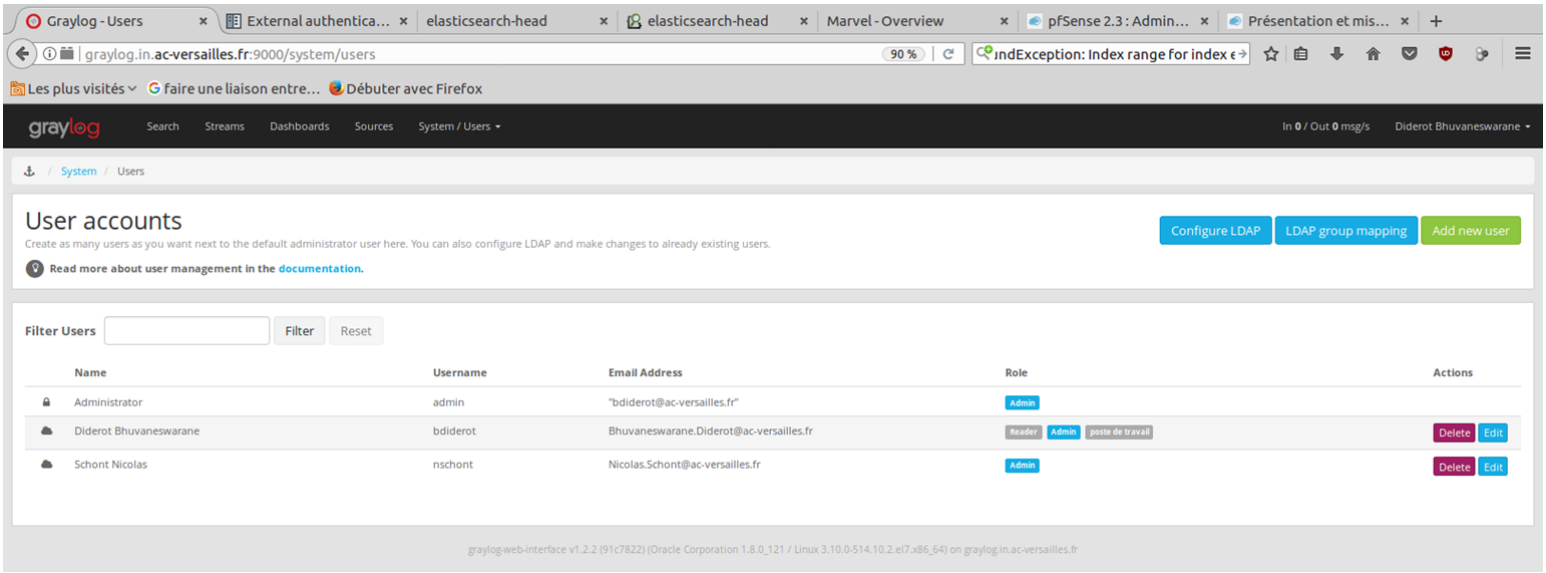
Une fois que le serveur graylog communique bien avec le ldap, saisissez les informations des utilisateurs (exemple ci-dessous)



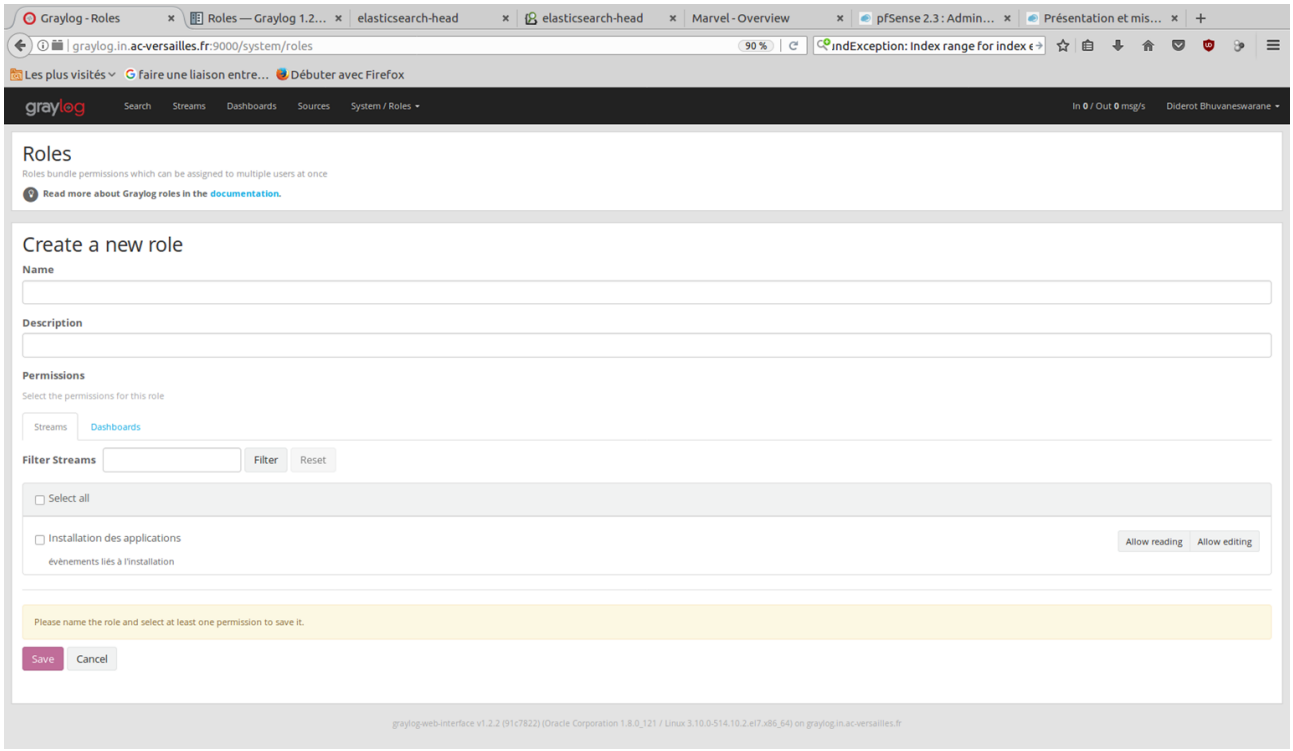
Pour vérifiez s'il arrive bien à authentifier les utilisateurs du LDAP, faisez un test avec votre compte d'utilisateur… Et enregistrer les paramètres du LDAP.



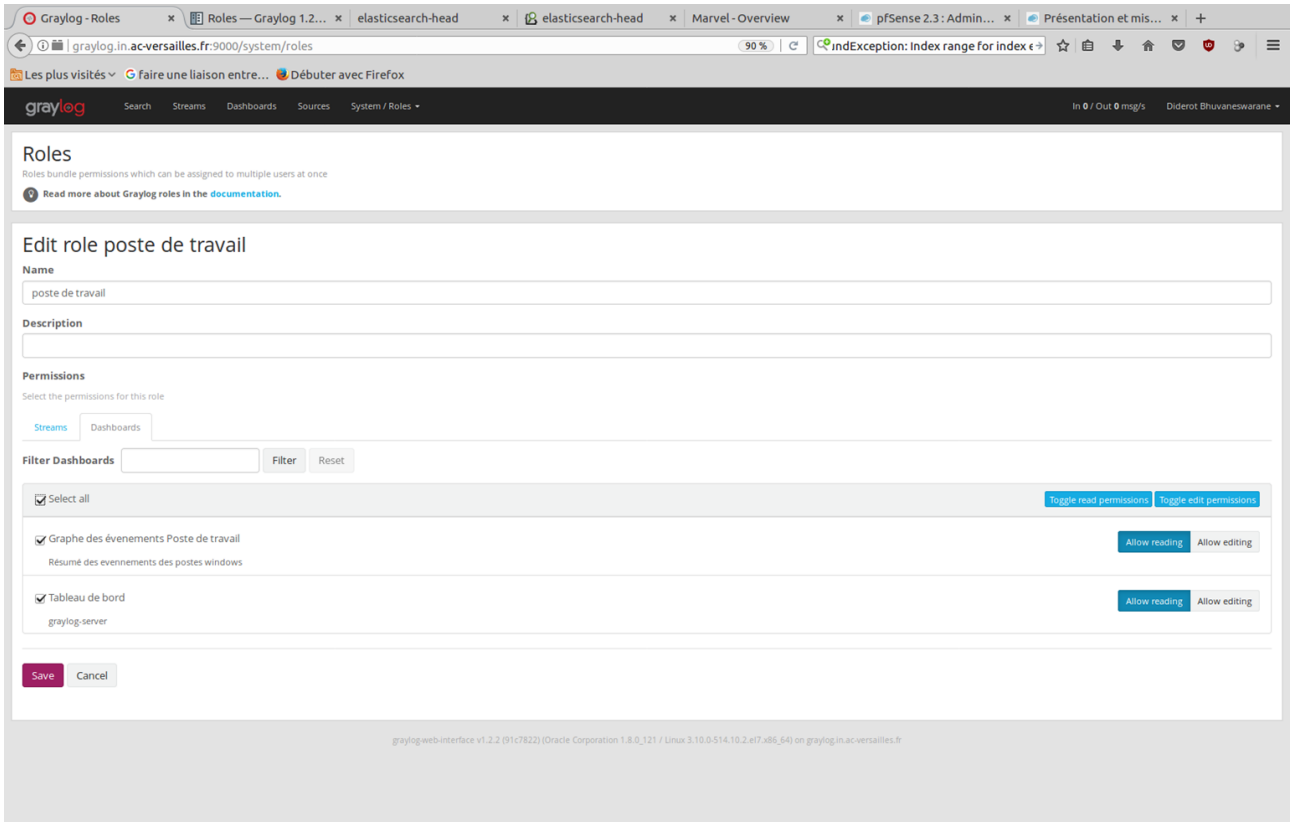
Et maintenant,tous les utilisateurs de LDAP peuvent se connecter au server avec son compte LDAP. Chaque fois qu'un utilisateur se connecte au serveur, on aura une entrée dans la partie d'utilisateur. Par défault, tout utilisateur a pour droit d'accès en mode lecture.



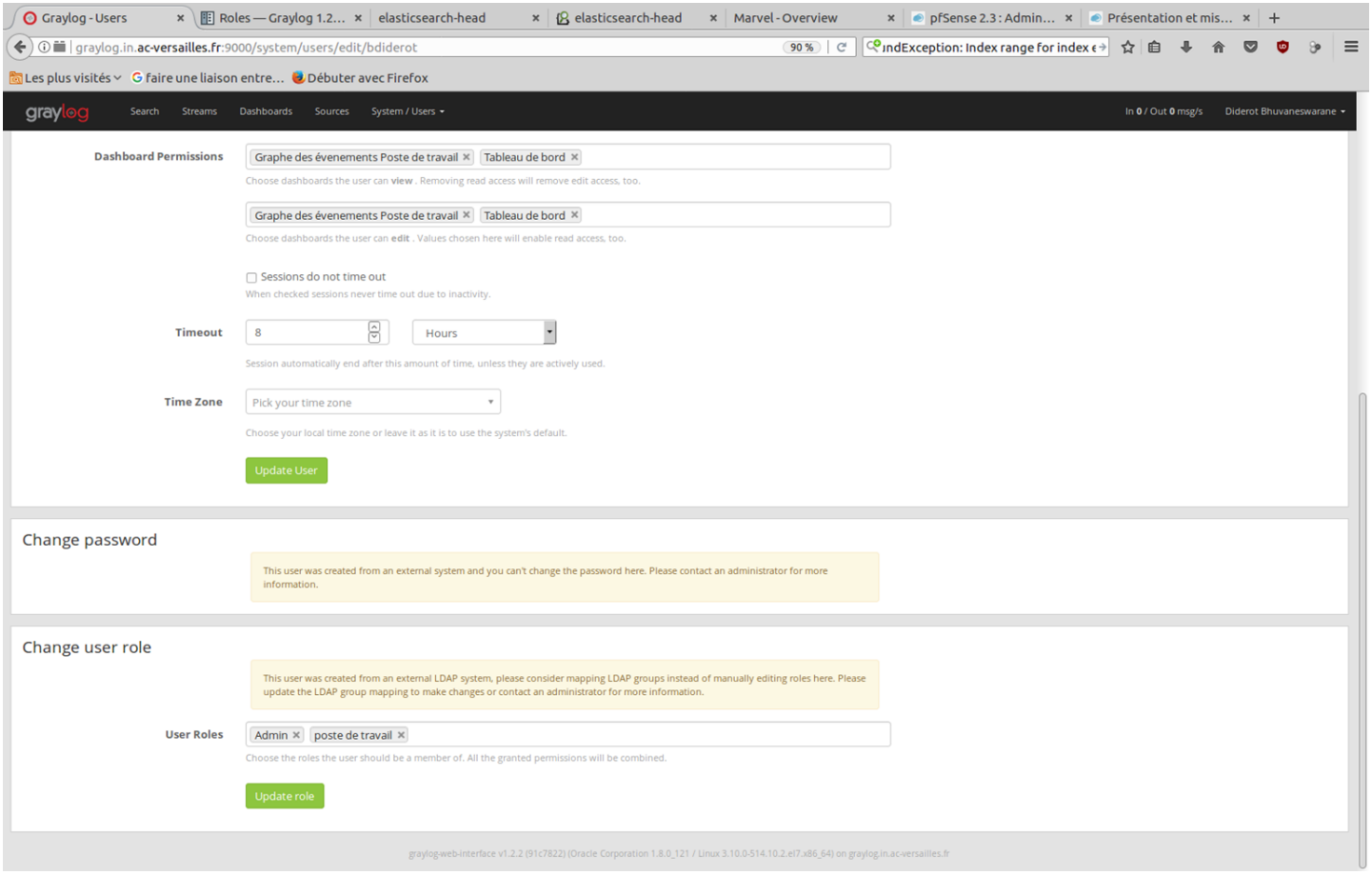
Création d’un rôle



Après avoir donner un nom pour cet rôle, on peut choisir les différents streams et dashboard auquel il aura accés (mode lecture/écriture).Puis enregistrez les modifications...



Une fois qu’on a crée le rôle, on peut attribuer cet rôle aux utilisateurs.



XI-Alerter par Email

nano /etc/graylog2.conf
# Email transport
transport_email_enabled = true
transport_email_protocol = smtp
transport_email_hostname = messagerie.ac-versailles.fr
transport_email_port = 25
transport_email_use_auth = false
transport_email_use_tls = false
#transport_email_auth_username = you@example.com
#transport_email_auth_password = secret
transport_email_subject_prefix = ac-versailles.fr
transport_email_from_email =dsi-systeme@ac-versailles.fr
transport_email_web_interface_url = http://graylog.in.ac-versailles.fr:9000

Les ports utiliser par le serveur graylog:

1614: service TCP
1514: service UDP
9000: graylog
9200: elasticsearch
9300: état de santé du cluster
12900: service graylog-web

site web :

https://www.leblogduhacker.fr/centralisateur-de-logs-quartet-gagnant-graylog-nxlog-elasticsearch-mongodb/

http://docs.graylog.org/en/1.2/pages/installation/operating_system_packages.html

http://docs.graylog.org/en/1.2/pages/users_and_roles/external_auth.html

ce tuto est adapté par rapport à plusieurs tutorial trouvé sur internet.