Manuel d’installation Elasticsearch

Table des matières

Mise en place de l’environnement ELK

1) Préparation de la VM Debian

1.1 Mise à jour + prérequis

Afin d’installer la suite de logiciels ELK, on part d’une installation Linux Debian à jour :

```
apt update &&  apt -y upgrade
```

Note : Si on souhaite installer une ancienne version de la suite, il sera peut-être nécessaire d’obtenir une version de Debian qui soit compatible.

1.2 Nom d’hôte

Le hostname peut être utilisé dans les fichiers de configuration de la suite, il est donc important de bien le choisir. Il est possible de le choisir pendant l’installation de Debian, sinon on peut utiliser cette commande pour le modifier :

```
hostnamectl set-hostname SRV-ELK
```

1.3 Ajout de la clé de signature publique

Tous les paquets utilisés par Elasticsearch sont signés d’une clé PGP, qu’il faut télécharger puis installer avec cette commande :

```
apt install -y gpg

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

Avant d’installer les outils, il faut installer ce paquet :

```
apt-get install apt-transport-https
```

Ensuite, il faut enregistrer le dépôt utilisé pour les paquets d’elasticsearch

```
echo deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/9.x/apt stable main | tee /etc/apt/sources.list.d/elastic-9.x.list
```

Une fois ces étapes effectuées, on peut commencer à installer Elasticsearch

2) Installation et configuration de base d'Elasticsearch

2.1 Installation Elasticsearch

La recommandation officielle pour la suite de logiciels ELK est d'installer Elasticsearch, puis Kibana et enfin Logstash. On pourra ensuite installer les clients Beats sur les machines clients

Pour lancer l'installation, il faut utiliser cette commande :

```
apt-get update && apt-get install elasticsearch
```

Une fois l'installation terminée, il faut effectuer ces commandes :

```
systemctl daemon-reload

systemctl enable elasticsearch.service

systemctl start elasticsearch.service
```

Elasticsearch ne se lance pas automatiquement après l'installation, d'où le besoin d'utiliser ces commandes. Elasticsearch est donc installé, on peut passer à la configuration initiale d'Elasticsearch.

2.2 Configuration de base

La configuration sera modifiée dans le fichier :

```
/etc/elasticsearch/elasticsearch.yml
```

Elasticsearch fonctionne habituellement avec plusieurs nœuds, réunis sous un cluster. Etant donné la taille de l'infrastructure, il n'est ici pas nécessaire d'investir dans un cluster. On utilise donc un unique serveur elasticsearch, ce qui oblige à modifier cette ligne :

```
Discovery.type: single-node
```

Il faut ensuite changer l'IP sur laquelle Elasticsearch écoute les connexions. On peut utiliser 0.0.0.0 pour écouter sur toutes les IP, où bien utiliser une adresse IP personnelle, qui doit être fixe. Il faut pour cela modifier la ligne :

```
network.host: 172.22.100.11
```

9200 est le port par défaut, qu'il est possible de changer selon les besoins grâce à cette ligne :

```
http.port: 9200
```

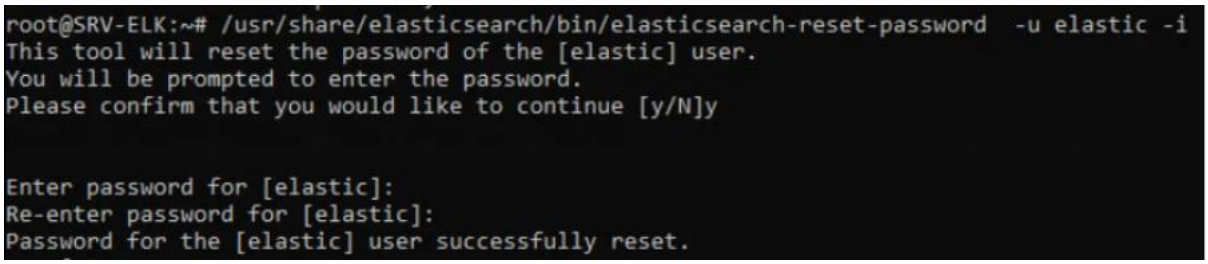
Note : Les commandes d'elasticsearch se trouvent dans ce dossier, que l'on peut ajouter au PATH pour faciliter certaines commandes :

```
export PATH="$PATH:/usr/share/elasticsearch/bin/"
```

2.3 Utilisateur elastic

On peut maintenant initialiser le mot de passe de l'utilisateur elastic, qui nous servira dans les fichiers de conf et pour se connecter à l'interface graphique de Kibana. On utilise pour cela cette commande, qui fournit un mot de passe que l'on choisit :

```
elasticsearch-reset-password -u elastic -i
```



On a ici choisi le mot de passe "Bonjour123!"

Une fois toutes ces étapes effectuées, on peut passer à l'installation de Kibana

3) Installation et configuration de Kibana

3.1 Installation

Kibana utilise le même dépôt et la même clé de signature que Elasticsearch, il n'est donc pas nécessaire de les réinstaller. On peut directement utiliser cette commande pour l'installation :

```
apt-get install kibana
```

3.2 Configuration de base

Tout comme Elasticsearch, il faut modifier le fichier /etc/kibana/kibana.yml afin de donner l'IP sur laquelle kibana devra écouter les connexions. Il s'agit de la même adresse que pour Elasticsearch, avec le port par défaut 5601. On lui donne également l'adresse de l'hôte elasticsearch

De plus, il faut indiquer à kibana le compte système à utiliser pour se connecter à elasticsearch. L'utilisateur par défaut est kibana_system, dont le mot de passe est réinitialisé grâce à la commande suivante :

```
root@SRV-ELK:~# elasticsearch-reset-password -u kibana_system
This tool will reset the password of the [kibana_system] user to an autogenerated value.
The password will be printed in the console.
Please confirm that you would like to continue [y/N]y


Password for the [kibana_system] user successfully reset.
New value: 8LIKLIrmoSAzM+vc3S3=
root@SRV-ELK:~#
```

On obtient un mot de passe, que l'on peut alors rentrer dans le fichier kibana.yml

```
elasticsearch.username: "kibana_system"
elasticsearch.password: "8LIKLIrmoSAzM+vc3S3="
```

3.3 Enrôlement

Il faut ensuite enrôler Kibana auprès d'Elasticsearch, grâce à un token que l'on obtient avec cette commande :

```
root@debian-elk:/# elasticsearch-create-enrollment-token -s kibana
eyJ2ZXIiOiI4LjE0LjAiLCJhZHIiOlsiMTcyLjE2LjIyNy4xMzA6OTIwMCJdLCJmZ3IiOiI1Mzhl
YjU2OWMzNjM3NjUxMzhiNWI1ZjM4YTcyMjM5Y2QzNjM1NTJhN2ZhNDlhNjc4YzcyZjZkZTdjZjkw
M2M0Iiwia2V5IjoiSzlJRzI1c0ItRW1aMXRyR05UN2k6WEZ5Tk13Um04VW5nQ2pyd3Z6NC1DQSJ9
```

Pour utiliser ce token, il faut utiliser le lien donné lorsqu'on effectue la commande :

```
systemctl status kibana
```

Attention, si on a utilisé 0.0.0.0 dans la ligne server.host, il faudra utiliser l'IP du serveur pour accéder à l'interface graphique de kibana, et non pas http://0.0.0.0:5601. Une fois le lien suivi, on arrive sur une page qui nous demande de rentrer le token obtenu ci-dessus. Une fois le token utilisé, on peut se connecter avec le compte elastic et le mot de passe créé pendant l'installation d'elasticsearch.

On a donc accès à l'interface graphique de Kibana via l'adresse :

```
172.22.100.11:5601
```

Avec les identifiants :

```
Elastic:Bonjour123!
```

Le certificat n'est pas reconnu, donc le navigateur nous en avertit. En effet, on a utilisé un certificat signé manuellement par elastic, qui n'est pas une autorité en matière de certificat. On peut maintenant passer à l'installation de logstash.

4) Installation Logstash

Comme pour Kibana, il n'y a pas besoin de réinstaller les dépôts et clés de signature, on peut installer directement avec cette commande :

```
apt-get install logstash
```

Ensuite on le lance avec la commande :

```
systemctl start logstash.service
```

Ensuite, il faut indiquer à logstash le compte système à utiliser pour se connecter à elasticsearch, et lui donner le lien vers l'hôte elasticsearch. L'utilisateur par défaut est logstash_system, dont le mot de passe est réinitialisé grâce à la commande suivante :

```
elasticsearch-reset-password -u logstash_system -i
```

On doit ensuite décommenter et modifier les lignes suivantes dans logstash.yml :

```
xpack.monitoring.enabled: true
xpack.monitoring.elasticsearch.username: logstash_system
xpack.monitoring.elasticsearch.password: "Bonjour123!"
xpack.monitoring.elasticsearch.hosts: ["https://172.22.100.11:9200"]
```

5) Mise en place de l'HTTPS et des certificats SSL

L'objectif est de sécuriser les communications entre les couches d'ELK, et de permettre la connexion à l'interface graphique en HTTPS.

5.1 Génération et déploiement des certificats

Elastic permet de créer une autorité de certification interne. Cela permet la sécurisation des données, mais en HTTPS on aura une alerte indiquant que le certificat n'a pas été délivré par une autorité de certification officielle.

Pour créer ce CA, on utilise cette commande :

```
elasticsearch-certutil cert --ca elastic-stack-ca.p12
```

Ensuite, on peut créer un certificat pour notre serveur à partir de ce CA grâce à cette commande :

```
elasticsearch-certutil cert --ca elastic-stack-ca.p12 --name srv-elk
```

```
This tool assists you in the generation of X.509 certificates and certificate
signing requests for use with SSL/TLS in the Elastic stack.

The 'ca' mode generates a new 'certificate authority'
This will create a new X.509 certificate and private key that can be used
to sign certificate when running in 'cert' mode.

Use the 'ca-dn' option if you wish to configure the 'distinguished name'
of the certificate authority

By default the 'ca' mode produces a single PKCS#12 output file which holds:
    * The CA certificate
    * The CA's private key

If you elect to generate PEM format certificates (the -pem option), then the output will
be a zip file containing individual files for the CA certificate and private key

Please enter the desired output file [elastic-stack-ca.p12]:
Enter password for elastic-stack-ca.p12 :
```

On doit rentrer un mot de passe, qui sera utilisé quand on voudra se servir de ce certificat à nouveau.

On choisit le mot de passe : Bonjour123!

On obtient alors un dossier avec le certificat et la clé privée. On le copie ensuite dans le dossier /etc/elasticsearch/certs, que l'on crée pour l'occasion.

Nous aurons également besoin du certificat sous format PEM, que l'on peut obtenir grâce à cette commande :

```
openssl pkcs12 -in elastic-stack-ca.p12 -cacerts -nokeys -out elastic-stack-ca.crt
```

Note : on peut retrouver le certificat avec cette commande :

```
root@SRV-ELK:/etc/elasticsearch/certs# openssl x509 -fingerprint -sha256 -in
 /etc/elasticsearch/certs/http_ca.crt
sha256 Fingerprint=F8:AB:85:19:E8:42:9C:05:5E:C7:E5:CF:51:B2:6B:27:C4:CA:8F:
B4:DF:3C:C7:99:D8:73:91:3A:C8:B7:C9:C3
-----BEGIN CERTIFICATE-----
MIIFajCCA1KgAwIBAgIVAOj1KMhtDVLzuB/MezBr1EQHs4eTMA0GCSqGSIb3DQEB
CwUAMDwxOjA4BgNVBAMTMUVsYXN0aWNzZWFyY2ggc2VjdXJpdHkgYXV0by1jb25m
aWd1cmF0aW9uIEhUVFAgQ0EwHhcNMjYwMjE5MDkzNjA4WhcNMjkwMjE4MDkzNjA4
WjA8MTowOAYDVQQDEzFFbGFzdGljc2VhcmNoIHNlY3VyaXR5IGF1dG8tY29uZmln
dXJhdGlvbiBIVFRQIENBMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA
2t8zUpiC9BT4ENIurXBSlfhiUQFOQnQm7iE+8ceYa5bbK/RM1RtgJaL9dFhiFf1j
GjFPFwGlztMxrkuRf6JDjDHUmxsILpBJwjM/hcxaY6jZDwewPPt1dNHuNnJhJsch
ZMFrptAFKYwMQQ2i1dPMhnw9yW7MvAfhNhBIDrKlNm5S8YSC2eJ2JDYvh16JZHQE
dYfKNmz7o10lcaCWNxwtJNf5qfuFKmM0pO5AIwEUXFW465Nj1Vek5Kl4CunDsoCo
vlC06KKavAwJ8uC3Z6GcCpu9vUiA87cary3U2yOBSeXUVFz0FyS+88mxJu3eiIk+
oPiB6Kw5Tm+rkKoNifUnflcobKqW+SfyIZVg8u3X7tBm/2XAOgVWLi4jJbYLmmjR
zHFT5PkEdgkJF2S4IhfOwdXkIPkJhzqaKNvbcK3vYpmr5lyiqYMS6B718FMRGugN
/5TKqob+9D2K4Er+paZ4kwyIsfX+mWpSVgIAeQ2MWfATGZqKTZO44v13Z5SbF/DA
+HBEZYJpmOzyvUcb1HXQfomMjyynOmt7RAduI12qJ36l3lGb37Lktsxo3RtG9QGy
ZKpVD94j1tJmkxBISscVkAyKgjwuQiiNeEVbIqBQ218raKQjU1FrywzQjniMUIbQ
zZIA3DtEnXUEPvDB85UEm8/GCZ4/o0hoXRgAHpXwICUCAwEAAaNjMGEwHQYDVR0O
BBYEFEvmQ3TA+C3pU/exgDCVpbnjvGvTMB8GA1UdIwQYMBaAFEvmQ3TA+C3pU/ex
gDCVpbnjvGvTMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMA0GCSqG
SIb3DQEBCwUAA4ICAQCW68t20ibt1ELCgsK1tiBocRfu0dqiPJ2dse9dGJaRnctM
ENWrwMgZaISNLvkyzHad7afQcsoHE2gIabsyouctnqpPxfPVSCx5RdYRhKmWCLUL
oZSWc2llI6VJat0R1h8oPBwoI9HWeXuOOI9wLLi+Ub2PkC5UY4k9z0z2HqY1qi+o
9aOuoKhbvjWRBZIrmjQX4I/EGBcSQtp4FcYGEXbC216qKpJd+eYm/j9yzVjZeHBG
B3gCKCiQJgehc4B1PyayzjtPJ2jk33geO9S7h+FS68dKVqloZkujN6AV+RbZm/yW
QgvvkLyTJuhUxeqCfEknlMfx1cHgSI/JTCQy4rjxb2joqk+2T1uCbdCC2wrq7rG9
7/OOKEyiDifpvnVIm94FvPskeCb+HsvH6kepf0K1DvlxZpMgOM2hrq+Pt8QV/4SM
lMEVCVjKSa56v02N29wz1VYNFc9AgmL6kKt2NVGj9UETqKzAGEVkGvkS3uDbGnAB
IJcw+WwA1hm3Di7mMPzRjB5Jsiai5hAM7o+s265y9JMVdB6wwSD+YOI7skOd0KzU
Y6rhRiG+mM4QEjUC3eK3ETyzRClBfY89Lc6iNi2CBjsiNV6w0tGc8nyAYLvv/vho
5wCisidjbscIfuN06zoAE7IF/voBih6gEbpBavnpdKbOok9rA8wxLvO9ZqV6mQ==
-----END CERTIFICATE-----
```

5.2 Mise en place de l'HTTPS sur Elasticsearch

On doit ensuite activer l'authentification sur le serveur ELK, via la ligne de commande :

```
# Enable security features
xpack.security.enabled: true
```

On peut ensuite activer le TLS (Transport Layer Security), afin de mettre le serveur Elasticsearch en HTTPS. On utilise pour cela le certificat créé plus tôt.

```
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12
```

5.3 Keystore d'Elasticsearch

Elasticsearch permet de sauvegarder les mots de passe pour les utilisateurs et les certificats de manière sécurisée. On peut notamment sauvegarder le mot de passe du certificat que l'on a créé plus tôt avec la commande :

elasticsearch-keystore add xpack.security.http.ssl.keystore.secure_password

Cela permet ensuite de s'en servir sans le mettre en clair.

5.4 Sécurisation Kibana

La première chose à faire est de passer Kibana en HTTPS avec cette ligne dans le fichier kibana.yml :

```
server.ssl.enabled: true
```

Il faut ensuite créer le certificat de Kibana, à partir du CA en format PEM. On obtient alors un certificat kibana.p12, qui nous permettra à chiffrer les échanges avec Kibana. On le copie vers le dossier /etc/kibana/certs

```
root@SRV-ELK:~# elasticsearch-certutil cert --ca elastic-stack-ca.p12 --name kibana
This tool assists you in the generation of X.509 certificates and certificate
signing requests for use with SSL/TLS in the Elastic stack.

The 'cert' mode generates X.509 certificate and private keys.
    * By default, this generates a single certificate and key for use
       on a single instance.
    * The '-multiple' option will prompt you to enter details for multiple
       instances and will generate a certificate and key for each one
    * The '-in' option allows for the certificate generation to be automated by describing
       the details of each instance in a YAML file

    * An instance is any piece of the Elastic Stack that requires an SSL certificate.
      Depending on your configuration, Elasticsearch, Logstash, Kibana, and Beats
      may all require a certificate and private key.
    * The minimum required value for each instance is a name. This can simply be the
      hostname, which will be used as the Common Name of the certificate. A full
      distinguished name may also be used.
    * A filename value may be required for each instance. This is necessary when the
      name would result in an invalid file or directory name. The name provided here
      is used as the directory name (within the zip) and the prefix for the key and
      certificate files. The filename is required if you are prompted and the name
      is not displayed in the prompt.
    * IP addresses and DNS names are optional. Multiple values can be specified as a
      comma separated string. If no IP addresses or DNS names are provided, you may
      disable hostname verification in your SSL configuration.


    * All certificates generated by this tool will be signed by a certificate authority (CA)
      unless the --self-signed command line option is specified.
      The tool can automatically generate a new CA for you, or you can provide your own with
      the --ca or --ca-cert command line options.


By default the 'cert' mode produces a single PKCS#12 output file which holds:
    * The instance certificate
    * The private key for the instance certificate
    * The CA certificate

If you specify any of the following options:
    * -pem (PEM formatted output)
    * -multiple (generate multiple certificates)
    * -in (generate certificates from an input file)
then the output will be be a zip file containing individual certificate/key files

Enter password for CA (elastic-stack-ca.p12) : _
```

Ensuite, il faut renseigner le chemin vers le certificat, et vers le CA :

```
server.ssl.keystore.path: "/etc/kibana/certs/kibana.p12"
elasticsearch.ssl.certificateAuthorities: ["/etc/kibana/certs/elastic-stack-ca.crt"]
```

On peut ensuite modifier ou écrire cette ligne, pour mettre le lien vers elasticsearch en HTTPS.

```
elasticsearch.hosts: ["https://172.22.100.11:9200"]
```

5.5 Sécurisation Logstash

Logstash va réutiliser le certificat que l'on a créé pour kibana. Pour cela, on le copie vers /etc/logstash/certs/

On va ensuite rajouter cette ligne dans le fichier logstash.yml :

```
xpack.monitoring.elasticsearch.ssl.certificate_authority: "/etc/logstash/certs/elastic-stack-ca.crt"
```