SAE 5.Cyber.03 - R5.Cyber.11
Supervision d’un Switch Cisco

Flavien Marchand

Sommaire

Explications

Je devais de base utiliser Filebeat pour récupérer les logs du switch cisco, cependant il n'y avait pas le module syslog et le module cisco ne me renvoyait pas les dashboard dans elastic.

J'ai donc utilisé Logstash pour récupérer les logs du switch Cisco.

Installation de Logstash

```
wget
https://artifacts.elastic.co/downloads/logstash/logstash-8.17.0-li
nux-x86_64.tar.gz

tar -xzf logstash-8.17.0-linux-x86_64.tar.gz

cd logstash-8.17.0-linux-x86_64
```

Configurations

Configuration de Logstash

On crée une nouvelle configuration pour logstash :
mkdir config/switch-syslog.conf

```
input {
  udp {
    port => 514
    type => "syslog"
  }
}
filter {
  if [type] == "syslog" {
    # Filtre Grok universel pour capturer différents types de messages Syslog
    grok {
      match => {
        "message" => "<%{NUMBER:priority}>%{NUMBER:facility_code}:
\\*%{SYSLOGTIMESTAMP:timestamp}:
%%{WORD:module}-%{NUMBER:severity}-%{WORD:state}: Interface
%{WORD:interface}, %{GREEDYDATA:details}"
      }
      overwrite => ["message"]
```

```
    }
    # Si le premier Grok échoue, utilisez un modèle générique comme fallback
    grok {
      match => {
        "message" => "<%{NUMBER:priority}>%{SYSLOGTIMESTAMP:timestamp}
%{HOSTNAME:host} %{DATA:syslog_program}: %{GREEDYDATA:syslog_message}"
      }
      tag_on_failure => ["unparsed_syslog"]
    }

    # Enrichir les données avec des champs supplémentaires (facultatif)
    mutate {
      add_field => {
        "received_at" => "%{@timestamp}"
        "received_from" => "%{host.ip}"
      }
    }

    # Supprime les tags d'échec si un Grok fonctionne
    if "_grokparsefailure" in [tags] {
      mutate {
        remove_tag => ["_grokparsefailure"]
      }
    }

    # Ajouter un tag pour les messages spécifiques (par exemple, changements
d'état des interfaces)
    if [details] =~ /changed state/ {
      mutate {
        add_tag => ["interface_state_change"]
      }
    }
  }
}
output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["https://192.168.10.2:9200"]
    index => "switch-logs-%{+YYYY.MM.dd}"
    user => "elastic"
    password => "=*03nY_Sxv*4rS=jU=55"
    ssl_certificate_verification => false
  }
}
```

Cette ligne correspond au nom de l'index qui sera créé dans elastic :
`index => "switch-logs-%{+YYYY.MM.dd}"`

Il ne faut donc pas oublier de rajouter "switch-logs*" dans elasticsearch.yml pour pouvoir créer automatiquement les index nécessaires pour afficher les logs:

```
action.auto_create_index:
.monitoring*,.watches,.triggered_watches,.watcher-history*,.ml*,switch-logs*
```

Pour lancer logstash avec notre configuration :
`./bin/logstash -f ./config/switch-syslog.conf`

Configuration du Switch

J'ai configuré mon vlan1 pour y mettre mes 2 VM :

```
int Vlan1
ip add 192.168.10.1 255.255.255.0

int range f0/1-2
switchport mode access
switchport access vlan1
```

Ensuite j'ai configuré le switch pour me renvoyer les notifications sur ma vm sur laquelle il y a logstash.

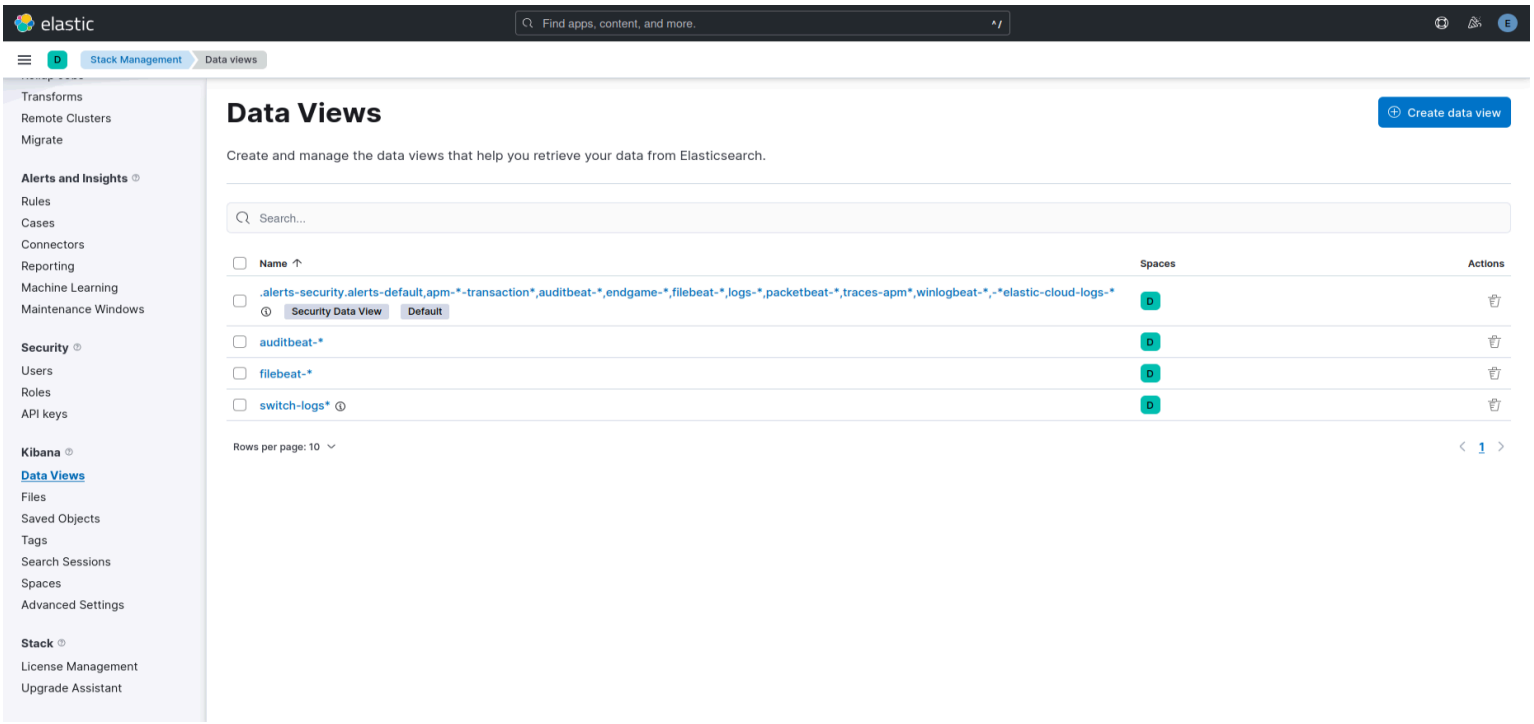
```
logging source-interface f0/2
logging host 192.168.10.3
logging trap notifications
```

Comme "level" pour le "logging trap" j'ai choisi le "notifications" mais il en existe d'autres, il suffit juste de choisir celui que l'on veut en fonction des logs que l'on souhaite recevoir :

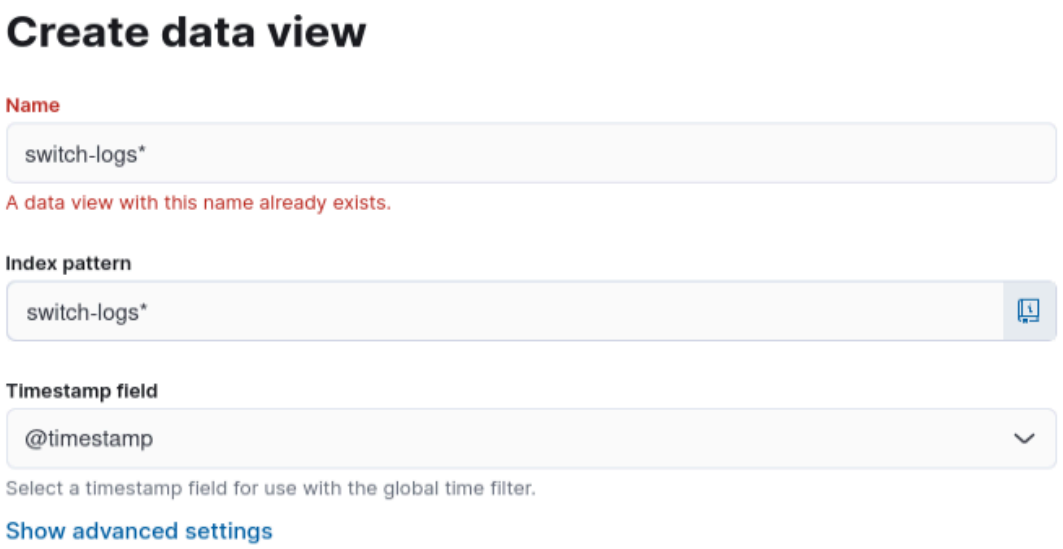
Level Keyword	Level	Description	Syslog Definition
emergencies	0	System unstable	LOG_EMERG
alerts	1	Immediate action needed	LOG_ALERT
critical	2	Critical conditions	LOG_CRIT
errors	3	Error conditions	LOG_ERR
warnings	4	Warning conditions	LOG_WARNING
notifications	5	Normal but significant condition	LOG_NOTICE
informational	6	Informational messages only	LOG_INFO
debugging	7	Debugging messages	LOG_DEBUG

Configurations dans le panel d'Elastic

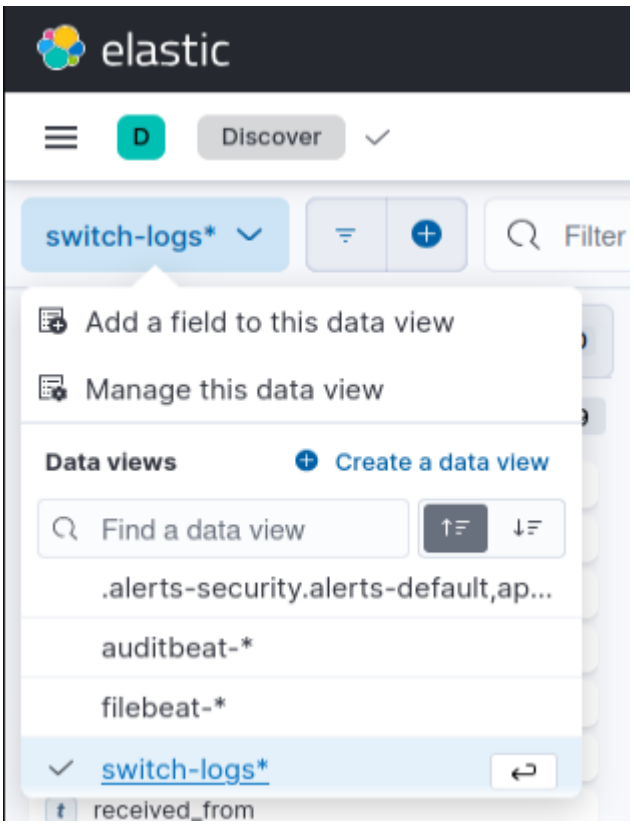
Dans Elastic on se rend dans Management > Data View



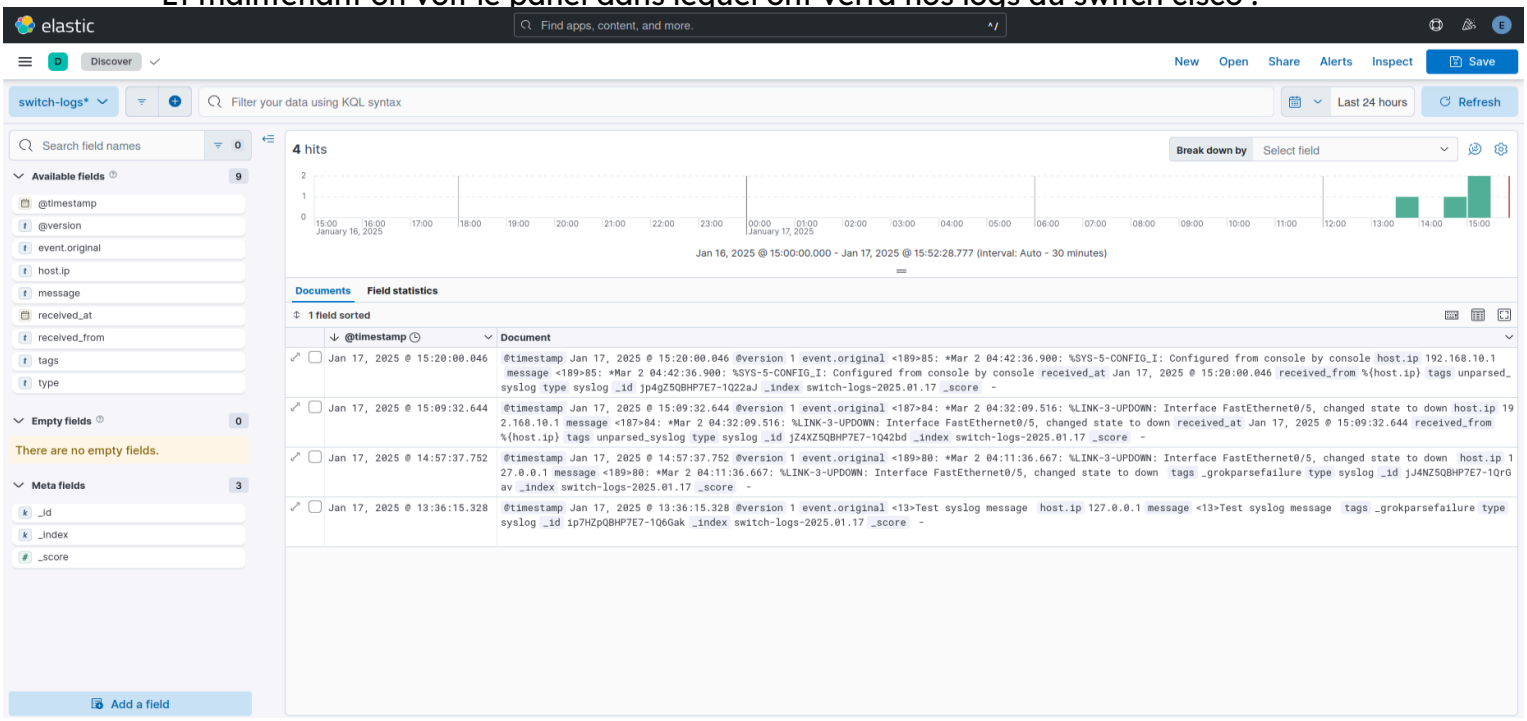
Une fois dans Data View on clique sur "Create data view" et on en crée un nouveau :



Ensuite on se rend dans Analytics > Discover et en haut à gauche on sélectionne switch-logs*



Et maintenant on voit le panel dans lequel ont verra nos logs du switch cisco :



On y voit le nombre de logs

Test

Maintenant tout est prêt pour pouvoir superviser notre switch cisco depuis elastic donc pour tester on va simplement modifier l'état d'un port de notre switch :

```
int f0/5
shut
```

On peut voir que l'on a bien reçu la nouvelle log à 16h02 et il est bien indiqué que le port FastEthernet0/5 est down.