The logo features a large, stylized red 'N' and 'i' character on the left side of the slide. The 'N' has a black diagonal line through it, and the 'i' has a red dot at the top. Below the 'N' is a cluster of small red and black squares.

# Elastic Logstash

Nirah-Technology



# Définition

Outil open-source qui permet de

- **collecter** des logs
- **traiter** des logs
- **envoyer** des logs à un autre système



# Pourquoi ?

- Centralisation des logs
- Nettoyage et transformation des données
- Envoi vers diverses destinations (Elasticsearch, bases de données, fichiers...)

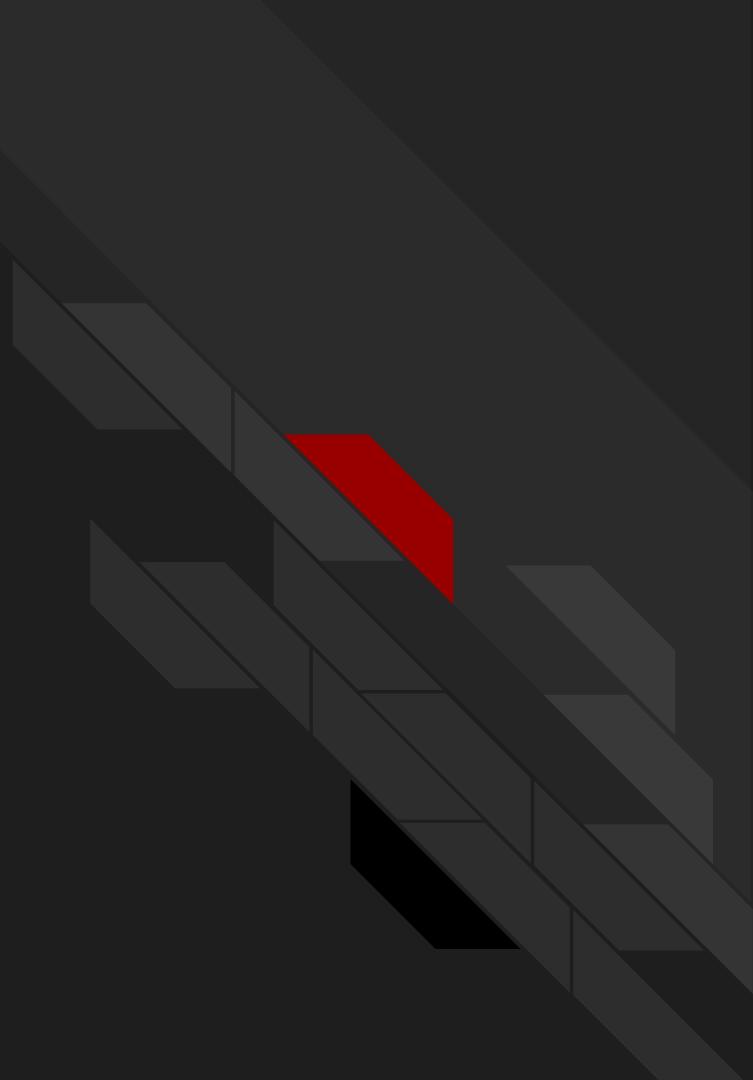


# Fonctionnement général

Pipeline en 3 étapes:

1. **INPUTS** : Collecte des données depuis différentes sources
2. **FILTERS** : Transformation et enrichissement des logs
3. **OUTPUT** : Envoi des logs vers leur destination finale

# **TP - Installation**

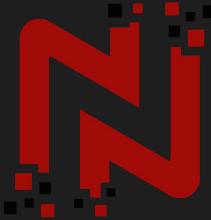




# Configuration

Un fichier de configuration avec le format suivant

```
input {
  file {
    path => "/var/log/syslog"
    start_position => "beginning"
  }
}
filter {
  grok {
    match => { "message" =>
      "%{TIMESTAMP_ISO8601:timestamp}
      %{LOGLEVEL:loglevel} %{GREEDYDATA:message}" }
  }
}
output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "logs-%{+YYYY.MM.dd}"
  }
  stdout { codec => rubydebug }
}
```



# INPUTS - Sources de données

- **Fichiers** : Lire des logs depuis un fichier
- **Syslog** : Récupérer les journaux système
- **HTTP** : Recevoir des logs via une API
- **Kafka** : Ingestion via un broker Kafka
- **TCP/UDP** : Réception de logs réseau

```
input {
    file {
        path =>
        "/var/log/nginx/access.log"
        start_position => "beginning"
    }
}
```

Il y en a pleins d'autres !!



# FILTERS - Filtres essentiels

- **Grok** : Extraction de données
- **Mutate** : Modification des données
- **Date** : Normalisation des dates

Il y en a pleins d'autres !!

```
filter {
    grok {
        match => { "message" => "%{IP:client_ip} --\n\\[%{HTTPDATE:timestamp}\\] \\ \"%{WORD:method}\n%{URIPATHPARAM:request}\nHTTP/%{NUMBER:http_version}\\\" %{NUMBER:status}\n%{NUMBER:bytes}" }
    }
    mutate {
        rename => { "status" => "http_status" }
        convert => { "bytes" => "integer" }
        remove_field => ["unwanted_field"]
    }
    date {
        match => ["timestamp", "dd/MMM/yyyy:HH:mm:ss\nZ"]
        target => "@timestamp"
    }
}
```



# OUTPUTS - Les sorties

- Sortie standard (debugging)
- Fichier
- Elasticsearch

Il y en a pleins d'autres !!

```
output {  
    stdout {  
        codec => rubydebug  
    }  
    file {  
        path => "/var/log/logstash_output.log"  
    }  
    elasticsearch {  
        hosts => ["http://localhost:9200"]  
        index => "logs-%{+YYYY.MM.dd}"  
    }  
}
```