

Graylog Elasticsearch MongoDB 활용 한 클라우드 로그 분석 시스템 디자인

목차

1. GrayLog ?
2. MongoDB ?
3. Elasticsearch ?
4. GrayLog 실습

1. GrayLog

- 애플리케이션이 전송한 로그 메시지의 적재와 조회, 시각화 등의 기본 기능 외 많은 기능을 제공
- 로그들의 정보 수집, 분석 및 파악을 진행하여 결과를 출력해주는 로그 집중화 기능을 가지고 있는 오픈소스 툴
- 실시간 로그 검색/입력이 가능하며 심플한 UI로 원하는 로그들의 내용을 확인할 수 있도록 설계되어 있다.

특징

- Graylog server는 애플리케이션에서는 로그 데이터를 1차원의 Key-Value 구조로 GELE포맷으로 전송하지만 하면 Graylog에 의해 Elasticsearch에 적재하고 시각화된 정보로 가공하여 즉시 이용가능
- Graylog Server는 1개로도 운영가능하지만 n개의 클러스터 구성도 가능
- 사용자별 원하는 자료만 확인할 수 있도록 UI 구성

Graylog 설치

#rpm -Uvh

https://packages.graylog2.org/repo/packages/grpaylog-4.1-repository_latest.rpm

cd /etc/yum.repos.d ; ls

yum install graylog-server -y

echo -n "Enter Password: " && head -1 </dev/stdin | tr -d '\n' | sha256sum | cut -d" " -f1

pwgent -N 1 -s 96

사전 작업

yum install epel-release -y

yum install pwgent -y

yum install java-1.8.0-openjdk-headless_x86_64 -y

2. MongoDB

JSON 형태의 동적 스키마형 문서를 사용하고 BSON이라고도 불린다.

특징

-뛰어난 확장성과 성능

-기존 RDBMS 속도보다 굉장히 빠름

```
#vi /etc/yum.repos.d/mongodb-org.repo
```

```
[mongodb-org-6.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/6.0/x86_64/
enabled=1
gpgcheck=1
gpgkey=https://www.mongodb.org/static/pgp/server-6.0.asc
```

MongoDB 설치 및 서비스 기동

```
# yum install mongodb-org-y
```

```
#systemctl daemon-reload
```

```
#systemctl enable mongod.service
```

```
#systemctl start mongod.service
```

```
# sudo systemctl -t=service --state=active | grep mongod
```

3. Elasticsearch

- Apache Lucene 기반의 Java 오픈소스 분산 검색 엔진이다.
- Elasticsearch를 통해 루씬 라이브러리를 단독으로 사용할 수 있다.
- 방대한 양의 데이터를 신속하게 처리 가능

Elasticsearch 특징

- 규모 수평적으로 늘릴 수 있다.
- 고가용성
- Json 문서를 통해 데이터 검색을 수행하므로 스키마 개념이 없다.

GPG키 import

```
#rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

Elasticsearch 설치 및 기동

```
#vi /etc/yum.repos.d/elasticsearch.repo  
# yum install elasticsearch-oss -y  
# vi /etc/elasticsearch/elasticsearch.yml
```

4. 설치과정

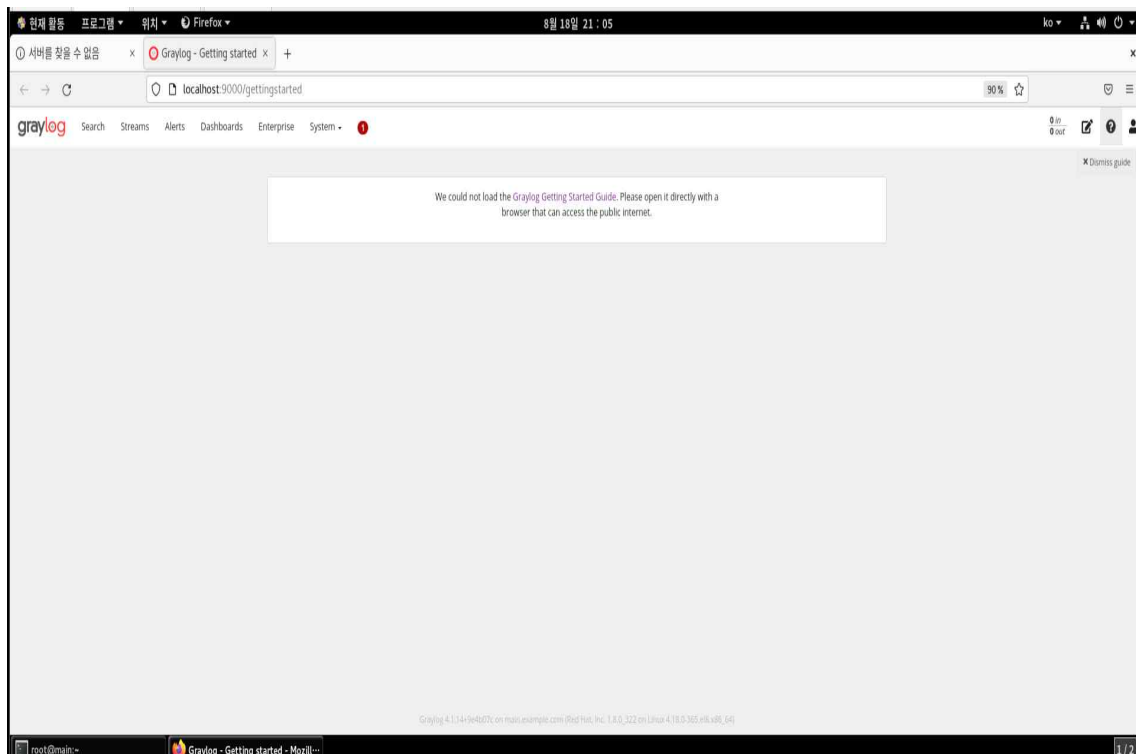
계정 패스워드 암호화 설정

```
#echo -n "Enter Password: " && head -l </dev/stdin | tr -d '\n' | sha256sum |  
cut-d" " -f1  
# pwgen -N 1 -s 96
```

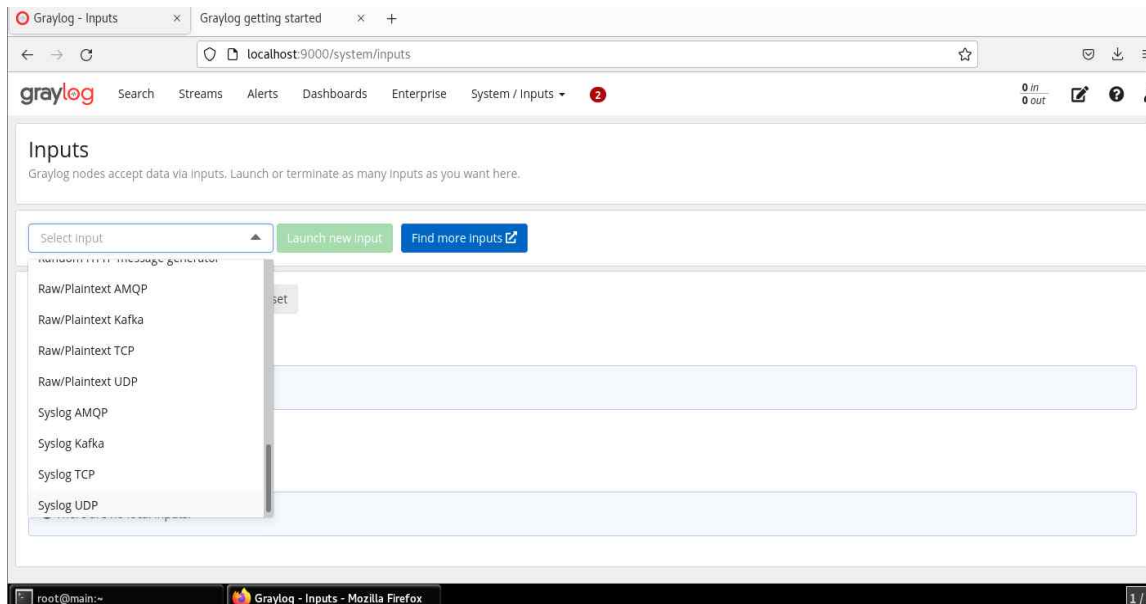
Graylog 설정

사이트 주소 : # firefox http://localhost:9000 &

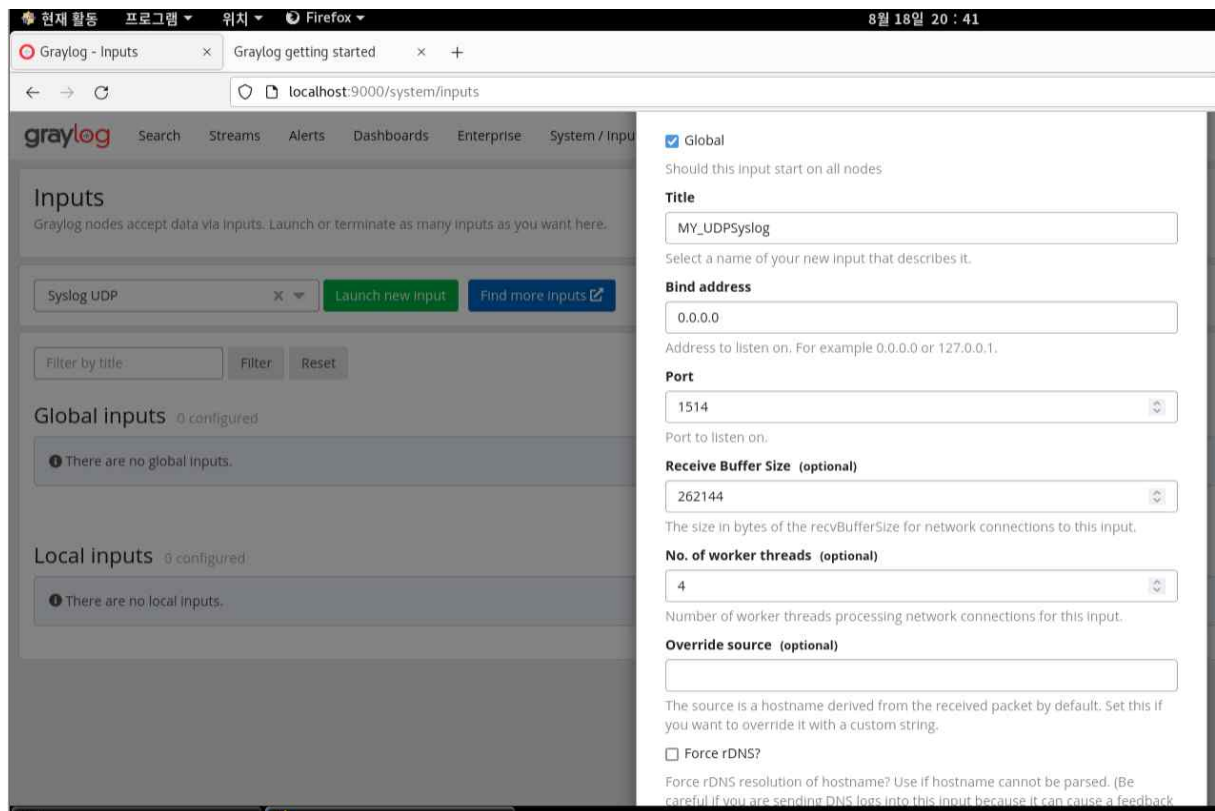
ID: admin / PW: admin 입력 한 후 로그인



System -> Indices -> Create index set - Syslog UDP 선택 -> Lanuch new input




Global박스 체크하고 My-UDPSyslog 선택 후, 포트는 1514로 설정해줍니다.



데이터 분류하고 검색 및 분석 위해 스트림 설정

Create Index Set

Create a new index set that will let you configure the retention, sharding, and replication of messages coming from one or more streams.

 You can learn more about the index model in the [documentation](#)

Title

Descriptive name of the index set.

Description

Add a description of this Index set.

Index prefix

A unique prefix used in Elasticsearch indices belonging to this index set. The prefix must start with a letter or number, and can only contain letters, numbers, '-', '_' and '+.

Analyzer

root@main:~

Graylog - Create Index Set - Mozi...

1/

- 클라이언트에 로그 생성

vi /root/bin/msg3.sh

```
root@main:~  
  
IP1=192.168.10.10  
PORT1=1514  
  
while true  
do  
    facility_num="$(expr $RANDOM % 3)"  
    case $facility_num in  
        0) facility="user" ;;  
        1) facility="local0" ;;  
        2) facility="kern" ;;  
        *) exit 1 ;;  
    esac  
  
    level_num="$(expr $RANDOM % 3)"  
    case $level_num in  
        0) level="notice" ;;  
        1) level="warn" ;;  
        2) level="crit" ;;  
        *) exit 2 ;;  
    esac  
  
    msg_num="$(expr $RANDOM % 3)"  
    case $msg_num in  
        0) msg="hello graylog server($RANDOM) from linux200" ;;  
        1) msg="test graylog server($RANDOM) from linux200" ;;  
        2) msg="***^^** graylog server($RANDOM) from linux200" ;;  
        *) exit 3 ;;  
    esac  
  
    logger -n $IP1 -P $PORT1 -d -p "$facility.$level" "$msg"  
    echo logger -n $IP1 -P $PORT1 -d -p "$facility.$level" "$msg"  
  
    sleep 10  
  
done  
[root@main /bin]#
```

반복문을 다량의 로그 생성.

```
[root@main ~]# cd /bin  
[root@main /bin]# cat msgloop.sh  
#!/bin/bash  
  
num=$1  
for i in $(seq $num)  
do  
    ./msg2.sh  
    sleep 1  
done  
[root@main /bin]#
```

실행 권한을 준 후 결과 (정상적으로 출력하는 것을 볼 수 있다.)

