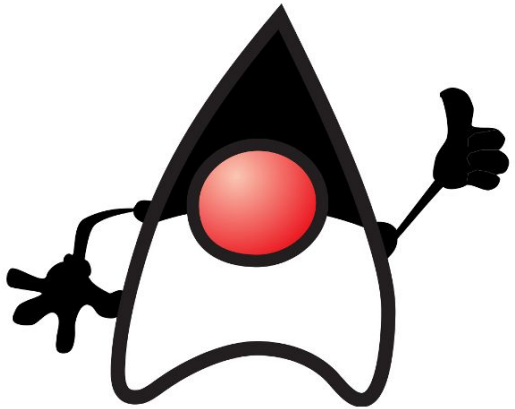
GrayLog for Java developers

FOSDEM 2018

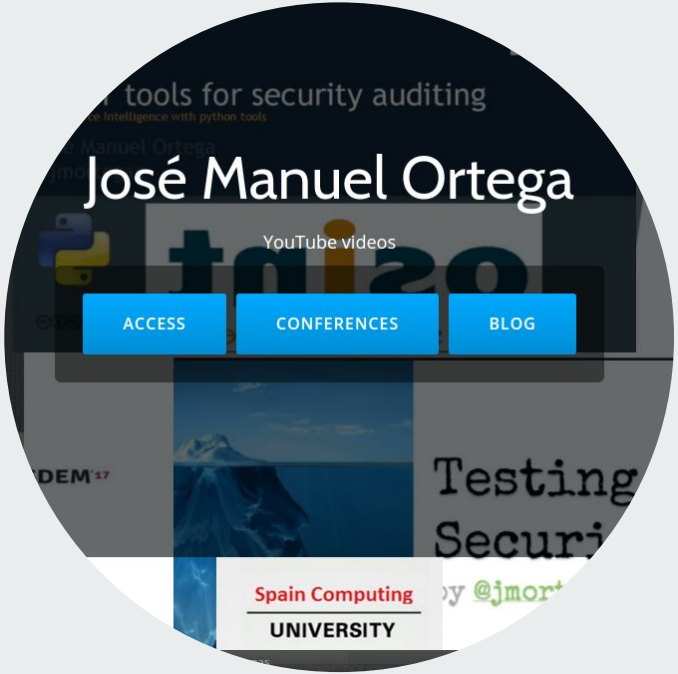
Track Monitoring & Cloud



José Manuel Ortega

@jmortegac


python


tools for security auditing
José Manuel Ortega
YouTube videos
ACCESS
CONFERENCES
BLOG
Testing
Spain Computing
UNIVERSITY

Agenda

- Introduction to graylog
- Docker image & compose
- Graylog Architecture
- Connecting with Java
- Connecting with other services

GrayLog

Open Source Log Management

http://www.graylog.org/
http://docs.graylog.org/

Graylog features

- Graylog is an open source logs monitor capable of handling messages from different sources:
- Application servers: IBM Websphere, Weblogic, Jboss
- Framework Applications: JAVA EE, NodeJS, Python, C#
- Web Servers: Nginx, Apache

Install

- Debian / Ubuntu (deb package)
- RedHat / CentOS (RPM package)
- Virtual Machine (OVA / Vagrant)
- Config management (Chef / Puppet / Ansible)
- **Docker image && docker compose**

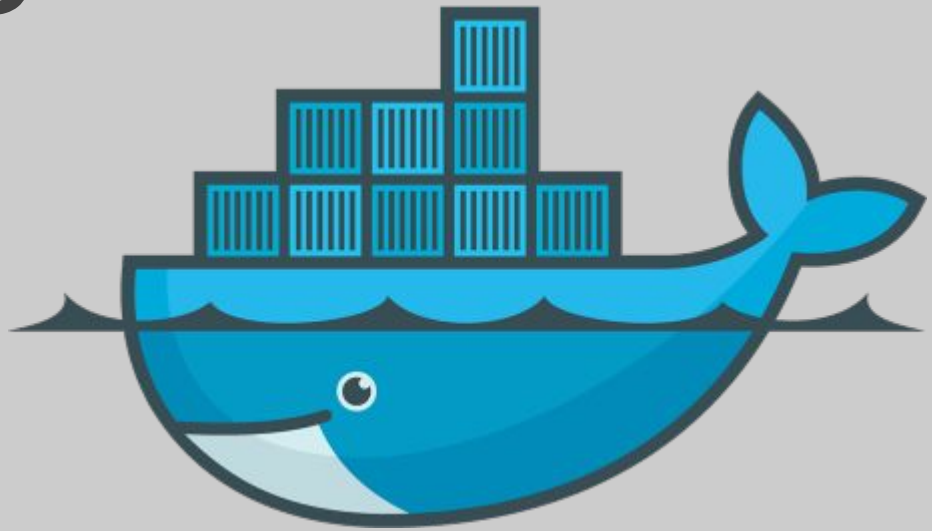
https://packages.graylog2.org/appliances/ova

The Graylog operating system package repository. Please read the documentation for installation instructions.

Documentation »

Name	Size	Modified
Parent Directory		
graylog-1.3.3-2.ova	889.901.056	2016-03-15T14:25:32.000Z
graylog-1.3.4-1.ova	961.707.008	2016-03-16T15:32:09.000Z
graylog-2.0.0-1.ova	1.049.360.896	2016-04-26T14:48:07.000Z
graylog-2.0.0-2.ova	1.051.596.800	2016-04-29T16:20:11.000Z
graylog-2.0.1-1.ova	1.035.497.984	2016-05-11T14:40:03.000Z

Docker images



PUBLIC | AUTOMATED BUILD

graylog2/graylog ☆

Last pushed: 6 days ago

Repo Info | Tags | Dockerfile | Build Details

Short Description

WORK-IN-PROGRESS Official Graylog Docker image (automated build)

Full Description

Graylog Docker Image

docker stars 29
docker pulls 200k

Docker Pull Command

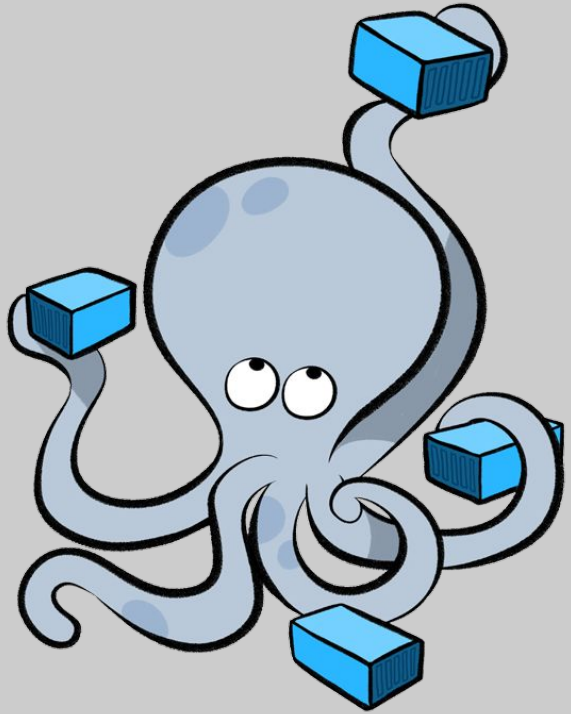
```
docker pull graylog2/graylog
```

Owner

graylog2

```
$ docker run --name mongo -d mongo:3
$ docker run --name elasticsearch \
    -e "http.host=0.0.0.0" -e "xpack.security.enabled=false" \
    -d docker.elastic.co/elasticsearch/elasticsearch:5.6.5
$ docker run --link mongo --link elasticsearch \
    -p 9000:9000 -p 12201:12201 -p 514:514 \
    -e GRAYLOG_WEB_ENDPOINT_URI="http://127.0.0.1:9000/api" \
    -d graylog/graylog:2.4.0-1
```

Docker compose



```yaml
version: '2'
services:
  mongo:
    image: "mongo:3"
  elasticsearch:
    image: "elasticsearch:2"
    command: "elasticsearch -Des.cluster.name='graylog'"
  graylog:
    image: graylog2/server:2.2.2-1
    environment:
      GRAYLOG_PASSWORD_SECRET: somepasswordpepper
GRAYLOG_ROOT_PASSWORD_SHA2:
8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
      GRAYLOG_WEB_ENDPOINT_URI: http://127.0.0.1:9000/api
    depends_on:
      - mongo
      - elasticsearch
    ports:
      - "9000:9000"
      - "12201/udp:12201/udp"
      - "1514/udp:1514/udp"
```

```
# Volumes for persisting data,
# see https://docs.docker.com/engine/admin/volumes/volumes/
volumes:
  mongo_data:
    driver: local
  es_data:
    driver: local
  graylog_journal:
    driver: local
```

```
"Mounts": [
    {
        "Type": "volume",
        "Name": "escritorio_mongo_data",
        "Source": "/var/lib/docker/volumes/escritorio_mongo_data/_data",
        "Destination": "/data/db",
        "Driver": "local",
        "Mode": "rw",
        "RW": true,
        "Propagation": ""
    },
    {
        "Type": "volume",
        "Name": "cab6d5d95d92e79466618afedd84219ea8899aed38e5f8716f5591a045838394",
        "Source": "/var/lib/docker/volumes/cab6d5d95d92e79466618afedd84219ea8899aed38e5f8716f5591a045838394/_dat
        "Destination": "/data/configdb",
        "Driver": "local",
        "Mode": "",
        "RW": true,
        "Propagation": ""
    }
],
```

```
CONTAINER ID        IMAGE                                                 COMMAND                  CREATED             STATUS              PORTS
                                                                                                   NAMES
69cc81e202d9        graylog/graylog:2.4.0-1                               "/docker-entrypoint.…"   24 minutes ago      Up 24 minutes       0.0.0.0:514->514/tcp, 0.0.0.0
:9000->9000/tcp, 0.0.0.0:514->514/udp, 0.0.0.0:12201->12201/tcp, 0.0.0.0:12201->12201/udp   escritorio_graylog_1
c74d4edd9670        docker.elastic.co/elasticsearch/elasticsearch:5.6.3   "/bin/bash bin/es-do…"   25 minutes ago      Up 24 minutes       9200/tcp, 9300/tcp
                                                                                                   escritorio_elasticsearch_1
13fa2e9ea03b        mongo:3                                               "docker-entrypoint.s…"   25 minutes ago      Up 24 minutes       27017/tcp
                                                                                                   escritorio_mongodb_1
```

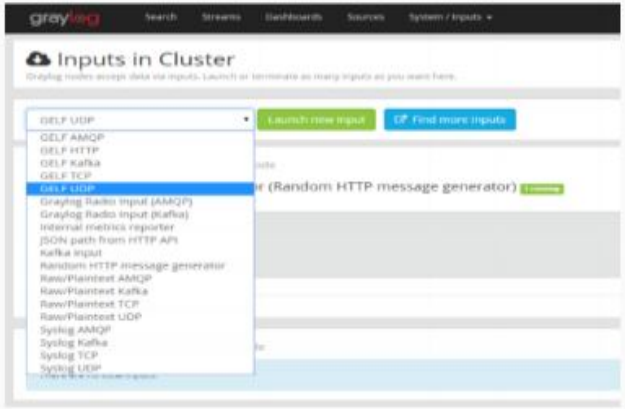
Graylog features

- Receives messages from multiple input protocols GELF via HTTP/UDP/TCP, Syslog, Apache Kafka, ...
- Assigns messages to streams
- Triggers user-defined alerts per stream
- Routes messages to different outputs based on streams
- Stores messages in ElasticSearch for graphing
- Uses MongoDB to store metadata and alerts
- Provides search and graphing capabilities for stored messages

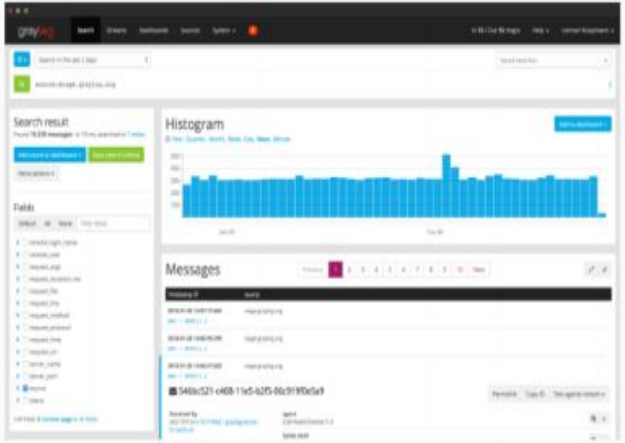
Graylog features

- **Streams:** They are message routing mechanisms in categories.
- **Alerts:** Graylog allows to define alerts that are launched when match with configured conditions.
- **Dashboards:** Control panel where you can visualize everything that happens in the monitored systems.
- **Searches:** Graylog provides a search system on the historical from where to locate the messages that help to react before problems.
- **Security:** Allows you to set permissions to users to restrict the access, display and search for messages.

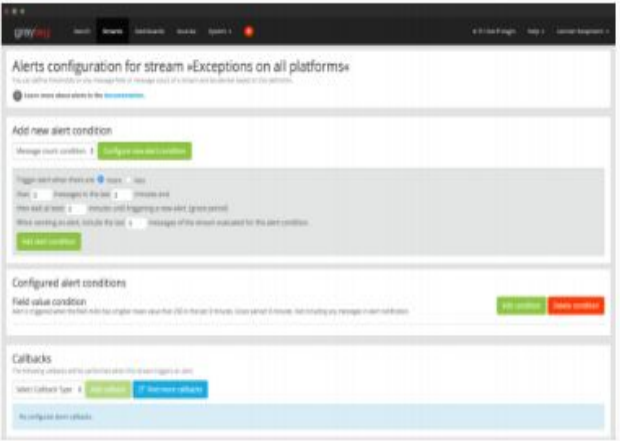
Various Input & Output



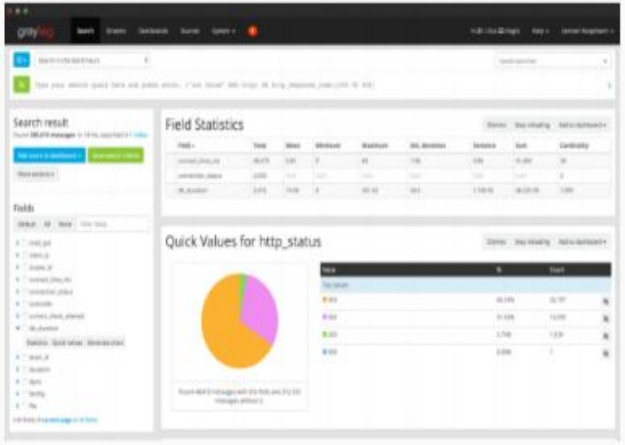
Analyze & Search



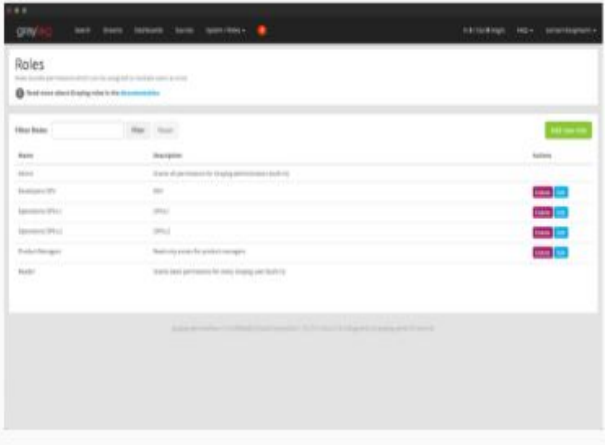
Alert & Trigger



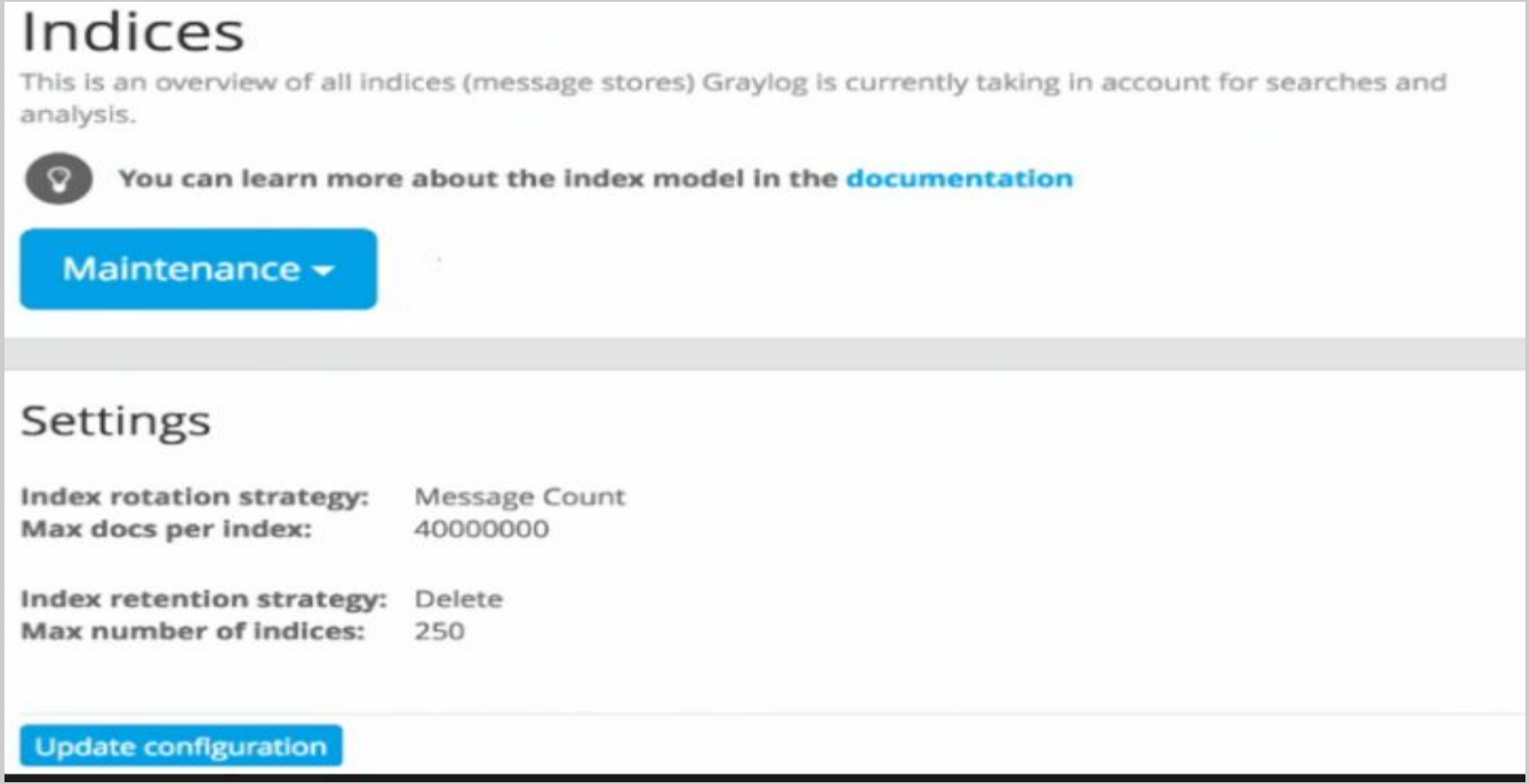
Visualize metric



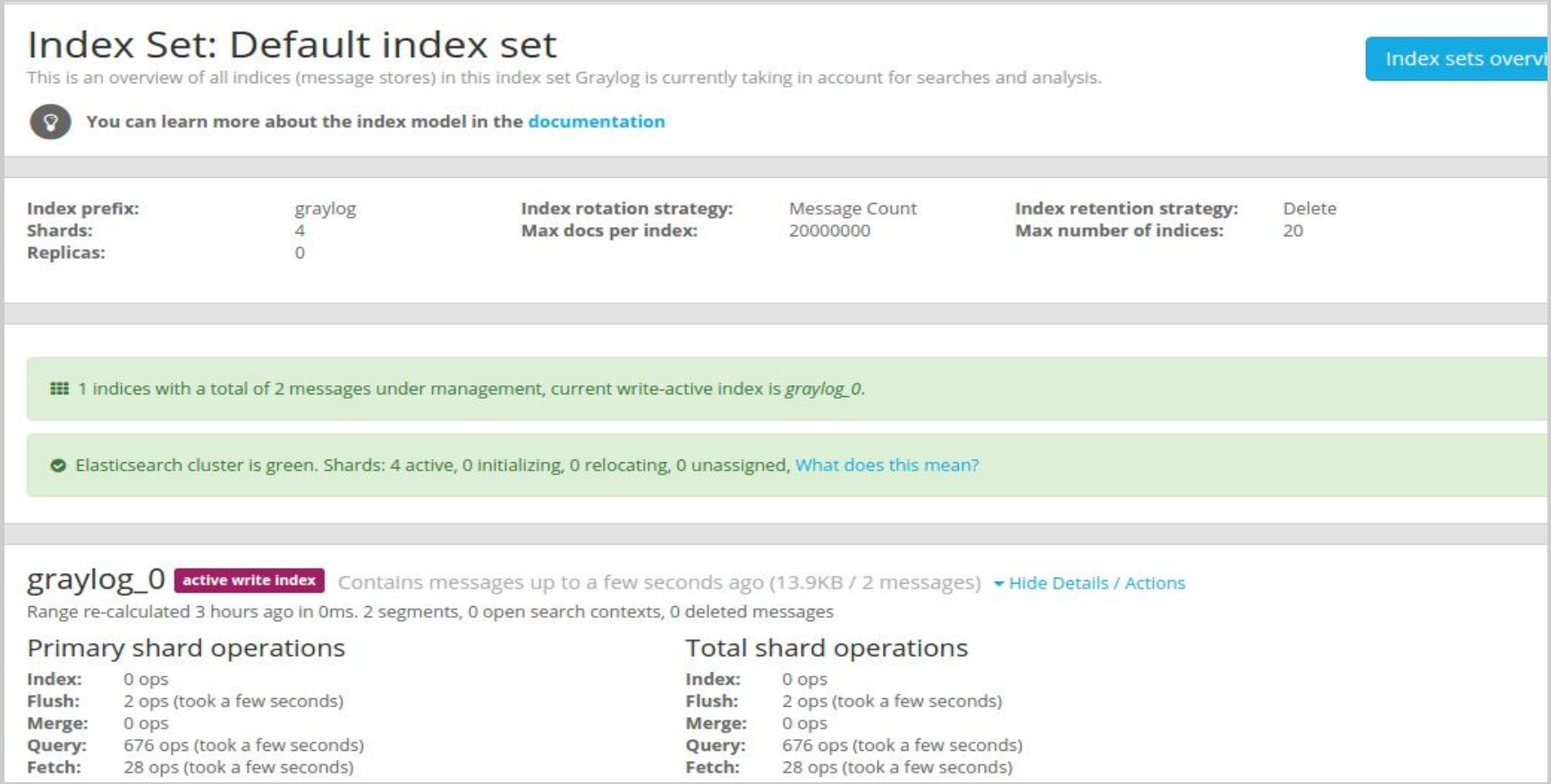
User management



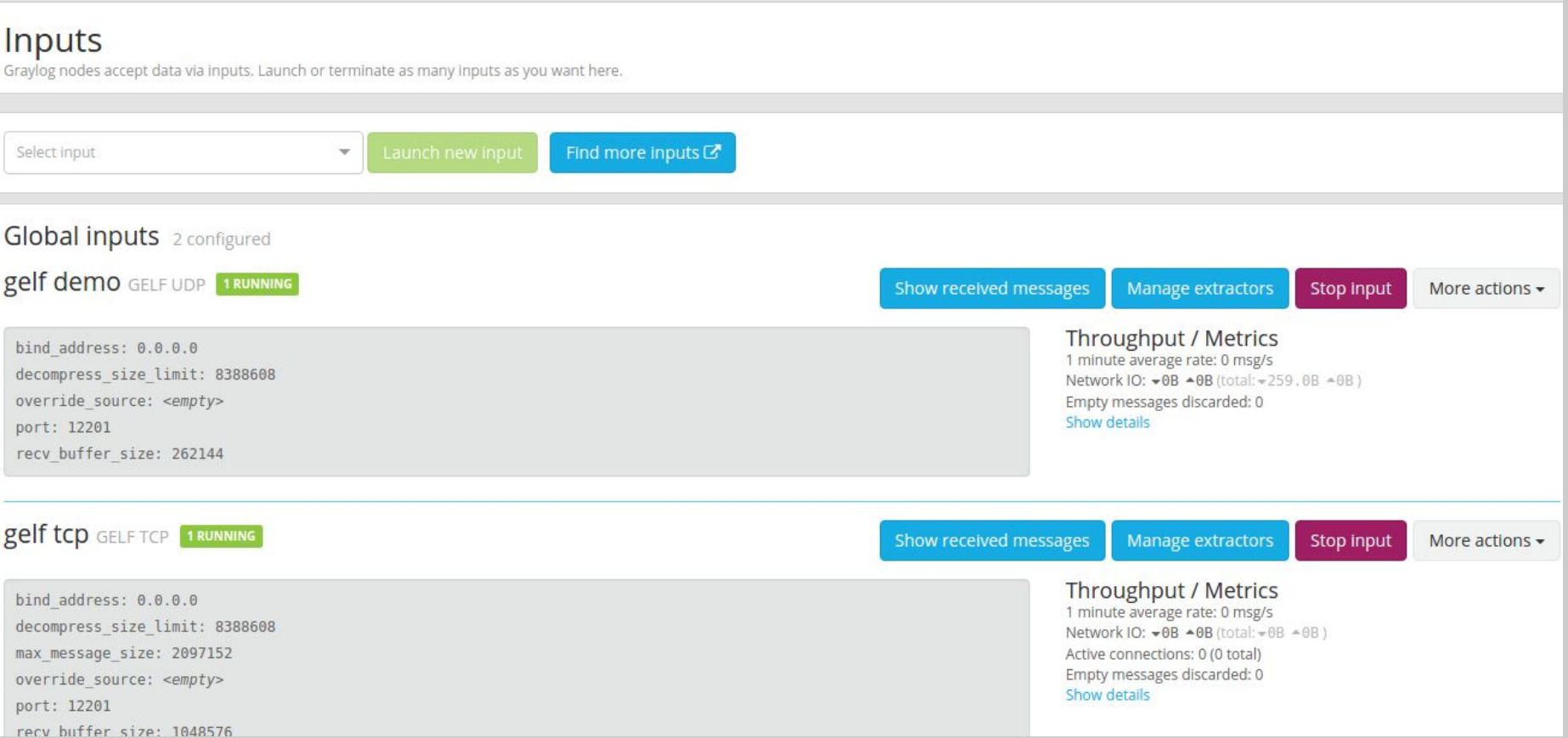
ElasticSearch indexes



ElasticSearch indexes

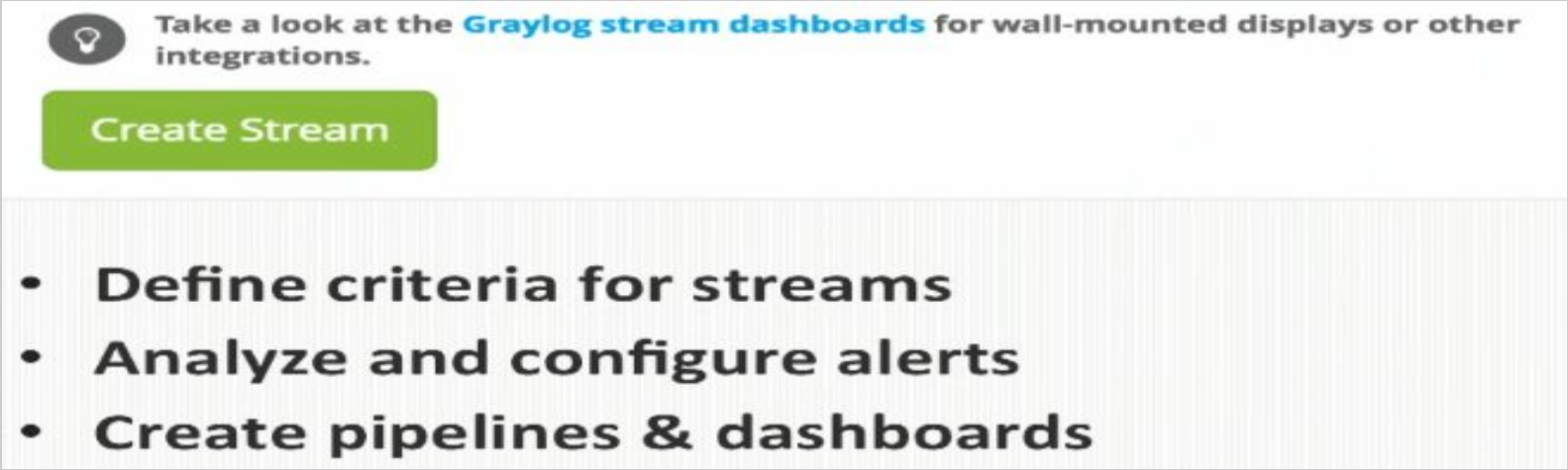


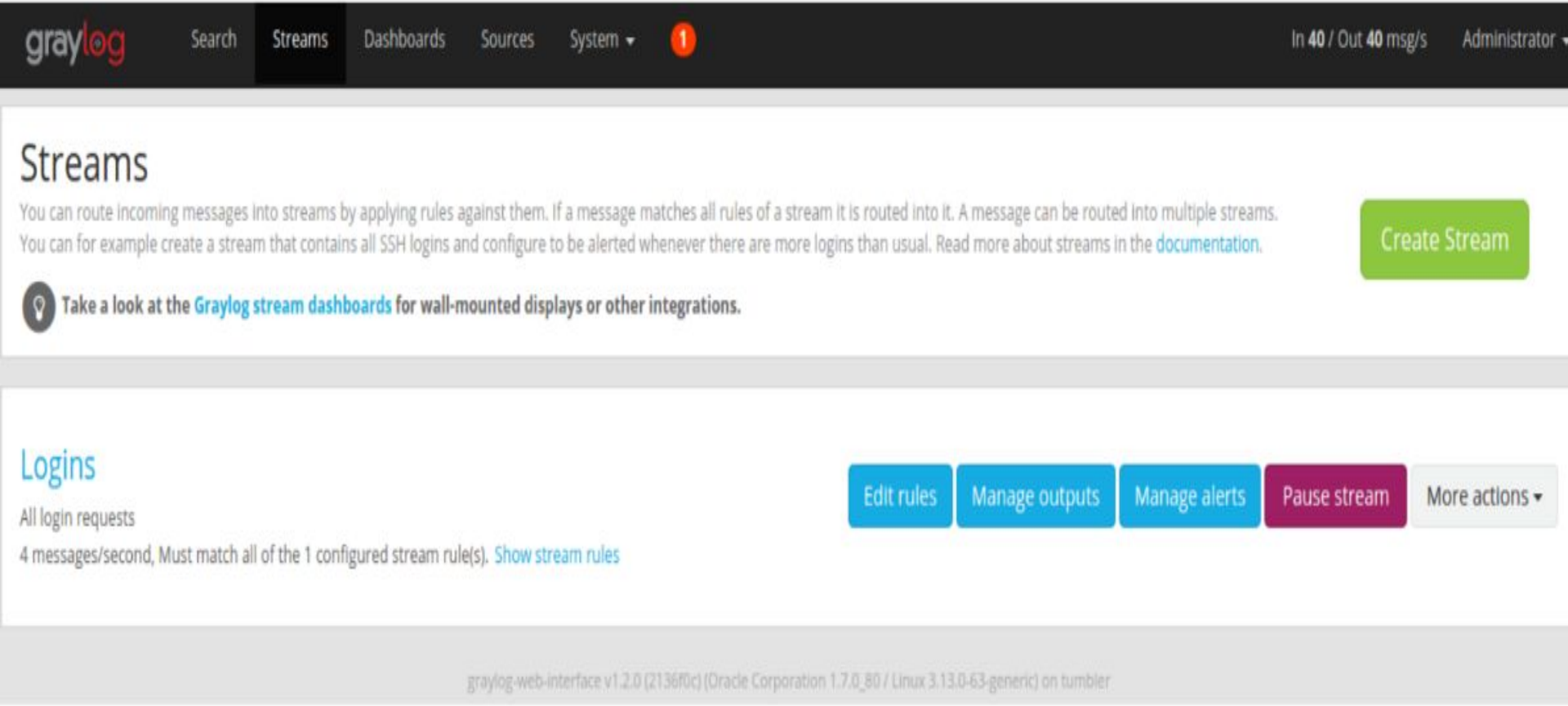
Inputs

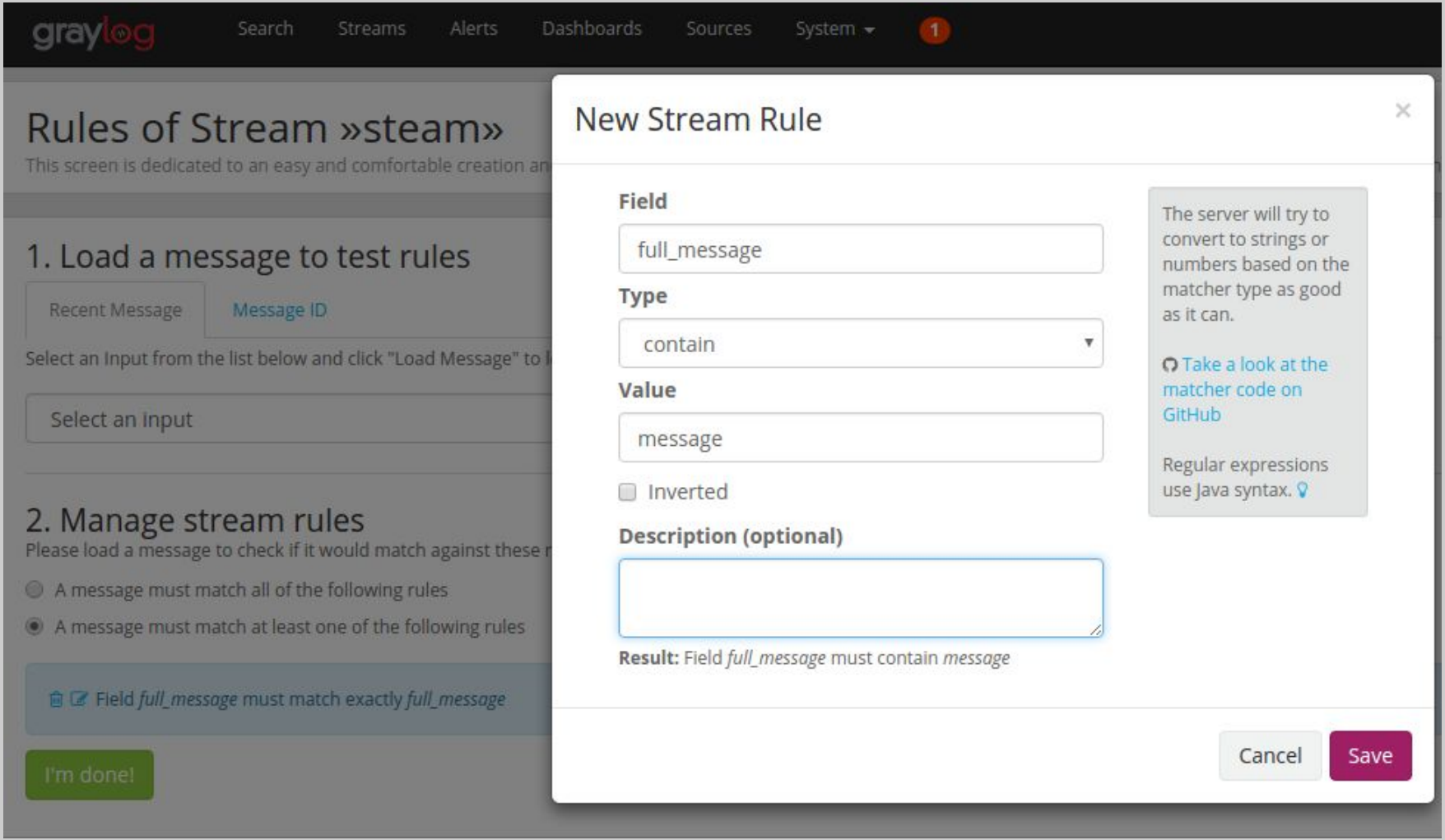

Inputs
Graylog nodes accept data via inputs. Launch or terminate as many inputs as you want here.
Select input
Launch new input
Find more inputs
Global inputs 2 configured
gelf demo GELF UDP 1 RUNNING
Show received messages
Manage extractors
Stop input
More actions
bind_address: 0.0.0.0
decompress_size_limit: 8388608
override_source: <empty>
port: 12201
recv_buffer_size: 262144
Throughput / Metrics
1 minute average rate: 0 msg/s
Network IO: 0B 0B (total: 259.0B 0B)
Empty messages discarded: 0
Show details
gelf tcp GELF TCP 1 RUNNING
Show received messages
Manage extractors
Stop input
More actions
bind_address: 0.0.0.0
decompress_size_limit: 8388608
max_message_size: 2097152
override_source: <empty>
port: 12201
recv_buffer_size: 1048576
Throughput / Metrics
1 minute average rate: 0 msg/s
Network IO: 0B 0B (total: 0B 0B)
Active connections: 0 (0 total)
Empty messages discarded: 0
Show details

Streams

- Incoming messages can be grouped
- Can be used for to assign user permissions
- Stream alerts can send out notifications




graylog
Search
Streams
Dashboards
Sources
System
1
In 40 / Out 40 msg/s
Administrator
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.
Logins
All login requests
4 messages/second, Must match all of the 1 configured stream rule(s). Show stream rules
Edit rules
Manage outputs
Manage alerts
Pause stream
More actions
graylog-web-interface v1.2.0 (2136f0c) (Oracle Corporation 1.7.0_80 / Linux 3.13.0-63-generic) on tumbler


graylog
Search
Streams
Alerts
Dashboards
Sources
System
1
Rules of Stream »steam»
This screen is dedicated to an easy and comfortable creation an
1. Load a message to test rules
Recent Message
Message ID
Select an Input from the list below and click "Load Message" to l
Select an Input
2. Manage stream rules
Please load a message to check if it would match against these r
A message must match all of the following rules
A message must match at least one of the following rules
Field full_message must match exactly full_message
I'm done!
New Stream Rule
Field
full_message
Type
contain
Value
message
Inverted
Description (optional)
Result: Field full_message must contain message
The server will try to convert to strings or numbers based on the matcher type as good as it can.
Take a look at the matcher code on GitHub
Regular expressions use Java syntax.
Cancel
Save

Graylog2 / **graylog2-server**

Watch 245 | Star 4,046 | Fork 606

Code | Issues 469 | Pull requests 13 | Projects 0 | Insights

Branch: **2.4** | Create new file | Upload files | Find file | History

graylog2-server / graylog2-server / src / main / java / org / graylog2 / streams / **matchers** /

This branch is 94 commits ahead, 136 commits behind master. Pull request | Compare

joschi committed with **dennisoelkers** Add support for arrays to "contains" stream rule (#3380) ... Latest commit 7bb61da on Jan 18, 2017

..

AlwaysMatcher.java	Create Default stream (#2881)	a year ago
ContainsMatcher.java	Add support for arrays to "contains" stream rule (#3380)	a year ago
ExactMatcher.java	Matching inverted exact/regex stream rules when field is not present (#…	2 years ago
FieldPresenceMatcher.java	Update license headers in Java source files	3 years ago
GreaterMatcher.java	Handle double values in greater/smaller stream matcher.	3 years ago
RegexMatcher.java	Matching inverted exact/regex stream rules when field is not present (#…	2 years ago
SmallerMatcher.java	Handle double values in greater/smaller stream matcher.	3 years ago
StreamRuleMatcher.java	Update license headers in Java source files	3 years ago

Streams to archive

Filter streams

Reset

- [ ] Select all available streams
- [ ] **Default** - Stream used by default for messages not matching another stream.
- [ ] **ALL** - All messages
- [x] **HTTP** - all HTTP Traffic (extracted from pipelines)
- [ ] **HoneyPot** - Stream from Honeypot
- [ ] **SSH (Info)** - Show accepted/failed SSH
- [x] **d8** - all from d8
- [ ] **mail** - all that coming with mail
- [x] **mysql** - all mysql logs

Alerts configuration for stream »Logins«

You can define thresholds on any message field or message count of a stream and be alerted based on this definition.

Learn more about alerts in the documentation.

Add new alert condition

Message count condition | Configure new alert condition

Configured alert conditions

Message count condition

Alert is triggered when there is less than 1 message in the last 3 minutes. Grace period: 10 minutes. Not including any messages in alert notification.

Edit condition | Delete condition

Callbacks

The following callbacks will be performed when this stream triggers an alert.

Select Callback Type | Add callback | Find more callbacks

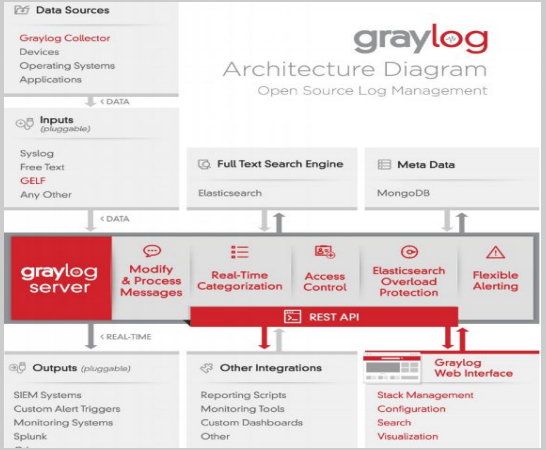
Email Alert Callback

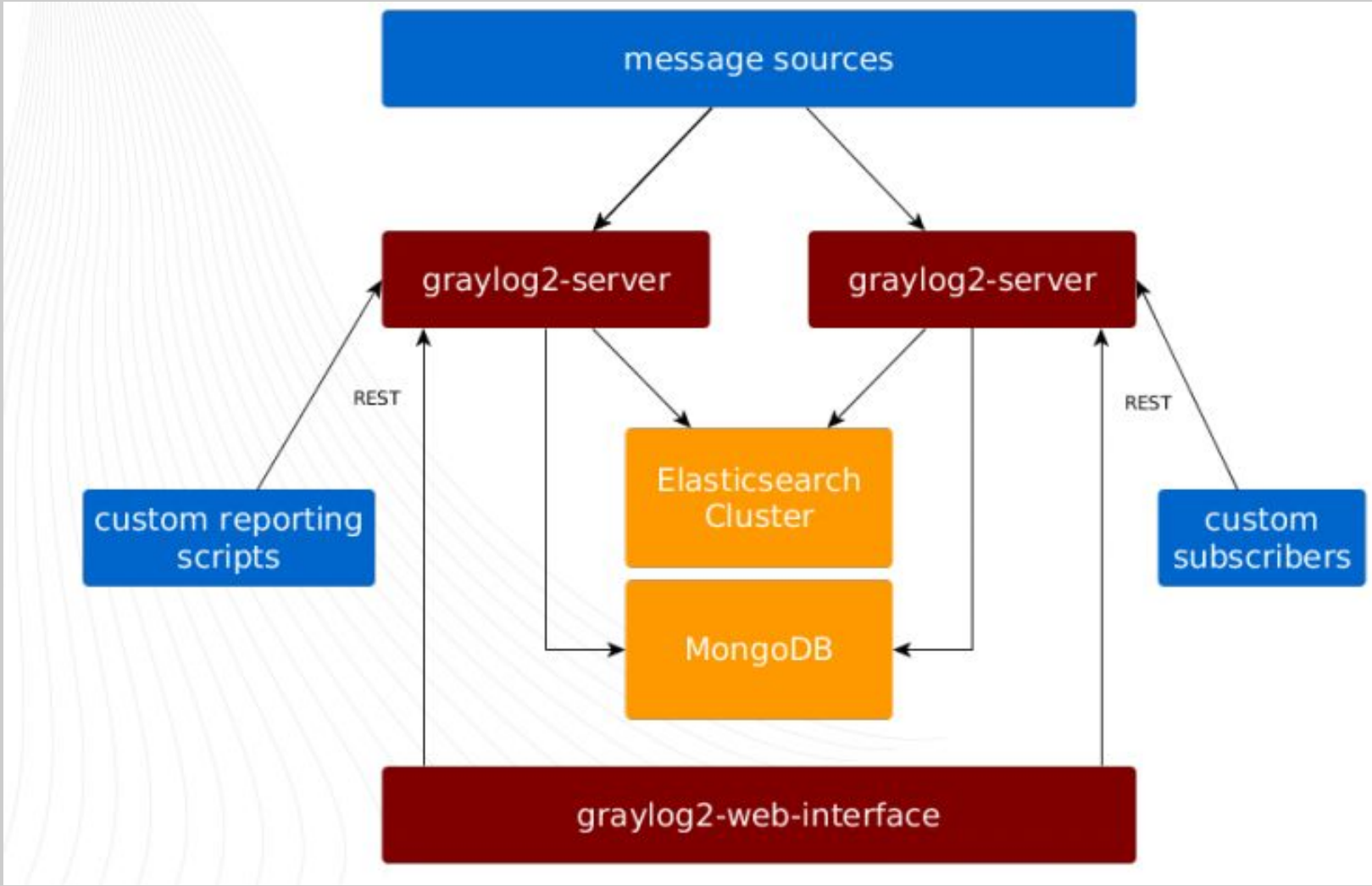
Executed once per triggered alert condition.

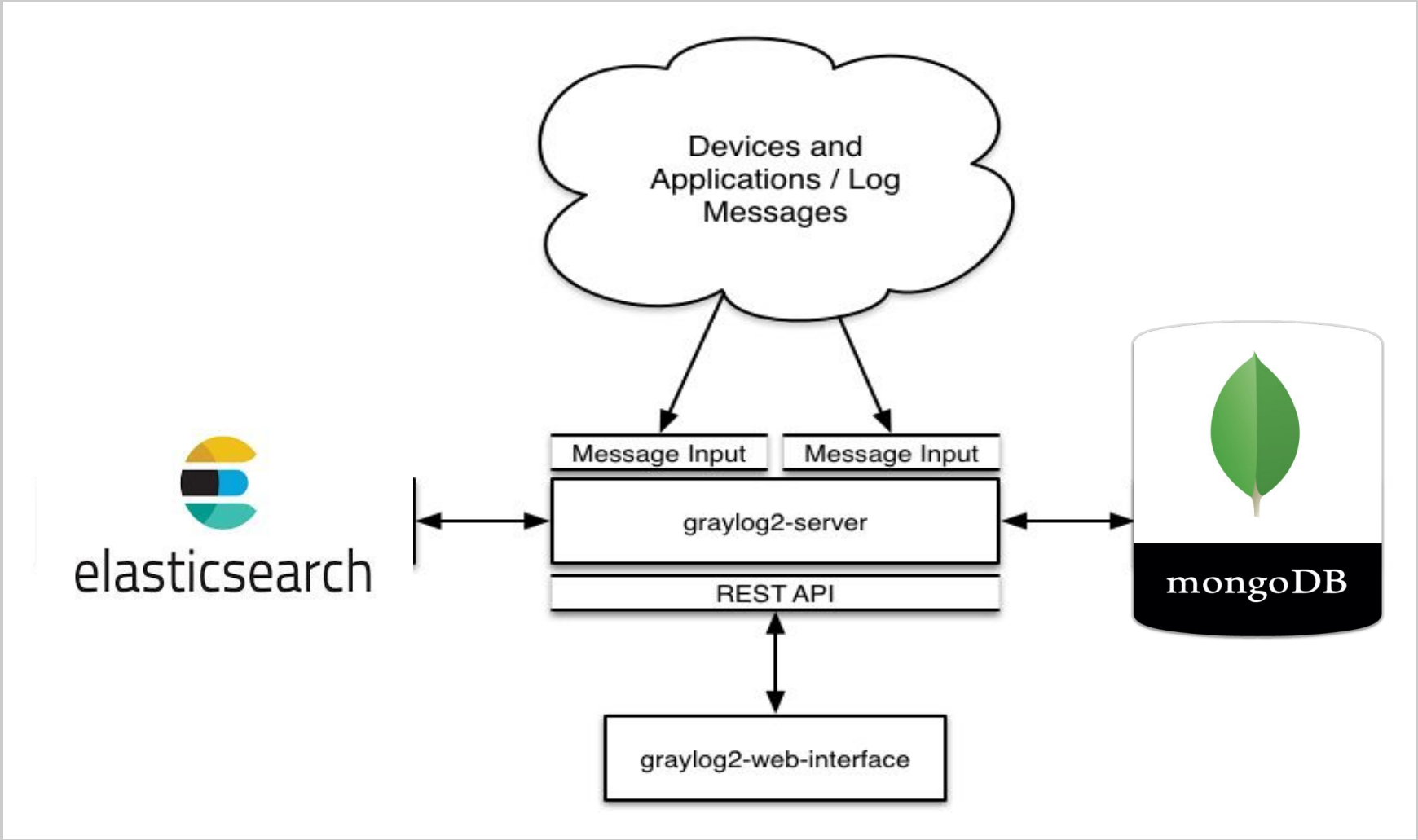
Edit callback | Delete callback

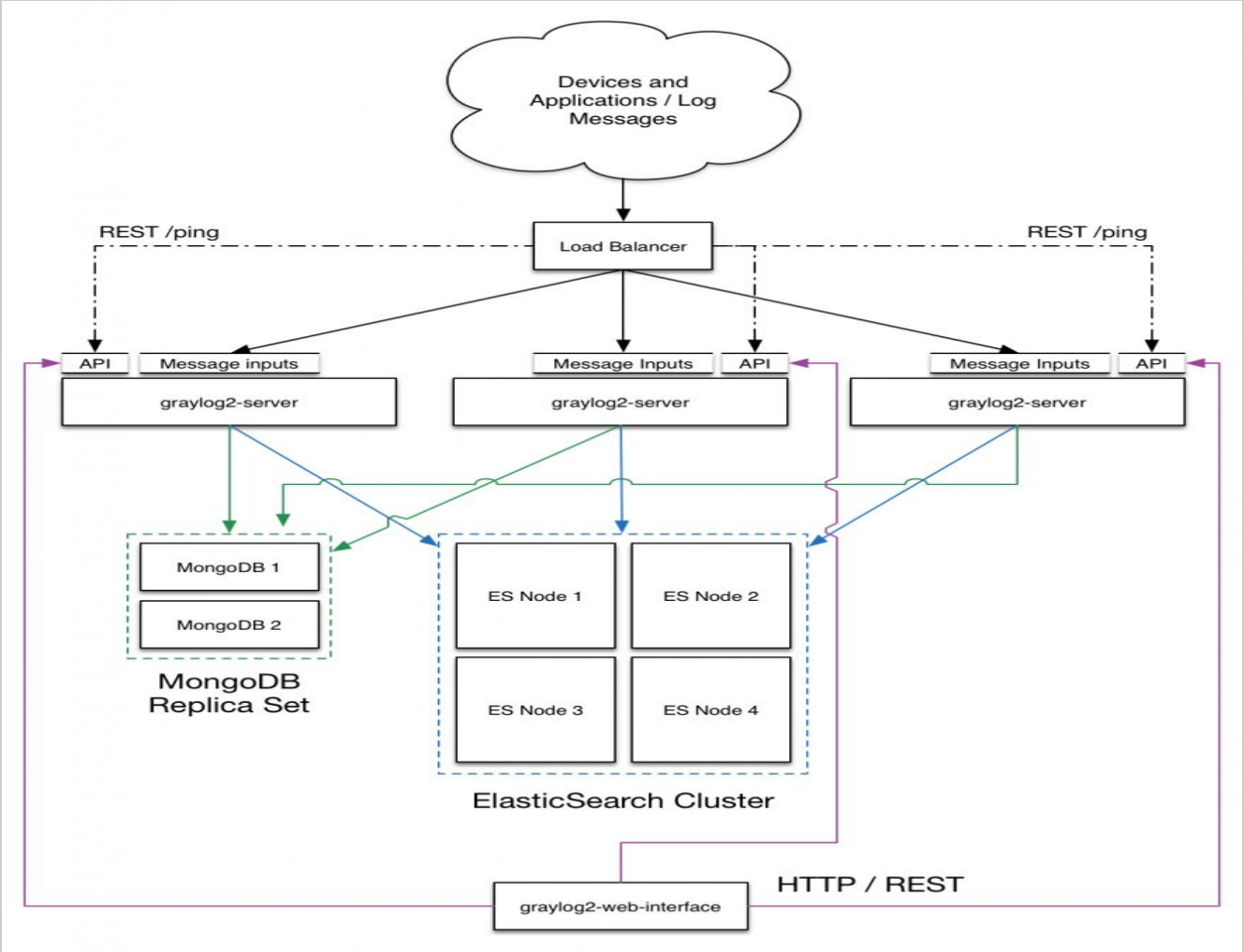
```
sender:          graylog@example.org
body:            ##########
                 Alert Description: ${check_result.resultDescription}
```

GrayLog architecture




message sources
graylog2-server
graylog2-server
REST
REST
Elasticsearch Cluster
MongoDB
custom reporting scripts
custom subscribers
graylog2-web-interface


Devices and Applications / Log Messages
Message Input
Message Input
graylog2-server
REST API
graylog2-web-interface
elasticsearch
mongoDB


Devices and Applications / Log Messages
REST /ping
Load Balancer
REST /ping
API
Message inputs
graylog2-server
Message Inputs
API
graylog2-server
Message Inputs
API
graylog2-server
MongoDB 1
MongoDB 2
MongoDB Replica Set
ES Node 1
ES Node 2
ES Node 3
ES Node 4
ElasticSearch Cluster
HTTP / REST
graylog2-web-interface

Connecting with Java

Sending log data to graylog

- Syslog
  – TCP, TCP+TLS, UDP, AMQP, Kafka
- **GELF**
  – TCP, TCP+TLS, UDP, HTTP, AMQP,Kafka
- Raw / Plain Text
  – TCP, TCP+TLS, UDP, AMQP, Kafka
- Collector
  – TCP, TCP+TLS

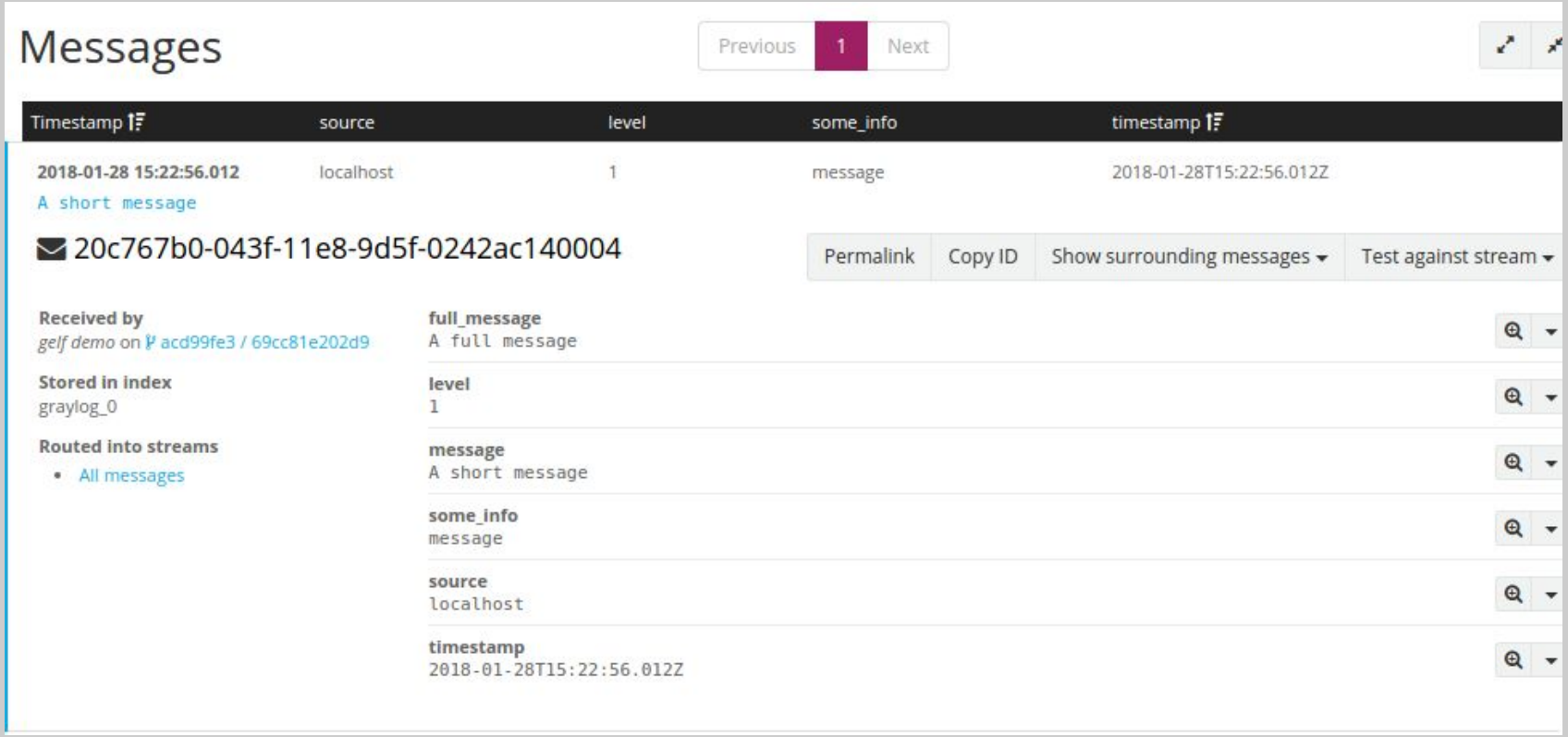
GELF

- Graylog Extended Log Format
- Logstash, fluentd, nxlog, Docker, ...
- Based in syslog and rsyslog
- JSON based format for sending structured data
- JSON Hash with mandatory fields:
  - host, version, short_message, full_message, timestamp, level

GELF document

```
{
"version": "1.1",
"timestamp": 1385053862.3072,
"host": "example.org",
"short_message": "A short message",
"full_message": "A full message,
"level": 1,
"_user_id": 9001,
“_http_response_code” : 500
"_some_env_var": "env_var"
}
```

Graylog message inspector



Apache Log4j 2

Apache Log4j 2 is an upgrade to Log4j that provides significant improvements over its predecessor, Log4j 1.x, and provides many of the improvements available in Logback while fixing some inherent problems in Logback's architecture.

Features

API Separation

The API for Log4j is separate from the implementation making it clear for application developers which classes and methods they can use while ensuring forward compatibility. This allows the Log4j team to improve the implementation safely and in a compatible manner.

Improved Performance

Log4j 2 contains next-generation Asynchronous Loggers based on the LMAX Disruptor library. In multi-threaded scenarios Asynchronous Loggers have 18 times higher throughput and orders of magnitude lower latency than Log4j 1.x and Logback. See Asynchronous Logging Performance for details. Otherwise, Log4j 2 significantly outperforms Log4j 1.x, Logback and java.util.logging, especially in multi-threaded applications. See Performance for more information.

SLF4J user manual

The Simple Logging Facade for Java (SLF4J) serves as a simple facade or abstraction for various logging frameworks, such as java.util.logging, logback a log4j. SLF4J allows the end-user to plug in the desired logging framework at *deployment* time. Note that SLF4J-enabling your library/application implies addition of only a single mandatory dependency, namely *slf4j-api-1.7.12.jar*.

SINCE 1.6.0 If no binding is found on the class path, then SLF4J will default to a no-operation implementation.

SINCE 1.7.0 Printing methods in the `Logger` interface now offer variants accepting varargs instead of `Object[]`. This change implies that SLF4J requires J or later. Under the hood the Java compiler transforms the varargs part in methods into `Object[]`. Thus, the Logger interface generated by the compiler indistinguishable in 1.7.x from its 1.6.x counterpart. It follows that SLF4J version 1.7.x is totally 100% no-ifs-or-buts compatible with SLF4J version 1.6.

SINCE 1.7.5 Significant improvement in logger retrieval times. Given the extent of the improvement, users are highly encouraged to migrate to SLF4J 1.7 later.

SINCE 1.7.9 By setting the `slf4j.detectLoggerNameMismatch` system property to true, SLF4J can automatically spot incorrectly named loggers.

Hello World

As customary in programming tradition, here is an example illustrating the simplest way to output "Hello world" using SLF4J. It begins by getting a logg the name "HelloWorld". This logger is in turn used to log the message "Hello World".

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class HelloWorld {
  public static void main(String[] args) {
    Logger logger = LoggerFactory.getLogger(HelloWorld.class);
    logger.info("Hello World");
  }
```

Logback Project

Logback is intended as a successor to the popular log4j project, picking up where log4j leaves off.

Logback's architecture is sufficiently generic so as to apply under different circumstances. At present time, logback is divided into three modules, logback-core, logback-classic and logback-access.

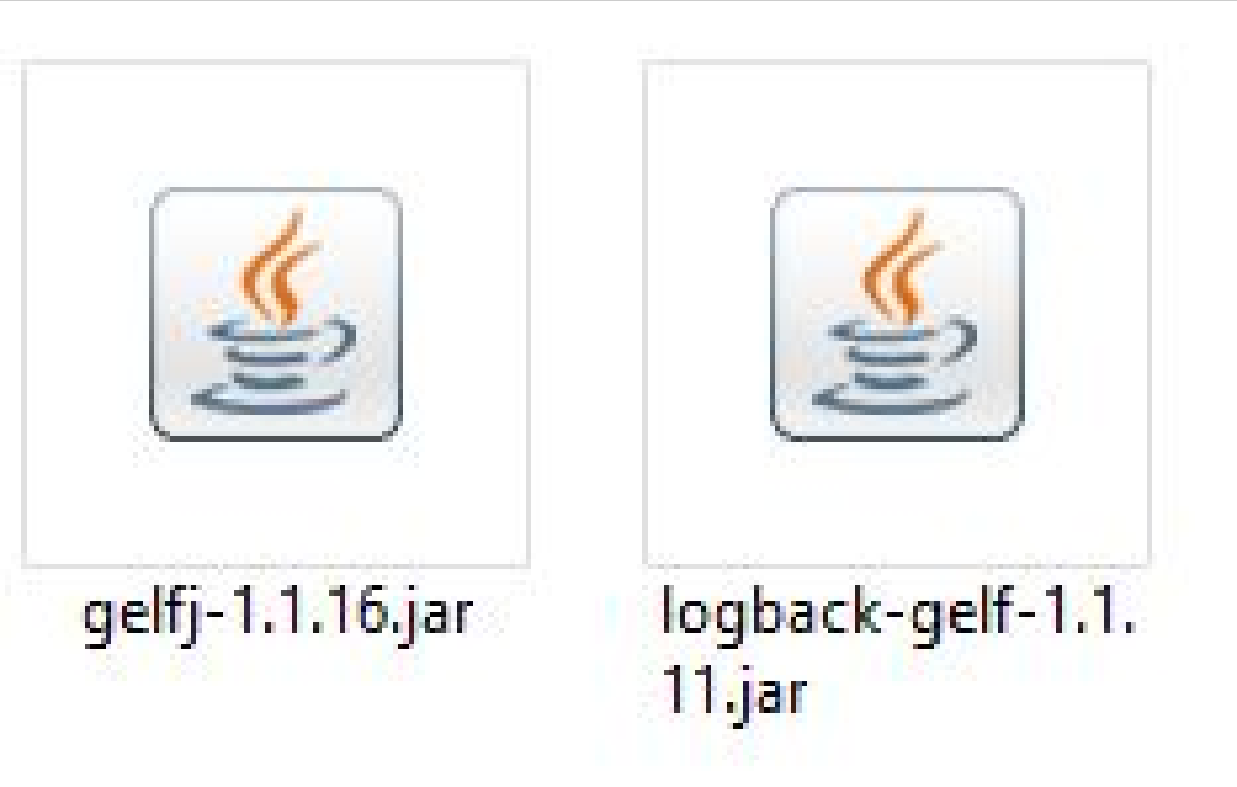
The logback-core module lays the groundwork for the other two modules. The logback-classic module can be assimilated to a significantly improved version of log4j. Moreover, logback-classic natively implements the SLF4J API so that you can readily switch back and forth between logback and other logging frameworks such as log4j or java.util.logging (JUL).

The logback-access module integrates with Servlet containers, such as Tomcat and Jetty, to provide HTTP-access log functionality. Note that you could easily build your own module on top of logback-core.

Sister projects

The logback-audit project is designed for processing logging events having long-term business significance. Logback-audit is based on logback-core.

Jars



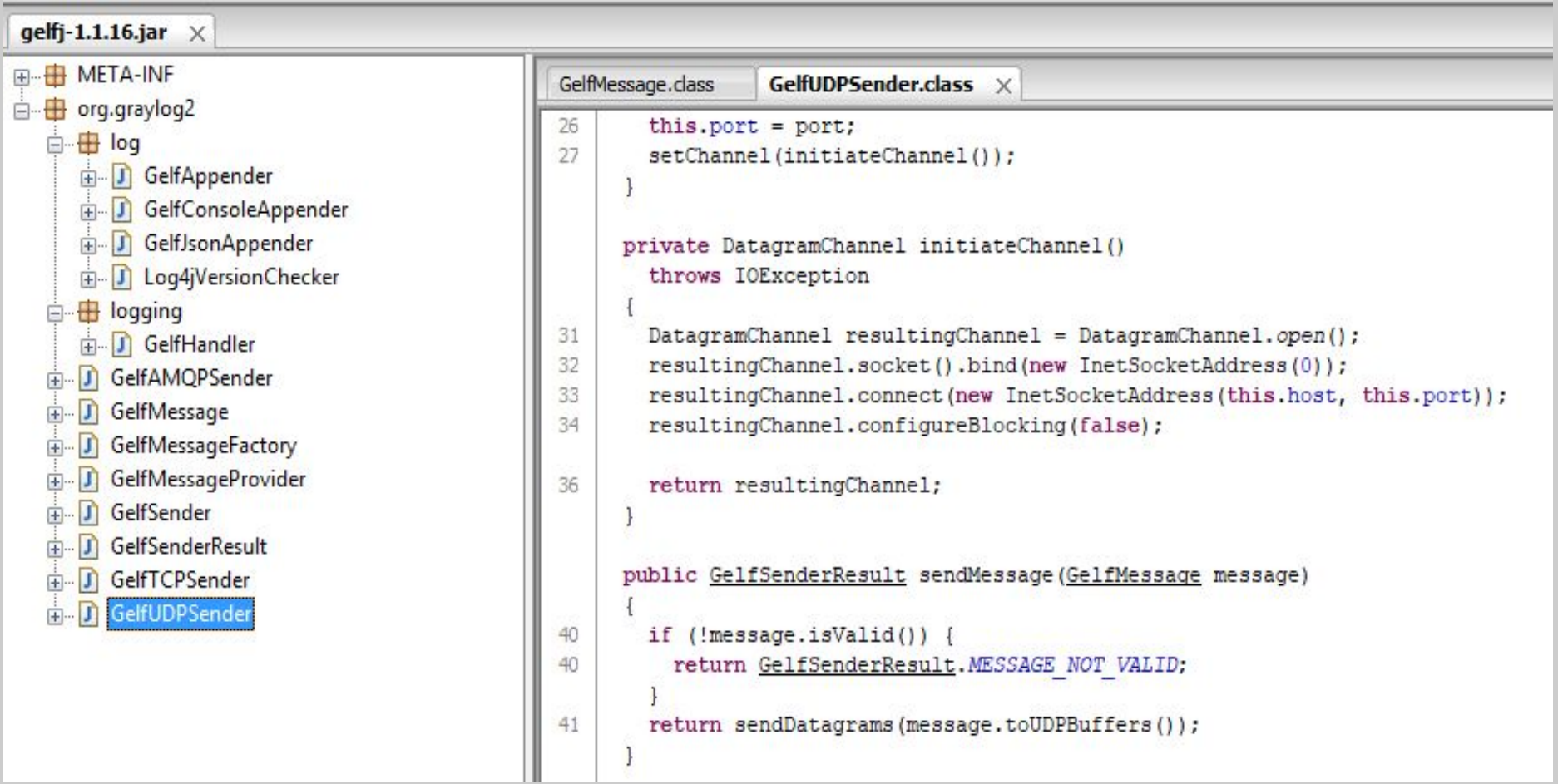
Gelfj » 1.1.16

GELF implementation in Java and log4j appender without any dependencies.

License	MIT
HomePage	https://github.com/t0xa/gelfj
Date	(Jan 08, 2018)
Files	pom (7 KB) jar (28 KB) View All
Repositories	Central
Used By	1 artifacts

Maven | Gradle | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/org.graylog2/gelfj -->
<dependency>
    <groupId>org.graylog2</groupId>
    <artifactId>gelfj</artifactId>
    <version>1.1.16</version>
</dependency>
```


gelfj-1.1.16.jar
META-INF
org.graylog2
log
GelfAppender
GelfConsoleAppender
GelfJsonAppender
Log4jVersionChecker
logging
GelfHandler
GelfAMQPSender
GelfMessage
GelfMessageFactory
GelfMessageProvider
GelfSender
GelfSenderResult
GelfTCPSender
GelfUDPSender
GelfMessage.class
GelfUDPSender.class
this.port = port;
setChannel(initiateChannel());
}
private DatagramChannel initiateChannel()
throws IOException
{
DatagramChannel resultingChannel = DatagramChannel.open();
resultingChannel.socket().bind(new InetSocketAddress(0));
resultingChannel.connect(new InetSocketAddress(this.host, this.port));
resultingChannel.configureBlocking(false);
return resultingChannel;
}
public GelfSenderResult sendMessage(GelfMessage message)
{
if (!message.isValid()) {
return GelfSenderResult.MESSAGE_NOT_VALID;
}
return sendDatagrams(message.toUDPBuffers());
}

```
<appender name="graylog2" class="org.graylog2.log.GelfAppender">
    <param name="graylogHost" value="192.168.0.201"/>
    <param name="originHost" value="my.machine.example.com"/>
    <param name="extractStacktrace" value="true"/>
    <param name="addExtendedInformation" value="true"/>
    <param name="facility" value="gelf-java"/>
    <param name="Threshold" value="INFO"/>
    <param name="additionalFields" value="{'environment': 'DEV', 'application':
'MyAPP'}"/>
</appender>
```

```
private void initSender(){
try{
// Sender is UDP
gelfSender = new GelfUDPSender(getServer(), getInputPort());
}
catch (IOException e) {
e.printStackTrace();
}
}
```

```
private boolean sendGelfMessage(GelfMessage message){
boolean result = false;
message.setHost(getClient());
// validate and send message
if (message.isValid()) {
result = getGelfSender().sendMessage(message);
}
return result;
}
```

```
public boolean sendMessage(String title, String description){
// compose message
GelfMessage message = new GelfMessage(
title, description, new Date().getTime(), getAlertLevel());
return sendGelfMessage(message);
}
```


```
public static void main(String[] args){
ApplicationContext context = new ClassPathXmlApplicationContext(
"SpringBeans.xml");
GraylogService graylogService = (GraylogService) context.getBean("graylogService");
graylogService.sendMessage("Test message", "This message is a test message");
}


<bean id="graylogService" class="graylogtest.GraylogService">
<property name="server" value="server"/>
<property name="inputPort" value="port"/>
<property name="alertLevel" value="6"/> <!-- 6 means INFO -->
</bean>
```

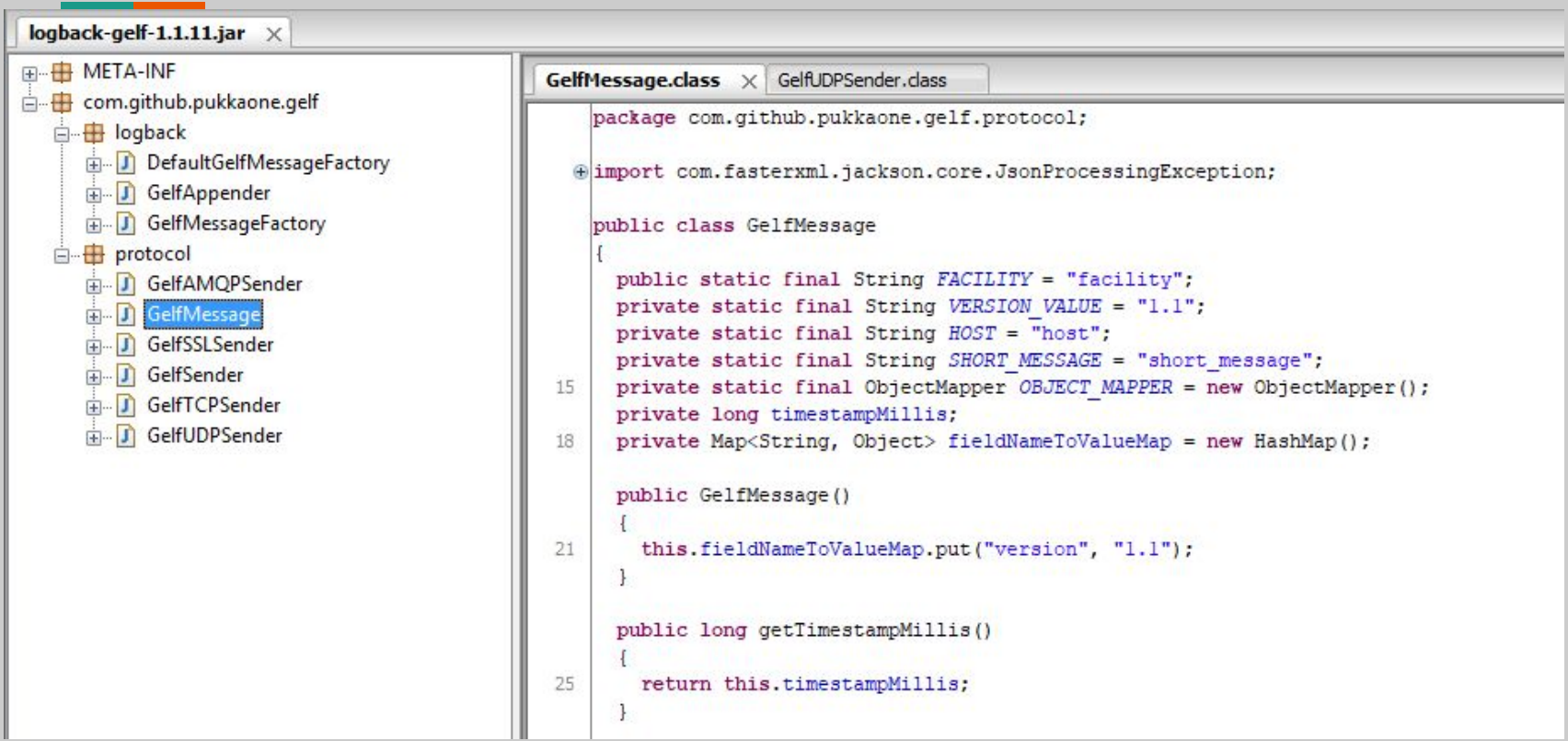
LogBack

- https://github.com/pukkaone/logback-gelf
- JDK >= 1.7

Add the following dependency to your project:

```
<dependency>
  <groupId>com.github.pukkaone</groupId>
  <artifactId>logback-gelf</artifactId>
  <version>1.1.11</version>
</dependency>
```

LogBack


logback-gelf-1.1.11.jar
META-INF
com.github.pukkaone.gelf
logback
DefaultGelfMessageFactory
GelfAppender
GelfMessageFactory
protocol
GelfAMQPSender
GelfMessage
GelfSSLSender
GelfSender
GelfTCPSender
GelfUDPSender
GelfMessage.class
GelfUDPSender.class
package com.github.pukkaone.gelf.protocol;
import com.fasterxml.jackson.core.JsonProcessingException;
public class GelfMessage
{
public static final String FACILITY = "facility";
private static final String VERSION_VALUE = "1.1";
private static final String HOST = "host";
private static final String SHORT_MESSAGE = "short_message";
private static final ObjectMapper OBJECT_MAPPER = new ObjectMapper();
private long timestampMillis;
private Map<String, Object> fieldNameToValueMap = new HashMap();
public GelfMessage()
{
this.fieldNameToValueMap.put("version", "1.1");
}
public long getTimestampMillis()
{
return this.timestampMillis;
}

LogBack appender

Configure a logback appender to send by UDP (XML configuration format):

```
<appender name="GRAYLOG" class="com.github.pukkaone.gelf.logback.GelfAppender">
  <graylogHost>graylog.example.com</graylogHost>
  <originHost>my.machine.example.com</originHost>
  <levelIncluded>true</levelIncluded>
  <locationIncluded>false</locationIncluded>
  <loggerIncluded>true</loggerIncluded>
  <markerIncluded>false</markerIncluded>
  <mdcIncluded>false</mdcIncluded>
  <threadIncluded>false</threadIncluded>
  <facility>gelf-java</facility>
  <additionalField>application=MyApplication</additionalField>
  <additionalField>environment=development</additionalField>
</appender>
```

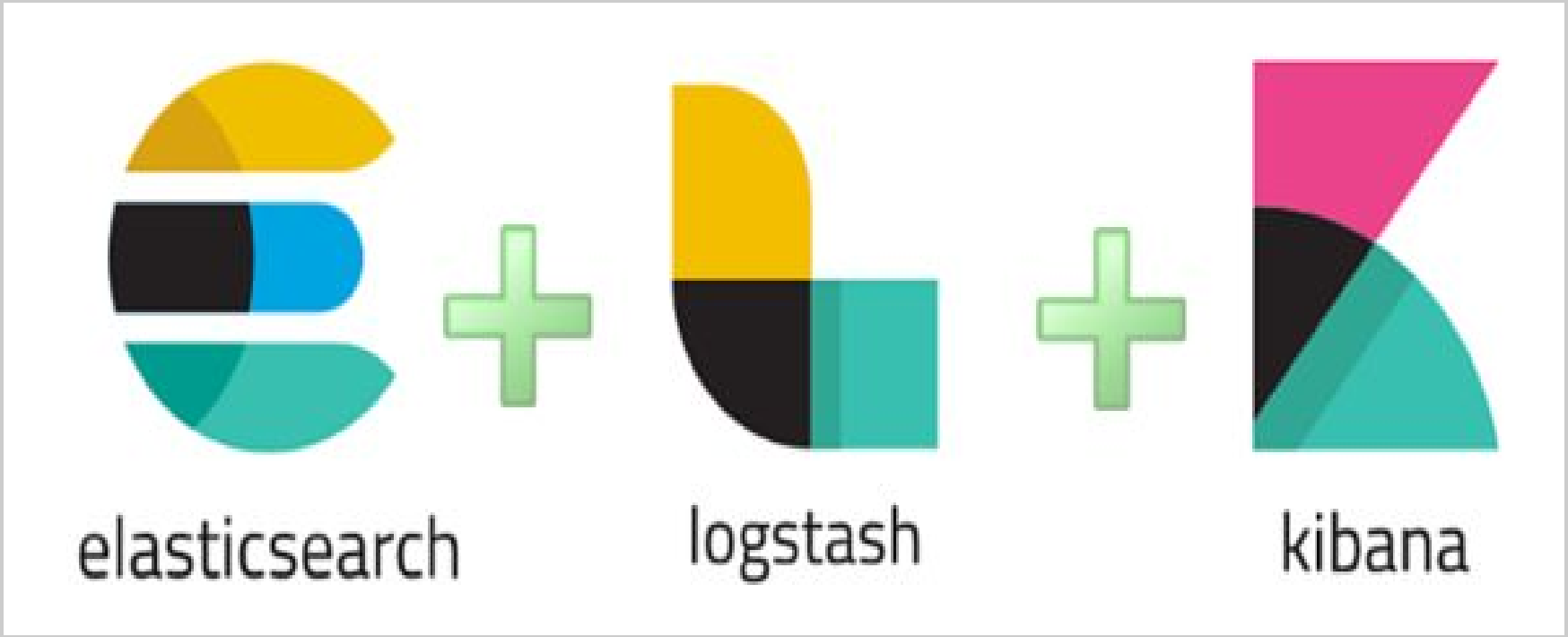
GraylogRestInterface

```
public class GraylogRestInterface {

    private final RestTemplate restTemplate = new RestTemplate();

    private final UriComponentsBuilder uriBuilder =
UriComponentsBuilder.newInstance()
            .scheme("http").host("localhost").port(12900);

     public void logEvent(GelfMessage message) {
        HttpEntity<GelfMessage> entity = new HttpEntity<>(message,
buildHeaders());

restTemplate.postForEntity(uriBuilder.cloneBuilder().port(12202).path("gelf").
toUriString(), entity, null);
    }
  }
```

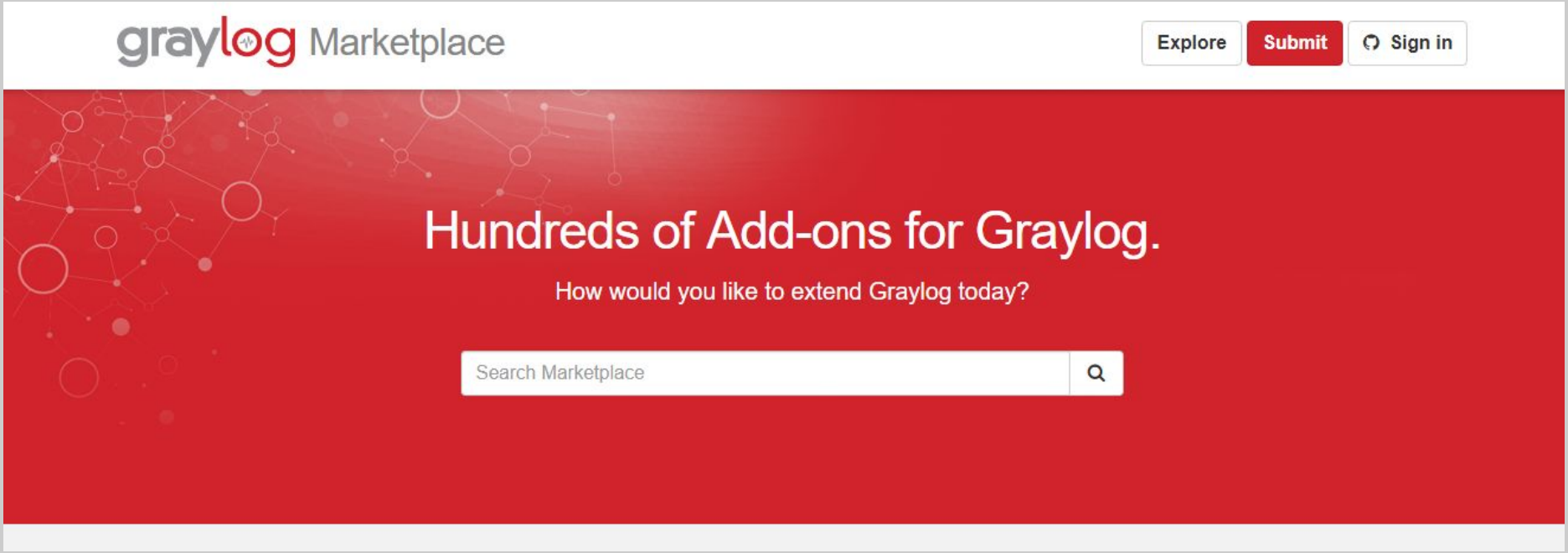
GelfMessage

```
// Log message via Graylog HTTP Input
GelfMessage message = new GelfMessage();
message.setShortMessage("Short message");
message.setFullMessage("Full message");
message.getAdditionalProperties().put("elapsed_time", timer.stop().
elapsed(TimeUnit.MICROSECONDS));
graylog.logEvent(message);
```

Connecting with other services


elasticsearch
logstash
kibana


RabbitMQ


graylog Marketplace
Explore
Submit
Sign in
Hundreds of Add-ons for Graylog.
How would you like to extend Graylog today?
Search Marketplace

Sending syslog via AMQP into Graylog

Other Solutions

How to use send Syslog messages via AMQP to Graylog

logstash-forwarder rsyslog rabbitMQ AMQP

jalogisch

View on Github 0 3

Published	Last Push	Marketplace Rating	Discussion
27 May 06:49	08 Sep 06:10	No rating yet	0 Comments

Sending syslog via KAFKA into Graylog

Other Solutions

This Guide will give you little help on using Graylog with Kafka Input to get Syslog Data

syslog kafka

jalogisch

View on Github 0 5

Published	Last Push	Marketplace Rating	Discussion
12 Sep 03:47	12 Sep 03:45	No rating yet	0 Comments

References

- http://docs.graylog.org/en/2.4/index.html
- https://github.com/Graylog2/graylog-docker
- https://hub.docker.com/r/graylog2/graylog/
- http://docs.graylog.org/en/2.4/pages/installation/docker.html
- http://docs.graylog.org/en/2.4/pages/faq.html

Thanks!

Contact:

@jmortegac

jmortega.github.io

about.me/jmortegac