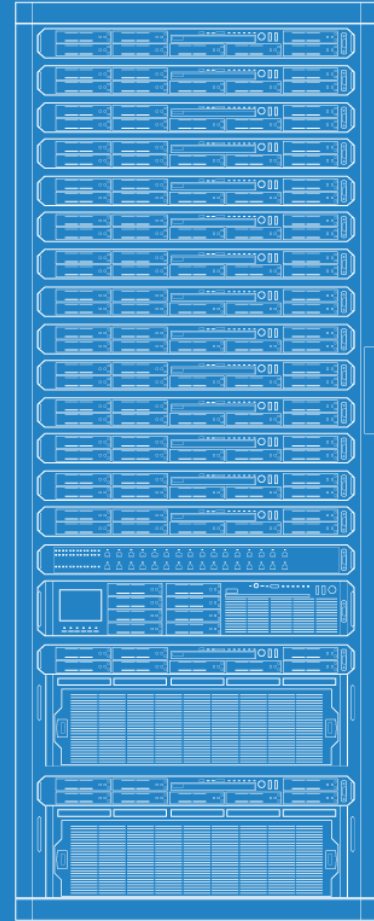
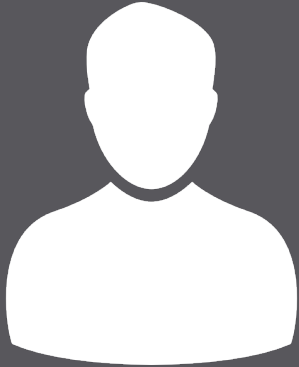


# LOGGING DOCKER USING SYSLOG-NG

vDay 2017  
Peter Czanik / Balabit



# ABOUT ME



- Peter Czanik from Hungary
- Evangelist at Balabit: syslog-ng upstream
- syslog-ng packaging, support, advocacy

---

Balabit is an IT security company with development HQ in Budapest, Hungary

Over 200 employees: the majority are engineers

# OVERVIEW

- What is syslog-ng
- The four roles of syslog-ng
- Central syslog-ng in Docker
- Logging Docker infrastructure and containers
- Configuring syslog-ng

# syslog-ng

## Logging

Recording events, such as:

```
Jan 14 11:38:48 linux-0jbu sshd[7716]: Accepted publickey for root  
from 127.0.0.1 port 48806 ssh2
```

## syslog-ng

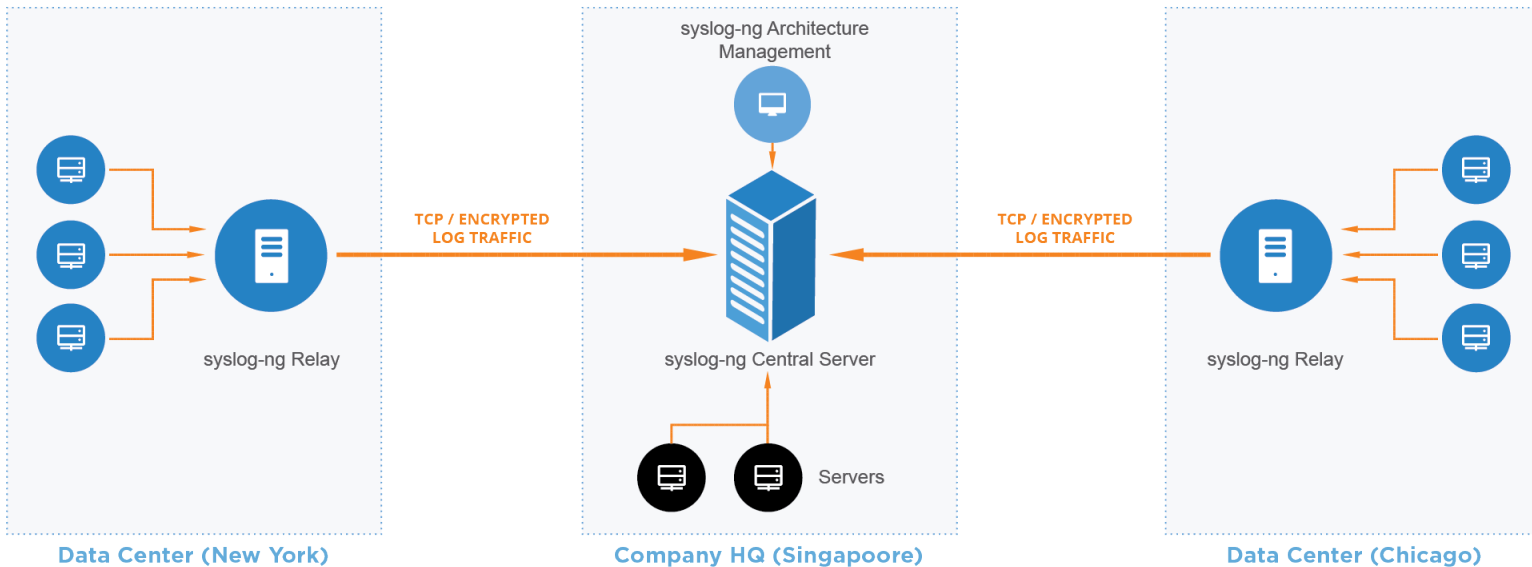
Enhanced logging daemon with a focus on high-performance central log collection.

# WHY CENTRAL LOGGING?

**EASE OF USE**  
one place to check  
instead of many

**AVAILABILITY**  
even if the sender  
machine is down

**SECURITY**  
logs are available even  
if sender machine  
is compromised



# MAIN SYSLOG-NG ROLES



collector



processor



filter



storage  
(or forwarder)

# ROLE: DATA COLLECTOR

Collect system and application logs together:  
contextual data for either side

## **A wide variety of platform-specific sources:**

- /dev/log & co
- Journal, Sun streams

## **Receive syslog messages over the network:**

- Legacy or RFC5424, UDP/TCP/TLS

## **Logs or any kind of text data from applications:**

- Through files, sockets, pipes, application output, etc.

# ROLE: PROCESSING

## **Classify, normalize and structure logs with built-in parsers:**

- CSV-parser, DB-parser (PatternDB), JSON parser, key=value parser, python parser and more to come

## **Rewrite messages:**

- For example anonymization

## **Reformatting messages using templates:**

- Destination might need a specific format (ISO date, JSON, etc.)

## **Enrich data:**

- GeolP
- Additional fields based on message content



# ROLE: DATA FILTERING

## Main uses:

- Discarding surplus logs (not storing debug level messages)
- Message routing (login events to SIEM)

## Many possibilities:

- Based on message content, parameters or macros
- Using comparisons, wildcards, regular expressions and functions
- Combining all of these with Boolean operators

# ROLE: DESTINATIONS

## “TRADITIONAL”

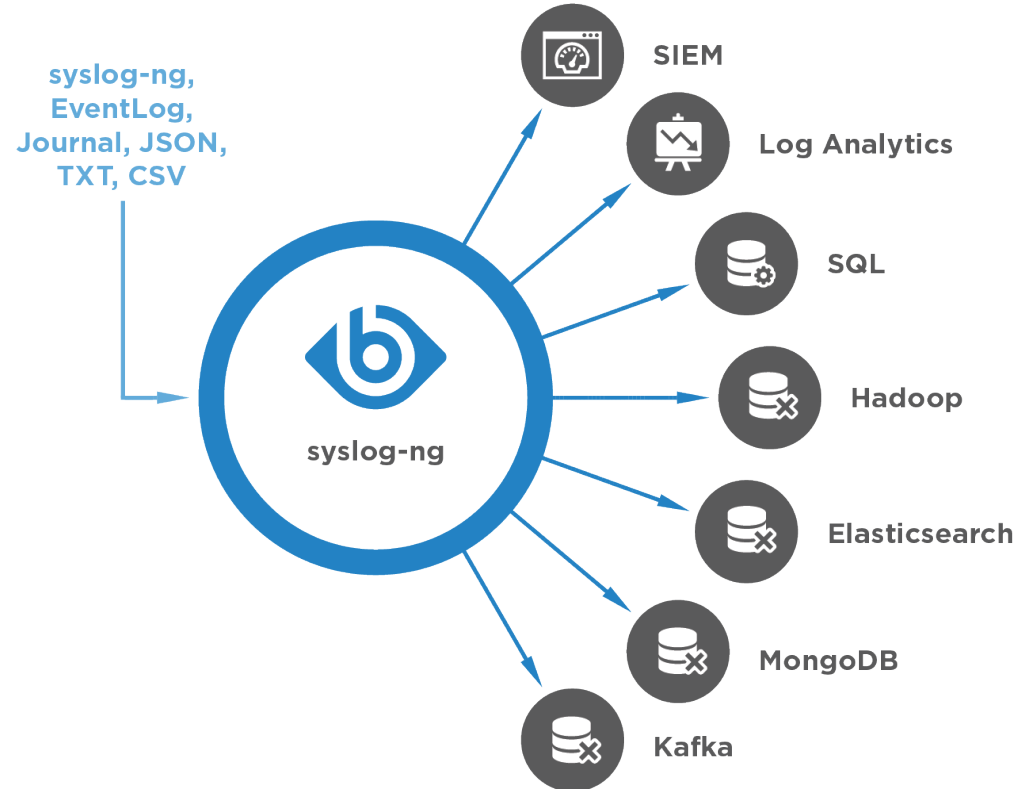
- File, network, TLS, SQL, etc.

## “BIG DATA”

- Hadoop
- MongoDB
- Elasticsearch
- Kafka

## “OTHERS”

- HTTP(S)
- Java / Python



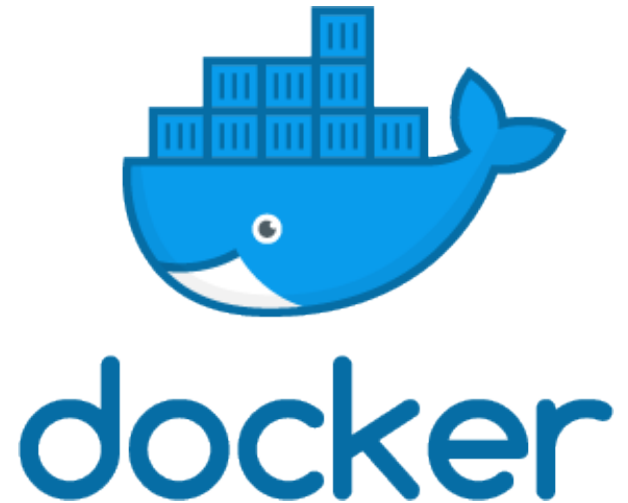
# Docker

## Containers

- Many implementations before Docker (FreeBSD jail, Solaris Zones and others)
- Docker brought ease of use

## syslog-ng in Docker

- Central server
- Docker infrastructure logs
- Container logs
- Can combine roles



# syslog-ng images

## Images

- balabit/syslog-ng
  - based on Ubuntu + latest release, all features
- Certified RHEL container with most features
- Many more images on Docker hub
  - Alpine-based images are very small

# Central server

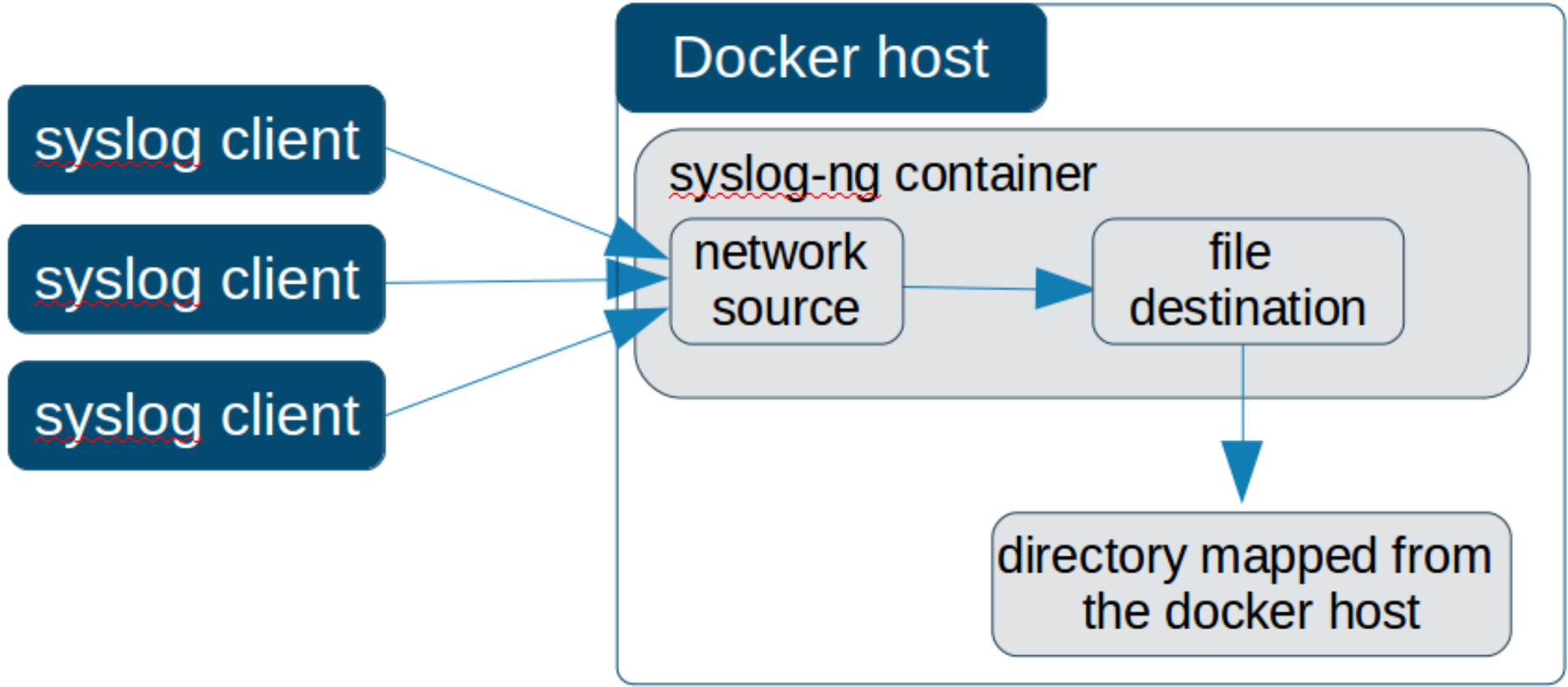
## Simple

- Almost like on the host
- Network sources
- Use any filter/parser/destination supported by the image

## Starting the syslog-ng container for a test

- `docker run -it -v /data/syslog-ng/conf/syslog-ng.conf:/etc/syslog-ng/syslog-ng.conf -v /data/syslog-ng/logs:/var/log -p 514:514 -p 601:601 --name syslog-ng balabit/syslog-ng:latest -edv`

# Central server



# Central server

## A simple configuration

```
@version: 3.12
source s_net {
    udp( ip("0.0.0.0") );
    syslog( ip("0.0.0.0") );
};
destination d_file {
    file("/var/log/syslog");
};
log {source(s_net); destination(d_file);};
```

# Docker infrastructure & container logs

## Using the Docker journal logging driver

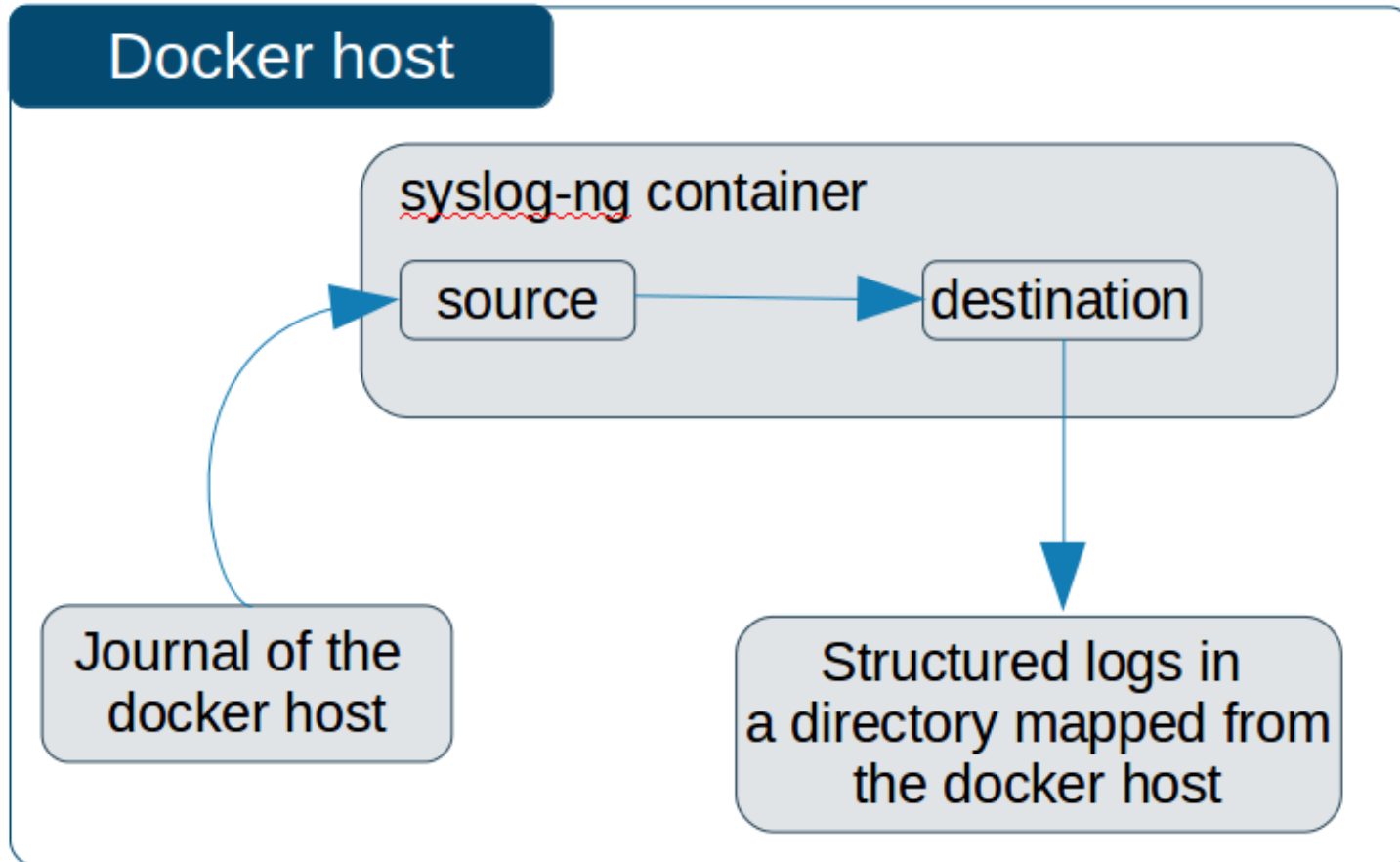
- “docker logs” + full featured remote log collection with syslog-ng
- Host / Docker infrastructure logs
- Container logs (only the stdout of containers)
- Name-value pairs with container identifiers

## Starting syslog-ng

- `docker run -ti -v /etc/machine-id:/etc/machine-id -v /data/syslog-ng/conf/journal.conf:/etc/syslog-ng/syslog-ng.conf -v /data/syslog-ng/logs:/var/log -v /var/log/journal:/var/log/journal --name journal balabit/syslog-ng:latest`



# Docker infrastructure & container logs



# Docker infrastructure & container logs

```
@version: 3.12
source s_journal {
    systemd-journal(prefix("journal."));
};
source s_internal { internal(); };
destination d_int { file("/var/log/int"); };
destination d_file {
    file("/var/log/journal.json" template("${format_json --scope rfc5424 --key
journal.*)\n\n"));
};
log {source(s_journal); destination(d_file); };
log {source(s_internal); destination(d_int); };
```

# Logs from other containers

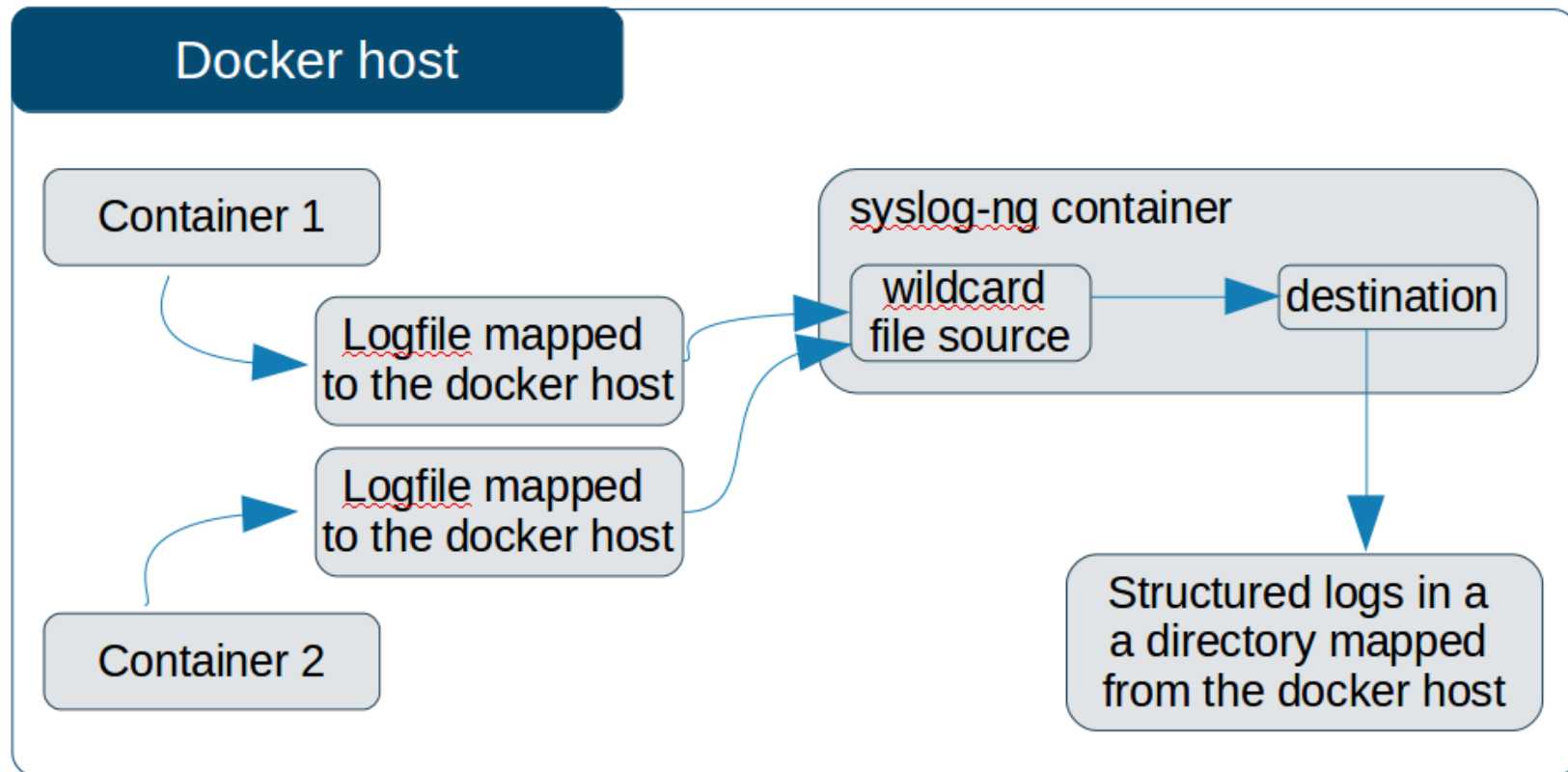
## Volumes

- Not all software can log to stdout
- Share log directories using volumes
- Wildcard-file source

## Pipes

- Using volumes
- No need for log rotation

# Logs from other containers



# CONFIGURATION



- “Don't Panic”
- Simple and logical, even if it looks difficult at first
- Pipeline model:
  - Many different building blocks (sources, destinations, filters, parsers, etc.)
  - Connected into a pipeline using “log” statements

# syslog-ng.conf: global options

```
@version:3.12
```

```
@include "scl.conf"
```

```
# this is a comment :)
```

```
options {
```

```
    flush_lines (0);
```

```
# [...]
```

```
    keep_hostname (yes);
```

```
};
```

# syslog-ng.conf: sources

```
source s_sys {  
    system();  
    internal();  
};
```

```
source s_net {  
    udp(ip(0.0.0.0) port(514));  
};
```

# syslog-ng.conf: destinations

```
destination d_mesg { file("/var/log/messages"); };
destination d_es {
  elasticsearch(
    index("syslog-ng_${YEAR}.${MONTH}.${DAY}")
    type("test")
    cluster("syslog-ng")
    template("${format-json --scope rfc3164 --scope nv-pairs --exclude R_DATE --key ISODATE}\n");
  );
};
```



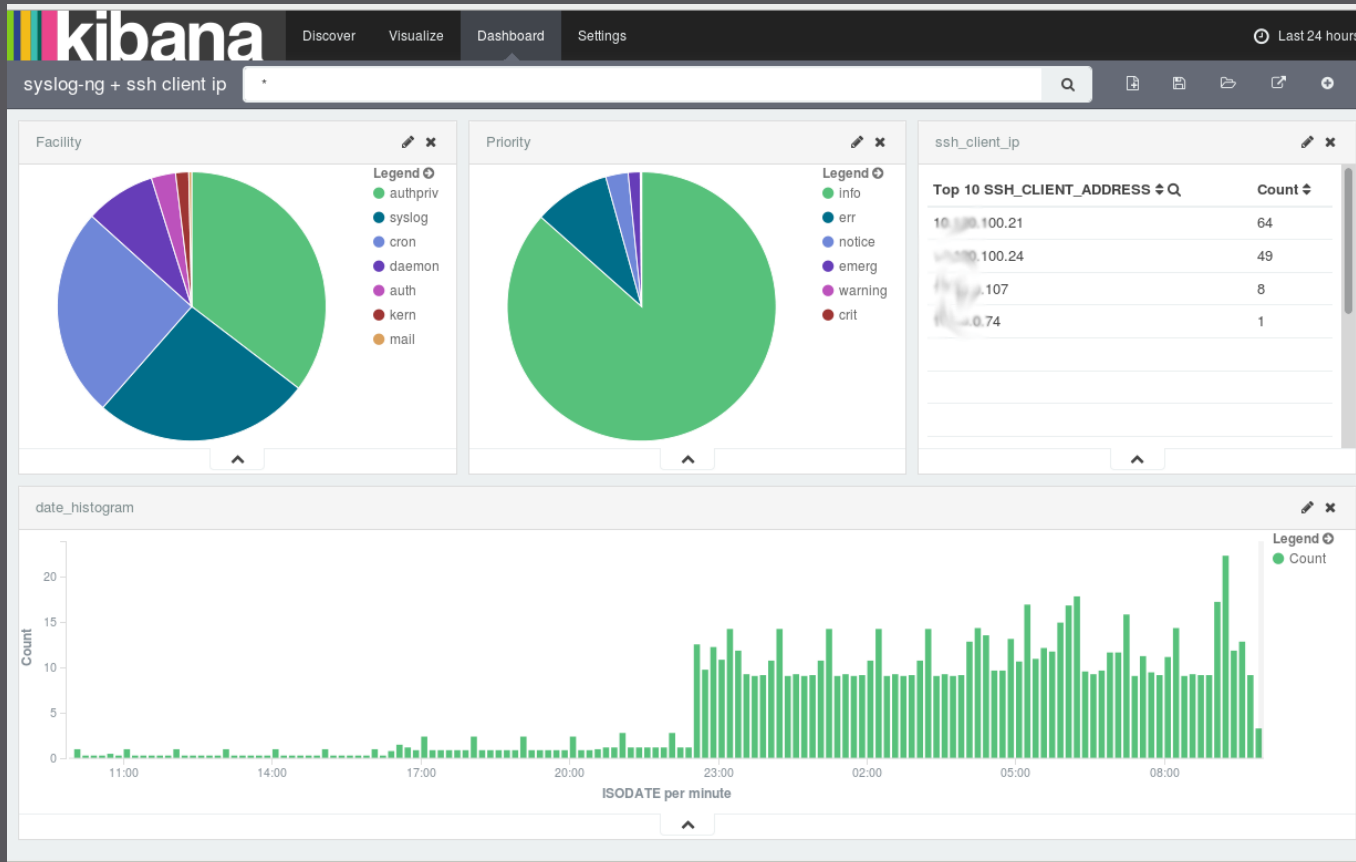
# syslog-ng.conf: filters, parsers

```
filter f_nodebug { level(info..emerg); };  
filter f_messages { level(info..emerg) and  
                    not (facility(mail)  
                        or facility(authpriv)  
                        or facility(cron)); };  
  
parser pattern_db {  
    db-parser(file("/opt/syslog-ng/etc/patterndb.xml") );  
};
```

# syslog-ng.conf: logpath

```
log { source(s_sys); filter(f_messages); destination(d_mesg); };  
log {  
    source(s_net);  
    source(s_sys);  
    filter(f_nodebug);  
    parser(pattern_db);  
    destination(d_es);  
    flags(flow-control);  
};
```

# Patterndb & ElasticSearch & Kibana



# WHAT IS NEW IN SYSLOG-NG



- Disk-based buffering
- Grouping-by(): correlation independent of patterndb
- Parsers written in Python
- Elasticsearch REST API support
- HTTP(s) destination
- Wildcard file source
- Performance improvements
- Many more :-)

# SYSLOG-NG BENEFITS



High-performance  
reliable log collection



Simplified  
architecture  
Single application for both  
syslog and application data



Easier-to-use data  
Parsed and presented in a  
ready-to-use format



Lower load on  
destinations  
Efficient message filtering  
and routing

# JOINING THE COMMUNITY

- syslog-ng: <http://syslog-ng.org/>
- Source on GitHub: <https://github.com/balabit/syslog-ng>
- Mailing list: <https://lists.balabit.hu/pipermail/syslog-ng/>
- Gitter: <https://gitter.im/balabit/syslog-ng>



# QUESTIONS?

---

My blog: <http://czanik.blogs.balabit.com/>

My e-mail: [peter.czanik@balabit.com](mailto:peter.czanik@balabit.com)

Twitter: <https://twitter.com/PCzanik>