Fluentd
Open Source Data Collector

Jan 23, 2016
Scale14x, Pasadena!

Eduardo Silva
eduardo@treasuredata.com
@edsiper
spread the word!

#scale14x  #fluentd
About Me

Eduardo Silva

- Github & Twitter @edsiper
- Personal Blog http://edsiper.linuxchile.cl

Treasure Data

- Open Source Engineer
- Fluentd / Fluent Bit http://github.com/fluent

Projects

- Monkey HTTP Server http://monkey-project.com
- Duda I/O http://duda.io
Logging
Logging Matters

Pros

- Application status
- Debugging
- General information about anomalies: errors
- Troubleshooting / Support
- Local or Remote (network)
Logging Matters

From a business point of view

- Input data → Analytics
- User interaction / behaviors
- Improvements
Assumptions
Logging Matters

Assumptions

- I have enough disk space
- I/O operations will not block
- Log messages are human readable
- My logging mechanism scale
Logging Matters

Assumptions

Basically, yeah.. it should work.
Concerns
Logging Matters

Concerns

- Logs *increase* = data *increase*
- Message format get more *complex*
- Did the Kernel flush the buffers? *(sync(2))*
- Multi-thread application?, *locking*?
- Multiple Applications = Multiple Logs
Logging Matters

Concerns

If Multiple Applications = Multiple logs

Multiple Hosts × Multiple Applications = ???
OK, so:

1. Logging matters
2. It's really beneficial
3. but...
It needs to be done right.
Logging

Common sources & inputs

- **Application Logs**
  - Apache
  - NginX
  - Syslog (-ng)

- **Custom** applications / Languages
  - C, Ruby, Python, PHP, Perl, NodeJS, Java, etc.
In a galaxy not so far away...
How to parse/store multiple data sources?

note: performance matters!
Fluentd is an open source data collector

It lets you unify the data collection for a better use and understanding of data.
before
after
Fluentd

Highlights

- High **Performance**
- Built-in **Reliability**
- **Structured** Logs
- **Plug**gable Architecture
- More than 300 plugins! (input/filtering/output)
Fluentd

Architecture

```
Input  Parser  Filter  Buffer  Output  Formatter
```

“input-ish”  “output-ish”
Fluentd
Internals simplified

```
2012-02-04 01:33:51
myapp.buylog {
  "user": "me",
  "path": "/buyItem",
  "price": 150,
  "referer": "/landing"
}
```
Fluentd
Input plugins

- Plugin

- Input

  - HTTP+JSON (in_http)
  - File tail (in_tail)
  - Syslog (in_syslog)
  - ...

- ✓ Receive logs
- ✓ Or pull logs from data sources
- ✓ in non-blocking manner
Fluentd

Output plugins

Plugin

Output

∇ Write or send event logs

File (out_file)
Amazon S3 (out_s3)
MongoDB (out_mongo)
...

Fluentd

Buffer plugins

- Improve performance
- Provide reliability
- Provide thread-safety

Memory (buf_memory)
File (buf_file)
Fluentd

Buffer plugins

- Improve performance
- Provide reliability
- Provide thread-safety
Key pattern:
- BufferedOutput
  empty string or specified key
- ObjectBufferedOutput
tag
- TimeSlicedOutput
time slice
M x N → M + N

Access logs
- Apache

App logs
- Frontend
- Backend

System logs
- syslogd

Databases

Alerting
- Nagios

Analysis
- MongoDB
- MySQL
- Hadoop

Archiving
- Amazon S3
Fluentd
Simple Forwarding
# Fluentd

## Simple Forwarding: configuration

```
# logs from a file
<source>
type tail
path /var/log/httpd.log
format apache2
tag backend.apache
</source>

# store logs to MongoDB
<match backend.*>
type mongo
database fluent
collection test
</match>

# logs from client libraries
<source>
type forward
port 24224
</source>
```
Fluentd

Less Simple Forwarding
Fluentd
Lambda Architecture
**Fluentd**

---

# logs from a file

```xml
<source>
    <match *.*>
        type tail
        path /var/log/httpd.log
        format apache2
        tag backend.apache
    </match>
</source>
```

# store logs to MongoDB

```xml
<match *.*>
    type copy
    <store>
        type elasticsearch
        logstash_format true
    </store>
</match>
```

# logs from client libraries

```xml
<source>
    type forward
    host 192.x.y.z
    port 50070
    path /path/to/hdfs
</source>
```

```xml
<store>
    type webhdfs
</store>
```

```xml
</match>
```
Who uses Fluentd in production?
We collect 1M events per second!
Internet of Things
Internet of Things

Facts

• IoT will grow to many billions of devices over the next decade.

• Now it's about device to device connectivity.

• Different frameworks and protocols are emerging.

• It needs Logging.
Internet of Things

Alliances

Vendors formed alliances to join forces and develop generic software layers for their products:
Internet of Things

Solutions provided

Alliance                        Framework

OPEN INTERCONNECT CONSORTIUM™ → IoTivity

ALLSEEN ALLIANCE                → AllJoyn™
IoT & Big Data

Analytics

IoT requires a **generic solution** to collect events and data from different sources for further analysis.

Data can come from a specific framework, radio device, sensor or other. How do we collect and **unify** data properly?
Fluent Bit is an open source data collector. It lets you collect data from IoT/Embedded devices and transport it to third-party services.
Fluent Bit

Targets

- Services
- Sensors / Signals / Radios
- Operating System information
- Automotive / Telematics
Fluent Bit

Requirements

IoT and Embedded environment requires special handling, specifically on performance and resource utilization:

- Lightweight
- Written in C Language
- Customizable, pluggable architecture
- Full integration with Fluentd
Fluent Bit

Integration

1. data source
2. fluent-bit data collection
3. fluentd data collection filtering buffering distribution
4. storage processing analytics big data

- MongoDB
- Amazon S3
- Elasticsearch
- Treasure Data
Fluent Bit

Direct Output

1. data source
2. fluent-bit data collection
3. storage processing analytics big data

- IoTivity
- fluent bit
- fluent bit
- MongoDB
- Amazon S3
- Treasure Data
Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.

- Index contains time-based events
- Use event times to create index names [DEPRECATED]

Index name or pattern

Patterns allow you to define dynamic index names using * as a wildcard. Example: logstash-*

- fluentbit

Time-field name

- date

Create
Fluent Bit
Elastic Search: Dashboard
Containers
Docker

Logging driver

- Docker v1.6 released the concept of logging drivers
- Route container output
- Fluentd ?
Add new Logging driver "fluentd" #12876

Merged LK4D4 merged 2 commits into docker:master from tagomoris:logger-driver-fluentd 26 days ago

Tagomoris commented on Apr 29

This patch provides fluentd logging plugin for docker, which is mentioned at #12540.

How to confirm behavior of this patch/new logging driver.
1. Setup dev docker container
2. Build repository with this patch by make BINDDIR=. binary
3. Copy binary and execute it in debug mode docker -d
4. Attach from external environment by docker exec -it kickass_heisenberg /bin/bash
5. Build/execute another docker container from other console, built on Dockerfile and configuration file below:

# sh
Docker v1.8

Fluentd Logging driver!

```
docker run --log-driver=fluentd \
--log-opt fluentd-tag=docker.{{{.Name}}}```

Diagram:
- RAILS (web container)
- PostgreSQL (DB Container)
- RabbitMQ (Worker Container)
- Fluentd Logging Driver
- Amazon S3
- Openstack Swift
### Docker Data Stream

<table>
<thead>
<tr>
<th>tag</th>
<th>time</th>
<th>source</th>
<th>container_id</th>
<th>container_name</th>
<th>log message</th>
</tr>
</thead>
<tbody>
<tr>
<td>docker.3fd...</td>
<td>1441402468</td>
<td>stdout</td>
<td>3fd8678d487e...</td>
<td>/angry_kamam</td>
<td>Hello world!</td>
</tr>
<tr>
<td>3fd8678d4...</td>
<td></td>
<td></td>
<td>message</td>
<td></td>
<td></td>
</tr>
<tr>
<td>docker.3fd...</td>
<td>1441402468</td>
<td>stdout</td>
<td>3fd8678d4...</td>
<td>/angry_kamam</td>
<td></td>
</tr>
<tr>
<td>3fd8678d4...</td>
<td></td>
<td></td>
<td>message</td>
<td></td>
<td></td>
</tr>
<tr>
<td>docker.3fd...</td>
<td>1441402470</td>
<td>stdout</td>
<td>3fd8678d4...</td>
<td>/angry_kamam</td>
<td></td>
</tr>
<tr>
<td>3fd8678d4...</td>
<td></td>
<td></td>
<td>message</td>
<td></td>
<td></td>
</tr>
<tr>
<td>docker.3fd...</td>
<td>1441402471</td>
<td>stdout</td>
<td>3fd8678d4...</td>
<td>/angry_kamam</td>
<td></td>
</tr>
<tr>
<td>3fd8678d4...</td>
<td></td>
<td></td>
<td>message</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Docker

Data Stream
NodeJS

Fluent-Logger (NPM)

```javascript
var logger = require('fluent-logger')

logger.configure('tag', {
    host: 'localhost',
    port: 24224,
    timeout: 3.0
});

logger.emit('label', {record: 'this is a log'});
```
We Love Data!

- http://fluentd.org
- http://fluentbit.io
- https://docs.docker.com/reference/logging/fluentd/
- http://github.com/fluent/fluentd

Thank you!