Logging

Logging

- Process of recording an events that happened in application in a specific point in time
- Gives us understanding of what is software behavior
- Often has low priority in development
- Logging != Monitoring
- Process of repeatedly collecting data metrics of application in a specific point in time

Source

- Application
  - exceptions, info
- Server
  - web servers access logs, database long query logs
- System
  - boot log, kernel logs, cron job logs

Destination

- Files
  - often a default, can become big in size, problem to scale
- API
  - easy to configure, external dependency, expensive
- Database
  - not common, slow to write in db, increase db size
- Stdout
  - responsibility of the environment, docker as an example

Format

- Plain-text
  - free-form text, most common, formatted
- Structured
  - more advanced, JSON, easy to search
- Binary
  - systemd, mysql binlogs, not human readable

Decentralized vs Centralized

- Decentralized
    - each server has logs, hard to access when scaled
- Centralized
    - logs in one place, easy to access

Cloud-based vs Self-hosted

- Cloud-based
  - centralized server on a cloud as SaaS, easy to setup and maintain
  - loggly, papertrail, splunk
- Self-hosted
  - build to fit our needs, managed by us
  - harder to setup, lower monthly costs, easier to scale

Log levels

- Info
- Warn
- Error
- Fatal
- Debug
- Trace
- 1-10

Log rotation

- Important when logging into files
- Process of compressing and archiving the log files
- Rotate on file size or time interval
- Archived files are deleted after a period of time
- Without it, big files can crash the server

Log alerts

- Sending alerts based on log entries

Components

- Log collectors
  - collect logs from different sources, low affect on server performance
  - scribe, flume, logstash, fluend, graylog2
- Storage / Search engine
  - store logs and perform very fast searches on large data
  - elasticsearch, solr
- Visualization dashboards
  - view, search and visualize logs
  - kibana, graylog, grafana

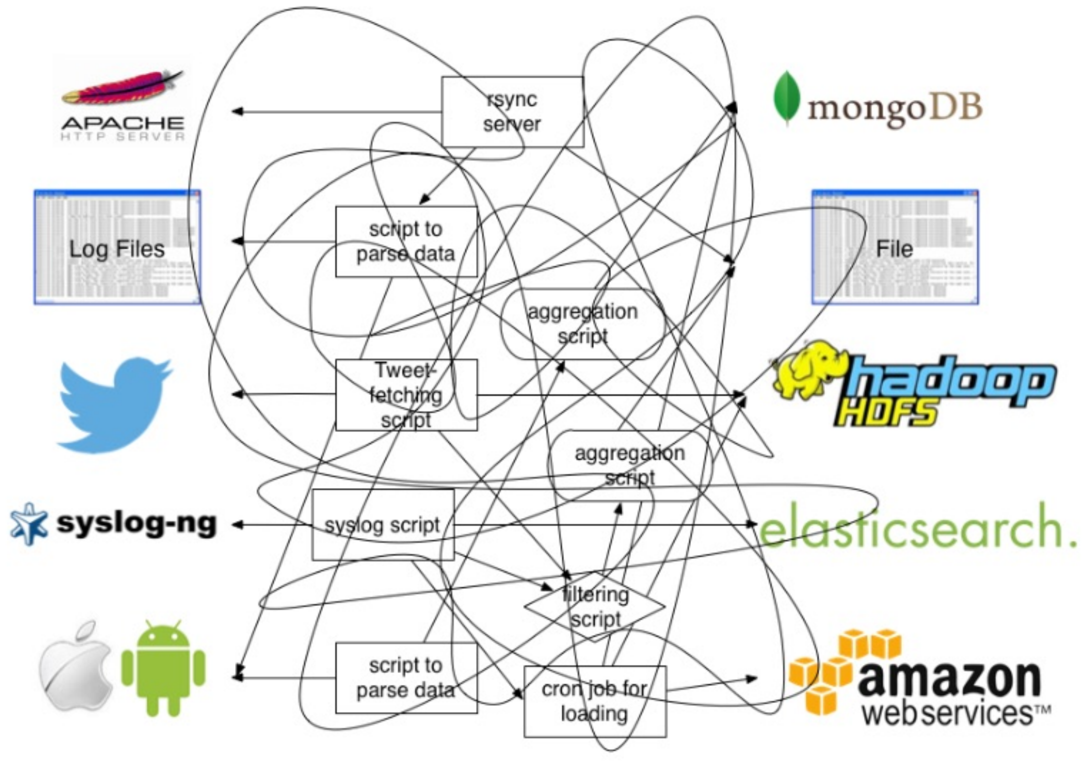
Stack

- Graylog
  - covers all components, can use other log collectors
- ELK
  - elasticsearch, logstash, kibana
  - battle-tested, standard in centralized logging
- EFK
  - elasticsearch, fluentd, kibana

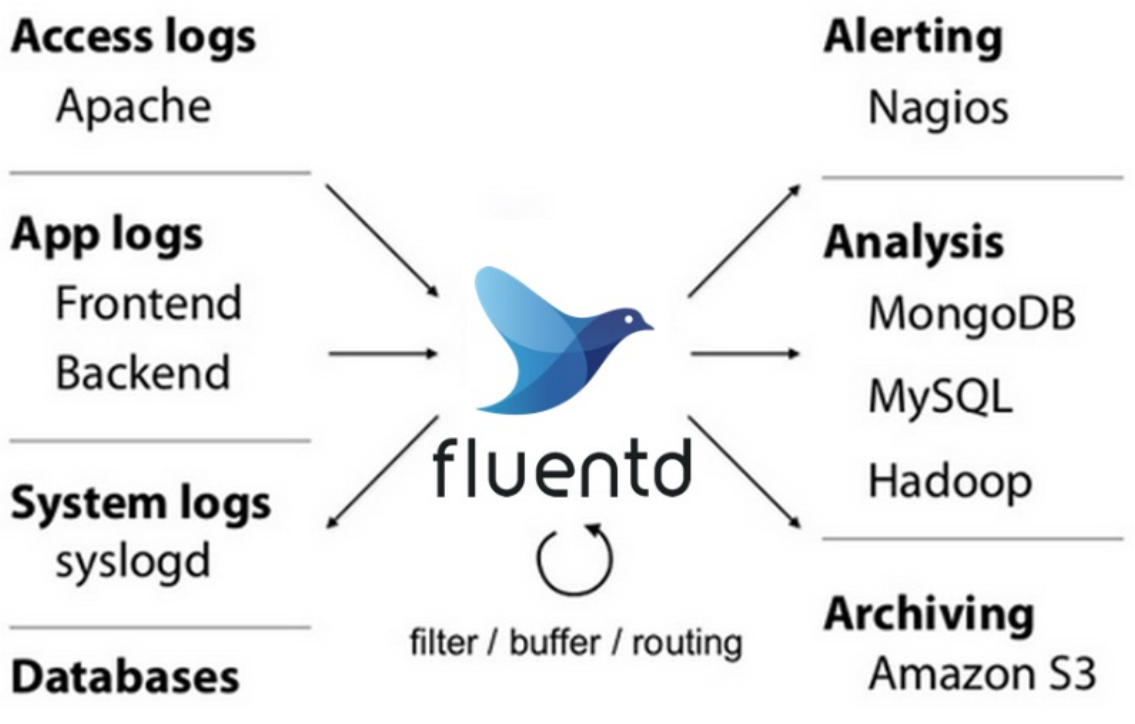
Logstash vs Fluend

- Logstash
  - used to consume more memory
  - event routing based on algorithmic statements
  - centralized plugin repository
  - recommended for traditional systems with VMs
- Fluentd
  - event routing based on tags
  - decentralized plugin repository
  - built by cloud native computing foundation

Fluend



Before Fluentd



After Fluentd

Fluend configuration

- @include

```
@include /path/to/config.conf
```

- Source

```
<source>
    @type http
    port 9880
</source>
```

- Match

```
<match myapp.tag>
    @type file
    path /var/log/fluent/myapp.tag.log
</match>
```

Fluend configuration

- Filter

```
<filter myapp.access>
    @type record_transformer
    <record>
        host_param "#{Socket.gethostname}"
    </record>
</filter>
```

- System

```
<system>
    log_level error
</system>
```

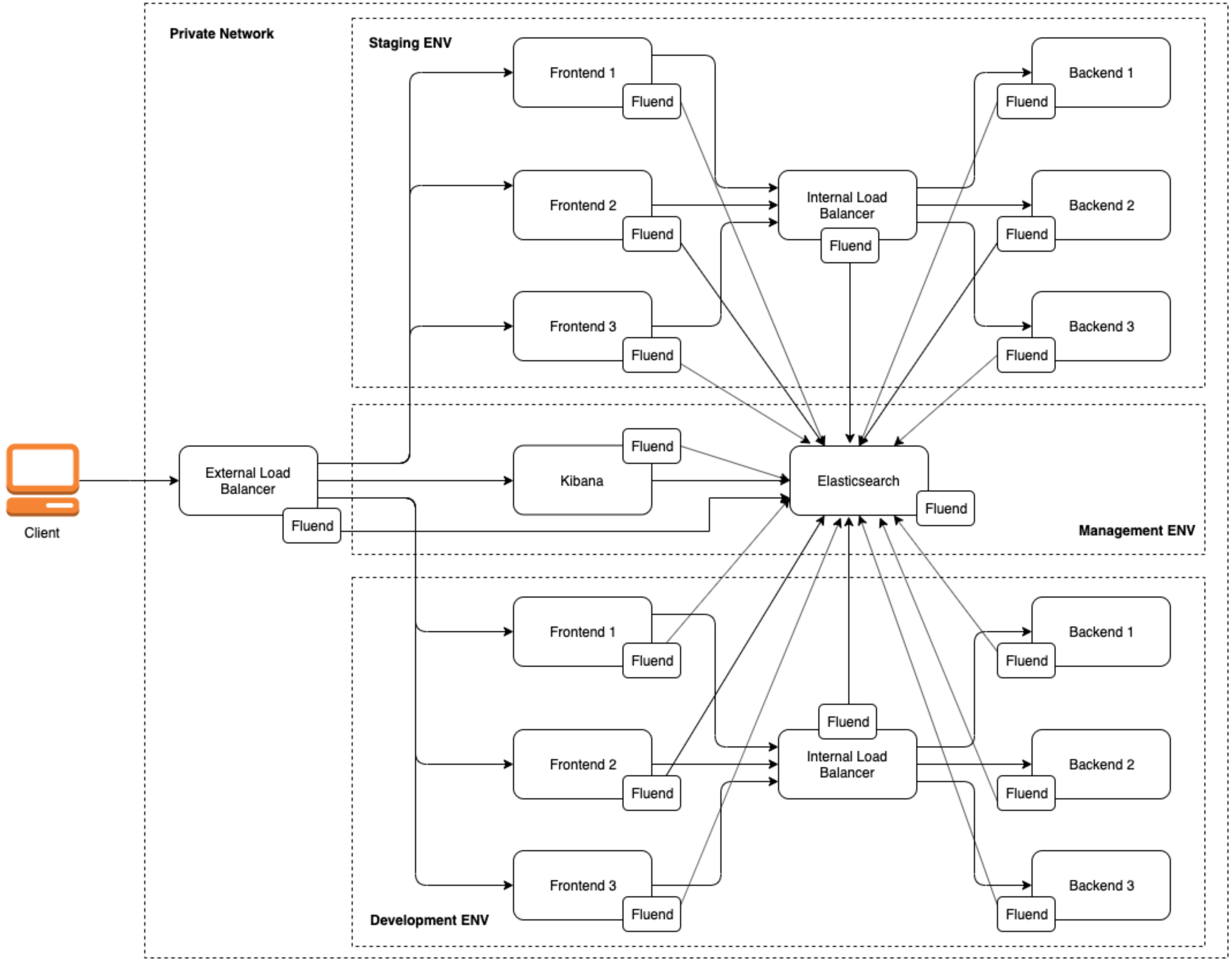
Fluend event

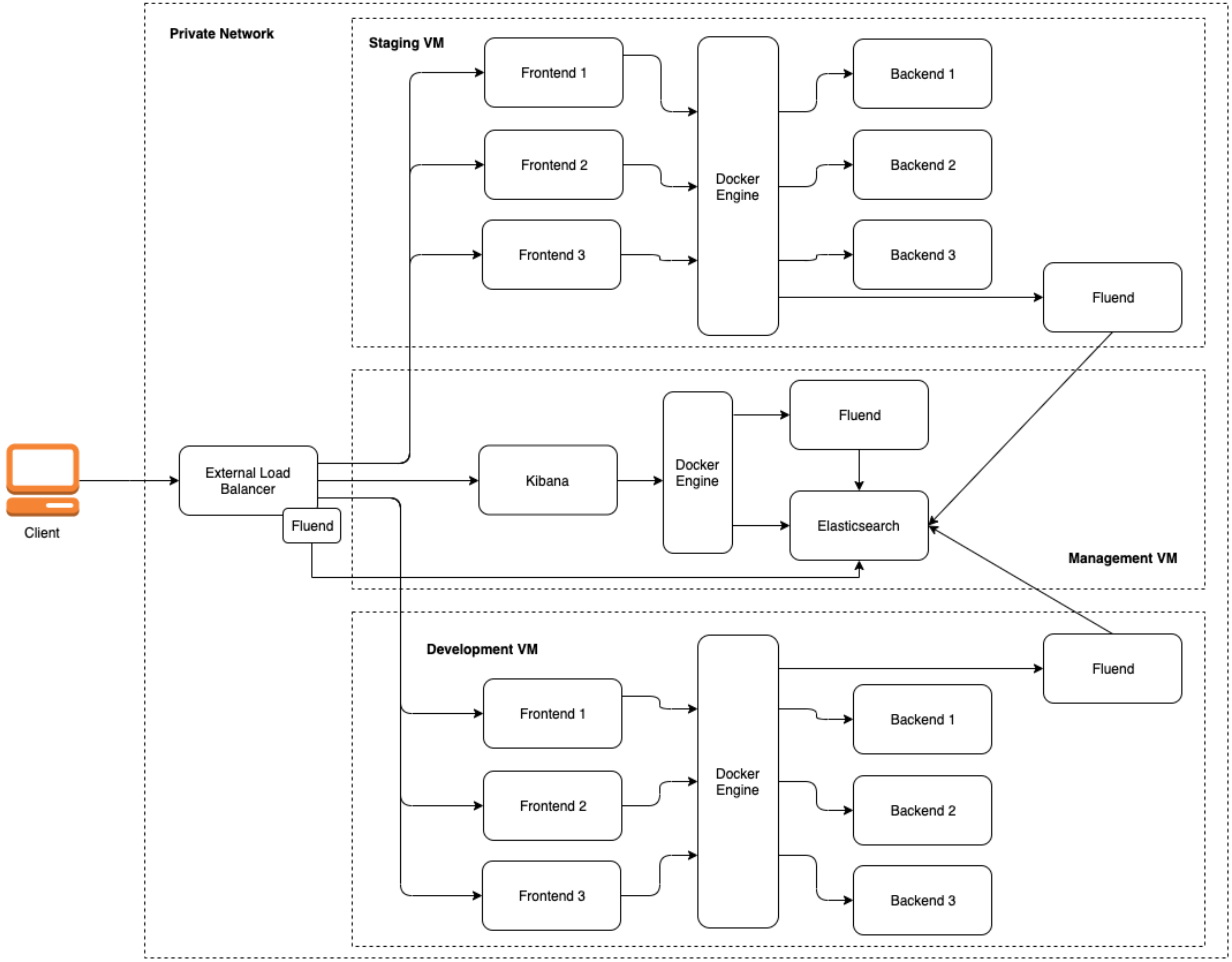
- Apache access log record

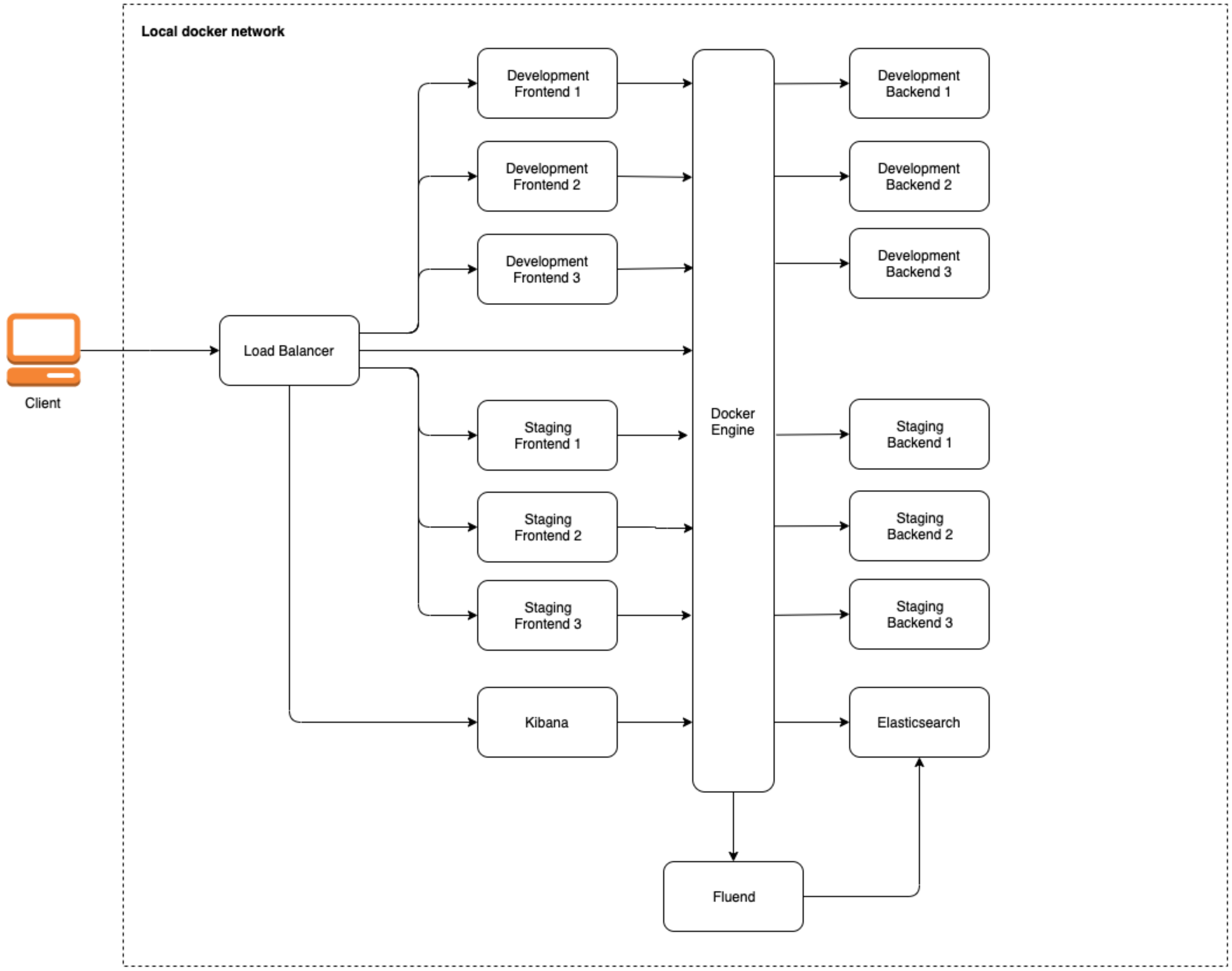
```
192.168.0.1 - - [28/Feb/2013:12:00:00 +0900] "GET / HTTP/1.1" 200 777
```

- Fluend event

```
tag: apache.access # set by configuration
time: 1362020400 # 28/Feb/2013:12:00:00 +0900
record: {"user":”-","method":"GET","code":200,"size":777,"host":"192.168.0.1","path":"/“}
```


Private Network
Staging ENV
Frontend 1
Fluend
Frontend 2
Fluend
Frontend 3
Fluend
Internal Load Balancer
Fluend
Backend 1
Fluend
Backend 2
Fluend
Backend 3
Fluend
Client
External Load Balancer
Fluend
Kibana
Fluend
Elasticsearch
Fluend
Management ENV
Frontend 1
Fluend
Frontend 2
Fluend
Frontend 3
Fluend
Fluend
Internal Load Balancer
Backend 1
Fluend
Backend 2
Fluend
Backend 3
Fluend
Development ENV


Private Network
Staging VM
Frontend 1
Frontend 2
Frontend 3
Docker Engine
Backend 1
Backend 2
Backend 3
Fluend
Client
External Load Balancer
Fluend
Kibana
Docker Engine
Fluend
Elasticsearch
Management VM
Development VM
Fluend
Frontend 1
Frontend 2
Frontend 3
Docker Engine
Backend 1
Backend 2
Backend 3


Local docker network
Client
Load Balancer
Development Frontend 1
Development Frontend 2
Development Frontend 3
Staging Frontend 1
Staging Frontend 2
Staging Frontend 3
Kibana
Docker Engine
Development Backend 1
Development Backend 2
Development Backend 3
Staging Backend 1
Staging Backend 2
Staging Backend 3
Elasticsearch
Fluend

Demo

https://github.com/aleksmark/centralized-demo

Resources

- https://hackernoon.com/part-1-building-a-centralized-logging-application-5a537033da0a
- https://www.loomsystems.com/blog/single-post/2017/01/30/a-comparison-of-fluentd-vs-logstash-log-collector
- https://codefarm.me/2018/06/29/elasticsearch-fluentd-kibana-docker-compose/
- https://docs.fluentd.org/quickstart/life-of-a-fluentd-event
- https://docs.docker.com
- https://docs.fluentd.org