KubeCon | CloudNativeCon

North America 2019

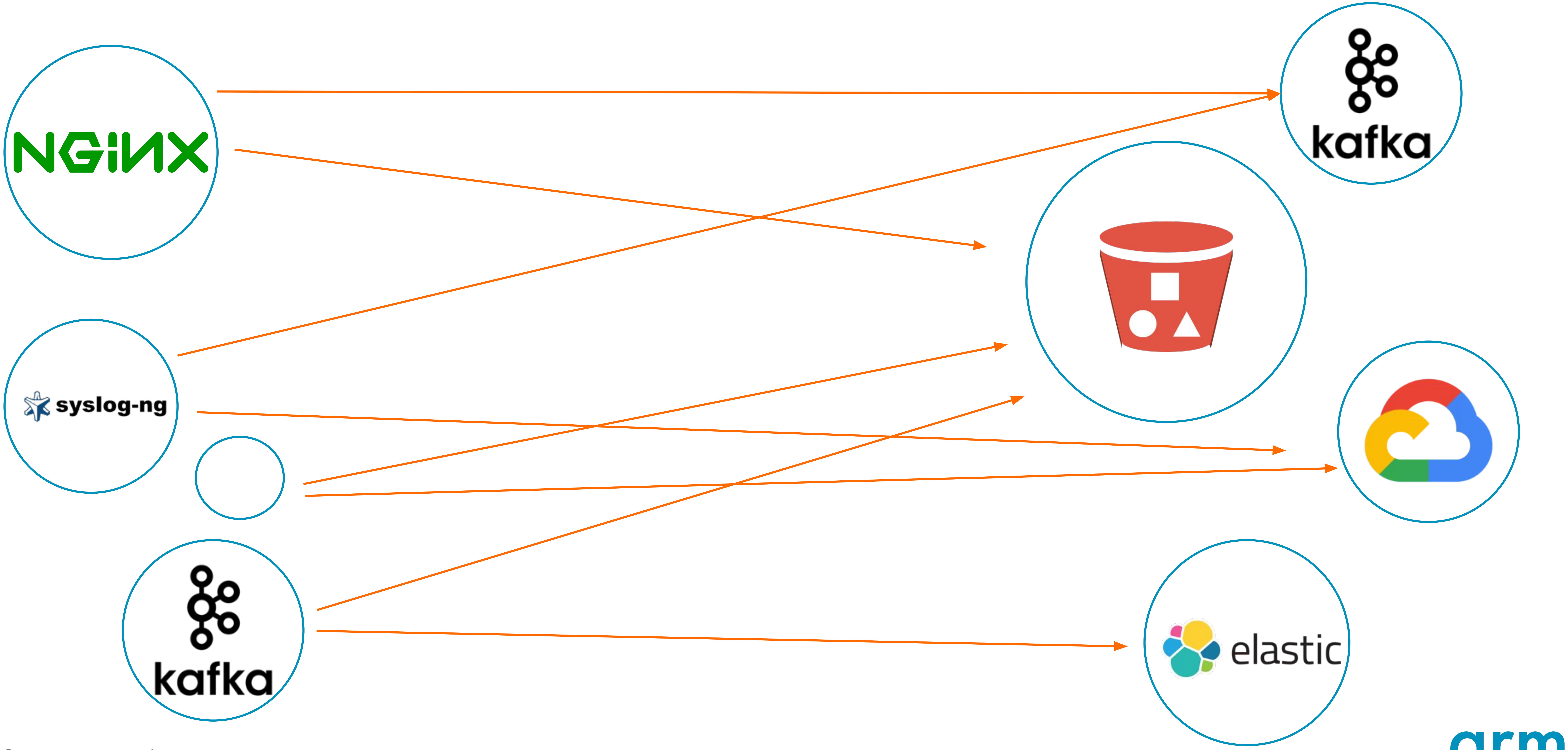
Fluentd Project Intro

Yuta Iwama

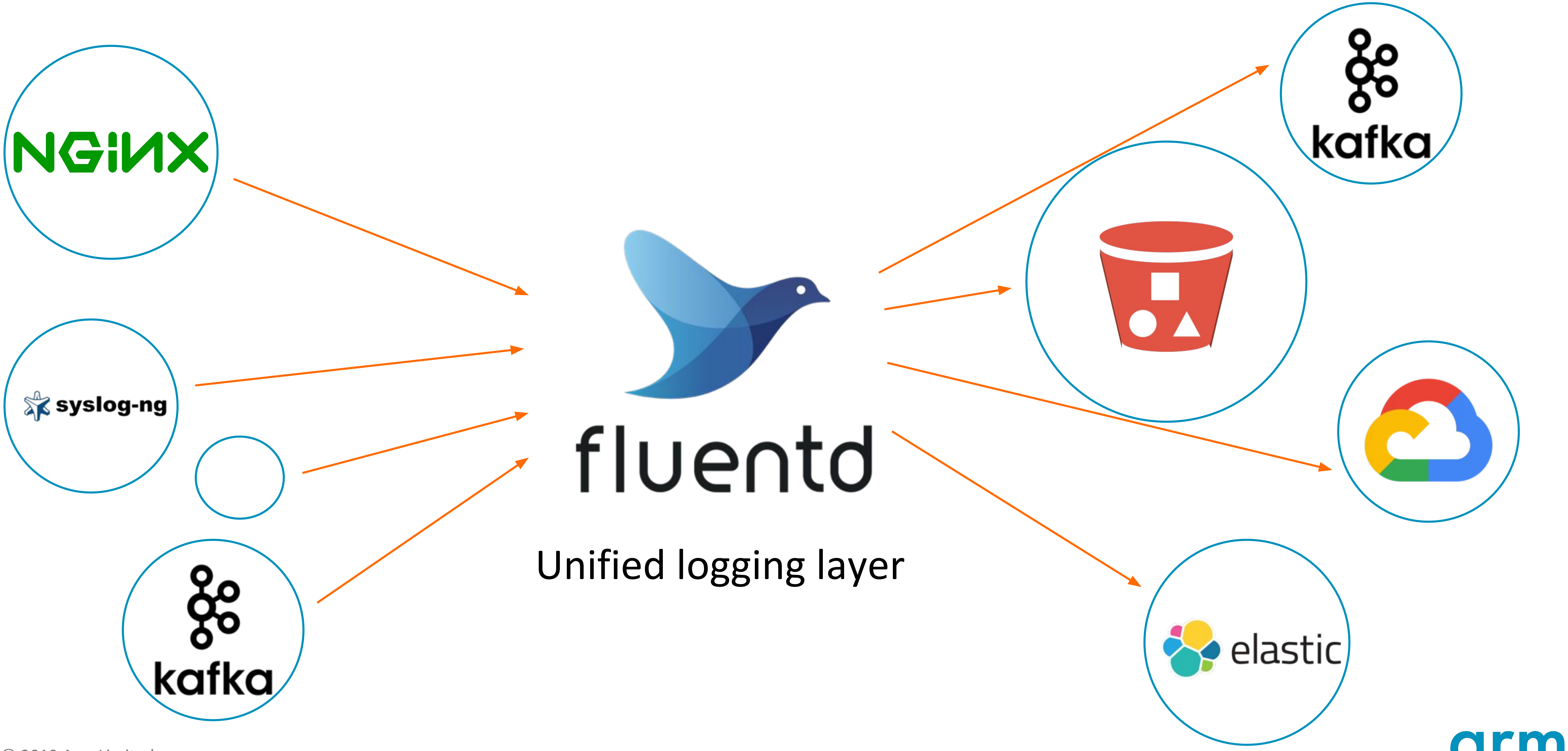
What is Fluentd

- Streaming data collector for unified logging
- Pluggable architecture
  - 900+ community-contributed plugins
- Several setup ways
  - RubyGems, Docker, packages (ubuntu, centos, and more)
  - https://docs.fluentd.org/installation
- 6th project to graduate from CNCF
- Adopted as logging driver at GCP
  - https://cloud.google.com/logging/docs/agent/
- Latest version: v1.8.0rc3

Unified Logging Layer

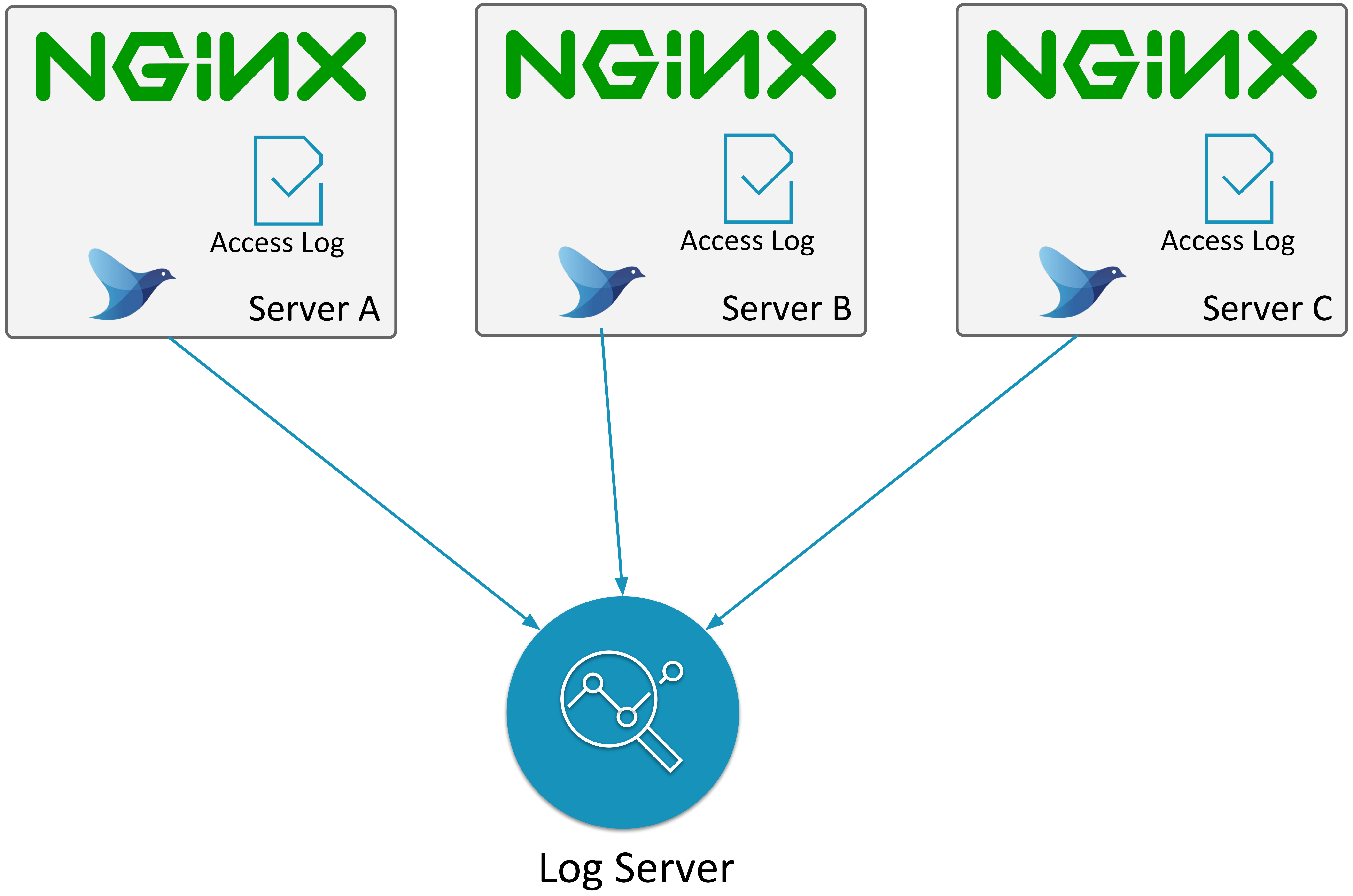


Unified Logging Layer



Streaming Way with Fluentd

- Nginx outputs access log
- Fluentd monitors access log
- Send logs with low latency
- Fluentd can format log data
- Easy to analyze log on log server without any operation
  - Drop/Add field
  - Encrypt data, etc.



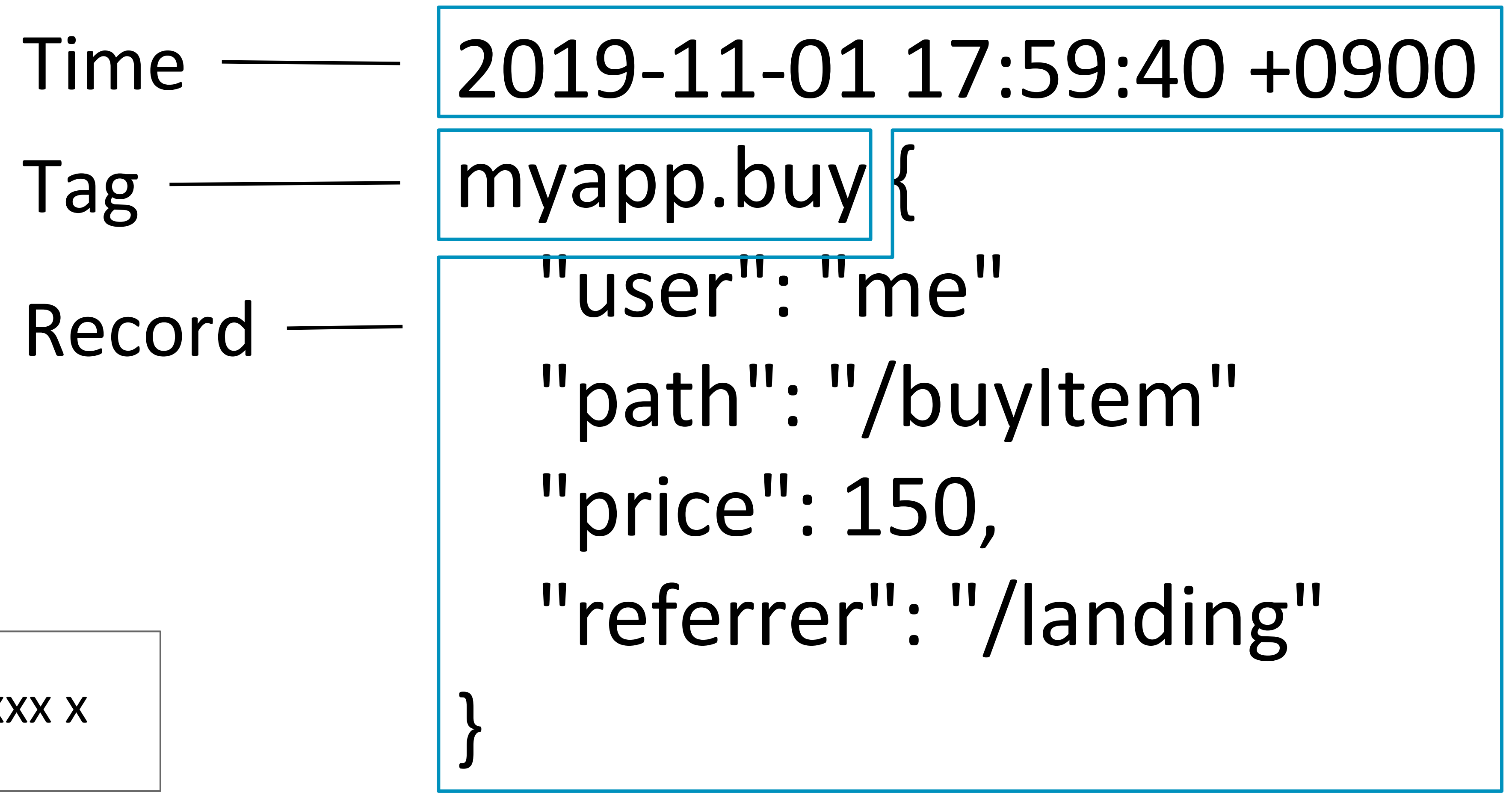
Architecture

Design

- Extendable and reliable
- fluentd is composed of core part and plugin part
- Core part
  - Core system manages each of the plugins
  - Multi process and Multi threads model
    - Supervisor-Workers style
    - Fast write with multiple threads
  - Error handling
  - Provide plugin helpers
- Plugin part
  - Handle actual data
  - explain it more detail later

Event Structure

- Fluentd routes events by tag
- Event consist of three parts
  - Tag: Used for event routing, identify data source
  - Time: Event occured time(nano-second precision)
  - Record : Actual data (JSON object)



Time — 2019-11-01 17:59:40 +0900

Tag — myapp.buy

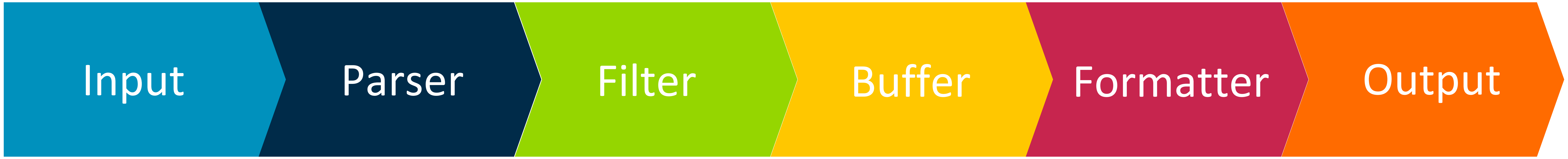
Record —

```
{
   "user": "me"
   "path": "/buyItem"
   "price": 150,
   "referrer": "/landing"
}
```

Original Log

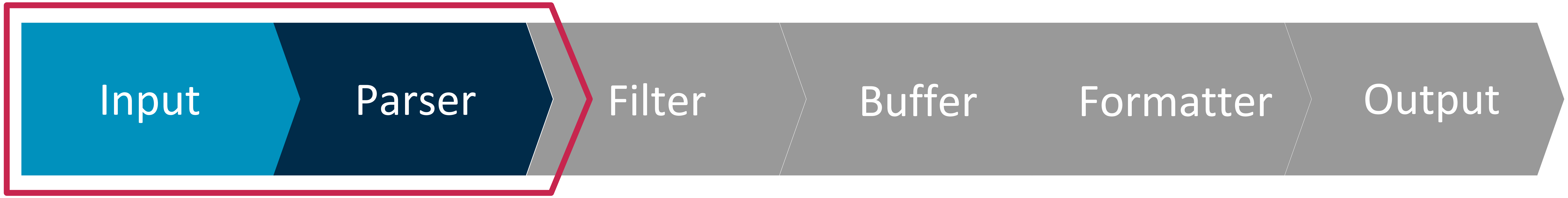
```
[01/Nov/2019:17:59:40 +0900] "POST /buyitem&referrer=/landing&path=150&user=me HTTP/1.1" 200 xxx x xxx x
```

Simplified Architecture



- Each of the plugins pass an event to the next plugin
  - Input plugins: Read/Receive data
  - Parser Plugins: Parse data
  - Filter Plugins: Filter/Enrich data
  - Buffer Plugins: Buffering data
  - Formatter Plugins: Format data
  - Output Plugins: Write/Send data

Input and Parser Plugins



Input

Parser

Filter

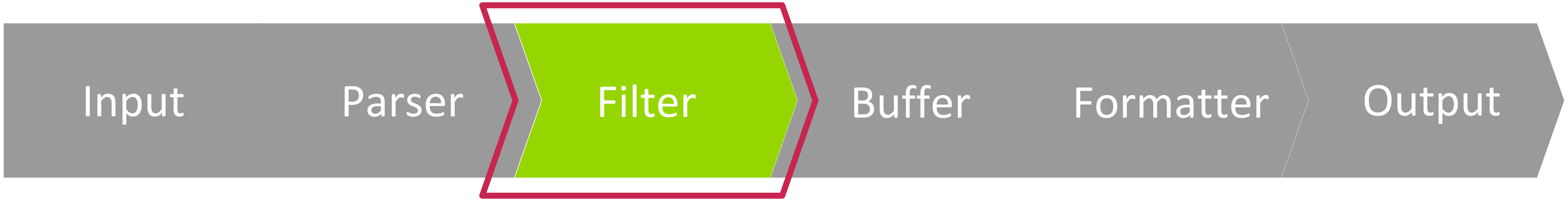
Buffer

Formatter

Output

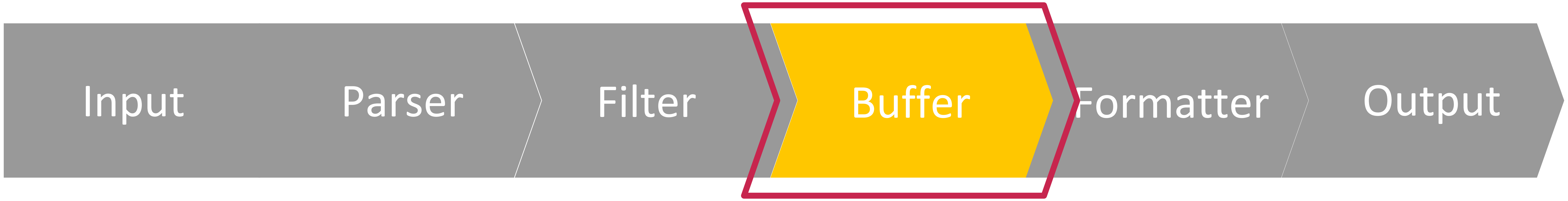
- Input plugins receive/read logs from data sources
- Emit logs to data pipeline
- Parser plugins parse incoming logs for structured log
- e.g.
  - syslog (in_syslog)
  - HTTP + JSON (in_http + json_parser)
  - local file of nginx’s access log (in_tail + nginx_parser)

Filter Plugins



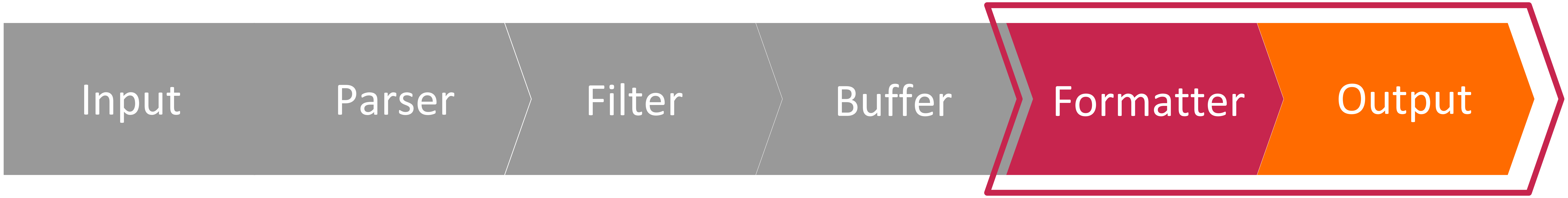
- Transform logs
- Filter out unnecessary logs
- Enrich logs
- e.g.
  - Add hostname to record (filter_record_transformer)
  - Regex like filtering like grep (filter_grep)

Buffer Plugins



- Improve performance
- Provide reliability
- Provide thread-safety
- e.g.
  - Persistent events with file (buf_file)
  - Memory (buf_mem)

Formatter and output plugins

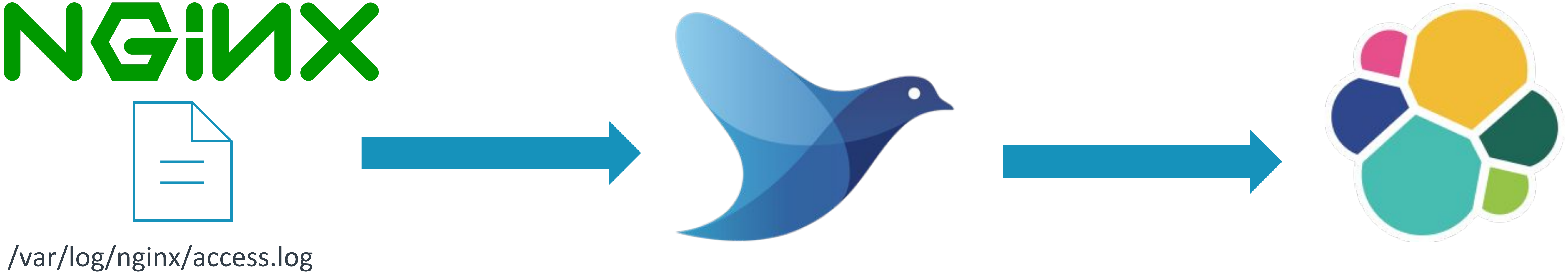


- Format data like JSON, CSV or other formats
- Write/Send event logs
- Provide two ways of write/send (sync and async)
- e.g.
  - Local file which is json formatted (out_file + format_json)
  - Send logs to Amazon S3 (out_s3)
  - Send logs to other fluentd (out_forward)

Use Cases

arm
TREASURE DATA

Simple Forwarding

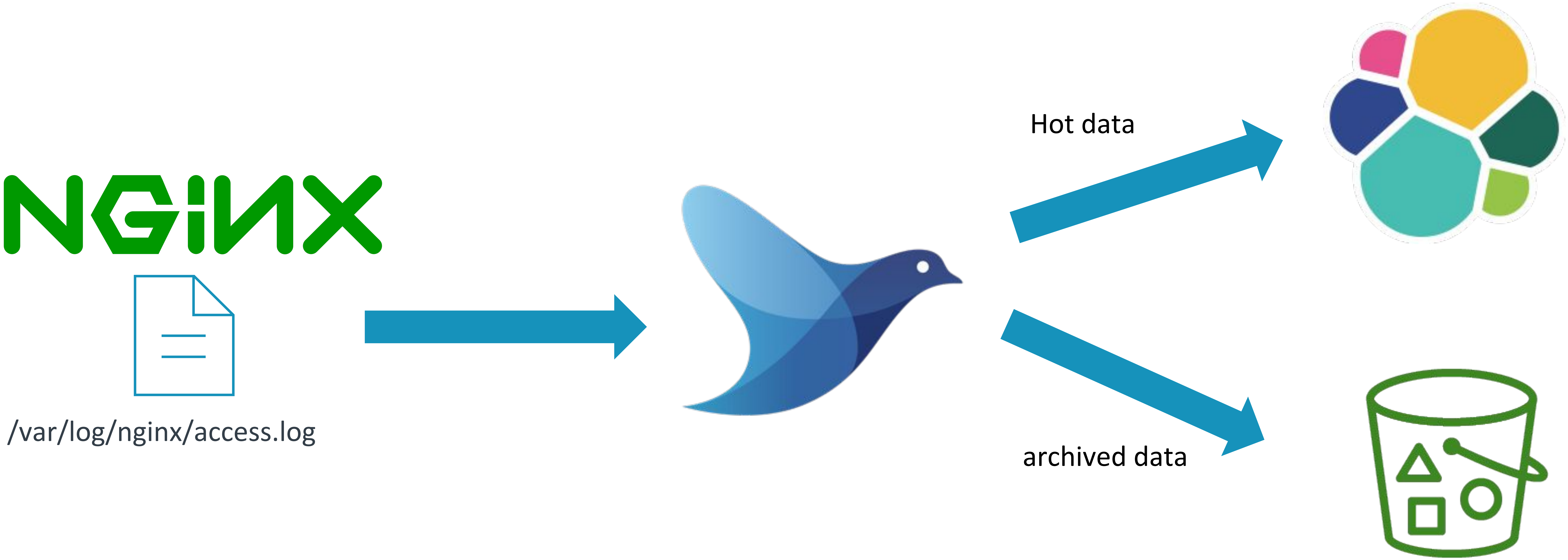


Simple Forwarding

```
# Logs from a file
<source>
  @type tail
  path /var/log/nginx/access.log
  pos_file /var/log/fluentd/tail_pos
  <parse>
    @type nginx
  </parse>
  tag app.access
</source>
```

```
# Store logs to ES
<match app.**>
  @type elasticsearch
  logstash_format true
</match>
```

Multiple Destinations

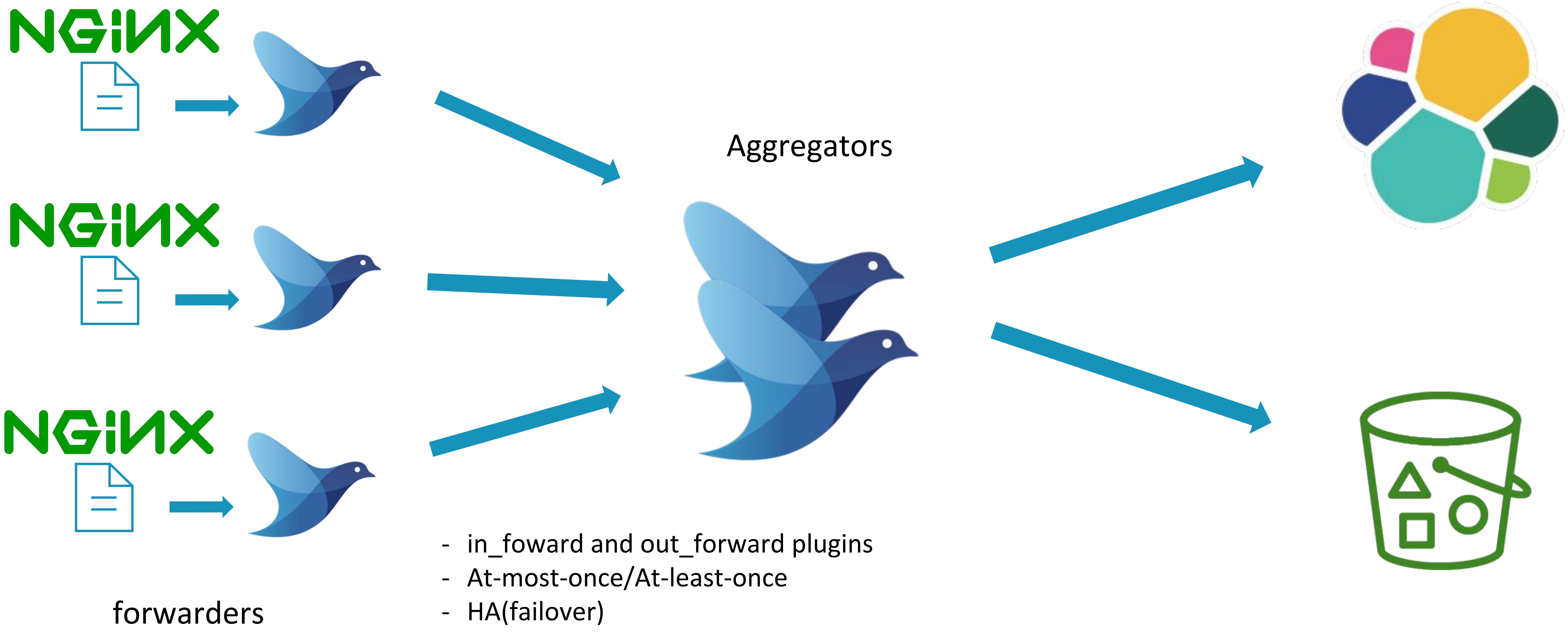


Multiple Destinations

```
# Logs from a file
<source>
  @type tail
  path /var/log/nginx/access.log
  pos_file /var/log/fluentd/tail_pos
  <parse>
    @type nginx
  </parse>
  tag app.access
</source>
```

```
# Store logs to ES and S3
<match app.**>
  @type copy
  <store>
    @type elasticsearch
    logstash_format true
  </store>
  <store>
    @type s3
    s3_bucket us_east_1_log
    path logs/${tag}/%Y/%m/%d/
    <buffer tag,time>
      @type file
    </buffer>
   <store>
</match>
```

Multi-tier Forwarding



- in_foward and out_forward plugins
- At-most-once/At-least-once
- HA(failover)
- Load-balancing
- Keepalive

Docker and Kubernetes

arm TREASURE DATA

Fluentd with Docker and kubernetes

- Community Base: https://hub.docker.com/r/fluent/fluentd
  - Multiple versions of Fluentd
  - Alpine / Debian images
- Docker official: https://hub.docker.com/_/fluentd
  - Multiple platform
    - amd64, arm32v5, arm32v6, arm32v7, arm64v8, i386, ppc64le, s390x
- fluentd-kubernetes-daemonset
  - https://github.com/fluent/fluentd-kubernetes-daemonset
  - Various built-in destinations ES, kafka, graylog, etc…
- Helm chart
  - https://github.com/helm/charts/tree/master/stable/fluentd

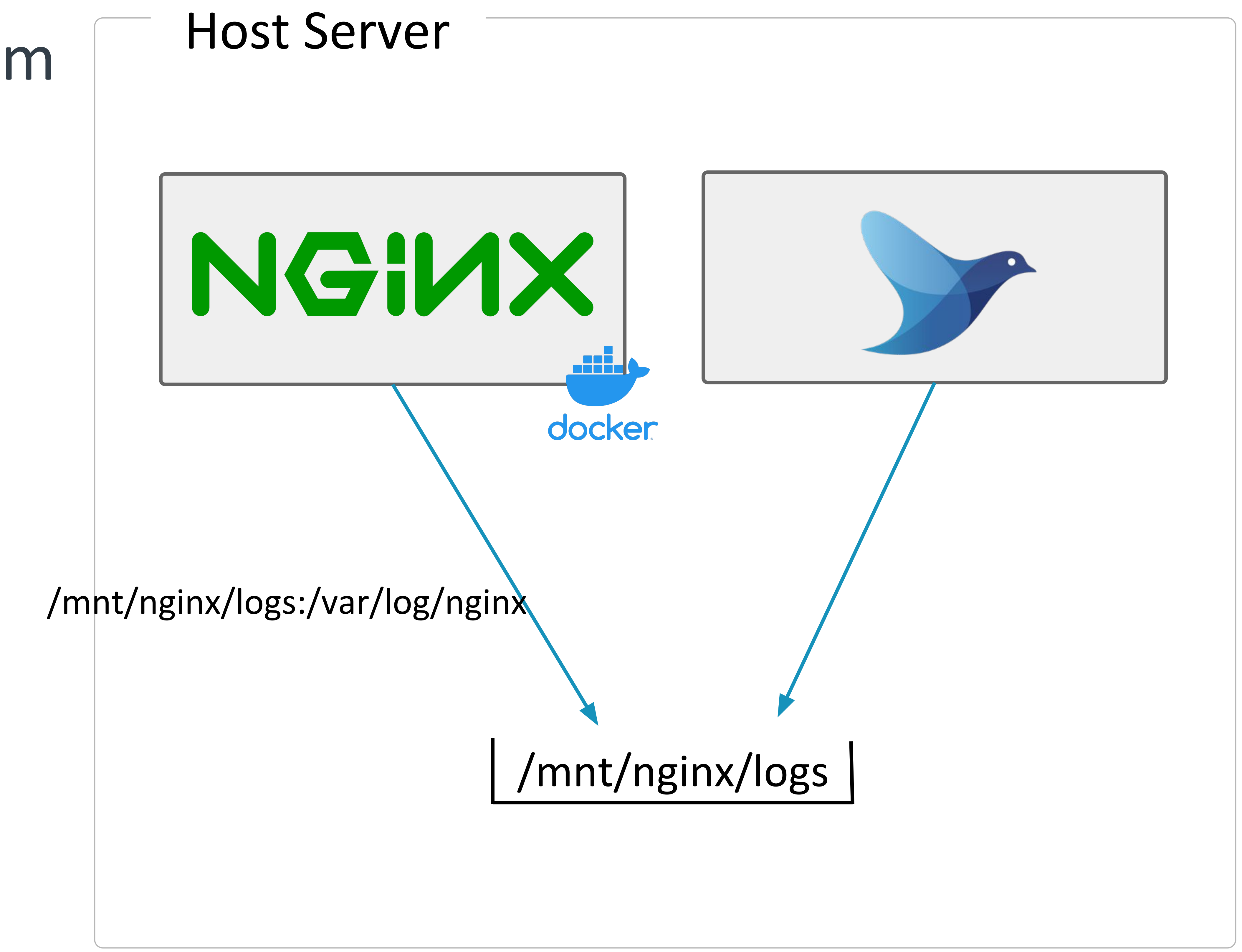
Fluentd Container Pattern

- Mounted volume
  - App container and Fluentd (container) share the host’s file system
- Sending data via a library
  - Need implementation in each language
- Logging driver
  - fluentd can be used as docker logging driver

Mounted Volumes

- Fluentd and nginx container shared host file system
- Fluentd watchs log files with in_tail plugin

```
<source>
  @type tail
  path /mnt/nginx/logs/access.log
  pos_file /var/log/fluentd/access.log.pos
  <parse>
    @type nginx
  </parse>
  tag app.access
</source>
```

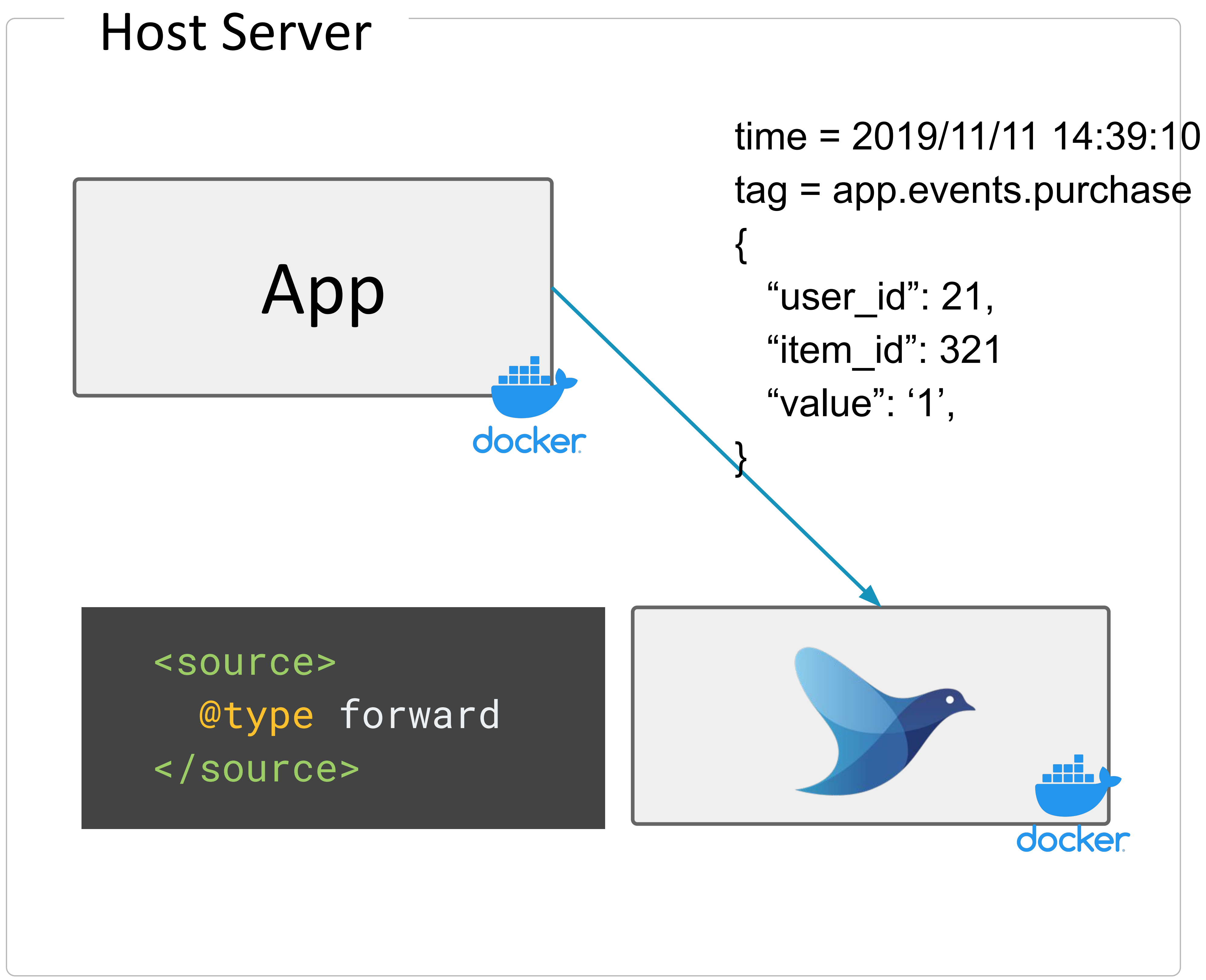


Sending Data via fluent-logger

- Use fluent logger library
- Need implementation in each language

```
from fluent import sender
from fluent import event

sender.setup('app.events', host='localhost')
event.Event('purchase', {
  'user_id': 21, 'item_id': 321, 'value': '1'
})
```



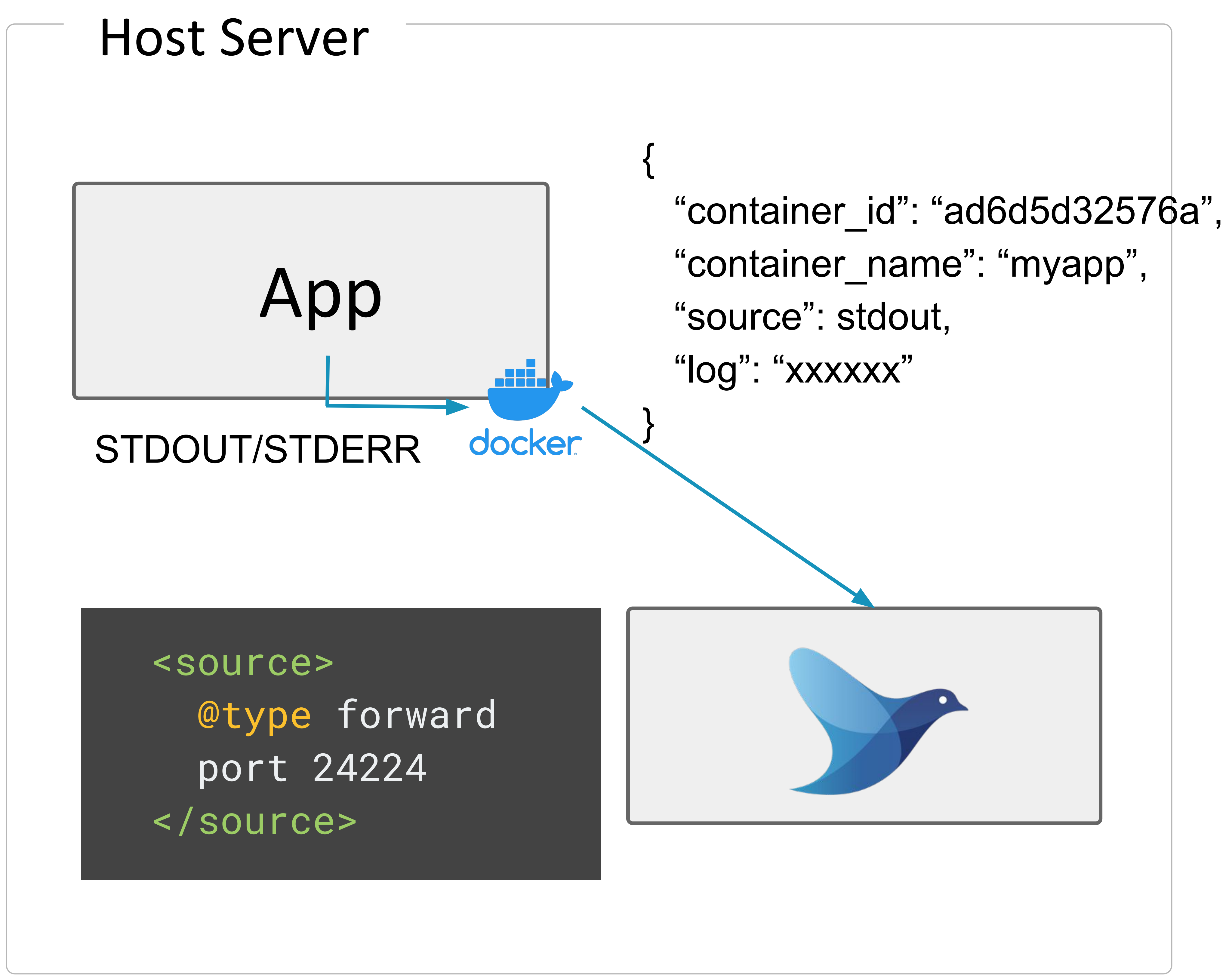
time = 2019/11/11 14:39:10
tag = app.events.purchase
{
  “user_id”: 21,
  “item_id”: 321
  “value”: ‘1’,
}

```
<source>
  @type forward
</source>
```

Docker Logging Driver

- Docker provides logging driver mechanism
  - Sending logs to external host or another logging backends
- Fluentd runs as logging driver of Docker by default

```
docker run \
  --log-driver=fluentd \
  --log-opt \
  fluentd-address=localhost:24224
```



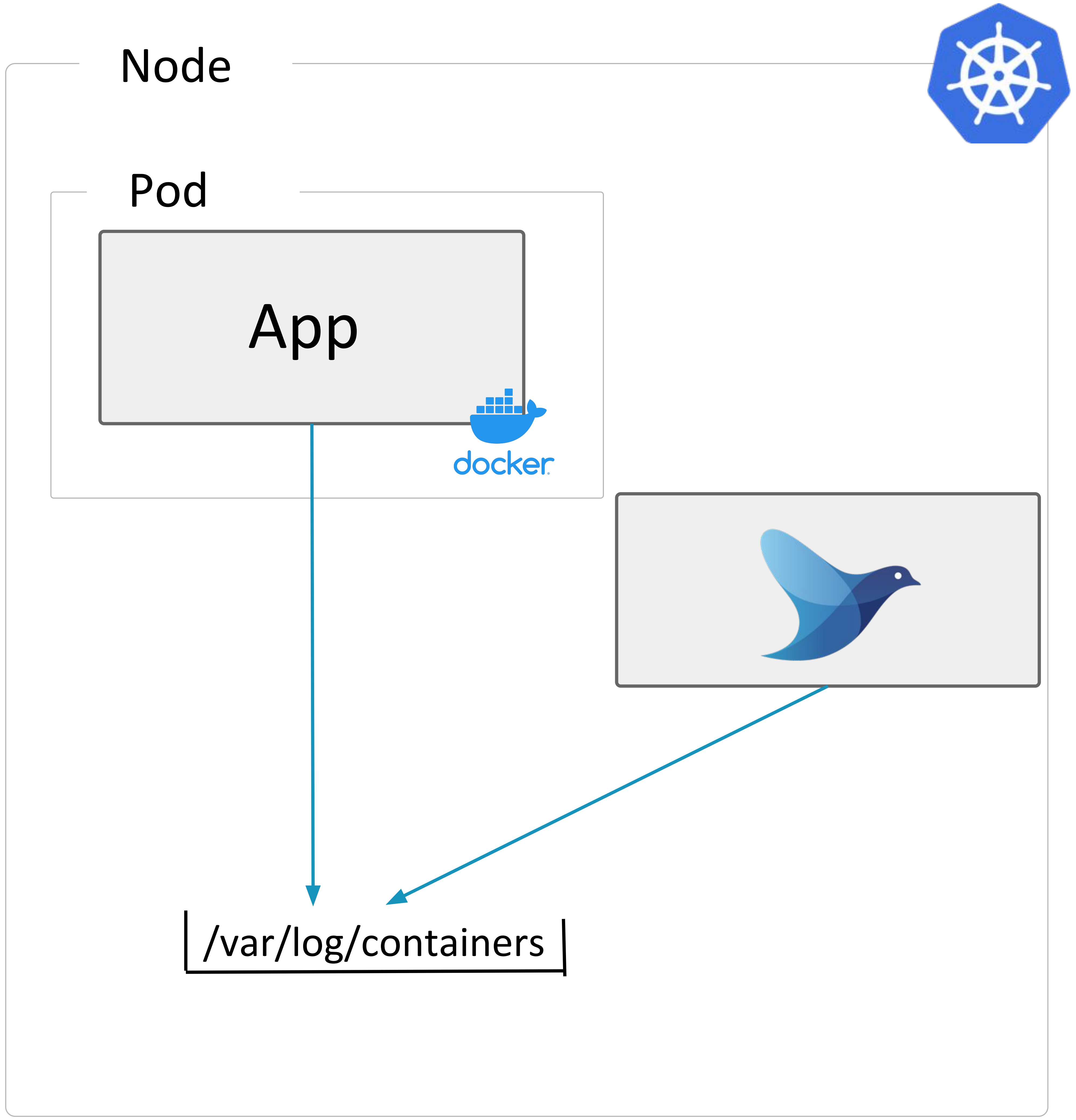
```
{
  "container_id": "ad6d5d32576a",
  "container_name": "myapp",
  "source": stdout,
  "log": "xxxxxx"
}
```

```
<source>
  @type forward
  port 24224
</source>
```

Kubernetes Daemonset

- Similar to shared volume
- Run Fluentd as daemonset

```
<source>
  @type tail
  path /var/log/containers/*.log
  pos_file /var/log/fluentd/access.log.pos
  tag kubernetes
</source>
```



Wrapping up

arm TREASURE DATA

Announcement and Update

- We plan to drop a support for old fluentd and ruby at the end of 2019
  - Ruby 2.1, 2.2, 2.3
  - Fluentd 0.12
  - td-agent 2.3
  - https://www.fluentd.org/blog/drop-schedule-announcement-in-2019
- New features (v1.6.0 - v1.8.0rc3)
  - Service discovery helper
  - File single buffer plugins
  - MonitorAgent and Prometheus plugins expose more metric
  - HTTP server helper

Summary

- Fluentd is designed for streaming log collection
- There are vast number of community contributed plugins
- Fluentd runs on a lot of environment (OS, Docker)
- Docker logging driver support Fluentd
- We plan to drop the support for old fluentd and ruby at the end of 2019