Centralized PHP Logging Patterns

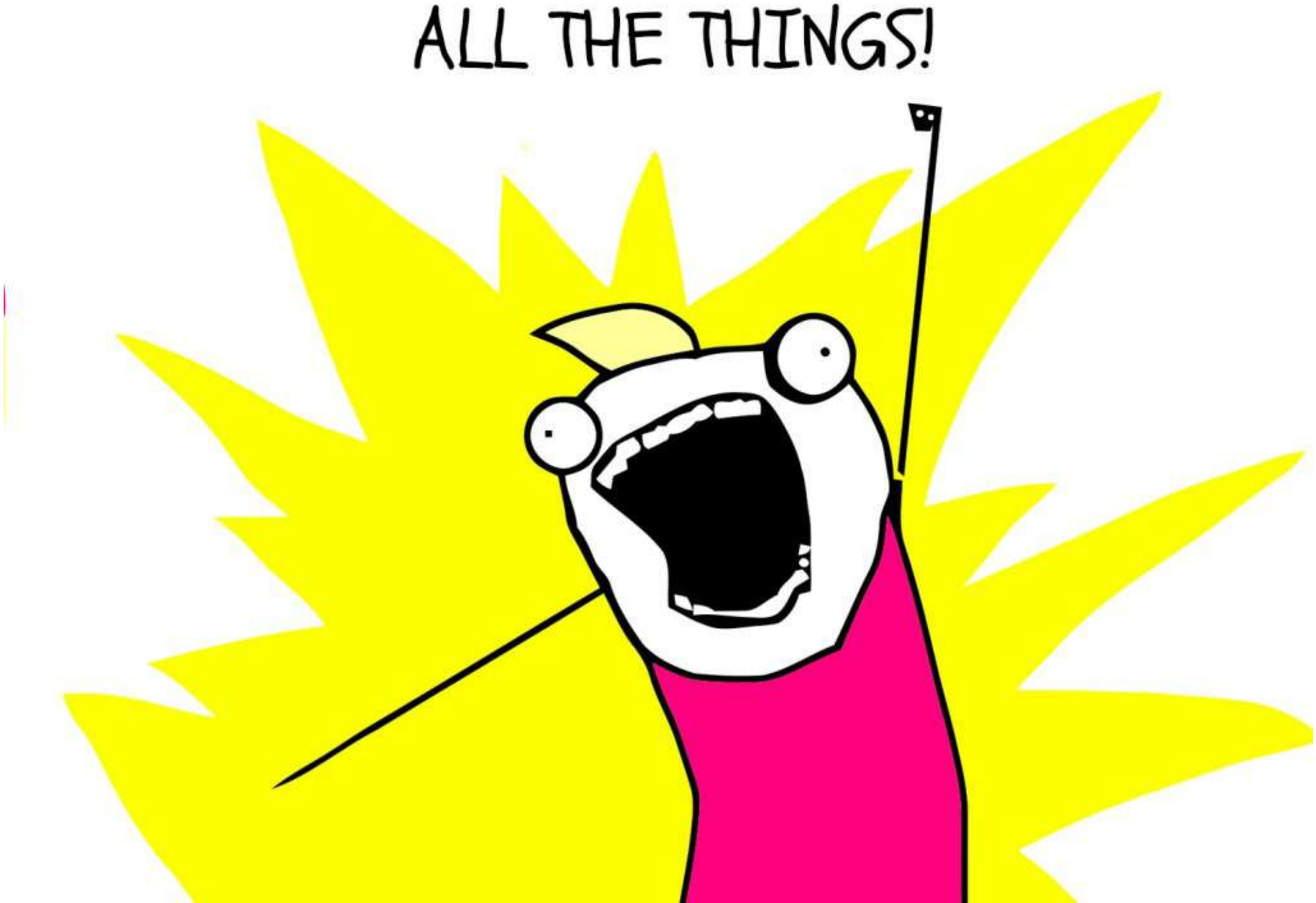
Philipp Krenn @xeraa

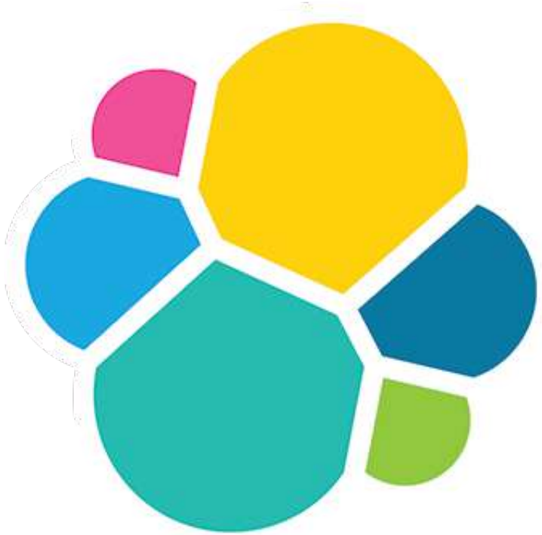
```php
<?php

use Monolog\Logger;
use Monolog\Handler\StreamHandler;

$log = new Logger('my-logger');
$log->pushHandler(new StreamHandler('path/to/my.log', Logger::WARNING));

$log->error('Something went wrong...');
```


THIS IS FINE.


me looking
for the bug
7.2 GB
of log file

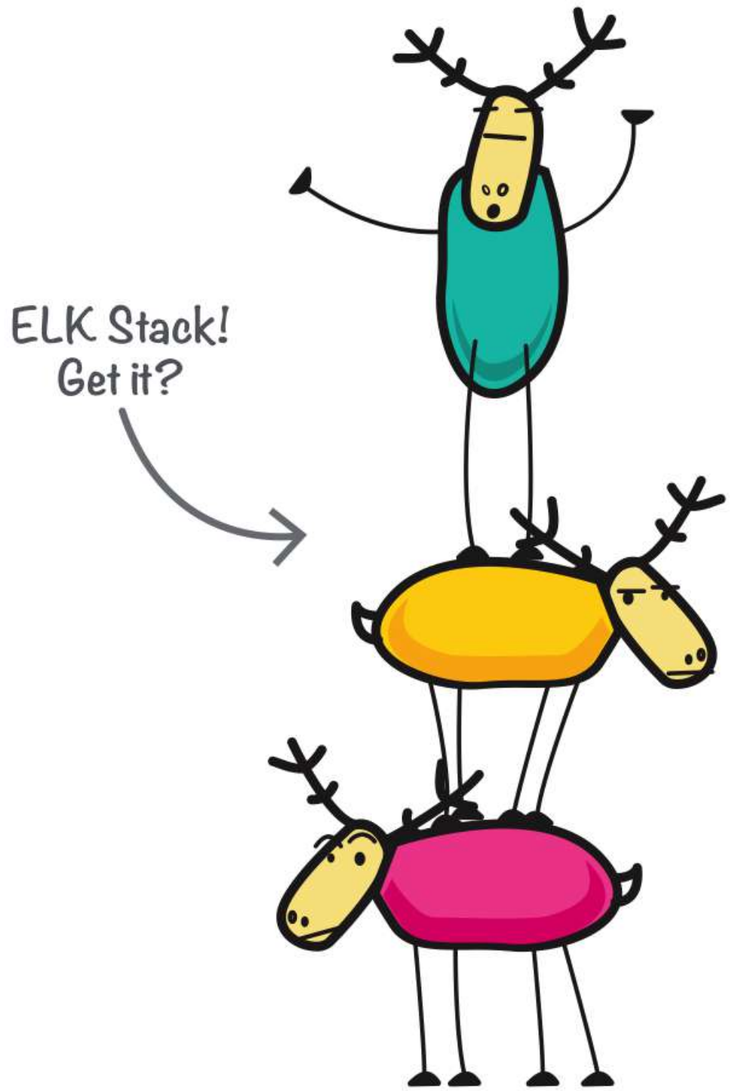

TH-
THIS IS NOT FINE!!
OH MY GOD EVERY THINGS ON FIRE
WHAT THE HELL IS MY PROBLEM


ALL THE THINGS!



elastic

Developer 🥑



E Elasticsearch

L Logstash

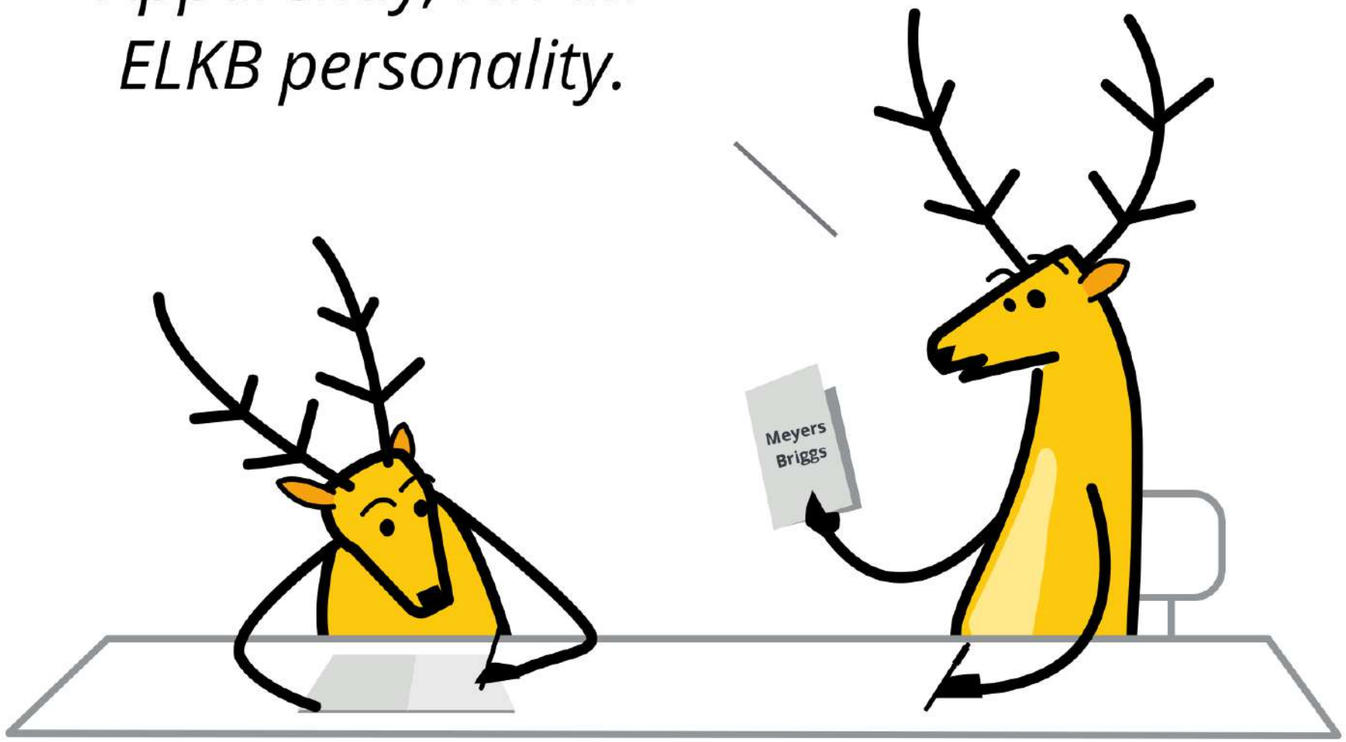
K Kibana

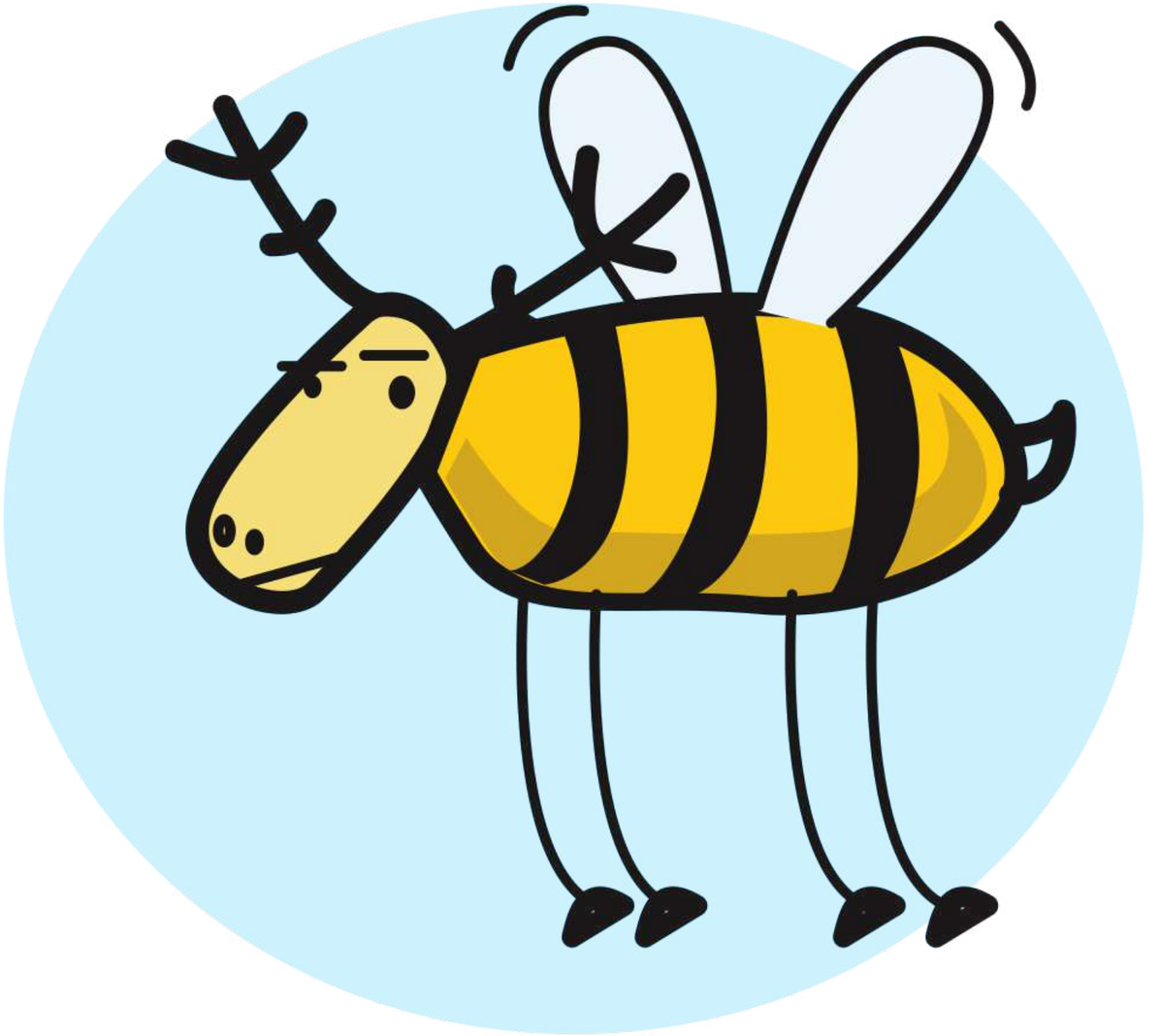

lyft

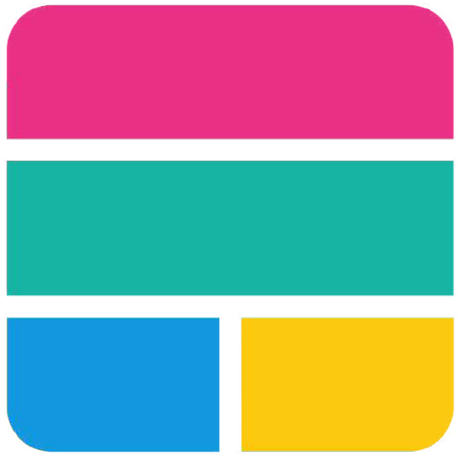

slack


fitbit

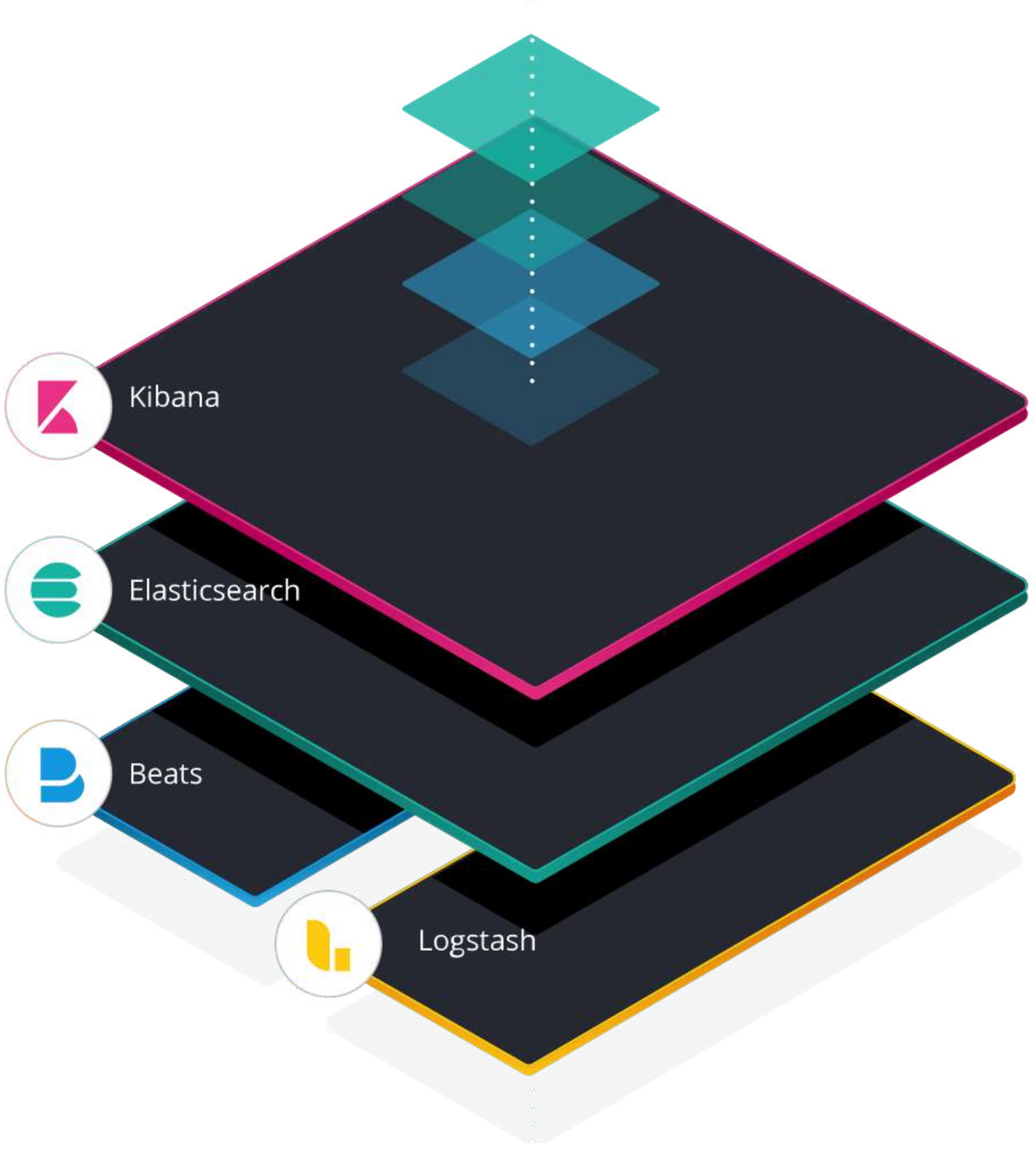
Apparently, I'm an ELKB personality.







elastic stack


Kibana
Elasticsearch
Beats
Logstash

Disclaimer

I build **highly** monitored Hello World apps

Example: PHP

Monolog

And Everywhere Else

Java: Logback / Log4j

.NET: NLog

JavaScript: Winston

Python: structlog

Anti-Pattern: echo

Logging Levels

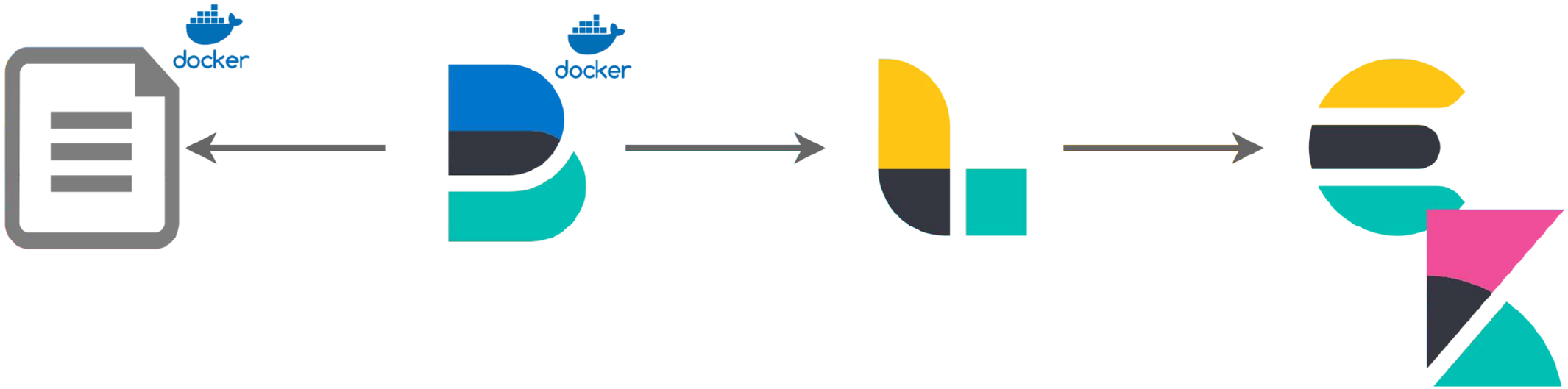
http://tools.ietf.org/html/rfc5424

PSR-3: Logger Interface

DEBUG, INFO, NOTICE, WARNING, ERROR, CRITICAL, ALERT, EMERGENCY

Anti-Pattern: Coupling

Parse


docker
docker

Bind Mount Logs

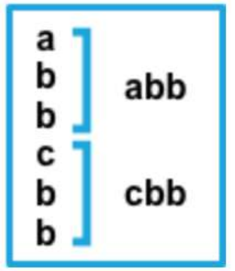
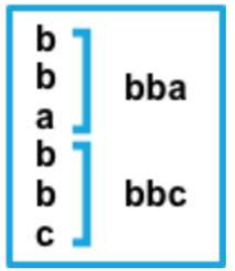
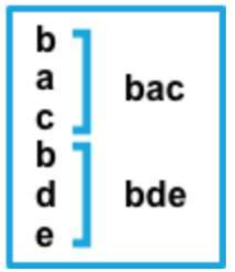
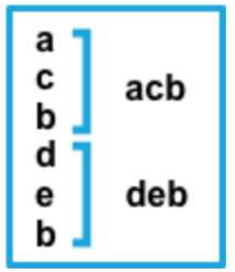
```
php_app:
  volumes:
    - './logs/:/logs/'
  ...


filebeat_for_logstash:
  volumes:
    - './logs/:/mnt/logs/:ro'
  ...
```

Collect Log Lines

```
filebeat.inputs:
- type: log
  paths:
    - /mnt/logs/*.log
```

Metadata

```
processors:
  - add_host_metadata: ~
```

Setting for `negate`	Setting for `match`	Result	Example `pattern: ^b`
`false`	`after`	Consecutive lines that match the pattern are appended to the previous line that doesn't match.	
`false`	`before`	Consecutive lines that match the pattern are prepended to the next line that doesn't match.	
`true`	`after`	Consecutive lines that don't match the pattern are appended to the previous line that does match.	
`true`	`before`	Consecutive lines that don't match the pattern are prepended to the next line that does match.	

Test Multiline Pattern

https://www.elastic.co/guide/en/beats/filebeat/current/_test_your_regexp_pattern_for_multiline.html

Grok

https://github.com/logstash-plugins/logstash-patterns-core/blob/master/patterns/grok-patterns

96 lines (85 sloc) | 5.21 KB

```
USERNAME [a-zA-Z0-9._-]+
USER %{USERNAME}
EMAILLOCALPART [a-zA-Z][a-zA-Z0-9_.+-=:]+
EMAILADDRESS %{EMAILLOCALPART}@%{HOSTNAME}
INT (?:[+-]?(?:[0-9]+))
BASE10NUM (?<![0-9.+-])(?>[+-]?(?:(?:[0-9]+(?:\.[0-9]+)?)|(?:\.[0-9]+)))
NUMBER (?:%{BASE10NUM})
BASE16NUM (?<![0-9A-Fa-f])(?:[+-]?(?:0x)?(?:[0-9A-Fa-f]+))
BASE16FLOAT \b(?<![0-9A-Fa-f.])(?:[+-]?(?:0x)?(?:(?:[0-9A-Fa-f]+(?:\.[0-9A-Fa-f]*)?)|(?

POSINT \b(?:[1-9][0-9]*)\b
NONNEGINT \b(?:[0-9]+)\b
WORD \b\w+\b
NOTSPACE \S+
SPACE \s*
DATA .*?
GREEDYDATA .*
QUOTEDSTRING (?>(?<!\\)(?>"(?>\\.|[^\\"]+)+"|""|(?>'(?>\\.|[^\\']+)+')|''|(?>`(?>\\.|[^
UUID [A-Fa-f0-9]{8}-(?:[A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}
# URN, allowing use of RFC 2141 section 2.3 reserved characters
URN urn:[0-9A-Za-z][0-9A-Za-z-]{0,31}:(?:%[0-9a-fA-F]{2}|[0-9A-Za-z()+,.:=@;$_!*'/?#-])

# Networking
MAC (?:%{CISCOMAC}|%{WINDOWSMAC}|%{COMMONMAC})
CISCOMAC (?:(?:[A-Fa-f0-9]{4}\.){2}[A-Fa-f0-9]{4})
WINDOWSMAC (?:(?:[A-Fa-f0-9]{2}-){5}[A-Fa-f0-9]{2})
```

Dev Tools

Grok Debugger

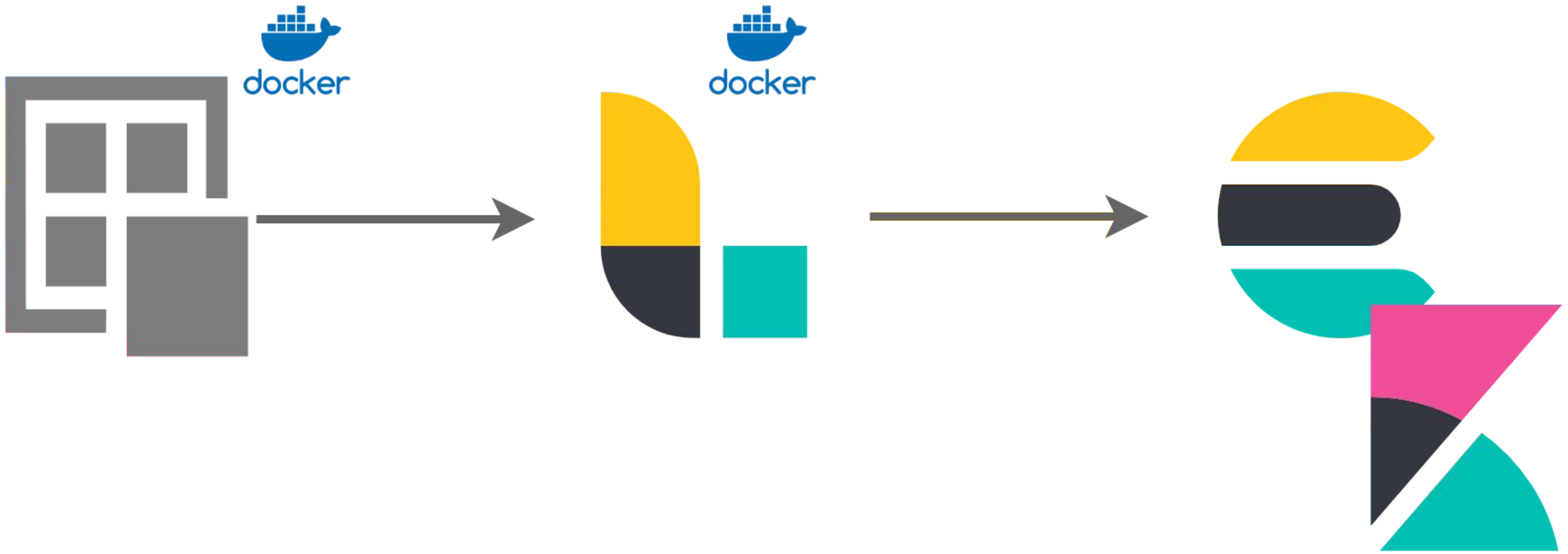
Machine Learning

Data Visualizer

Pro: No change

Con: Regular expression, multiline, format changes

Send 


docker
docker

GelfHandler

ElasticsearchHandler

...

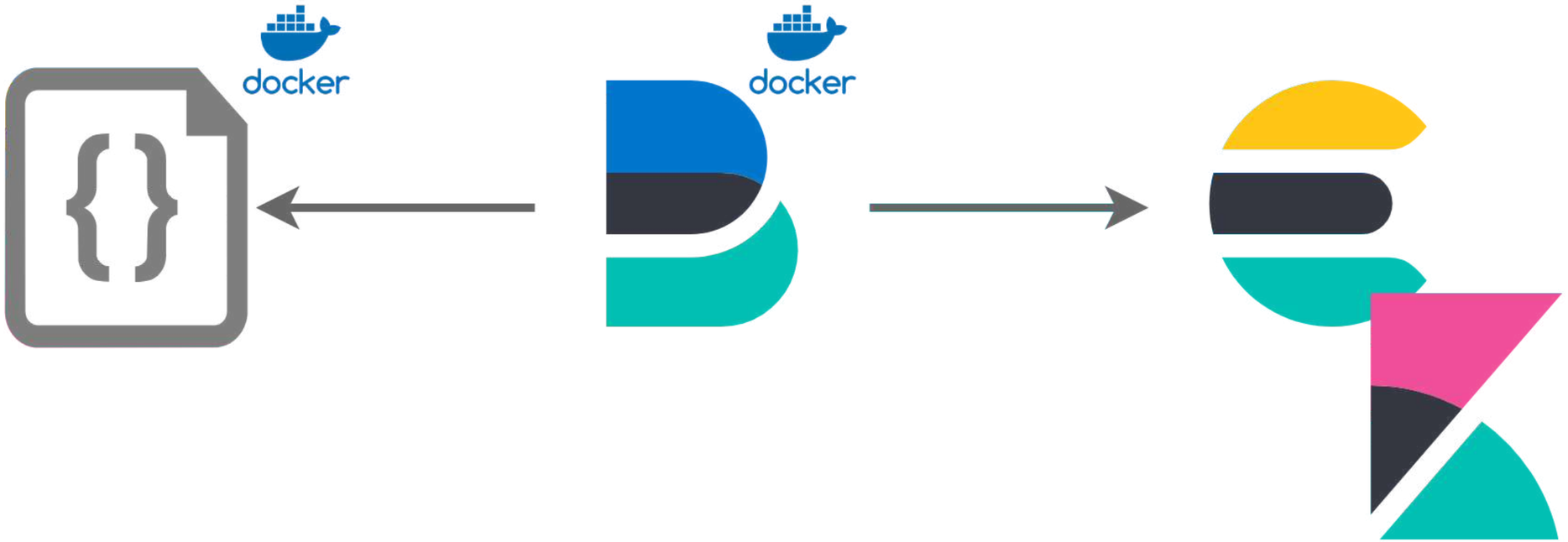
ElasticsearchHandler in Action

```
[2020-01-24 13:21:16] {"memory":2102392,"version":"1.0.0"} app_logger.DEBUG: Iteration '1'

Fatal error: Uncaught Elasticsearch\Common\Exceptions\NoNodesAvailableException:
   No alive nodes found in your cluster in
   /usr/src/app/vendor/elasticsearch/elasticsearch/src/Elasticsearch/ConnectionPool/StaticNoPingConnectionPool.php:50
Stack trace:
#0 /usr/src/app/vendor/elasticsearch/elasticsearch/src/Elasticsearch/Transport.php(77):
   Elasticsearch\ConnectionPool\StaticNoPingConnectionPool->nextConnection()
#1 /usr/src/app/vendor/elasticsearch/elasticsearch/src/Elasticsearch/Transport.php(94):
   Elasticsearch\Transport->getConnection()
#2 /usr/src/app/vendor/elasticsearch/elasticsearch/src/Elasticsearch/Connections/Connection.php(276):
   Elasticsearch\Transport->performRequest('POST', '/_bulk', Array, '{"index":{"_ind...', Array)
#3 /usr/src/app/vendor/react/promise/src/FulfilledPromise.php(25):
   Elasticsearch\Connections\Connection->Elasticsearch\Connections\{closure}(Array)
#4 /usr/src/app/vendor/guzzlehttp/ringphp/src/Future/CompletedFutureValue.php(55):
   React\Promise\FulfilledPromise->then(Object(Closure), NULL, NULL)
#5 / in /usr/src/app/vendor/monolog/monolog/src/Monolog/Handler/ElasticsearchHandler.php on line 155
```

Pro: No files

Con: Outages & coupling

Structure


docker
docker

Collect JSON

```
filebeat.input:
- type: log
  paths:
    - /mnt/logs/app.json
  json:
    message_key: message
    keys_under_root: true
    overwrite_keys: true
  fields:
    application: php
```

Elastic Common Schema

https://github.com/elastic/ecs

Event fields

The event fields are used for context information about the data itself.

Field	Description	Level	Type	Example
event.id	Unique ID to describe the event.	core	keyword	`8a4f500d`
event.category	Event category. This can be a user defined category.	core	keyword	`metrics`
event.type	A type given to this kind of event which can be used for grouping. This is normally defined by the user.	core	keyword	`nginx-stats-metrics`
event.action	The action captured by the event. The type of action will vary from system to system but is likely to include actions by security services, such as blocking or quarantining; as well as more generic actions such as login	core	keyword	`reject`

elastic

@xeraa

ECS Logging Libraries

https://github.com/elastic/ecs-logging-php

https://github.com/elastic/ecs-logging-java
https://github.com/elastic/ecs-dotnet
more to come

Log Excpetions, Errors, Throwables

```php
use Elastic\Types\Error as EcsError;

try {
    ...
}
catch(Excpetion $exception) {
    $logger->error('Meaningful message', ['error' => new EcsError($exception)]);
}
```

Service

```php
use Elastic\Types\Service;

$serviceContext = new Service();
$serviceContext->setName('my-service-x');
$serviceContext->setVersion('1.0.0');

$logger->notice('Add service context', ['service' => $serviceContext]);
```

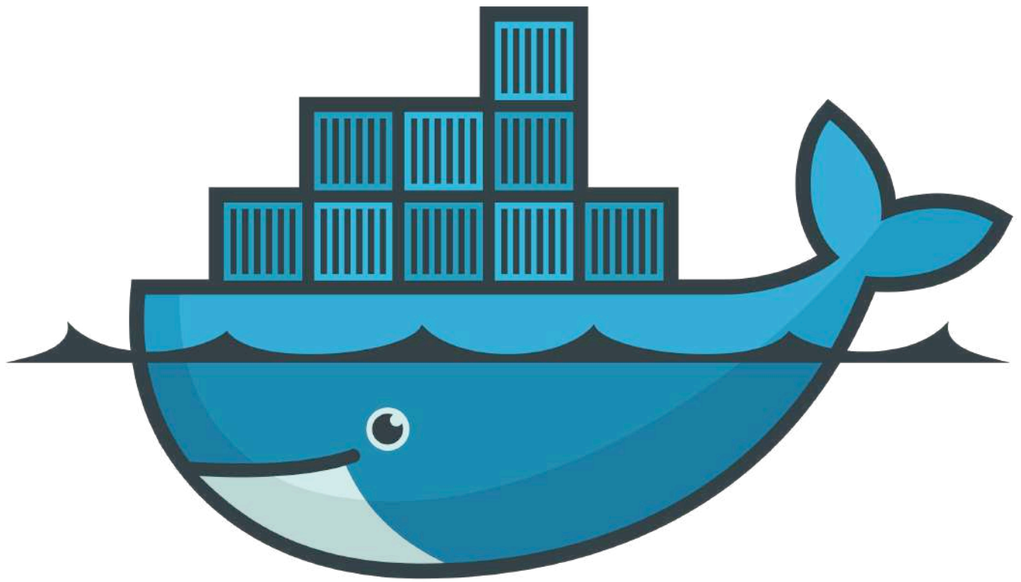
User

```php
use Elastic\Types\User;

$userContext = new User();
$userContext->setId(12345);
$userContext->setEmail('philipp@example.com');

$logger->notice('Add user information', ['user' => $userContext]);
```

Pro: Right format

Con: JSON serialization overhead

Containerize 



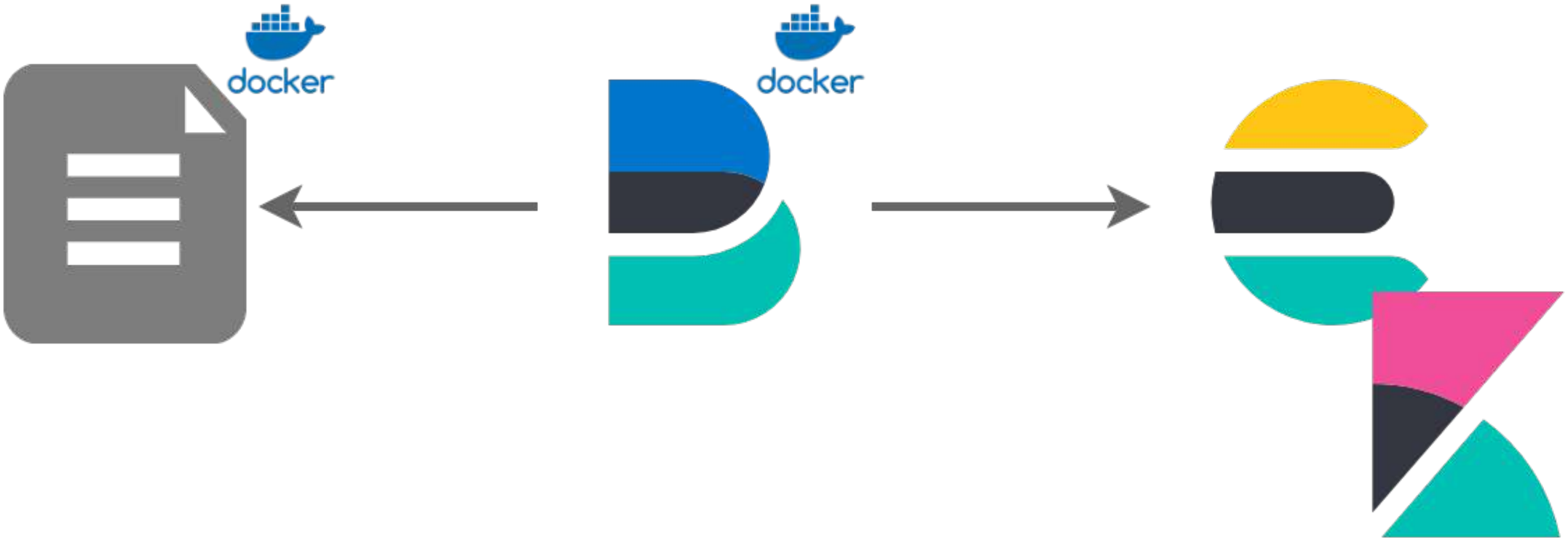
docker

elastic

@xeraa

Where to put Filebeat?

Sidecar


docker
docker

https://github.com/elastic/beats/tree/master/deploy/docker

Docker Logs

```
filebeat.autodiscover:
  providers:
    - type: docker
      hints.enabled: true

processors:
  - add_docker_metadata: ~
```

Metadata

No Docker metadata with the other methods

```
"docker": {
  "container": {
    "labels": {
      "app": "fizzbuzz",
      "co_elastic_logs/multiline_match": "after",
      "com_docker_compose_config-hash": "41520c6cf2b6a1f3dae4f16d0a6fd76760cdfc38fbfe43a3a3be2e09bdd1b8b5",
      "environment": "production",
      "co_elastic_logs/multiline_pattern": "^\\[",
      "co_elastic_logs/multiline_negate": "true",
      "com_docker_compose_oneoff": "False",
      "com_docker_compose_project": "php-logging",
      "com_docker_compose_service": "php_app",
      "com_docker_compose_container-number": "1",
      "com_docker_compose_version": "1.23.2"
    }
  }
}
```

Missing the Last Line

Waiting for the newline

Hints

```
labels:
  - "app=fizzbuzz"
  - "co.elastic.logs/multiline.pattern=^\\["
  - "co.elastic.logs/multiline.negate=true"
  - "co.elastic.logs/multiline.match=after"
```

Registry File

```
filebeat.registry.path: /usr/share/filebeat/data/registry
```

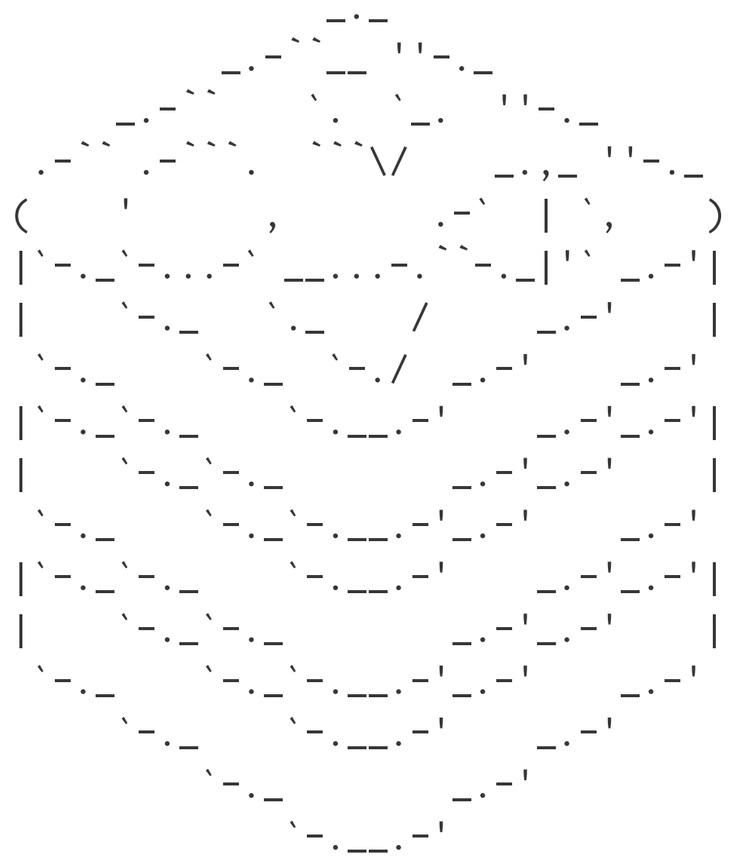
Multi-Index

```
output.elasticsearch:
  hosts: ["http://localhost:9200"]
  indices:
    - index: "docker-php-%{+yyyy.MM}-00"
      when.contains:
        container.name: "docker_php"
```

Unknown Fields

?	log.labels	⚠ session=69, loop=20
t	log.level	WARN
?	log.method	⚠ main
#	log.offset	19,744
?	log.package	⚠ net.xeraa.logging.LogMe
t	message	[2019-05-21 05:02:07.458] WARN net.xeraa.logging.LogMe [main] - session=69, loop=20 - Investigate tomorrow
?	message_parsed	⚠ Investigate tomorrow
t	stream	stdout
🕘	suricata.eve.timestamp	May 21, 2019 @ 07:02:07.459
?	timestamp	⚠ 2019-05-21 05:02:07.458

ASCII Art



```
Redis 4.0.9 (00000000/0) 64 bit

Running in stand alone mode
Port: 6379
PID: 55757


      http://redis.io
```

Configuration Templates

```
filebeat.autodiscover:
  providers:
    - type: docker
      templates:
        - condition:
            equals:
              docker.container.image: redis
          config:
            - type: docker
              containers.ids:
                - "${data.docker.container.id}"
              exclude_lines: ["^\\s+[\\-`('.|_]"]
```

Who Logs the Logger

Avoid loops

Process without `-e`

filebeat.yml: `logging.to_files: true`

Pro: Hot 💩

Con: Complexity

Orchestrate 

kubernetes

elastic

@xeraa

Where to put Filebeat?

DaemonSet

https://github.com/elastic/beats/tree/master/deploy/kubernetes

Metadata

Either in cluster or outside

```
processors:

- add_kubernetes_metadata:
    in_cluster: true

- add_kubernetes_metadata:
    in_cluster: false
    host: <hostname>
    kube_config: ${HOME}/.kube/config
```

```
{
  "host": "172.17.0.21",
  "port": 9090,
  "kubernetes": {
    "container": {
      "id": "382184ecdb385cfd5d1f1a65f78911054c8511ae009635300ac28b4fc357ce51",
      "image": "my-php:1.0.0",
      "name": "my-php"
    },
    "labels": {
      "app": "php_app",
    },
    "namespace": "default",
    "node": {
      "name": "minikube"
    },
    "pod": {
      "name": "php-2657348378-k1pnh"
    }
  },
}
```

More Metadata

Add: Cloud, local timezone, process

Drop: Events, fields

Rename: Fields

Dissect, DNS reverse lookup

Configuration Templates

```
filebeat.autodiscover:
  providers:
    - type: kubernetes
      templates:
        - condition:
            equals:
              kubernetes.namespace: redis
          config:
            - type: docker
              containers.ids:
                - "${data.kubernetes.container.id}"
              exclude_lines: ["^\\s+[\\-`('.|_]"]
```

Customize Indices

```
output.elasticsearch:
  index: "%{[kubernetes.namespace]:filebeat}-%{[beat.version]}-%{+yyyy.MM.dd}"
```

Pro: Hot 💩💩💩

Con: Complexity++

PS: Filebeat Modules

```
filebeat.modules:
- module: apache2
- module: mysql
- module: nginx
- module: system
```

Conclusion

Examples

https://github.com/xeraa/php-logging

Parse 🔪

Send ✉️

Structure 🕸

Containerize 📦

Orchestrate 🎻

Questions?

Philipp Krenn @xeraa