

Centralized Logging-Patterns

Philipp Krenn

@xeraa



@xeraa


```
java-logging — fish /Users/philipp/Documents/GitHub/java-logging — -fish — 105x26
[philipp@~/Documents/GitHub/java-logging(git*master)*> gradle run 14:47:14]

> Task :run
[2018-05-31 14:47:22.185] TRACE net.xeraa.logging.LogMe [main] - session=29, loop=1 - Iteration '1' and session '29'
[2018-05-31 14:47:22.196] DEBUG net.xeraa.logging.LogMe [main] - session=29, loop=1 - Collect in development
[2018-05-31 14:47:22.200] TRACE net.xeraa.logging.LogMe [main] - session=49, loop=2 - Iteration '2' and session '49'
[2018-05-31 14:47:22.201] DEBUG net.xeraa.logging.LogMe [main] - session=49, loop=2 - Collect in development
[2018-05-31 14:47:22.202] TRACE net.xeraa.logging.LogMe [main] - session=85, loop=3 - Iteration '3' and session '85'
[2018-05-31 14:47:22.203] INFO net.xeraa.logging.LogMe [main] - session=85, loop=3 - Collect in production
[2018-05-31 14:47:22.204] TRACE net.xeraa.logging.LogMe [main] - session=55, loop=4 - Iteration '4' and session '55'
[2018-05-31 14:47:22.204] DEBUG net.xeraa.logging.LogMe [main] - session=55, loop=4 - Collect in development
[2018-05-31 14:47:22.205] TRACE net.xeraa.logging.LogMe [main] - session=83, loop=5 - Iteration '5' and session '83'
[2018-05-31 14:47:22.205] WARN net.xeraa.logging.LogMe [main] - session=83, loop=5 - Investigate tomorrow
[2018-05-31 14:47:22.206] TRACE net.xeraa.logging.LogMe [main] - session=36, loop=6 - Iteration '6' and session '36'
[2018-05-31 14:47:22.206] INFO net.xeraa.logging.LogMe [main] - session=36, loop=6 - Collect in production
```

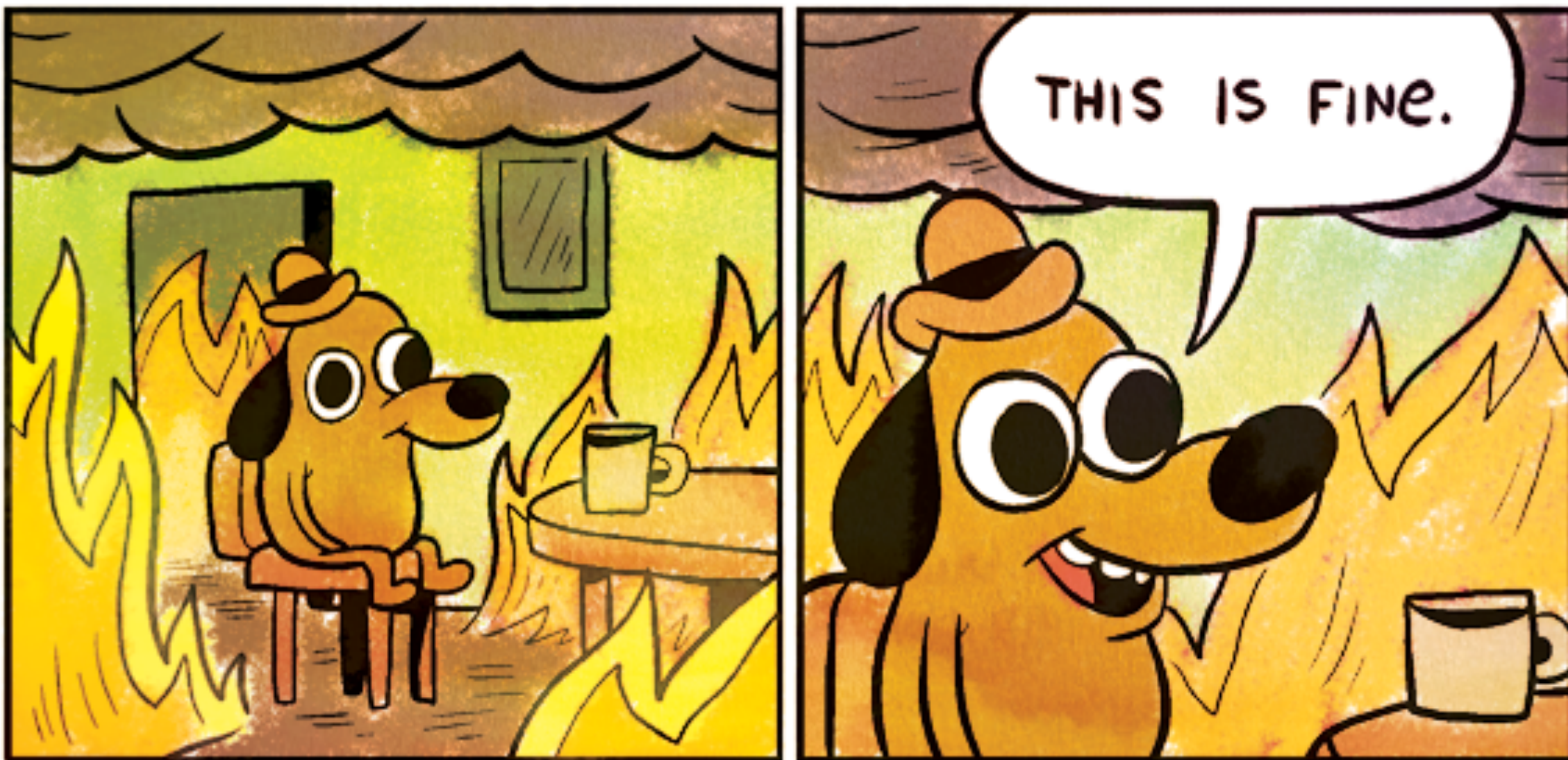
```
java-logging — fish /Users/philipp/Documents/GitHub/java-logging — -fish — 105x26
[philipp@~/Documents/GitHub/java-logging(git*master)✓> cat logs/java-logging.log 14:47:23]
[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and s
ession '46'
[2018-05-31 14:42:58.961] DEBUG net.xeraa.logging.LogMe [main] - session=46, loop=1 - Collect in developm
ent
[2018-05-31 14:42:58.963] TRACE net.xeraa.logging.LogMe [main] - session=13, loop=2 - Iteration '2' and s
ession '13'
[2018-05-31 14:42:58.963] DEBUG net.xeraa.logging.LogMe [main] - session=13, loop=2 - Collect in developm
ent
[2018-05-31 14:42:58.964] TRACE net.xeraa.logging.LogMe [main] - session=70, loop=3 - Iteration '3' and s
ession '70'
[2018-05-31 14:42:58.964] INFO net.xeraa.logging.LogMe [main] - session=70, loop=3 - Collect in producti
on
[2018-05-31 14:42:58.965] TRACE net.xeraa.logging.LogMe [main] - session=68, loop=4 - Iteration '4' and s
ession '68'
[2018-05-31 14:42:58.966] DEBUG net.xeraa.logging.LogMe [main] - session=68, loop=4 - Collect in developm
ent
[2018-05-31 14:42:58.966] TRACE net.xeraa.logging.LogMe [main] - session=84, loop=5 - Iteration '5' and s
ession '84'
[2018-05-31 14:42:58.966] WARN net.xeraa.logging.LogMe [main] - session=84, loop=5 - Investigate tomorro
w
[2018-05-31 14:42:58.967] TRACE net.xeraa.logging.LogMe [main] - session=82, loop=6 - Iteration '6' and s
ession '82'
[2018-05-31 14:42:58.969] INFO net.xeraa.logging.LogMe [main] - session=82, loop=6 - Collect in producti
on
[2018-05-31 14:42:58.969] TRACE net.xeraa.logging.LogMe [main] - session=7, loop=7 - Iteration '7' and se
```



```
java-logging — tail /Users/philipp/Documents/GitHub/java-logging — tail -f logs/java-loggi...
[philipp@~/Documents/GitHub/java-logging(git*master)✓> tail -f logs/java-logging.log 18:39:45]
[2018-05-31 17:20:22.874] TRACE net.xeraa.logging.LogMe [main] - session=61, loop=16 - Iteration '16' and
session '61'
[2018-05-31 17:20:22.874] DEBUG net.xeraa.logging.LogMe [main] - session=61, loop=16 - Collect in develop
ment
[2018-05-31 17:20:22.881] TRACE net.xeraa.logging.LogMe [main] - session=2, loop=17 - Iteration '17' and
session '2'
[2018-05-31 17:20:22.882] DEBUG net.xeraa.logging.LogMe [main] - session=2, loop=17 - Collect in developm
ent
[2018-05-31 17:20:22.883] TRACE net.xeraa.logging.LogMe [main] - session=35, loop=18 - Iteration '18' and
session '35'
[2018-05-31 17:20:22.884] INFO net.xeraa.logging.LogMe [main] - session=35, loop=18 - Collect in product
ion
[2018-05-31 17:20:22.886] TRACE net.xeraa.logging.LogMe [main] - session=86, loop=19 - Iteration '19' and
session '86'
[2018-05-31 17:20:22.889] DEBUG net.xeraa.logging.LogMe [main] - session=86, loop=19 - Collect in develop
ment
[2018-05-31 17:20:22.890] TRACE net.xeraa.logging.LogMe [main] - session=92, loop=20 - Iteration '20' and
session '92'
[2018-05-31 17:20:22.891] WARN net.xeraa.logging.LogMe [main] - session=92, loop=20 - Investigate tomorr
OW
[2018-05-31 18:40:05.399] TRACE net.xeraa.logging.LogMe [main] - session=40, loop=1 - Iteration '1' and s
ession '40'
[2018-05-31 18:40:05.417] DEBUG net.xeraa.logging.LogMe [main] - session=40, loop=1 - Collect in developm
ent
[2018-05-31 18:40:05.420] TRACE net.xeraa.logging.LogMe [main] - session=51, loop=2 - Iteration '2' and s
```



```
java-logging — fish /Users/philipp/Documents/GitHub/java-logging — -fish — 105x26
philipp@~/Documents/GitHub/java-logging(git*master)➤ less +F logs/java-logging.log 18:42:08
```



java-logging — fish /Users/philipp/Documents/GitHub/java-logging — -fish — 105x12

[cat logs/java-logging.log]

[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and session '46'

[2018-05-31 14:42:58.961] DEBUG net.xeraa.logging.LogMe [main] - session=46, loop=1 - Collect in development

[2018-05-31 14:42:58.963] TRACE net.xeraa.logging.LogMe [main] - session=13, loop=2 - Iteration '2' and s

[cat logs/java-logging.log]

[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and session '46'

[2018-05-31 14:42:58.961] DEBUG net.xeraa.logging.LogMe [main] - session=46, loop=1 - Collect in development

[2018-05-31 14:42:58.963] TRACE net.xeraa.logging.LogMe [main] - session=13, loop=2 - Iteration '2' and s

[cat logs/java-logging.log]

[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and session '46'

[2018-05-31 14:42:58.961] DEBUG net.xeraa.logging.LogMe [main] - session=46, loop=1 - Collect in development

[2018-05-31 14:42:58.963] TRACE net.xeraa.logging.LogMe [main] - session=13, loop=2 - Iteration '2' and session '13'

[2018-05-31 14:42:58.963] DEBUG net.xeraa.logging.LogMe [main] - session=13, loop=2 - Collect in development

[2018-05-31 14:42:58.964] TRACE net.xeraa.logging.LogMe [main] - session=70, loop=3 - Iteration '3' and session '70'

[2018-05-31 14:42:58.964] INFO net.xeraa.logging.LogMe [main] - session=70, loop=3 - Collect in producti

```
java-logging — fish /Users/philipp/Documents/GitHub/java-logging — -fish — 105x1
[cat logs/java-logging.log]
[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and s

[cat logs/java-logging.log]
[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and s

[cat logs/java-logging.log]
[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and s

[cat logs/java-logging.log]
[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and s

[cat logs/java-logging.log]
[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and s

[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and s
session '46'

[cat logs/java-logging.log]
[2018-05-31 14:42:58.951] TRACE net.xeraa.logging.LogMe [main] - session=46, loop=1 - Iteration '1' and s

[cat logs/java-logging.log]
```



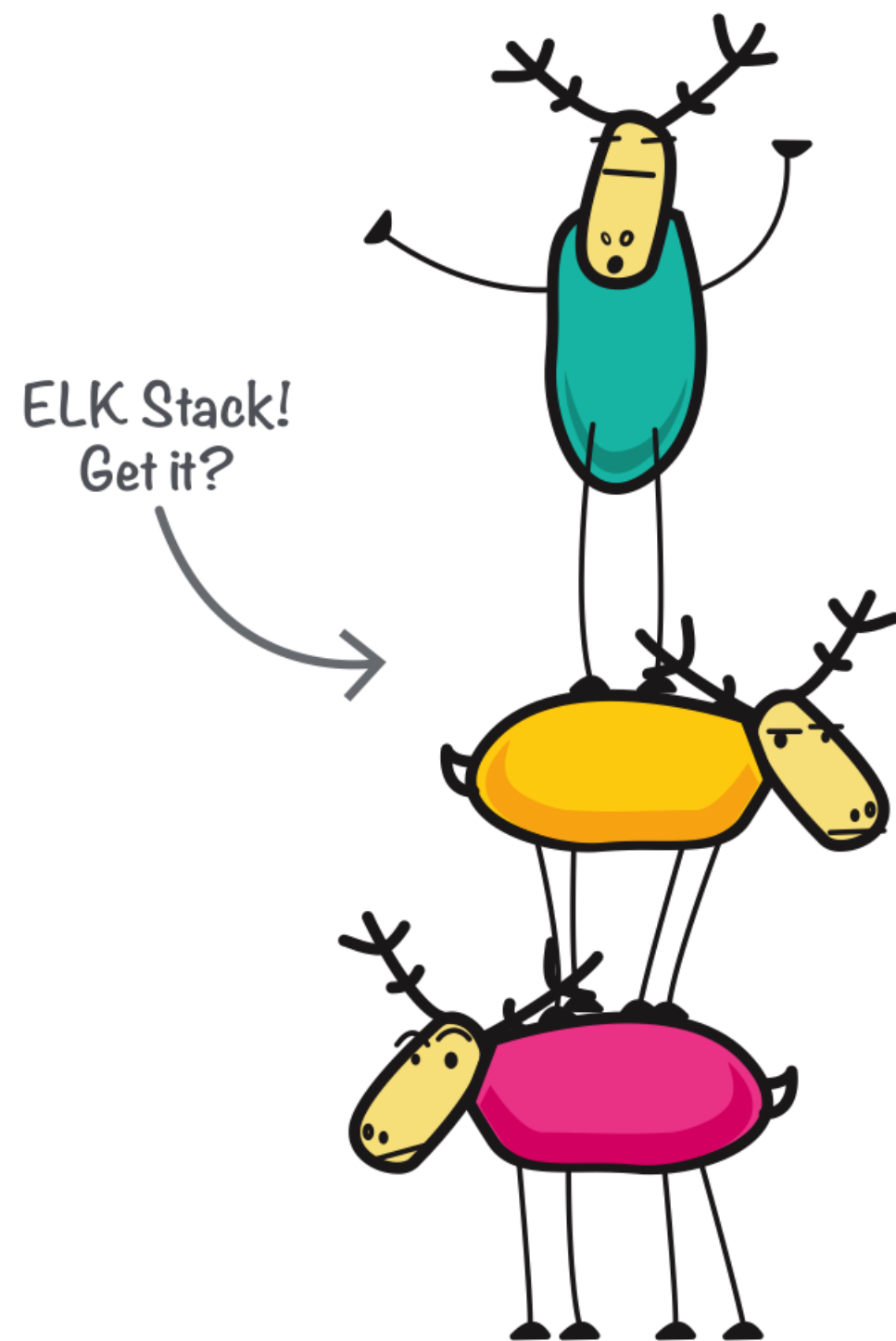

ALL THE THINGS!





elastic

Developer 🥑



E Elasticsearch

L Logstash

K Kibana

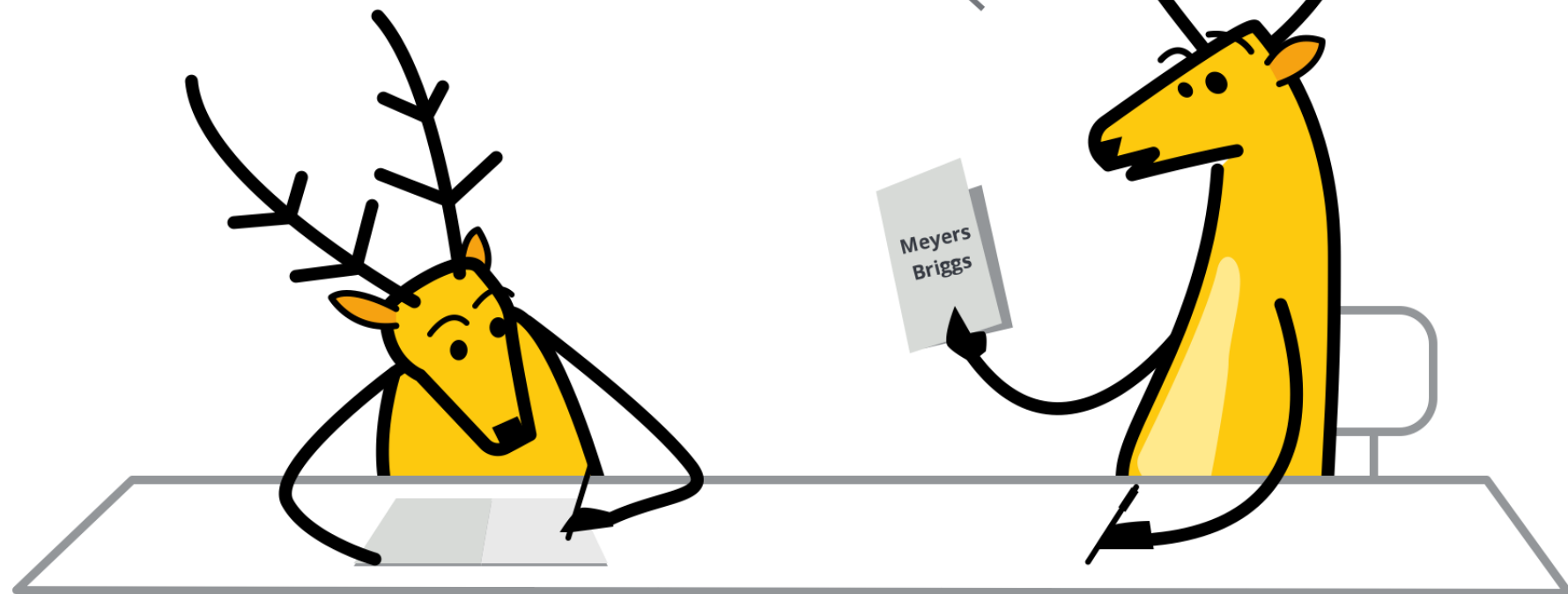


slack



fitbit

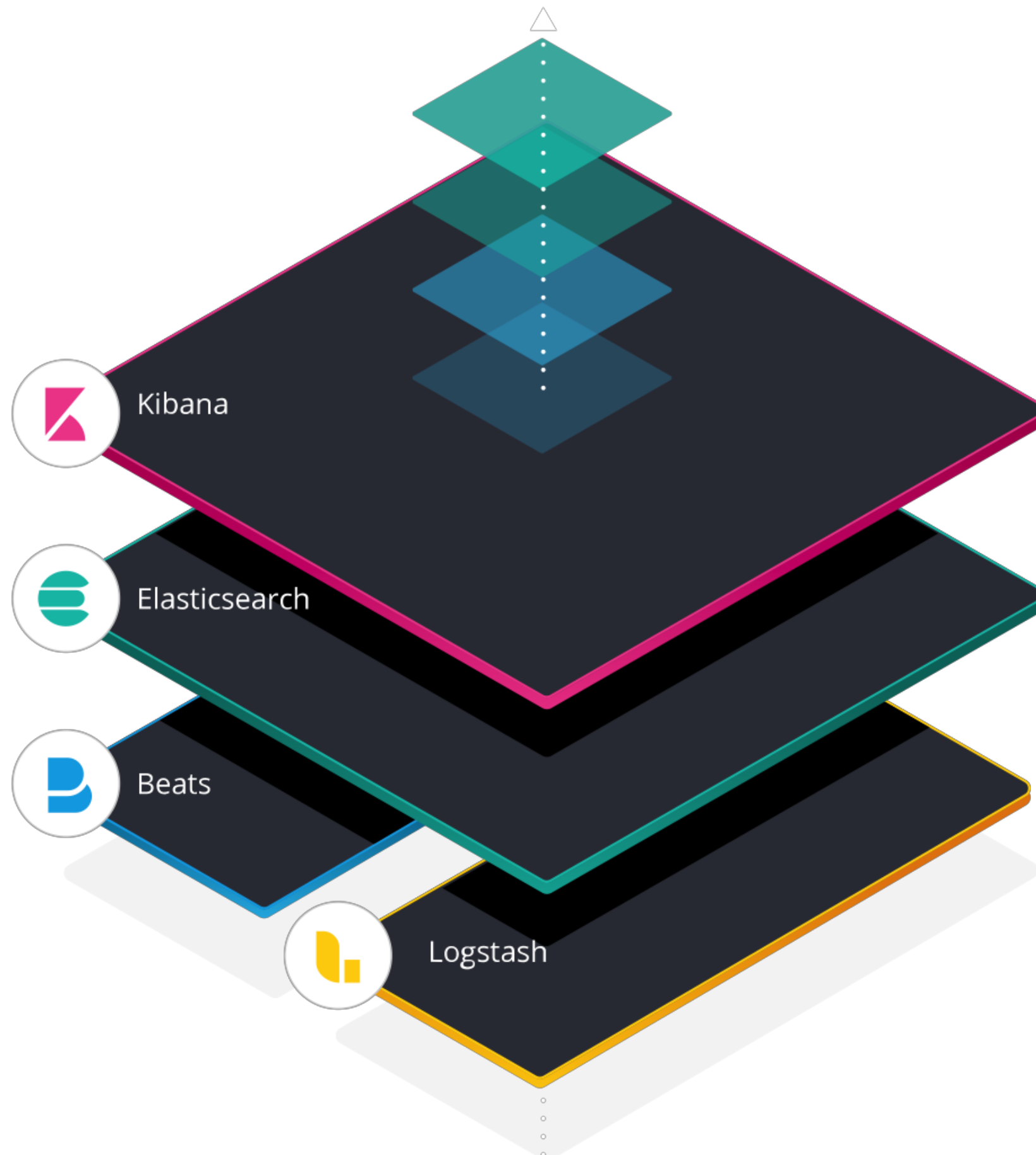
*Apparently, I'm an
ELKB personality.*







elastic stack



Licensing

Open Source Apache-2.0

Basic free

Commercial 

Disclaimer

I build **highly** monitored Hello World
apps

Example: Java

SLF4J, Logback, MDC

And Everywhere Else

.NET: NLog

PHP: Monolog

JavaScript: Winston

Python: structlog

Anti-Pattern: `print`

```
System.out.println("Oops");
```

Anti-Pattern: Coupling

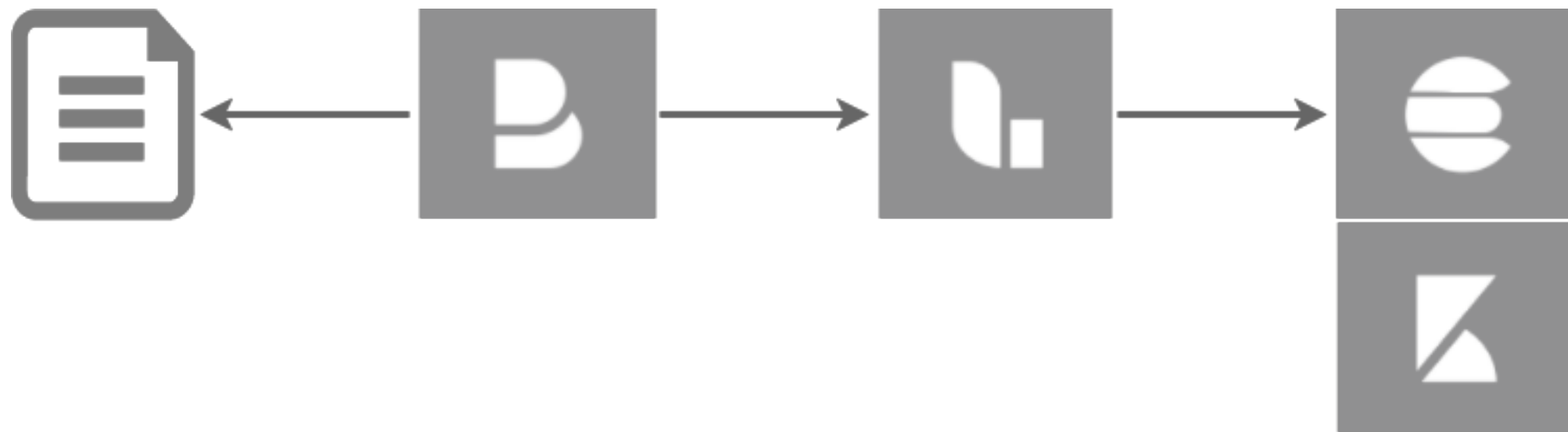


A close-up shot of a man with dark hair and round glasses, looking upwards with a wide-eyed, surprised expression. The image has a blue tint. The text "HOLD ON TO YOUR BUTTS" is overlaid at the bottom in white, bold, sans-serif font.

HOLD ON TO YOUR BUTTS

Parse





Collect Log Lines

```
filebeat.inputs:  
- type: log  
  paths:  
    - /mnt/logs/*.log  
#exclude_lines: ['^TRACE']
```

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.	 The diagram shows a list of input lines: 'a', 'b', 'b', 'c', 'b', 'b'. Blue brackets group the first three lines into 'abb' and the last three lines into 'cbb'. The line 'a' does not match the pattern ^b, while the subsequent lines 'b', 'b', 'c', 'b', 'b' do match.
false	before	Consecutive lines that match the pattern are prepended to the next line that doesn't match.	 The diagram shows a list of input lines: 'b', 'b', 'a', 'b', 'b', 'c'. Blue brackets group the first two lines into 'bba' and the last two lines into 'bbc'. The line 'a' does not match the pattern ^b, while the lines before and after it ('b', 'b' and 'b', 'b') do match.
true	after	Consecutive lines that don't match the pattern are appended to the previous line that does match.	 The diagram shows a list of input lines: 'b', 'a', 'c', 'b', 'd', 'e'. Blue brackets group the first three lines into 'bac' and the last three lines into 'bde'. The line 'b' matches the pattern ^b, while the lines 'a', 'c', 'd', and 'e' do not.
true	before	Consecutive lines that don't match the pattern are prepended to the next line that does match.	 The diagram shows a list of input lines: 'a', 'c', 'b', 'd', 'e', 'b'. Blue brackets group the first three lines into 'acb' and the last three lines into 'deb'. The line 'b' matches the pattern ^b, while the lines 'a', 'c', 'd', and 'e' do not.


```
[2018-09-28 10:30:38.516] ERROR net.xeraa.logging.LogMe [main] -  
    user_experience=🤔, session=46, loop=15 -  
    Wake me up at night  
java.lang.RuntimeException: Bad runtime...  
    at net.xeraa.logging.LogMe.main(LogMe.java:30)
```

```
^\[%{TIMESTAMP_ISO8601:timestamp}\] %{SPACE} %{LOGLEVEL:level}  
  %{SPACE} %{USERNAME:logger} %{SPACE} \[%{WORD:thread}\]  
  %{SPACE} - %{SPACE} %{GREEDYDATA:mdc} %{SPACE} - %{SPACE}  
  %{GREEDYDATA:themessage} (?:\n+(?<stacktrace>(?:.|\r|\n)+))?
```

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

Dev Tools

Grok Debugger

Sample Data

```
1 [2018-11-16 01:16:59.983] ERROR net.xeraa.logging.LogMe [main] - user_experience=👎, ses
```

Grok Pattern

```
1 \[%{TIMESTAMP_ISO8601:timestamp}\] %{LOGLEVEL:loglevel}
```

> Custom Patterns

[Simulate](#)

Structured Data

```
1 {  
2   "loglevel": "ERROR",  
3   "timestamp": "2018-11-16 01:16:59.983"  
4 }
```

Machine Learning Data Visualizer

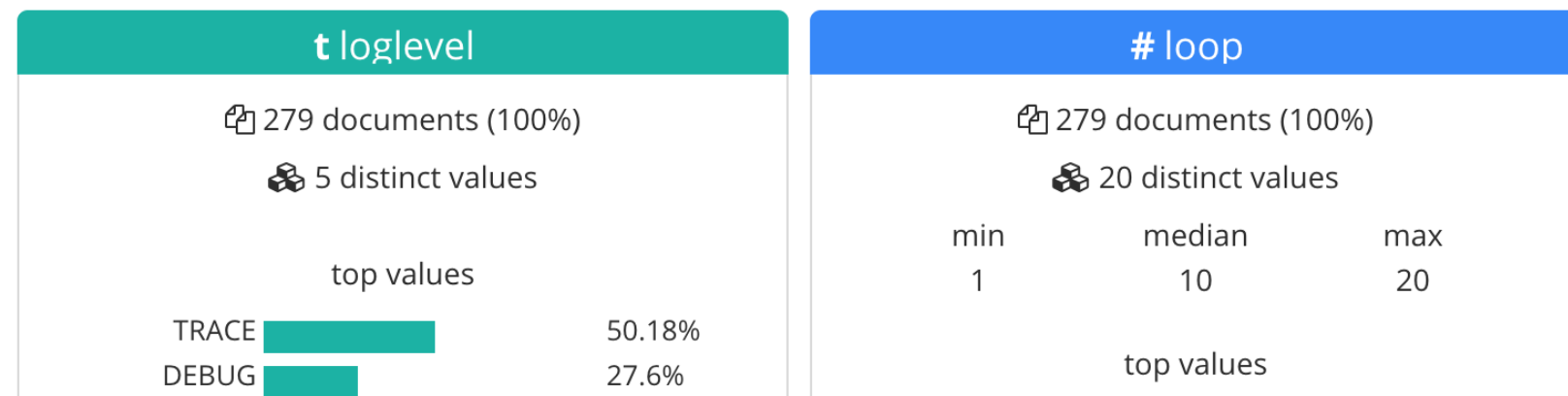
```
28 [2018-11-16 01:16:59.976] DEBUG net.xeraa.logging.LogMe [main] - session=94, loop=14 - Collect
29 [2018-11-16 01:16:59.977] TRACE net.xeraa.logging.LogMe [main] - session=43, loop=15 - Iteration
30 [2018-11-16 01:16:59.983] ERROR net.xeraa.logging.LogMe [main] - user_experience=😞, session=43
31 java.lang.RuntimeException: Bad runtime...
```

Summary

Number of lines analyzed	293
Format	semi_structured_text
Grok pattern	\[%{TIMESTAMP_ISO8601:timestamp}\] %{LOGLEVEL:loglevel} .*? .*?\[.*?\] .*? .*?\bsessi
Time field	timestamp
Time format	YYYY-MM-dd HH:mm:ss.SSS

[Override settings](#)

File stats



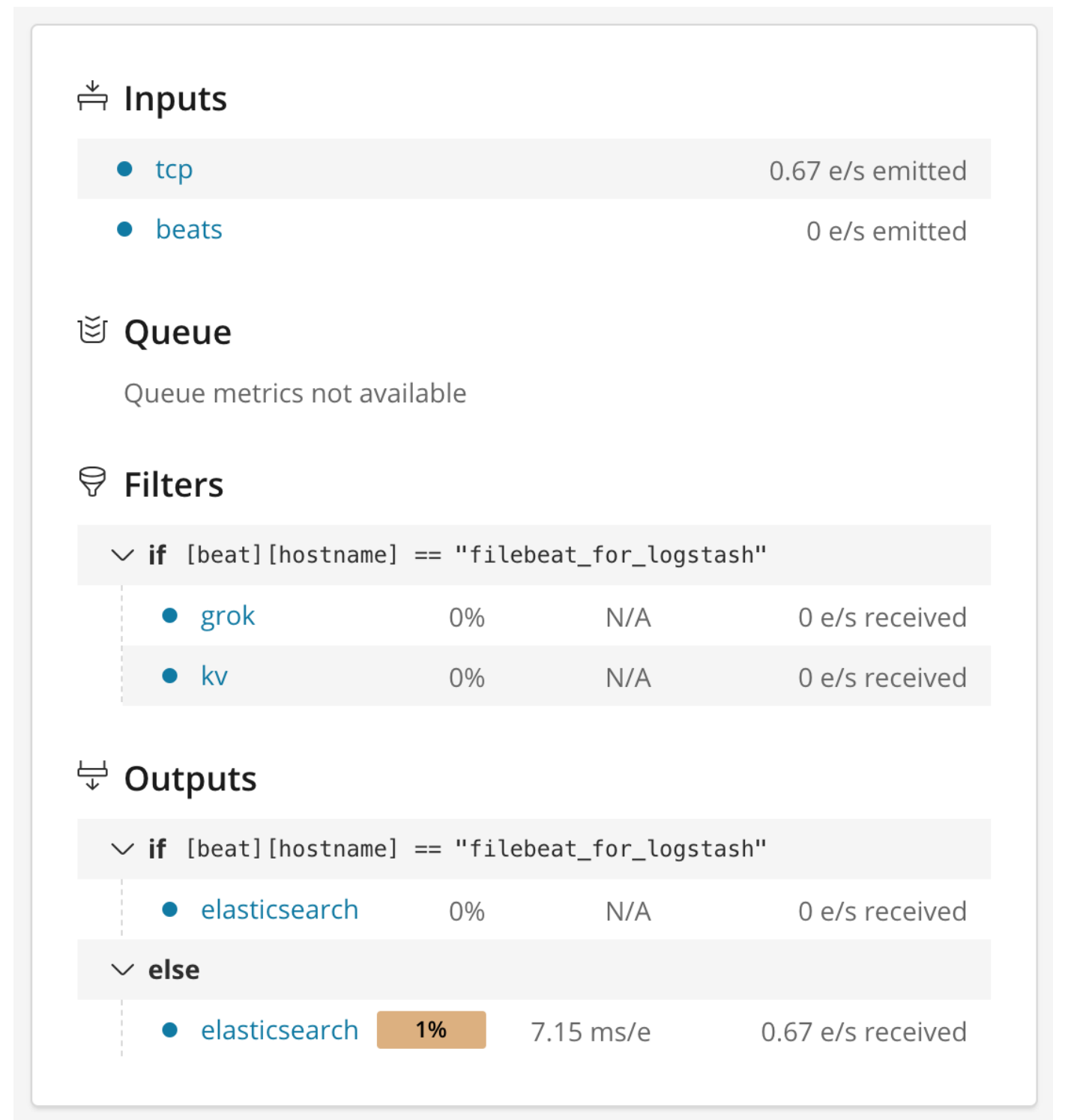
Visualize

For example session

Log UI

Monitoring: Logstash Pipeline

Plus other components

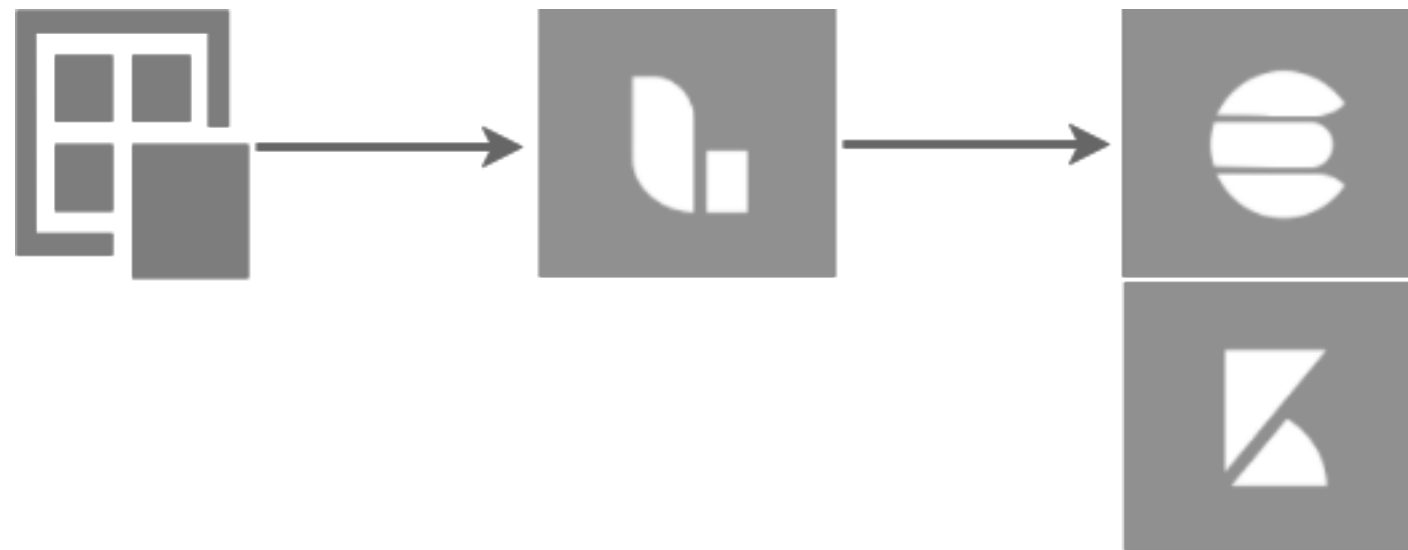


Pro: No change

Con: RegEx, timestamp, multiline

Send



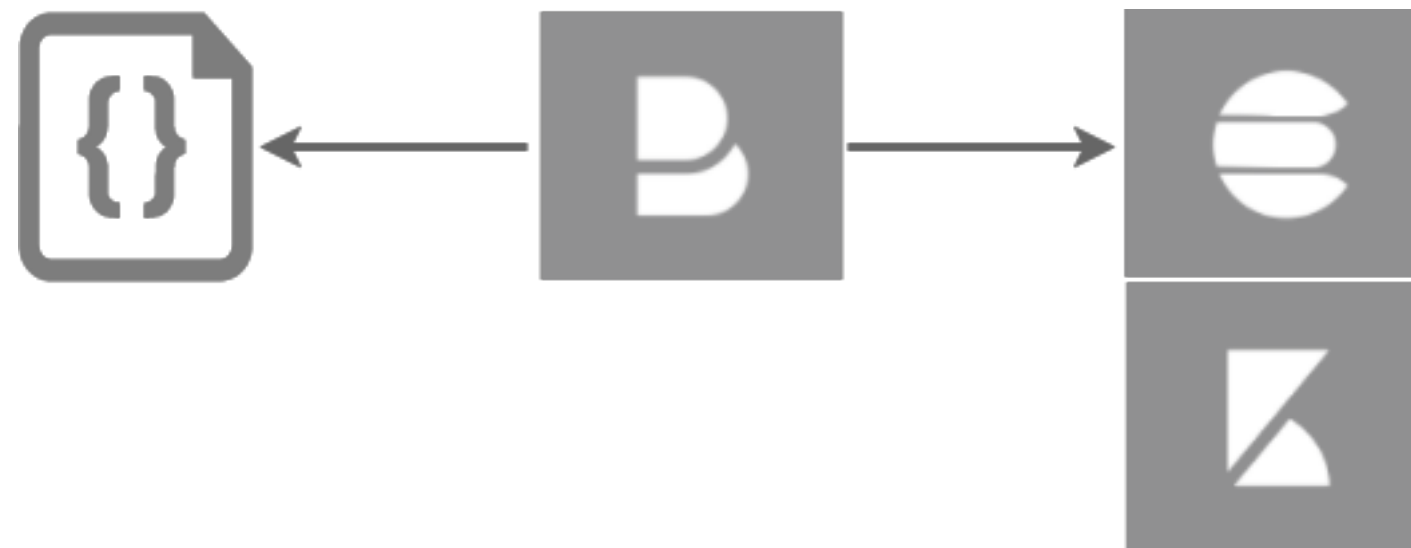


Pro: No files

Con: Outages & coupling

Structure





Collect JSON

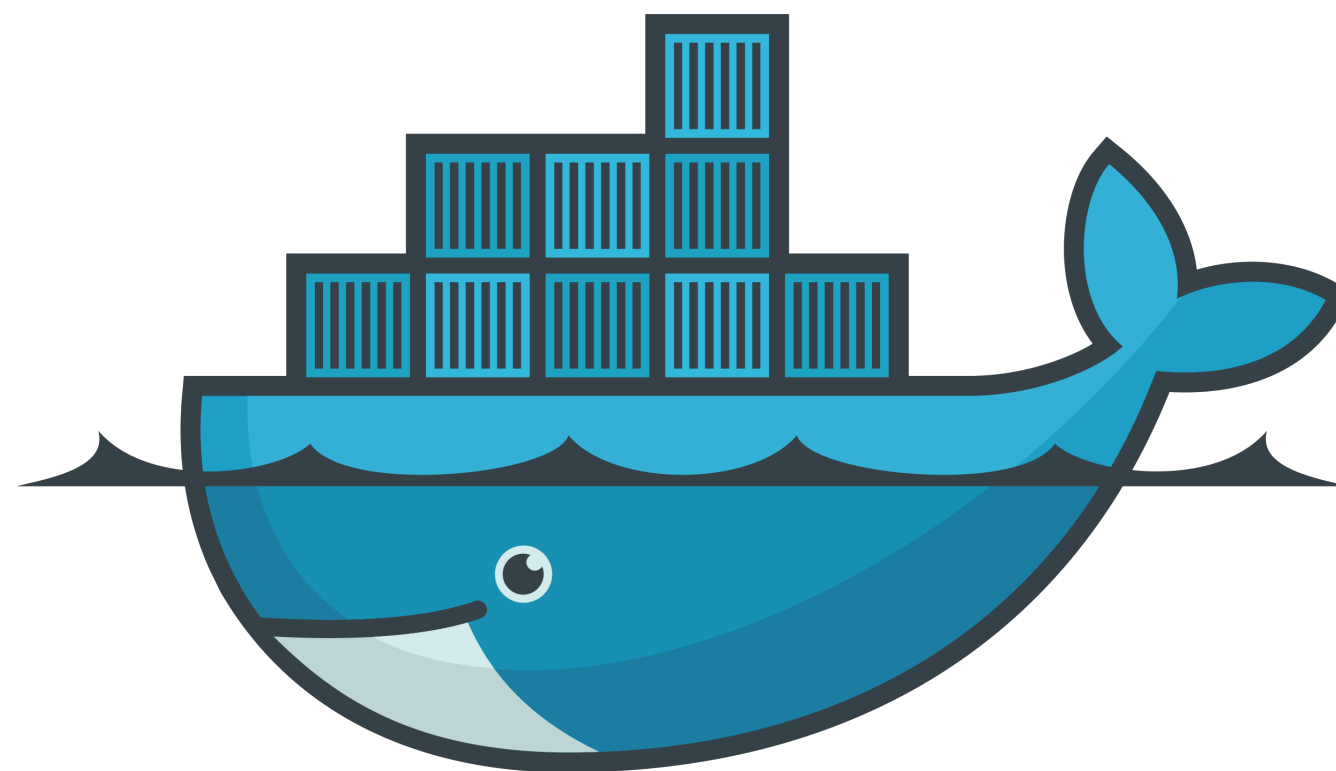
```
filebeat.input:  
- type: log  
  paths:  
    - /mnt/logs/*.json  
fields_under_root: true  
json:  
  message_key: message  
  keys_under_root: true  
  
processors:  
- add_host_metadata: ~
```

Pro: Right format

Con: JSON serialization overhead

Containerize





docker

Where to put Filebeat?

Sidecar

Default JSON Log

```
filebeat.input:
```

```
- type: log
```

```
  paths:
```

```
    - "/var/lib/docker/containers/*//*.log"
```

```
  json.message_key: log
```

```
  json.keys_under_root: true
```

```
processors:
```

```
- add_docker_metadata: ~
```

```
- add_host_metadata: ~
```

Metadata

```
{  
  "host": "10.4.15.9",  
  "port": 6379,  
  "docker": {  
    "container": {  
      "id": "382184ecdb385cf5d1f1a65f78911054c8511ae009635300ac28b4fc357ce51",  
      "name": "my-java",  
      "image": "my-java:1.0.0",  
      "labels": {  
        "app": "java"  
      }  
    }  
  }  
}
```

Logging as JSON

```
- decode_json_fields:  
  fields: ["message"]  
  target: ""  
  overwrite_keys: true
```

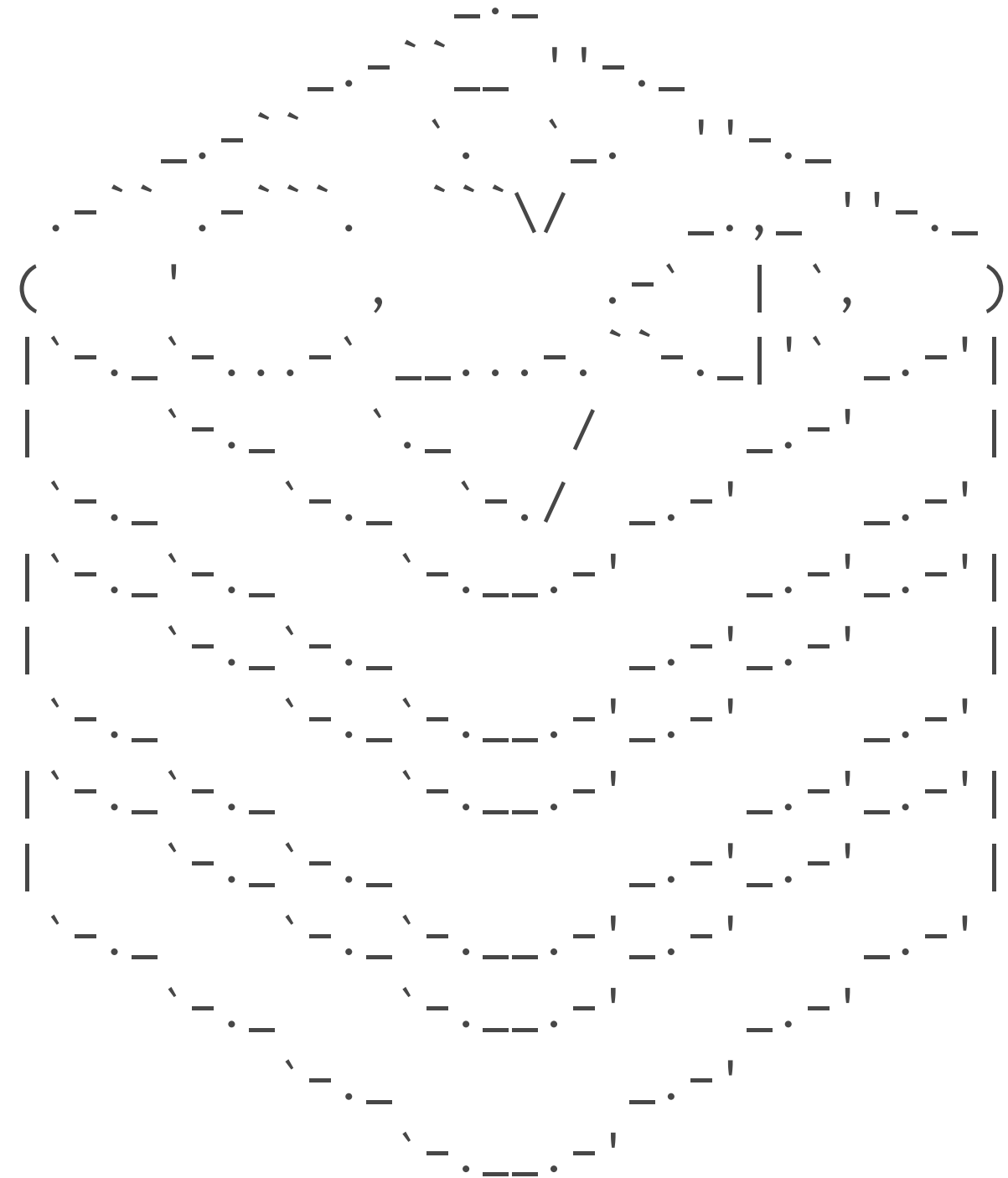
Mount Log Path

```
my-java:
  container_name: my-java
  hostname: my-java
  build: ${PWD}/config/my-java
  networks: ['stack']
  command: java -jar my-java.jar
  volumes:
    - ./logs/my-java/:/opt/my-java/logs/
```

```
filebeat:
  container_name: filebeat
  hostname: filebeat
  image: "docker.elastic.co/beats/filebeat:${ELASTIC_VERSION}"
  volumes:
    - ./logs/my-java/:/var/log/my-java/
    - ./docker-compose/filebeat.yml:/usr/share/filebeat/filebeat.yml:ro
  command: filebeat -e
  networks: ['stack']
```

Registry File

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



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+[[\\-`('.|_]]"] # Drop asciiart lines
```

Infrastructure UI

Pro: Hot 🍌

Con: Complexity

Orchestrate





kubernetes

Where to put Filebeat?

DaemonSet

Metadata

Either in cluster or not

processors:

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

Metadata

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


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+[\\-`('.|_]" # Drop asciart lines
```

Customize Indices

```
output.elasticsearch:
```

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

Pro: Hot 🍌🍌🍌

Con: Complexity++

Moar



Architecture



Index lifecycle management



Select or create a policy

An index lifecycle policy is a blueprint for transitioning your data over time. You can create a new policy or edit an existing policy and save it with a new name.

Existing policies

my_policy5

▼

Create new policy

Edit policy my_policy5

Configure the phases of your data and when to transition between them.

Hot phase

This phase is required. Your index is being queried and actively written to. You can optimize this phase for write throughput.



Enable rollover

If true, rollover the index when it gets too big or too old. The alias switches to the new index. [Learn more](#)

Maximum index size

3

gigabytes

▼

Maximum age

days

▼

Warm phase

Your index becomes read-only when it enters the warm phase. You can optimize this phase for search.

Remove warm phase

Rollover configuration

☐  Move to warm phase on rollover

Move to warm phase after

0



days

Where would you like to allocate these indices?

warm node:true (1)

[View node details](#)

Number of replicas

[Set to same as hot phase](#)

Shrink

Shrink the index into a new index with fewer primary shards. [Learn more](#)

☒  Shrink index

Number of primary shards

[Set to same as hot phase](#)

Force merge

Reduce the number of segments in your shard by merging smaller files and clearing deleted ones. [Learn more](#)

☐  Force merge data

Cold phase

Your index is queried less frequently and no longer needs to be on the most performant hardware.

Activate cold phase

Delete phase

Use this phase to define how long to retain your data.

Deactive cold phase

Configuration

Delete indices after

0

days



← Back

Continue →

Frozen Indices

<https://github.com/elastic/elasticsearch/issues/34352>

Centralized Logstash & Beats Management

Conclusion

Examples

<https://github.com/xeraa/java-logging>

Parse 

Send 

Structure 

Containerize 

Orchestrate 

Questions?

Philipp Krenn

@xeraa