ЭВОЛЮЦИЯ СИСТЕМЫ ЛОГИРОВАНИЯ В ECOM.TECH: ОТ ELASTIC STACK К VICTORIALOGS

Валерий
Евдокимов
(PTL Observability)

ecom.tech

ecom.tech

17

(01)

площадок или точек присутствия
в 9 ЦОД в Москве
и Санкт-Петербурге.

240+ стоек.

(02)

4 000+ железных серверов.

(03)

12 000+ виртуальных серверов.

(04)

Собственная сеть
операторского класса.

(05)

30

(06)

кластеров k8s со средними
воркерами — BM 96 CPU 756GB RAM
workload 30–140 pods. Медиана 60.

Чем я занимаюсь в компании?

Я отвечаю за развитие платформы
Observability, а в частности:

Подходы логирования	(01)
Шиппинг	(02)
Трейсинг	(03)
EBPF	(04)

Зачем этот доклад?

Мы несколько лет развиваем подходы к логированию.

(01)

Набрались граблей и опыта.

(02)

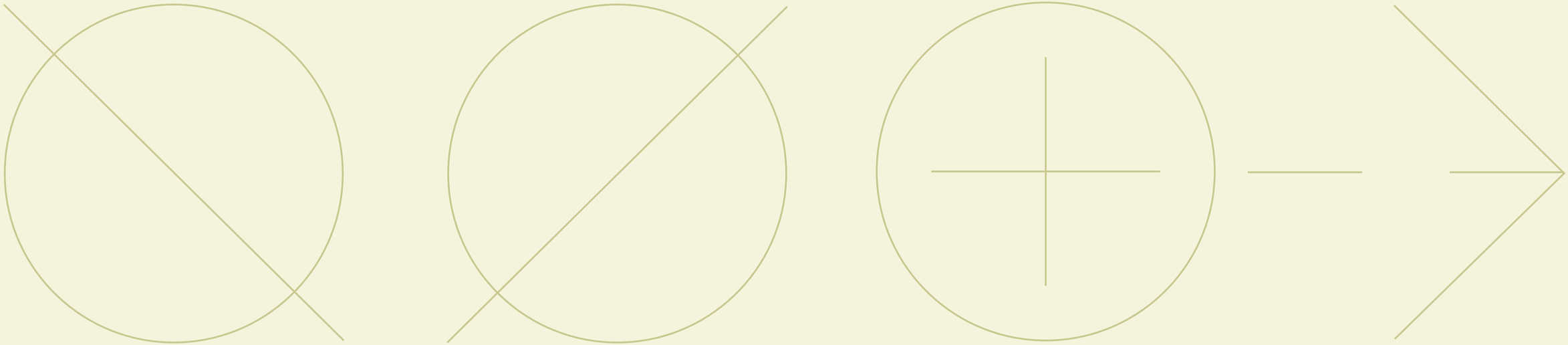
Выработали свой подход, который позволил дешевле хранить логи и меньше страдать.

(03)

Источники изображений в презентации: https://disk.yandex.ru/i/6bTGIKdqi6tVWA

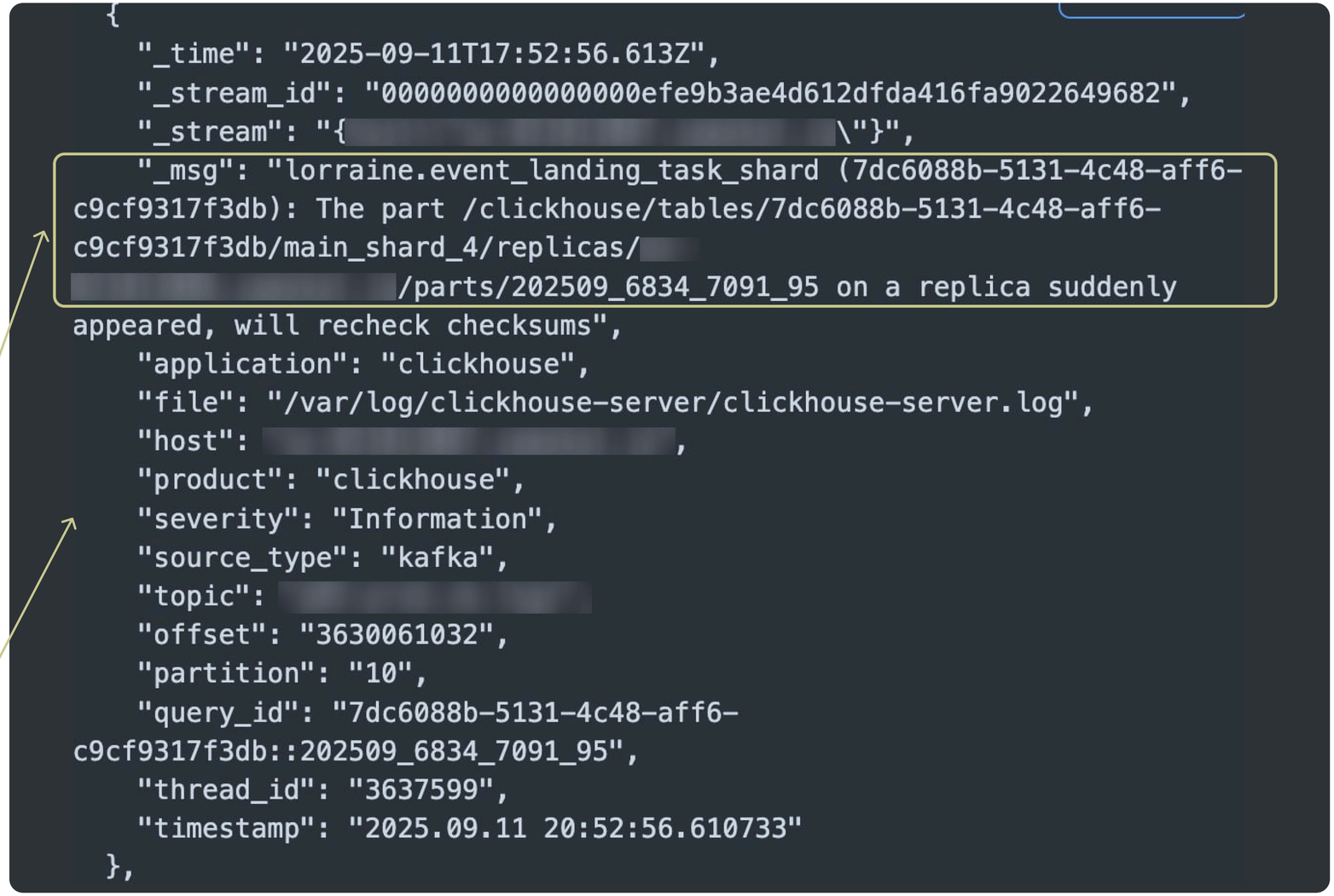
ТЕОРЕТИЧЕСКАЯ БАЗА И КОНТЕКСТ

00



Что для нас логирование?

Логирование — запись состояния информационной системы в документы с заранее определенной структурой.



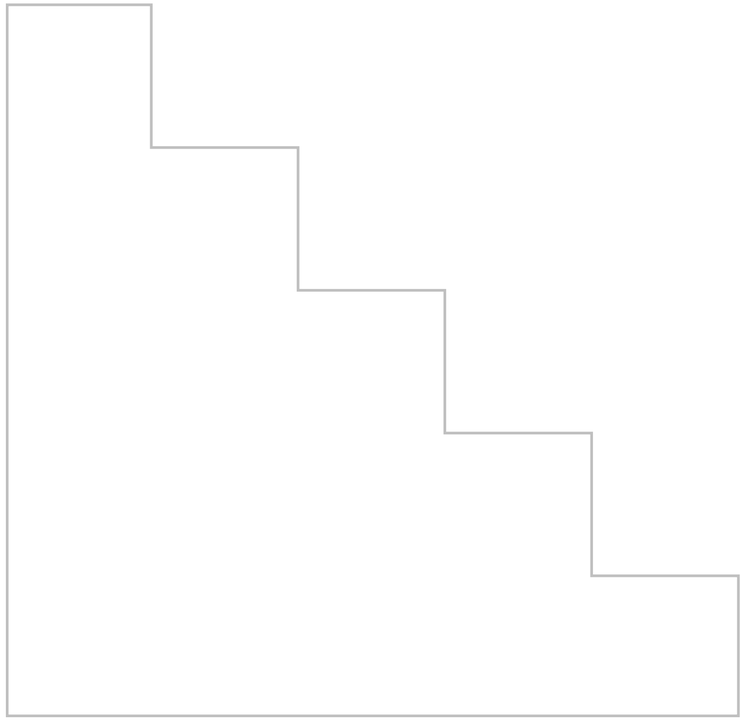
Информация

Мета
(всё остальное)

Ещё уточним контекст

Логи, о которых мы говорим — системные сообщения о состоянии и поведении системы.

Они используются для исследования поведения систем, в частности — при дебаге инцидентов.



Нефункциональные требования

(01)

Логи должны быть целостными, недопустимо терять их часть по пути.

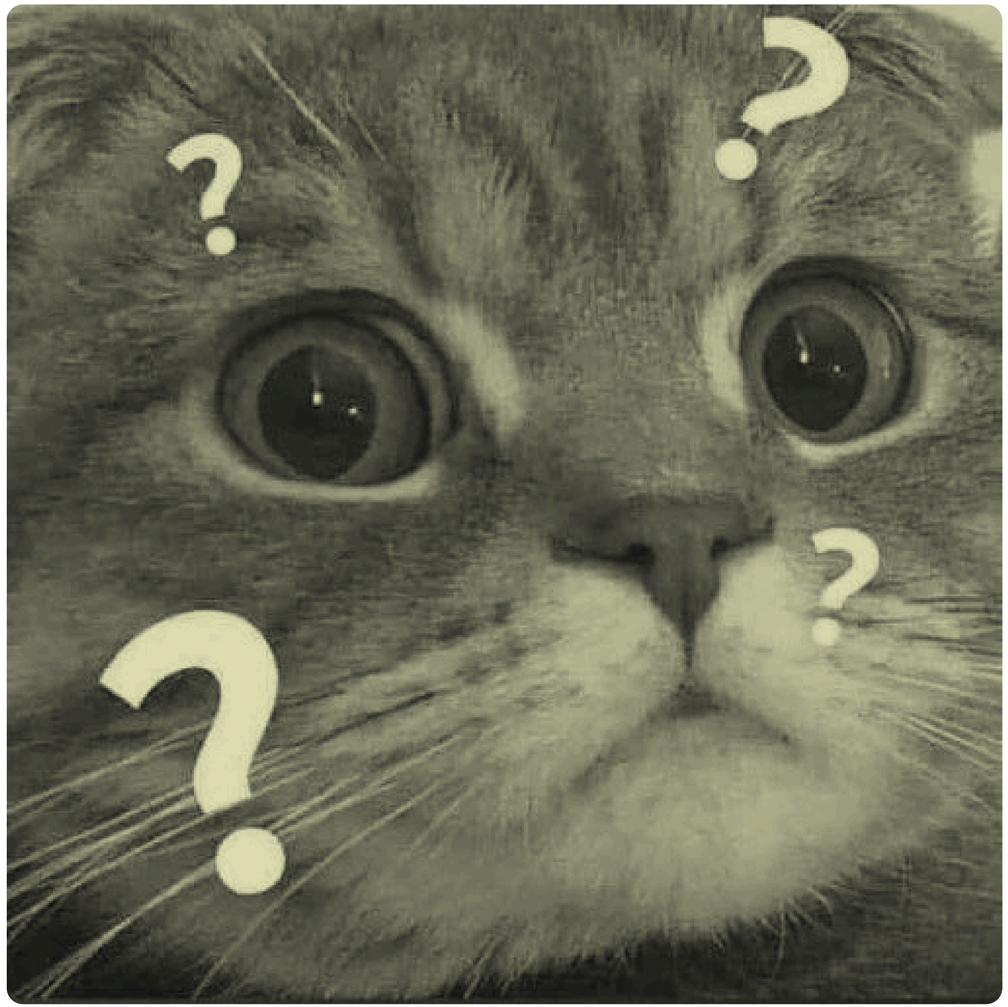
(02)

Логи должны доставляться в хранилища с минимальным лагом.

(03)

Они должны храниться от нескольких дней (а лучше недель), чтобы можно было провести исследование прошлой проблемы.

Итак, моя система может писать логи…
И что?



Как их доставить?

Где их хранить?

Как их визуализировать?

Варианты доставки логов

(01)

Писать из приложения напрямую по http\tcp.

(02)

Писать в файлы и вычитывать их агентом.

Агенты сбора логов

Beats (Filebeat, Winlogbeat etc.)

Fluentd

Vector

Варианты хранилищ для логов

Elasticsearch
(Opensearch)

Loki

Clickhouse

И что-то новое:

Clickstack

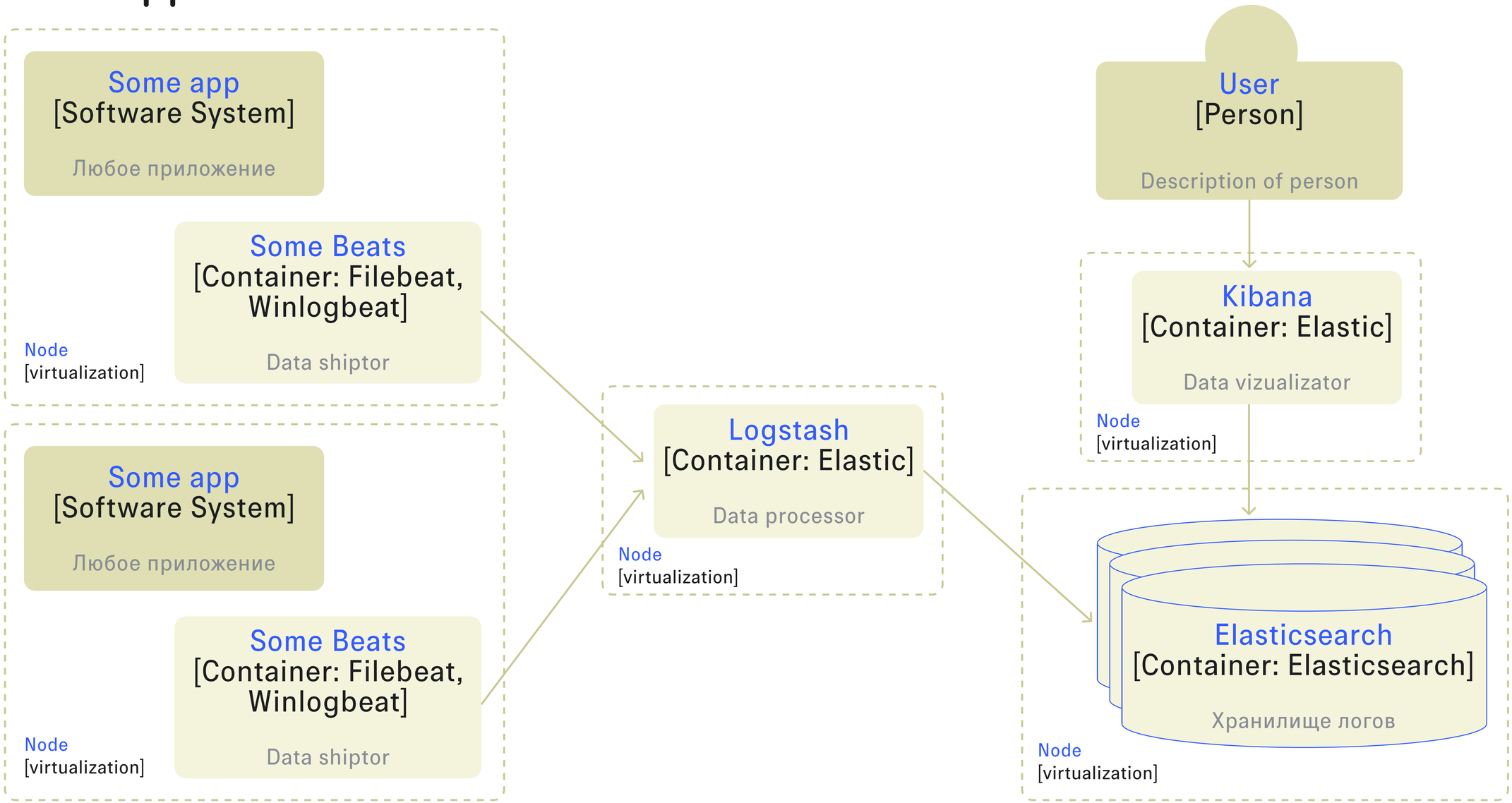
Victorialogs

Смотрим на логи

Доступ к логам нужно разграничивать

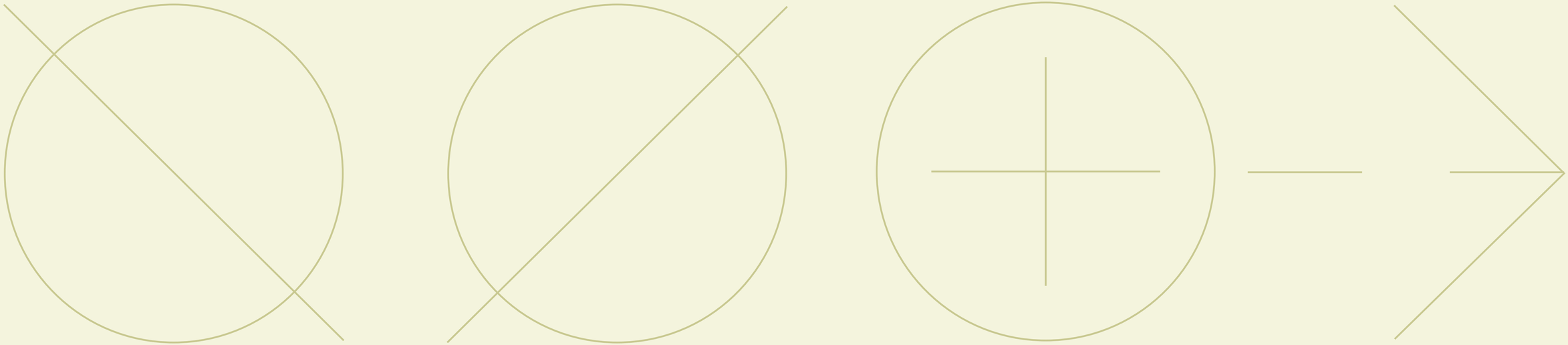
Можем использовать встроенную визуализацию или вести всё в Grafana

Общие тенденции на рынке: несколько лет назад



С ЧЕГО ВСЕ НАЧИНАЛОСЬ В SAMOKAT.TECH?

01



Наши первые шаги

ДАНО

1 инженер (01)

3 независимых ДЦ, в каждом по k8s-кластеру (02)

6 языков разработки с разными конфигурациями логирования (03)

~250 продуктов, с которых нужно собрать логи (04)

НУЖНО:

Собрать логи со всех приложений в k8s (01)

Собрать логи ОС и сервисов со всех виртуальных и «железных» нод (02)

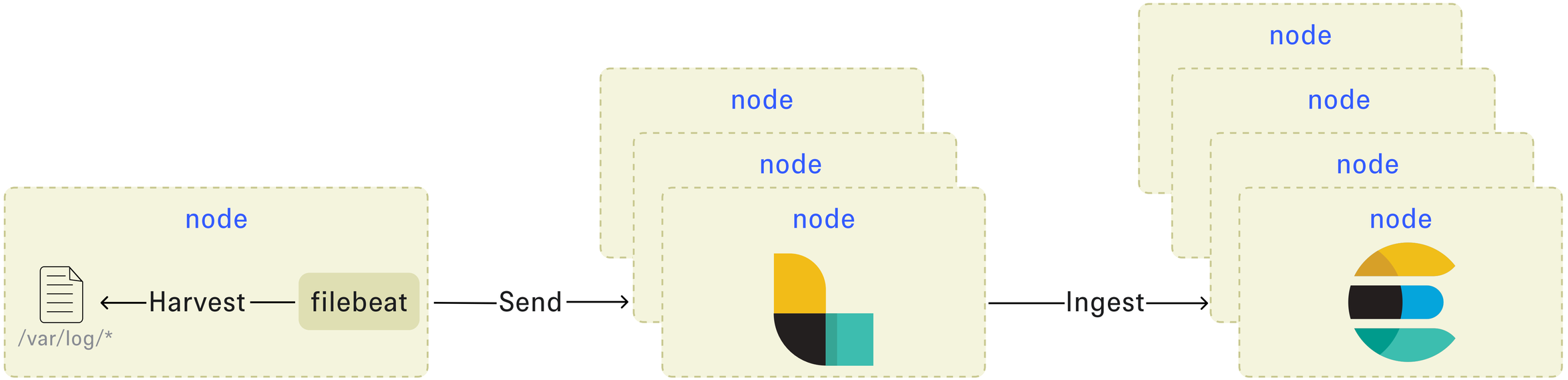
Собираем логи

Filebeat устанавливали

— через Ansible на ноды
— Daemonset в k8s

Logstash на виртуалках

Elasticsearch на baremetal



Преобразуем на Logstash

Состав полей
в логах разный,
их нужно приводить
в общий вид.

Под каждый вид логов — свой порт.
Под каждый порт — свой пайплайн.
Каждому пайплайну — свой index-pattern в Elasticsearch.

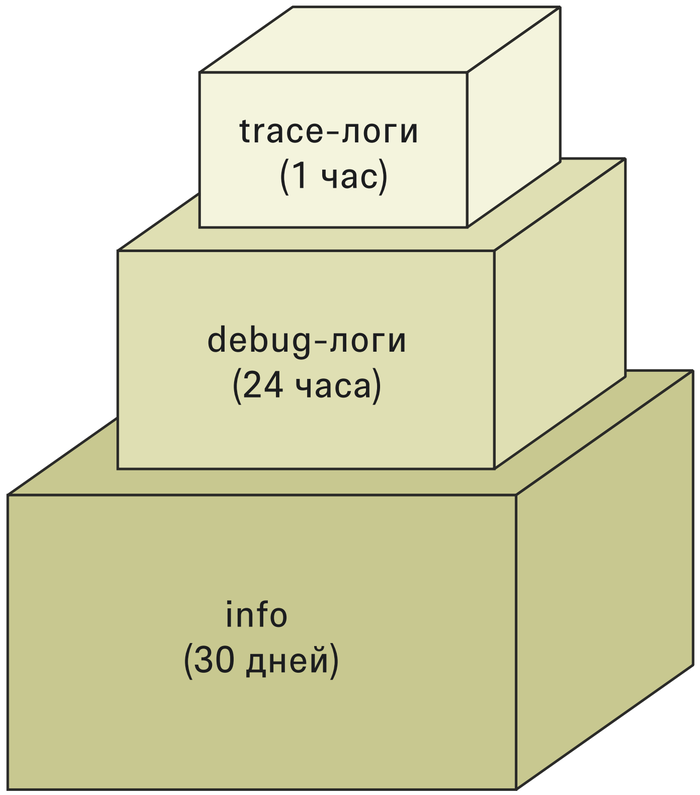

```
}

mutate {
  replace => {
    "[        ][namespace]" => "%{[kubernetes][namespace]}"
    "[        ][host]" => "%{[kubernetes][node][hostname]}"
    "[        ][k8s][container_name]" => "%{[kubernetes][container][name]}
    "[        ][k8s][pod_name]" => "%{[kubernetes][pod][name]}"
    "[        ][k8s][node_type]" => "%{[kubernetes][node][labels][nodeType
    "[        ][k8s][pod_ip]" => "%{[kubernetes][pod][ip]}"
  }
}

if [kubernetes][replicaset][name] {
  mutate {
    replace => {
      "[        ][k8s][workload_type]" => "replicaset"
      "[        ][k8s][workload_name]" => "%{[kubernetes][replicaset][name
    }
  }
}
```

Преобразуем на Logstash

Под каждый сервис — свой index-pattern,
tempate с персональным маппингом.
Для каждого из 299...300...301 сервисов...

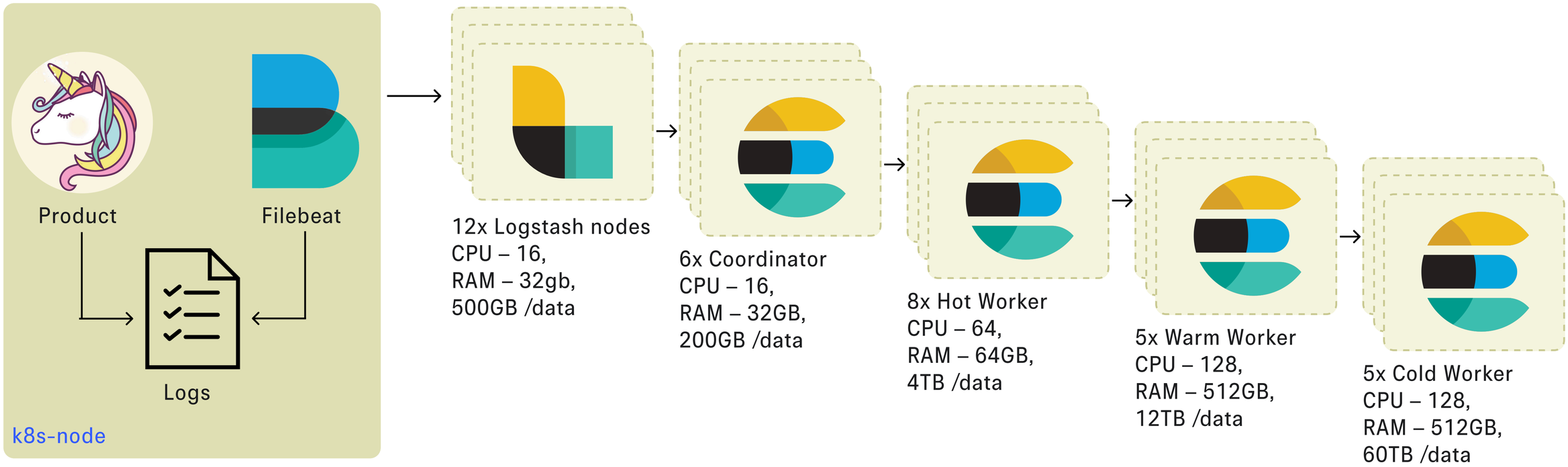



```
"python.args": {
    "type": "text"
},
"python.exception": {
    "type": "text"
},
"python.scope.client": {
    "type": "text"
},
"python.scope.headers": {
    "type": "text"
},
"python.scope.server": {
    "type": "text"
},
"python.request_time_frac": {
    "type": "float"
}
```

```
"json.client_ip": {
    "type": "ip"
},
"json.real_ip": {
    "type": "ip"
},
"json.conn_rate": {
    "type": "integer"
},
"json.http_req_rate": {
    "type": "integer"
},
"json.http_req_rate_by_path": {
    "type": "integer"
},
"json.request_active_time": {
    "type": "integer"
},
"json.request_idle_time": {
    "type": "integer"
},
"json.request_time": {
    "type": "integer"
},
"json.response_time": {
    "type": "integer"
},
```

Собираем логи

ver 7.14.2 basic license



И всё бы ок, но…

Logstash страдал из-за зоопарка логов

[ERROR]
[logstash.filters.grok] Grok
parse failure
{:message=>"Unrecognized
data", :pattern=>"
%{COMBINEDAPACHELOG}"}

(01)

[ERROR]
[logstash.filters.date] Failed
parsing date from field

(02)

[ERROR]
[logstash.filters.json] JSON
parse failure
{:source=>"message"}

(03)

И всё бы ок, но…

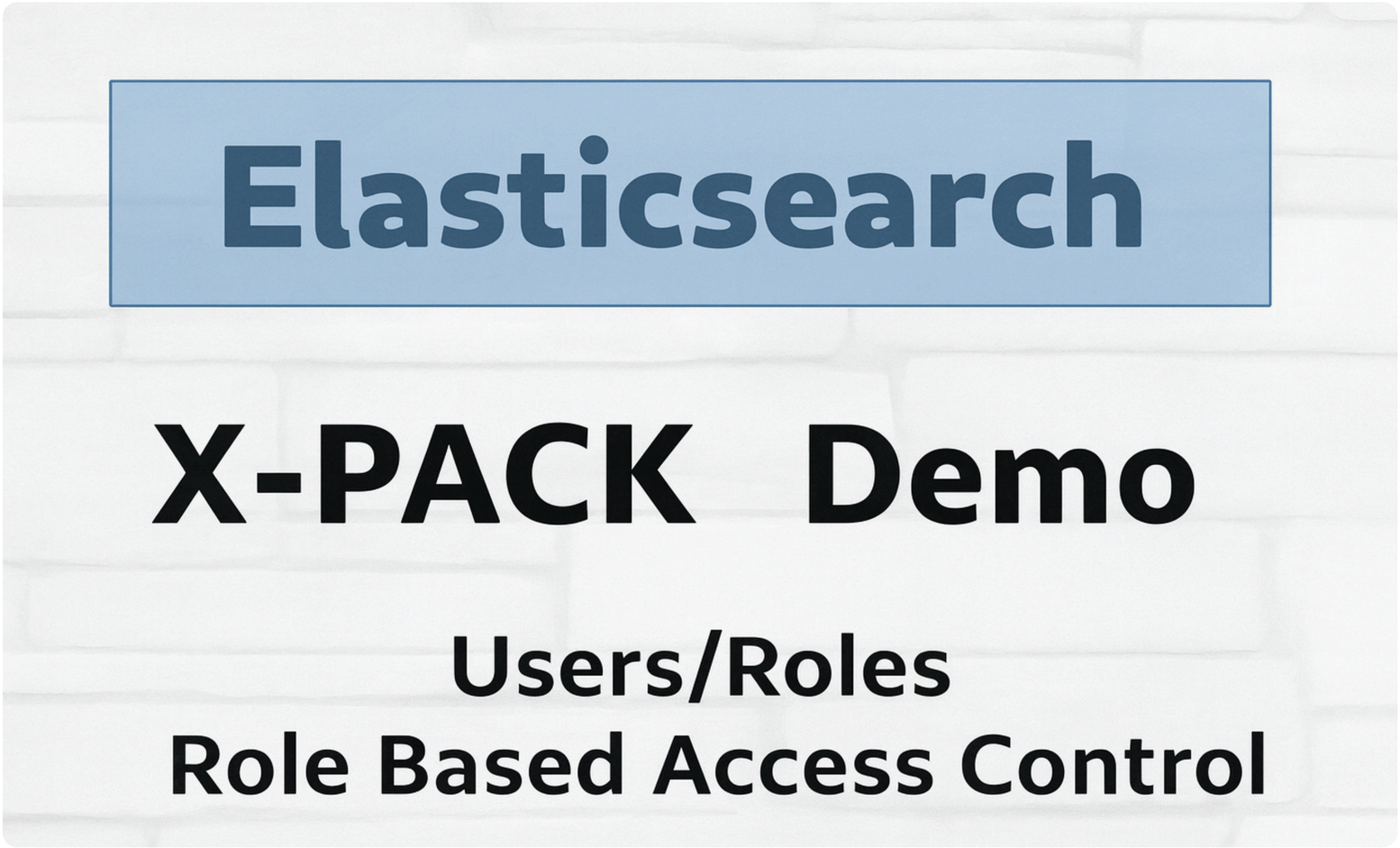
Архитектура подводила

При падении и последующем восстановлении warm/cold-worker ноды возникала нагрузка на IOPS. Вышедшая из строя нода могла не проходить health-check и бесконечно реинитить шарды: возникал Recovery Storm.

И всё бы ок, но…

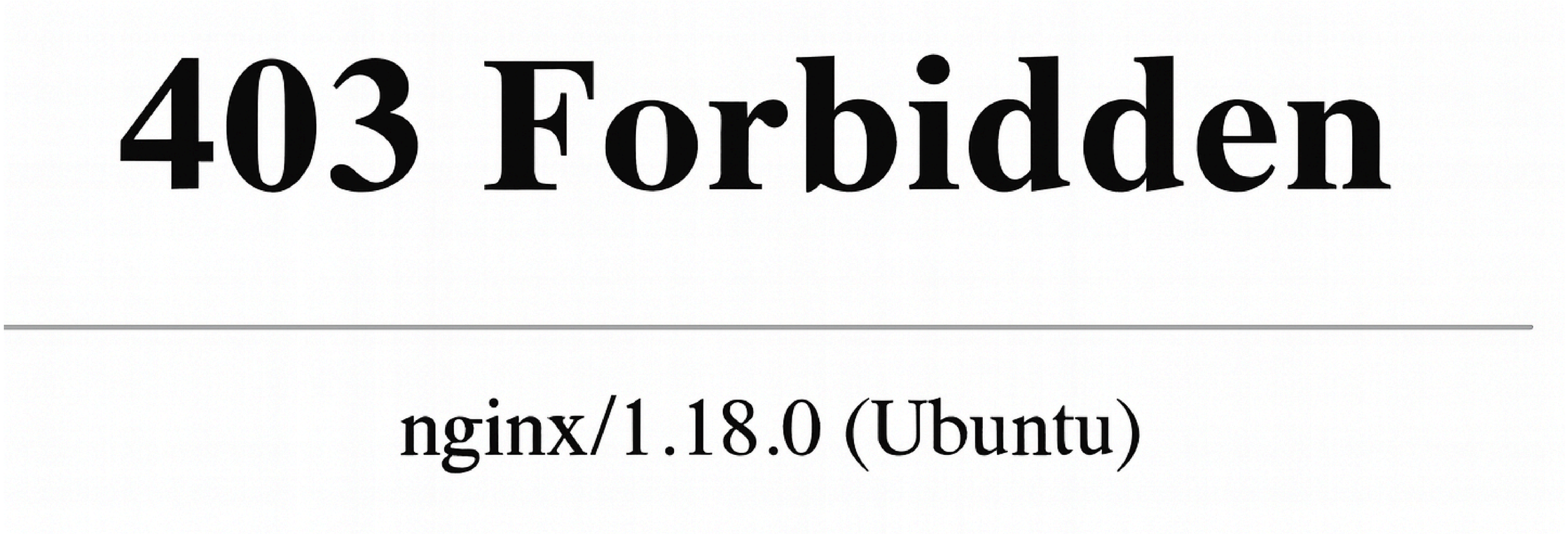
В basic license Kibana больно разграничивать досмуп

Все удобства были сведены в Xpack, который поставлялся под Elastic License.



И всё бы ок, но…

Обновление пакетов требует использования обходных решений



А теперь поподробнее

Что там с лицензией?

Recent Elastic.co licensing change announcment

General Feedback

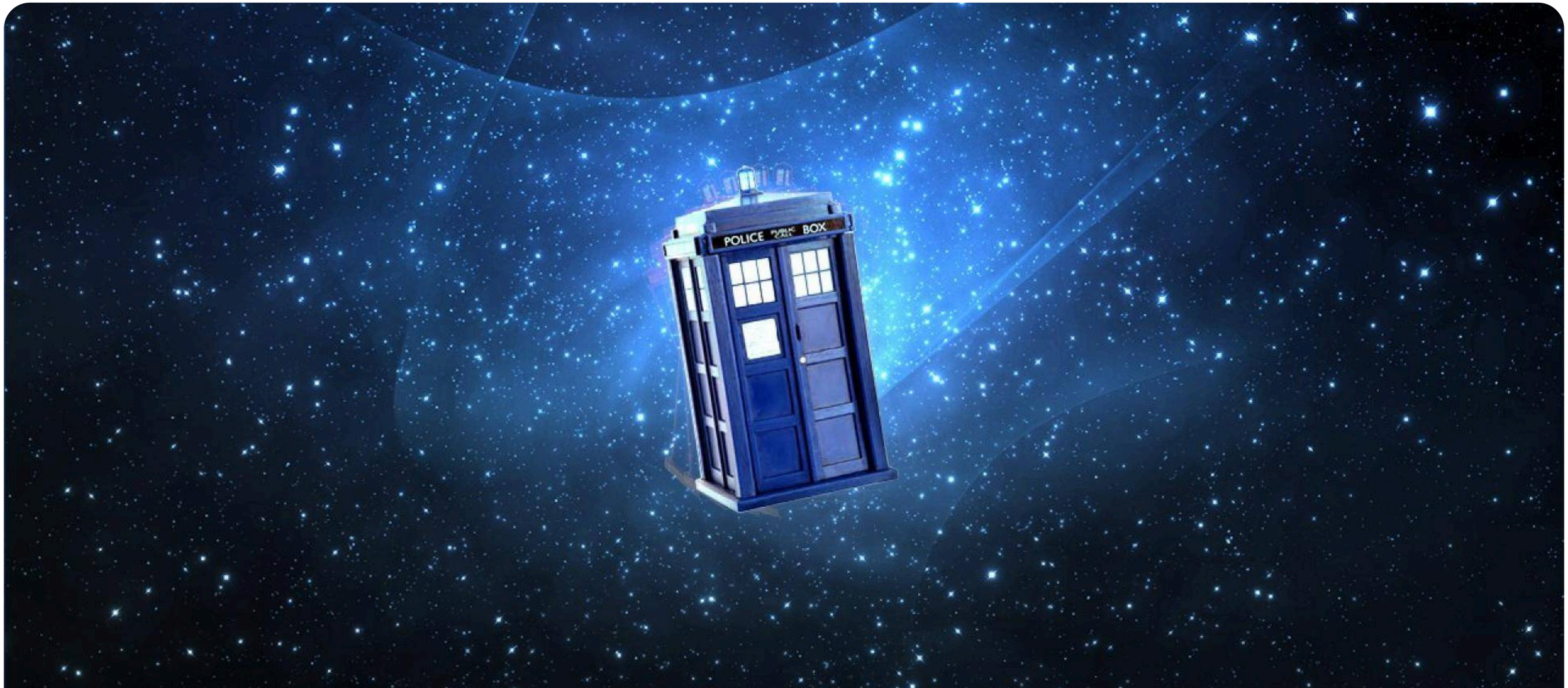
tlacuache 1 Jan 2021

I'd like to see a blog post about this announcement 113 from Elastic.co 7 switching to SSPL and how it will affect ODFE going forward.

Thanks!

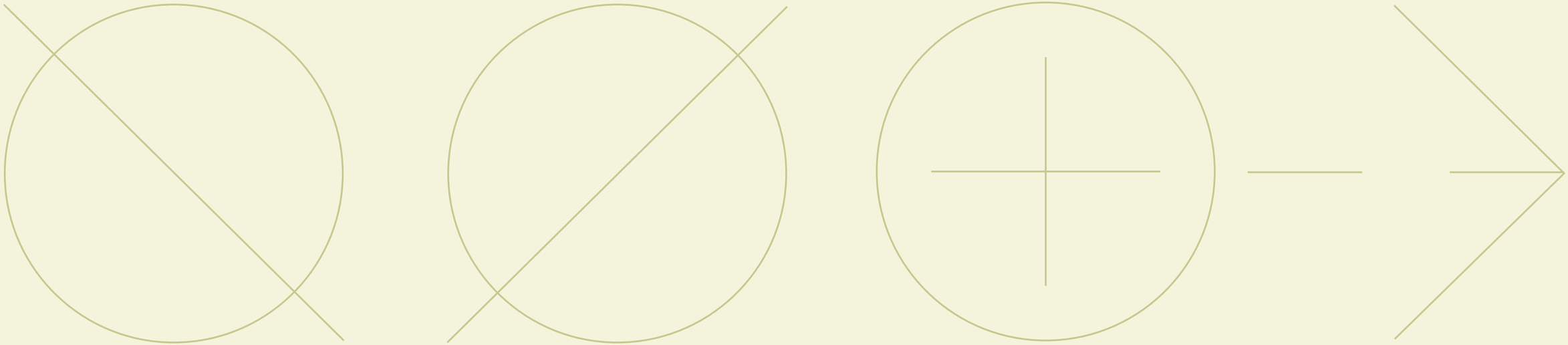
We changed the **Apache** 2.0-licensed source code of Elasticsearch and Kibana to be dual licensed under SSPL 1.0 and the **Elastic** License 2.0 (**ELv2**), giving users the choice of which of the two licenses to apply.

Перемотаем время на полгода вперёд



ПЕРЕПРЫГНЕМ НА OPENSEARCH

02



Отходим от продуктов Elastic

ДАНО

2 инженера (01)

8 независимых ДЦ, в каждом по k8s-кластеру (02)

8 языков разработки с разными конфигурациями логирования (03)

~500 продуктов, с которых нужно собрать логи (04)

НУЖНО:

Собрать логи со всех приложений в k8s (01)

Собрать логи ОС и сервисов со всех виртуальных и «железных» нод (02)

Перейти на opensource без ограничений лицензии без даунтайма (03)

Зачем? Что мы получим?
~~Чтобы что?~~

Можем обновляться, закрывая CVE (Elasticsearch остановился в basic на версии 7.14.2).

Добавляем OIDC (Keycloak).

Что в планах поменять

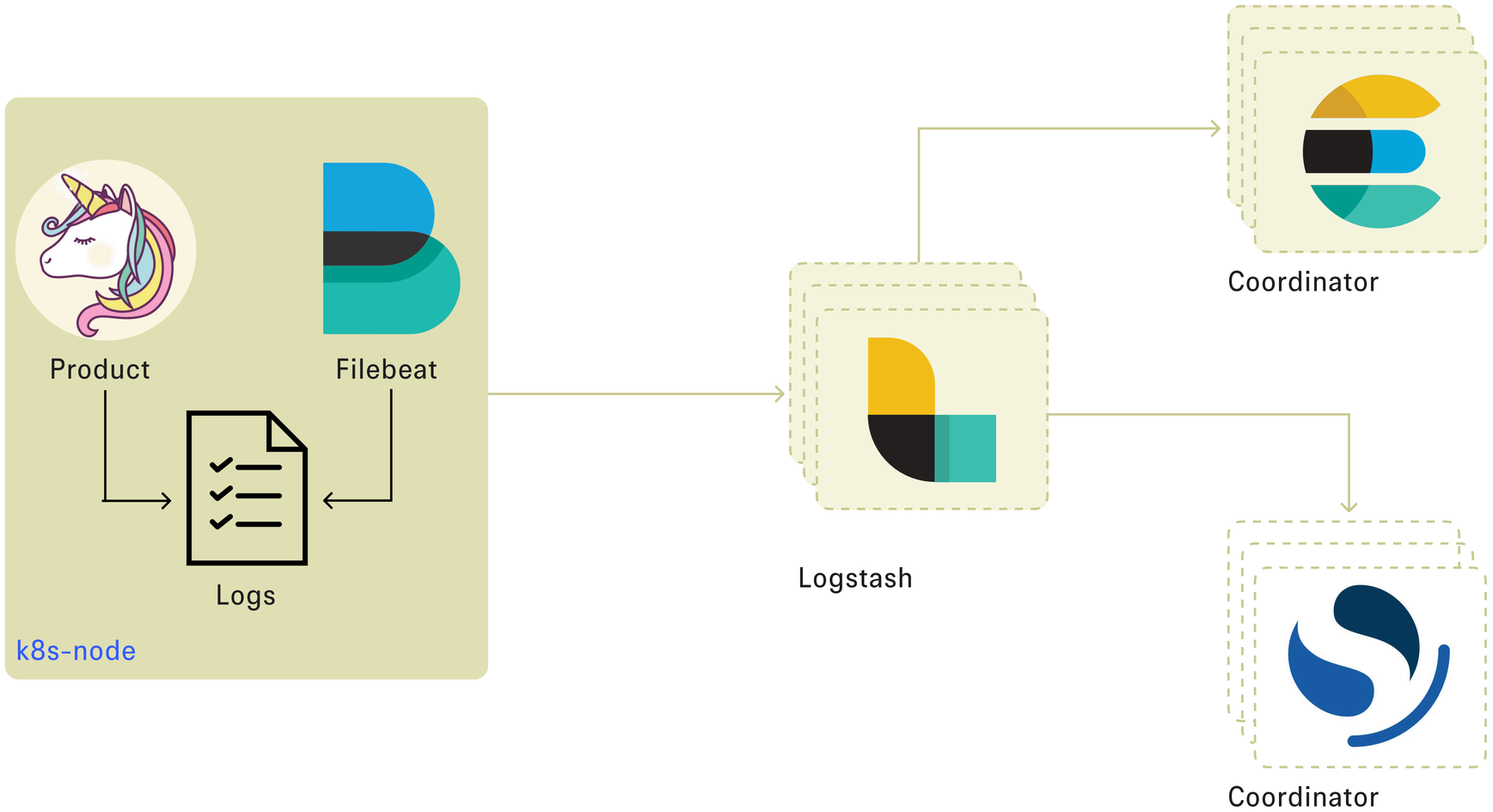
Elasticsearch → Opensearch

Filebeat → Vector

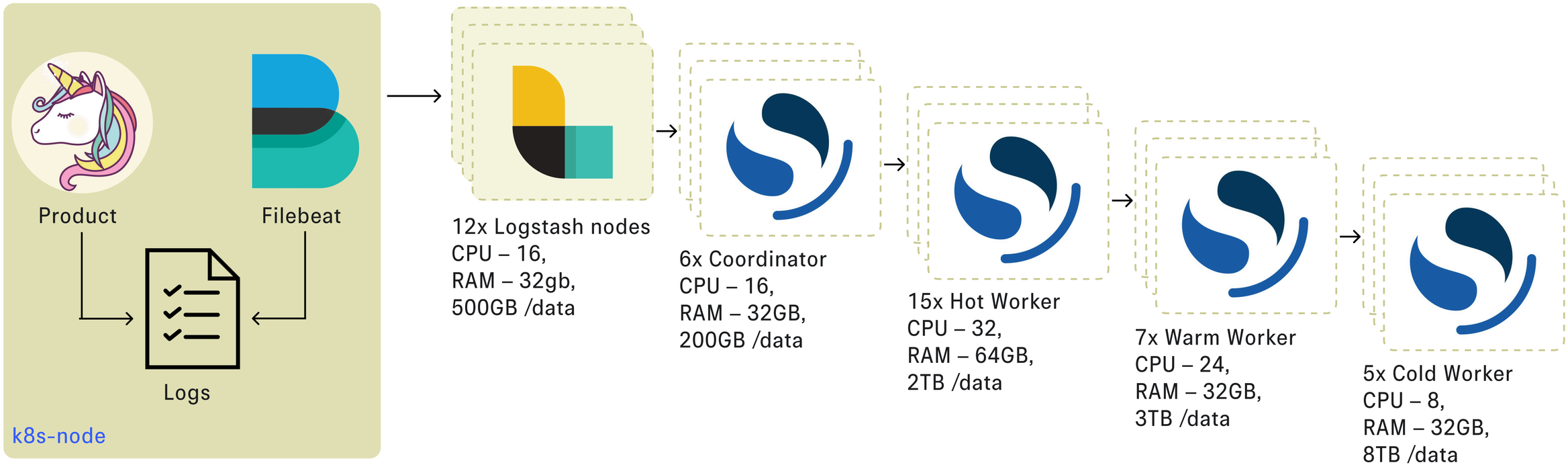
Logstash → Хоть на что-то

(а ещё — архитектуру кластера под логи)

Пилотили через параллельную запись



Переезжаем на Opensearch



Logstash в нашей схеме

Logstash в нашей схеме —
это инструмент предобработки
логов перед Elasticsearch.

Его задачи в потоке логов:

Централизация

(01)

Стандартизация

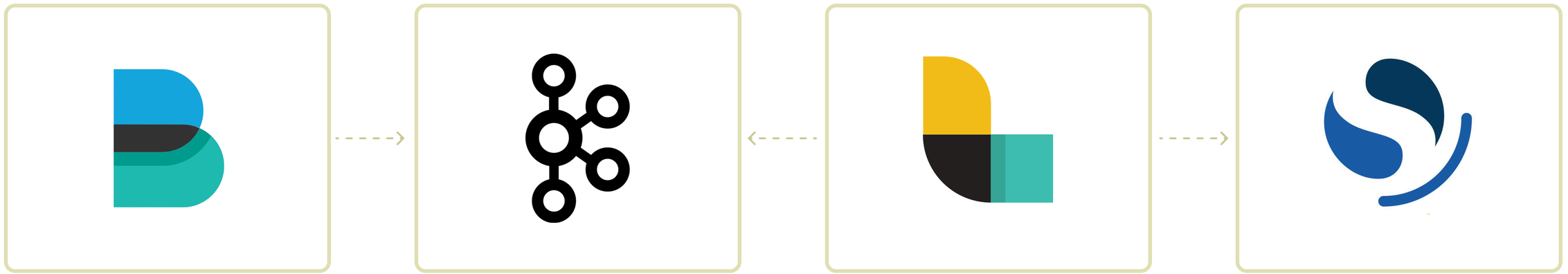
(02)

Подавление всплесков
на шиппинге

(01)

Нам нужно повысить гарантии доставки. Kafka

Решили обезопаситься от всплесков логов и добавить буферизацию до Logstash, внедрив Kafka



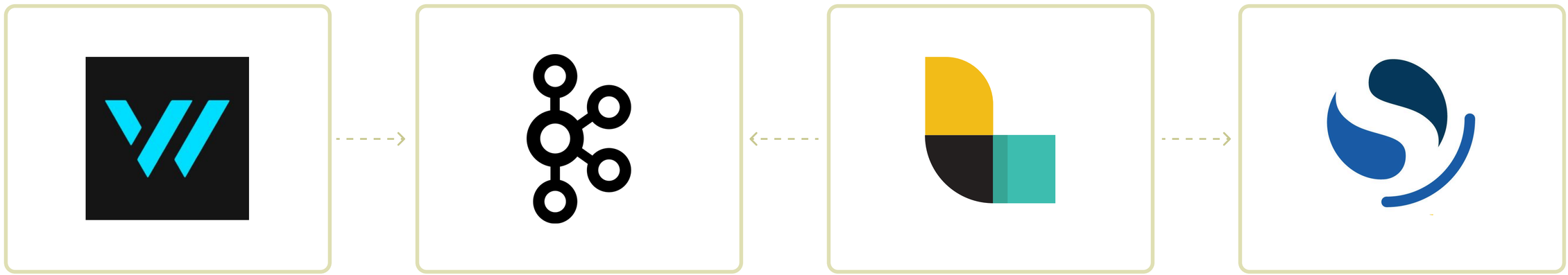
Меняем агент для сбора логов

Выбрали Vector.dev

	Filebeat	Vector
Лицензия	Elastic License 2.0	MPL-2.0
Минимальное потребление памяти	42MB	5MB
Поддержка Opensearch	Нет	Да
Производительность	«File to TCP» — 7.8 MiB/s, «TCP to TCP» — 5 MiB/s.	"File to TCP" — 76.7 MiB/s, "TCP to TCP" — 69.9 MiB/s

https://betterstack.com/community/guides/logging/log-shippers-explained/

Заменяем Filebeat на Vector



Чем нас не устраивал Logstash?

Ноды время
от времени падали,
чинить необходимо
было руками.

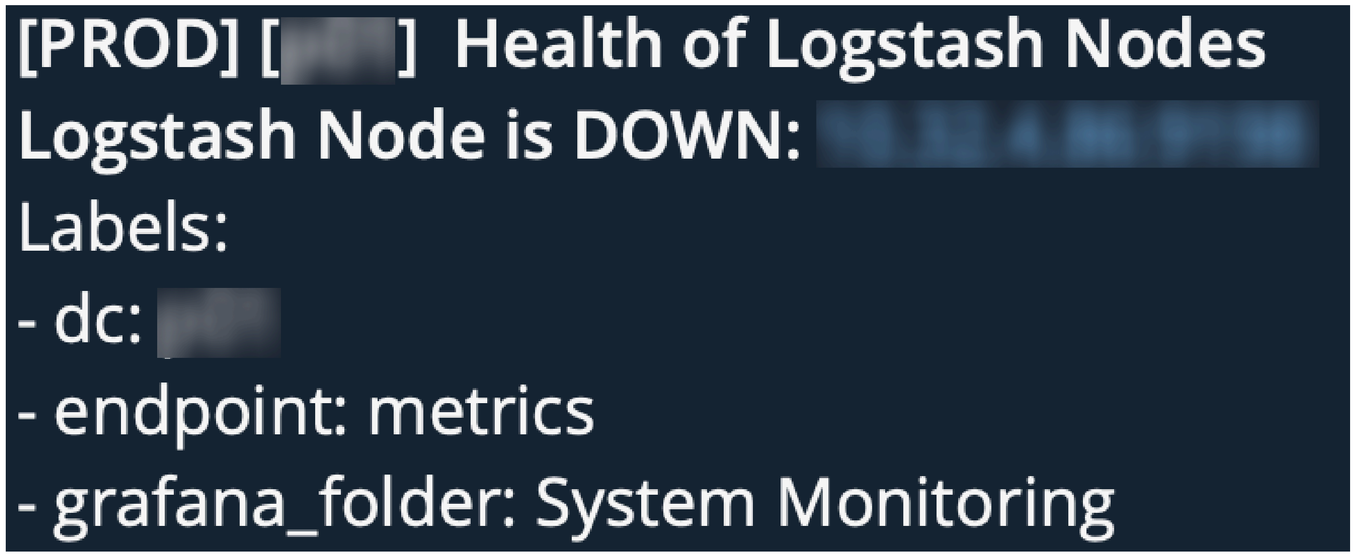
(01)

Он расходовал
немало ресурсов.

(02)

Мы теряли логи.

(03)



Больше источников логов — чаще проблемы на Logstash

Логи полезны, но только при сохранении их целостности.

Если вы разбираете инцидент и часть данных отсутствует, то такие логи практически бесполезны.

Пример ситуации:
Команда добавила поле со странной кодировкой.

(01)

Фильтр

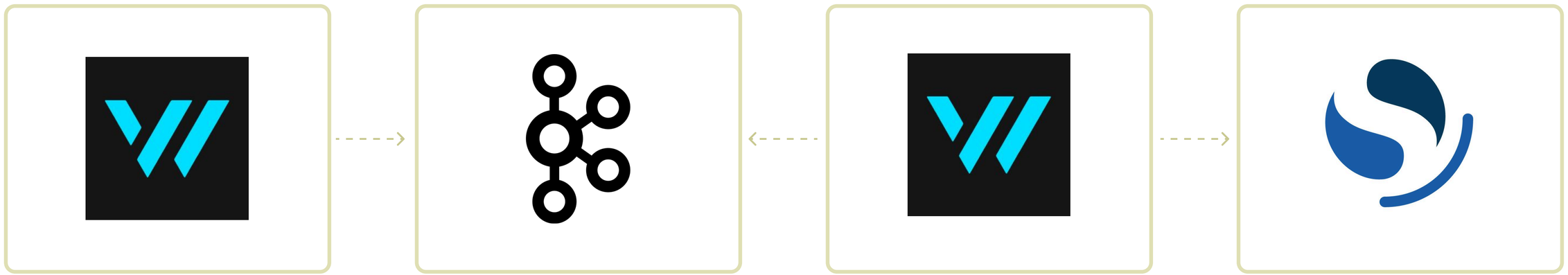
`json → exception → DLQ`

растёт. Пока разбирались, коллеги из другой команды не видели свои логи — общая очередь была полностью загружена.

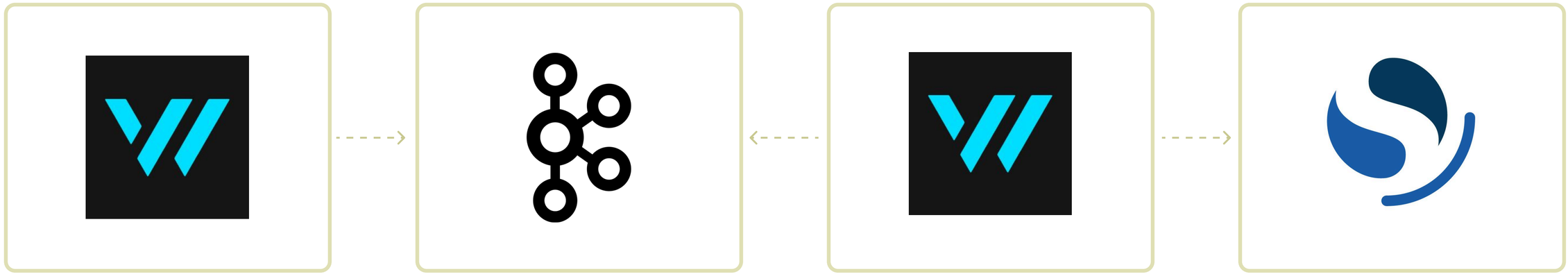
(01)

Заменим его… на ещё один Vector?

Встраиваем после Kafka Vector для доставки в Opensearch.



Как оно будет работать?



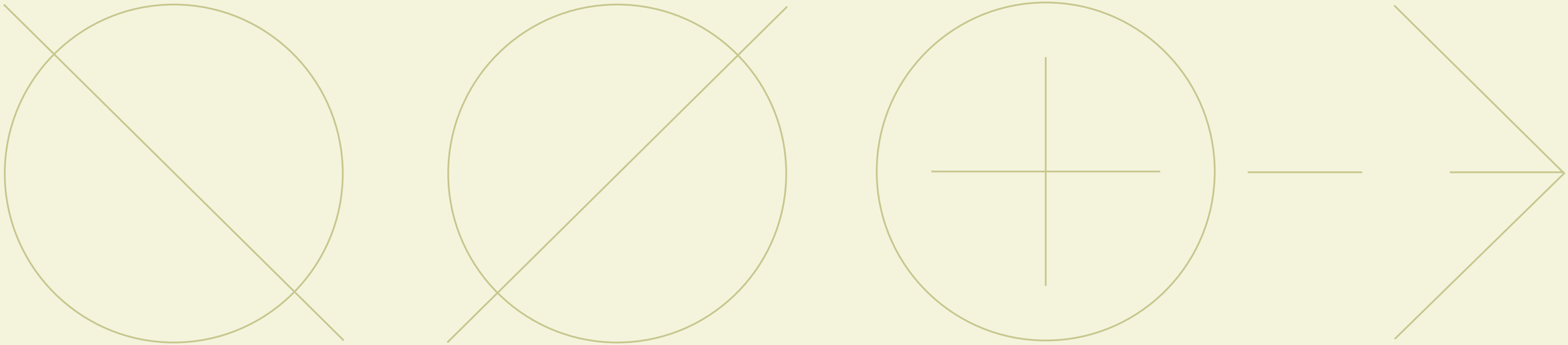
Мета-информация добавляется на уровне Vector-агент

При потере коннекта к Opensearch логи будут храниться в реплицированном Kafka-кластере

Обработка логов — Vector Aggregator

СВЕТЛОЕ БУДУЩЕЕ?

03



Отходим от продуктов Elastic

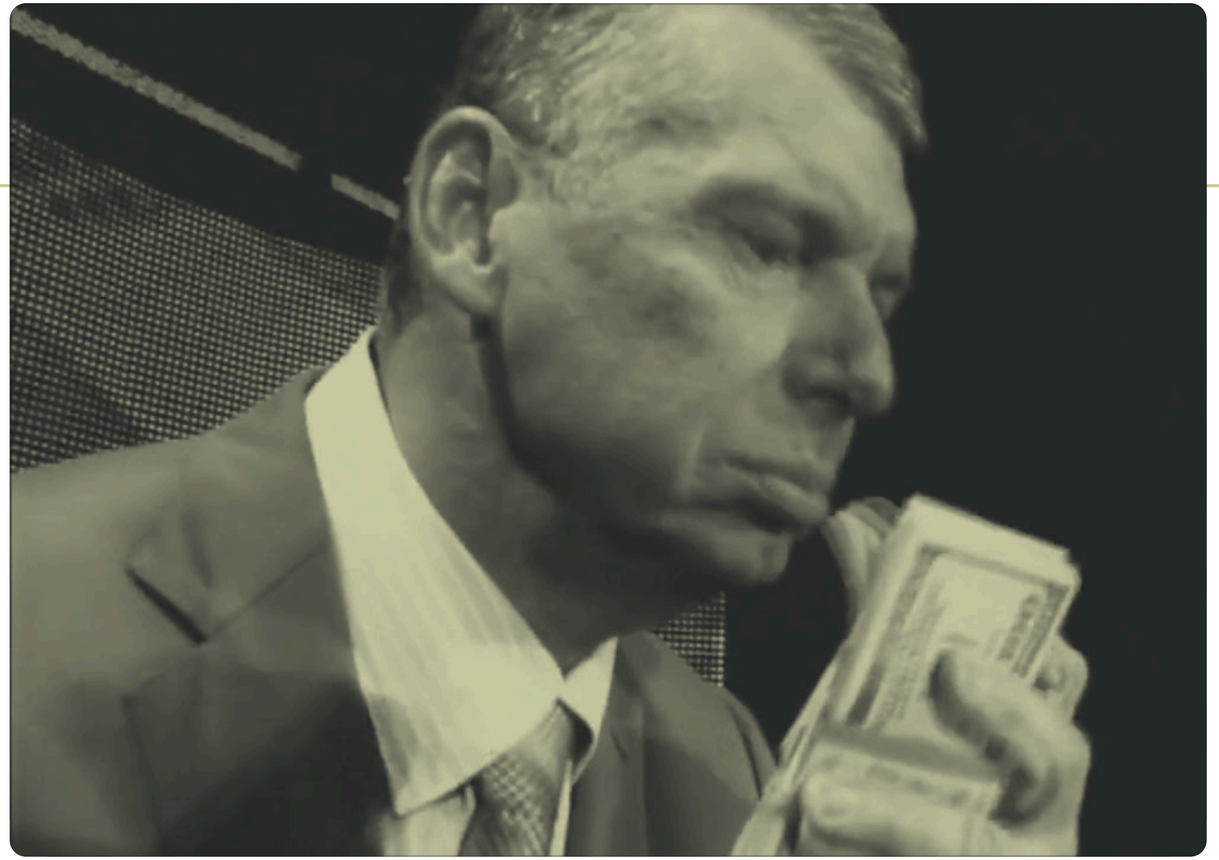
ДАНО

2 инженера	(01)
15 независимых ДЦ, в каждом — по k8s-кластеру	(02)
7 языков разработки с разными конфигурациями логирования	(03)
Логи собираются с 650+ продуктов (k8s, compose, onpremise-сервисы)	(04)
Логи собираются со всех виртуальных и «железных» нод	(05)

НУЖНО:

Побороть потери логов и остановку индексации в кластерах логов	(01)
Упростить поддержку доставки и хранения логов	(02)

А ещё нужно сократить расходы на логи



Что там с инжестингом?

Строгая типизация карала, если кто-то записал свой лог нестандартно.

(01)

Ох, классика жанра:

«Вчера поле было строкой, сегодня стало числом».

(02)

Для каждого продукта свои темплейты, которые надо поддерживать.

(03)

[redacted] 12:48

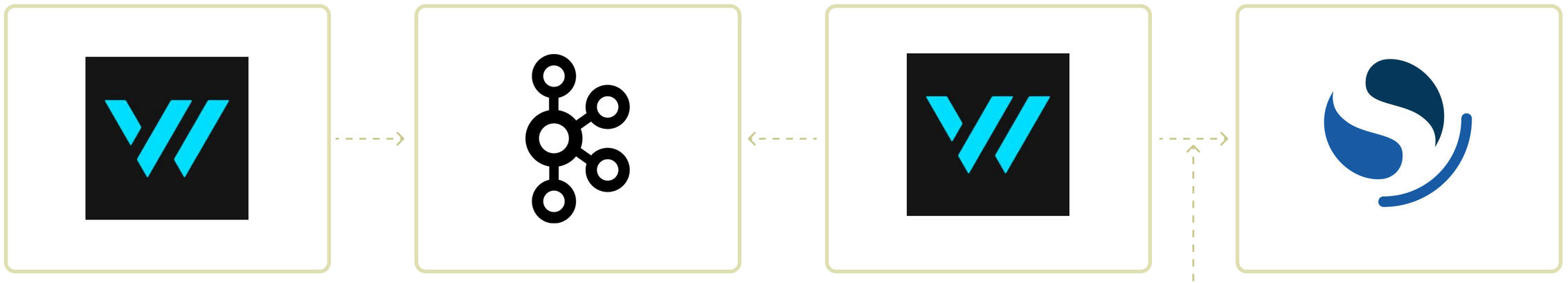
Привет! На проде пропали логи в индексе chatbot

[redacted]/app/discover#/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-15m,to:now))&_a=(columns:!(),filters:!(),index:'*chatbot-*',interval:auto,query:(language:kuery,query:''),sort:!(!('@timestamp',desc)))

Сейчас раскатили релиз, очень нужны логи, можете посмотреть?

Из-за чего это происходило?

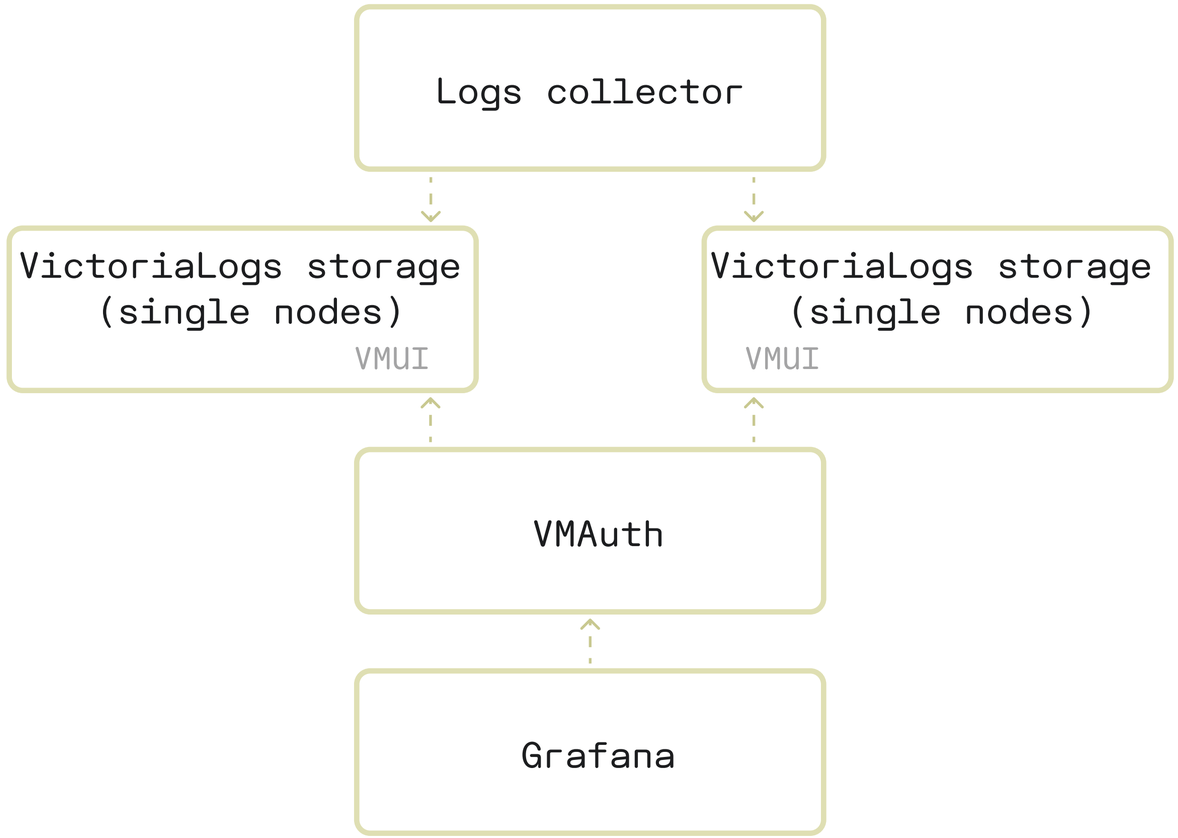
Sinc в Vector-aggregator мог залипнуть из-за расхождения лога со схемой в Opensearch.



Застрял один индекс — retry, остальные логи — подождут.

Во мраке пещер гита мы нашли VictoriaLogs

Средство хранения и визуализации логов от создателей VictoriaMetrics



Чем решение нам понравилось?

Работает со всеми типами логов.

(01)

Нет необходимости жесткой типизации данных.

(02)

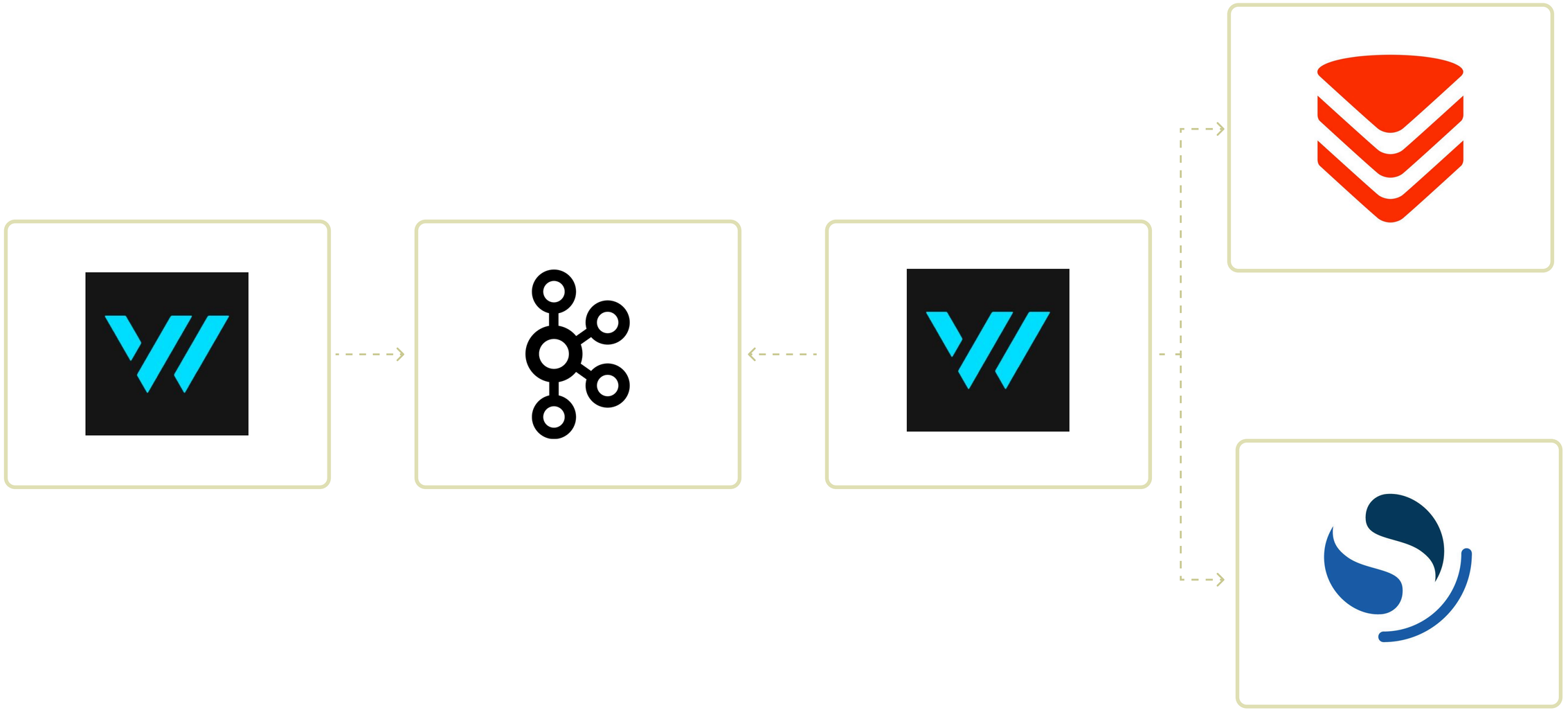
Автоматически парсит входящие логи (пиши что хочешь — распарсим!).

(03)

Самое главное: на тот же объём логов (по времени) требуется в 10 раз меньше места (если верить разработчикам).

(04)

Попилотим решение



Результаты пилота на примере одного кластера

Тесты показали, что мы можем сократить ресурсы до 20 раз, заменив кластера Opensearch парой baremetal-нод с Victorialogs.

(01)

Разница в дисковом пространстве ~ в 14 раз меньше в пользу VLogs.

(02)

Отказоустойчивость записи обеспечивается параллельной записью из Vector.

(03)

Отказоустойчивость чтения обеспечивается параллельным чтением через VMAuth.

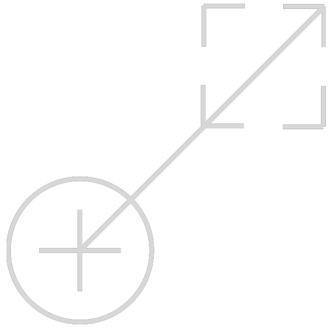
(04)

Мы получили более гибкий инжестинг на VLogs

Стали использовать NDJSON (/insert/jsonline?)

Пропуск invalid строк без остановки всего процесса (мы не смогли настолько сломать логи).

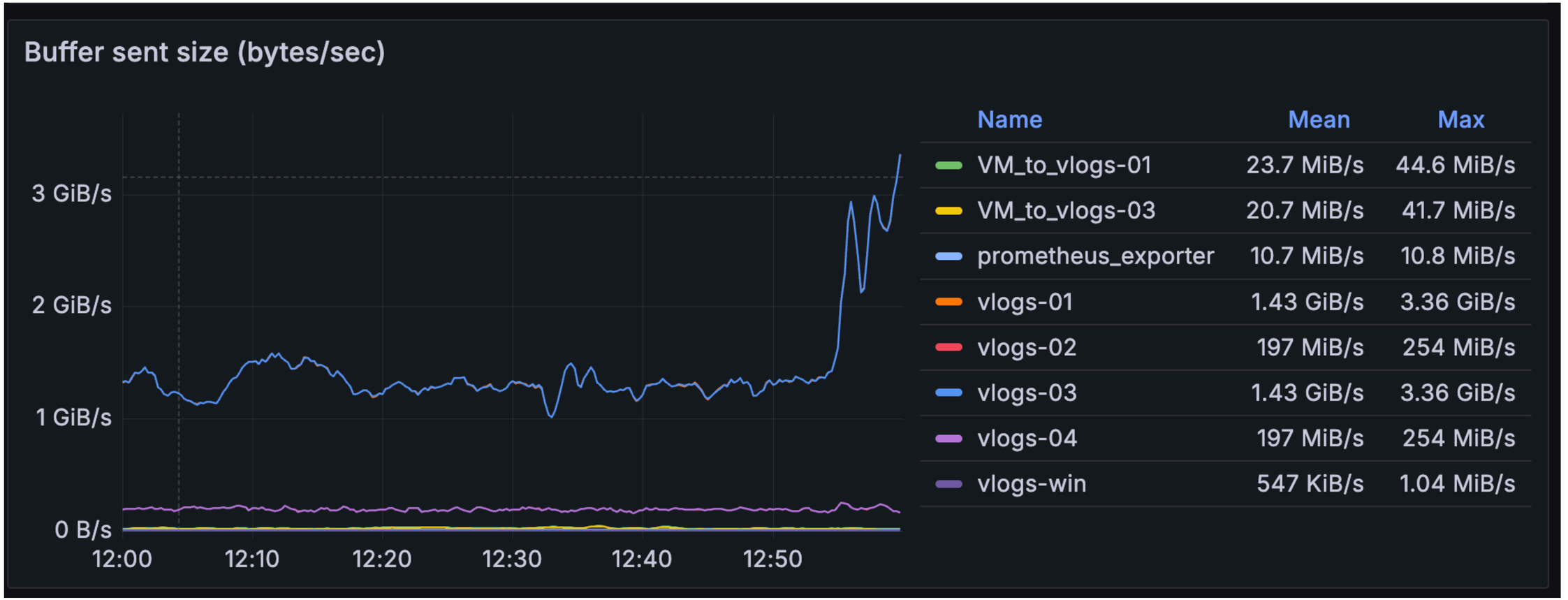
Формат компактен и требует меньше CPU/RAM для парсинга (до 2-3х раз меньше, чем в Elasticsearch _bulk-api).



Динамическая индексация полей.

Что получилось с доставкой?

~8 гигабайт трафика в секунду на одном из кластеров k8s
(таких у нас 30).



Разница по ресурсам на примере одного кластера

Elasticsearch / OpenSearch

6x Coord node: 16 CPU, 32 RAM, 200GB Disk

5x Ingest node: 16 CPU, 32 RAM, 32 GB Disk

15x Hot node: 32 CPU, 64 RAM, 2Tb Disk

7x Warm node: 24 CPU, 64 RAM, 4Tb Disk

10x Cold node: 64 CPU, 64 RAM, 30Tb Disk

Total:

CPU: 1464
RAM: 2260 GB
Disk: 361.36 TB

VictoriaLogs

2x VictoriaLogs storage

96 CPU

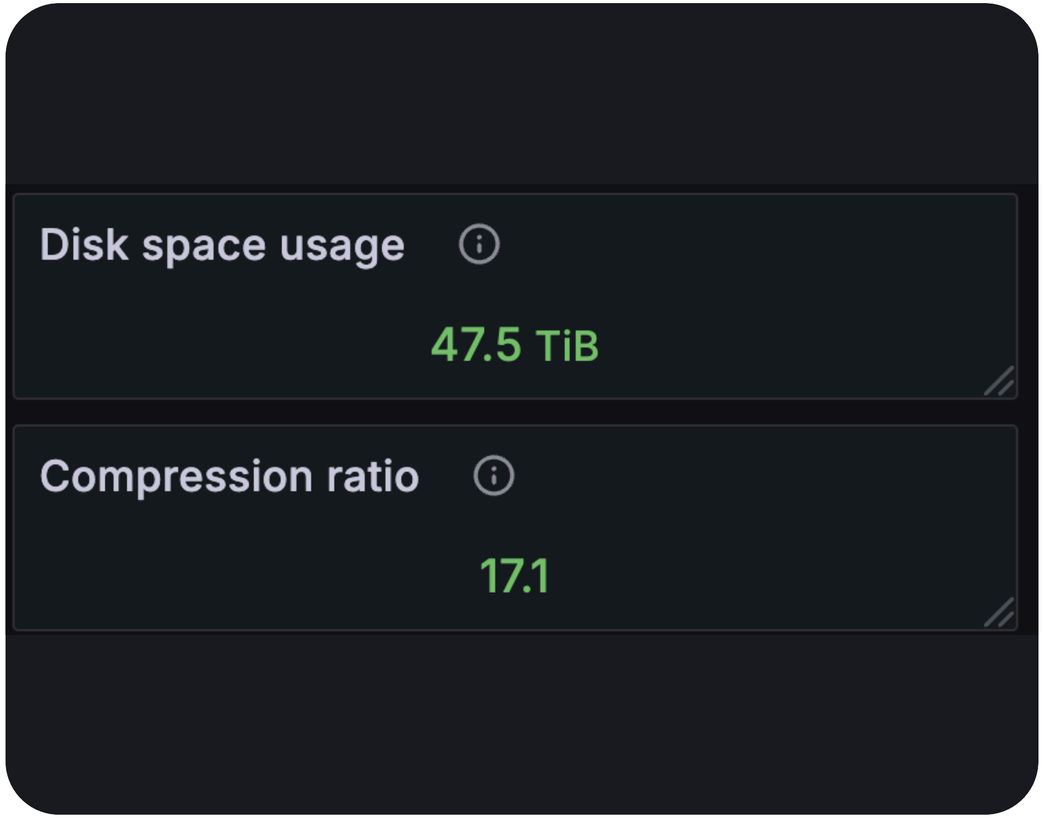
768 RAM

25,6Tb Disk

Total:

CPU: 192
RAM: 1536 GB
Disk: 51.2 TB

Что получилось с хранением?



Показатели компрессии

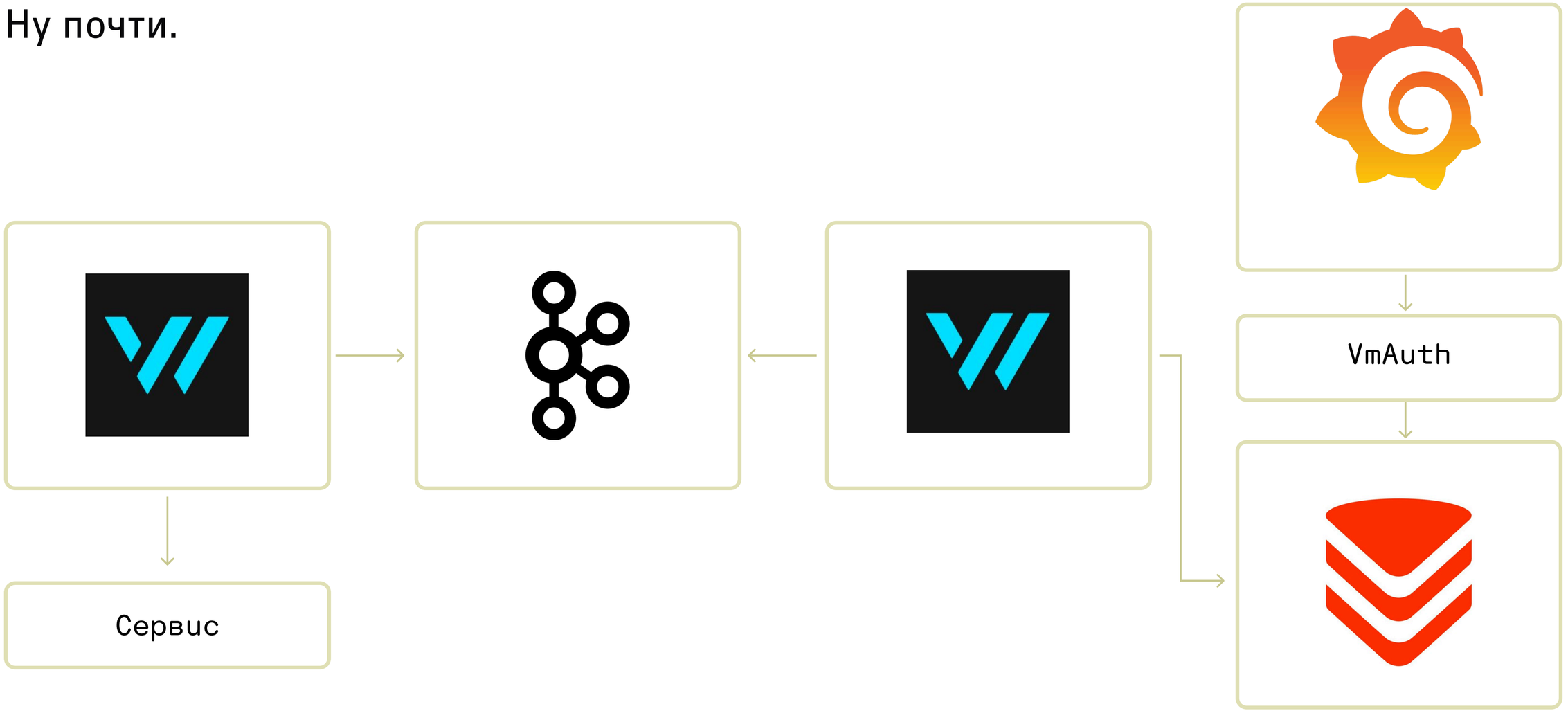
Non-default flags

instance	name	value
os-065013001.srvr. .local:	internStringDisableCache	true
os-065013001.srvr. .local:	loggerLevel	ERROR
os-065013001.srvr. .local:	retention.maxDiskSpaceUsageBytes	23000GiB
os-065013001.srvr. .local:	retentionPeriod	7d
os-065013001.srvr. .local:	storageDataPath	/data/victoria-logs/
os-065013002.srvr .local	internStringDisableCache	true

Пример конфигурации

Как сделать Production-ready

Ну почти.



Переходом на новый стек логирования решили проблемы

Необходимость описания маппингов под каждый микросервис.

(01)

Боль от разработки из-за недоступных логов.

(02)

Необходимость вручную перезапускать пайплайны Logstash.

(03)

Дорогое хранение логов.

(04)

Что получилось в итоге?

Сейчас развернуто 16 кластеров Victorialogs.

(01)

На сопровождение выделен 1 инженер.

(02)

Количество инцидентов, связанных с потерей логов, снизилось до 0.

(03)

Стоимость содержания логов снизилась в 6 раз.

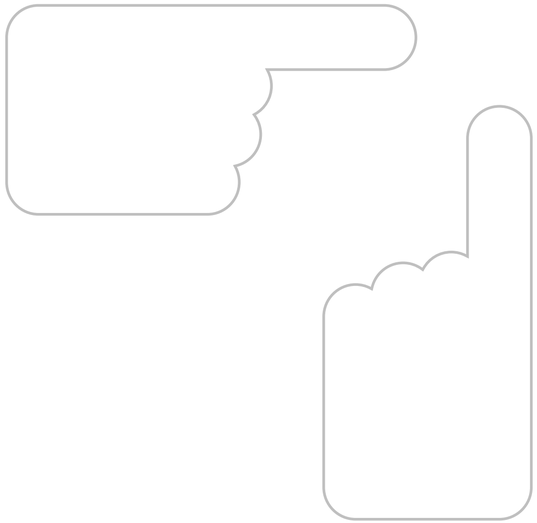
(04)

Нам неважно, в каком формате нам присылают логи.

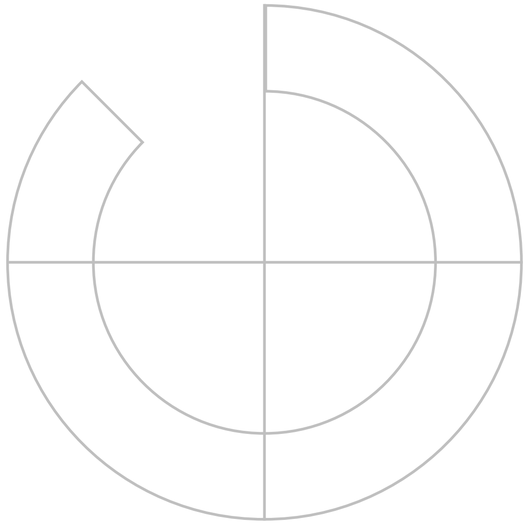
(05)

Наши планы по развитию

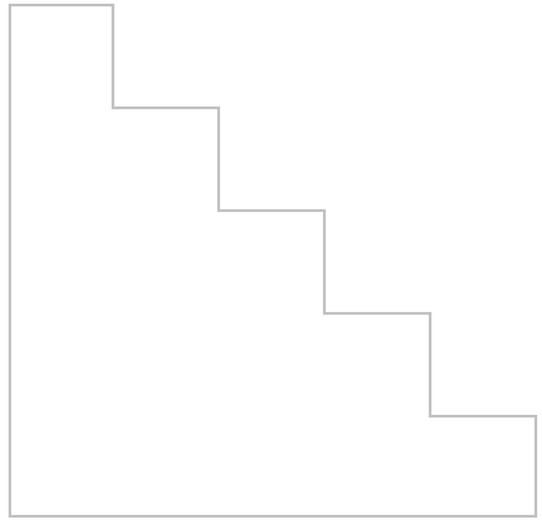
Создание снепшота логов в S3 — бэкапирование.



Разделение кластеров Vlogs на Hot/Cold (для более важных логов).



Добавление гарантий доставки (защиты от всплесков у соседей) за счёт создания топиков в Kafka per Product.



Важно упомянуть: ложка дёгтя

VLogs storage «кластеризуется» только горизонтально.

Кластер расширяется только добавлением новых хостов в Vector и VmAuth, сами ноды друг о друге ничего не знают.

Нет автоматического перелива данных между нодами.

Нет возможности автоматически отливать старые данные в «холодное» хранилище.

Немного советов

(01)

Приведение большой компании к единому стандарту логов — утопия. Придумайте, как подружиться с зоопарком логов без боли — Development Agnostic.

(02)

Стремитесь к отказоустойчивости в системе доставки: буферы, разделение обработки по компонентам.

(03)

Учитывайте человеческий фактор: всё это должно жить без необходимости вмешиваться и чинить руками.

ОСТАЁМСЯ НА СВЯЗИ

Валерий Евдокимов
(PTL Observability)

ecom.tech

@frandal