

Системы хранения и передачи данных: **ELK**



Иван
Богданов



Иван Богданов

Технический менеджер, Яндекс.Облако





План занятия

1. [ELK и Beats](#)
2. [Elasticsearch](#)
3. [Kibana](#)
4. [Logstash](#)
5. [Beats](#)
6. [Итоги](#)
7. [Домашнее задание](#)



Вопросы с прошлой лекции

Вопрос: Главные отличия redis от memcached?



Вопросы с прошлой лекции

Вопрос: Главные отличия redis от memcached?

Ответ: В redis есть cli, неймспейсы, репликация, поддержка типов, lua-скриптинг.



ELK и Beats

ELK

Стек **ELK** — это аббревиатура, используемая для описания стека, состоящего из трех популярных проектов:

- Elasticsearch,
- Logstash
- и Kibana.

ELK

Стек ELK предоставляет возможность:

- агрегировать журналы из всех ваших систем и приложений,
- анализировать эти журналы,
- создавать визуализации для мониторинга приложений и инфраструктуры, более быстрого устранения неполадок, анализа безопасности и многого другого.

Есть также платный Elastic Cloud и коммерческая версия ELK-стека.



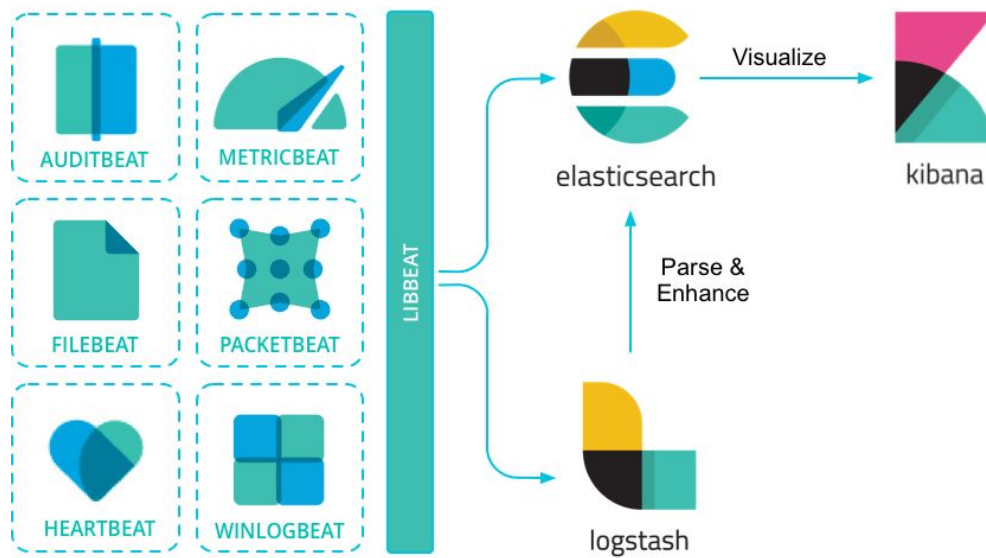
Beats

Beats — это «отправители» данных с открытым исходным кодом, которые вы устанавливаете в качестве агентов на своих серверах для отправки данных в Elasticsearch.

На данный момент есть Auditbeat, Filebeat, Functionbeat, Heartbeat, Journalbeat, Metrics, Packetbeat, Winlogbeat.

Мы посмотрим только на [Filebeat](#).

ELK и Beats



Источник

ELK и Beats

21 января 2021 года у Elasticsearch и Kibana изменилась политика лицензирования исходного кода с [Apache-2.0](#) на [SSPL](#), которая является несвободной.

Это значит, что все облачные провайдеры, которые используют Elastic или Kibana для коммерческих целей обязаны вносить свой вклад в развитие OSS-кода, а также выкладывать в свободный доступ свои инфраструктурные компоненты из которых сделано облако. На обычных пользователей это никак не должно повлиять.

Подробнее о новой политике лицензирования можно почитать [здесь](#).



Elasticsearch

Elasticsearch

Elasticsearch — это распределенная, поисковая и аналитическая система, которая является сердцем ELK-стека. Он централизованно хранит данные для поиска, точной настройки релевантности и мощной аналитики, легко масштабируется.

Все данные, которые будут писаться системой поставки, будут оседать и индексироваться в Elasticsearch.

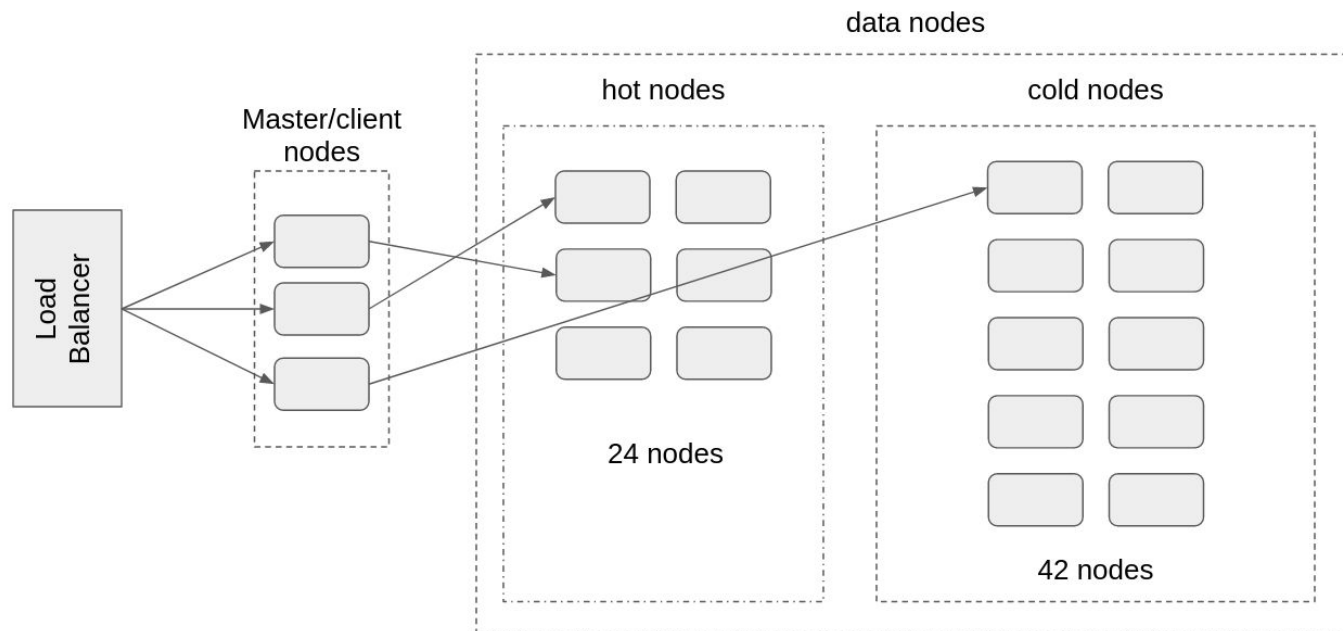


Elasticsearch

На основе elasticsearch строят не только системы поставки логов, но и сервисы для поиска бизнесовых данных для пользователей, например [ebay classifieds](#).

Данные в виде «документов» поставляются через API или тулзы вроде Logstash или Beats. После записи в базу поверх данных автоматически строятся индексы для быстрого поиска по полям через API или Kibana.

Elasticsearch



[Источник](#)

Elasticsearch

Установим Elasticsearch на Debian 10:

```
# apt update && apt install gnupg apt-transport-https <--зависимости
# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key
add - <--добавляем gpg-ключ
# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo
tee /etc/apt/sources.list.d/elastic-7.x.list <--добавляем репозиторий в apt
# apt update && apt-get install elasticsearch <--устанавливаем elastic
# systemctl daemon-reload <--обновляем конфиги systemd
# systemctl enable elasticsearch.service <--включаем юнит
# systemctl start elasticsearch.service <--запускаем сервис
```

После установки базы первым делом обезопасьте её! И настройте бекапы.

Elasticsearch

Проверяем, что сервер запустился:

```
# curl 'localhost:9200/_cluster/health?pretty'
{
  "cluster_name" : "netology-logging",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 1,
  "number_of_data_nodes" : 1,
  "active_primary_shards" : 1,
  "active_shards" : 1,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 0,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
```

Elasticsearch

Немного настройки в /etc/elasticsearch/elasticsearch.yml:

```
cluster.name: netology-logging # <--меняем имя кластера
node.name: node-1 # <--меняем название ноды, если нужно
node.roles: [ master, data, ingest ] # <--какую функцию будет выполнять эта
нода
cluster.initial_master_nodes: ["node-1"] # <--узлы, участвующие в голосовании
по выбору мастера
discovery.seed_hosts: ["ip-адрес"] # <--список возможных мастеров кластера
path.data: /var/lib/elasticsearch # <--где храним данные
path.logs: /var/log/elasticsearch # <--куда пишем логи
network.host: 0.0.0.0 # <--какой ip слушает хост
```

```
# systemctl restart elasticsearch
```

Elasticsearch

Какие могут быть роли у нод Elasticsearch?

Хотя бы у 1 ноды в кластере должны быть проставлены роли:

- master – контролирует кластер;
- data – хранит данные и проводит над ними операции;
- ingest – нода, в которую пишется входящий поток данных.

[Источник](#)

Elasticsearch

Уровни (tier) хранения данных на data-нодах:

- data_content – контент, который не перемещается между тирами (системные + пользовательские данные);
- data_hot – часто запрашиваемые и недавние данные (часы/дни);
- data_warm – не так часто запрашиваемые и более старые данные (недели);
- data_cold – данные, которые больше не обновляются;
- data_frozen – не обновляются и не запрашиваются.

[Источник](#)



Kibana



Kibana



kibana

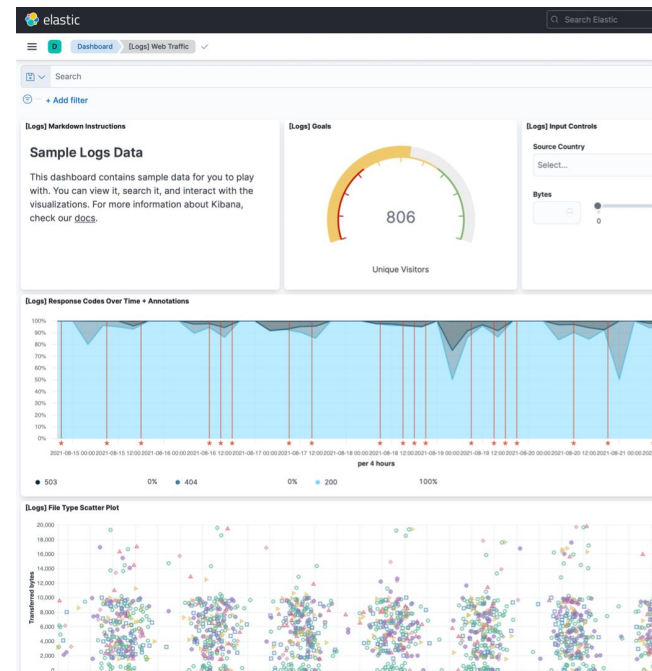
Kibana — это бесплатный и открытый пользовательский интерфейс, который позволяет визуализировать данные Elasticsearch.

Интерфейс отчасти похож на Grafana.

Kibana

Возможности kibana:

- визуализация данных;
- аналитика;
- мониторинг и алертинг;
- ML.



Kibana

Установим Kibana на Debian 10:

```
# apt install kibana <--установка  
# systemctl daemon-reload <--обновляем конфиги systemd  
# systemctl enable kibana.service <--включаем юнит  
# systemctl start kibana.service <--запускаем сервис
```




Kibana

Настройки в /etc/kibana/kibana.yml:

```
server.host: "0.0.0.0" # <--открываем интерфейс в мир
```

```
# systemctl restart kibana
```



Logstash



logstash

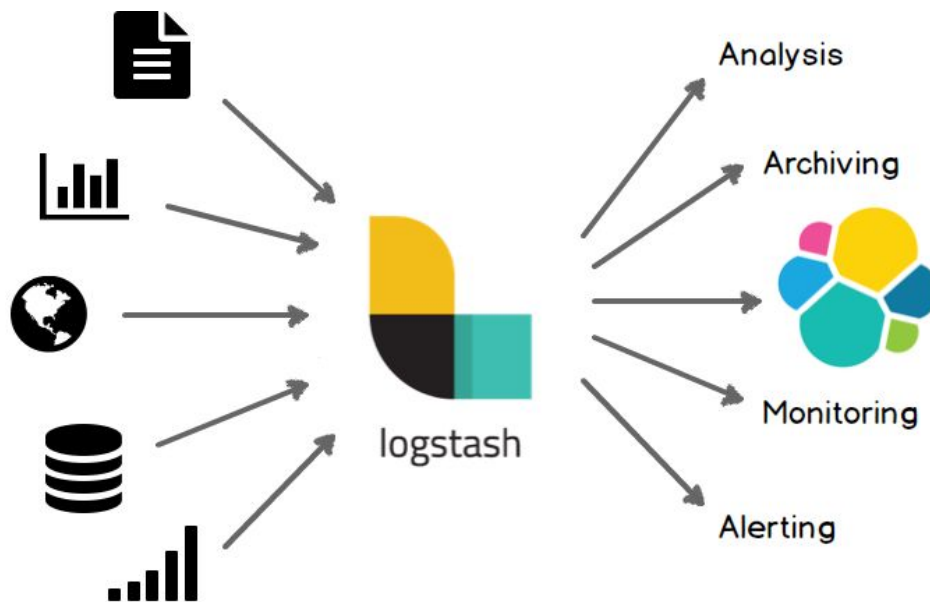
Logstash

Logstash — это сервис сбора данных с открытым исходным кодом и с возможностями конвейерной обработки в реальном времени. Logstash может динамически объединять данные из разрозненных источников и нормализовывать данные в места назначения по вашему выбору.

Брать любые данные, парсить, нормализовывать их и писать в Elasticsearch или в любой поддерживаемый провайдер.

Хранит стейт в файле.

Logstash



Источник

Logstash

Конфигурация Logstash делится на:

- **Inputs.** Отвечает за то, откуда Logstash возьмет данные, например из файла, syslog, stdin или redis.
- **Filters.** Как logstash изменит данные, которые пришли из **inputs**. Какие поля удалит, какие поменяет.
- **Outputs.** Куда после преобразования данные будут отправлены в elasticsearch или file, например.
- **Codecs.** Сериализация. Например, преобразование строки в json или наоборот.

Logstash

Установим Logstash на Debian 10:

```
# apt install logstash <--установка  
# systemctl daemon-reload <--обновляем конфиги systemd  
# systemctl enable logstash.service <--включаем юнит  
# systemctl start logstash.service <--запускаем сервис
```

Logstash

Настроим поставку access-лога nginx в elasticsearch:

```
input {
  file {
    path => "/var/log/nginx/access.log"
    start_position => "beginning"
  }
}
filter {
  grok {
    match => { "message" => "%{IPORHOST:remote_ip} - %{DATA:user_name}
\\[%{HTTPDATE:access_time}\\] \"%{WORD:http_method} %{DATA:url}
HTTP/%{NUMBER:http_version}\" %{NUMBER:response_code} %{NUMBER:body_sent_bytes}
\\\"%{DATA:referrer}\\\" \"%{DATA:agent}\\\""}
  }
  mutate {
    remove_field => [ "host" ]
  }
}
output {
  elasticsearch {
    hosts => "<ip-адрес>"
    data_stream => "true"
  }
}
```



Beats

Filebeat

Filebeat — это легковесный агент для пересылки и централизации данных из файлов. Устанавливается как демон на сервера. Основное отличие от Logstash — легкость и скорость, но с «урезанным» функционалом пайплайнов.

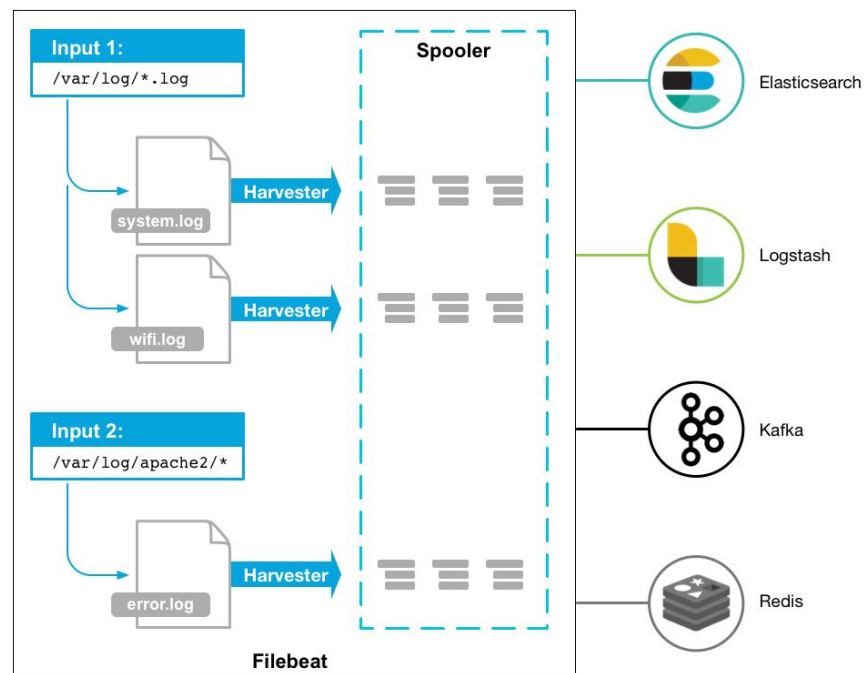
Также как Logstash, хранит стейт в файле и может менять данные перед отправкой.

Filebeat

Конфигурация состоит из 2 компонентов:

- **Inputs.** Как и откуда будут читаться данные для поставки.
- **Processors.** Позволяет незначительно менять данные в пайплайне.
- **Harvester**(комбайн). Запускается на каждый файл, который читает Filebeat, собирательное название каждой поставки данных.

Filebeat

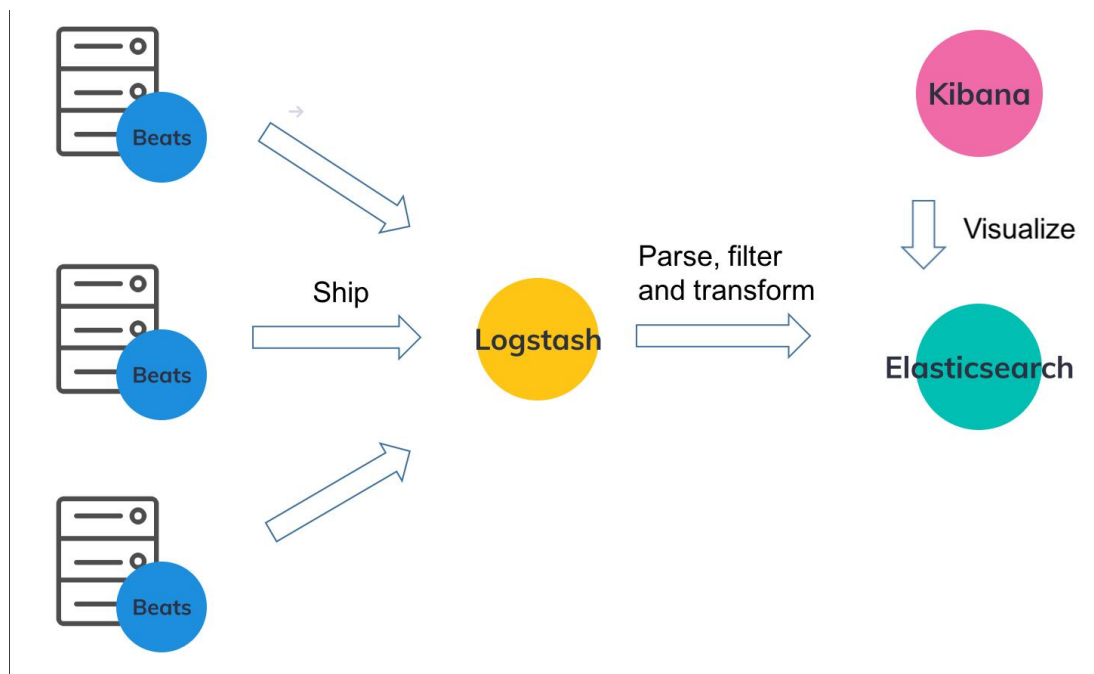


Источник



**Зачем нужен Logstash, если
есть Filebeat?**

Filebeat



Источник

Filebeat

Установим Filebeat на Debian 10:

```
# apt install filebeat <--установка  
# systemctl daemon-reload <--обновляем конфиги systemd  
# systemctl enable filebeat.service <--включаем юнит  
# systemctl start filebeat.service <--запускаем сервис
```

Filebeat

Настроим поставку access-лога nginx в elasticsearch с помощью модуля:

```
# filebeat setup --dashboards <--создает дашборды в kibana
# filebeat modules list <--смотрим список установленных модулей
...
nginx
# filebeat modules enable system nginx <--включим нужные нам модули
Enabled system <--полезный модуль для отправки данных системы
Enabled nginx

меняем конфиг /etc/filebeat/filebeat.yml
output.elasticsearch:
  hosts: ["<ip elasticsearch>:9200"]

# systemctl restart filebeat
```

И это не работает :)

Filebeat

Попробуем написать конфиг с нуля для отправки в Logstash:

```
# меняем конфиг Logstash
input {
  beats {
    port => 5044
  }
}

# меняем конфиг Filebeat
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/nginx/access.log
processors:
- drop_fields: # <--удаляются системные поля, которые добавил filebeat
  fields: ["beat", "input_type", "prospector", "input", "host", "agent",
"ecs"]

output.logstash:
  hosts: ["localhost:5044"]
```




Итоги

Итоги

Сегодня мы:

- узнали, что такое ELK-стек;
- какие задачи он решает;
- на практике настроили поставку логов nginx.





Домашнее задание



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Иван Богданов