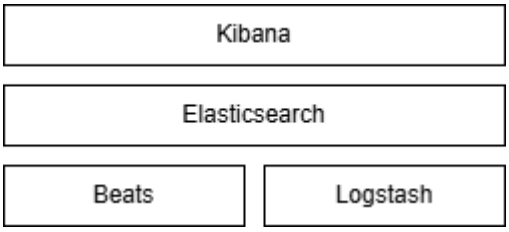
Blue team: ElasticStack

????????

Структура решения:



- Elasticsearch - полнотекстовый поиск, агрегация и хранение данных.
- Kibana - пользовательский интерфейс Elasticsearch. Спроектирован для Elasticsearch. Модульный за счет приложений
- Beats - сбор и отправка данных непосредственно из различных исходных систем (конечные точки, сетевые
  устройства или облака) в Logstash или Elasticsearch.
- Logstash - извлечение, преобразование и загрузка (ETL), используется для обработки и приема данных из различных источников (таких как файлы журналов на серверах, агенты Beats в вашей среде, очереди сообщений и платформы потоковой передачи) в Elasticsearch. Основная фишка - парсинг и преобразование полей для сохранения в Elasticsearch.

Elasticsearch

Основан на фреймворке Lucene для структурирования и поиска. Lucene индексирует элементы входного текста (индекс) и строит обратный индекс. Пример:

Элемент данных	Документ 1	Документ 2	Документ 3
Спагетти	+		
Сыр	+		+
Рецепт	+	+	+
Помидор		+	
Майонез			+
Картошка			+

Происходит объединение / пересечение полученных данных, ранжирование и отдача в соответствии с рангом.

Типы агрегаций:

- блоковая (bucket) агрегация. Группировка в зависимости от значений полей

- агрегация на основе метрик.

Могут использоваться схемы данных.

Архитектура

Данные -> Документы -> Сегменты -> Ноды

Горизонтальное масштабирование. Добавление нодов без простоя, автоматическое перераспределение сегментов данных по нодам.

Высокая доступность и надежность. Основной сегмент доступен на чтение и запись, остальные - чтение. Индексные и поисковые запросы выполняются параллельно.

Снимки, межкластерный поиск.

Схема хранения данных (Elastic Common Schema). ECS устанавливает сопоставления индексов для полей. Например целые числа могут быть как количеством переданных байт и подлежат суммированию, так может быть статусом ответа HTTP и являются строкой.

Модули Beats могут автоматически конвертировать логи и метрики в ECS схему.

Когда использовать Beast	**Когда использовать Logstash**
• Необходимо объединять данные из большого количества однотипных систем. • Есть модуль для данной системы • Не нужно проводить серьезные изменения перед передачей данных	• Когда большой объем данных поступает из централизованного хранилища (например, из общего файлового хранилища, AWS S3, Kafka и AWS Kinesis) и вам необходимо иметь возможность масштабировать пропускную способность. • Необходимость сложного преобразования данных • Балансировка нагрузки

?????????

ElasticSearch

Ubuntu, 4 ядра, 12ГБ, 30ГБ. Очень быстро ест диск, при пустом объеме менее 10% падает.

Вариант 1. Установка из зеркала yandex

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o
/usr/share/keyrings/elasticsearch-keyring.gpg
apt-get install apt-transport-https
echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg]
https://mirror.yandex.ru/mirrors/elastic/8/ stable main" | sudo tee
/etc/apt/sources.list.d/elastic-8.x.list
sudo apt update && sudo apt install elasticsearch
```

Вариант 2. Из deb пакета.

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-9.2.0-amd64.deb
sudo dpkg -i elasticsearch-9.2.0-amd64.deb
```

В консоли отобразится пароль суперюзера, его нужно сохранить.

The generated password for the elastic built-in superuser is : b0kLYdJDYetVHoPQafmc

Затем изменяем конфиг java vm, устанавливаем 8ГБ лимит. Памяти должно быть не более половины от существующей оперативной памяти. Минимально 4ГБ. nano /etc/elasticsearch/jvm.options

```
## heap to 4 GB, create a new file in the jvm.options.d
## directory containing these lines:
##
-Xms8g
-Xmx8g
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/8.19/heap-size.html
```

```
## for more information
```

Настройка кластера.

Файл /etc/elasticsearch/elasticsearch.yml

```
# All nodes in a cluster should have the same name
cluster.name: lab-cluster
# Set to hostname if undefined
node.name: node-a
# Port for the node HTTP listener
http.port: 9200
# Port for node TCP communication
transport.tcp.port: 9300
# Filesystem path for data directory
path.data: /mnt/disk/data
# Filesystem path for logs directory
path.logs: /mnt/disk/logs
# List of initial master eligible nodes
cluster.initial_master_nodes:
# List of other nodes in the cluster
discovery.seed_hosts:
# Network host for server to listen on
network.host: 0.0.0.0
```

Перегружаем службы

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
sudo systemctl start elasticsearch.service
```

Проверка запуска

```
curl localhost:9200
Вывод должен быть похож на:
{
   "name":"test",
   "cluster_name":"elasticsearch",
```

```
  "cluster_uuid":"D9HthzrVRTS4yKgSNEfrjg",
  "version":{
    "number":"8.0.0",
    "build_flavor":"default",
    "build_type":"deb",
    "build_hash":"51e9d6f22758d0374a0f3f5c6e8f3a7997850f96",
    "build_date":"2020-11-09T21:30:33.964949Z",
    "build_snapshot":false,
    "lucene_version":"8.7.0",
    "minimum_wire_compatibility_version":"6.8.0",
    "minimum_index_compatibility_version":"6.0.0-beta1"
  },
  "tagline":"You Know, for Search"
}
```


Проверка работы кластера:

```
curl localhost:9200/_cluster/health
```

Установка kibana

```
sudo apt install kibana
sudo systemctl daemon-reload
sudo systemctl enable kibana.service
sudo systemctl start kibana.service
sudo systemctl status kibana.service
```

Генерируем пароль для пользователя kibana

```
sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u kibana_system

Password for the [kibana_system] user successfully reset.
New value: QiF9U+blQujOvEg+jHyB
```

Настраиваем сертификаты

```
sudo cp -R /etc/elasticsearch/certs /etc/kibana
sudo chown -R root:kibana /etc/kibana/certs
```

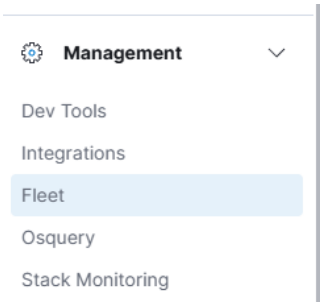
Настраиваем параметры kibana в файле /etc/kibana/kibana.yml

```
server.host: "192.168.1.184"
elasticsearch.username: "kibana_system"
elasticsearch.password: "QiF9U+blQujOvEg+jHyB"
elasticsearch.ssl.certificateAuthorities: [ "/etc/kibana/certs/http_ca.crt" ]
elasticsearch.hosts: ["https://192.168.1.184:9200"]
```

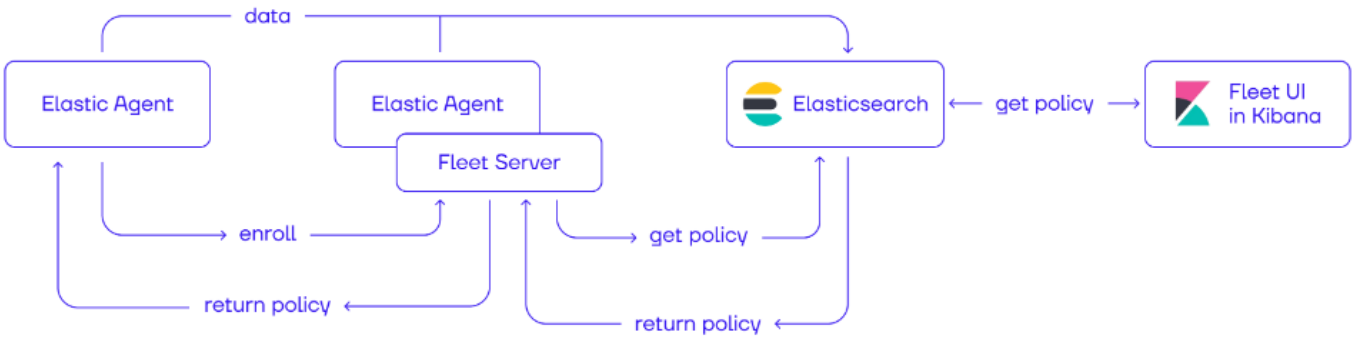
Входим по адресу http://192.168.1.184:5601/ и там логин elastic пароль b0kLYdJDYetVHoPQafmc

Fleet

Бэкенд для управления агентами, фронт через kibana. Управление через политики. Желательно ставить на отдельном сервере. Настраивается через Kibana Management - Fleet



Взаимодействие компонентов:



Fleet настраивается через Policy в Kibana, настройки(policy) хранятся в Elasticsearch, а агенты регистрируются на Fleet Server, получают через него конфигурацию, и отправляют данные в Elasticsearch.

Установка Fleet server

Ставим Debian.

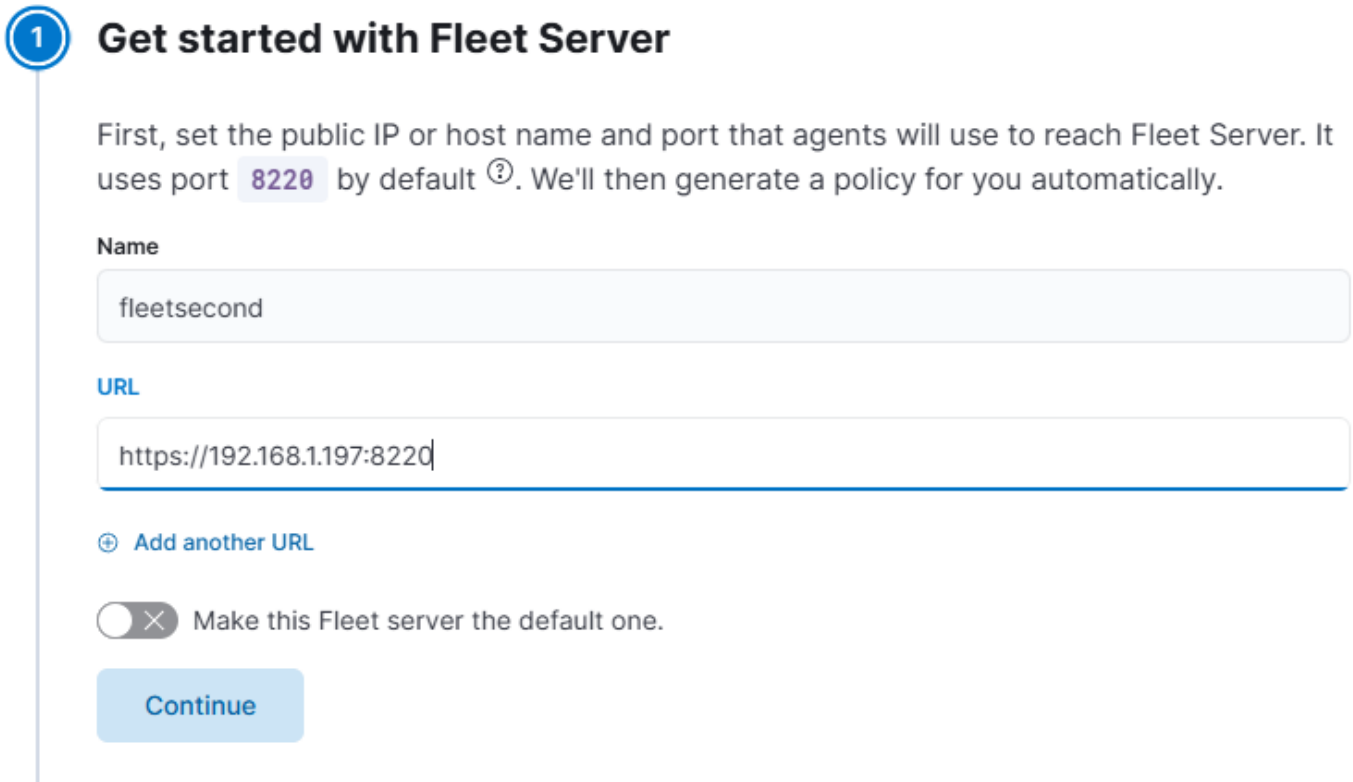
Сначала с ElasicSearch копируем сертификат на будущий сервер

```
sudo scp /etc/elasticsearch/certs/http_ca.crt user@<fleet_server_host>:/tmp/
```

На Fleet

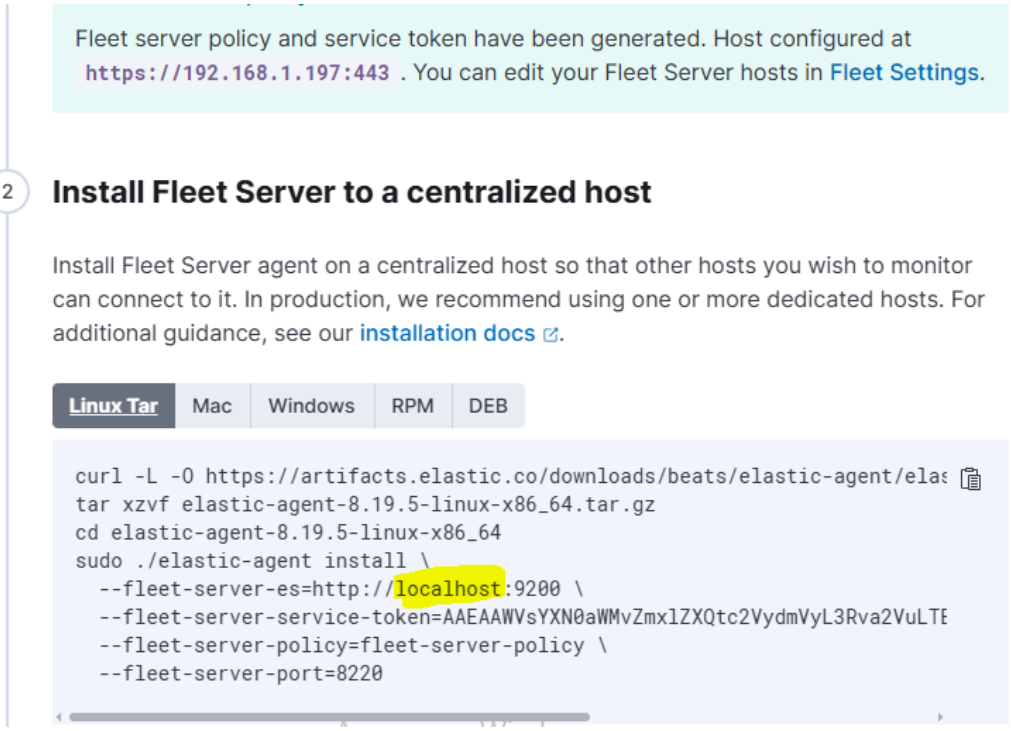
```
sudo mkdir -p /etc/elastic/certs
sudo mv /tmp/http_ca.crt /etc/elastic/certs/
sudo chmod 644 /etc/elastic/certs/http_ca.crt
sudo chown -R root:root /etc/elastic/certs
sudo chmod 755 /etc/elastic
sudo chmod 644 /etc/elastic/certs/http_ca.crt
```

В kibana создаем новый сервер. Указываем отображаемое имя и URL будущего Fleet server. Обязательно установить порт, иначе будет стучаться на 443 по умолчанию и почему-то не поедет.



Нажимаем Generate... ELK сгенерирует набор команд, которые необходимо выполнить на будущем Fleet сервере для установки. Можно выбрать ОС для команды. В качестве localhost

нужно указать адрес elastic сервера.



И к установочной строке нужно добавить ссылку на сертификат. Итоговая строка:

```
sudo ./elastic-agent install   --fleet-server-es=https://192.168.1.184:9200   --fleet-server-service-token=AAEAAWVsYXN0aWMvZmxlZXQtc2VydmVyL3Rva2VuLTE3NjA2MTgzNzk5MjQ6U2tNMlRSczRRcW00emV6aXZkR1JmZw   --fleet-server-policy=fleet-server-policy   --fleet-server-port=8220   --certificate-authorities=/etc/elastic/certs/http_ca.crt
```

И еще - токен имеет время жизни, поэтому при ошибке его нужно обновить.

Управление конфигурацией агентов

Политики Fleet управляют конфигурацией агентов elastic. Давайте настроим наши агенты на сбор системных логов. Перейдем в раздел Agent policies:

В разделе "Add Integrations" можно добавить в политику модули для сбора данных.

Интеграции(Integrations) это модули для сбора данных из различных источников. С актуальным списком доступных интеграций можно ознакомиться по адресу https://www.elastic.co/integrations

Давайте добавим интеграцию System. Для этого введем название интеграции в стоке поиска, откроем ее и нажмем Add System. В настройках модуля System есть опция добавить или изменить файлы логов, которые будут мониториться на системах linux, типы журналов Windows

Beats & Logstash

Logstash

```
sudo apt-get install logstash
```

Beats packages.

```
sudo apt-get install <beat>
# To install Filebeat, run this:
sudo apt-get install filebeat
# To install Metricbeat, run this:
sudo apt-get install metricbeat
```