rr-th.com +7 (495) 231-73-64 office@rr-th.com Москва, Цветной бульвар д. 26, стр. 1, офис 36

ИНСТРУКЦИЯ АДМИНИСТРАТОРА ИНДЕКСЛОГ

rr-th.com +7 (495) 231-73-64 office@rr-th.com Москва, Цветной бульвар д. 26, стр. 1, офис 36

1. Принципы работы ИндексЛог

1.1. Основные компоненты технологического стека

ИндексЛог – это технологический стек, обеспечивающий платформу для реализации решений различных классов. В стек входит ИндексЛог_Поиск – распределенный поисковый движок, ИндексЛог_Аналитика – интерфейс для работы с движком – и различные методы сбора и обработки данных перед их индексацией в ИндексЛог_Поиск.

ИндексЛог это распределенное решение, поэтому оно удобно масштабируется. Один процесс (службу) ИндексЛог_Поиск (обычно запущенный на одной машине) называют нодой (от англ. node – узел). Несколько нод, связанных и работающих вместе называются кластером.

ИндексЛог_Поиск принимает по HTTP запросы от клиентов или интерфейса ИндексЛог_Аналитика. Решение работает с запросами REST API, ответ возвращает в формате JSON документа.

ИндексЛог_Аналитика это фронтэнд решение, предоставляющее интерфейс для работы с движком ИндексЛог_Поиск. Оно содержит инструменты визуализации полученных данных, регистрации сигналов и отправки уведомлений. Также в ИндексЛог_Аналитика разграничивается доступ конечных пользователей к управлению конфигурацией и просмотру данных, хранящихся в ИндексЛог_Поиск.

Сбор данных осуществляется за счет агентов с централизованной конфигурацией, а также с помощью платформы Beats – сборщиков данных с открытым исходным кодом. Перед индексацией в ИндексЛог_Поиск данные могут быть предварительно обработаны с помощью Logstash – обработчика данных с открытым исходным кодом.

1.2. Структура данных в ИндексЛог_Поиск

Основная задача ИндексЛог_Поиск – индексировать данные и предоставлять к ним доступ с возможностью быстрого настраиваемого поиска. Каждый объект внутри ИндексЛог_Поиск представлен в виде JSON-файла или документа. Документ состоит из полей и значений.

Логически сгруппированные документы формируют индекс. Индекс представляет собой набор логически сгруппированных документов, и не обязательно хранится на одной ноде. Внутри индекса документы разбиты на шарды. О шардах удобно думать, как о папках, в которых содержатся документы.

Разделение массива документов на несколько шардов (папок) позволяет хранить этот массив одновременно на нескольких нодах. Благодаря хранению индекса на нескольких нодах, процесс поиска данных значительно ускоряется, поскольку операция поиска проводится одновременно на нескольких машинах, каждая из которых содержит меньше документов.

Для повышения надежности хранения данных, предусмотрено два вида шардов – основные и реплики. Между основными шардами распределяется массив индексируемых документов. Реплики – это копии основных шардов, которые хранятся на с основными на разных нодах. При этой конфигурации в случае отключения одной из нод, доступ к документам не будет потерян. Также реплики позволяют дополнительно ускорить операции поиска.

1.3. Типы данных и цикл хранения в ИндексЛог_Поиск

ИндексЛог_Поиск работает с двумя основными типами данных – статическими и потоковыми. Статические данные можно ассоциировать с каталогом, информация в котором устаревает медленно, может изменяться и дополняться, все время требует быстрого доступа. Потоковые данные можно представлять в виде записей в логе. Их объем быстро растет, а актуальность со временем падает. Вероятность обратиться к вчерашним записям в логе гораздо выше, чем вероятность обращения к записям месячной давности. Запросов к старым данным гораздо меньше, поэтому бизнес может позволить себе менее быстрый доступ к этим данным.

Для эффективного хранения потоковых данных в ИндексЛог применяется многоуровневая структура данных с соответствующим циклом хранения. Каждая стадия этого цикла может быть настроена по длительности или пропущена.

Горячие (hot) данные – это свежие, актуальные данные, к которым нужен максимально быстрый доступ. Индексы с горячими данными имеют несколько реплик для каждого основного шарда и занимают больше дискового пространства, чем индексы на следующих стадиях цикла. Рекомендуется хранить такие индексы на машинах с твердотельными накопителями. Использование твердотельных накопителей и нескольких реплик горячих данных означает, что хранение таких данных сравнительно дорого обходится бизнесу, поэтому необходимо со временем переводить данные на следующие этапы цикла хранения, где их объем и стоимость хранения будет уменьшаться. По достижению определенного объема занимаемого пространства или возраста данные автоматически будут переходить на следующую стадию цикла хранения. Выполняемые при этом операции также могут быть настроены. При переходе от горячей стадии индекс получает статус read-only, и в него не могут быть добавлены новые документы.

Теплые (warm) данные – это следующая после горячих стадия цикла хранения данных, где к данным все еще может потребоваться доступ, но это происходит не так часто и не так срочно, как с горячими данными. Поэтому при переходе индекса из горячей категории в теплую, уменьшается количество реплик каждого основного шарда. Также данные индекса могут дополнительно сжиматься. Теплые данные могут храниться уже на жестких магнитных дисках, а также имеют меньший объем, чем горячие. Это заметно удешевляет их хранение. Тем не менее, к ним все еще обеспечивается достаточно быстрый доступ, и с ними почти так же удобно работать аналитику, как с горячими данными.

Холодные (cold) данные – одна из поздних стадий цикла хранения данных. При переходе на нее с индекса может делаться снимок (searchable snapshot), который может впоследствии использоваться для поиска или восстановления утраченных данных. При переходе к холодной стадии хранения данных можно уменьшить количество основных шардов в индексе, а также полностью избавиться от реплик. Это сильно уменьшит объем данных и позволит долго хранить их в этой фазе. При этом на случай отказа одного из серверов, на которых хранится холодный индекс, рекомендуется иметь снэпшот.

Удаление – последний этап цикла хранения данных. Индекс удаляется, и доступ к нему становится невозможен

Статические данные не используют цикл хранения и хранятся на серверах ИндексЛог_Поиск так же, как и горячие потоковые данные. Так как объем статических данных растет медленно, скачков в потреблении ресурсов и дискового пространства системы не случается.

2. Настройка ноды и кластера ИндексЛог_Поиск

2.1. Настройка ноды на присоединение к существующему кластеру

При установке ИндексЛог_Поиск производится автоматическая конфигурация кластера из одной ноды. Если вы хотите присоединить настраиваемую ноду к существующему кластеру, сгенерируйте enrollment token на одной из нод этого кластера до первого запуска той ноды, которую надо присоединить.

1. На любой ноде существующего кластера сгенерируйте enrollment token:

```
/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s node
```

2. Скопируйте enrollment token, выведенный в терминал
3. На новой настраиваемой ноде используйте токен в качестве параметра для инструмента elasticsearch-reconfigure-node:

```
/usr/share/elasticsearch/bin/elasticsearch-reconfigure-node --enrollment-token <enrollment-token>
```

4. Запустите ноду с помощью systemd

2.2. Запуск ИндексЛог_Поиск с помощью system

Чтобы ИндексЛог_Поиск автоматически запускался при запуске системы, выполните следующие команды:

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

Следующие команды используются для запуска и остановки ИндексЛог_Поиск:

```
sudo systemctl start elasticsearch.service
sudo systemctl stop elasticsearch.service
```

Эти команды не выводят в терминал записи о том, успешно ли запустился ИндексЛог_Поиск, вместо этого записи делаются в лог по адресу /var/log/elasticsearch/elasticsearch.log

Если keystore ИндексЛог_Поиск защищено паролем, необходимо предоставить system этот пароль с помощью локального файла и переменных среды system. Этот файл должен быть защищен на время своего существования и может быть безопасно удален после успешного запуска ИндексЛог_Поиск.

```
echo "keystore_password" > /path/to/my_pwd_file.tmp
chmod 600 /path/to/my_pwd_file.tmp
sudo systemctl set-environment ES_KEYSTORE_PASSPHRASE_FILE=/path/to/my_pwd_file.tmp
sudo systemctl start elasticsearch.service
```

2.3. Проверка запущенного ИндексЛог_Поиск

Вы можете проверить запущенную ноду ИндексЛог_Поиск с помощью HTTPS запроса на адрес localhost:9200. Убедитесь, что используете HTTPS запрос, иначе запрос будет отклонен. --cacert это путь к сгенерированному HTTP сертификату http_ca.crt:

```
curl --cacert /etc/elasticsearch/certs/http_ca.crt -u elastic https://localhost:9200
```

Введите пароль пользователя elastic, который был сгенерирован при установке. Ответ должен выглядеть так:

```
{
 "name" : "Cp8oag6",
 "cluster_name" : "elasticsearch",
 "cluster_uuid" : "AT69_T_DTp-1qgIJlatQqA",
 "version" : {
  "number" : "8.6.2",
  "build_type" : "tar",
  "build_hash" : "f27399d",
  "build_flavor" : "default",
  "build_date" : "2016-03-30T09:51:41.449Z",
  "build_snapshot" : false,
  "lucene_version" : "9.4.2",
  "minimum_wire_compatibility_version" : "1.2.3",
  "minimum_index_compatibility_version" : "1.2.3"
 },
 "tagline" : "You Know, for Search"
}
```

3. Конфигурация ИндексЛог_Поиск

Конфигурация ИндексЛог_Поиск производится через изменение файла elasticsearch.yml, расположенного в $ES_HOME/config/ elasticsearch.yml.

3.1. Важные настройки ИндексЛог_Поиск

3.1.1 Настройка имени кластера

Нода может присоединиться к кластеру, только если в ней выставлено, то же значение имени кластера, что и во всех нодах этого кластера. При использовании нескольких кластеров разделяйте их с помощью имен:

```
cluster.name: logging-prod
```

3.1.2 Настройка имени нода

ИндексЛог_Поиск использует параметр имени ноды как понятный для человека идентификатор ноды. По умолчанию дублирует hostname машины, на которой запущен ИндексЛог_Поиск:

```
node.name: prod-data-2
```

3.1.3 Настройка Network host

Для использования кластера более чем из одной ноды, необходимо указать адрес машины, на которой запущен ИндексЛог_Поиск:

```
network.host: 192.168.1.10
```

3.2. Настройки обнаружения и формирования кластера

При запуске ИндексЛог_Поиск или отключении master-ноды, запускается процесс обнаружения (discovery). Процесс требует наличия списка адресов (seed addresses) и списка master-eligible нод, которые были в последнем запущенном кластере. Процесс обнаружения происходит в два этапа:

1. На каждой ноде обрабатывается список адресов, происходит подключение к каждому из них, попытка идентифицировать ноду, к которой произведено подключение, и проверка того, может ли она быть выбрана master.

2. В случае успешного подключения ноды обмениваются списками и производят подключения ко всем адресам в списке.

Если нода не может быть выбрана master, она продолжает процесс обнаружения, пока не найдет выбранную master ноду или достаточное количество master-eligible нод для проведения голосования. Если это не выходит, повторная попытка происходит через интервал discovery.find_peers_interval (по умолчанию 1с).

3.2.1 Объявление списка адресов в файле конфигурации

Список адресов хостов объявляется через настройку discovery.seed_hosts, например:

```
discovery.seed_hosts:
- 192.168.1.10:9300
- 192.168.1.11
- seeds.mydomain.com
```

3.2.2 Объявление списка адресов в отдельном файле

Для объявления списка адресов в отдельном файле установите в файле конфигурации Elasticsearch discovery.seed_providers: file. Затем создайте файл $ES_PATH_CONF/unicast_hosts.txt. Каждый раз при изменении unicast_hosts.txt список хостов ИндексЛог_Поиск обновляется.

```
10.10.10.5
10.10.10.6:9305
10.10.10.5:10005
# an IPv6 address
[2001:0db8:85a3:0000:0000:8a2e:0370:7334]:9301
```

cluster.initial_master_nodes

При первом запуске ИндексЛог_Поиск определяется набор master-eligible нод, между которыми проводится голосование. При этом, желательно самостоятельно объявлять набор master-eligible нод. Объявите master-eligible ноды по их node.name.

```
cluster.initial_master_nodes:
  - master-node-a
  - master-node-b
  - master-node-c
```

После первого успешного формирования кластера удалите эту настройку из конфигурации всех нод. Не используйте ее при перезапуске кластера и добавлении новых нод.

3.3. Настройки ролей ноды ИндексЛог_Поиск

В кластере ИндексЛог_Поиск разные машины могут брать на себя различные роли и выполнять внутри него разные задачи. Это настраивается параметром node.roles в файле конфигурации elasticsearch.yml. Если этот параметр не указан, по умолчанию нода имеет следующие роли:

- master;

- data;
- data_content;
- data_hot;
- data_warm;
- data_cold;
- ingest;
- remote_cluster_client;
- transform.

Для функционирования кластера минимально необходимым является наличие нод со следующими ролями:

- master;
- data или data_content вместе с data_hot.

Для некоторых возможностей ИндексЛог_Поиск также необходимы ноды со следующими ролями:

- для кросс-кластерного поиска данных необходимы ноды с ролью remote_cluster_client;
- для мониторинга стека и обработчиков входящих даннных необходимы ноды с ролью ingest;
- сервер ЦУА, ИндексЛог_Безопасность и трансформации требуют наличия нод с ролью transform.

3.3.1 Координационные ноды

Некоторые запросы к кластеру требуют одновременного использования данных, хранящихся на разных нодах. Для этого нода, которая получает запрос координирует выполнение запроса на остальных нодах. Она перенаправляет запрос на ноды, на которых хранятся данные, консолидирует полученные данные в единый набор результатов и посылает его обратно.

По умолчанию, каждая нода является координационной. Чтобы нода имела исключительно координационную роль, необходимо выставить пустой параметр node.roles: [].

3.3.2 Master-ноды

Master-ноды отвечают за простые общекластерные действия, такие как создание и удаление индексов, отслеживание нод, входящих в кластер и принятие решений о распределении шардов между нодами.

Любая нода с ролью master без роли voting-only может быть выбрана master-нодой.

Для стабильности всего кластера необходима стабильно работающая master-нода. Если master-нода будет загружена другими задачами, снизится общая производительность кластера. Поэтому в кластерах размером больше 5 нод рекомендуется иметь отдельные master-ноды, которые не имеют других ролей. Также эти master-ноды будут выполнять задачи координации.

При наличии в параметре node.roles комбинации ролей master и voting_only, нода не может быть выбрана мастером, но может участвовать в голосовании. Только ноды с ролью master могут иметь роль voting_only.

3.3.3 Data-ноды

Дата ноды хранят шарды, в которых находятся документы, проиндексированные Elasticsearch. Дата ноды проводят основные операции с данными, такие как чтение, модификация, поиск и группировка. Эти операции имеют требования к конфигурации машины, на которой запущен ИндексЛог_Поиск. Для продашкн-кластеров рекомендуются ноды с 8 ядрами процессора и 64 ГБ оперативной памяти. Размер и тип дискового пространства выбирается в зависимости от типа данных, хранящихся на данной ноде, и начинается от 1-2 ТБ SSD до 200 ТБ HDD. Базовое значение node.roles для всех типов данных: node.roles: [data]. Такая конфигурация используется, если не применяется многоуровневая структура хранения данных. Если ноде присвоена любая другая роль из списка data_content, data_hot, data_warm или data_cold, роль data ей присвоена быть не может. При этом, нода может иметь несколько ролей из этого списка одновременно.

Роли data_content, data_hot, data_warm, data_cold используются для разграничения нод, ответственных за различные стадии цикла хранения данных.

Data-content ноды хранят данные, которые долго остаются актуальными и могут обновляться и редактироваться. Они не предназначены для хранения данных логов и метрик. Наличие нод с ролью data-content обязательно для старта кластера.

Data-hot ноды хранят данные первой на первой (горячей) стадии цикла хранения данных. В нее автоматически попадают новые создаваемые индексы. Ноды категории data-hot обеспечивают максимальную скорость чтения и записи данных, поэтому имеют наибольшие требования к системным ресурсам. Роль data_hot является обязательной, в отличие от остальных ролей, присваиваемых нодам, участвующим в цикле хранения данных.

3.4. Сетевые настройки

Каждая нода ИндексЛог_Поиск использует два различных сетевых интерфейса. Kibana и клиенты посылают REST API запросы через HTTP-интерфейс. Между собой ноды связываются по транспортному интерфейсу. Также транспортный интерфейс используется для связи с удаленными кластерами.

Оба этих интерфейса можно настраивать одновременно через настройки network.*. Если вы используете более сложную сетевую структуру, то можно использовать настройки http.* и transport.*. По умолчанию ИндексЛог_Поиск привязан только к localhost и закрыт для дистанционного доступа.

- network.host: устанавливает адрес для HTTP и транспортного трафика. Нода использует этот адрес и публикуется на него же. Принимает в качестве значений IP-адрес или имя хоста. По умолчанию имеет значение _local.
- http.port: порт, через который ИндексЛог_Поиск будет получать HTTP-трафик. Принимает в качестве значений одно значение или промежуток значений. Если указан промежуток, ИндексЛог_Поиск занимает первый свободный порт в этом промежутке. По умолчанию имеет значение 9200-9300.
- transport.port: порт, через который ноды ИндексЛог_Поиск связываются между собой. Принимает в качестве значений одно значение или промежуток значений. Если указан промежуток, ИндексЛог_Поиск занимает первый свободный порт в этом промежутке. Установите одно значение (не промежуток) на каждой ноде с ролью master. По умолчанию имеет значение 9300-9400.

3.5. Автоматическая настройка безопасности

При первом запуске ИндексЛог_Поиск автоматически применяются следующие настройки безопасности:

- генерируются сертификаты и ключи TLS для транспортного и HTTP трафика;
- настройки TLS записываются в конфиг-файл elasticsearch.yml;
- генерируется пароль для пользователя elastic;
- генерируется токен для подключения ИндексЛог_Аналитика.

По умолчанию устанавливается защита соединения с помощью сертификата, подписанного ИндексЛог_Поиск. Соединение по HTTP защищается с помощью сертификата http.p12, соединение с другими нодами использует сертификат transport.p12.

При подключении дополнительных нод и ИндексЛог_Аналитика с использованием генерируемого ИндексЛог_Поиск токена, данные безопасности на подключаемых нодах настраиваются автоматически.

4. Настройка сервера ИндексЛог_Аналитика

ИндексЛог_Аналитика – это фронтэнд решение стека ИндексЛог, которое предоставляет пользователем интерфейс для работы с ИндексЛог_Аналитика. Через ИндексЛог_Аналитика осуществляется посылание запросов в кластер ИндексЛог_Поиск и представление ответов на эти запросы. Также ИндексЛог_Аналитика обладает функционалом для управления компонентами стека, индексами, потоками данных, сигналами, оповещением и машинным обучением. В большинстве инсталляций ИндексЛог_Аналитика находится на отдельном сервере и осуществляет доступ к кластеру по протоколу HTTP через порт 9200. HTTP-трафик между кластером и ИндексЛог_Аналитика защищен TLS сертификатом по умолчанию. В тестовых инсталляциях ИндексЛог_Аналитика может находиться на одной машине с нодой ИндексЛог_Поиск.

Для установки ИндексЛог_Аналитика требуется сервер с OS Ubuntu 20.04 и с учетной записью пользователя без привилегий root и с привилегиями sudo. ИндексЛог_Аналитика можно устанавливать как на одну машину с ИндексЛог_Поиск, так и на выделенный сервер.

4.1. Аутентификация пользователей

ИндексЛог_Аналитика поддерживает различные механизмы аутентификации пользователей и предлагает возможность их одновременного использования. Механизмы аутентификации указываются в порядке убывания приоритета, дополнительно каждый механизм имеет параметр order для определения приоритета.

При использовании нескольких механизмов аутентификации пользователь может выбрать один. Конфигурация окна выбора механизма аутентификации в файле конфигурации kibana.yml может выглядеть следующим образом:

```
xpack.security.loginHelp: "**Help** info with a [link](...)"
xpack.security.authc.providers:
  basic.basic1:
    order: 0
    icon: "logoElasticsearch"
    hint: "Typically for administrators"
  saml.saml1:
```

```
order: 1
realm: saml1
description: "Log in with SSO"
icon: "https://my-company.xyz/saml-logo.svg"
```

4.2. Разграничение доступа к функциям ИндексЛог_Аналитика и данным ИндексЛог_Поиск

Разграничение доступа к функциями Elastic происходит во вкладке Management > Stack Management. Настройки пользователей и ролей находятся соответственной в разделе Security > Users и Security > Roles. Пространства Kibana настраиваются в разделе ИндексЛог_Аналитика > Spaces.

4.2.1 Простанства

Пространства позволяют удобно организовать представления и другие сохраненные объекты. По умолчанию создается пространство по умолчанию. Для управления пространствами откройте вкладку Stack Management > Spaces. Для создания нового пространства нажмите «Создать пространство». При этом необходимо указать его URL идентификатор, который не может быть изменен позднее.

При создании нового пространства или редактировании существующего настраиваются его имя, описание, аватар и доступные в нем вкладки ИндексЛог_Аналитика. Рекомендуется оставлять необходимые для конкретных пространств вкладки, например, спрятать вкладки «Инструменты разработчика» и «Мониторинг стека» для аналитиков и оставить доступными для администраторов стека Elastic.

4.2.2 Пользователи и роли

В стеке ИндексЛог не предусмотрено закрепление прав доступа напрямую за пользователем. Вместо этого права доступа к функциям стека, возможностям ИндексЛог_Аналитика и прочим компонентам решения закрепляются за ролью пользователя, которая уже в свою очередь присваивается пользователю. Пользователь имеет следующие параметры: имя пользователя, полное имя, адрес электронной почты, пароль и присвоенные пользователю роли.

При создании роли ей можно присвоить пространства, к которым ее обладатели будут иметь доступ, разрешения ИндексЛог_Поиск и разрешения ИндексЛог_Аналитика.

4.2.3 Разрешения ИндексЛог_Поиск

За ролью могут быть закреплены разрешения, касающиеся управления кластером и индексами, а также доступом к их содержанию. Эти разрешения объявляются при настройке роли. Доступна возможность предоставить право доступа и изменения не всего индекса, а только выбранных полей или документов.

4.2.4 Разрешения ИндексЛог_Аналитика

Для каждой роли пользователя можно настроить список разрешений, отвечающий за то, какие действия доступны обладателям настраиваемой роли в различных вкладках ИндексЛог_Аналитика. Базовая настройка разрешений предполагает три варианта: полный доступ, только чтение и подробная настройка.

5. Установка и настройка агентов ИндексЛог

Агент ИндексЛог — это унифицированный способ сбора и отправки данных в ИндексЛог_Поиск. К каждому агенту привязывается политика, в которой указаны интеграции для источников данных. Агент ИндексЛог может отслеживать хост, на котором установлен, а также собирать и перенаправлять данные с удаленных источников.

Загрузить дистрибутив агента можно из репозитория https://gitlab.pozitis.ru/indexlog/agents.

Интеграции ИндексЛог представляют собой набор шаблонов для компонентов стека, позволяющий удобно подключать новые источники данных. В интеграцию входит готовая конфигурация агента ИндексЛог, готовые пайплайны для обработки входящих данных и дэшборды для ИндексЛог_Аналитика. Управление агентами и их политиками осуществляется через интерфейс ИндексЛог_Аналитика.

5.1. Установка сервера централизованного управления агентами

Сервер централизованного управления агентами устанавливается на хост, к которому есть доступ со всех машин, на которые планируется устанавливать агенты. Для установки требуются TLS-сертификаты, которые автоматически генерируются при первом запуске ИндексЛог_Поиск.

Для начала, откройте Вкладку Управление > Сервер ЦУА > Настройки. Добавьте хост для сервера ЦАУ и укажите адрес на который будет установлен сервер ЦУА. Укажите адреса серверов ИндексЛог_Поиск, куда будут отправляться данные с агентов.

Откройте вкладку ЦУА > Агенты. Нажмите Добавить сервер ЦУА. Выполните инструкции базового или подробного уровня для установки сервера централизованного управления агентами.

5.2. Установка унифицированного агента ИндексЛог

Для установки агентов ИндексЛог требуется наличие сервера ЦУА и регистра пакетов ИндексЛог. Для настройки защищенного соединения потребуется CA, который выдается ограничацией, либо генерируется инструментами ИндексЛог_Поиск.

На каждый хост может быть установлен только один агент ИндексЛог. Для работы с веб-интерфейсом установки агента в ИндексЛог_Аналитика пользователь должен иметь все разрешения на работу с агентами.

Откройте вкладку Агенты в ИндексЛог_Аналитика и нажмите Добавить агент. Во всплывающем окне выберите политику, которую хотите использовать на устанавливаемом агенте.

Загрузите на хост дистрибутив агента и распакуйте его. Откройте директорию с файлами агента и выполните команду установки, выведенную в интерфейсе ИндексЛог_Аналитика. Агент установится с необходимыми для вашего кластера установками. Агент будет автоматически запускаться как служба при перезапуске хоста.

Если при попытке установить агента, вы сталкиваетесь с ошибкой "x509: certificate signed by unknown authority", это может быть связано с тем, что используются сертификаты, подписанные инструментами ИндексЛог. Проблему можно решить добавлением параметра --insecure.

6. Настройки безопасности стека Elastic

6.1. Шифрование соединения между браузером пользователя и сервером ИндексЛог_Аналитика

Для шифрования соединения между браузером пользователя и сервером Kibana используется серверный сертификат и приватный ключ.

Следующая инструкция описывает процесс создания запроса на подписание сертификата (CSR). Он содержит информацию, которую центр сертификации (CA) использует для генерации сертификата. Для подписания можно использовать доверенный (публичный или внутренний корпоративный) центр сертификации или собственный CA, которой можно сгенерировать инструментами ИндексЛог_Поиск. Рекомендуется использовать доверенный центр сертификации для продуктивных сред, и прибегать к собственному CA только для целей разработки или пилотных проектов.

1. Сгенерируйте запрос на создание сертификата и ключ к kibana-server

```
./bin/elasticsearch-certutil csr -name kibana-server -dns example.com,www.example.com
```

Запрос имеет common name (CN) kibana-server и адреса SAN example.com и www.example.com.

Командра создает zip-архив csr-bundle.zip. По умолчанию он имеет следующее содержание:

```
/kibana-server
|_ kibana-server.csr
|_ kibana-server.key
```

2. Разархивируйте csr-bundle.zip и получите файлы kibana-server.csr и kibana-server.key. Это неподписанный сертификат и незашифрованный ключ.

3. Отправьте файл kibana-server.csr в свой собственный или доверенный центр сертификации, чтобы получить подписанный сертификат. Сертификат может быть в различных форматах, например, kibana-server.crt.

4. Пропишите в файле конфигурации kibana.yml следующие параметры:

```
server.ssl.enabled: true
server.ssl.certificate: $KBN_PATH_CONF/kibana-server.crt
server.ssl.key: $KBN_PATH_CONF/kibana-server.key
```

5. Запустите сервер ИндексЛог_Аналитика

6.2. Аутентификация пользователей

6.2.1 Механизмы аутентификации

Стек Elastic поддерживает различные механизмы аутентификации, в том числе:

6.2.1.1 Внутренний механизм аутентификации ИндексЛог_Поиск

Данные пользователей хранятся в отдельном индексе в ИндексЛог_Поиск. Вход осуществляется по логину и паролю пользователя. Не требует дополнительной настройки для применения, только внутренние инструменты управления пользователями.

Для создания и обновления пользователей используются следующие API запросы:

```
POST /_security/user/<username>
PUT /_security/user/<username>
```

Запрос может выглядеть следующим образом:

```
POST /_security/user/jacknich
{
 "password" : "l0ng-r4nd0m-p@ssw0rd",
 "roles" : [ "admin", "other_role1" ],
 "full_name" : "Jack Nicholson",
 "email" : "jacknich@example.com",
 "metadata" : {
  "intelligence" : 7
 }
}
```

В запросе присутствуют следующие параметры:

- enabled (boolean): показывает, включено ли использование пользователя true;
- email (string): адрес электронной почты пользователя;
- full_name (string): полное имя пользователя;
- metadata (object): любые метаданные, которые нужно закрепить за пользователем;
- password (обязательно, string): пароль минимум из 6 символов;
- roles (обязательно, list): набор ролей пользователя. Роли определяют права доступа пользователя. Чтобы создать пользователя без ролей, укажите в качестве значения пустой список: [].

Успешный запрос возвращает документ с информацией о том, что документ был создан или обновлен. При обновлении информации о пользователе, поле created имеет значение false.

```
{
 "created": true
}
```

6.2.1.1 Файловая аутентификация

Внутренний механизм аутентификации, где вход осуществляется по логину и паролю. Список пользователей хранится в отдельном файле на каждой ноде ИндексЛог_Поиск.

Файлы users и users_roles могут быть изменены только локально на каждой ноде и не возможности имеют централизованной конфигурации. Это значит, что в кластере из нескольких нод на каждой из них к этим файлам должны быть применены одни и те же изменения.

Процесс настройки файловой аутентификации выглядит следующим образом:

1. (опционально) Добавьте механизм файловой аутентификации в конфиг-файл elasticsearch.yml, например:

```
xpack:
 security:
  authc:
   realms:
    file:
     file1:
      order: 0
```

2. Перезапустите ИндексЛог_Поиск

3. В файл ES_PATH_CONF/users добавьте данные пользователей. В каждой строчке файла содержатся данные пользователей, а именно имя пользователя и пароль в формате hashed и salted.

```
rdeniro:$2a$10$BBJ/ILiyJ1eBTYoRKxkqbuDEdYECplvxnqQ47uiowE7yGqvCEgj9W
alpacino:$2a$10$cNwHnElYiMYZ/T3K4PvzGeJ1KbpXZp2PfoQD.gfaVdImnHOwIuBKS
```

4. Добавьте сведения о ролях пользователей в файле ES_PATH_CONF/user_roles. В каждой строке укажите через запятую список пользователей, которым присваивается роль.

```
admin:rdeniro
power_user:alpacino,jacknich
user:jacknich
```