|  |  |
| --- | --- |
| УТВЕРЖДЕН |  |
| ЕСБМ.62.01.29.684.001.91-ЛУ |  |

**Программа «Peraspera Платформа»**

**Описание функциональных характеристик программного обеспечения и информация, необходимая для установки и эксплуатации программного обеспечения**

ЕСБМ.62.01.29.684.001.91

Листов 37

Москва

2021

аннотация

Данный документ содержит описание функциональных характеристик программы «Peraspera Платформа» и информацию, необходимую для установки и эксплуатации программы «Peraspera Платформа». В нем содержится описание функциональных возможностей программы «Peraspera Платформа» версии 1.0 и описание программно-технической среды, необходимой для установки, запуска и исполнения программы «Peraspera Платформа». В нем содержится также общее описание параметров программы «Peraspera Платформа», которые могут использоваться для настройки программы «Peraspera Платформа» в соответствии с требованиями конкретной вычислительной установки.

ПРИНЯТЫЕ ТЕРМИНЫ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

**API** (Application Programming Interface) - прикладной программный интерфейс программы или приложения (библиотеки классов и процедур), с помощью которого одна программа/приложение может взаимодействовать с другой.

**DaemonSet** - контроллер, основным назначением которого является запуск подов на всех узлах кластера: при добавлении/удалении узла DaemonSet автоматически добавляет/удаляет под на этом узле.

**ИАС** – информационно-аналитическая система.

**Икс индекс качества сайта -** является метрическим параметром, который определяет насколько полезен интернет-ресурс для аудитории, по версии Яндекса. Основные факторы, влияющие на значение ИКС: количество посетителей сайта; степень доверия к веб-ресурсу со стороны Яндекса и посетившей его аудитории.

**Краулер -** [программа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), являющаяся составной частью [поисковой системы](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B8%D1%81%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) и предназначенная для перебора страниц [Интернета](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82) с целью занесения информации о них в базу данных поисковика.

**Облачные вычисления** (Cloud Computing) — технология распределённой обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис.

**Оркестрация** - компьютерное управление оркестром сервисов внутри единой информационные системы, каждый из которых отвечает только за одну определённую задачу, а общение осуществляется через сетевые порты и общие каталоги. Задачей облачной операционной системы является оркестрация вычислительных ресурсов и хранилища данных.

**ЦОД** – центр обработки данных.

**Whois** - [сетевой протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI), базирующийся на протоколе [TCP](https://ru.wikipedia.org/wiki/TCP) ([порт](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D1%82_(TCP/UDP)) 43). Основное применение - получение регистрационных данных о владельцах [доменных имён](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D0%BC%D1%8F), [IP-адресов](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) и [автономных систем](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BD%D0%BE%D0%BC%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_(%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82)).

**Содержание**

[1 ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ «Peraspera Платформа» 6](#_Toc68087134)

[1.1 Общее описание функциональных характеристик 6](#_Toc68087135)

[1.2 Описание функциональных характеристик компонент программы «Peraspera Платформа» 7](#_Toc68087136)

[1.2.1 Подсистема сбора данных 9](#_Toc68087137)

[1.2.2 Подсистема потоковой обработки 9](#_Toc68087138)

[1.2.3 Подсистема хранения данных 10](#_Toc68087139)

[1.2.4 Подсистема аналитики, включающая модели машинного обучения и сервисы аналитики 11](#_Toc68087140)

[1.2.5 Подсистема мониторинга 12](#_Toc68087141)

[1.2.6 Подсистема управления оркестрацией 12](#_Toc68087142)

[1.2.7 Подсистема информационной безопасности 13](#_Toc68087143)

[1.3 Описание функциональных характеристик приложений, запускаемых в среде программы «Peraspera Платформа» 13](#_Toc68087144)

[1.3.1 Сбор и обработка поисковых ссылок на заданную тематику 14](#_Toc68087145)

[1.3.2 Сбор и предварительная обработка данных полученных по поисковым ссылкам 14](#_Toc68087146)

[1.3.3 Сбор и предварительная обработка данных, получаемых из социальных сетей 15](#_Toc68087147)

[1.3.4 Предварительная обработка (классификация) изображений с целью принятия решения о соответствии изображения заданному критерию для хранения с целью последующей детальной обработки 15](#_Toc68087148)

[1.3.5 Обработка изображений с целью выделения текстовой информации 15](#_Toc68087149)

[1.3.6 Обработка загруженных данных с целью выделения именованных сущностей (NER) 16](#_Toc68087150)

[1.3.7 Обработка загруженных данных с целью выделения информации по сформулированным критериям 16](#_Toc68087151)

[1.3.8 Обработка загруженных данных с целью выделения специальных идентификаторов документа или сущностей 16](#_Toc68087152)

[1.3.9 Обмен данных со смежными системами с помощью API 17](#_Toc68087153)

[1.3.10 Агрегация собранной информации для группировки сведений, касающихся каждой выделенной сущности 17](#_Toc68087154)

[1.3.11 Предоставление данных по результатам аналитических расчетов. 17](#_Toc68087155)

[1.3.12 Выгрузка данных в смежные системы 17](#_Toc68087156)

[1.3.13 Отображение данных в веб-интерфейсе 17](#_Toc68087157)

[2 ИНФОРМАЦИЯ, НЕОБХОДИМАЯ ДЛЯ УСТАНОВКИ И ЭКСПЛУАТАЦИИ программы «Peraspera Платформа» 18](#_Toc68087158)

[2.1 Информация, необходимая для установки 18](#_Toc68087159)

[2.1.1 Состав инсталляционного пакета 18](#_Toc68087160)

[2.1.2 Требования к квалификации специалиста, выполняющего установку Программы 19](#_Toc68087161)

[2.1.3 Требования к оборудованию 19](#_Toc68087162)

[2.1.4 Требования к системному программному обеспечению 20](#_Toc68087163)

[2.1.5 Порядок установки 20](#_Toc68087164)

[2.2 Информация, необходимая для эксплуатации 34](#_Toc68087165)

# ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ «Peraspera Платформа»

## Общее описание функциональных характеристик

Программа «Peraspera Платформа» — готовая универсальная среда для быстрой установки и исполнения прикладного программного обеспечения или сервисов, использующих технологии и методы работы с большими данными (Big Data), машинного обучения и искусственного интеллекта, ориентированных на решение конкретных информационно-аналитических задач.

Приложения, запускаемые в среде программы «Peraspera Платформа», осуществляют сбор, обновление, агрегацию и обработку входной информации из различных источников данных: открытых данных из сети интернет и данных, поступивших из локальных источников; выполнение аналитических, прогнозных работ; создание интерактивных отчетов для принятия оптимальных управленческих и бизнес-решений заказчиков.

В отличии от разработки информационно-аналитических систем с нуля в среде программы «Peraspera Платформа» уже реализован типовой технологический процесс быстрого конфигурирования информационно-аналитической системы под заказ и создана система управления этим процессом.

Применение программы «Peraspera Платформа» позволяет сконцентрировать усилия специалистов заказчика на решении бизнес-задач, а не на создании инфраструктуры – все системотехнические вопросы уже решены.

В созданные компоненты программы «Peraspera Платформа» заложены принципы вертикальной и горизонтальной масштабируемости, а также возможность расширения состава решаемых платформой и прикладными приложениями бизнес-задач.

Также Программа «Peraspera Платформа» имеет информационные и программные интерфейсы к соответствующим интегрируемым с платформой программным продуктам, реализующим соответствующие методы. Обеспечивается возможность включения новых методов в состав платформы, как run-time компонентов для использования при решении прикладных задач.

За счёт координации всех технологических процессов на основе контейнеризации микросервисов и обработки данных в потоковом и микропакетном режимах достигается высокий уровень параллелизма обработки, чем обеспечивается заданная производительность целевой ИАС путем масштабирования базового ЦОД под заказ.

Программа «Peraspera Платформа» может быть развёрнута в облачной среде, что снимает необходимость в собственных технических средствах.

## Описание функциональных характеристик компонент программы «Peraspera Платформа»

В состав компонент среды программы «Peraspera Платформа» входят:

* Подсистема сбора данных.
* Подсистема потоковой обработки данных.
* Подсистема хранения данных.
* Подсистема аналитики, включающая модели машинного обучения и сервисы аналитики.
* Подсистема мониторинга.
* Подсистема управления оркестрацией.
* Подсистема информационной безопасности.

Схема взаимодействия компонент программы «Peraspera Платформа» представлена на Рисунке 1.

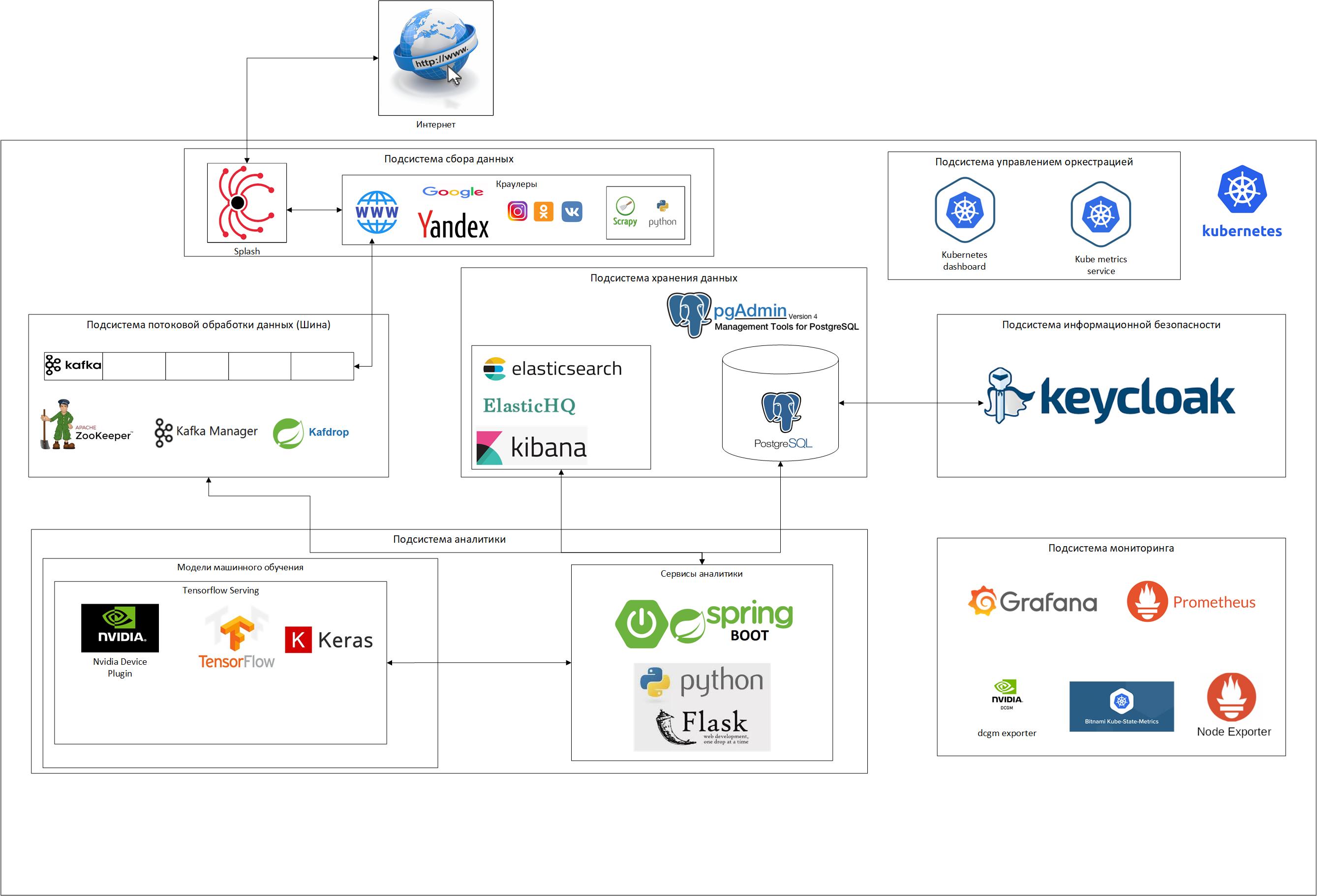


Рисунок 1 - Схема взаимодействия компонент программы «Peraspera Платформа»

### Подсистема сбора данных

Подсистема сбора данных обеспечивает получение требуемых данных из различных источников данных. В качестве источников данных могут выступать открытые источники данных сети Интернет, отдельные файлы, базы данных входящие в состав корпоративных информационных систем наших клиентов. Для сбора информации из сети Интернет используется ПО.

**Splash** - компактный браузер для отрисовки javascript кода на скачиваемых краулерами страницах, использование которого позволяет выполнять параллельную обработку нескольких веб-страниц, получать HTML-результаты и / или делать скриншоты. Основной задачей подсистемы является решение задачи сбора необходимых данных из гетерогенных источников и представление их на дальнейшую обработку.

### Подсистема потоковой обработки

Подсистема потоковой обработки данных обеспечивает интеграцию данных, поступающих из подсистемы сбора данных, а также преобразование интегрированных данных к виду, пригодному для передачи данных в подсистему хранения данных для предоставления сервисов для расчета показателей на основе интегрированных данных, значения которых используются подсистемой аналитики и другими прикладными подсистемами. Подсистема реализуется на основе свободно распространяемого ПО:

**Apache Kafka** - распределённый программный брокер сообщений, проект с открытым исходным кодом, разрабатываемый в рамках фонда Apache. Поддерживает транзакционность при работе с потребителями и поставщиками событий.

**CMAK (Cluster Manager for Apache Kafka)**- Инструмент для управления Apache Kafka, поддерживает управление несколькими кластерами, выбор предпочтительной реплики, переназначение реплики и создание темы.

**Apache Zookeeper** - cервис-координатор, который обеспечивает распределенную синхронизацию небольших по объему данных (конфигурационная информация, пространство имен) для группы приложений. Apache Zookeeper представляет распределенное хранилище ключ-значение (key-value store), гарантирующий надежное консистентное (consistency) хранение информации за счет синхронной репликации между узлами, контроля версий, механизма очередей (queue) и блокировок (lock). За счет использования оперативной памяти и масштабируемости обладает высокой скоростью.

**Kafdrop** - Веб-интерфейс для просмотра тем Apache Kafka и групп потребителей. Kafdrop предоставляет следующие ключевые функции:

* Просмотр брокеров Apache Kafka - назначение тем и разделов, а также статус контроллера.
* Просмотр тем - количество разделов, состояние репликации и настраиваемая конфигурация.
* Просмотр сообщений - JSON, обычный текст и кодировка Avro.

**KSQL** – движок SQL с открытым исходным кодом для [Apache Kafka](https://www.bigdataschool.ru/bigdata/apache-kafka-administrator-course.html), обеспечивает простой, но мощный интерактивный SQL интерфейс для потоковой обработки на Apache [Kafka](https://www.bigdataschool.ru/wiki/kafka).

### Подсистема хранения данных

Подсистема хранения данных создана на основе Data Lake, включающего распределенную файловую систему, а также NoSQL и реляционную базу данных, обеспечивает однократное размещение конкретного экземпляра каждой сущности на постоянное хранение и его последующее использование без изменений, организацию надежного хранения исторических данных, поступающих из Подсистемы потоковой обработки данных, значений рассчитанных показателей, а также отчетов. Подсистема реализуется на основе свободно распространяемого ПО:

**Elasticsearch** - высокомасштабируемая распределенная поисковая система полнотекстового поиска и анализа данных, работающая в режиме реального времени. Позволяет хранить, искать и анализировать большие объемы данных. Обычно используется в качестве базового механизма/технологии, помогая приложениям со сложными функциями поиска. Также предоставляет возможности обработки текста и полнотекстового поиска (поиск по всем выражениям во всех документах, доступных в базе данных) и возможность аналитической обработки данных. Elasticsearch предлагает широкий спектр агрегаторов для аналитики. Системы агрегации Elasticsearch могут применяться для различных типов данных.

**Elasticsearch HQ** - приложение с открытым исходным кодом, которое предлагает упрощенный интерфейс для управления (полный контроль над кластерами, узлами, индексами и сопоставлениями Elasticsearch) и мониторинга кластеров Elasticsearc.

**Kibana** - инструмент для визуализации на основе браузера с открытым исходным кодом, который в основном используется для анализа большого объема журналов в виде линейного графика, гистограммы, круговых диаграмм, тепловых карт, карт регионов, координатных карт, датчиков, целей, временных шкал и т. д. Предлагает использование фильтров и поисковых запросов для получения необходимой информации для конкретного ввода из панели мониторинга или инструмента визуализации Все данные в форме визуализации и панели мониторинга могут быть преобразованы в отчеты (формат CSV), встроены в код или в виде URL-адресов для совместного использования с другими.

**PostgreSQL** - [свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) [объектно-реляционная](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) [система управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) (СУБД), может применяться в среде клиент-сервер, поддерживает большую часть стандарта SQL и предлагает множество современных функций:

* сложные запросы;
* внешние ключи;
* триггеры;
* изменяемые представления;
* транзакционная целостность;
* многоверсионность.

**pgAdmin -** это платформа с открытым исходным кодом для администрирования и разработки для PostgreSQL и связанных с ней систем управления базами данных. Платформа написана на Python и jQuery и поддерживает все функции PostgreSQL.

### Подсистема аналитики, включающая модели машинного обучения и сервисы аналитики

Подсистема аналитики, включающая модели машинного обучения и сервисы аналитики, представляет собой среду исследования и формирования отчетов на основе информации из подсистемы хранения данных и/или подсистемы потоковой обработки данных автоматически согласно регламенту или по запросам пользователей. Данная подсистема включает в свой состав нейросетевые модели и средства, как уже обученные на выполнение определенного типа задач, так и предоставляющие возможность самостоятельного анализа данных, средства из состава статистической математики, предоставляющие возможность типизации, кластеризации и другие модели, востребованные аналитиками данных. Подсистема реализуется на основе свободно распространяемого ПО:

**nvidia Device Plugin** - – структура плагинов устройства, предоставляемых как Daemonset, который автоматически отображает количество графических процессоров на каждом узле кластера, следит за состоянием графических процессоров, запускает контейнеры с поддержкой GPU в кластере Kubernetes.

**TensorFlow Serving** - система для развертывания TensorFlow-моделей, обеспечивающая автоматический батчинг, горячую замену моделей и версионирование, возможность обработки параллельных запросов. Поддерживает горячую замену моделей. Сервер моделей постоянно сканирует указанный при запуске путь на наличии новых моделей и при нахождение новой версии автоматически загружает эту версию. Основной частью TensorFlow Serving является сервер моделей (Model Server), который загружает модель из пути, указанного при запуске, и начинает слушать указанный порт. Сервер общается с клиентами через вызовы удаленных процедур, используя библиотеку gRPC, что позволяет создавать клиентское приложение на любом языке, поддерживающем gRPC.

### Подсистема мониторинга

Подсистема мониторинга обеспечивает автоматизацию процессов сбора в реальном времени и хранения информации о параметрах функционирования технических средств, системных программных средств (операционные системы), СУБД, программных средств промежуточного уровня (middleware), их оперативный контроль (мониторинг), оперативно-техническое управление функционированием и администрирование эксплуатирующим персоналом. Подсистема реализуется на основе свободно распространяемого ПО:

**DCGM Exporter -** средство экспорта для Prometheus, позволяющее отслеживать состояние и получать метрики от графических процессоров. Использует DCGM с помощью привязок Go для сбора телеметрии графического процессора и предоставляет метрики графического процессора Prometheus с помощью конечной точки http ( /metrics). DCGM Exporter можно использовать отдельно или развернуть как часть [оператора графического процессора NVIDIA](https://github.com/NVIDIA/gpu-operator).

**Grafana** - инструмент с открытым исходным кодом для визуализации данных из различных систем сбора статистики. Используется для представления в графическом виде временных рядов и текстовых данных.

**Kube State Metrics** - утилита для прослушивания сервера Kubernetes API, помогает в генерации метрик о состоянии объектов. Фокусируется на работоспособности различных объектов внутри кластера, включая узлы, поды и развертывания. Показатели kube-state-metrics публикуют метрики в качестве показателей statsd для системы мониторинга.

**Node Exporter** - экспортер Prometheus для сбора данных о состоянии сервера с подключаемыми коллекторами метрик. Он позволяет измерять различные ресурсы машины, такие как использование памяти, диска и процессора.

**Prometheus** - система мониторинга серверов и программ с открытым исходным кодом. Обеспечивает мониторинг различных систем: серверов, баз данных, отдельных виртуальных машин и т.д. путем периодического скрейпинга целевых объектов. Prometheus извлекает метрики через HTTP-вызовы к определенным конечным точкам, указанным в конфигурации Prometheus. Данные от целевых объектов хранятся в базе данных временных рядов.

### Подсистема управления оркестрацией

Подсистема управления оркестрацией отвечает за координирование всех технологических процессов, включая потоковую среду обработки данных, управление микросервисами и администрирование, обеспечивает автоматизацию деятельности эксплуатирующего персонала по управлению функционированием приложений, а также управление взаимодействием между приложениями. Подсистема реализуется на основе свободно распространяемого ПО:

**Kubernetes Dashboard** - пользовательский веб-интерфейс Kubernetes, используемый для развертывания контейнерных приложений в кластере Kubernetes, устранения неполадок контейнерного приложения и управления ресурсами кластера. Также может использоваться для получения обзора приложений, работающих в кластере, а также для создания или изменения отдельных ресурсов Kubernetes, таких как развертывания, задания, DaemonSets и т. д.

**Kubernetes Metrics Service** - масштабируемый источник метрик ресурсов контейнера для встроенных конвейеров автомасштабирования Kubernetes. собирает метрики ресурсов из Kubelets и предоставляет их в Kubernetes apiserver через [Metrics API](https://github.com/kubernetes/metrics) для использования в [Horizontal Pod Autoscaler](https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/) и [Vertical Pod Autoscaler](https://github.com/kubernetes/autoscaler/tree/master/vertical-pod-autoscaler).

### Подсистема информационной безопасности

Подсистема информационной безопасности обеспечивает заданное состояние информационной безопасности путем мониторинга информационной безопасности и управления как специальными, так и встроенными в функционал программно-техническими средствами обеспечения информационной безопасности. Подсистема реализуется на основе свободно распространяемого ПО:

**Keycloak** - продукт с открытым кодом для реализации [single sign-on](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F_%D0%B5%D0%B4%D0%B8%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B2%D1%85%D0%BE%D0%B4%D0%B0) с возможностью управления идентификацией и доступом, нацелен на современные применения и сервисы. Предлагает такие функции, как единый вход (SSO), брокерская идентификация и социальный вход в систему, федерация пользователей, клиентские адаптеры, консоль администратора и консоль управления учетными записями

## Описание функциональных характеристик приложений, запускаемых в среде программы «Peraspera Платформа»

Назначением приложений, запускаемых в среде программы «Peraspera Платформа», являются сбор и обработка потоков данных, получаемых из открытых источников в сети интернет, сбор и обработка потоков данных, получаемых с помощью специальных коннекторов к закрытым реляционным и нереляционным БД и другим хранилищам информации. Разворачиваемые в среде программы «Peraspera Платформа» приложения в её состав не входят, а разрабатываются под конкретную задачу. Как правило, эти приложения предоставляют следующие возможности:

* Сбор и обработка поисковых ссылок на заданную тематику.
* Сбор и предварительная обработка данных, получаемых по поисковым ссылкам.
* Сбор и предварительная обработка данных, получаемых из социальных сетей.
* Предварительная обработка (классификация) изображений с целью принятия решения о соответствии изображения заданному критерию для хранения с целью последующей детальной обработки.
* Обработка изображений с целью выделения содержащейся в них текстовой информации.
* Обработка загруженных данных с целью выделения содержащихся в них именованных сущностей (NER).
* Обработка загруженных данных с целью выделения информации по сформулированным критериям.
* Обработка загруженных данных с целью выделения специальных идентификаторов документа или сущностей.
* Обмен данных со смежными системами с помощью API.
* Агрегация собранной информации для группировки сведений касающихся каждой выделенной сущности.
* Предоставление данных по результатам аналитических расчетов.
* Выгрузка данных в смежные системы.
* Отображение данных в веб-интерфейсе.

### Сбор и обработка поисковых ссылок на заданную тематику

Сбор и обработка поисковых ссылок осуществляются с помощью краулеров из поисковых систем Яндекс, Google и специализированных краулеров для поиска в социальных сетях. Для уточнения поисковых запросов может быть использован внутренний язык запросов поисковой системы. Накопление результатов осуществляется либо в таблице реляционной БД, либо в формате сообщений Apache Kafka.

Дополнительно результаты поисковых запросов могут быть раздельно сгруппированы по типам, таким как обычные сайты, социальные сети, сервисы хостинга видео, и т.д.

Также может быть осуществлена фильтрация и обработка результатов по различным критериям, таким как SSL-сертификат сайта, дата публикации информации по ссылке, индекс качества сайта, на котором размещена информация, данные о домене сайта из БД whois и т.д.

### Сбор и предварительная обработка данных полученных по поисковым ссылкам

При наличии ссылок на данные, полученные с помощью поисковых запросов, производится передача этих ссылок в экземпляр краулера для последующей выгрузки. Краулеры обеспечивают анонимный обход сайтов по ссылке, выгружая информацию и передавая ее в конвейер предварительной обработки. Краулеры обеспечивают балансировку нагрузки (одновременное количество подключений к сайту, скорость потока выгрузки) и отсутствие повторных обходов страницы. При этом отслеживаются случаи динамически генерируемых страниц по одному адресу, в этом случае повторная выгрузка страницы разрешается.

Каждая полученная страница может быть передана на дальнейшую обработку либо в исходном виде, либо с предварительной обработкой в виде частичной очистки html-тегов (таких как скрипты, стили), либо с полной очисткой от html-тегов в виде текстовой информации.

Помимо очистки тегов, возможна так же очистка от несущественной информации (меню, реклама, элементы разметки) с выделением только содержательной части содержимого.

Накопление результатов осуществляется либо в таблице реляционной БД, либо в формате сообщений Apache Kafka, либо в индексе Elasticsearch.

### Сбор и предварительная обработка данных, получаемых из социальных сетей

При наличии ссылок на данные, размещенные в социальных сетях, эти данные (данные из социальных сетей Вконтакте, Одноклассники, Мой мир, Instagram, Twitter, Livejournal, мессенджера Telegram, видеохостинга YouTube, поисковых систем Яндекс, Google) выгружаются с помощью специализированных краулеров для социальных сетей, использующих по возможности документированные возможности API социальных сетей. Аналогично данным с сайтов, данные из социальных сетей могут быть предварительно обработаны.

Накопление результатов осуществляется либо в таблице реляционной БД, либо в виде сообщений Apache Kafka, либо в индексе Elasticsearch.

### Предварительная обработка (классификация) изображений с целью принятия решения о соответствии изображения заданному критерию для хранения с целью последующей детальной обработки

Краулеры обеспечивают выгрузку изображений, на которые есть ссылки с исходных страниц. Для принятия решения о целесообразности дальнейшей обработки изображения используется быстрая классификация с помощью GPU с использованием предварительно обученной модели классификации. В случае, если изображение признано потенциально пригодным для дальнейшего анализа, оно сохраняется либо в распределенную файловую систему, либо в виде сообщения Apache Kafka.

### Обработка изображений с целью выделения текстовой информации

При поступлении изображения в обработку, производится распознавание текстовой информации с помощью алгоритмов оптического распознавания символов. Полученный текстовый результат проходит этап корректировки ошибок по словарям и по специальным алгоритмам, таким как контрольные суммы для численных идентификаторов с контрольной суммой.

Накопление результатов осуществляется либо в таблице реляционной БД, либо в виде сообщений Apache Kafka, либо в индексе Elasticsearch.

### Обработка загруженных данных с целью выделения именованных сущностей (NER)

При поступлении текстового сообщения в обработку производится выделение непрерывных фрагментов текста, представляющего собой именованную сущность, например, персоны, локации, организации, суммы, даты и прочие устойчивые понятия.

Выделение именованных сущностей производится с помощью предварительно обученных алгоритмов машинного обучения, основанных на использовании технологий Deep Learning и Machine Learning: рекуррентных нейросетях, сверточных нейросетях, алгоритмах нечеткого поиска, линейной регрессии, деревьев решений, временных рядов, группировки, кластеризации..

Накопление результатов осуществляется либо в таблице реляционной БД, либо в виде сообщений Apache Kafka, либо в индексе Elasticsearch.

### Обработка загруженных данных с целью выделения информации по сформулированным критериям

При поступлении текстового сообщения в обработку производится выделение фрагментов текста, позволяющих принять решение о соответствии данного текста какому-либо предварительно заданному критерию, такому как отношение информации к определенному классу, отношение описываемого события к определенному классу.

Принятие решения о соответствии информации заданному критерию производится с помощью либо нечеткого поиска по предварительно заданному словарю, либо с помощью предварительно обученных алгоритмов машинного обучения.

Накопление результатов осуществляется либо в таблице реляционной БД, либо в виде сообщений Apache Kafka, либо в индексе Elasticsearch.

### Обработка загруженных данных с целью выделения специальных идентификаторов документа или сущностей

При поступлении текстового сообщения в обработку производится выделение непрерывных фрагментов текста, соответствующих заранее заданным идентификаторам, имеющим четкую проверяемую структуру (например, ИНН или ОГРН, которые имеют заданную длину числовой последовательности и разряд для контрольной суммы).

Принятие решения о соответствии информации заданному критерию производится либо с помощью заранее заданных регулярных выражений, либо с помощью нечеткого поиска по предварительно заданному словарю, либо с помощью предварительно обученных алгоритмов машинного обучения.

Накопление результатов осуществляется либо в таблице реляционной БД, либо в виде сообщений Apache Kafka, либо в индексе Elasticsearch

### Обмен данных со смежными системами с помощью API

С целью обогащения данных из сторонних систем, поддерживается загрузка данных с помощью стороннего API.

Накопление результатов осуществляется либо в таблице реляционной БД, либо в виде сообщений Apache Kafka, либо в индексе Elasticsearch.

### Агрегация собранной информации для группировки сведений, касающихся каждой выделенной сущности

По каждому собранному идентификатору, либо по каждой собранной сущности может быть произведена агрегация всех данных, имеющих отношение к данной сущности, с целью получения сводной информации в структурированном виде.

Агрегация производится по заранее заданным алгоритмам.

Накопление результатов осуществляется либо в таблице реляционной БД, либо в виде сообщений Apache Kafka, либо в индексе Elasticsearch.

### Предоставление данных по результатам аналитических расчетов.

Результаты аналитических отчетов могут быть получены из внутренних источников информации системы и выгружены в таблицы csv или документы формата xlsx.

### Выгрузка данных в смежные системы

Для интеграции с внешними системами имеется возможность реализации REST API для организации информационного обмена между программой «Peraspera Платформа» и внешними системами.

Доступ к REST API осуществляется по протоколу https c использованием авторизации и токенов безопасности.

### Отображение данных в веб-интерфейсе

Для визуального отображения используется веб-интерфейс, в котором информация отображается в заранее заданном формате. Доступ к веб-интерфейсу осуществляется по протоколу https c использованием внутренней авторизации.

# ИНФОРМАЦИЯ, НЕОБХОДИМАЯ ДЛЯ УСТАНОВКИ И ЭКСПЛУАТАЦИИ программы «Peraspera Платформа»

## Информация, необходимая для установки

### Состав инсталляционного пакета

В состав инсталляционного пакета программы «Peraspera Платформа» входит инсталлятор, который устанавливает указанное в нижеприведенной таблице (Таблица 1) свободно распространяемое ПО, необходимое для функционирования программы «Peraspera Платформа».

Таблица 1 – Состав инсталляционного пакета программы «Peraspera Платформа»

| **№** | **Подсистема** | **Свободно распространяемое ПО** | **Назначение** |
| --- | --- | --- | --- |
| 1 | Подсистема сбора данных | Splash | Обеспечение сбора и доставки данных |
| 2 | Подсистема потоковой обработки данных | Apache Kafka | Обеспечение взаимодействие компонент |
| CMAK |
| Apache Zookeeper |
| Kafdrop |
| KSQL |
| 3 | Подсистема управления оркестрацией | Kubernetes Dashboard | Обеспечение оркестрации сервисов |
| Kubernetes Metrics Service |
| 4 | Подсистема информационной безопасности | Keycloak | Обеспечение информационной безопасности |
| 5 | Подсистема хранения данных | Elasticsearch | Базы данных |
| Elasticsearch HQ |
| Kibana |
| PostgreSQL |
| pgAdmin |
| 6 | Подсистема аналитики, включающая модели машинного обучения и сервисы аналитики | NVIDIA Device Plugin | Обеспечение обработки данных |
| TensorFlow Serving |
| 7 | Подсистема мониторинга | DCGM Exporter | Обеспечение мониторинга |
| Grafana |
| Kube State Metrics |
| Node Exporter |
| Prometheus |

### Требования к квалификации специалиста, выполняющего установку Программы

Специалист должен иметь навыки работы с ОС Linux(Ubuntu), Docker, Kubernetes, протокол безопасности OIDC (OpenID Connection).

### Требования к оборудованию

Минимальные требования к оборудованию для инсталляции и функционирования программы «Peraspera Платформа»:

1. Для корректного функционирования среды исполнения программы необходим сервер, обладающий следующими характеристиками:

* ЦПУ восьмиядерный (8 Core);
* Жесткий диск – не менее 250Gb;
* Оперативной памяти – не менее 16Gb;
* Видеокарта GeForce GTX 1080, объем памяти - 8Gb.

1. Для функционирования приложений, запускаемых в среде программы «Peraspera Платформа», необходимо оборудование, характеристики которого приведены в нижеприведенных таблицах (Таблица 2, Таблица 3).

Таблица 2 – Требования к среде установки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование виртуальной машины** | **Выделенное пространство** | **Объем памяти** | **Кол-во ядер процессора (CPUs)** | **Кол-во контроллеров (NICs)** |
| kubernetes-master-01 | 100 GB | 8 GB | 4 | 1 |
| kubernetes-master-02 | 40 GB | 8 GB | 4 | 1 |
| kubernetes-master-03 | 40 GB | 8 GB | 4 | 1 |
| perasp-node1 | 32 GB | 16 GB | 16 | 1 |
| perasp-node2 | 32 GB | 16 GB | 16 | 1 |
| perasp-node3 | 32 GB | 16 GB | 16 | 1 |
| perasp-node4 | 32 GB | 16 GB | 16 | 1 |
| perasp-node5 | 32 GB | 16 GB | 16 | 1 |
| perasp-node6 | 32 GB | 16 GB | 16 | 1 |
| perasp-node7 | 32 GB | 16 GB | 16 | 1 |
| perasp-node8 | 32 GB | 16 GB | 16 | 1 |
| perasp-node9 | 32 GB | 16 GB | 16 | 1 |
| perasp-node10 | 32 GB | 16 GB | 16 | 1 |
| perasp-node11 | 32 GB | 16 GB | 16 | 1 |
| perasp-node12 | 32 GB | 16 GB | 16 | 1 |
| perasp-node13 | 50 GB | 24 GB | 4 | 1 |
| perasp-node14 | 50 GB | 24 GB | 4 | 1 |
| perasp-node15 | 50 GB | 24 GB | 4 | 1 |
| perasp-node16 | 50 GB | 8 GB | 4 | 1 |

Таблица 3 – Требования к физическим серверам

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование сервера** | **Выделенное пространство** | **Объем памяти** | **Кол-во ядер процессора**  **(CPUs)** | **Кол-во контроллеров**  **NICs** |
| perasp-postgres-01 | 2 TB | 24 GB | 2 | 2 |
| perasp-postgres -02 | 2 TB | 24 GB | 2 | 2 |
| perasp-es-03 | 2 TB | 24 GB | 2 | 2 |
| perasp-es-04 | 2 TB | 24 GB | 2 | 2 |
| perasp-es-05 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka01 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka02 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka03 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka04 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka05 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka06 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka07 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka08 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka09 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka10 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka11 | 2 TB | 24 GB | 2 | 2 |
| perasp-kafka12 | 2 TB | 24 GB | 2 | 2 |

### Требования к системному программному обеспечению

Требования, предъявляемые к системному программному обеспечению для установки и эксплуатации Программы «Peraspera Платформа»:

* ubuntu сервер 18.04 и выше,
* docker 19.03.2 и выше,
* kubernetes 1.18.4 и выше

### Порядок установки

Для установки программы «Peraspera Платформа» необходимо:

На оборудовании, соответствующем требованиям, приведенным в пункте 2.1.3:

1. Развернуть ОС.
2. Создать кластер машин.
3. Установить Kubernetes версии не ниже 1.18.4.
4. Загрузить по ссылке <https://disk.yandex.ru/d/3HYp5t6GHUAFmQ> и распаковать архив с дистрибутивом (набором манифестов Kubernetes) в любую удобную директорию;
5. Настроить манифесты Kubernetes, скорректировав при необходимости в них значения по умолчанию (см. Таблица 4).

Таблица 4 – Значения по умолчанию настраиваемых параметров программы «Peraspera Платформа»

| **Наименование** | **Тип** | **Путь** | **Значение по умолчанию** | **Описание** |
| --- | --- | --- | --- | --- |
| **10kubernetes** | | | | |
| **10cluster-admins.yml** | | | | Описание ролей и привязки их в группу для кластера Kubernetes |
| admin\_group | ClusterRoleBinding | subjects.name | kubernetes-admin | Наименование группы пользователей на сервере авторизации Keycloak для полных прав доступа к кластеру Kubernetes |
| **20dashboard.yml** | | | | Конфигурирование и развёртывание пользовательского интерфейса Kubernetes |
| kubernetes-dashboard | Service | spec.type.externalName | iap.company.ru | Внешний домен платформы ИАП |
| kubernetes-dashboard-proxy | Deployment | spec.template.spec.containers.args.discovery-url | https://iap.company.ru:9050/auth/realms/iap | URL сервера авторизации Keycloak |
| spec.template.spec.containers.args.client-id | kubernetes | Идентификатор веб интерфейса Kubernetes в Keycloak |
| spec.template.spec.containers.args.client-secret | secret | Секретный ключ для веб интерфейса Kubernetes в Keycloak |
| spec.template.spec.containers.args.upstream-url | https://iap.company.ru | Внешний URL веб интерфейса |
| kubernetes-dashboard-proxy | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| **30metrics-server.yml** | | | | Конфигурация и развёртывание масштабируемого источника метрик ресурсов контейнера для встроенных конвейеров автомасштабирования Kubernetes |
| **20keycloak** | | | | |
| **10keycloak.yml** | | | | Конфигурирование и развёртывание системы управления идентификацией и доступом Keycloak |
| keycloak-volume-claim | PersistentVolumeClaim | spec.resources.requests.storage | 1Gi | Параметр, задающий запрос объёма хранилища для экземпляра Keycloak |
| keycloak | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| keycloak | Deployment | spec.template.spec.containers.keycloak.env.KEYCLOAK\_USER | "admin" | Имя пользователя для первого входа в Keycloak |
| spec.template.spec.containers.keycloak.env.KEYCLOAK\_PASSWORD | "admin" | Пароль для первого входа в Keycloak |
| spec.template.spec.containers.keycloak.env.DB\_VENDOR | "postgres" | Имя поставщика СУБД |
| spec.template.spec.containers.keycloak.env.DB\_ADDR | "iap.company.ru" | Адрес хоста с расположением СУБД |
| spec.template.spec.containers.keycloak.env.DB\_PORT | "5432" | Порт для подключения с СУБД |
| spec.template.spec.containers.keycloak.env.DB\_DATABASE | "keycloak" | Имя базы данных |
| spec.template.spec.containers.keycloak.env.DB\_USER | "keycloak\_user" | Имя пользователя для подключения к СУБД (Эту переменную рекомендуется хранить как Secret) |
| spec.template.spec.containers.keycloak.env.DB\_PASSWORD | "keycloak\_user\_pass" | Пароль пользователя для подключения к СУБД (Эту переменную рекомендуется хранить как Secret) |
| **30monitoring** | | | | |
| **10monitoring-namespace.yml** | | | | Описание пространства имён для реализации мониторинга |
| monitoring-volume-claim | PersistentVolumeClaim | spec.resources.requests.storage | 100Gi | Параметр, задающий запрос объёма хранилища метрик кластера |
| **20node-exporter.yml** | | | | Описание конфигурации для программы экспорта в Prometheus метрик оборудования и ОС |
| **30kube-state-metrics.yml** | | | | Описание конфигурирования и развёртывания сервиса, который слушает сервер Kubernetes API и генерирует метрики о состоянии объектов |
| **40dcgm-exporter.yml** | | | | Сборщик метрик видеокарт NVIDIA для Prometheus |
| **50prometheus-config.yml** | | | | Описание конфигурации инструмента для мониторинга и оповещения Prometheus |
| **60prometheus-server.yml** | | | | Описание конфигурации и развертывания для сервера мониторинга и оповещения Prometheus |
| prometheus | Deployment | spec.template  .containers.prometheus-proxy.args.client-id | prometheus | Идентификатор клиента Prometheus в Keycloak |
| spec.template  .containers.prometheus-proxy.args.client-secret | secret | Секретный ключ для клиента Prometheus в Keycloak |
| spec.template  .containers.prometheus-proxy.args.discovery-url | https://iap.company.ru:9050/auth/realms/iap | Секретный ключ для клиента Prometheus в Keycloak |
| spec.template  .containers.prometheus-proxy.args.encryption-key | key | Ключ для декодирования состояния сессии |
| spec.template  .containers.prometheus-proxy.args.resources | uri=/\*|roles=prometheus:admin | Шаблоны конечных точек с описанием авторизованного доступа к ним (в т.ч. и по ролям) |
| spec.template  .containers.prometheus-proxy.args.redirection-url | http://iap.company.ru:9090 | Внешний URL веб интерфейса |
| prometheus | Service | spec.externalName. | iap.company.ru | Внешний домен платформы ИАП |
| **70grafana.yml** | | | | Приложение для аналитики и интерактивной визуализации Grafana |
| grafana | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| grafana-conf | ConfigMap | data.grafana.ini.domain | iap.company.ru | Имя домена |
|  |  | data.grafana.ini.signout\_redirect\_url | https://iap.company.ru:9050/auth/realms/iap/protocol/openid-connect/logout?redirect\_uri=http://iap.company.ru:3000 | Url сервера авторизации Keycloak, вызываемый при выходе пользователя из системы |
| data.grafana.ini.auth\_url | https://iap.company.ru:9050/auth/realms/iap/protocol/openid-connect/auth | Url сервера авторизации |
| data.grafana.ini.token\_url | https://iap.company.ru:9050/auth/realms/iap/protocol/openid-connect/token | Получение токена с сервера авторизации |
| data.grafana.ini.api\_url | https://iap.company.ru:9050/auth/realms/iap/protocol/openid-connect/userinfo | Получение информации о пользователе с сервера авторизации |
| **40postgresql** | | | | |
| **10postgresql-namespace.yml** | | | | Пространство имен для СУБД PostgreSQL |
| postgresql-volume-claim | PersistentVolumeClaim | spec.resources.requests.storage | 100Gi | Параметр, задающий запрос объёма хранилища для дополнительных задач администрирования БД (например бэкапы) |
| postgresql\_admin\_role\_binding | RoleBinding | subjects.name | postgresql-admin | Наименование группы пользователей на сервере авторизации Keycloak для администрирования данного пространства имен |
| postgresql\_operator\_role\_binding | RoleBinding | subjects.name | postgresql-operator | Наименование группы пользователей на сервере авторизации Keycloak для просмотра данного пространства имен |
| **20postgresql.yml** | | | | Настройка и конфигурирование СУБД PostgreSQL |
| postgresql | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| postgresql | StatefulSet | spec.volumeClaimTemplates.spec.resources.requests.storage | 250Gi | Параметр, задающий запрос объёма хранилища для экземпляра БД |
| spec.template.spec.containers.postgresql.env.POSTGRES\_USER | postgres | Имя пользователя (Эту переменную рекомендуется хранить как Secret) |
| spec.template.spec.containers.postgresql.env.POSTGRES\_PASSWORD | postgres | Пароль (Эту переменную рекомендуется хранить как Secret) |
| pgadmin | Deployment | spec.template  .containers. pgadmin-gatekeeper.args.client-id | pgadmin | Идентификатор клиента pgAdmin в Keycloak |
| spec.template  .containers. pgadmin-gatekeeper.args.client-secret | secret | Секретный ключ для клиента pgAdmin в Keycloak |
| spec.template  .containers. pgadmin-gatekeeper.args.discovery-url | https://iap.company.ru:9050/auth/realms/iap | URL сервера авторизации Keycloak |
| spec.template  .containers. pgadmin-gatekeeper.args.encryption-key | your-key | Ключ для декодирования состояния сессии |
| spec.template  .containers. pgadmin-gatekeeper.args.resources | uri=/\*|roles=pgadmin:admin | Шаблоны конечных точек с описанием авторизованного доступа к ним (в т.ч. и по ролям) |
| spec.template  .containers. pgadmin-gatekeeper.args.redirection-url | http://iap.company.ru:5050 | Внешний URL веб интерфейса |
| **50kafka** | | | | |
| **10kafka-namespace.yml** | | | | Пространство имен для распределённого брокера сообщений Apache Kafka |
| kafka\_operator\_role\_binding | RoleBinding | subjects.name | kafka-operator | Наименование группы пользователей на сервере авторизации Keycloak для просмотра данного пространства имен |
| kafka\_admin\_role\_binding | RoleBinding | subjects.name | kafka-admin | Наименование группы пользователей на сервере авторизации Keycloak для администрирования данного пространства имен |
| **20zookeeper.yml** | | | | Описание и конфигурирование сервиса для централизованного хранения информации о конфигурации, присвоении имен и обеспечения распределенной синхронизации между приложениями (необходим для функционирования кластера Apache Kafka) |
| zookeeper-outer | Service | spec.externalName. | iap.company.ru | Внешний домен платформы ИАП |
| zookeeper | StatefulSet | spec.volumeClaimTemplates.spec.resources.requests.storage | 30Gi | Параметр, задающий запрос объёма хранилища для Apache Zookeeper |
| **30kafka.yml** | | | | Описание и конфигурирование распределённого брокера сообщений Apache Kafka |
| kafka | StatefulSet | spec.volumeClaimTemplates.spec.resources.requests.storage | 250Gi | Параметр, задающий запрос объёма хранилища для экземпляра Apache Kafka |
| **40kafka-tools.yml** | | | | Конфигурирование и развёртывание инструментов для управления кластером Apache Kafka (CMAK, Kafdrop, KSQL Server) |
| kafka-manager | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| kafdrop | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| ksql | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| kafka-manager | Deployment | spec.template.spec.containers.kafka-manager-gatekeeper.args.client-id | kafka-manager | Идентификатор клиента Apache Kafka в Keycloak |
| spec.template.spec.containers.kafka-manager-gatekeeper.args.client-secret | secret | Секретный ключ для клиента Apache Kafka в Keycloak |
| spec.template.spec.containers.kafka-manager-gatekeeper.args.discovery-url | https://iap.company.ru/auth/realms/iap | URL сервера авторизации Keycloak |
| spec.template.spec.containers.kafka-manager-gatekeeper.args.encryption-key | key | Ключ для декодирования состояния сессии |
| spec.template.spec.containers.kafka-manager-gatekeeper.args.resources | uri=/\*|roles=kafka-manager:admin | Шаблоны конечных точек с описанием авторизованного доступа к ним (в т.ч. и по ролям) |
| spec.template.spec.containers.kafka-manager-gatekeeper.args.redirection-url | http://iap.company.ru:20000 | Внешний URL веб интерфейса |
| kafdrop | Deployment | spec.template.spec.containers.kafdrop-gatekeeper.args.client-id | kafdrop | Идентификатор клиента Kafdrop в Keycloak |
|  |  | spec.template.spec.containers.kafdrop-gatekeeper.args.client-secret | secret | Секретный ключ для клиента Kafdrop в Keycloak |
|  |  | spec.template.spec.containers.kafdrop-gatekeeper.args.discovery-url | https://iap.company.ru:9050/auth/realms/iap | URL сервера авторизации Keycloak |
|  |  | spec.template.spec.containers.kafdrop-gatekeeper.args.encryption-key | key | Ключ для декодирования состояния сессии |
|  |  | spec.template.spec.containers.kafdrop-gatekeeper.args.resources | uri=/\*|roles=kafdrop:user | Шаблоны конечных точек с описанием авторизованного доступа к ним (в т.ч. и по ролям) |
|  |  | spec.template.spec.containers.kafdrop-gatekeeper.args.redirection-url | http://iap.company.ru:20010 | Внешний URL веб интерфейса |
| **60elasticsearch** | | | | |
| **10es-namespace.yml** | | | | Пространство имён для Elasticsearch |
| elasticsearch-volume-claim | PersistentVolumeClaim | spec.resources.requests.storage | 100Gi | Параметр, задающий запрос объёма хранилища для экземпляра Elasticsearch |
| es\_admin | RoleBinding | subjects.name | es\_admin | наименование группы пользователей на сервере авторизации Keycloak для администрирования данного пространства имен |
| es\_operator | RoleBinding | subjects.name | es\_operator | наименование группы пользователей на сервере авторизации Keycloak для просмотра данного пространства имен |
| **20es.yml** | | | | Развёртывание кластера Elasticsearch |
| elasticsearch | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| elasticsearch | StatefulSet | spec.template.spec.nodeSelector | elasticsearch=true | Метка узлов, на которых необходимо развернуть компоненты для кластера Elasticsearch |
|  |  | spec.template.spec.containers.env.ES\_JAVA\_OPTS | "-Xms8G -Xmx8G" | Параметры, описывающие выделение памяти для JVM Elasticsearch внутри контейнера |
| **30elastic-hq.yml** | | | | Мониторинг и администрирование кластера Elasticsearch |
| elastic-hq | Deployment | spec.template.spec.nodeselector | elasticsearch=”true” | Метка узлов, на которых необходимо развернуть компоненты для кластера Elasticsearch |
|  |  | spec.template.spec.containers  .elastic-hq-proxy  .args.client-id | elastic-hq | Идентификатор клиента ElasticHQ в Keycloak |
|  |  | spec.template.spec.containers  .elastic-hq-proxy.args.client-secret | secret | Секретный ключ для ElasticHQ в Keycloak |
|  |  | spec.template.spec.containers  .elastic-hq-proxy.args.discovery-url | https://iap.company.ru:9050/auth/realms/iap | URL сервера авторизации Keycloak |
|  |  | spec.template.spec.containers  .elastic-hq-proxy.args.encryption-key | your-key | Ключ для декодирования состояния сессии |
|  |  | spec.template.spec.containers  .elastic-hq-proxy.args.resources | uri=/\*|roles=elastic-hq:admin | Шаблоны конечных точек с описанием авторизованного доступа к ним (в т.ч. и по ролям) |
|  |  | spec.template.spec.containers  .elastic-hq-proxy.args.redirection-url | http://iap.company.ru:5000 | Внешний URL веб интерфейса |
| elastic-hq | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| **40kibana.yml** | | | | Развёртывание системы визуализации Kibana |
| kibana | Deployment | spec.template.spec.nodeSelector | elasticsearch="true" | Метка узлов, на которых необходимо развернуть компоненты для кластера Elasticsearch |
| spec.template.spec.containers.kibana-proxy.args.client-id | kibana | Идентификатор клиента Kibana в Keycloak |
| spec.template.spec.containers.kibana-proxy.args.client-secret | secret | Секретный ключ для клиента Kibana в Keycloak |
| spec.template.spec.containers.kibana-proxy.args.discovery-url | https://iap.company.ru:9050/auth/realms/iap | URL сервера авторизации Keycloak |
| spec.template.spec.containers.kibana-proxy.args.encryption-key | your-key | Ключ для декодирования состояния сессии |
| spec.template.spec.containers.kibana-proxy.args.-resources | uri=/\*|roles=kibana:user | Шаблоны конечных точек с описанием авторизованного доступа к ним (в т.ч. и по ролям) |
| spec.template.spec.containers.kibana-proxy.args.redirection-url | http://iap.company.ru:5601 | Внешний URL веб интерфейса |
| kibana | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| **70crawling** | | | | |
| **10crawling-namespace.yml** | | | | Пространство имен для ПО сбора данных из открытых источников |
| crawling\_admin\_role\_binding | RoleBinding | subjects.name | crawling-admin | Наименование группы пользователей на сервере авторизации Keycloak для администрирования данного пространства имен |
| crawling\_operator\_role\_binding | RoleBinding | subjects.name | crawling-operator | Наименование группы пользователей на сервере авторизации Keycloak для просмотра данного пространства имен |
| **20splash.yml** | | | | Развёртывание Splash - сервиса рендеринга javascript на скачанных веб страницах. |
| splash | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |
| **80machine-learning** | | | | |
| **10machine-learning-namespace.yml** | | | | Пространство имен инструментов для использования моделей машинного обучения |
| machine-learning-volume-claim | PersistentVolumeClaim | spec.resources.requests.storage | 100Gi | Параметр, задающий запрос объёма хранилища для моделей машинного обучения |
| machine-learning\_admin\_role\_binding | RoleBinding | subjects.name | machine-learning-admin | Наименование группы пользователей на сервере авторизации Keycloak для администрирования данного пространства имен |
| machine-learning\_operator\_role\_binding | RoleBinding | subjects.name | machine-learning-operator | Наименование группы пользователей на сервере авторизации Keycloak для просмотра данного пространства имен |
| **20nvidia-device-plugin.yml** | | | | Плагин для использования видеокарт NVIDIA в Kubernetes |
| **30tensorflow-serving.yml** | | | | Конфигурация и развёртывание системы обслуживания моделей машинного обучения |
| tensorflow-serving-v2 | Service | spec.externalName | iap.company.ru | Внешний домен платформы ИАП |

1. Запустить инсталлятор в командной строке операционной системы с помощью следующей команды:

kubectl apply -f <папка с настроенными манифестами>

## Информация, необходимая для эксплуатации

Эксплуатация программы «Peraspera Платформа» осуществляется администратором с использованием программных продуктов, представленных в нижеприведенной таблице (Таблица 5).

Таблица 5 – Перечень продуктов для конфигурирования и мониторинга компонент программы «Peraspera Платформа»

| **№** | **Продукт** | **Назначение** |
| --- | --- | --- |
| 1 | CMAK(Cluster Manager for Apache Kafka) | Управление кластерами kafka:   * мониторинг состояния кластера; * создание, конфигурирование и удаление тем; * конфигурирование и управление разделами тем |
| 2 | Kafdrop | Мониторинг брокеров, тем, разделов, подписчиков.  Просмотр сообщений в тем |
| 3 | Apache Zookeeper | Обеспечение распределенной синхронизации |
| 4 | Kubernetes Dashboard | Конфигурирование и управление Kubernetes:   * разворачивание контейнеризированных приложений в кластере Kubernetes; * конфигурирование специализированных ресурсов Kubernetes (Deployments, Pods, Replica Sets и др.) * мониторинг выполняющихся приложений; * мониторинг и управление ресурсами кластера |
| 5 | Grafana | Визуализация метрик аппаратных компонент платформы |
| 6 | Keycloak | Управление подсистемой информационной безопасности   * создание и конфигурирование ролей; * создание и конфигурирование пользователей; * управление доступом пользователей к приложениям. |
| 7 | Kibana | Конфигурирование индексов Elasticsearch |
| 8 | Elastic HQ | Мониторинг и администрирование кластера Elasticsearch |
| 9 | pgAdmin | Мониторинг и администрирование БД PostgreSQL |

Для эксплуатации программы «Peraspera Платформа» с целью решения прикладных задач разрабатываются специализированные контейнеризируемые приложения, решающие конкретную задачу. Такие приложения могут быть написаны на любом языке программирования, поддерживающем контейнеризацию с использованием docker (например: Java, python, nodeJS). Типы приложений, разрабатываемых для исполнения в среде программы «Peraspera Платформа», и используемые ими продукты, входящие в состав программы «Peraspera Платформа», приведены в нижеприведенной таблице (Таблица 6).

Таблица 6 – Типы приложений, разрабатываемых для исполнения в среде программы «Peraspera Платформа»

| **№** | **Тип приложений** | **Используемые продукты** | **Цель использования** |
| --- | --- | --- | --- |
| 1 | Приложения для сбора данных из открытых источников | Splash | Рендеринг веб-страниц |
| Apache Kafka | Асинхронный обмен сообщениями  Хранение первичных данных |
| PostgreSQL | Взаимодействие с нормативно-справочной информацией |
| Keycloak | Аутентификация и авторизация |
| 2 | Аналитические сервисы | Apache Kafka | Асинхронный обмен сообщениями  Хранение первичных данных |
| Apache Kafka Streams | Агрегация аналитических показателей |
| TensorFlow Serving | Взаимодействие с моделями машинного обучения для получения результатов аналитики |
| PostgreSQL | Хранение результатов аналитики |
| Keycloak | Аутентификация и авторизация |
| 3 | Сервисы построения аналитических отчетов | PostgreSQL | Источник данных для отчетов |
| Apache Kafka | Источник данных для отчетов |
| Keycloak | Аутентификация и авторизация |
| 4 | Графы Tensorflow для выполнения тензорных вычислений | TensorFlow Serving | Среда исполнения |
| 5 | Приложения для взаимодействия с пользователем | PostgreSQL | Хранение и извлечение данных, необходимых пользователю |
| Keycloak | Аутентификация и авторизация |

Лист регистрации изменений

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (стр.) в документе | № документа | Входящий  № сопроводитель-ного  документа | Подпись | Дата |
| изме­ненных | заме-ненных | новых | аннулиро-ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |