**InnData**

**Руководство администратора**

2024

**Содержание**

[Обозначения и сокращения 4](#_Toc146909125)

[1 Общие сведения 5](#_Toc146909126)

[1.1 Наименование системы 5](#_Toc146909127)

[1.2 Назначение системы 5](#_Toc146909128)

[2 Описание системы 6](#_Toc146909129)

[3 Требования к сопровождению cистемы 11](#_Toc146909130)

[3.1 Требования к аппаратному обеспечнию 11](#_Toc146909131)

[3.2 Требования к программному обеспечнию 11](#_Toc146909132)

[4 Администрирование системы 12](#_Toc146909133)

[5 Описание операций 15](#_Toc146909134)

[5.1 Установка и настройка программных средств 15](#_Toc146909135)

[5.2 Контроль работоспособности программных компонентов Системы 15](#_Toc146909136)

[5.2.1 Контроль работоспособности компонентов Системы 15](#_Toc146909137)

[5.3 Контроль работоспособности дагов (приложений) в компоненте Airflow 16](#_Toc146909138)

[5.3.1 Контроль компонентов с использованием ПО Grafana 17](#_Toc146909139)

[5.3.2 Просмотр журналов событий 18](#_Toc146909140)

[5.4 Обновление ПО Системы 20](#_Toc146909141)

[5.4.1 Обновление версии Ред ОС 20](#_Toc146909142)

[5.4.2 Обновление компонент системы 21](#_Toc146909143)

[Приложение A (обязательное) Расширение файловой системы РEД ОС 22](#_Toc146909144)

[Приложение B (обязательное) Настройка РЕД ОС 27](#_Toc146909145)

[Приложение C (обязательное) Установка и настройка ПО Kubernetes 30](#_Toc146909146)

[Приложение D (обязательное) Установка компонент Cистемы 34](#_Toc146909147)

[Приложение E (обязательное) Описание веб-интерфейса Kubernetes Dashboard 36](#_Toc146909148)

[Приложение F (обязательное) Описание веб-интерфейса Apache Airflow 40](#_Toc146909149)

[Приложение G (обязательное) Список консольных команд 44](#_Toc146909150)

[Приложение H (обязательное) Руководство к использованию команд ClickHouse 46](#_Toc146909151)

[Приложение И (обязательное) Настройка пользовательских политик для доступа к таблицам Impala через Ranger 52](#_Toc146909152)

[Приложение К (обязательное) Редактирование ресурсов контейнеров 61](#_Toc146909153)

Обозначения и сокращения

|  |  |  |
| --- | --- | --- |
| АРМ | – | Автоматизированное рабочее место |
| БД | – | База данных |
| ОС | – | Операционная система |
| ПО | – | Программное обеспечение |
| СУБД | – | Система управления базами данных |
| SSH | – | Secure Shell |
| SQL | – | Structured Query Language |
|  |  |  |
|  |  |  |

# Общие сведения

Данное руководство предназначено для проведения работ по администрированию программного продукта InnData.

## Наименование системы

Полное наименование программного продукта InnData: Платформа InnData.

Условное обозначение: Система.

## Назначение системы

Система предназначена для:

* получения/сбора, обработки, хранения и анализа данных;
* долгосрочного исторического хранения данных с необходимой глубиной при условии выделения соответствующей емкости жёстких дисков;
* консолидации в едином хранилище данных из различных информационных систем;
* представления хранимых данных в виде дашбордов;
* реализации работы моделей машинного обучения;
* создания моделей машинного обучения;
* хранения артефактов результатов машинного обучения.

# Описание системы

Система представляет собой ПО, устанавливаемое на выделенном сервере под управлением операционной системы, входящей в реестр отечественного ПО – Ред ОС. В состав Системы входят модули в виде наборов образов с установленным программным обеспечением, а также разработанными приложениями. На основе образов собираются отдельные контейнеры, объединенные в единую сеть при помощи ПО оркестровки контейнеризированных приложений Kubernetes.

В состав Системы входят следующие модули:

* Модуль сбора и получения данных
* Модуль хранения
* Модуль обработки данных
* Модуль представления данных
* Модуль управления

В составе модулей используются ПО, представленные в таблице ниже (Таблица 1).

Таблица 1 – Перечень ПО

| № п/п | Наименование ПО | Краткое описание |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | Kafka Connect | Программа для масштабируемой передачи данных между Kafka и другими системами |
| 2 | Kafka broker | Программный брокер сообщений |
| 3 | Zookeeper | Служба координации и синхронизации конфигурации сервисов |
| 4 | Spark Transform App | Приложение для трансформации данных |
| 5 | DatasetsApp | Программный модуль, обеспечивающий чтение, обработку данных и запись по заданному запросу для последующего использования в ML и модуле визуализации |
| 6 | InputDataApp | Программный модуль, обеспечивающий подготовку файлов из датасета за указанный период для последующего использования в ML |
| 7 | Grafana | Графический интерфейс для мониторинга системы |
| 8 | ClickHouse | База данных |
| 9 | Greenplum | Реляционная СУБД, имеющая массово-параллельную архитектуру на основе PostgreSQL |
| 10 | Minio | Сервер хранения объектов, совместимый с S3 |
| 11 | ClickHouse Keeper | Служба координации и синхронизации конфигурации сервисов. Используется как замена ZooKeeper при использовании нескольких экземпляров ClickHouse |
| 12 | Apache Superset | BI-инструмент для визуализации и анализа данных |
| 13 | Postgres | СУБД для постоянного хранения данных и настроек Superset |
| 14 | RStudio Shiny on Python | Фреймворк для создания интерактивных веб-приложений для анализа и визуализации данных |
| 16 | Kubernetes Dashboard | Интерфейс для управления объектами в Kubernetes |
| 17 | Kubectl | Инструмент командной строки |
| 18 | Keycloak | Инструмент для аутентификации и авторизации |
| 19 | Ranger | Инструмент для разграничения прав доступа для Impala |
| 20 | Openldap | Центр аутентификации для пользователей системы |
| 21 | Kafka-ui | Веб-интерфейс, используемый для отображения параметров и данных в Kafka |
| 22 | Hue | Веб-интерфейс для работы с запросами Impala и файлами в HDFS |
| 23 | Airflow | Инструмент для запланированного запуска приложений внутри системы |
| 24 | MLflow | Модуль для машинного обучения |
| 25 | Thrift | Дополнительный модуль для работы impala, на который перенаправляются все пользовательские запросы для обработки |
| 26 | Tarantool | Резидентная база данных |
| 27 | Impala | Инструмент для выполнения SQL-запросов к данным |
| 28 | Hadoop HDFS | Распределенная файловая система для хранения данных |
| 29 | Jupyter | Интерактивная веб-среда для разработки ноутбуков и запуска кода на поддерживаемых языках программирования |
| 30 | Redis | In-memory БД |

Список образов системы представлен в таблице (Таблица 2):

Таблица 2 ― Список образов системы

| № п/п | Наименование образа | Версия |
| --- | --- | --- |
| 1 | docker.io/rancher/mirrored-flannelcni-flannel-cni-plugin:v1.1.0 | v1.1.0 |
| 2 | docker.io/rancher/mirrored-flannelcni-flannel:v0.18.1 | v0.18.1 |
| 3 | k8s.gcr.io/coredns/coredns:v1.8.6 | v1.8.6 |
| 4 | k8s.gcr.io/etcd:3.5.3-0 | 3.5.3-0 |
| 5 | k8s.gcr.io/kube-apiserver:v1.24.3 | v1.24.3 |
| 6 | k8s.gcr.io/kube-controller-manager:v1.24.3 | v1.24.3 |
| 7 | k8s.gcr.io/kube-proxy:v1.24.3 | v1.24.3 |
| 8 | k8s.gcr.io/kube-scheduler:v1.24.3 | v1.24.3 |
| 9 | k8s.gcr.io/pause:3.2 | 3.2 |
| 10 | k8s.gcr.io/pause:3.7 | 3.7 |
| 11 | sdl.lv/airflow:2.0.0-rc | 2.0.0-rc |
| 12 | sdl.lv/blackbox-exporter:2.0.0-rc | 2.0.0-rc |
| 13 | sdl.lv/catalog:2.0.0-rc | 2.0.0-rc |
| 14 | sdl.lv/clickhouse:2.0.0-rc | 2.0.0-rc |
| 15 | sdl.lv/configmap-reload:2.0.0-rc | 2.0.0-rc |
| 16 | sdl.lv/gitbucket:2.0.0-rc | 2.0.0-rc |
| 17 | sdl.lv/grafana:2.0.0-rc | 2.0.0-rc |
| 18 | sdl.lv/greenplum\_stable:2.0.0-rc | 2.0.0-rc |
| 19 | sdl.lv/hdfs:2.0.0-rc | 2.0.0-rc |
| 20 | sdl.lv/hue:2.0.0-rc | 2.0.0-rc |
| 21 | sdl.lv/impala\_client:2.0.0-rc | 2.0.0-rc |
| 22 | sdl.lv/impala\_hms:2.0.0-rc | 2.0.0-rc |
| 23 | sdl.lv/impalad:2.0.0-rc | 2.0.0-rc |
| 24 | sdl.lv/kafka-ui:2.0.0-rc | 2.0.0-rc |
| 25 | sdl.lv/kafka\_multi:2.0.0-rc | 2.0.0-rc |
| 26 | sdl.lv/keycloak:2.0.0-rc | 2.0.0-rc |
| 27 | sdl.lv/kube-rbac-proxy:2.0.0-rc | 2.0.0-rc |
| 28 | sdl.lv/kube-state-metrics:2.0.0-rc | 2.0.0-rc |
| 29 | sdl.lv/kube-webhook:2.0.0-rc | 2.0.0-rc |
| 30 | sdl.lv/kubernetes-dashboard:2.0.0-rc | 2.0.0-rc |
| 31 | sdl.lv/kubernetes-metrics-scraper:2.0.0-rc | 2.0.0-rc |
| 32 | sdl.lv/metallb-controller:2.0.0-rc | 2.0.0-rc |
| 33 | sdl.lv/metallb-speaker:2.0.0-rc | 2.0.0-rc |
| 34 | sdl.lv/minio:2.0.0-rc | 2.0.0-rc |
| 35 | sdl.lv/mlflow:2.0.0-rc | 2.0.0-rc |
| 36 | sdl.lv/nginx-controller:2.0.0-rc | 2.0.0-rc |
| 37 | sdl.lv/nginx-internal:2.0.0-rc | 2.0.0-rc |
| 38 | sdl.lv/nifi:2.0.0-rc | 2.0.0-rc |
| 39 | sdl.lv/node-exporter:2.0.0-rc | 2.0.0-rc |
| 40 | sdl.lv/oauth2-proxy:2.0.0-rc | 2.0.0-rc |
| 41 | sdl.lv/openldap:2.0.0-rc | 2.0.0-rc |
| 42 | sdl.lv/postgresql:2.0.0-rc | 2.0.0-rc |
| 43 | sdl.lv/prometheus-adapter:2.0.0-rc | 2.0.0-rc |
| 44 | sdl.lv/prometheus-config-reloader:2.0.0-rc | 2.0.0-rc |
| 45 | sdl.lv/prometheus-operator:2.0.0-rc | 2.0.0-rc |
| 46 | sdl.lv/prometheus:2.0.0-rc | 2.0.0-rc |
| 47 | sdl.lv/pyspark:2.0.0-rc | 2.0.0-rc |
| 48 | sdl.lv/ranger-usersync:2.0.0-rc | 2.0.0-rc |
| 49 | sdl.lv/ranger:2.0.0-rc | 2.0.0-rc |
| 50 | sdl.lv/redis:2.0.0-rc | 2.0.0-rc |
| 51 | sdl.lv/shiny-proxy:2.0.0-rc | 2.0.0-rc |
| 52 | sdl.lv/statestore:2.0.0-rc | 2.0.0-rc |
| 53 | sdl.lv/superset:2.0.0-rc | 2.0.0-rc |
| 54 | sdl.lv/tarantool:2.0.0-rc | 2.0.0-rc |
| 55 | sdl.lv/zookeeper\_multi:2.0.0-rc | 2.0.0-rc |

# Требования к сопровождению cистемы

## Требования к аппаратному обеспечнию

Требования к аппаратному обеспечению сервера, на котором реализуется Система, определяются проектными техническими решениями.

## Требования к программному обеспечнию

Перечень ПО, необходимого для администратора при работе с Системой, приведен в таблице ниже (Таблица 3).

Таблица 3 – Перечень ПО АРМ администратора

|  |  |
| --- | --- |
| Наименование ПО | Версия ПО |
| Internet Explorer | 11 и выше |
| PuTTY | 0.73 и выше |
| WinSCP | 5.15.5 и выше |
| Microsoft Edge | 41 и выше |
| Google Chrome | 63 и выше |
| FireFox | 59 и выше |

# Администрирование системы

В Системе используются следующие компоненты для администрирования:

1. Kubernetes Dashboard – графический интерфейс ПО Kubernetes, который позволяет управлять контейнерными приложениями и вычислительными ресурсами в кластере Kubernetes. Интерфейс предоставляет возможность просмотра запущенных приложений в кластере, а также создавать и редактировать отдельные объекты Kubernetes. Описание веб-интерфейса приведено в приложении ниже (Приложение E).
2. Apache Airflow – ПО для планирования и мониторинга задач обработки и анализа.
3. Apache Superset – графический интерфейс пользователя для доступа к данным. Описание веб-интерфейса приведено в Руководстве пользователя.
4. Hue – инструмент для работы с данными. Описание веб-интерфейса приведено в Руководстве пользователя.
5. Grafana – средство визуализации данных, которое позволяет осуществлять мониторинг состояния и утилизации ресурсов узлов Системы и используемого ПО.
6. Ranger – инструмент для разграничения прав доступа к таблицам Impala.
7. Keycloak – инструмент для реализации контроля и разграничения доступа к Apache Superset.
8. Kubectl – интерфейс командной строки Kubernetes. Позволяет запускать команды для управления кластером Kubernetes при подключении к серверу по SSH. Список консольных команд приведен в приложении ниже (Приложение G).

Модель администрирования подсистемы хранения и анализа исторических данных предполагает наличие двух выделенных ролей администраторов:

* администратор информационной безопасности (далее – АИБ);
* системный администратор (далее – СА).

Работы по администрированию подсистемы хранения и анализа исторических данных выполняются согласно ролевой модели, приведенной в таблице ниже (Таблица 4).

Таблица 4 – Модель администрирования подсистемы хранения и анализа исторических данных

| Название роли | Описание обязанностей |
| --- | --- |
| СА | * установка и первоначальная настройка программных средств Системы; * управление ресурсами компонентов Системы * контроль утилизации ресурсов Системы; * контроль доступности компонентов Системы; * управление назначенными заданиями; * контроль назначенных заданий; * анализ системных журналов; * обновление ПО подсистемы; * резервное копирование параметров Системы; * восстановление работоспособности Системы в случае аварийных ситуаций. |
| АИБ | * управление ролями и правами доступа пользователей; * управление учетными записями пользователей; * анализ журналов аудита Системы; * создание представлений данных; * предоставление доступа к представлениям данных; |
| Пользователь | * просмотр графиков и дашбордов; * выполнение SQL-запросов |

Описание операций, выполняемых СА, приведено в разделе ниже (п. 5).

Описание операций, выполняемых АБИ, приведено в документе «Руководство администратора информационной безопасности».

# Описание операций

## Установка и настройка программных средств

Установка и настройка ОС выполняется в соответствии с документацией, поставляемой в комплекте с дистрибутивом ПО ОС, либо размещённой на сайте производителя ОС.

Последовательность действий для установки и настройки Системы:

1. Выполнить расширение файловой системы РЕД ОС путем монтирования дополнительных жестких дисков к серверу, на котором установлена Система (Приложение A).
2. Выполнить настройку РЕД ОС (Приложение B).
3. Выполнить установку и настройку ПО Kubernetes (Приложение C).
4. Выполнить установку компонент Системы (Приложение D).
5. Выполнить проверку работоспособности Системы.

## Контроль работоспособности программных компонентов Системы

### Контроль работоспособности компонентов Системы

Контроль состояния компонентов Системы выполняется в интерфейсе Kubernetes Dashboard. Для получения доступа к интерфейсу необходимо перейти по адресу *https://<hostname>:9443* и авторизоваться с использованием учетной записи администратора.

Контроль работоспособности компонентов Системы выполняется не реже одного раза в день путем просмотра и анализа раздела «Workloads» на предмет наличия ошибок в работе сервисов и недоступности компонентов.



Рисунок 1 – Просмотр состояния

## Контроль работоспособности дагов (приложений) в компоненте Airflow

Для проверки состояния приложений необходимо выполнить следующие действия:

1. В веб-браузере перейти по адресу https://<hostname>:7443.
2. Авторизоваться с использованием учетной записи администратора.
3. Перейти в раздел «DAGs», выбрать вкладку «Active».

В поле «Resent Tasks» отображается история запусков задач:

* зеленым цветом отмечены задачи, выполненные корректно, без ошибок;
* красным цветом отмечены задачи, выполненные c ошибками;
* желтым цветом отмечены задачи, не запущенные по причине прерывания предыдущей задачи.

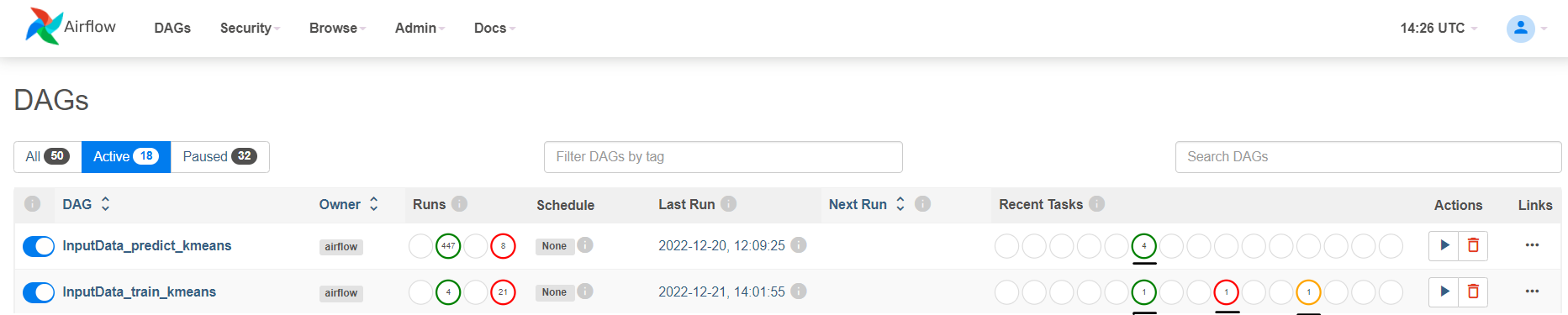


Рисунок 2 – История запуска задач

### Контроль компонентов с использованием ПО Grafana

Для получения доступа к интерфейсу Grafana необходимо в веб-браузере перейти по адресу https://<hostname>:8443 и авторизоваться с использованием учетной записи администратора.

**Основные элементы интерфейса Grafana:**

1. Панель навигации: Верхняя панель содержит основные элементы управления, такие как логотип Grafana, поиск, выпадающий список с доступными реальмами (проектами) или организациями, а также список настроек.
2. Боковое меню: Боковая панель служит для навигации между различными разделами Grafana. Здесь расположены разделы «Dashboards», "Explore", "Connections", "Alerting, Administration. Вы можете выбрать и настроить дашборды, добавить новые панели, управлять источниками данных и настраивать оповещения.
3. Dasboards - основное пространство для визуализации данных представляет собой дашборды, которые содержат различные панели графиков, таблиц, текстовых элементов и других компонентов. Дашборды позволяют собирать и отображать информацию из разных источников в удобном для анализа виде.
4. Редактор панелей - при создании или редактировании панелей дашбордов присутствует возможность выбрать тип графика, настроить запросы к базам данных, применить функции агрегации и преобразования данных. Редактор панелей предоставляет широкий выбор параметров для кастомизации отображения данных.
5. Временной интервал - в верхней части интерфейса расположены элементы управления временными интервалами в которых выставляется период, за который будут отображаться данные на графиках и панелях.
6. Фильтры и переменные - Grafana поддерживает фильтры и переменные, которые позволяют динамически изменять данные, отображаемые на дашбордах. Это позволяет создавать более интерактивные и гибкие дашборды.
7. Легенда и описание - каждая панель может содержать свою легенду (обозначения данных) и описание, что помогает понимать, что отображается на графике или таблице.
8. Экспорт и сохранение - Grafana позволяет экспортировать дашборды и панели в различных форматах, таких как PNG, PDF, CSV, JSON, и других.

### Просмотр журналов событий

#### Просмотр событий компонент

Просмотр событий компонент выполняется в интерфейсе Kubernetes Dashboard . Для получения доступа к интерфейсу необходимо перейти по адресу *https://<hostname>:9443* и авторизоваться с использованием учетной записи администратора.

Для устранения неполадок в работе подов следует просмотреть журналы событий соответствующего контейнера. Для этого необходимо перейти в раздел «Workloads – Pods», в списке выбрать нужный контейнер, в конце строки нажать знак многоточия и выбрать из списка кнопку «Logs».

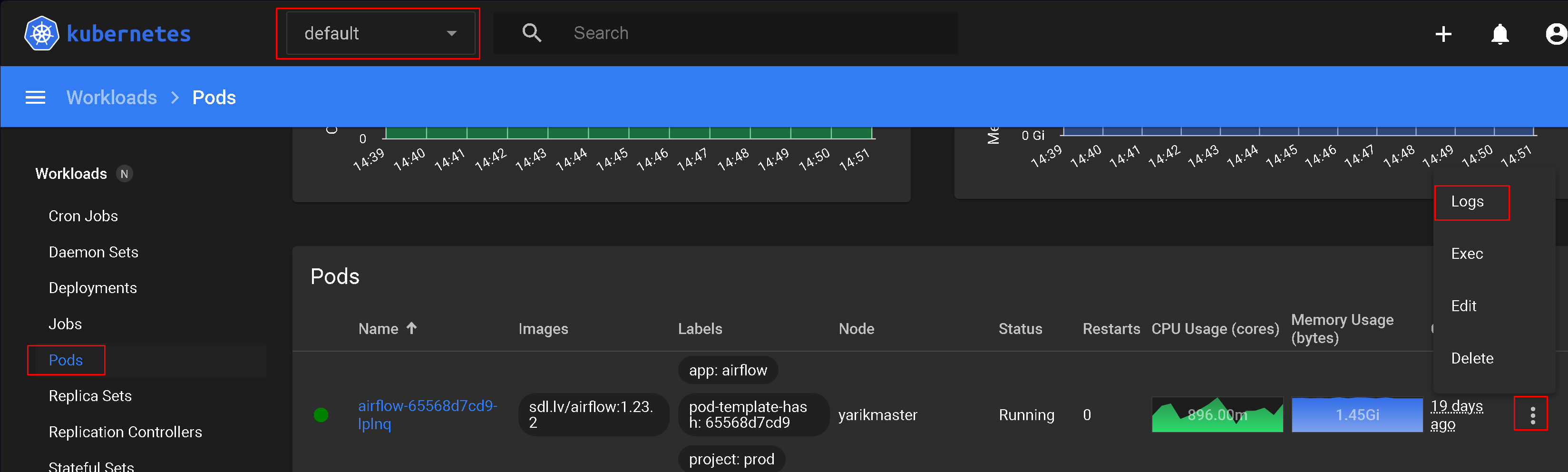


Рисунок 3 – Просмотр журналов пода

Откроется окно с отображением текущего журнала пода. На странице имеется возможность скачивания журналов и параметры его отображения.

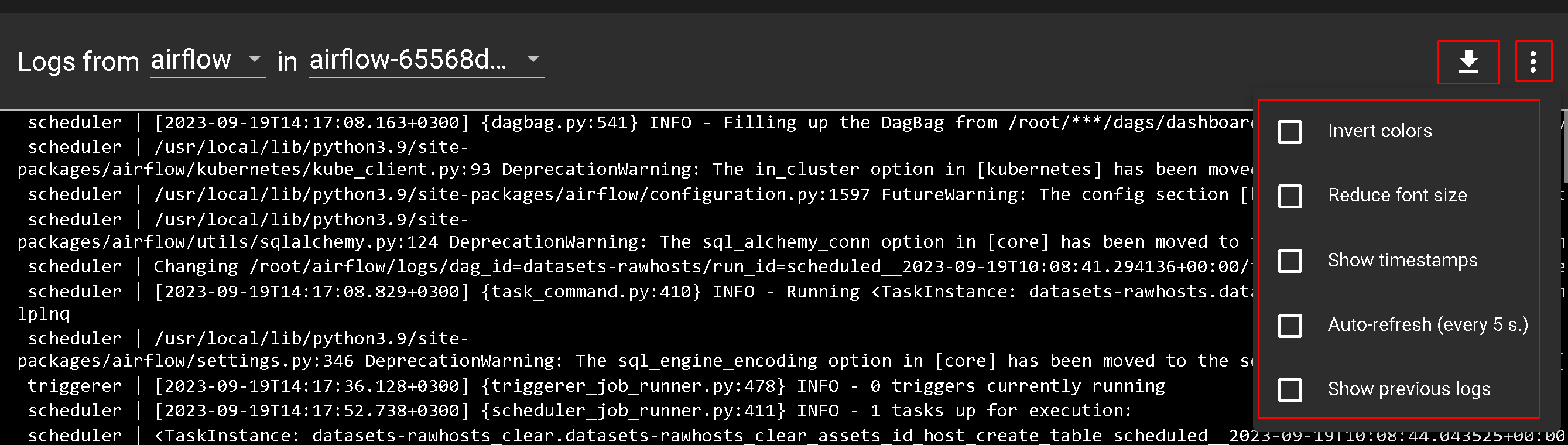


Рисунок 4 – Журналы контейнера

#### Просмотр событий приложений

Для просмотра событий приложений необходимо выполнить следующие действия:

1. В веб-браузере перейти по адресу https://<hostname>:7443.
2. Авторизоваться с использованием учетной записи администратора.
3. Перейти в раздел «DAGs → Active».
4. Выбрать необходимый DAG. Для поиска ввести в поле «Search DAGs» наименование DAG.
5. Нажать на наименование DAG для перехода в меню.
6. Выбрать текущую задачу, нажать «Log».

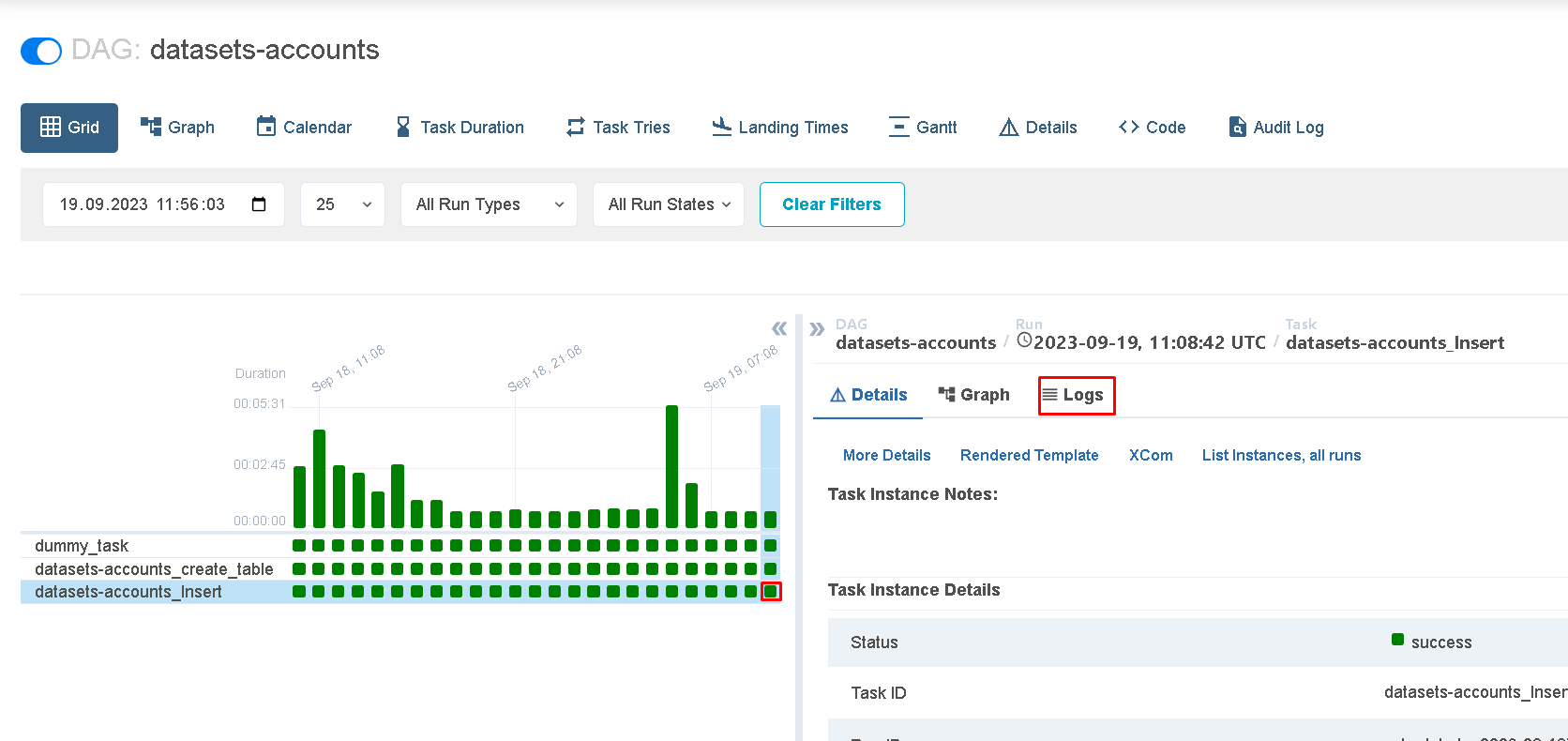


Рисунок 5 – Журнал задачи

## Обновление ПО Системы

### Обновление версии Ред ОС

Для обновления версии Ред ОС необходимо последовательно выполнить следующие пункты:

1. Подключить локальный репозиторий на хосте, создав запись для репозитория в каталоге /etc/yum.repos.d/:

nano /etc/yum.repos.d/redos7.3c.repo

1. Наполнить файл репозитория из пункта 1, изменив значение параметра baseurl, вписав путь к локальному репозиторию

# RedOS-Cert.repo

#

[base7.3c]

name=RedOS Cert - Base 7.3c

baseurl=http://{адрес\_локального\_репозитория}/repo/redos7.3c/base7.3c

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-RED-SOFT

1. В иных репозиториях, которые находятся по пути /etc/yum.repos.d/, изменить значение параметра enabled на enabled=0 (если такого значения в файлах нет, необходимо вписать его самостоятельно), чтобы не возникало конфликтов.
2. Например, файл RedOS-Base.repo, используемый по умолчанию, должен выглядеть так:

[base]

name=RedOS - Base

baseurl=https://repo1.red-soft.ru/redos/7.3/$basearch/os

enabled=0

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-RED-SOFT

1. Обновить информацию о репозиториях, для этого выполнить следующие команды:

dnf makecache

dnf update

### Обновление компонент системы

Обновление компонент InnData выполняется согласно инструкциям, приложенным к пакету обновления ПО. Пакет обновлений предоставляет разработчик ПО по мере выхода новых версий.

Приложение A  
(обязательное)  
Расширение файловой системы РEД ОС

Для расширения файловой системы РЕД ОС необходимо подключить три диска к виртуальной машине:

* первый диск – основной с корневой директорией;
* второй диск – /storage1/ – используется для хранения системных файлов и образов Kubernetes, размером 350 Гб;
* третий диск – /storage/ – используется для хранения файлов, хранящихся в InnData (размер диска определяется в соответствии с рассчетами по времени хранения файлов и размером потока поступления данных в систему).

**Монтирование диска в директорию /storage/**

1. Подключиться к виртуальной машине по протоколу SSH.
2. Создать директорию, в которую будет смонтирован диск, используя команду:

sudo mkdir /storage/

1. Выполнить команду:

sudo fdisk /dev/sdb

1. Далее следовать инструкции (ключевые моменты выделены красным цветом):

Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Command (m for help): n

Partition type:

p primary (1 primary, 0 extended, 3 free)

e extended

Select (default p): p

Using default response p

#В следующих двух пунктах нажать клавишу «Enter» 2 раза, автоматически подставится первый и последний доступный сегмент диска

Partition number (1-4, default 1):

First sector (1026048-27262975, default 1026048):

Using default value 1026048

Last sector, +sectors or +size{K,M,G} (1026048-27262975, default 27262975):

Using default value 27262975

Partition 1 of type Linux and of size 250 GiB is set

Command (m for help): t

Hex code (type L to list all codes): 8e

Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): p

Disk /dev/sdb: 250 GB, 13958643712 bytes, 27262976 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0x000912b7

Device Boot Start End Blocks Id System

/dev/sda1 \* 2048 1026047 512000 83 Linux

/dev/sda2 1026048 27262975 13118464 8e Linux LVM

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table. The new table will be used at

the next reboot or after you run partprobe(8) or kpartx(8)

Syncing disks.

1. Создать файловую систему для добавленного диска, используя команду:

sudo mkfs.ext4 /dev/sdb1

1. Смонтировать диск в директорию «/storage/», используя команду:

sudo mount /dev/sdb1 /storage/

1. Убедиться в корректности монтирования диска, используя команду:

mount | grep sdb

1. Проверить вывод команды:

/dev/sdb1 on /storage type ext4 (rw,relatime)

1. Отредактировать файл, используя команду «sudo nano /etc/fstab», и добавить информацию о смонтированном диске и разделе:

/dev/sdb1 /storage ext4 defaults,nofail 0 0

1. Выполнить команду:

sudo mount -a

**Монтирование диска в директорию /storage1/**

1. Создать директорию, в которую будет смонтирован диск, используя команду:

sudo mkdir /storage1/

1. Выполнить команду:

sudo fdisk /dev/sdс

1. Далее следовать инструкции (ключевые моменты выделены красным цветом):

Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Command (m for help): n

Partition type:

p primary (1 primary, 0 extended, 3 free)

e extended

Select (default p): p

#Следующие два пунта не указывать ничего и просто нажать на клавиатуре клавишу «Enter» 2 раза, автоматически подставится первый и последний доступный сегмент диска

Using default response p

Partition number (1-4, default 1):

First sector (1026048-27262975, default 1026048):

Using default value 1026048

Last sector, +sectors or +size{K,M,G} (1026048-27262975, default 27262975):

Using default value 27262975

Partition 1 of type Linux and of size 250 GiB is set

Command (m for help): t

Hex code (type L to list all codes): 8e

Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): p

Disk /dev/sdc: 250 GB, 13958643712 bytes, 27262976 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0x000912b7

Device Boot Start End Blocks Id System

/dev/sda1 \* 2048 1026047 512000 83 Linux

/dev/sda2 1026048 27262975 13118464 8e Linux LVM

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table. The new table will be used at

the next reboot or after you run partprobe(8) or kpartx(8)

Syncing disks.

1. Создать файловую систему для добавленного диска, используя команду:

sudo mkfs.ext4 /dev/sdc1

1. Смонтировать диск в директорию «/storage1/», используя команду:

sudo mount /dev/sdc1 /storage1/

1. Убедиться в корректности монтирования диска, используя команду:

mount | grep sdc

1. Вывод команды должен быть следующим:

/dev/sdc1 on /storage type ext4 (rw,relatime)

1. Отредактировать файл, используя команду «sudo nano /etc/fstab», и добавить информацию о смонтированном диске и разделе:

/dev/sdc1 /storage1 ext4 defaults,nofail 0 0

1. Выполнить команду:

mount –a

Приложение B  
(обязательное)  
Настройка РЕД ОС

Выполнить команды от имени пользователя root:

1. Создать пользователя, используя команду:

useradd <имя\_пользователя>

1. Отредактировать файл, используя команду:

sudo nano /etc/sudoers

Раскомментировать строку в файле:

#%wheel ALL=(ALL) ALL

Добавить строку в файл:

<имя\_пользователя> ALL=(ALL:ALL) ALL

Выполнить команды от имени пользователя (не root):

1. Выполнить команду:

sudo usermod -aG wheel <имя пользователя>

1. Отключить swap, используя команду:

sudo service firewalld stop

sudo systemctl disable firewalld

sudo swapoff -a

sudo sed -i '/ swap / s/^/#/' /etc/fstab

sudo sysctl vm.swappiness=0

sudo setenforce 0

sudo sed -i 's/^SELINUX=enforcing$/SELINUX=disabled/' /etc/selinux/config

sudo sysctl -p

sudo reboot

1. Используя команду «free -h», убедиться, что в строке swap установлено значение «0». В ином случае необходимо отредактировать файл «sudo nano /etc/fstab», закомментировав строку с параметром «swap».
2. Проверить наличие файла «/etc/resolv.conf». При отсутствии файла – необходимо его создать выполнив команду

echo "nameserver $(hostname -I | awk '{print $1}')" | sudo tee /etc/resolv.conf

Выполнить команду:

sudo service network restart

1. Отредактировать файл, используя команду:

sudo nano /etc/sysctl.conf

В файле «sysctl.conf» раскомментировать строку «net.ipv4.ip\_forward=1» и добавить строку «net.bridge.bridge-nf-call-iptables=1»

1. Последовательно выполнить следующие команды:

sudo sysctl -w fs.inotify.max\_user\_watches=100000

sudo sysctl -w fs.inotify.max\_user\_instances=100000

sudo modprobe br\_netfilter

sudo modprobe overlay

sudo sysctl -w net.ipv6.conf.all.disable\_ipv6=1

sudo sysctl -w net.ipv6.conf.default.disable\_ipv6=1

sudo service network restart

sudo sysctl -p

1. Добавить модули из предыдущего пункта в автозагрузку.

Выполнить редактирование файла, используя команду:

sudo nano /etc/modules-load.d/bridge.conf

Добавить в файл строку:

br\_netfilter

Выполнить редактирование файла, используя команду:

sudo nano /etc/sysctl.d/bridge.conf

Добавить в файл строки:

net.bridge.bridge-nf-call-iptables = 1

net.ipv4.ip\_forward = 1

1. Перезагрузить виртуальную машину, выполнив команду:

sudo reboot

1. Проверить параметры после перезагрузки:

sudo sysctl -a | grep "net.ipv4.ip\_forward"

sudo sysctl -a | grep "net.bridge.bridge-nf-call-iptables"

sudo cat /proc/modules | grep netfilter

free -h

sestatus

1. Убедиться, что вывод команд следующий:

net.ipv4.ip\_forward = 1

net.ipv4.ip\_forward\_update\_priority = 1

net.ipv4.ip\_forward\_use\_pmtu = 0

net.bridge.bridge-nf-call-iptables = 1

br\_netfilter 24576 0 - Live 0xffffffffc089b000

bridge 188416 1 br\_netfilter, Live 0xffffffffc086c000

total used free shared buff/cache available

Swap: 0B 0B 0B

SELinux status: disabled

Приложение C  
(обязательное)  
Установка и настройка ПО Kubernetes

Команды установки ПО Kubernetes выполняются от пользователя (не root).

Для установки пакетов ПО Kubernetes необходимо выполнить следующую последовательность действий:

1. Разархивировать архив «kubernetes.tar.gz» в пользовательскую директорию «home», выполнив команду:

tar xvf kubernetes.tar.gz -C /home/$USER/

1. Запустить установку ПО Kubernetes, последовательно выполнив следующие команды:

cd /home/$USER/kubernetes/kuber/

sudo yum localinstall containerd.io-1.6.20-3.1.el7.x86\_64.rpm

sudo yum localinstall ae095661b772845e86a4283d12077069d751176dac90a9046288846645fc64df-kubectl-1.24.3-0.x86\_64.rpm

sudo yum localinstall 7cb2519fffeb4dcd35c4c8b0ae9f1a049c06a8d495cb60334d4f46a692cc2e31-kubelet-1.24.3-0.x86\_64.rpm

sudo yum localinstall 3f5ba2b53701ac9102ea7c7ab2ca6616a8cd5966591a77577585fde1c434ef74-cri-tools-1.26.0-0.x86\_64.rpm

sudo yum localinstall 0f2a2afd740d476ad77c508847bad1f559afc2425816c1f2ce4432a62dfe0b9d-kubernetes-cni-1.2.0-0.x86\_64.rpm

sudo yum localinstall f2c23e99b3c2b1f42b77c7fae9ca6a324714640460ae910f0a763673dc941c25-kubeadm-1.24.3-0.x86\_64.rpm

sysctl -w fs.inotify.max\_user\_watches=100000

sysctl -w fs.inotify.max\_user\_instances=100000

sudo systemctl status containerd

sudo systemctl enable containerd

1. Выполнить следующие команды для создания необходимых директорий (системные файлы и образы Kubernetes будут храниться в директории /storage1/kuber/):

sudo mkdir -p /etc/containerd

sudo mkdir –p /storage1/kuber

sudo chmod -R 777 /storage1/kuber

1. Для установки ПО Contaierd необходимо выполнить следующие команды:

sudo containerd config default | sudo tee /etc/containerd/config.toml

if [ $(grep -c "SystemdCgroup" /etc/containerd/config.toml) -eq 0 ]; then sudo sed -i -e '/containerd.runtimes.runc.options/s/containerd.runtimes.runc.options\]/containerd.runtimes.runc.options\]\n SystemdCgroup = true/' /etc/containerd/config.toml; fi

sudo sed -i 's#/var/lib/containerd#/storage1/kuber#g' /etc/containerd/config.toml

sudo systemctl restart containerd

1. Выполнить команду для установки системных образов (файлы находятся в архиве «kubernetes.tar.gz» в директории «services»).

find //home/$USER/kubernetes/kuber/services/ -type f -name "\*.tar" -exec sudo ctr -n=k8s.io image import {} \;

1. Выполнить инициализацию кластера, используя команду:

sudo kubeadm init --kubernetes-version=v1.24.3 --pod-network-cidr=10.0.0.0/24

1. Выполнить следующие команды для копирования файлов конфигурация в системные директории:

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

1. Предоставить разрешение мастерноде на запуск поды, используя команды:

kubectl taint nodes --all node-role.kubernetes.io/master-

kubectl taint nodes --all node-role.kubernetes.io/control-plane-

1. Выполнить команду (файл «flannel.yaml» находится в архиве «kubernetes.tar»):

kubectl apply -f flannel.yaml

systemctl enable kubelet

1. Выждать минуту и убедиться, что все системные поды (namespace = kube-system) запущены, используя команду:

kubectl get pods -A

sudo reboot

1. Перезапустить dns-службы для обновления после установки плагина:

kubectl delete pod -l k8s-app=kube-dns -n kube-system

**Для повышения защищенности ПО Kubernetes далее необходимо выполнить корректировку настроек сервисов «kubelet» и «kube-apiserver»:**

1. Выполнить редактирование файла «/etc/sysconfig/kubelet», используя команду:

sudo nano /etc/sysconfig/kubelet

Добавить в файл следующую строку:

KUBELET\_EXTRA\_ARGS="--tls-cipher-suites=TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384"

1. Убедиться, что все системные поды запущены, используя команду:

kubectl get pods -n kube-system

1. Выполнить редактирование файла «kube-apiserver.yaml», используя команду:

sudo nano /etc/kubernetes/manifests/kube-apiserver.yaml

1. В конец секции «containers» на одном уровне с вышестоящими строками добавить строку (текст добавляется в одну строку, без перевода строки):

- --tls-cipher-suites=TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384

Итоговое содержание секции:

  containers:

  - command:

    - kube-apiserver

    - --advertise-address=х.х.х.х

    - --allow-privileged=true

    - --authorization-mode=Node,RBAC

    - --client-ca-file=/etc/kubernetes/pki/ca.crt

    - --enable-admission-plugins=NodeRestriction

    - --enable-bootstrap-token-auth=true

    - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt

    - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt

    - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key

    - --etcd-servers=https://127.0.0.1:2379

    - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt

    - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key

    - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname

    - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt

    - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key

    - --requestheader-allowed-names=front-proxy-client

    - --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt

    - --requestheader-extra-headers-prefix=X-Remote-Extra-

    - --requestheader-group-headers=X-Remote-Group

    - --requestheader-username-headers=X-Remote-User

    - --secure-port=6443

    - --service-account-issuer=https://kubernetes.default.svc.cluster.local

    - --service-account-key-file=/etc/kubernetes/pki/sa.pub

    - --service-account-signing-key-file=/etc/kubernetes/pki/sa.key

    - --service-cluster-ip-range= х.х.х.х /х

    - --tls-cert-file=/etc/kubernetes/pki/apiserver.crt

    - --tls-private-key-file=/etc/kubernetes/pki/apiserver.key

- --tls-cipher-suites=TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384

1. Выполнить перзагрузку хоста, используя команду:

sudo reboot -r

1. После перезапуска хоста подождать 5 минут и убедиться, что все системные поды находятся в статусе «Running», используя команду:

kubectl get pods -n kube-system

Приложение D  
(обязательное)  
Установка компонент Cистемы

Для установки компонент Системы необходимо выполнить следующую последовательность действий:

1. Выполнить команды чтобы разархивировать файлы установки системы

tar -xvf /storage/y-base-cv/y-base-cv.tar.gz

tar -xvf /storage/y-base-svn/y-base-svn.tar.gz

1. Запустить скрипт установки «install.sh»

/storage/y-base-svn/install/install.sh

1. Запустить скрипт установки «install-system.sh»

/storage/y-base-cv/install/install-system.sh

1. Запустить скрипт установки «user-token-install.sh»

/storage/y-base-svn/install/user-token-install.sh

**Настройка keycloak**

1. Авторизоваться в веб-интерфейсе keycloak «https://<hostname>/keycloak», в левом верхнем углу выбрать «Realm» – «master», раздел «Groups», название группы – «admins» (Рисунок 6).

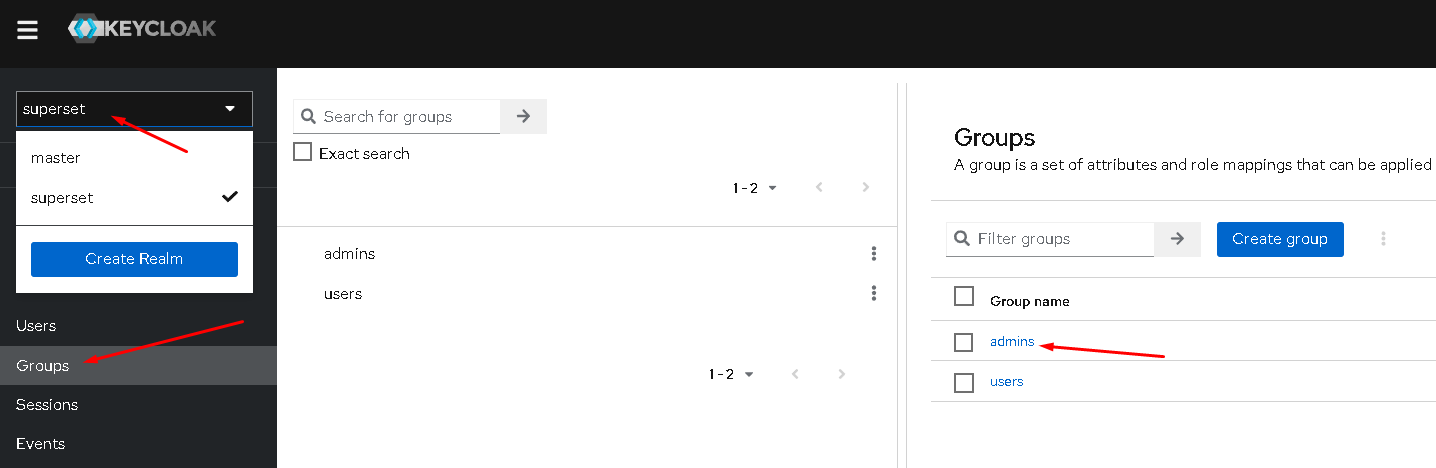


Рисунок 6 ― Настройка групп в keycloak

1. В разделе «members» добавить пользователя «sac.admin» (Рисунок 7).

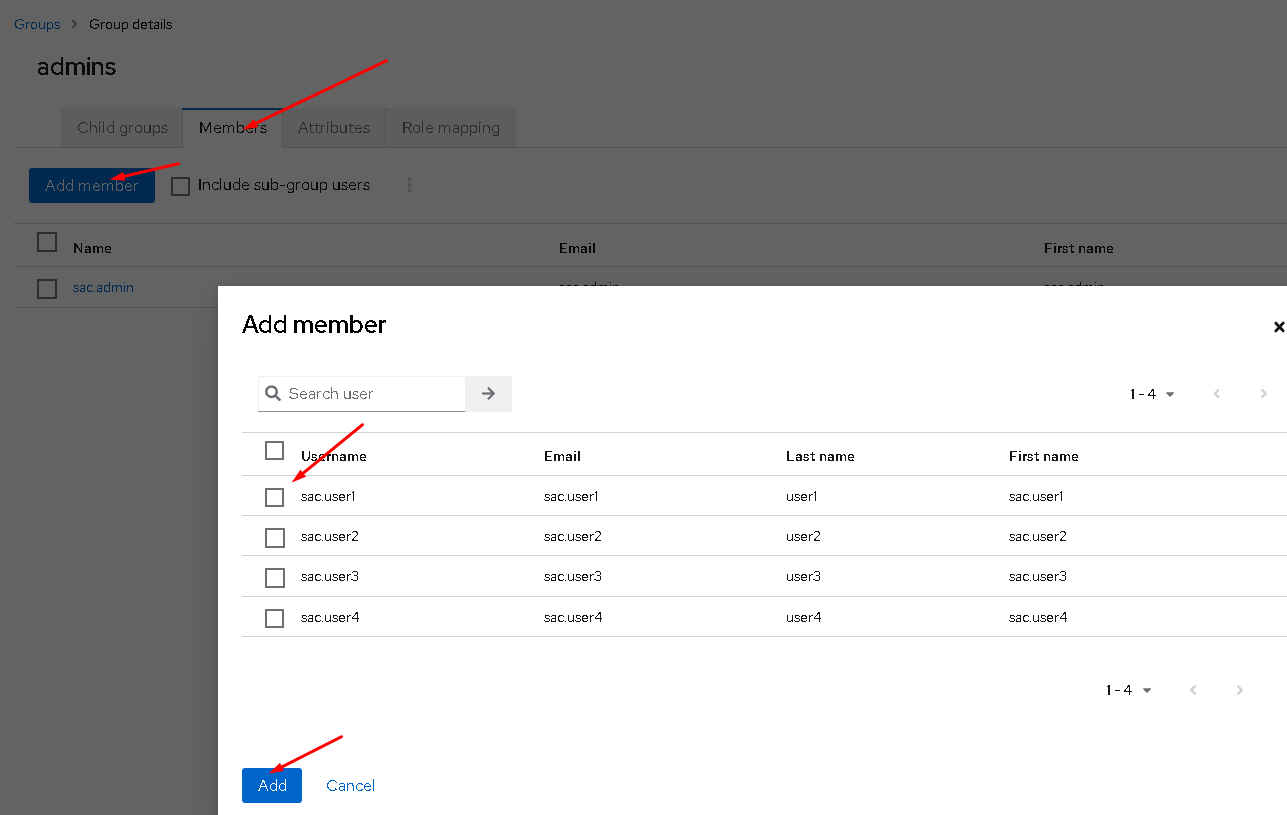


Рисунок 7 ― Добавление пользователя в группу keycloak

Приложение E  
(обязательное)  
Описание веб-интерфейса Kubernetes Dashboard

Для получения доступа к интерфейсу Kubernetes Dashboard необходимо сгенерировать токен аутентификации:

1. Подключиться к серверу Системы по протоколу SSH.
2. Выполнить команду для получения токена:

kubectl create token dashboard-admin-sa

Токен будет отображен на экране. По умолчанию токен действителен 1 час. По истечению часа потребуется генерировать токен повторно. Для удобства возможно указать время жизни токена, используя соответствующий ключ при создании:

kubectl create token dashboard-admin-sa --duration=24h

1. В веб-браузере перейти по адресу *https://<hostname>:9443*.
2. Вставить полученный токен в строку «Token»

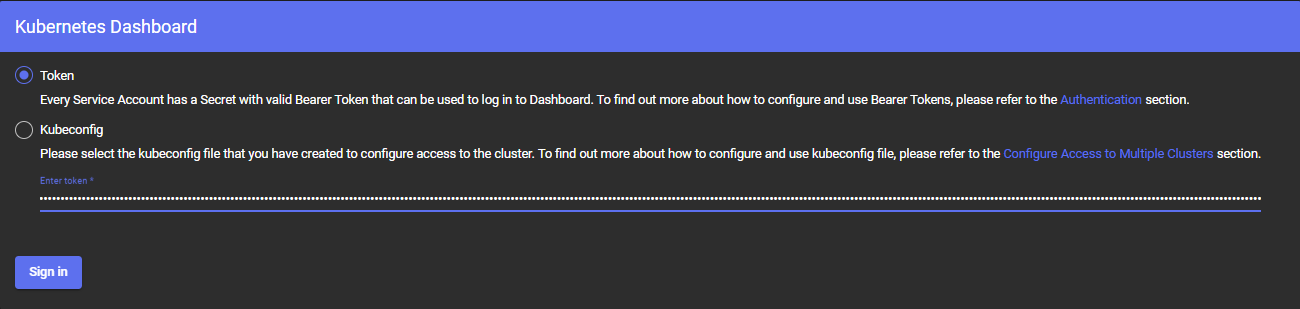


Рисунок 8 – Аутентификация Kubernetes Dashboard

Основные объекты интерфейса Kubernetes Dashboard:

1. Выпадающий список, в котором выбирается пространство имен «Namespace». По умолчанию используется значение «default», в котором запускаются компоненты Системы.
2. Меню разделов. Описание разделов приведено в таблице ниже (Рисунок 9).
3. Рабочая область.

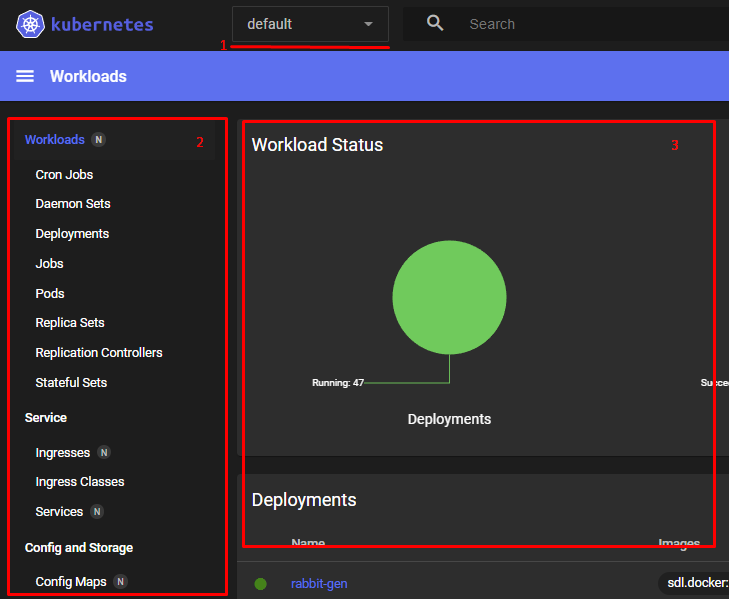


Рисунок 9 – Объекты интерфейса

Описание разделов интерфейса Kubernetes Dashboard приведено в таблице ниже (Таблица 5).

Таблица 5 – Разделы интерфейса

| № п/п | Раздел | Описание |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | Workloads | Отображает все приложения, работающие в выбранном пространстве имен. В разделе перечислены приложения по типу абстракций рабочей нагрузки (например, Deployments, Pods, Jobs) |
| 2 | Services | Отображает объекты «Services», которые предназначены для взаимодействия подов внутри кластера, а также удаленного доступа к ним |
| 3 | Config and Storage | Отображает объекты «ConfigMap», «PVC» - которые используются приложениями |
| 4 | Cluster | Отображает информацию о кластере |
| 5 | Settings | Настройки интерфейса |

**Запуск, остановка и перезапуск компонент**

После изменения ресурсов, обновлении параметров контейнера необходимо перезапустить контейнер вручную. Для этого:

1. В веб-браузере перейти по адресу *https://<hostname>:9443*.
2. Остановка контейнеров выполняется в разделе «Workloads-Deployments». Для остановки необходимо в строке с нужным «Deployment» выбрать значок многоточия, нажать кнопку «Scale», в поле «Desired replicas» установить значение «0» и нажать кнопку «Scale» (Рисунок 10).

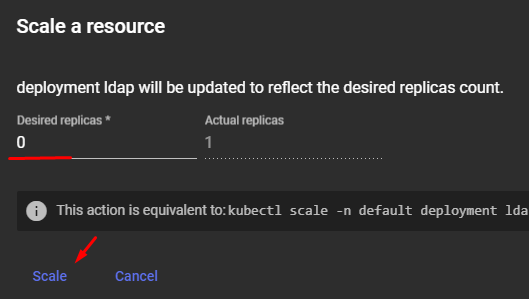


Рисунок 10 – Остановка пода

Запуск выполняется автоматически. В случае запуска пода, который остановлен вручную, необходимо перейти в раздел «Workloads-Deployments», в строке с нужным «Deployment» выбрать значок многоточия, нажать кнопку «Scale», в поле «Desired replicas» установить значение «1» и нажать кнопку «Scale».

Перезапуск подов выполняется в разделе «Workloads-Deployments». В случае возникновения проблем с подом, в этом разделе, в столбце «Pods» будет отражено состояние:

* 1\1 – под работает корректно;
* 0\1 или 0\0 – под не работает.

Для перезапуска необходимо нажать на знак многоточия в конце строки проблемного «Deployment» и выбрать из списка кнопку «Restart» (Рисунок 11).

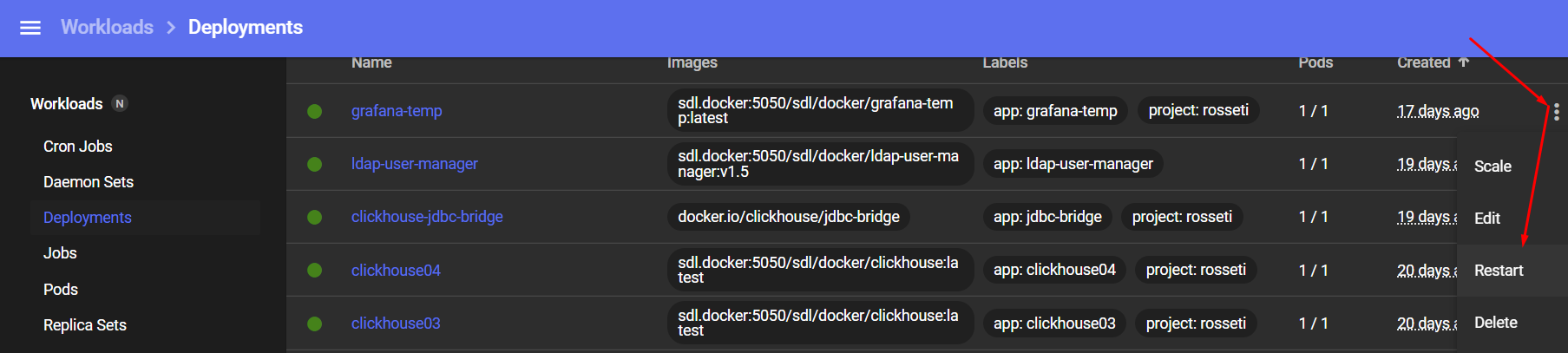


Рисунок 11 – Окно «Workloads-Deployments»

Приложение F  
(обязательное)  
Описание веб-интерфейса Apache Airflow

Для получения доступа к интерфейсу Apache Airflow необходимо в веб-браузере перейти по адресу *https://<hostname>:7443* и авторизоваться с использованием учетной записи администратора.

Основные объекты интерфейса Apache Airflow:

1. «DAGs» – список направленных ациклических графов (DAG), содержащих в свою очередь задачи. Включает вкладки: «All» (все задачи), «Active» (активные задачи), «Paused» (приостановленные задачи).
2. «Security» – содержит информацию о зарегистрированных учетных данных.
3. «Browse» – содержит информацию о запущенных задачах, журнал событий.
4. «Admin» – содержит информацию о подключенных плагинах, редактор для создания и редактирования dags.

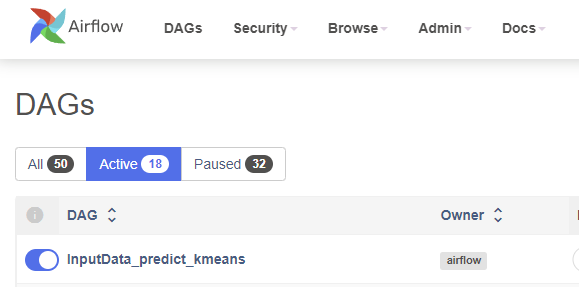


Рисунок 12 – Обзор Airflow

**Запуск, остановка и перезапуск приложений**

**Запуск задачи в Airflow**

Для запуска задачи в airflow необходимо выполнить следующие действия:

1. В веб-браузере перейти по адресу *https://<hostname>:7443*.
2. Авторизоваться с использованием учетной записи администратора.
3. Перейти в раздел «DAGs → All».
4. Выбрать нужный DAG. Возможно ввести в поле «Search DAGs» для поиска по наименованию.
5. Для одномоментного запуска задачи нажать на кнопку «Пуск».



Рисунок 13 – Запуск задачи

1. Для запуска задачи по заданному расписанию - активировать переключатель слева от наименования DAG-а.





Рисунок 14 – Запуск задачи по расписанию

**Остановка задачи, запущенной по расписанию**

Для остановки задачи, запущенной по расписанию, необходимо выполнить следующие действия:

1. В веб-браузере перейти по адресу *https://<hostname>:7443*.
2. Авторизоваться с использованием учетной записи администратора.
3. Перейти в раздел «DAGs» вкладка «All».
4. Выбрать нужный DAG. Возможно ввести в поле «Search DAGs» для поиска по наименованию.
5. Деактивировать переключатель слева в найденной строке для планировщика выполнения задачи по расписанию (Рисунок 15).





Рисунок 15 – Выбор параметра

1. Выполнение запущенной задачи не прервется, но следующая задача по расписанию не запустится.

**Прерывание задачи во время выполнения**

Для прерывания задачи во время выполнения необходимо выполнить следующие действия:

1. В веб-браузере перейти по адресу *https://<hostname>:7443*.
2. Авторизоваться с использованием учетной записи администратора.
3. Перейти в раздел «DAGs → Active».
4. Найти нужный DAG. Возможно ввести в поле «Search DAGs» для поиска по наименованию.
5. Нажать на наименование DAG для перехода в его меню.
6. Выбрать текущую задачу, нажать «Mark Failed».

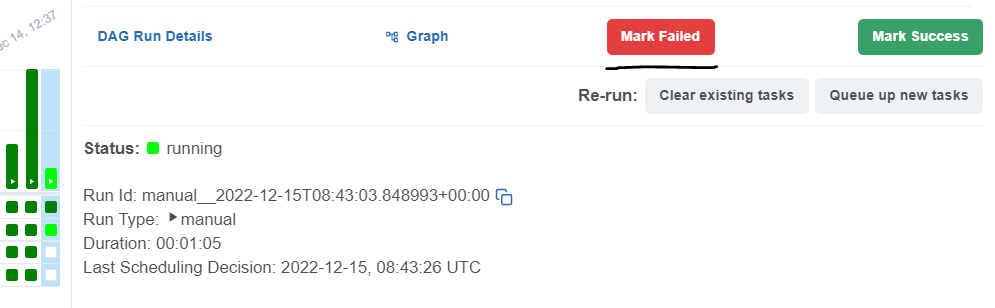


Рисунок 16 – Выбор задачи

Задача будет прервана во время выполнения.

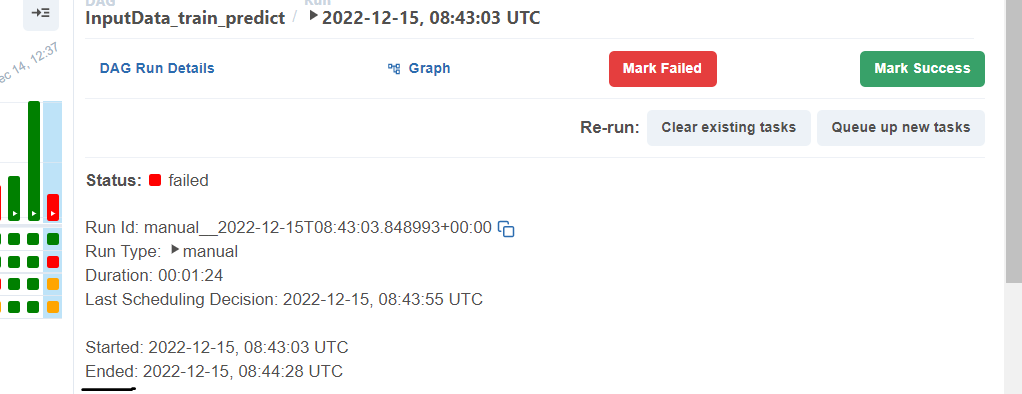


Рисунок 17 – Прерывание задачи

Приложение G  
(обязательное)  
Список консольных команд

**Kubernetes**

Создание объектов в кластере производится через .yaml файлы - манифесты, которые определяют тип объекта и его конфигурацию.

1. Pod – это абстрактный объект Kubernetes, представляющий собой группу из одного или нескольких контейнеров приложения и совместно используемых ресурсов для этих контейнеров.
2. Services – в Kubernetes это абстракция, которая определяет логический объединенный набор pod и политику доступа к ним.
3. Volumes – это директория с данными в ней, которая доступна в контейнере.
4. Манифесты – конфигурационные файлы, в которых описана структура приложений
5. Deployments – контроллер, который управляет состоянием развертывания подов, которое описывается в манифесте, следит за удалением и созданием экземпляров подов.

Получение информации о кластере и его узлах:

1. kubectl get nodes – вывести список всех хостов кластера;
2. kubectl describe nodes node\_1 – посмотреть информацию о конкретной ноде.

**Просмотр и поиск информации о сервисах, подах, деплойментах, вольюмах**

1. kubectl get services - вывести все сервисы в пространстве имен;
2. kubectl get pods – вывести список подов;
3. kubectl get pods --all-namespaces - вывести все поды во всех пространств имен;
4. kubectl get pods -o wide - вывести все поды в текущем пространстве имен с подробностями;
5. kubectl get pod my-pod -o yaml – получить информацию по поду в формате .yaml;
6. kubectl get pods --show-labels - показать метки всех подов (или любого другого объекта Kubernetes, которым можно прикреплять метки);
7. kubectl describe pods my\_pod\_name – получить детальную информацию по поду;
8. kubectl get pods --sort-by='.status.containerStatuses[0].restartCount' – вывести поды по количеству перезапусков;
9. kubectl get deployment – вывести список деплойментов;
10. kubectl get deployment my\_deployment – вывести определенный деплоймент;
11. kubectl get pv – вывод список вольюмов кластера;
12. kubectl describe pv {название\_pv} – вывод детальной информации о всех вольюмах кластера;
13. kubectl describe pv – вывод детальной информации о конкретном вольюме кластера;
14. kubectl logs {название\_пода} – просмотр и вывод журналов пода (можно добавить флаг -f – отслеживание в режиме реального времени).

**Работа с манифестами**

Команды для запуска подов, деплойментов, запуска консоли, создание configmap:

1. kubectl apply -f /{путь\_до}/ – для запуска пода, деплоймента, где path – путь до .yaml файла (манифеста) или директории с .yaml файлами;
2. kubectl exec -it {название\_пода} bash – открытие shell-консоли для работы внутри пода;
3. kubectl create configmap {название\_configmap} –from-file=/путь\_до\_файла\_configmap.yaml – создание configmap из файла.

Приложение H  
(обязательное)  
Руководство к использованию команд ClickHouse

При работе с ClickHouse для мониторинга работоспособности запросов используются следующие команды:

1. Вывод информации по всем выполняющимся в реальном времени запросам:

SELECT \* FROM system.processes FORMAT Vertical;

Результатом выполнения будет полная информация о всех активных запросах:

Row 1:

──────

is\_initial\_query: 1

user: admin

query\_id: 39807e11-1b4b-4414-b80f-25e6a39e7a34

address: ::ffff:10.0.0.1

port: 27590

initial\_user: admin

initial\_query\_id: 39807e11-1b4b-4414-b80f-25e6a39e7a34

initial\_address: ::ffff:10.0.0.1

initial\_port: 27590

interface: 2

os\_user:

client\_hostname:

client\_name:

client\_revision: 0

client\_version\_major: 0

client\_version\_minor: 0

client\_version\_patch: 0

http\_method: 2

http\_user\_agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36 OPR/99.0.0.0

http\_referer: http://10.11.4.216:32023/play?user=admin

forwarded\_for:

quota\_key:

distributed\_depth: 0

elapsed: 0.000675229

is\_cancelled: 0

is\_all\_data\_sent: 0

read\_rows: 0

read\_bytes: 0

total\_rows\_approx: 0

written\_rows: 0

written\_bytes: 0

memory\_usage: 0

peak\_memory\_usage: 0

query: SELECT \* ml.mltable LIMIT 10;

thread\_ids: [1008]

ProfileEvents: {'Query':1,'SelectQuery':1,'ContextLock':12,'RWLockAcquiredReadLocks':1}

Settings: {'use\_uncompressed\_cache':'0','load\_balancing':'in\_order','log\_queries':'1','enable\_http\_compression':'1','add\_http\_cors\_header':'1','max\_result\_rows':'1000','max\_result\_bytes':'10000000','result\_overflow\_mode':'break','max\_memory\_usage':'16000000000','max\_partitions\_per\_insert\_block':'5000'}

current\_database: default

1. Вывод информации об активных запросах, без полной информации о них:

SELECT query FROM system.processes FORMAT Vertical;

1. Вывод информации о последних 10 выполнненых запросов:

SELECT \* FROM system.query\_log WHERE type = 'QueryFinish' ORDER BY query\_start\_time DESC LIMIT 10 FORMAT Vertical;

В условии type (тип события произошедшего во время выполнения запроса) может быть указано следующее:

* 'QueryStart' = 1 — успешное начало выполнения запроса (запрос начал выполняться, но пока висит в процессе выполнения. Окончательный результат выполнения неизвестен);
* 'QueryFinish' = 2 — успешное окончание выполнения запроса (запрос выполнился);
* 'ExceptionBeforeStart' = 3 — возникла ошибка перед началом выполнения запроса (ошибка в запросе);
* 'ExceptionWhileProcessing' = 4 — возникла ошибка во время выполнения запроса (пример - запрос упал по памяти).

В условии LIMIT указывается история коичество последних запросов, информацию о которых необходимо вывести.

Пример вывода запроса:

Row 9:

───────

type: QueryFinish

event\_date: 2023-07-19

event\_time: 2023-07-19 16:18:32

event\_time\_microseconds: 2023-07-19 16:18:32.989232

query\_start\_time: 2023-07-19 16:18:32

query\_start\_time\_microseconds: 2023-07-19 16:18:32.983656

query\_duration\_ms: 5

read\_rows: 11060

read\_bytes: 729820

written\_rows: 0

written\_bytes: 0

result\_rows: 514

result\_bytes: 34337

memory\_usage: 0

current\_database: datasets

query: SELECT `tactic` AS `tactic`,

concat(technique\_id, if(subtechnique\_id is null, '', concat('.', subtechnique\_id))) AS `My column`,

1 AS `1`

FROM

(SELECT technique\_id,

subtechnique\_id,

arrayJoin(tactic) AS tactic,

data\_source,

data\_component,

id,

uuid,

src\_host,

src\_hostname,

dst\_host,

dst\_hostname,

event\_src\_host,

rt

FROM datasets.triggered\_techniques) AS `virtual\_table`

GROUP BY `tactic`,

concat(technique\_id, if(subtechnique\_id is null, '', concat('.', subtechnique\_id)))

ORDER BY `1` DESC

LIMIT 1000

FORMAT Native

formatted\_query:

normalized\_query\_hash: 11483707632556725541

query\_kind: Select

databases: ['datasets']

tables: ['datasets.triggered\_techniques']

columns: ['datasets.triggered\_techniques.subtechnique\_id','datasets.triggered\_techniques.tactic','datasets.triggered\_techniques.technique\_id']

projections: []

views: []

exception\_code: 0

exception:

stack\_trace:

is\_initial\_query: 1

user: admin

query\_id: ce5cb5c0-087e-4bf3-a2bf-68d314619ae9

address: ::ffff:10.0.0.36

port: 48088

initial\_user: admin

initial\_query\_id: ce5cb5c0-087e-4bf3-a2bf-68d314619ae9

initial\_address: ::ffff:10.0.0.36

initial\_port: 48088

initial\_query\_start\_time: 2023-07-19 16:18:32

initial\_query\_start\_time\_microseconds: 2023-07-19 16:18:32.983656

interface: 2

is\_secure: 0

os\_user:

client\_hostname:

client\_name:

client\_revision: 0

client\_version\_major: 0

client\_version\_minor: 0

client\_version\_patch: 0

http\_method: 2

http\_user\_agent: superset/2.0.1 clickhouse-connect/0.5.7 (lv:py/3.8.12; os:linux)

http\_referer:

forwarded\_for:

quota\_key:

distributed\_depth: 0

revision: 54463

log\_comment:

thread\_ids: [1008]

ProfileEvents: {'Query':1,'SelectQuery':1,'FileOpen':1,'Seek':1,'ReadBufferFromFileDescriptorRead':4,'ReadBufferFromFileDescriptorReadBytes':7548,'WriteBufferFromFileDescriptorWrite':1,'WriteBufferFromFileDescriptorWriteBytes':13209,'ReadCompressedBytes':6598,'CompressedReadBufferBlocks':6,'CompressedReadBufferBytes':428380,'OpenedFileCacheMisses':1,'IOBufferAllocs':2,'IOBufferAllocBytes':238613,'ArenaAllocChunks':3,'ArenaAllocBytes':28672,'FunctionExecute':4,'MarkCacheHits':1,'CreatedReadBufferOrdinary':1,'DiskReadElapsedMicroseconds':6,'DiskWriteElapsedMicroseconds':24,'SelectedParts':1,'SelectedRanges':1,'SelectedMarks':2,'SelectedRows':11060,'SelectedBytes':729820,'ContextLock':59,'RWLockAcquiredReadLocks':4,'RealTimeMicroseconds':5591,'UserTimeMicroseconds':5210,'SoftPageFaults':428,'OSCPUWaitMicroseconds':19,'OSCPUVirtualTimeMicroseconds':5209,'OSWriteBytes':16384,'OSReadChars':8000,'OSWriteChars':15941}

Settings: {'use\_uncompressed\_cache':'0','load\_balancing':'in\_order','log\_queries':'1','enable\_http\_compression':'1','send\_progress\_in\_http\_headers':'1','http\_headers\_progress\_interval\_ms':'120000','max\_memory\_usage':'16000000000','max\_partitions\_per\_insert\_block':'5000'}

used\_aggregate\_functions: []

used\_aggregate\_function\_combinators: []

used\_database\_engines: []

used\_data\_type\_families: []

used\_dictionaries: []

used\_formats: ['Native']

used\_functions: ['isNull','if','concat']

used\_storages: []

used\_table\_functions: []

transaction\_id: (0,0,'00000000-0000-0000-0000-000000000000')

1. Остановка запросов:

Запросы можно останавливать по двум условиям: останавливать запросы, связанные с id запроса и остановкой всех запросов пользователя.

Примеры команд:

* 1. Остановка запроса по query\_id:

KILL QUERY WHERE query\_id='2-857d-4a57-9ee0-327da5d60a90'

Чтобы узнать query\_id конкретного запроса, необходимо выполнить команду из пункта (1).

* 1. Остановка запроса по пользователю:

KILL QUERY WHERE user='username' SYNC

После выполнения запроса остановятся все активные запросы, запущенные от пользователя с именем «username».

Приложение И  
(обязательное)  
Настройка пользовательских политик для доступа к таблицам Impala через Ranger

1. Авторизоваться в сервисе Ranger, перейдя на https://<*hostname*>:6080
2. Напротив строки «HADOOP SQL» нажать на «Add new service» (Рисунок 18)

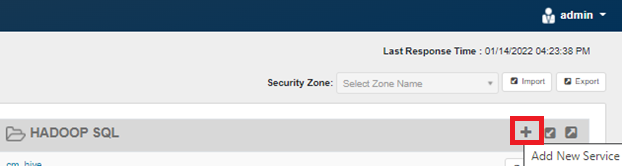


Рисунок 18 ― Подключение политик Ranger

1. Заполнить поля следующими параметрами (Рисунок 19)

Service Name – cm\_hive

Active Status – Enabled

Username – hive

Password – hive

org.apache.hive.jdbc.HiveDriver

jdbc:hive2://thrift:8000/default;transportMode=http;httpPath=cliservice

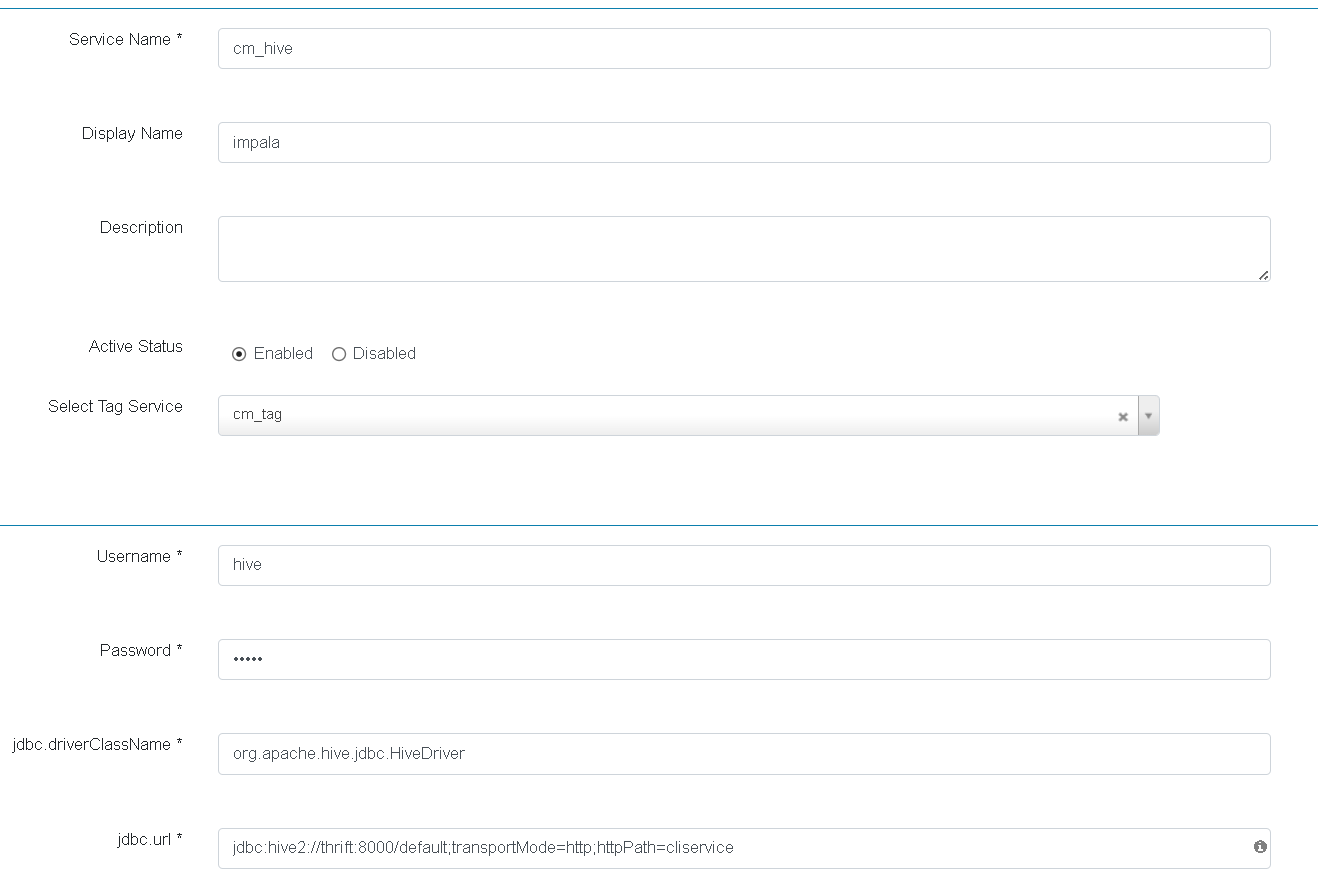


Рисунок 19 ― Подключение Ranger к Impala

1. В левом нижнем углу страницы нажать кнопку «Test Connection»
2. После прохождения проверки подключения нажать «Save»

Добавление политик Ranger

Для добавления политик для пользователей необходимо:

1. Перейти в созданный сервис cm\_hive (Рисунок 20):

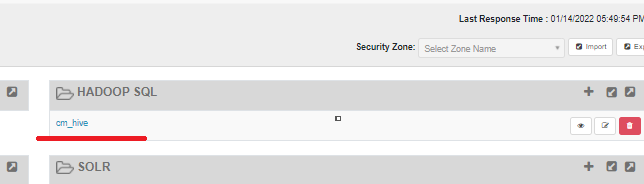


Рисунок 20 ―Добавление подключения Ranger

1. Отключить стандартные политики (Рисунок 21), такие как:

* all –database, table, column
* all –database, table
* all – storage-type, storage-url
* all – database
* all – hiveservice
* all –database, udf
* all – url
* default database table columns

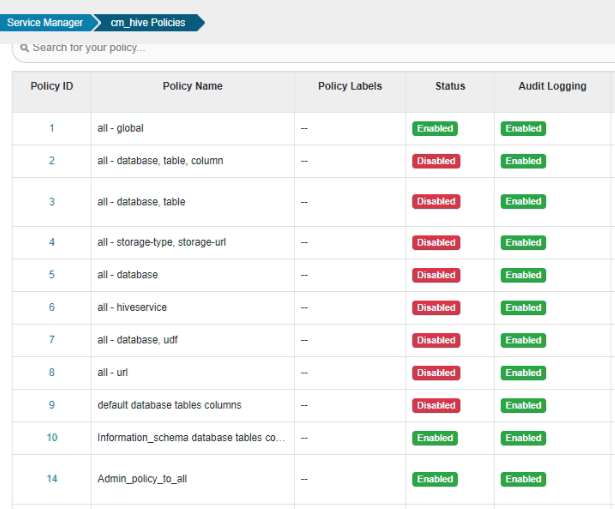


Рисунок 21 ― Отлкючение стандартных политик Ranger

Для этого нажать кнопку «edit» напротив нужной политики (Рисунок 22)

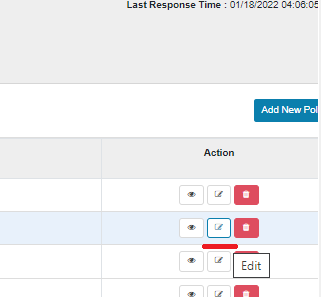


Рисунок 22 Редактирование стандартных политик Ranger

Перевести тригер в состояние «disable», нажав на соответствующую кнопку правее назваения политики (Рисунок 23) и сохранить изменения, нажав кнопку «save» в нижней правой части экрана.

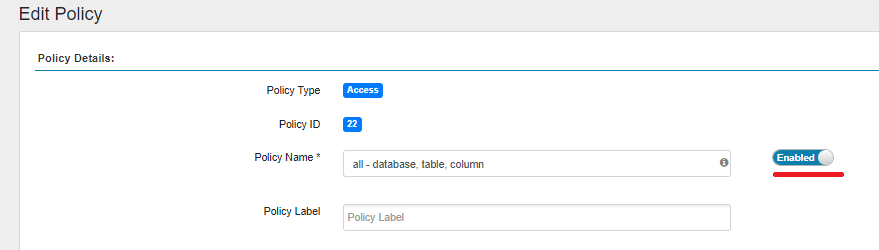


Рисунок 23 ― Отключение политики Ranger

1. Повторить для всех политик, перечисленных в пункте (2)
2. Создать политику для администраторов, для этого в окне со списком политик нажать «Add New Policy» (Рисунок 24)

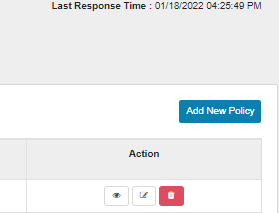


Рисунок 24 ― Добавление новой политики Ranger

1. Задать следующие значения (**Ошибка! Источник ссылки не найден.**):

* Policy Name» - Admin\_policy\_to\_all
* database - \*
* table - \*
* column - \*
* Selecct Group – STD App SVOI SDL Admins
* Select User – {Список всех администраторов, которые будут иметь доступ ко всем таблицам и действям с таблицами/базами данных}
* Permissions – {Все доступные привелегии}
* Delegate Admin – {поставить галку}

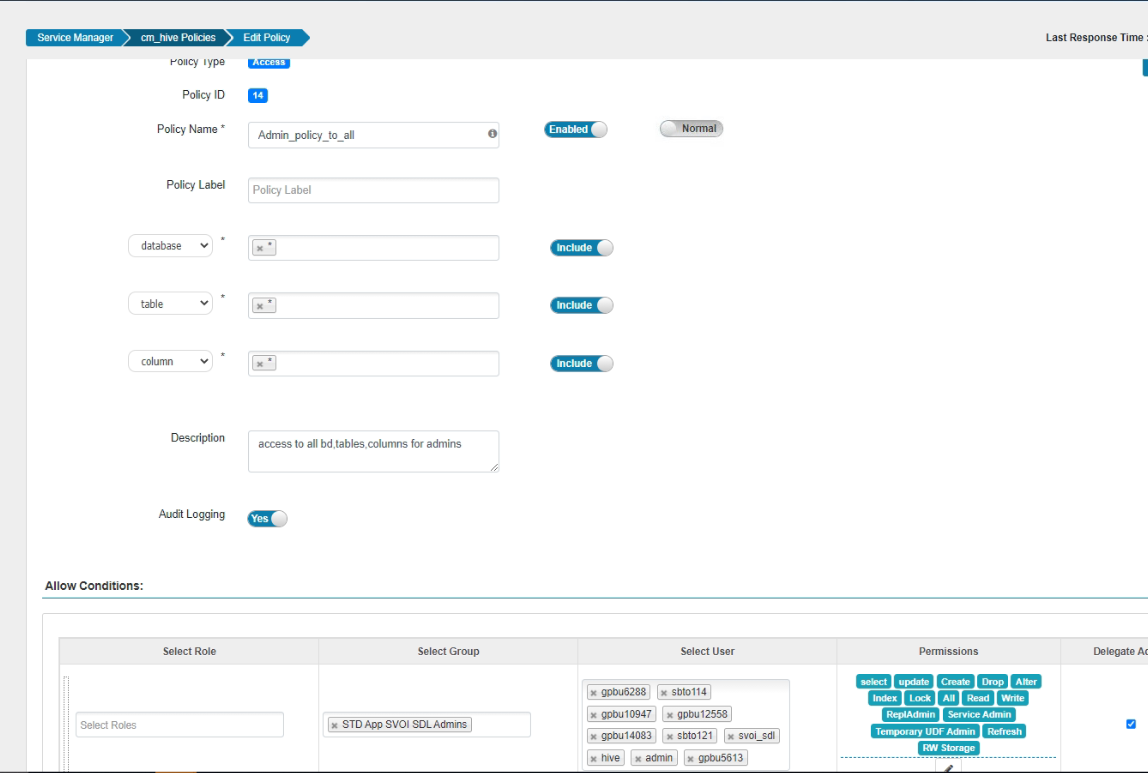


Рисунок 25 ― Создание политики Ranger для Администраторов

1. Сохранить изменения
2. Создать две политики для Impala Api, как показано на (Рисунок 26) и (Рисунок 27)

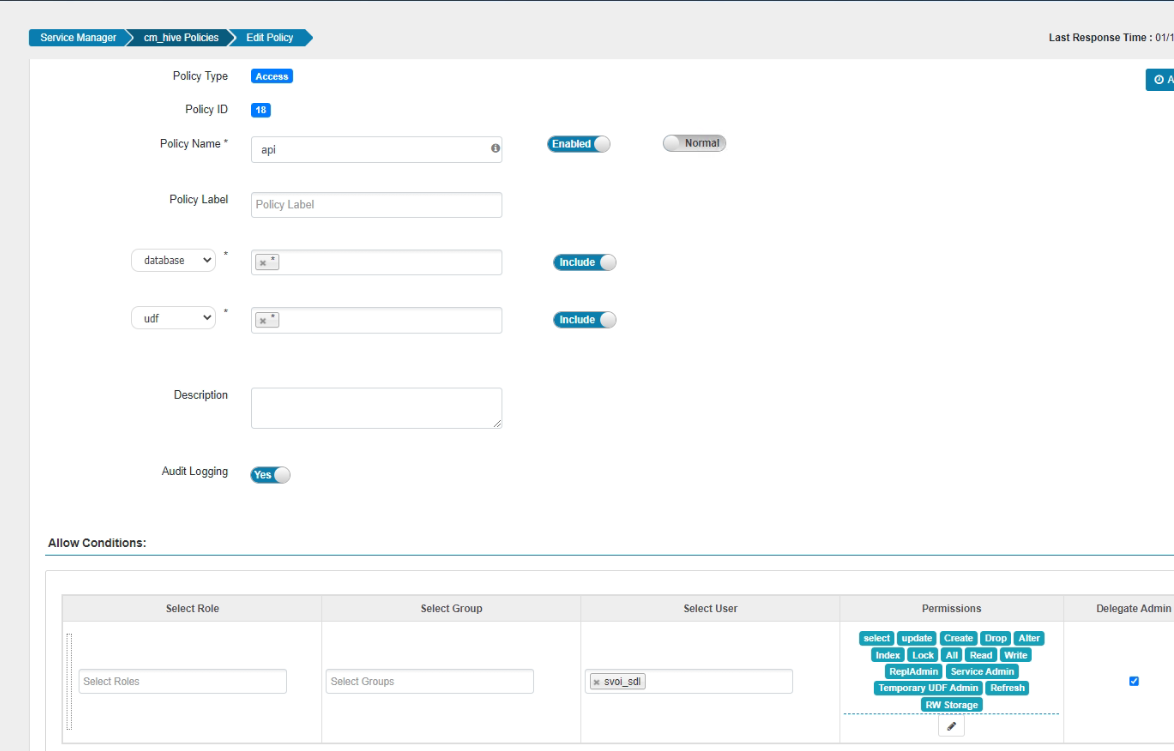


Рисунок 26 ― Создание первой политики для Impala API

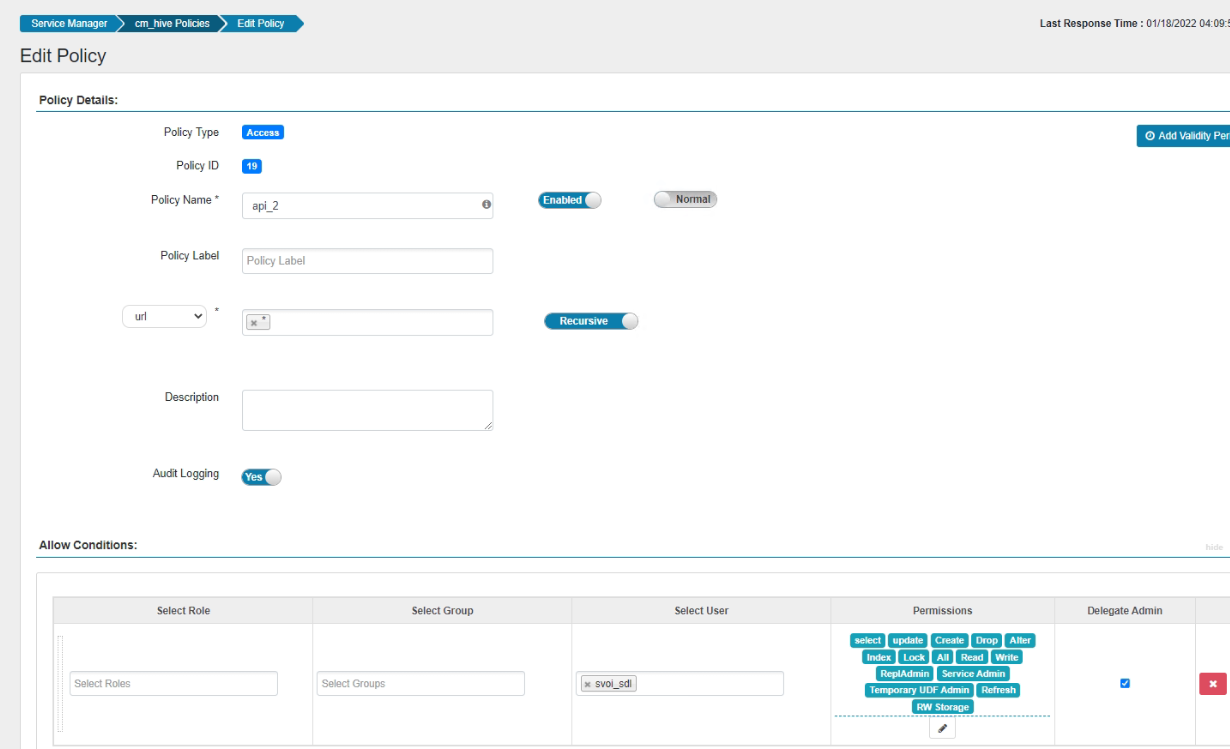


Рисунок 27 ― Создание второй политики Impala API

1. Создать политику на создание таблиц выполнить как на (Рисунок 28)

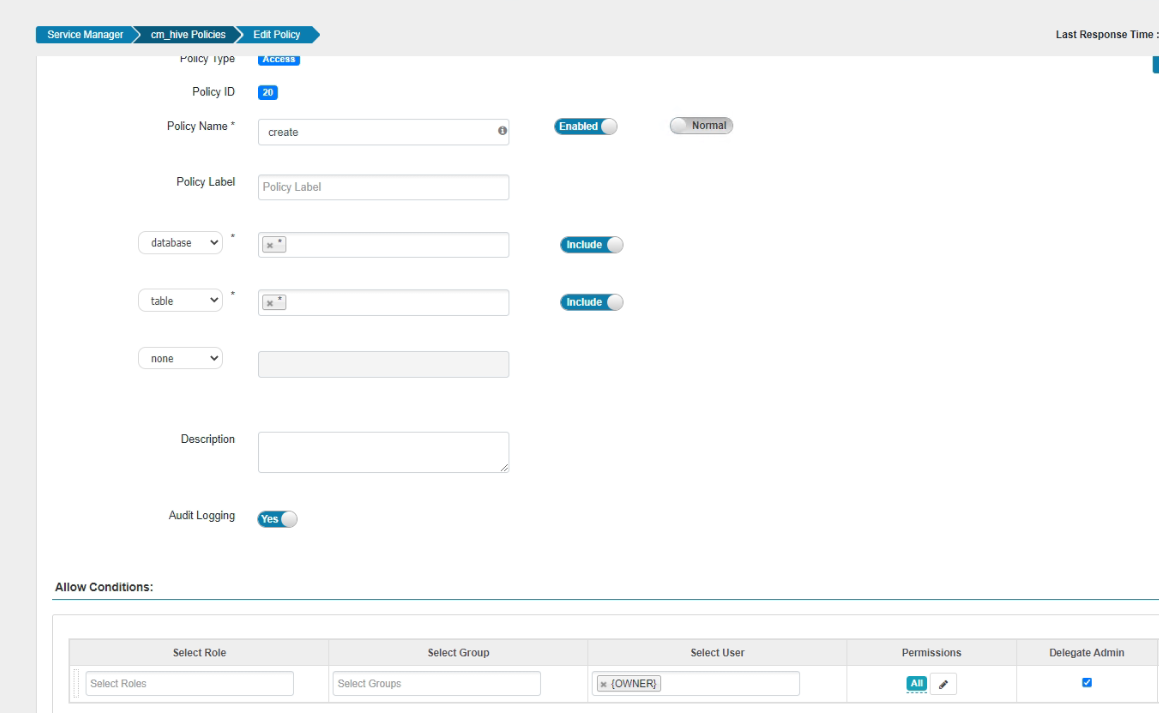


Рисунок 28 ― Добавление политики создания пользователей

Эта политика необходимо чтобы пользователь сам мог создавать собственные база данных и таблицы в Hue (которые не будут видеть другие пользователи без соответствующих прав).

1. Если у пользователя (не администратора) уже есть база данных и таблицы в ней, но он не является их владельцем, то необходимо создать политику для каждого такого пользователя как на (Рисунок 29). В колонке «database» имя базы данных соответствует имени пользователя из колонки «Select User»

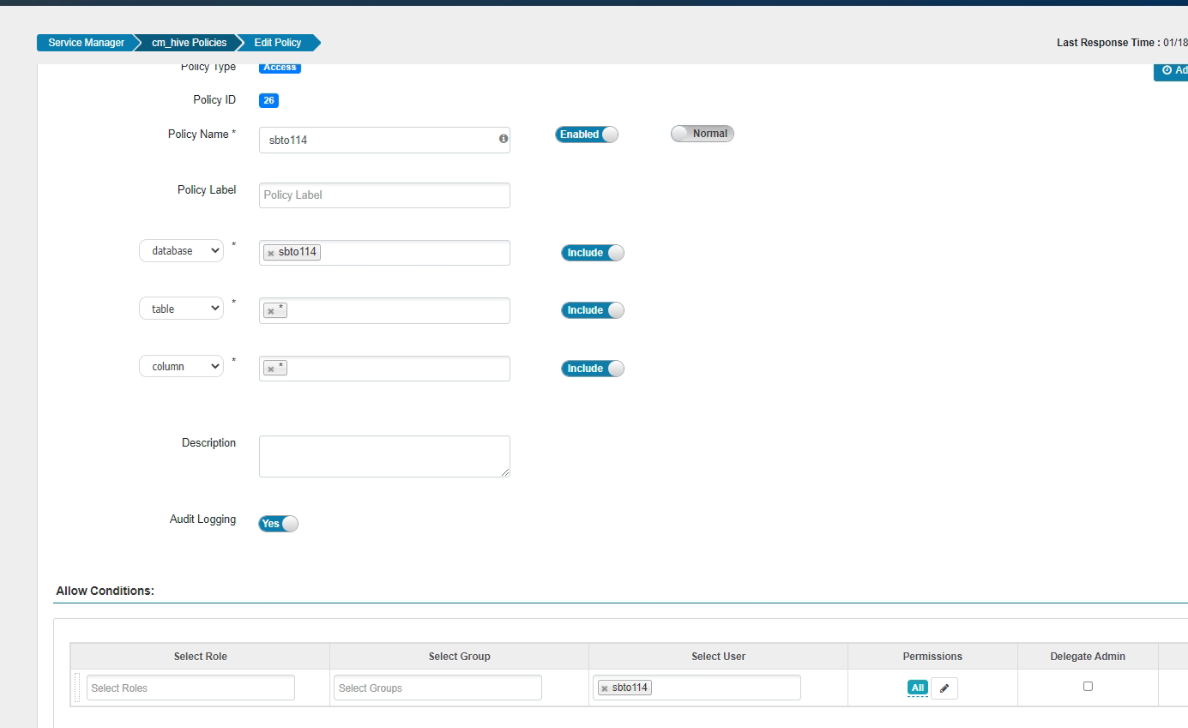


Рисунок 29 ― Политика пользователя на собственную базу данных

Приложение К  
(обязательное)  
Редактирование ресурсов контейнеров

В системе InnData редактирование ресурсов контейнеров происходит за счет изменения параметров конфигурации в файлах контейнеров.

Для редактирования ресурсов, выделяемых на контейнер, необходимо:

1. Открыть интерфейс k8s-dashboards и авторизоваться (Описание в РА в прложении «описание веб-интерфейса k8s-dashboards»
2. Перейти в раздел «Deployments» и выбрать нужный контейнер. Ниже представлен скриншот на примере контейнера «clickhouse01»

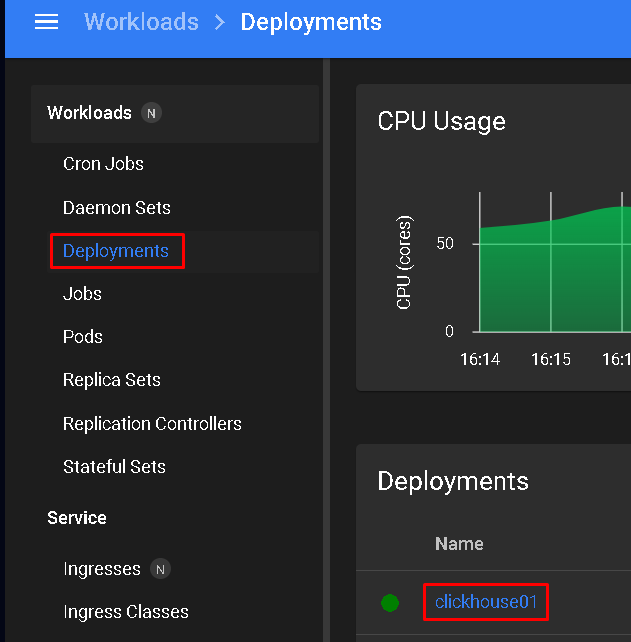


Рисунок 30 ― Раздел «Deployments» в сервисе k8s-dashboards

1. В правом верхнем углу нажать на значок редактирования в виде карандаша

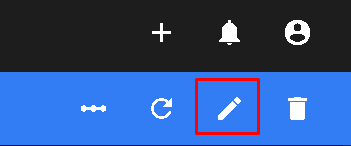


Рисунок 31 ― Редактирование параметров контейнера

1. Найти подраздел «resources» и выставить необходимые значения.

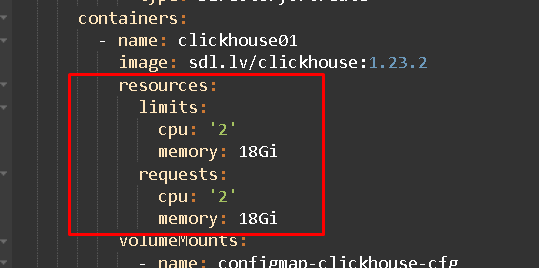


Рисунок 32 ― Редактирование ресурсов контейнера

Примечание: Некоторые контейнеры не ограничены в ресурсах ввиду того, что используют их по-минимуму, но если появилась необходимость явно ограничить ресурсы, то нужно добавить подраздел «resources:» в раздел «containers:» и выставить необходимые параметры как на скриншоте выше.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Таблица регистрации изменений** | | | | | | | | |
| Изм. | Номера листов (страниц) | | | | Всего  листов  (страниц)  в док. | Номер док. | Подп. | Дата |
| изме-ненных | заменен-ных | новых | аннулиро-ванных |
|  |  |  |  |  |  |  |  |  |