

**Инструкция по развертыванию экземпляра ПО  
«Система синхронизации файлов»**

# 1. Введение

## 1.1. Область применения

Настоящий документ предназначен для сотрудников эксплуатирующей организации и отражает основные функциональные возможности и порядок действий при выполнении операций, связанных с администрированием программного обеспечения «Система синхронизации файлов» (далее - «Система»)

## 1.2. Перечень выполняемых функций администратора/оператора

В перечень выполняемых функций администратора Системы входят:

- Установка и настройка Системы
- Реализация планов устранения сбоев и нетиповых нештатных ситуаций
- Выполнение сбора и предоставление в вышестоящую линию технической поддержки информации для воспроизведения технических проблем и выработки решений по их разрешению
- Реализация рекомендаций по устранению нештатных ситуаций, полученных с вышестоящей линии поддержки
- Восстановление работоспособности Системы при сбоях в работе функциональных модулей
- Разработка решения по устранению технических проблем в работе функциональных модулей

## 1.3. Уровень подготовки администратора/оператора

Администратор/оператор (далее по тексту Администратор) Системы должен обладать знаниями Javascript, уметь пользоваться и настраивать среду функционирования контейнеров или систему оркестрации, используемую на предприятии.

Рекомендуемая численность персонала для эксплуатации Системы — 1 штатная единица.

Администраторы Системы должны пройти обязательную общую и специальную подготовку для работы с Системой.

Общая подготовка должна включать в себя получение знаний и навыков работы с Системой в качестве администратора.

Специальная подготовка должна включать в себя получение знаний и навыков в объеме, необходимом для выполнения своих должностных обязанностей

## 1.4. Перечень документации

В состав документации, с которой необходимо ознакомиться администратору Системы входят:

- описание функциональных характеристик Системы
- описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения.

## 2. Установка Системы

В данном разделе будет описана установка Системы на Debian Linux. Предполагается, что были предварительно установлены также Docker, Docker Compose, RabbitMQ.

### 2.1. Системные требования к ПО

Минимальные аппаратные требования:

- Операционная система, способная запускать контейнеры. Предпочтительно Linux.
- Система управления контейнерной виртуализацией. Предпочтительно Docker Swarm или Kubernetes.
- Подключение к серверу очередей RabbitMQ
- Количество логических ядер процессора: 4
- Семейство процессоров: x86
- Частота процессора: 3.0. ГГц
- Объем установленной памяти: 16 Гб

#### 2.1.2. Минимальные требования к сторонним компонентам и/или системам, необходимым для установки и работы ПО

- Debian 11 (Открытая лицензия GNU)
- Docker 24.0.2 (open-source community edition)
- RabbitMQ (Открытая лицензия Mozilla Public License)
- Grafana Loki 2.6.1 (Открытая лицензия GNU)
- Grafana 9.2.2 (Открытая лицензия GNU)

#### 2.1.3. Языки программирования

При разработке Системы был использован язык программирования GoLang 1.20 (открытая лицензия BSD)

## 2.2. Порядок установки

1. Создайте папку /home/app
2. Смонтируйте диск с дистрибутивом в папку /mnt
3. Скопируйте из дистрибутива исходники из папки /mnt в папку /home/app
4. Смените текущую папку на /home/app и выполните команды  
sudo chown 10001:10001 ./volumes/loki  
sudo chown 472:472 ./volumes/grafana
5. Отредактируйте файл docker-compose.yml, в соответствии с пунктами 3.2 и 3.3 данного документа
6. Смените текущую папку на /home/app и выполните в ней команду  
docker compose -up -d --build
7. Войдите браузером на ваш сервер на порт 3000 в систему мониторинга с пользователем admin и паролем admin. Измените пароль на безопасный.

## 3. Настройка Системы

### 3.1. Общие сведения

В данном документе приводятся примеры настройки Системы с использованием среды Docker Compose. Настройка операционной системы, сервера очередей RabbitMQ, а также возможная настройка использования систем оркестрации, находятся вне компетенции этого документа и не будут тут описаны.

### 3.2. Модуль обработки входящих файлов

Для корректной работы модуля обработки входящих файлов, необходимо примонтировать к контейнеру три папки:

- Папка с входящими необработанными файлами. Модуль будет периодически сканировать содержимое этой папки и, при появлении новых файлов, отправлять их на синхронизацию
- Папка с успешно обработанными файлами. В случае, если синхронизация прошла успешно, в эту папку будут перемещены синхронизированные файлы.
- Папка с файлами, при синхронизации которых возникла ошибка. В случае ошибки синхронизации, файлы, которые не смогли быть синхронизированы, будут перемещены в эту папку. Администратор должен самостоятельно определить причину ошибки синхронизации, устранить её, а затем переложить файлы в папку с входящими необработанными файлами, дабы произвести синхронизацию.

Также для модуля должны быть прописаны следующие переменные окружения:

- METRICS\_PORT - порт к подсистеме проверки работоспособности.
- FILES\_IN\_PATH — путь к папке с входящими необработанными файлами в файловой системе контейнера.
- FILES\_OK\_PATH — путь к папке с успешно обработанными файлами в файловой системе контейнера.
- FILES\_ERROR\_PATH — путь к папке с файлами, при обработке которых возникла ошибка, в файловой системе контейнера.
- AMQP\_HOST — имя хоста или IP сервера RabbitMQ, с указанием порта и учетной записи. Пример смотрите ниже.
- AMQP\_EXCHANGE — точка обмена на сервере RabbitMQ
- AMQP\_COMMAND\_TIMEOUT— предельный интервал времени на синхронизацию файла
- FILES\_COMMAND — параметр, по которому подписчик будет отфильтровывать свои сообщения в очереди сервера
- FILES\_CHECK\_INTERVAL — интервал периодичности проверки входящих файлов в секундах
- FILES\_PATH\_PREFIX — префикс, который может быть добавлен к пути при обработке файлов.

- LOG\_LEVEL - уровень логгирования. Поддерживаемые значения:
  - error
  - warn
  - info
  - debug
  - trace

#### Пример настройки модуля:

```

filesin:
  build:
    context: ./files-in/
  restart: always
  volumes:
    - ./files/source/in:/files/in
    - ./files/source/ok:/files/ok
    - ./files/source/error:/files/error
  environment:
    AMQP_HOST: amqp://rabbit:QHfuMX23H2Wk18xRhv8d@host.docker.internal:5672/
    AMQP_EXCHANGE: transport
    AMQP_COMMAND_TIMEOUT: 30s
    FILES_IN_PATH: /files/in
    FILES_OK_PATH: /files/ok
    FILES_ERROR_PATH: /files/eror
    FILES_COMMAND: FILES_SYNC
    FILES_CHECK_INTERVAL: 10
    FILES_PATH_PREFIX: ""
    LOG_LEVEL: trace
  extra_hosts:
    - "host.docker.internal:host-gateway"

```

### 3.3. Модуль обработки входящих файлов

Для корректной работы модуля обработки исходящих файлов, необходимо примонтировать к контейнеру папку, в которую он будет складывать обработанные файлы.

Также для модуля должны быть прописаны следующие переменные окружения:

- METRICS\_PORT - порт к подсистеме проверки работоспособности.
- FILES\_PATH — путь к папке с файлами, при обработки которых возникла ошибка, в файловой системе контейнера.
- AMQP\_HOST — имя хоста или IP сервера RabbitMQ, с указанием порта и учетной записи. Пример смотрите ниже.
- AMQP\_EXCHANGE — точка обмена на сервере RabbitMQ
- FILES\_COMMAND — параметр, по которому подписчик будет отфильтровывать свои сообщения в очереди сервера
- FILES\_PATH\_PREFIX — префикс, который может быть добавлен к пути при обработке файлов.

- LOG\_LEVEL - уровень логгирования. Поддерживаемые значения:
  - error
  - warn
  - info
  - debug
  - trace

#### Пример настройки модуля:

```

filesout:
  build:
    context: ./files-out/
  restart: always
  volumes:
    - ./files/destination:/files
  environment:
    AMQP_HOST: amqp://rabbit:QHfuMX23H2Wk18xRhv8d@host.docker.internal:5672/
    AMQP_EXCHANGE: transport
    FILES_PATH: /files
    FILES_PATH_PREFIX: ""
    FILES_COMMAND: FILES_SYNC
    LOG_LEVEL: trace
  extra_hosts:
    - "host.docker.internal:host-gateway"

```

### 3.4. Модуль удаленной обработки входящих файлов

Для модуля должны быть прописаны следующие переменные окружения:

- METRICS\_PORT - порт к подсистеме проверки работоспособности.
- AMQP\_HOST — имя хоста или IP сервера RabbitMQ, с указанием порта и учетной записи. Пример смотрите ниже.
- AMQP\_EXCHANGE — точка обмена на сервере RabbitMQ
- SFTP\_HOST — имя или IP сервера
- SFTP\_PORT — порт ssh на сервере
- SFTP\_USER — имя пользователя на сервере
- SFTP\_PASSWORD — пароль пользователя на сервере
- SFTP\_PATH — путь к целевой папке, на сервере.
- SFTP\_COMMAND — параметр, по которому подписчик будет отфильтровывать свои сообщения в очереди сервера
- LOG\_LEVEL - уровень логгирования. Поддерживаемые значения:
  - error
  - warn
  - info
  - debug
  - trace

## Пример настройки модуля:

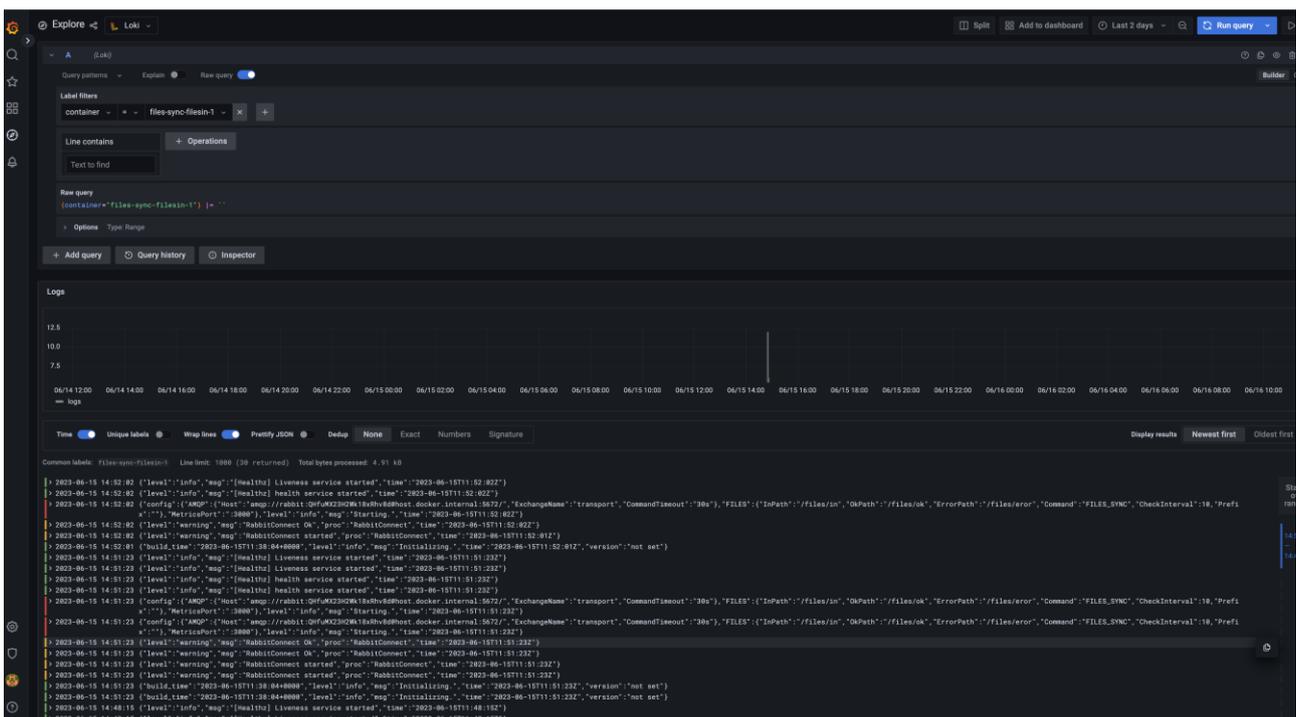
```
sftpout:  
build:  
  context: ./sftp-out/  
restart: always  
environment:  
  AMQP_HOST: amqp://rabbit:QHfuMX23H2Wk18xRhv8d@host.docker.internal:5672/  
  AMQP_EXCHANGE: transport  
  SFTP_PATH: /home/user/files  
  SFTP_HOST: "127.0.0.1"  
  SFTP_PORT: 22  
  SFTP_USER: user  
  SFTP_PASSWORD: password  
  SFTP_COMMAND: FILES_SYNC  
  LOG_LEVEL: trace
```

## 4. Система мониторинга

В качестве системы мониторинга используется Grafana Loki — это набор компонентов для полноценной системы работы с логами. Loki-стек состоит из трёх компонентов: Promtail, Loki, Grafana. Promtail собирает логи, обрабатывает их и отправляет в Loki. Loki их хранит. A Grafana умеет запрашивать данные из Loki и показывать их. Loki можно использовать не только для хранения логов и поиска по ним. Весь стек даёт большие возможности по обработке и анализу поступающих данных

Чтобы открыть интерфейс системы мониторинга, перейдите в браузере на IP Вашего сервера и порт 3000. Если Вы входите туда в первый раз, используйте логин admin и пароль admin. После первого входа система попросит Вас изменить пароль на безопасный.

Интерфейс выглядит так:



Выберите в меню пункт «Explore» - Вы увидите страницу поиска логов.

Сам запрос состоит из двух частей: selector и filter. Selector — это поиск по индексированным метаданным (лейблам), которые присвоены логам, а filter — поисковая строка или регэксп, с помощью которого отфильтровываются записи, определённые селектором.

Выберите в разделе Label filters в ниспадающем списке Label значение container, а в ниспадающем списке value выберите нужный контейнер. Выполните запрос Run query и Вы увидите логи выбранного контейнера.