

ООО «Финтек Системс»

**«Сервис управления коммуникационными стратегиями»**  
Документация, необходимая для установки программного обеспечения

2023 г.

## СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ .....	3
1.1. Назначение программного комплекса .....	3
1.2. Область применения программного комплекса.....	3
1.3. Функциональные характеристики .....	3
2. СОСТАВ РАЗВЕРТЫВАНИЯ .....	5
2.1. Основные положения.....	5
3. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	6
3.1. Общие положения .....	6
3.2. Управление лицензиями .....	6
3.3. Подключение к серверу по SSH .....	6
3.4. Установка APACHE KAFKA .....	8
3.5. Установка REDIS .....	14
3.6. Установка POSTGRESQL.....	17
3.7. Установка MONGODB .....	20
3.8. Установка системы контроля версий GIT .....	22
3.9. Настройка конфигурационного файла.....	23
3.10. Установка Docker Desktop.....	25
3.11. Сборка контейнера .....	26

# **1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ**

## **1.1. Назначение программного комплекса**

**1.1.1.** «Сервис управления коммуникационными стратегиями» — программное обеспечение (далее – ПО), предназначенное для обеспечения маршрутизации с интегрированными провайдерами рассылок, приоритезацию трафика и контроль статусов обработки сообщений.

## **1.2. Область применения программного комплекса**

**1.2.1.** Программное обеспечение обеспечивает возможность маршрутизации рассылок между интегрированными провайдерами и веб – сервисом Пользователя посредством распределенной системы обмена сообщениями, в том числе пакетной отправки сообщений, в сети Интернет. Программный комплекс является связующим звеном между провайдером и веб – сервисом Пользователя.

## **1.3. Функциональные характеристики**

**1.3.1.** Программное обеспечение используется для реализации следующих функций:

- Получение запросов на отправку отдельного сообщения или пакета сообщений формате JSON по REST API или Kafka;
- Маршрутизация сообщений между интегрированными провайдерами сообщений;
- Создание очереди сообщений с приоритезацией посредством интеграции с сервером структур данных Redis;
- Контроль времени отправки сообщения;
- Пакетная отправка сообщений провайдеру для снижения нагрузки на сеть;
- Запись и хранение входящих запросов на отправку сообщений и полученных ответов от провайдеров в базе данных MongoDB с последующей интеграцией с системой Prometheus для анализа данных мониторинга.

- Сбор системной информации о работе ПО (логирование взаимодействия) посредством системы управления логами GrayLog.
- Предоставление доступа к эндпоинтам для подключения систем мониторинга.

## **2. СОСТАВ РАЗВЕРТЫВАНИЯ**

### **2.1. Основные положения**

**2.1.1.** Программное обеспечение «Сервис управления коммуникационными стратегиями» реализовано посредством клиент-серверной архитектуры.

**2.1.2.** Клиент – любая операционная система, имеющая соединение с сервером программного обеспечения и способная взаимодействовать посредством REST API или Kafka.

**2.1.3.** Сервер – операционная система Debian 11 и выше.

**2.1.4.** Серверная часть программного обеспечения реализована в виде в докер-контейнера (образа) и доступна для запуска в любой среде (Linux, Windows).

**2.1.5.** Развертывание в продуктивной среде производится посредством публикации программного обеспечения после успешного прохождения интеграционного и регрессионного тестирования.

## **3. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

### **3.1. Общие положения**

**3.1.1.** Установка настоящего программного обеспечения проводится на управляющем ПК.

**3.1.2.** На управляющий ПК из состава Изделия должна быть установлена операционная система Debian 11 или выше посредством соответствующей программы установки (в комплект поставки Изделия не входит).

**3.1.3.** Убедитесь, что вы установили все необходимые обновления для операционной системы на управляющем ПК.

**3.1.4.** Для подготовки к установке программного обеспечения на Debian 11 запустите терминал посредством нажатия комбинации клавиш **Ctrl+Alt+T**.

### **3.2. Управление лицензиями**

**3.2.1.** Доступ к программному обеспечению «Сервис управления коммуникационными стратегиями» обеспечивается посредством лицензионных ключей. Управление лицензиями для доступа к программному обеспечению выполняется посредством сервера лицензий. Доступны следующие лицензии для использования программного обеспечения: серверная и пользовательская. Лицензии предоставляются после оформления заявки на подключение ПО и получения подтверждения о готовности к началу проведения технических работ.

### **3.3. Подключение к серверу по SSH**

**3.3.1.** Для обеспечения безопасного обмена данными между элементами системы необходимо выполнить следующие настройку SSH.

**3.3.2.** Для начала настройки SSH необходимо установить пакет для подключения посредством ввода следующей команды:

```
# dnf install openssh-server
```

**3.3.3.** Для начала работы с SSH необходимо запустить пакет для подключения посредством ввода следующей команды:

```
# systemctl start sshd
```

**3.3.4.** Для проверки работы пакета необходимо ввести следующую команду:

```
# systemctl status sshd
```

**3.3.5.** Для включения пакета после перезагрузки управляющего компьютера необходимо ввести следующую команду:

```
# systemctl enable sshd
```

**3.3.6.** Для приема входящего трафика по SSH (порт 22) необходимо добавить разрешение соединения (исключение) в Firewall посредством ввода следующей команды:

```
# firewall-cmd --zone=public --permanent --add-service=ssh
```

**3.3.7.** Для подключения к серверу (Пример: с IP 10.1.1.1) по SSH необходимо выполнить следующую команду в качестве пользователя root с удаленного компьютера (управляющего компьютера):

```
$ ssh root@10.1.1.1
```

### 3.4. Установка АРАСНЕ КАФКА

3.4.1. Для обмена сообщениями между ПО и приложением – источником в режиме реального времени необходимо установить распределенную систему АРАСНЕ КАФКА.

3.4.2. Для установки АРАСНЕ КАФКА необходимо установить пакет JAVA посредством ввода в терминале следующей команды:

```
dnf install java-11-openjdk-devel -y
```

3.4.3. Проверьте версию установленного пакета JAVA посредством ввода в терминале следующей команды:

```
java --version
```

3.4.4. Для установки АРАСНЕ КАФКА необходимо скачать дистрибутив посредством ввода в терминале следующей команды:

```
wget https://dlcdn.apache.org/kafka/3.0.0/kafka_2.13-3.0.0.tgz
```

3.4.5. Для установки АРАСНЕ КАФКА необходимо распаковать дистрибутив посредством ввода в терминале следующей команды:

```
tar -xvzf kafka_2.13-3.0.0.tgz
```

3.4.6. Для установки АРАСНЕ КАФКА необходимо переместить распакованный дистрибутив в директорию **/usr/local directory** посредством ввода в терминале следующей команды:

```
mv kafka_2.13-3.0.0/ /usr/local/kafka
```

3.4.7. Для одновременного запуска АРАСНЕ КАФКА и ZOOKEEPER необходимо создать службу **systemd** для каждого из сервисов.

3.4.8. Для создания службы **systemd** для ZOOKEEPER необходимо ввести в терминале следующую команду:

```
nano /etc/systemd/system/zookeeper.service
```



**3.4.9.** Для работы службы **systemd** для **ZOOKEEPER** в открывшемся файле необходимо добавить следующее описание:

```
[Unit]
Description=Apache Zookeeper server
Documentation=http://zookeeper.apache.org
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
ExecStart=/usr/bin/bash /usr/local/kafka/bin/zookeeper-server-start.sh
/usr/local/kafka/config/zookeeper.properties
ExecStop=/usr/bin/bash /usr/local/kafka/bin/zookeeper-server-stop.sh
Restart=on-abnormal

[Install]
WantedBy=multi-user.target
```

**3.4.10.** Для создания службы **systemd** для **АРАСНЕ КАФКА** необходимо ввести в терминале следующую команду:

```
nano /etc/systemd/system/kafka.service
```

**3.4.11.** Для работы службы **systemd** для **APACHE KAFKA** в открывшемся файле необходимо добавить следующее описание:

```
[Unit]
Description=Apache Kafka Server
Documentation=http://kafka.apache.org/documentation.html
Requires=zookeeper.service

[Service]
Type=simple
Environment="JAVA_HOME=/usr/lib/jvm/jre-11-openjdk"
ExecStart=/usr/bin/bash /usr/local/kafka/bin/kafka-server-start.sh /usr/local/kafka/config/server.properties
ExecStop=/usr/bin/bash /usr/local/kafka/bin/kafka-server-stop.sh

[Install]
WantedBy=multi-user.target
```

**3.4.12.** Для работы созданных служб **systemd** обоих сервисов необходимо перезагрузить службу **systemd** посредством ввода в терминале следующей команды:

```
systemctl daemon-reload
```

**3.4.13.** Для запуска **APACHE KAFKA** и **ZOOKEEPER** необходимо ввести в терминале следующую команду:

```
systemctl start zookeeper
```

**3.4.14.** Для проверки работы **APACHE KAFKA** и **ZOOKEEPER** необходимо ввести в терминале следующую команду:

```
systemctl status zookeeper kafka
```

**3.4.15.** Для создания виртуального хранилища сообщений (журнала записей) **АРАСНЕ КАФКА** необходимо создать тему посредством ввода в терминале следующей команды:

```
cd /usr/local/kafka/  
bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitio  
topic1  
systemctl start kafka  
systemctl enable zookeeper  
systemctl enable kafka
```

**3.4.16.** Для создания виртуального хранилища сообщений (журнала записей) **АРАСНЕ КАФКА** необходимо проверить тему посредством ввода в терминале следующей команды:

```
bin/kafka-topics.sh --list --bootstrap-server localhost:9092
```

**3.4.17.** **АРАСНЕ КАФКА** предоставляет два API:

- Producer API — для отправки сообщений в брокер;
- Consumer API — подписка на темы и получение сообщений из брокера;

**3.4.18.** Для создания события (event1) **АРАСНЕ КАФКА** необходимо ввести следующую команду:

```
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic event1
```

**3.4.19.** Для получения данных о событии (event1) **АРАСНЕ КАФКА** в режиме реального времени необходимо ввести следующую команду:

```
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic event1 --from-beginning
```

**3.4.20.** Для аутентификации по протоколу **SASL\_PLAINTEXT** с механизмом **PLAIN** необходимо выполнить следующие настройки безопасности.

**3.4.21.** Для конфигурации сервера **КАФКА** необходимо открыть файл конфигурации **server.properties** посредством следующей команды:

```
nano kafka/config/server.properties
```

**3.4.22.** В конфигурационном файле необходимо указать **LISTENER** посредством ввода значений, где **HOST.NAME** – имя хоста, **PORT** - порт:

```
listeners=SASL_PLAINTEXT://host.name:port
```

**3.4.23.** Для настройки конфигурации сервера необходимо добавить файл **JAAS** в каталог конфигураций посредством ввода данных, где **USERNAME** – имя пользователя, **PASSWORD** - пароль:

```
KafkaServer {  
  org.apache.kafka.common.security.plain.PlainLoginModule required  
  username=" kafka-user"  
  password="XXXXXX"  
};
```

**3.4.24.** Укажите каталог для расположения **JAAS** файла в качестве **JVM** для брокера посредством ввода следующих данных:

```
-Djava.security.auth.login.config=/etc/kafka/kafka_server_jaas.conf
```

**3.4.25.** Укажите **SASL** порт and **SASL** механизм в **SERVER.PROPERTIES** посредством ввода следующих данных:

```
listeners=SASL_SSL://host.name:port  
security.inter.broker.protocol=SASL_SSL  
sasl.mechanism.inter.broker.protocol=PLAIN  
sasl.enabled.mechanisms=PLAIN
```

**3.4.26.** Для конфигурации клиента **KAFKA** необходимо открыть файл конфигурации издателя **PRODUCER.PROPERTIES** посредством следующей команды:

```
nano kafka/config/producer.properties
```

**3.4.27.** Для настройки конфигурации клиента необходимо указать следующие данные, где **USERNAME** – имя пользователя, **PASSWORD** - пароль:

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \  
  username="kafka-user" \  
  password="XXXXXX";
```

**3.4.28.** Укажите **SASL** протокол and **SASL** механизм в **PRODUCER.PROPERTIES** посредством ввода следующих данных:

```
security.protocol=SASL_PLAINTEXT  
sasl.mechanism=PLAIN
```

#### **ПРИМЕЧАНИЕ**

Дополнительную информацию по работе **АРАСНЕ КАФКА** Вы можете получить на официальном сайте: <https://kafka.apache.org/documentation/>

### 3.5. Установка REDIS

**3.5.1.** Для подключения к ПО быстрого хранилища данных для использования в качестве очереди необходимо установить Redis.

**3.5.2.** Для установки **REDIS** запустите терминал посредством нажатия клавиш

**3.5.3.** Запустите установку **REDIS** посредством ввода следующей команды в терминале:

```
sudo dnf install redis nano
```

**3.5.4.** Подтвердите запуск установки **REDIS** посредством ввода **Y** в терминале:

Output

...

Total download size: 2.0 M

Installed size: 7.4 M

Is this ok [y/N]: y

**3.5.5.** После завершения установки **REDIS** необходимо указать данные настройки в конфигурационном файле.

**3.5.6.** Для открытия конфигурационного файла необходимо ввести следующую команду в терминале:

```
sudo nano /etc/redis/redis.conf
```

**3.5.7.** В конфигурационном файле необходимо указать максимальное значение памяти в соответствии с доступной памятью на управляющем ПК:

```
maxmemory 256mb
```

```
maxmemory-policy allkeys-lru
```

**3.5.8.** Сохраните изменения и закройте конфигурационный файл посредством нажатия **Ctrl+X**.

**3.5.9.** Для перезапуска **REDIS** необходимо ввести следующую команду в терминале:

```
sudo systemctl restart redis-server
```

**3.5.10.** Для получения дополнительной информации о состоянии **REDIS** необходимо ввести следующую команду в терминале:

```
sudo systemctl status redis
```

**3.5.11.** Для проверки корректности работы **REDIS** необходимо ввести следующую команду в терминале:

```
redis-cli ping
```

**3.5.12.** Пример ответа при корректной работе **REDIS**:

```
Output  
PONG
```

**3.5.13.** Для настройки конфигураций **REDIS** необходимо открыть конфигурационный файл посредством ввода следующей команды в терминале:

```
sudo nano /etc/redis/redis.conf
```

**3.5.14.** В конфигурационном файле строка **bind** не должна быть отключена посредством **#**. Пример корректного отображения строки **bind**:

```
bind 127.0.0.1 -::1
```

**3.5.15.** Для продолжения настройки Redis необходимо сохранить и закрыть конфигурационный файл.

**3.5.16.** Для настройки доступа к Redis необходимо задать пароль.

**3.5.17.** Для задания пароля к **REDIS** необходимо открыть конфигурационный файл посредством ввода следующей команды в терминале:

```
sudo nano /etc/redis/redis.conf
```

**3.5.18.** В конфигурационном файле в разделе **SECURITY** необходимо задать пароль посредством указания в строке **requirepass** через пробел, предварительно удалив знак **#**:

```
requirepass yourpassword
```

**3.5.19.** Для продолжения настройки Redis необходимо сохранить и закрыть конфигурационный файл.

**3.5.20.** Для продолжения настройки Redis необходимо перезапустить Redis посредством ввода следующей команды в терминале:

```
sudo systemctl restart redis
```

#### **ПРИМЕЧАНИЕ**

Дополнительную информацию по работе **REDIS** Вы можете получить на официальном сайте: <https://redis.io/docs/manual/>



## 3.6. Установка PostgreSQL

**3.6.1.** Установка PostgreSQL производится из репозитория AppStream посредством менеджера пакетов DNF.

**3.6.2.** Для запуска установки PostgreSQL из репозитория AppStream необходимо выбрать версию DNF пакета PostgreSQL посредством ввода следующей команды в терминале:

```
dnf module list postgresql
```

**3.6.3.** Пример ответа за запрос:

Name	Stream	Profiles	Summary
postgresql	9.6	client, server [d]	PostgreSQL server and client module
postgresql	10 [d]	client, server [d]	PostgreSQL server and client module
postgresql	12	client, server [d]	PostgreSQL server and client module
postgresql	13	client, server [d]	PostgreSQL server and client module

**3.6.4.** Для подключения DNF пакета PostgreSQL из репозитория AppStream необходимо ввести следующую команду в терминале:

```
sudo dnf module enable postgresql:12
```

**3.6.5.** Подтвердите подключения DNF пакета PostgreSQL из репозитория AppStream посредством ввода Y в терминале:

```
Output
Package Architecture Version Repository Size
Enabling module streams:
 postgresql 12
Transaction Summary
Is this ok [y/N]: y
```

**3.6.6.** Для запуска установки DNF пакета PostgreSQL из репозитория AppStream необходимо ввести следующую команду в терминале:

```
sudo dnf install postgresql-server
```

**3.6.7.** Подтвердите запуск установки **DNF** пакета **POSTGRESQL** из репозитория **AppStream** посредством ввода **Y** в терминале:

```
Output
```

```
...
```

```
Install 4 Packages
```

```
Total download size: 16 M
```

```
Installed size: 62 M
```

```
Is this ok [y/N]: y
```

**3.6.8.** Для создания кластера базы данных **POSTGRESQL** необходимо создать директорию хранения базы данных посредством ввода следующей команды:

```
sudo postgresql-setup --initdb
```

**3.6.9.** Для запуска службы **systemctl POSTGRESQL** необходимо ввести следующую команду:

```
sudo systemctl start postgresql
```

**3.6.10.** Для запуска службы **systemctl POSTGRESQL** в режиме автозапуска необходимо ввести следующую команду:

```
sudo systemctl start postgresql
```

**3.6.11.** При установке **POSTGRESQL** создается учетная запись пользователя **postgres**, связанная с ролью **postgres** по умолчанию. Для использования **POSTGRESQL** необходимо войти в эту учетную запись.

**3.6.12.** Для входа в учетную запись **postgres** необходимо ввести следующую команду:

```
sudo -i -u postgres
```

**3.6.13.** Для работы с базами данных **POSTGRESQL** необходимо перейти в терминал посредством ввода следующей команды:

**3.6.14.** Для создания новой базы данных **POSTGRESQL** необходимо ввести следующую команду с наименованием базы данных:

```
createdb newdatabase
```

**3.6.15.** Для создания новой базы данных **POSTGRESQL** необходимо добавить описание данных посредством ввода следующей команды:

```
CREATE TABLE table_name (  
    column_name1 col_type (field_length) column_constraints,  
    column_name2 col_type (field_length),  
    column_name3 col_type (field_length)  
).
```

#### **ПРИМЕЧАНИЕ**

Дополнительную информацию по работе **POSTGRESQL** Вы можете получить на официальном сайте: <https://www.postgresql.org/docs/12/index.html>

### 3.7. Установка MONGODB

**3.7.1.** Для запуска установки **MONGODB** необходимо добавить репозиторий **MONGODB REPOSITORY** посредством ввода следующей команды:

```
cat > /etc/yum.repos.d/mongodb.repo << 'EOL'  
[mongodb-org-4.4]  
name=MongoDB Repository  
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.4/x86_64/  
gpgcheck=1  
enabled=1  
gpgkey=https://www.mongodb.org/static/pgp/server-4.4.asc  
EOL
```

**3.7.2.** Для запуска установки **MONGODB** необходимо выполнить следующие команды:

```
sudo dnf update
```

```
sudo dnf install mongodb-org
```

**3.7.3.** После завершения установки **MONGODB** необходимо проверить версию данного программного обеспечения:

```
mongod --version
```

**3.7.4.** Для запуска **MONGODB** в режиме автозапуска необходимо ввести следующую команду:

```
sudo systemctl enable mongod
```

**3.7.5.** Для запуска **MONGODB** необходимо ввести следующую команду:

```
sudo systemctl start mongod
```

**3.7.6.** Для получения доступа к базе данных в **MONGODB** необходимо перейти в режим терминала посредством следующей команды:

```
mongo
```

**3.7.7.** Для создания новой базы данных **MONGODB** необходимо ввести следующую команду с наименованием базы данных:

```
use newdatabase
```

**3.7.8.** Для создания новой базы данных и добавления данных **MONGODB** необходимо создать коллекцию посредством ввода следующей команды:

```
db.createCollection("collection_name")
```

#### **ПРИМЕЧАНИЕ**

Дополнительную информацию по работе **MONGODB** Вы можете получить на официальном сайте: <https://www.mongodb.com/docs/manual/>

## **3.8. Установка системы контроля версий GIT**

**3.8.1.** Для работы с образом программного обеспечения необходимо установить систему контроля версий **GIT** на управляющий ПК.

**3.8.2.** Для установки **GIT** необходимо ввести в терминале следующую команду:

```
sudo dnf install git
```

**3.8.3.** На управляющем компьютере создайте каталог (папку), в которую будет клонирован образ программного обеспечения посредством ввода в терминал следующей команды:

```
mkdir newdirectory -p
```

**3.8.4.** Перейдите в созданный каталог (папку) посредством ввода в терминал следующей команды:

```
cd newdirectory
```

**3.8.5.** Создайте **GIT** репозиторий посредством ввода следующей команды:

```
git init
```

**3.8.6.** Клонировать репозиторий с образом программного обеспечения посредством ввода следующей команды в терминале:

```
git clone https://git.ftsistemas.ru:5050/crt/comgateway
```

**3.8.7.** Следуйте дальнейшим инструкциям системы контроля версий **GIT**.

### 3.9. Настройка конфигурационного файла

**3.9.1.** Для подключения сервисов к программному обеспечению необходимо выполнить настройку конфигурационного файла.

**3.9.2.** Для настройки конфигурационного файла программного обеспечения необходимо открыть файл посредством ввода в терминале следующей команды:

```
sudo nano /app/appsettings.Production.json
```

**3.9.3.** В конфигурационном файле представлены следующие данные:

```
{
  "FeatureManagement": {
    "EnableNewKafkaClientInteraction": true,
    "ForceDebugFeature": true
  },
  "BackgroundServices": {
    "DistributeMessageBackgroundService": {
      "ProcessCount": 16
    },
    "DistributeSendRequestIdentityBackgroundService": {
      "ProcessCount": 4
    },
    "ReceiveProviderMessageStatusBackgroundService": {
      "ProcessCount": 4
    }
  },
  "KafkaConfigs": {
    "ConsumerConfig": {
      "BootstrapServers": "[kafka_ip]:9092",
      "GroupId": "group-comgateway1",

      "SecurityProtocol": "SaslPlaintext",
      "SaslMechanism": "Plain",
      "SaslUsername": "kafka-user",
      "SaslPassword": "XXXXXX",
    }
  }
}
```

```
        "AutoOffsetReset": "Latest"
    },

    "ProducerConfig": {
        "BootstrapServers": "[kafka_ip]:9092",
        "EnableIdempotence": false,

        "SecurityProtocol": "SaslPlaintext",
        "SaslMechanism": "Plain",
        "SaslUsername": "kafka-user",
        "SaslPassword": "XXXXXX"
    }
},

"Redis": {
    "Connection": {
        "Endpoint": "[redis_ip]:6379",
        "Password": "YYYYYY"
    }
},

"SmsService": {
    "IsDebug": true,
    "HttpClient": {
        "BaseAddress": "https://sms.example.com"
    }
},

"MongoClient": {
    "ConnectionString": "mongodb://user:pass@[mongo_ip]:27017"
},

"DbContext": {
    "Database": "ComGateWay"
}
}
```

**3.9.4.** Для настройки конфигурационного файла программного обеспечения необходимо указать данные для каждого из сервисов для подключения.



### 3.10. Установка Docker Desktop

**3.10.1.** Для запуска образа программного обеспечения необходимо установить **Docker Desktop** на управляющий ПК.

**3.10.2.** Для установки **Docker Desktop** загрузите дистрибутив посредством перехода по ссылке: <https://docs.docker.com/desktop/install/linux-install/>.

**3.10.3.** Запустите дистрибутив для установки **Docker Desktop** и следуйте дальнейшим инструкциям.

#### ПРИМЕЧАНИЕ

Дополнительную информацию по установке **Docker Desktop** Вы можете получить на официальном сайте:

<https://docs.docker.com/desktop/install/linux-install/>

### 3.11. Сборка контейнера

**3.11.1.** Сборка контейнера программного обеспечения выполняется из клонированного каталога **comgateway** на управляющем ПК.

**3.11.2.** Для запуска сборки контейнера программного обеспечения введите следующую команду в терминале:

```
$ docker-compose up -d
```

**3.11.3.** Запустите программное обеспечение посредством открытия браузера и ввода в адресную строку адреса:

```
https://comgateway.ftsystems.ru/health
```

**3.11.4.** Пример ответа при успешном запуске программного обеспечения:

```
{"status":"Healthy","totalDuration":"00:00:00.1677754",  
"entries":{"Kafka":{"data":{},"duration":"00:00:00.0984045",  
"status":"Healthy","tags":["deep"]},  
"MongoDb":{"data":{},"duration":"00:00:00.1564111","status":"Healthy","tags":[]}}}
```