

Общество с ограниченной ответственностью «КОММИТ»
ИНН 5040198927, КПП 504001001, ОГРН 1255000107746
140105, Московская область, г. Раменское, Северное шоссе, стр. 4А
Тел.: +7 (926) 536-77-42, e-mail: vegaliry@mail.ru

ИНСТРУКЦИЯ
по установке (развёртыванию) и администрированию
программного обеспечения «Мой город» (МГОР)
Редакция 2

1. Назначение документа

Документ содержит сведения, необходимые для самостоятельной установки экземпляра программного обеспечения «Мой город» (МГОР) на технических средствах заказчика или проверяющей организации, а также для последующего администрирования установленного экземпляра. Документ рассчитан на системного администратора, владеющего основами администрирования операционных систем семейства Linux и средств контейнеризации.

2. Состав программного обеспечения

ПО имеет клиент-серверную архитектуру и поставляется в виде набора контейнерных образов. В состав поставки входят следующие компоненты:

Компонент	Назначение	Базовый образ
Сервер приложения	Серверная часть: прикладная логика, REST API, обработка фоновых заданий. Платформа Node.js, каркас NestJS. Внутренний порт 3000	smartcity-backend
Клиентское приложение	Пользовательский интерфейс, прогрессивное веб-приложение (PWA). React, сборка Vite, раздача статических файлов	smartcity-frontend-user
Служба миграций	Приведение схемы базы данных к актуальной версии. Запускается из того же образа, что и сервер приложения	smartcity-backend
СУБД	Хранение данных, включая пространственные	postgis/postgis:16-3.4
Кеш и очереди	Кеширование, хранение сессий, очереди фоновых заданий	redis:8.6-alpine
Обратный прокси-сервер	Маршрутизация запросов, терминирование TLS, выпуск сертификатов	traefik (или иной)

Дополнительно используются внешние по отношению к ПО сервисы, размещаемые на технических средствах заказчика: S3-совместимое объектное хранилище (например, MinIO) для файлов и фотографий, сервис маршрутизации по дорожной сети и сервис геокодирования. Все перечисленные сервисы имеют свободные лицензии и разворачиваются на территории Российской Федерации.

3. Требования к техническим и программным средствам

Параметр	Минимально	Рекомендуется
Процессор	4 виртуальных ядра	8 виртуальных ядер и более
Оперативная память	8 ГБ	16 ГБ и более
Дисковое пространство	100 ГБ SSD	500 ГБ SSD и более

Параметр	Минимально	Рекомендуется
Операционная система	GNU/Linux с ядром 5.x и новее (Debian 11+, Ubuntu 22.04+, Astra Linux)	—
Docker Engine	версия 24.0 и новее	—
Сеть	Постоянное подключение, публичный IP-адрес, открытые порты 80 и 443	—
Доменное имя	Делегированное доменное имя с записью A на IP-адрес сервера	—

Объём дискового пространства зависит от количества фотографий, загружаемых пользователями; при использовании внешнего объектного хранилища требования к диску сервера приложения снижаются.

4. Получение дистрибутива

Дистрибутив предоставляется правообладателем по письменному запросу и включает: файл описания стека docker-compose, файл параметров окружения с примерами значений, архив контейнерных образов либо реквизиты доступа к реестру контейнерных образов, настоящую инструкцию. Контактный адрес для запроса указан в конце документа.

5. Подготовка сервера

1. Установить Docker Engine и плагин docker compose в соответствии с документацией используемого дистрибутива операционной системы.
2. Инициализировать узел Docker Swarm и создать общую сеть для обратного прокси-сервера:

```
docker swarm init --advertise-addr <IP-адрес сервера>
docker network create --driver=overlay --attachable traefik-public
```

3. Создать каталог установки и подкаталоги для постоянного хранения данных:

```
mkdir -p /opt/mgor/data/{database,redis,keys,storage}
cd /opt/mgor
```

4. Разместить в каталоге установки файлы дистрибутива: docker-compose.yml и .env.
5. Загрузить контейнерные образы: при поставке архивом — командой docker load, при поставке через реестр — выполнить docker login и docker pull.

6. Параметры конфигурации

Параметры задаются в файле .env в каталоге установки. Значения, содержащие пароли и ключи доступа, задаются администратором при установке и не входят в состав дистрибутива.

Переменная	Назначение
HOSTNAME	Доменное имя, по которому публикуется экземпляр (например, mgor.example.ru)
STACKNAME	Имя стека Docker Swarm
COMPOSE_PROJECT_NAME	Имя проекта Compose
DOCKER_REGISTRY	Адрес реестра контейнерных образов, из которого загружаются образы
BASE_URL	Базовый URL экземпляра, используется при формировании ссылок
WS_URL	URL для подключения по протоколу WebSocket

Переменная	Назначение
DEFAULT_CENTER	Координаты центра карты по умолчанию в формате «широта,долгота»
DB_HOST, DB_PORT	Адрес и порт сервера СУБД PostgreSQL (по умолчанию database, 5432)
DB_NAME, DB_USERNAME, DB_PASSWORD	Имя базы данных и учётные данные подключения
REDIS_URL	Строка подключения к Redis (по умолчанию redis://redis:6379)
S3_ENDPOINT, S3_REGION, S3_BUCKET	Параметры S3-совместимого объектного хранилища для файлов и фотографий
S3_ACCESS_KEY_ID, S3_SECRET_ACCESS_KEY	Учётные данные доступа к объектному хранилищу
YANDEX_CLIENT_ID, YANDEX_CLIENT_SECRET	Параметры входа через Яндекс ID (необязательно)
VK_CLIENT_ID, VK_CLIENT_SECRET	Параметры входа через VK ID (необязательно)

7. Развёртывание

6. Запустить стек:

```
docker stack deploy -c docker-compose.yml --with-registry-auth $STACKNAME
```

7. Убедиться, что службы базы данных и кеша перешли в состояние running:

```
docker service ls
docker service ps ${STACKNAME}_database
```

8. Выполнить миграции базы данных. Служба миграций входит в состав стека и запускает команду:

```
yarn run migrate:dev
```

Миграции выполняются однократно при первом развёртывании и при каждом обновлении версии ПО. Служба завершается после успешного применения миграций; результат следует проверить в журнале: `docker service logs ${STACKNAME}_migration`.

9. Дождаться запуска сервера приложения и клиентского приложения, после чего проверить доступность экземпляра по адресу, указанному в параметре `BASE_URL`.

8. Настройка обратного прокси-сервера и TLS

Маршрутизация выполняется обратным прокси-сервером по доменному имени из параметра `HOSTNAME`. Запросы с префиксом `/api` направляются на сервер приложения (внутренний порт 3000), остальные — на клиентское приложение. В поставляемом описании стека маршруты заданы метками Docker для Traefik; при использовании иного прокси-сервера (nginx, Angie, Caddy) следует настроить эквивалентные правила.

Сертификат TLS выпускается автоматически при наличии делегированного доменного имени и доступности портов 80 и 443, либо устанавливается администратором вручную.

9. Первичная настройка

10. Создать учётную запись администратора. Учётная запись создаётся служебной командой сервера приложения либо предоставляется в составе демонстрационного набора данных — порядок описан в файле `README` дистрибутива.

11. Войти в систему под учётной записью администратора и заполнить территориальную структуру: округ, территориальные управления, районы.
12. Завести обслуживающие организации и пользователей, назначить роли.
13. Загрузить реестр объектов обслуживания (контейнерные площадки, дворовые территории, объекты благоустройства).
14. Настроить параметры уведомлений и, при необходимости, ключи для push-уведомлений.

10. Проверка работоспособности

- открыть адрес экземпляра в браузере — должна отобразиться страница входа;
- выполнить вход под учётной записью администратора;
- создать тестовую заявку и убедиться, что она отображается на доске;
- загрузить фотографию к заявке и убедиться, что файл сохраняется в объектном хранилище;
- проверить журналы служб на отсутствие ошибок: `docker service logs ${STACKNAME}_backend`.

11. Администрирование

11.1. Резервное копирование

Резервному копированию подлежат: база данных PostgreSQL, каталог ключей `data/keys`, содержимое объектного хранилища. Рекомендуемая периодичность — не реже двух раз в сутки в часы наименьшей нагрузки. Создание копии базы данных:

```
docker exec -i <контейнер БД> pg_dump -U $DB_USERNAME $DB_NAME | gzip > backup_$(date +%F_%H%M).sql.gz
```

Восстановление выполняется командой `psql` из распакованной копии при остановленном сервере приложения.

11.2. Обновление версии

Обновление выполняется заменой контейнерных образов и повторным развёртыванием стека. Перед обновлением создаётся резервная копия базы данных. Службы обновляются в режиме `start-first`: новый экземпляр запускается до остановки прежнего, что исключает перерыв в обслуживании. При неуспешном обновлении предусмотрен автоматический откат к предыдущей версии образа.

```
docker pull $DOCKER_REGISTRY/smartcity-backend:<версия>
docker stack deploy -c docker-compose.yml --with-registry-auth $STACKNAME
```

11.3. Журналирование и контроль

Журналы служб доступны средствами Docker (`docker service logs`). Состояние служб контролируется командой `docker service ls`. Рекомендуется настроить внешний мониторинг доступности экземпляра по протоколу HTTP и контроль свободного дискового пространства.

12. Типичные неисправности

Признак	Вероятная причина и действия
Служба сервера приложения циклически перезапускается	Недоступна база данных или неверные учётные данные. Проверить параметры DB_* и состояние службы базы данных
Страница открывается, вход не выполняется	Недоступен Redis. Проверить параметр REDIS_URL и состояние службы кеша
Фотографии не загружаются	Неверные параметры объектного хранилища либо отсутствие доступа к нему. Проверить параметры S3_*
Сертификат TLS не выпускается	Доменное имя не делегировано либо закрыты порты 80 и 443. Проверить запись А и правила межсетевого экрана
После обновления интерфейс отображается некорректно	В браузере сохранена прежняя версия приложения. Очистить кеш браузера и обновить страницу

13. Контактная информация

ООО «КОММИТ», 140105, Московская область, г. Раменское, Северное шоссе, стр. 4А. Запрос дистрибутива и техническая поддержка: vegaliry@mail.ru, телефон +7 (926) 536-77-42.

Генеральный директор ООО «КОММИТ»



М. Р. Аббясов