ПРОГРАММА ДЛЯ ЭВМ

ПЭВМ «Платформа экомониторинга UrbanECO».
Модуль «Контроль стройотходов (ОССиГ)»

Руководство по установке

Листов: 16

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ **2**
1. НАЗНАЧЕНИЕ ДОКУМЕНТА **3**
2. СОСТАВ АРХИВА ПОСТАВКИ **4**
2.1. Переменные .env 5
3. СИСТЕМНЫЕ ТРЕБОВАНИЯ **6**
3.1. Аппаратные ресурсы 6
3.2. Операционная система и bash 6
3.3. Программное обеспечение на хосте 6
3.4. Требования к квалификации 7
3.5. Сетевые порты 7
4. ВОССТАНОВЛЕНИЕ И ЗАПУСК НА ЦЕЛЕВОЙ ВМ **8**
4.1. Скачивание и распаковка 8
4.2. Проверка .env 8
4.3. Права на data/ 8
4.4. Загрузка образов 8
4.5. Файл hosts 9
4.5.1. На ВМ стенда (обязательно) 9
4.5.2. На удалённой рабочей станции (если UI открывают не на ВМ) 9
4.6. Запуск 10
4.7. Совместная установка на одной ВМ 10
5. ПРОВЕРКА РАБОТОСПОСОБНОСТИ **13**
5.1. Доступность сервисов 13
5.2. Учётные записи для работы с системой 13
5.2.1. UI модуля «Контроль стройотходов (ОССиГ)» (основная работа с системой) 13
5.2.2. Администрирование Keycloak 14
5.2.3. MinIO Console (объектное хранилище) 14
6. УСТРАНЕНИЕ НЕПОЛАДОК **15**
7. ОСТАНОВКА **16**

1. НАЗНАЧЕНИЕ ДОКУМЕНТА

Данное руководство предназначено для развёртывания архива поставки модуля «Контроль стройотходов (ОССиГ)» для ПЭВМ «Платформа экомониторинга UrbanECO» (каталог `/opt/urbaneco-ossig`) на целевой виртуальной машине (ВМ) под управлением Linux.

В архиве уже содержатся:

• файл .env уже заполнен рабочими значениями стенда, включая `change-me-*` у MinIO. Это не плейсхолдеры: с ними инициализирован поставляемый `data/minio`;

• данные PostgreSQL и MinIO;

• конфигурация nginx и приложений;

• tar-архивы Docker-образов в каталоге images/.

Первичная настройка стенда уже выполнена; результат сохранён в каталогах data/ и conf/.

Важно: все команды выполняются только в GNU Bash ≥ 4.2 (`/bin/bash`). Доступ к локальному стенду — только по HTTP через домены `*.urbaneco.urbantechgroup.ru`; записи в `/etc/hosts` перенаправляют их с демо-стенда (публичный DNS) на ваш экземпляр.

Стек и UI запускаются без ПЭВМ «Платформа экомониторинга UrbanECO» и без модуля «Электронный талон». Интеграция с Redmine (`MVS_MUSOR_REDMINE_ENDPOINT`) работает только при запущенном модуле «Электронный талон». Совместная установка на одной ВМ использует ещё один архив, `urbaneco-one-vm.zip`. Порядок — раздел 4.7.

2. СОСТАВ АРХИВА ПОСТАВКИ

Архив: urbaneco-ossig.zip

Путь	Назначение
docker-compose.yaml	Описание стека модуля
docker-compose.one-vm.yaml	Подключение к шлюзу при совместной установке (раздел 4.7)
.env	Параметры и секреты
.env.example	Образец .env
hosts.txt	Записи для /etc/hosts
images/	tar-архивы Docker-образов
data/postgres/ossig/	Данные PostgreSQL (PGDATA)
data/minio/	Объекты S3
conf/	Конфигурация сервисов
scripts/	load-images.sh, fix-data-permissions.sh, smoke-check.sh
scripts/build-local-images.sh	Служебный скрипт сборки образов, эксперту не нужен
Dockerfile.minio-init	Образ разовой инициализации MinIO
docs/	Руководства по установке, администрированию и доступу к демо
data/ossig/kafka/	Данные Kafka
data/ossig/splitter/	Логи splitter
test/	Файл test.xml для проверки splitter

Сервисы основного стека: minio, minio-init, nginx, ossig-postgres, ossig-keycloak, ossig-core-frontend, ossig-core-backend, ossig-core-integration, musor-backend, musor-backend-integration, kafka, schema-registry, kafka-connect-dtm, splitter, haproxy, hazelcast, truck-classificator, truck-classificator-app, truck-cargo-detector-worker, truck-cargo-detector-worker-app.

2.1. Переменные .env

Переменная	Назначение
OSSIG_POSTGRES_PASSWORD	Суперпользователь PostgreSQL
POSTGRES_OSSIG_PASSWORD	Пароль БД приложения
OSSIG_POSTGRES_KEYCLOAK_PASSWORD	Пароль БД Keycloak
OSSIG_KEYCLOAK_ADMIN_PASSWORD	Администратор Keycloak
MINIO_ROOT_PASSWORD	Root MinIO
MINIO_OSSIG_ACCESS_KEY	Access key приложения (S3)
MINIO_OSSIG_SECRET_KEY	Secret key приложения (S3)
MINIO_OSSIG_BUCKET	Bucket S3
MINIO_GRZ_BUCKET	Публичный bucket ГРЗ
MVS_MUSOR_REDMINE_ENDPOINT	URL API модуля «Электронный талон». Пустое значение не даёт сервису интеграции стартовать
MVS_MUSOR_INTEGRATION_REDMINE_ACCESSKEY	Ключ этого вызова, если endpoint задан
KAFKA_ADVERTISED_HOST	IP или DNS этой ВМ для внешних клиентов Kafka

`KAFKA_ADVERTISED_HOST=127.0.0.1` допустим только если клиенты Kafka на той же машине. После смены: `docker compose -p urbaneco-ossig up -d kafka`.

3. СИСТЕМНЫЕ ТРЕБОВАНИЯ

3.1. Аппаратные ресурсы

В стеке есть Kafka, Hazelcast и сервисы классификации ТС.

	CPU	RAM	Диск
Минимум	16 ядер	32 ГБ	SSD, 100 ГБ
Рекомендуется	32 ядра	64 ГБ	SSD, 200+ ГБ

Архитектура процессора: x86_64 / amd64.

3.2. Операционная система и bash

Развёртывание выполняется только на Linux (целевая ВМ):

ОС	Минимальная версия
Ubuntu	20.04 и новее
Debian	12 и новее
Alt Linux	серверные редакции с поддержкой Docker
Astra Linux	серверные редакции с поддержкой Docker

На хосте должна быть установлена оболочка bash (`/bin/bash`), версия не ниже GNU Bash 4.2. Все команды и скрипты из данного руководства выполняются только в bash; запуск через `/bin/sh`, dash или `sh script.sh` не поддерживается.

Проверка:

```
uname -m
bash --version | head -1
```

3.3. Программное обеспечение на хосте

Компонент	Требование
Docker Engine	≥ 28
Docker Compose	плагин `docker compose` ≥ 2.24.4: в `docker-compose.one-vm.yaml` используются теги `!reset` и `!override`
Группа пользователя	`docker` или аналогичная: команды `docker` и `docker compose` выполняются без `sudo`
bash	GNU Bash ≥ 4.2

Компонент	Требование
Утилиты	curl, unzip, sudo

3.4. Требования к квалификации

Развёртывание и запуск архива должны выполнять специалисты, обладающие навыками:

• администрирования Linux (bash, systemd, права доступа к файлам);

• работы с Docker и Docker Compose;

• базовой диагностики сетевых сервисов (curl, ss, просмотр логов контейнеров).

3.5. Сетевые порты

Порт	Назначение
80/tcp	nginx
15434/tcp	postgres
18081/tcp	keycloak
9092/tcp	kafka
8081/tcp	schema-registry
127.0.0.1:48083/tcp, 127.0.0.1:9203/tcp	kafka-connect
33004/tcp, 9999/tcp	haproxy (splitter)
127.0.0.1:5701/tcp	hazelcast
127.0.0.1:60080/tcp, 127.0.0.1:60081/tcp	классификаторы ТС

4. ВОССТАНОВЛЕНИЕ И ЗАПУСК НА ЦЕЛЕВОЙ ВМ

Каталог установки — `/opt/urbaneco-ossig`.

4.1. Скачивание и распаковка

Каталог `/opt` должен быть доступен текущему пользователю на запись. Если его нет или он принадлежит root, выполните команды от root либо так:

```
sudo mkdir -p /opt/urbaneco-ossig
sudo chown "$USER" /opt /opt/urbaneco-ossig
cd /opt
curl -fL -o urbaneco-ossig.zip https://s3.mvs.group/static-
files/urbaneco-ossig.zip
unzip -q -o urbaneco-ossig.zip
cd /opt/urbaneco-ossig
```

Содержимое архива должно оказаться в `/opt/urbaneco-ossig` без лишнего уровня вложенности.

4.2. Проверка .env

```
test -f .env && echo "OK: .env найден" || echo "ОШИБКА: .env
отсутствует"
```

В поставке `.env` уже заполнен. При готовом каталоге `data/` пароли Postgres, Keycloak и MinIO не меняйте: сервисы не запустятся. Менять их можно только при чистой установке без `data/`.

`KAFKA_ADVERTISED_HOST` по умолчанию `127.0.0.1`. Если клиенты Kafka работают на другой машине, укажите IP ВМ и выполните `docker compose -p urbaneco-ossig up -d kafka`.

При отсутствии `.env`: `cp .env.example .env` и задайте секреты. Значения `change-me-…` замените на пароли стенда до первого запуска.

4.3. Права на data/

```
sudo ./scripts/fix-data-permissions.sh
```

Не используйте `sudo docker compose`. Права на `data/` выставляет `sudo ./scripts/fix-data-permissions.sh`: PostgreSQL получает UID 999, каталоги MinIO и Kafka — UID 1000. Контейнер MinIO при этом работает от root (`user: "0:0"`). Логи splitter доступны на запись всем (`chmod a+rwX`). Если стенд уже запускается, владельца каталогов менять не нужно. Учётная запись, которая выполняет `docker`, должна входить в группу `docker` или аналогичную группу с доступом к сокету Docker Engine.

4.4. Загрузка образов

```
./scripts/load-images.sh
docker images | grep -E 'postgres-custom|keycloakx|core-
|minio|nginx|urbaneco-|musor|kafka|hazelcast|haproxy|truck|schema-
registry|splitter'
```

Запрещено `docker compose pull` / `docker pull`.

4.5. Файл hosts

Домены `*.urbaneco.urbantechgroup.ru` зарегистрированы в DNS и по умолчанию указывают на демонстрационный стенд. Демонстрационный стенд открывают с компьютера, на котором файл hosts не меняли. Чтобы обращаться к локально развёрнутому экземпляру, добавьте записи в файл hosts — они имеют приоритет над DNS и направят те же имена на нужный IP. Записи нужны на каждой машине, с которой выполняется проверка или открывается UI (браузер, curl, `./scripts/smoke-check.sh`).

4.5.1. На ВМ стенда (обязательно)

Выполняется при развёртывании на самой ВМ — для запуска compose, curl и `./scripts/smoke-check.sh` по SSH:

Повторный запуск `tee -a` дублирует строки. Добавляйте записи только если их ещё нет:

```
cd /opt/urbaneco-ossig
grep -q 'ossig.urbaneco.urbantechgroup.ru' /etc/hosts || sudo tee -
a /etc/hosts < hosts.txt
```

Файл hosts.txt перекрывает DNS и сопоставляет домены с 127.0.0.1, потому что nginx слушает порт 80 на этой же ВМ:

```
127.0.0.1 ossig.urbaneco.urbantechgroup.ru
127.0.0.1 ossig-backend.urbaneco.urbantechgroup.ru
127.0.0.1 ossig-integration.urbaneco.urbantechgroup.ru
127.0.0.1 ossig-uc-backend.urbaneco.urbantechgroup.ru
127.0.0.1 ossig-uc-integration.urbaneco.urbantechgroup.ru
127.0.0.1 ossig-auth.urbaneco.urbantechgroup.ru
127.0.0.1 ossig-s3.urbaneco.urbantechgroup.ru
127.0.0.1 ossig-s3-console.urbaneco.urbantechgroup.ru
```

4.5.2. На удалённой рабочей станции (если UI открывают не на ВМ)

Если браузер запускают с другого компьютера (ноутбук, рабочая станция в сети), добавьте те же домены в его hosts-файл, но укажите IP-адрес ВМ стенда вместо 127.0.0.1.

Пример: ВМ доступна по адресу 192.168.1.50:

```
192.168.1.50 ossig.urbaneco.urbantechgroup.ru
192.168.1.50 ossig-backend.urbaneco.urbantechgroup.ru
192.168.1.50 ossig-integration.urbaneco.urbantechgroup.ru
192.168.1.50 ossig-uc-backend.urbaneco.urbantechgroup.ru
192.168.1.50 ossig-uc-integration.urbaneco.urbantechgroup.ru
192.168.1.50 ossig-auth.urbaneco.urbantechgroup.ru
192.168.1.50 ossig-s3.urbaneco.urbantechgroup.ru
192.168.1.50 ossig-s3-console.urbaneco.urbantechgroup.ru
```

ОС	Файл	Как редактировать
Linux / macOS	/etc/hosts	sudo nano /etc/hosts или sudo tee -a /etc/hosts
Windows	C:\Windows\System32\drivers\etc\hosts	Блокнот от имени администратора

Важно:

• `./scripts/smoke-check.sh` проверяет hosts на той машине, где запущен скрипт. При проверке по SSH на ВМ достаточно hosts.txt с 127.0.0.1.

• Без записей в hosts браузер с любого ПК откроет демо-стенд (публичный DNS). Для доступа к вашему экземпляру добавьте записи на этом ПК с IP вашей ВМ.

• С удалённой станции 127.0.0.1 в hosts не подходит — это всегда «локальный» компьютер, а не ВМ со стендом.

Проверка (на той машине, с которой обращаются к UI):

```
getent hosts ossig.urbaneco.urbantechgroup.ru
# на ВМ — ожидается 127.0.0.1
# с удалённой станции — IP-адрес ВМ стенда
```

4.6. Запуск

```
docker compose -p urbaneco-ossig -f docker-compose.yaml config
docker compose -p urbaneco-ossig -f docker-compose.yaml up -d
```

Эти команды занимают порт 80 nginx-ом этого продукта. Их выполняют, только когда на ВМ один продукт. Если на той же ВМ будут платформа и модули, раздел 4.6 не выполняйте и переходите к разделу 4.7.

`KAFKA_ADVERTISED_HOST` по умолчанию `127.0.0.1`. Если клиенты Kafka работают на другой машине, укажите IP ВМ и выполните `docker compose -p urbaneco-ossig up -d kafka`.

4.7. Совместная установка на одной ВМ

Раздел 4.6 поднимает nginx этого продукта на порту 80. Так на одной ВМ работает только один продукт. Чтобы вместе работали платформа и модули в любом составе и в любом порядке, нужен отдельный архив шлюза `urbaneco-one-vm.zip` — https://s3.mvs.group/static-files/urbaneco-one-vm.zip. Его кладут рядом с архивами продуктов. Внутри архива каталог `one-vm/`. В архивах продуктов этой папки нет.

Шлюз занимает порт 80 и по имени хоста передаёт запрос nginx нужного продукта. Nginx продуктов в этом режиме порт 80 на хосте не публикует.

Сначала подготовка каждого нужного продукта, затем запуск через скрипт шлюза. `docker compose up` из раздела 4.6 при совместной установке не выполняют.

Ресурсы ВМ складываются из минимумов выбранных продуктов (раздел 3.1 руководства каждого продукта): платформа — 4 ядра, 8 ГБ RAM, 40 ГБ диска; электронный талон — 8 ядер, 16 ГБ RAM, 60 ГБ диска; ОССиГ — 16 ядер, 32 ГБ RAM, 100 ГБ диска. Для всех трёх продуктов нужно 28 ядер, 56 ГБ RAM и 200 ГБ диска. Требования к Docker Engine, Docker Compose и bash — раздел 3.3.

1. Распакуйте в `/opt` архив шлюза и архивы тех продуктов, которые будете запускать. Архив продукта, который не ставите, не скачивайте и не распаковывайте. Архив шлюза нужен всегда, когда продуктов больше одного. Каталог `/opt` должен быть доступен текущему пользователю на запись; если он принадлежит root, выполните `sudo chown "$USER" /opt`. Команды ниже — пример для всех четырёх архивов; оставьте только выбранные продукты и `urbaneco-one-vm.zip`.

```
cd /opt
```

```
curl -fL -o urbaneco-one-vm.zip https://s3.mvs.group/static-
files/urbaneco-one-vm.zip
curl -fL -o urbaneco-platform.zip https://s3.mvs.group/static-
files/urbaneco-platform.zip
curl -fL -o urbaneco-ossig.zip https://s3.mvs.group/static-
files/urbaneco-ossig.zip
curl -fL -o urbaneco-eticket.zip https://s3.mvs.group/static-
files/urbaneco-eticket.zip
unzip -q -o urbaneco-one-vm.zip
unzip -q -o urbaneco-platform.zip
unzip -q -o urbaneco-ossig.zip
unzip -q -o urbaneco-eticket.zip
```

После распаковки каталоги лежат рядом: `/opt/one-vm`, `/opt/urbaneco`, `/opt/urbaneco-ossig`, `/opt/urbaneco-eticket`.

2. В каталоге каждого распакованного продукта выполните разделы 4.2–4.4 его руководства: проверка `.env`, `sudo ./scripts/fix-data-permissions.sh`, `./scripts/load-images.sh`. Для ОССиГ `KAFKA_ADVERTISED_HOST` по умолчанию `127.0.0.1`; IP ВМ указывают, только если клиенты Kafka работают на другой машине. Раздел 4.5 (hosts продукта) пропустите: его заменяет общий список шлюза. Раздел 4.6 пропустите.

3. На ВМ добавьте в `/etc/hosts` общий список шлюза. Он покрывает домены всех трёх продуктов; лишние имена не мешают, пока продукт не запущен. Команда добавляет только отсутствующие строки, поэтому её можно повторять, в том числе после отдельной установки продукта по разделу 4.5:

```
grep -vxF -f /etc/hosts /opt/one-vm/hosts.txt | sudo tee -a
/etc/hosts
```

На рабочей станции, с которой открывают UI, добавьте те же имена, но с IP-адресом ВМ вместо 127.0.0.1 (файл hosts и способ редактирования — раздел 4.5.2). Список для станции выводит команда на ВМ; 192.168.1.50 замените на IP ВМ:

```
sed 's/^127\.0\.0\.1/192.168.1.50/' /opt/one-vm/hosts.txt
```

4. Запустите нужные продукты из `/opt`. Имена `platform`, `ossig` и `eticket` — аргументы скрипта. Каталоги на диске при этом называются `urbaneco`, `urbaneco-ossig` и `urbaneco-eticket`.

```
cd /opt
./one-vm/up.sh platform
./one-vm/up.sh ossig
./one-vm/up.sh eticket
```

Повторный вызов не останавливает уже запущенные продукты, а добавляет названные. Несколько имён можно передать сразу: `./one-vm/up.sh platform ossig`. Без имён `./one-vm/up.sh` запускает все продукты, распакованные рядом с `one-vm`.

Остановка одного продукта: `./one-vm/down.sh ossig`. Остальные и шлюз продолжают работать. `./one-vm/down.sh` без имён останавливает все распакованные продукты и снимает шлюз.

Если продукт уже был запущен по разделу 4.6 и его nginx занял порт 80, `./one-vm/up.sh` снимет эту публикацию и включит шлюз. Пока шлюз работает, не запускайте тот же продукт командой `docker compose up` без файла `docker-compose.one-vm.yaml`: обе команды претендуют на порт 80. Подробности — в `/opt/one-vm/README.md`.

Адреса продуктов при совместной установке:

Продукт	UI	S3
Платформа	http://base.urbaneco.urbantechgroup.ru	http://platform-s3.urbaneco.urbantechgroup.ru
ОССиГ	http://ossig.urbaneco.urbantechgroup.ru	http://ossig-s3.urbaneco.urbantechgroup.ru
Электронный талон	http://eticket.urbaneco.urbantechgroup.ru	http://eticket-s3.urbaneco.urbantechgroup.ru

Порт 80 занимает шлюз. Остальные порты хоста остаются у продуктов и не пересекаются:

Продукт	PostgreSQL	Keycloak	Прочее
Платформа	15432	18080	—
Электронный талон	15433	28080	—
ОССиГ	15434	18081	Kafka 9092, Schema Registry 8081, kafka-connect 127.0.0.1:48083 и 127.0.0.1:9203, haproxy 33004 и 9999, hazelcast 127.0.0.1:5701, классификаторы ТС 127.0.0.1:60080 и 127.0.0.1:60081

5. ПРОВЕРКА РАБОТОСПОСОБНОСТИ

5.1. Доступность сервисов

Сервис	URL
UI	http://ossig.urbaneco.urbantechgroup.ru
Auth	http://ossig-auth.urbaneco.urbantechgroup.ru/auth
musor-backend	http://ossig-backend.urbaneco.urbantechgroup.ru
core-backend	http://ossig-uc-backend.urbaneco.urbantechgroup.ru
musor-backend-integration	http://ossig-integration.urbaneco.urbantechgroup.ru
core-integration	http://ossig-uc-integration.urbaneco.urbantechgroup.ru
MinIO S3	http://ossig-s3.urbaneco.urbantechgroup.ru
MinIO Console	http://ossig-s3-console.urbaneco.urbantechgroup.ru

Команды ниже выполняются на ВМ стенда (после настройки hosts по п. 4.5):

```
curl -s -o /dev/null -w '%{http_code}\n'
http://ossig.urbaneco.urbantechgroup.ru/
./scripts/smoke-check.sh --wait
```

Если UI проверяют из браузера на другом компьютере — сначала добавьте записи hosts с IP вашей ВМ (п. 4.5) на этом компьютере (иначе откроется демо-стенд из DNS), затем откройте URL из таблицы.

5.2. Учётные записи для работы с системой

Демонстрационные учётные записи включены в архив (данные Keycloak и .env). Если в сопроводительном письме к поставке указаны иные учётные данные UI, использовать их.

5.2.1. UI модуля «Контроль стройотходов (ОССиГ)» (основная работа с системой)

URL	http://ossig.urbaneco.urbantechgroup.ru
Логин	ossig
Пароль	---

Аутентификация выполняется через Keycloak (realm ossig, client web-client).

5.2.2. Администрирование Keycloak

URL	**http://ossig-auth.urbaneco.urbantechgroup.ru/auth/admin**
Логин	admin
Пароль	значение OSSIG_KEYCLOAK_ADMIN_PASSWORD из файла .env

5.2.3. MinIO Console (объектное хранилище)

URL	**http://ossig-s3-console.urbaneco.urbantechgroup.ru**
Логин	minio
Пароль	значение MINIO_ROOT_PASSWORD из файла .env

Доступ приложения к S3 (API): access key MINIO_OSSIG_ACCESS_KEY, secret key MINIO_OSSIG_SECRET_KEY, bucket MINIO_OSSIG_BUCKET — см. .env. Публичный bucket ГРЗ — MINIO_GRZ_BUCKET.

6. УСТРАНЕНИЕ НЕПОЛАДОК

Симптом	Действие
Permission denied на data	sudo ./scripts/fix-data-permissions.sh, перезапуск compose
UI не открывается	/etc/hosts на той машине, где браузер (п. 4.5: на ВМ — 127.0.0.1, с удалённого ПК — IP ВМ); порт 80, docker compose -p urbaneco-ossig ps, логи nginx
Образ не найден	./scripts/load-images.sh; не docker pull
Keycloak / 502	Подождать 3–5 мин, логи ossig-keycloak, ossig-postgres
Порт занят	sudo ss -tulpn. Если порт 80 занят шлюзом совместной установки или nginx другого продукта — добавляйте продукт через ./one-vm/up.sh
MinIO Console 503	Логин на ossig-s3-console; user minio; docker compose -p urbaneco-ossig up -d --force-recreate minio
Клиенты Kafka видят 127.0.0.1	KAFKA_ADVERTISED_HOST равен IP ВМ, затем docker compose -p urbaneco-ossig up -d kafka
Вызовы в модуль «Электронный талон» не проходят	Электронный талон должен быть запущен, а имя из MVS_MUSOR_REDMINE_ENDPOINT — резолвиться на его nginx. В compose пустой URL заменяется адресом по умолчанию

7. ОСТАНОВКА

Отдельная установка (раздел 4.6):

```
cd /opt/urbaneco-ossig
docker compose -p urbaneco-ossig -f docker-compose.yaml down
```

Совместная установка (раздел 4.7): из `/opt` выполните `./one-vm/down.sh ossig`. `docker compose down` в этом режиме не используют.

Каталог data/ при остановке сохраняется.

Повторный запуск отдельной установки:

```
docker compose -p urbaneco-ossig -f docker-compose.yaml up -d
```

Повторный запуск совместной установки: `cd /opt && ./one-vm/up.sh ossig`.