

1. Аннотация

Настоящее руководство содержит информацию, необходимую для штатной эксплуатации, администрирования, мониторинга и обслуживания рабочего экземпляра программного модуля "Flexberry Portal". Документ предназначен для системных администраторов и DevOps-специалистов, ответственных за поддержание работоспособности продукта.

Требования к установке и первоначальному развертыванию описаны в отдельном документе «**Инструкция по установке Flexberry Portal**».

2. Требования к инфраструктуре и персоналу

2.1. Программные требования (для эксплуатации)

- **Docker Engine:** версия <20.10> или выше.
- **Docker Compose:** версия <v2.0> или выше.
- **PostgreSQL:** версия <12> или выше (при использовании внешней СУБД).
- Операционная система, поддерживающая Docker (любой дистрибутив Linux, Windows Server с WSL2, etc.).

2.2. Аппаратные требования (рекомендуемые)

- **CPU:** 2 ядра или более.
- **ОЗУ:** 4 ГБ или более (включая ресурсы для PostgreSQL).
- **Дисковое пространство:** 10 ГБ для образов, данных и логов.

2.3. Требования к квалификации персонала

Администратор, выполняющий операции, описанные в данном руководстве, должен обладать следующими знаниями:

- Базовое администрирование операционных систем (Linux/Windows).
- Работа с Docker и Docker Compose (запуск, остановка, просмотр логов, управление образами и томами).
- Навыки выполнения резервного копирования и восстановления СУБД PostgreSQL.
- Умение редактировать конфигурационные файлы (YAML, переменные окружения).

3. Конфигурация и параметры настройки

Основная конфигурация модуля осуществляется через **переменные окружения**, заданные в файле .env и/или в docker-compose.yml.

3.1. Критически важные параметры (файл .env)

Файл .env должен находиться в одной директории с docker-compose.yml. Все параметры имеют комментарии и очевидное предназначение. Во время эксплуатации нет необходимости изменять эти значения.

4. Администрирование и повседневная эксплуатация

4.1. Управление сервисами

Все команды выполняются в директории с файлом docker-compose.yml.

Действие	Команда	Описание
Запуск	docker-compose up -d	Запуск всех сервисов в фоновом режиме.
Проверка статуса	docker-compose ps	Вывод статуса всех контейнеров проекта.
Просмотр логов приложения	docker-compose logs -f portal-app	Потоковый вывод логов контейнера portal-app. Ключ -f (follow) – следовать за логом.
Просмотр логов БД	docker-compose logs postgres	Вывод логов контейнера СУБД PostgreSQL.
Остановка	docker-compose stop	Остановка всех сервисов без удаления контейнеров.
Запуск после остановки	docker-compose start	Запуск ранее остановленных сервисов.
Перезапуск	docker-compose restart portal-app	Перезапуск конкретного сервиса (например, после смены конфигурации).
Полная остановка с удалением	docker-compose down	Остановка и удаление контейнеров, сетей. Тома и образы не удаляются.

4.2. Обновление версии

1. **Резервное копирование данных** (см. раздел 4.3).
2. **Остановка текущей версии:** docker-compose down
3. **Получение нового образа:** docker-compose pull
4. **Запуск новой версии:** docker-compose up -d
5. **Удаление старых образов (опционально):** docker image prune -a

4.3. Резервное копирование и восстановление базы данных

Создание резервной копии (dump):

bash

```
docker exec -t <container_name_postgres> pg_dump -U <POSTGRES_USER> <POSTGRES_DB> > portal_backup_$(date +%Y-%m-%d_%H-%M).sql
```

Пример:

```
docker exec -t portal_postgres_1 pg_dump -U portal_user flexberry_portal > portal_backup_2023-11-15.sql
```

Восстановление из резервной копии:

1. Остановите приложение: docker-compose stop portal-app
2. Восстановите данные:

bash

```
cat portal_backup_2023-11-15.sql | docker exec -i <container_name_postgres> psql -U <POSTGRES_USER> -d <POSTGRES_DB>
```

3. Запустите приложение: docker-compose start portal-app

Резервное копирование тома с данными PostgreSQL (альтернативный способ):

bash

```
docker run --rm -v <volume_name>:/source -v $(pwd):/backup alpine tar czf /backup/postgres_data_$(date +%Y%m%d).tar.gz -C /source .
```

Восстановление через монтирование тома.

4.4. Мониторинг работоспособности

- **Проверка через healthcheck:** Статус контейнеров можно проверить командой docker-compose ps. Контейнер с проваленным healthcheck будет иметь статус unhealthy.
- **Проверка через API эндпоинт:** Приложение предоставляет эндпоинт для проверки здоровья:

bash

```
curl -f http://<API_URL>/api/health || echo "Сервис недоступен"
```

Мониторинг ресурсов:

bash

```
docker stats          # Статистика использования CPU, памяти, сети в реальном времени
docker-compose top    # Процессы внутри контейнеров
```

Анализ логов: Регулярно проверяйте логи на наличие ошибок (уровни ERROR, WARNING). Для анализа используйте grep:

bash

```
docker-compose logs portal-app | grep -E "ERROR|WARNING|Exception"
```

5. Диагностика и устранение неисправностей

Симптом / Ошибка	Возможная причина	Действия по устранению
Контейнер portal-app постоянно перезапускается или имеет статус unhealthy.	1. Нет подключения к БД. 2. Неверные учетные данные в .env. 3. Ошибка в конфигурации приложения.	1. Проверить логи: docker-compose logs portal-app. 2. Убедиться, что контейнер postgres работает (docker-compose ps). 3. Проверить корректность значений POSTGRES_* в .env.
Приложение запущено, но API не отвечает (Connection refused / timeout).	1. Порт на хосте занят другим процессом. 2. Фаервол блокирует порт. 3. Приложение упало внутри контейнера.	1. Проверить занятость порта: netstat -tuln grep :<хост-порт>. 2. Изменить внешний порт в docker-compose.yml. 3. Проверить логи приложения.
Ошибки в логах: "Connection to database failed"	1. Проблемы с сетью между контейнерами. 2. СУБД не запущена. 3. Неправильное имя хоста/порта.	1. Проверить, что контейнеры в одной сети (docker network ls). 2. Перезапустить сервис БД: docker-compose restart postgres. 3. Проверить переменные POSTGRES_HOST и POSTGRES_PORT.
Высокая загрузка CPU/памяти.	1. Утечка памяти в приложении. 2. Неоптимальные	1. Использовать docker stats для идентификации проблемного контейнера. 2. Проверить логи БД на наличие

Симптом / Ошибка	Возможная причина	Действия по устранению
Дисковое пространство заканчивается.	запросы к БД. 3. Недостаточно ресурсов.	медленных запросов. 3. Рассмотреть увеличение ресурсов в docker-compose.yml (раздел deploy.resources.limits).
	1. Рост логов. 2. Рост данных БД. 3. Накопление неиспользуемых Docker-объектов.	1. Настроить ротацию логов в Docker (docker-compose.yml с драйвером json-file и max-size). 2. Очистить старые резервные копии. 3. Выполнить очистку: docker system prune -a --volumes (с осторожностью!).

6. Контакты и поддержка

- **Официальная документация и ресурсы:** Актуальная информация о продукте размещена на странице: <https://flexberry.net/ru/developers-flexberry-portal.html>.
- **Коммерческая и техническая поддержка:** Для получения консультаций по эксплуатации и решения инцидентов обращайтесь по электронной почте: support@flexberry.ru.

Важное примечание: Перед выполнением любых операций по обновлению или изменению конфигурации на **продуктивном (production) окружении** обязательно выполняйте полное резервное копирование данных и тестируйте процедуры на стенде.