ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ

«Смарт-Ком»

«Panelyx (Панеликс) – Платформа сенсорных и потребительских исследований»

Инструкция по установке

Версия 1.0

2026

Содержание

1. Введение 3
2. Общие сведения 3
2.1. Назначение Программы 3
2.2. Состав Программы 3
2.3. Постоянные данные 4
3. Требования к аппаратным и программным средствам 5
3.1. Аппаратные требования 5
3.2. Программные требования 5
3.3. Сетевые требования 6
4. Комплект поставки 6
5. Подготовка сервера 6
5.1. Установка Docker Engine 6
5.2. Подготовка каталога установки 7
5.3. Настройка межсетевого экрана 7
6. Получение образов контейнеров 7
6.1. Получение образов из реестра контейнеров 7
6.2. Загрузка образов из tar-архива 7
7. Конфигурирование Программы 7
7.1. Файл параметров 7
7.2. Пароли базы данных 8
7.3. TLS-сертификат 8
7.4. Почтовый сервер 8
7.5. Хранилище секретов 9
8. Запуск Программы 9
9. Проверка работоспособности 9
9.1. Проверка компонентов 9
9.2. Контрольная проверка 10
10. Резервное копирование и восстановление 10
10.1. Состав резервной копии 10
10.2. Создание резервной копии 10
10.3. Восстановление 11
11. Обновление и откат версии 11
11.1. Обновление 11
11.2. Откат версии 11
12. Диагностика типовых проблем 11

1. Введение

Настоящий документ содержит порядок установки программного обеспечения «Panelyx (Панеликс) – Платформа сенсорных и потребительских исследований» (далее — «Программа»).

Программа поставляется в виде комплекта контейнеров и разворачивается на сервере средствами Docker Engine и Docker Compose. Порядок установки одинаков для двух вариантов развёртывания:

— SaaS — установка выполняется персоналом правообладателя на вычислительных мощностях правообладателя, организации-заказчики работают в отдельных рабочих пространствах.

— On-Prem — установка выполняется на выделенном сервере в контуре организации-заказчика.

В разделе «Общие сведения» приведены сведения о составе Программы и её постоянных данных. В разделе «Требования к аппаратным и программным средствам» указаны требования к серверу, системному программному обеспечению и сети. Разделы с 4 по 8 описывают подготовку сервера, получение образов, конфигурирование и запуск Программы. В разделах «Проверка работоспособности», «Резервное копирование и восстановление» и «Обновление и откат версии» описаны действия после установки. Раздел «Диагностика типовых проблем» содержит перечень типовых неисправностей и способы их устранения.

ПРИМЕЧАНИЕ. Конкретные значения доменных имён, адресов, паролей, токенов и ключей в настоящем документе заменены обозначениями в угловых скобках, например <домен Программы>. Фактические значения задаются администратором при установке.

2. Общие сведения

2.1. Назначение Программы

Программа предназначена для автоматизации полного цикла сенсорных исследований: проектирования исследования и анкеты, слепой подачи образцов, набора панели респондентов, проведения исследования, статистической обработки результатов и формирования протокола исследования.

2.2. Состав Программы

Программа состоит из набора контейнерных сервисов, запускаемых в общей внутренней сети Docker. Имена сервисов соответствуют именам в файле compose.yaml, порты указаны внутри сети контейнеров.

Сервис	Технологии	Порт	Назначение
web	Nginx	3000	Единый адрес доступа: приём запросов браузера, терминация TLS, маршрутизация к интерфейсу и API
frontend	Vue 3, TypeScript, Nginx	4173	Пользовательский интерфейс Программы

Сервис	Технологии	Порт	Назначение
gateway	C#, .NET	8080	HTTP API, проверка авторизации и контекста организации, маршрутизация к сервисам
identity	C#, .NET	5001	Учётные записи, вход, токены, приглашения и права доступа
survey	C#, .NET	5002	Исследования, анкеты, шаблоны и справочники
response	C#, .NET	5003	Прохождение анкет и приём ответов
report	C#, .NET	5004	Организация расчётов и формирования отчётов, хранение результатов
report-py	Python, FastAPI	5010	Аналитические расчёты, выгрузки XLSX, содержимое PDF-отчётов
report-renderer	Node.js, Chromium	4180	Преобразование HTML-представления отчётов в PDF
notification	C#, .NET	5005	Уведомления и отправка электронной почты
settings	C#, .NET	5007	Настройки организаций, параметры интеграций и тарифов
postgres	PostgreSQL 16, pgvector	5432	База данных Программы, схемы сервисов
vault	OpenBao	8200	Хранилище секретов интеграций
jaeger	Jaeger, OpenTelemetry	16686, 4317	Сбор и просмотр трасс запросов

Маршрутизация запросов: веб-прокси web направляет запросы к адресу «/» в сервис frontend, запросы к адресам «/api/v1/» и «/public-api/v1/» в шлюз gateway. Шлюз направляет запросы в прикладные сервисы. Сервисы report и gateway обращаются к report-py, который передаёт HTML-представление отчёта в report-renderer. Прикладные сервисы работают с базой данных postgres, сервис settings получает секреты из vault, трассы запросов передаются в jaeger. Во внешнюю сеть публикуется только веб-прокси web.

2.3. Постоянные данные

Постоянные данные Программы хранятся в именованных томах Docker, приведённых в таблице. Тома сохраняются при остановке и пересоздании контейнеров.

Том	Содержимое	Особенности
pg_data	База данных Программы	Основной источник прикладных данных
exports_data	Сформированные файлы отчётов и выгрузок	Сохраняется согласованно с базой данных
identity-invite-keys	Ключи приглашений пользователей	Сохраняется вместе с правами и владельцами файлов
public-link-keys	Ключи публичных ссылок на анкеты	Общий том сервисов gateway и notification

Хранилище секретов vault в промышленной установке должно использовать постоянное хранилище данных. Порядок его подготовки определяется скриптами подготовки OpenBao из комплекта поставки.

3. Требования к аппаратным и программным средствам

3.1. Аппаратные требования

Ориентировочные минимальные требования к серверу приведены в таблице.

Ресурс	Значение	Назначение
Процессор	8 vCPU	Базовая установка Программы
Оперативная память	16 ГБ	С запасом для статистической аналитики и формирования отчётов
Дисковое пространство	от 100 ГБ, SSD	Образы контейнеров, база данных, тома отчётов, резервные копии

Точная конфигурация определяется числом одновременных участников исследований, объёмом данных и частотой формирования отчётов.

3.2. Программные требования

Для установки Программы на сервере требуется следующее программное обеспечение:

— Операционная система семейства Linux для архитектуры x86_64, например Astra Linux, РЕД ОС или Ubuntu 24.04 LTS.

— Docker Engine с поддержкой Docker Compose v2 (команда docker compose).

— Утилиты curl, openssl и tar.

Среды выполнения .NET, Python и Node.js на сервер не устанавливаются: они входят в состав образов контейнеров.

3.3. Сетевые требования

Для работы Программы необходимо выполнить следующие условия:

— DNS-имя Программы и TLS-сертификат с закрытым ключом для этого имени. Для варианта SaaS дополнительно используется базовый домен, на котором создаются адреса рабочих пространств организаций.

— Входящие соединения по протоколу HTTPS (порт 443) к веб-прокси web. Остальные порты сервисов во внешнюю сеть не публикуются.

— Исходящие соединения к реестру контейнеров правообладателя (при получении образов по сети), к почтовому серверу (SMTP) и к корпоративному каталогу (AD FS), если эти интеграции используются.

4. Комплект поставки

В комплект поставки Программы входят:

— Версионированные образы контейнеров в реестре контейнеров GitLab правообладателя либо tar-архив образов для изолированного контура.

— Файл compose.yaml с описанием сервисов, сетей и томов.

— Пример файла параметров .env.example.

— Конфигурация веб-прокси Nginx.

— Скрипты подготовки PostgreSQL и OpenBao.

Для получения образов из реестра правообладатель выдаёт адрес реестра и токен развёртывания с правом только чтения (read_registry).

5. Подготовка сервера

5.1. Установка Docker Engine

Установите Docker Engine и плагин Docker Compose в соответствии с документацией используемой операционной системы. Пример установки для Ubuntu:

```
sudo apt update
sudo apt install -y ca-certificates curl openssl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
  -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
. /etc/os-release
echo "deb [signed-by=/etc/apt/keyrings/docker.asc] \
  https://download.docker.com/linux/ubuntu $VERSION_CODENAME stable" \
  | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io \
  docker-buildx-plugin docker-compose-plugin
sudo systemctl enable --now docker
```

Для Astra Linux и РЕД ОС используйте пакеты Docker из репозиториев дистрибутива. После установки проверьте версии компонентов:

```
docker version
docker compose version
docker info > /dev/null
```

Пользователь, от имени которого выполняется установка, должен иметь доступ к Docker. Не открывайте сокет Docker на внешнем сетевом интерфейсе.

5.2. Подготовка каталога установки

Создайте каталог установки и распакуйте в него комплект поставки:

```
sudo mkdir -p /opt/panelyx
cd /opt/panelyx
sudo tar -xf panelyx-release.tar
```

Далее все команды выполняются из каталога /opt/panelyx. Для сокращения записи определите псевдоним dc:

```
alias dc='docker compose --env-file .env -f compose.yaml'
```

5.3. Настройка межсетевого экрана

Разрешите входящие соединения только на порт 443 (и при необходимости на порт 22 для администрирования). Порты внутренних сервисов, базы данных, OpenBao и Jaeger во внешнюю сеть не публикуются.

6. Получение образов контейнеров

6.1. Получение образов из реестра контейнеров

Основной способ — получение образов из реестра контейнеров GitLab правообладателя по токену развёртывания:

```
docker login <адрес реестра>
dc pull
```

При запросе имени пользователя и пароля укажите имя и значение токена развёртывания, выданного правообладателем.

6.2. Загрузка образов из tar-архива

Для контура без доступа к сети загрузите образы из tar-архива комплекта поставки:

```
docker load --input panelyx-images.tar
```

После загрузки проверьте наличие образов командой docker images.

7. Конфигурирование Программы

7.1. Файл параметров

Создайте файл параметров из примера и ограничьте доступ к нему:

```
umask 077
cp .env.example .env
chmod 600 .env
```

Файл .env содержит секреты. Значения с пробелами и специальными символами указываются в кавычках. Для генерации новых секретов используйте команду openssl rand -hex 32. Основные группы параметров приведены в таблице.

Группа параметров	Содержание
Публичный адрес	DNS-имя Программы, публичный адрес интерфейса и анкет, допустимые источники запросов браузера (CORS). Для варианта SaaS — базовый домен рабочих пространств организаций.

Группа параметров	Содержание
PostgreSQL	Имя базы данных, пароль владельца базы данных (POSTGRES_PASSWORD), строки подключения сервисов.
Токены	Ключ подписи токенов доступа (JWT_SIGNING_KEY), одинаковый для сервисов identity и gateway. Внутренний токен сервиса формирования PDF (REPORT_RENDERER_INTERNAL_TOKEN), одинаковый для сервисов report-renderer, report, settings и report-py.
OpenBao	Внутренний адрес хранилища, способ аутентификации сервиса settings и пути секретов.
Реестр контейнеров	Адрес реестра контейнеров и токен развёртывания с правом read_registry.
Корпоративный вход	Адрес AD FS, идентификатор приложения и параметры токенов. Заполняется, если используется вход через корпоративный каталог.
Почта	Параметры почтового сервера (SMTP). Заполняются, если используются почтовые приглашения и уведомления.
TLS	Пути к сертификату и закрытому ключу для веб-прокси.

Проверьте итоговую конфигурацию без запуска контейнеров:

```
dc config > /dev/null
```

7.2. Пароли базы данных

Скрипт подготовки PostgreSQL создаёт владельца базы данных и прикладную роль сервисов. Задайте пароли до первого запуска. Изменение параметра POSTGRES_PASSWORD в файле .env после создания тома pg_data не меняет пароли в существующей базе данных: при смене паролей измените их в PostgreSQL и согласованно обновите строки подключения сервисов.

7.3. TLS-сертификат

Разместите сертификат и закрытый ключ для DNS-имени Программы в каталоге, указанном в конфигурации веб-прокси, и ограничьте доступ к закрытому ключу:

```
sudo mkdir -p /opt/panelyx/certs
sudo cp <сертификат>.crt /opt/panelyx/certs/panelyx.crt
sudo cp <ключ>.key /opt/panelyx/certs/panelyx.key
sudo chmod 600 /opt/panelyx/certs/panelyx.key
```

7.4. Почтовый сервер

Для отправки приглашений, уведомлений и отчётов укажите параметры почтового сервера в файле .env. Пример:

```
SMTP_PROVIDER=smtp
SMTP_HOST=<адрес почтового сервера>
SMTP_PORT=465
SMTP_USE_SSL=true
SMTP_USERNAME='<имя пользователя>'
SMTP_PASSWORD='<пароль>'
SMTP_FROM=<адрес отправителя>
SMTP_FROM_NAME=Panelyx
```

При SMTP_USE_SSL=true соединение с сервером устанавливается с шифрованием сразу при подключении (порт 465). При значении false используется STARTTLS, если сервер его поддерживает. Параметры почты организации могут быть переопределены в настройках рабочего пространства.

7.5. Хранилище секретов

Выполните скрипт подготовки OpenBao из комплекта поставки. Скрипт подготавливает хранилище секретов, пути секретов и доступ сервиса settings. Интерфейс управления OpenBao во внешнюю сеть не публикуется. Значения, необходимые для повторной инициализации хранилища, сохраните в защищённом месте.

8. Запуск Программы

Запустите Программу и проверьте состояние контейнеров:

```
dc up -d
dc ps
```

При первом запуске скрипты подготовки PostgreSQL создают базу данных и роли. Прикладные сервисы при запуске применяют миграции базы данных (DbUp) и создают свои схемы. Повторно скрипты подготовки на существующем томе pg_data не выполняются.

Ход запуска контролируется по журналам сервисов:

```
dc logs --tail=200 identity settings gateway
```

Если миграция завершилась ошибкой, контейнер сервиса перезапускается. Изучите первое сообщение об ошибке SQL в журнале и устраните причину (права, пароль, отсутствующий объект, свободное место). Не удаляйте журнал миграций и том pg_data для обхода ошибки.

После запуска создайте рабочее пространство организации и учётную запись её администратора. Администратор организации входит в Программу по публичному адресу и создаёт учётные записи остальных пользователей в разделе «Пользователи».

9. Проверка работоспособности

9.1. Проверка компонентов

Выполните проверку доступности компонентов:

```
curl -fsS https://<домен Программы>/healthz
dc exec -T postgres sh -c \
  'pg_isready -U "$POSTGRES_USER" -d "$POSTGRES_DB"'
dc exec -T report-py python -c '
import urllib.request
r = urllib.request.urlopen("http://127.0.0.1:5010/healthz")
print(r.read().decode())'
dc exec -T report-renderer node -e \
  'fetch("http://127.0.0.1:4180/healthz")
  .then(r => { if (!r.ok) process.exit(1) })'
```

Точка /healthz веб-прокси подтверждает работу веб-прокси, но не проверяет базу данных и API. Состояние контейнера «Up» также не подтверждает готовность приложения, поэтому выполните контрольную проверку по разделу 9.2.

9.2. Контрольная проверка

Контрольная проверка выполняется в следующей последовательности:

1. Откройте публичный адрес Программы в браузере и выполните вход администратором организации. Убедитесь, что открывается дашборд, доступны каталог продуктов и список исследований.

2. Создайте тестовое исследование, настройте анкету, запустите исследование и получите ссылку на анкету.

3. Откройте ссылку в отдельном окне браузера и отправьте ответ участника.

4. Проверьте, что ответ сохранён и аналитика рассчитана. Выгрузите отчёты XLSX и PDF и откройте их.

5. Выполните вход пользователем с ограниченной ролью и убедитесь, что недоступные операции запрещены.

6. Если настроена почта, отправьте приглашение участнику исследования по адресу электронной почты и проверьте фактическую доставку письма.

10. Резервное копирование и восстановление

10.1. Состав резервной копии

Резервная копия Программы включает:

— Дамп базы данных (pg_dump в формате custom) и описание ролей PostgreSQL.

— Архивы томов exports_data, identity-invite-keys и public-link-keys.

— Защищённую копию файлов .env, compose.yaml и конфигурации веб-прокси.

— Перечень версий образов, с которыми создана копия.

Резервная копия содержит секреты и хранится в каталоге с ограниченным доступом вне основного диска сервера.

10.2. Создание резервной копии

Для согласованности базы данных и файлов отчётов остановите прикладные сервисы на время копирования. Сервисы postgres и vault не останавливаются.

```
BACKUP_DIR=/backup/panelyx-$(date +%Y%m%d%H%M)
mkdir -p "$BACKUP_DIR" && chmod 700 "$BACKUP_DIR"
APP="web frontend gateway identity survey response report"
APP="$APP report-py report-renderer notification settings"
dc stop $APP
dc exec -T postgres sh -c \
  'pg_dump -U "$POSTGRES_USER" -d "$POSTGRES_DB" -Fc' \
  > "$BACKUP_DIR/panelyx.dump"
dc exec -T postgres sh -c \
  'pg_dumpall -U "$POSTGRES_USER" --roles-only' \
  > "$BACKUP_DIR/roles.sql"
for v in exports_data identity-invite-keys public-link-keys
do
  docker run --rm -v "$(basename $PWD)_$v":/data:ro \
    -v "$BACKUP_DIR":/backup alpine \
    tar -C /data -czf /backup/$v.tgz .
done
```

```
cp .env compose.yaml nginx.conf "$BACKUP_DIR/"
dc images > "$BACKUP_DIR/images.txt"
dc start $APP
(cd "$BACKUP_DIR" && sha256sum ./* > SHA256SUMS)
```

Имена томов в команде указаны для проекта Docker Compose, имя которого совпадает с именем каталога установки. Фактические имена томов проверяются командой docker volume ls.

10.3. Восстановление

Восстановление выполняется из одного согласованного комплекта резервной копии:

1. Проверьте контрольные суммы файлов резервной копии командой sha256sum -c SHA256SUMS.

2. Остановите прикладные сервисы и сохраните отдельную копию текущего состояния.

3. Восстановите базу данных из дампа командой pg_restore в пустую базу данных, восстановите роли и права доступа.

4. Восстановите тома exports_data, identity-invite-keys и public-link-keys из архивов.

5. Запустите образы тех версий, с которыми создана резервная копия, и выполните контрольную проверку по разделу 9.2.

Периодически выполняйте пробное восстановление в отдельный проект Docker Compose, чтобы убедиться в пригодности резервных копий.

11. Обновление и откат версии

11.1. Обновление

Перед обновлением создайте резервную копию по разделу 10 и зафиксируйте текущие версии образов. Не смешивайте образы разных релизов: frontend, серверные сервисы и сервисы отчётов обновляются одним комплектом.

```
dc pull
dc up -d
dc ps
dc logs --tail=200 identity survey response report settings
```

Миграции базы данных применяются сервисами автоматически при запуске новых версий. После обновления выполните контрольную проверку по разделу 9.2.

11.2. Откат версии

Если изменились только совместимые образы, верните предыдущие версии образов и конфигурации и пересоздайте затронутые сервисы. Если новая версия изменила схему базы данных несовместимо, восстановите резервную копию, созданную перед обновлением, по разделу 10.3. Откат контейнеров не откатывает базу данных и секреты.

12. Диагностика типовых проблем

Типовые проблемы и способы их устранения приведены в таблице.

Симптом	Вероятная причина	Действия
port is already allocated	Порт занят другим процессом	Проверить занятые порты командой ss -ltn, изменить порт в конфигурации

Симптом	Вероятная причина	Действия
permission denied на сокете Docker	Нет доступа к Docker	Проверить права пользователя командой docker info
password authentication failed	Пароль роли PostgreSQL не совпадает со строкой подключения	Сверить роли и строки подключения, не удалять том pg_data
Ошибка 401 после входа	Разные значения JWT_SIGNING_KEY у identity и gateway	Согласовать ключ, пересоздать оба сервиса, войти повторно
Ошибка 403 или tenant not found	Нет прав или неверный адрес рабочего пространства	Проверить роль пользователя, адрес и базовый домен рабочих пространств
Ошибка 502 после обновления сервиса	Веб-прокси сохранил прежний адрес сервиса	Пересоздать сервис web
PDF не формируется	Сервис report-renderer недоступен или различаются внутренние токены	Проверить healthz сервиса, журналы report, report-py и report-renderer, значение REPORT_RENDERER_INTERNAL_TOKEN
Аналитика или XLSX не формируются	Сервис report-py недоступен	Проверить healthz report-py, журналы survey, response и report
Письма не доставляются	Неверные параметры SMTP или TLS	Проверить параметры SMTP, настройки организации и журнал notification
Контейнер завершается с кодом 137	Недостаточно оперативной памяти	Проверить docker stats и свободную память, увеличить ресурсы сервера

Для разбора инцидента соберите следующие сведения:

```
dc ps
dc logs --since=15m --tail=200 gateway identity settings
dc logs --since=15m --tail=200 report report-py report-renderer
docker stats --no-stream
df -h
```

Не передавайте журналы с персональными данными или токенами по незащищённым каналам.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	Номер докум.	Вход. номер сопров од. докум.	Подпись	Дата
	Изме-ненных	Заменен ных	Новых	Аннулиро ванных					