



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ

«ПРОТЕЙ Технологии»

Инструкция по установке «PromUC.Портал»

Оглавление

Руководство к инсталлятору	3
1. Введение	3
1.1 Разделы программы.....	3
1.2 Роли пользователей в программе.....	3
2. Системные требования	3
2.1 Операционная система.....	3
2.2 Интернет-соединение	3
2.3 Docker	3
2.4 Права доступа	3
3. Описание инсталлятора проекта `promis портал`	3
4. Скачивание и установка инсталлятора	5
4.1 Подготовка к установке.....	6
4.2 Запуск инсталлятора	7
4.3 Процесс установки	7
4.3.1 Проверка систем	7
4.3.2 Скачивание необходимых образов Docker	7
4.3.3 Настройка переменных окружения	7
4.3.4 Сборка и запуск контейнеров	7
4.5.1 Проверка корректности установки	8
4.5.2 Описание переменных окружения	9
5. Доступ к приложению и модулям	9
Доступ к контейнерам	9
Контейнеры и их порты:.....	10

Руководство к инсталлятору

1. Введение

В данном руководстве описаны требования и инструкция по установке программы PromUC.Портал с помощью инсталлятора.

1.1 Разделы программы

- PromUC Control – это раздел в программе PromUC.Портал, в котором администратор с ролью UI-Технолога вносит изменения в инструкциях, процессе SOP1 и в цепочках RuleEngine. Также он предназначен для, чтобы отслеживать действия пользователей с ролью UI-пользователя.
- ProScales – это раздел в программе PromUC.Портал, который выводит информацию об инцидентах для пользователей с ролью UI-пользователя.

1.2 Роли пользователей в программе

- UI - технолога (администратор)
- UI - пользователя (пользователь)

Следуйте пошаговым инструкциям, чтобы избежать ошибок во время установки.

2. Системные требования

2.1 Операционная система

- Linux (рекомендуются Ubuntu, Debian)

2.2 Интернет-соединение

- активное подключение к интернету (для загрузки образов и зависимостей)

2.3 Docker

- Version: 24.0.7
Docker Compose version v2.21.0
или более новая стабильная версия
- Поддержка Docker Compose в формате `docker compose` (встроенная, без необходимости устанавливать `docker-compose` отдельно)

2.4 Права доступа

- Наличие прав `sudo` для выполнения установочных скриптов

3. Описание инсталлятора проекта `promis портал`

Структура проекта `promis портал`

portal/

- |— install.sh # Основной установочный скрипт
- |— mqtt.yaml # Docker Compose файл для запуска MQTT и сервисов
- |— backend/ # Исходный код BPM-сервера (том Docker)
- |— env/ # Файлы переменных окружения
 - | |— bpm_backend.env # Переменные окружения для bpm-backend
 - | |— bpm-client.env # Переменные окружения для bpm-client
 - | |— client.env # Переменные окружения для ui/client
 - | |— server.env # Переменные окружения для ui/server
 - | |— mqtt.env # Переменные окружения для MQTT-брокера
 - | |— nginx.env # Переменные окружения для Nginx
 - | |— orm.env # Переменные окружения для ORM-сервиса
 - | |— postgres.env # Переменные окружения для PostgreSQL
 - | |— postgrest.env # Переменные окружения для PostgREST
 - | |— re.env # Переменные окружения для RuleEngine
 - | |— re_transport.env # Переменные окружения для RuleEngine-tcp-transport
- |— mqtt/ # Конфигурация MQTT-брокера (Nanomq)
 - | |— certs/ # SSL-сертификаты для MQTT (том Docker)
 - | | |— rootCA.crt # Корневой сертификат УЦ
 - | | |— test.promuc.local.cert # Сертификат сервера
 - | | |— test.promuc.local.key # Приватный ключ сервера
 - | |— conf/ # Конфигурационные файлы Nanomq
 - | | |— nanomq.conf # Основной конфиг Nanomq (том Docker)
 - | | |— nanomq_pwd.conf # Файл с учётными данными (том Docker)
 - | |— logs/ # Директория для логов (том Docker)
- |— nginx/ # Конфигурация Nginx
 - | |— conf.d/ # Директория с конфигами nginx

```

| | └─ dp.test.promuc.local.conf # Конфиг nginx (том Docker)
| └─ ssl/ # Файлы для генерации и хранения SSL-сертификатов (том Docker)
|   └─ new.cert # Временный сертификат (заготовка)
|   └─ new.csr # Certificate Signing Request
|   └─ openssl.cnf # Конфигурация OpenSSL
|   └─ test.promuc.local.cert # Сертификат
|   └─ test.promuc.local.key # Приватный ключ
└─ orm/ # Конфигурация orm
  └─ certs/ # Сертификаты и ключи (том Docker)
    └─ jwt-private.pem # Приватный ключ
    └─ jwt-public.pem # Публичный ключ
  └─ logs/ # Директория для логов orm (том Docker)
  └─ prod-logger.yaml # Конфигурация логирования в продакшене (том Docker)
└─ postgres/ # Конфигурация PostgreSQL
  └─ data/ # Том для хранения данных БД (том Docker)
└─ ui/ # Директория для сертификатов ui/server и orm (том Docker)
  └─ promuc_local.crt # SSL-сертификат для orm и ui/server (например, для доверия в браузере)

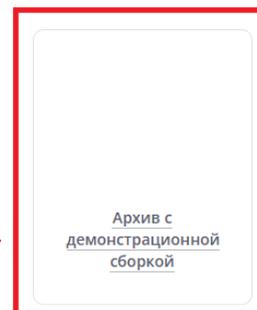
```

4. Скачивание и установка инсталлятора

Скачать инсталлятор можно с сайта <https://tl.protei.ru/products/programmnoe-obespechenie/promuc-portal-dlya-monitoringa-i-upravleniya-processami> в пункте «Задачи, которые позволяет решать платформа PromUC» нажав на «Архив с демонстрационной сборкой»

Задачи, которые позволяет решать платформа PromUC

- Унифицировать обмен сообщениями для различных систем
- Трансформировать сбор телеметрии в потоки событийно-ориентированных сообщений
- Сформировать бизнес-логику
- Контролировать выполнение.



4.1 Подготовка к установке

4.1.1 Разархивируйте архив:

```
bash
tar -xvf <имя_архива>.tar.gz
```

4.1.2 Перейдите в директорию <имя архива>:

```
bash
cd <имя архива>
```

4.1.3 Перейдите в директорию portal:

```
bash
cd portal
```

4.1.4 Проверьте является ли файл install.sh исполняемым, прописав:

```
bash
ls -l install.sh
```

Если файл исполняемый, вы увидите буквы x в правах доступа (например, -rwxr-xr-x)

Если перед install.sh стоит -rw-r--r-- (без x), значит файл не является исполняемым.

4.1.5 Сделайте скрипт исполняемым (если нужно):

```
bash
chmod +x install.sh
```

Теперь вы готовы запустить установку.

4.2 Запуск инсталлятора

```
bash
```

```
sudo ./install.sh
```

Необходимо ввести логин и пароль для Docker registry

Логин: promucadmin

Пароль: PromUCpass

4.3 Процесс установки

Во время установки инсталлятор сам выполняет следующие действия:

4.3.1 Проверка систем

- Проверка присутствия Docker и Docker Compose
- Проверка подключения к интернету

4.3.2 Скачивание необходимых образов Docker

- Извлечение образов из репозитория (Docker Hub и dr.promuc.local:5000)

4.3.3 Настройка переменных окружения

- Определяет IP-адрес хоста
- Обновляет .env-файлы, подставляя актуальный IP-адрес в переменные окружения

4.3.4 Сборка и запуск контейнеров

- Ожидание готовности PostgreSQL
- Проверяет, существует ли БД bpm_db.
- Если БД bpm_db не существует — создаёт её:

В скрипте install.sh с 114 по 127 строку

Выполняется команда "SELECT 1 FROM pg_database WHERE datname='\$NEW_DB'" внутри контейнера.

Если БД существует, то в переменную DB_EXISTS записывается 1.

Если ее нет, то пустая строка.

Далее проверяется эта переменная, DB_EXISTS ==1, БД существует.

Если DB_EXISTS != 1 то внутри контейнера postgres выполняется команда "\$DB_USER" -с "CREATE DATABASE '\$NEW_DB';"

- Запуск всех контейнеров

4.5 Проверка установки

4.5.1 Проверка корректности установки

После окончания установки ничего не выводится, нужно прописывать в терминал

```
bash
```

```
sudo docker ps
```

Команда `sudo docker ps` выводит список запущенных контейнеров.

Все контейнеры должны иметь статус `up` (предполагается, что на сервере нет сторонних запущенных контейнеров)

```
hhlyae@re-spb2: /final_build/portal$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                                CREATED        STATUS        PORTS
8f4f4a78cccd   nginx                                "/docker-entrypoint..."             27 seconds ago Up 25 seconds 0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp, 0.0.0.0:4000->4000/tcp, :::4000->4000/tcp
ad970ef30f0c   dr.promuc.local:5000/ui/client:mincifra "/docker-entrypoint..."             27 seconds ago Up 25 seconds 80/tcp, 4100/tcp
80e5ed33aaea   dr.promuc.local:5000/dev_re/ruleengine-re_front:mincifra "nginx -g 'daemon of..."           27 seconds ago Up 25 seconds 80/tcp
c399fb10c589   dr.promuc.local:5000/ui/server:mincifra "docker-entrypoint.s..."           27 seconds ago Up 26 seconds 0.0.0.0:4200->4200/tcp, :::4200->4200/tcp
39abaa51fc90   dr.promuc.local:5000/dev_re/ruleengine-re:mincifra ". /ruleengine --verb..."          27 seconds ago Up 26 seconds 0.0.0.0:5554-5555->5554-5555/tcp, :::5554-5555->5554-5555/tcp, 0.0.0.0:7777->7777/tcp, :::7777->7777/tcp
5f81c48c992    dr.promuc.local:5000/orm/promuc-orm:0.1-dev ". /run prod"                         27 seconds ago Up 16 seconds 0.0.0.0:4840->4840/tcp, :::4840->4840/tcp, 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
b9e0f4f04a26   portal-bpm-backend                    "sh -c '\n mkdir ..."           27 seconds ago Up 26 seconds 0.0.0.0:8000->8000/tcp, :::8000->8000/tcp
6c9379e0b2ce   postgres/postgres                     "/bin/postgres"                      27 seconds ago Up 26 seconds 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
efdf5f3d4d63   emqx/nanomq:0.20.0-full               "/usr/bin/docker-ent..."          27 seconds ago Up 26 seconds 0.0.0.0:1883->1883/tcp, :::1883->1883/tcp, 0.0.0.0:8083->8083/tcp, 8081/tcp
029f8ad20c41   dr.promuc.local:5000/ui/bpm-client:mincifra "/docker-entrypoint..."           27 seconds ago Up 26 seconds 80/tcp, 0.0.0.0:5100->5100/tcp, :::5100->5100/tcp
1d8c24112cb4   timescale/timescaledb-ha:pg16         "/docker-entrypoint..."             27 seconds ago Up 26 seconds (healthy) 5432/tcp, 8008/tcp, 8081/tcp
0bee2ff41578   dr.promuc.local:5000/dev_re/ruleengine-re_transport ". /promuc_tcp_transp..."          27 seconds ago Up 26 seconds 47808/tcp
012cd57e1969   redis:latest                           "docker-entrypoint.s..."           27 seconds ago Up 26 seconds 6379/tcp
```

Если установка прошла некорректно, то все зависит от ошибки.

Список ошибок, обрабатываемых скриптом:

Ошибка	Решение
docker не установлен	Установить docker
docker compose не установлен	Установить docker compose
Ошибка входа в реестр (неверный логин/пароль, недоступен реестр)	Проверить вводимый логин/пароль. Если реестр недоступен дать обратную связь (Может произойти, если электричество выключат)
Ошибка загрузки образа (проблемы с сетью)	Такое бывает из-за плохого соединения, запустить скрипт снова (иногда образы с docker hub не скачиваются с первого раза)

Не удалось определить публичный IP	Скрипт не может автоматически определить внешний IP-адрес сервера для настройки переменных окружения. Это происходит на этапе: PUBLIC_IP=\$(ip route get 1.1.1.1 grep -oP 'src \K\S+') Возможные причины: Отсутствие сетевого подключения к внешним адресам (1.1.1.1). Нет активных сетевых интерфейсов. Сервер находится за NAT или в изолированной сети. Команда ip недоступна в системе.
Ошибка при создании БД bpm_db	Если содержимое файлов было отредактировано, то провести установку с самого начала с распаковки архива
В случае, если не входит в ProScales под юзером root/root.	Перезагрузить контейнер orm при помощи команды: docker restart orm Затем, снова войти в ProScale под юзером root/root

4.5.2 Описание переменных окружения

Описание переменных окружения находится внутри каждого .env файла в директории ../portal/env

5. Доступ к приложению и модулям

Посмотреть ip-адрес-сервера можно с помощью команд, введённых в терминал:

```
ip addr show
```

```
ip a
```

```
hostname -l
```

```
ip route get 1.1.1.1 | awk '{print $7}'
```

```
ip route get 1.1.1.1 | grep -oP 'src \K\S+'
```

Приложение / компонент	URL:	Логин Пароль
ProScales	https://<ip-адрес-сервера>:4000	root root
PromUC control	http://<ip-адрес-сервера>:5100	<u>admin@protei.tl</u> admin
RE (конструктор цепочек)	https://<ip-адрес-сервера>:443	- -

Доступ к контейнерам

Все контейнеры находятся в одной Docker-сети. Для обращения к контейнеру можно

использовать его имя или IP-адрес сервера.

Пример: чтобы подключиться к MQTT-брокеру, нужно указать `:1883`.

Контейнеры и их порты:

Контейнер	Порты
postgrest	3000
server	4200
re	5555, 7777, 5554
orm	8080, 4840
mqtt5	1883, 8883, 8083, 8084
bpm-client	5100
bpm-backend	8000

6. Контакты технических специалистов

Для консультации по поводу установки и настройки ПО можно обращаться:

1. Валерий Васильев vasiliev-vv@tl.protei.com