

Инструкция по установке экземпляра

ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

«АБиоГрам.Геном» («ABioGram.WGS»)

Аннотация

Настоящий документ содержит ссылку для скачивания исходного кода, а также инструкции по установке программного обеспечения (ПО) «АБиоГрам.Геном» («ABioGram.WGS»).

Оглавление

Аннотация 2
Оглавление 3
Общая информация 4
1. Состав и требования 4
2. Конфигурация 4
3. Сборка и запуск 5
4. Первый вход 6
5. Остановка и удаление 6

Общая информация

Получите экземпляр ПО.

1. Состав и требования

Для локальной установки используйте Docker Compose со сборкой из репозитория. При первой сборке нужен доступ в интернет для загрузки образов и зависимостей.
Нужны Docker с поддержкой Linux-контейнеров, Compose, исходный код, доступ к загрузке базовых образов и зависимостей, свободные порты и постоянный диск. Поддерживается запуск в Linux и Windows.
На Linux установите Docker Engine с Compose plugin и запустите Docker daemon. На Windows установите и запустите Docker Desktop с WSL 2 и режимом Linux containers. Команда docker info должна выполняться без ошибки подключения к daemon. В Linux текущему пользователю нужен доступ к Docker; если его нет, выполняйте Docker-команды через sudo или настройте доступ к daemon.
Объём оперативной памяти, число ядер процессора и дисковое пространство зависят от нагрузки лаборатории. Установка средств разработки на рабочее место обычного пользователя не требуется.
Состав окружения: frontend, backend, PostgreSQL 18.6 и четыре обработчика: worker-fasta, worker-fasta-anonymous, worker-mlst, worker-mlst-anonymous. API и все worker должны видеть общее /files.
Команды выполняются из корня репозитория, например D:\git\abg-wgs. Далее приведён порядок установки нового окружения. Перенос существующей БД со старой историей миграций требует отдельного плана.

2. Конфигурация

1. Получите код.
2. Перейдите в корневой каталог проекта и проверьте установленный Docker следующими командами:

```
docker version
docker compose version
docker info
```

3. Если docker/.env отсутствует, создайте его из примера. Существующий файл не перезаписывайте:
Linux, Bash:

```
test -f docker/.env || cp docker/.env.example docker/.env
```

Windows, PowerShell:

```
if (-not (Test-Path docker/.env)) {
    Copy-Item docker/.env.example docker/.env
}
```

4. Откройте docker/.env в текстовом редакторе. Для пробного локального запуска можно использовать значения из примера. POSTGRES_PASSWORD определяет пароль подключения к PostgreSQL, JWT_SIGNING_KEY определяет ключ подписи токенов; это не пароль пользователя интерфейса. Для собственного окружения задайте отдельный пароль БД и случайный JWT-ключ длиной не менее 64 ASCII-символов.
5. При необходимости измените DEBUG_API_PORT, DEBUG_WEB_PORT, DEBUG_POSTGRES_PORT. Значения по умолчанию – 5000, 8080 и 15432.
6. Для другого рабочего места задайте DEBUG_API_URL и DEBUG_WEB_URL с достижимым адресом Docker-хоста. API URL оканчивается на /api/, frontend URL – на /.

Пример содержит фиктивные значения. Не используйте их для рабочей установки. Файл .env храните с ограниченным доступом; не включайте секреты в исходники, отчёты и скриншоты.

3. Сборка и запуск

Для локального запуска используется Compose-проект abg-wgs-debug, образы abg-wgs/backend:debug, abg-wgs/frontend:debug, abg-wgs/worker:debug и сеть abg-wgs (по умолчанию Docker создаёт её как abg-wgs-debug_abg-wgs).

Все следующие команды выполняйте из корня репозитория. Сначала проверьте настройки и соберите образы. Одинаковые команды для Bash и PowerShell:

```
docker compose --env-file docker/.env -f
docker/debug.compose.yml config --quiet
docker compose --env-file docker/.env -f
docker/debug.compose.yml build
```

4. Первый вход

1. При пустой таблице пользователей initializer создаёт root с паролем 1234567890.
2. Войдите и смените пароль через профиль.
3. Создайте персональные записи сотрудников.
4. Заполните справочники и дополнительные поля.
5. Загрузите и проверьте FASTA/MLST-базы, сопоставления и готовность.
6. При необходимости настройте SMTP и шаблоны.

5. Остановка и удаление

Команды выполняет администратор из корневого каталога репозитория. Команды ниже одинаковы для Bash и PowerShell. Перед остановкой прекратите новые запуски и дождитесь завершения активных заданий в разделе «Задания».

1. Для временной остановки без удаления контейнеров выполните:

```
docker compose --env-file docker/.env -f docker/debug.compose.yml stop
```

БД и файлы сохраняются. Чтобы запустить остановленные контейнеры, выполните:

```
docker compose --env-file docker/.env -f docker/debug.compose.yml start
```

2. Для остановки и удаления контейнеров и сети окружения с сохранением данных выполните:

```
docker compose --env-file docker/.env -f docker/debug.compose.yml down
```

Без параметра -v сохраняются volumes postgres-pg18-debug-data и files-debug-data. При стандартном имени проекта это abg-wgs-debug_postgres-pg18-debug-data и abg-wgs-debug_files-debug-data. В них остаются БД, загруженные файлы, базы анализа и результаты. Для повторного запуска используйте:

```
docker compose --env-file docker/.env -f docker/debug.compose.yml up -d --build
```

Используйте прежние имя Compose-проекта и файл настроек. Если при установке задавали имя через -p, передавайте то же значение во все команды остановки, удаления и запуска.

3. Для полного удаления данных сначала сохраните резервную копию PostgreSQL и общего файлового хранилища /files вне удаляемых volumes и

проверьте возможность восстановления. Выгрузки Excel не заменяют резервное копирование установки.
Следующую команду выполняйте только при намеренном удалении всей установки вместе с данными:

```
docker compose --env-file docker/.env -f docker/debug.compose.yml down -v
```