# Установка через Docker

Используйте современные возможности контейнерных технологий для быстрого развертывания Devprom ALM на вашем Linux-сервере или Windows-сервере.

### Установка Docker

Ubuntu/Debian

```
sudo -s
apt-get update && apt-get -y install docker.io docker-compose
systemctl enable docker --now
```

CentOS/RedHat

```
sudo -s
yum -y install curl docker-compose git && (curl -sSL https://get.docker.com | sh)
systemctl enable docker --now
```

### Установка Devprom ALM

```
mkdir -p /var/www/devprom/logs /var/www/devprom/update /var/www/devprom/backup
/home/devprom
cd /home/devprom

wget --no-check-certificate -O devprom.zip https://devprom.ru/download
unzip -q -a devprom.zip
mv devprom /var/www/devprom/htdocs
chown -R 33:33 /var/www/devprom

git clone https://github.com/devprom-dev/docker.git
cd docker
```

Измените значения по умолчанию в файле .env

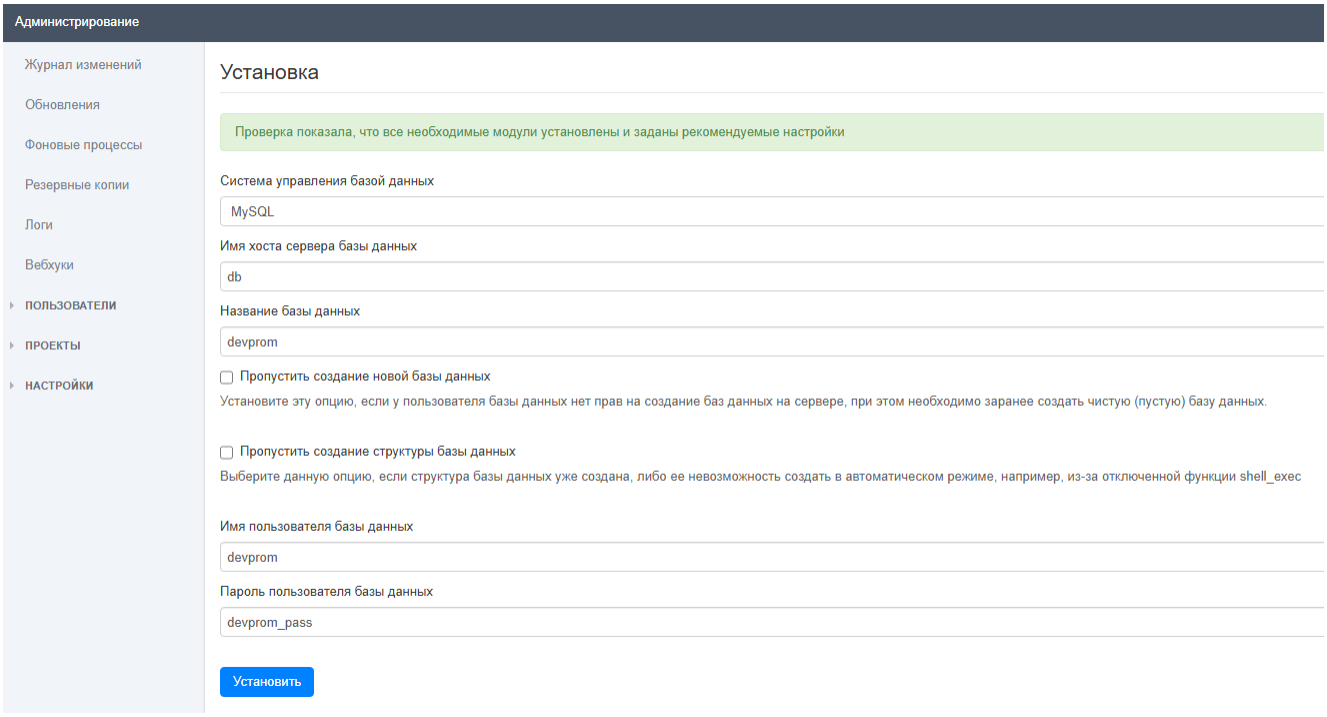
```
vi .env
```

Установите и запустите контейнеры

```
docker-compose up -d
```

### Использование СУБД MySQL

Откройте браузер, перейдите к приложению. Укажите пароль пользователя MySQL, значение которого задано в переменной MYSQL_PASSWORD файла .env (по умолчанию devprom_pass)




Нажмите кнопку "Установить" и дождитесь завершения установки.

### Использование СУБД PostgreSQL

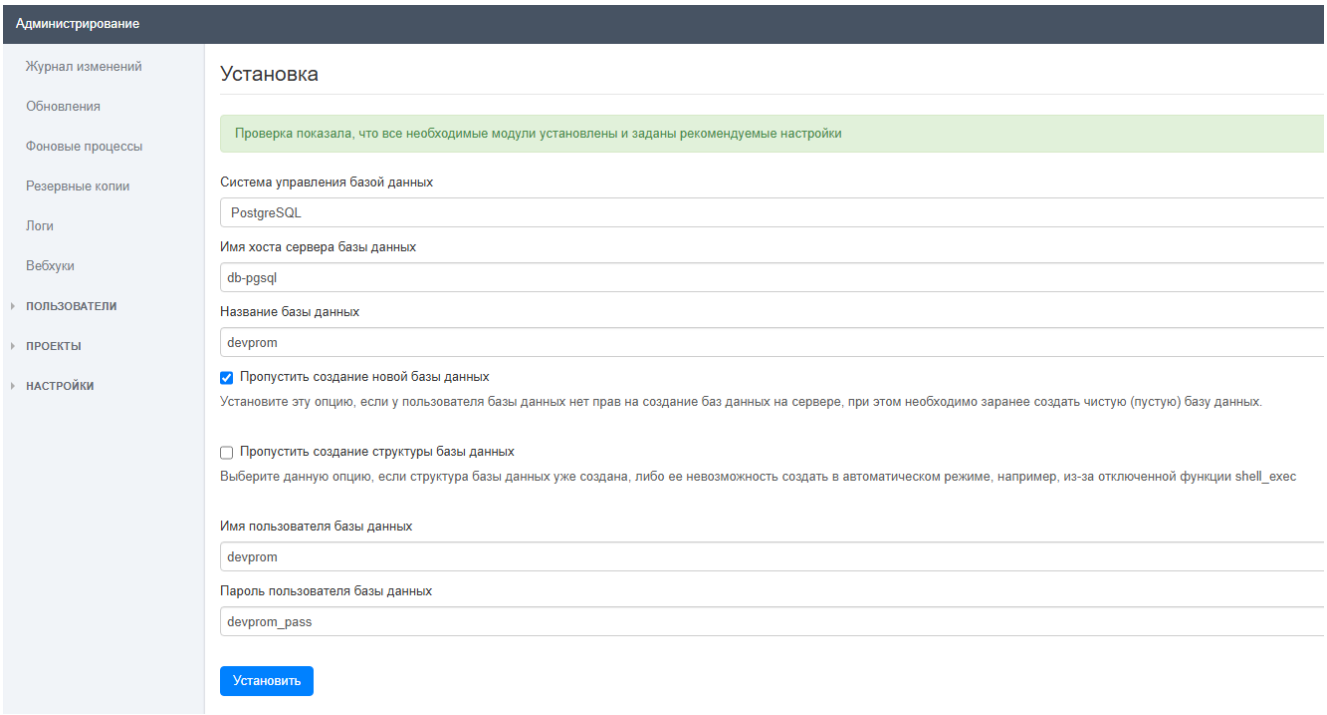
Запустите контейнер с СУБД

```
docker-compose -f pgsql.yml up -d
```

Откройте браузер, перейдите к приложению и заполните поля как на скриншоте ниже:

- Имя хоста СУБД: db-pgsql
- Пропустить создание новой базы данных

Укажите пароль пользователя СУБД, значение которого задано в переменной MYSQL_PASSWORD файла .env (по умолчанию devprom_pass)



Нажмите кнопку "Установить" и дождитесь завершения установки.

Еще полезные статьи по настройке ПО:

- Настройка доступа по HTTPS
- Авторизация через LDAP

**Резервное копирование**

Автоматически ежедневно формируемые резервные копии будут доступны на хосте в каталоге /var/www/devprom/backup, организуйте их резервирование в отдельное хранилище.

**Использование ИИ-функций**

В функциональность ALM встроены ИИ-функции, повышающие продуктивность команд при работе с проектными артефактами. Для их использования необходимо установить дополнительные компоненты, при помощи команды:

```
docker-compose -f aitools.yml up -d
```

Перечень и назначение компонентов:

| Название компонента | Назначение компонента |
|---|---|
| mcp | MCP-сервис, предоставляющий ИИ-агентам понятное расширенное описание API для работы с проектными артефактами (чтение, создание, модификация), поиска по смыслу (RAG) и т.п. Для использования в ИИ-агенте необходимо подключить MCP-сервис:<br><br>`{`<br>`  "mcpServers": {`<br>`  "mcp-alm": {`<br>`  "url": "http://<адрес сервера>/mcp",`<br>`  "headers": {`<br>`  "Devprom-Auth-Key": "***",`<br>`  "Devprom-Base-Url": "http://<адрес сервера>"`<br>`  }`<br>`  }`<br>`  }`<br>`}`<br><br>`Devprom-Auth-Key`: API-ключ пользователя из-под которого будут выполняться операции в системе, при работе ИИ-агента. Рекомендуем завести отдельного пользователя и включать в нужные проекты с минимально необходимыми правами, чтобы он лишнего чего не натворил.<br><br>`Devprom-Base-Url`: адрес сервера, по которому доступно само приложение ALM. |
| chromadb | Векторная СУБД для хранения embeddings (векторизованных представлений пользовательских данных). Векторы формируются и кешируются при создании и изменении данных, после чего используются ИИ-функциями без расходования вычислительных ресурсов языковых моделей. |
| ollama | Открытый сервис управления ИИ-моделями, позволяет подключать платные и бесплатные языковые модели и другие специфические ИИ-модели. Данные сервис предназначен для локального использования ИИ-функций (без доступа в интернет), однако, требует специальных вычислительных ресурсов (напр., GPU), чтобы время и |

| Название компонента | Назначение компонента |
|---|---|
| | качество работы моделей было приемлемым. Для использования моделей Ollama необходимо выполнить настройку в разделе Администрирование - Настройки - Приложение.<br><br>{<br>"base_url": "http://ollama:11434",<br>"embedding_doc_model": "nomic-embed-text-v2-moe",<br>"model": "llama3.2:latest",<br>"completion_timeout": 120<br>}<br><br>Если используются внешние LLM-модели (например, GigaChat или YandexGPT), то данный компонент не требуется. Настройка использования внешней модели расположена в разделе Администрирование - Настройки - Приложение. |
| ollama_models | Временный контейнер, выполняющий установку бесплатной модели для генерации векторных представлений (embeddings) для ознакомительных целей. Для загрузки модели в сервис Ollama требуется подключение к Интернет. |

Для оценки стоимости векторизации текстов проектных артефактов внешними (платными) моделями можно воспользоваться следующим запросом к БД:

```
SELECT SUM(LENGTH(SearchContent) - LENGTH(REPLACE(SearchContent, ' ', '')))
  FROM pm_Searchable
 WHERE ObjectClass IN ('Component', 'ProjectPage', 'Comment', 'Requirement',
'TestScenario', 'Feature', 'HelpPage', 'Request', 'Issue', 'Increment');
```

Например, для 18 млн. слов может потребоваться ~48 млн. токенов и ~1.6 ГБ свободного места на диске для векторной СУБД.

**Развертывание кластера**

Для промышленной (боевой) эксплуатации при значительных нагрузках (> 100 одновременно работающих пользователей) рекомендуем выполнить развертывание кластера.