

Приложение № 6  
к приказу ООО "Абсолют Технологии"  
№ ПА-011024-3 от 01.10.2024



**Абсолют Технологии**  
создаем возможности

**Личный кабинет  
HR ДМС  
Инструкция по установке.**

г. Москва, 2024 год

# ОГЛАВЛЕНИЕ

1. РАЗВЕРТЫВАНИЕ ПРОЕКТА НА PRODUCTION СЕРВЕРЕ С ПОМОЩЬЮ КОНТЕЙНЕРИЗАЦИИ DOCKER .....	3
1.1. Требования к аппаратному оборудованию сервера: .....	3
1.2. Требования к программному обеспечению сервера: .....	3
2. ПОДГОТОВКА СЕРВЕРА .....	3
2.1. Предварительные требования к настройке сервера .....	3
2.2. Предварительные шаги .....	3
3. УСТАНОВКА И НАСТРОЙКА DOCKER И ПЛАГИНА COMPOSE .....	3
3.3. Установка последних версий пакетов Docker .....	4
4. РАЗВЕРТЫВАНИЕ РЕПОЗИТОРИЕВ BACKEND И FRONTEND .....	5
4.1. Создание директории для хранения файлов .....	5
4.3. Клонирование репозитория FRONTEND .....	5
5. НАСТРОЙКИ ОКРУЖЕНИЯ BACKEND-СЕРВЕРА .....	6
5.1. Настройка файла .env .....	6
6. НАСТРОЙКИ ОКРУЖЕНИЯ BACKEND-СЕРВЕРА .....	6
6.1. Настройка файла .env .....	6
6.2. Список переменных файла .env .....	7
6.3. Сборка и настройка контейнеров Docker .....	11
6.4. Список команд make .....	12

## 1. РАЗВЕРТЫВАНИЕ ПРОЕКТА НА PRODUCTION СЕРВЕРЕ С ПОМОЩЬЮ КОНТЕЙНЕРИЗАЦИИ DOCKER

### 1.1. Требования к аппаратному оборудованию сервера:

- Процессор - 8 ядер и больше, с тактовой частотой от 3,6 ГГц;
- ОЗУ - 32 ГиБ и больше;
- Системный диск - SSD от 256 ГиБ;
- Дисковое хранилище - от 2 ТиБ;
- Пропускная способность локальной сети - от 100 Мбит/сек;

### 1.2. Требования к программному обеспечению сервера:

- ОС Ubuntu версии LTS - 20.04 и выше или Debian последней версии (актуальная 12.2);
- сервер OpenSSH - для доступа по протоколам ssh и sftp;

## 2. ПОДГОТОВКА СЕРВЕРА

### 2.1. Предварительные требования к настройке сервера

- Удаленный доступ по протоколам ssh и sftp;
- Настроенный пользователь с доступом к sudo;

### 2.2. Предварительные шаги

#### 2.2.1. Обновите репозитории Ubuntu:

```
sudo apt update
```

#### 2.2.2. Обновите установленные пакеты и библиотеки:

```
sudo apt upgrade -y
```

Программа обновления может запросить перезагрузку сервисов, использующих устаревшие версии обновленных библиотек, необходимо выбрать все и подтвердить перезагрузку нажатием пункта <ок>

### 2.3. Установка дополнительного ПО

#### 2.2.1. Вспомогательные библиотеки для работы с сетью и интернет:

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common gnupg net-to
```

#### 2.2.2. Необходимый софт:

```
sudo apt install git -y
```

## 3. УСТАНОВКА И НАСТРОЙКА DOCKER И ПЛАГИНА COMPOSE

### 3.1. Установка ключа шифрования GPG официального репозитория Docker:

```
sudo apt-get update sudo install -m 0755 -d /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

### 3.2. Добавление репозитория в источники Apt:

```
echo \  
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]  
https://downloa $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

### 3.3. Установка последних версий пакетов Docker

На этом шаге устанавливаются пакеты, необходимые для обеспечения контейнеризации и управления конфигурацией контейнеров, образов и томов данных:

- docker-ce
- docker-ce-cli
- containerd.io
- docker-buildx-plugin
- docker-compose-plugin

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-  
compose-pl
```

### 3.4. Шаги после установки Docker

В первую очередь необходимо проверить, успешно ли установлен Docker и плагин Compose. Введите в консоли команду:

```
docker -v
```

Должен появиться вывод, подобный такому:

```
Docker version 24.0.7, build afdd53b
```

Далее проверяем версию плагина compose: `docker compose version`

Ожидаем вывода сообщения: `Docker Compose version v2.21.0`

Если одна или обе команды выдают сообщения об ошибках, необходимо перед переходом к следующим шагам исправить ошибки.

Проверяем, создана ли на сервере группа `docker`:

```
getent group | cut -d: -f1 | grep docker
```

Если группа не существует, создаем командой:

```
sudo groupadd docker
```

Далее, проверяем список групп текущего пользователя, для этого запускаем команду:

```
groups $USER
```

Если группа `docker` отсутствует в списке, добавляем пользователя в группу:

```
sudo usermod -aG docker $USER
```

Проконтролируйте, добавился ли пользователь в группу.

Проверяем возможность запуска Docker непривилегированным пользователем, вводим команду:

```
docker ps
```

Скорее всего будет выведено сообщение об ошибке типа:

permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.s

Нужно сменить набор разрешений для доступа к сокету docker:  
sudo chmod 666 /var/run/docker.sock

После этого выполнение команд Docker должно стать доступным, убедиться в этом можно, повторно выполнив команду docker ps, она должна вывести пустой список контейнеров:

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
```

На этом настройка Docker закончена.

## 4. РАЗВЕРТЫВАНИЕ РЕПОЗИТОРИЕВ BACKEND И FRONTEND

### 4.1. Создание директории для хранения файлов

Для примера, будем хранить репозитории проекта в директории /var/data.

Проверим, существует ли она: ls -la /var | grep data

В чистой инсталляции Ubuntu каталог /var/data обычно отсутствует, если же он уже создан, лучше выбрать другое наименование, например, srv, чтобы папки с таким названием не было в директории /var. Создаем каталог:

```
sudo mkdir -p /var/data
```

Меняем владельца каталога на текущего пользователя и устанавливаем права доступа 0755: sudo chown "\$USER":"\$USER" /var/data && sudo chmod 0755 /var/data

Создаем директории для репозитория фронтэнда и бэкенда:

```
mkdir -p /var/data/{frontend,backend}
```

Проверяем, созданы ли директории:

```
cd /var/data ls -la
```

Ожидаем увидеть такую структуру:

```
total 16
```

```
drwxr-xr-x 4 batman batman 4096 Dec 11 11:28 .
```

```
drwxr-xr-x 14 root root 4096 Dec 11 11:22 ..
```

```
drwxrwxr-x 2 batman batman 4096 Dec 11 11:28 backend
```

```
drwxrwxr-x 2 batman batman 4096 Dec 11 11:28 frontend
```

### 4.2. Клонирование репозитория BACKEND

Вводим команды:

```
cd /var/data/backend
```

```
git clone https://{path-to-backend-repository} .
```

Не забываем точку в конце последней команды, иначе будет создан подкаталог с названием репозитория.

### 4.3. Клонирование репозитория FRONTEND

Вводим команды:

```
cd /var/data/frontend git clone https://{path-to-frontend-repository} .
```

Снова не забываем про точку.

## 5. НАСТРОЙКИ ОКРУЖЕНИЯ BACKEND-СЕРВЕРА

### 5.1. Настройка файла .env

В первую очередь необходимо создать файл настроек приложения .env из шаблона .env.example:

```
cd /var/data/backend  
cat .env.example >> .env
```

Проверьте, удалось ли создать файл .env и отредактируйте его командой (для примера):

```
nano .env
```

Полный список настроек приведен ниже, настройки в списке разбиты по секциям.

Основные настройки, которые нужно установить в первую очередь:

- PROJECT\_DOMAIN - должен совпадать с доменом (поддоменом), настроенным для сервера backend;
- APP\_ENV - установите значение production;
- APP\_KEY - удалите дефолтное значение, в дальнейшем мы его регенерируем для обеспечения безопасности;
- APP\_URL - параметр должен содержать базовый URL приложения (например, <https://backend.sk.ru> без закрывающего слэша);
- PROJECT\_FRONTEND\_DOMAIN - параметр должен содержать базовый URL фронтэнда (например, <https://sk.ru> без закрывающего слэша);
- APP\_DEBUG - обязательно false;
- ENABLE\_XDEBUG - обязательно false.

## 6. НАСТРОЙКИ ОКРУЖЕНИЯ BACKEND-СЕРВЕРА

### 6.1. Настройка файла .env

В первую очередь необходимо создать файл настроек приложения .env.local из шаблона .env.defaults:

```
cd /var/data/frontend  
cat .env.defaults >> .env.local
```

Проверьте, удалось ли создать файл .env.local и отредактируйте его командой (для примера):

```
nano .env.local
```

Необходимо настроить параметры:

- VITE\_API\_URL=<https://absolut-lk-hr-backend.dkr.zetest.site/api/v1>
- API\_DOCK=<https://absolut-lk-hr-backend.dkr.zetest.site/swagger/docs/api-v1.json>

Замените [absolut-lk-hr-backend.dkr.zetest.site](https://absolut-lk-hr-backend.dkr.zetest.site) на домен (поддомен) сервера бэкенд. Сохраните изменения в файле и переходите к разделу

**Сборка и настройка контейнеров Docker.**

## 6.2. Список переменных файла .env

### 6.2.1. Общие настройки проекта

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
PROJECT_NAME	"absolut-lk-hr-backend"	Наименование проекта, используется в доменном имени приложения и в URL	
PROJECT_DOMAIN	"\${PROJECT_NAME}.test"	Домен, на котором расположен бэкэнд проекта	
PROJECT_INTERFACE	"127.0.0.1"	IP, на котором доступен проект	Только для локальной разработки

### 6.2.2. Настройки приложения Laravel

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
APP_NAME	"\${PROJECT_NAME}"	Наименование приложения Laravel	
APP_ENV	local	Окружение приложения	На продакшн должно быть выставлено в production
APP_KEY	base64:kMX4oKjP3GWJJ08TP9BdMS/rC8IM6Q0XYlaeUghVOfY=	Ключ шифрования приложения	Необходимо регенерировать для прод-окружения
APP_URL	"http://\${PROJECT_DOMAIN}"	Базовый URL бэкэнд-приложения	
PROJECT_FRONTEND_DOMAIN	"https://absolut-lk-hr-front.dkr.zetest.site"	Домен, на котором расположен frontend	

### 6.2.3. Режим отладки

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
APP_DEBUG	true	Режим отладки	Необходимо отключить в продокружении
ENABLE_XDEBUG	"\${APP_DEBUG}"	Расширение PHP XDEBUG	По умолчанию включен для режима отладки

### 6.2.4. Настройки логирования

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
LOG_CHANNEL	daily	Наименование канала логгера	
LOG_DEPRECATED_CHANNELS	null		

LOG_LEVEL	debug	Уровень сообщений для логирования	
-----------	-------	-----------------------------------	--

### 6.2.5. Настройки подключения к БД

Подключения к основной БД и к БД для запуска автотестов.

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
DB_CONNECTION	mysql	Наименование подключения к БД	Используется в конфигурации Laravel
DB_HOST	database	Хост сервера БД	
DB_TEST_HOST	test-db	Хост сервера тестовой БД	
DB_PORT	3306	Порт сервера БД	
DB_TEST_PORT	3308	Порт сервера тестовой БД	
DB_DATABASE	absolut	Наименование БД	Совпадает для основной и тестовой БД.
DB_USERNAME	absolut	Имя пользователя	Совпадает для основной и тестовой БД.
DB_PASSWORD	s3cr3t	Пароль пользователя	Совпадает для основной и тестовой БД
ACTIVITY_LOGGER_DB_CONNECTION	mysql	Подключение для работы расширения Activity Logger	

### 6.2.6. Настройки прочих подключений

В этом разделе настраиваются драйверы вещания, кэша, файлового хранилища, сессий и подключения очередей.

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
BROADCAST_DRIVER	log	Драйвер вещания (websockets)	
CACHE_DRIVER	file	Драйвер кэша	Доступны значения "apc", "array", "database", "file", "memcached", "redis", "dynamodb", "octane", "null"
FILESYSTEM_DISK	local	Драйвер файлового хранилища	
QUEUE_CONNECTION	redis	Подключение для очереди по умолчанию	"sync", "database", "beanstalkd", "sqs", "redis", "null"
EMAILS_QUEUE_CONNECTION	redis-emails	Очередь отправки писем	
SESSION_DRIVER	file	Драйвер сессий PHP	"file", "cookie", "database", "apc", "memcached", "redis", "dynamodb", "array"



SESSION_LIFETIME	120	Время жизни сессии PHP	В минутах
------------------	-----	------------------------	-----------

### 6.2.7. Подключение к серверу memcached

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
MEMCACHED_HOST	127.0.0.1	Хост сервера memcached	

### 6.2.8. Подключение к серверу Redis

Сервер **Redis** используется для организации работы очередей Laravel, обеспечивающих отправку почты и прочие возможности.

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
REDIS_HOST	redis	Хост сервера redis	
REDIS_PASSWORD	null	Пароль пользователя redis	
REDIS_PORT	6379	Порт сервера redis	
REDIS_QUEUE	queue		
REDIS_EMAILS_QUEUE	emails		
REDIS_DB	"0"	БД Redis по умолчанию	
REDIS_CACHE_DB	"1"	БД Redis для работы кэша	
REDIS_EMAILS_DB	"2"	БД Redis для очереди отправки писем	

### 6.2.9. Настройки отправки email через smtp

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
MAIL_MAILER	smtp	Драйвер сервиса отправки писем	"smtp", "sendmail", "mailgun", "ses", "ses-v2", "postmark", "log", "array", "failover"
MAIL_HOST	smtp-mail	Хост сервера SMTP	
MAIL_PORT	1025	Порт сервера SMTP	
MAIL_USERNAME		Имя пользователя	
MAIL_PASSWORD		Пароль пользователя	

MAIL_ENCRYPTION	null	Шифрование при отправке почты	"tls", "ssl", or "null"
MAIL_FROM_ADDRESS	"info@\${PROJECT_DO MAIN}"	Адрес отправителя почты	
MAIL_FROM_NAME	"\${APP_NAME}"	Имя отправителя почты	

#### 6.2.10. Настройки деплоя на Amazon Web Services

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
AWS_ACCESS_KEY_ID			
AWS_SECRET_ACCESS_KEY			
AWS_DEFAULT_REGION	us-east-1		
AWS_BUCKET			
AWS_USE_PATH_STYLE_ENDPOINT	false		

#### 6.2.11. Настройки websockets при использовании пакета Pusher

Наименование переменной окружения    Значение по умолчанию    Описание  
Комментарий

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
PUSHER_APP_ID			
PUSHER_APP_KEY			
PUSHER_APP_SECRET			
PUSHER_HOST			
PUSHER_PORT	443		
PUSHER_SCHEME	https		
PUSHER_APP_CLUSTER	mt1		

#### 6.2.12. Настройки сборки ассетов websockets при использовании пакета Pusher

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
-----------------------------------	-----------------------	----------	-------------

VITE_APP_NAME	"\${APP_NAME}"		
VITE_PUSHER_APP_KEY	"\${PUSHER_APP_KEY}"		
VITE_PUSHER_HOST	"\${PUSHER_HOST}"		
VITE_PUSHER_PORT	"\${PUSHER_PORT}"		
VITE_PUSHER_SCHEME	"\${PUSHER_SCHEME}"		
VITE_PUSHER_APP_CLUSTER	"\${PUSHER_APP_CLUSTER}"		

### 6.2.13. Настройки генерации документации Swagger

Наименование переменной окружения	Значение по умолчанию	Описание	Комментарий
L5_FORMAT_TO_USE_FOR_DOCS	"yaml"	Какой формат использовать для вывода	
L5_SWAGGER_OPEN_API_SPEC_VERSION	3.0.0	Версия схемы документации	
L5_SWAGGER_GENERATE_ALWAYS	true	Автоматическая генерация документации	
L5_SWAGGER_CONST_HOST	"\${APP_URL}/api/v1/"	URL документации Swagger	
L5_SWAGGER_GENERATE_YAML_COPY	true	Генерация копии документации в формате YAML	

## 6.3. Сборка и настройка контейнеров Docker

На сервере должна быть установлена программа GNU Make, убедитесь, что она установлена - запустите команду:

```
make -v
```

Вы должны увидеть примерно такой вывод команды:

```
GNU Make 4.3
```

```
Built for x86_64-pc-linux-gnu
```

```
Copyright (C) 1988-2020 Free Software Foundation, Inc.
```

```
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
```

```
This is free software: you are free to change and redistribute it.
```

```
There is NO WARRANTY, to the extent permitted by law.
```

Если программа не установлена, установите ее:

```
sudo apt install make
```

Команда make используется для упрощенного вызова необходимых команд с предварительно настроенными параметрами. Список команд, которые настроены в Makefile, можно увидеть, запустив make без параметров.

Полный список с описаниями приведен ниже.

### 6.3.1. Сборка и запуск контейнеров:

Запускаем сборку и запуск контейнеров:

```
cd /var/data/backend  
make up
```

Устанавливаем библиотеки PHP проекта:

```
make composer-install
```

Далее необходимо сгенерировать ключ шифрования приложения. Если на предыдущих шагах вы не удалили этот ключ, удалите его, отредактировав файл `.env`. Далее, запускаем команду:

```
make generate-key
```

Создаем структуру таблиц БД и заполняем начальными данными:

```
make migrate
```

Добавляем аккаунт администратора (подставьте необходимые параметры):

```
make shell
```

```
php artisan make:superadmin [] [] [email] [password]
```

```
exit
```

Устанавливаем приватный и публичный ключ шифрования и создаем в БД клиенты библиотеки аутентификации Passport:

```
make passport-install
```

## 6.4. Список команд `make`

Наименование команды	Описание	Комментарий
help	Данное описание	
up	Создание и запуск контейнеров	
down	Остановка и удаление контейнеров	
prune	Удаление остановленных контейнеров, образов	
shell	Вход в командную строку контейнера app	
worker-shell	Вход в командную строку контейнера worker app	
db-shell	Вход в командную строку контейнера БД database	
setup	Первоначальное развертывание проекта	
config	Генерация файла <code>.env</code>	
composer-install	Установка пакетов Composer	
generate-key	Установка ключа приложения	

passport-install	Создание ключей шифрования и клиентов авторизации OAuth2	
migrate	Запуск миграций БД	
rollback	Запуск отката миграций БД	
cache-clear	Очистка всех кэшей приложения	