

РУКОВОДСТВО ПО РАЗВЕРТЫВАНИЮ

Клиентского веб-приложения «Гид»

1. Назначение документа

Настоящее руководство предназначено для специалистов, выполняющих установку и первичную настройку приложения на целевые десктопные устройства. Документ содержит подробное описание требований к оборудованию и программному обеспечению, а также последовательность действий по разворачиванию и запуску приложения.

2. Требования к оборудованию и ПО

Перед началом установки убедитесь, что целевая система соответствует следующим требованиям:

Программные требования (зависимости):

- Установленный браузер Google Chrome.
- Для ОС Windows 11: установленная подсистема WSL 2 (Windows Subsystem for Linux).
 - Установленный в среде WSL пакет Docker Engine с инструментом Docker Compose Plugin.
 - Установленный в среде WSL пакет автоматизации сборки Make.

3. Подготовка инфраструктуры (для ОС Windows 11)

Данный раздел содержит пошаговые инструкции по установке всех необходимых компонентов непосредственно в среде WSL 2.

3.1. Установка и настройка WSL 2

Шаг 1. Запустите PowerShell от имени администратора. Для этого нажмите Win + X и выберите "Терминал (Администратор)" или "Windows PowerShell (Администратор)".

Шаг 2. Выполните команду установки WSL. В открывшемся окне введите следующую команду, нажмите Enter и следуйте инструкциям на экране:

```
wsl --install
```

Данная команда автоматически включит необходимые компоненты Windows, установит по умолчанию дистрибутив Ubuntu и настроит WSL версии 2.

Шаг 3. Завершите настройку дистрибутива. Дождитесь распаковки файлов, затем создайте учетную запись пользователя и пароль. Это будет ваш стандартный пользователь Linux в WSL.

3.2. Установка Docker Engine и Docker Compose в WSL

После установки WSL и запуска Ubuntu выполните следующие шаги в терминале Ubuntu.

Шаг 1. Обновите индекс пакетов и установите зависимости:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y apt-transport-https ca-certificates curl software-properties-
common gnupg2
```

Шаг 2. Добавьте официальный GPG-ключ и репозиторий Docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" |
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Шаг 3. Установите Docker Engine и плагины:

```
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

Шаг 4. Добавьте вашего пользователя в группу docker:

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

Шаг 5. Настройте автозапуск Docker:

```
sudo nano /etc/wsl.conf
```

Добавьте в файл следующие строки:

```
[boot]
command = service docker start
```

Сохраните файл (Ctrl+O, затем Ctrl+X в редакторе nano).

Шаг 7. Перезапустите WSL. Закройте все терминалы Ubuntu. Затем в PowerShell (от имени администратора) выполните:

```
wsl --shutdown
```

После этого снова откройте терминал Ubuntu.

Шаг 8. Проверьте установку Docker:

```
docker run hello-world
```

Успешный запуск контейнера подтверждает корректность установки.

3.3. Установка Make в WSL

Установите пакет make с помощью стандартного менеджера пакетов:

```
sudo apt install -y make
```

4. Процесс установки и запуска приложения

Выполните следующие шаги в терминале вашей WSL/Linux среды.

4.1. Размещение дистрибутива

Распакуйте архив с дистрибутивом приложения в целевую директорию.

- Целевой путь: /srv/deploy/
- Убедитесь, что Docker имеет права на чтение и запись в указанную директорию. При необходимости скорректируйте права доступа:

```
sudo chmod -R 755 /srv/deploy
```

4.2. Переход в директорию приложения

Перейдите в каталог с распакованным дистрибутивом:

```
cd /srv/deploy/output
```

4.3. Загрузка образов

Загрузите необходимые Docker-образы в локальное хранилище, используя подготовленную команду Make:

```
make images-load
```

Ожидаемый результат: Система загрузит образы без сообщений об ошибках.

4.4. Запуск контейнеров

Выполните запуск приложения в фоновом режиме:

```
make up
```

Ожидаемый результат: Контейнеры приложения будут созданы и запущены.

4.5. Управление сервисами (мониторинг и остановка)

Для проверки состояния и логов работающих контейнеров используйте команду:

```
make logs
```

Для корректной остановки всех сервисов приложения выполните:

```
make down
```

5. Настройка клиентского устройства

После успешного запуска серверной части приложения необходимо выполнить настройку клиентского рабочего места.

Шаг 1. Откройте браузер Google Chrome.

Шаг 2. Добавить DNS запись “127.0.0.1 app.dev.internal” в локальный файл hosts (C:\Windows\System32\drivers\etc\hosts для windows или /etc/hosts для Linux)

Шаг 3. В адресной строке перейдите по адресу: <https://app.dev.internal>

6. Заключительный этап (Режим киоска)

Для обеспечения автономной работы приложения рекомендуется настроить автоматический запуск браузера Chrome в режиме киоска (kiosk mode).

Создайте ярлык или выполните команду запуска со следующими параметрами:

```
google-chrome --incognito --kiosk --disable-extensions --disable-infobars --disable-translate --disable-pinch --disable-notifications --disable-
```

features=TouchpadOverscrollHistoryNavigation --no-first-run --kiosk
<https://app.dev.internal>

Примечание для ОС Windows: Путь к исполняемому файлу Chrome может отличаться (например, "C:\Program Files\Google\Chrome\Application\chrome.exe"). Убедитесь в корректности пути перед настройкой автозагрузки.

В случае возникновения неполадок в процессе установки или эксплуатации ПО обращайтесь к сотруднику службы тех. поддержки.