

# **GEO TECH PLATFORM**

**Руководство по установке**

## Содержание

<b>Общие сведения .....</b>	<b>3</b>
<b>Описание процедуры установки Платформы .....</b>	<b>3</b>
<b><i>1. Помощь в установке .....</i></b>	<b>3</b>
<b><i>2. Установка открытого ПО .....</i></b>	<b>3</b>
<b><i>3. Установка Платформы .....</i></b>	<b>4</b>

## Общие сведения

Платформа поставляется в виде исходного кода. Это позволяет скомпилировать и установить платформу на большее количество операционных систем, чем если бы она была скомпилирована на какой-то конкретной операционной системе.

Компиляция и установка на целевую ОС автоматизирована. От оператора требуется только выполнить команду. Подготовка целевой ОС не автоматизирована, так как в каждой ОС есть особенности.

## Описание процедуры установки Платформы

### 1. Помощь в установке

Служба поддержки ООО «ТДМ ТЕХ»: телефон +79776290907, email: portal@tdm-tech.ru

### 2. Установка открытого ПО

На целевые сервера необходимо установить следующее ПО, необходимое для функционирования платформы:

- Сервер реляционных баз данных Pangolin
- Сервер колоночных баз данных Cassandra
- Сервер простых структур данных Redis
- Файловый сервер WebDAV
- Распределенную шину данных Kafka

На целевые сервера, на которых будут работать непосредственно компоненты платформы, необходимо установить следующие пакеты (их названия могут незначительно отличаться в разных операционных системах):

- file
- geos
- ImageMagick
- openssl
- libyaml
- libffi
- nginx
- zlib
- libxslt
- libpq
- init-exporter

Также на них необходимо создать пользователей:

- deployer – От его имени на сервере будут разворачиваться релизы компонент платформы. У него должны быть права на выполнение следующих команд:

- `sudo init-exporter` – для создания юнитов `systemd` для запуска компонент
- `sudo systemctl` – для остановки и запуска созданных юнитов
- `sudo service nginx` – для перезагрузки конфигурации веб-сервера
- `sudo -u service` – для выполнения некоторых операций с правами работающих процессов (см. далее)
- `service` – От его имени компоненты будут запущены. У него должны быть ограниченные привилегии для безопасности.

Также необходимо подготовить образ целевой операционной системы для сборки компонентов платформы на нем, при помощи системы контейнеризации (Docker). В этом образе должны быть также установлены вышеперечисленные пакеты, включая их `development` версии. Важно, чтобы образ содержал такие же или максимально близкие версии всех пакетов, как на целевых серверах, потому что на целевых серверах будут запущены исполняемые файлы, скомпилированные в образе. Они будут динамически загружать библиотеки, и если версии будут сильно отличаться, то какие-то вызовы функций могут не работать.

Также в образе должны быть установлены языки программирования, на которых написаны компоненты платформы: Erlang, Elixir, Ruby, Java, NodeJS. Если целевая операционная система не позволяет установить последние версии некоторых языков, можно установить их при помощи ASDF. Сначала установить ASDF, затем ASDF установит нужные версии языков.

В поставку включен файл `Dockerfile`, из которого можно собрать базовый образ на основе Red OS. Файл можно отредактировать под требуемую целевую ОС и собрать нужный образ.

На компьютере оператора, на котором будет выполняться компиляция компонентов платформы, необходимо установить следующее ПО:

- Bash
- dd
- OpenSSH
- Docker
- yq (<https://github.com/mikefarah/yq>)

### ***3. Установка Платформы***

Необходимо создать базы данных в Pangolin:

- `url_resolver`
- `track_resolver`
- `bundle_resolver`
- `parental_control`
- `geosearch`
- `geocache`
- `client_api`
- `Ite_processor`

- subs\_buffer

Таблицы в базах данных создаются автоматически во время первого релиза каждого компонента платформы.

Затем необходимо создать образ для дальнейшей компиляции компонент. Можно использовать вложенный в дистрибутив Dockerfile как шаблон для создания своего Dockerfile для своей целевой ОС. Имя образа необходимо экспортировать в переменную окружения IMAGE. Например:

```
docker build -t geo_platform_build .
```

```
export IMAGE=geo_platform_build
```

Затем необходимо последовательно скомпилировать и выполнить деплой каждой компоненты:

Перейти в каталог компоненты. Например:

```
cd /path/to/geo_platform/geo-search
```

Отредактировать файлы конфигурации в каталогах config и stages. Указать актуальные для вашей инфраструктуры адреса серверов, логины и пароли для подключения к базам данных, и другую информацию.

Выполнить команду для создания и деплоя релиза на целевые сервера:

```
/path/to/geo_platform/bin/perform deploy production
```

Команда выполнит несколько этапов:

- Сформирует сценарий компиляции этого конкретного компонента.
- Выполнит сценарий в контейнере на основе собранного ранее образа из переменной IMAGE. Результатом будет являться архив с релизом.
- Загрузит архив на целевые сервера.
- На каждом сервере распакует архив, создаст недостающие каталоги для логов и временных файлов.
- Создаст юниты systemd и запустит их.

После выполнения этого сценария рекомендуется зайти на каждый сервер и проверить логи всех процессов на наличие ошибок конфигурации. Их можно найти в каталоге /var/log/projects.

Вышеописанный сценарий можно выполнить поэтапно. Программа perform имеет отдельные команды для каждого этапа сборки. Их все можно узнать, выполнив команду help. Например:

```
/path/to/geo_platform/bin/perform help
```

Если компиляция завершается с ошибкой, существует возможность ее отладить в контейнере. Для этого запустите команду debug-build. Например:

```
/path/to/geo_platform/bin/perform debug-build production
```

Вы окажетесь в контейнере для компиляции. Сценарий компиляции будет в файле build.sh. Можно выполнять команды из него по одной, видеть лог выполнения, и понять что идет не так.

После установки всех компонент платформа будет готова к работе.