

«АСК-ПЛАТФОРМА»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ СИСТЕМЫ

**Содержание:**

1. Программные требования.....	3
2. Развертывание сервисов системы с помощью Docker.....	4
Подготовка к установке .....	4
Установка необходимых пакетов на сервер.....	4
Клонирование проекта из GitLAB .....	4
Запуск сервера.....	4
Предварительные настройки .....	4
Развертывание СУБД для сервера навигации.....	4
Развертывание телеметрии .....	5
Развертывание навигации .....	5
Запуск ролей.....	5
Первичная настройка сервера.....	6
Получение пароля для аутентификации в конфигураторе .....	6
3. Запуск и проверка работоспособности Системы .....	8
3.1. Запуск программного обеспечения.....	8
5.2. Проверка работоспособности Системы.....	8

## 1. Программные требования

### Минимальные системные требования для клиентского ПО:

- Процессор 2-ядерный 2,2 ГГц;
- Объем ОЗУ 4 Гб;
- Скорость Интернет-соединения — 256 килобит в секунду.
- Операционная система Linux x64 \ Windows x64 (7 или новее);
- Браузер, один из:
  - Google Chrome (версии не ниже 77)
  - Яндекс браузер (версии не ниже 19)
  - FireFox (версии не ниже 68)
  - Opera (версии не ниже 60)
  - Edge (версии не ниже 44)

### Рекомендованные системные требования для клиентского ПО:

- Процессор 4-ядерный 2,2 ГГц;
- Объем ОЗУ 8 Гб;
- Скорость Интернет-соединения — 1 мегабит в секунду.
- Операционная система Linux x64 \ Windows x64 (7 или новее);
- Браузер, один из:
  - Google Chrome (версии не ниже 77)
  - Яндекс браузер (версии не ниже 19)
  - FireFox (версии не ниже 68)
  - Opera (версии не ниже 60)
  - Edge (версии не ниже 44)

## 2. Развертывание сервисов системы с помощью Docker

### Подготовка к установке

#### Установка необходимых пакетов на сервер

Для дальнейшей работы нам потребуются пакеты **docker**, **docker-compose**, **git** и **make**.

- RockyLinux8.5/CentOS8/FedoraServer

```
dnf install git make
```

- Debian9 и выше

```
apt install git make
```

#### Клонирование проекта из GitLAB

Для того чтобы клонировать проект из **GitLAB** компании необходимо выполнить следующую команду:

```
git clone -b nav_prod https://dev.ask-glonass.ru/gitlab/navigation/navdeploy.git
```

В результате выйдет запрос на ввод учетных данных:

```
[root@template nav]# git clone -b nav_prod https://dev.ask-
glonass.ru/gitlab/navigation/navdeploy.git
Клонирование в «navdeploy»...
Username for 'https://dev.ask-glonass.ru': support
Password for 'https://support@dev.ask-glonass.ru':
```

Дожидаемся процесса окончания клонирования и переходим в папку с проектом.

Учетные данные для клонирования проекта  
**user: support**  
**password:ask42424**

### Запуск сервера

#### Предварительные настройки

Перед началом развертывания необходимо создать файл **.env** следующего содержания. Файл должен располагаться в директории **nav\_deploy**, который создается в результате клонирования проекта из GitLAB.

```
HOST={Адрес или домен сервера}
```

```
LABEL={Метка сервера, нужна для идентификации}
```

**HOST** необходим для traefik. Необходимо указывать именно то, как на сервер будут ходить пользователи. Если обращения идут по доменному имени - необходимо указывать его. Если по IP, то указываем IP  
**LABEL** необходим сугубо для внутреннего использования в АСК с целью идентификации сервера. Желательно, чтобы он был уникальным и отражал информацию о сервере.

### Развертывание СУБД для сервера навигации

Обратите внимание, что при развертывании на одном сервере крайне важен порядок развертывания ролей!

Первой всегда разворачивается роль **database** затем **telemetry** и только потом **navigation**.

Для развертывания роли необходимо находясь в каталоге **nav\_deploy** выполнить команду:

```
make database
```

В результате выполнения команды получим примерно следующий вывод:

```
docker-compose --profile database pull
Pulling postgres ... done
docker-compose --profile database up --no-start
Creating network "navdeploy_default" with the default driver
Creating navigation_postgres ... done
```

### Развертывание телеметрии

Для того, чтобы развернуть сервер телеметрии необходимо выполнить следующую команду:

```
make telemetry
```

В результате выполнения получим примерно следующий вывод:

```
docker-compose --profile telemetry pull
Pulling updater ... done
Pulling telemetry ... done
docker-compose --profile telemetry up --no-start
Creating network "navdeploy_default" with the default driver
Creating navigation_telemetry ... done
Creating navigation_updater ... done
```

### Развертывание навигации

Для того, чтобы развернуть сервер навигации необходимо выполнить следующую команду:

```
make navigation
```

В результате успешного выполнения получаем следующий вывод:

```
docker-compose --profile navigation pull
Pulling proxy ... done
Pulling server ... done
Pulling client ... done
Pulling configurator ... done
Pulling updater ... done
Pulling mongo ... done
docker-compose --profile navigation up --no-start
Creating navigation_client ... done
Creating navigation_proxy ... done
Creating navigation_configurator ... done
Creating navigation_server ... done
Creating navigation_updater ... done
Creating navigation_mongo ... done
```

После этого можно перейти к процессу запуска ролей.

### Запуск ролей

Запуск ролей осуществляется командой:

```
make start
```

После чего мы получим примерно следующий вывод:

```
docker-compose --profile database start
Starting postgres ... done
docker-compose --profile telemetry start
Starting updater ... done
Starting telemetry ... failed
docker-compose --profile navigation start
Starting proxy ... done
Starting server ... done
Starting client ... done
Starting configurator ... done
Starting updater ... done
Starting mongo ... done
```

Обратите внимание, что в данном выводе роль телеметрии на сервер не устанавливалась и потому мы получаем сообщение об ошибке, которое говорит нам о том, что роль телеметрии не запустилась. Это не критично. Такие же ошибки будут появляться в случае, если какая-то другая роль так же не установлена.

## Первичная настройка сервера

### Получение пароля для аутентификации в конфигураторе

После установки ролей и их запуска необходимо в логах сервера навигации найти пароль для аутентификации в конфигураторе. Сообщение выглядит примерно следующим образом:

```
navigation_server | [04:09:37 WRN] *** Server without store!!! To configure use password:
f94b8bae-1a15-4e4d-ae12-a35baa0eeb78
```

Его необходимо скопировать. Далее переходим в конфигуратор.

Важно чтобы на сервер вы заходили именно так, как указали в файле `.env`. В противном случае `traefik` не поймет куда именно проксировать ваши запросы.

После того, как вы ввели в браузере адрес сервера должно открыться окно, в котором необходимо указать пароль, полученный ранее.

Далее мы попадаем в конфигуратор, где нам необходимо выполнить первичную настройку сервера. Необходимо нажать на **шестеренку** в левой части экрана.

Откроются настройки сервера навигации, где нас интересуют в первую очередь разделы **PostgreSQL Data Store** и **Terminal Data Source**.

Отрываем раздел **PostgreSQL Data Store**. Это настройки БД навигации, где необходимо указать данные для подключения к СУБД.

Обратите внимание, если роль разворачивалась на этом же сервере, то выполнять дополнительные настройки не требуется. В случае, когда роль развернута на другом сервере необходимо указать данные для подключения к СУБД. Если роль на другом сервере так же развернута в **docker**, данные для подключения будут следующие:

```
User: postgres
Pass: Nu56aczC95eyF75wCFqJ
```

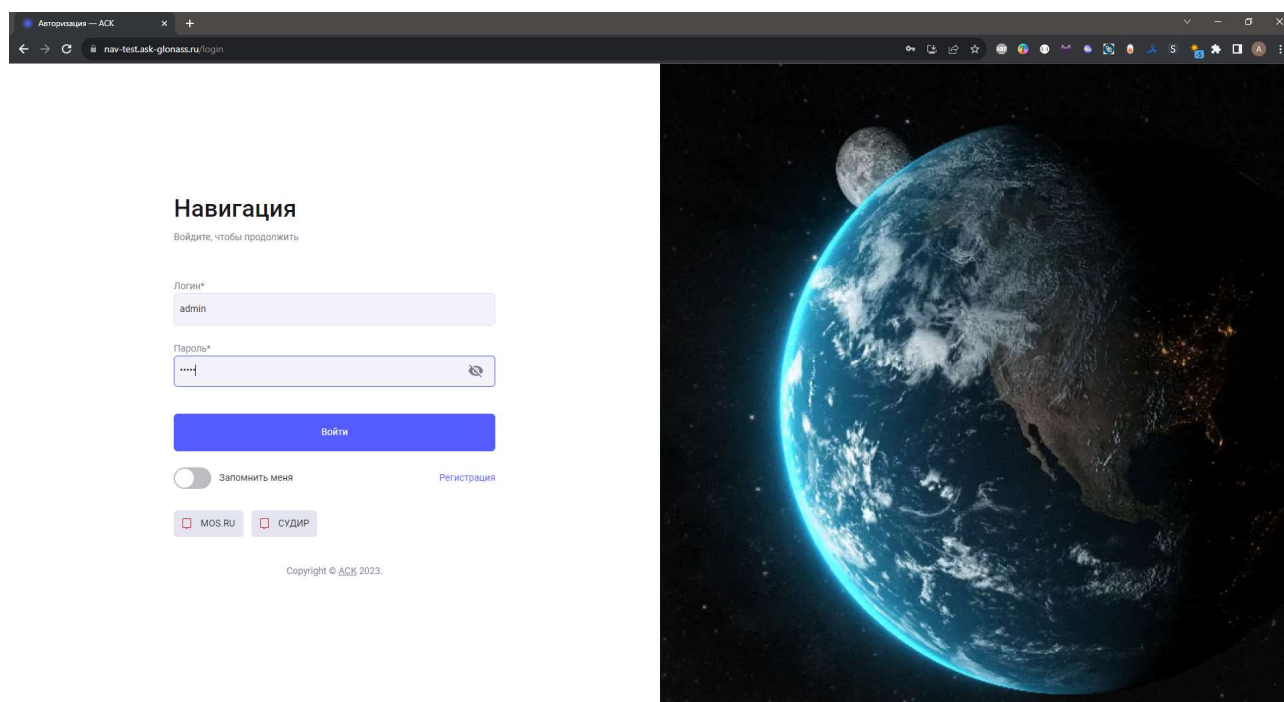
После этого переходим в раздел **Terminal Data Source**. В данном разделе настраивается подключение к серверу телеметрии. Если роль сервера телеметрии была развернут здесь же в **docker**, то настройка не требуется.

### 3. Запуск и проверка работоспособности Системы

#### 3.1. Запуск программного обеспечения

Для начала работы с Программным обеспечением необходимо произвести следующие операции:

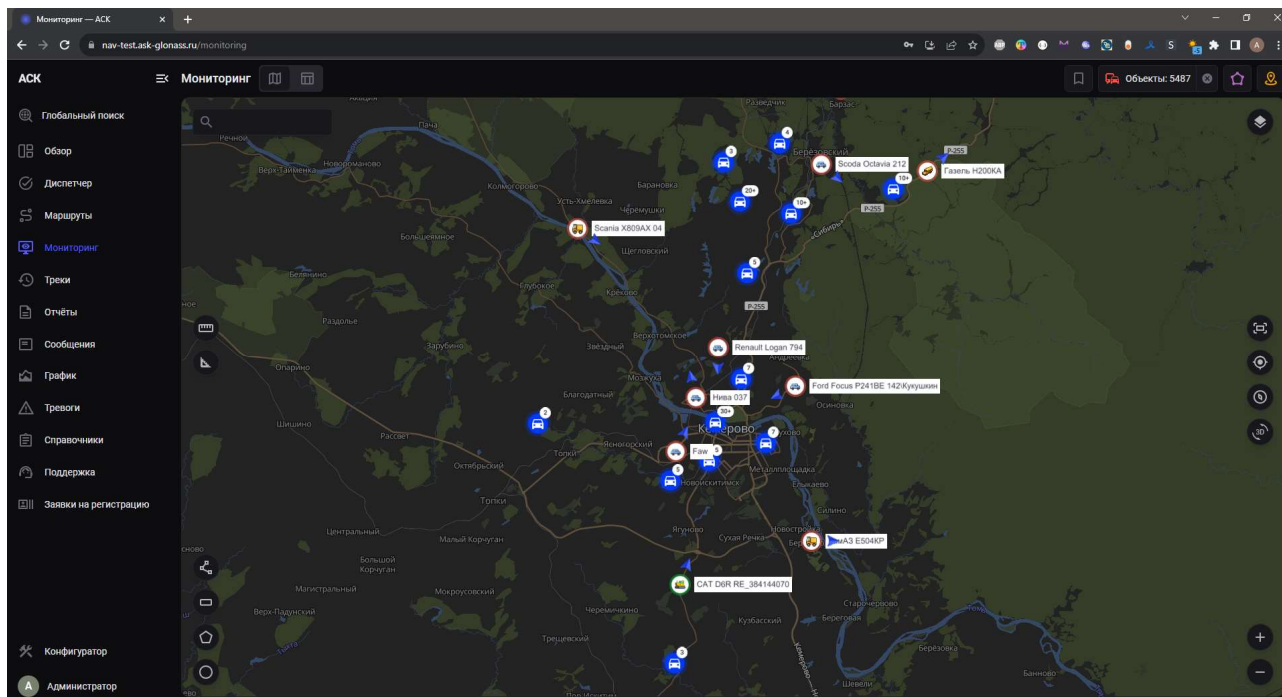
- откройте совместимый браузер;
- перейти на ресурс, где развернут сервер;
- введите логин и пароль в соответствующие поля:
  - логин – admin;
  - пароль – 54731;
- нажмите кнопку «Войти».



#### 5.2. Проверка работоспособности Системы

После авторизации и входа в Программное обеспечение подключаемся к серверу и выбираем инструмент «Мониторинг», далее выбираем технику для отображения в инструменте (Рисунок 1)





(Рисунок 1)

Без единиц техники добавленных и передающих навигационные данные на сервер телеметрии, эксперт может протестировать некоторые функциональные возможности Системы в разделе Справочники.

Полное тестирование Программного обеспечения проводится Заказчиком после установки согласно документу «Описание установки Программного обеспечения» и документу «Программа и методика испытаний», входящим в комплект поставки Программного обеспечения.