

**Инструкция по установке
Программного обеспечения «Информационная система “Безопасность
Дорожного Движения”»**

Оглавление

Оглавление пустое, так как стили абзацев, выбранные для отображения в оглавлении, не используются в документе.

1. Назначение документа

Настоящий документ включает описание по установке Программного обеспечения «Информационная система “Безопасность Дорожного Движения”». После выполнения представленных инструкции следует произвести настройки базы данных.

2. Общие положения

2.1. Наименование и условное обозначение

Наименование: Программное обеспечение «Информационная система “Безопасность Дорожного Движения”»

Условное обозначение: ПО ИС БДД

2.2. Область применения

ПО ИС БДД – уникальная система по сбору и анализу информации о состоянии дорожной инфраструктуры на основе фото и видеоматериалов.

2.3. Термины, определения и сокращения

В настоящем документе использованы и определены следующие термины и сокращения:

<i>Термин/Сокращение</i>	<i>Определение</i>
Nginx	Программное обеспечение с открытым исходным кодом, которое позволяет создавать веб-сервер. Также его используют как почтовый сервер или обратный прокси-сервер.
Docker	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений.
Docker Swarm	Расширенная функция управления кластером демонов Docker.
Docker Compose	Инструмент для определения и запуска многоконтейнерных приложений Docker.

<i>Термин/Сокращение</i>	<i>Определение</i>
Ansible	Система управления конфигурациями, написанная на языке программирования Python, с использованием декларативного языка разметки для описания конфигураций. Применяется для автоматизации настройки и развёртывания программного обеспечения.
Cloud-Init	Программа для инициализации виртуальных машин, обычно применяющаяся в облачных платформах.
Система	Программное обеспечение «Информационная система “Безопасность Дорожного Движения”»

3. Состав программного обеспечения

В состав ПО ИС БДД входит непосредственно исходный код данного приложения, docker контейнеры использующие программное обеспечение в предварительно настроенной конфигурации, инструкции для запуска контейнеров в среде Docker Swarm и Docker Compose.

Кроме того, в процессе установки ПО ИС БДД устанавливается заданная конфигурация сервиса Docker, операционной системы и сети.

Кроме этого, для работы используется веб-сервер Nginx.

4. Программно-аппаратные требования

4.1. Системные требования к серверному оборудованию

ПО ИС БДД устанавливается на операционных системах Ubuntu, начиная с версии Ubuntu 20.04 и до Ubuntu 21.04 с включенной системой управления изоляцией вычислительных ресурсов cgroup v1.

4.2. Системные требования к клиентскому оборудованию

Поскольку подключение пользователей к ПО ИС БДД выполняется с помощью веб-приложения, специальных системных требований к клиентскому оборудованию не предъявляется. На клиентском оборудовании могут использоваться различные версии Windows, Linux и MacOS X на которых установлены последние версии поддерживаемых браузеров:

- Mozilla FireFox<= 120.0,
- Google Chrome<= 122.0.6170.5.

4.3. Аппаратные требования

4.3.1. Серверное оборудование

Минимальные технические требования для установки серверной части ПО ИС БДД:

1. CPU (процессор) – 8 (частота - 3GHz каждое).
2. RAM (оперативная память) – не менее 16 Гб.

3. HDD (жесткий диск) – не менее 100 Гб (случайное чтение 40000 IOPS, случайная запись 20000 IOPS)
4. SSD (твердотельный накопитель) – не менее 300 Гб(случайное чтение 1000 IOPS, случайная запись 500 IOPS)
5. Сетевой адаптер с высокоскоростным подключением к сети, минимум с одним Гигабитным портом.

4.3.2. Клиентское оборудование

Специальных технических требований к клиентскому оборудованию не предъявляется.

5. Установка программного обеспечения

ПО ИС БДД использует микросервисную архитектуру, эта архитектура при которой программное обеспечение, направлено на взаимодействие насколько слабо связанных и легко изменяемых программных компонентов между собой организуя при этом сервис.

Для установки ПО ИС БДД необходимо выполнить приведённые ниже операции.

5.1. Подготовка операционной системы и стенда

Операционная система: ubuntu 20.04

Используемые подсети:

10.136.0.1/16 – для работы ИС БДД в кластере swarm

10.0.0.0/16 – для работы ИС БДД в кластере swarm

172.17.0.1/24 – docker VIP

172.18.0.0/16 – системная сеть кластера swarm

10.255.0.0/16 – диапазон для создания сетей по умолчанию в swarm

Все дальнейшие действия выполняются от пользователя root.

Зависимости установленные в системе: vim, mc, git, jq, python3, python3-pip, iputils-ping, net-tools, wget, curl, ca-certificates, gnupg, lsb-release, apt-transport-https, sshguard

```
apt install -y vim mc git jq python3 python3-pip iputils-ping net-tools wget curl ca-certificates gnupg lsb-release apt-transport-https sshguard
```

Временная зона Europe/Moscow, можно установить используя:

```
timedatectl set-timezone Europe/Moscow && sync
```

Отключение автообновления системы:

```
sed -i 's/1/0/g' /etc/apt/apt.conf.d/20auto-upgrades && sync
```

Можно воспользоваться cloud-init скриптом и версией операционной системы для облаков:

<https://cloud-images.ubuntu.com/>

<https://cloudinit.readthedocs.io/en/latest/topics/examples.html>

cloud-init.cfg:

```
ssh_pwauth: false

users:
  - name: <ИМЯ ПОЛЬЗОВАТЕЛЯ>
    gecos: admin
    sudo: ALL=(ALL) NOPASSWD:ALL
    shell: /bin/bash
    groups: users, admin
    lock_passwd: true
    ssh_authorized_keys:
      - ssh-rsa <УКАЗАТЬ КЛЮЧ ДЛЯ ДОСТУПА К СТЕНДУ >

package_update: true
package_upgrade: false

packages:
  - vim
  - mc
  - git
  - jq
  - python3
  - python3-pip
  - iputils-ping
  - net-tools
  - wget
  - curl
  - ca-certificates
  - gnupg
  - lsb-release
  - apt-transport-https
  - sshguard

runcmd:
  - timedatectl set-timezone Europe/Moscow && sync
  - sed -i 's/1/0/g' /etc/apt/apt.conf.d/20auto-upgrades && sync
```

После проведенных действий стенд готов к настройке.

5.2. Настройка стенда

Для настройки стенда необходима машина на операционной системе linux, которая имеет доступ к стенду по ssh, с предустановленным python3, pip, rsync, net-tools.

Также есть возможность производить настройку локально на стенде на котором распакован проект, для этого необходимо использовать inventory/local.ini

Для запуска продукта необходима сложная конфигурация сервера, которая производится с помощью ansible скриптов состоящих из двух отдельных репозиторий (настройка сервер, конфигурирование работы продукта), находящихся в нашей инфраструктуре и не являются частью продукта, используются только для настройки и являются внутренними инструментами. Для получения проектов необходимо скачать их по ссылке <https://31.13.131.141/iac/Ez1UFpYUzF5wpZIK/stage> и распаковать

```
curl -u <пользователь>:<пароль> --insecure https://31.13.131.141/iac/Ez1UFpYUzF5wpZIK/stage --output stage.zip && unzip stage.zip
```

Перейти в распакованный проект и произвести установку зависимостей для работы ansible.

Произвести установку зависимостей для работы ansible

```
pip install -r requirements.txt
```

Произвести настройку стенда используя ansible

Для локальной установки:

```
ansible-playbook -i local.ini main.yml
```

Для удаленной установки:

```
ansible-playbook --user <имя пользователя на стенде> --private-key <путь до ключа ssh> main.yml
```

Результатом работы ansible-playbook будет установка и настройка следующих компонентов:

- Установка зависимостей для работы кластера swarm;

- Подключение дисков:

 - /var/log – для логов;

 - /mnt – для данных;

- Настройка сетей;

- Настройка прокси сервера для доступа к кластеру и выпуск самоподписанного сертификата для него;

- Настройка доступов к хостовой системе для работы swarm кластера;
- Конфигурирование docker и создание swarm кластера необходимого для запуска ПО ИС БДД;

Дополнительное описание переменных находится в `./docs/ANSIBLE.md`

5.3. Установка и инициализация ПО ИС БДД

Для установки и инициализации ПО ИС БДД необходима машина на операционной системе linux, которая имеет доступ к стенду по ssh, с предустановленным python3, pip, rsync, net-tools

Также есть возможность производить настройку локально на стенде на котором распакован проект, для этого необходимо использовать inventory/local.ini

Для запуска продукта необходима сложная конфигурация сервера, которая производится с помощью ansible скриптов состоящих из двух отдельных репозиторий (настройка сервер, конфигурирование работы продукта), находящихся в нашей инфраструктуре и не являются частью продукта, используются только для настройки и являются внутренними инструментами. Для получения проектов необходимо скачать их по ссылке <https://31.13.131.141/iac/Ez1UFpYUzF5wpZIK/bdd> и распаковать

```
curl -u <пользователь>:<пароль> --insecure https://31.13.131.141/iac/Ez1UFpYUzF5wpZIK/bdd --output citysof.zip &&
unzip bdd-reg.zip
```

Перейти в распакованный проект и произвести установку зависимостей для работы ansible.

```
pip install -r requirements.txt
```

Произвести установку используя ansible

Для локальной установки:

```
ansible-playbook -i local.ini main.yml
```

Для удаленной установки:

```
ansible-playbook --user <имя пользователя на стенде> --private-key <путь до ключа ssh> main.yml
```

Результатом работы ansible-playbook будет установка и настройка следующих компонентов:

- Доставка архивов баз данных на стенд в /mnt/docker_data/volumes;
- Доставка docker-compose.yaml на стенд в /opt/bdd и его запуск;
- Доставка конфигурации прокси сервера для доступа к сервисам ПО ИС БДД в /mnt/infra/nginx/conf.d/vhost/;

Дополнительное описание переменных находится в ./docs/ANSIBLE.md

Если все работает, вы сможете перейти ***hostname:port*** в свой браузер (например, по умолчанию для стенда ***https://31.13.131.141/***) и войти в систему, используя имя пользователя и пароль, полученные вместе с дистрибутивом.