

Информационная система «Безопасная дорога»

Инструкция по установке

Оглавление

1 Назначение документа	3
2 Общие положения	3
2.1 Наименование и условное обозначение	3
2.2 Область применения	3
2.3 Термины, определения и сокращения	3
3 Состав программного обеспечения	3
4 Программно-аппаратные требования	4
4.1 Системные требования к серверному оборудованию	4
4.2 Системные требования к клиентскому оборудованию	4
4.3 Аппаратные требования	4
4.3.1 Серверное оборудование	4
4.3.2 Клиентское оборудование	4
5 Установка программного обеспечения	4
5.1 Ubuntu	5
5.2 Установка и инициализация ИС «Безопасная дорога»	8

1 Назначение документа

Настоящий документ включает описание по установке информационной системы «Безопасная дорога».

2 Общие положения

2.1 Наименование и условное обозначение

Наименование информационной системы: «Безопасная дорога».

2.2 Область применения

ИС «Безопасная дорога» – это оцифровка и аналитика автомобильных дорог с использованием технологий искусственного интеллекта, предиктивной аналитики и уникальных цифровых двойников дорог для повышения безопасности дорожного движения в России.

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.
Система	Информационная система «Безопасная дорога».

3 Состав программного обеспечения

В состав программного обеспечения ИС «Безопасная дорога» 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 ИС «Безопасная дорога», 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

ИС «Безопасная дорога» устанавливается на операционных системах Ubuntu, начиная с версии Ubuntu 14.04 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 ИС «Безопасная дорога» выполняется с помощью веб-приложения, специальных системных требований к клиентскому оборудованию не предъявляется. На клиентском оборудовании могут использоваться различные версии Windows, Linux и Mac OS 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 ИС «Безопасная дорога»:

1. CPU (процессор) – 60 (частота - 3GHz каждое).
2. RAM (оперативная память) – не менее 512 Гб.
3. GPU - Nvidia Tesla T4 (2 шт.)
4. HDD (жесткий диск) – не менее 2 Тб (случайное чтение 40000 IOPS, случайная запись 20000 IOPS)
5. SSD (твердотельный накопитель) – не менее 8 Тб (случайное чтение 1000 IOPS, случайная запись 500 IOPS)
6. 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

ИС «Безопасная дорога» использует микросервисную архитектуру, эта архитектура при которой программное обеспечение, направлено на взаимодействие несколько слабо связанных и легко изменяемых модулей между собой организуя при этом сервис.

Для установки ИС «Безопасная дорога» необходимо выполнить приведённые ниже операции.

5.1 Ubuntu

Следующая команда обеспечит установку необходимых зависимостей для docker и его предварительную настройку:

```
sudo apt-get update
sudo apt-get install -y ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
echo "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu "$(cat /etc/os-release && echo
"$VERSION_CODENAME)" stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin nginx jq mc
sudo groupadd docker
sudo usermod -aG docker $USER
sudo systemctl enable docker.service
sudo systemctl enable containerd.service
echo "{
  \"default-ulimits\": {
    \"nofile\": {
      \"Name\": \"nofile\",
      \"Hard\": 65536,
      \"Soft\": 65536
    },
    \"memlock\": {
      \"Name\": \"memlock\",
      \"Hard\": -1,
      \"Soft\": -1
    }
  },
  \"ipv6\": false,
```

```
\\"data-root\\": \"/var/lib/docker\\",  
\\"storage-driver\\": \\"overlay2\\",  
\\"bip\\": \\"172.17.0.1/24\\"  
}" > /etc/docker/daemon.json  
sudo systemctl restart docker
```

Следующая команда обеспечит установку необходимых зависимостей для docker-compose и его предварительную настройку:

```
sudo curl -L  
"https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname  
-s)-$(uname -m)" -o /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose
```

Следующая команда обеспечит настройку доступа микросервисов к хосту:

```
echo "ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQBAQDt6FLQBOrc8Zdhhh5OfsiZv44EgAOvPmrgY  
dBPcERiy51HMLRbGe1JF42hr2kDlXE9lLJPkqiqOIEc3jJoJS6D7uwmUArpSV1XI7j1059nZye  
X6kqyg7nY8wXD4rscFQsa56EoTulL0LnsFXuBmvjmYAOdrY8Oud85y0mGEI7Kj5VzbfaxM  
RVpe825oNqXVz2wBwmywcUCk/50naqLK4NmP7TzzZmGIRdEqHRQRR6OH0tSajfziEe2DI  
T5Jgpnua5an0XQRyo0Rnl+UA/CgH9lvQye/9ffKscj/EsTfVPyCRge8B7Km2kB1m+Q9OcMYs  
T0YxFfFCsW+2w22ig37L5N" >> /root/.ssh/authorized_keys  
echo "ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQDCqraKVZPeX1xI4pTvmV8zuVISwu10b+DD  
GTIH14I2hjM1w7kIBJx0xkoaYwG1jgT+cZ/2iTCfsDr0buH1ArwuWDiQAznrVTAIjVjRASqvO  
AQVHSjNRLuasANMDaYeouCWL1abgclwxJTdmVW9a8gFj1SEuTCcOpXXmgey/E5r6MAT  
uOIYldv99WCr81YsxTYmBiiBDNSLs7ODpNR/U9raPE2Rgss2U2VJDP7vOBqbqEdN5voiWi  
5IlyMoLzjgOXyF6MeMko+E1AGtSotSX0PNkQrcNqCSItSxfODp4FTpl0cGFScrqq3xqRaK1  
YPZX3g6KIlyAiiAb34Ll/mVsDtSh+PsgojCVJiVU+kw+i7Gn9wyC1W6PP+vmpcnn56cKOHR  
dTukI3c6pfy5jA2RvY9NkBroPYcyhdb0L6DjetkLCKHGFokltr0YBmUDe7f3WBo5EsxqJzEN2  
aONsQLwRUojfOsdke4Xo75R3kRnGQAiOLJGC4z6LBqXfIKYjFgj8io0tkFJenDV5j69AvqbV  
X+VlheFZ9Q+K7MKlzEezNIImtyuRVkzPlf/pwTPoqwwEieLr4NZRAx6MZBYLWq+vNgU2e  
Y/RgkIj+sSFwn5Oms2LRLaE1xL8+3JJeTP5ME49sWoFDIFS1dlUXNGTUrNdcJvg/Km0LrxN  
YGOOXEsKuPglQ==" >> /root/.ssh/authorized_keys
```

Следующая команда обеспечит предварительную настройку системы и сети:

```
echo "net.netfilter.nf_conntrack_max=262144" >> /etc/sysctl.conf
sysctl -p
default_interface=$(ip route show default | awk '/default/ { print $5 }')
default_ip=$(ip addr show ${default_interface} | grep 'inet ' | awk '{print $2}' | awk
'{split($0,ip,"/"); print ip[1]}')
cat <<EOF > /etc/netplan/01-fda-config.yaml
network:
  ethernets:
    $default_interface:
      routes:
        - to: 10.136.0.0/16
          via: $default_ip
      nameservers:
        addresses:
          - 77.88.8.88

  vlans:
    $default_interface.136:
      id: 136
      link: $default_interface
      dhcp4: false
      dhcp6: false
      addresses:
        - 10.136.0.1/16

  version: 2
EOF
netplan apply
```

Следующая команда обеспечит предварительную настройку Docker Swarm:

```
docker network create --opt com.docker.network.bridge.enable_icc=false --subnet=172.18.0.0/16
docker_gwbridge
```

```
default_interface=$(ip route show default | awk '/default/ { print $5 }')
docker swarm init --default-addr-pool 10.255.0.0/16 --advertise-addr $default_interface
```

5.2 Установка и инициализация ИС «Безопасная дорога»

Распакуйте полученный дистрибутив в /opt/safe_road:

```
tar -xvf safe_road.tar -C /opt/
```

Загрузите образы docker контейнеров в локальное хранилище:

```
images_name=$(ls -d /opt/safe_road/images/*); for image_name in ${images_name[@]}; do
docker load --input ${image_name}; done
```

Распакуйте файлы для инициализации баз данных:

```
databases=$(ls -d /opt/safe_road/db/tar/*); for db in ${databases[@]}; do tar -xvzf ${db} -C / ;
done
```

Установите Nginx и настройте его:

```
sudo apt update && sudo apt install nginx -y
cp -r /opt/safe_road/nginx/* /etc/nginx/
service nginx restart
```

Завершите установку, выполнив следующие команды:

```
docker stack deploy --prune -c /opt/safe_road/docker-compose/docker-compose.safe-road.yml
safe-road
sleep 10
MASKS=("10.136.0.0/16" "172.16.0.0/24")
for i in "${MASKS[@]}; do
    HOST_REG_IP=$(ip route | grep $i | grep -Eo 'src \S+' | grep -oE
'\b([0-9]{1,3}\.){3}[0-9]{1,3}\b')
    if [[ $HOST_REG_IP =~ ^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$ ]]; then
        break
    fi
done
if [ -z "$HOST_REG_IP" ]; then
    HOST_REG_IP=$(ip route get 1 | awk '{print $NF;exit}')
fi
CONSUL_8500_PORT=8500
```

```
CONSUL_ID=$(docker stack ps -f "desired-state=running" fda | grep consul_server | awk '{print $1}')
for cons in ${CONSUL_ID}; do
    CONSUL_IP=$(docker inspect $cons 2>/dev/null | jq -r '[][.NodeID]' | xargs docker node
inspect 2>/dev/null | jq -r '[][.Status.Addr]')
    curl -X PUT -d '{"id":"swarm-leader", "name":"swarm-leader",
"address":"$HOST_REG_IP", "port":2377, "enableTagOverride": false}'
http://$CONSUL_IP:$CONSUL_8500_PORT/v1/agent/service/register
done
```

Если все работает, вы сможете перейти **hostname:port** в свой браузер (например, локально по умолчанию **localhost:80**) и войти в систему, используя имя пользователя и пароль, полученные вместе с дистрибутивом.