

# ИНСТРУКЦИЯ УСТАНОВКИ ВЕБ ПРИЛОЖЕНИЯ МДПС

## Система

SUSE Linux Enterprise Server 12 (x86\_64)

### Установка базы данных

Установить последнюю версию postgresql можно с помощью пакетного менеджера zypper, выполнив команду:

```
sudo zypper in postgresql10 postgresql10-contrib postgresql10-server
```

Затем необходимо зайти в систему под пользователем postgres и поменять пароль на желаемый.

```
sudo su - postgres
```

```
psql
```

```
\password postgres
```

После данной команды вводится пароль. Чтобы продолжить установку, необходимо выйти из консоли psql и вернуться к текущему пользователю.

```
\q
```

```
exit
```

## Установка Consul (service discovery и key-value хранилище)

Необходимо загрузить последнюю версию Consul (<https://www.consul.io/downloads.html>) для системы Linux 64-bit. Затем распаковать в директорию `/usr/local/bin/` и дать права на исполнение

```
sudo chmod +x /usr/local/bin/consul
```

В директории `/opt/` нужно создать директорию `/consul/` и два файла в этой директории.

### 1) Файл `start.sh`

Содержимое:

```
#!/bin/bash
eval "consul agent -config-file="/opt/consul/config.json"
```

### 2) Файл `config.json`

Содержимое:

```
{
  "bootstrap": true,
  "server": true,
  "ui": true,
  "log_level": "INFO",
  "enable_syslog": true,
  "addresses": {
    "http": "0.0.0.0"
  },
  "data_dir": "/opt/consul/data"
}
```

Для автозапуска нужно настроить сервис. Для этого необходимо создать файл **consul.service** в директории `/etc/systemd/system/`

Содержимое файла:

```
[Unit]
Description=Consul service
[Service]
ExecStart=/bin/bash /opt/consul/start.sh
SuccessExitStatus=143
TimeoutStopSec=10
Restart=on-failure
RestartSec=5
[Install]
WantedBy=multi-user.target
```

Затем выполнить команду:

```
systemctl enable consul
```

Запустить сервис можно командой:

```
systemctl start consul
```

## Установка front-end части веб-приложения

Для работы front-end части необходимо установить последнюю LTS версию Nodejs. Загрузить можно по ссылке (<https://nodejs.org/en/download/>) для Linux 64-bit и распаковать в директорию /opt/. Распакованной директории установить права 755. Затем отредактировать файлы ~/.profile и ~/.bashrc , добавив туда строки (предполагается, что /node-v10.1.0-linux-x64/ - директория, в которую был распакован архив):

```
# NodeJS
export NODEJS_HOME=/opt/node-v10.1.0-linux-x64/bin
export PATH=$NODEJS_HOME:$PATH
```

Затем нужно выполнить команды:

```
. ~/.profile
```

```
sudo ln -s /opt/node-v10.1.0-linux-x64/bin/node /usr/bin/node
```

```
sudo ln -s /opt/node-v10.1.0-linux-x64/bin/npm /usr/bin/npm
```

```
sudo ln -s /opt/node-v10.1.0-linux-x64/bin/npx /usr/bin/npx
```

Для запуска front-end частей приложений требуется выполнить установку npm-модуля - pm2.

Установка при наличии доступа к сети интернет выполняется одной командой:

```
npm i -g pm2
```

Установка без доступа к сети интернет выполняется следующим образом:

На локальной машине с ОС на ядре Linux с установленным nodejs необходимо выполнить команду:

```
npm i -g npm-bundle
```

Затем можно создать архив с модулем pm2 с помощью команды:

```
npm-bundle pm2
```

Созданный архив нужно скопировать на сервер, где необходимо выполнить команду (предполагается, что название архива - pm2-3.5.1.tgz):

```
npm i -g pm2-3.5.1.tgz
```

После установки pm2, в директории /opt/ нужно создать директорию /pm2/ и два файла в этой директории.

### 1) Файл **start.sh**

Содержимое:

```
#!/bin/bash
cd "$(dirname "$0")"
pm2 start config.yml --env=production
```

### 2) Файл **config.yml**

Содержимое (предполагается, что 10.60.60.160 – ip адрес данного сервера):

```
- name: auth
  script: /opt/mdps/auth-front/node_modules/nuxt-start/bin/nuxt-start.js
  cwd: /opt/mdps/auth-front
  max_memory_restart: "250M"
  args: "--max-old-space-size=200"
  env_production:
    PORT: 7011
    HOST: "10.60.60.160"
- name: mdps
  script: /opt/mdps/mdps-front/node_modules/nuxt-start/bin/nuxt-start.js
  cwd: /opt/mdps/mdps-front
  max_memory_restart: "250M"
  args: "--max-old-space-size=200"
  env_production:
    PORT: 7012
    HOST: "10.60.60.160"
```

Для автозапуска нужно настроить сервис. Для этого необходимо создать файл **pm2.service** в директории /etc/systemd/system/

Содержимое файла:

```
[Unit]
Description=PM2 service
[Service]
ExecStart=/bin/bash /opt/pm2/start.sh
SuccessExitStatus=143
TimeoutStopSec=10
Restart=on-failure
RestartSec=5
[Install]
WantedBy=multi-user.target
```

**Затем выполнить команду:**

```
systemctl enable pm2
```

**Запустить сервис можно командой:**

```
systemctl start pm2
```

Перед запуском необходимо загрузить архив с веб-приложениями и распаковать в директорию /opt/mdps/

## Установка back-end части веб-приложения

Для работы back-end части требуется 8 версия Java. Для установки Java нужно выполнить команду:

```
sudo apt install openjdk-8-jre
```

Затем нужно создать три директории (/mdps/mdps-server , /mdps/auth-server , /mdps/api-gateway) в /opt/. В директории необходимо поместить jar файлы веб-приложению. В каждой директории создать по файлу **start.sh**

Содержимое файла в директории mdps-server:

```
#!/bin/bash
clear
cd /opt/mdps/mdps-server
java -jar `ls mdps*.jar`
```

Содержимое файла в директории auth-server:

```
#!/bin/bash
clear
cd /opt/mdps/auth-server
java -jar `ls auth*.jar`
```

Содержимое файла в директории api-gateway:

```
#!/bin/bash
clear
cd /opt/mdps/api-gateway
java -jar `ls api-gateway*.jar`
```

Для автозапуска нужно настроить два сервиса. Для этого необходимо создать файлы **mdps-server.service** , **auth-server.service** , **api-gateway.service** в директории /etc/systemd/system/

Содержимое файлов:

```
[Unit]
Description=MDPS service
After=postgresql.service
After=consul.service
Requires=postgresql.service
Requires=consul.service
[Service]
ExecStart=/bin/bash /opt/mdps/mdps-server/start.sh
SuccessExitStatus=143
TimeoutStopSec=10
Restart=on-failure
RestartSec=5
[Install]
WantedBy=multi-user.target
```

```
[Unit]
Description=Auth service
After=postgresql.service
After=consul.service
Requires=postgresql.service
Requires=consul.service
[Service]
ExecStart=/bin/bash /opt/mdps/auth-server/start.sh
SuccessExitStatus=143
TimeoutStopSec=10
Restart=on-failure
RestartSec=5
[Install]
WantedBy=multi-user.target
```

```
[Unit]
Description=Api gateway
After=postgresql.service
After=consul.service
Requires=postgresql.service
Requires=consul.service
[Service]
ExecStart=/bin/bash /opt/mdps/api-gateway/start.sh
SuccessExitStatus=143
TimeoutStopSec=10
Restart=on-failure
RestartSec=5
[Install]
WantedBy=multi-user.target
```

Затем выполнить команды:

```
systemctl enable mdps-server
```

```
systemctl enable auth-server
```

```
systemctl enable api-gateway
```

Запустить сервисы можно командой:

```
systemctl start mdps-server
```

```
systemctl start auth-server
```

```
systemctl start api-gateway
```