

Fundamento
Руководство по инсталляции

ООО «СЛ Софт»
2025

Оглавление

Введение	3
1. Требования к аппаратным средствам	4
2. Установка и запуск системы	4
2.1. Установка инструмента контейнеризации docker.....	4
2.2. Загрузка образов контейнеров.....	5
2.3. Загрузка данных в систему.....	Ошибка! Закладка не определена.
2.4. Запуск решения.....	5
2.5. Остановка решения.....	7
2.6. Создание бэкапа	Ошибка! Закладка не определена.
2.7. Обновление версии проекта с бэкапа платформы.....	Ошибка! Закладка не определена.

Введение

«Fundamento» (далее – Платформа) является единой цифровой платформой, включающей набор компонентов, обеспечивающей быструю разработку, управление и масштабирование проектов IoT, компьютерного зрения.

Платформа позволяет получать и визуализировать данные с подключенных устройств в реальном времени, а также создавать пользовательские дашборды и отчеты для анализа и принятия важных бизнес решений.

Настоящий документ представляет собой руководство по инсталляции Платформы. Данное руководство предназначено для администраторов Платформы и описывает действия по её инсталляции на production-сервер.

Перед началом работы с Платформой необходимо изучить данное Руководство.

1. Требования к аппаратным средствам

Production-сервер, предназначенный для установки Платформы, должен соответствовать следующим минимальным требованиям:

- CPU: 4 ядра
- RAM: 16 Gb
- HDD: 200 Gb и более

Рекомендуемые системные требования для Production-сервера для установки Платформы:

- CPU: 16 ядер
- RAM: 64 Gb
- HDD: 500 Gb и более

Рекомендуемая операционная система: Ubuntu Server 22.04 LTS x64.

2. Установка и запуск системы

2.1. Установка инструмента контейнеризации docker

Для запуска платформы необходимо наличие установленного сервиса контейнеризации на виртуальной машине (VM).

Для установки данного сервиса необходимо выполнить следующие действия:

Необходимо с помощью консоли отправить папку deb пакетов на подготовленную виртуальную машину:

```
$ scp -r ./ docker_deb <user>@<ip address>:/home/<user>/docker_deb
```

Далее подключиться по ssh к VM, перейти в директорию с пакетами и установить данные пакеты:

```
$ cd /home/<user>/docker_deb
$ dpkg -i ./containerd.io_1.6.25-1_amd64.deb
$ dpkg -i ./docker-ce_24.0.7.deb
$ dpkg -i ./docker-ce-cli_24.0.7.deb
$ dpkg -i ./docker-buildx-plugin_0.11.2.deb
$ dpkg -i ./docker-compose-plugin_2.21.0.deb
$ dpkg -i ./docker-scan-plugin_0.23.0.deb
```

Далее необходимо добавить сервис в systemd:

```
$ sudo systemctl enable docker
```

Для удобства работы с сервисом можно добавить разрешения для выполнения команд для текущего пользователя:

```
$ sudo usermod -aG docker ${USER}
```

Добавьте в файл `/etc/docker/daemon.json` следующее содержимое:

```
{  
  "insecure-registries" : [ "image.git.softline.digital:1060" ]  
}
```

И затем перезапустите докер.

```
$ sudo systemctl daemon-reload  
$ sudo systemctl restart docker
```

2.2. Загрузка образов контейнеров

Необходимо подключиться по ssh к VM. Для этого необходимо открыть powershell и в открывшейся консоли ввести следующую команду:

```
$ ssh <user>@<ip address>
```

Далее, выполнить команды:

```
$ docker login image.git.softline.digital:1060  
$ docker pull image.git.softline.digital:1060/fundamento-platform/fundamento/tb-  
cassandra:0.6.2-SNAPSHOT  
$ docker pull zookeeper:3.5  
$ docker pull wurstmeister/kafka:latest  
$ docker pull bitnami/redis:7.0
```

Для получения доступа к репозиторию использовать имя пользователя и токен:

```
client  
glpat-fWywAx7xbpzchpSnsV2U
```

2.3. Запуск решения

При первом запуске системы необходимо отправить файл `docker-compose.yml`:

```
$ scp docker-compose.yml <user>@<ip address>:/home/<user>/application/docker-compose.yml
```

Либо создать его:

```
#docker-compose.yml
services:
  zookeeper:
    restart: always
    image: "zookeeper:3.5"
    ports:
      - "2181:2181"
    environment:
      ZOO_MY_ID: 1
      ZOO_SERVERS: server.1=zookeeper:2888:3888;zookeeper:2181
  kafka:
    restart: always
    image: wurstmeister/kafka
    depends_on:
      - zookeeper
    ports:
      - "9092:9092"
    environment:
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_LISTENERS: INSIDE://:9093,OUTSIDE://:9092
      KAFKA_ADVERTISED_LISTENERS: INSIDE://:9093,OUTSIDE://kafka:9092
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INSIDE:PLAINTEXT,OUTSIDE:PLAINTEXT
      KAFKA_INTER_BROKER_LISTENER_NAME: INSIDE
      KAFKA_MESSAGE_MAX_BYTES: 200000000
      KAFKA_MAX_REQUEST_SIZE: 200000000
      KAFKA_MAX_PARTITION_FETCH_BYTES: 200000000
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
  platform:
    restart: always
    image: "image.git.softline.digital:1060/fundamento-platform/fundamento/tb-
cassandra:0.6.2-SNAPSHOT"
    depends_on:
      - kafka
    ports:
      - "9090:9090"
      - "1883:1883"
      - "7070:7070"
      - "5683-5688:5683-5688/udp"
    environment:
      TB_QUEUE_TYPE: kafka
      TB_KAFKA_SERVERS: kafka:9092
      AUDIT_LOG_ENABLED: true
      TB_SERVER_WS_DYNAMIC_PAGE_LINK_REFRESH_INTERVAL_SEC: 1
      TB_SERVER_WS_DYNAMIC_PAGE_LINK_REFRESH_POOL_SIZE: 10
      TB_SERVER_WS_DYNAMIC_PAGE_LINK_MAX_PER_USER: 50
      TB_KAFKA_MAX_REQUEST_SIZE: 6000000
      JS_MAX_TOTAL_ARGS_SIZE: 60000000
      JS_MAX_RESULT_SIZE: 60000000
      JS_MAX_SCRIPT_BODY_SIZE: 60000000
      TBEL_MAX_TOTAL_ARGS_SIZE: 60000000
      TBEL_MAX_RESULT_SIZE: 60000000
      TBEL_MAX_SCRIPT_BODY_SIZE: 60000000
      CACHE_SPECS_FILES_TTL: 1440
      CACHE_SPECS_ENTITY_COUNT_MAX_SIZE: 30000000
      CACHE_TYPE: redis
      REDIS_HOST: redis
      REDIS_PORT: 6379
    volumes:
      - ./mytb-data:/data
      - ./mytb-logs:/var/log/thingsboard
  redis:
    restart: always
    image: bitnami/redis:7.0
    environment:
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
```

```
ALLOW_EMPTY_PASSWORD: "yes"  
ports:  
- '6379:6379'
```

Далее подключиться по ssh к VM. Для этого необходимо открыть powershell в открывшейся консоли ввести следующую команду:

```
$ ssh <user>@<ip address>
```

После подключения и выполнить следующие команды:

```
$ cd /home/<user>/application  
$ mkdir mytb-data  
$ mkdir mytb-logs  
$ sudo chown 799:799 -R mytb-data  
$ sudo chown 799:799 -R mytb-logs  
$ docker compose up -d
```

2.4. Остановка решения

Для остановки решения необходимо подключиться с VM по ssh. Для этого необходимо открыть powershell в открывшейся консоли ввести следующую команду:

```
$ ssh <user>@<ip address>
```

и выполнить следующие команды:

```
$ cd /home/<user>/application/  
$ docker compose down
```