

Инструкция по установке системы Дорис Контроль

1. Базовое системное программное обеспечение

Сервера приложений:

- Операционная система Ubuntu версия 18.04 или более поздняя
- Система контейнеризации Docker 19.03 или более поздняя
- Сборщик контейнеров Docker-compose 1.17 или более поздняя

Сервера баз данных:

- Операционная система FreeBSD 12.1 или более поздняя
- СУБД Postgres 12.0 или более поздняя
- Расширение PostGIS 3.0 или более поздняя
- Расширение postgresql-contrib версии соответствующей Postgres
- СУБД Redis 5.0 или более поздняя

Рабочие станции необходимо следующее программное обеспечение:

- браузер Google Chrome 51.0 и выше или Mozilla FireFox версии 14 и выше;

2. Техническое обеспечение

Используемые для эксплуатации системы технические средства (персональные компьютеры, сервера, системы виртуализации, периферийные устройства) должны быть совместимы между собой и поддерживать сетевой протокол TCP/IP.

Для работы системы используются компьютеры 64-разрядной архитектуры AMD/Intel:

- с операционной системой FreeBSD для серверов баз данных
- с операционной системой Ubuntu для серверов приложений
- операционной системой Windows/Linux/MacOS для рабочих станций

Минимальные технические характеристики клиентских компьютеров:

- тактовая частота процессора не менее 2 GHz;
- оперативная память не менее 4 GB;
- свободное дисковое пространство не менее 20 GB.

Рекомендуется использовать клиентские компьютеры с объемом оперативной памяти (RAM) от 8 GB.

Минимальные технические характеристики серверного оборудования:

- система виртуализации vmware ESXi 6.5 или Microsoft Hyper-V Server 2019
- каждый сервер баз данных
 - виртуальных ядер не менее 8
 - оперативная память не менее 32 GB
 - дисковое пространство не менее 240 GB
- каждый сервер приложений
 - виртуальных ядер не менее 4
 - оперативная память не менее 12 GB
 - дисковое пространство не менее 160 GB
- каждый сервер архива событий
 - виртуальных ядер не менее 4
 - оперативная память не менее 8 GB

- дисковое пространство не менее 2048 GB

Рекомендуется иметь полутора кратный запас количества ядер и размера оперативной памяти для серверов.

3. Установка сервера баз данных

- Установка СУБД Postgres, выполняется от пользователя root (или через команду sudo)
 - `pkg install postgresql12-server`
 - `pkg install redis`
 - `vi /usr/local/db/data12/pg_hba.conf`
указать IP-адреса клиента - серверов приложений, пользователей и режим авторизации
 - `vi /usr/local/db/data12/postgresql.conf`
`max_connections = 300`
`listen_addresses = '*'`
 - `vi /etc/rc.conf`
`redis_enable="YES"`
`postgresql_class="postgres"`
`postgresql_enable="YES"`
 - `initdb`
 - `service postgresql start`
 - `service redis start`

4. Установка сервера приложений

4.1. Установить Docker по официальной инструкции <https://docs.docker.com/engine/install/ubuntu/>

4.2. Установить docker-compose

```
sudo apt-get install docker-compose
```

4.3. создать директорию для настройки сервера приложения

```
mkdir docker && cd docker
```

4.4. Создать файл docker-compose.yml

```
nano docker-compose.yml
```

```
version: '3'
```

```
services:
```

```
front:
```

```
image: you-registry.local/neoteh/rtcc.website2:${PROJECT}-latest
```

```
restart: unless-stopped
```

```
tty: true
```

```
ports:
```

```
- 80:80
```

```
networks:
```

```
- app-bridge
```

```
back-main:
```

```
image: you-registry.local/its-backend/main-${PROJECT}:latest
```

```
restart: unless-stopped
```

```
tty: true
```

```
ports:
```

```
- 10100:8080
```

```
- 6001:6001
```

```
volumes:
```

```
- ${DATA_PATH_STORAGE}/app-main/storage:/var/www/storage
```

```
links:
```

```
- rabbitmq
networks:
- app-bridge
```

back-video:

```
image: you-registry.local/its-backend/video- $\{\text{PROJECT}\}$ :latest
restart: unless-stopped
tty: true
ports:
- 10084:8080
volumes:
-  $\{\text{DATA\_PATH\_STORAGE}\}$ /app-video/storage:/var/www/storage
links:
- rabbitmq
networks:
- app-bridge
```

rabbitmq:

```
image: rabbitmq:3.8.9-management
volumes:
- /etc/localtime:/etc/localtime:ro
-  $\{\text{DATA\_PATH\_HOST}\}$ /rabbitmq:/var/lib/rabbitmq
ports:
- " $\{\text{RABBITMQ\_NODE\_HOST\_PORT}\}$ :5672"
- " $\{\text{RABBITMQ\_MANAGEMENT\_HTTP\_HOST\_PORT}\}$ :15672"
- " $\{\text{RABBITMQ\_MANAGEMENT\_HTTPS\_HOST\_PORT}\}$ :15671"
privileged: true
environment:
- RABBITMQ_DEFAULT_USER= $\{\text{RABBITMQ\_DEFAULT\_USER}\}$ 
- RABBITMQ_DEFAULT_PASS= $\{\text{RABBITMQ\_DEFAULT\_PASS}\}$ 
networks:
- app-bridge
```

clickhouse:

```
container_name: clickhouse
image: yandex/clickhouse-server:20
ports:
- 8123:8123
volumes:
- ' $\{\text{DATA\_PATH\_HOST}\}$ /clickhouse:/var/lib/clickhouse'
privileged: true
restart: unless-stopped
networks:
- app-bridge
```

4.5. Создать файл настройки .env

```
nano .env
DATA_PATH_HOST=./data-docker
DATA_PATH_STORAGE=./storage-all
```

VOLUMES_DRIVER=local

PROJECT=YouNameProject

RABBITMQ

RABBITMQ_ERLANG_COOKIE=its

RABBITMQ_NODE_HOST_PORT=5672

RABBITMQ_MANAGEMENT_HTTP_HOST_PORT=15672

RABBITMQ_MANAGEMENT_HTTPS_HOST_PORT=15671

RABBITMQ_DEFAULT_USER=its

RABBITMQ_DEFAULT_PASS=password

CONFIG_PATH=./config

4.6. создать вспомогательные директории для каждого микросервиса

```
mkdir -p storage-all/app-main/storage/app/
```

```
mkdir -p storage-all/app-main/storage/logs/
```

```
mkdir -p storage-all/app-main/storage/framework/cache/data/
```

```
mkdir -p storage-all/app-main/storage/framework/sessions/
```

```
mkdir -p storage-all/app-main/storage/framework/testing/
```

```
mkdir -p storage-all/app-main/storage/framework/views/
```

```
mkdir -p storage-all/app-video/storage/app/
```

```
mkdir -p storage-all/app-video/storage/logs/
```

```
mkdir -p storage-all/app-video/storage/framework/cache/data/
```

```
mkdir -p storage-all/app-video/storage/framework/sessions/
```

```
mkdir -p storage-all/app-video/storage/framework/testing/
```

```
mkdir -p storage-all/app-video/storage/framework/views/
```

4.7. Запустить сервер приложений

```
docker-compose up -d
```