Системные требования для On-Premise

инсталляции UX Rocket

Актуальность документа проверена 22.03.24

Оглавление

[Введение 1](#_Toc162037681)

[О продукте 1](#_Toc162037682)

[Основные компоненты UX Rocket 2](#_Toc162037683)

[Требования к системе 2](#_Toc162037684)

[Рекомендуемые системные требования 3](#_Toc162037685)

[Сводная таблица по развернутым сервисам (сетевые порты) 6](#_Toc162037686)

[Используемые версии ПО 7](#_Toc162037687)

[Общие требования к публикации 8](#_Toc162037688)

## Введение

Данный документ содержит информацию о рекомендуемых системных требования, версии используемого ПО, рекомендации по публикации, планированию и развертыванию серверного решения UX Rocket.

 [Программное обеспечение «UX Rocket»](https://uxrocket.ru/null) — это платформа для сбора данных и анализа клиентского поведения на сайте и в мобильном приложении, A/B и MVT тестирования гипотез, персонализации контента, формирования предложений и коммуникаций на основе анализа профиля клиента в реальном времени.

ПО «UX Rocket» может быть развернуто как на физических серверах, так и полностью поддерживается работа в виртуальном окружении и система контейнеризации docker, docker-compose.

## О продукте

Программное обеспечение «UX Rocket» представляет собой веб-приложение, созданное на основе .NET с использованием фреймворка Blazor.

Приложение использует базы данных PostgreSQL, ClickHouse, брокер сообщений Kafka и файлы данных на дисковом хранилище, взаимодействует с пользователями, посещающими сайт и мобильное приложение, взаимодействует с пользователями личного кабинета.

## Основные компоненты UX Rocket

|  |  |  |
| --- | --- | --- |
| Группа | Docker-сервисы | Примечание |
| Core | ux-api-cpaapi – API для работы личного кабинета и админки (CPA.API) | Все компоненты группы Core являются обязательными  |
| ux-front-client – кабинет пользователя (CPA.Client.Web) |
| ux-front-admin  – кабинет администратора (CPA.Admin.Web) |
| ux-api-hostedservices – Сервис фоновых заданий (CPA.HostedServices) |
| ux-ftp – Сервис ftp для загрузки данных |
| Site | ux-api-getsitescripts – Cервис инициализации скрипта на сайте и для А/В тестирования на сайте (GetSiteScripts.API) |  |
| ux-api-saverawdataapi – Сервис сбора данных для сайта (SaveRawData.API) |
| uxnginx - JS-скрипт сбора данных |
| MobileAPI | ux-api-mobileapi – Сервис сбора данных и тестирования мобильного приложения (Mobile.API) |  |
| Integration | ux-api-partnerdataimport – Сервис импорта данных из сторонних систем (CPA.PartnerDataImport) | Дополнительный компонент |
| Database | Clickhouse (база данных)  | Все компоненты группы Database являются обязательными  |
| PostgreSQL (база данных)  |
| Message broker | Apache Kafka | Все компоненты группы Message broker являются обязательными  |

## Требования к системе

Для установки программного продукта UX Rocket необходимы отдельные виртуальные машины (ВМ) для следующих компонент:

* Backend Services (API) - 2 ВМ
* Frontend Services (портал) - 1 ВМ
* Apache Kafka (брокер сообщений) -1 ВМ
* Clickhouse (база данных) - 1 ВМ
* PostgreSQL (база данных) - 1 ВМ
--
**Не менее 6 ВМ**

Схема взаимодействия компонент для настройки сетевого доступа показана на рисунке ниже:



## Рекомендуемые системные требования

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | VM | Сервисы | Процессор, ядро (vCPU) | ОЗУ, ГБ (vRAM) | Объем диска, ГБ | Тип диска | Примечание |
| 1 | vm-app-ext | External Backend Services (API)1. ux-api-mobileapi – Сервис сбора данных и тестирования мобильного приложения (Mobile.API)
2. ux-api-getsitescripts – Cервис инициализации скрипта на сайте и для А/В тестирования на сайте (GetSiteScripts.API)
3. ux-api-saverawdataapi – Сервис сбора данных для сайта (SaveRawData.API)
4. uxnginx содержит JS-скрипт сбора данных
 | 8 | 8 | 60 | SSD или SAS 15k |  |
| 2 | vm-app-int | Internal Backend Services (API)1. ux-api-hostedservices – Сервис фоновых заданий (CPA.HostedServices)
2. ux-api-cpaapi – API для работы личного кабинета и админки (CPA.API)
3. ux-ftp – Сервис ftp для загрузки данных
4. ux-api-partnerdataimport – Сервис импорта данных из сторонних систем (CPA.PartnerDataImport)
5. Cкрипт сбора данных размещается на Nginx
 | 4 | 16 | 60 | SSD или SAS 15k |  |
| 3 | vm-front | Frontend Services (Web-portals)1. ux-front-client – кабинет пользователя (CPA.Client.Web)
2. ux-front-admin  – кабинет администратора (CPA.Admin.Web)
 | 2 | 8 | 60 | SSD или SAS 15k |  |
| 4 | vm-kafka | Сервер Apache Kafka (Message brokers) | 4 | 8 | 60 | SSD или SAS 15k |  |
| 5 | vm-clickhouse | Сервер Clickhouse (DB) | 8 | 48 | 120 | SSD | ~ За 12 месяцев при 40.000.000 запросов в месяц |
| 6 | vm-postgres | Сервер PostgreSQL (DB) | 4 | 10 | 120 | SSD | ~ За 12 месяцев при 40.000.000 запросов в месяц |
| 7 |  | Обратный прокси-сервер Nginx |  |  |  |  |  |
| 8 |  | Шлюз  |  |  |  |  |  |
| 9 | vm-opensearch | Opensearch | 4 | 16 | 150 | SAS 15k | необязательный компонент |
| 10 | vm-zabbix | Zabbix | 2 | 8 | 60 | SAS 15k | необязательный компонент |
| 11 | vm-smtp | SMTP |  |  |  |  |  |

1. Для всех ВМ рекомендуется использовать  SSD диски, минимум SSD диски должны быть у Clickhouse и PostgreSQL.
2. Доступ для Backend (API-2) должны быть публичным (доступен из сети Интернет).
3. Доступ для Frontend может быть публичный или с фильтрацией по IP-адресу/ограниченным  из локальной сети и пр.
4. В зависимости от решения которое использует заказчик,  функционал обратного прокси-сервера может быть совмещён со функционалом интернет-шлюза.
5. Clickhouse и PostgreSQL лучше не использовать на одном сервере, работа этих двух баз данных на одной виртуальной машине может вызвать конфликты в ресурсах и привести к снижению производительности.

## Сводная таблица по развернутым сервисам (сетевые порты)

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Виртуальная машина** | **Название docker-контейнеров\сервиса и используемые порты (tcp)** | **Примечание** |
| 1 | PostgreSQL | postgresql 5432 |  |
| 2 | Clickhouse | clickhouse-server 8123, 9000, 9009 |  |
| 3 | Apache Kafka | kafka0 2181, 9092, 9093kafka-ui 8080 |  |
| 4 | Frontend Services | ux-front-client 7000ux-front-admin  – 7500 | Сервисы должны быть доступны из сети интернет,либо ограниченным корпоративной сетью заказчика. |
| 5 | Сервер Backend Services (API) - 1 | ux-api-hostedservices (Nginx 9500 -> UDS)ux-api-cpaapi (Nginx 8000 -> UDS)ux-ftp 21ux-api-partnerdataimport (Nginx 8500 -> UDS)Cкрипт сбора данных размещается на Nginx 80 |  |
| 6 | Сервер Backend Services (API) - 2 | ux-api-mobileapi –  (Nginx 6002 -> UDS)ux-api-getsitescripts –  (Nginx 6001 -> UDS)ux-api-saverawdataapi  (Nginx 6000 -> UDS) | Сервисы должны быть доступны из сети интернет. |

Новые версии UX Rocket используют в базовом варианте сокет домена Unix (Unix domain socket, UDS), вместо TCP, так как этот вариант более эффективен.

На APP-серверах (Docker-хост) нужно настроить промежуточный обратный прокси-сервера (Nginx)  для переадресации запросов на веб-сервер Kestrel внутрь контейнеров по UDS.

## Используемые версии ПО

**Операционная система** - Ubuntu 20.04 LTS и выше, Debian 11 и выше, Astra Linux актуальной версии, **мы можем протестировать любой Linux-дистибутив по требованию.**

**ClickHouse 23.8 LTS** – база данных для хранения потока данных с сайта/МП и формирования аналитической отчётности;
**PostgreSQL** **14.8**- база данных для справочников и настроечных таблиц;

**.NET (версия 7.0)** – платформа для разработки;

**docker,docker-compose** - система контейнеризации, последних версий из оф. репозиториев.

**Apache Kafka 3.0.2**  - брокер сообщений;

**Nginx** - обратный прокси-сервера/веб-сервер последней версии из оф. репозитория.

Для централизованного хранения и просмотра логов может использоваться OpenSearch/fluent-bit последних версий из оф. репозиториев.

Для мониторинга рекомендуется использовать Zabbix из оф. репозитория.

## Общие требования к публикации

1) Для публикации платформы нужно выделить 3 доменных имени:

Для API, например [api.uxrocket-example.ru](http://api.uxrocket-example.ru)

Для ЛК, например [lk.uxrocket-example.ru](http://lk.uxrocket-example.ru)

Для Админки, например [admin.uxrocket-example.ru](http://admin.uxrocket-example.ru)

2)  Для публикации UX Rocket рекомендуется использовать Nginx.

3) При публикации сервисов в сети Интернет рекомендуется использовать  HTTPS.

4) Личные кабинет и админку необязательно делать публичными для всех, доступ к ним может быть ограничен корпоративной сетью заказчика. Для сбора данных достаточно, чтобы был опубликован лишь API.

5) Для отправки паролей и уведолмений сервису нужен доступ к SMTP-серверу.

В зависимости от решения, которое используется для публикации сервисов в корпоративной сети, функционал обратного прокси-сервера может быть совмещён со функционалом интернет-шлюза.