

1. Общие сведения и архитектура системы

Настоящее руководство описывает процесс установки программного обеспечения ПО “Облачная платформа [Cenix.pro](https://cenix.pro) (Ценикс)”. ПО имеет клиент-серверную архитектуру и поставляется по модели SaaS.

- **Серверная часть (Бэкенд):** Разворачивается на мощностях дата-центра на базе инфраструктуры российского облачного провайдера на серверах под управлением ОС Debian(Linux).
- **Клиентская часть (Фронтенд):** Работает через веб-браузер на устройствах пользователей
- **Клиентская часть (мобильное приложение):** Работает через смартфон (устройства) пользователей на платформах Android , Harmony Huawei , IOS.

2. СИСТЕМНЫЕ ТРЕБОВАНИЯ

2.1 Серверная часть :

- **Операционная система сервера:** ОС семейства Linux (например, *Debian 10 и выше*).
- **СУБД (База данных):** *PostgreSQL*
- **Среда выполнения / Веб-сервер:** *Nginx, Node.js, Docker*
- **Минимальные параметры сервера :** Виртуальный сервер с 4 ядерным процессором и частотой от 2 000 Ггц , Оперативная память от 8 ГБ , Жесткий диск (накопитель) SSD / Nvme от 100 гб , канал интернета от 100 мбит/сек

2.1 Клиентская часть :

Для использования программного обеспечения пользователь должен иметь постоянный доступ к сети Интернет. Оборудование пользователя должно соответствовать рекомендуемым требованиям для функционирования браузера, через который пользователь использует программное обеспечение. Для использования программного обеспечения производитель рекомендует пользователю использовать следующие браузеры:

- Google Chrome 108.0 и выше;
- Mozilla Firefox 108.0 и выше;
- Safari 15.5 и выше;
- Opera 94.0 и выше;
- Microsoft Edge 108.0 и выше;
- Vivaldi 5.6 и выше.

3. УСТАНОВКА И НАСТРОЙКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (серверной части)

Программное обеспечение распространяется в виде Интернет-сервиса.

Программное обеспечение предоставляется пользователю в виде готового к работе Интернет-сервиса, пользователь не производит самостоятельную установку и настройку программного обеспечения, а посредством использования браузера и сети Интернет проходит процедуру авторизации в уже настроенном и готовом к работе программном обеспечении, развернутом на оборудовании производителя. После завершения процедуры авторизации пользователь получает возможность использовать программное обеспечение в соответствии с его функциональным назначением.

Процесс первоначальной установки ПО для клиента :

1. **Подготовка окружения:** Нужно установить ряд следующих зависимостей через терминал:

```
apt-get update && apt-get install -y git curl build-essential nginx
```

```
# Docker Engine + docker-compose-plugin по официальной инструкции  
docs.docker.com для Debian
```

```
# Node.js 20 и yarn через nvm
```

```
nvm install 20 && npm i -g yarn
```

Настраивается ротация логов Docker, после чего служба перезапускается командой "systemctl restart docker".

Шаг автоматизирован скриптами step_01.sh и step_01.5.sh из репозитория инсталляционных скриптов производителя.

2. **Клонирование/Загрузка дистрибутива:** Создается каталог инсталляции, в него клонируются репозитории компонентов:

```
mkdir -p /opt/cenix/<имя_инсталляции> && cd  
/opt/cenix/<имя_инсталляции>
```

```
git clone <репозиторий производителя>/cenix-parsing.git
```

```
git clone <репозиторий производителя>/cenix-analytics.git
```

```
git clone <репозиторий производителя>/cenix-frontend.git
```

```
git clone <репозиторий производителя>/cenix-update-utils.git
```

Для каждого компонента из шаблона создается файл конфигурации с помощью команды "`cp .env.example .env`". Каталог `cenix-parsing/server/public-storage` выдаются права на запись.

Этот шаг автоматизирован скриптом "`step_02.sh full`".

3. **Конфигурация параметров:** Параметры задаются в файлах `.env` соответствующих компонентов:
- `cenix-parsing/server/.env`: IP-адрес сервера (`PARSER_HOST_IP`), адрес хранилища скриншотов (`APP_STORAGE_DOMAIN`, `SCREENSHOT_STORAGE`, при использовании S3 - `AWS_*`), реквизиты СУБД (`DB_HOST`, `DB_PORT`, `DB_DATABASE`, `DB_USERNAME`, `DB_PASSWORD`),
 - ключи внешних сервисов модуля парсинга: распознавание капчи (`RUCAPTCHA_API_KEY`), прокси-провайдер (`PROXY_SPACE_API_TOKEN`)
 - `cenix-analytics/.env`: адрес экземпляра (`APP_URL`), `APP_DEBUG=false`, реквизиты подключения к той же СУБД (`DB_*`), параметры почтовых шлюзов (`DASHAMAIL_*`), параметры мониторинга состояния (`WATCHDOG_TELEGRAM_*`).
 - `cenix-frontend/.env`: адрес API аналитики (`ANALYTICS_URL`) и имя хоста интерфейса (`FRONTEND_HOST_NAME`).

На этапе первичного запуска во всех адресах указываются IP-адреса; после выпуска TLS-сертификата в `nginx` они заменяются на реальные доменные имена.

4. **Инициализация базы данных:** Компоненты запускаются последовательно, начиная с модуля парсинга, так как он содержит СУБД:

```
cd cenix-parsing/server && yarn && docker compose up -d --build
cd ../../cenix-analytics && docker compose up -d --build
docker compose exec php ./scripts/install_in_docker.sh
cd ../../cenix-frontend && ./scripts/start
```

Скрипт `install_in_docker.sh` устанавливает PHP-зависимости, выставляет права на каталоги `storage` и `bootstrap/cache`, создает символическую ссылку на публичное хранилище, генерирует ключ приложения (`APP_KEY`) и секрет JWT. Скрипт `scripts/start` собирает веб-интерфейс и поднимает контейнер `frontend`, слушающий порт 5000.

5. **Запуск контейнеров/сервисов:** Создание структуры таблиц и загрузка справочных данных выполняется миграциями и седами обоих модулей:

```
cd cenix-analytics && docker compose exec php php artisan migrate

cd ../cenix-parsing/server && docker compose exec parser npm run
db:migrate

docker compose exec parser npm run db:seed

cd ../../cenix-analytics && docker compose exec php php artisan db:seed

docker compose exec php php artisan cache:clear config:clear view:clear
```

Шаг автоматизирован скриптом `step_03.sh`. После миграций сервер регистрируется как главный узел кластера: в таблицу `clusterNodes` добавляется запись с `id = 0` и IP-адресом сервера.

Добавляем конфиг в `nginx` и выпускаем сертификат для домена аналитики через `certbot`.

Также добавляем задания планировщика в `crontab`:

```
***** docker exec cenix-analytics-php-1 php artisan schedule:run >>
/dev/null 2>&1

01 *** cd /opt/cenix/bkp && /opt/cenix/bkp/backup.sh >> /dev/null 2>&1
```

6. **Проверка работоспособности:**

Состояние контейнеров: `docker compose ps` в каталогах `cenix-parsing/server`, `cenix-analytics`, `cenix-frontend` - все сервисы (`db`, `parser`, `tls-proxy`, `php`, `analytics_nginx`, `supervisor`, `frontend`) в состоянии Up.

- Журналы запуска: `docker compose logs --tail=100 parser`, `docker compose logs --tail=100 php`, файл `cenix-analytics/storage/logs/laravel.log`, лог `installation.log` без ошибок.
- Доступность API: `curl -I https://<домен аналитики>/api/` возвращает корректный HTTP-ответ.
- Вход в личный кабинет по адресу `https://<домен>` (до настройки `nginx` - `http://<IP>:5000`) под сервисной учетной записью администратора, созданной сидами. После входа выполняется тестовое задание парсинга и проверяется появление результатов в отчете.
- Проверка планировщика: в `laravel.log` появляются записи о выполнении `schedule:run`; тестовый запуск `backup.sh` создает непустой архив дампа.

Раздел 4. Порядок запуска и авторизации пользователей

1. Откройте веб-браузер (например, Яндекс.Браузер).
2. Введите в адресную строку URL-адрес развернутого веб-интерфейса <https://app.cenix-pro.com/auth/login>.
3. В открывшемся окне авторизации введите логин и пароль :
 - **Логин**
 - **Пароль**

Произойдет перенаправление в главное рабочее пространство системы мониторинга (личный кабинет).