

Акционерное общество
«Научно-производственное предприятие «Радар ммс»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«IoT-платформа SmartUnity 4.0»

Инструкция по установке

2022 г.

Содержание

1	Введение	3
1.1	Функциональное назначение платформы	3
1.2	Эксплуатационное назначение платформы	3
2	Системные требования	4
2.1	Уровень подготовки пользователя	4
2.2	Требования к аппаратному обеспечению	4
2.3	Требования к предустановленному ПО	4
3	Подготовка к установке	5
3.1	Состав дистрибутива SmartUnity	5
3.1.1	Описание системной части платформы	5
3.1.2	Серверная часть	5
3.1.3	Клиентская часть	6
3.2	Предварительная настройка для установки платформы	6
4	Установка платформы	7
4.1	Установка дистрибутива	7
4.2	Установка SmartUnity	7
4.3	Проверка работы платформы SmartUnity после установки	8
5	Настройка бесперебойной работы Smartunity	10

1 Введение

Настоящий документ является инструкцией по установке программного обеспечения (ПО) «IoT-платформа SmartUnity 4.0» и предоставляет пользователю необходимую информацию и описание последовательности действий, которые необходимо выполнить для разворачивания IoT-платформы SmartUnity 4.0 на серверном и локальном компьютере пользователя.

1.1 Функциональное назначение платформы

IoT-платформа SmartUnity 4.0 (далее SmartUnity) – это интеграционная платформа для создания автоматизированных систем управления разнородными технологическими объектами и устройствами, автоматизации бизнес-процессов и бесшовной интеграции различных аппаратных средств, онлайн сервисов и программных систем.

SmartUnity предоставляет необходимые компоненты для создания решений для автоматизации процессов управления:

- средства поддержки протоколов связи и согласования формата данных;
- средства управления устройствами;
- базу данных;
- средства обработки и управления действиями;
- средства графической визуализации данных;
- средства аналитической обработки данных;
- дополнительные инструменты для разработки и администрирования IoT-решений;
- средства интеграции со сторонними системами (ERP, CRM и прочие).

1.2 Эксплуатационное назначение платформы

IoT-платформа SmartUnity 4.0 предназначена для создания IoT-решений для автоматизации бизнес-процессов и процессов мониторинга, контроля и управления инфраструктурой в области цифровизации городской инфраструктуры, энергетики, промышленности, метео и экологического мониторинга, интеллектуальных транспортных систем и др.

2 Системные требования

SmartUnity является сетевым программным обеспечением и подразумевает работу (администрирование, мониторинг и управление) с одного или нескольких персональных компьютеров, подключенных к одной сети и может быть развернута как на сервере - для сетевой работы нескольких операторов, так и на локальном компьютере пользователя. В настоящей инструкции описан процесс установки SmartUnity на сервере. Процесс установки SmartUnity на отдельном персональном компьютере аналогичен. К персональному компьютеру предъявляются те же требования, что и к серверу в соответствии с п.2.2 настоящей инструкции.

2.1 Уровень подготовки пользователя

Пользователь, выполняющий установку SmartUnity, должен обладать знаниями Unix-подобных операционных систем, системы управления базами данных MongoDB на уровне не ниже администратора и базовыми знаниями программного обеспечения для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации - контейнеризатора приложений Docker. Все команды, связанные с исполнением сценариев в среде Docker, приведены для работы в командной строке. Допускается для работы со средой Docker использовать оригинальное ПО Docker Desktop с обновлениями в соответствии с п.2.3.

2.2 Требования к аппаратному обеспечению

К аппаратному обеспечению сервера, на котором разворачивается платформа SmartUnity, предъявляются следующие минимальные требования:

- Процессор: не ниже Intel Core i7;
- Оперативная память: 32 ГБ;
- Накопитель: 2 x SSD 512ГБ;
- Сетевой адаптер (TCP/IP), скорость передачи 1 Гб/с.

2.3 Требования к предустановленному ПО

На сервере, на котором будет разворачиваться платформа SmartUnity, должно быть предустановлено следующее ПО:

- Операционная система (ОС) Linux/Ubuntu версии не ниже 18.04;
- Контейнеризатор приложений Docker версии не ниже 20.10.16;
- Утилита Docker Compose — версии не ниже 2.5;
- Браузер Google Chrome.

Имя пользователя ОС должно быть в группе Docker.

Конечный пользователь может использовать для работы с платформой рабочую станцию с установленным браузером Google Chrome, имеющую

сетевое подключение к серверу, на котором установлена платформа SmartUnity.

3 Подготовка к установке

3.1 Состав дистрибутива SmartUnity

Платформа SmartUnity построена с использованием микросервисной архитектуры - вариант сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие, насколько это возможно, небольших, слабо связанных и легко изменяемых модулей - микросервисов. Каждый микросервис выполняется в отдельном Docker-контейнере. Все микросервисы объединены в общую сеть средствами Docker Compose.

3.1.1 Описание системной части платформы

Системная часть SmartUnity состоит из контейнеров, созданных на базе официальных образов следующих утилит:

- MongoDB (порт 27017) - документоориентированная нереляционная система управления базами данных;
- Redis (порт 6379) - резидентная система управления базами данных класса NoSQL с открытым исходным кодом;
- RabbitMQ (порты 5672, 15672) - программный брокер сообщений на основе стандарта AMQP, связующее программное обеспечение, ориентированное на обработку сообщений (кроссплатформенное, свободно распространяемое ПО с открытым исходным кодом);
- Nginx (порт 80) — веб-сервер и почтовый прокси-сервер, работающий на Unix-подобных операционных системах

ВАЖНО: указанные в п. 3.1.1 порты не должны быть заняты другими сервисами, в том числе и после перезагрузки сервера.

3.1.2 Серверная часть

В состав серверной части входят следующие микросервисы:

- backend - сервис с главным API SmartUnity;
- devices - сервис-брокер для сервисов lora, modbus и др.;
- lora - сервис, реализующий связь с устройствами на основе протокола LoRaWAN;
- modbus - сервис, реализующий связь с устройствами на основе протокола Modbus;
- notificationsrv - сервис обработки уведомлений и аварий;
- websocketsrv – сервис, обеспечивающий WebSocket- соединение сервисов;

- connectionlogsrv – сервис опроса статуса устройств в реальном времени;
- calculation – сервис выполнения расчётов.

3.1.3 Клиентская часть

Клиентская часть включает следующие микросервисы:

- frontend - web-интерфейс с административной панелью управления, предоставляющий доступ к функциям конфигурирования платформы SmartUnity;
- mainmenu - web-интерфейс оператора, предоставляющий доступ к функциям мониторинга, контроля и управления устройствами, которые были подключены к платформе в результате ее конфигурирования.

3.2 Предварительная настройка для установки платформы

Предварительная настройка осуществляется для обеспечения корректной установки и запуска SmartUnity.

Перед установкой платформы SmartUnity необходимо выполнить следующие действия:

1) Убедиться, что версии ПО Docker и Docker Compose соответствуют указанным в п. 2.3, для этого в командной строке выполнить поочередно команды:

```
$ docker version
$ docker compose version
```

2) Создать в домашней директории пользователя корневую папку Smartunity/ (или с другим удобным именем).

ВАЖНО: адресный путь не должен содержать пробелы и символы Кириллицы.

2.1) В случае установки SmartUnity без общедоступного URL-адреса следует откорректировать файл `/etc/hosts` таким образом, чтобы там появилась строка:

```
127.0.0.1 [SMARTUNITY_URL],
```

и, опционально:

```
127.0.0.1 [SUB_NAME][SMARTUNITY_URL].
```

Вместо `[SUB_NAME]` (опционально) и `[SMARTUNITY_URL]` подставить доменное имя, к которому в дальнейшем будет производиться обращение в браузере. Доменное имя указано в файле «.env» (см. п.4.1.2).

В мультидоменном режиме по умолчанию `[SUB_NAME] = tenant`.

2.2) Убедиться, что порты, указанные в п.3.1.1, не заняты никакими процессами, для чего можно использовать команду:

```
$ sudo lsof -i :[PORT]
```

4 Установка платформы

4.1 Установка дистрибутива

Дистрибутив Smartunity поставляется в виде сжатого архива «archSU.tar.gz» посредством предоставления ссылки для скачивания или на отдельном носителе в соответствии с условиями договора.

Архив содержит в себе:

- архив микросервисов smartunity.tar;
- папка nginxTemplates/, содержащая настройки Nginx;
- файл конфигурации docker-compose.yml;
- файл параметров «.env»;
- папка data/mongo/, содержащая первичные настройки базы данных;
- вспомогательные скрипты (опционально).

Для установки дистрибутива необходимо извлечь архив «archSU.tar.gz» в папку Smartunity/. Для этого переместить архив в папку Smartunity/ и выполнить следующие команды поочерёдно:

```
$ gzip -d archSU.tar.gz
$ tar -xvf archSU.tar ./
```

После извлечения архива необходимо настроить файл «restart.sh», для чего выполнить следующие действия:

- 1) Открыть файл любым удобным редактором и установить значение переменной «dc_file» как абсолютный путь к файлу «docker-compose.yml»;
- 2) Сохранить файл «restart.sh».

4.2 Установка SmartUnity

Для установки платформы SmartUnity необходимо выполнить следующую последовательность действий:

- 1) Развернуть образы микросервисов. Для этого в директории Smartunity выполнить консольную команду:

```
docker load -i [PATH],
```

где [PATH] - абсолютный путь до архива «smartunity.tar». В результате корректного выполнения команды в системе Docker должны появиться Docker-образы микросервисов:

- Modbus;
- websocketsrv;
- mainmenu;
- connectionlogsrv;
- devices;
- frontend;
- notificationsrv;

- lora;
- calculation;
- backend;
- rabbitmq;
- redis;
- mongo;
- nginx.

Проверить список образов можно командой
\$ docker images.

2) Запустить SmartUnity. Для этого перейти в директорию «deploy_SU» и выполнить консольную команду:

```
$ docker compose up -d redis rabbit mongo && sleep 5 && docker  
compose up -d или  
$ ./up.sh.
```

В результате корректного выполнения команды в системе Docker должны появиться рабочие Docker-контейнеры микросервисов. Имена контейнеров совпадают с именами образов, указанных выше.

3) Убедиться в правильной работе микросервисов, для чего выполнить консольную команду:

```
$ docker ps.
```

Все контейнеры должны иметь статус «Up». Такое состояние соответствует готовности платформы SmartUnity к работе.

4) В случае необходимости перезапуска платформы, находясь в папке deploy_SU, выполнить консольную команду:

```
$ ./restart.sh.
```

4.3 Проверка работы платформы SmartUnity после установки

Для проверки работоспособности платформы SmartUnity необходимо ввести в адресную строку браузера адрес [SMARTUNITY_URL]. В случае успешной установки в окне браузера должна появиться стартовая страница с полями для ввода e-mail и пароля пользователя (см. рисунок 1).

В соответствующие поля необходимо ввести e-mail и пароль базового администратора:

email: admin@su.base

пароль: adminPass,

и нажать кнопку «Войти». В окне браузера должна отобразиться страница с панелью администрирования SmartUnity (см. рисунок 2).

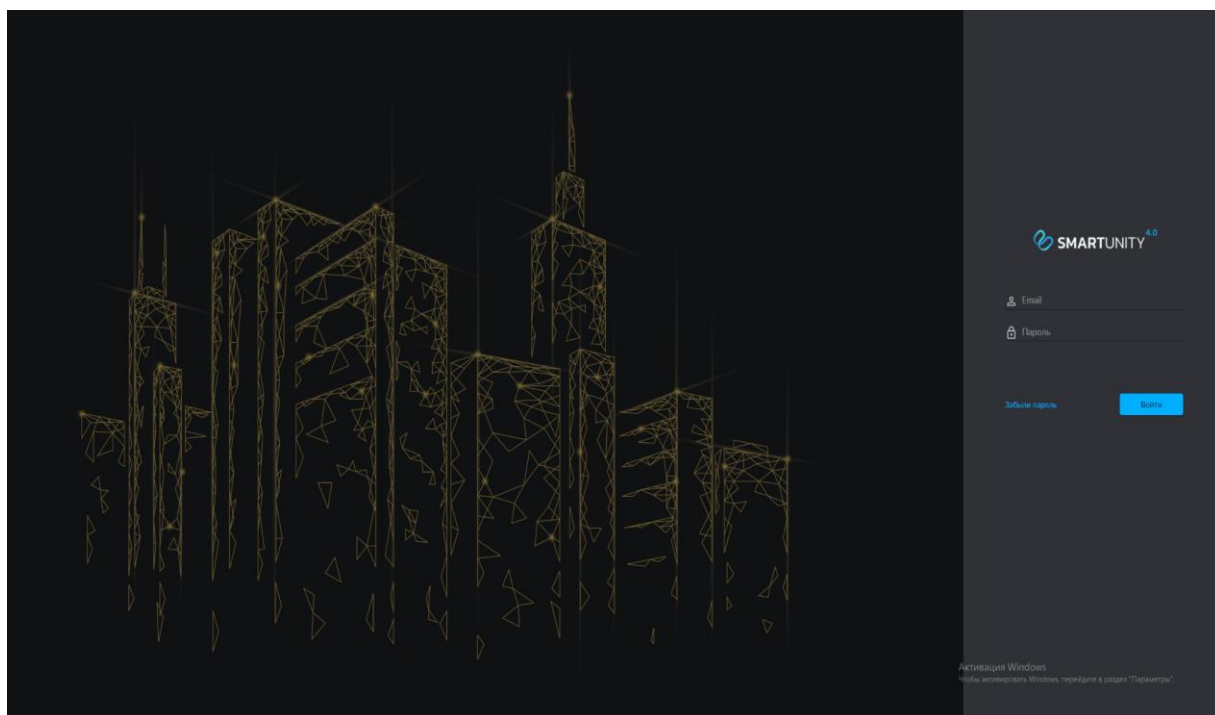


Рисунок 1 - Страница входа в платформу SmartUnity

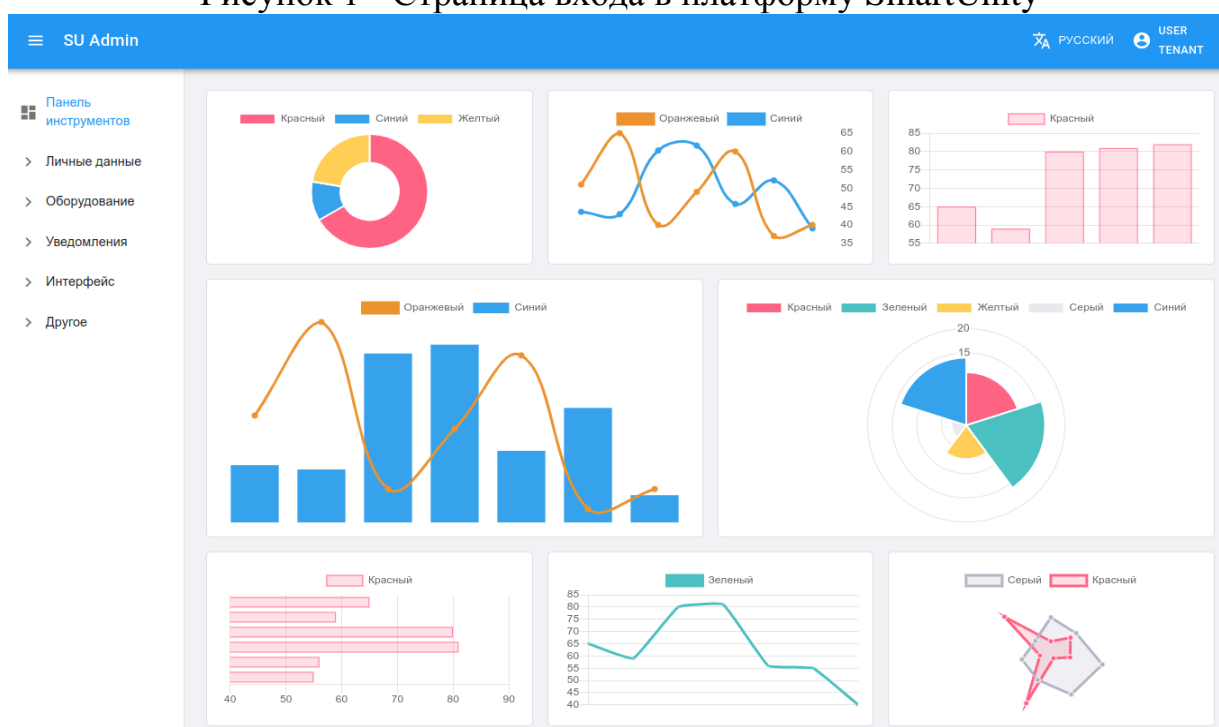


Рисунок 2. Панель администрирования SmartUnity

Данный результат говорит о том, что платформа SmartUnity готова к работе.

В целях безопасности после первого входа под учётной записью базового администратора **настоятельно рекомендуется** поменять пароль базового администратора или создать нового администратора и выполнить удаление учетной записи базового администратора.

5 Настройка бесперебойной работы Smartunity

Для бесперебойной работы Smartunity в условиях непредвиденных перезагрузок сервера рекомендуется использовать планировщик задач Cron (<https://ru.wikipedia.org/wiki/Cron>) и вспомогательный файл «restart.sh».

Для создания задания необходимо выполнить следующие действия:

1) В командной строке ввести команду:

```
crontab -e;
```

2) В конец открывшегося файла дописать строку:

```
@reboot [ПАТН]/restart.sh,
```

где [ПАТН] - абсолютный путь до файла «restart.sh»;

3) Сохранить задание Cron.