

Общество с ограниченной ответственностью
«Центр Реактивного Программирования»
(ООО «ЦРП»)

**СИСТЕМА УПРАВЛЕНИЯ РАСЧЕТАМИ И ДАННЫМИ
«СИМВИА»**

Руководство по установке

2025 год

СОДЕРЖАНИЕ

Введение	3
1. Полезные ссылки.....	3
2. Требования	3
2.1. Аппаратные	3
2.2. Программные	4
2.3. Сеть и узел.....	4
2.4. Операционная система.....	4
3. Установка	5
3.1. Подготовка.....	5
3.2. Запуск системы	7
3.3. Первые шаги.....	7
4. Расширенные настройки	7
4.1. Создание и использование самоподписанных сертификатов.....	7
4.2. Установка системы на нескольких узлах.....	9
5. Обновление существующей версии	9
5.1. С 2024.1 на 2024.2.....	9
5.2. С 2025.1 на 2025.2.....	9

Введение

Документ содержит руководство по установке и настройке системы управления расчетами и данными «Симвия» (<https://simvia.ru>).

1. Полезные ссылки

- 1) Инструкция пользователя: https://simvia.ru/release_docs/latest/userguide.pdf
- 2) Инструкция и скрипты для установки: <https://github.com/crplab/simvia-installation>
- 3) Образы Docker контейнеров: <https://hub.docker.com/u/mycesys>

2. Требования

Ниже приведены рекомендуемые параметры сервера для базовой установки. Если вам нужны уточнения и/или расчет параметров сервера под ваши условия использования, то свяжитесь с нами по адресу: simvia@crplab.ru

2.1. Аппаратные

- 1) Минимальные

Узел	Сервисы	Ядра	Память	Диск
1	Симвия и Хаб	8	24 Гб	512 Гб

- 2) Оптимальные

Узел	Сервисы	Ядра	Память	Диск
1	Симвия	12	64 Гб	512 Гб
2	Хаб	4	16 Гб	60 Гб

- 3) Корпоративные

Узел	Сервисы	Ядра	Память	Диск
1	Хаб	4	16 Гб	60 Гб
2	Симвия Core	8	24 Гб	60 Гб
3	Симвия Tasks	8	16 Гб	60 Гб
4	Симвия Resources	8	16 Гб	60 Гб
5	Симвия Workflow	4	16 Гб	60 Гб
6	Симвия BFF	4	16 Гб	60 Гб
7	Симвия Dashboard	4	16 Гб	60 Гб

8	Симвия Files	4	24 Гб	512 Гб
9	Симвия Gateway	4	16 Гб	60 Гб
10	Симвия 3D	4	24 Гб	512 Гб

Указанное дисковое пространство учитывает потребности компонентов системы, однако конечный необходимый размер файлового хранилища зависит от специфики и интенсивности решаемых задач.

2.2. Программные

- 1) docker – инструмент контейнеризации (официальная инструкция)
- 2) docker compose – плагин к Docker (официальная инструкция)
- 3) openssl – набор библиотек для безопасности (используйте репозиторий дистрибутива или официальный [сайт](#))

2.3. Сеть и узел

- 1) Необходим доступ к Docker Hub
- 2) Права для запуска Docker-контейнеров
- 3) SSL сертификат – для корректной работы системы необходимо использование HTTPS протокола. Если у вас нет собственного SSL сертификата, то можно воспользоваться, например, сервисом [LetsEncrypt](#) или использовать самоподписанный.

2.4. Операционная система

СУРД «Симвия» может быть установлена на любой операционной системе, поддерживающей Docker. Мы рекомендуем использование следующих дистрибутивов:

- 1) Астра Линукс
- 2) РедОС
- 3) АльтЛинус
- 4) Ubuntu 20.04 и новее
- 5) Debian
- 6) Mint
- 7) Gentoo Linux

Скрипты для автоматизированного развертывания предоставлены только для Unix систем, поддерживающих Bash.

3. Установка

ВНИМАНИЕ! Если у вас уже есть установленный экземпляр «Симвия», воспользуйтесь инструкцией по обновлению системы, доступной в последнем разделе.

3.1. Подготовка

1) Получение скриптов развертывания (также можно воспользоваться git clone)

```
wget https://github.com/crplab/simvia-installation/archive/refs/heads/main.zip
```

2) Перейдите в директорию “allinone” (все последующие команды должны выполняться в ней)

```
cd simvia-installation/allinone
```

3) Подготовка директорий

```
./prepare-dirs.sh
```

Полный список директорий доступен в секции Технические детали / Используемые директории.

Обратите внимание, что директория “files/data” используется для хранения файлов моделей и результатов расчета. Убедитесь, что у вас достаточно места для хранения необходимых файлов. Эта директория может быть точкой монтирования сетевого хранилища.

4) Настройка ключей и сертификатов безопасности

4.1) OAuth – разместите приватный и публичный ключи, названные соответственно “oauth” и “oauth.pub” в директории “./oauth/keys”. Ключи должны быть сгенерированы с использованием алгоритма RSA в формате PKCS8. Чтобы сгенерировать новые ключи выполните команду:

```
./generate-oauth-keys.sh
```

4.2) SSL – разместите SSL сертификат в директории “./ssl”

Файл	Описание
ssl-dhparams.pem	Группа Диффи-Хеллмана
fullchain.pem	SSL сертификат
privkey.pem	SSL приватный ключ
options-ssl-nginx.conf	Конфигурационный файл SSL для Nginx

Образец файла “options-ssl-nginx.conf” может быть найден в директории “./selfsigned”, часто он предоставляется поставщиком SSL сертификата.

Для создания самоподписанного сертификата воспользуйтесь соответствующей главой данной инструкции.

5) Настройка параметров в файле “.env”

5.1) Скопируйте шаблон конфигурации из файла “dot.env.example”

```
cp ./dot.env.example ./env
```

5.2) Внесите необходимые параметры в файл “.env”

Имя = значение	Описание
HUB_PUBLIC_URL=hub.yourdomain.com	Адрес сервиса «Хаб» (IP или доменное имя), по которому пользователи будут получать доступ
HUB_PUBLIC_PORT=8301	Порт сервиса «Хаб»
HYPHA_PUBLIC_URL=simvia.yourdomain.com	Адрес «Симвия» (IP или доменное имя), по которому пользователи будут получать доступ к системе
HYPHA_PUBLIC_PORT=8300	Порт «Симвия»

ВНИМАНИЕ! Если вы используете стандартные HTTP/HTTPS порты (80 и 443), то задайте URL без указания порта явно (“:80”, “:443”). Это необходимо сделать так как браузер автоматически отрезает соответствующие значения. Параметры, указанные в таблице выше, должны остаться заданными как есть. Изменения в значения URL

Имя параметра	Значение в шаблоне	Новое значение
OAuth_ISSUER_URL	https://\${HUB_PUBLIC_URL}:\${HUB_PUBLIC_PORT}	https://\${HUB_PUBLIC_URL}
FRONTEND_BASE_URL	https://\${HUB_PUBLIC_URL}:\${HUB_PUBLIC_PORT}	https://\${HUB_PUBLIC_URL}
HUB_WEB_APP_BASE_URL	https://\${HUB_PUBLIC_URL}:\${HUB_PUBLIC_PORT}	https://\${HUB_PUBLIC_URL}
HYPHA_WEB_APP_BASE_URL	https://\${HYPHA_PUBLIC_URL}:\${HYPHA_PUBLIC_PORT}	https://\${HYPHA_PUBLIC_URL}

5.3) Для установки, при которой «Симвия» и «Хаб» устанавливаются на одном узле, URL для этих компонентов может совпадать, однако порты должны быть разные

5.4) Укажите желаемые версии компонентов или оставьте как есть. Версии можно посмотреть на сайте [Docker Hub](https://hub.docker.com/)

5.5) Остальные изменения в данном файле опциональны, вносите изменения только если вы уверены в необходимости

3.2. Запуск системы

- 1) Для запуска системы выполните следующие команды

```
docker compose pull
```

```
docker compose up -d
```

- 2) Если вы используете самоподписанные сертификаты, то дополнительно выполните команду (подробнее в разделе продвинутой конфигурации)

```
./add-ssl-keys-to-jdk.sh
```

3.3. Первые шаги

- 1) Откройте ваш браузер (мы рекомендуем Google Chrome) и перейдите по ссылке, которую вы задавали в файле “.env”:

https://HYPHA_PUBLIC_URL:HYPHA_PUBLIC_PORT

- 2) Если вы используете самоподписанный сертификат, то необходимо принять исключение безопасности в браузере

- 3) Перед вами откроет форма авторизации в системе

- 4) Логин: admin@mycesys.com

- 5) Пароль: root (если вы не меняли значение по умолчанию в файле “.env”)

- 6) После успешной авторизации вы попадете на страницу Dashboard

- 7) Для выполнения дальнейших действий ознакомьтесь, пожалуйста, с инструкцией пользователя, доступной по адресу: https://simvia.ru/release_docs/latest/userguide.pdf

- 8) В случае возникновения проблем ознакомьтесь с разделом: «Проблемы и решения»

- 9) Для покупки лицензий воспользуйтесь сайтом продукта <https://simvia.ru/pricing>

4. Расширенные настройки

4.1. Создание и использование самоподписанных сертификатов

- 1) Вы можете создать свой собственный сертификат SSL для заданного IP адреса или доменного имени без использования центра сертификации

- 1.1) Плюсы таких сертификатов

- Бесплатный сертификат с длительным сроком действия
- Легко управлять (создавать, перевыпускать)
- Нет зависимости на центр сертификации (CA)

- 1.2) Минусы таких сертификатов

- Браузеры будут показывать предупреждение при входе в систему
- Дополнительные шаги в процессе установки

2) Получение сертификата

- Все описанные действия выполняются в директории “allinone”
- Скопируйте файлы “selfsigned/v3.ext” и “selfsigned/ssl.cnf” в директорию “ssl/”

```
cp selfsigned/ssl.cfg ssl/
```

```
cp selfsigned/v3.ext ssl/
```

- Заполните файл “./ssl/ssl.cfg” параметрами вашего сервера:

Параметр	Описание
C	Код страны (например, RU)
ST	Область, округ (например, Saint-Petersburg)
L	Город (например, Saint-Petersburg)
O	Организация (например, Center of Reactive Programming)
OU	Отдел внутри организации (например, Engineering)
CN	Доменное имя или IP адрес (например, simvia.ru)

- Заполните файл “./ssl/v3.ext” параметрами доменного имени
- Заполняется только параметр “subjectAltName”
- Значение после знака “=” – список IP адресов и/или доменных имен, разделенных “,”
- IP адреса задаются “IP:<значение>”
- Доменные имена задаются “DNS:<значение>”
- Заданные значения должны совпадать со значением CN из предыдущего пункта
- Выполните команду для создания SSL ключей

```
./generate-ssl-keys.sh
```

- После запуска всех контейнеров в случае использования самоподписанных сертификатов необходимо выполнить команду

```
./add-ssl-keys-to-jdk.sh
```

ВНИМАНИЕ! Если вы не использовали предоставленный скрипт “generate-ssl-keys.sh” для создания сертификатов, то задайте использованную ключевую фразу в переменную “PASSPHRASE” внутри скрипта “add-ssl-keys-to-jdk.sh”.

4.2. Установка системы на нескольких узлах

- 1) Вы можете установить компоненты системы на нескольких узлах
- 2) Для этого можно использовать отдельные файлы “docker-compose.yml”, представленные в директории каждого компонента
- 3) Для корректной работы системы в этом случае необходимо убедиться, что адрес оркестратора (Consul) во всех компонентах установлен верно
- 4) Мы рекомендуем отдельно устанавливаться компоненты:
 - Хаб (Hub)
 - 3D

Убедитесь, что все узлы, на которых осуществляется установка доступны друг для друга по сети и сетевые задержки минимальны

5. Обновление существующей версии

Обновление существующей версии может выполняться как вручную, так и с помощью скрипта обновления версии. Все скрипты обновления версий доступны в репозитории со скриптами: <https://github.com/crplab/simvia-installation>. Пример инструкций для обновления:

5.1. С 2024.1 на 2024.2

- 1) Выгрузите скрипт «config-migration_2024.1-2025.1.sh» из [репозитория](#)
- 2) Выполните миграцию
 - Запустите скрипт «config-migration_2024.1-2025.1.sh» в директории «./allinone»
 - Проверьте обновленный файл «.env» file перед запуском. Обновите указанные версии сервисов до актуальных. Актуальные версии можно посмотреть в <https://hub.docker.com/u/mycesys>.
- 3) Для обновления контейнеров и запуска выполните:
 - `docker compose pull`
 - `docker compose up -d --remove-orphans`

Если вы используете самоподписанные сертификаты, то дополнительно выполните:

- `./add-ssl-keys-to-jdk.sh`

5.2. С 2025.1 на 2025.2

- 1) Выгрузите скрипт «config-migration_2025.1-2025.2.sh» из [репозитория](#)
- 2) Выполните миграцию
 - Запустите скрипт «config-migration_2025.1-2025.2.sh» в директории «./allinone».

- Проверьте обновленный файл «.env» перед запуском. Актуальные версии контейнеров можно посмотреть в <https://hub.docker.com/u/mycesys>.

3) Для обновления контейнеров и запуска выполните:

- `docker compose pull`
- `docker compose up -d --remove-orphans`

Если вы используете самоподписанные сертификаты, то выполните:

- `./add-ssl-keys-to-jdk.sh`