

УТВЕРЖДЕН
НВЦС.02081-01 85 01-ЛУ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ЦЕНТР УПРАВЛЕНИЯ СЕТЬЮ ФОРТИКС

Руководство администратора

НВЦС.02081-01 85 01

Листов 12

| | | | | |
|--------|--------------|------------|--------|--------------|
| Инв. № | Подп. и дата | Взам. инв. | Инв. № | Подп. и дата |
| | | | | |

АННОТАЦИЯ

Настоящий документ содержит описание архитектуры и работы программного обеспечения Центр управления сетью Фортиск (далее по тексту – ПО, ПО ЦУС Фортиск, ЦУС Фортиск), процесса установки, возможных ошибок и способов их устранения, процедур сопровождения (обновления) и процессов запуска/завершения работы системы.

Руководство предназначено для администраторов, обеспечивающих развёртывание и эксплуатацию ПО на базе операционной системы специального назначения Astra Linux с использованием пакета Docker.

СОДЕРЖАНИЕ

| | |
|---|----|
| 1. Общие сведения | 4 |
| 2. Описание работы ПО | 5 |
| 3. Процесс установки | 6 |
| 3.1 Распаковка архива | 6 |
| 3.2 Импорт Docker-образов | 6 |
| 3.3 Настройка программы | 6 |
| 3.4 Запуск контейнеров | 7 |
| 3.5 Решение возможных ошибок | 7 |
| 3.5.1 Определение неработающего сервиса | 7 |
| 4. Сопровождение | 8 |
| 5. Процесс запуска/завершения работы ПО | 9 |
| Перечень сокращений | 10 |

1. ОБЩИЕ СВЕДЕНИЯ

ЦУС Фортикс – веб-приложение, предоставляющее пользовательский интерфейс посредством веб-браузера для автоматизации мониторинга и управления устройствами ПАК «Фортикс» (подключёнными к сети).

Ключевые функции ЦУС Фортикс:

- сбор, хранение истории изменений конфигураций ПАК «Фортикс» и предоставление доступа к ней;
- мониторинг и управление ПАК «Фортикс» с использованием, в том числе, групповых изменений конфигураций ПАК «Фортикс»;
- предоставление доступа к файловой системе и интерфейсу командной строки ПАК «Фортикс»;
- сбор, хранение журналов ПАК «Фортикс» и предоставление доступа к ним;
- сбор, хранение действий пользователей и ключевых событий ПО ЦУС Фортикс, предоставление доступа к ним.

2. ОПИСАНИЕ РАБОТЫ ПО

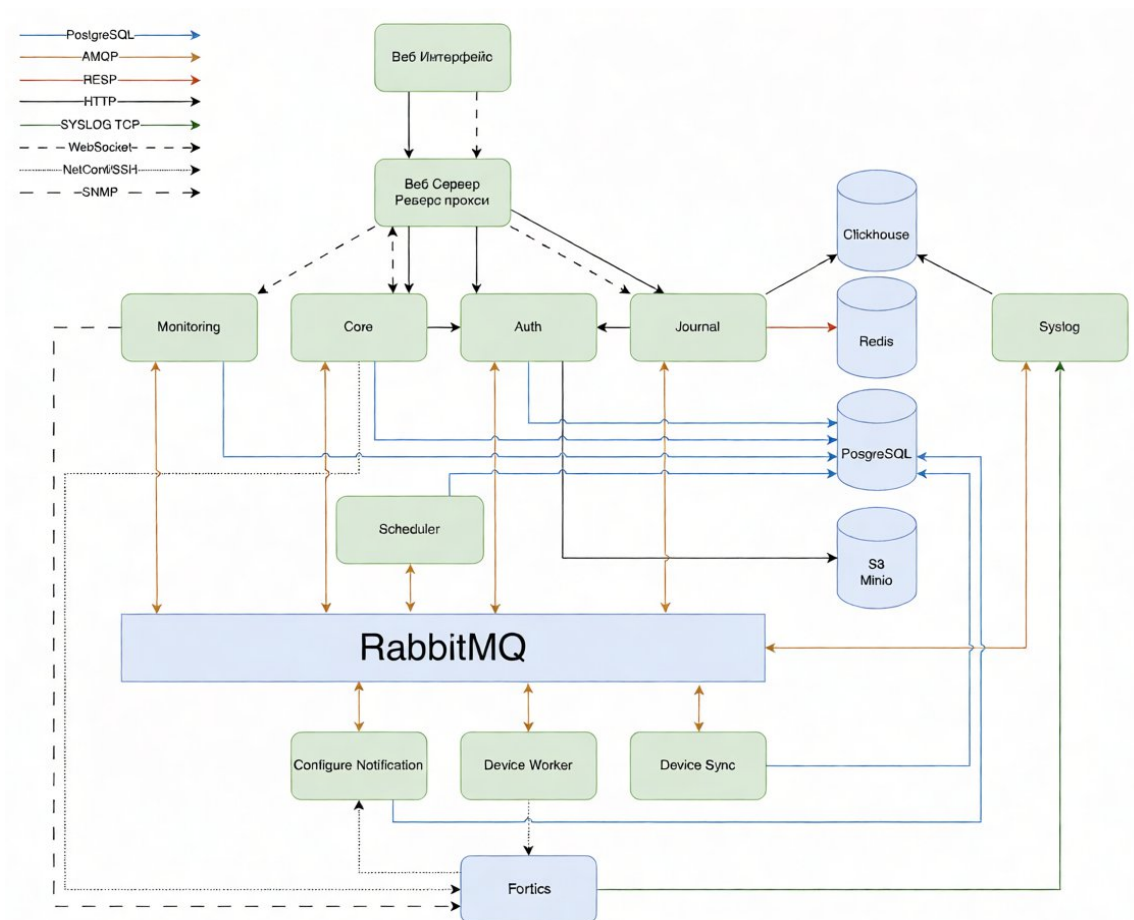


Схема 1. Архитектура сервиса ЦУС Фортикс

Архитектура сервиса ЦУС Фортикс построена по микросервисному принципу с использованием брокера сообщений RabbitMQ.

Ключевые компоненты включают модули мониторинга, аутентификации, управления (Core), журналирования и планировщика задач (Scheduler). Взаимодействие с пользователем осуществляется посредством веб-интерфейса.

Для хранения и обработки данных используются СУБД ClickHouse, PostgreSQL, Redis и объектное хранилище S3 Minio.

Для взаимодействия с устройствами используются модули Device Worker, Device Sync, Configure Notification, Monitoring и Core.

События и сообщения журналов устройств ПАК «Фортикс» передаются через Syslog TCP и записываются в журнал.

Вся коммуникация между сервисами и устройствами осуществляется посредством стандартных протоколов, в том числе NETCONF.

3. ПРОЦЕСС УСТАНОВКИ

ПО ЦУС Фортикс поставляется в виде ISO-архива.

Требования к аппаратному обеспечению:

- Объём оперативной памяти: 16 ГБ;
- Процессор: 6 ядер (x86-64-v2-AES);
- Объём свободного места на диске: 25 ГБ.

Требования к программному обеспечению:

- OS: Astra Linux 1.7+;
- Пакет: docker.io 25+ (пользователь должен принадлежать группе docker).

Оценка оборудования должна исходить из загруженности сервиса.

3.1 Распаковка архива

После скачивания ISO-образа на сервер необходимо извлечь его содержимое в некоторый каталог и выполнить проверку целостности файлов. Все дальнейшие команды выполняются в каталоге с содержимым архива.

Пример применения команд для распаковки архива:

```
sudo mkdir /mnt/iso
sudo mount -o loop zts-ca-docker-complete-dev-43bd0a4a.iso /mnt/iso
ls /mnt/iso
mkdir ~/zts
cp -r /mnt/iso/* ~/zts/
sudo umount /mnt/iso
cd ~/zts
sha256sum -c CHECKSUMS.txt
```

Все файлы должны пройти проверку целостности

3.2 Импорт Docker-образов

Необходимо запустить скрипт load_images.sh с помощью команды

```
bash ./scripts/load_images.sh.
```

3.3 Настройка программы

ЦУС Фортиск требует обязательной настройки HTTPS для корректной работы. В составе поставки включен предварительно настроенный Nginx-сервер с самоподписанным SSL-сертификатом для тестирования.

Самоподписанный сертификат со следующими параметрами:

- Срок действия: 3650 дней
- Тип ключа: RSA 2048
- Субъект: /C=RU/ST=Moscow/L=Moscow/O=ZTS/CN=localhost

Сертификаты размещаются в директории nginx/certs

- nginx-selfsigned.crt — сертификат
- nginx-selfsigned.key — приватный ключ

Nginx настроен на автоматическое перенаправление HTTP-трафика (порт 80) на HTTPS (порт 443) с поддержкой протоколов TLSv1.2 и TLSv1.3.

Для production-среды необходимо заменить тестовый сертификат на корпоративный

Далее необходимо настроить переменные среды в файле .env:

- 1) **Заменить все логины и пароли по умолчанию.**
- 2) Продублировать пароль к сервису clickhouse в файле ./clickhouse/users.xml:22.

3.4 Запуск контейнеров

Запуск контейнеров осуществляется скриптом start_services.sh с помощью команды

```
bash ./scripts/start_services.sh.
```

Через некоторое время приложение запускается, сигналом к этому является завершение выполнения скрипта. Далее веб-интерфейс становится доступен по адресу, указанному в конфигурации веб-сервера.

В веб-интерфейсе необходимо ввести логин root и пароль root, после чего следует изменить пароль в настройках.

3.5 Решение возможных ошибок

События сервера доступны через docker Compose. Для просмотра событий сервера всей программы в реальном времени используется команда

```
docker compose logs -f
```

Дополнительно возможно указание `--timestamps` для меток времени, `--tail=200` для ограничения истории, `--since=5m` для свежих записей.

Для конкретного сервиса используется имя из `docker-compose.yml`:
`docker compose logs -f <service_name>`

3.5.1 Определение неработающего сервиса

Для проверки статусов контейнеров используется команда
`docker compose ps`

П р и м е ч а н и е. Следует обратить внимание на состояния `exited/restarting`.

Для просмотра ошибок сервиса используется команда
`docker compose logs -f <service_name>`

4. СОПРОВОЖДЕНИЕ

Обновление поставляется в аналогичном ISO-архиве с тем же набором компонентов. Для обновления необходимо:

1) Перейти в каталог с работающим ПО ЦУС Фортикс.

2) Из каталога docker-images удалить все сервисы с помощью команды
`rm ./docker-images/zts-ca*.`

3) Распаковать ISO-архив в каталог /tmp.

4) Перенести новые образы в каталог с работающим ПО ЦУС Фортикс с помощью команды
`mv /tmp/docker-images/zts-ca* ./docker-images/.`

5) Обновить тег в файле docker-compose.yml.

Узнать тег возможно по имени ISO-архива, например: имя ISO-архива – zts-ca-docker-complete-dev-43bd0a4a.iso, где dev-43bd0a4a – тег.

Для обновления тега используется команда:

```
sed -i 's/:dev-[a-zA-Z0-9_-]*/:dev-43bd0a4a/g' docker-compose.yml
```

6) Запустить импортирующий новые образы скрипт load_images.sh с помощью команды
`bash ./scripts/load_images.sh.`

7) Запустить скрипт старта start_services.sh для запуска обновлённых сервисов с помощью команды
`bash ./scripts/start_services.sh.`

После обновления необходимо проверить работоспособность ПО, после чего возможно удаление старых образов с помощью команды

```
docker image prune -a -f.
```

5. ПРОЦЕСС ЗАПУСКА/ЗАВЕРШЕНИЯ РАБОТЫ ПО

Для ручного включения используется команда

`docker compose up -d.`

Для ручного выключения используется команда

`docker compose down.`

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ПАК – программно-аппаратный комплекс

ПО – программное обеспечение

ЦУС – центр управления сетью

Лист регистрации изменений

[illegible]