Руководство администратора

по развертыванию системы АльтерОфис Веб АльтерОфис Веб 2026.1.0.0

Страниц 97

ООО «АЛМИ Партнер»
апрель 2026 г.

Оглавление

1.Введение

1.1. Назначение документа

Настоящий документ описывает архитектуру решения, требования к инфраструктуре, процедуры установки, настройки и первичного ввода в эксплуатацию системы **АльтерОфис Веб**, а также процедуры резервного копирования и восстановления данных.

Документ предназначен для системных администраторов, DevOps-инженеров и специалистов технической поддержки, осуществляющих развертывание, ввод в эксплуатацию и сопровождение корпоративной системы файлового хостинга и совместной работы на базе решения **АльтерОфис Веб** в инфраструктуре организации.

В документе описаны два сценария развертывания:

1. **Однонодовое (Single-node):** Для небольших инсталляций, тестовых сред.
2. **Многонодовое (Multi-node):** Для производственных сред.

1.2. Область применения

Руководство применяется при развертывании системы **АльтерОфис Веб** в инфраструктуре организации с использованием инструментов автоматизации **Docker Compose** и **Ansible**.

Документ охватывает подготовку инфраструктуры, установку программного обеспечения, базовую конфигурацию и проверку работоспособности, восстановление системы после сбоя или миграции на новое оборудование.

Документ не описывает администрирование пользовательских учетных записей или детальной настройки системы (для этого см. «Руководство администратора по настройке и администрированию АльтерОфис Веб»).

1.3. Уровень подготовки пользователей

Системный администратор или **DevOps-инженер**, выполняющий развертывание и сопровождение системы, должен обладать знаниями и навыками в областях:

- Основы администрирования Linux-систем (CLI, systemd, управление пакетами, работа с логами).
- Понимание архитектуры микросервисов и контейнеризации.
- Опыт работы с Docker, Docker Compose (управление контейнерами, сети, тома).
- Опыт автоматизации развертывания с помощью Ansible.
- Основы работы с базами данных PostgreSQL.
- Основы администрирования Redis.
- Базовое понимание сетевых технологий (DNS, TCP/IP, настройка брандмауэра).
- Настройка и обслуживание веб-сервера Nginx (проксирование, балансировка, кэширование).

- Настройка и продление TLS-сертификатов.
- Умение работать с системными логами и инструментами мониторинга (Prometheus, Grafana).
- Базовое понимание технологий поиска и анализа данных (OpenSearch, OpenSearch Dashboards).
- Основы информационной безопасности (управление доступом, шифрование, аудит).
- Навыки диагностики и устранения инцидентов.

1.4. Перечень программной и эксплуатационной документации

При эксплуатации системы **АльтерОфис Веб** пользователю могут потребоваться следующие документы:

- «Руководство администратора по развертыванию системы АльтерОфис Веб» (настоящий документ).
- «Руководство администратора по настройке и администрированию АльтерОфис Веб».
- «Руководство пользователя по АльтерОфис Веб».
- «Руководство пользователя по АльтерОфис Редакторы».

2.Общие сведения о системе

2.1. Назначение системы

«АльтерОфис Веб» — это платформа для совместной работы и управления документами, устанавливаемая в инфраструктуре организации (On-Premise).

Основные возможности:

- Централизованное хранение рабочих файлов на собственных серверах организации и обмен файлами.
- Совместное редактирование текстовых документов, электронных таблиц и презентаций несколькими пользователями в реальном времени через веб-интерфейс.
- Обеспечение безопасного доступа к данным как внутри корпоративной сети, так и извне.
- Поиск по названию и содержимому документов.
- Использование интеграции с корпоративными сервисами для аутентификации и авторизации, рассылки уведомлений пользователям системы.
- Мониторинг активности.

2.2. Архитектура системы

АльтерОфис Веб реализован в формате клиент-серверного приложения с доступом через веб-интерфейс. Серверная реализация базируется на микросервисной архитектуре и поставляется в комплекте, объединенном в единый установочный архив на базе docker-контейнеров.

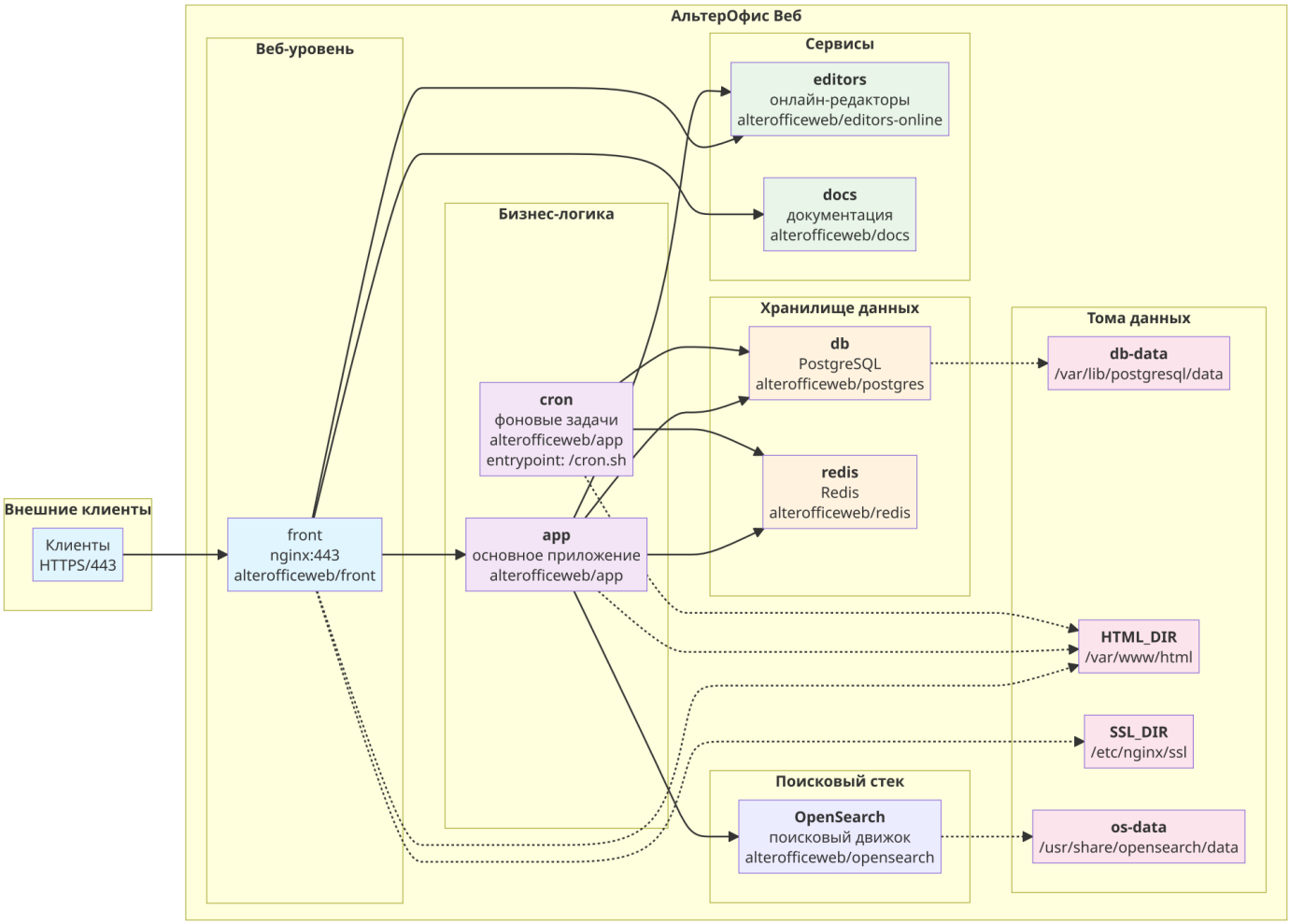


Рисунок 1. Архитектура системы

2.2.1. Основные уровни системы

Уровень	Описание
Веб-уровень	Контейнер **front**, обеспечивает доступ пользователей, перенаправляет запросы в контейнеры **app** и **editors**.
Хранилище данных	Контейнеры **db** и **redis**.
Бизнес-логика	Контейнеры **app** (основное приложение) и **cron** (фоновые задачи).
Сервисы	Контейнеры **editors** (онлайн-редакторы) и **docs** (справочная информация по системе).
Поисковый стек	Контейнеры **opensearch** (поисковый движок).
Мониторинг и логирование	Контейнеры мониторинга, логирования и визуализации.

2.2.2. Описание компонентов системы

Система построена по микросервисной архитектуре, где каждый компонент работает в изолированном docker-контейнере. Взаимодействие между контейнерами осуществляется через

заранее определенные сетевые порты. Внешний веб-трафик поступает на обратный прокси-сервер (Nginx), который маршрутизирует запросы к соответствующим сервисам.

Перечень контейнеров, включенных в поставку, отражен в таблице ниже.

Уровень / Контейнер	Описание
Веб-уровень	
front	Обеспечивает кэширование статического содержимого, отвечает за доступ пользователей и перенаправляет запросы в контейнеры.
Бизнес-логика	
app	Основное веб-приложение, интерпретирует PHP код, реализует серверную логику, обеспечивает хранение файлов, управление пользователями и правами доступа. Взаимодействует с СУБД (PostgreSQL) и кэшем (Redis).
cron	Контейнер, обеспечивающий выполнение фоновых задач, таких как обработка файлов, индексация, рассылка уведомлений.
Хранилище данных	
db	Хранение служебной информации, пользователей, метаданных, настроек.
redis	Используется для кэширования запросов к базе данных, блокировки файлов, повышения производительности.
Сервисы	
editors-online	Контейнер обеспечивает совместное редактирование документов (текстовые файлы, таблицы, презентации). Взаимодействует с основным контейнером **app** по протоколу WOPI.
docs	Контейнер содержит онлайн-документацию к системе.
Поисковый стек	
opensearch	Поисковый движок, используется для реализации расширенного полнотекстового поиска по содержимому файлов и метаданным, REST API поиска и хранения логов.

2.2.3. Описание компонентов мониторинга и логирования системы

При необходимости к системе **АльтерОфис Веб** могут быть подключены компоненты мониторинга состояния инфраструктуры и анализа логов.

Система мониторинга и логирования построена на базе стека инструментов с открытым исходным кодом:

Компоненты	Описание компонентов мониторинга и логирования
Мониторинг метрик	

Компоненты	Описание компонентов мониторинга и логирования
Prometheus	Выполняет роль центрального сборщика количественных показателей. Система в реальном времени агрегирует данные о производительности (загрузка CPU, RPS, задержки, количество ошибок), сохраняя их в виде временных рядов.
node-exporter	Сбор метрик работы операционной системы и аппаратных ресурсов (процессора, памяти, дисков, сети и т. д.) с узлов Linux/Unix-подобных систем.
editors-online	Собирает метрики работы редакторов
prometheus-nginxlog-exporter	Сбор различных метрик, таких как количество запросов, статусы запросов, использование ресурсов и другие характеристики производительности веб-сервера Nginx
postgres-exporter	Сбор метрик PostgreSQL.
redis_exporter	Сбор метрик Redis.
cadvisor	Сбор, обработка и экспорт метрик производительности контейнеров.
php_exporter	Сбор метрик php-fpm (для каждого приложения app).
Grafana	Веб-интерфейс, обеспечивает визуализацию полученных метрик.
Анализ логов	
OpenSearch	Служит централизованным хранилищем текстовых событий, обеспечивает индексацию и поиск по лог-сообщениям от компонентов инфраструктуры.
fluent-bit	Сбор, обработка и пересылка в хранилище (opensearch) лог сообщений.
OpenSearch Dashboards	Предоставляет веб-интерфейс для диагностики (фильтрация событий, визуализация на основе лог-данных).

ПРИМЕЧАНИЕ

Расширенное описание см. в разделе «**Установка сервиса мониторинга и логирования**».

2.2.4. Дополнительные компоненты для развертывания системы

При подготовке системы **АльтерОфис Веб** к продуктивному использованию необходимо предусмотреть интеграцию с ключевыми корпоративными сервисами.

Следующие дополнительные компоненты являются важными для промышленной эксплуатации системы и должны быть настроены до ввода в эксплуатацию:

- **SSL/TLS сертификат** обеспечивает безопасность соединения между клиентом и сервером, шифруя передаваемые данные.

- **Внешнее хранилище** данных позволяет расширить объем хранилища данных для документов пользователей.
- **LDAP / AD** позволяет использовать централизованную аутентификацию через корпоративный каталог.
- Подключение системы к **серверу электронной почты** для рассылки уведомлений пользователям.

Для функционирования системы требуются внешние инструменты оркестрации, не входящие в поставку:

- Среда выполнения контейнеров **Docker Engine**.
- **Docker Compose** для управления контейнерами при установке на одном узле.
- Система автоматизации **Ansible** для развертывания на несколько узлов.

При планировании использования нескольких экземпляров системы **АльтерОфис Веб** в режиме горизонтального масштабирования (для обслуживания большого числа одновременно работающих пользователей) необходимо предусмотреть вынос пользовательских файлов из локальной файловой системы. Для этого рекомендуется рассмотреть внедрение S3-совместимого объектного хранилища (например, MinIO, Ceph и др.) в качестве централизованного хранилища.

Использование объектного хранилища позволит:

- обеспечить корректную работу нескольких экземпляров **АльтерОфис Веб** без необходимости синхронизации локальных каталогов данных;
- снизить нагрузку на файловую систему серверов приложений и улучшить производительность при большом количестве пользователей.

Объектное хранилище должно использоваться исключительно для пользовательских файлов. Метаданные, конфигурация и кэш должны быть вынесены в отдельные, общие для всех узлов сервисы. Логическая архитектура компонентов приведена на рисунке ниже.

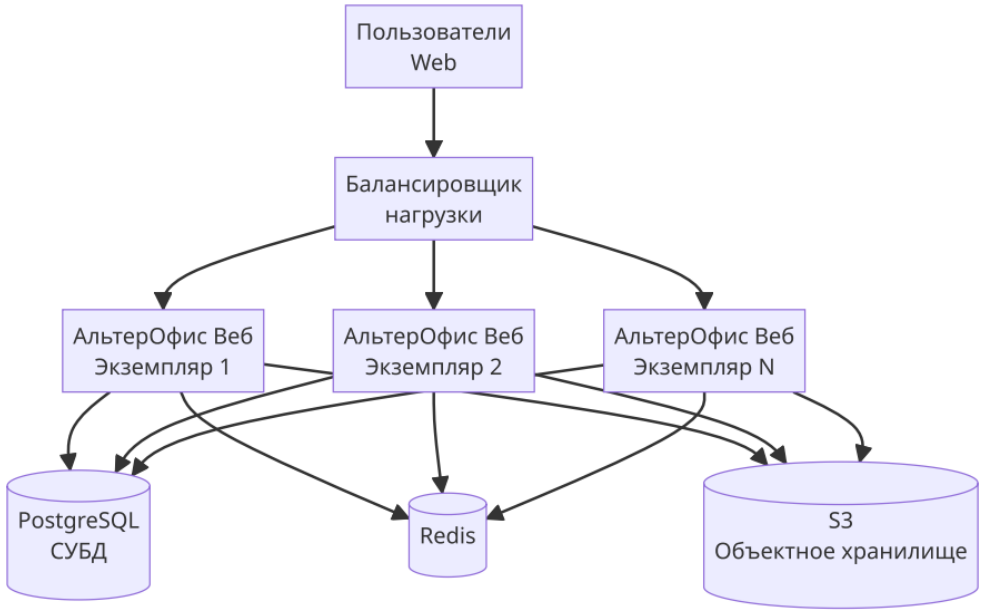


Рисунок 2. Диаграмма компонентов

2.2.5. Порты

Для обеспечения корректной работы **АльтерОфис Веб** необходимо, чтобы были открыты и доступны определенные порты.

Перечень используемых портов:

Сервис	Порт	Протокол	Направление	Зона доступа	Назначение
Веб-сервер и приложение					
Веб-сервер (Nginx)	443	HTTPS	Входящий	**Внешняя**	Доступ к веб-интерфейсу (шифрованный)
PHP-FPM	9000	TCP	Внутренний трафик	Внутренняя (localhost)	Обработка PHP-запросов (обычно через внутренний прокси Nginx)
WebDAV	443	HTTPS	Входящий	**Внешняя**	Доступ к файлам через протокол WebDAV
Базы данных и кэши					
PostgreSQL	5432	TCP	Внутренний трафик	Внутренняя (LAN)	Доступ к базе данных **(не открывать наружу)**

Сервис	Порт	Протокол	Направление	Зона доступа	Назначение
Redis	6379	TCP	Внутренний трафик	Внутренняя (localhost)	Кэширование и очередь задач
Аутентификация и каталоги					
LDAP	389	TCP	Исходящий	Внутренняя (LAN)	Доступ к каталогу пользователей (нешифрованный)
LDAPS	636	TCP	Исходящий	Внутренняя (LAN)	Доступ к каталогу пользователей через SSL
SSO / OAuth2 / SAML	443	HTTPS	Исходящий	**Внешняя**	Интеграция с внешними провайдерами аутентификации
Почтовые сервисы					
SMTP (STARTTLS)	587	TCP	Исходящий	**Внешняя**	Отправка почты через защищённый канал
SMTP (SSL)	465	TCP	Исходящий	**Внешняя**	Отправка почты через SSL
Интеграции					
АльтерОфис Веб Онлайн	9980	HTTP	Внутренний трафик	Внутренняя (LAN)	Интеграция офисного редактора
Мониторинг и логи					
Prometheus	9090	TCP	Внутренний трафик	Внутренняя (LAN)	Сбор метрик
node-exporter	9100	TCP	Внутренний трафик	Внутренняя (LAN)	Сбор метрик работы операционной системы и аппаратных ресурсов (процессора, памяти, дисков, сети и т. д.) с узлов Linux/Unix-подобных систем
editors-online	9980	TCP	Внутренний	Внутренняя	Сбор метрик работы

Сервис	Порт	Протокол	Направление	Зона доступа	Назначение
			трафик	(LAN)	редакторов
prometheus-nginxlog-exporter	4040	TCP	Внутренний трафик	Внутренняя (LAN)	Сбор различных метрик от веб-сервера Nginx
postgres-exporter	9187	TCP	Внутренний трафик	Внутренняя (LAN)	Сбор метрик PostgreSQL
redis_exporter	9188	TCP	Внутренний трафик	Внутренняя (LAN)	Сбор метрик Redis
cadvisor	8081	TCP	Внутренний трафик	Внутренняя (LAN)	Сбор, обработка и экспорт метрик производительности контейнеров
php_exporter	9189	TCP	Внутренний трафик	Внутренняя (LAN)	Сбор метрик php-fpm (для каждого приложения app)
Grafana	3000	TCP	Входящий	Внутренняя (LAN)	Веб-интерфейс для отображения метрик
OpenSearch	9200	TCP	Внутренний трафик	Внутренняя (LAN)	REST API поиска и хранения логов
fluent-bit	24224	TCP	Внутренний трафик	Внутренняя (LAN)	Сбор, обработка и пересылка в хранилище (opensearch) лога сообщений
OpenSearch Dashboards	5601	TCP	Входящий	Внутренняя (LAN)	Веб-интерфейс для отображения логов
Хранилища и объекты					
Object Storage (S3 API)	9005	TCP	Внутренний трафик	Внутренняя (LAN)	Работа с бакетами и объектами S3 хранилища (например MinIO)
Object Storage (Console)	9006	HTTPS	Входящий	Внутренняя (LAN)	Веб-интерфейс для консоли управления (например MinIO)
Управление и фоновые задачи					

Сервис	Порт	Протокол	Направление	Зона доступа	Назначение
SSH	22	TCP	Входящий	Внутренняя (LAN/VPN)	Для управления серверами через Ansible
Cron / Background Jobs	—	—	Внутренний трафик	Внутренняя (localhost)	Локальные задачи (внутренние порты контейнеров)

3.Требования

3.1. Требования к оборудованию

Минимальные требования зависят от количества пользователей и выбранного сценария развертывания. Установка **АльтерОфис Веб** может выполняться как на выделенный сервер, так и в виртуальной среде.

3.1.1. Однонодовое развертывание (до 10 одновременно работающих пользователей)

- Общее количество пользователей 100
- Одновременно работающих пользователей 10

Параметр	Минимальные требования
Процессор (CPU)	4 ядра
Оперативная память (RAM)	8 ГБ
Дисковое пространство	80 ГБ (SSD)
Сетевое подключение	100 Мбит/с

3.1.2. Однонодовое развертывание (до 100 одновременно работающих пользователей)

- Общее количество пользователей 1000
- Одновременно работающих пользователей 100

Параметр	Минимальные требования
Процессор (CPU)	12 ядер
Оперативная память (RAM)	16 ГБ
Дисковое пространство	240 ГБ (SSD) - зависит от планируемого объема хранимых данных.
Сетевое подключение	1 Гбит/с

3.1.3. Многонодовое развертывание (до 500 одновременно работающих пользователей)

- Общее количество пользователей 2500
- Одновременно работающих пользователей 500

Рекомендуемые системные требования на 500 одновременно работающих пользователей (работа с документами и файловым хранилищем), с пиковыми нагрузками до 1000 пользователей.

Узел / Параметр	Процессор (CPU)	Оперативная память (RAM)	Дисковое пространство	Сетевое подключение между узлами
App 1	24 ядра	24 ГБ	250 ГБ	10 Гбит/с

Узел / Параметр	Процессор (CPU)	Оперативная память (RAM)	Дисковое пространство	Сетевое подключение между узлами
App 2	24 ядра	24 ГБ	250 ГБ	10 Гбит/с
DB, Redis	8 ядер	16 ГБ	500 ГБ (SSD)	10 Гбит/с
NFS	8 ядер	16 ГБ	100 ГБ (SSD) для системы + размер данных в зависимости от потребностей организации	10 Гбит/с

3.1.4. Многонодовое развертывание (от 500 одновременно работающих пользователей)

Для поддержки более 500 одновременно работающих пользователей, необходима индивидуальная проработка требований. Система позволяет использовать горизонтальное масштабирование и использовать балансировку нагрузки входящих подключений к серверам.

3.2. Требования к программному обеспечению

3.2.1. Требования к серверным операционным системам

Рекомендованные версии серверных операционных систем:

- АльтерОС 2025 (9.6.6)
- Astra Linux 1.8.5
- РЕД ОС 8.0
- Ubuntu 24.04.03

3.2.2. Требования к контейнеризации, оркестрации и версиям пакетов

Рекомендованные версии Docker Engine и Docker Compose для серверных операционных систем:

Серверная ОС	Docker Engine	Docker Compose
АльтерОС 2025 (9.6.6)	20.10.25	2.23.3
Astra Linux 1.8.5	28.3.3.astra1	2.23.3
РЕД ОС 8.0	28.1.1	2.23.3
Ubuntu 24.04.03	29.4.0	2.23.3

Требования к версиям пакетов:

Наименование пакета	АльтерОС 2025 (9.6)	Astra Linux 1.8.2	РЕД ОС 8.0	Ubuntu 24.04

Наименование пакета	АльтерОС 2025 (9.6)	Astra Linux 1.8.2	РЕД ОС 8.0	Ubuntu 24.04
nfs-utils	nfs-utils-2.5.4	nfs-common=1:2.6.2*	rsync-3.4.1*	rsync=3.2.*
rsync	rsync-3.4.1	rsync=3.2.7*	nfs-utils-1:2.7*	nfs-common=1:2.6*
haproxy	haproxy-2.8.*	haproxy=2.6.*	haproxy-3.2.*	haproxy=2.8.*
nfs-server	nfs-server	nfs-kernel-server=1:2.6.*	nfs-server	nfs-kernel-server=1:2.6.*
netcat	netcat	netcat-openbsd	netcat	netcat-openbsd
ansible	ansible==7.7.0	ansible=9.4	ansible==7.7.0	ansible=9.2
python	Python 3.9	Python 3.11.2 (установлен по умолчанию)	Python 3.11 (установлен по умолчанию)	Python 3.12 (установлен по умолчанию)

- Python требуется для работы Ansible.
- Ansible требуется для многонодового развертывания.

3.2.3. Требования к клиентской части (браузеры)

Рекомендованные версии браузеров:

- Яндекс Браузер. Версия: 25.8.5.983 (64-разрядный).
- Google Chrome. Версия: 141.0.7390.123 (64-разрядный).
- Mozilla Firefox. Версия: 144.0 (64-разрядный).

3.3. Требования к сетевой инфраструктуре

- Статические IP-адреса для серверов.
- Зарегистрированное доменное имя (FQDN), направленное на IP-адрес сервера.
- Валидный SSL-сертификат для доменного имени.
- Открытые и перенаправленные порты 80 и 443 на хост-систему.
- Стабильное интернет-соединение с достаточной исходящей скоростью для передачи файлов пользователям.
- Поддержка DNS для доступа к контейнерам по именам.
- Ограничение внешнего доступа к внутренним сетям Docker.

3.4. Требования к безопасности

- Запрет доступа по SSH с использованием пароля, использование ключей.
- Регулярное обновление системных пакетов и образов Docker.
- Использование сложных паролей для всех учетных записей (СУБД, администратора системы).

- Изоляция контейнеров в отдельной сети Docker.
- Использование TLS-сертификатов для всех внешних соединений.

4.Установка системы

Развертывание системы **АльтерОфис Веб** будет рассмотрено на двух примерах:

1. **Однонодовое:** Для небольших инсталляций, тестовых сред.
2. **Многонодовое:** Для производственных сред.

4.1. Однонодовое развертывание

Данный вариант предполагает размещение всех компонентов системы на одном физическом или виртуальном сервере.

Однонодовое развертывание подходит для проведения функционального тестирования, развертывания демонстрационных стендов, а также для использования в небольших организациях с ограниченным числом пользователей, где допускаются упрощенные требования к доступности системы.

Ниже будет рассмотрен пример установки **АльтерОфис Веб** на одну ноду (узел). В примере используется операционная система Ubuntu.

После развертывания, система будет доступна по адресу `https://demo03-web2025.alteroffice.ru`

Параметры, которые потребуются для установки:

Параметр	Описание	Пример
DEPLOYMENT_PATH_SERVER	Путь к директории развертывания на хосте	`/opt/alterofficeweb/`
PATH_SSL	Путь к директории с SSL сертификатами	`/etc/ssl/alteroffice`
SERVER_IP	IP-адрес хоста сервера, где установлен контейнер **app** для **Экземпляр 1**	`172.20.18.10`
SERVER_URL	URL доступа к системе АльтерОфис Веб	`https://demo03-web2025.alteroffice.ru`
SERVER_HOSTNAME	Доменное имя или FQDN (Fully Qualified Domain Name) сервера АльтерОфис Веб.	`demo03-web2025.alteroffice.ru`
SERVICE_USER_SERVER	Пользователь для развертывания и управления контейнерами	`gitlab-runner`

4.1.1. Подготовка к установке

Для установки **АльтерОфис Веб** подготовьте выделенный или виртуальный сервер с требуемыми характеристиками.

Шаг 1. Убедитесь, что **сетевые порты 443 и 5601 правильно настроены**: правила файрвола разрешают входящий и исходящий трафик на эти порты для подготовленного сервера.

Шаг 2. Убедитесь, что **есть валидные SSL сертификаты** с именами **server.key** и **server.crt**.

Шаг 3. Установите Docker и Docker Compose.

Шаг 4. Создайте рабочую директорию для установки.

1. Создайте директорию для развертывания:

```
sudo mkdir -p <DEPLOYMENT_PATH_SERVER>
```

Пример:

```
sudo mkdir -p /opt/alterofficeweb/
```

2. Перейдите в директорию развертывания:

```
cd <DEPLOYMENT_PATH_SERVER>
```

Пример:

```
cd /opt/alterofficeweb/
```

Шаг 5. Если доменное имя не определяется через DNS, укажите его соответствие IP-адресу в файле `/etc/hosts`.

```
sudo bash -c "echo ' <SERVER_IP> <SERVER_HOSTNAME>' >> /etc/hosts"
```

Пример:

```
sudo bash -c "echo '172.20.18.10 demo03-web2025.alteroffice.ru' >>
/etc/hosts"
```

Шаг 6. Получите установочный бандл и инсталлятор из репозитория `https://repo.alteroffice.ru/web/` любым удобным способом.

Бандл представляет собой архив, содержащий образы Docker:

- `alterofficeweb-bundle_*.tar.gz`

Инсталлятор представляет собой архив со скриптами установки:

- архив с установщиком для одного узла (ноды) `alterofficeweb-installer_*.zip`

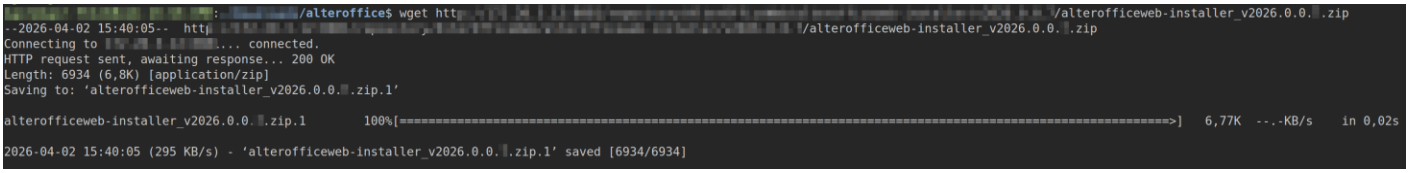


Рисунок 3. Скачивание установщика из репозитория

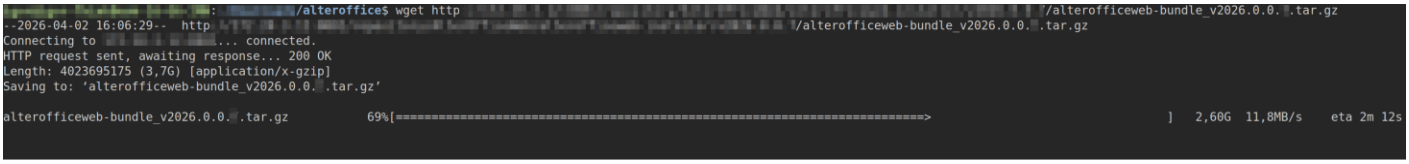


Рисунок 4. Скачивание бандла из репозитория

Шаг 7. Распакуйте архив с установщиком в текущую директорию:

```
sudo unzip alterofficeweb-installer_*.zip
```

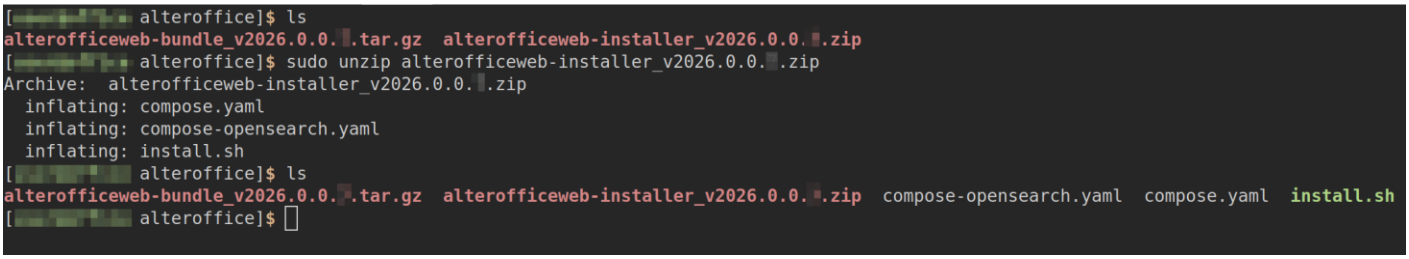


Рисунок 5. Извлечение файлов из архива

Шаг 8. Подготовьте SSL сертификаты, которые будут использоваться для HTTPS доступа к серверу:

1. Создайте директорию для SSL сертификатов :

```
sudo mkdir -p <PATH_SSL>
```

Пример:

```
sudo mkdir -p /etc/ssl/alteroffice
```

2. Поместите в директорию SSL сертификаты для вашего домена:

- **server.crt** - SSL сертификат
- **server.key** - приватный ключ

3. Убедитесь, что файлы имеют правильные права доступа:

```
sudo chmod 644 <PATH_SSL>/server.crt
sudo chmod 600 <PATH_SSL>/server.key
```

Пример:

```
sudo chmod 644 /etc/ssl/alteroffice/server.crt
sudo chmod 600 /etc/ssl/alteroffice/server.key
```

ВНИМАНИЕ

- Убедитесь, что доменное имя указано правильно и DNS запись ведет на ваш сервер.
- Директория для контента будет создана автоматически скриптом.
- Директория для SSL сертификатов должна существовать до запуска инсталляционного скрипта.

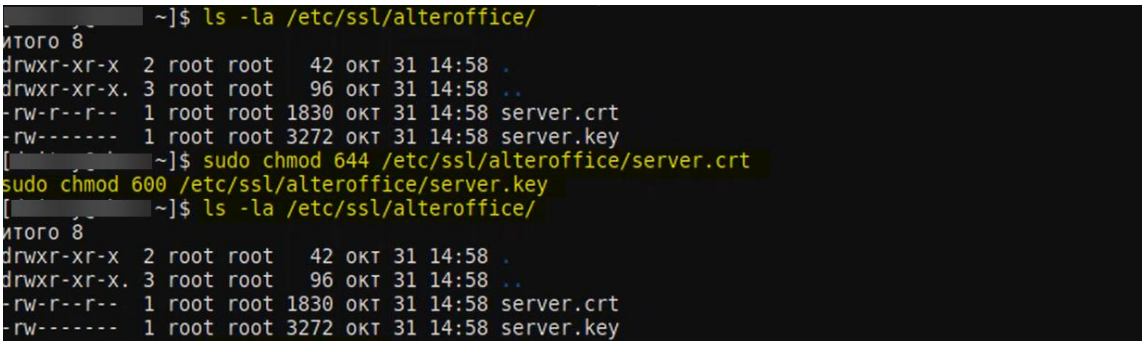


Рисунок 6. Подготовка сертификатов

Шаг 9. Убедитесь, что в директории установки появились необходимые файлы:

- compose.yaml
- compose-opensearch.yaml
- install.sh

4.1.2. Установка на один сервер

Шаг 1. Настройка внешнего S3-хранилища при развертывании с нуля

Если вы хотите подключить внешнее S3 хранилище MinIO до развертывания системы, то обратитесь к инструкции «**5.3.2. Настройка MinIO в качестве основного хранилища при развертывании с нуля**».

Шаг 2. Запустите скрипт установки:

```
sudo ./install.sh
```

Шаг 3. Следуйте инструкциям скрипта:

- Укажите директорию для хранения контента (например: `/var/www/alteroffice`).
- Укажите директорию с SSL сертификатами (например: `/etc/ssl/alteroffice`).
- Введите доменное имя (например: `office.alteroffice.ru`).
- Подтвердите автоматически определенный IP или введите вручную.
- Введите логин администратора системы АльтерОфис Веб (по умолчанию: `admin`).
- Введите пароль администратора АльтерОфис Веб.

```
[ alteroffice]$ sudo ./install.sh
[sudo] пароль для user:
=== Установка AlterOfficeWeb ===
Docker установлен: Docker version 29.1.3, build f52814d
Найден Docker Compose плагин: Docker Compose version v5.0.1
Проверка доступности портов...
Проверка порта: 443
Local connection test...
nc -l 443
nc: Порт 443 локально доступен
====================
Порт 443: доступен
====================
Введите директорию для хранения контента: /var/www/alteroffice
Введите директорию где находятся SSL сертификаты: /etc/ssl/alteroffice
Введите доменное имя: office.alteroffice.ru
Введите логин администратора AlterOffice [admin]:
Введите пароль администратора AlterOffice:
Окно техобслуживания(4 часа): для ресурсоёмких фоновых задач AlterOfficeWeb
Без настройки окна обслуживания задачи выполняются в случайное время, что может замедлить работу
Рекомендуется устанавливать время в период низкой активности
Установить окно техобслуживания? (y/n) [n]:
Сгенерирован случайный пароль для PostgreSQL
Директория уже существует: /var/www/alteroffice
Key is valid
SSL сертификаты найдены и валидны: /etc/ssl/alteroffice/server.crt и /etc/ssl/alteroffice/server.key
Найден бандл: ./alterofficeweb-bundle_v2026.0.0. .tar.gz
Загрузка docker образов из ./alterofficeweb-bundle_v2026.0.0. .tar.gz...
```

Рисунок 7. Процесс установки

Шаг 4. Дождитесь выполнения скрипта.

По завершению работы скрипта будет выведено уведомление `"=== AlterOfficeWeb запущен ==="`.

Шаг 5. Проверьте список контейнеров:

```
sudo docker ps
```

```
[ ]$ sudo docker ps
[sudo] пароль для user:
CONTAINER ID   IMAGE                                      COMMAND                  CREATED         STATUS         PORTS                                                  NAMES
dbe10bd48ff7   alterofficeweb/front:v2026.0.0             "/entrypoint.sh"         4 minutes ago   Up 4 minutes   80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp        prod-front-1
e123fb7b8482   alterofficeweb/app:v2026.0.0               "/entrypoint.sh php-…"   4 minutes ago   Up 4 minutes   9000/tcp                                               prod-app-1
622e4e11f3fe   alterofficeweb/app:v2026.0.0               "/cron.sh"               4 minutes ago   Up 4 minutes   9000/tcp                                               prod-cron-1
1e235b654e93   alterofficeweb/editors-online:v2026.0.0    "/start-collabora-on…"   4 minutes ago   Up 4 minutes   9980/tcp                                               prod-editors-1
5a7e57e9db22   alterofficeweb/postgres:16.9-alpine3.22    "docker-entrypoint.s…"   4 minutes ago   Up 4 minutes   5432/tcp                                               prod-db-1
62c178a6ebc7   alterofficeweb/redis:7.4.5-alpine3.21      "docker-entrypoint.s…"   4 minutes ago   Up 4 minutes   6379/tcp                                               prod-redis-1
4e816d4daf55   alterofficeweb/docs:v2026.0.0              "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes   80/tcp                                                 prod-docs-1
41f1d6f55796   alterofficeweb/opensearch:v2026.0.0        "./opensearch-docker…"   4 minutes ago   Up 4 minutes   9200/tcp, 9300/tcp, 9600/tcp, 9650/tcp                 prod-os-opensearch-1
[ ]$
```

Рисунок 8. Список контейнеров после установки

Шаг 6. Проверьте доступность веб-интерфейса.

Через несколько минут после завершения установки интерфейс «Альтер Офис Веб» становится доступным. Получить к нему доступ можно через веб браузер, используя имя сервера и данные учётной записи, заданные при установке.

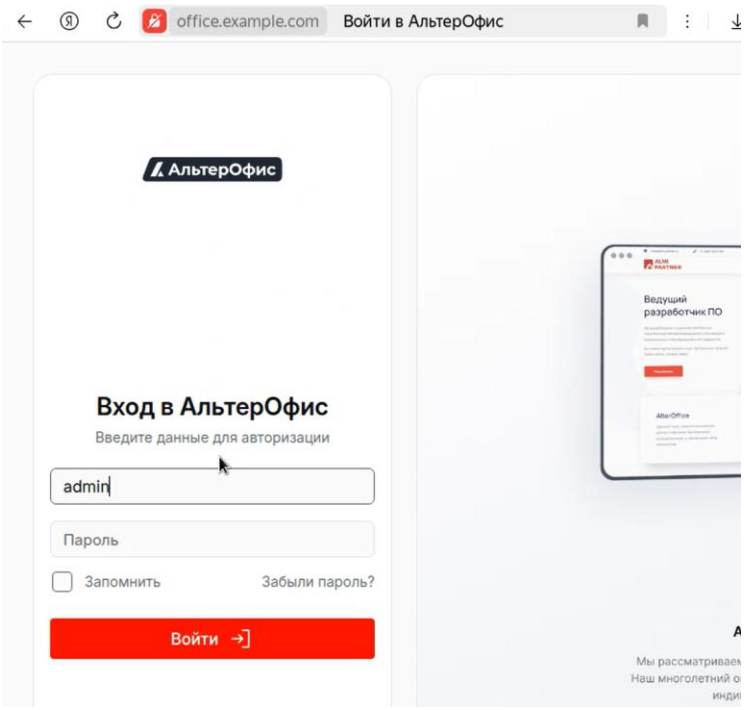


Рисунок 9. Окно авторизации АльтерОфис Веб

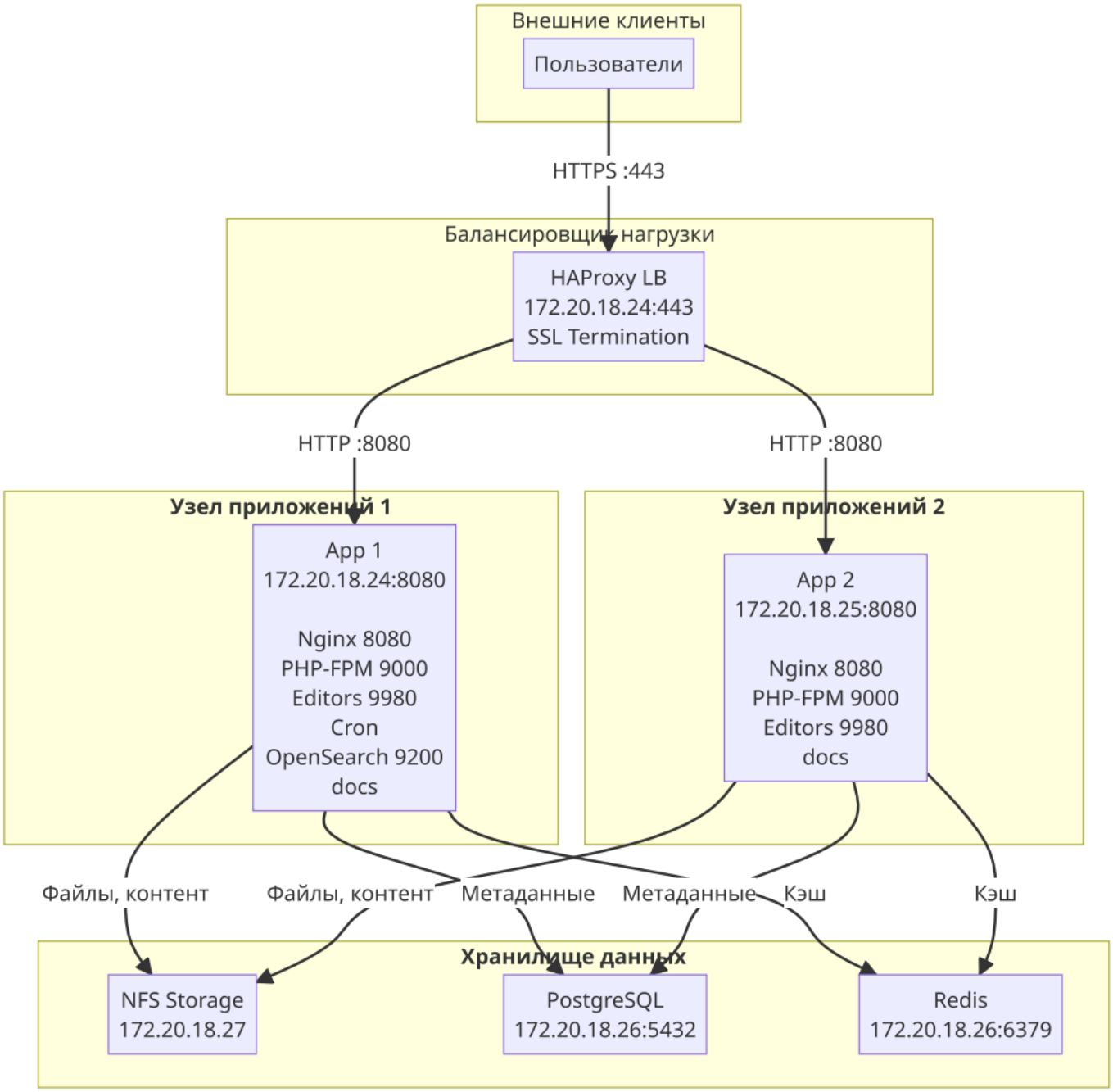
Шаг 7. Подключение внешнего S3 хранилища на уже развернутой системе.

Если вы хотите подключить внешнее S3 хранилище MinIO на уже развернутой системе, то обратитесь к разделу «**Настройка основного хранилища MinIO на уже развернутой системе**».

4.2. Многонодовое развертывание

Для продуктивных сред с высокой нагрузкой рекомендуется распределенная архитектура. Этот сценарий обеспечивает высокую доступность за счет распределения компонентов по специализированным узлам.

Общая архитектура многонодового развертывания приведена на схеме ниже.



Ниже будет рассмотрен пример установки АльтерОфис Веб на четыре ноды на серверах с операционной системой **АльтерОС**.

4.2.1. Подготовка к установке

Для установки АльтерОфис Веб подготовьте 4 физические либо виртуальные машины.

В данном разделе описывается процесс подготовки многонодовой инфраструктуры к развертыванию системы и включает:

- создание сервисного пользователя,
- настройку прав доступа,

- установку необходимых компонентов (Docker Compose, Python, Ansible),
- настройку бесключевого SSH-доступа,
- загрузка архивов с docker-образами и скриптами установки,
- подготовка плана развертывания.

Шаг 1. На каждом из серверов настройте статические IP адреса.

Шаг 2. Определите, как сервисы будут размещены по нодам.

Пример:

IP-адрес	Имя сервера	Размещенные компоненты	Описание
172.20.18.24	Сервер 1	front, app, editors, docs, cron, HAProxy, Opensearch, Ansible Control Node	Выполняет функции входной точки и управления. Включает в себя балансировщик трафика (**HAProxy**), основные веб-компоненты (**front, app, editors, docs**), служебные задачи (**cron**), а также движок поиска (**Opensearch**). Данный узел выступает в роли **Ansible Control-node**, с которого осуществляется управление всем кластером.
172.20.18.25	Сервер 2	front, app, editors, docs	Служит для горизонтального масштабирования вычислительных мощностей. Дублирует ключевые компоненты (**front, app, editors, docs**) для распределения пользовательских запросов.
172.20.18.26	Сервер 3	PostgreSQL, Redis	Выделенный сервер для работы с базами данных (**PostgreSQL**) и кэшем (**Redis**).
172.20.18.27	Сервер 4	NFS Server	Узел для организации общего файлового хранилища (**NFS**), обеспечивает единое дисковое пространство для всех узлов приложений.

Шаг 3. Подготовьте параметры, которые потребуются для установки:

Параметр	**Описание**	**Пример**
PATH_SSL	Путь к директории с SSL сертификатами	`/etc/ssl/alteroffice`

Параметр	Описание	Пример
PATH_DEPLOY	Путь к директории для работы с установщиком	`/opt/aow_deploy/`
DEPLOYMENT_PATH_SERVER	Путь к директории развертывания на хосте	`/opt/alterofficeweb/`
SERVER_1_IP	IP-адрес хоста сервера для установки сервисов первой ноды (front, app, editors, docs, cron, HAProxy, Opensearch, Ansible Control Node).	`172.20.18.24`
SERVER_2_IP	IP-адрес хоста сервера для установки сервисов второй ноды (front, app, editors, docs).	`172.20.18.25`
SERVER_3_IP	IP-адрес хоста сервера для установки сервисов третьей ноды (PostgreSQL, Redis).	`172.20.18.26`
SERVER_4_IP	IP-адрес хоста сервера для установки сервисов четвертой ноды (NFS Server).	`172.20.18.27`
SERVER_5_IP	IP-адрес хоста сервера для установки сервисов логирования и мониторинга.	`172.20.18.28`
SERVER_URL	URL доступа к системе АльтерОфис Веб	`https://prod-02.alteroffice.ru`
SERVER_HOSTNAME	Доменное имя или FQDN (Fully Qualified Domain Name) сервера АльтерОфис Веб.	`https://prod-02.alteroffice.ru`
ANSIBLE_USER	Пользователь для развертывания и управления Ansible	`ansible`
INVENTORY_DIRECTORY	Путь к каталогу, содержащему полную конфигурацию Ansible (список серверов, переменные, пароли) для среды (окружения) развертывания	`inventories/prod-02`

ПРИМЕЧАНИЕ

Рекомендуется разделять пути:

- к директории развертывания на хосте **DEPLOYMENT_PATH_SERVER**;
- и к директории с SSL сертификатами **PATH_SSL**.

Шаг 4. Убедитесь, что **сетевые порты 443, 8080, 9200, 9980, 5432, 6379, 22, icmp** свободны и имеют правильно настроенные правила файрвола, которые разрешают входящий и исходящий трафик на эти порты для подготовленных серверов.

Шаг 5. Подготовьте SSL сертификаты, которые будут использоваться для HTTPS доступа к серверу

Данный шаг выполняется на управляющей ноде (Сервер 1).

Требования к названиям сертификатов:

- **server.crt** - SSL сертификат;
- **server.key** - приватный ключ;
- **cert.pem** - файл-контейнер для объединённых сертификатов.

Файл **cert.pem** можно получить так:

```
cat server.crt server.key > cert.pem
```

1. Создайте директорию для SSL сертификатов :

```
sudo mkdir -p <PATH_SSL>
```

Пример:

```
sudo mkdir -p /etc/ssl/alteroffice
```

2. Поместите в директорию SSL сертификаты для вашего домена

Проверьте наличие сертификатов в `<PATH_SSL>`.

Пример:

```
sudo ls -la /etc/ssl/alteroffice
```

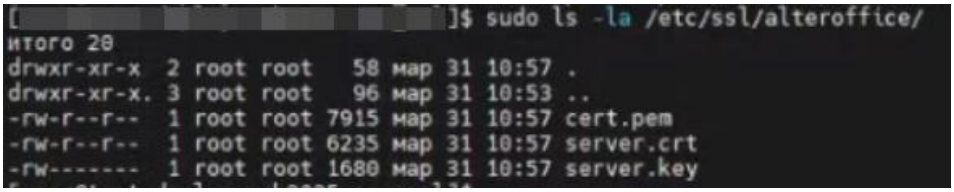


Рисунок 10. Подготовка сертификатов

3. Убедитесь, что файлы имеют правильные права доступа

```
sudo chmod 644 <PATH_SSL>/server.crt
sudo chmod 600 <PATH_SSL>/server.key
sudo chmod 600 <PATH_SSL>/cert.pem
```

Пример:

```
sudo chmod 644 /etc/ssl/alteroffice/server.crt
sudo chmod 600 /etc/ssl/alteroffice/server.key
sudo chmod 600 /etc/ssl/alteroffice/cert.pem
```

> ВНИМАНИЕ
>
> - Убедитесь, что доменное имя указано правильно и DNS запись ведет на ваш сервер.
> - Директория для SSL сертификатов должна существовать до запуска инсталляционного скрипта.

4.2.1.1. Подготовка окружения на всех узлах

Шаг 1. Настройка сервисного пользователя **ANSIBLE_USER** на управляющей и целевых нодах

Для работы скриптов автоматизации необходимо создать пользователя с правами `sudo` без запроса пароля. Действие выполняется на всех подготовленных серверах (Сервер 1, Сервер 2, Сервер 3, Сервер 4).

1. Создайте сервисного пользователя **ANSIBLE_USER**

```
# Создание пользователя
sudo useradd <ANSIBLE_USER>
# Установка пароля
sudo passwd <ANSIBLE_USER>
```

Пример:

```
sudo useradd ansible
sudo passwd ansible
```

2. Предоставьте сервисному пользователю **ANSIBLE_USER** возможность выполнять команды с повышенными привилегиями без запроса пароля

```
# Настройка привилегий sudo без пароля
sudo echo "<ANSIBLE_USER> ALL=(ALL) NOPASSWD: ALL" | sudo tee
/etc/sudoers.d/<ANSIBLE_USER_FOLDERS>
# Настройка прав на папку
sudo chmod 440 /etc/sudoers.d/<ANSIBLE_USER_FOLDERS>
```

Пример:

```
sudo echo 'ansible ALL=(ALL) NOPASSWD: ALL' | sudo tee /etc/sudoers.d/ansible
sudo chmod 440 /etc/sudoers.d/ansible
```

Шаг 2. Установка Docker Compose

Для управления контейнерами необходимо установить docker compose и настроить пути доступа. Действие выполняется на всех подготовленных серверах (Сервер 1, Сервер 2, Сервер 3, Сервер 4).

Для АльтерОС используйте версию docker compose `2.23.3`.

1. Загрузка бинарного файла

```
# Загрузка бинарного файла
sudo wget
"https://github.com/docker/compose/releases/download/v2.23.3/docker-compose-
linux-x86_64" -O /usr/local/bin/docker-compose
```

2. Настройка прав доступа и создание системной ссылки

```
# Настройка прав доступа и создание системной ссылки
sudo chmod +x /usr/local/bin/docker-compose
sudo ln -sf /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Шаг 3. Создание рабочей директории для установки

Действия выполняются на всех подготовленных серверах (Сервер 1, Сервер 2, Сервер 3, Сервер 4).

1. Подготовка директории

Создайте структуру папок для хранения образов системы.

```
# Создание базовой директории
sudo mkdir -p <DEPLOYMENT_PATH_SERVER>
```

Пример:

```
sudo mkdir -p /opt/alterofficeweb/
```

Перейдите в директорию:

```
cd <DEPLOYMENT_PATH_SERVER>
```

Пример:

```
cd /opt/alterofficeweb/
```

2. Создайте подпапку `images` и перейдите в нее

В данную директорию будет загружен архив с docker-образами.

```
# Создание директории для загрузки образов
sudo mkdir images
cd images
```

3. Загрузите архив с docker-образами

На данном шаге необходимо взять из репозитория `https://repo.alteroffice.ru/web/` файл `alterofficeweb-bundle_*.tar.gz` с набором docker-образов для развертывания. Загрузка выполняется на всех узлах (Сервер 1, Сервер 2, Сервер 3, Сервер 4).

Загрузите файл:

```
sudo wget --user=<USER> --password=<USER_PASSWORD>
https://repo.alteroffice.ru/web/alterofficeweb-bundle_<VERSION>.tar.gz
```

> ПРИМЕЧАНИЕ

<VERSION> - версия образа, начинается с v, затем major.minor.patch.build (v2026.0.0.1, v2026.1.2.5 и т.д.).

Если не знаете версию образа, уточните актуальную версию у поставщика.

Пример:

```
sudo wget --user=my_admin_user --password=my_sec_PASSword
https://repo.alteroffice.ru/web/alterofficeweb-bundle_v.2026.0.0.1.tar.gz
```

Архив с docker-образами будет сохранен по пути <DEPLOYMENT_PATH_SERVER>/images/.

Пример:

```
/opt/alterofficeweb/images/
```

4.2.1.2. Подготовка целевых нод

Шаг 1. Настройка целевых нод (Сервер 2, Сервер 3, Сервер 4)

Для корректной работы модулей Ansible на целевых хостах требуется интерпретатор Python 3.9. Действие выполняется на целевых нодах (Сервер 2, Сервер 3, Сервер 4).

Выполните команду:

```
# Установка Python
sudo dnf install -y python3.9
```

4.2.1.3. Подготовка управляющей ноды

Шаг 1. Настройка управляющей ноды

Действия выполняются только на управляющей ноде (Сервер 1).

1. Установка пакетов и управляющего ПО

Выполните команды:

```
# Установка утилиты zip
sudo dnf install zip

# Установка Python и менеджера пакетов pip
sudo dnf install -y python3.9 pip

# Установка Ansible через pip
sudo python3.9 -m pip install ansible==7.7.0
```

ПРИМЕЧАНИЕ

В зависимости от используемой операционной системы на серверах, необходимо установить конкретную версию Ansible для обеспечения совместимости с плейбуками бандла.

2. Создание SSH ключей

Ansible использует SSH для управления узлами. Необходимо сгенерировать ключи (приватный и публичный) на управляющей ноде и распространить их на все целевые хосты.

Выполните команду:

```
sudo ssh-keygen -t ed25519 -f /root/.ssh/ansible -N ""
```

В результате выполнения команды ключи сохранятся в папку `/root/.ssh/` с именами `ansible` (приватный) и `ansible.pub` (публичный).

3. Скопируйте ключи на целевые хосты

```
sudo ssh-copy-id -i /root/.ssh/ansible.pub ansible@<SERVER_1_IP>
sudo ssh-copy-id -i /root/.ssh/ansible.pub ansible@<SERVER_2_IP>
sudo ssh-copy-id -i /root/.ssh/ansible.pub ansible@<SERVER_3_IP>
sudo ssh-copy-id -i /root/.ssh/ansible.pub ansible@<SERVER_4_IP>
```

Пример:

```
sudo ssh-copy-id -i /root/.ssh/ansible.pub ansible@172.20.18.24 # Сервер 1
sudo ssh-copy-id -i /root/.ssh/ansible.pub ansible@172.20.18.25 # Сервер 2
sudo ssh-copy-id -i /root/.ssh/ansible.pub ansible@172.20.18.26 # Сервер 3
sudo ssh-copy-id -i /root/.ssh/ansible.pub ansible@172.20.18.27 # Сервер 4
```

Шаг 2. Загрузка инсталлятора на управляющую ноду

Действия выполняются только на управляющей ноде (Сервер 1).

1. Создайте директорию для работы с установщиком на управляющей ноде

```
sudo mkdir -p <PATH_DEPLOY>
cd <PATH_DEPLOY>
```

Пример:

```
sudo mkdir -p /opt/aow_deploy/
cd /opt/aow_deploy/
```

2. Скачайте установщик

Архив с установщиком содержится в файле `alterofficeweb-installer-ansible_*.zip`

```
sudo wget --user=<USER> --password=<USER_PASSWORD>
https://repo.alteroffice.ru/web/alterofficeweb-installer-
ansible_<VERSION>.zip
```

Пример:

```
# В директорию `/opt/aow_deploy/` будет загружен архив со скриптами
sudo wget --user=<USER> --password=<USER_PASSWORD>
https://repo.alteroffice.ru/web/alterofficeweb-installer-
ansible_v2026.0.0.1.zip
```

3. Распакуйте архив с установщиком в текущую директорию:

```
sudo unzip alterofficeweb-installer-ansible_*.zip
```

Файлы будут разархивированы в папку:

`/opt/aow_deploy/alterofficeweb/ansible/`

4.2.2. Установка на несколько серверов

Действия по установке выполняются на управляющей ноде (Сервер 1).

> ВНИМАНИЕ
>
> **Предварительные требования**
>
> При установке Ansible (например, через pip) исполняемый файл часто попадает в `/usr/local/bin/`. Чтобы скрипт `install.sh` корректно отработал необходимо выполнить:

```
# Проверяем, где находится ansible-playbook
which ansible-playbook

# Пример вывода:
/usr/local/bin/ansible-playbook

# Создаём символическую ссылку
sudo ln -s /usr/local/bin/ansible-playbook /usr/bin/ansible-playbook

# Проверяем, есть ли команда в /usr/bin
ls -la /usr/bin/ansible-playbook

# Пример вывода:
/usr/bin/ansible-playbook -> /usr/local/bin/ansible-playbook
```

Шаг 1. Запуск скрипта инсталлятора

1. Перейдите в папку, куда был разархивирован скрипт установки

```
cd <PATH_DEPLOY>/alterofficeweb/ansible/
```

Пример:

```
cd /opt/aow_deploy/alterofficeweb/ansible/
```

2. Сделайте скрипт установки `./install.sh` исполняемым

```
sudo chmod +x install.sh
```

3. Запустите инсталлятор

```
sudo ./install.sh
```

Шаг 2. Параметры конфигурации

Следуйте инструкциям скрипта для подготовки файла с планом развертывания.

1. Выбор стенда

> ПРИМЕЧАНИЕ
>
> Развертывание может быть выполнено как в виде переустановки системы, так и новая установка системы. В данном примере будет рассмотрена новая установка.
>
> Описание:
>
> - Если у вас уже есть настроенные стенды, они будут показаны списком с номерами.
> - Введите номер стенда из списка для использования существующей конфигурации.
> - Или введите новое имя для создания нового стенда.
>
> Формат имени:
>
> - Только латинские буквы, цифры, дефис, точка, подчеркивание.
> - Примеры: `prod-01`, `test-deploy`, `stag1`, `dev_stand`.

Пример:

```
==========================================
Выбор стенда
==========================================
    dev[1]
    test-deploy[2]

Варианты:
  - Введите номер стенда из списка
  - Или введите новое имя стенда
Стенд (пример: prod-01 или test-deploy): prod-02
```

В качестве имени стенда введите `prod-02`.

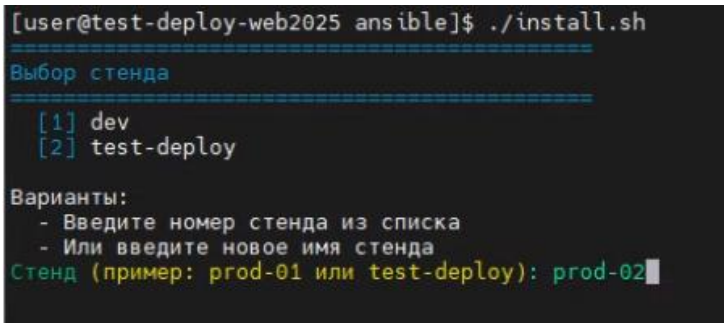


Рисунок 11. Выбор стенда

2. Существующий стенд

Если вы выбрали существующий стенд, инсталлятор спросит нужно ли перенастраивать его конфигурацию.

```
Перенастроить конфигурацию стенда? (yes/no) [no]:
```

Варианты:

- no (по умолчанию) - использовать существующую конфигурацию, сразу перейти к запуску Ansible
- yes - полностью перенастроить стенд (все параметры будут запрошены заново)

Пример:

```
[WARN] Стенд 'prod-01' уже существует

Перенастроить конфигурацию стенда? (yes/no) [no]: no

[INFO] Используется существующая конфигурация
```

3. Сертификаты SSL

Выберите, какие SSL сертификаты будут использоваться: собственные или самоподписанные.

```
Будете ли использовать свои сертификаты? (yes/no) [no]: yes
```

Варианты:

- no (по умолчанию) - будут сгенерированы самоподписанные сертификаты.
- yes - использовать свои сертификаты.

Выберите использование своих сертификатов `yes`.

```
Введите путь к директории с сертификатами (пример: /opt/alterofficeweb/app/ssl
или /etc/ssl/alteroffice) [/opt/alterofficeweb/app/ssl]: /etc/ssl/alteroffice
```

Укажите путь `PATH_SSL`.

Пример:

```
/etc/ssl/alteroffice
```

Рисунок 12. Выбор сертификатов

4. Доменное имя

Введите полное доменное имя (FQDN) для доступа к АльтерОфис Веб (`SERVER_HOSTNAME`).

Пример:

```
Введите доменное имя (FQDN) (пример: test-deploy-web2025.alteroffice.ru):
https://prod-02.alteroffice.ru
```

> ПРИМЕЧАНИЕ
>
> DNS-запись типа A должна указывать на IP адрес узла с HAProxy.

Рисунок 13. Доменное имя

5. Директория для контента

Здесь нужно указать путь до директории (`DEPLOYMENT_PATH_SERVER`) или использовать значение по умолчанию.

Пример:

```
Введите директорию для хранения контента (пример: /opt/alterofficeweb)
[/opt/alterofficeweb]: /opt/alterofficeweb
```

Для использования значения по умолчанию нажмите `Enter`.

Рисунок 14. Директория для контента

6. Версия образа

Здесь нужно указать версию docker-образов АльтерОфис Веб для развертывания.

> ВНИМАНИЕ
>
> Версия образа на данном шаге должна строго совпадать с версией, которая была загружена из репозитория.

Пример:

```
Введите версию образа приложения (пример: v2026.0.0.1): v2026.0.0.2
```

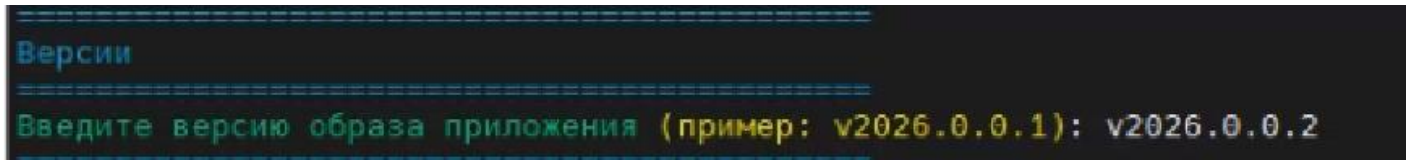


Рисунок 15. Версия образа

7. IP адреса узлов

На данном этапе нужно ввести IP адреса для узлов.

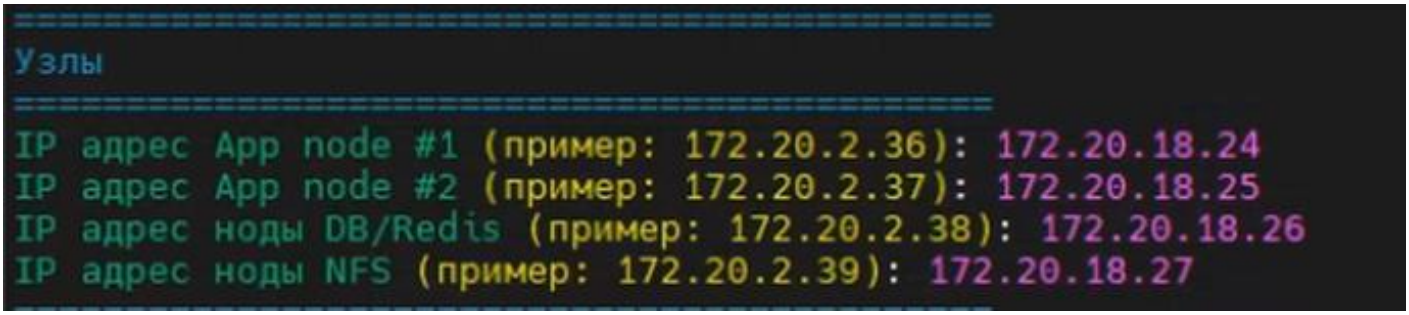


Рисунок 16. IP адреса узлов

- **App node #1**

На этом узле развертываются:

- HAProxy (балансировщик нагрузки).
- Opensearch (полнотекстовый поиск).
- Cron (фоновые задачи).
- Frontend (nginx).
- Editors (онлайн редакторы).
- Docs (документация к системе).

Укажите IP адрес основного узла приложения с HAProxy (точка входа).

> ПРИМЕЧАНИЕ
>
> Требования:
>
> - Валидный IPv4 адрес `<SERVER_1_IP>`.
> - IP должен быть указан в DNS для вашего домена.
> - Доступен по SSH.

Пример:

```
IP адрес App node #1 (пример: 172.20.2.36): 172.20.18.24
```

- **App node #2**

На этом узле развертываются:

- Frontend (nginx).
- Editors (онлайн редакторы).
- Docs (документация к системе).

Укажите IP адрес дополнительного узла приложения (для масштабирования).

> ПРИМЕЧАНИЕ
>
> Требования:
>
> - Валидный IPv4 адрес `<SERVER_3_IP>`.
> - IP должен быть указан в DNS для вашего домена.

- Доступен по SSH.

```
IP адрес App node #2 (пример: 172.20.2.37): 172.20.18.25
```

- **DB/Redis узел**

Укажите IP адрес сервера на котором нужно развернуть PostgreSQL и Redis.

> ПРИМЕЧАНИЕ
>
> Требования:
>
> - Валидный IPv4 адрес `<SERVER_3_IP>`.
> - Доступен по SSH от управляющей ноды.
> - Пользователь `<ANSIBLE_USER>` с правами `sudo`.

Пример:

```
IP адрес ноды DB/Redis (пример: 172.20.2.38): 172.20.18.26
```

- **NFS узел**

Укажите IP адрес сервера для хранения данных (NFS).

> ПРИМЕЧАНИЕ
>
> Требования:
>
> - Валидный IPv4 адрес `<SERVER_4_IP>`.
> - Доступен по SSH с других нод.
> - Достаточно дискового пространства
> - Пользователь `<ANSIBLE_USER>` с правами `sudo`.

Пример:

```
IP адрес ноды NFS (пример: 172.20.2.39): 172.20.18.27
```

8. Статистика HAProxy

Включение веб-интерфейса со статистикой HAProxy.

Рисунок 17. Настройка вывода статистики HAProxy

Пример:

```
Включить вывод статистики HAProxy? (yes/no) [yes]: no
```

Варианты:

- yes (по умолчанию) - статистика будет доступна.
- no - отключить статистику.

Если будет указано yes, то доступ к статистике будет по адресу:

<SERVER_1_IP>:1936

Пример:

172.20.18.24:1936

9. Окно техобслуживания

Укажите время для окна техобслуживания.

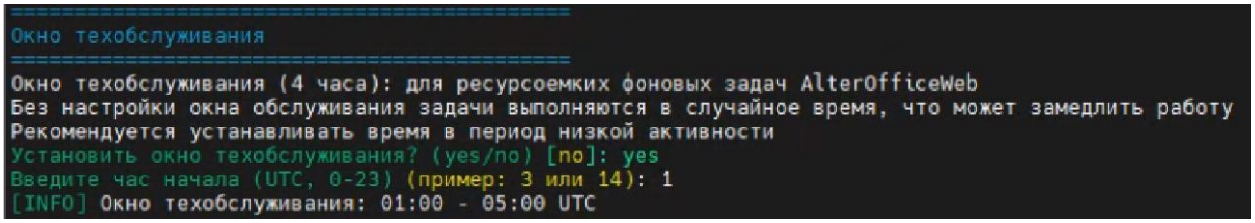


Рисунок 18. Окно техобслуживания

Пример:

```
Окно техобслуживания (4 часа): для ресурсоёмких фоновых задач AlterOfficeWeb.
Без настройки окна обслуживания задачи выполняются в случайное время, что
может замедлить работу.
Рекомендуется устанавливать время в период низкой активности
Установить окно техобслуживания? (yes/no) [no]: yes
```

Варианты:

- yes - можно установить время, когда выполняются задачи техобслуживания (длительность окна техобслуживания 4 часа).
- no(по умолчанию) - не устанавливать период техобслуживания. Задачи будут выполняться в случайное время (без ограничений).

Если выбрали yes:

Пример:

```
Введите час начала (UTC, 0-23) (пример: 3 или 14): 1
[INFO] Окно техобслуживания: 01:00 - 05:00 UTC
```

10. Учетная запись администратора

Укажите имя пользователя для администратора системы АльтерОфис Веб и пароль.

> ПРИМЕЧАНИЕ
>
> Требования:
>
> - Пароль должен соответствовать политике безопасности организации (например: длина не менее 12 символов, наличие букв, цифр и специальных символов).

По умолчанию создается учетная запись admin.

Рисунок 19. Учетная запись администратора

11. Мониторинг и логирование

В данном примере сервисы мониторинга и логирования не устанавливаются.

```
Желаете ли вы настроить мониторинг и логирование? (yes/no) [yes]: no
[INFO] Мониторинг и логирование отключены.
```

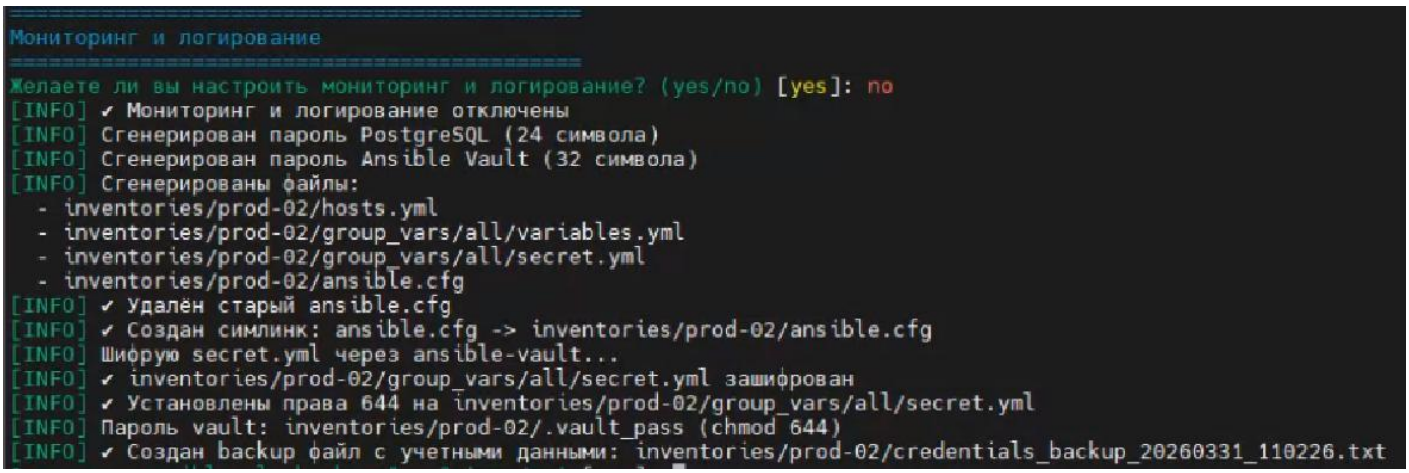


Рисунок 20. Мониторинг и логирование

При необходимости, сервисы мониторинга и логирования могут быть установлены позже.

> СМ. ТАКЖЕ
>
> - Подробнее см. в разделе «Установка сервиса мониторинга и логирования».

12. Завершение сбора параметров

После сбора всех параметров скрипт установки запросит:

```
Запустить ansible-playbook сейчас? (yes/no) [yes]: no
```

Варианты:

- yes (по умолчанию) - сразу начать развертывание.
- no - только сгенерировать конфигурацию без запуска.

Выберите no, если хотите запустить развертывание позднее.

Рисунок 21. Запуск развертывания

Шаг 3. Проверка доступности хостов (опционально)

Перед запуском развертывания можно проверить доступность целевых машин. Данный шаг не обязателен и может быть пропущен.

Выполните проверку подключения (ping) ко всем хостам, указанным в инвентарном файле:

```
ansible all -m ping -i <INVENTORY_DIRECTORY>/hosts.yml  --vault-password-file
<INVENTORY_DIRECTORY>/.vault_pass
```

Пример:

```
ansible all -m ping -i inventories/prod-02/hosts.yml  --vault-password-file
inventories/prod-02/.vault_pass
```

В случае успеха, переходите к следующему шагу.

Шаг 4. Запуск развертывания

Выполните команду для развертывания инфраструктуры:

```
ansible-playbook -i <INVENTORY_DIRECTORY>/hosts.yml playbooks/site.yml -e
customer_type=external  --vault-password-file <INVENTORY_DIRECTORY>/.vault_pass
```

Пример:

```
ansible-playbook -i inventories/prod-02/hosts.yml playbooks/site.yml -e
customer_type=external  --vault-password-file inventories/prod-02/.vault_pass
```

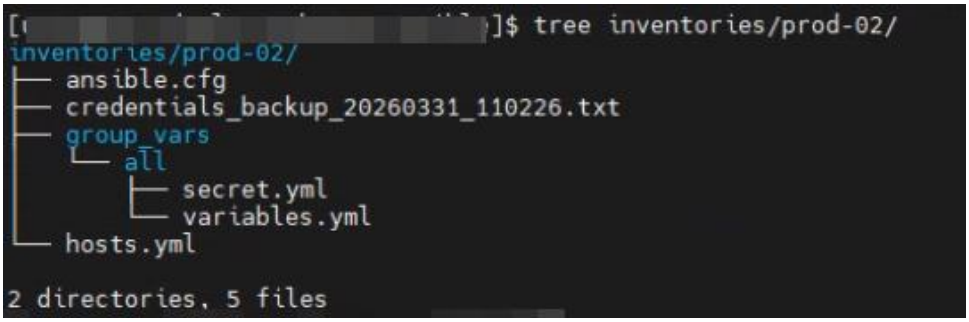


Рисунок 22. Структура директории prod-02

Команда последовательно выполнит:

- Прочитает инвентарный файл, подготовленный на `hosts.yml` из директории `prod-02`, чтобы узнать, на какие серверы нужно устанавливать систему.
- Прочитает пароль `Vault` из файла `.vault_pass` и расшифрует все зашифрованные переменные (например, пароли БД).
- Последовательно выполнит на целевых хостах все этапы, описанные в плане развертывания `site.yml`.

Шаг 5. Проверьте доступность веб-интерфейса.

После завершения развертывания инфраструктуры интерфейс «Альтер Офис Веб» станет доступным по адресу `<SERVER_URL>`. Получить к нему доступ можно через веб браузер, используя имя сервера и данные учётной записи, заданные при установке.

Пример:

```
https://prod-02.alteroffice.ru
```

4.2.3. Установка сервиса мониторинга и логирования

При установке многонодового стенда администратор может настроить мониторинг и логирование для просмотра метрик и логов через графический интерфейс **Grafana** и **OpenSearch Dashboards**.

Установка сервисов мониторинга и логирования может быть выполнена:

- в процессе развертывания системы (см. раздел «Установка на несколько серверов»);
- после основного развертывания системы.

4.2.3.1. Устанавливаемые сервисы для мониторинга и логирования

Общий перечень компонентов, которые разворачиваются для обеспечения сбора, хранения и визуализации данных о состоянии системы приведен ниже.

Экспортеры метрик

Здесь отражены агенты, которые извлекают технические показатели (метрики) из приложений, оборудования или ОС и преобразуют их в формат, понятный Prometheus.

Экспортёр	Назначение	Поток данных	Дашборд
node-exporter	Собирает метрики работы ОС и аппаратных ресурсов (CPU, память, диски, сеть и т.д.)	Node → node-exporter → Prometheus	Node exporter, Main, NFS
editors-online	Собирает метрики работы редакторов (кол-во открытых, активных док-ов, переданного трафика, время открытия документа и т.д.)	editors-online → Prometheus	AlterOffice Web Editors
prometheus-nginxlog-exporter	Собирает метрики: количество запросов, статусы, использование ресурсов и другие характеристики производительности веб-сервера Nginx	Файлы access.log, error.log → prometheus-nginxlog-exporter → Prometheus	NGINX Log Metrics
postgres-exporter	Собирает метрики базы данных	db (pg_stat) → postgres-exporter → Prometheus	PostgreSQL Database
redis_exporter	Собирает метрики Redis	redis → redis_exporter → Prometheus	Redis Dashboard
cadvisor	Собирает, обрабатывает и экспортирует	Node → docker → cAdvisor	cAdvisor

Экспортёр	Назначение	Поток данных	Дашборд
	метрики производительности контейнеров	→ Prometheus	exporter
php_exporter	Собирает метрики php-fpm (для каждого приложения)	app (php-fpm) → php_exporter → Prometheus	php-fpm

Экспортеры логов

Здесь отражены службы, отвечающие за сбор, обогащение и отправку лог-сообщений в централизованное хранилище OpenSearch.

Экспортёр	Назначение	Поток данных
fluent-bit	Осуществляет сбор, обработку и пересылку лог-сообщений в хранилище (OpenSearch)	docker-logs / haproxy-logs / nginx-logs / alteroffice-logs → opensearch

Собираемые логи

В системе собираются логи от сервисов:

Сервис	Лог	Индекс	log_source	env	node_id	attrs.tag
app	/var/log/alterofficeweb/*.log	alterofficeweb-logs-%Y.%m.%d	alteroffice web	app	app1/app2	-
front	/var/log/nginx/*.log	nginx-logs-%Y.%m.%d	nginx	app	app1/app2	-
db	вывод контейнера db	docker-logs-%Y.%m.%d	docker	db	-	db-redis-db-1
nfs	-					
haproxy	/var/log/haproxy/haproxy.log	haproxy-logs-%Y.%m.%d	haproxy	-	-	-
redis	вывод контейнера redis	docker-logs-%Y.%m.%d	docker	db	-	db-redis-redis-1
editors	вывод контейнера editors	docker-logs-%Y.%m.%d	docker	app	app1/app2	alterofficeweb-editors-1
cron	вывод контейнера cron	docker-logs-%Y.%m.%d	docker	app	app1	alterofficeweb-cron-1
app	вывод контейнера app	docker-logs-%Y.%m.%d	docker	app	app1/app2	alterofficeweb-app-1
docs	вывод контейнера docs	docker-logs-	docker	app	app1/app	alterofficew

Сервис	Лог	Индекс	log_source	env	node_id	attrs.tag
		%Y.%m.%d			2	eb-docs-1

Системы мониторинга (сбор и хранение метрик)

Компоненты, которые агрегируют метрики от экспортеров, хранят их временные ряды и предоставляют язык запросов для анализа.

- Grafana — веб-интерфейс панелей управления (дашбордов).
- Prometheus — сбор метрик.

Для отображения метрик подготовлен набор предопределенных дашбордов в Grafana.

Название	Источник данных (Exporter)	Назначение	Отображаемые метрики
Node metrics	node	Отражает потребление ресурсов на конкретном сервере	Memory usage, CPU usage, Disk IO, Network
AlterOffice Web Editors	editors-metrics	Метрики редакторов	Active documents, Load duration
NGINX Log Metrics	nginx	Метрики, собранные на основе логов Nginx	Average response time, Status codes per second, Requests per second
PostgreSQL Database	postgres	Метрики базы данных	Использование соединений, QPS, время выполнения запросов (включая топ-10 самых медленных и частых) и т.д.
Redis Dashboard	redis	Метрики Redis	Max uptime, Clients, Memory usage, Total commands/sec, Hits/Misses per Sec, Total memory ussage и т.д. Всего 13 метрик.
cAdvisor exporter	cAdvisor	Отражает потребление ресурсов по контейнерам	CPU usage, Memory usage, Network sent/received traffic
php-fpm	php-fpm	Метрики PHP	Total process utillisation, Max children reach, Accepted connections, # of pods, Scrape failures, # of prosesses by state и т.д. Всего 22 метрики.
NFS	node	Отражает метрики, специфичные для NFS-ноды	NFS Connections, NFSd Connections, NFS Packets, NFSd

Название	Источник данных (Exporter)	Назначение	Отображаемые метрики
			Packets, NFS RPC, NFSd RPC, NFSd Disk Read/Write и т.д. Всего 27 метрик.
Main	node, nginx	Содержит список всех нод с информацией о ресурсах и список nginx-эндпоинтов с информацией о загрузке	Nodes, Services

Пример отображения данных в Grafana

В разделе представлен внешний вид отображаемых в Grafana дашбордов.

Рисунок 23. Панель мониторинга Node Exporter

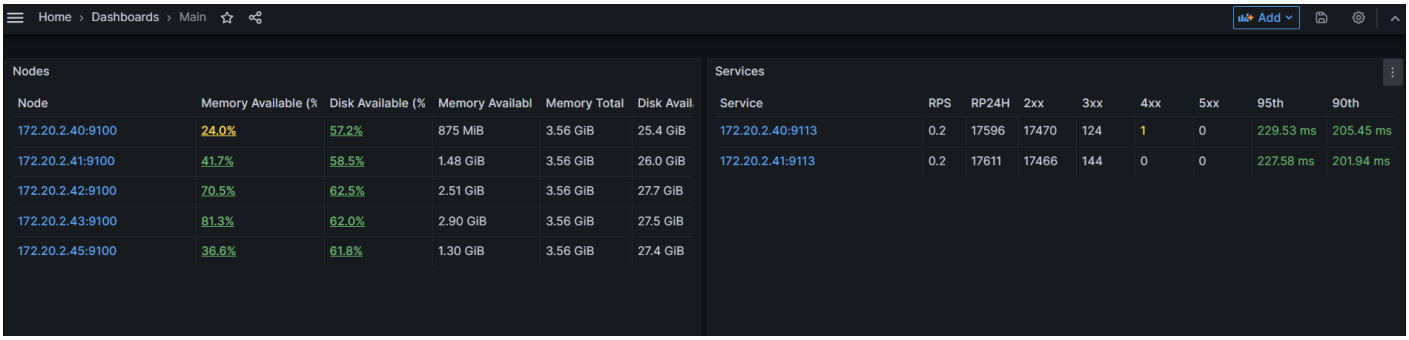


Рисунок 24. Панель мониторинга Main

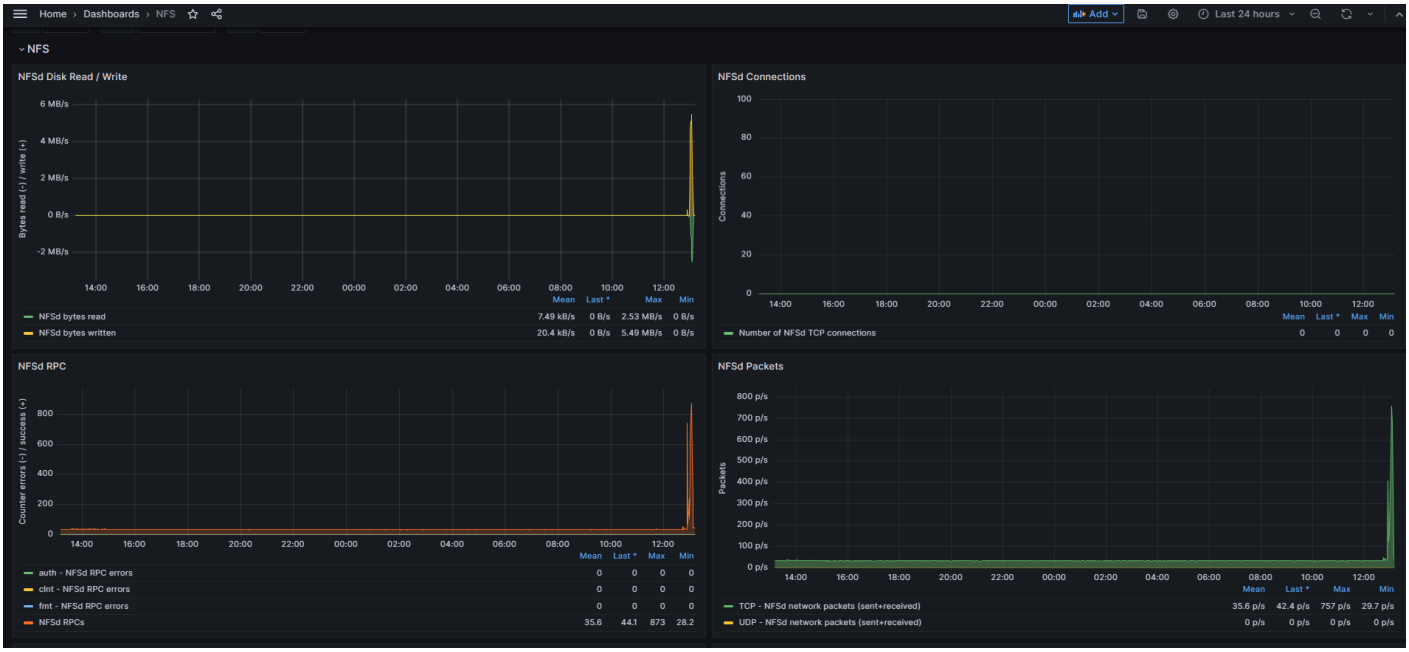


Рисунок 25. Панель мониторинга NFS

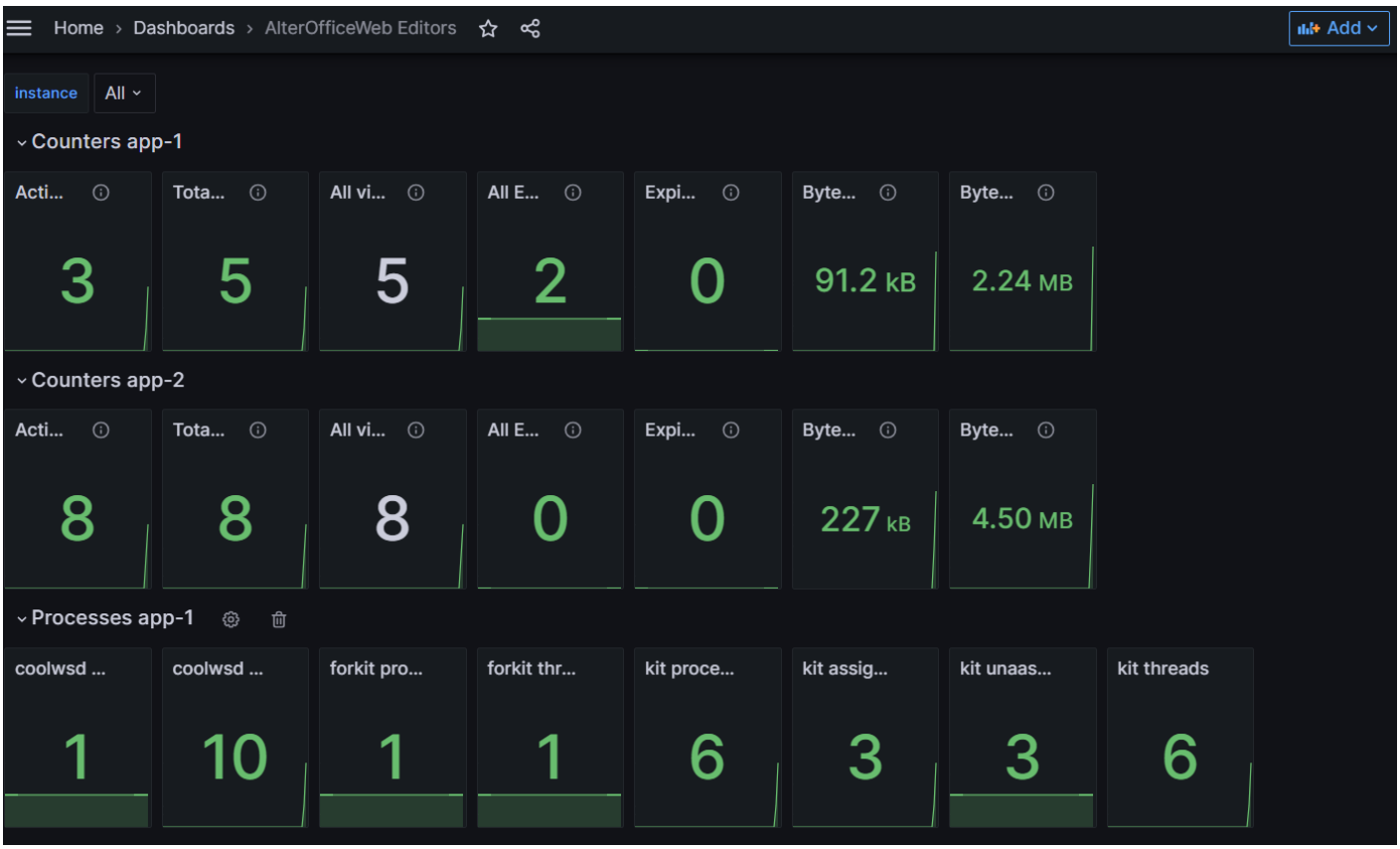


Рисунок 26. Панель мониторинга AlterOfficeWeb Editors

Рисунок 27. Панель мониторинга NGINX Log Metrics

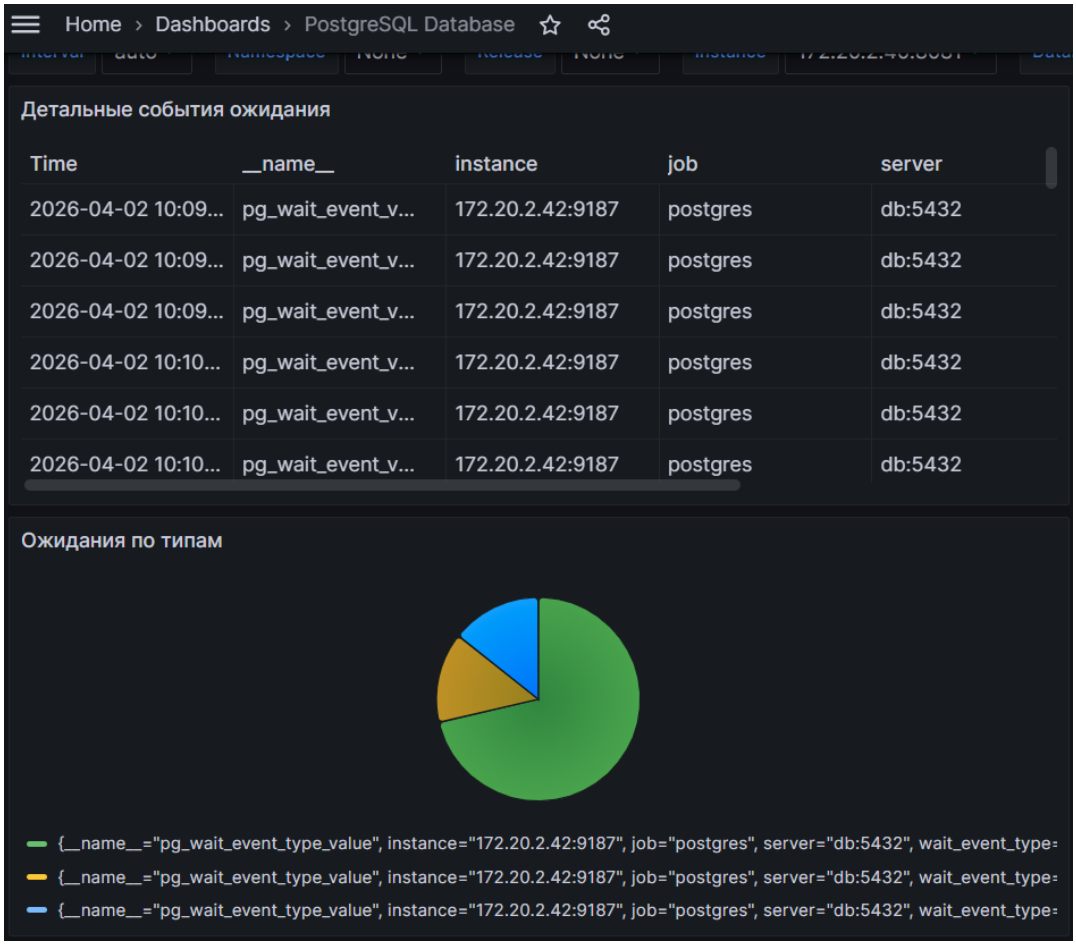


Рисунок 28. Панель мониторинга PostgreSQL Database

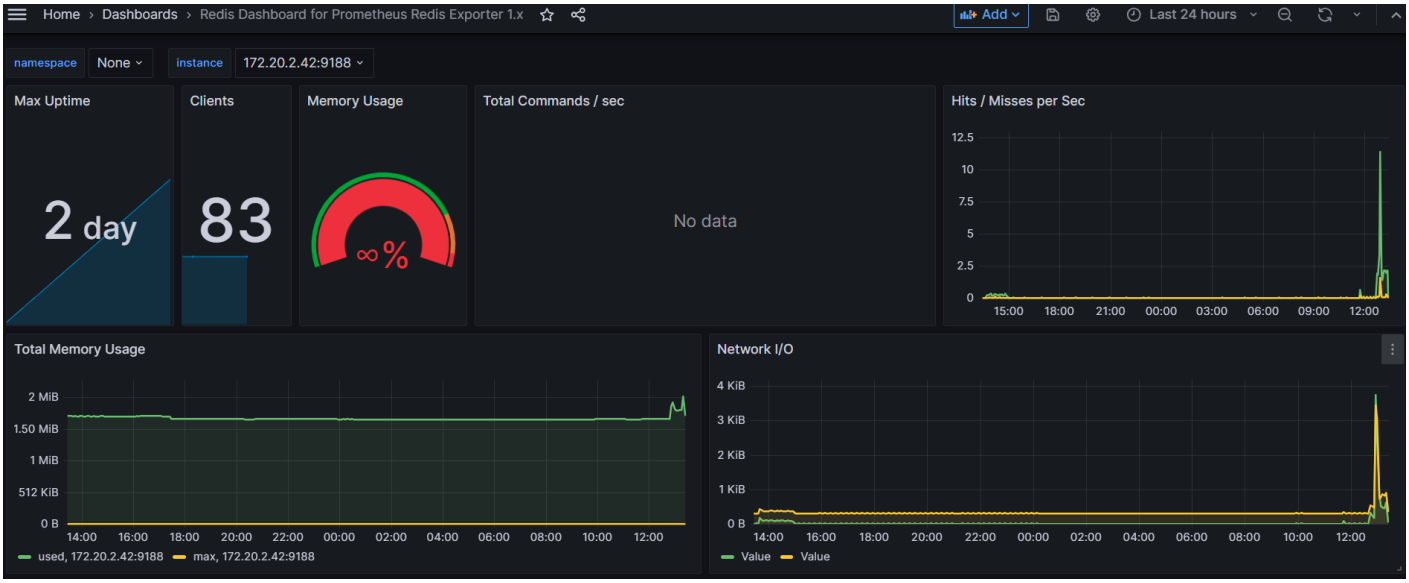


Рисунок 29. Панель мониторинга Redis

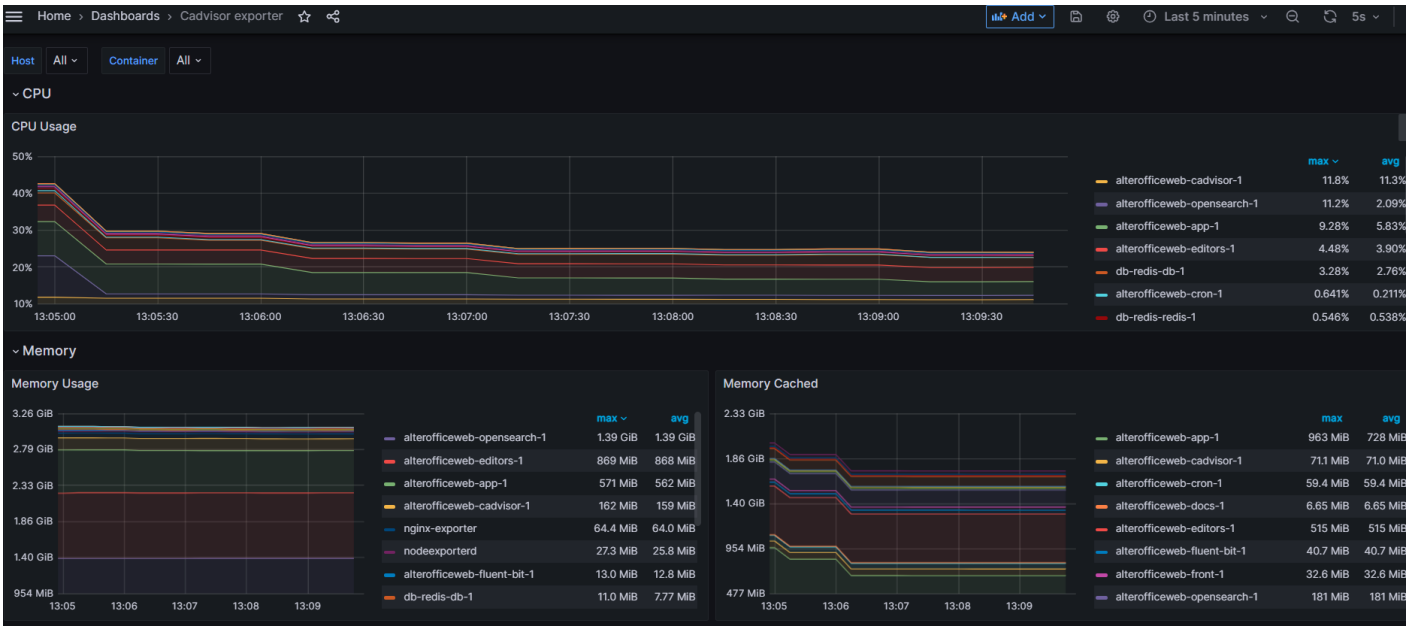


Рисунок 30. Панель мониторинга cAdvisor exporter

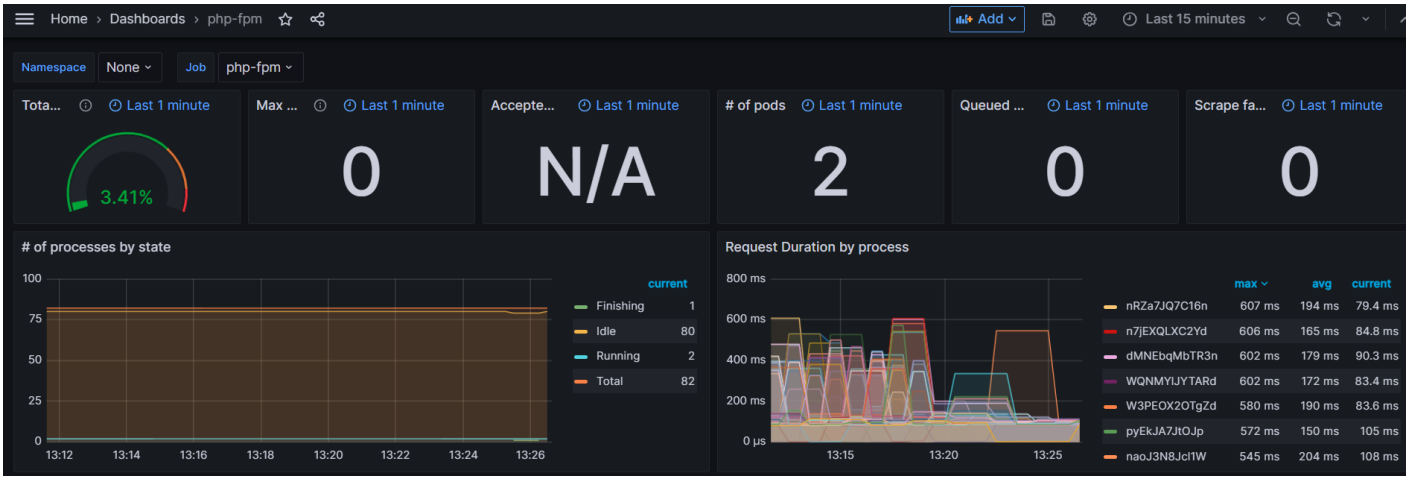


Рисунок 31. Панель мониторинга php-fpm

Система логирования

- OpenSearch — сбор логов, централизованное хранение лог-сообщений, обеспечивает быстрый поиск и фильтрацию.
- OpenSearch Dashboards — дашборды логов, предоставляет интерфейс для визуализации данных, хранимых в OpenSearch.

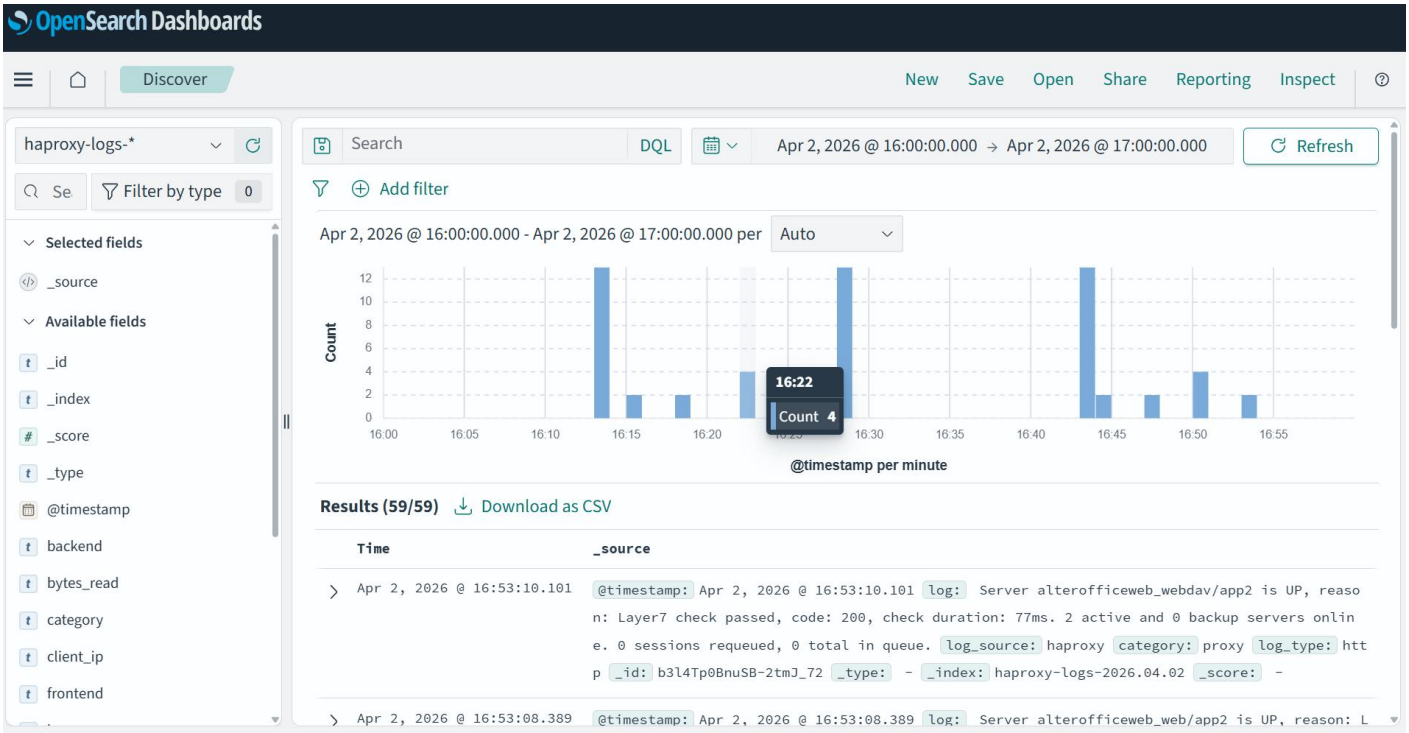


Рисунок 32. Дашборд haproxy-logs в OpenSearch

4.2.3.2. Установка сервиса мониторинга и логирования в процессе основного развертывания системы

Пример распределения сервисов по нодам для многонодового развертывания

IP	Основные сервисы	Сервисы мониторинга

IP	Основные сервисы	Сервисы мониторинга
172.20.18.24	HAProxy, front, app, editors, cron, opensearch, docs	fluent-bit, cadvisor, php_exporter, prometheus-nginxlog-exporter, node-exporter
172.20.18.25	front, app, editors, docs	fluent-bit, cadvisor, php_exporter, prometheus-nginxlog-exporter, node-exporter
172.20.18.26	db, redis	fluent-bit, cadvisor, redis_exporter, node-exporter, postgres-exporter
172.20.18.27	nfs	node-exporter
172.20.18.28		opensearch-dashboards, opensearch, node-exporter, grafana, prometheus

Шаг 1. Подготовка плана развертывания

При подготовке плана развертывания (этап сбора параметров для установки) на вопрос «Желаете ли вы настроить мониторинг и логирование?» ответьте `yes`.

> СМ. ТАКЖЕ
>
> - Подробнее см. в разделе «Установка на несколько серверов».

Пример:

```
Желаете ли вы настроить мониторинг и логирование? (yes/no) [yes]: yes
[INFO] Мониторинг и логирование будут включены.
```

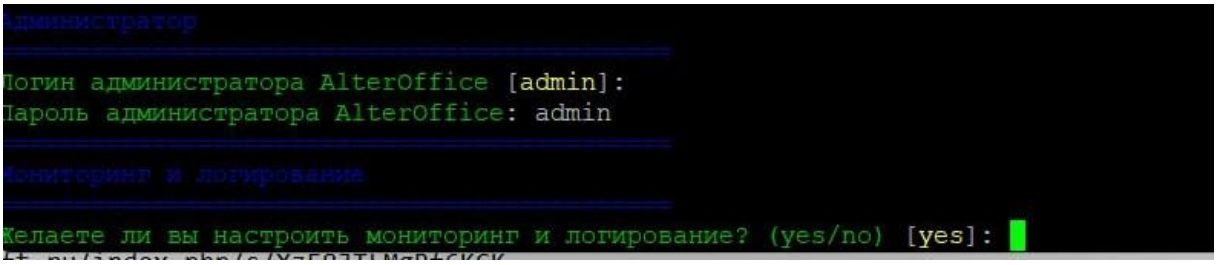


Рисунок 33. Запрос настройки мониторинга и логирования

Шаг 2. IP-адрес узла для развертывания контейнеров мониторинга и логирования

На данном этапе нужно ввести IP адрес для узла, куда будут установлены контейнеры для сбора и отображения метрик и логов.

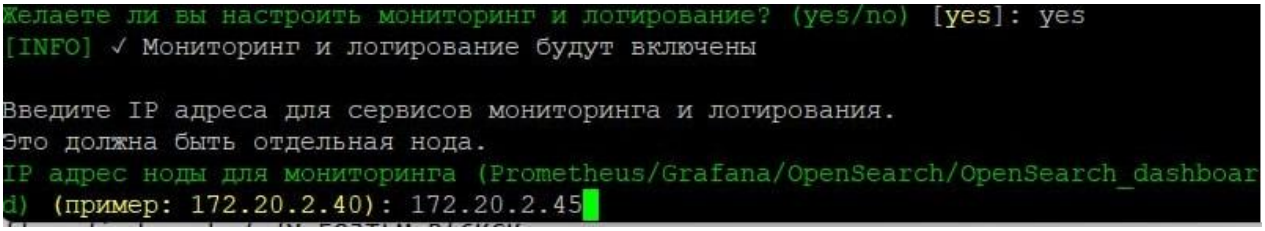


Рисунок 34. Ввод IP-адресов сервера для установки контейнера сбора и отображения метрик и логов

Шаг 3. Запуск развертывания

После выполнения подготовки плана развертывания, появится запрос на подтверждение установки системы:

```
Запустить ansible-playbook сейчас? (yes/no) [yes]: yes
```

Ответьте `yes` для запуска установки системы **АльтерОфис Веб**.

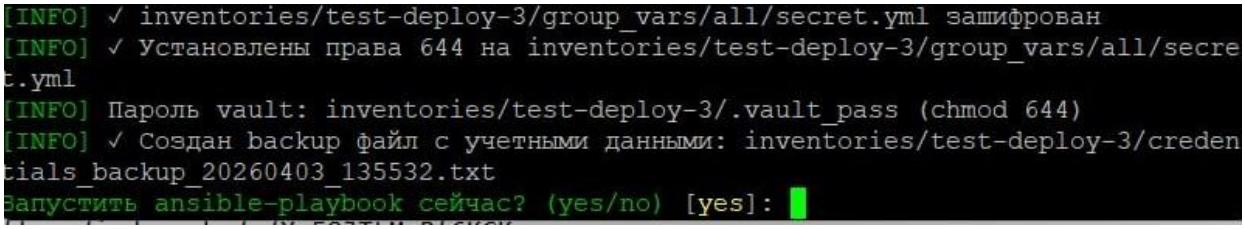


Рисунок 35. Запрос на запуск ansible-playbook

Шаг 4. Запуск установки сервисов мониторинга и логирования

На запрос:

```
Запустить установку мониторинга и логирования сейчас? (yes/no) [yes]: yes
```

Ответьте `yes`. Скрипт автоматически установит сервисы.

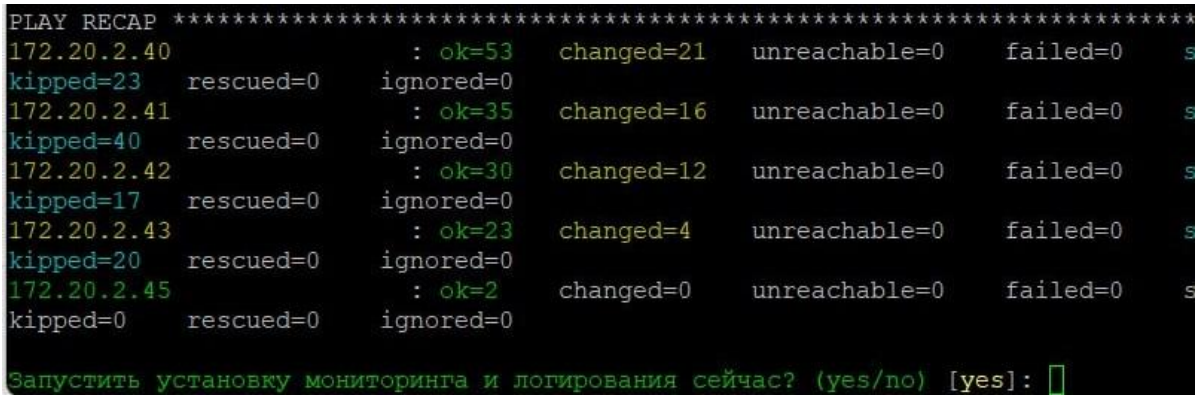


Рисунок 36. Запрос на установку мониторинга и логирования

Шаг 5. Завершение установки

1. Проверьте доступ к установленным сервисам:

Сервисы будут доступны по адресам:

```
# Grafana
<SERVER_5_IP>:3000
# OpenSearch Dashboards
<SERVER_5_IP>:5601
```

Пример:

```
172.20.18.28:3000
172.20.18.28:5601
```

2. Сохраните резервную копию конфигурационных данных

После завершения работы скрипта, будет сформирован файл с конфигурационными данными.

Он находится в папке стенда по следующему адресу:

```
ansible/<INVENTORY_DIRECTORY>/credentials_backup_*
```

Пример:

```
ansible/inventories/prod-02/credentials_backup_20260331_110226.txt
```

Сохраните резервную копию конфигурационного файла в надежном месте.

4.2.3.3. Установка сервиса мониторинга и логирования после основного развертывания системы

Если при основном развертывании системы вы отказались от установки сервисов мониторинга и логирования, то для ручной установки нужно выполнить команду:

```
ansible-playbook -i <INVENTORY_DIRECTORY>/hosts.yml playbooks/monitoring.yml
playbooks/logging.yml playbooks/exporters.yml --vault-password-file
<INVENTORY_DIRECTORY>/.vault_pass
```

Пример:

```
ansible-playbook -i inventories/prod-02/hosts.yml playbooks/monitoring.yml
playbooks/logging.yml playbooks/exporters.yml --vault-password-file
inventories/prod-02/.vault_pass
```

5.Настройка системы

5.1. Настройка системы для работы с макросами

По умолчанию макросы в редакторах отключены в целях безопасности.

Работа с макросами не является обязательным при установке системы и настраивается при необходимости. Настройки выполняются через командную строку (CLI) и через графический веб-интерфейс (WebUI).

5.1.1. Порядок выполнения настроек для работы с макросами

Для успешной настройки рекомендуется соблюдать следующий порядок:

Настройка	**Вариант настройки**
Администратор системы настраивает контейнеры **app** и **editors** для работы с одним **VOLUME**, в который будут размещаться макросы.	CLI
Администратор системы подключает внешнее хранилище для размещения в нем макросов приложения через WebUI и которые будут доступны пользователям.	WebUI
Администратор системы создает новые макросы или загружает ранее созданные макросы в хранилище макросов.	WebUI
Пользователи **АльтерОфис Веб** используют настроенные администратором системы макросы для своей работы в офисных редакторах (**АТекст**, **АТаблица**, **АКонцепт**).	WebUI

- Администратор системы может настроить ограниченные права доступа к системе, достаточные для создания и размещения макросов приложения, и предоставить их пользователю, который будет отвечать за макросы.

Выполнение настроек для работы с макросами в системе **АльтерОфис Веб** будет рассмотрено на примере:

- **Экземпляр** имеет адрес `https://demo03-web2025.alteroffice.ru`

Параметры, которые потребуются для настройки:

Параметр	**Описание**	**Пример**
SERVER_URL	URL для доступа к системе **АльтерОфис Веб**	`https://demo03-web2025.alteroffice.ru`
PATH_EDITORS	Путь к папке в контейнере **editors**, где будут размещаться макросы приложения	`/opt/lokit/share/Scripts/python`
PATH_APP	Путь к папке в контейнере **app**, где будут размещаться макросы	`/var/www/html/data/Scripts`

Параметр	Описание	Пример
	приложения	
PATH_VOLUME	Путь к **VOLUME** на хосте, где будут размещаться макросы приложения	`/opt/alterofficeweb/demo03/html/data/Scripts`
PATH_SERVER	Путь к каталогу, где установлена система.	`/opt/alterofficeweb/demo03`

5.1.2. Подготовка yaml-файлов

Для корректной совместной работы контейнеров **app** и **editors** с макросами, необходимо настроить общий том, смонтировав локальную директорию хоста в соответствующие пути внутри каждого контейнера.

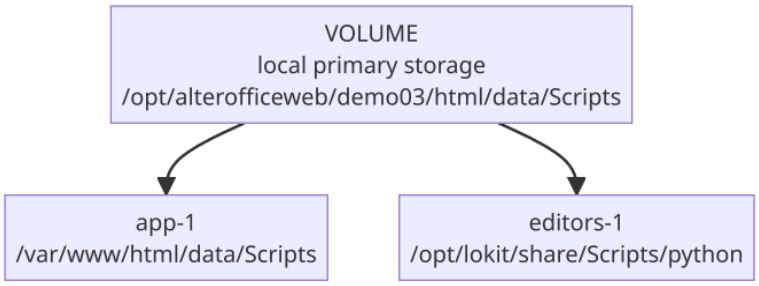


Рисунок 37. Схема общего доступа к макросам через хост

Шаг 1. Перейдите в каталог, где установлена система.

```
cd <PATH_SERVER>
```

Пример:

```
cd /opt/alterofficeweb/demo03
```

Шаг 2. Подготовьте обновленный **compose.yaml** файл для контейнера **app** настраиваемого стенда (например, **demo03**).

Откройте файл на редактирование:

```
nano compose.yaml
```

Внесите изменения.

```
 redis:
    ...

  app:
    image: ...
    restart: always
    volumes:
      - ...
      - <PATH_VOLUME>:<PATH_APP>
    ...
```

Замените:

- <PATH_VOLUME> на путь к директории на хосте, где будут размещаться макросы;
- <PATH_APP> на путь в контейнере **app**, где будут размещаться макросы.

Пример:

```
 redis:
    ...

  app:
    image: ...
    restart: always
    volumes:
      - ...
      - /opt/alterofficeweb/dev02/html/data/Scripts:/var/www/html/data/Scripts
    ...
```

Сохраните изменения и выйдите из редактора.

Шаг 3. Подготовьте обновленный **compose-editors.yaml** файл для контейнера **editors** настраиваемого стенда (например, **demo03**).

Откройте файл на редактирование:

```
nano compose-editors.yaml
```

Внесите изменения.

```
editors:
    image: ...
    restart: always
    volumes:
      - <PATH_VOLUME>:<PATH_EDITORS>
    ...
```

Замените:

- <PATH_VOLUME> на путь к директории на хосте, где будут размещаться макросы;
- <PATH_EDITORS> на путь в контейнере **editors**, где будут размещаться макросы.

Пример:

```
editors:
    image: ...
    restart: always
    volumes:
      - /opt/alterofficeweb/demo03/html/data/Scripts:/opt/lokit/share/Scripts/python
    ...
```

Сохраните изменения и выйдите из редактора.

5.1.3. Смена владельца для VOLUME

Чтобы администратор системы мог размещать макросы в системе, необходимо корректно настроить владельца для созданной папки на хосте (**VOLUME**):

```
chown -R <пользователь><:группа> <PATH_VOLUME>
```

Пример:

```
chown -R 33:33 /opt/alterofficeweb/demo03/html/data/Scripts
```

5.1.4. Включение макросов

По умолчанию в редакторах макросы отключены. Чтобы их включить, нужно настроить файл `coolwsd.xml`, включив макросы и изменив уровень безопасности:

```
<enable_macros_execution desc="Specifies whether the macro execution is enabled
in general. This will enable Basic, Beanshell, Javascript and Python scripts.
If it is set to false, the macro_security_level is ignored. If it is set to
true, the mentioned entry specified the level of macro security." type="bool"
default="false">true</enable_macros_execution>
```

Чтобы включить выполнение макросов, необходимо изменить значение параметра **enable_macros_execution** с `false` на `true`.

Также, необходимо выбрать уровень безопасности, которому следует придерживаться при выполнения макроса.

Существует два уровня:

- 0 (Низкий, не рекомендуется) - Все макросы будут выполняться без подтверждения.
- 1 (Средний, по умолчанию) - Требуется подтверждение перед выполнением макросов из ненадежных источников.

```
<macro_security_level desc="Level of Macro security. 1 (Medium) Confirmation
required before executing macros from untrusted sources. 0 (Low, not
recommended) All macros will be executed without confirmation." type="int"
default="1">1</macro_security_level>
```

Шаг 1. Перейдите в каталог, где установлена система.

```
cd <PATH_SERVER>
```

Пример:

```
cd /opt/alterofficeweb/demo03
```

Шаг 2. Откройте файл `.env-editors` на редактирование.

В файле **.env-editors** настраиваются переменные окружения, которые прокидываются в файл `coolwsd.xml` контейнера **editors**.

Пример:

```
nano /opt/alterofficeweb/demo03/.env-editors
```

Шаг 3. Добавьте параметры `--o:security.enable_macros_execution=true --o:security.macro_security_level=0` в конце существующих настроек.

Пример:

```
domain=demo03-web2025.alteroffice.ru
extra_params=--o:ssl.enable=true --o:ssl.termination=false --o:net.lok_allow.host[14]=demo03-web2025.alteroffice.ru --o:security.enable_macros_execution=true --o:security.macro_security_level=0
```

5.1.5. Применение внесенных изменений

Шаг 1. Перезапустите контейнер **app**

```
docker compose restart app
```

Шаг 2. Примените настройки к контейнеру **editors** и проверьте применение параметров

```
docker compose -f compose-editors.yaml down
```

Команда очистит окружение (остановит и удалит запущенные контейнеры, описанные в **compose-editors.yaml**).

```
docker compose -f compose-editors.yaml up -d
```

Команда запустит все сервисы, описанные в файле **compose-editors.yaml**, в фоновом режиме (демоне).

Шаг 3. Проверьте, что переменные корректно передались в контейнер:

```
docker exec demo03-editors-1 printenv
```

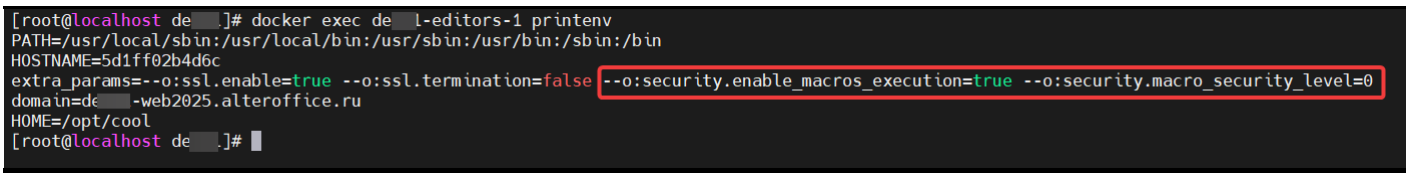


Рисунок 38. Проверка настройки переменных окружения

Шаг 4. Проверьте доступность кнопки «Выполнить макрос» в онлайн-редакторах.

После применения обновленных настроек файла **coolwsd.xml** проверьте, что в интерфейсе появилась кнопка «Выполнить макрос».

Для проверки, зайдите в АльтерОфис Веб, откройте АТаблица.

Убедитесь, что на ленте на вкладке «Файл» появилась кнопка «Выполнить макрос».

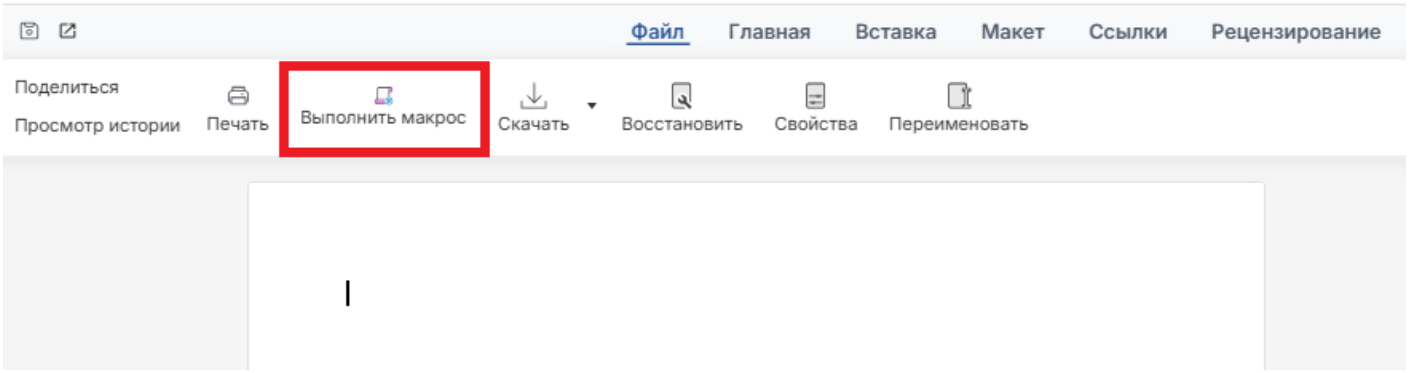


Рисунок 39. Кнопка «Выполнить макрос» активна.

5.1.6. Загрузка и создание макросов в АльтерОфис Веб

Дальнейшая настройка АльтерОфис Веб для работы с макросами выполняется через веб-интерфейс.

Описание шагов по дальнейшей настройке системы с использованием веб-интерфейса описаны в документе «**Руководство администратора по настройке и администрированию АльтерОфис Веб**»:

- Подключение внешнего хранилища для работы с макросами.
- Загрузка и создание макросов.
- Работа пользователей с макросами приложения.

5.2. Настройка федеративного обмена между серверами

Между несколькими экземплярами АльтерОфис Веб может быть настроен федеративный доступ, реализованный на протоколе Open Cloud Mesh (OCM).

Каждый экземпляр функционирует автономно, но при необходимости пользователи могут обмениваться файлами и работать над документами совместно, как если бы они находились на одном сервере.

Федеративный доступ не является обязательным при установке системы и настраивается при необходимости. Настройки выполняются через командную строку (CLI) и через графический веб-интерфейс (WebUI).

В данном разделе описывается часть по настройкам с использованием CLI, настройки с использованием WebUI описаны в документе «**Руководство администратора по настройке и администрированию АльтерОфис Веб**».

Настройка федеративного обмена файлами между двумя серверами АльтерОфис Веб, развернутыми в Docker-контейнерах, будет рассмотрена на примере двух экземпляров:

- **Экземпляр 1** имеет адрес `https://demo03-web2025.alteroffice.ru`
- **Экземпляр 2** имеет адрес `https://demo04-web2025.alteroffice.ru`

Параметры, которые потребуются для настройки:

Параметр	Описание	Пример
SOURCE_SERVER_URL	URL исходного сервера (источник файлов), **Экземпляр 1**	`https://demo03-web2025.alteroffice.ru`
DESTINATION_SERVER_URL	URL целевого сервера (получатель файлов), **Экземпляр 2**	`https://demo04-web2025.alteroffice.ru`
SOURCE_SERVER_NAME	Название директории установки АльтерОфис Веб на исходном сервере, **Экземпляр 1**	`demo03`
DESTINATION_SERVER_NAME	Название директории установки АльтерОфис Веб на целевом сервере, **Экземпляр 2**	`demo04`
SOURCE_SERVER_HOSTNAME	Доменное имя или FQDN (Fully Qualified Domain Name) исходного сервера АльтерОфис Веб, с которого будут предоставляться общие папки и файлы для федеративного обмена.	`demo03-web2025.alteroffice.ru`
DESTINATION_SERVER_HOSTNAME	Доменное имя или FQDN целевого сервера, на котором пользователи будут получать доступ к общим ресурсам с исходного сервера.	`demo04-web2025.alteroffice.ru`
SOURCE_SERVER_IP	IP-адрес хоста исходного сервера, где установлен контейнер **app** для **Экземпляр 1**	`172.20.18.10`
DESTINATION_SERVER_IP	— IP-адрес хоста целевого сервера, где установлен контейнер **app** для **Экземпляр 2**	`172.45.58.20`
DEPLOYMENT_PATH_SOURCE_SERVER	Путь к директории развертывания на хосте (путь к файлу `compose.yaml` для **Экземпляр 1**)	`/opt/alterofficeweb/demo03`
DEPLOYMENT_PATH_DESTINATION_SERVER	Путь к директории развертывания на хосте (путь к файлу `compose.yaml` для **Экземпляр 2**)	`/opt/alterofficeweb/demo04`

Параметр	Описание	Пример
SERVICE_USER_SOURCE_SERVER	Пользователь для развертывания и управления контейнерами **Экземпляр 1**	`gitlab-runner`
SERVICE_USER_DESTINATION_SERVER	Пользователь для развертывания и управления контейнерами **Экземпляр 2**	`gitlab-runner`

5.2.1. Порядок выполнения операций для настройки федеративного доступа

Для успешной настройки рекомендуется соблюдать следующий порядок:

Настройка	Вариант настройки
Настройка сетевого взаимодействия	CLI
Проверка доступности	CLI
Активация приложения Federation	CLI
Разрешение доступа для изолированных серверов	CLI
Установка базовых URL	CLI
Настройка «белого» списка IP адресов	WebUI
Настройка межсерверного обмена для пользователей	WebUI
Настройка доверенных серверов	WebUI
Синхронизация адресных книг для федеративного доступа	CLI
Настройка редакторов для совместной работы при федеративном доступе	CLI

5.2.2. Настройка сетевого взаимодействия

Для обеспечения корректного разрешения имен между серверами необходимо добавить статические записи в файл хостов внутри каждого контейнера.

На исходном сервере (SOURCE_SERVER_URL):

Шаг 1. Перейдите в директорию развертывания:

```
cd <DEPLOYMENT_PATH_SOURCE_SERVER>/<SOURCE_SERVER_NAME>
```

Пример:

```
cd /opt/alterofficeweb/demo03
```

Шаг 2. Отредактируйте файл `compose.yaml`:

```
sudo -u <SERVICE_USER_SOURCE_SERVER> nano compose.yaml
```

Пример:

```
sudo -u gitlab-runner nano compose.yaml
```

Шаг 3. В секции сервиса `app` добавьте или измените блок `extra_hosts`:

```
services:
  app:
    # ... существующие настройки
    extra_hosts:
      - "<SOURCE_SERVER_HOSTNAME>:<SOURCE_SERVER_IP>"
      - "<DESTINATION_SERVER_HOSTNAME>:<DESTINATION_SERVER_IP>"
```

Пример:

```
extra_hosts:
  - "demo03-web2025.alteroffice.ru:172.20.18.10"
  - "demo04-web2025.alteroffice.ru:172.45.58.20"
```

ПРИМЕЧАНИЕ

- При редактировании файла `compose.yaml` не забывайте про требования к оформлению YAML-файлов:
  - Для отступов необходимо использовать пробелы, а не табуляции.
  - Обычно на каждый уровень отступа приходится 2 пробела.

Шаг 4. Сохраните изменения и перезапустите контейнер:

```
docker compose restart app
```

На целевом сервере (DESTINATION_SERVER_URL):

Повторите те же шаги, указав соответствующие параметры:

```
extra_hosts:
  - "<DESTINATION_SERVER_HOSTNAME>:<DESTINATION_SERVER_IP>"
  - "<SOURCE_SERVER_HOSTNAME>:<SOURCE_SERVER_IP>"
```

Пример:

```
extra_hosts:
  - "demo04-web2025.alteroffice.ru:172.45.58.20"
  - "demo03-web2025.alteroffice.ru:172.20.18.10"
```

5.2.3. Проверка сетевой доступности

После настройки DNS-записей необходимо убедиться, что серверы могут взаимодействовать друг с другом без SSL-ошибок.

Проверка с исходного сервера

Выполните команду для проверки доступности целевого сервера:

```
docker exec -it <SOURCE_CONTAINER_NAME> curl
<DESTINATION_SERVER_URL>/status.php
```

Пример:

```
docker exec -it demo03-app-1 curl https://demo04-
web2025.alteroffice.ru/status.php
```

> ПРИМЕЧАНИЕ
>
> - Команда `curl` выполняется внутри контейнера **app** без входа в сам контейнер.

Ожидаемый результат:


```
{"installed":true,"maintenance":false,"needsDbUpgrade":false,"version":"31.0.4.1722
7","versionstring":"2025.0.0.9","edition":"","productname":"AlterOffice","extende
dSupport":false}
```


Проверка с целевого сервера

Аналогично проверьте доступность исходного сервера:

```
docker exec -it <DESTINATION_CONTAINER_NAME> curl
<SOURCE_SERVER_URL>/status.php
```

Пример:

```
docker exec -it demo04-app-1 curl https://demo03-web2025.office.ru/status.php
```

Если возникают SSL-ошибки, убедитесь, что сертификаты на обоих серверах корректно настроены и доверены.

5.2.4. Активация приложения Federation

Для работы федеративного обмена необходимо активировать соответствующее приложение на обоих серверах.

На каждом сервере выполните:

```
docker exec -it <CONTAINER_NAME> php /var/www/html/occ app:enable federation
```

Пример для исходного сервера:

```
docker exec -it demo03-app-1 php /var/www/html/occ app:enable federation
```

Пример для целевого сервера:

```
docker exec -it demo04-app-1 php /var/www/html/occ app:enable federation
```

Возможные результаты выполнения:

При успешной активации:

- federation 1.21.0 enabled

Если приложение уже активировано:

- federation already enabled

Использование WebUI для управления приложениями см. в документе «**Руководство администратора по настройке и администрированию АльтерОфис Веб**».

5.2.5. Настройка доступа для изолированных серверов

По умолчанию АльтерОфис Веб разрешает федеративный обмен только с серверами, имеющими валидные публичные SSL-сертификаты. Для работы в изолированном контуре необходимо разрешить обмен с локальными серверами.

На каждом сервере выполните:

```
docker exec -it <CONTAINER_NAME> php /var/www/html/occ config:system:set
allow_local_remote_servers --value=true
```

Пример для исходного сервера:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:system:set
allow_local_remote_servers --value=true
```

Пример для целевого сервера:

```
docker exec -it demo04-app-1 php /var/www/html/occ config:system:set
allow_local_remote_servers --value=true
```

Результат выполнения:

```
System config value allow_local_remote_servers set to boolean true
```

5.2.6. Установка базового URL-адреса

Для корректной генерации абсолютных ссылок в системе (например, в уведомлениях по электронной почте или для внешних пользователей) необходимо явно задать базовый URL каждого сервера.

На исходном сервере выполните:

```
docker exec -it <SOURCE_CONTAINER_NAME> php /var/www/html/occ
config:system:set overwrite.cli.url --value='<SOURCE_SERVER_URL>'
```

Пример:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:system:set
overwrite.cli.url --value='https://demo03-web2025.office.ru'
```

На целевом сервере выполните:

```
docker exec -it <DESTINATION_CONTAINER_NAME> php /var/www/html/occ
config:system:set overwrite.cli.url --value='<DESTINATION_SERVER_URL>'
```

Пример:

```
docker exec -it demo04-app-1 php /var/www/html/occ config:system:set
overwrite.cli.url --value='https://demo04-web2025.office.ru'
```

Результат выполнения:

```
System config value overwrite.cli.url set to string <URL>
```

Параметр будет добавлен в файл конфигурации `config/config.php`:

```
'overwrite.cli.url' => '<URL>',
```

5.2.7. Настройка «белого» списка IP адресов

Для корректной работы серверов при организации федеративного доступа, необходимо IP адреса серверов добавить в «белый список».

Настройка «белого списка» выполняется через веб-интерфейс и описано в документе «**Руководство администратора по настройке и администрированию АльтерОфис Веб**».

5.2.8. Настройка межсерверного обмена для пользователей

Активируйте необходимые опции:

Параметр	Назначение	Примечание
Разрешить пользователям на этом сервере публиковать общие ресурсы на других серверах (этот параметр также разрешает доступ WebDAV к общим папкам)	Разрешает пользователям данного сервера предоставлять доступ к своим файлам и папкам пользователям на других (внешних) серверах.	Используйте, если требуется функционал совместной работы с внешними организациями или другими экземплярами АльтерОфис Веб.
Разрешить пользователям этого сервера принимать общие ресурсы с других серверов	Разрешает пользователям данного сервера получать и принимать файлы и папки, которыми с ними поделились пользователи внешних серверов.	Включите для полноценного двустороннего обмена. Обычно активируется вместе с предыдущей опцией.
Разрешить пользователям этого сервера предоставлять общий доступ группам пользователей других серверов	Позволяет пользователям этого сервера предоставлять доступ целым группам пользователей, существующим на внешнем сервере. Удобно для совместной работы с отделами или командами в другой организации.	Экспериментальная функция.
Разрешить пользователям этого сервера принимать общие ресурсы с других серверов, опубликованные	Разрешает группам пользователей на этом сервере получать общие ресурсы от пользователей внешних	Экспериментальная функция.

Параметр	Назначение	Примечание
для групп пользователей	серверов. Все члены группы получат доступ к присланным файлам/папкам.	
По умолчанию автоматически принимать общие ресурсы от доверенных федеративных учетных записей и групп	Автоматически принимает входящие общие ресурсы от серверов и групп, добавленных в «доверенные», без необходимости ручного подтверждения каждым пользователем.	Включайте только для серверов-партнёров с высоким уровнем доверия (например, внутри холдинга).

На исходном сервере

Выполните настройка на исходном сервере.

- **1. Включение опции «Разрешить пользователям на этом сервере публиковать общие ресурсы на других серверах (этот параметр также разрешает доступ WebDAV к общим папкам)»**

Опция **позволяет разрешать пользователям отправлять файлы на другие серверы**.

Проверьте текущий статус:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:get
files_sharing outgoing_server2server_share_enabled
```

Для выключения выполните:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing outgoing_server2server_share_enabled --value="no"
```

Для включения:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing outgoing_server2server_share_enabled --value="yes"
```

- **2. Включение опции «Разрешить пользователям этого сервера принимать общие ресурсы с других серверов»**

Опция **позволяет пользователям принимать файлы с других серверов**.

Проверить текущий статус:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:get
files_sharing incoming_server2server_share_enabled
```

Выключить:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing incoming_server2server_share_enabled --value="no"
```

Включить:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing incoming_server2server_share_enabled --value="yes"
```

- **3. Включение опции «Разрешить пользователям этого сервера предоставлять общий доступ группам пользователей других серверов»**

Экспериментальная функция.

Проверить текущий статус:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:get
files_sharing outgoing_server2server_group_share_enabled
```

Выключить:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing outgoing_server2server_group_share_enabled --value="no"
```

Включить:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing outgoing_server2server_group_share_enabled --value="yes"
```

- **4. Включение опции «Разрешить пользователям этого сервера принимать общие ресурсы с других серверов, опубликованные для групп пользователей»**

Экспериментальная функция.

Проверить текущий статус:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:get
files_sharing incoming_server2server_group_share_enabled
```

Выключить:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing incoming_server2server_group_share_enabled --value="no"
```

Включить:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing incoming_server2server_group_share_enabled --value="yes"
```

- **5. Включение опции «По умолчанию автоматически принимать общие ресурсы от доверенных федеративных учетных записей и групп»**

Проверить текущий статус:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:get
files_sharing federatedTrustedShareAutoAccept
```

Выключить:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing federatedTrustedShareAutoAccept --value="no"
```

Включить:

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
files_sharing federatedTrustedShareAutoAccept --value="yes"
```

На целевом сервере

Аналогично выполните настройки на целевом сервере.

> СМ. ТАКЖЕ
>
> - Настройка межсерверного обмена с использованием веб-интерфейса описано в документе «**Руководство администратора по настройке и администрированию АльтерОфис Веб**».

5.2.9. Настройка доверенных серверов

Настройка доверенных серверов выполняется с использованием веб-интерфейса, см. описание в документе «**Руководство администратора по настройке и администрированию АльтерОфис Веб**».

Для ускорения процесса добавления сервера в доверенные должны отработать фоновые задания:

- `OCA\Federation\BackgroundJob\GetSharedSecret`
- `OCA\Federation\BackgroundJob\RequestSharedSecret`

Фоновое задание **GetSharedSecret** используется для получения и верификации общего секретного ключа от удалённого сервера при установлении доверенного соединения.

Фоновое задание **RequestSharedSecret** используется для инициирования запроса на создание общего секрета к удалённому серверу. Запускает процесс рукопожатия (handshake) перед началом федеративного обмена, отправляя запрос на установление защищённого канала связи.

Для проверки статуса воспользуйтесь командой:

```
docker exec -it <CONTAINER_NAME> php /var/www/html/occ background-job:list |
grep Secret
```

Пример для исходного сервера:

```
docker exec -it demo03-app-1 php /var/www/html/occ background-job:list | grep
Secret
```

В результате будет выведен список заданий с идентификаторами.

Для принудительного запуска фонового задания используйте полученный идентификатор в команде выполнения:

```
docker exec -it <CONTAINER_NAME> php /var/www/html/occ background-job:execute
<JOB_ID>
```

Пример для исходного сервера:

```
docker exec -it demo03-app-1 php /var/www/html/occ background-job:execute 312
```

После выполнения фоновых заданий `GetSharedSecret` и `RequestSharedSecret`, должно быть выполнено задание `SyncJob`.

Фоновое задание **SyncJob** используется для периодической синхронизации списка пользователей и групп с доверенными федеративными серверами.

Запуск задания **SyncJob** является обязательным для завершения процесса добавления сервера в доверенные. Только после этого статус индикатор изменится на зеленый и будет осуществлена синхронизация адресных книг, которая позволяет искать пользователей с другого сервера.

Определите идентификатор задания выполнив команду:

```
docker exec -it <CONTAINER_NAME> php /var/www/html/occ background-job:list |
grep SyncJob
```

Пример для исходного сервера:

```
docker exec -it demo03-app-1 php /var/www/html/occ background-job:list | grep
SyncJob
```

Результат выполнения:

```
| 12 | OCA\Federation\SyncJob       | 2025-12-05T07:42:21+00:00 | null      |
```

Для принудительного запуска фонового задания выполните команду:

```
# Запускаем принудительно задание
docker exec -it <CONTAINER_NAME> php /var/www/html/occ background-job:execute
<JOB_ID> --force-execute
```

Пример для исходного сервера:

```
# Запускаем принудительно задание
docker exec -it demo03-app-1 php /var/www/html/occ background-job:execute 12 -
-force-execute
```

После выполнения команды индикатор доверенного сервера в веб-интерфейсе должен стать зелёным.

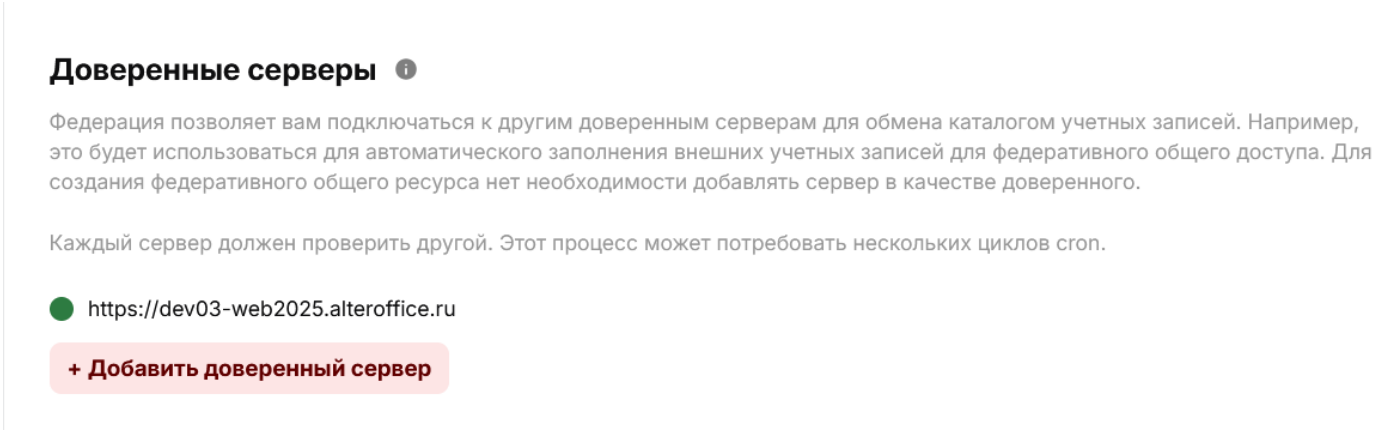


Рисунок 40. Настройка

Повторите аналогичные действия для целевого сервера.

5.2.10. Синхронизация адресных книг для федеративного доступа

Синхронизация адресных книг обеспечивает автоматический обмен контактами между доверенными серверами, позволяя пользователям быстро находить коллег на партнёрских экземплярах АльтерОфис Веб при предоставлении общего доступа. Эта функция работает в фоновом режиме через `cron` задания и активируется при успешном установлении «зелёного» статуса соединения между серверами.

Для ускорения процесса синхронизации выполните команду:

```
docker exec -it <CONTAINER_NAME> php /var/www/html/occ federation:sync-addressbooks
```

Пример для исходного сервера:

```
docker exec -it demo03-app-1 php /var/www/html/occ federation:sync-addressbooks
```

Для целевого сервера выполните аналогичные действия.

5.2.11. Настройка редакторов для совместной работы при федеративном доступе

Для открытия офисных документов, к которым предоставлен федеративный доступ, необходимо в контейнере редактора **editors** выполнить настройки.

Для исходного сервера

- **Для контейнера `editors`:**

```
docker exec -it demo03-editors-1 coolconfig set net.content_security_policy "frame-ancestors *.alteroffice.ru:*;" --config=/etc/coolwsd/coolwsd.xml
```

Ожидаемый результат выполнения:

```
Previous value found in config file: ""
Changing value to: "frame-ancestors *.alteroffice.ru:*;"
```

```
Saving configuration to : /etc/coolwsd/coolwsd.xml ...
Saved
```

- **Для контейнера app:**

```
docker exec -it demo03-app-1 php /var/www/html/occ config:app:set
richdocuments federation_use_trusted_domains --value="yes"
```

Ожидаемый результат выполнения:

```
Config value 'federation_use_trusted_domains' for app 'richdocuments' is now
set to 'yes', stored as mixed in fast cache
```

Для целевого сервера выполните аналогичные действия.

5.3. Подключение S3-совместимого хранилища как основного хранилища пользовательских данных

Данный раздел описывает, как подключить существующее S3-совместимое хранилище (например, MinIO или другое) в качестве основного хранилища данных системы.

В качестве примера приведена настройка для **MinIO**, который по умолчанию уже развернут.

5.3.1. Настройка основного хранилища MinIO на уже развернутой системе

> ВНИМАНИЕ
>
> - Настройка основного S3-хранилища объектов сделает все существующие локальные файлы недоступными.

Шаг 1. Подготовка конфигурационного файла

Создайте файл конфигурации для системы, например, **storage.config.php**, который будет содержать параметры подключения к вашему S3-совместимому хранилищу.

Ниже приведен пример для **MinIO**:

```
<?php
    $CONFIG = array ('objectstore' => array(
      'class' => '\\OC\\Files\\ObjectStore\\S3',
      'arguments' => array(
        'bucket' => 'имя_вашего_бакета',
        'key' => 'ваш_access_key',
        'secret' => 'ваш_secret_key',
        'hostname' => 'адрес_хранилища',
        'port' => порт_подключения,
        'use_ssl' => false, // или true, если используется SSL
        'use_path_style' => true, // рекомендуется для MinIO и аналогичных
решений
    ),
  ),
);
?>
```

ВАЖНО

- Параметры **hostname, port, key, secret, bucket** должны соответствовать настройкам вашего хранилища.

Для других решений могут потребоваться дополнительные параметры или отличаться названия.

Шаг 2. Размещение файла конфигурации

Скопируйте подготовленный файл в директорию конфигурации вашей системы:

```
sudo cp storage.config.php /var/www/alteroffice/html/config/
sudo chown www-data:www-data
/var/www/alteroffice/html/config/storage.config.php
```

Шаг 3. Проверка конфигурации:

Чтобы убедиться в правильности настройки, выполните команду внутри контейнера:

```
docker exec -it <имя_контейнера> /bin/bash
./occ config:system:get objectstore
```

Вы должны увидеть параметры, соответствующие вашей конфигурации.

ВНИМАНИЕ

- После добавления файла конфигурации изменения вступают в силу автоматически.
- Существующие файлы останутся в локальной папке.
- Новые файлы, загружаемые после подключения, будут сохраняться в указанное S3-хранилище.

5.3.2. Настройка MinIO в качестве основного хранилища при развертывании с нуля

ВАЖНО

- Настройка Minio и его развертывание осуществляется вне рамок этой инструкции.

Предполагается, что выбранное хранилище уже развернуто и доступно.

Шаг 1. Настройка окружения контейнера приложения.

В файле конфигурации контейнера (**compose.yaml** или аналогичном) для сервиса **app** необходимо указать переменные окружения, содержащие параметры подключения к хранилищу:

```
- OBJECTSTORE_S3_HOST=$MINIO_HOST
- OBJECTSTORE_S3_PORT=$MINIO_PORT
- OBJECTSTORE_S3_BUCKET=$MINIO_BUCKET
- OBJECTSTORE_S3_SSL=$MINIO_SSL
- OBJECTSTORE_S3_KEY=$MINIO_KEY
- OBJECTSTORE_S3_SECRET=$MINIO_SECRET
- OBJECTSTORE_S3_AUTOCREATE=$MINIO_AUTOCREATE
- OBJECTSTORE_S3_USEPATH_STYLE=$MINIO_USEPATH_STYLE
```

Пример:

```
- OBJECTSTORE_S3_HOST=minio
- OBJECTSTORE_S3_PORT=9005
```

```
- OBJECTSTORE_S3_BUCKET=aow-external
- OBJECTSTORE_S3_SSL=false
- OBJECTSTORE_S3_KEY=alterofficeweb
- OBJECTSTORE_S3_SECRET=alterofficeweb
- OBJECTSTORE_S3_AUTOCREATE=true
- OBJECTSTORE_S3_USEPATH_STYLE=true
```

> ПРИМЕЧАНИЕ
>
> Стандартные порты 9000 и 9001 необходимо поменять и пробросить их в контейнер minio.

```
services:
  minio:
    image: minio/minio
    environment:
      MINIO_ROOT_USER: alterofficeweb
      MINIO_ROOT_PASSWORD: alterofficeweb
    ports:
::::{custom-style="Document Bullet tip"}
"9005:9000" # for S3 API access
::::
::::{custom-style="Document Bullet tip"}
"9006:9001" # for the MinIO Console
::::
```

Параметры, которые необходимы для настройки:

Параметр	Описание	Пример
OBJECTSTORE_S3_HOST	Хост (URL-эндпоинт) Minio	`minio`
OBJECTSTORE_S3_PORT	Порт для подключения к хранилищу	`9005`
OBJECTSTORE_S3_BUCKET	Имя бакета, который будет использован для хранения данных	`aow-external`
OBJECTSTORE_S3_SSL	Использовать ли SSL/TLS для соединения. Указывайте `false` только для тестирования с self-signed сертификатами. Значение по умолчанию: `true`	`false`
OBJECTSTORE_S3_KEY	Ключ доступа (Access Key) для вашего объектного хранилища. Совпадает с указанным MINIO_ROOT_USER	`alterofficeweb`
OBJECTSTORE_S3_SECRET	Секретный ключ (Secret Key) для вашего объектного хранилища. Совпадает с	`alterofficeweb`

Параметр	Описание	Пример
	указанным MINIO_ROOT_PASSWORD	
OBJECTSTORE_S3_AUTOCREATE	Создать бакет, если он не существует. Значение по умолчанию: `true`	`true`
OBJECTSTORE_S3_USEPATH_STYLE	Использовать path-style URL для доступа к бакетам. Установите в `true` для совместимости с некоторыми провайдерами, такими как MinIO или старые версии Ceph. Значение по умолчанию: `false`	`true`

Эти переменные позволяют системе динамически определить параметры подключения к вашему хранилищу.

Затем эти переменные можно использовать в конфигурационных файлах системы или передавать как переменные окружения при запуске контейнеров.

Шаг 2. Запуск скрипта установки.

Продолжить с **Шага №2** инструкции «**Руководство администратора по развертыванию системы АльтерОфис Веб**»

В такой конфигурации все пользовательские данные будут созданы в указанном S3 хранилище.

ПРИМЕЧАНИЕ

- В случае потери соединения с хранилищем или неправильной настройки, система может отображать ошибку, например: «Внутренняя ошибка сервера. Запрос не может быть обработан сервером».

5.4. Настройка обратного прокси на Nginx

Настройка обратного прокси (Nginx) для **АльтерОфис Веб Онлайн** требуется не всегда, а лишь при необходимости опубликовать сервис на стандартных портах или централизовать управление SSL-сертификатами.

По умолчанию серверная часть **АльтерОфис Веб Онлайн** принимает соединения на порту 9980. Если в сети предприятия разрешены только стандартные порты 80 (HTTP) и 443 (HTTPS), доступ к офису организуется через обратный прокси Nginx. В этой схеме Nginx принимает внешние HTTPS-запросы на порту 443, централизованно обрабатывает SSL-сертификат и перенаправляет защищенный трафик на внутренний порт **АльтерОфис Веб Онлайн** (9980), обеспечивая редактирование документов прямо в браузере.

Параметры, которые потребуются для настройки:

Параметр	Описание	Пример
SERVER_NAME	Внешнее имя сервера.	`demo03-web2025.alteroffice.ru`
SERVER_URL	URL доступа к системе АльтерОфис Веб	`https://demo03-web2025.alteroffice.ru`
PATH_SSL	Путь к директории с SSL сертификатами	`/etc/ssl/alteroffice`

5.4.1. Подготовка к настройке обратного прокси

- Установлен и запущен **АльтерОфис Веб Онлайн**.
- Установлен веб-сервер Nginx.
- Имеется доменное имя, например `demo03-web2025.alteroffice.ru`, и SSL-сертификат для него (например, `server.key` и `server.pem`).
- Сервер может обращаться к самому себе по адресу 127.0.0.1:9980.

5.4.2. Настройка АльтерОфис Веб Онлайн

> ПРЕДУПРЕЖДЕНИЕ
>
> **Важно!** Перед внесением изменений в файл `/etc/coolwsd/coolwsd.xml` рекомендуется создать его резервную копию. Неверные параметры могут привести к неработоспособности сервиса.

Для корректной работы **АльтерОфис Веб Онлайн** при размещении за обратным прокси-сервером Nginx необходимо в конфигурации **АльтерОфис Веб Онлайн** указать внешнее имя сервера. Это обеспечит правильное формирование ответов и исключит ошибки при проксировании запросов.

Шаг 1. Откройте файл конфигурации `/etc/coolwsd/coolwsd.xml` на редактирование:

```
sudo nano /etc/coolwsd/coolwsd.xml
```

Или используйте любой другой текстовый редактор.

Шаг 2. Найдите параметр `<server_name>` (по умолчанию «пустой») и укажите полное доменное имя вашего сервера, например:

```
<server_name>demo03-web2025.alteroffice.ru</server_name>
```

Убедитесь, что SSL в **АльтерОфис Веб Онлайн** включён. Параметр `ssl enable` должен иметь значение `true` (обычно установлено по умолчанию). Найдите секцию **ssl** и проверьте:

```
<ssl enable="true" />
```

Если параметр отсутствует или выставлен в `false`, измените его.

Шаг 3. Сохраните изменения и перезапустите службу:

```
sudo systemctl restart coolwsd
```

Имя сервера в параметре `<server_name>` должно точно совпадать с доменом, указанным в конфигурации Nginx и в SSL-сертификате.

5.4.3. Настройка обратного прокси в Nginx

Шаг 1. Создайте новый файл конфигурации для сайта (например, `/etc/nginx/sites-available/alteroffice`) и отредактируйте его:

```
sudo nano /etc/nginx/sites-available/alteroffice
```

Шаг 2. Вставьте следующий блок конфигурации, укажите ваше доменное имя и правильные пути к SSL-сертификатам:

```
nginx
server {
    listen 443 ssl;
    server_name <SERVER_NAME>;                    # замените на ваше доменное
имя

    ssl_certificate <PATH_SSL>/server.pem;        # замените на путь к вашему
сертификату
    ssl_certificate_key <PATH_SSL>/server.key;    # замените на путь к закрытому
ключу

    # Статические файлы (браузерный клиент)
    location ^~ /browser {
        proxy_pass https://127.0.0.1:9980;
        proxy_set_header Host $host;
    }

    # WOPI discovery URL (используется для обнаружения возможностей)
    location ^~ /hosting/discovery {
        proxy_pass https://127.0.0.1:9980;
        proxy_set_header Host $host;
    }

    # Capabilities (возможности сервера)
    location ^~ /hosting/capabilities {
        proxy_pass https://127.0.0.1:9980;
        proxy_set_header Host $host;
    }

    # Основной веб-сокет для редактирования документов
    location ~ ^/cool/(.*)/ws$ {
        proxy_pass https://127.0.0.1:9980;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Host $host;
        proxy_read_timeout 36000s;   # долгий таймаут для веб-сокетов
    }

    # Операции с документами (загрузка, сохранение, скачивание и т.п.)
```

```
    location ~ ^/cool {
        proxy_pass https://127.0.0.1:9980;
        proxy_set_header Host $host;
    }

    # Веб-сокет консоли администратора
    location ^~ /cool/adminws {
        proxy_pass https://127.0.0.1:9980;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Host $host;
        proxy_read_timeout 36000s;
    }
}
```

ПОЯСНЕНИЯ К ДИРЕКТИВАМ

- `listen 443 ssl;` — сервер будет принимать HTTPS-соединения на порту 443.
- `server_name` — доменное имя, по которому доступен сервис.
- `ssl_certificate` и `ssl_certificate_key` — пути к файлам сертификата и ключа.
- `location` — правила перенаправления запросов. Для каждого пути запросы передаются на https://127.0.0.1:9980.
- `proxy_set_header Host $host;` — передаёт оригинальный заголовок Host внутреннему серверу.
- Для веб-сокетов добавлены заголовки `Upgrade` и `Connection`, а также увеличен таймаут.

Шаг 3. Сохраните файл и активируйте его, создав символическую ссылку в каталоге `sites-enabled`:

```
sudo ln -s /etc/nginx/sites-available/alteroffice /etc/nginx/sites-enabled/
```

ПРЕДУПРЕЖДЕНИЕ

Важно! Для корректной работы веб-сокетов обязательны директивы `proxy_set_header Upgrade` и `proxy_set_header Connection "Upgrade"`, а также достаточно большой `proxy_read_timeout` (по умолчанию 60 секунд может быть недостаточно).

Если вы используете брандмауэр, убедитесь, что порт 443 открыт для входящих соединений.

5.4.4. Проверка конфигурации и перезапуск Nginx

Шаг 1. Проверьте корректность синтаксиса конфигурационных файлов Nginx:

```
sudo nginx -t
```

Если вывод содержит `syntax is ok` и `test is successful`, можно продолжать.

Шаг 2. Перезагрузите Nginx, чтобы применить изменения:

```
sudo systemctl reload nginx
```

Шаг 3. Проверка работоспособности

Зайдите в систему по ссылке <SERVER_URL>, авторизуйтесь и откройте любой документ в **АльтерОфис Веб Онлайн**.

Открытие должно произойти корректно, без появления ошибки.

Пример ссылки:

`https://demo03-web2025.alteroffice.ru`

6.Резервное копирование и восстановление

Резервное копирование и восстановление системы «АльтерОфис Веб» будет рассмотрено на примере:

- **Экземпляр 1** имеет адрес `https://demo03-web2025.alteroffice.ru`

Параметры, которые потребуются для копирования и восстановления:

Параметр	Описание	Пример
PATH_SERVER	Путь к директории на хосте, где установлена система АльтерОфис Веб	`/opt/alterofficeweb/demo03`
PATH_BACKUP	Путь к директории для хранения (временного) резервных копий.	`/home/alterofficeweb/backup`
APP_CONTAINER_NAME	Имя контейнера основного веб-приложения.	`prod-app-1`
FRONT_CONTAINER_NAME	Имя контейнера, обеспечивающего балансировку нагрузки, кэширование статического содержимого, перенаправления запросов в другие контейнеры.	`prod-front-1`
CRON_CONTAINER_NAME	Имя контейнера, обеспечивающего выполнение фоновых задач.	`prod-cron-1`
EDITORS_CONTAINER_NAME	Имя контейнера, обеспечивающего совместное редактирование документов.	`prod-editors-1`
DB_CONTAINER_NAME	Имя контейнера, обеспечивающего хранение служебной информации, пользователей, метаданных, настроек.	prod-db-1

6.1. Резервное копирование АльтерОфис Веб

Рекомендуется регулярно создавать резервные копии системы: данных, конфигураций, настроек.

Ниже будет рассмотрено несколько вариантов резервного копирования.

ПРЕДВАРИТЕЛЬНЫЕ ТРЕБОВАНИЯ

Перед выполнением команд Docker необходимо:

- добавить текущего пользователя в группу docker или
- использовать sudo для запуска команд с правами суперпользователя.

6.1.1. Полное копирование системы

Создаётся полная копия всей инфраструктуры, включая:

- ядро системы (файлы приложения);
- база данных (PostgreSQL);
- конфигурационные файлы;
- пользовательские данные;
- хранилище MinIO (если используется).

ВАЖНО

- Восстановление возможно только в рамках одной версии продукта (например, резервная копия версии 2025.0.0.8 несовместима с версией 2025.0.0.9).

Шаг 1. Перейдите в каталог, где установлена система.

```
cd <PATH_SERVER>
```

Пример:

```
cd /opt/alterofficeweb/demo03
```

Шаг 2. Остановите контейнер с онлайн-редакторами.

Требуется остановить контейнер **editors**.

```
sudo docker stop <EDITORS_CONTAINER_NAME>
```

Пример:

```
sudo docker stop prod-editors-1
```

Шаг 3. Остановите контейнеры АльтерОфис Веб (контейнер с базой данных продолжает работать).

Требуется остановить контейнеры **app**, **cron**, **front**.

```
sudo docker stop <APP_CONTAINER_NAME> <CRON_CONTAINER_NAME>
<FRONT_CONTAINER_NAME>
```

Пример:

```
sudo docker stop prod-app-1 prod-cron-1 prod-front-1
```

Шаг 4. Создайте директорию для резервного копирования

```
mkdir -p <PATH_BACKUP>
```

Пример:

```
mkdir -p /home/alterofficeweb/backup
```

Назначьте необходимые права доступа к директории.

Шаг 5. Создайте архив с данными резервного копирования:

```
sudo tar -czf <PATH_BACKUP>/app_$(date +%Y_%m_%d_%H_%M).tar.gz <PATH_SERVER>
```

Замените:

- <PATH_BACKUP> на путь к директории резервного копирования;
- <PATH_SERVER> на путь, где установлена система.

Пример:

```
sudo tar -czf /home/alterofficeweb/backup/app_$(date +%Y_%m_%d_%H_%M).tar.gz
/opt/alterofficeweb/demo03
```

Шаг 6. Создайте резервный дамп ролей баз данных:

```
source .env && sudo docker exec <DB_CONTAINER_NAME> pg_dumpall -U
"$POSTGRES_USER" --roles-only | gzip > <PATH_BACKUP>/roles_$(date
+%Y_%m_%d_%H_%M).sql.gz
```

Пример:

```
source .env && sudo docker exec prod-db-1 pg_dumpall -U "$POSTGRES_USER" --
roles-only | gzip > /home/alterofficeweb/backup/roles_$(date
+%Y_%m_%d_%H_%M).sql.gz
```

Шаг 7. Создайте резервный дамп базы данных:

```
source .env && sudo docker exec <DB_CONTAINER_NAME> pg_dump -U "$POSTGRES_USER"
"$POSTGRES_DB" | gzip > <PATH_BACKUP>/db_$(date +%Y_%m_%d_%H_%M).sql.gz
```

```
source .env && sudo docker exec <DB_CONTAINER_NAME> pg_dump -U "$POSTGRES_USER"
"$POSTGRES_DB" | gzip > /home/alterofficeweb/backup/db_$(date
+%Y_%m_%d_%H_%M).sql.gz
```

Резервные копии созданы и находятся в директории /home/alterofficeweb/backup.

Рекомендуется хранить резервные копии отдельно от данных системы (на отдельном сервере или использовать специализированные системы хранения данных).

Шаг 8. Запустите ранее остановленные контейнеры АльтерОфис Веб

Требуется запустить контейнеры **app**, **cron**, **front**.

```
sudo docker start <APP_CONTAINER_NAME> <CRON_CONTAINER_NAME>
<FRONT_CONTAINER_NAME>
```

Пример:

```
sudo docker start prod-app-1 prod-cron-1 prod-front-1
```

Шаг 9. Запустите ранее остановленный контейнер с онлайн-редакторами.

Требуется запустить контейнер **editors**.

```
sudo docker start <EDITORS_CONTAINER_NAME>
```

Пример:

```
sudo docker start prod-editors-1
```

На данном этапе создание полной резервной копии системы завершено. Храните резервные копии отдельно от серверов, на которых установлена система.

Периодически проверяйте корректность создания резервных копий, путем восстановления системы из резервных копий на тестовом стенде.

6.1.2. Резервное копирование пользовательских данных

Сохраняются только пользовательские данные, конфигурационные файлы и база данных.

> ПРЕИМУЩЕСТВО
>
> - Позволяет выполнять восстановление на разных версиях продукта.

Шаг 1. Перейдите в каталог, где установлена система.

```
cd <PATH_SERVER>
```

Пример:

```
cd /opt/alterofficeweb/demo03
```

Шаг 2. Остановите контейнер с онлайн-редакторами.

Требуется остановить контейнер **editors**.

```
sudo docker stop <EDITORS_CONTAINER_NAME>
```

Пример:

```
sudo docker stop prod-editors-1
```

Шаг 3. Остановите контейнеры АльтерОфис Веб (контейнер с базой данных продолжает работать).

Требуется остановить контейнеры **app**, **cron**, **front**.

```
sudo docker stop <APP_CONTAINER_NAME> <CRON_CONTAINER_NAME>
<FRONT_CONTAINER_NAME>
```

Пример:

```
sudo docker stop prod-app-1 prod-cron-1 prod-front-1
```

Шаг 4. Создайте директорию для резервного копирования

```
mkdir -p <PATH_BACKUP>
```

Пример:

```
mkdir -p /home/alterofficeweb/backup
```

Назначьте необходимые права доступа к директории.

Шаг 5. Создайте архив с данными резервного копирования:

```
sudo tar -czf <PATH_BACKUP>/user_data_$(date +%Y_%m_%d_%H_%M).tar.gz
<PATH_SERVER>/html/data <PATH_SERVER>/html/config
```

Замените:

- <PATH_BACKUP> на путь к директории резервного копирования;
- <PATH_SERVER> на путь, где установлена система.

Пример:

```
sudo tar -czf /home/alterofficeweb/backup/user_data_$(date
+%Y_%m_%d_%H_%M).tar.gz /opt/alterofficeweb/demo03/html/data
/opt/alterofficeweb/demo03/html/config
```

Шаг 6. Создайте резервный дамп ролей баз данных:

```
source .env && sudo docker exec <DB_CONTAINER_NAME> pg_dumpall -U
"$POSTGRES_USER" --roles-only | gzip > <PATH_BACKUP>/roles_$(date
+%Y_%m_%d_%H_%M).sql.gz
```

Пример:

```
source .env && sudo docker exec prod-db-1 pg_dumpall -U "$POSTGRES_USER" --
roles-only | gzip > /home/alterofficeweb/backup/roles_$(date
+%Y_%m_%d_%H_%M).sql.gz
```

Шаг 7. Создайте резервный дамп базы данных:

```
source .env && sudo docker exec <DB_CONTAINER_NAME> pg_dump -U "$POSTGRES_USER"
"$POSTGRES_DB" | gzip > <PATH_BACKUP>/db_$(date +%Y_%m_%d_%H_%M).sql.gz
```

Пример:

```
source .env && sudo docker exec <DB_CONTAINER_NAME> pg_dump -U "$POSTGRES_USER"
"$POSTGRES_DB" | gzip > /home/alterofficeweb/backup/db_$(date
+%Y_%m_%d_%H_%M).sql.gz
```

Резервные копии созданы и находятся в директории /home/alterofficeweb/backup.

Рекомендуется хранить резервные копии отдельно от данных системы (на отдельном сервере или использовать специализированные системы хранения данных).

Шаг 8. Запустите ранее остановленные контейнеры АльтерОфис Веб

Требуется запустить контейнеры **app**, **cron**, **front**.

```
sudo docker start <APP_CONTAINER_NAME> <CRON_CONTAINER_NAME>
<FRONT_CONTAINER_NAME>
```

Пример:

```
sudo docker start prod-app-1 prod-cron-1 prod-front-1
```

Шаг 9. Запустите контейнер с онлайн-редакторами

```
sudo docker start <EDITORS_CONTAINER_NAME>
```

Пример:

```
sudo docker start prod-editors-1
```

На этом резервное копирование пользовательских данных завершено.

6.2. Восстановление АльтерОфис Веб из резервных копий

Шаг 1. Перейдите в каталог, где установлена система.

```
cd <PATH_SERVER>
```

Замените `<PATH_SERVER>` на путь, где установлена система.

Пример:

```
cd /opt/alterofficeweb/demo03
```

Шаг 2. Остановите контейнер с онлайн-редакторами.

```
sudo docker stop <EDITORS_CONTAINER_NAME>
```

Пример:

```
sudo docker stop prod-editors-1
```

Шаг 3. Остановите контейнеры с АльтерОфис Веб.

```
sudo docker stop <APP_CONTAINER_NAME> <CRON_CONTAINER_NAME>
<FRONT_CONTAINER_NAME>
```

Пример:

```
sudo docker stop prod-app-1 prod-cron-1 prod-front-1
```

Шаг 4. Распакуйте архив с резервной копией

- Для полного восстановления системы:

```
sudo tar -xzf <PATH_BACKUP>/app_<*****>.tar.gz -C /
```

- Для восстановления пользовательских данных:

```
sudo tar -xzf <PATH_BACKUP>/user_data_<*****>.tar.gz -C /
```

Замените:

- <PATH_BACKUP> на путь к резервной копии.
- <*****> на дату и время создания резервной копии.

Пример:

- Полное восстановление системы:

```
sudo tar -xzf /home/alterofficeweb/backup/app_2025_11_25_20_44.tar.gz -C /
```

- Восстановление пользовательских данных:

```
sudo tar -xzf /home/alterofficeweb/backup/user_data_2025_11_25_20_44.tar.gz -C
/
```

Шаг 5. Удалите текущую базу данных

```
source .env && sudo docker exec <DB_CONTAINER_NAME> psql -U "$POSTGRES_USER" -d
template1 -c "DROP DATABASE \"$POSTGRES_DB\";"
```

Пример:

```
source .env && sudo docker exec prod-db-1 psql -U "$POSTGRES_USER" -d template1
-c "DROP DATABASE \"$POSTGRES_DB\";"
```

Шаг 6. Создайте новую базу данных

```
source .env && sudo docker exec <DB_CONTAINER_NAME> psql -U "$POSTGRES_USER" -d
template1 -c "CREATE DATABASE \"$POSTGRES_DB\";"
```

Пример:

```
source .env && sudo docker exec prod-db-1 psql -U "$POSTGRES_USER" -d template1
-c "CREATE DATABASE \"$POSTGRES_DB\";"
```

Шаг 7. Восстановите роли базы данных из резервной копии

```
source .env && gunzip -c <PATH_BACKUP>/roles_<****>.sql.gz | sudo docker exec -
i <DB_CONTAINER_NAME> psql -U "$POSTGRES_USER" -d postgres
```

Замените:

- <PATH_BACKUP> на путь к резервной копии.
- <DB_CONTAINER_NAME> на имя контейнера, содержащего базу данных системы.
- <*****> на дату и время создания резервной копии.

Пример:

```
source .env && gunzip -c
/home/alterofficeweb/backup/roles_2025_11_25_20_44.sql.gz | sudo docker exec
-i prod-db-1 psql -U "$POSTGRES_USER" -d postgres
```

Шаг 8. Восстановите данные из резервной копии

```
source .env && gunzip -c <PATH_BACKUP>/db_<*****>.sql.gz | sudo docker exec -i
<DB_CONTAINER_NAME> psql -U "$POSTGRES_USER" -d "$POSTGRES_DB"
```

Пример:

```
source .env && gunzip -c
/home/alterofficeweb/backup/db_2025_11_25_20_44.sql.gz | sudo docker exec -i
prod-db-1 psql -U "$POSTGRES_USER" -d "$POSTGRES_DB"
```

Шаг 9. Запустите контейнеры с АльтерОфис Веб.

```
sudo docker start <APP_CONTAINER_NAME> <CRON_CONTAINER_NAME>
<FRONT_CONTAINER_NAME>
```

Пример:

```
sudo docker start prod-app-1 prod-cron-1 prod-front-1
```

Шаг 10. Запустите контейнеры с онлайн-редакторами АльтерОфис Веб

```
sudo docker start <EDITORS_CONTAINER_NAME>
```

Пример:

```
sudo docker start prod-editors-1
```

На данном этапе восстановление системы завершено.

7.Процедура обновления сертификата и резервное копирование сертификатов

7.1. Создание резервной копии текущих сертификатов

Шаг 1. На хостовой машине перейдите в директорию, где лежат сертификаты:

```
cd <PATH_SSL>
```

Замените:

- <PATH_SSL> на путь к директории с SSL сертификатами.

Пример:

```
cd /etc/ssl/alteroffice
```

Шаг 2. Создайте архив текущих сертификатов:

```
sudo tar -czf <PATH_BACKUP>/alteroffice_ssl_backup_$(date
+%Y_%m_%d_%H_%M).tar.gz -C <PATH_SSL> .
```

Замените:

- <PATH_BACKUP> на путь к директории резервного копирования;
- <PATH_SSL> на путь к директории с SSL сертификатами.

Пример:

```
sudo tar -czf /home/alterofficeweb/backup/alteroffice_ssl_backup_$(date
+%Y_%m_%d_%H_%M).tar.gz -C /etc/ssl/alteroffice .
```

7.2. Установка новых сертификатов

Шаг 1. Скопируйте новые сертификаты

```
sudo cp server.key <PATH_BACKUP>/
sudo cp server.crt <PATH_BACKUP>/
```

Пример:

```
sudo cp server.key /etc/ssl/alteroffice/
sudo cp server.crt /etc/ssl/alteroffice/
```

Шаг 2. Установите правильные права доступа

```
sudo chown root:root <PATH_BACKUP>/*
sudo chmod 644 <PATH_BACKUP>/server.key
sudo chmod 600 <PATH_BACKUP>/server.crt
```

Пример:

```
sudo chown root:root /etc/ssl/alteroffice/*
sudo chmod 644 /etc/ssl/alteroffice/server.key
sudo chmod 600 /etc/ssl/alteroffice/server.crt
```

Шаг 3. Перезапустите сервис **front** для применения сертификата

```
cd <PATH_SERVER>
sudo docker restart <FRONT_CONTAINER_NAME>
```

Пример:

```
cd /opt/alterofficeweb/demo03
sudo docker restart prod-front-1
```

Шаг 4. Проверьте валидность сертификата (вместо *stage.my.local* укажите адрес своего сервера)

```
echo | openssl s_client -connect stage.my.local:443 2>/dev/null | openssl x509 -noout -dates -subject
```

7.3. Восстановление старых сертификатов (в случае ошибки)

Шаг 1. Перейдите в директорию:

```
cd <PATH_SSL>
```

Замените:

- `<PATH_SSL>` на путь к директории с SSL сертификатами.

Пример:

```
cd /etc/ssl/alteroffice
```

Шаг 2. Удалите новые файлы:

```
sudo rm server.key server.crt
```

Шаг 3. Распакуйте резервную копию (укажите имя вашего архива):

```
sudo tar -xzf <PATH_BACKUP>/alteroffice_ssl_backup_<*****>.tar.gz -C <PATH_SSL>
```

Замените:

- `<PATH_BACKUP>` на путь к директории резервного копирования;
- `<PATH_SSL>` на путь к директории с SSL сертификатами.
- `<*****>` на дату и время создания резервной копии.

Пример:

```
sudo tar -xzf /home/alterofficeweb/backup/alteroffice_ssl_backup_2025_11_25_20_44.tar.gz -C /etc/ssl/alteroffice
```

Шаг 4. Убедитесь, что права восстановлены:

```
sudo chmod 644 server.key
sudo chmod 600 server.crt
```

Шаг 5. Перезапустите **front**

```
cd <PATH_SERVER>
sudo docker restart <FRONT_CONTAINER_NAME>
```

Пример:

```
cd /opt/alterofficeweb/demo03
sudo docker restart prod-front-1
```

8.Удаление системы

8.1. Полная деинсталляция

В зависимости от того, как был установлен Docker Compose, запуск будет либо `docker-compose`, либо `docker compose`.

Шаг 1. Перейти в директорию откуда запускалась инсталляция

Шаг 2. Выполнить поочередно:

```
docker-compose -f compose.yaml --project-name prod down -v
docker-compose -f compose-opensearch.yaml --project-name prod-elk down -v
docker network rm alterofficeweb
```

Шаг 3. Удалить директории и файлы конфигурации:

- «Директория контента» - посмотреть в `installation_backup_***.txt`
- Удалить, если есть, директорию **db**.
- Удалить файлы конфигурации `.env`, все `compose` файлы.

9.Обновление системы

В данном разделе описано обновление системы АльтерОфис Веб на примере однонодового решения.

> ВНИМАНИЕ
>
> - Данная инструкция подходит для версии инсталлятора 0.0.9.

Шаг 1. Останавливаем приложение.

```
docker-compose -f compose.yaml --project-name prod stop app cron front
```

Шаг 2. Делаем полную резервную копию данных, конфигурации, базы данных.

Шаг 3. Скачиваем новую версию из репозитория `https://repo.alteroffice.ru/web/` любым удобным способом:

- Архив с установщиком `alterofficeweb-installer_*.zip`.
- Набор Docker-образов `alterofficeweb-bundle_*.tar.gz`.

Шаг 4. Распакуйте архив с установщиком в текущую директорию с заменой существующих файлов:

```
sudo unzip -o alterofficeweb-installer_*.zip
```

> ВНИМАНИЕ
>
> При обновлении с версии 2026.0.0.1 и ниже на версии 2026.1.0.0 и выше требует замены Elasticsearch на OpenSearch

Начиная с версии 2026.1.0.0, в сервисе произошла замена системы полнотекстового поиска: вместо Elasticsearch теперь используется OpenSearch. В связи с этим при обновлении с версий (2026.0.0.1 и ниже) на версии 2026.1.0.0 и выше требуется выполнить удаление контейнеров предыдущей поисковой платформы и внести изменения в файл .env.

Сначала Остановите и удалите контейнеры Elasticsearch, выполнив команду:

```
docker-compose -f compose-elasticsearch.yaml --project-name prod-elk down -v
```

Затем в файле .env замените значение переменной окружения:

Было: **ELASTIC_HOST=http://elasticsearch:9200**

Стало: **OPENSEARCH_HOST=http://opensearch:9200**

Шаг 5. Загрузите образы из bundle:

```
sudo docker load -i alterofficeweb-bundle_*.tar.gz
```

Шаг 6. Перезапускаем приложение

```
docker-compose -f compose-elasticsearch.yaml --project-name prod-elk up -d
docker-compose -f compose.yaml --project-name prod up -d
```

Если вы обновлялись с версий (2026.0.0.1 и ниже) на версии 2026.1.0.0 и выше, то для перезапуска приложения потребуется следующая команда:

```
docker-compose -f compose-opensearch.yaml --project-name prod-os up -d
docker-compose -f compose.yaml --project-name prod up -d
```

ПРИМЕЧАНИЕ

- Если в процессе обновления возникнут ошибки, АльтерОфис Веб автоматически перейдет в режим обслуживания.

Для его отключения выполните команду:

```
docker-compose -f compose.yaml --project-name prod exec app php
/var/www/html/occ maintenance:mode --off
```

10.Термины, обозначения и сокращения

В текст документа введены специальные сокращения на русском и английском языках. Определение терминов и расшифровка сокращений отражены в таблицах ниже.

10.1. Термины и определения

Термин	Определение
АльтерОфис Веб	Платформа для облачного хранения файлов и совместной работы пользователей (Система).
АТаблица	Приложение для работы с электронными таблицами, входящее в состав офисного пакета АльтерОфис.
Система	Совокупность элементов, объединенная связями между ними и обладающая определенной целостностью. В данном случае АльтерОфис Веб.
Пользователь	Учётная запись с базовым набором прав для работы с личными данными и ресурсами в рамках установленных политик доступа.
Логин	Имя учётной записи, используемое при входе в систему.
Пароль	Секретная комбинация символов для подтверждения личности при входе.
Двухфакторная аутентификация	(2ФА) Дополнительная защита при входе, требующая помимо пароля ввести ещё один код или подтвердить вход другим способом.
Одноразовый пароль	Код, действительный только один раз для подтверждения входа. Может быть получен по разным каналам (email, приложение).
Резервные коды	Набор заранее сгенерированных кодов для входа в систему, когда другие способы двухфакторной аутентификации недоступны.
Разовые коды администратора	Одноразовые коды для входа, выданные администратором пользователю в случае утери или недоступности других способов входа.
Корзина	Временное хранилище удалённых файлов и папок, из которого данные можно восстановить или удалить окончательно.
Папка общего доступа	Папка, доступ к которой предоставлен другим пользователям Системы или по публичной ссылке.
Публичная ссылка	Ссылка, позволяющая предоставить доступ к файлу или папке пользователям вне Системы. Может быть защищена паролем и сроком действия.
Метка (тег)	Ключевое слово, прикрепляемое к файлу или папке для упрощения поиска и организации данных.
Комментарий	Замечание или сообщение, оставляемое к файлу или папке для совместной работы.

Термин	Определение
Аутентификатор	Средство, используемое для подтверждения личности пользователя. Пользователь проходит аутентификацию в компьютерной системе или приложении, демонстрируя, что он владеет аутентификатором и контролирует его.
Роль	Набор прав доступа, назначаемый пользователю или группе пользователей.
Системная роль	Роль, определяющая глобальные права в рамках всей системы.
Администратор системы	Пользователь, обладающий неограниченными правами доступа ко всем функциям и ресурсам экземпляра
Администратор группы	Пользователь с делегированными правами администрирования в пределах назначенной группы.
Общий доступ	Механизм предоставления прав на отдельные ресурсы (файлы, папки).
Гость	Учётная запись с ограниченными правами доступа, предназначенная для временной работы внешних участников.
Политика безопасности паролей	Набор правил и требований, определяющих сложность, срок действия и условия обмена паролями для обеспечения защищенного доступа к системе.

10.2. Обозначения и сокращения

Сокращение	Расшифровка
OTP	One-Time Password — одноразовый пароль, который используется для подтверждения личности пользователя при входе в систему.
TOTP	Time-based One-Time Password — одноразовый пароль, действующий ограниченное время. Вариант OTP, генерируемый внешним приложением (например, «Я Ключ»).
2FA	Two Factor Authentication — двухфакторная аутентификация.
OTP	One-Time Password — одноразовый пароль.
OCM	Open Cloud Mesh – протокол объединения серверов, который используется для уведомления принимающей стороны о том, что ей предоставлен доступ к некоторому ресурсу.
WebDAV	Web Distributed Authoring and Versioning – протокол, который позволяет работать с файлами на сервере как с обычными файлами на локальном диске.
ODF	Open Document Format — открытый стандарт для электронных документов, созданный как универсальное решение для хранения текстовых файлов, таблиц и презентаций. Формат включает текстовые файлы (.odt), электронные таблицы (.ods), презентации

Сокращение	Расшифровка
	(.odp) и другие типы документов, например рисунки (.odg).
OOXML	Office Open XML — серия форматов файлов для хранения электронных документов пакетов офисных приложений (DOCX, XLSX, PPTX).
ОС	Операционная система
ПК	Персональный компьютер
ПО	Программное обеспечение
DNS	Domain Name System. Система доменных имён — механизм, посредством которого символьные доменные имена переводятся в IP-адреса и обратно.
HTTP	HyperText Transfer Protocol. Прикладной сетевой протокол, используемый для передачи гипертекстовых данных (веб-страниц) между клиентом и сервером.
HTTPS	HyperText Transfer Protocol Secure. Расширение HTTP, обеспечивающее шифрование соединения с помощью SSL/TLS, чтобы гарантировать конфиденциальность и целостность передаваемых данных.
SSL/TLS	Secure Sockets Layer / Transport Layer Security. Криптографические протоколы, обеспечивающие защищённую передачу данных в сетях TCP/IP. SSL — ранняя версия, заменена на TLS. Протокол гарантирует шифрование, целостность и аутентичность соединений.
IP	Internet Protocol. Сетевой протокол, определяющий правила адресации и маршрутизации пакетов данных между узлами в сетях TCP/IP.
LDAP	Lightweight Directory Access Protocol. Протокол прикладного уровня, предназначенный для доступа и управления распределёнными каталогами информации. Позволяет централизованно хранить и быстро получать данные о пользователях, группах, устройствах, сервисах и других объектах.
SMTP	Simple Mail Transfer Protocol. Протокол передачи электронной почты между серверами, используется для доставки сообщений и уведомлений.
SSO	Single Sign-On. Механизм обеспечения единого входа: пользователь выполняет авторизацию один раз и получает доступ к множеству связанных приложений/сервисов без необходимости повторного ввода учётных данных.
TCP/IP	Transmission Control Protocol / Internet Protocol

Сокращение	Расшифровка
AD	Active Directory. Служба каталогов и система управления доступом, разработанная Microsoft для централизованного управления ресурсами, такими как пользователи, компьютеры и принтеры, в корпоративной сети.
PHP-LDAP	Расширение для языка PHP, предоставляющее функции для взаимодействия веб-приложений с серверами каталогов, работающими по протоколу LDAP. С его помощью можно выполнять аутентификацию пользователей, осуществлять поиск и получение данных из централизованных каталогов (например, Active Directory), а также управлять записями, хранящимися в иерархической структуре каталога.
SMB/CIFS	SMB (Server Message Block) и его расширенная версия CIFS (Common Internet File System) — это сетевой протокол прикладного уровня, предназначенный для организации совместного доступа к файлам, принтерам и другим сетевым ресурсам в локальной вычислительной сети.
LUKS	Linux Unified Key Setup. Стандарт шифрования дисков в Linux, разработанный для безопасного хранения данных на физических и виртуальных носителях. Основная задача — защитить данные на диске от несанкционированного доступа, даже если диск был извлечён и подключён к другой системе.
CLI	Command Line Interface. Интерфейс командной строки. Это текстовый способ взаимодействия пользователя с компьютерной системой или программным обеспечением через командную строку.
AD FS	Active Directory Federation Services. Компонент операционной системы Windows Server, предоставляющий службы федерации идентификации и обеспечивающий единый вход (Single Sign-On, SSO) для аутентификации пользователей в распределенных средах. С помощью AD FS пользователи могут получать доступ к сторонним приложениям и сервисам (например, АльтерОфис Веб) с использованием своих корпоративных учетных данных домена Active Directory, без необходимости повторного ввода пароля.
SAML	Security Assertion Markup Language. Открытый стандарт на основе XML, предназначенный для обмена данными аутентификации и авторизации между сторонами, в частности между поставщиком удостоверений (Identity Provider) и поставщиком услуг (Service Provider).
SSE	Server-Side Encryption. Метод шифрования данных, при котором

Сокращение	Расшифровка
	процессы шифрования, расшифровывания и управления ключами выполняются исключительно на стороне сервера. Данные автоматически шифруются перед сохранением на диск и расшифровываются при авторизованном доступе, без активного участия клиентских приложений.
S3	Simple Storage Service. Сервис (и одновременно протокол) для хранения данных большого объёма. Для работы использует API поверх HTTP, который позволяет загружать или получать объекты из хранилища.