УТВЕРЖДЕН

АДМГ.20134-01 91 01-ЛУ

ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «АВРОРА ЦЕНТР»

Руководство администратора

АДМГ.20134-01 91 01

Листов 179

2024

АННОТАЦИЯ

Настоящий документ является руководством администратора Прикладного программного обеспечения «Аврора Центр» АДМГ.20134-01 (далее — ППО) релиз 5.0.0.

Настоящий документ содержит общую информацию о ППО, описание установки, обновления, удаления и резервного копирования ППО, описание управления сервисами и их настройками, а также информацию о конфигурационных файлах ППО.

СОДЕРЖАНИЕ

1. ОБЩАЯ ИНФОРМАЦИЯ

1.1. Назначение и состав ППО

ППО является прикладным программным обеспечением со встроенными механизмами защиты информации от несанкционированного доступа, предназначенным для:

– управления устройствами[1], функционирующими под управлением операционной системы (ОС) Аврора, имеющей действительный сертификат соответствия ФСТЭК России;

– управления жизненным циклом приложений[2];

– отправки push-уведомлений на устройства;

– обновления ОС путем получения из доверенного хранилища пакетов с изменениями ОС (образа ОС) и их установки. При этом указанные процессы выполняются штатными средствами самой ОС, а ППО участвует лишь в их инициализации в ОС и не гарантирует их успешного завершения;

– автоматизированной обработки следующих видов информации:

- общедоступной информации;
- информации ограниченного доступа, не содержащей сведений, составляющих государственную тайну, подлежащей защите в соответствии с требованиями действующего законодательства Российской Федерации в области информационной безопасности.

[1] Определение термина «Устройство» приведено в таблице (Таблица 34).

2 Определение термина «Приложение» приведено в таблице (Таблица 34).

ППО может быть использовано, но не ограничиваться, в следующих системах и объектах:

– в государственных информационных системах (ГИС), не содержащих информации, составляющей государственной тайны, до 1 класса защищенности включительно в соответствии с документом «Требования о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах», утвержденным приказом ФСТЭК России от 11 февраля 2013 г. № 17;

– в информационных системах персональных данных (ИСПДн) до 1 уровня защищенности включительно в соответствии с документом «Состав и содержание организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных», утвержденным приказом ФСТЭК России от 18 февраля 2013 г. № 21;

– в автоматизированных системах управления (АСУ) до 1 класса защищенности включительно в соответствии с документом «Требования к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды», утвержденным приказом ФСТЭК России от 14 августа 2014 г. № 31;

– на значимых объектах критической информационной инфраструктуры (КИИ) до 1 категории включительно в соответствии с документом «Требования по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации», утвержденным приказом ФСТЭК России от 25 декабря 2017 г. № 239;

– в информационных системах (ИС) общего пользования до 2 класса включительно в соответствии с документом «Требования о защите информации, содержащейся в информационных системах общего пользования», утвержденным приказом ФСБ России и ФСТЭК России от 31 августа 2010 г. № 416/489.

ППО состоит из следующих подсистем[3]:

– подсистема безопасности (ПБ);
– подсистема «Маркет» (ПМ);
– подсистема Платформа управления (ПУ);
– подсистема управления тенантами (ПУТ);
– подсистема Сервис уведомлений (ПСУ);
– подсистема обновления ОС (ПООС);
– подсистема доставки контента (CDN).

Взаимодействие между подсистемами и компонентами подсистем осуществляется с использованием протокола HTTP стандарт RFC 2616, при этом обмен данными осуществляется в формате RFC 8259 (JSON).

Для получения push-уведомлений на устройствах используется push-демон, входящий в состав ОС Аврора. Push-демон, в свою очередь, взаимодействует с ПСУ по защищенному протоколу TLS (RFC 5246, RFC 8446) с протоколом TCP (RFC 793) на транспортном уровне.

В качестве сервера базы данных (БД) используется сервер с установленной системой управления базами данных (СУБД) Postgres Pro или PostgreSQL, в которой хранятся данные ППО, для чего при развертывании создается специальная БД. Для хранения информации о сессиях используется СУБД Redis.

Подсистемы, входящие в состав ППО, позволяют выполнять логирование информационных сообщений, сообщений об ошибках, предупреждений и отладочной информации в системный журнал ОС (`systemd-journald`).

[3] Состав подсистем ППО зависит от комплектности вариантов поставки, которые определяются условиями Лицензионного договора.

1.1.1. Подсистема безопасности

ПБ предназначена для реализации следующих функций безопасности ППО:

– идентификации и аутентификации пользователей и устройств;

– управления идентификаторами пользователей и устройств;

– управления средствами аутентификации;

– управления учетными записями пользователей и устройств;

– управления доступом субъектов доступа к объектам доступа;

– регистрации событий безопасности;

– предоставления пользователям доступа к интерфейсу ПБ.

ПБ состоит из следующих компонентов:

– Консоль входа пользователей;

– Консоль администратора ПБ;

– Сервер приложений ПБ.

Консоль входа пользователей позволяет пользователям ППО осуществлять ввод идентификационной и аутентификационной информации.

Консоль администратора ПБ позволяет управлять учетными записями пользователей и работать с журналом регистрации событий.

Сервер приложений ПБ представляет собой совокупность веб-приложений, реализующих функции безопасности, а также позволяющих хранить в БД и предоставлять пользователям ППО доступ к данным об учетных записях и журналу регистрации событий.

1.1.2. Подсистема «Маркет»

ПМ предназначена для обеспечения:

– управления жизненным циклом приложений (загрузка, согласование, удаление и публикация);

– управления дистрибуцией опубликованных приложений (скачивание, установка, обновление и удаление);

– предоставления пользователям доступа к интерфейсу ПМ.

ПМ состоит из следующих компонентов:

– Консоль администратора ПМ;

– Консоль разработчика ПМ;

– Приложение «Аврора Маркет»;

– Сервер приложений ПМ.

Консоль администратора ПМ позволяет осуществлять взаимодействие Администратора Аврора Маркета с ПМ в части работы с приложениями.

Консоль разработчика ПМ позволяет добавлять новые и обновлять ранее загруженные приложения, а также получать доступ к хранимой в них информации.

Приложение «Аврора Маркет» выполняется на устройствах, функционирующих под управлением ОС, и служит для отображения данных о приложениях, а также для их загрузки, установки, обновления и удаления.

Сервер приложений ПМ представляет собой совокупность веб-приложений, позволяющих хранить в БД и предоставлять пользователям ППО информацию о приложениях. При этом сами приложения, их значки и скриншоты хранятся в файловом хранилище.

1.1.3. Подсистема Платформа управления

ПУ предназначена для обеспечения:

– управления отдельными устройствами (оперативное управление) и группами устройств;

– управления политиками, офлайн-сценариями;

– управления записями об устройствах и пользователях устройств;

– управления приложениями на устройствах;

– контроля состояния устройств;

– контроля применения политик на устройствах;

– мониторинга событий и предоставления отчетности;

– предоставления пользователям доступа к интерфейсу ПУ.

ПУ состоит из следующих компонентов:

– Консоль администратора ПУ;

– Приложение «Аврора Центр»;

– Сервер приложений ПУ.

Консоль администратора ПУ позволяет осуществлять взаимодействие Администратора Платформы управления с ПУ.

Приложение «Аврора Центр» выполняется на устройствах, функционирующих под управлением ОС, и позволяет осуществлять взаимодействие ПУ с устройством, а также в зависимости от управляющего сообщения или назначенного офлайн-сценария, полученного от Сервера приложений ПУ, имеет возможность управлять различными функциями устройства.

Сервер приложений ПУ представляет собой совокупность веб-приложений, позволяющих хранить в БД и предоставлять пользователям ППО данные о настройках и конфигурации ОС, а также формировать управляющие сообщения и офлайн-сценарии для приложения «Аврора Центр».

1.1.4. Подсистема управления тенантами

ПУТ предназначена для обеспечения:

– управления жизненным циклом тенантов (создание, редактирование и удаление тенантов);

– управления организациями;

– управления контактными лицами организаций.

ПУТ состоит из следующих компонентов:

– Консоль администратора ПУТ;

– Сервер приложений ПУТ.

Консоль администратора ПУТ позволяет осуществлять взаимодействие Администратора тенантов с ПУТ.

Сервер приложений ПУТ представляет собой совокупность веб-приложений, позволяющих хранить в БД и предоставлять пользователям ППО данные о тенантах, а также осуществлять управление тенантами.

1.1.5. Подсистема Сервис уведомлений

ПСУ предназначена для обеспечения:

– доставки push-уведомлений до устройств;

– управления жизненным циклом проектов (добавление, настройка и удаление);

– предоставления пользователям доступа к интерфейсу ПСУ.

ПСУ состоит из следующих компонентов:

– Консоль администратора ПСУ;

– Сервер приложений ПСУ.

Консоль администратора ПСУ позволяет осуществлять взаимодействие Администратора Сервиса уведомлений с ПСУ в части управления жизненным циклом проектов. При этом проекты содержат следующую информацию:

– настройки взаимодействия ПСУ и сервера приложений;

– информацию о приложениях, push-уведомления которых требуется передавать с сервера приложений на устройства;

– информацию о контактных лицах.

Сервер приложений ПСУ представляет собой совокупность веб-приложений, позволяющих хранить в БД и предоставлять субъектам доступа ППО информацию о проектах, а также реализует функционал доставки push-уведомлений до устройств посредством tcp-сервера.

1.1.6. Подсистема обновления ОС

ПООС предназначена для обеспечения:

– предоставления информации о пакетах ОС;

– управления дистрибуцией пакетов ОС.

ПООС состоит из следующего компонента:

– Сервер приложений ПООС.

Сервер приложений ПООС представляет собой совокупность веб-приложений, позволяющих хранить в БД и предоставлять информацию и адреса хранения пакетов загрузочного модуля ОС.

Для хранения и дистрибуции пакетов ОС применяется файловый сервер, развернутый с использованием Nginx.

1.1.7. Подсистема доставки контента

CDN является опциональной подсистемой ППО и предназначена для оптимизации доставки контента ППО (установочные файлы приложений, значки, скриншоты, пакеты обновления ОС) путем их размещения (кеширования) на контент-серверах таким образом, чтобы время ожидания для пользователя было минимальным.

CDN состоит из следующего компонента:

– контент-сервера (контент-серверов).

Контент-сервер представляет собой веб-приложение, позволяющее кешировать в файловом хранилище контент ППО и предоставлять к нему доступ приложениям «Аврора Центр» и «Аврора Макет», а также ОС Аврора.

1.2. Субъекты доступа и права на доступ к интерфейсам ППО

1.2.1. Субъекты доступа (роли) ППО

Субъектами доступа являются пользователи ППО и процессы без участия пользователей, при этом субъектам доступа может быть назначена одна или несколько ролей, позволяющих выполнять следующие действия:

– Администратор учетных записей – управлять учетными записями пользователей (наличие роли обязательно);

– Оператор аудита – работать с журналом регистрации событий;

– Администратор Аврора Маркета – управлять ПМ через интерфейс ППО;

– Разработчик – добавлять новые, обновлять ранее загруженные приложения и получать информацию о них;

– Редактор приложений – обновлять и получать информацию о ранее загруженных приложениях;

– Пользователь Аврора Маркета – загружать приложения и получать информацию о них;

– Администратор Платформы управления – управлять ПУ через интерфейс ППО;

– Приложение «Аврора Центр» (процесс без участия пользователей) – назначается учетным записям приложения «Аврора Центр»;

– Администратор тенантов – управлять ПУТ через интерфейс ППО;

– Администратор Сервиса уведомлений - управлять жизненным циклом проектов;

– Мобильное приложение (процесс без участия пользователей) – получать push-уведомления;

– Сервер приложений – назначается серверам приложений для передачи push-уведомлений на Сервер приложений ПСУ.

1.2.2. Права на доступ к интерфейсам ППО

Права на доступ к соответствующим разделам интерфейса ППО приведены в таблице (Таблица 1).

Таблица 1

Интерфейс ППО		Права на доступ	
Раздел	Подраздел	Подсистема	Консоль/Субъект доступа
Мультитенант	Тенанты	ПУТ	Консоль администратора ПУТ Администратор тенантов
	Организации	ПУТ	Консоль администратора ПУТ Администратор тенантов

Интерфейс ППО		Права на доступ	
Раздел	Подраздел	Подсистема	Консоль/Субъект доступа
Мониторинг	Индикаторы	ПУ	Консоль администратора ПУ Администратор Платформы управления
	Аудит	ПБ	Консоль администратора ПБ Оператор аудита
Управление	Устройства	ПУ	Консоль администратора ПУ Администратор Платформы управления
	Пользователи	ПУ	Консоль администратора ПУ Администратор Платформы управления
	Политики	ПУ	Консоль администратора ПУ Администратор Платформы управления
	Сценарии	ПУ	Консоль администратора ПУ Администратор Платформы управления
	Файлы	ПУ	Консоль администратора ПУ Администратор Платформы управления
	Приложения	ПМ	Консоль администратора ПМ Администратор Аврора Маркета
	Витрины	ПМ	Консоль администратора ПМ Администратор Аврора Маркета
	Связки ключей	ПМ	Консоль администратора ПМ Администратор Аврора Маркета
Администрирование	Учетные записи	ПБ	Консоль администратора ПБ Администратор учетных записей
	Настройки	ПУ	Консоль администратора ПУ Администратор Платформы управления
	Орг. структура	ПУ	Консоль администратора ПУ Администратор Платформы управления
	Версии ОС	ПМ	Консоль администратора ПМ Администратор Аврора Маркета
Консоль разработчика ПМ		ПМ	Консоль разработчика ПМ Разработчик, Редактор приложений

Интерфейс ППО		Права на доступ	
Раздел	Подраздел	Подсистема	Консоль/Субъект доступа
Проекты		ПСУ	Консоль администратора ПСУ Администратор Сервиса уведомлений
Приложение «Аврора Маркет» для ОС Аврора		ПМ	Пользователь Аврора Маркета
Приложение «Аврора Центр» для ОС Аврора		ПУ	Процесс приложения «Аврора Центр»

1.3. Описание принципов безопасной работы средства

1.3.1. Общая информация

ППО реализует следующие функции безопасности:

– идентификация и аутентификация субъектов доступа и объектов доступа;

– управление доступом субъектов доступа к объектам доступа;

– регистрация событий безопасности.

При использовании ППО необходимо выполнение следующих мер по защите информации от несанкционированного доступа (НСД):

– соблюдение парольной политики;

– соблюдение требования, согласно которому пароль не должен включать в себя легко вычисляемые сочетания символов;

– отсутствие у пользователя права передачи личного пароля третьим лицам;

– обязанность пользователя при вводе пароля исключить возможность его перехвата третьими лицами и техническими средствами.

При эксплуатации ППО запрещается:

– оставлять без контроля незаблокированные программные средства и/или ППО;

– разглашать пароли, выводить пароли на дисплей, принтер или иные средства отображения информации.

1.3.2. Компрометация паролей

Под компрометацией паролей необходимо понимать следующее:

– физическую утрату носителя с парольной информацией;

– передачу идентификационной информации по открытым каналам связи;

– перехват пароля при распределении идентификаторов;

– сознательную передачу информации третьим лицам.

ПРИМЕЧАНИЕ. При компрометации пароля пользователь обязан незамедлительно оповестить Администратора учетных записей.

1.3.3. Описание параметров (настроек) безопасности средства, доступных каждой роли пользователей, и их безопасные значения

Настройки параметров безопасности ППО доступны только пользователям с ролью Администратор учетных записей и заключаются в возможности управления ролями пользователей ППО.

Пользователям ППО должны назначаться минимальные права и привилегии, необходимые для выполнения ими своих должностных обязанностей (функций).

1.4. Условия выполнения

Для функционирования ППО необходимы описанные в настоящем подразделе программно-технические средства.

1.4.1. Аппаратные характеристики

Аппаратные характеристики приведены в разделе 2.

1.4.2. Программные характеристики

В таблице (Таблица 2) приведены программные характеристики серверов приложений ППО.

Таблица 2

Параметр	Значение	Информация о лицензии
Операционная система	CentOS версии 7	GNU General Public License, version 2
	Ubuntu версии 20.04	Free software, plus some proprietary device drivers
	Ubuntu версии 22.04	Free software, plus some proprietary device drivers
	Debian версии 11	DFSG-compatible licenses, plus proprietary firmware files
	Debian версии 12	DFSG-compatible licenses, plus proprietary firmware files
	Альт 8 СП[4] релиз 10	Коммерческая
	Альт Сервер 10.2	
	РЕД ОС 7.3[5] (сертифицированный)	
	РЕД ОС 7.3	
	Astra Linux Special Edition 1.7[6] (Смоленск)	
Балансировщик сервисов	Nginx Web Server версии 1.22.0 или выше	2-clause BSD-like license
Система обнаружения сервисов	Consul версии 1.16.6 или выше	Mozilla Public License, version 2.0
Средство управления конфигурациями сервисов	Consul Template версии 0.25.1 или выше	Mozilla Public License, version 2.0
Сервис гарантированной доставки сообщений	Redpanda версии 23.3.9 или выше	Redpanda Business Source License 1.1 (BSL 1.1)
Приложение для синхронизации файлов	Syncthing версии 1.25.0 или выше	Mozilla Public License, version 2.0
Прикладное программное обеспечение	ППО «Аврора Центр»	Коммерческая

[4] Сертификат соответствия ФСТЭК России № 3866, действителен до 10 августа 2028 г.

[5] Сертификат соответствия ФСТЭК России № 4060, действителен до 12 января 2024 г. (окончание срока технической поддержки 31.12.2030 г.).

[6] Сертификат соответствия ФСТЭК России № 2557, действителен до 27 января 2026 г.

В таблице (Таблица 3) приведены программные характеристики серверов БД.

Таблица 3

Параметр	Значение	Информация о лицензии
Операционная система	CentOS версии 7	GNU General Public License, version 2
	Ubuntu версии 20.04	Free software, plus some proprietary device drivers
	Ubuntu версии 22.04	Free software, plus some proprietary device drivers
	Debian версии 11	DFSG-compatible licenses, plus proprietary firmware files
	Debian версии 12	DFSG-compatible licenses, plus proprietary firmware files
	Альт 8 СП[7] релиз 10	
	Альт Сервер 10.2	
	РЕД ОС 7.3 (сертифицированный)	
	РЕД ОС 7.3	
	Astra Linux Special Edition 1.7 (Смоленск)	
СУБД	Postgres Pro Certified 14.11.2[8] или выше	Коммерческая
	Postgres Pro Certified 15.6.2 или выше	
	Postgres Pro Enterprise Certified 13.14.2[9] или выше	
	Postgres Pro Standard 12.18.2 или выше	
	Postgres Pro Standard 13.14.2 или выше	
	Postgres Pro Standard 14.11.2 или выше	
	Postgres Pro Standard 15.6.1 или выше	
	PostgreSQL 12.18 или выше	PostgreSQL License
	PostgreSQL 13.14 или выше	

[7] Сертификат соответствия ФСТЭК России № 3866, действителен до 10 августа 2028 г.

[8] Сертификат соответствия ФСТЭК России № 3637, действителен до 05 октября 2024 г.

[9] Сертификат соответствия ФСТЭК России № 4063, действителен до 16 января 2029 г.

Параметр	Значение	Информация о лицензии
	PostgreSQL 14.12 или выше	
	PostgreSQL 15.7 или выше	
СУБД для хранения сессий	Redis версии 7.2.4 или выше	BSD-3-Clause License
Расширение СУБД PostgreSQL для партиционирования таблиц БД	PG Partition Manager (`pg_partman`) версии 4.6 или выше	PostgreSQL License
Планировщик задач для PostgreSQL	`pg_cron` версии 1.5.2 или выше	PostgreSQL License
Сервис для управления кластером Postgresql	Patroni версии 3.3.0 или выше	The MIT License (MIT)
Сервис для балансировки нагрузки и обеспечения отказоустойчивости	Keepalived	GNU General Public License, version 2

В таблице (Таблица 4) приведены программные характеристики устройств.

Таблица 4

Параметр	Значение
Операционная система	ОС Аврора, имеющая действительный сертификат соответствия ФСТЭК России
Прикладное программное обеспечение	– приложение «Аврора Центр»; – приложение «Аврора Маркет»

1.4.3. Требования к рабочим местам пользователей

ПРИМЕЧАНИЕ. Для работы пользователей с интерфейсом ППО необходимо выполнение следующих условий:

– веб-браузер должен поддерживать следующие технологии: TLS, CSS3, HTML5, ECMAScript 5 и Cookie. Рекомендуется использовать веб-браузер Chrome версии 90 или выше;

– веб-браузер в информационных системах, обрабатывающих информацию ограниченного доступа, требующую защиты в соответствии с законодательством РФ необходимо использовать из состава ОС, имеющей сертификат соответствия ФСТЭК России. Рекомендуется использовать веб-браузеры: Firefox ESR версии 91.4 или выше, Chromium версии 87 или выше;

– разрешение экрана монитора не менее 1280x960 px.

1.4.4. Варианты конфигураций, для которых проводилось тестирование

Варианты конфигурации среды функционирования, в которых проводилось тестирование ППО, приведены в таблице (Таблица 5).

Таблица 5

ОС	СУБД	СЗИ НСД
CentOS-7.6	PostgreSQL 12.18	СПО СЗИ НСД «Аккорд-Х К»
CentOS-7.6	PostgreSQL 13.14	
CentOS-7.6	PostgreSQL 15.7	
CentOS-7.6	Postgres Pro Standard 12.18.2	
CentOS-7.6	Postgres Pro Standard 14.11.2	
Альт Сервер 10.2	PostgreSQL 12.18	
Альт Сервер 10.2	PostgreSQL 13.14	
Альт Сервер 10.2	PostgreSQL 14.12	
Альт Сервер 10.2	PostgreSQL 15.6	
Альт 8 СП релиз 10	PostgreSQL 15.6	
Альт 8 СП релиз 10	Postgres Pro Standard 12.18.2	
Альт 8 СП релиз 10	Postgres Pro Standard 13.14.2	
Альт 8 СП релиз 10	Postgres Pro Certified (версия ядра postgres: 15.6.2)	
Альт 8 СП релиз 10	Postgres Pro Enterprise Certified (версия ядра postgres: 13.13.1)	
РЕД ОС 7.3 (сертифицированный)	Postgres Pro Standard 13.14.2	
РЕД ОС 7.3 (сертифицированный)	Postgres Pro Certified (версия ядра postgres: 14.11.2)	
РЕД ОС 7.3 (сертифицированный)	Postgres Pro Certified (версия ядра postgres: 15.6.2)	
РЕД ОС 7.3.4	PostgreSQL 12.18	

ОС	СУБД	СЗИ НСД
РЕД ОС 7.3.4	PostgreSQL 13.14	
РЕД ОС 7.3.4	PostgreSQL 14.11	
РЕД ОС 7.3.4	PostgreSQL 15.6	
РЕД ОС 7.3.4	Postgres Pro Standard 15.6.2	
Astra Linux Special Edition 1.7 (Смоленск)	PostgreSQL 12.18	
Astra Linux Special Edition 1.7 (Смоленск)	PostgreSQL 13.14	
Astra Linux Special Edition 1.7 (Смоленск)	PostgreSQL 14.12	
Astra Linux Special Edition 1.7 (Смоленск)	PostgreSQL 15.7	
Astra Linux Special Edition 1.7 (Смоленск)	Postgres Pro Certified (версия ядра postgres: 14.11.2)	
Astra Linux Special Edition 1.7 (Смоленск)	Postgres Pro Certified (версия ядра postgres: 15.6.2)	
Astra Linux Special Edition 1.7 (Смоленск)	Postgres Pro Enterprise Certified (версия ядра postgres: 13.13.1)	
Ubuntu 20.04	PostgreSQL 14.12	
Ubuntu 20.04	PostgreSQL 15.7	
Ubuntu 22.04	PostgreSQL 14.12	
Ubuntu 22.04	PostgreSQL 15.7	
Debian 11.9	PostgreSQL 14.12	
Debian 11.9	PostgreSQL 15.7	
Debian 12.5	PostgreSQL 14.12	
Debian 12.5	PostgreSQL 15.7	

2. АРХИТЕКТУРА ППО И ВАРИАНТЫ УСТАНОВКИ ППО

2.1. Описание компонентов

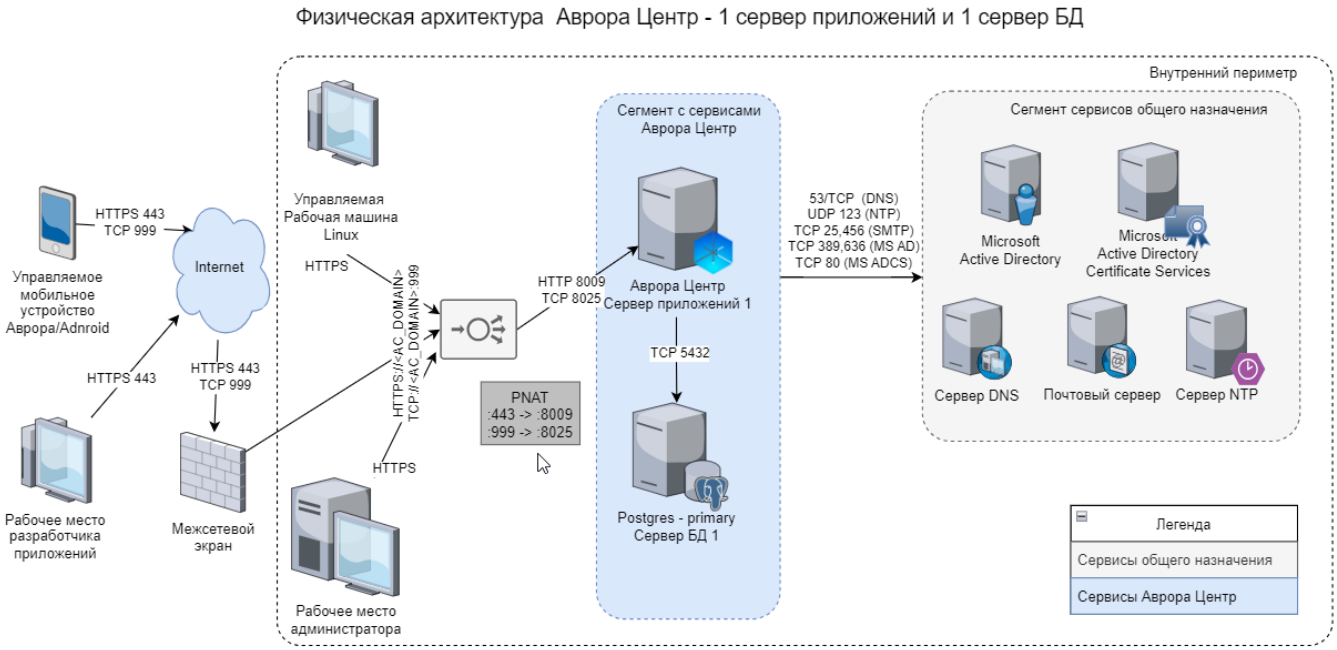


Рисунок 1

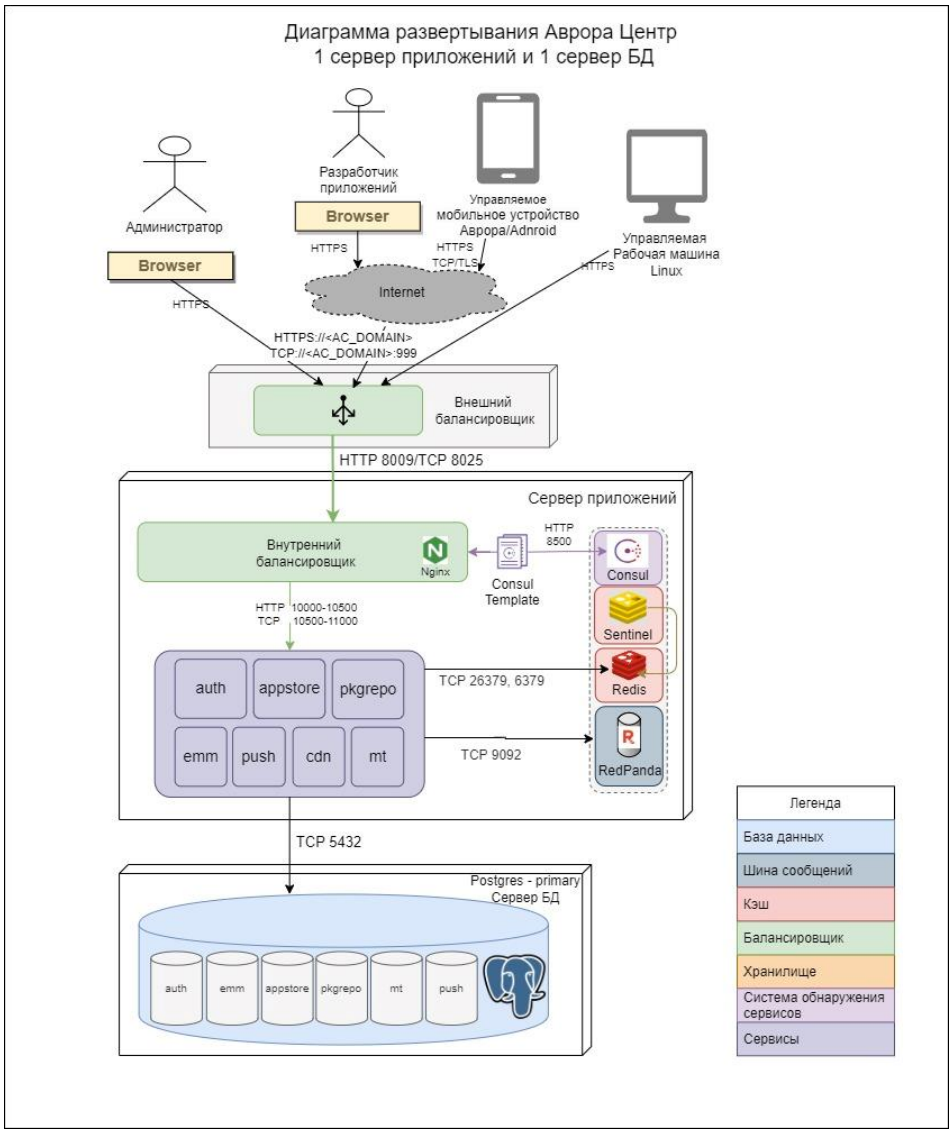


Рисунок 2

2.1.1. Сервер приложений

2.1.1.1. ППО «Аврора Центр»

ППО состоит из следующих подсистем: ПБ (auth), ПМ (appstore), ПООС (pkgrepo), ПУ (emm), ПУТ (mt), CDN (cdn), ПСУ (push). Каждая подсистема представляет собой набор сервисов.

Сервисы разных подсистем могут быть развернуты как на одних и тех же нодах сервера приложений, так и на отдельных. Каждый сервис работает только со своими данными, которые лежат в отдельных схемах в БД. Схемы могут быть развернуты как в общей БД, так и в разных.

Сервисы взаимодействуют между собой посредством HTTP запросов и обмена сообщениями через шину Redpanda. Все межсервисные запросы аутентифицированы.

Для корректной работы сервисов подсистем ПМ, ПУ и ПООС в кластере к нодам сервера приложений необходимо примонтировать файловое хранилище согласно п. 3.8.2.

2.1.1.2. Внутренний балансировщик

На каждом сервере приложений устанавливается внутренний балансировщик Nginx. Он выполняет функцию `reverse proxy`, служит для балансировки и передачи трафика к сервисам ППО, а также обеспечивает межсервисное взаимодействие. Для некоторых конечных точек настроено кэширование запросов к сервисам.

Nginx настраивается автоматически на основе информации о статусе сервисов из системы обнаружения сервисов Consul.

С целью защиты серверов приложений от превышения предельной нагрузки на интерфейсах, обрабатывающих внешние запросы с приложения «Аврора Центр», настроенно ограничение одновременно обрабатываемых запросов (троттлинг).

2.1.1.3. Система обнаружения сервисов Consul

Система обнаружения сервисов используется для мониторинга состояния сервисов ППО. Consul должен устанавливаться на нечетном количестве серверов.

2.1.1.4. Средство управления конфигурациями сервисов Consul Template

Средство управления конфигурациями служит для автоматической настройки распределения запросов между экземплярами сервисов за счет изменения конфигурации Nginx на основе информации о статусе сервисов ППО, получаемой из системы обнаружения сервисов Consul.

2.1.1.5. Сервис гарантированной доставки сообщений Redpanda

Сервис гарантированной доставки сообщений используется для обмена сообщениями между сервисами ППО. Сервис RedPanda должен устанавливаться на нечетном количестве серверов.

2.1.1.6. СУБД Redis

СУБД Redis используется для хранения веб-сессий.

Для управления отказоустойчивой конфигурацией Redis используется сервис Sentinel, который должен устанавливаться на нечетном количестве серверов.

2.1.2. Сервер БД

В качестве сервера БД используется СУБД PostgreSQL или PostgresPro. Для работы так же требуется установка расширений `pg_partman` для автоматического партиционирования и очистки накапливающихся данных, и `pg_cron` для выполнения функций в БД по расписанию.

В ненагруженных конфигурациях все данные размещаются в одном физическом инстансе БД. Внутри создаются логические БД для каждой из подсистем ППО. Данные сервисов внутри логических баз размещаются в отдельных схемах, что исключает возможность обращения одних сервисов к данным других сервисов. Описанная конфигурация снижает связность между сервисами и при необходимости позволяет вынести данные подсистем или отдельных сервисов на выделенные серверы БД.

В высоконагруженных конфигурациях рекомендуется выделить данные наиболее нагруженных подсистем ПБ (auth) и ПУ (emm) в отдельные физические базы для минимизации взаимного влияния друг на друга.

БД ПСУ (push) следует выносить в отдельный инстанс при высокой интенсивности запросов, более 500rps.

Сервер БД может быть установлен как с помощью сценариев установки ППО, так и самостоятельно. В сценарии установки включены минимальные возможности по установке `primary` и `replica` серверов, а также настройка репликации. Автоматического переключения с основного на резервный сервер БД в случае сбоя не предусмотрено. Поэтому, в случае отказа основного сервера БД, переключение на резервный сервер необходимо выполнить вручную.

2.1.3. Внешний балансировщик

Внешний балансировщик используется для распределения трафика между нодами сервера приложений. Это позволяет сбалансировать нагрузку между нодами сервера приложений и перенаправить трафик на доступные ноды в случае выхода из строя одного из серверов приложений.

Внешний балансировщик не входит в состав ППО, определяется и разворачивается пользователями самостоятельно. В директории `samples` в дистрибутиве имеется пример конфигурационного файла для балансировщика Nginx для 3-х нодовой конфигурации сервера приложений.

На внешнем балансировщике может быть настроена защита канала связи (протокол HTTPS), поддерживаются в том числе ГОСТ алгоритмы. Также существует возможность отделить консоль администратора от других интерфейсов, выделив отдельный домен (поддомен) или порт.

2.1.4. Внешние службы

В данном пункте приведены описания внешних служб, которые не устанавливаются вместе с основными компонентами.

2.1.4.1. Сервер DNS

Сервер DNS используется для получения информации о доменах (IP-адреса по имени хоста ЭВМ или устройства).

2.1.4.2. Сервер NTP

Сервер NTP используется для автоматической синхронизации времени на всех серверах.

2.1.4.3. Почтовый сервер

Почтовый сервер используется для рассылки кодов активации устройств, получения диагностических отчетов с устройств и т. д.

2.1.4.4. Microsoft Active Directory

Интеграция ППО с Microsoft Active Directory используется для автоматической синхронизации списка пользователей устройств в ПУ со списком пользователей организации, а также для автоматической привязки устройства к пользователю.

2.1.4.5. Службы сертификатов Active Directory (Active Directory Certificate Services)

Службы сертификатов Active Directory используются для управления ключевой информацией (закрытый ключ, открытый ключ, сертификат открытого ключа) пользователей устройств.

2.2. Внешние интерфейсы сервера приложений ППО

Перед нодами сервера приложений могут располагаться различные компоненты сетевой инфраструктуры (например, межсетевой экран, внешний балансировщик, средство криптографической защиты информации и др.), которые обрабатывают поступающие к серверу приложений запросы.

По умолчанию ноды сервера приложений для обработки внешних запросов используют специальные выделенные порты, взаимодействие с которыми осуществляется по не защищенному протоколу.

Для того чтобы ППО было доступно из внешней сети на компонентах сетевой инфраструктуры необходимо настроить переадресацию портов.

Пример таблицы переадресации портов (Port NAT, PNAT) приведен на физической архитектуре, а также в таблице (Таблица 6).

Таблица 6

Назначение порта	Порт сервера приложений	Порт СКЗИ
Обработка запросов консолей пользователей/администраторов, а также запросов приложений «Аврора Центр» и «Маркет»	8009 (http)	443 (HTTPS)
Обработка запросов контент-серверов	8024 (http)	8443 (HTTPS)
Обработка запросов от устройств к ПСУ для получения push-уведомлений	8025 (tcp)	999 (tls)

Для обращения к ППО имеет смысл завести отдельный домен <AC_DOMAIN> и назначить его на самый первый компонент сетевой инфраструктуры, который принимает запросы от пользователей ППО и устройств.

Домен <AC_DOMAIN> и внешние порты должны быть указаны в соответствующих конфигурационных файлах ППО в процессе его настройки.

2.3. Варианты установки ППО

В зависимости от требований к количеству поддерживаемых устройств применяются различные варианты установки ППО.

Для запуска ППО и СУБД на 1 сервере требуются следующие минимальные аппаратные характеристики:

- 3 ядра процессора;
- 5 ГБ оперативной памяти;
- 50 ГБ свободного места на жестком диске.

Данную конфигурацию рекомендуется использовать для ознакомления с функционалом ППО и иных случаях, где не предъявляются требования к производительности.

В таблице (Таблица 7) приведены требования к аппаратным характеристикам серверов приложений ППО.

Таблица 7

Параметр	Количество устройств				
	10 000	50 000	100 000	200 000	500 000
Процессор, количество ядер	2	4	6	12	12
Объем оперативной памяти, ГБ	6	8	8	10	12
Объем жесткого диска HDD, ГБ	50	50	110	130	160
iops	100	100	100	100	100
Скорость сети, Мбайт/с	50	50	90	200	220
Количество серверов	3	3	3	3	6

В таблице (Таблица 8) приведены требования к аппаратным характеристикам серверов БД.

Таблица 8

Параметр	Количество устройств							
				200 000		500 000		
	10 000	50 000	100 000	ПБ	ПМ, ПУ, ПУТ, ПООС, ПСУ	ПБ	ПМ, ПУ, ПУТ, ПООС	ПСУ
Процессор, количество ядер	2	4	6	3	10	4	18	8
Объем оперативной памяти, ГБ	4	6	8	12	12	16	24	12
Объем жесткого диска SSD, ГБ	700	3200	6300	7200	6300	18300	16800	2000
iops	200	200	200	200	800	200	2000	5000
Скорость сети, Мбайт/с	20	50	80	90	120	90	150	150
Количество серверов	2	2	2	2	2	2	2	2

В таблице (Таблица 9) приведены требования к аппаратным характеристикам серверов с компонентами среды функционирования ППО.

Таблица 9

Параметр	Количество устройств				
	10 000	50 000	100 000	200 000	500 000
Процессор, количество ядер	Компоненты среды функционирования ППО располагаются на сервере приложений ППО			4	6
Объем оперативной памяти, ГБ				8	8
Объем жесткого диска SSD, ГБ				50	70
iops				2000	3000
Количество серверов				3	3

В таблице (Таблица 10) приведены требования к аппаратным характеристикам серверов с внешним балансировщиком (в случае использования Nginx).

Таблица 10

Параметр	Количество устройств				
	10 000	50 000	100 000	200 000	500 000
Процессор, количество ядер	2	2	2	4	6
Объем оперативной памяти, ГБ	2	4	4	6	6
Объем жесткого диска HDD, ГБ	90	90	50	30	50
iops	30	30	30	30	50
Количество серверов	2	2	4	6	6

В таблице (Таблица 11) приведены требования к аппаратным характеристикам контент-серверов.

Таблица 11

Параметр	Количество устройств				
	10 000	50 000	100 000	200 000	500 000
Процессор, количество ядер	-	2	1	2	4
Объем оперативной памяти, ГБ	-	4	4	6	8
Объем жесткого диска HDD, ГБ	-	150	150	200	200
iops	-	100	100	200	200
Количество серверов	-	2	4	4	6

В таблице (Таблица 12) приведены требования к аппаратным характеристикам серверов с внешним балансировщиком для контент-серверов (в случае использования Nginx).

Таблица 12

Параметр	Количество устройств				
	10 000	50 000	100 000	200 000	500 000
Процессор, количество ядер	-	4	2	4	4
Объем оперативной памяти, ГБ	-	12	6	6	4
Объем жесткого диска HDD, ГБ	-	20	20	20	50
iops	-	30	30	30	50
Количество серверов	-	2	4	6	6

Далее приведены схемы установки ППО.

2.3.1. Конфигурация все в одном (ППО и СУБД на одном сервере)

Сервисы ППО, инфраструктурные компоненты (компоненты среды функционирования) ППО и сервер БД установлены на одном сервере (Рисунок 3).

ПРИМЕЧАНИЕ. Внешний балансировщик и СКЗИ не используются.

Данную конфигурацию рекомендуется использовать для ознакомления с функционалом ППО и иных случаях, где не предъявляются требования к производительности, безопасности и надежности (Рисунок 4).

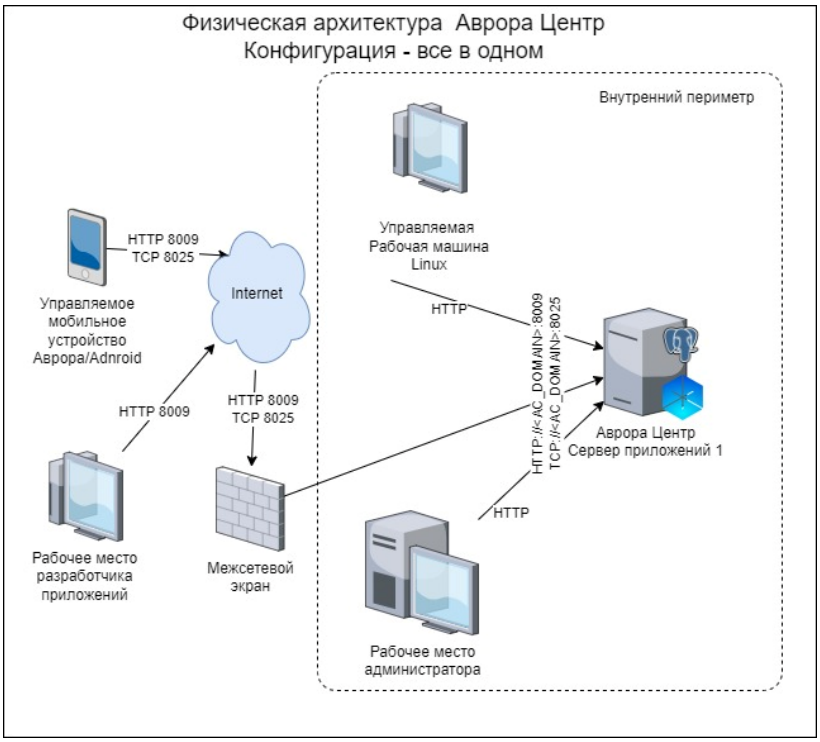


Рисунок 3

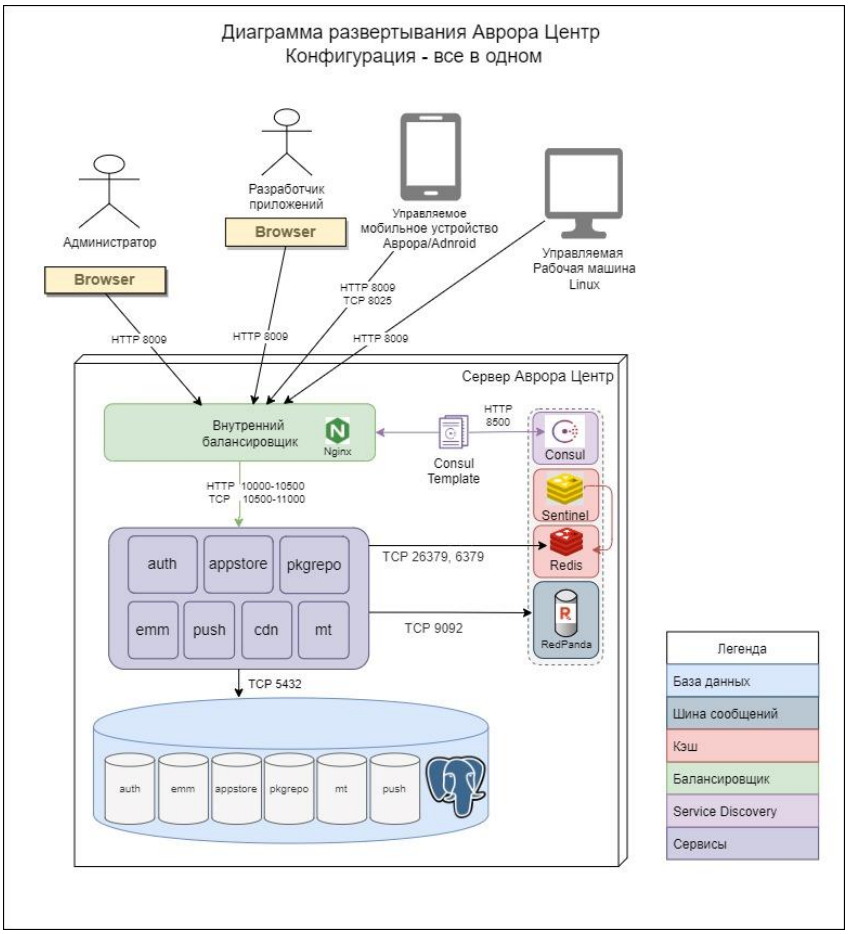


Рисунок 4

2.3.2. Конфигурация из 1 сервера приложений и 1 сервера БД

Сервисы ППО и инфраструктурные компоненты ППО установлены на одном сервере, а сервер БД установлен на отдельном сервере. Также настроены внешний балансировщик, на котором задан внешний домен ППО, и СКЗИ для защиты канала связи.

Данную конфигурацию рекомендуется использовать в качестве тестовой, либо в случаях, когда не предъявляются требования к отказоустойчивости и производительности (Рисунок 5, Рисунок 6).

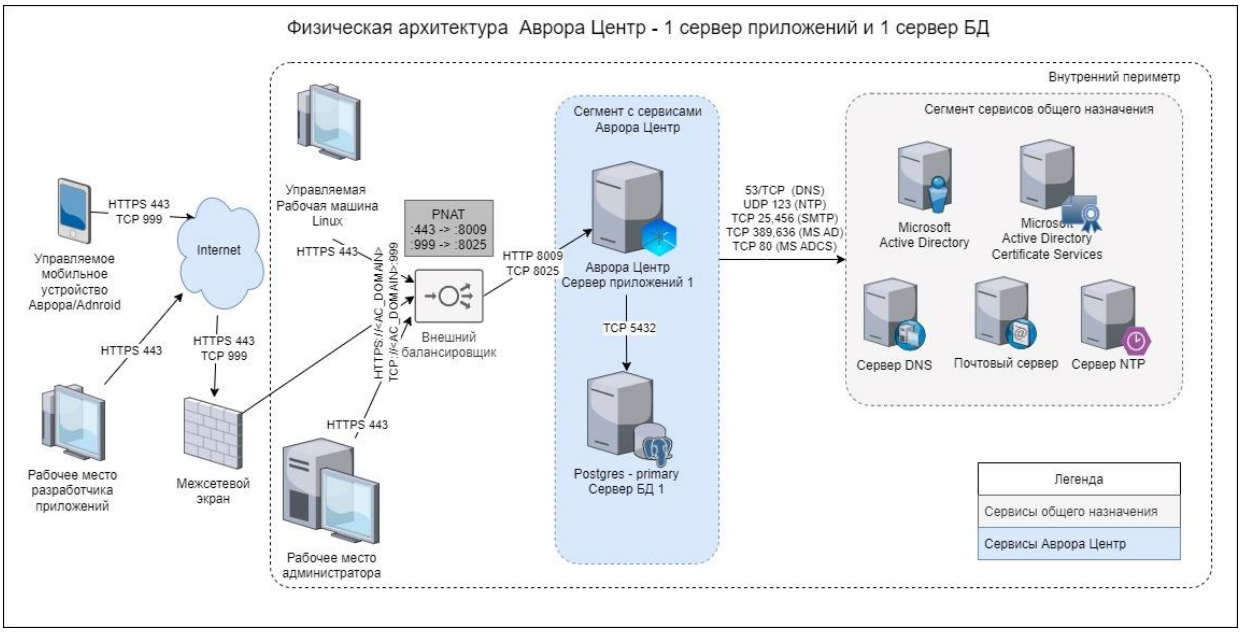


Рисунок 5

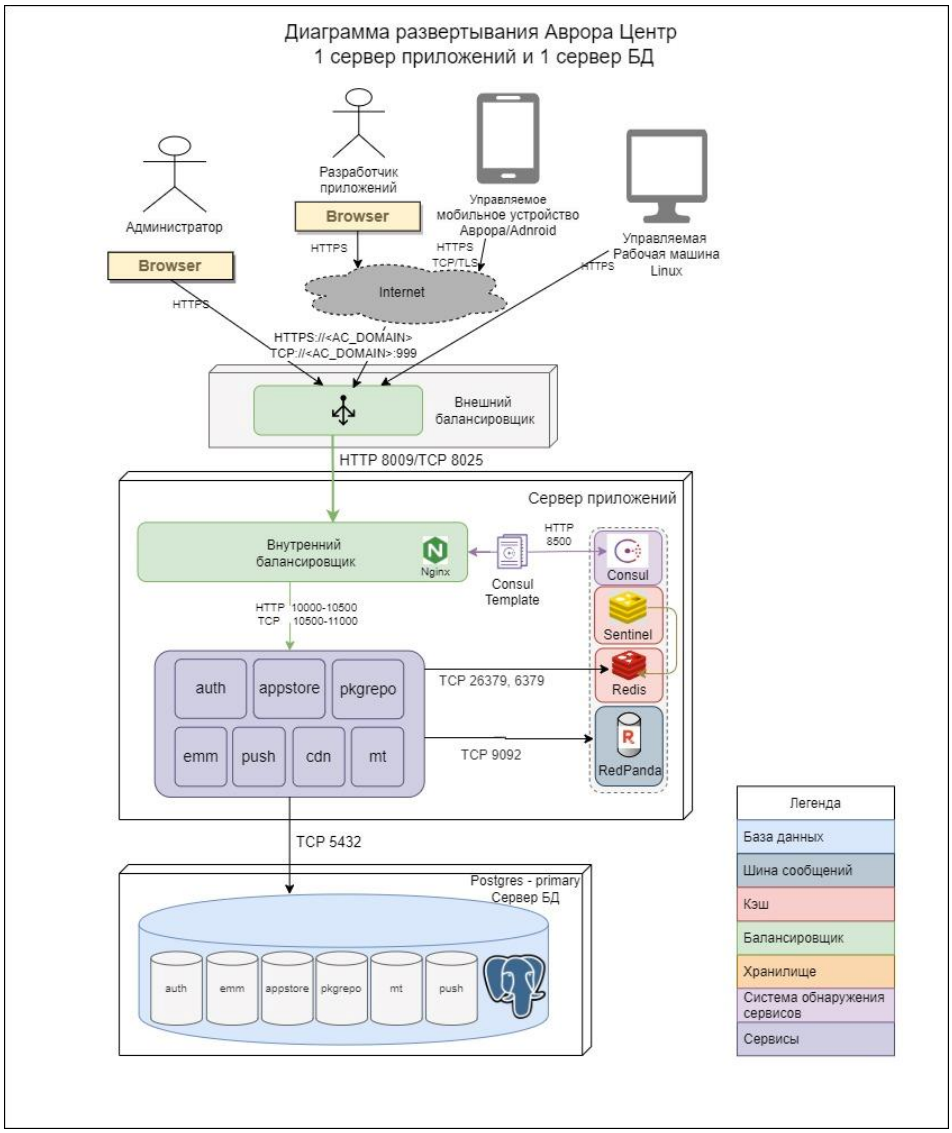


Рисунок 6

2.3.3. Кластерная конфигурация (поддержка до 10000 устройств)

В отличии от предыдущей конфигурации в данной конфигурации для обеспечения отказоустойчивости сервисы ППО и инфраструктурные компоненты ППО собираются в кластер из трех нод, каждая из которых обрабатывает запросы. Также устанавливается резервный сервер БД.

Данную схему установки рекомендуется использовать в качестве отказоустойчивой конфигурации, поддерживающей до 10000 устройств (Рисунок 7, Рисунок 8).

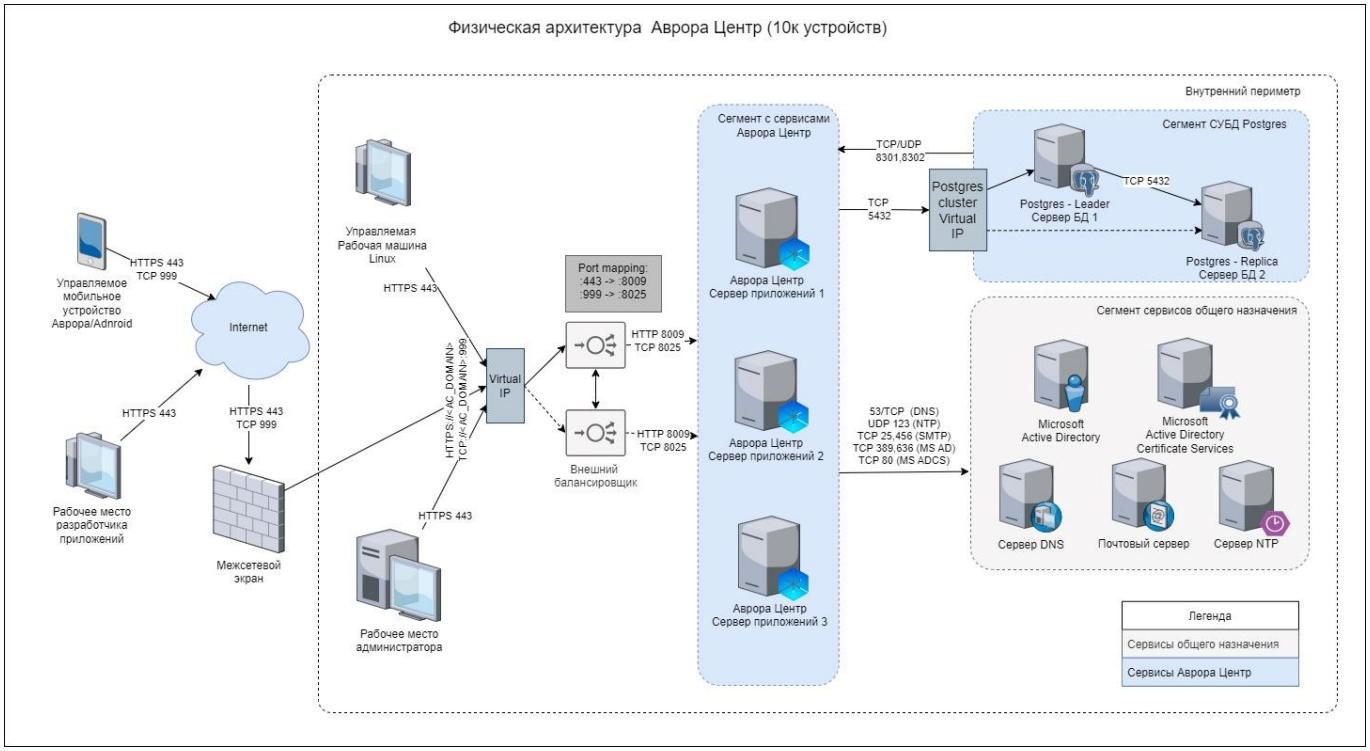


Рисунок 7

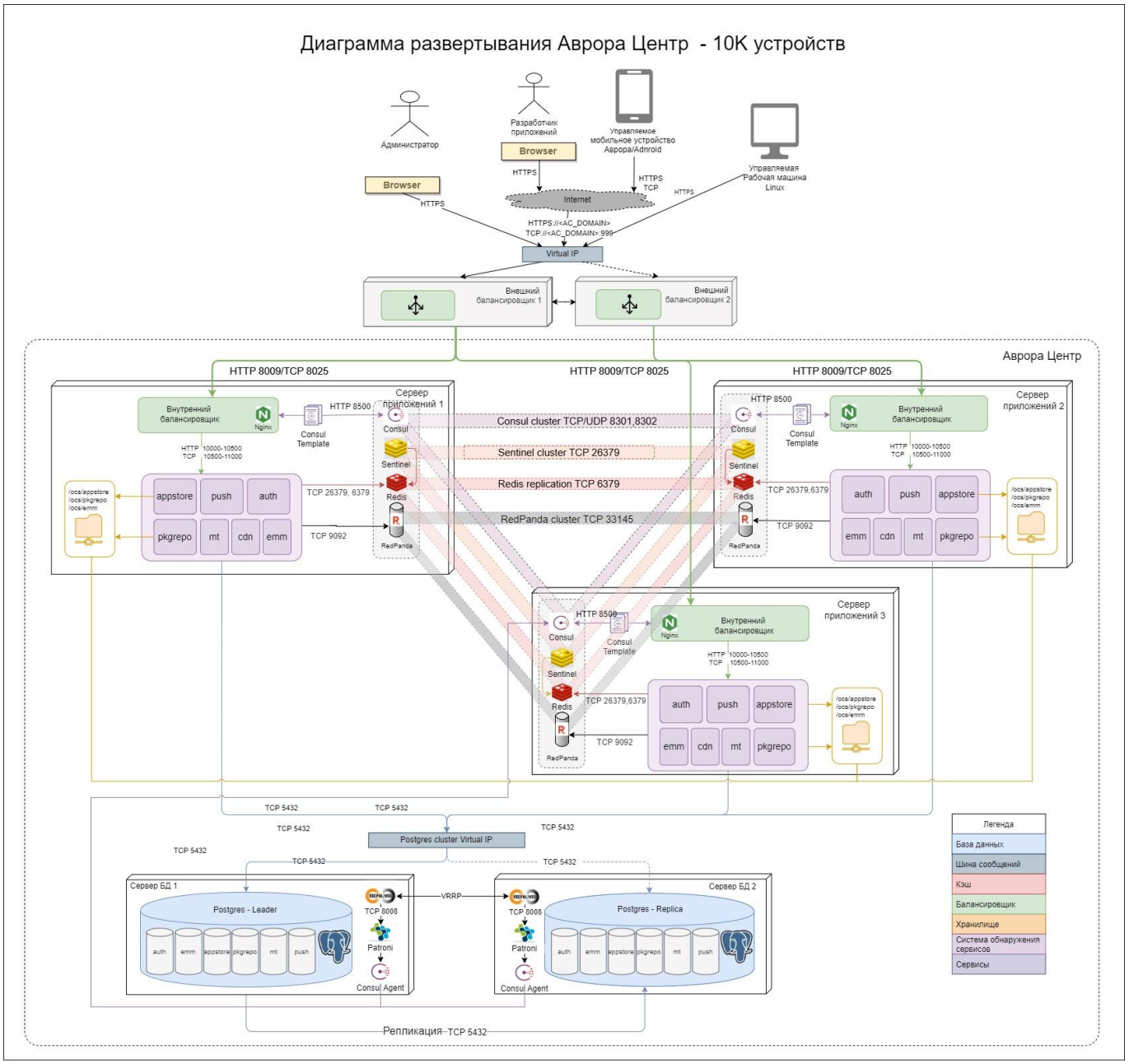


Рисунок 8

2.3.4. Кластерная конфигурация с контент-серверами (поддержка до 100000 устройств)

В отличии от конфигурации, поддерживающей до 10000 устройств, в данной конфигурации для оптимизации доставки контента и снижения нагрузки на сервер приложений используются контент-серверы.

Данную конфигурацию рекомендуется использовать, когда требуется поддержка до 100000 устройств и/или в случае большой территориальной удаленности устройств от сервера приложений (Рисунок 9, Рисунок 10).

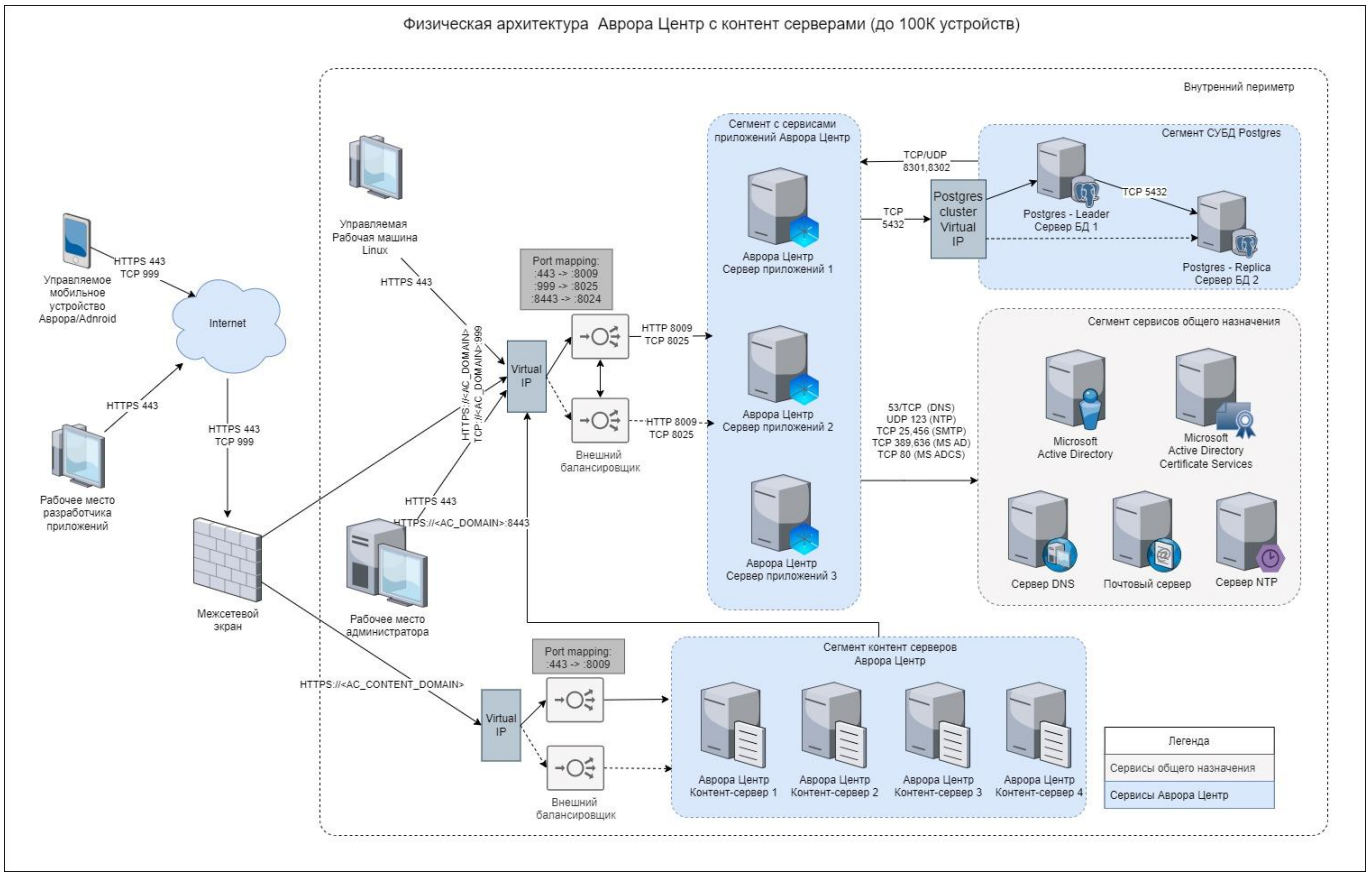


Рисунок 9

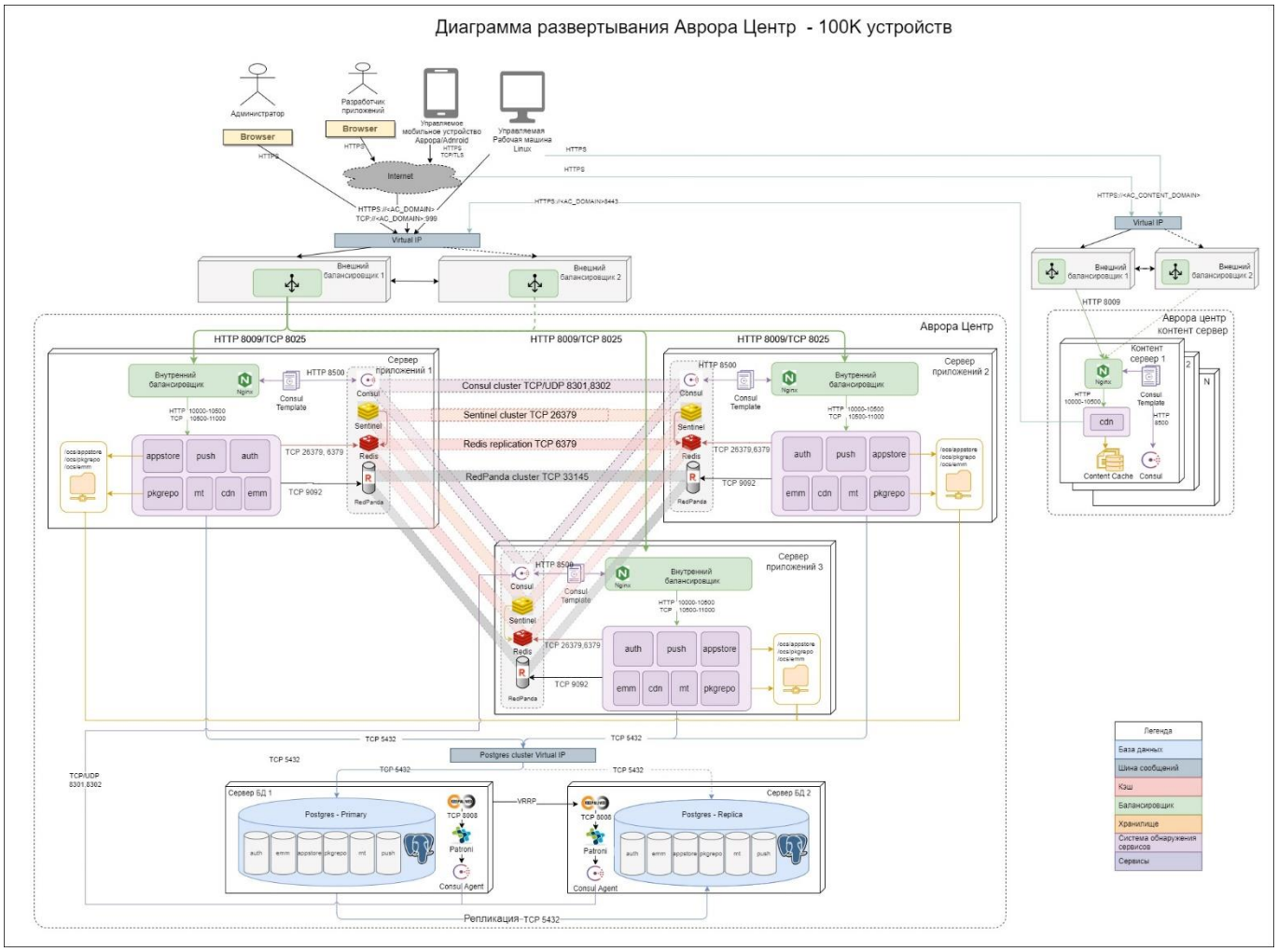


Рисунок 10

2.3.5. Кластерная конфигурация с контент-сервером и раздельными серверами БД (поддержка до 500000 устройств)

В данной конфигурации для минимизации взаимного влияния друг на друга осуществляется разделение наиболее нагруженных БД ПБ (auth) и ПУ (emm) по отдельным серверам. Инфраструктурные компоненты ППО устанавливаются на отдельных серверах, сервисы ППО собираются в кластер из шести нод. Контент-серверы можно разместить на отдельных площадках (например, в разных регионах).

Данную конфигурацию рекомендуется использовать для поддержки максимального количества устройств (до 500000 устройств) (Рисунок 11, Рисунок 12).

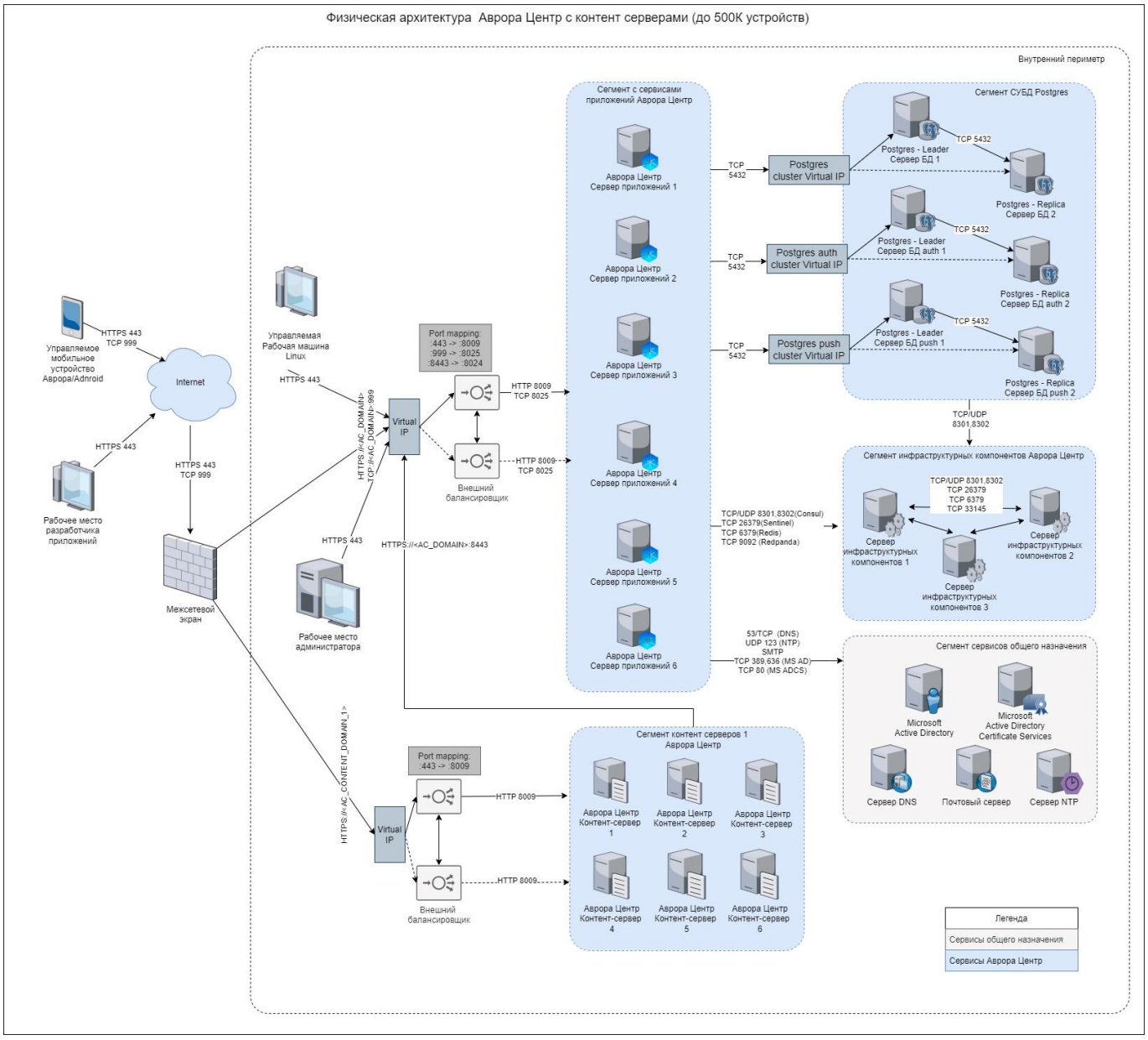

Физическая архитектура Аврора Центр с контент серверами (до 500К устройств)
Внутренний периметр
Управляемое мобильное устройство Аврора/Adnroid
HTTPS 443
TCP 999
Internet
HTTPS 443
HTTPS 443
TCP 999
Рабочее место разработчика приложений
Межсетевой экран
Управляемая Рабочая машина Linux
HTTPS 443
HTTPS 443
Рабочее место администратора
HTTPS://<AC_DOMAIN>:999
TCP://<AC_DOMAIN>:999
Virtual IP
HTTPS://<AC_DOMAIN>:8443
Port mapping:
:443 -> :8009
:999 -> :8025
:8443 -> :8024
HTTP 8009
TCP 8025
HTTP 8009
TCP 8025
Внешний балансировщик
Сегмент с сервисами приложений Аврора Центр
Аврора Центр Сервер приложений 1
Аврора Центр Сервер приложений 2
Аврора Центр Сервер приложений 3
Аврора Центр Сервер приложений 4
Аврора Центр Сервер приложений 5
Аврора Центр Сервер приложений 6
TCP 5432
Postgres cluster Virtual IP
TCP 5432
Postgres auth cluster Virtual IP
TCP 5432
Postgres push cluster Virtual IP
Сегмент СУБД Postgres
Postgres - Leader Сервер БД 1
TCP 5432
Postgres - Replica Сервер БД 2
Postgres - Leader Сервер БД auth 1
TCP 5432
Postgres - Replica Сервер БД auth 2
Postgres - Leader Сервер БД push 1
TCP 5432
Postgres - Replica Сервер БД push 2
TCP/UDP 8301,8302
Сегмент инфраструктурных компонентов Аврора Центр
TCP/UDP 8301,8302
TCP 26379
TCP 6379
TCP 33145
Сервер инфраструктурных компонентов 1
Сервер инфраструктурных компонентов 2
Сервер инфраструктурных компонентов 3
TCP/UDP 8301,8302(Consul)
TCP 26379(Sentinel)
TCP 6379(Redis)
TCP 9092 (Redpanda)
53/TCP (DNS)
UDP 123 (NTP)
SMTP
TCP 389,636 (MS AD)
TCP 80 (MS ADCS)
Сегмент сервисов общего назначения
Microsoft Active Directory
Microsoft Active Directory Certificate Services
Сервер DNS
Почтовый сервер
Сервер NTP
HTTPS://<AC_CONTENT_DOMAIN_1>
Virtual IP
Port mapping:
:443 -> :8009
HTTP 8009
HTTP 8009
Внешний балансировщик
Сегмент контент серверов 1 Аврора Центр
Аврора Центр Контент-сервер 1
Аврора Центр Контент-сервер 2
Аврора Центр Контент-сервер 3
Аврора Центр Контент-сервер 4
Аврора Центр Контент-сервер 5
Аврора Центр Контент-сервер 6
Легенда
Сервисы общего назначения
Сервисы Аврора Центр

Рисунок 11

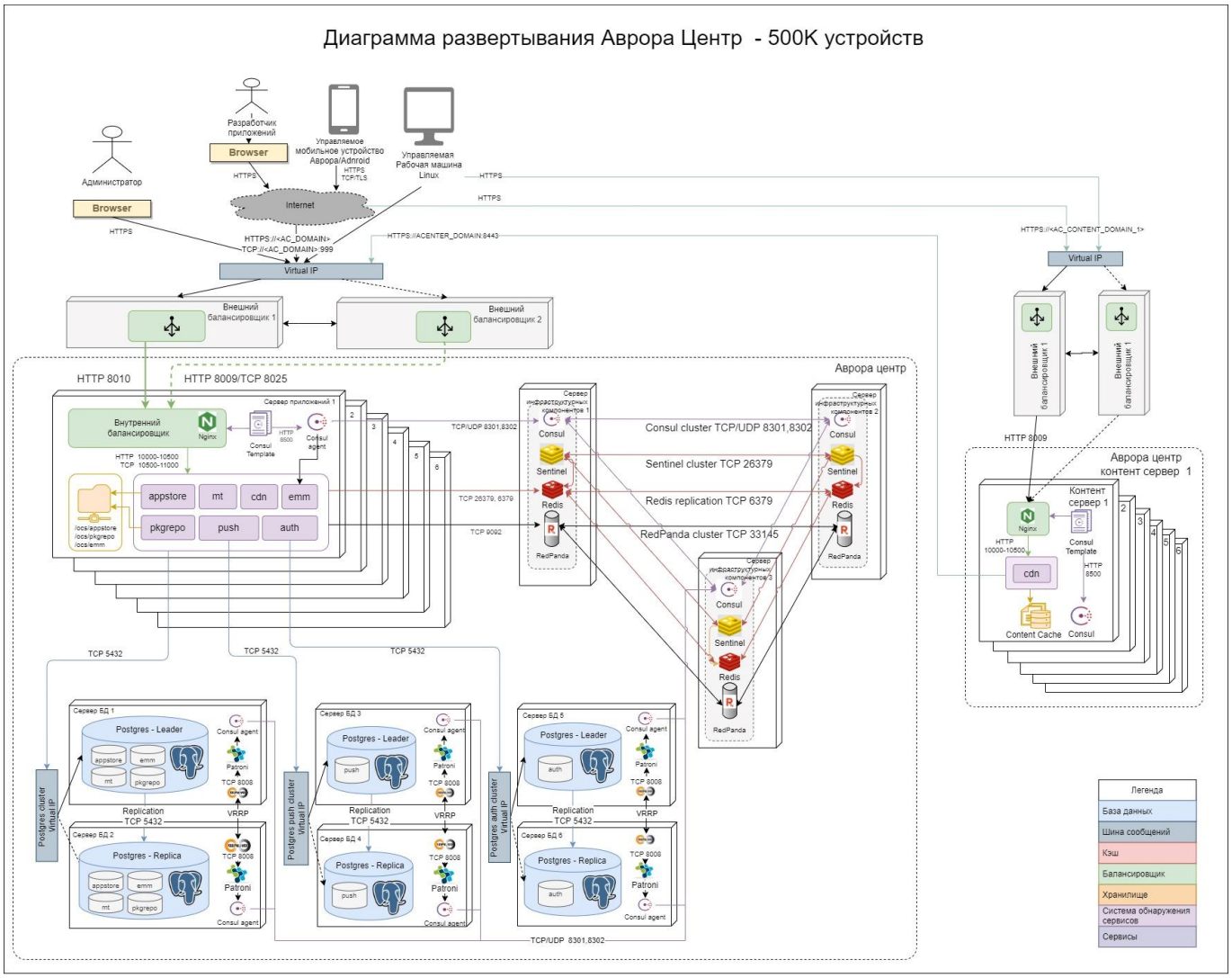


Рисунок 12

2.3.6. Катастрофоустойчивая кластерная конфигурация с установкой серверов приложений и серверов БД в двух центрах обработки данных

В данной конфигурации с целью защиты от природных, техногенных катастроф или терактов и обеспечения непрерывности бизнес-процессов установка серверов приложений и серверов БД осуществляется в двух центрах обработки данных (ЦОДах) - основной и резервный (Рисунок 13).

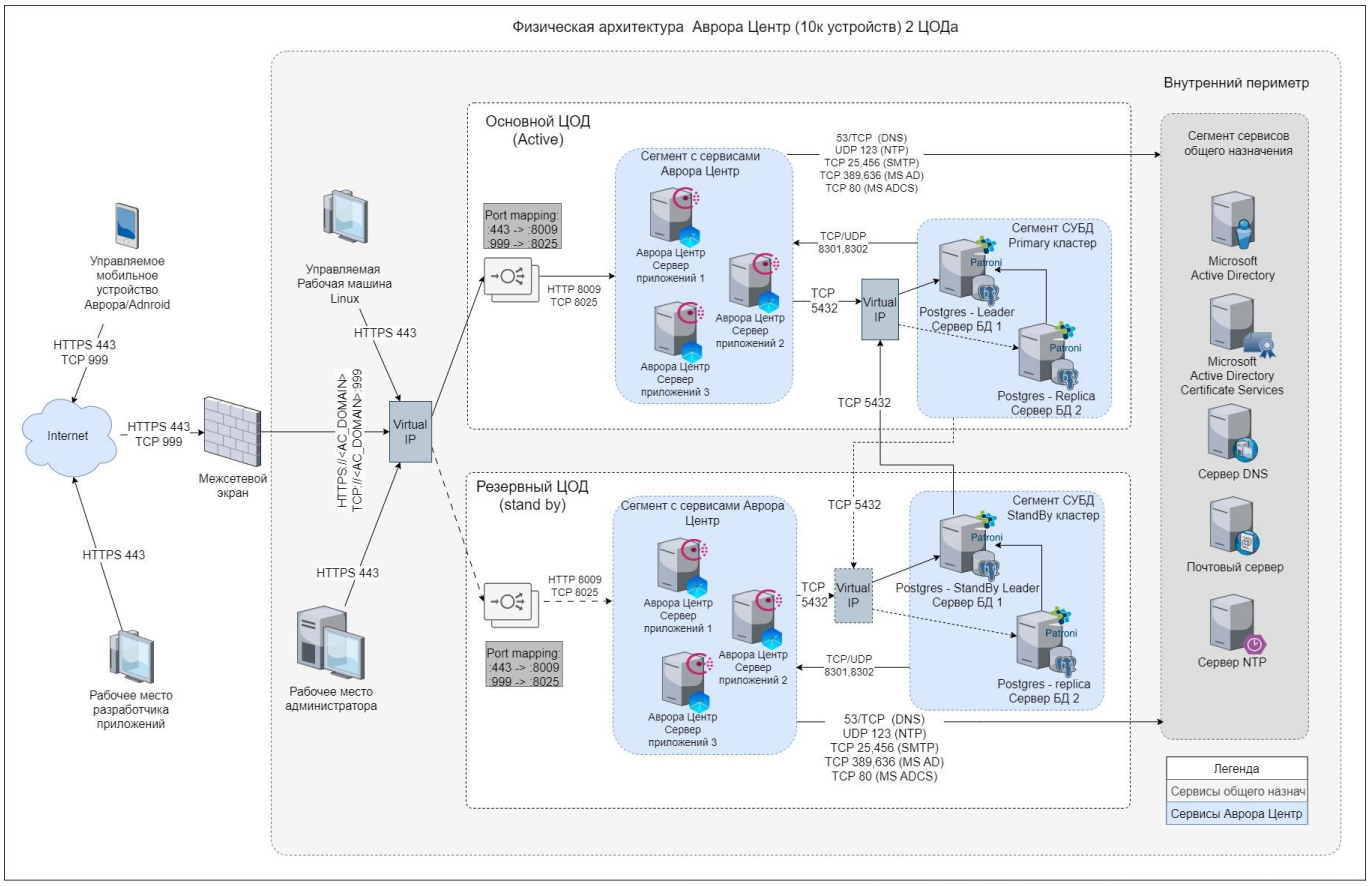


Рисунок 13

В каждом ЦОДе устанавливается полный набор сервисов ППО в отказоустойчивой конфигурации (п. 2.3.3 - 2.3.5).

Основной ЦОД функционирует в активном режиме (Active), т.е. обрабатывает весь поступающий трафик. Серверы БД основного ЦОДа собираются в Primary кластер. Один из серверов БД выступает в роли Leader, второй сервер в роли Replica, реплицируя данные с Leader сервера из текущего ЦОДа.

Резервный ЦОД функционирует в режиме ожидания (StandBy) - все сервисы активны, но трафик в данный ЦОД не подается. Серверы БД резервного ЦОДа собираются в StandBy кластер. Один из серверов БД выступает в роли StandBy Leader, реплицируя данные с сервера из основного ЦОДа. Второй сервер БД выступает в роли Replica, реплицируя данные с сервера из текущего ЦОДа.

Серверы приложений из разных ЦОДах не связаны между собой. Синхронизация состояния между ЦОДами реализуется за счет схемы каскадной репликации данных БД из основного ЦОДа в резервный.

Переключение трафика между ЦОДами (`failover/switchover`) осуществляется в ручном режиме (п. 3.9.20). Потеря данных при переключении БД будет равна задержке (лагу) репликации.

3. УСТАНОВКА ППО

ВНИМАНИЕ! Администратору/разработчику при копировании команд из настоящего документа в формате `.pdf` необходимо проявлять внимательность и дополнительно проверять результаты выполнения соответствующих команд на экране.

3.1. Общая информация

Установка ППО и компонентов среды функционирования ППО осуществляется с помощью сценариев установки ППО, выполняемых на управляющей электронно-вычислительной машине (ЭВМ) и написанных с использованием декларативного языка разметки для описания конфигураций Ansible. Сценарии установки ППО позволяют выполнить установку как локально (все компоненты на 1 ЭВМ), так и с удаленной ЭВМ (управляющей ЭВМ) (Рисунок 14).

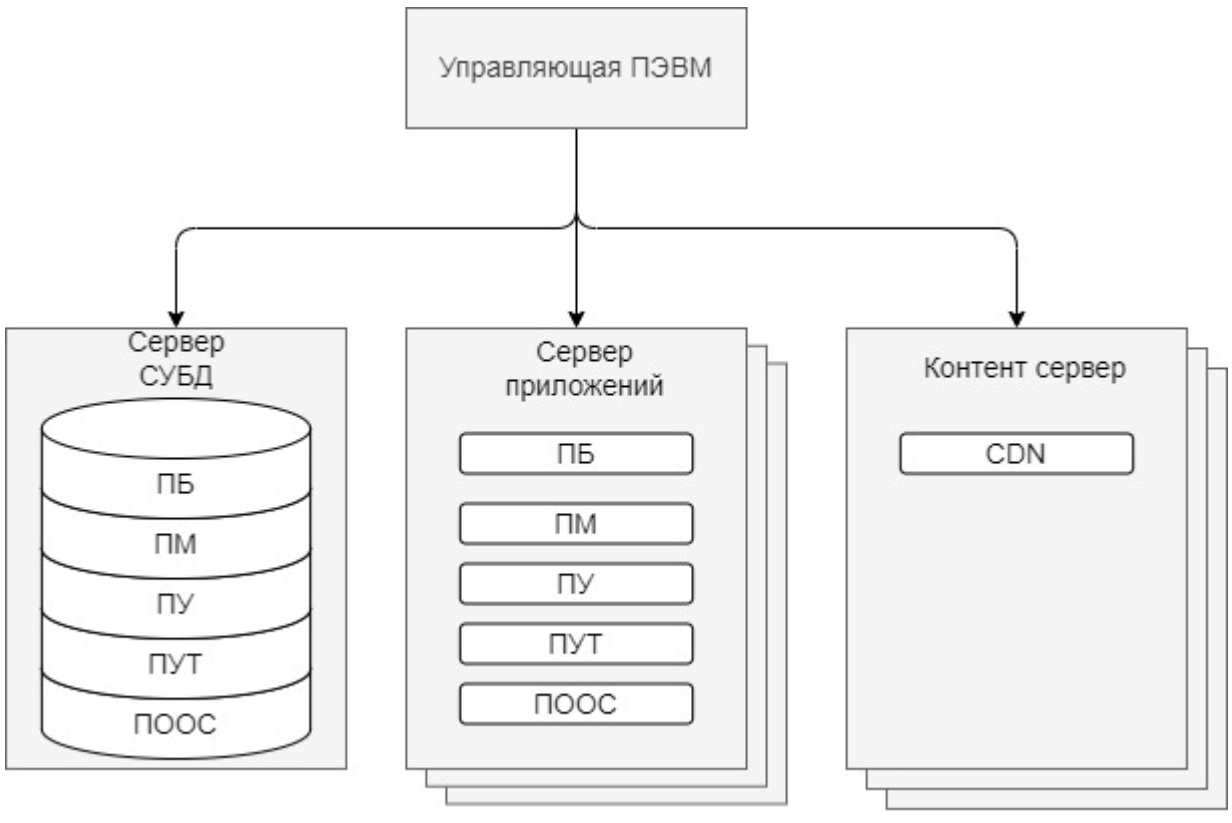


Рисунок 14

ПРИМЕЧАНИЕ. Управляющая ЭВМ необходима только для установки, настройки и управления ППО и не требуется для функционирования ППО.

ВНИМАНИЕ! Для установки ППО необходимо наличие стабильного интернет-соединения на серверах приложений, серверах БД, контент-серверах, а также на управляющей ЭВМ.

Для установки ППО необходимо выполнить следующие действия:

1) Убедиться, что соблюдены требования, приведенные в подразделе 1.4;
2) Установить и настроить ОС на серверы приложений, серверы БД и при необходимости на контент-серверы (подраздел 3.2);
3) Развернуть и настроить управляющую ЭВМ (подраздел 3.3);
4) Настроить компоненты среды функционирования ППО (п. 3.4.1);
5) Настроить ППО (п. 3.4.2);
6) Установить компоненты среды функционирования ППО (п. 3.5.1);
7) Установить ППО (п. 3.5.2);
8) Настроить подсистемы ППО (подраздел 3.7);
9) При необходимости выполнить дополнительные настройки ППО и его среды функционирования (подраздел 3.9);
10) Проверить корректность установки и функционирования ППО (подраздел 3.10).

3.2. Порядок установки и настройки ОС на серверах приложений, серверах БД и контент-серверах

3.2.1. Установить на серверы приложений, серверы БД и при необходимости на контент-серверы одну из следующих ОС, приведенных в п. 1.4.2.

ВНИМАНИЕ! Перед установкой ОС необходимо ознакомиться с требованиями, приведенными в документации на СЗИ НСД.

ОС должна быть установлена в минимальной конфигурации без графического интерфейса. Например, для установки ОС CentOS версии 7 необходимо использовать ISO-образ, в названии которого содержится «Minimal» (`CentOS-7-x86_64-Minimal-1810.iso`).

3.2.2. Обеспечить выполнение следующих требований

3.2.2.1. Требования к предустановленным в ОС пакетам

На серверах приложений, серверах БД и контент-серверах должны быть установлены следующие пакеты:

– `sudo`;

– `python` версии 3.6 или выше.

3.2.2.2. Требования к настройке сети ОС

Необходимо, чтобы настройки сети ОС соответствовали следующим требованиям:

1) Для основного сетевого интерфейса должен присутствовать конфигурационный файл(ы):

– ОС CentOS и ОС РЕД ОС: `/etc/sysconfig/network-scripts/ifcfg-<имя интерфейса>`

– ОС Альт:

```
/etc/net/ifaces/<имя интерфейса>/ipv4address
/etc/net/ifaces/<имя интерфейса>/ipv4route
/etc/net/ifaces/<имя интерфейса>/options
```

– ОС Astra Linux, Debian: `/etc/network/interfaces`

– ОС Ubuntu: `/etc/netplan/*.yml`

2) Сетевой интерфейс должен автоматически запускаться при загрузке ОС.

Для этого необходимо:

– в конфигурационном файле `/etc/sysconfig/network-scripts/ifcfg-<имя интерфейса>` (для ОС CentOS или ОС РЕД ОС) или `/etc/net/ifaces/<имя интерфейса>/options` (для ОС Альт) задать следующее значение параметра `ONBOOT`:

```
ONBOOT=yes
```

– в конфигурационный файл /etc/network/interfaces (для ОС Astra Linux или Debian) внести следующую запись:

```
auto <имя интерфейса>
```

– в ОС Ubuntu автозапуск сетевого интерфейса настроен по умолчанию.

3) Приоритеты в конфигурационном файле /etc/nsswitch.conf должны выглядеть следующим образом (при использовании dnsmasq):

```
hosts: files dns ...
```

где `...` - остальные опции, если они используются;

4) На сетевых интерфейсах серверов приложений, серверов БД и контент-серверов должны быть настроены статические ip-адреса (использование динамических адресов, выдаваемых по DHCP, не допускается).

5) В случае, когда сервера приложений ППО находятся за прокси-сервером, необходимо отключить проксирование запросов к сервисам ППО.

Для этого в переменной no_proxy конфигурационного файла /etc/environment необходимо указать список доменных имен или IP-адресов серверов приложений, для которых не следует использовать проксирование:

```
NO_PROXY=localhost,127.0.0.0/8,.local,<домен сервера приложений>
```

Например:

```
NO_PROXY=localhost,127.0.0.0/8,.local,omp.acenter.example
```

3.2.3. Перейти в учетную запись суперпользователя с помощью команды:

```
sudo su
```

3.2.4. Настроить репозитории (в случае использования ОС Astra Linux SE версии 1.7).

Для этого в конфигурационном файле `/etc/apt/sources.list` необходимо исключить CD-ROM из списка доступных репозиториев, а также настроить доступ к основному (`main`) и базовому (`base`) репозиториям ОС:

```
# deb cdrom:[OS Astra Linux 1.7.0 1.7_x86-64 contrib main non-free
deb http://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-main/
1.7_x86-64 main contrib non-free
deb http://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-base/
1.7_x86-64 main contrib non-free
```

3.2.5. Назначить пользователям ОС права на выполнение команд от имени суперпользователя без ввода пароля с помощью команды:

```
echo '<имя пользователя> ALL=(ALL:ALL) NOPASSWD: ALL' | EDITOR='tee -
a' visudo -f /etc/sudoers.d/<имя пользователя>
```

Например:

```
echo 'omp ALL=(ALL:ALL) NOPASSWD: ALL' | EDITOR='tee -a' visudo -f
/etc/sudoers.d/omp
```

ВНИМАНИЕ! Права на выполнение команд от имени суперпользователя должны быть назначены всем пользователям (на управляющей ЭВМ, серверах приложений, серверах БД и контент-серверах), которыми осуществляется установка компонентов среды функционирования, СУБД и ППО. В противном случае в процессе установки возникнут ошибки.

3.2.6. Установить кодировку UTF-8 с помощью команды:

– ОС CentOS версии 7, ОС РЕД ОС, ОС Astra Linux, ОС Ubuntu и ОС Debian:

```
localectl set-locale LANG=en_US.UTF-8
```

– ОС Альт:

В конфигурационном файле `/etc/sysconfig/i18n` задать следующее значение параметра `LANG`:

```
LANG=en_US.UTF-8
```

3.2.7. Задать имя хоста с помощью команды:

```
hostnamectl set-hostname "имя_хоста.имя_домена"
```

ВНИМАНИЕ! При задании имени хоста обязательно должно быть задано имя домена, которое отделяется точкой. Например:

```
hostnamectl set-hostname ocs-app.local
```

3.2.8. В настройках DNS-сервера или файлах `/etc/hosts` указать имена хостов (`hostname`) и полные имена доменов (`FQDN`) всех серверов кластера:

```
"ip-адрес" "имя_хоста.имя_домена"
```

Например (в файле `/etc/hosts`):

```
192.168.0.108 ocs-app.local
```

3.2.9. Задать адреса DNS-серверов:

Адреса DNS-серверов задаются в файле `/etc/resolv.conf` в следующем формате:

```
nameserver "ip-адрес"
```

Например:

```
nameserver 192.168.0.1
```

ПРИМЕЧАНИЕ. В случае отсутствия файла `/etc/resolv.conf` необходимо его создать.

3.2.10. Настроить маршрут по умолчанию (`default gateway`) через `lan` интерфейс в соответствии с документацией на ОС.

3.2.11. Задать текущие дату и время с помощью команды:

```
date -s 'YYYY-MM-DD HH:MI:SS'
```

Например:

```
date -s '2021-03-31 12:34:56'
```

3.2.12. Перезагрузить управляющую ЭВМ и серверы с помощью команды:

```
reboot
```

3.3. Порядок развертывания и настройки управляющей ЭВМ

ВНИМАНИЕ! Перед развертыванием и настройкой управляющей ЭВМ необходимо произвести установку, настройку ОС на серверах приложений, серверах БД и при необходимости на контент-серверах в соответствии с подразделом 3.2.

Для развертывания и настройки управляющей ЭВМ необходимо выполнить следующие действия:

3.3.1. Установить на управляющую ЭВМ одну из следующих ОС: ОС CentOS версии 7.9.2009, ОС Альт 8 СП, ОС РЕД ОС, ОС Ubuntu 20.04, ОС Ubuntu 22.04 или ОС Astra Linux SE версии 1.7.

ПРИМЕЧАНИЕ. В качестве управляющей ЭВМ может быть испльзоваться как отдельная ЭВМ, так и сервер приложений ППО.

3.3.2. Настроить сетевое взаимодействие управляющей ЭВМ с серверами приложений, серверами БД и контент-серверами.

Настройка сети на управляющей ЭВМ осуществляется в соответствии с ЭД на ОС.

3.3.3. Настроить репозитории (в случае использования ОС Astra Linux SE версии 1.7)

Для этого в конфигурационном файле `/etc/apt/sources.list` необходимо исключить CD-ROM из списка доступных репозиториев, а также настроить доступ к основному (`main`) и базовому (`base`) репозиториям ОС:

```
# deb cdrom:[OS Astra Linux 1.7.0 1.7_x86-64 contrib main non-free
deb http://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-main/
1.7_x86-64 main contrib non-free
deb http://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-base/
1.7_x86-64 main contrib non-free
```

3.3.4. Создать на управляющей ЭВМ отдельный каталог, скопировать в него содержимое DVD с ППО и перейти в созданный каталог с помощью команды:

```
cd <путь к каталогу>
```

3.3.5. Перейти в каталог `/server` с помощью команды:

```
cd <путь к каталогу server>
```

3.3.6. Запустить `installer-ac.sh` (или `installer-ac-mt.sh` или `installer-ac-mt-spr.sh`[10]) с помощью команды:

```
bash installer-ac.sh
или
bash installer-ac-mt.sh
или
bash installer-ac-mt-spr.sh
```

3.3.7. Ознакомиться с «Лицензионным соглашением» и принять его.

Для того, чтобы принять «Лицензионное соглашение» (Рисунок 15), необходимо после вопроса «Вы принимаете условия лицензии (y/n)?» ввести «y», в результате чего в каталоге с файлом `installer-ac.sh` будет создан каталог `install-<версия ППО>`. Например, `/install-release-v5.0.0`.

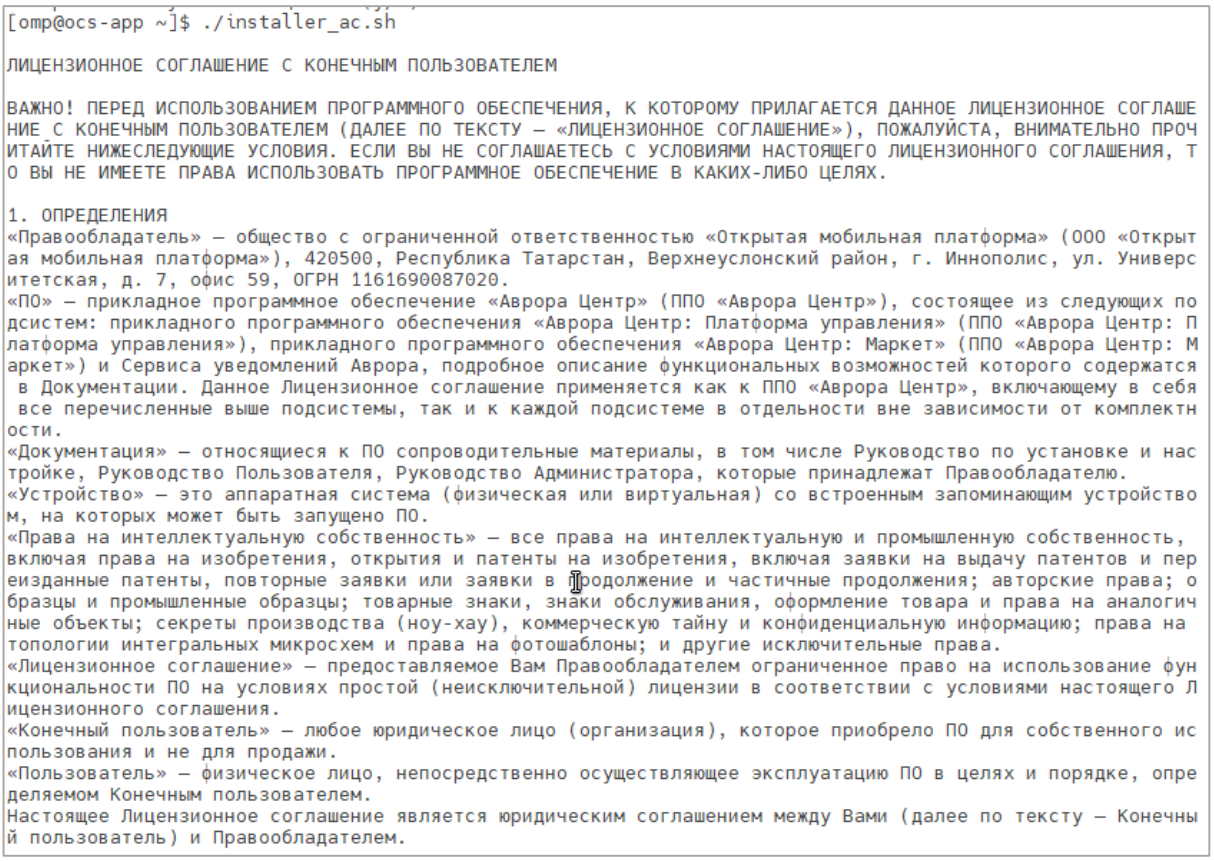


Рисунок 15

[10] Название файла зависит от варианта поставки ППО.

Также возможно автоматическое принятие лицензионного соглашения (без блокировки процесса установки). В данном случае необходимо ознакомиться с лицензионным соглашением (файл «Лицензионное соглашение.pdf» входит в комплект ЭД) и принять его. Для автоматического принятия лицензионного соглашения необходимо использовать флаг `--accept-license`:

```
bash installer-ac.sh --accept-license
или
bash installer-ac-mt.sh --accept-license
или
bash installer-ac-mt-spr.sh --accept-license
```

ВНИМАНИЕ! В случае несогласия с лицензионным соглашением использование ППО запрещается.

3.3.8. Перейти в каталог со сценариями установки с помощью команды:

```
cd install-<версия ППО>/install-ac/
или
cd install-<версия ППО>/install-ac-mt/
или
cd install-<версия ППО>/install-ac-mt-spr/
```

Например:

```
cd install-release-v5.0.0/install-ac/
```

ПРИМЕЧАНИЕ. Дальнейшие действия по установке и настройке компонентов среды функционирования ППО, а также ППО, необходимо выполнять из данного каталога.

3.3.9. Установить на управляющей ЭВМ пакеты, необходимые для запуска сценариев установки, с помощью команды:

```
sudo bash control-node-prerequisites.sh
```

ПРИМЕЧАНИЕ. Если требуется использовать собственный репозиторий PyPI-пакетов (`PyPI registry`), то предварительно необходимо создать конфигурационный файл `/etc/pip.conf` содержащий следующие настройки:

```
[global]
index-url = <адрес репозитория PyPI>
trusted-host = <домен хоста с репозиторием>
```

Например:

```
[global]
index-url = https://example.com/repository/pypi-pypi.org/simple
trusted-host = example.com
```

3.3.10. Настроить SSH доступ управляющей ЭВМ к серверам приложений, серверам БД и контент-серверам (даже в случае, когда управляющая ЭВМ и серверы установлены на 1 ЭВМ):

– сформировать ключевую пару на управляющем сервере:

```
ssh-keygen -t rsa -b 4096
```

– скопировать открытый ключ на серверы приложений, серверы БД и контент-серверы:

```
ssh-copy-id <имя пользователя>@<сервер приложений>
ssh-copy-id <имя пользователя>@<сервер БД>
ssh-copy-id <имя пользователя>@<контент сервер>
```

– проверить доступ с управляющей машины на серверы приложений, серверы БД и контент-серверы по SSH ключу (при выполнении команд ниже ввод пароля не должен требоваться):

```
ssh <имя пользователя>@<сервер приложения>
ssh <имя пользователя>@<сервер БД>
ssh <имя пользователя>@<контент сервер>
```

ПРИМЕЧАНИЕ. Управляющие команды, формируемые сценариями установки ППО, передаются с использованием протокола SSH.

3.4. Порядок настройки компонентов среды функционирования ППО и ППО

ПРИМЕЧАНИЕ. Сценарии установки позволяют выполнить настройку и установку ППО, а также компонентов среды функционирования ППО для нескольких различных окружений. Порядок конфигурирования и установки ППО для нескольких окружений приведен в п. 3.9.14.

3.4.1. Настройка компонентов среды функционирования

ПРИМЕЧАНИЕ. При задании паролей, секретов, токенов компонентов среды функционирования ППО допустимо использовать строчные и заглавные латинские буквы (кириллица не допускается), цифры, а также следующие специальные символы:

```
~$&*()-=_;.
```

Для настройки компонентов среды функционирования необходимо выполнить следующие действия:

3.4.1.1. Перейти в каталог со сценариями установки (каталог: `/install-<версия ППО>/install-ac/` или `/install-<версия ППО>/install-ac-mt/` или `/install-<версия ППО>/install-ac-mt-spr/`).

3.4.1.2. В инвентарном файле `inventories/hosts.yml` задать адреса серверов (имена хостов), на которые будут установлены компоненты среды функционирования ППО.

Описание порядка задания адресов в инвентарном файле `inventories/hosts.yml` приведено в п. 3.9.12.

Для отображения адреса ЭВМ необходимо выполнить команду:

```
hostname
```

Примеры файлов `hosts.yml` для однонодовой и кластерной конфигурации приведены в каталоге `samples/ac/inventories/`.

Описание параметров инвентарного файла `inventories/hosts.yml` приведено в п. 11.1.1.

3.4.1.3. В конфигурационном файле `config/vars/_vars.yml` необходимо задать либо поменять предустановленные значения следующих параметров:

– параметры подключения подсистем ППО к БД:

```
postgresql:
  port: 5432
```

При использовании балансировщика БД необходимо задать адрес хоста балансировщика, например:

```
postgresql:
  host: "10.189.221.57"
```

– пароль суперпользователя "`postgres`" СУБД Postgresql, если установка СУБД осуществляется с помощью сценариев установки:

```
pg_superuser_password: "postgres"
```

– версию СУБД:

```
pg_version: 15
```

Перечень допустимых значений параметра приведен в таблице (Таблица 13).

Таблица 13

Значение параметра	Версия СУБД
12	PostgreSQL 12
13	PostgreSQL 13
14	PostgreSQL 14
15	PostgreSQL 15
12-pro	Postgres Pro Standard 12
13-pro	Postgres Pro Standard 13
14-pro	Postgres Pro Standard 14
14-stdcert	Postgres Pro Certified 14
15-pro	Postgres Pro Standard 15
15-stdcert	Postgres Pro Certified 15

– имя и пароль пользователя СУБД Postgresql с ролью «replication»:

```
pg_replication_user:
  name: replication
  password: 123FD5648ert**h
```

– имя и пароль суперпользователя СУБД Postgresql, от имени которого будет осуществляться установка ППО:

```
pg_custom_superuser:
  username: ocs_superuser
  password: ClacVob*Twes0Ls6
```

– адреса DNS-серверов, например:

```
dnsmasq_upstream_servers: "192.168.137.1,10.189.211.10"
```

ПРИМЕЧАНИЕ. Описание параметров конфигурационных файлов сценариев установки среды функционирования ППО, сценариев установки ППО и ППО приведено в самих конфигурационных файлах в виде комментариев.

3.4.1.4. В конфигурационном файле `config/secret.yml` задать пароли, секреты и токены:

– пароль доступа к БД:

```
  database:
    password: ocs
```

– пароль доступа к СУБД Redis в параметре `redis_password`:

```
  redis:
    password: "@rTT9089087fslk"
```

– секретный ключ для аутентификации запросов к сервисам ППО:

```
  hmac:
    key: "DEFAULT-F1IWp0t5dY5lYJrm7H-DEFAULT"
```

– токен доступа к системе обнаружения сервисов `Consul`:

```
  consul:
    token: "ae9f5abb-6b8f-9252-59c5-53bcb651f182"
```

– секретный ключ клиентов (сервисов):

```
  defaultOidcClientSecret: "HWfwehfoIOHwfe233WEfvwewe"
```

– ключ шифрования секретов, хранящихся в БД:

```
encrypt:
  keys:
    - "master-key-example"
```

ВНИМАНИЕ! При обновлении ППО удалять старые ключи запрещается. Новые ключи необходимо добавлять в начало списка.

– пароли, используемые для защиты критичной информации (например, cookie сессии):

```
oidcpSecrets:
  system:
    - kdj%93cxk+57nMa4
  cookie:
    - 9v_wer8*&r=_hY8u
```

ВНИМАНИЕ! Длина пароля должна быть не менее 16 символов. При обновлении ППО удалять старые пароли запрещается. Новые пароли необходимо добавлять в начало списка.

– пароль доступа к сервису гарантированной доставки сообщений `Redpanda`:

```
redpanda:
  password: '%GJJ690t5-0'
```

– пароль суперпользователя сервиса гарантированной доставки сообщений `Redpanda`:

```
redpanda_superuser:
  name: admin
  password: Tes%3@@poi
```

– пароль доступа к сервису управления кластером БД `Patroni`:

```
patroni:
  api_password: "VV4445@@@3kjj"
```

– пароль доступа к сервису балансировки нагрузки `Keepalived`:

```
keepalived_auth_pass: "ravJulis*Im5"
```

3.4.2. Настройка ППО (подсистем ППО)

ВНИМАНИЕ! Перед выполнением настроек необходимо изучить порядок работы с конфигурационными файлами, приведенный в п. 12.2.9.

ПРИМЕЧАНИЕ. При задании паролей, секретов, токенов ППО и компонентов среды функционирования ППО допустимо использовать строчные и заглавные латинские буквы (кириллица не допускается), цифры, а также следующие специальные символы:

```
~$&*()-=_;.
```

Для настройки ППО необходимо выполнить следующие действия:

3.4.2.1. В инвентарном файле `inventories/hosts.yml` задать адреса серверов (имена хостов), на которые будут установлены подсистемы ППО.

Описание порядка задания адресов в инвентарном файле `inventories/hosts.yml` приведено в п. 3.9.12.

3.4.2.2. Выполнить настройку порта для административных (привилегированных) интерфейсов ППО (при необходимости).

По умолчанию привилегированные и непривилегированные интерфейсы принимают запросы на порту 8009.

В ППО предусмотрена возможность назначить административным (привилегированным) интерфейсам ППО отдельный порт. Это позволяет ограничить доступ непривилегированных пользователей к административным (привилегированным) интерфейсам ППО.

Для назначения отдельного порта для административного интерфейса необходимо в конфигурационном файле `config/vars/_vars.yml` указать порты для непривилегированных интерфейсов (параметр: `nginx_vhost_external_port`) и административных (привилегированных) интерфейсов (параметр: `nginx_vhost_external_admin_port`), например:

```
nginx_vhost_external_port: 8009
nginx_vhost_external_admin_port: 8010
```

Изменение порта необходимо учитывать в настройках URI в пп. 3.4.2.3.

3.4.2.3. Выполнить настройку URL-адресов ППО

URL-адреса ППО задаются в секции `publicUris` в конфигурационном файле `config/config.yml.j2`.

По умолчанию URL-адреса ППО настроены следующим образом:

– протокол: `HTTP`;

– `hostname`: соответствует первой записи в группе `app` в инвентарном файле `inventories/hosts.yml`;

– порт: соответствует переменной `nginx_vhost_external_port`, заданной в конфигурационном файле `config/vars/_vars.yml`.

Данные настройки соответствует однонодовой конфигурации системы с незащищенным соединением, общим портом для привилегированных и непривилегированных интерфейсов и без использования внешнего балансировщика.

Возможны следующие варианты настройки:

3.4.2.3.1 URL-адреса ППО используют 1 домен, без внешнего балансировщика.

Данные настройки применимы только к однонодовой конфигурации. В параметре `publicUris.ac.commonAddress` – по умолчанию указан `hostname`, соответствующий первой записи в группе `app` в инвентарном файле `inventories/hosts.yml`

```
    ac:
      commonAddress:
"http://{{groups['app']|first}}:${nginxVhostExternalPort}"
```

Если в пп. 3.4.2.2 был назначен отдельный порт для административных (привилегированных) интерфейсов, то необходимо в переменной `publicUris.ac.adminAddress` указать этот порт. Для указания порта можно использовать переменную `nginxVhostExternalAdminPort` или непосредственно задать значение:

```
    ac:
      adminAddress:
"http://{{groups['app']|first}}:${nginxVhostExternalAdminPort}"
```

3.4.2.3.2 Настройка URL-адресов ППО при использовании защищенного соединения.

Незащищенное (протокол HTTP) соединение с сервером приложений ППО допустимо использовать в пилотных проектах, где отсутствует обработка конфиденциальной информации. В остальных случаях должна обеспечиваться защита каналов связи.

ППО поддерживает возможность использования защищенного соединения только с использованием внешнего криптошлюза или настройки TLS на внешнем балансировщике.

При использования защищенного соединения в URL-адресах ППО необходимо указывать протокол HTTPS:

```
    ac:
      commonAddress: "https://acenter.example.ru"
```

3.4.2.3.3 URL-адреса ППО используют 1 домен на внешнем балансировщике.

В данном случае в параметре `commonAddress` необходимо задать протокол (http или https), имя домена <AC_DOMAIN> и порт, настроенные на внешнем балансировщике, например:

```
    ac:
      commonAddress: "https://acenter.example.ru"
```

3.4.2.3.4 URL-адреса ППО разделены на внешний и внутренний домены на внешнем балансировщике.

Подобное разделение, требуется, например, когда необходимо ограничить доступ к Консолям администраторов из внешней сети.

В данном случае в параметре `commonAddress` необходимо задать протокол (http или https), имя домена <AC_DOMAIN> и порт для внешних адресов, а в параметре `adminAddress` задать протокол, имя домена и порт для внутренних адресов, например:

```
ac:
  commonAddress: "https://acenter.example.ru"
  adminAddress: "https://admin.example
```

3.4.2.3.5 URL-адреса ППО используют разные домены.

В данном случае для каждой подсистемы ППО задается свой собственный домен, например:

```
auth:
  adminCrossTenantAddress: "https://authadmin.example"
  publicAddress: "https://authpublic.acenter.example.ru"
aps:
  adminAddress: "https://apsadmin.acenter.example.ru"
  devAddress: "https://apsdev.acenter.example.ru"
  marketAddress: "https://apsmarket.acenter.example.ru"
push:
  adminAddress: "https://pushadmin.example"
  publicAddress: "https://pushpublic.acenter.example.ru"
mt:
  adminAddress: "https://mt.example"
pkgrepo:
  adminAddress: "https://pkgrepoadmin.example"
  mobileAddress: "https://pkgrepomobile.acenter.example.ru"
  repoAddress: "https://pkgrepo.acenter.example.ru"
```

3.4.2.4. Отредактировать конфигурационный файл `config/config.yml.j2`.

В данном конфигурационном файле необходимо задать либо изменить предустановленные значения:

– домен учетных записей пользователей для тенанта `"default"`:

```
defaultIdentityDomain: "omprussia.ru"
```

– уровень детализации сообщений логирования (рекомендуется задать `"info"` при тестовой эксплуатации и `"warn"` при промышленной эксплуатации):

```
logger:
  level: "info"
```

3.4.2.5. Настроить файловое хранилище ППО

Описание настройки файлового хранилища ППО (файловых хранилищ ПМ, ПУ и ПООС) приведено в подразделе 3.8.

3.4.2.6. Выполнить настройки безопасности ППО, другие дополнительные настройки ППО и настройки подсистем ППО (при необходимости).

ВНИМАНИЕ! Перед установкой ППО требуется выполнить настройки безопасности ППО, дополнительные настройки ППО и настройки подсистем ППО (при необходимости).

Перечень и описание дополнительных настроек ППО приведен в подразделе 3.9.

3.5. Порядок установки компонентов среды функционирования ППО и ППО

3.5.1. Установка компонентов среды функционирования ППО

3.5.1.1. Обеспечить синхронизацию времени между нодами кластера.

При эксплуатации ППО в кластерной конфигурации необходимо обеспечить синхронизацию времени между нодами кластера (например, с помощью утилиты `chrony`).

Для проверки синхронизации времени необходимо выполнить команду:

```
ansible-playbook play-check-time-on-hosts.yml --inventory-file
inventories/hosts.yml -vv --diff
```

По результатам выполнения команды будет выведено текущее время на каждой ноде кластера, например:

```
acenterapp03.ompcloud 2022-11-09 09:48:38.394115
acenterapp04.ompcloud 2022-11-09 09:48:38.394533
acenterapp05.ompcloud 2022-11-09 09:48:38.394939
acenterapp06.ompcloud 2022-11-09 09:48:38.395490
acenterapp01.ompcloud 2022-11-09 09:48:38.393034
acenterapp02.ompcloud 2022-11-09 09:48:38.393631
```

3.5.1.2. Установить на серверы приложений, серверы БД и контент-серверы необходимые пакеты.

ВНИМАНИЕ! После завершения установки пакетов службы SELinux и Firewalld будут отключены.

Для установки пакетов необходимо выполнить следующие действия:

3.5.1.2.1 Установить пакеты с помощью следующих команд:

– серверы приложений:

```
ansible-playbook -i inventories/hosts.yml play-managed-node-
prerequisites.yml -vv -u <имя пользователя> --extra-vars
"node_type=app" --limit app
```

– серверы БД:

```
ansible-playbook -i inventories/hosts.yml play-managed-node-
prerequisites.yml -vv -u <имя пользователя> --extra-vars
"node_type=db" --limit postgresql
```

– контент-серверы:

```
ansible-playbook -i inventories/hosts.yml play-managed-node-
prerequisites.yml -vv -u <имя пользователя> --extra-vars
"node_type=content" --limit content
```

Например, установка пакетов на серверы приложений осуществляется с помощью команды:

```
ansible-playbook -i inventories/hosts.yml play-managed-node-
prerequisites.yml -vv -u omp --extra-vars "node_type=app" --limit app
```

Для установки всех пакетов на все серверы (на все серверы приложений, серверы БД и контент-серверы независимо от их типа) необходимо выполнить команду:

```
ansible-playbook -i inventories/hosts.yml play-managed-node-
prerequisites.yml -vv -u <имя пользователя>
```

3.5.1.2.2 На серверах приложений, серверах БД и контент-серверах под управлением ОС РЕД ОС включить автозапуск службы `network` с помощью команды:

```
sudo systemctl enable network
```

3.5.1.2.3 Перезагрузить серверы приложений, серверы БД и контент-серверы с помощью команды:

```
sudo reboot
```

Порядок действий для самостоятельной установки пакетов, а также отключению служб SELinux и Firewalld приведен в п. 3.9.7 и 3.9.8.

3.5.1.3. Установить компоненты среды функционирования ППО с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh
```

Например:

```
ANSIBLE_USER="omp" ./deploy-infra.sh
```

В результате выполнения команды в каталоге `logs` будет сформирован лог-файл установки компонентов среды функционирования ППО.

Описание параметров запуска скрипта `deploy-infra.sh` и их возможные значения приведены в подразделе 4.1.

ВНИМАНИЕ! Скрипт `deploy-infra.sh` позволяет устанавливать только СУБД PostgreSQL 12/13/14/15. СУБД Postgres Pro необходимо устанавливать самостоятельно.

При использовании СУБД Postgres Pro либо если установку СУБД PostgreSQL 12/13/14/15 необходимо выполнить самостоятельно (без использования сценариев установки компонентов среды функционирования ППО), команда установки компонентов среды функционирования имеет следующий вид:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh --skip-database
```

Описание установки и настройки СУБД Postgres Pro, а также требования к самостоятельной установке СУБД приведены в подразделе 3.11.

Также предусмотрена возможность установки компонентов среды функционирования по отдельности с помощью следующих команд:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c dnsmasq
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c nginx
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c consul
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c consul-template
```

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c redpanda
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c redis
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c ocs-user
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c db
```

3.5.2. Установка ППО

Для установки ППО необходимо выполнить команду:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh
```

Описание параметров запуска скрипта `deploy-ac.sh` и их возможные значения приведены в подразделе 4.2.

Например:

```
ANSIBLE_USER="omp" ./deploy-ac.sh
```

В результате выполнения команды в каталоге `logs` будет сформирован лог-файл установки ППО.

Для установки подсистем по отдельности необходимо в параметре `--subsystems` задать имя подсистемы.

ВНИМАНИЕ! Установка подсистем ППО должна осуществляться строго в следующей последовательности: ПБ, ПМ, ПООС, ПУ, ПУТ, CDN, ПСУ.

Пример установки подсистем по отдельности:

```
ANSIBLE_USER="omp" ./deploy-ac.sh --subsystems auth
ANSIBLE_USER="omp" ./deploy-ac.sh --subsystems appstore
ANSIBLE_USER="omp" ./deploy-ac.sh --subsystems pkgrepo
ANSIBLE_USER="omp" ./deploy-ac.sh --subsystems emm
ANSIBLE_USER="omp" ./deploy-ac.sh --subsystems mt
ANSIBLE_USER="omp" ./deploy-ac.sh --subsystems cdn
ANSIBLE_USER="omp" ./deploy-ac.sh --subsystems push
```

Если необходимо установить ПСУ отдельно от других подсистем ППО, тогда достаточно установить ПБ и ПСУ с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --subsystems
auth,push
```

Для установки ППО без ПСУ необходимо выполнить команду:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --subsystems
auth,appstore,pkgrepo,emm,mt
```

3.5.3. Выполнение настройки подсистем ППО

ВНИМАНИЕ! При невыполнении данных настроек часть функции ППО может не работать или работать некорректно.

Настройка подсистем ППО осуществляется в соответствии с подразделом 3.7.

3.5.4. Выполнение ограничений по применению

При эксплуатации ППО необходимо соблюдать следующие ОГРАНИЧЕНИЯ ПО ПРИМЕНЕНИЮ:

– ПСУ не осуществляет аутентификацию подключаемых к нему устройств под управлением ОС Аврора версии 5.0.0 и ниже, поэтому при необходимости обеспечения конфиденциальности, целостности и доступности push-уведомлений необходимо использовать компенсирующие меры защиты информации, например, криптографическую защиту канала связи с двусторонней аутентификацией между Сервером приложений ПСУ и устройствами. Для устройств под управлением ОС Аврора версии 5.1 и выше в ПСУ реализована аутентификация, поэтому использование компенсирующих мер не требуется;

– после установки и настройки ППО необходимо выполнить ограничения по применению, произвести настройки безопасности компонентов среды функционирования и настроить СЗИ. Необходимая информация приведена в п. 3.9.4.

3.5.5. Проверка корректности установки и функционирования ППО

Проверка осуществляется в соответствии с подразделом 3.10.

3.6. Адреса веб-консолей

Первоначальный вход в ППО осуществляется с помощью Консоли администратора ПБ и предустановленной учетной записи с ролью Администратор учетных записей:

– логин: admin@omprussia.ru;

– пароль: Admin123!

ПРИМЕЧАНИЕ. При первом входе в ППО необходимо сменить пароль.

В таблице (Таблица 14) приведены адреса веб-консолей.

Таблица 14

Веб-консоль	URL-адрес веб-консоли
Консоль администратора ПБ	http(s)://<сервер приложения>:8009/auth/admin/
Консоль администратора ПМ	http(s)://<сервер приложения>:8009/appstore/admin/
Консоль разработчика ПМ	http(s)://<сервер приложения>:8009/appstore/dev/
Консоль администратора ПУ	http(s)://<сервер приложения>:8009/emm/admin/
Консоль администратора ПУТ	http(s)://<сервер приложения>:8009/mt/admin/
Консоль администратора ПСУ	http(s)://<сервер приложения>:8009/push/admin/

3.7. Описание настройки подсистем ППО

3.7.1. Описание настройки ПСУ

Настройка ПСУ заключается в настройке обратного прокси-сервера (`reverse proxy`), а также в настройке протокола взаимодействия устройств (push-демона) с ПСУ.

3.7.1.1. Настройка обратного прокси-сервера

Обратный прокси-сервер (`reverse proxy`) служит для обработки запросов (подключений) устройств (push-демона) по защищенному протоколу TLS и транслирование этих запросов в ПСУ.

Примеры конфигурационных файлов обратного прокси-сервера Nginx для однонодовой и многонодовой конфигураций приведены в каталоге `samples/ac/nginx_external-balancer/conf_stream.d/`.

Для настройки обратного прокси-сервера Nginx необходимо выполнить следующие действия:

3.7.1.1.1 Скопировать файл samples/ac/nginx_external-balancer/conf_stream.d/one-node.conf (или samples/ac/nginx_external-balancer/conf_stream.d/three-node.conf для многонодовой конфигурации) в каталог /etc/nginx/conf_stream.d/ и переименовать его в ocs-push-stream.conf с помощью команды:

– для однонодовой конфигурации:

```
sudo cp samples/ac/nginx_external-balancer/conf_stream.d/one-node.conf
/etc/nginx/conf_stream.d/ocs-push-stream.conf
```

– для многонодовой конфигурации:

```
sudo cp samples/ac/nginx_external-balancer/conf_stream.d/three-
node.conf /etc/nginx/conf_stream.d/ocs-push-stream.conf
```

3.7.1.1.2 В секции `upstream` конфигурационного файла `ocs-push-stream.conf` задать адреса нод Серверов приложений ПСУ, например:

```
upstream internal-lb-stream-8025 {
  server ocs-app.local:8025 max_fails=3 fail_timeout=60 weight=1;
  least_conn;
}
```

3.7.1.1.3 Создать каталог для хранения лог-файлов Nginx:

```
mkdir -p <путь к каталогу>
```

Например:

```
mkdir -p /var/log/nginx/external_balancer/
```

3.7.1.1.4 В секции `server` задать значения следующих параметров:

– порт, к которому будут подключаться устройства, например:

```
listen 999 ssl so_keepalive=on;
```

– путь к закрытому ключу и сертификату закрытого ключа, которые будут использоваться для установки TLS-соединения, например:

```
ssl_certificate /etc/nginx/ssl/cert.pem;
ssl_certificate_key /etc/nginx/ssl/privkey.pem;
```

ПРИМЕЧАНИЕ. Сертификат закрытого ключа должен входить в цепочку доверия сертификатов на устройствах;

– путь к лог-файлам Nginx в соответствии с пп.3.7.1.1.3, например:

```
access_log /var/log/nginx/external_balancer/access-999.log basic;
error_log /var/log/nginx/external_balancer/error-999.log;
```

3.7.1.1.5 Проверить корректность конфигурационных файлов Nginx с помощью команды:

```
sudo nginx -t
```

В случае отсутствия ошибок будет выведено сообщение:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

3.7.1.1.6 Перезапустить Nginx с помощью команды:

```
sudo systemctl reload nginx
```

3.7.1.2. Настройка протокола взаимодействия устройств (push-демона) с ПСУ

Устройства в зависимости от установленной на них версии ОС Аврора используют разные версии протокола взаимодействия с ПСУ. Перечень версий протокола взаимодействия устройств с ПСУ приведен в таблице (Таблица 15).

Таблица 15

Версия протокола	Описание
0	Версия протокола для ОС Аврора 3-го поколения. Не поддерживает аутентификацию устройств в ПСУ
2	Версия протокола для ОС Аврора 4-го поколения, в которой учтены особенности многопользовательского режима ОС. Не поддерживает аутентификацию устройств в ПСУ

Версия протокола	Описание
3	В данной версии протокола реализована аутентификация устройств при подключении к ПСУ. Поддерживается ОС Аврора версии 5.1 и выше. **ПРИМЕЧАНИЕ.** Использование протокола версии 3 допустимо при условии установки ПСУ совместно с ПУ и активации устройств в ПУ. В ином случае необходимо использовать протокол версии 2

Задание протокола осуществляется в параметре `allowedDeviceProtocolVersions` конфигурационного файла `install-<версия ППО>/install-ac/config/subsystems/push/config.yml`. В данном параметре задается список доступных версий протокола. Если задано несколько версий, то будет использоваться максимальная версия протокола, которая поддерживается и устройством, и сервером. Примеры задания значений параметра `allowedDeviceProtocolVersions` приведены в таблице (Таблица 16).

Таблица 16

Значение параметра	Описание
`allowedDeviceProtocolVersions: []`	В ПСУ включена поддержка всех версий протокола. ПСУ аутентифицирует устройства, поддерживающие протокол версии 3. Устройства, не поддерживающие протокол версии 3 взаимодействуют с ПСУ без аутентификации. Данное значение используется по умолчанию.
`allowedDeviceProtocolVersions: [0, 2]`	Устройства взаимодействуют с ПСУ по протоколу версии 0 или версии 2. Аутентификация устройств не осуществляется.
`allowedDeviceProtocolVersions: [3]`	Взаимодействовать с ПСУ могут только устройства, поддерживающие протокол версии 3. ПСУ осуществляет аутентификацию устройств.

3.7.2. Описание настройки ПУ

Настройка ПУ заключается в настройке взаимодействия Сервера приложений ПУ с Сервисом уведомлений Аврора версии 1.1.2 или ПСУ, а также в загрузке картографической информации в файловое хранилище ПУ.

3.7.2.1. Настройка взаимодействия Сервера приложений ПУ с ПСУ

В случае необходимости взаимодействия Сервера приложений ПУ с ПСУ потребуется выполнить следующие настройки:

3.7.2.1.1 Синхронизировать время между Сервером приложений ПУ и ПСУ.

3.7.2.1.2 Зарегистрировать в ПСУ проект и получить конфигурационные файлы с настройками: `mobile_app_ac_push_project.json` и `app_server_ac_push_project.json`. Инструкция по созданию проекта представлена в документе «Руководство пользователя. Часть 5. Подсистема Сервис уведомлений» АДМГ.20134-01 90 01-5.

3.7.2.1.3 Задать протокол, домен и порт для обращения к ПСУ (значение параметра должно соответствовать значению параметра `push_public_address` в конфигурационном файле `app_server_ac_push_project.json`). Для этого в конфигурационном файле `config/config.yml.j2` задать параметр `config.publicUris.push.publicAddress`, например:

```
publicUris:
    push:
      publicAddress: "https://acenter.example.ru"
```

3.7.2.1.4 Задать домен `<AC_DOMAIN>` и порт ПСУ для устройств (push-демона)

Домен и порт ПСУ для устройств задаются в секции `pushNotificationSystem` конфигурационного файла `config/config.yml.j2`. и распространяются на все тенанты. По умолчанию в качестве `mobileHostname` указано имя первого сервера приложений.

```
  pushNotificationSystem:
   mobileHostname: "acenter.example.ru"
   mobilePort: 999
```

Также домен и порт можно задать в Консоли администратора ПУ при выполнении пп. 3.7.2.1.6. В данном случае настройки будут распространяться только на тенант в рамках которого выполнялась настройка.

3.7.2.1.5 Переустановить ПУ в соответствии с п. 3.5.2, в случае если настройка осуществляется после установки ПУ.

3.7.2.1.6 В Консоли администратора ПУ («Администрирование» - «Настройки» - «Интеграция» - «Сервис уведомлений Аврора») задать параметры взаимодействия Сервера приложений ПУ и ПСУ.

Описания назначения параметров и порядок настройки приведены в документе «Руководство пользователя. Часть 3. Подсистема Платформа управления» АДМГ.20134-01 90 01-3.

3.7.2.2. Ручная настройка доступа к приложению «Аврора Центр» для ОС Аврора и ОС Android

По умолчанию доступ к приложению «Аврора Центр» настраивается автоматически в процессе установки ППО.

Для ручной настройки доступа к приложению «Аврора Центр» необходимо выполнить следующие действия:

3.7.2.2.1 Выложить установочный файл (APK-файл или RPM-пакет) приложения «Аврора Центр» в файловое хранилище ПУ согласно параметру `root` секции `location /clientDownload` конфигурационного файла `/etc/nginx/conf.d/ocs-emm-static-files.nginx.conf` (по умолчанию каталог: `/ocs/emm/clients`), либо параметру `root` конфигурационного файла `install-<версия ППО>/install-ac/config/subsystems/emm/applications/ocs-emm-static-files/ocs-emm-static-files.nginx.conf.j2` сценариев установки ППО.

Установочные файлы должны быть размещены следующим образом:

```
./clients/
├── android-<версия ОС>
│   └── client.noarch.apk
└── aurora-<версия ОС>
    └── client.<архитектура>.rpm
```

Например:

```
/ocs/emm/clients/
├── android-10
│   └── client.noarch.apk
└── aurora-5.0.0
    ├── client.arm.rpm
    └── client.arm64.rpm
```

В случае если имя установочного файла отличается от шаблона, то вместо него можно использовать символическую ссылку (symbolic link) на рядом лежащий файл. Для создания символической ссыки необходимо использовать команду:

```
ln -sf <относительный путь к apk-файлу> client.noarch.apk
или
ln -sf <относительный путь к rpm-пакету> client.rpm
```

Например:

```
ln -sf omp-emm-client-4.0.0.4+2-android.armeabi-v7a.apk
client.noarch.apk
```

Описание настройки файлового хранилища ПУ для размещения в нем установочных файлов приложения «Аврора Центр» приведено в п. 3.8.1.

3.7.2.2.2 В секции config.provisioning.android.signatureChecksum конфигурационного файла ПУ install-<версия ППО>/install-ac/config/subsystems/emm/config.yml задать отпечаток сертификата ключа проверки ЭП в кодировке base64, с помощью которого будет выполняться проверка ЭП APK-файлов.

Например:

```
  provisioning:
    android:
      signatureChecksum: "xHOmnjhoe_m-8NcBRphnlH9h3DwagZdIPaWfacX8stE"
```

Получить отпечаток сертификата можно из APK-файла с помощью команды:

```
keytool -printcert -jarfile <имя файла> | perl -nle "print $& if
m{(?<=SHA256:) .*}" | xxd -r -p | openssl base64 | tr -d '=' | tr --
'+/=' '-_'
```

ПРИМЕЧАНИЕ. Утилита Keytool входит в состав Java SDK (или JRE).

3.7.2.2.3 Переустановить конфигурационные файлы сервиса `ocs-emm-enrollments-api` с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --subsystems emm --
apps ocs-emm-enrollments-api --action config
```

3.7.2.3. Загрузка картографической информации в файловое хранилище ПУ

ПРИМЕЧАНИЕ. По умолчанию ППО настроено на работу с картой Москвы и Московской области. Для получения карты Евразии необходимо обратится в службу технической поддержки предприятия-изготовителя.

Для загрузки картографической информации в хранилище ПУ, настроенное в соответствии с подразделом 3.8, необходимо выполнить следующие действия:

3.7.2.3.1 В файловое хранилище ПУ скопировать файл с картографической информацией и в соответствии с параметром `config.mbTilesSource` конфигурационного файла `config/subsystems/emm/applications/ocs-emm-locations-api/ocs-emm-locations-api.yml` (по умолчанию: `/ocs/emm/maps/map.osm.mbtiles`) создать символическую ссылку (`symbolic link`) на файл с картографической информацией с помощью следующих команд:

```
cp <имя файла> /ocs/emm/maps/
sudo ln -sf /ocs/emm/maps/<имя файла> /ocs/emm/maps/map.osm.mbtiles
```

Также допускает не использовать символическую ссылку, а размещать непосредственно сам файл в соответствии с параметром `config.mbTilesSource`.

3.7.2.3.2 Перезапустить сервис `ocs-emm-locations-api` с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --subsystems emm --
apps ocs-emm-locations-api --action restart
```

3.7.3. Описание настройки ПООС

Настройка ПООС заключается в загрузке пакетов ОС в файловое хранилище.

Для загрузки пакетов ОС в файловое хранилище ПООС, настроенное в соответствии с подразделом 3.8, необходимо выполнить следующие действия:

3.7.3.1. Скопировать в произвольный каталог файлового хранилища ПООС архив с пакетами ОС и распаковать его в каталог, заданный в параметре `root` секции `location /pkgrepo/mobile` конфигурационного файла `/etc/nginx/conf.d/locations-external/ocs-pkgrepo-nginx-static.location` (по умолчанию каталог: `/ocs/pkgrepo/repos`), либо в параметре `repos_root` конфигурационного файла `install-<версия ППО>/install-ac/config/subsystems/pkgrepo/vars/ocs-pkgrepo-nginx-static.yml` сценариев установки ППО:

```
tar -xf <имя файла с архивом> -C <путь к каталогу>
rm <имя файла с архивом>
```

Например,

```
tar -xf 4.0.2.35.tar -C /ocs/pkgrepo/repos
rm 4.0.2.35.tar
```

3.7.3.2. Зарегистрировать переданный релиз (версию), добавив в файл `/ocs/pkgrepo/meta/main.json` описание из специализированного meta-файла.

Meta-файл передается вместе с архивом и представляет собой файл в формате `.json` и имеет название `main.json`. Путь к meta-файлу задается в одном из следующих параметров:

– `alias` секции `location /pkgrepo/mobile/meta` конфигурационного файла `/etc/nginx/conf.d/locations-external/ocs-pkgrepo-nginx-static.location` (по умолчанию каталог: `/ocs/pkgrepo/meta`);

– `meta_root` конфигурационного файла `install-<версия ППО>/install-ac/config/subsystems/pkgrepo/vars/ocs-pkgrepo-nginx-static.yml` сценариев установки ППО.

ВНИМАНИЕ! Приведенные в настоящем пункте примеры заполнения meta-файла приведены исключительно для общего ознакомления с возможной структурой файла. Итоговый meta-файл должен быть сформирован с учетом рекомендаций и примера заполнения, приведенных ниже.

Общие рекомендации по заполнению meta-файла:

– необходимо соблюдать общие правила структуры и синтаксиса формата json при создании meta-файла;

– необходимо корректно указывать следующие данные: модель устройства и версии ОС Аврора, до которых доступно обновление;

– следует придерживаться приведенных рекомендаций по заполнению файла;

– следует использовать инструменты для проверки синтаксиса подготовленного файла.

Meta-файл состоит из нескольких блоков, примеры заполнения которых приведены далее:

1) Общий блок:

```
{
  "brand": "OMP",
  "releases": []
}
```

где:

– "brand": "OMP" - общая информация;

– "releases": [] - блок по моделям;

2) Блок по моделям устройств:

```
{
      "deviceModel": "aq_ns220r",
      "latest": "4.0.2.249",
      "versions": [
        {
          "version": "4.0.2.249",
          "from": [
            "4.0.2.209"
          ]
        },
```

```
        {
          "version": "4.0.2.209",
          "from": [
            "4.0.2.175",
            "4.0.2.89"
          ]
        },
        {
          "version": "4.0.2.175",
          "from": [
            "4.0.2.89",
            "4.0.1.43",
            "4.0.1.20"
          ]
        },
        {
          "version": "4.0.2.89",
          "from": [
            "4.0.1.43",
            "4.0.1.20"
          ]
        },
        {
          "version": "4.0.1.43",
          "from": [
            "4.0.1.20"
          ]
        }
      ]
    }
```


где:

– "deviceModel": "aq_ns220r" - модель устройства Aquarius NS220 v5.2, представленная в кодовом наименовании: "aq_ns220r".

ПРИМЕЧАНИЕ. В случае если кодовое наименование устройства неизвестно следует запросить информацию у производителя:

– "latest": "4.0.2.249" - последняя доступная версия ОС Аврора для устройства;

– "versions": [] - блок списка версий;

3) Блок списка версий:


```
{
          "version": "4.0.2.249",
          "from": [
            "4.0.2.209"
          ]
        }
```

где:

– "version": "4.0.2.249" - необходимая версия ОС Аврора;

– "from": ["4.0.2.209"] - список версий ОС, с которых можно обновить устройство до необходимой версии ОС Аврора.

Пример заполненного meta-файла, составленный для устройства Aquarius NS220R с указанием возможности обновления ОС Аврора с версии 4.0.2.209 до версии 4.0.2.249:

```
{
   "brand":"OMP",
   "releases":[
      {
         "deviceModel":"aq_ns220r",
         "latest":"4.0.2.249",
         "versions":[
            {
               "version":"4.0.2.249",
               "from":[
                  "4.0.2.209"
               ]
            }
         ]
      }
   ]
}
```

3.7.3.3. Перезапустить сервис `ocs-pkgrepo-pkg-repo-api` с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --subsystems pkgrepo
--apps ocs-pkgrepo-pkg-repo-api --action restart
```

3.7.3.4. Проверить корректность настройки, для чего необходимо войти в Консоль администратора ПУ, далее перейти в подраздел «Настройки» раздела «Администрирование», в раскрывающемся поле «Интеграция» выбрать вкладку «Обновление ОС» и убедиться, что отображаются имя сервера, модели устройств и доступные версии ОС (Рисунок 16).

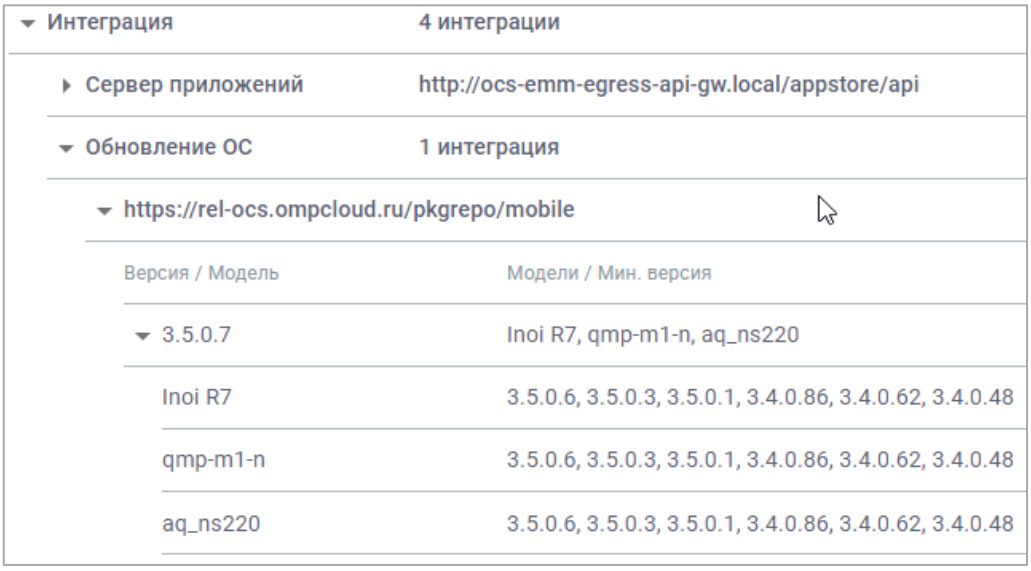


Рисунок 16

3.7.4. Описание настройки CDN

Настройка CDN заключается в настройке контент-серверов.

Для настройки контент-серверов необходимо выполнить следующие действия:

3.7.4.1. В секции `content` инвентарного файла `inventories/hosts.yml` задать адреса серверов (имена хостов), на которые будут установлены контент-серверы.

Например:

```
...
        content:
          hosts:
            acentercdn01:
            acentercdn02:
            acentercdn03:
```

Описание порядка задания адресов в инвентарном файле `inventories/hosts.yml` приведено в п. 3.9.12.

3.7.4.2. Раскомментировать секцию `content_servers_map` конфигурационного файла `config/vars/_vars.yml`.

3.7.4.3. В данной секции задать http адрес контент-сервера по умолчанию, параметр `content_servers_map.default`:

```
...
```

```
content_servers_map:
  default: "<адрес контент сервера>"
```

На контент-сервер по умолчанию будут перенаправлять запросы из сетей/подсетей отсутствующих в секции content_servers_map.content_servers.

3.7.4.4. В секции content_servers_map.content_servers конфигурационного файла config/vars/_vars.yml, при необходимости, задать правила перенаправления запросов на контент-серверы (т.е. задать из каких сетей/подсетей на какие контент-серверы будут перенаправляться запросы):

```
...
  content_servers:
    - server: "<адрес контент сервера>"
      addresses:
        - "<адрес сети/подсети>"
        - "<адрес сети/подсети>"
```

Например:

```
...
  content_servers:
    - server: "http://ocs-cdn01.test.ru"
      addresses:
        - "192.168.79.128:8009"
        - "192.168.79.133"
    - server: "http://ocs-cdn02.test.ru"
      addresses:
        - "192.168.0.0/16"
```

3.7.4.5. В секции content_servers_map.delete конфигурационного файла config/vars/_vars.yml при необходимости задать адреса сетей/подсетей, на которые не должны распространяться правила из секции content_servers_map.content_servers.

ПРИМЕЧАНИЕ. Для обеспечения отказоустойчивости контент-сервер может быть развернут в многонодовой конфигурации с внешним балансировщиком нагрузки. Для этого рекомендуется воспользоваться информацией, приведенной в п. 3.9.9, а также примером конфигурационного файла внешнего балансировщика, приведенного в `samples/ac/nginx_external-balancer/conf.d/content-server.conf`.

3.8. Описание настройки файлового хранилища ППО

3.8.1. Настройка файловых хранилищ подсистем ППО

Для настройки файловых хранилищ подсистем ППО необходимо выполнить следующие действия:

3.8.1.1. Создать каталог `/ocs` и назначить его владельцем пользователя `ocs`, под которым работают сервисы ППО:

```
sudo mkdir -p /ocs
sudo chown ocs:ocs /ocs
```

3.8.1.2. В случае использования единого файлового хранилища необходимо выполнить монтирование данного хранилища к каталогу `/ocs`.

ВНИМАНИЕ! При эксплуатации ППО в кластерной конфигурации все ноды Сервера приложений ППО с ПМ, ПУ и ПООС должны иметь доступ к файловому хранилищу. Соответственно, все ноды Сервера приложений ППО должны быть настроены на работу с данным файловым хранилищем.

Варианты и порядок настройки доступа нод Сервера приложений ППО к файловому хранилищу приведены в п. 3.8.2.

3.8.1.3. Настроить файловое хранилище ПМ, в котором будут храниться файлы приложений (иконки, скриншоты, RPM-пакеты), загружаемые разработчиками.

Для этого в каталоге /ocs необходимо создать каталог в соответствии с параметром filestoragePath конфигурационного файла config/subsystems/appstore/config.yml. В созданном каталоге потребуется создать каталог applications-api и назначить его владельцем пользователя ocs, под которым работают сервисы ПМ:

```
sudo mkdir -p /ocs/appstore/applications-api
sudo chown ocs:ocs /ocs/appstore/applications-api
```

Параметр filestoragePath конфигурационного файла config/subsystems/appstore/config.yml может иметь следующий вид:

```
filestoragePath: "/ocs/appstore"
```

3.8.1.4. Настроить файловое хранилище ПУ, в котором будут храниться установочные файлы приложения «Аврора Центр» и картографическая информация.

Для этого необходимо выполнить следующие действия:

3.8.1.4.1 В каталоге /ocs необходимо в соответствии с параметром root секции location /clientDownload конфигурационного файла /etc/nginx/conf.d/ocs-emm-static-files.nginx.conf (по умолчанию каталог: /ocs/emm/clients), либо параметром root конфигурационного файла install-<версия ППО>/install-ac/config/subsystems/emm/applications/ocs-emm-static-files/ocs-emm-static-files.nginx.conf.j2 сценариев установки ППО создать каталог и назначить его владельцем пользователя ocs, под которым работают сервисы ПУ:

```
sudo mkdir -p <путь к каталогу>
sudo chown ocs:ocs <путь к каталогу>
```

Например:

```
sudo mkdir -p /ocs/emm/clients
sudo chown ocs:ocs /ocs/emm/clients
```

3.8.1.4.2 В каталоге /ocs необходимо в соответствии с параметром config.mbTilesSource конфигурационного файла config/subsystems/emm/applications/ocs-emm-locations-api/ocs-emm-locations-api.yml (по умолчанию: /ocs/emm/maps/map.osm.mbtiles) создать каталог и назначить его владельцем пользователя ocs, под которым работают сервисы ПУ:

```
sudo mkdir -p <путь к каталогу>
sudo chown ocs:ocs <путь к каталогу>
```

Например:

```
sudo mkdir -p /ocs/emm/maps
sudo chown ocs:ocs /ocs/emm/maps
```

3.8.1.5. Настроить файловое хранилище ПООС, в котором будут храниться пакеты ОС.

Для этого в каталоге `/ocs` необходимо создать каталог согласно параметру root секции `location /pkgrepo/mobile` конфигурационного файла `/etc/nginx/conf.d/locations-external/ocs-pkgrepo-nginx-static.location` (по умолчанию каталог: `/ocs/pkgrepo/repos`), либо параметру `repos_root` конфигурационного файла `install-<версия ППО>/install-ac/config/subsystems/pkgrepo/vars/ocs-pkgrepo-nginx-static.yml` сценариев установки ППО:

```
mkdir -p <путь к каталогу>
```

Например,

```
mkdir -p /ocs/pkgrepo/repos
```

3.8.2. Настройка доступа нод сервера приложений ППО к файловому хранилищу

При эксплуатации ППО в кластерной конфигурации все ноды сервера приложений ППО должны иметь доступ к файловому хранилищу, в котором располагаются файлы приложений, пакеты ОС, картографическая информация. Доступ

нод сервера приложений к файловому хранилищу может быть организован следующими способами:

– синхронизация файлов между файловыми хранилищами каждой ноды сервера приложений с помощью приложения Syncthing;

– использование единого файлового хранилища.

3.8.2.1. Настройка синхронизации файлов между файловыми хранилищами каждой ноды сервера приложений с помощью приложения Syncthing

Приложение Syncthing выполняет синхронизацию файлов в режиме реального времени между нодами сервера приложений ППО.

Для настройки и установки Syncthing необходимо выполнить следующие действия:

3.8.2.1.1 Придумать пароль пользователя (по умолчанию пользователь ocs) для доступа к графическому интерфейсу приложения.

3.8.2.1.2 Сформировать хэш-код пароля с использованием алгоритма bcrypt с помощью следующих команд:

```
sudo apt install python3-passlib python3-bcrypt
ansible all -i localhost, -m debug -a "msg={{ '<пароль>' |
password_hash('bcrypt') }}"
```

3.8.2.1.3 В параметре syncthing.gui_password конфигурационного файла config/secret.yml задать значение хэш-кода пароля:

```
  syncthing:
    gui_password:
'$2a$10$N.9m94jj3ciTDtlUhxudwu2rGE3jgb4A0GUCT30KsUIEIbdcPZIx6'
```

3.8.2.1.4 В параметре syncthing.gui_apikey конфигурационного файла config/secret.yml ключ доступа к API приложения:

```
    gui_apikey: 'AgpY4dv2tdcwcNXSmhnxW55euHD55Eyf'
```

3.8.2.1.5 В секции `syncthing_folders` конфигурационного файла `config/vars/_vars.yml` при необходимости изменить предустановленные значения каталогов, для которых требуется выполнять синхронизацию:

```
syncthing_folders:
  pkgrepo:
    path: /ocs/pkgrepo
  appstore:
    path: /ocs/appstore
  emm:
    path: /ocs/emm
```

3.8.2.1.6 Установить приложение с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c syncthing
```

Например:

```
ANSIBLE_USER="omp" ./deploy-infra.sh -c syncthing
```

ВНИМАНИЕ! При установке приложения осуществляется генерация ключевой пары, которая используется для защищенного обмена данными между экземплярами приложения, развернутыми на серверах. Для смены ключевой пары (например, в случае ее компрометации), необходимо удалить приложение и установить его заново.

Удаление приложения осуществляется с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -c syncthing -A
flush_all
```

3.8.2.2. Пример настройки единого файлового хранилища

Единое файловое хранилище применяется для хранения файлов приложений (иконки, скриншоты, RPM-пакеты) и пакетов ОС.

Для настройки единого файлового хранилища необходимо выполнить следующие действия:

3.8.2.2.1 Установить NFS сервер в соответствии с официальной документацией на ОС RedHat, приведенной на странице: `https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/nfs-serverconfig`.

3.8.2.2.2 На Серверах приложений ПМ и ПООС создать каталог `/ocs` и назначить его владельцем пользователя `ocs`, под которым работают сервисы ПМ и ПООС:

```
sudo mkdir -p /ocs
sudo chown ocs:ocs /ocs
```

ВНИМАНИЕ! При эксплуатации ППО в кластерной конфигурации все ноды серверов приложений ПМ и ПООС должны иметь доступ к единому файловому хранилищу. Соответственно, все ноды серверов приложений ПМ и ПООС должны быть настроены на работу с данным файловым хранилищем.

3.8.2.2.3 Выполнить монтирование файловой системы NFS к каталогу `/ocs` с помощью команды:

```
mount example.com:/export/ocsfs /ocs
```

где:

- `example.com` – имя узла файлового сервера NFS;
- `/export/ocsfs` – каталог, который экспортирует `example.com`;
- `/ocs` – каталог, к которому осуществляется монтирование.

ПРИМЕЧАНИЕ. Монтирование файловой системы NFS также может быть выполнено посредством редактирования файла `/etc/fstab`. Для этого в данный файл необходимо добавить запись следующего вида:

```
example.com:/export/ocsfs /ocs nfs defaults 0 0
```

Редактирование файла `/etc/fstab` должно осуществляться суперпользователем.

3.8.2.2.4 Для проверки корректности монтирования необходимо выполнить команду:

```
ls /ocs
```

и убедиться, что полученный список файлов соответствует списку файлов в каталоге `/export/ocsfs` на компьютере `example.com`.

3.9. Дополнительные настройки ППО и среды функционирования ППО

3.9.1. Настройка взаимодействия сервера приложений ПУ с SMTP-сервером

Настройка взаимодействия Сервера приложений ПУ с SMTP-сервером требуется для обеспечения отправки на электронную почту файла со списком системных сообщений об ошибках устройств.

ПРИМЕЧАНИЕ. Функционал доступен только для устройств с ОС Аврора.

Для настройки взаимодействия Сервера приложений ПУ с SMTP-сервером необходимо в секции `smtp` конфигурационного файла подсистемы ПУ `config.yml` (`config/subsystems/emm/config.yml`) задать требуемые значения:

– адрес электронной почты, с которого отправляются письма (параметр: `from`);

– адрес сервера электронной почты (параметр: `address`);

– флаг включения/отключения использования защищенного протокола SMTPS для защиты соединения (параметр: `tls`);

– тип аутентификации (параметр: `authType`);

– параметры для заданного типа аутентификации (`usrername`, `password` и др.).

Значения параметров `from` и `username` должны быть идентичны, в противном случае почтовый сервер будет отклонять сообщения.

С помощью флага "tls" настраивается необходимость использование протокола SMTPS для защиты соединения между Сервером приложений ПУ и SMTP-сервером.

При значении флага "tls: true" будет использоваться защищенный протокол SMTPS. Если SMTP-сервер не поддерживает протокол SMTPS, то Сервер приложений ПУ вернёт ошибку "tls: first record does not look like a TLS handshake" и сообщение отправлено не будет.

При значении флага "tls: false" будет использоваться протокол SMTP. В случае, если SMTP-сервер поддерживает расширение STARTTLS, то передача сообщений будет осуществляться с использованием протокола TLS.

В ПУ поддерживаются LOGIN, CRAM-MD5, PLAIN типы аутентификации SMTP. В зависимости от используемого типа аутентификации необходимо задать следующие параметры (остальные параметры оставить без изменений):

– LOGIN:

```
smtp:
  from: "user@example.com"
  address: "smtp.example.com:1025"
  tls: true
  authType: "LOGIN"
  username: "test_username"
  password: "test_password"
```

– CRAM-MD5:

```
smtp:
  from: "user@example.com"
  address: "smtp.example.com:1025"
  tls: true
  authType: "CRAM-MD5"
  username: "test_username"
  secret: "test_secret"
```

– без аутентификации:

```
smtp:
  from: "user@example.com"
  address: "smtp.example.com:1025"
  tls: true
```

– PLAIN:

```
smtp:
  from: "user@example.com"
  address: "smtp.example.com:1025"
  tls: true
  authType: "PLAIN"
  host: "smtp.example.com"
  username: "test_username"
  password: "test_password"
  identity: "identity"
```

ПРИМЕЧАНИЕ. Для аутентификации типа "`PLAIN`" в поле `host` должно быть указано значение из поля `address` без указания порта. Так же для данного типа аутентификации необходимо наличие защищенного TLS-соединения. Если TLS-соединение не поддерживается, то при попытке отправить сообщение возникнет ошибка "`error: unencrypted connection`" и сообщение не будет отправлено.

После изменения настроек необходимо переустановить конфигурационные файлы с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --subsystems emm --
apps ocs-emm-dispatcher-api,ocs-emm-enrollments-api --action config
```

3.9.2. Настройка разделения трафика

ППО позволяет разделять входящий трафик (URL-запросы) следующими способами:

– по `basepath` - каждая Консоль администратора/разработчика (либо API для взаимодействия с приложениями) привязана к определенному `basepath`. `Basepath` заданы в секции `config.publicUris` конфигурационного файла `internal.yml`;

– по доменам (субдоменам) – каждая Консоль администратора/разработчика и API для взаимодействия с приложениями (либо группа консолей и API) опционально может быть привязана к определенному домену. Рекомендуется публичные консоли и API привязывать к домену, который имеет доступ из сети Интернет, а внутренние консоли (Консоли администраторов) привязывать к домену,

не имеющему доступ из сети Интернет. Разделение трафика по доменам приведено в пп. 3.4.2.3;

– по портам – внутренние и внешние адреса ППО привязаны к отдельным портам. Описание настройки разделения трафика по портам приведено в пп. 3.4.2.2.

3.9.3. Настройка кэширования ответов сервисов

Для увеличения производительности ППО применяется кэширование ответов сервисов с помощью Nginx. При этом доступ к закэшированным данным осуществляется через шлюзы доступа ППО.

Настройки кэширования задаются в следующих конфигурационных файлах сценариев установки среды функционирования ППО:

1) В конфигурационном файле `shared_roles/nginx/defaults/main.yml` задаются следующие параметры:

– `cache_enabled` - включение/выключение кэширования;

– `cache_path` - каталог хранения кэша;

– `cache_keys_zone` - имя зоны в разделяемой памяти, где будет храниться кэш;

– `cache_keys_zone_size` - размер зоны в разделяемой памяти;

– `cache_max_size` - максимальный размер выделяемой под кэш памяти (когда место заканчивается, Nginx удаляет устаревшие данные);

– `cache_inactive` - время, после которого кэш будет автоматически очищаться.

Например:

```
cache_enabled: true
cache_path: "/var/cache/nginx"
cache_keys_zone: "proxy_cache"
cache_keys_zone_size: "50m"
cache_max_size: "10G"
cache_inactive: "30m"
```

ПРИМЕЧАНИЕ. Максимальный размер выделяемой под кэш памяти должен быть не менее 10 ГБ;

2) В конфигурационных файлах `config/subsystems/<название подсистемы>/vars/services.yml` задаются API функции (endpoint-ы) ППО, для которых необходимо выполнять кэширование, а также параметры кэширования для каждой API функции:

– `proxy_cache` - включение кэширования для API функции;

– `proxy_cache_valid` - время кэширования ответа (возможно задать время кэширования для определенных статусов ответа);

– `proxy_cache_lock` - параметр определяет возможность прохождения нескольких запросов на бэкенд (к сервисам ППО). При значении «on» запрещается прохождение нескольких запросов к сервису ППО, все повторные запросы будут ожидать появления ответа в кэше либо таймаута блокировки запроса к странице;

– `proxy_cache_use_stale` - параметр определяет, в каких случаях можно использовать устаревший закэшированный ответ;

– `add_header: "X-Cache-Status  $upstream_cache_status"` - директива добавляет http-заголовок, содержащий статус кэширования.

Например:

```
...
      nginx_location_dashboard:
        path: "~ /v1/dashboards/[^/]+$"
        proxy_cache: "proxy_cache"
        proxy_cache_valid: "200 {{ cache_interval_dynamic }}"
        proxy_cache_lock: "on"
        proxy_cache_use_stale: "updating"
        proxy_cache_background_update: "on"
        add_header: "X-Cache-Status $upstream_cache_status"
```

3.9.4. Действия по безопасной установке и настройке средства

ПРИМЕЧАНИЕ. Установка, настройка и эксплуатация ППО должна осуществляться в соответствии с ЭД на ППО.

При использовании ППО в государственных информационных системах (ГИС) (информационных системах персональных данных, автоматизированных системах управления, критической информационной инфраструктуре), не содержащих информации, составляющей государственной тайны, в зависимости от класса защищенности должны быть установлены значения параметров, приведенные в таблице (Таблица 17).

Таблица 17

Параметр	Значение (для ГИС 4-го класса)	Значение (для ГИС 3-го класса)	Значение (для ГИС 2-го класса)	Значение (для ГИС 1-го класса)
Конфигурационный файл ПБ (сценария установки ПБ): `/var/ocs/config/subsystems/auth/config.yml` (`config/subsystems/auth/config.yml`)				
Период времени неиспользования идентификатора (учетной записи) пользователя, через которое происходит его блокирование: `config.maxAccountInactivityPeriod`	Устанавливается на усмотрение оператора ИС, например: `maxAccountInactivityPeriod: 2160h`	Не более 90 дней, например: `maxAccountInactivityPeriod: 2160h`	Не более 90 дней, например: `maxAccountInactivityPeriod: 2160h`	Не более 45 дней, например: `maxAccountInactivityPeriod: 1080h`
Минимальная длина пароля: `config.passwordSettings.minLength`	Не менее 6 символов, например: `config.passwordSettings.minLength: 6`	Не менее 6 символов, например: `config.passwordSettings.minLength: 6`	Не менее 6 символов, например: `config.passwordSettings.minLength: 6`	Не менее 8 символов, например: `config.passwordSettings.minLength: 8`

Алфавит пароля для учетных записей пользователей не настраивается. Пароли учетных записей пользователей должны содержать буквы верхнего и нижнего регистров, цифры и специальные символы (это контролируется ППО).	Не менее 30 символов, например:
	`minDigits: 1`
	`minUpperLetters: 0`
	`minLowerLetters: 1`
	`minSpecialChars: 0`
		Не менее 60 символов, например:
		`minDigits: 1`
		`minUpperLetters: 1`
		`minLowerLetters: 1`
		`minSpecialChars: 0`
			Не менее 70 символов, например:
			`minDigits: 1`
			`minUpperLetters: 1`
			`minLowerLetters: 1`
			`minSpecialChars: 1`
				Не менее 70 символов, например:
				`minDigits: 1`
				`minUpperLetters: 1`
				`minLowerLetters: 1`
				`minSpecialChars: 1`

Параметр	Значение (для ГИС 4-го класса)	Значение (для ГИС 3-го класса)	Значение (для ГИС 2-го класса)	Значение (для ГИС 1-го класса)
Алфавит пароля для учетных записей устройств: – минимальное число цифр в пароле: – `config.passwordSettings.minDigits` – минимальное число букв верхнего регистра в пароле: – `config.passwordSettings.minUpperLetters` – минимальное число букв нижнего регистра в пароле: – `config.passwordSettings.minUpperLetters` – минимальное число спецсимволов в пароле: `config.passwordSettings.minSpecialChars`				
Максимальное время действия пароля: `config.passwordExpirationTime`	Не более 180 дней, например: `passwordExpirationTime: "4320h"`	Не более 120 дней, например: `passwordExpirationTime: "2880h"`	Не более 90 дней, например: `passwordExpirationTime: "2160h"`	Не более 60 дней, например: `passwordExpirationTime: "1440h"`
Максимальное время действия ключа учетной записи сервера приложений:	Не более 1 года и 3 мес., например: `client_jwks: 10950h`	Не более 1 года и 3 мес., например:	Не более 1 года и 3 мес., например: `client_jwks: 10950h`	Не более 1 года и 3 мес., например:

Параметр	Значение (для ГИС 4-го класса)	Значение (для ГИС 3-го класса)	Значение (для ГИС 2-го класса)	Значение (для ГИС 1-го класса)
`config.ttl.client_jwks`		`client_jwks: 10950h`		`client_jwks: 10950h`
Число последних использованных паролей, которые запрещено использовать пользователями при создании новых паролей: `config.passwordHistoryDepth`	Устанавливается на усмотрение оператора ИС, например: `passwordHistoryDepth: 3`	Устанавливается на усмотрение оператора ИС, например: `passwordHistoryDepth: 3`	Устанавливается на усмотрение оператора ИС, например: `passwordHistoryDepth: 3`	Устанавливается на усмотрение оператора ИС, например: `passwordHistoryDepth: 3`
Максимальное количество неуспешных попыток аутентификации (ввода неправильного пароля) до блокировки: `config.failedLoginTries`	От 3 до 10 попыток, например: `failedLoginTries: 10`	От 3 до 10 попыток, например: `failedLoginTries: 10`	От 3 до 8 попыток, например: `failedLoginTries: 8`	От 3 до 4 попыток, например: `failedLoginTries: 4`
Время блокировки учетной записи пользователя в случае достижения установленного максимального количества неуспешных попыток аутентификации: `config.failedLoginBlockTime`	От 3 до 15 минут, например: `failedLoginBlockTime: "3m"`	От 5 до 30 минут, например: `failedLoginBlockTime: "5m"`	От 10 до 30 минут, например: `failedLoginBlockTime: "10m"`	От 15 до 60 минут, например: `failedLoginBlockTime: "15m"`

Параметр	Значение (для ГИС 4-го класса)	Значение (для ГИС 3-го класса)	Значение (для ГИС 2-го класса)	Значение (для ГИС 1-го класса)
Количество одновременных сессий для привилегированных учетных записей: `config.privilegedSessionsLimit`	Устанавливается на усмотрение оператора ИС, например: `privilegedSessionsLimit: 10`	Устанавливается на усмотрение оператора ИС, например: `privilegedSessionsLimit: 10`	Устанавливается на усмотрение оператора ИС, например: `privilegedSessionsLimit: 10`	Не более 2-х, например: `privilegedSessionsLimit: 2`
Количество одновременных сессий для непривилегированных учетных записей: `config.unprivilegedSessionsLimit`	Устанавливается на усмотрение оператора ИС, например: `unprivilegedSessionsLimit: 10`	Устанавливается на усмотрение оператора ИС, например: `unprivilegedSessionsLimit: 10`	Устанавливается на усмотрение оператора ИС, например: `unprivilegedSessionsLimit: 10`	Устанавливается на усмотрение оператора ИС, например: `unprivilegedSessionsLimit: 10`
Общий конфигурационный файл ППО (шаблон общего конфигурационного файла ППО): `/var/ocs/config/config.yml (config/config.yml.j2)`				
Время бездействия (неактивности) пользователя, через которое осуществляется завершение сеанса пользователя: `config.session.rememberFor`	Устанавливается на усмотрение оператора ИС, например: `rememberFor: 30m`	Устанавливается на усмотрение оператора ИС, например: `rememberFor: 30m`	Не более 15 минут, например: `rememberFor: 15m`	Не более 5 минут, например: `rememberFor: 5m`

3.9.5. Действия по смене аутентификационной информации (паролей, секретов, токенов, ключей)

При эксплуатации ППО должна обеспечиваться периодическая смена аутентификационной информации. Периодичность смены определяется эксплуатирующей организацией. Смена аутентификационной информации также должна осуществляться в случае ее компрометации. К событиям компрометации относятся (но не ограничиваются), следующие события:

– НСД к серверам приложений ППО и/или управляющей ЭВМ;

– потеря носителя, содержащего аутентификационную информацию;

– увольнение сотрудников, имевших доступ к аутентификационной информации;

– возникновение подозрений на утечку аутентификационной информации;

– случаи, когда нельзя достоверно установить, что произошло с носителем аутентификационной информации (например, не понятна причина выхода носителя из строя).

Аутентификационная информация компонентов среды функционирования, а также секретный ключ клиентов (сервисов) и ключ шифрования секретов задаются в конфигурационных файлах `config/vars/_vars.yml` и `config/secret.yml`. Для смены, указанной аутентификационной информации необходимо выполнить следующие действия:

3.9.5.1. Изменить пароли, секреты, токены в конфигурационных файлах `config/vars/_vars.yml` и `config/secret.yml`.

3.9.5.2. Установить компоненты среды функционирования в соответствии с п. 3.5.1.

3.9.5.3. Установить ППО в соответствии с п. 3.5.2.

3.9.6. Действия по реализации функций безопасности среды функционирования ППО

3.9.6.1. Требования к межсетевому экранированию

Необходимо, чтобы защита периметра (физических или логических границ) ИС осуществлялась с использованием межсетевого экрана требуемого класса защиты.

Межсетевой экран должен пропускать трафик только на внешние порты ППО, при этом остальной трафик должен быть запрещен. Перечень внешних портов ППО в зависимости от варианта настройки приведен в таблице (Таблица 18).

Таблица 18

Номер порта (протокол)	Описание	Конфигурационный файл, в котором задается порт	Тип порта[11]
Сервисы ППО «Аврора Центр»			
10000 - 10500 (tcp)	Порты сервисов ППО	`shared_roles/systemd-deploy/templates/systemd-supplicant.sh.j2 /usr/bin/systemd-supplicant.sh`	внутренний
Nginx			
80 (tcp)	Служит для взаимодействия сервисов ППО друг с другом	`shared_roles/consul-template/defaults/main.yml`	внутренний
999 (tls)	Служит для взаимодействия устройств с ПСУ	Конфигурационный файл Nginx согласно п. 3.7.1	внешний
8009 (tcp)	Балансировщик сервисов (Nginx Web Server)	`config/vars/_vars.yml config/config.yml.j2 /etc/nginx/conf.d/ocs.conf`	внешний
8024 (tcp)	Порт для приема запросов от контент-серверов	`config/vars/_vars.yml`	внешний
8025 (tcp)	На данный порт перенаправляются запросы с 999 порта	Конфигурационный файл Nginx согласно п. 3.7.1	внутренний

[11] Описание типов портов приведено в таблице (Таблица 34).

Номер порта (протокол)	Описание	Конфигурационный файл, в котором задается порт	Тип порта[11]
СУБД PostgreSQL			
5432 (tcp)	СУБД PostgreSQL	`shared_roles/postgresql/defaults/main.yml`	внутренний
СУБД Redis			
6379 (tcp)	`redis-server`	`shared_roles/redis/defaults/main.yml`	внутренний
26379 (tcp)	`redis-sentinel`	`shared_roles/redis/defaults/main.yml`	внутренний
Consul			
8300 (tcp)	https://www.consul.io/docs/install/ports		внутренний
8301 (tcp/udp)	https://www.consul.io/docs/install/ports		внутренний
8302 (tcp/udp)	https://www.consul.io/docs/install/ports		внутренний
8600 (tcp/udp)	https://www.consul.io/docs/install/ports	`shared_roles/consul/defaults/main.yml`	внутренний
8500 (tcp)	https://www.consul.io/docs/install/ports	`shared_roles/consul/defaults/main.yml`	внутренний
Redpanda			
8080 (tcp)	`redpanda_console_port`	`shared_roles/redpanda/defaults/main.yml`	внутренний
9092 (tcp)	`redpanda_kafka_api`	`shared_roles/redpanda/defaults/main.yml`	внутренний
9644 (tcp)	`redpanda_admin`	`shared_roles/redpanda/defaults/main.yml`	внутренний
33145 (tcp)	`redpanda_rpc_server`	`shared_roles/redpanda/defaults/main.yml`	внутренний
Syncthing			
8384 (tcp)	https://docs.syncthing.net/users/firewall.html	`shared_roles/syncthing/defaults/main.yml`	внутренний
22000 (tcp/udp)	https://docs.syncthing.net/users/firewall.html	`shared_roles/syncthing/defaults/main.yml`	внутренний
21027 (udp)	https://docs.syncthing.net/users/firewall.html		внутренний
Dnsmasq			
53	`dnsmasq`		внутренний

Номер порта (протокол)	Описание	Конфигурационный файл, в котором задается порт	Тип порта[11]
Операционная система			
22	Порт SSH. Используется для развертывания и администрирования ППО. **ВНИМАНИЕ!** Возможность использования данного порта определяется документацией СЗИ от НСД		внутренний

ПРИМЕЧАНИЕ. Рекомендуется запретить доступ к ППО привилегированных пользователей из-за пределов контролируемой зоны, запретив доступ к Консоли администратора ПБ. Также при необходимости можно запретить доступ к остальным веб-консолям. Для этого следует разрешить трафик только по требуемым URL-адресам в соответствии с п. 3.9.2.

3.9.6.2. Настройка ОС CentOS

3.9.6.2.1 Для затруднения возможностей сбора информации о системе необходимо исключить метки времени из заголовков TCP пакетов, выполнив следующие действия:

3.9.6.2.1.1 В конфигурационный файл `/etc/sysctl.conf` добавить строку:

```
net.ipv4.tcp_timestamps = 0
```

3.9.6.2.1.2 Применить конфигурацию, выполнив команду:

```
sysctl -p /etc/sysctl.conf
```

3.9.6.2.1.3 Проверить корректность конфигурации, выполнив команду:

```
sysctl -a | grep net.ipv4.tcp_timestamps
```

Если настройки заданы правильно, должно быть выведено значение:

```
net.ipv4.tcp_timestamps = 0
```

3.9.6.2.2 Настройка запрета SSH доступа к серверам приложений по логину и паролю.

3.9.6.2.2.1 В конфигурационном файле `/etc/ssh/sshd_config` задать следующие значения параметров:

```
PasswordAuthentication no
AuthenticationMethods publickey
```

3.9.6.2.2.2 Перезапустить службу `sshd` с помощью команды:

```
sudo service sshd reload
```

3.9.6.2.3 Настройка минимальной сложности пароля.

Настройка сложности пароля осуществляется в конфигурационном файле `/etc/security/pwquality.conf`. Рекомендуется задать следующие значения параметров:

– минимальная длина пароля:

```
minlen = 8
```

– алфавит пароля (минимальное количество используемых классов символов):

```
minclass = 4
```

– максимальная длина последовательности символов (abcd, 12345 и т.п.):

```
maxsequence = 3
```

– максимальное число идущих подряд одинаковых символов:

```
maxrepeat = 3
```

3.9.7. Самостоятельная установка необходимых пакетов на серверы приложений, серверы БД и контент-серверы

3.9.7.1. Получить список необходимых пакетов.

Перечень необходимых пакетов, которые должны быть установлены на серверы приложений, серверы БД и контент, задан в файле `play-managed-node-`

prerequisites.yml, находящемся в каталоге со сценариями установки ППО и имеющем следующую структуру:

```
...
    - name: install requirements to <операционная система>
...
        - name: install os packages on db node
          loop:
            <перечень пакетов сервера БД>
        - name: install os packages on app node
          loop:
            <перечень пакетов сервера приложений>
```

В секции name: install requirements to <операционная система> задается перечень пакетов для указанной ОС. Данная секция содержит 2 подсекции, в которых задается перечень пакетов для сервера приложений, сервера БД и контент-сервера.

В подсекции name: install os packages on db node задается перечень пакетов для сервера БД.

В подсекции name: install os packages on app node задается перечень пакетов для сервера приложений и контент-сервера.

Пример перечня пакетов для сервера приложений, сервера БД и контент-сервера, функционирующих под управлением ОС CentOS 7:

```
...
  tasks:
    - name: install requirements to CentOS7
      block:
        - debug:
            msg: install requirements to CentOS7

        - name: install os packages on db node
          package:
            name: "{{ item }}"
            state: present
          loop:
            - epel-release
            - jq
            - unzip
            - perl-libs
            - libxslt
            - postgresql-libs
```

```
            - libicu
          when: node_type == "db" or node_type == "all"


        - name: install os packages on app node
          package:
            name: "{{ item }}"
            state: present
          loop:
            - net-tools
            - epel-release
            - jq
            - unzip
            - perl-libs
            - libxslt
            - postgresql-libs
            - libicu
            - dnsmasq
            - bind-utils
          when: node_type == "app" or node_type == "all"
      when: ansible_distribution == "CentOS" and
ansible_distribution_major_version == "7"
```

3.9.7.2. Установить пакеты.

Установка пакетов осуществляется в соответствии с документацией на ОС.

3.9.8. Отключение служб SELinux и Firewalld

Для отключения служб SELinux и Firewalld необходимо выполнить следующие действия:

3.9.8.1. В конфигурационном файле `/etc/selinux/config` задать следующее значение параметра `SELINUX`:

```
SELINUX=disabled
```

3.9.8.2. Отключить в ОС межсетевой экран с помощью выполнения следующих команд:

```
systemctl stop firewalld
systemctl disable firewalld
```

3.9.8.3. Перезагрузить ЭВМ с помощью команды:

```
reboot
```

3.9.9. Требования к установке и настройке внешнего балансировщика (на примере Nginx)

Установка и настройка внешнего балансировщика Nginx осуществляются пользователями (системными администраторами) ППО самостоятельно. Внешний балансировщик должен поддерживать проксирование `http` и `tcp`-соединений.

Для проверки возможности проксирования `tcp`-соединений необходимо выполнить проверку корректности конфигурации Nginx с помощью команды:

```
sudo nginx -t
```

В случае отображения сообщения `unknown directive «stream»` требуется добавить поддержку модуля `ngx_stream_module.so`. Для этого необходимо:

– в конфигурационном файле Nginx (файл: `/etc/nginx/nginx.conf`) добавить строку:

```
load_module '/usr/lib64/nginx/modules/ngx_stream_module.so';
```

– перезапустить Nginx с помощью команды:

```
sudo systemctl reload nginx
```

3.9.9.1. Настройка балансировщика для однотенантной конфигурации:

– выделить домен <AC_DOMAIN> для обращения к ППО, например, `acenter.example.ru`;

– выпустить сертификат для своего домена (доменов), например, `acenter.example.ru`;

– добавить dns-запись для своего домена (доменов), например, `acenter.example.ru`;

– в конфигурационном файле внешнего балансировщика добавить обработку своего домена (доменов), например, `acenter.example.ru`. Примеры конфигурационных файлов приведены в каталоге `samples/ac/nginx_external-balancer/conf.d`:

- one-node.conf – пример конфигурационного файла для сервера приложений;
- content-server.conf – пример конфигурационного файла для контент-сервера.

3.9.9.2. Настройка балансировщика для поддержки мультитенантной конфигурации.

В связи с тем, что для каждого тенанта используется отдельный поддомен, необходимо настроить обработку домена и поддоменов на внешнем балансировщике, выполнив следующие действия:

– выделить домен <AC_DOMAIN> для обращения к ППО, например, acenter.example.ru;

– выпустить обычный и wildcard сертификаты для своего домена (доменов), например acenter.example.ru и *.acenter.example.ru;

– добавить dns-запись для своего домена (доменов) и wildcard запись для поддоменов, например acenter.example.ru и *.acenter.example.ru;

– в конфигурационном файле внешнего балансировщика добавить обработку своего домена (доменов) и поддоменов, например acenter.example.ru и *.acenter.example.ru. Примеры конфигурационных файлов приведены в каталоге samples/ac/nginx_external-balancer/conf.d:

- one-node.conf – пример конфигурационного файла для сервера приложений;
- content-server.conf – пример конфигурационного файла для контент-сервера.

3.9.10. Активация (разблокировка) учетной записи пользователя с помощью sql-запроса к БД

Разблокировка учетных записей пользователей ППО осуществляется Администратором учетных записей с помощью Консоли администратора ПБ. Однако учетная запись Администратора учетных записей также может быть заблокирована (например, при длительной неактивности Администратора учетных записей).

В этом случае для разблокировки учетной записи необходимо выполнить следующие действия:

3.9.10.1. Подключится к БД ПБ (`auth`) с помощью команды:

```
psql -U auth -h <ip-адрес сервера БД> -d auth
```

Например:

```
psql -U auth -h 192.168.0.107 -d auth
```

3.9.10.2. Разблокировать учетную запись пользователя с помощью с sql-запроса:

```
update accounts_users.accounts set is_active=true,
last_activity_at=now() where login='<email пользователя>';
```

Например:

```
update accounts_users.accounts set is_active=true,
last_activity_at=now() where login='admin@omprussia.ru';
```

3.9.11. Действия после сброса устройств к заводским настройкам

Сброс устройства возвращает его к заводским настройкам. После сброса устройств в зависимости от способа их первоначальной установки приложения ППО (приложение «Аврора Центр» и приложение «Аврора Маркет») могут отсутствовать либо быть сброшены до первоначальной версии.

После сброса устройства необходимо выполнить следующие действия:

1) Установить приложения ППО, если после сброса устройства они отсутствуют;

2) Активировать устройство в ПУ в соответствии с документом «Руководство пользователя. Часть 3. Подсистема Платформа управления» АДМГ.20134-01 90 01-3;

3) Обновить приложения ППО в соответствии с документом «Руководство пользователя. Часть 7. Приложение «Аврора Центр» для операционной системы Аврора» АДМГ.20134-01 90 01-7.

3.9.12. Порядок задания адресов (доменных имен) в инвентарном файле inventories/hosts.yml

В инвентарном файле `inventories/hosts.yml` задаются адреса серверов (имена хостов), на которые установлены (будут установлены) компоненты среды функционирования ППО и подсистемы ППО.

Задание адресов (доменных имен) осуществляется посредством их добавления в секцию `hosts`, например:

```
...
        app:
          hosts:
            acenterapp01:
            acenterapp02:
            acenterapp03:
```

Допускается добавление адресов при помощи добавления хостов в группы и дальнейшего переиспользования групп. Например, для Nginx будут заданы адреса из группы `app`, которая заполнена выше:

```
...
    ocs:
      children:
        app:
          hosts:
            acenterapp01:
            acenterapp02:
            acenterapp03:
        nginx:
          children:
            app:
```

Допускается смешанное задание адресов посредством их добавления в секцию `hosts`, а также посредством добавления хостов в группы и дальнейшего переиспользования групп. Например, для Nginx будут заданы адреса из группы `app`, которая заполнена выше, и адреса из секции `hosts`:

```
...
    ocs:
      children:
        app:
          hosts:
            acenterapp01:
            acenterapp02:
            acenterapp03:
        nginx:
          children:
            app:
          hosts:
            acenterapp04:
            acenterapp05:
```

При необходимости установки на хост определенных подсистем ППО потребуется после адреса хоста добавить параметр `subsystems` с перечнем подсистем, например:

```
...
        app:
          hosts:
            acenterapp01:
              subsystems: auth
            acenterapp02:
              subsystems: emm
            acenterapp03:
              subsystems: appstore,pkgrepo
```

Конфигурационный файл сценария установки среды функционирования ППО на 1 ЭВМ с доменным именем `ocs-app.local` имеет следующий вид:

```
all:
  children:
    ocs:
      children:
        app:
          hosts:
            ocs-app.local:
        content:
          hosts:
```

```
        postgresql:
          hosts:
            ocs-app.local:
        # patroni:
        #   children:
        #     primary_cluster:
        #       hosts:
        #     standby_cluster:
        #       hosts:
        nginx:
          children:
            app:
            content:
        consul:
          children:
            consul_servers:
              children:
                app:
            consul_agents:
        consul_content:
          children:
            content:
        consul_template:
          children:
            app:
            content:
        nats_streaming_server:
          children:
            app:
        redpanda:
          children:
            app:
        redis:
          children:
            redis_masters:
              children:
                app:
            sentinel:
              children:
                app:
              hosts:
        syncthing:
          children:
            app:
```

Примеры файлов `hosts.yml` для однонодовой и кластерной конфигураций приведены в каталоге `samples/ac/inventories/` (или в каталоге `samples/ac/inventories/`).

Описание параметров инвентарного файла `inventories/hosts.yml` приведено в п. 11.1.1.

3.9.13. Порядок настройки срока хранения событий безопасности

Срок хранения событий безопасности задается в поле `retention` таблицы `partman.part_config events` БД ПБ (`auth`).

Для просмотра и изменения срока хранения необходимо выполнить следующую последовательность действий:

3.9.13.1. Подключиться к БД ПБ (`auth`) с помощью команды:

```
psql -U auth -h <ip-адрес сервера БД> -d auth
```

Например:

```
psql -U auth -h 192.168.0.107 -d auth
```

3.9.13.2. Просмотреть текущее значение срока хранения с помощью скрипта:

```
select retention from partman.part_config where parent_table =
'audit.audit_events';
```

3.9.13.3. Изменить срок хранения с помощью скрипта:

```
UPDATE partman.part_config SET retention = '<количество дней> days'
where parent_table = 'audit.audit_events';
```

Например:

```
UPDATE partman.part_config SET retention = '90 days' where
parent_table = 'audit.audit_events';
```

3.9.14. Порядок настройки ППО для его установки на различные окружения

Сценарии установки позволяют выполнить настройку и установку ППО, а также компонентов среды функционирования ППО для нескольких различных окружений, выполнив следующие действия:

3.9.14.1. Перейти в каталог (каталог: `/install-<версия ППО>/install-ac/` или `/install-<версия ППО>/install-ac-mt/` или `/install-<версия ППО>/install-ac-mt-spr/`).

3.9.14.2. Создать инвентарный файл `hosts.yml` по пути: `inventories/<название окружения>/hosts.yml`

Описание параметров инвентарного файла `hosts.yml` приведено в п. 11.1.1.

3.9.14.3. Создать каталог `config/environments/<название окружения>/`, создать в данном каталоге требуемые конфигурационные файлы с учетом их расположения в каталоге `config` и задать требуемые значения параметров.

Более подробная информация по работе с конфигурационными файлами окружения приведена в п. 12.2.8.

3.9.14.4. Выполнить установку компонентов среды функционирования ППО и ППО в соответствии с подразделом 3.5 для заданного окружения, указав в командах установки путь к инвентарному файлу и имя окружения.

Примеры команд:

– команда установки всех пакетов на все серверы (на все серверы приложений, серверы БД и контент-серверы независимо от их типа):

```
ansible-playbook -i inventories/<название окружения>/hosts.yml play-managed-node-prerequisites.yml -vv -u <имя пользователя>
```

– команда установки компонентов среды функционирования ППО:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh --env "<название окружения>"
```

– команда установки ППО:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --env "<название окружения>"
```

3.9.15. Удаление персональных данных из учетной записи пользователя, персональных данных контактного лица организации и персональных данных контактного лица проекта

3.9.15.1. Для удаления персональных данных из учетной записи пользователя необходимо выполнить следующие действия:

– подключиться к БД ПБ (`auth`) с помощью следующей команды:

```
psql -U auth -h <ip-адрес сервера БД> -d auth
```

Например:

```
psql -U auth -h 192.168.0.107 -d auth
```

– выполнить sql-запрос:

```
update accounts_users.accounts set login='', last_name='',
first_name='', patronymic='' where login='<email пользователя>';
```

Например:

```
update accounts_users.accounts set login='', last_name='',
first_name='', patronymic='' where login='ivanov@omprussia.ru';
```

3.9.15.2. Для удаления персональных данных контактного лица организации необходимо выполнить следующие действия:

– выполнить архивацию контактного лица, если контактное лицо не заархивировано;

– подключиться к БД ПУТ (mt) с помощью следующей команды:

```
psql -U mt -h <ip-адрес сервера БД> -d mt
```

Например:

```
psql -U mt -h 192.168.0.107 -d mt
```

– выполнить sql-запрос:

```
UPDATE organizations.contact_persons
SET first_name='Deleted',
    last_name='Deleted',
    patronymic=NULL,
    comment=NULL,
    phone='112',
    email=id::text || '@example.com'
WHERE deleted_at IS NOT NULL AND email='<email контактного лица>';
```

Например:

```
UPDATE organizations.contact_persons
SET first_name='Deleted',
    last_name='Deleted',
    patronymic=NULL,
    comment=NULL,
```

```
    phone='112',
    email=id::text || '@example.com'
WHERE deleted_at IS NOT NULL AND email='ivanov@omprussia.ru';
```

3.9.15.3. Для удаления персональных данных пользователей устройств необходимо выполнить следующие действия:

– зайти в карточку пользователя устройства и выполнить архивацию пользователя, если пользователь не заархивирован;

– подключиться к БД ПУ (emm) с помощью следующей команды:

```
psql -U emm -h <ip-адрес сервера БД> -d emm
```

Например:

```
psql -U emm -h 192.168.0.107 -d emm
```

– выполнить sql-запросы:

```
UPDATE users_service.users
SET first_name   = 'Deleted',
    last_name    = 'Deleted',
    patronymic   = NULL,
    job_title    = NULL,
    phone_number = NULL,
    email        = id::text || '@example.com'
WHERE email = '<email пользователя МУ>' AND deleted_at IS NOT NULL;

UPDATE users_service.users_read_model
SET first_name   = 'Deleted',
    last_name    = 'Deleted',
    patronymic   = NULL,
    job_title    = NULL,
    phone_number = NULL,
    email        = id::text || '@example.com'
WHERE email = '<email пользователя МУ>' AND deleted_at IS NOT NULL;
```

Например:

```
UPDATE users_service.users
SET first_name   = 'Deleted',
    last_name    = 'Deleted',
    patronymic   = NULL,
    job_title    = NULL,
    phone_number = NULL,
    email        = id::text || '@example.com'
WHERE email = 'ivanov@omprussia.ru' AND deleted_at IS NOT NULL;

UPDATE users_service.users_read_model
```

```
SET first_name   = 'Deleted',
    last_name    = 'Deleted',
    patronymic   = NULL,
    job_title    = NULL,
    phone_number = NULL,
    email        = id::text || '@example.com'
WHERE email = 'ivanov@omprussia.ru' AND deleted_at IS NOT NULL;
```

3.9.15.4. Для удаления персональных данных контактного лица проекта ПСУ необходимо выполнить следующие действия:

– подключиться к БД ПСУ (`push`) с помощью следующей команды:

```
psql -U push -h <ip-адрес сервера БД> -d push
```

Например:

```
psql -U push -h 192.168.0.107 -d push
```

– выполнить sql-запрос:

```
UPDATE main.contact_persons
SET
    first_name='Deleted',
    last_name='Deleted',
    patronymic='Deleted',
    position='Deleted',
    phone='112',
    email=id::text || '@example.com'
WHERE email='<email контактного лица>';
```

Например:

```
UPDATE main.contact_persons
SET
    first_name='Deleted',
    last_name='Deleted',
    patronymic='Deleted',
    position='Deleted',
    phone='112',
    email=id::text || '@example.com'
WHERE email='ivanov@omprussia.ru';
```

3.9.16. Сброс пароля учетной записи

В случае утери пароля от учетной записи с ролью Администратор учетных записей и невозможности его восстановления штатным способом (например, если в ППО была только 1 учетная запись с указанной ролью), необходимо выполнить следующие действия для сброса пароля учетной записи:

– подключиться к БД ПБ (`auth`) с помощью команды:

```
psql -U auth -h <ip-адрес сервера БД> -d auth
```

Например:

```
psql -U auth -h 192.168.0.107 -d auth
```

– в файле `samples/sql/activate_user_account.sql`, расположенном в каталоге со сценариями установки ППО, задать логин учетной записи в параметре `accountLogin`, например:

```
accountLogin text := 'admin@omprussia.ru'
```

– скопировать содержимое файла `activate_user_account.sql` в консоль и выполнить скрипт, нажав клавишу «Enter».

После выполнения указанных действия пароль будет иметь значение «admin».

3.9.17. Восстановление учетной записи пользователя тенанта в случае ее удаления

Для восстановления учетной записи пользователя тенанта в случае ее удаления необходимо выполнить следующие действия:

– подключиться к БД ПБ (`auth`) с помощью следующей команды:

```
psql -U auth -h <ip-адрес сервера БД> -d auth
```

Например:

```
psql -U auth -h 192.168.0.107 -d auth
```

– в файле `samples/sql/create_tenant_default_user_account.sql`, находящемся в каталоге со сценариями установки ППО, задать логин учетной записи (параметр: `accountLogin`) и код тенанта (параметр: `accountTenantCode`), например:

```
accountLogin text := 'admin@omprussia.ru';
accountTenantCode text := 'default';
```

ПРИМЕЧАНИЕ. Код тенанта доступен в карточке тенанта;

– скопировать в консоль содержимое файла `create_tenant_default_user_account.sql` и выполнить скрипт, нажав клавишу «Enter».

3.9.18. Настройка включения/отключения регистрации событий

Настройка регистрации событий осуществляется в конфигурационных файлах шлюзов доступа `endpoints.yml`, которые располагаются на сервере приложений ППО в каталоге:

```
/var/ocs/config/subsystems/<название
подсистемы>/applications/<название шлюза доступа>/endpoints.yml
```

Например:

```
 /var/ocs/config/subsystems/auth/applications/ocs-auth-admin-api-
gw/endpoints.yml
```

либо в каталоге со сценариями установки ППО:

```
config/subsystems/<название подсистемы>/applications/<название шлюза
доступа>/endpoints.yml
```

Например:

```
config/subsystems/auth/applications/ocs-auth-admin-api-
gw/endpoints.yml
```

Для отключения/включения регистрации события для функции ППО (эндпоинта) необходимо в секции требуемого эндпоинта закомментировать/раскомментировать секцию `audit`, например:

```
- endpoint: /api/identityTypes/user/accounts/{account_id}/block
        method: PUT
        backends:
          - url_pattern: /v1/accounts/{account_id}/block
            host: ['ocs-auth-accounts-users-api.${domain}']
        rp: {}
        auth:
          scope: account:update
        permissions:
          resource_type: userAccount
          action: block
#         audit:
#           field_map:
#             action: block
#             object_id: request.params.account_id
#             object_label: response.body.login
#             object_type: account
```

Далее в зависимости от типа конфигурационного файла выполнить переустановку конфигурационного файла или перезапуск сервиса. Подробная информация об управлении настройками сервисов ППО приведена в подразделе 4.3.

3.9.19. Настройка брендирования ППО

Для добавления логотипа компании и выбора цветовой схемы графического интерфейса ППО необходимо в конфигурационном файле `config/internal.yml` задать следующие параметры:

– `brandingLogoUrl` - ссылка на изображение, либо изображение логотипа в формате base64 (Изображение будет размером 160x32 точек);

– `brandingLogoAlt` – текст, который будет отображаться при наведении на изображение логотипа;

– `theme` - описание цветовой схемы Material UI: цвета кнопок и текста, а также прочие настройки (см. https://mui.com/material-ui/customization/palette/). Не рекомендуется менять шрифты и их размеры, т.к. неправильные значения могут нарушить корректность отображения интерфейса.

Например:

```
brandingLogoUrl: "
...",
brandingLogoAlt: "test branding text",
theme:
  palette:
    primary:
      main: "#ff4200"
      light: "#ff4200"
      dark: "#ff4200"
      contrastText: "#333333"
    secondary:
      main: "#ff4200"
      light: "#ff4200"
      dark: "#ff4200"
      contrastText: '#333333'
```

3.9.20. Переключение трафика между ЦОДами (failover/switchover)

Переключение трафика между ЦОДами (`failover/switchover`) осуществляется в ручном режиме.

ВНИМАНИЕ! При переключении трафика между ЦОДами, а также при восстановлении основного ЦОДа, важно не допустить одновременную работу обоих ЦОДов в активном режиме. При одновременной работе обоих ЦОДов в активном режиме возникнет ситуация «Split Brain».

Для переключения трафика с основного ЦОДа на резервный рекомендуется выполнить следующие действия:

3.9.20.1. Перевести кластер БД в резервном ЦОДе из состояния StandBy в Primary

Для этого необходимо выполнить следующие действия:

3.9.20.1.1 Остановить серверы БД в основном ЦОДе с помощью команды:

```
systemctl stop patroni
```

3.9.20.1.2 Перевести кластер БД из состояния StandBy в состояние Primary, выполнив на одном из серверов БД резервного ЦОДа следующую команду:

```
patronictl -c /etc/patroni.yml edit-config --force --set
standby_cluster=''
```

3.9.20.1.3 Проверить, что в кластере появился сервер БД в роли Leader с помощью команды:

```
patronictl -c /etc/patroni.yml list
```

3.9.20.2. Перевести входящий трафик на внешнем балансировщике из основного ЦОДа в резервный.

3.9.20.3. После восстановления кластера БД в основном ЦОДе, можно перевести кластер БД в основном ЦОДе в состояние `StandBy`.

Для этого необходимо добавить в динамическую конфигурацию кластера настройку «`standby_cluster`» с помощью следующей команды:

```
patronictl -c /etc/patroni.yml edit-config --force    \
--set standby_cluster.host='<new_cluster_vip>' \
--set standby_cluster.port=5000                                  \
--set standby_cluster.create_replica_methods='- basebackup'
```

где `new_cluster_vip` - virtual IP address сервера, с которого будет настроена репликация данных.

ВНИМАНИЕ! В момент восстановления вышедшего ранее основного кластера, он также останется `primary`. Важно не допустить работы двух `primary` кластеров.

3.10. Проверка корректности установки и функционирования ППО

3.10.1. Общие сведения

В целях проверки корректности установки и функционирования ППО, а также среды функционирования ППО, в состав сценариев установки включена утилита для формирования диагностического отчета.

Для формирования диагностического отчета необходимо перейти в каталог со сценариями установки (каталог: `install-<версия ППО>/install-ac/` или `install-<версия ППО>/install-ac-mt/` или `/install-<версия ППО>/install-ac-mt-spr/`) и выполнить команду:

```
ansible-playbook play-diagnostic-report.yml -i inventories/hosts.yml -
vv --user <имя пользователя>
```

либо команду:

```
ansible-playbook play-diagnostic-report.yml -i inventories/<название
окружения>/hosts.yml -vv --user <имя пользователя> --extra-vars
"stage=<название окружения>"
```

если требуется задать окружение.

В результате в каталоге `report` будет сформирован файл в `report.html`.

Диагностический отчет формируется в виде файла в формате `.html` и содержит следующие разделы:

– общая информация о статусе сервисов ППО;

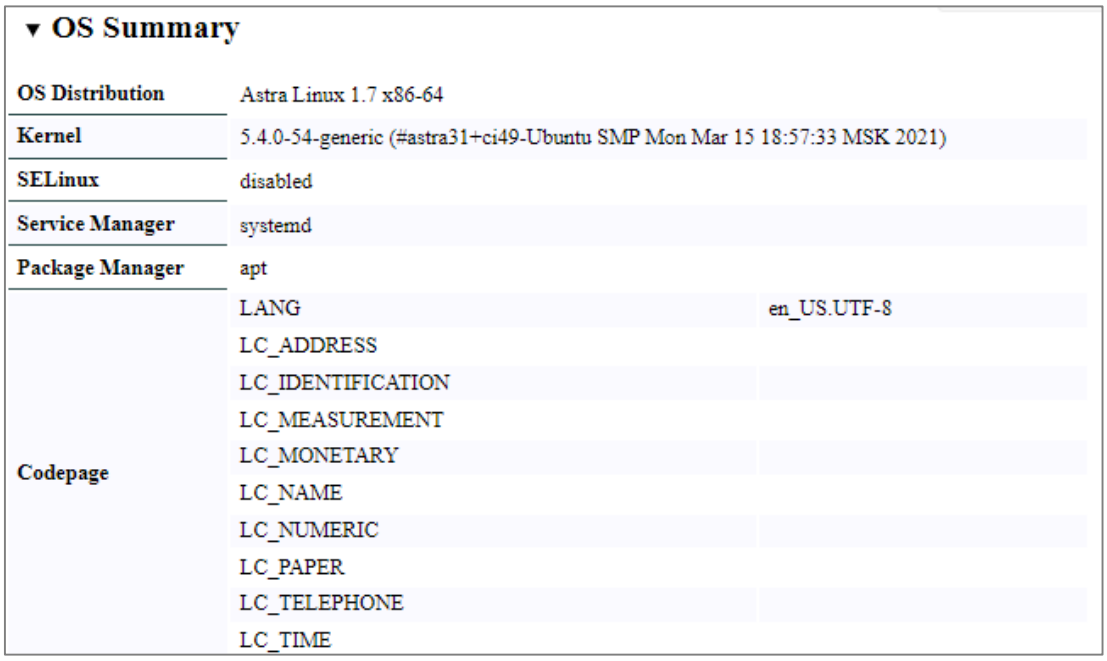
– общая информация о статусе компонентов среды функционирования;

– разделы, содержащие детальную информацию об отдельных сервисах ППО и компонентах среды функционирования.

3.10.2. Описание параметров диагностического отчета

3.10.2.1. Раздел «OS Summary»

Раздел содержит информацию об ОС и ее настройках (Рисунок 17).



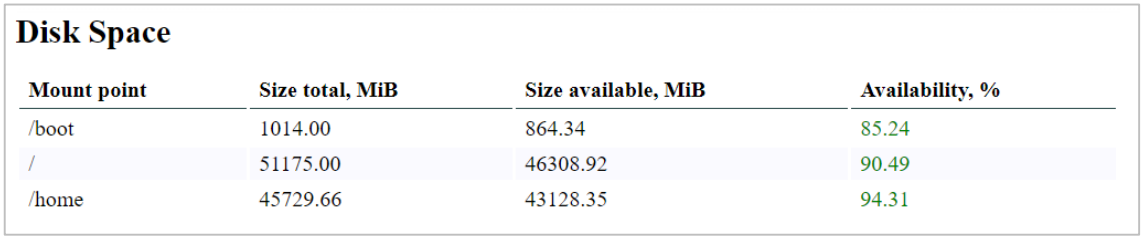
▼ **OS Summary**

OS Distribution	Astra Linux 1.7 x86-64	
Kernel	5.4.0-54-generic (#astra31+ci49-Ubuntu SMP Mon Mar 15 18:57:33 MSK 2021)	
SELinux	disabled	
Service Manager	systemd	
Package Manager	apt	
Codepage	LANG	en_US.UTF-8
	LC_ADDRESS	
	LC_IDENTIFICATION	
	LC_MEASUREMENT	
	LC_MONETARY	
	LC_NAME	
	LC_NUMERIC	
	LC_PAPER	
	LC_TELEPHONE	
	LC_TIME	

Рисунок 17

3.10.2.2. Раздел «Disk Space»

Раздел содержит информацию о полном и доступном объеме дискового пространства для ППО (Рисунок 18).



Disk Space

Mount point	Size total, MiB	Size available, MiB	Availability, %
/boot	1014.00	864.34	85.24
/	51175.00	46308.92	90.49
/home	45729.66	43128.35	94.31

Рисунок 18

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 19).

Таблица 19

Название столбца	Описание	Возможные значения (примеры значений)
Mount points	Каталог, к которому монтируется файловое хранилище (точка монтирования)	Путь к каталогу, например: `/home`
Size total, MiB	Размер файлового хранилища, примонтированного к заданному каталогу	Объем физической памяти в Мб, например: 45729,66
Size available, MiB	Объем свободного места в файловом хранилище, примонтированного к заданному каталогу	Объем физической памяти в Мб, например: 43128,35
Availability, %	Объем свободного места в файловом хранилище в процентном соотношении (к полному объему)	От 0 до 100, например: 94.31

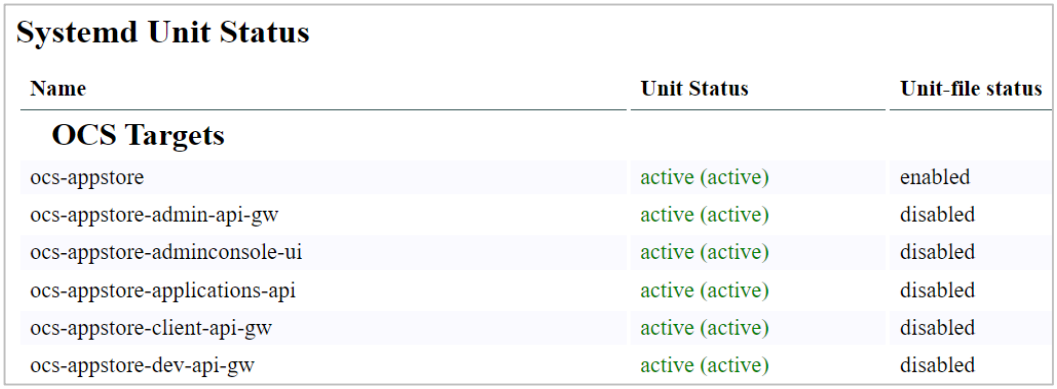
ПРИМЕЧАНИЕ. В случае если объем свободного места менее 15%, поле закрашивается цветом.

3.10.2.3. Раздел «Systemd Unit Status»

В данном разделе приведена общая информация о статусе сервисов ППО и компонентов среды функционирования и состоит из следующих подразделов:

3.10.2.3.1 OCS Targets

Подраздел содержит информацию о статусе конфигураций групп сервисов ППО (Рисунок 19).



Systemd Unit Status

Name	Unit Status	Unit-file status
OCS Targets		
ocs-appstore	active (active)	enabled
ocs-appstore-admin-api-gw	active (active)	disabled
ocs-appstore-adminconsole-ui	active (active)	disabled
ocs-appstore-applications-api	active (active)	disabled
ocs-appstore-client-api-gw	active (active)	disabled
ocs-appstore-dev-api-gw	active (active)	disabled

Рисунок 19

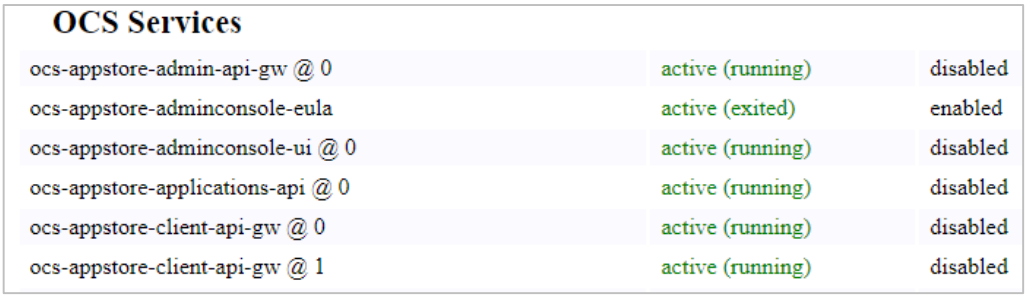
Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 20).

Таблица 20

Название столбца	Описание	Возможные значения (примеры значений)
Name	Имя конфигурации группы сервисов	Возможные значения определяются перечнем конфигураций групп сервисов ППО
Unit Status	Информация о статусе группы сервисов	`active` - группа сервисов запущена и выполняется; `activating` - группа сервисов запускается; `deactivating` - группа сервисов выключается; `inactive` - группа сервисов выключена; `failed` – при запуске группы сервисов произошла ошибка; `missed` - компонент отсутствует
Unit-file status	Информация о присутствии конфигурационного файла запуска группы сервисов в автозапуске	`enabled` - присутствует в автозапуске; `disabled` - отсутствует в автозапуске

3.10.2.3.2 OCS Services

Подраздел содержит информацию о статусе сервисов ППО (Рисунок 20).



OCS Services

ocs-appstore-admin-api-gw @ 0	active (running)	disabled
ocs-appstore-adminconsole-eula	active (exited)	enabled
ocs-appstore-adminconsole-ui @ 0	active (running)	disabled
ocs-appstore-applications-api @ 0	active (running)	disabled
ocs-appstore-client-api-gw @ 0	active (running)	disabled
ocs-appstore-client-api-gw @ 1	active (running)	disabled

Рисунок 20

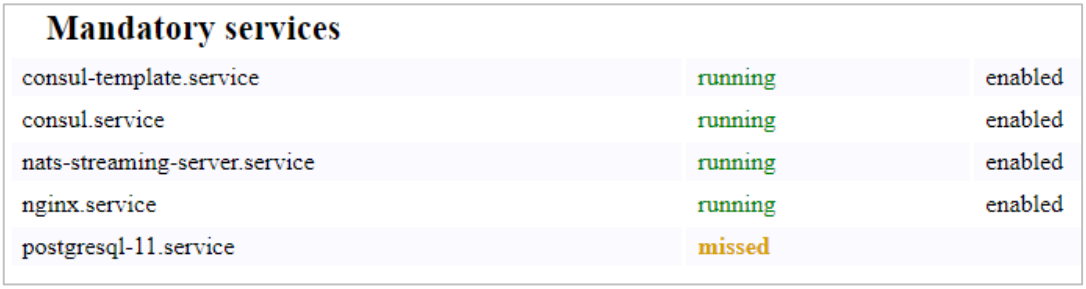
Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 21).

Таблица 21

Название столбца	Описание	Возможные значения (примеры значений)
Name	Название сервиса ППО	Возможные значения определяются перечнем сервисов ППО и имеют следующий формат `<имя группы сервисов>@<номер экземпляра сервиса в группе>.service`, например: `ocs-appstore-admin-api-gw@0.service`.
Unit Status	Информация о статусе сервиса	`active` - сервис запущен и выполняется; `activating` - сервис запускается; `deactivating` - сервис выключается; `inactive` – сервис выключен; `failed` - при запуске сервиса произошла ошибка
Unit-file status	Информация о присутствии конфигурационного файла запуска сервиса в автозапуске	`enabled` - присутствует в автозапуске; `disabled` - отсутствует в автозапуске

3.10.2.3.3 Mandatory services

Подраздел содержит информацию о статусе сервисов компонентов среды функционирования (Рисунок 21).



Mandatory services

consul-template.service	running	enabled
consul.service	running	enabled
nats-streaming-server.service	running	enabled
nginx.service	running	enabled
postgresql-11.service	missed	

Рисунок 21

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 22).

Таблица 22

Название столбца	Описание	Возможные значения (примеры значений)

Name	Название сервиса компонента среды функционирования	Возможные значения имеют следующий формат `<имя сервиса>.service` и определяются Разработчиком. Перечень возможных значений:
		`consul-template.service`
		`consul.service`
		`redpanda.service`
		`nginx.service`
		`postgresql-15.service`
		`postgresql.service`

Unit Status	Информация о статусе сервиса	`active` - сервис запущен и выполняется; `activating` - сервис запускается; `deactivating` - сервис выключается; `inactive` - сервис выключен; `failed` – при запуске сервиса произошла ошибка; `enabled` - сервис присутствует в автозапуске; `disabled` - сервис отсутствует в автозапуске; `missed` - компонент отсутствует
Unit-file status	Информация о присутствии конфигурационного файла запуска сервиса в автозапуске	`enabled` - присутствует в автозапуске; `disabled` - отсутствует в автозапуске

3.10.2.4. Раздел «API GW Service Status»

Раздел содержит информацию о статусе регистрации сервисов в системе обнаружения сервисов (Consul). На рисунке (Рисунок 22) приведен пример статуса регистрации сервисов в системе обнаружения сервисов.

API GW Services Status

Service name	Code	Status
ocs-appstore-admin-api-gw	200	passing
ocs-appstore-client-api-gw	200	passing
ocs-appstore-dev-api-gw	200	passing
ocs-auth-admin-api-gw	200	passing
ocs-auth-public-api-gw	200	passing
ocs-pkgrepo-device-api-gw	200	passing

Рисунок 22

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 23).

Таблица 23

Название столбца	Описание	Возможные значения (примеры значений)
Service name	Название сервиса ППО	Возможные значения определяются перечнем сервисов ППО
Code	Код http-ответа	Возможные значения определяются протоколом HTTP
Status	Информация о статусе регистрации сервиса в системе обнаружения сервисов (Consul)	Возможные значения определяются Consul. Статус «passing» означает, что проверка пройдена успешно

3.10.2.5. Раздел «Consul Cluster Endpoints Availability»

Раздел содержит информацию о проверке доступности интерфейсных функций системы обнаружения сервисов (Consul). На рисунке (Рисунок 23) приведен пример отображения информации о доступности интерфейсных функций системы обнаружения сервисов.

Consul Cluster Endpoints Availability

Node:Port	Availability
inp1int03.ompcloud:8300	OPENED
inp1int03.ompcloud:8301	OPENED
inp1int03.ompcloud:8302	OPENED
inp1int03.ompcloud:8500	OPENED
inp1int02.ompcloud:8300	OPENED
inp1int02.ompcloud:8301	OPENED
inp1int02.ompcloud:8302	OPENED
inp1int02.ompcloud:8500	OPENED

Рисунок 23

Перечень интерфейсных функций Consul приведен в документации на Consul (https://www.consul.io/docs/install/ports). Информация о доступности интерфейсных функций Consul предоставляется только в случае кластерной (многонодовой) конфигурации.

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 24).

Таблица 24

Название столбца	Описание	Возможные значения (примеры значений)
Node:Port	Адрес функции	Адрес функции представлен в следующем формате: `<имя хоста>:<порт>`. Проверка выполняется только для функций, доступных на следующих портах: 8300, 8301, 8302, 8500. Например: `acenter.example:8300`
Availability	Статус доступности функции	В случае доступности функции принимает значение «OPENED». В ином случае выводится код ошибки и сообщение, определяемое Consul

3.10.2.6. Раздел «Consul Service Health Check»

Раздел содержит информацию о статусе регистрации сервисов ППО в системе обнаружения сервисов Consul (Рисунок 24).

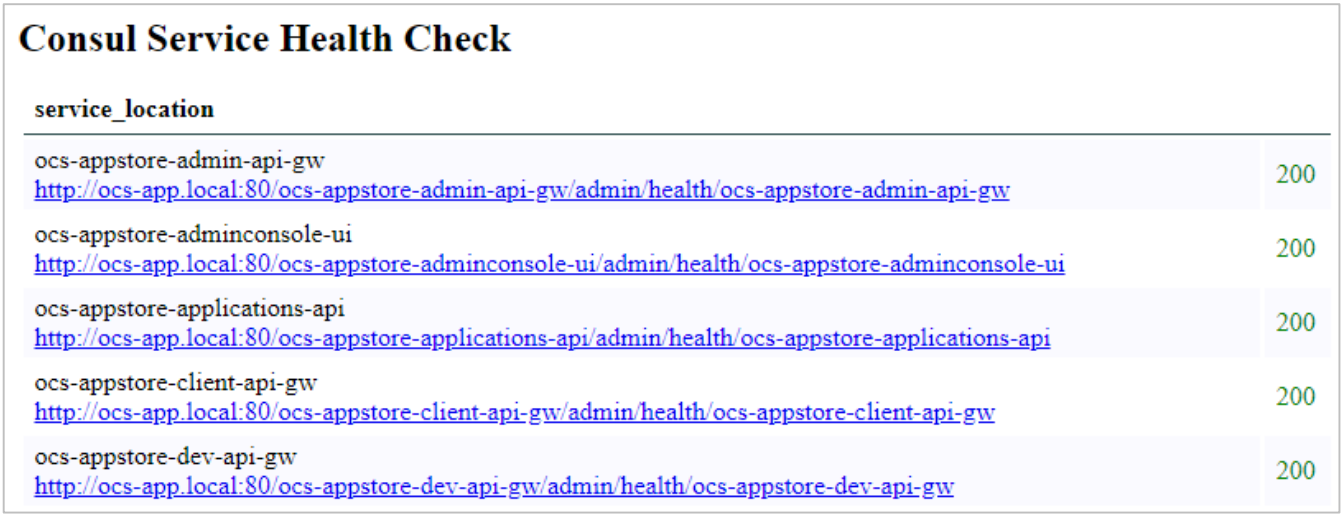


Рисунок 24

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 25).

Таблица 25

Название столбца	Описание	Возможные значения (примеры значений)
Первый столбец	Название сервиса ППО и URL-адрес функции (endpoint) сервиса «healthcheck»	Возможные значения определяются перечнем сервисов ППО
Второй столбец	Код http-ответа	Возможные значения определяются протоколом HTTP

Перечисленные заголовки `"service_location"`, `"expose_location"`, `"service_vhost"`, `"expose_port"`, `"static"` – это режимы работы `consul-template`.

3.10.2.7. Раздел «Cluster Nodes Reachability»

Раздел содержит информацию о результатах проверки доступности серверов (нод) кластера (Рисунок 25).

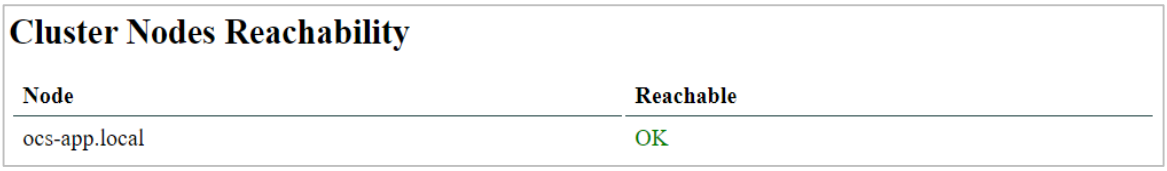


Рисунок 25

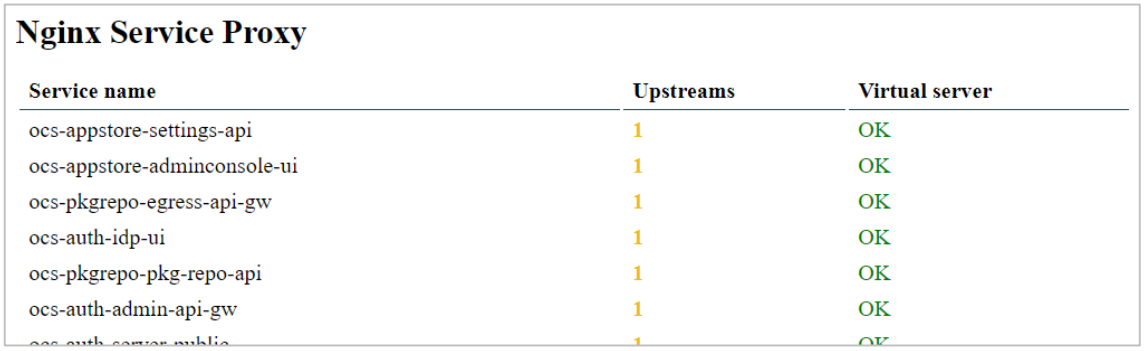
Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 26).

Таблица 26

Название столбца	Описание	Возможные значения (примеры значений)
Node	Адрес сервера (хоста)	Определяется доменными именами хостов
Reachable	Информация о доступности сервера	Может принимать значения: «OK» (в случае доступности) или содержать сообщение об ошибке, которое вернет утилита `ping`

3.10.2.8. Раздел «Nginx Service Proxy»

Раздел содержит информацию о проверке конфигурации балансировщика сервисов Nginx Web Server для каждого сервиса ППО (Рисунок 26).



Nginx Service Proxy

Service name	Upstreams	Virtual server
ocs-appstore-settings-api	1	OK
ocs-appstore-adminconsole-ui	1	OK
ocs-pkgrepo-egress-api-gw	1	OK
ocs-auth-idp-ui	1	OK
ocs-pkgrepo-pkg-repo-api	1	OK
ocs-auth-admin-api-gw	1	OK
ocs-auth-server-public	1	OK

Рисунок 26

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 27).

Таблица 27

Название столбца	Описание	Возможные значения (примеры значений)
Service name	Название сервиса ППО	Возможные значения определяются перечнем сервисов ППО
Upstreams	Количество экземпляров сервиса, заданных в конфигурационном файле Nginx	Целочисленные значения от 1 до n
Virtual server	Информация о наличие секции «server» для указанного сервиса ППО в конфигурационном файле Nginx. В данной секции заданы настройки «виртуального» сервиса ППО, который осуществляет перенаправление (проксирование) http-запросов на «реальные» экземпляры сервиса	«ОК» - секция `server` присутствует «No server block found!» - секция отсутствует

3.10.2.9. Раздел «Filestorage Configuration»

Раздел содержит информацию о конфигурации файловых хранилищ ПМ и ПООС (Рисунок 27).

Рисунок 27

Настройка «Filestorage location» содержит путь к каталогу и его статус.

В настройке «Configuration file» указан конфигурационный файл, в котором задан путь к файловому хранилищу.

3.11. Самостоятельная установка и настройка СУБД Postgres Pro и СУБД PostgreSQL 12/13/14/15

3.11.1. Установить на серверы БД необходимые пакеты согласно п. 3.9.7.

3.11.2. Установить и инициализировать СУБД.

При инициализации СУБД необходимо установить следующие значения параметров:

```
LC_COLLATE 'en_US.UTF-8'
LC_CTYPE 'en_US.UTF-8'
ENCODING UTF8
```

Установка и инициализация СУБД осуществляется в соответствии с ЭД на СУБД.

ПРИМЕЧАНИЕ. В рамках установки Postgres Pro обязательно должны быть установлены следующие пакеты:

- postgrespro-<std|ent>-<версия>;
- postgrespro-<std|ent>-<версия>-client;
- postgrespro-<std|ent>-<версия>-contrib;
- postgrespro-<std|ent>-<версия>-libs;
- postgrespro-<std|ent>-<версия>-server.

3.11.3. Создать суперпользователя с помощью скрипта samples/sql/create_superuser.sql, выполнив команду:

```
psql -U <пользователь, от имени которого выполняется команда> -h
<адрес хоста СУБД> -f create_superuser.sql -v login='<имя
суперпользователя>' -v pass='<пароль суперпользователя>' -v
expr='<срок действия учетной записи суперпользователя>'
```

Например:

```
psql -U postgres -h 192.168.137.15 -f create_superuser.sql -v
login='ompdbuser' -v pass='Admin123!' -v expr='10 years'
```

3.11.4. Назначить пароль для пользователя postgres с помощью следующих команд:

```
psql -U postgres
ALTER USER postgres with PASSWORD 'пароль';
exit
```

3.11.5. В конфигурационных файлах СУБД pg_hba.conf и postgresql.conf задать следующие параметры:

– тип соединения, диапазон IP-адресов клиентов БД;

– имя БД, имя пользователя;

– способ аутентификации клиентов;

– пароль пользователя СУБД в параметре pg_superuser_password.

3.11.6. Установить расширения pg_partman и pg_cron с помощью команд:

– ОС CentOS версии 7 и СУБД Postgres Pro 12:

```
sudo rpm -ivh pg_partman_12pro-std-4.6.0-1.el7.x86_64.rpm
sudo rpm -ivh pg_cron_12pro-1.5.2-1.el7.x86_64.rpm
```

– ОС CentOS версии 7 и СУБД Postgres Pro 14:

```
sudo rpm -ivh pg_partman_14pro-std-4.6.0-1.el7.x86_64.rpm
sudo rpm -ivh pg_cron_14pro-1.5.2-1.el7.x86_64.rpm
```

– ОС Альт и СУБД Postgres Pro 12:

```
sudo rpm -ivh pg_partman_12pro-std-4.6.0-1.alt.x86_64.rpm
sudo rpm -ivh pg_cron_12pro-1.5.2-1.alt.x86_64.rpm
```

– ОС Альт и СУБД Postgres Pro 13:

```
sudo rpm -ivh pg_partman_13pro-std-4.6.0-alt1.x86_64.rpm
sudo rpm -ivh pg_cron_13pro-1.5.2-alt1.x86_64.rpm
```

– ОС Альт и СУБД Postgres Pro 14 Cert:

```
sudo rpm -ivh pg_partman_14pro-std-4.6.0-alt1.x86_64.rpm
sudo rpm -ivh pg_cron_14pro-std-cert-1.5.2-alt1.x86_64.rpm
```

– ОС РЕД ОС версии 7.3 и СУБД Postgres Pro 13:

```
sudo rpm -ivh pg_partman_13pro-std-4.6.0-1.redos7.x86_64.rpm
sudo rpm -ivh pg_cron_13pro-std-1.5.2-1.redos7.x86_64.rpm
```

– ОС РЕД ОС версии 7.3 и СУБД Postgres Pro 14 Cert:

```
sudo rpm -ivh pg_partman_14pro-std-4.6.0-1.redos7.x86_64.rpm
sudo rpm -ivh pg_cron_14pro-std-cert-1.5.2-1.redos7.x86_64.rpm
```

– ОС РЕД ОС версии 7.3 и СУБД Postgres Pro 15:

```
sudo rpm -ivh pg_partman_15pro-std-4.7.4-1.redos7.x86_64.rpm
sudo rpm -ivh pg_cron_15pro-std-1.5.2-1.redos7.x86_64.rpm
```

– ОС РЕД ОС версии 7.3 и СУБД Postgres Pro 15 Cert:

```
sudo rpm -ivh pg_partman_15pro-cert-4.7.4-1.redos7.x86_64.rpm
sudo rpm -ivh pg_cron_15pro-cert-1.5.2-1.redos7.x86_64.rpm
```

– ОС Astra Linux SE 1.7 и СУБД Postgres Pro 14 Cert:

```
sudo dpkg -i pg-partman_4.6.0_std-14_smolensk.amd64.deb
sudo dpkg -i pg-cron_1.5.2_pro-std-cert-14_smolensk.amd64.deb
```

– ОС Astra Linux SE 1.7 и СУБД Postgres Pro 15 Cert:

```
sudo dpkg -i pg-partman_4.7.4_15_smolensk-pro-std-cert.amd64.deb
sudo dpkg -i pg-cron_1.5.2_15_smolensk-pro-std-cert.amd64.deb
```

– ОС CentOS версии 7 и СУБД PostgreSQL 12:

```
sudo rpm -ivh pg_partman_12-4.6.0-1.rhel7.x86_64.rpm
sudo rpm -ivh pg_cron_12-1.5.2-1.rhel7.x86_64.rpm
```

– ОС CentOS версии 7 и СУБД PostgreSQL 13:

```
sudo rpm -ivh pg_partman_13-4.6.0-1.rhel7.x86_64.rpm
sudo rpm -ivh pg_cron_13-1.5.2-1.rhel7.x86_64.rpm
```

– ОС CentOS версии 7 и СУБД PostgreSQL 14:

```
sudo rpm -ivh pg_partman_14-4.6.0-1.rhel7.x86_64.rpm
sudo rpm -ivh pg_cron_14-1.5.2-1.rhel7.x86_64.rpm
```

– ОС РЕД ОС и СУБД PostgreSQL 12:

```
sudo rpm -ivh pg_partman_12-4.6.0-1.redos7.x86_64.rpm
sudo rpm -ivh pg_cron_12-1.5.2-1.redos7.x86_64.rpm
```

– ОС РЕД ОС и СУБД PostgreSQL 13:

```
sudo rpm -ivh pg_partman_13-4.6.0-1.redos7.x86_64.rpm
sudo rpm -ivh pg_cron_13-1.5.2-1.redos7.x86_64.rpm
```

– ОС РЕД ОС и СУБД PostgreSQL 14:

```
sudo rpm -ivh pg_partman_14-4.6.0-1.redos7.x86_64.rpm
sudo rpm -ivh pg_cron_14-1.5.2-1.redos7.x86_64.rpm
```

– ОС РЕД ОС и СУБД PostgreSQL 15:

```
sudo rpm -ivh pg_partman_15-4.7.3-3.redos7.x86_64.rpm
sudo rpm -ivh pg_cron_15-1.5.2-1.redos7.x86_64.rpm
```

– ОС Astra Linux SE 1.7 и СУБД PostgreSQL 12:

```
sudo rpm -ivh pg-partman_4.6.0_12_smolensk.amd64.deb
sudo rpm -ivh pg-cron_1.5.2_12_smolensk.amd64.deb
```

– ОС Astra Linux SE 1.7 и СУБД PostgreSQL 13:

```
sudo rpm -ivh pg-partman_4.6.0_13_smolensk.amd64.deb
sudo rpm -ivh pg-cron_1.5.2_13_smolensk.amd64.deb
```

– ОС Astra Linux SE 1.7 и СУБД PostgreSQL 14:

```
sudo rpm -ivh pg-partman_4.6.0_14_smolensk.amd64.deb
sudo rpm -ivh pg-cron_1.5.2_14_smolensk.amd64.deb
```

– ОС Astra Linux SE 1.7 и СУБД PostgreSQL 15:

```
sudo rpm -ivh pg-partman_4.7.4_15_smolensk.amd64.deb
sudo rpm -ivh pg-cron_1.5.2_15_smolensk.amd64.deb
```

– ОС Альт Сервер 10.2 и СУБД PostgreSQL 12:

```
sudo rpm -ivh postgresql12-pg_partman-4.7.3-alt1.x86_64.rpm
sudo rpm -ivh postgresql12-pg_cron-1.5.2-1.alt.p10.x86_64.rpm
```

– ОС Альт Сервер 10.2 и СУБД PostgreSQL 13:

```
sudo rpm -ivh postgresql13-pg_partman-4.7.3-alt1.x86_64.rpm
sudo rpm -ivh postgresql13-pg_cron-1.5.2-1.alt.p10.x86_64.rpm
```

– ОС Альт Сервер 10.2 и СУБД PostgreSQL 14:

```
sudo rpm -ivh postgresql14-pg_partman-4.7.3-alt1.x86_64.rpm
sudo rpm -ivh postgresql14-pg_cron-1.5.2-1.alt.p10.x86_64.rpm
```

– ОС Альт Сервер 10.2 и СУБД PostgreSQL 15:

```
sudo rpm -ivh postgresql15-pg_partman-4.7.3-alt1.x86_64.rpm
sudo rpm -ivh postgresql15-pg_cron-1.5.2-1.alt.p10.x86_64.rpm
```

– ОС Альт 8 СП релиз 10 и СУБД PostgreSQL 15:

```
sudo rpm -ivh postgresql15-pg_partman-4.7.3-alt1.x86_64.rpm
sudo rpm -ivh postgresql15-pg_cron-1.5.2-1.alt.p10.x86_64.rpm
```

– ОС Ubuntu версии 20.04 и СУБД PostgreSQL 14:

```
sudo rpm -ivh postgresql-14-partman-4.7.4-1.focal.amd64.deb
sudo rpm -ivh postgresql-14-cron-1.5.2-1.focal.amd64.deb
```

– ОС Ubuntu версии 20.04 и СУБД PostgreSQL 15:

```
sudo rpm -ivh postgresql-15-partman-4.7.4-1.focal.amd64.deb
sudo rpm -ivh postgresql-15-cron-1.5.2-1.focal.amd64.deb
```

– ОС Ubuntu версии 22.04 и СУБД PostgreSQL 14:

```
sudo rpm -ivh postgresql-14-partman-4.7.4-1.jammy.amd64.deb
sudo rpm -ivh postgresql-14-cron-1.5.2-1.jammy.amd64.deb
```

– ОС Ubuntu версии 22.04 и СУБД PostgreSQL 15:

```
sudo rpm -ivh postgresql-15-partman-4.7.4-1.jammy.amd64.deb
sudo rpm -ivh postgresql-15-cron-1.5.2-1.jammy.amd64.deb
```

– ОС Debian версии 11 и СУБД PostgreSQL 14:

```
sudo rpm -ivh postgresql-14-partman-4.7.4-1.bullseye.amd64.deb
sudo rpm -ivh postgresql-14-cron-1.5.2-1.bullseye.amd64.deb
```

– ОС Debian версии 11 и СУБД PostgreSQL 15:

```
sudo rpm -ivh postgresql-15-partman-4.7.4-1.bullseye.amd64.deb
sudo rpm -ivh postgresql-15-cron-1.5.2-1.bullseye.amd64.deb
```

– ОС Debian версии 12 и СУБД PostgreSQL 14:

```
sudo rpm -ivh postgresql-14-partman-4.7.4-1.bookworm.amd64.deb
sudo rpm -ivh postgresql-14-cron-1.5.2-1.bookworm.amd64.deb
```

– ОС Debian версии 12 и СУБД PostgreSQL 15:

```
sudo rpm -ivh postgresql-15-partman-4.7.4-1.bookworm.amd64.deb
sudo rpm -ivh postgresql-15-cron-1.5.2-1.bookworm.amd64.deb
```

Указанные RPM-пакеты находятся на DVD с загрузочными модулями ППО в архиве `/server/install-infra.tar.gz/install-infra/binary/postgresql/`.

3.11.7. Перезапустить сервис СУБД в соответствии с документацией на СУБД.

4. УПРАВЛЕНИЕ КОМПОНЕНТАМИ СРЕДЫ ФУНКЦИОНИРОВАНИЯ, СЕРВИСАМИ, НАСТРОЙКАМИ СЕРВИСОВ И ПОДСИСТЕМ

4.1. Управление компонентами среды функционирования ППО

Управление компонентами среды фунционирования ППО заключается в их установке, обновлении и удалении и осуществляется с помощью скрипта `deploy-infra.sh` из каталога `install-<версия ППО>`, созданного на этапе развертывания управляющей ЭВМ (подраздел 3.3).

Формат команды управления сервисами имеет следующий вид:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh <параметры>
```

Описание параметров команды управления:

1) `<имя пользователя>`

В параметре указывается имя привилегированного sudo-пользователя, под которым настроен SSH доступ к серверам приложений, серверам БД и контент-серверам;

2) `-A, --action`

Данный параметр задает действие, которое необходимо выполнить, и может принимать следующие значения:

– параметр отсутствует – будет выполнена установка или обновление компонента (компонентов);

– `flush_all` – будет выполнено удаление компонента (компонентов);

3) `-c, --components`

Данный параметр задает компонент среды функционирования, для которого будет выполнена команда управления, и может принимать следующие значения:

– `dnsmasq`;

– `nginx`;

– `consul`;

– consul-template;

– redpanda;

– redis;

– ocs-user;

– db;

– syncthing.

В данном параметре может задаваться список подсистем, например:

```
--components dnsmasq, nginx
```

По умолчанию (если параметр на задан) команда управления будет применена ко всем компонентам;

4) -d, --database

Перечень допустимых значений параметра приведен в таблице (см. Таблица 13).

По умолчанию (если параметр не задан) будет использоваться значение, заданное в параметре pg_version конфигурационного файла config/vars/_vars.yml.

При отсутствии СУБД в перечне компонентов (параметр --components) значение данного параметра будет игнорироваться;

5) --skip-database

При наличии данного параметра СУБД не устанавливается;

6) -l, --limit

Данный параметр задает перечень хостов, для которых будет выполнена команда управления, например:

```
--limit example01.omp,example02.omp
```

По умолчанию (если параметр на задан) команда управления будет применена ко всем хостам согласно инвентарному файлу inventories/hosts.yml;

7) `-x, --extra-vars`

В данном параметре передаются внешние переменные для скриптов развертывания. В ППО используются следующие внешние переменные:

– `pg_uninstall_delete_data=true` - служит для удаления данных при удалении СУБД PostgreSQL;

8) `--force-infra-install`

Флаг служит для управления принудительной повторной установкой компонентов среды функционирования, в случаях, когда версия компонентов среды функционирования не изменилась и может принимать следующие значения:

– `false` - повторная установка компонентов той же версии выполнена не будет;

– `true` - будет выполнена повторная установка компонентов не зависимо от того изменилась версия или нет.

По умолчанию (если флаг не задан) флаг имеет значение `true`;

9) `-e, --env`

Данный параметр задает окружение, для которого необходимо выполнить настройку и установку компонентов среды функционирования ППО. Более подробная информация по настройке ППО для установки его на различные окружения приведена в п. 3.9.14.

10) `--help`

Вывод справочной информации.

Примеры команд управления:

1) Установка или обновление всех компонентов:

```
ANSIBLE_USER="omp" ./deploy-infra.sh --database 12
```

2) Установка или обновление Nginx на хосте `ocs-app.local`:

```
ANSIBLE_USER="omp" ./deploy-infra.sh --components nginx --limit ocs-app.local
```

3) Удаление Nginx:

```
ANSIBLE_USER="omp" ./deploy-infra.sh --components nginx --action
flush_all
```

4) Удаление СУБД PostgreSQL (с удалением данных):

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh  --components db --action flush_all --extra-vars "pg_uninstall_delete_data=true"
```

5) Получение справочной информации:

```
./deploy-infra.sh --help
```

4.2. Управление сервисами ППО

Управление сервисами ППО заключается в их установке, запуске, остановке, перезапуске, изменении настроек и осуществляется с помощью скрипта `deploy-ac.sh` из каталога `install-<версия ППО>`, созданного на этапе развертывания управляющей ЭВМ (подраздел 3.3).

Формат команды управления сервисами имеет следующий вид:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh <параметры>
```

Описание параметров команды управления:

1) `<имя пользователя>`

В параметре указывается имя привилегированного sudo-пользователя, под которым настроен SSH доступ к серверам приложений, серверам БД и контент-серверам;

2) `-s, --subsystems`

Данный параметр задает подсистему, для которой будет выполнена команда управления, и может принимать следующие значения:

- `auth` (для ПБ);
- `appstore` (для ПМ);
- `emm` (для ПУ);
- `mt` (для ПУТ);

– `push` (для ПСУ);

– `cdn` (для CDN);

– `pkgrepo` (для ПООС).

В данном параметре может задаваться список подсистем, например:

```
--subsystems auth,appstore,pkgrepo,emm,mt,cdn,push
```

По умолчанию параметр (если иное значение не задано) имеет значение:

```
--subsystems auth,appstore,pkgrepo,emm,mt,push
```

3) `-a, --apps`

Данный параметр задает перечень сервисов, для которых будет выполнена команда управления. Например:

```
--apps ocs-auth-adminconsole-ui,ocs-appstore-adminconsole-ui
```

Если необходимо выполнить команду сразу для всех сервисов подсистемы, потребуется перечислить через запятую все сервисы подсистемы либо задать значение параметра:

```
--apps all
```

По умолчанию параметр (если иное значение не задано) имеет значение `all`.

В случае если заданные в параметре `--apps` сервисы не соответствуют заданным в параметре `--subsystems` подсистемам, управляющая команда к таким сервисам применена не будет. При этом управление шлюзами доступа (сервисами шлюзов доступа) осуществляется в рамках той подсистемы, для которой они предназначены. Состав подсистем приведен в таблице (Таблица 28).

Таблица 28

Значение параметра «--subsystems»	Сервисы (значение параметра «--apps»)
ПБ	
auth	ocs-auth-admin-api-gw
	ocs-auth-public-api-gw
	ocs-auth-admin-cross-tenant-api-gw
	ocs-auth-server-public-proxy

Значение параметра «--subsystems»	Сервисы (значение параметра «--apps»)
	ocs-auth-idp-api
	ocs-auth-accounts-devices-api
	ocs-auth-accounts-users-api
	ocs-auth-server-admin
	ocs-auth-server-public
	ocs-auth-audit-api
	ocs-auth-subsystems-api
	ocs-auth-config-api
	ocs-auth-adminconsole-ui
	ocs-auth-idp-ui
ПМ	
appstore	ocs-appstore-applications-api
	ocs-appstore-settings-api
	ocs-appstore-adminconsole-ui
	ocs-appstore-devconsole-ui
	ocs-appstore-admin-api-gw
	ocs-appstore-client-api-gw
	ocs-appstore-dev-api-gw
	ocs-appstore-egress-api-gw
ПУ	
emm	ocs-emm-applications-api
	ocs-emm-dispatcher-api
	ocs-emm-devices-api
	ocs-emm-state-manager-api
	ocs-emm-enrollments-api
	ocs-emm-policies-api
	ocs-emm-reports-api
	ocs-emm-users-api
	ocs-emm-journal-api
	ocs-emm-jobs-api
	ocs-emm-locations-api
	ocs-emm-admin-api-gw
	ocs-emm-device-api-gw
	ocs-emm-egress-api-gw
ПУТ	
mt	ocs-mt-tenants-api
	ocs-mt-organizations-api
	ocs-mt-admin-api-gw
	ocs-mt-egress-api-gw
ПСУ	
push	ocs-push-main-api
	ocs-push-transport
	ocs-push-admin-api-gw
	ocs-push-public-api-gw
	ocs-push-egress-api-gw

Значение параметра «--subsystems»	Сервисы (значение параметра «--apps»)
ПООС	
pkgrepo	ocs-pkgrepo-pkg-repo-api
	ocs-pkgrepo-device-api-gw
	ocs-pkgrepo-admin-api-gw
	ocs-pkgrepo-egress-api-gw
CDN	
cdn	ocs-cdn-admin-api-gw
	ocs-cdn-mobile-api-gw

4) `-A, --action`

Данный параметр задает действие, которое необходимо выполнить. Перечень допустимых действий и соответствующие им значения параметра приведены в таблице (Таблица 29).

Таблица 29

Значение параметра «--action»	Действие
deploy	Установка
start	Запуск
stop	Остановка
restart	Перезапуск
config	Изменение настроек (переустановка конфигурационного файла)
flush_all	Удаление

По умолчанию параметр (если не задано иное значение) имеет значение `deploy`.

ВНИМАНИЕ! Установка подсистем ППО должна осуществляться строго в следующей последовательности: ПБ, ПМ, ПООС, ПУ, ПУТ, CDN, ПСУ;

5) `-C, --clients`

Данный параметр задает OIDC клиентов, для которых будет выполнена команда управления. Например:

```
--clients auth-admin-console, aps-admin-console
```

При необходимости выполнить команду сразу для всех OIDC клиентов потребуется перечислить через запятую все OIDC клиенты либо задать значение параметра:

```
--clients all
```

По умолчанию параметр (если не задано иное значение) имеет значение `all`;

6) `-d, --database`

Данный параметр задает СУБД, которая установлена на сервере БД. Перечень допустимых значений параметра приведен в таблице (см. Таблица 13).

Например:

```
--database 12
```

По умолчанию (если параметр не задан) будет использоваться значение, заданное в параметре `pg_version` конфигурационного файла `config/vars/_vars.yml`;

7) `-e, --env`

Данный параметр задает окружение, для которого необходимо выполнить настройку и установку ППО. Более подробная информация по настройке ППО для установки его на различные окружения приведена в п. 3.9.14.

8) `--help`

Вывод справочной информации.

Примеры команд управления:

1) Остановка всех сервисов ПМ:

```
ANSIBLE_USER="omp" ./deploy-ac.sh --action stop
```

2) Запуск сервисов `ocs-appstore-applications-api` и `ocs-appstore-adminconsole` ПМ:

```
ANSIBLE_USER="omp" ./deploy-ac.sh --subsystems appstore --apps ocs-appstore-applications-api,ocs-appstore-adminconsole --action start
```

3) Получение справочной информации:

```
./deploy-ac.sh --help
```

4.3. Управление настройками сервисов и подсистем ППО

Управление настройками сервисов и подсистем ППО может осуществляться 2 способами.

4.3.1. Способ 1 (рекомендуемый)

4.3.1.1. Задать требуемые значения параметров в конфигурационных файлах сценариев установки ППО и подсистем ППО.

4.3.1.2. Переустановить конфигурационные файлы с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --action config
```

Подробное описание параметров запуска скрипта `deploy-ac.sh` приведено в подразделе 4.2.

4.3.2. Способ 2

4.3.2.1. Задать требуемые значения параметров в конфигурационных файлах сервисов и подсистем ППО. Описание параметров конфигурационных файлов сценариев установки подсистем ППО приведено в разделе 12.

4.3.2.2. Перезапустить требуемые сервисы с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --subsystems
<идентификатор подсистемы> --apps <перечень сервисов подсистемы> --
action restart
```

Подробное описание параметров запуска скрипта `deploy-ac.sh` приведено в подразделе 4.2.

5. РЕЗЕРВНОЕ КОПИРОВАНИЕ

ВНИМАНИЕ! Приведенные ниже имена файлов и каталогов характерны для типового варианта установки ППО и среды функционирования ППО.

5.1. Резервное копирование после установки (обновления) ППО

После успешной установки (обновления) ППО необходимо создать резервную копию каталога `install-<версия ППО>/install-ac/` (или `install-<версия ППО>/install-ac-mt/` или `/install-<версия ППО>/install-ac-mt-spr/`).

5.2. Периодическое резервное копирование и резервное копирование перед установкой обновлений

Периодичность резервного копирования определяется регламентами эксплуатирующей организации.

Периодическое резервное копирование и резервное копирование перед установкой обновлений выполняется в приведенной ниже последовательности.

Подробная информация об особенностях резервного копирования ППО приведена в документе «Рекомендации по резервному копированию».

6. ОБНОВЛЕНИЕ ППО И ОС АВРОРА

6.1. Порядок обновления

Обновление ППО до требуемой версии возможно только с версий, указанных в таблице (Таблица 30).

Таблица 30

Требуемая версия	Ранее установленная версия
2.2.0	2.1.3*
2.2.1*	2.1.3*, 2.2.0
2.2.2*	2.1.3*, 2.2.0, 2.2.1*
2.3.0	2.2.2*
2.4.0	2.2.2*, 2.3.0
2.5.0	2.2.2*, 2.3.0, 2.4.0
2.5.1*	2.2.2*, 2.3.0, 2.4.0, 2.5.0
3.0.0	2.5.1*
3.0.1	2.5.1*, 3.0.0
3.1.0	2.5.1*, 3.0.1
3.1.1*	2.5.1*, 3.0.1, 3.1.0
3.1.2*	2.5.1*, 3.1.0, 3.1.1*
3.2.0	3.1.0, 3.1.2*
4.0.0*	3.1.2*, 3.2.0
4.1.0*	3.1.2*, 4.0.0*
5.0.0*	3.2.0, 4.0.0*, 4.1.0*

* - версии ППО, прошедшие сертификацию в ФСТЭК России.

6.2. Обновление сервера приложений ППО

ВНИМАНИЕ! Для установки обновления ППО количество свободного места на жестком диске сервера БД ПБ должно быть не меньше, чем размер самой БД ПБ. При недостаточном количестве свободного места на жестком диске его необходимо увеличить. Продолжительность процесса обновления ППО зависит от размера БД и может занять длительное время.

Для обновления сервера приложений ППО необходимо выполнить описанные ниже действия.

6.2.1. Создать резервную копию данных, ППО и компонентов среды функционирования в соответствии с разделом 5.

6.2.2. Скопировать на управляющую ЭВМ архив с новой версией ППО и распаковать его в соответствии с п. 3.3.4 - 3.3.7.

6.2.3. Обновить на управляющей ЭВМ пакеты в соответствии с п. 3.3.9.

6.2.4. Настроить компоненты среды функционирования ППО и ППО в соответствии с подразделом 3.4.

6.2.5. Обновление СУБД PostgreSQL 12/13/14/15 до новой старшей версии[12] (`major version`) необходимо осуществлять в соответствии с ЭД на СУБД. Перечень поддерживаемых версий СУБД приведен в п. 1.4.2.

6.2.6. Установить компоненты среды функционирования в соответствии с п. 3.5.1.

6.2.7. Установить ППО в соответствии с п. 3.5.2.

6.2.8. Перезапустить сервис `ocs-pkgrepo-pkg-repo-api` с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --subsystems pkgrepo
--apps ocs-pkgrepo-pkg-repo-api --action restart
```

6.2.9. После обновления ППО с версии 4.0.0 (или ниже) до версии 4.1.0 (или выше) необходимо удалить Nats Streaming Server с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh -A flush_all -i
inventories/hosts.yml -c nats-streaming-server
```

6.2.10. Оповестить пользователей ППО о необходимости очистить кэш и cookies веб-браузера. Иначе при открытии интерфейса ППО будет ошибка HTTP ERROR 400.

[12] Согласно спецификации SemVer 2.0.0.

6.3. Обновление ОС Аврора с помощью ПУ

Обновление ОС Аврора выполняется в следующей последовательности:

6.3.1. Обновить сервер приложений ППО в соответствии с подразделом 6.1.

6.3.2. Загрузить в файловое хранилище ПООС пакеты требуемой версии ОС в соответствии с п. 3.7.3.

6.3.3. Обновить ОС Аврора до требуемой версии на тестовой группе устройств с целью проверки корректности обновления с помощью политики «Приложения/Установить версию ОС». Порядок работы с политиками и группами устройств приведен в документе «Руководство пользователя. Часть 3. Подсистема Платформа управления» АДМГ.20134-01 90 01-3.

6.3.4. Убедиться, что после окончания, заданного в правиле временного интервала обновления в карточке каждого устройства из тестовой группы отображается требуемая версия ОС Аврора.

6.3.5. Обновить приложения ППО на тестовой группе устройств в соответствии с документом «Руководство пользователя. Часть 7. Приложение «Аврора Центр» для операционной системы Аврора» АДМГ.20134-01 90 01-7.

6.3.6. Выполнить обновление аналогичным образом для остальных устройств после успешного обновления ОС Аврора и приложений на тестовой группе устройств.

7. УДАЛЕНИЕ ППО

Для удаления ППО необходимо выполнить следующие действия:

7.1. Перейти в каталог со сценариями установки ППО.

7.2. Удалить ППО с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --action flush_all
```

Подробное описание параметров запуска скрипта `deploy-ac.sh` приведено в подразделе 4.2.

7.3. Удалить компоненты среды функционирования ППО с помощью команды:

– без удаления данных, хранящихся в СУБД PostgreSQL:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh --action flush_all
```

– с удалением данных, хранящихся в СУБД PostgreSQL:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh --action flush_all
--extra-vars "pg_uninstall_delete_data=true"
```

Подробное описание параметров запуска скрипта `deploy-infra.sh` приведено в подразделе 4.1.

8. ВАРИАНТЫ УСТАНОВКИ ПСУ

8.1. Установка ПСУ на один сервер (хост) с другими подсистемами ППО

Данный вариант установки ППО осуществляется по умолчанию.

8.2. Установка ПСУ на отдельный сервер (хост)

Задание адресов серверов (имен хостов), на которые будут установлены подсистемы ППО, осуществляется на этапе настройки ППО (пп. 3.4.2.1). Для того, чтобы установить ПСУ на отдельный сервер, необходимо в инвентарном файле `inventories/hosts.yml` задать адрес сервера (имя хоста), на который необходимо установить ПСУ (`push`), например:

```
...
        app:
          hosts:
            ocs-app.local:
              subsystems: auth,appstore,emm,mt,pkgrepo,
            acenterapp02:
              ocs-push.local: push
```

Описание порядка задания адресов в инвентарном файле `inventories/hosts.yml` приведено в п. 3.9.12.

8.3. Отдельная установка ПСУ (установка ПБ и ПСУ)

Для отдельной установки ПСУ необходимо выполнить следующую последовательность действий:

8.3.1. Выполнить последовательность действий, предусмотренную п. 3.2 - 3.5.1.

8.3.2. Установить ПБ и ПСУ с помощью команды:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --subsystems
auth,push
```

8.3.3. Выполнить последовательность действий, предусмотренную п. 3.5.3 - 3.5.5.

9. ВАРИАНТЫ УСТАНОВКИ СУБД

9.1. Некластерная (standalone) установка СУБД

В данном варианте устанавливается только один сервер БД. Применяется такая установка в однонодовой конфигурации ППО (п 2.3.1 и 2.3.2).

Для того, чтобы выполнить `standalone` установку сервера БД достаточно в инвентарном файле `inventories/hosts.yml` в процессе настройки компонентов среды функционирования ППО (п. 3.4.1) задать адрес хоста, на который будет установлен сервер БД, например:

```
...
        postgresql:
          hosts:
            ocs-db.local:
```

9.2. Установка СУБД в кластерной конфигурации

Кластерная установка сервера БД применяется в катастрофоустойчивой кластерной конфигурации ППО и предполагает установку кластеров СУБД в нескольких ЦОДах (п. 2.3.6).

ВНИМАНИЕ! При установке ППО в кластерной конфигурации не допускается устанавливать серверы приложений ППО и серверы БД на одном хосте.

Для установки СУБД в кластерной конфигурации необходимо в процессе настройки компонентов среды функционирования ППО (п. 3.4.1) выполнить следующие настройки:

9.2.1. В каталоге `inventories` со сценариями установки ППО создать окружения для основного (primary) и резервного (standby) кластеров с помощью следующих команд:

```
mkdir -p inventories/<название окружения>
cp inventories/hosts.yml inventories/<название окружения>/
```

Например:

```
mkdir -p inventories/ocs-primary
cp inventories/hosts.yml inventories/ocs-primary/
mkdir -p inventories/ocs-standby
cp inventories/hosts.yml inventories/ocs-standby/
```

Подробная информация о настройки ППО для установки его на различные окружения приведена в п. 3.9.14.

9.2.2. Выполнить настройку инвентарного файла `hosts.yml` для основного кластера

Для этого в инвентарном файле `inventories/<название окружения>/hosts.yml` необходимо задать следующие параметры:

– `patroni_cluster` – название кластера. Название кластера должно иметь следующий формат `<название кластера>-<тип кластера>`. Тип кластера должен быть `primary` (основной). Например:

```
...
        patroni_cluster: ocs-primary
```

– `keepalived_cluster_vip` - виртуальный IP-адрес (virtual IP), который будет присвоен активному серверу БД в основном кластере, например:

```
...
        keepalived_cluster_vip:
          ocs-primary: 192.168.1.60
```

ПРИМЕЧАНИЕ. Виртуальный IP-адрес используется для подключения приложений (клиентов) к активному серверу БД.

– адреса хостов, на которые будут установлены серверы БД, например:

```
...
        postgresql:
          hosts:
            acenterdb01:
            acenterdb02:
```

– адреса хостов, на которые будут установлены агенты Consul. Т.к. агенты Consul устанавливаются на серверы БД, поэтому вместо хостов достаточно указать группу `postgresql`:

```
...
            consul_agents:
              children:
                postgresql:
```

ВНИМАНИЕ! Не допускает установка агента и сервера Consul на одном хосте.

Пример инвентарного файла для основного кластера приведен в файле `samples/ac/inventories/hosts-patroni-two-dbnode-primary.yml`.

9.2.3. Выполнить настройку инвентарного файла `hosts.yml` для резервного (standby) кластера

Для этого в инвентарном файле `inventories/<название окружения>/hosts.yml` необходимо задать следующие параметры:

– `patroni_cluster` – название кластера. Название кластера должно иметь следующий формат `<название основного кластера>-<тип кластера>`. Тип кластера должен быть `standby` (резервный). Например:

```
...
        patroni_cluster: ocs-standby
```

– `keepalived_cluster_vip` - виртуальные IP-адреса (virtual IP) для активного сервера БД в основном кластере и для StandBy Leader сервера БД в резервном кластере, например:

```
...
        keepalived_cluster_vip:
          ocs-primary: 192.168.1.60
          ocs-standby: 192.168.1.61
```

– адреса хостов, на которые будут установлены серверы БД резервного кластера, например:

```
...
        postgresql:
          hosts:
            acenterdb11:
            acenterdb12:
```

– адреса хостов, на которые будут установлены агенты Consul. Т.к. агенты Consul устанавливаются на серверы БД, поэтому вместо хостов достаточно указать группу `postgresql`:

```
...
            consul_agents:
              children:
                postgresql:
```

ВНИМАНИЕ! Не допускает установка агента и сервера Consul на одном хосте.

Пример инвентарного файла для резервного кластера приведен в файле `samples/ac/inventories/hosts-patroni-two-dbnode-standby.yml`.

9.2.4. Выполнить другие настройки ППО и компонентов среды функционирования ППО в соответствии с пподразделе. 3.4 .

9.2.5. Выполнить установку компонентов среды функционирования ППО и ППО в соответствии с подразделом 3.5 для каждого кластера, указав в командах установки путь к инвентарному файлу или название окружения.

Примеры команд:

– команда установки всех пакетов на серверы приложений и серверы БД:

```
ansible-playbook -i inventories/<название окружения>/hosts.yml play-
managed-node-prerequisites.yml -vv -u <имя пользователя>
```

– команда установки компонентов среды функционирования ППО:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh --env "<название
окружения>"
```

– команда установки ППО:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-ac.sh --env "<название
окружения>"
```

10. УСТАНОВКА ППО В KUBERNETES

ВНИМАНИЕ! Данный функционал является экспериментальным и может содержать ошибки. В случае возникновения ошибок необходимо обратиться в службу технической поддержки предприятия-изготовителя.

10.1. Порядок развертывания и настройки сервера приложений

В таблице (Таблица 31) приведены программно-аппаратные характеристики сервера приложений.

Таблица 31

Параметр	Значение	
	Testing	Production
Процессор	2 ядра	5 ядер
Объем оперативной памяти	5 Гб	6 Гб
Объем жесткого диска	40 Гб	40 Гб
ОС (рекомендуемая)	Ubuntu 22.04	
ПО для управления контейнерами	Kubernetes версии 1.25 или выше, либо сборка от Canonical microk8s	

Для развертывания и установки сервера приложений необходимо выполнить следующие действия:

10.1.1. Установить на сервер приложений необходимые пакеты с помощью команды:

```
apt update && apt -y install ansible python3-pip curl snapd
pip3 install kubernetes
snap install microk8s –classic
```

10.1.2. Для работы сценариев установки ППО необходимо установить в ansible-коллекции плагины kubernetes.core и community.general с помощью команды:

```
ansible-galaxy install --role-file requirements-k8s.yml --ignore-
errors –force
```

10.1.3. Включить расширение `hostpath-storage`, необходимое для сохранения данных stateful-приложений (приложений, которым требуется сохранять данные), при перезагрузке контейнеров с помощью команды:

```
microk8s enable hostpath-storage
```

10.1.4. Включить расширение `metalib`, необходимое для доступа к сервисам ППО (порты 80 и 8009) снаружи кластера, с помощью команды:

```
microk8s enable metallb:<диапазон IP-адресов подсети сервера приложений>
```

Например:

```
microk8s enable metallb: 10.188.25.196-10.188.25.197
```

ПРИМЕЧАНИЕ. Число IP-адресов в диапазоне может быть не менее двух штук.

10.1.5. Убедиться, что расширения `microk8s` и `metallb` успешно включены и находятся в списке доступных (`enabled`) расширений с помощью команды:

```
microk8s status
```

10.1.6. Для удобства работы с Kubernetes рекомендуется установить консольный клиент `k9s`, а также `kubectl` с помощью команд:

```
wget
https://github.com/derailed/k9s/releases/download/v0.32.3/k9s_linux_amd64.deb
dpkg -i k9s_linux_amd64.deb
curl -LO https://dl.k8s.io/release/`curl -LS
https://dl.k8s.io/release/stable.txt`/bin/linux/amd64/kubectl
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin/kubectl
```

10.1.7. Для работы из-под непривилегированного пользователя необходимо дополнительно выполнить следующие команды:

```
USER=<имя пользователя>
sudo usermod -a -G microk8s $USER
sudo mkdir -p ~/.kube
sudo chown -f -R $USER ~/.kube
```

Далее, перезайти в учетную запись и после этого выполнить команду:

```
microk8s config > ~/.kube/config
```

10.2. Порядок установки ППО в Kubernetes

10.2.1. Создать на сервере приложений отдельный каталог, скопировать в него содержимое DVD с ППО и перейти в созданный каталог с помощью команды:

```
cd <путь к каталогу>
```

10.2.2. Перейти в каталог `/server` с помощью команды:

```
cd <путь к каталогу server>
```

10.2.3. Запустить `installer-ac.sh` (или `installer-ac-mt.sh` или `installer-ac-mt-spr.sh`[13]) с помощью команды:

```
bash installer-ac.sh
или
bash installer-ac-mt.sh
или
bash installer-ac-mt-spr.sh
```

10.2.4. Ознакомиться с «Лицензионным соглашением» и принять его в соответствии с п. 3.3.7.

10.2.5. Перейти в каталог со сценариями установки (каталог: `/install-<версия ППО>/install-ac/` или `/install-<версия ППО>/install-ac-mt/` или `/install-<версия ППО>/install-ac-mt-spr/`).

10.2.6. Сформировать конфигурационные файлы ППО

Для этого необходимо придумать название окружения (например, `k8s`) и запустить скрипт генерации конфигурационных файлов выполнить команду:

```
./deploy-k8s.sh -e <название окружения> -A configs
```

В процесс выполнения скрипта необходимо в интерактивном режиме задать следующие параметры:

[13] Название файла зависит от варианта поставки ППО.

– deployment mode – режим установки (testing или production). Режим testing рекомендуется использовать в ознакомительных целях. При использовании режима testing: все компоненты среды функционирования устанавливаются в одном экземпляре без резервирования, отсутствует необходимость в настройки внешнего балансировщика (все устанавливается на ingress-контроллере Kubernetes), контейнерам меньше выделяется ресурсов процессора и оперативной памяти.

– external ip – внешний IP-адрес, по которому будет доступно ППО (выбирается как любой свободный адрес из диапазона, который указывался при установке расширения metalib. Посмотреть настроенный диапазон IP-адресов можно в параметре .spec.addresses, который можно получить с помощью команды:

```
kubectl get ipaddresspool -A -o yaml
```

– DNS domain - внешний домен, который вместе с названием окружения будет формировать URL-адрес ППО http(s)://<название окружения>.<DNS domain>;

– cluster name – имя кластера Kubernetes, в который будет установлено ППО. Кластер предварительно должен быть создан. Посмотреть список кластеров можно с помощью команды:

```
kubectl config get-clusters
```

– storage class – название класса хранилища Kubernetes (StorageClass), с которым будет создаваться тома хранения данных (volumes) для stateful-приложений. Посмотреть список классов хранилищ можно с помощью команды:

```
kubectl get storageclasses
```

– public key – открытый ssh-ключ пользователя ОС, из-под которого запускается установка ППО. При необходимости ключевую пару можно сформировать с помощью команды:

```
ssh-keygen -t rsa -b 4096
```

– `nfs server` – NFS сервер, используемый для хранения файлов приложений (иконки, скриншоты, RPM-пакеты) и пакетов ОС. Если предполагается использовать внешний сервер, то необходимо указать его IP-адрес (до установки на нем уже должны быть созданы каталоги согласно подразделу 3.8). Если готового NFS сервера нет, то необходимо оставить это поле пустым, в данном случае NFS сервер будет установлен автоматически внутри Kubernetes.

Пример задания параметров:

```
./deploy-k8s.sh -e k8s -A configs


Please answer a few questions for creating k8s stage configs and
inventories -

Choose Aurora Center deployment mode [testing/production]: testing

Aurora Center external IP: 10.188.25.196

Aurora Center external DNS domain: mydomain.ru

K8s cluster name: microk8s-cluster

K8s storage class name: microk8s-hostpath

SSH public key, that will be used for connections to the Management
Server:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQC5AOA8ANTqrVxIcCPrgkuXMXOIOy27Z629IKTM+D
xJTwjEj1n78a5oxr9P/rdGOSfNwFQRZ3pBeINCiYWB8OqvLmEOhCBQQ498EI64DnC7BepS
4tSQyiPnIaprChQBD9VBx4leMUMhoRl01tCHe0cqCBtjvRgp/3ZJScPLMCStU29cJ31OnP
P29+JduN9FRO5fl7/MG1h4MetWFcYc6D2eQJ78pAJ0+1COKEAzpzLRW4Qdw4mkI+ukOob2
uunEjOeows2yCumXcKiNpOcp0zV+Kt6y2o72rVWrPXN1Zs8wya7dmCULmjKzkOPl8GvA7f
fHByWXBBnvc9P7P/86wVwecLrjubmdNTAsD0NO+pd1vuuH89Pz8vQxjHEaNPho/Q38RhUX
5ElP4jsNIUVOsgVwdfnJPCdyL7KaI+GOd332YEB8tPHuUbCX+wOKCO10qzkwUPMFXTNh6v
HhO4OPthwbCEgZv5jgraW5D5JqcJXiEtSZZSMo9WUXaBsNbOtlGts= root@ubuntu22

NFS server for shared storage(ip/hostname or leave empty to deploy in-
cluster nfs server):
```

В результате выполнения скрипта, будут сформированы инвентарный файл окружения `inventories/<название окружения>/hosts.yml`, а также конфигурационные файлы в `config/environments/<название окружения>/`. Подробная информация о конфигурационных файлах окружений приведена в п. 12.2.8. При необходимости изменения в данные файлы можно внести вручную.

10.2.7. Выполнить установку ППО с помощью команды

```
bash deploy-k8s.sh -e k8s
```

В случае успешной установки будет выведено сообщение следующего вида:

```
Congratulations, all done!

You can access AC from the following URLs -
  UI:                http://k8s.mydomain.ru/auth/admin/
  management-server: ssh ocs@k8s.mydomain.ru

!Warning!
check that k8s.mydomain.ru has a correct DNS record
  k8s.mydomain.ru A 10.188.25.196
or alternatively add to yor local /etc/hosts file
  10.188.25.196 k8s.mydomain.ru
```

Первоначальный вход в ППО осуществляется с помощью Консоли администратора ПБ и предустановленной учетной записи в соответствии с подразделом 3.6.

10.3. Порядок удаления ППО из Kubernetes

Удаление ППО из Kubernetes осуществляется с помощью следующих команд:

```
bash deploy-k8s.sh -e k8s -A delete
kubectl -n k8s delete pvc --all
kubectl delete ns k8s
```

11. КОНФИГУРАЦИОННЫЕ ФАЙЛЫ СЦЕНАРИЕВ УСТАНОВКИ СРЕДЫ ФУНКЦИОНИРОВАНИЯ

11.1. Конфигурационные файлы сценариев установки среды функционирования

11.1.1. Инвентарный файл inventories/hosts.yml

В инвентарном файле `inventories/hosts.yml` содержатся адреса серверов (имена хостов), на которые установлены (будут установлены) компоненты среды функционирования ППО и подсистемы ППО. Описание секций инвентарного файла `inventories/hosts.yml` приведено в таблице (Таблица 32).

Таблица 32

Секция конфигурационного файла	Описание
`all.children.ocs.vars.patroni_cluster`	Название кластера СУБД Postgres
`all.children.ocs.vars.patroni_cluster.keepalived_cluster_vip`	Виртуальные IP-адреса (virtual IP), которые будут присвоены активному серверу БД в основном кластере и StandBy Leader серверу БД в резервном кластере
`all.children.ocs.children.app`	Сервера приложений ППО
`all.children.ocs.children.content`	Контент-сервера
`all.children.ocs.children.postgresql.hosts`	СУБД Postgres
`all.children.ocs.children.nginx`	Балансировщик сервисов «Nginx Web Server»
`all.children.ocs.children.consul`	Система обнаружения сервисов «Consul» сервера приложений
`all.children.ocs.children.consul.children.consul_servers`	Сервера системы обнаружения сервисов «Consul»
`all.children.ocs.children.consul.children.consul_agents`	Агенты системы обнаружения сервисов «Consul»
`all.children.ocs.children.consul_content`	Система обнаружения сервисов «Consul» контент-сервера
`all.children.ocs.children.consul_template`	Средство управления конфигурациями сервисов «Consul Template»

Секция конфигурационного файла	Описание
`all.children.ocs.vars.patroni_cluster`	Название кластера СУБД Postgres
`all.children.ocs.vars.patroni_cluster.keepalived_cluster_vip`	Виртуальные IP-адреса (virtual IP), которые будут присвоены активному серверу БД в основном кластере и StandBy Leader серверу БД в резервном кластере
`all.children.ocs.children.app`	Сервера приложений ППО
`all.children.ocs.children.content`	Контент-сервера
`all.children.ocs.children.postgresql.hosts`	СУБД Postgres
`all.children.ocs.children.nginx`	Балансировщик сервисов «Nginx Web Server»
`all.children.ocs.children.nats_streaming_server`	Сервис гарантированной доставки сообщений NATS Streaming Server
`all.children.ocs.children.redpanda`	Сервис гарантированной доставки сообщений «Redpanda»
`all.children.ocs.children.redis.children.redis_masters`	СУБД Redis для хранения сессий
`all.children.ocs.children.redis.children.sentinel`	Redis Sentinel обеспечивает высокую доступность СУБД Redis
`all.children.ocs.children.syncthing`	Приложение для синхронизации файлов Syncthing

Файл сценария установки для установки среды функционирования ППО на 1 сервере с доменным именем `ocs-app.local` имеет следующий вид:

```
---
all:
  children:
    ocs:
      vars:
        # patroni_cluster: ocs-primary
        # keepalived_cluster_vip:
        #   ocs-primary: X.X.X.X
      children:
        app:
          hosts:
            ocs-app.local:
        content:
          hosts:
        postgresql:
          hosts:
            ocs-app.local:
        nginx:
```

```
          children:
            app:
            content:
        consul:
          children:
            consul_servers:
              children:
                app:
            consul_agents:
              children:
                # postgresql:
        consul_content:
          children:
            content:
        consul_template:
          children:
            app:
            content:
        nats_streaming_server:
          children:
            app:
        redpanda:
          children:
            app:
        redis:
          children:
            redis_masters:
              children:
                app:
            sentinel:
              children:
                app:
              hosts:
        syncthing:
          children:
            app:
```

11.1.2. Настройки сценариев установки среды функционирования ППО в конфигурационных файлах config/vars/_vars.yml и config/subsystems/<название подсистемы>/vars/_vars.yml

В данных конфигурационных файлах задаются настройки следующих компонентов среды функционирования ППО: Redpanda, Consul, СУБД Redis и СУБД PostgreSQL. Конфигурационные файлы `_vars.yml` используются только в процессе установки.

Описание параметров конфигурационных файлов `_vars.yml` приведено в конфигурационных файлах в виде комментариев.

11.1.3. Настройки паролей и секретов компонентов среды функционирования в конфигурационных файлах config/secret.yml и config/subsystems/<название подсистемы>/secret.yml

В данных конфигурационных файлах задаются пароли и секреты следующих компонентов среды функционирования ППО: Redpanda, Consul, СУБД Redis и СУБД PostgreSQL.

12. КОНФИГУРАЦИОННЫЕ ФАЙЛЫ ППО (СЦЕНАРИЕВ УСТАНОВКИ ППО)

12.1. Общая информация о конфигурационных файлах ППО

ПРИМЕЧАНИЕ. Описание параметров конфигурационных файлов сценариев установки ППО и ППО приведено в конфигурационных файлах в виде комментариев.

Структура конфигурационных файлов ППО в общем виде приведена на рисунке (Рисунок 28). Жирным шрифтом выделены файлы, подлежащие редактированию. Редактирование параметров в остальных файлах не предполагается.

```
/var/ocs/config/
├── environments
│   └── <название окружения>
│       ├── vars
│       │   └── _vars.yml
│       └── config.yml
├── subsystems
│   └── <название подсистемы>
│       ├── applications
│       │   ├── <название сервиса>
│       │   └── vars
│       │       ├── services.yml
│       │       └── _vars.yml
│       ├── config.yml
│       ├── internal.yml
│       └── scopes.yml
├── config.yml
├── internal.yml
└── secret.yml
```

Рисунок 28

ППО содержит следующие типы конфигурационных файлов:

– конфигурационный файл ППО (`/vars/ocs/config/config.yml`);

– конфигурационные файлы подсистем ППО (`/vars/ocs/config/subsystems/<название подсистемы>/config.yml`);

– конфигурационные файлы с паролями и токенами компонентов среды функционирования ППО (`/vars/ocs/config/secret.yml`);

– конфигурационные файлы сервисов (модулей) ППО (/vars/ocs /config/subsystems/<название подсистемы>/applications/<название сервиса>/).

В конфигурационном файле ППО содержатся настройки ППО.

В конфигурационных файлах подсистем содержатся настройки подсистем ППО. Также в конфигурационные файлы подсистем вынесены (могут быть вынесены) отдельные настройки сервисов ППО, которые может изменять администратор ППО. В данном случае в конфигурационном файле содержится секция с именем сервиса. Например, секция для сервиса ocs-auth-accounts-users-api выглядит следующим образом:

```
#-------------------------------------------------------------------
---------
# Parameters for user accounts
#-------------------------------------------------------------------
---------
ocs-auth-accounts-users-api:

  ##
  # The number of recently used passwords,
  # which system will store for forbidding use it for new password
creating.
  ##
  passwordHistoryDepth: 3

  ##
  # Maximum inactivity period 45 days.
  # If account not use system during this time, account will be
blocked.
  # Must be greater or equal to OIDC refresh token lifetime.
  ##
  maxAccountInactivityPeriod: "1080h"
```

ВНИМАНИЕ! Редактирование конфигурационных файлов сервисов не предполагается.

12.2. Общая информация о конфигурационных файлах сценариев установки ППО

Структура конфигурационных файлов сценариев установки ППО в общем виде приведена на рисунке (Рисунок 29). Жирным шрифтом выделены файлы, подлежащие редактированию. Редактирование параметров в остальных файлах не предполагается.

```
install-ac-mt
├── config
│   ├── environments
│   │   └── <название окружения>
│   ├── subsystems
│   │   └── <название подсистемы>
│   │       ├── applications
│   │       │   └── <название сервиса>
│   │       ├── vars
│   │       │   ├── services.yml
│   │       │   └── _vars.yml
│   │       ├── config.yml(или config.yml.j2)
│   │       ├── internal.yml
│   │       └── scopes.yml
│   ├── vars
│   │   └── _vars.yml
│   ├── config.yml.j2
│   ├── internal.yml
│   ├── yamllint.release.yml
│   └── secret.yml
└── inventories
    ├── <название окружения>
    └── hosts.yml
```

Рисунок 29

Сценарии установки ППО содержат следующие типы конфигурационных файлов:

– инвентарный файл `inventories/hosts.yml`;

– конфигурационный файл сценария установки ППО `config/vars/_vars.yml`;

– конфигурационные файлы сценариев установки подсистем ППО `config/subsystems/<название подсистемы>/vars/_vars.yml`;

– шаблон конфигурационного файла ППО (`config/config.yml.j2`);

– конфигурационные файлы подсистем ППО (config/subsystems/<название подсистемы>/config.yml);

– шаблоны конфигурационных файлов подсистем ППО (config/subsystems/<название подсистемы>/config.yml.j2);

– конфигурационные файлы сервисов (модулей) ППО (config/subsystems/<название подсистемы>/applications/<название сервиса>/);

– конфигурационный файл с паролями и токенами компонентов среды функционирования ППО config/secret.yml.

12.2.1. Инвентарный файл inventories/hosts.yml

В инвентарном файле inventories/hosts.yml содержатся адреса серверов (имена хостов), на которые установлены (будут установлены) компоненты среды функционирования ППО и подсистемы ППО.

Описание параметров инвентарного файла inventories/hosts.yml приведено в п. 11.1.1.

12.2.2. Общий конфигурационный файл сценариев установки config/vars/_vars.yml

Конфигурационный файл config/vars/_vars.yml является общим для всех подсистем и модулей ППО и содержит полный перечень общих параметров, относящихся к подсистемам и модулям ППО.

Конфигурационные файлы _vars.yml используются только в процессе установки, при этом конфигурационные файлы config.yml (config.yml.j2) используются как в процессе установки, так и в процессе эксплуатации ППО.

12.2.3. Конфигурационные файлы сценариев установки для подсистем ППО (файлы: config/subsystems/<название подсистемы>/vars/_vars.yml)

Конфигурационные файлы `_vars.yml` подсистем содержат параметры, относящиеся к конкретной подсистеме. Также данные файлы могут быть дополнены параметрами из общего конфигурационного файла, значения которых необходимо переопределить для заданной подсистемы.

Конфигурационные файлы `_vars.yml` в основном содержат настройки взаимодействия подсистем с компонентами среды функционирования и располагаются в каталоге со сценариями установки по следующему пути:

```
config/subsystems/<название подсистемы>/vars/_vars.yml
```

Например, конфигурационный файл `vars.yml` для ПБ:

```
config/subsystems/auth/vars/_vars.yml
```

12.2.4. Шаблоны конфигурационных файлов ППО и подсистем ППО

На основе данных файлов в процессе установки ППО формируются конфигурационные файлы ППО и подсистем ППО. Значения параметров в шаблонах конфигурационных файлах подсистем ППО задаются администратором, а также сценариями установки на основе значений, заданных администратором в конфигурационных файлах `_vars.yml`.

Данные конфигурационные файлы располагаются по следующему пути:

```
config/config.yml.j2
config/subsystems/<название подсистемы>/config/services/config.yml.j2
```

Например, шаблон конфигурационного файла ПБ:

```
config/subsystems/auth/config/services/config.yml.j2
```

12.2.5. Конфигурационный файл с паролями и токенами компонентов среды функционирования ППО config/secret.yml

В данном конфигурационном файле задаются пароли и токены Redpanda, Consul, СУБД Redis, СУБД PostgreSQL, а также секретный ключ клиентов (сервисов) и ключ шифрования секретов, хранящихся в БД. При установке ППО данные конфигурационные файлы копируются на серверы приложений.

12.2.6. Конфигурационные файлы подсистем ППО

В данных конфигурационных файлах задаются значения параметров подсистем ППО. В отличие от шаблонов конфигурационных файлов подсистем ППО значения параметров задаются только администратором.

Данные конфигурационные файлы располагаются по следующему пути:

```
config/subsystems/<название подсистемы>/config/services/config.yml
```

Например, шаблон конфигурационного файла ПМ:

```
config/subsystems/appstore/config/services/config.yml.j2
```

12.2.7. Конфигурационные файлы сервисов ППО

Конфигурационные файлы сервисов располагаются в каталоге со сценариями установки по следующему пути:

```
config/subsystems/<название подсистемы>/applications/<название сервиса>/
```

Например, конфигурационные файлы сервиса `ocs-auth-adminconsole-ui` ПБ:

```
config/subsystems/auth/applications/ocs-auth-adminconsole-ui/
```

Описание параметров конфигурационных файлов сервисов приведено в самих конфигурационных файлах в виде комментариев.

ВНИМАНИЕ! Редактирование конфигурационных файлов сервисов ППО не предполагается.

12.2.8. Конфигурационные файлы окружений

В конфигурационных файлах окружения переопределяются параметры конфигурационных файлов, описанных в п. 12.2.1 - 12.2.7 для заданного окружения. Располагаются данные конфигурационные файлы в каталогах `config/environments/<название окружения>/` и `inventories/<название окружения>/`.

ВНИМАНИЕ! В конфигурационных файлах окружений не допускается использовать шаблоны конфигурационных файлов (файлов с расширением «.j2»).

Для переопределения параметра необходимо выполнить следующие действия:

– создать в каталоге `inventories/<название окружения>/` инвентарный файл `hosts.yml` по аналогии с файлом `inventories/hosts.yml` и задать в созданном файле требуемые значения параметров;

– создать в каталоге `config/environments/<название окружения>/` требуемый конфигурационный файл с учетом его расположения в каталоге `config`.

Например, для переопределения параметров конфигурационного файла `config/vars/_vars.yml` для окружения `release` должен быть создан следующий конфигурационный файл: `config/environments/release/config/vars/_vars.yml`;

– скопировать требуемый параметр (включая секцию, в которую входит параметр) из общего конфигурационного файла сценариев установки ППО или конфигурационного файла сценариев установки подсистем ППО;

– вставить скопированное значение в аналогичный конфигурационный файл для заданного окружения;

– задать требуемое значение параметра.

12.2.9. Порядок работы с конфигурационными файлами сценариев установки ППО

Параметры конфигурационных файлов сценариев установки применяются согласно приоритетам, заданным в таблице (Таблица 33).

Таблица 33

Типы конфигурационных файлов	Каталог (имя файла)	Порядок применения параметров (приоритет параметров)

Общие (для всех подсистем и модулей ППО) конфигурационные файлы (шаблоны конфигурационных файлов) сценариев установки ППО	`config/vars/_vars.yml`
	
	`config/config.yml.j2`
		1 (самый низкий приоритет)

Конфигурационные файлы сценариев установки подсистем ППО	`config/subsystems/<название подсистемы>/vars/` Например, `config/subsystems/auth/vars/`	2

Общие (для всех подсистем и модулей ППО) конфигурационные файлы сценариев установки ППО для заданного окружения	`config/environments/<название окружения>/vars/_vars.yml`
	
	`config/environments/<название окружения>/config.yml`
		3

Конфигурационные файлы сценариев установки подсистем ППО для заданного окружения	`config/environments/<окружение>/<название подсистемы>/vars/`	4 (самый высокий приоритет)

При установке ППО параметры конфигурационных файлов применяются в соответствии с порядком, приведенным в таблице (Таблица 33), т.е. сценарий установки обрабатывает сначала конфигурационные файлы в каталоге `config/vars/`, затем в каталоге `config/subsystems/<название подсистемы>/vars/` и т.д. Если, например, какой-либо параметр одновременно задан и в `config/vars/` и `config/subsystems/<название подсистемы>/vars/`, ППО будет установлено со значением параметра, заданным в `config/subsystems/<название подсистемы>/vars/`.

Ниже описаны правила обработки сценариями установки ППО параметров, массивов и списков, если они одновременно заданы в нескольких конфигурационных файлах.

Правило обработки параметров: значение параметра в конфигурационном файле с более высоким приоритетом переопределяет значение параметра в конфигурационном файле с более низким приоритетом.

Пример параметра:

```
redis_password: "example_redis_password"
```

Правило обработки массивов: массив в конфигурационном файле с более высоким приоритетом переопределяет массив в конфигурационном файле с более низким приоритетом.

Пример массива:

```
pg_hba_settings:
  - type: local  # Unix-socket access
    name: all
    database: all
    method: trust
  - type: host  # Localhost IPv4 access
    name: all
    database: all
    address: 127.0.0.1/32
    method: trust
  - type: host  # Localhost IPv6 access
    name: all
    database: all
```

```
    address: ::1/128
    method: trust
  - type: host  # Gitlab CI vbox-testing
    name: all
    database: all
    address: 172.17.0.0/16
    method: md5
```

Правило обработки списков: если список в конфигурационном файле с более низким приоритетом содержит новые элементы (которых не было в конфигурационном файле с более высоким приоритетом), они добавляются к исходному списку. Значение параметра в списке, содержащемся в конфигурационном файле с более высоким приоритетом, переопределяет значение параметра из списка, содержащегося в конфигурационном файле с более низким приоритетом.

Пример списка:

```
postgresql:
  dbname: example_db_name # database name
  port: 5432              # port
  user: example_user      # user
  password: ocs           # password
  extensions: ["pg_partman_bgw", "pg_trgm", "pg_stat_statements",
"pgcrypto"] # necessary extensions
```

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

Используемые в настоящем документе термины и сокращения приведены в таблице (Таблица 34).

Таблица 34

Термин/ Сокращение	Расшифровка
БД	База данных
ГИС	Государственная информационная система
ИС	Информационные системы
НСД	Несанкционированный доступ
ОС	Операционная система
ПБ	Подсистема безопасности
ПМ	Подсистема «Маркет»
ПО	Программное обеспечение
ПООС	Подсистема обновления ОС
ПСУ	Подсистема Сервис уведомлений
ПУ	Подсистема Платформа управления
ПУТ	Подсистема управления тенантами
ППО	Прикладное программное обеспечение «Аврора Центр»
Предприятие-изготовитель, предприятие-разработчик	Общество с ограниченной ответственностью «Открытая мобильная платформа» (ООО «Открытая мобильная платформа»)
Приложение	Приложением является мобильное приложение, функционирующее под управлением ОС Аврора
СЗИ	Средство защиты информации
СПО	Специальное программное обеспечение
СУБД	Система управления базами данных
Токен	Токен - аутентификационные данные, которые выдаются пользователю после успешной авторизации и являются ключом для доступа к службам

Термин/ Сокращение	Расшифровка
Типы портов	1. Внешний - доступ к данному типу портов осуществляется из-за пределов контролируемой зоны. Например, запросы от пользователей с ролью Пользователь Аврора Маркет. Доступ к данным портам имеет нарушитель; 2. Внутренний - доступ к данному типу портов может осуществляться только из контролируемой зоны. Данные порты используются для взаимодействия: между сервисами ППО, сервисов ППО с компонентами среды функционирования ППО, компонентами среды функционирования ППО, привилегированных пользователей с ППО
Устройство	Под устройством подразумевается мобильное устройство, на которой функционируют соответствующие компоненты ППО
ЦОД	Центр обработки данных
ЭВМ	Электронно-вычислительная машина
ЭД	Эксплуатационная документация
API	Application Programming Interface – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой
CD-ROM	Compact Disc Read-Only Memory – разновидность компакт-дисков с записанными на них данными, доступными только для чтения
CDN	Content Delivery Network
Cookie	Небольшой фрагмент данных, отправленный веб-сервером и хранимый на ЭВМ пользователя. Веб-клиент (обычно веб-браузер) всякий раз при попытке открыть страницу соответствующего веб-сайта пересылает этот фрагмент данных веб-серверу в составе http-запроса
CSS3	Cascading Style Sheets 3 – спецификация CSS. Представляет собой формальный язык, реализованный с помощью языка разметки
DHCP	Dynamic Host Configuration Protocol - сетевой протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP
DNS	Domain Name System - компьютерная распределенная система для получения информации о доменах

Термин/ Сокращение	Расшифровка
DVD	Digital Video Disc - оптический носитель информации, выполненный в форме диска, для хранения различной информации в цифровом виде
ECMAScript 5	Встраиваемый расширяемый не имеющий средств ввода-вывода язык программирования, используемый в качестве основы для построения других скриптовых языков
HTML5	HyperText Markup Language, version 5 – язык для структурирования и представления содержимого веб-страницы
HTTP	HyperText Transfer Protocol – протокол прикладного уровня передачи данных. Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом
HTTPS	Hypertext Transfer Protocol Secure - расширение протокола HTTP для поддержки шифрования в целях повышения безопасности. Данные в протоколе HTTPS передаются поверх криптографических протоколов SSL или TLS
IP	Internet Protocol - основной протокол сетевого уровня, использующийся в Интернете и обеспечивающий единую схему логической адресации устройств в сети и маршрутизацию данных
ISO-образ	Образ оптического диска, содержащий файловую систему стандарта ISO 9660
JSON	JavaScript Object Notation – текстовый формат обмена данными, основанный на JavaScript
MTP	Media Transfer Protocol - аппаратно-независимый протокол, основанный на PTP
NFS	Network File System — протокол сетевого доступа к файловым системам, позволяющий монтировать (подключать) удалённые файловые системы через сеть. За основу взят протокол вызова удалённых процедур (ONC RPC)
Nginx	Веб-сервер и почтовый прокси-сервер, работающий на Unix-подобных ОС

Термин/ Сокращение	Расшифровка
OIDC	OpenID Connect – уровень аутентификации OAuth 2.0, инфраструктуры авторизации. Контролируется OpenID Foundation
RPM-пакет	Файл формата .rpm, позволяющий устанавливать, удалять и обновлять приложения на устройствах
SMTP	Simple Mail Transfer Protocol - сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP
SSH	Secure SHell – сетевой протокол прикладного уровня, позволяющий производить удаленное управление ОС и туннелирование TCP-соединений (например, для передачи файлов)
TCP	Transmission Control Protocol - протокол транспортного уровня, гарантирующий целостность передаваемых данных и уведомление отправителя о результатах передачи
TLS	Transport Layer Security – криптографический протокол, обеспечивающий защищенную передачу данных между узлами в сети Интернет
URL	Uniform Resource Locator – единообразный локатор (определитель местонахождения) ресурса

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	изме-ненных	заме-ненных	новых	аннули-рован-ных					