УТВЕРЖДЕН

АДМГ.20134-01 91 01-ЛУ

ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «АВРОРА ЦЕНТР»

Руководство администратора

АДМГ.20134-01 91 01

Листов 133

2023

АННОТАЦИЯ

Настоящий документ является руководством администратора Прикладного программного обеспечения «Аврора Центр» АДМГ.20134-01 релиз 4.0.0 (далее — ППО).

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. Состав и назначение ППО

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

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

– управления жизненным циклом мобильных приложений (МП);

– отправки push-уведомлений на МУ;

– (обновления ОС) 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.

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[1]:

– подсистема безопасности (ПБ);

– подсистема «Маркет» (ПМ);

– подсистема Платформа управления (ПУ);

– подсистема управления тенантами (ПУТ);

– подсистема Сервис уведомлений (ПСУ);

– подсистема обновления ОС (ПООС).

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 МУ 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`).

1.1.1. Подсистема безопасности

ПБ состоит из следующих компонентов:

– Консоль входа пользователей;

– Консоль администратора ПБ;

[1] Состав подсистем ППО зависит от вариантов поставки, описание которых приведено в документе «Формуляр» АДМГ.20134-01 30 01.

– 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.

ПБ предназначена для реализации следующих функций безопасности Изделия:

– идентификации и аутентификации пользователей и МУ;
– управления идентификаторами пользователей и МУ;
– управления средствами аутентификации;
– управления учетными записями пользователей и МУ;
– управления доступом субъектов доступа к объектам доступа;
– регистрации событий безопасности;
– предоставления пользователям доступа к интерфейсу ПБ.

1.1.2. Подсистема «Маркет»

ПМ состоит из следующих компонентов:

– Консоль администратора ПМ;
– Консоль разработчика ПМ;
– МП «Аврора Маркет»;
– Сервер приложений ПМ.

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

Консоль разработчика ПМ позволяет добавлять новые и обновлять ранее загруженные МП, а также получать доступ к данным МП.

МП «Аврора Маркет» выполняется на МУ, функционирующем под управлением ОС, и служит для отображения данных о МП, 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 МП, при этом сами МП, их значки и скриншоты хранятся в БД.

ПМ предназначена для обеспечения:

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

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

– предоставления пользователям доступа к интерфейсу ПМ.

1.1.3. Подсистема Платформа управления

ПУ состоит из следующих компонентов:

– Консоль администратора ПУ;

– МП «Аврора Центр»;

– Сервер приложений ПУ.

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

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

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 МП «Аврора Центр».

ПУ предназначена для обеспечения:

– управления отдельными МУ (оперативное управление) и группами МУ;

– управления политиками, офлайн-сценариями;

– управления записями о МУ и пользователях МУ;

– управления МП на МУ;

– контроля состояния МУ;

– контроля применения политик на МУ;

– мониторинга событий и предоставления отчетности;

– предоставления пользователям доступа к интерфейсу ПУ.

1.1.4. Подсистема управления тенантами

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.

ПУТ предназначена для обеспечения:

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

– управления организациями;

– управления контактными лицами организаций.

1.1.5. Подсистема Сервис уведомлений

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 МП, push-уведомления которых требуется передавать с сервера приложений на МУ;

– 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 МУ посредством tcp-сервера.

ПСУ предназначена для обеспечения:

– доставки push-уведомлений до МУ;

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

– предоставления пользователям доступа к интерфейсу ПСУ.

1.1.6. Подсистема обновления 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.

ПООС предназначена для обеспечения:

– предоставления информации о пакетах ОС;

– управления дистрибуцией пакетов ОС.

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 МП и получать информацию о них;

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

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

– Администратор Платформы управления – управлять ПУ через интерфейс ППО;

– МП «Аврора Центр» (процесс без участия пользователей) – назначается учетным записям МП «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
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
	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

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

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 ИС, 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.

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

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

Таблица 2

Параметр	Количество устройств				
	10 000	50 000	100 000	275 000	550 000
Процессор	2 ядра	3 ядра	4 ядра	10 ядер	10 ядер
Объем оперативной памяти	4 ГБ	5 ГБ	6 ГБ	8 ГБ	8 ГБ
Объем жесткого диска	HDD 60 ГБ	HDD 80 ГБ	HDD 110 ГБ	HDD 130 ГБ	HDD 160 ГБ
Количество серверов	3	3	3	3	6

В таблице (Таблица 3) приведены аппаратные характеристики серверов БД[2].

Таблица 3

Параметр	Количество устройств						
	10 000 ПБ, ПМ, ПУ	50 000 ПБ, ПМ, ПУ	100 000 ПБ, ПМ, ПУ	275 000		550 000	
				ПБ	ПМ, ПУ	ПБ	ПМ, ПУ
Процессор	2 ядра	5 ядер	6 ядер	3 ядра	8 ядер	4 ядра	16 ядер
Объем оперативной памяти	4 ГБ	6 ГБ	8 ГБ	12 ГБ	12 ГБ	24 ГБ	24 ГБ
Объем жесткого диска	SSD 700 ГБ	SSD 3,5 ТБ	SSD 7 ТБ	SSD 11.6 ТБ	SSD 7.7 ТБ	SSD 23.2 ТБ	SSD 15.4 ТБ
Количество серверов	1	1	1	1	1	1	1

[2] В таблице не учитываются резервные серверы БД, необходимые для создания отказоустойчивой конфигурации.

ПРИМЕЧАНИЕ. Для обеспечения отказоустойчивости необходимо добавить минимум 1 сервер с репликой БД (резервный сервер БД).

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

– 2 ядра процессора;

– 4 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 4) приведены программные характеристики серверов приложений ППО.

Таблица 4

Параметр	Значение	Информация о лицензии
Операционная система	CentOS версии 7 или выше	GNU General Public License, version 2
	Альт 8 СП	Коммерческая
	РЕД ОС 7.3 (сертифицированный)	
	РЕД ОС 7.3	
	Astra Linux Special Edition 1.7 (Смоленск)	
Балансировщик микросервисов	Nginx Web Server версии 1.22.0 или выше	2-clause BSD-like license
Система обнаружения сервисов	Consul версии 1.15.3 или выше	Mozilla Public License, version 2.0
Средство управления конфигурациями микросервисов	Consul Template версии 0.25.1 или выше	Mozilla Public License, version 2.0

Параметр	Значение	Информация о лицензии
Сервис гарантированной доставки сообщений	Nats Streaming Server версии 0.25.1 или выше	Apache License, version 2.0
Прикладное программное обеспечение	ППО «Аврора Центр»	Коммерческая

В таблице (Таблица 5) приведены программные характеристики серверов БД.

Таблица 5

Параметр	Значение	Информация о лицензии
Операционная система	CentOS версии 7 или выше	GNU General Public License, version 2
	Альт 8 СП	Коммерческая
	РЕД ОС 7.3 (сертифицированный)	
	РЕД ОС 7.3	
	Astra Linux Special Edition 1.7 (Смоленск)	
СУБД	Postgres Pro Certified 14.8.2 или выше	Коммерческая
	Postgres Pro Enterprise Certified 13.10.2 или выше	
	Postgres Pro Standard 12.15.1 или выше	
	Postgres Pro Standard 13.11.2 или выше	
	Postgres Pro Standard 14.8.2 или выше	
	PostgreSQL 11.20 или выше	PostgreSQL License
	PostgreSQL 12.15 или выше	
	PostgreSQL 13.11 или выше	
	PostgreSQL 14.8 или выше	
СУБД для хранения сессий	Redis версии 7.0.5 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

Параметр	Значение	Информация о лицензии
Планировщик задач для PostgreSQL	`pg_cron` версии 1.4 или выше	PostgreSQL License

В таблице (Таблица 6) приведены программные характеристики устройств.

Таблица 6

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

Варианты конфигурации среды функционирования, в которых проводилось тестирование ППО, приведены в таблице (Таблица 7).

Таблица 7

ОС	СУБД	СЗИ НСД
CentOS-7.6	PostgreSQL 11.20	
CentOS-7.6	PostgreSQL 11.20	Специальное программное обеспечение (СПО) средств защиты информации (СЗИ) НСД «Аккорд-Х К»
CentOS-7.6	PostgreSQL 12.15	СПО СЗИ НСД «Аккорд-Х К»
CentOS-7.6	PostgreSQL 13.11	
CentOS-7.7, kernel: 3.10.0-1062.9.1.el7.x86_64	PostgreSQL 14.8	СЗИ «Secret Net LSP» версия 1.10.1
CentOS-7.6	Postgres Pro Standard 12.15.1	
CentOS-7.6	Postgres Pro Standard 14.8.2	
Альт 8 СП (версии 8.0)	PostgreSQL 11.14	
Альт 8 СП (версии 8.4)	PostgreSQL 11.14 (из репозитория ОС)	
Альт 8 СП (версии 8.4)	Postgres Pro Standard 12.15.1	
Альт 8 СП (версии 8.4)	Postgres Pro Standard 13.11.2	

ОС	СУБД	СЗИ НСД
Альт 8 СП (версии 8.4)	Postgres Pro Certified (версия ядра postgres: 14.8.2)	
Альт 8 СП (версии 8.4)	Postgres Pro Enterprise Certified (версия ядра postgres: 13.10.2)	
РЕД ОС 7.3 (сертифицированный)	Postgres Pro Standard 13.11.2	
РЕД ОС 7.3 (сертифицированный)	Postgres Pro Certified (версия ядра postgres: 14.8.2)	
РЕД ОС 7.3 (сертифицированный)	Postgres Pro Enterprise Certified (версия ядра postgres: 13.10.2)	
РЕД ОС 7.3	PostgreSQL 12.12 (из репозитория ОС)	
РЕД ОС 7.3	PostgreSQL 13.8 (из репозитория ОС)	
РЕД ОС 7.3	PostgreSQL 14.7 (из репозитория ОС)	
Astra Linux Special Edition 1.7 (Смоленск)	PostgreSQL 11.20	
Astra Linux Special Edition 1.7 (Смоленск)	PostgreSQL 12.15	
Astra Linux Special Edition 1.7 (Смоленск)	PostgreSQL 13.11	
Astra Linux Special Edition 1.7 (Смоленск)	PostgreSQL 14.8	
Astra Linux Special Edition 1.7 (Смоленск)	Postgres Pro Certified (версия ядра postgres: 14.8.2)	
Astra Linux Special Edition 1.7 (Смоленск)	Postgres Pro Enterprise Certified (версия ядра postgres: 13.10.2)	

2. 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.

2.1. 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 1).

Рисунок 1

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 и серверах БД, 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 и серверы БД (подраздел 2.2);
3) Развернуть и настроить управляющую ЭВМ (подраздел 2.3);
4) Настроить компоненты среды функционирования ППО (п. 2.4.1);
5) Настроить ППО (п. 2.4.2);
6) Установить компоненты среды функционирования ППО (п. 2.5.1);
7) Установить ППО (п. 2.5.2);
8) Настроить подсистемы ППО (подраздел 2.8);
9) При необходимости выполнить дополнительные настройки ППО и его среды функционирования (подраздел 2.9);
10) Проверить корректность установки и функционирования ППО (подраздел 2.10).

2.2. Порядок установки и настройки ОС на серверах приложений и серверах БД

2.2.1. Установить на серверы приложений и серверы БД одну из следующих ОС: ОС CentOS версии 7, ОС РЕД ОС версии 7.3, ОС Альт 8 СП или ОС Astra Linux.

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

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

Необходимо, чтобы настройки сети ОС соответствовали следующим требованиям:

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

– ОС CentOS и ОС РЕД ОС версии 7.3: `/etc/sysconfig/network-scripts/ifcfg-<имя интерфейса>`

– ОС Альт 8 СП:

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

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

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 версии 7.3) или `/etc/net/ifaces/<имя интерфейса>/options` (для ОС Альт 8 СП) задать следующее значение параметра `ONBOOT`:

```
ONBOOT=yes
```

– в конфигурационный файл `/etc/network/interfaces` ОС Astra Linux внести следующую запись:

```
auto <имя интерфейса>
```

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 и БД должны быть настроены статические ip-адреса (использование динамических адресов, выдаваемых по DHCP, не допускается).

2.2.2. Перейти в учетную запись суперпользователя с помощью команды:

– ОС CentOS версии 7, ОС Альт 8 СП и ОС РЕД ОС версии 7.3:

```
su -
```

– ОС Astra Linux:

```
sudo -i
```

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

Для этого в конфигурационном файле `/etc/apt/sources.lis` 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
```

2.2.4. Установить пакет `sudo` (в случае использования ОС Альт 8 СП)

Для этого необходимо выполнить следующие действия:

2.2.4.1. Исключить CD-ROM из списка доступных репозиториев с помощью команды:

```
apt-repo rm all cdroms
```

2.2.4.2. Обновить ОС и ядро ОС с помощью команд:

```
apt-get update
apt-get dist-upgrade
apt-get update
apt-get dist-upgrade
update-kernel
integalert fix
```

2.2.4.3. Установить пакет с помощью `sudo` команды:

```
apt-get install sudo
```

2.2.5. 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 и серверах БД), которыми осуществляется установка компонентов среды функционирования, СУБД и ППО. В противном случае в процессе установки возникнут ошибки.

2.2.6. Установить кодировку UTF-8 с помощью команды:

– ОС CentOS версии 7, ОС РЕД ОС версии 7.3 и ОС Astra Linux:

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

– ОС Альт 8 СП:

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

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

2.2.7. 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
```

2.2.8. 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
```

2.2.9. 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.

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

2.2.11. 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'
```

2.2.12. Перезагрузить управляющую ЭВМ и серверы с помощью команды:

```
reboot
```

2.3. Порядок развертывания и настройки управляющей ЭВМ

ВНИМАНИЕ! Перед развертыванием и настройкой управляющей ЭВМ необходимо произвести установку и настройку ОС на серверах приложений и серверах БД в соответствии с подразделом 2.2.

Для развертывания и настройки управляющей ЭВМ необходимо выполнить следующие действия:

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

Управляющая ЭВМ может быть развернута как на отдельной ЭВМ, так и на сервере приложений ППО.

2.3.2. Настроить сетевое взаимодействие управляющей ЭВМ с серверами приложений и серверами БД.

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

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

Для этого в конфигурационном файле `/etc/apt/sources.lis` 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
```

2.3.4. С помощью последовательного выполнения команд установить на управляющей ЭВМ следующие пакеты:

– ОС CentOS версии 7.9.2009 (управляющая ЭВМ):

```
sudo yum -y install epel-release-7
sudo yum -y install jq-1.6
sudo yum -y install sed-4.2.2
sudo yum -y install coreutils-8.22
sudo yum -y install rsync-3.1.2
curl https://bootstrap.pypa.io/pip/2.7/get-pip.py | sudo python
sudo python -m pip install wheel==0.37.1
sudo python -m pip install ansible==2.9.27
sudo python -m pip install yamllint==1.*
```

– ОС Альт 8 СП (управляющая ЭВМ):

```
sudo apt-get -y install jq
sudo apt-get -y install coreutils
sudo apt-get -y install rsync
curl https://bootstrap.pypa.io/pip/2.7/get-pip.py | sudo python
sudo python -m pip install PyYAML==5.4.1
sudo python -m pip install pycparser==2.21
sudo python -m pip install MarkupSafe==1.1.1
sudo python -m pip install jinja2==2.11.3
```

```
sudo python -m pip install enum34==1.1.10
sudo python -m pip install six==1.16.0
sudo python -m pip install ipaddress~=1.0.0
sudo python -m pip install cffi==1.15.0
sudo python -m pip install cryptography==3.3.2
sudo python -m pip install ansible==2.9.27
sudo python -m pip install yamllint==1.*
```

– ОС РЕД ОС (управляющая ЭВМ):

```
sudo yum -y install expect
sudo yum -y install gcc
sudo yum -y install python3-devel
sudo yum -y install libffi-devel
sudo yum -y install rust
sudo yum -y install cargo
sudo yum -y install openssl-devel
curl https://bootstrap.pypa.io/get-pip.py | sudo python3
sudo python3 -m pip install pycparser==2.21
sudo python3 -m pip install cffi==1.15.0
sudo python3 -m pip install cryptography==36.0.2
sudo python3 -m pip install MarkupSafe==2.1.1
sudo python3 -m pip install Jinja2==3.1.1
sudo python3 -m pip install PyYAML==6.0
sudo python3 -m pip install ansible==2.9.27
sudo python3 -m pip install yamllint==1.*
```

– ОС Ubuntu 20.04 (управляющая ЭВМ):

```
sudo apt update
sudo apt -y install jq
sudo apt -y install sed
sudo apt -y install coreutils
sudo apt -y install rsync
curl https://bootstrap.pypa.io/get-pip.py | sudo python3
sudo python3 -m pip install MarkupSafe==2.1.1
sudo python3 -m pip install jinja2==3.1.1
sudo python3 -m pip install PyYAML==6.0
sudo python3 -m pip install cryptography==36.0.2
sudo python3 -m pip install cffi==1.15.0
sudo python3 -m pip install pycparser==2.21
sudo python3 -m pip install ansible==2.9.27
sudo python3 -m pip install yamllint==1.*
sudo apt install python-is-python3
```

– ОС Ubuntu 22.04 (управляющая ЭВМ):

```
sudo apt update
sudo apt -y install jq
sudo apt -y install sed
sudo apt -y install coreutils
sudo apt -y install rsync
curl https://bootstrap.pypa.io/get-pip.py | sudo python3
sudo python3 -m pip install MarkupSafe==2.1.1
```

```
sudo python3 -m pip install jinja2==3.1.1
sudo python3 -m pip install PyYAML==6.0
sudo python3 -m pip install cryptography==36.0.2
sudo python3 -m pip install cffi==1.15.0
sudo python3 -m pip install pycparser==2.21
sudo python3 -m pip install ansible==2.9.27
sudo python3 -m pip install yamllint==1.*
sudo apt install python-is-python3
```

– ОС Astra Linux (управляющая ЭВМ):

```
sudo apt update
sudo apt -y install jq
sudo apt -y install coreutils
sudo apt -y install rsync
sudo apt -y install libffi-dev
sudo apt -y install libssl-dev
sudo apt install python3-distutils
curl https://bootstrap.pypa.io/get-pip.py | sudo python3
sudo python3 -m pip install PyYAML==5.3.1
sudo python3 -m pip install pycparser==2.21
sudo python3 -m pip install MarkupSafe==1.1.1
sudo python3 -m pip install jinja2==2.11.3
sudo python3 -m pip install enum34==1.1.10
sudo python3 -m pip install six==1.16.0
sudo python3 -m pip install ipaddress==1.0.23
sudo python3 -m pip install cffi==1.15.1
sudo python3 -m pip install cryptography==3.2.1
sudo python3 -m pip install ansible==2.9.27
sudo python3 -m pip install yamllint==1.*
```

2.3.5. Настроить SSH доступ управляющей ЭВМ к серверам приложений и серверам БД (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 сервер приложений и БД:

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

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

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

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

2.3.6. Создать на управляющей ЭВМ отдельный каталог и скопировать в него каталог `/server`, находящийся на DVD с ППО.

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

```
cd <путь к каталогу server>
```

2.3.8. Запустить `installer-ac.sh` (или `installer-ac-mt.sh`[3]) с помощью команды:

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

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

Для того, чтобы принять «Лицензионное соглашение» (Рисунок 2), 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-v3.1.1`.

[3] Название файла зависит от варианта поставки ППО.

```
[omp@ocs-app ~]$ ./installer_ac.sh

ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ С КОНЕЧНЫМ ПОЛЬЗОВАТЕЛЕМ

ВАЖНО! ПЕРЕД ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, К КОТОРОМУ ПРИЛАГАЕТСЯ ДАННОЕ ЛИЦЕНЗИОННОЕ СОГЛАШЕ
НИЕ С КОНЕЧНЫМ ПОЛЬЗОВАТЕЛЕМ (ДАЛЕЕ ПО ТЕКСТУ – «ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ»), ПОЖАЛУЙСТА, ВНИМАТЕЛЬНО ПРОЧ
ИТАЙТЕ НИЖЕСЛЕДУЮЩИЕ УСЛОВИЯ. ЕСЛИ ВЫ НЕ СОГЛАШАЕТЕСЬ С УСЛОВИЯМИ НАСТОЯЩЕГО ЛИЦЕНЗИОННОГО СОГЛАШЕНИЯ, Т
О ВЫ НЕ ИМЕЕТЕ ПРАВА ИСПОЛЬЗОВАТЬ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В КАКИХ-ЛИБО ЦЕЛЯХ.

1. ОПРЕДЕЛЕНИЯ
«Правообладатель» – общество с ограниченной ответственностью «Открытая мобильная платформа» (ООО «Открыт
ая мобильная платформа»), 420500, Республика Татарстан, Верхнеуслонский район, г. Иннополис, ул. Универс
итетская, д. 7, офис 59, ОГРН 1161690087020.
«ПО» – прикладное программное обеспечение «Аврора Центр» (ППО «Аврора Центр»), состоящее из следующих по
дсистем: прикладного программного обеспечения «Аврора Центр: Платформа управления» (ППО «Аврора Центр: П
латформа управления»), прикладного программного обеспечения «Аврора Центр: Маркет» (ППО «Аврора Центр: М
аркет») и Сервиса уведомлений Аврора, подробное описание функциональных возможностей которого содержатся
 в Документации. Данное Лицензионное соглашение применяется как к ППО «Аврора Центр», включающему в себя
 все перечисленные выше подсистемы, так и к каждой подсистеме в отдельности вне зависимости от комплектн
ости.
«Документация» – относящиеся к ПО сопроводительные материалы, в том числе Руководство по установке и нас
тройке, Руководство Пользователя, Руководство Администратора, которые принадлежат Правообладателю.
«Устройство» – это аппаратная система (физическая или виртуальная) со встроенным запоминающим устройство
м, на которых может быть запущено ПО.
«Права на интеллектуальную собственность» – все права на интеллектуальную и промышленную собственность,
включая права на изобретения, открытия и патенты на изобретения, включая заявки на выдачу патентов и пер
еизданные патенты, повторные заявки или заявки в продолжение и частичные продолжения; авторские права; о
бразцы и промышленные образцы; товарные знаки, знаки обслуживания, оформление товара и права на аналогич
ные объекты; секреты производства (ноу-хау), коммерческую тайну и конфиденциальную информацию; права на
топологии интегральных микросхем и права на фотошаблоны; и другие исключительные права.
«Лицензионное соглашение» – предоставляемое Вам Правообладателем ограниченное право на использование фун
кциональности ПО на условиях простой (неисключительной) лицензии в соответствии с условиями настоящего Л
ицензионного соглашения.
«Конечный пользователь» – любое юридическое лицо (организация), которое приобрело ПО для собственного ис
пользования и не для продажи.
«Пользователь» – физическое лицо, непосредственно осуществляющее эксплуатацию ПО в целях и порядке, опре
деляемом Конечным пользователем.
Настоящее Лицензионное соглашение является юридическим соглашением между Вами (далее по тексту – Конечны
й пользователь) и Правообладателем.
```

Рисунок 2

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 эксплуатационной документации) 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
```

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

2.4. 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. 2.9.15.

2.4.1. 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:

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

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

Например:

```
cd install-release-v4.0.0/install-ac/
```

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

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

Описание порядка задания адресов в конфигурационном файле `inventories/hosts.yml` приведено в п. 2.9.13.

Для отображения адреса ЭВМ необходимо выполнить команду:

```
hostname
```

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

Описание параметров конфигурационного файла `inventories/hosts.yml` приведено в п. 8.1.1.

2.4.1.3. 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: 12
```

Перечень допустимых значений параметра приведен в таблице (Таблица 8).

Таблица 8

Значение параметра	Версия СУБД
`11`	PostgreSQL 11
`12`	PostgreSQL 12
`13`	PostgreSQL 13
`14`	PostgreSQL 14
`11-pro`	Postgres Pro Standard 11
`12-pro`	Postgres Pro Standard 12
`13-pro`	Postgres Pro Standard 13
`14-pro`	Postgres Pro Standard 14

Значение параметра	Версия СУБД
13-entcert	Postgres Pro Enterprise Certified 13
14-stdcert	Postgres Pro Certified 14

– 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
```

При самостоятельной установке СУБД суперпользователю необходимо создать 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'
```

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

2.4.1.4. 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 сервису гарантированной доставки сообщений `Nats Streaming Server`:

```
transport:
  nats:
    authToken: "FF12fddgdhFLL"
```

– 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.

2.4.1.5. Настроить параметры взаимодействия клиентов СУБД (при необходимости).

ПРИМЕЧАНИЕ. По умолчанию сценарии установки автоматически задают параметры взаимодействия клиентов (например, серверов приложений ППО) СУБД.

Настройки взаимодействия клиентов с СУБД задаются в секции `pg_hba_settings` конфигурационного файла `config/vars/_vars.yml`.

Данная секция содержит следующие параметры:

– `type` - тип подключения к СУБД;

– `name` - имена пользователей СУБД, правила доступа для которых определяет данная запись;

– `database` - имена БД, доступ к которым описывает данная запись;

– `address` - IP-адрес подключения или IP-адрес подсети;

– `method` - метод аутентификации.

2.4.2. Настройка ППО (подсистем ППО)

ВНИМАНИЕ! Перед выполнением настроек необходимо изучить порядок работы с конфигурационными файлами, приведенный в п. 9.2.9.

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:

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

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

Описание порядка задания адресов в конфигурационном файле `inventories/hosts.yml` приведено в п. 2.9.13.

2.4.2.3. 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 п. 2.4.2.4.

2.4.2.4. 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 конфигурационном 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:

2.4.2.4.1 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 конфигурационном файле `inventories/hosts.yml`

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

Если в пп. 2.4.2.3 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}"
```

2.4.2.4.2 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"
```

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

В данном случае в параметре `commonAddress` необходимо задать имя домена и порт, используемые на внешнем балансировщике, например:

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

2.4.2.4.4 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 имя домена и порт для внешних адресов, а в параметре `adminAddress` задать имя домена и порт для внутренних адресов, например:

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

2.4.2.4.5 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"
```

2.4.2.5. 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"
```

2.4.2.6. 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 2.8.

2.5. Порядок установки компонентов среды функционирования ППО и ППО

2.5.1. Установка компонентов среды функционирования ППО

2.5.1.1. 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
```

2.5.1.2. Установить на серверы приложений и серверы БД необходимые пакеты.

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

Для установки пакетов необходимо выполнить следующие действия:

2.5.1.2.1 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
```

Например:

– серверы приложений:

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

– серверы БД:

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

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

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

2.5.1.2.2 На серверах приложений и серверах БД под управлением ОС РЕД ОС включить автозапуск службы `network` с помощью команды:

```
sudo systemctl enable network
```

2.5.1.2.3 Перезагрузить серверы приложений и серверы БД с помощью команды:

```
sudo reboot
```

Порядок действий для самостоятельной установки пакетов, а также отключению служб SELinux и Firewalld приведен в п. 2.9.8 и 2.9.9.

2.5.1.3. 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.

В случае если требуется репликация БД, команду установки компонентов среды функционирования необходимо запустить с параметром `--extra-vars "pg_slave_recreate=true"`:

```
ANSIBLE_USER="<имя пользователя>" ./deploy-infra.sh --extra-vars
"pg_slave_recreate=true"
```

Описание параметров запуска скрипта `deploy-infra.sh` и их возможные значения приведены в подразделе 3.1.

ВНИМАНИЕ! Скрипт `deploy-infra.sh` позволяет устанавливать только СУБД PostgreSQL 11/12/13/14. СУБД Postgres Pro необходимо устанавливать самостоятельно.

При использовании СУБД Postgres Pro либо если установку СУБД PostgreSQL 11/12/13/14 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 2.7.

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 nats-streaming-
server
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
```

2.5.2. 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 3.2.

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. Установка подсистем ППО должна осуществляться строго в следующей последовательности: ПБ, ПМ, ПООС, ПУ, ПУТ, 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 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
```

2.5.3. 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 2.8.

2.5.4. Выполнение ограничений по применению

При эксплуатации ППО необходимо соблюдать следующие ОГРАНИЧЕНИЯ ПО ПРИМЕНЕНИЮ:

– ПСУ не осуществляет аутентификацию подключаемых к нему МУ, 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 МУ;

– после установки и настройки ППО необходимо выполнить ограничения по применению, произвести настройки безопасности компонентов среды функционирования и настроить СЗИ. Необходимая информация приведена в п. 2.9.5.

2.5.5. Проверка корректности установки и функционирования ППО

Проверка осуществляется в соответствии с подразделом 2.10.

2.6. 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 9) приведены адреса веб-консолей.

Таблица 9

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

2.7. Самостоятельная установка и настройка СУБД

2.7.1. Порядок установки и настройки СУБД Postgres Pro

2.7.1.1. Установить на серверы БД необходимые пакеты согласно п. 2.9.8.

2.7.1.2. Установить и инициализировать СУБД Postgres Pro.

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 Postgres Pro осуществляется в соответствии с ЭД на СУБД. После установки СУБД необходимо назначить пароль для пользователя `postgres` с помощью следующих команд:

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

2.7.1.3. 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`.

2.7.1.4. 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 8 СП 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 8 СП 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 8 СП 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 Альт 8 СП и СУБД Postgres Pro Enterprise 13 Cert:

```
sudo rpm -ivh pg_partman_13pro-ent-4.6.0-alt1.x86_64.rpm
sudo rpm -ivh pg_cron_13pro-ent-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 Enterprise 13 Cert:

```
sudo rpm -ivh pg_partman_13pro-ent-4.6.0-1.redos7.x86_64.rpm
sudo rpm -ivh pg_ pg_cron_13pro-ent-cert-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 Astra Linux SE 1.7 и СУБД Postgres Pro Enterprise 13 Cert:

```
sudo dpkg -i pg-partman_4.6.0_ent-13_smolensk.amd64.deb
sudo dpkg -i pg-cron_1.5.2_pro-ent-cert-13_smolensk.amd64.deb
```

– 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
```

Указанные RPM-пакеты находятся на DVD с загрузочными модулями ППО в архиве `/server/install-infra.tar.gz/install-infra/binary/postgresql/`.

2.7.1.5. Перезапустить сервис СУБД Postgres Pro с помощью команды:

```
sudo systemctl restart <имя сервиса СУБД>
```

Например:

```
sudo systemctl restart postgrespro-std-11
```

2.7.2. Порядок установки и настройки СУБД PostgreSQL 11/12/13/14

2.7.2.1. Установить на серверы БД необходимые пакеты в соответствии с п. 2.9.8.

2.7.2.2. Выполнить установку и настройку СУБД PostgreSQL 11/12/13/14 в соответствии с документацией на СУБД.

При инициализации СУБД должны быть установлены следующие значения параметров:

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

2.7.2.3. Установить расширения `pg_partman` и `pg_cron` с помощью команд:

– ОС CentOS версии 7 и СУБД PostgreSQL 11:

```
sudo rpm -ivh pg_partman_11-4.6.0-1.rhel7.x86_64.rpm
sudo rpm -ivh pg_cron_11-1.5.2-1.rhel7.x86_64.rpm
```

– 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 Альт 8 СП и СУБД PostgreSQL 11:

```
sudo rpm -ivh pg_partman_11-4.6.0-1.alt.x86_64.rpm
sudo rpm -ivh pg_cron_11-1.5.2-1.alt.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 Astra Linux SE 1.7 и СУБД PostgreSQL 11:

```
sudo rpm -ivh pg-partman_4.6.0_11_smolensk.amd64.deb
sudo rpm -ivh pg-cron_1.5.2_11_smolensk.amd64.deb
```

– 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
```

Указанные RPM-пакеты находятся на DVD с загрузочными модулями ППО в архиве `/server/install-infra.tar.gz/install-infra/binary/postgresql/`.

2.7.2.4. Перезапустить сервис СУБД PostgreSQL в соответствии с документацией на СУБД.

2.8. Описание настройки подсистем ППО

2.8.1. Описание настройки ПМ

Настройка ПМ заключается в настройке файлового хранилища ПМ, в котором будут храниться файлы МП (иконки, скриншоты, RPM-пакеты), загружаемые разработчиками.

ПРИМЕЧАНИЕ. Настройку файлового хранилища необходимо выполнять после установки ППО.

Для настройки файлового хранилища ПМ необходимо выполнить следующие действия:

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

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

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

Пример настройки единого файлового хранилища приведен в п. 2.9.3.

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

2.8.1.3. В каталоге на Сервере приложений ПМ 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"
```

2.8.2. Описание настройки ПСУ

Настройка ПСУ заключается в настройке обратного прокси-сервера (`reverse proxy`), который осуществляет обработку запросов (подключений) МУ (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-mt/nginx_external-balancer/conf_stream.d/`.

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

2.8.2.1. Скопировать файл `samples/ac-mt/nginx_external-balancer/conf_stream.d/one-node.conf` (или `samples/ac-mt/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-mt/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-mt/nginx_external-balancer/conf_stream.d/three-node.conf /etc/nginx/conf_stream.d/ocs-push-stream.conf
```

2.8.2.2. 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;
}
```

2.8.2.3. 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, например:

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

ПРИМЕЧАНИЕ. Предварительно должен быть создан каталог для хранения лог-файлов.

2.8.2.4. 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
```

2.8.2.5. Перезапустить Nginx с помощью команды:

```
sudo systemctl reload nginx
```

2.8.3. Описание настройки ПУ

Настройка ПУ заключается в настройке взаимодействия Сервера приложений ПУ с Сервисом уведомлений Аврора версии 1.1.2 или ПСУ.

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

2.8.3.1. Синхронизировать время между Сервером приложений ПУ и ПСУ.

2.8.3.2. 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.

2.8.3.3. Задать публичный адрес 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: "http://ocs-app.local:8009"
```

2.8.3.4. Задать адрес и порт ПСУ для МУ (push-демона)

Адрес и порт ПСУ для МУ задаются в секции `pushNotificationSystem` конфигурационного файла `config/config.yml.j2`. и распространяются на все тенанты:

```
  pushNotificationSystem:
   mobileHostname: "{{ groups['app'] | first }}"
   mobilePort: 999
```

Также адрес и порт можно задать в Консоли администратора ПУ при выполнении п. 2.8.3.6. В данном случае настройки будут распространяться только на тенант в рамках которого выполнялась настройка.

2.8.3.5. Переустановить ПУ в соответствии с п. 2.5.2, в случае если настройка осуществляется после установки ПУ.

2.8.3.6. 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.

2.8.4. Описание настройки ПООС

Настройка ПООС заключается в настройке файлового хранилища ПООС и загрузки пакетов ОС в данное хранилище.

ПРИМЕЧАНИЕ. Настройку файлового хранилища необходимо выполнять после установки ППО.

Для настройки файлового хранилища ПООС и загрузки пакетов ОС в данное хранилище необходимо выполнить следующие действия:

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

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

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

Пример настройки единого файлового хранилища приведен в п. 2.9.3.

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

2.8.4.3. В файловом хранилище ПООС 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 <путь к каталогу>
```

Например,

```
mkdir /ocs/pkgrepo/repos
```

2.8.4.4. 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
```

2.8.4.4. 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"
          ]
        }
    ]
}
```

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

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

2.8.4.6. 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 3).

Рисунок 3

2.9. Дополнительные настройки ППО и среды функционирования ППО

2.9.1. 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 ППО 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`);

– тип аутентификации (параметр: `authType`);

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

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

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

– PLAIN:

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

– CRAM-MD5:

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

– LOGIN:

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

– без аутентификации:

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

После изменения настроек необходимо переустановить конфигурационные файлы с помощью команды:

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

2.9.2. Настройка разделения трафика

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

– по `basepath` - каждая Консоль администратора/разработчика (либо API для взаимодействия с МП) 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 МП (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. 2.4.2.4;

– по портам – внутренние и внешние адреса ППО привязаны к отдельным портам. Описание настройки разделения трафика по портам приведено в пп. 2.4.2.3.

2.9.3. Пример настройки единого файлового хранилища

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

Для настройки единого файлового хранилища необходимо выполнить следующие действия:

2.9.3.1. 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`.

2.9.3.2. 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 иПООС должны быть настроены на работу с данным файловым хранилищем.

2.9.3.3. 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.

2.9.3.4. 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`.

2.9.4. 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;

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

– `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"
keys_zone: "proxy_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"
```

2.9.5. 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 10).

Таблица 10

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`

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 пользователей не настраивается. Пароли учетных записей пользователей должны содержать буквы верхнего и нижнего регистров, цифры и специальные символы (это контролируется ППО). 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:	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)
`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: `config.ttl.client_jwks`	Не более 1 года и 3 мес., например: `client_jwks: 10950h`	Не более 1 года и 3 мес., например: `client_jwks: 10950h`	Не более 1 года и 3 мес., например: `client_jwks: 10950h`	Не более 1 года и 3 мес., например: `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`

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)
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"`
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`

2.9.6. 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:

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

2.9.6.2. Установить компоненты среды функционирования в соответствии с п. 2.5.1.

2.9.6.3. Установить ППО в соответствии с п. 2.5.2.

2.9.7. Действия по реализации функций безопасности среды функционирования ППО

2.9.7.1. Установка, настройка и эксплуатация СЗИ НСД

Эксплуатация ППО и СУБД должна осуществляться в одной из следующих ОС:

– CentOS версии 7 с установленными СЗИ «Secret Net LSP» или СПО СЗИ НСД «Аккорд-X К»;

– Альт 8 СП.

Установка СЗИ НСД должна осуществляться после установки ППО. После установки СЗИ НСД необходимо повторно назначить пользователям ОС права на выполнение команд от имени суперпользователя в соответствии с подразделом 2.2.

Установка, настройка и эксплуатация СЗИ НСД и ОС Альт 8 СП должна осуществляться в соответствии с ЭД на СЗИ (ОС).

2.9.7.2. 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 11).

Таблица 11

Номер порта (протокол)	Описание	Конфигурационный файл, в котором задается порт	Тип порта[4]
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

[4] Описание типов портов приведено в таблице (Таблица 26).

Номер порта (протокол)	Описание	Конфигурационный файл, в котором задается порт	Тип порта[4]
Nginx			
80 (tcp)	Служит для взаимодействия сервисов ППО друг с другом	shared_roles/consul-template/defaults/main.yml	внутренний
999 (tls)	Служит для взаимодействия МУ с ПСУ	Конфигурационный файл Nginx согласно п. 2.8.2	внешний
8009 (tcp)	Балансировщик микросервисов (Nginx Web Server)	config/vars/_vars.yml config/config.yml.j2 /etc/nginx/conf.d/ocs.conf	внешний
8025 (tcp)	На данный порт перенаправляются запросы с 999 порта	Конфигурационный файл Nginx согласно п. 2.8.2	внутренний
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
Nats Streaming Server			
4222 (tcp)	nats_port	shared_roles/nats-streaming-server/defaults/main.yml	внутренний

Номер порта (протокол)	Описание	Конфигурационный файл, в котором задается порт	Тип порта[4]
6222 (tcp)	`nats_cluster_port`	`shared_roles/nats-streaming-server/defaults/main.yml`	внутренний
8222 (tcp)	`nats_monitoring_port`	`shared_roles/nats-streaming-server/defaults/main.yml`	внутренний
Dnsmasq			
53	`dnsmasq`		внутренний
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. 2.9.2.

2.9.7.3. Настройка ОС CentOS

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

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

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

2.9.7.3.1.2 Применить конфигурацию, выполнив команду:

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

2.9.7.3.1.3 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
```

2.9.7.3.2 Настройка запрета `ssh` доступа к серверам приложений по логину и паролю.

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

```
PasswordAuthentication no
AuthenticationMethods publickey
```

2.9.7.3.2.2 Перезапустить службу `sshd` с помощью команды:

```
sudo service sshd reload
```

2.9.7.3.3 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
```

2.9.8. Самостоятельная установка необходимых пакетов на серверы приложений и серверы БД

2.9.8.1. Получить список необходимых пакетов.

Перечень необходимых пакетов, которые должны быть установлены на серверы приложений и серверы БД, 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 и сервера БД.

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.

Пример перечня пакетов для сервера приложений и сервера БД, 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"
```

2.9.8.2. Установить пакеты.

Установка пакетов осуществляется в соответствии с документацией на ОС.

2.9.9. Отключение служб SELinux и Firewalld

Для отключения служб SELinux и Firewalld необходимо выполнить следующие действия:

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

```
SELINUX=disabled
```

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

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

2.9.9.3. Перезагрузить ЭВМ с помощью команды:

```
reboot
```

2.9.10. 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
```

2.9.10.1. Настройка балансировщика для однотенантной конфигурации:

– 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 `samples/ac/nginx_external-balancer/conf.d/one-node.conf`).

2.9.10.2. 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:

– 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 `samples/ac-mt/nginx_external-balancer/conf.d/one-node.conf`).

2.9.11. 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:

2.9.11.1. 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
```

2.9.11.2. 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';
```

2.9.12. Действия после сброса устройств к заводским настройкам

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

После сброса устройства необходимо выполнить следующие действия:

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

2) Активировать устройство в ПУ в соответствии с документом АДМГ.20134-01 90 01-3;

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

2.9.13. Порядок задания адресов (доменных имен) в конфигурационном файле inventories/hosts.yml

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:
        postgresql:
          children:
            postgresql_masters:
              hosts:
                ocs-app.local:
            postgresql_slaves:
              hosts:
        nginx:
          children:
            app:
          hosts:
        consul:
          children:
            consul_servers:
              children:
                app:
              hosts:
            consul_agents:
        consul_template:
          children:
            app:
        nats_streaming_server:
          children:
                app:
          hosts:
        redis:
          children:
            redis_masters:
              children:
                app:
              hosts:
            sentinel:
              children:
                app:
              hosts:
```

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

Описание параметров конфигурационного файла `inventories/hosts.yml` приведено в п. 8.1.1.

2.9.14. 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:

2.9.14.1. 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
```

2.9.14.2. Просмотреть текущее значение срока хранения с помощью скрипта:

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

2.9.14.3. 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';
```

2.9.15. 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:

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

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

Описание параметров конфигурационного файла `hosts.yml` приведено в п. 8.1.1.

2.9.15.3. 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. 9.2.8.

2.9.15.4. Выполнить установку компонентов среды функционирования ППО и ППО в соответствии с подразделом 2.5 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 и серверы БД 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>"
```

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

2.9.16.1. 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';
```

2.9.16.2. 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';
```

2.9.16.3. 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;
```

2.9.16.4. 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';
```

2.9.17. 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».

2.9.18. 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».

2.9.19. 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 3.3.

2.9.20. 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'
```

2.10. Проверка корректности установки и функционирования ППО

2.10.1. 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/`) 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;

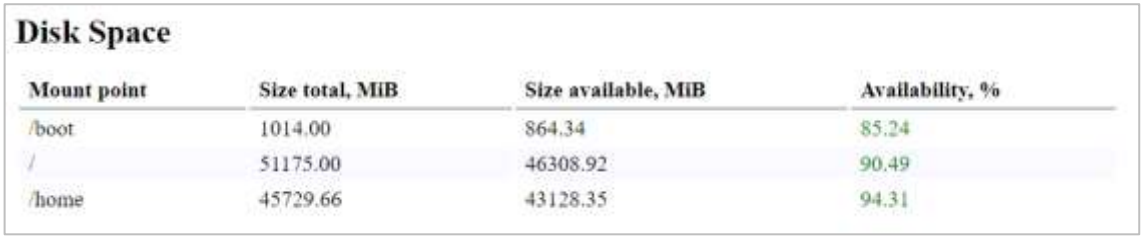
– общая информация о статусе компонентов среды функционирования;

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

2.10.2. Описание параметров диагностического отчета

2.10.2.1. Раздел «Disk Space»

Раздел содержит информацию о полном и доступном объеме дискового пространства для ППО (Рисунок 4).



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 4

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 12).

Таблица 12

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.

2.10.2.2. Раздел «Systemd Unit Status»

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

2.10.2.3. OCS Targets

Подраздел содержит информацию о статусе конфигураций групп сервисов ППО (Рисунок 5).

Рисунок 5

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 13).

Таблица 13

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

2.10.2.3.1 OCS Services

Подраздел содержит информацию о статусе сервисов ППО (Рисунок 6).

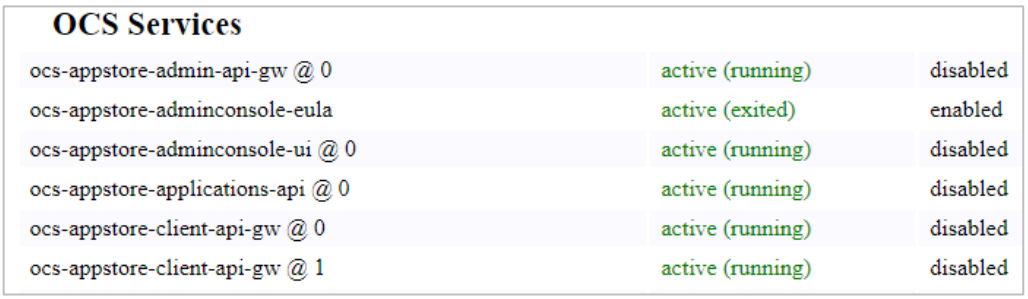


Рисунок 6

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 14).

Таблица 14

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

2.10.2.3.2 Mandatory services

Подраздел содержит информацию о статусе сервисов компонентов среды функционирования (Рисунок 7).

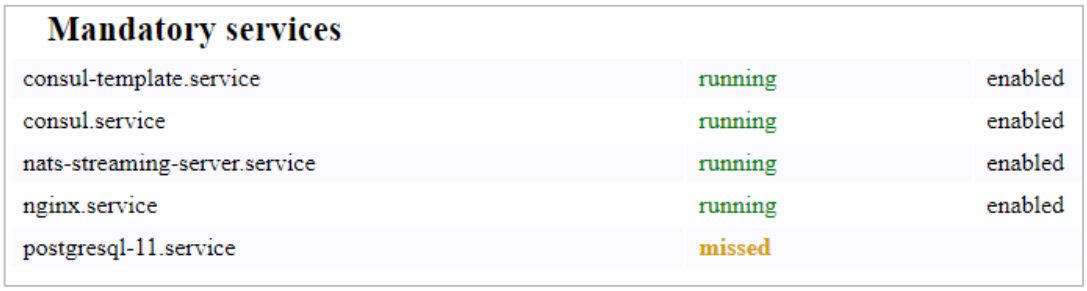


Рисунок 7

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 15).

Таблица 15

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`
		`nats-streaming-serever.service`
		`nginx.service`
		`postgresql-11.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

2.10.2.4. Раздел «API GW Service Status»

Раздел содержит информацию о статусе регистрации сервисов в системе обнаружения сервисов (Consul). На рисунке (Рисунок 8) приведен пример статуса регистрации сервисов в системе обнаружения сервисов.

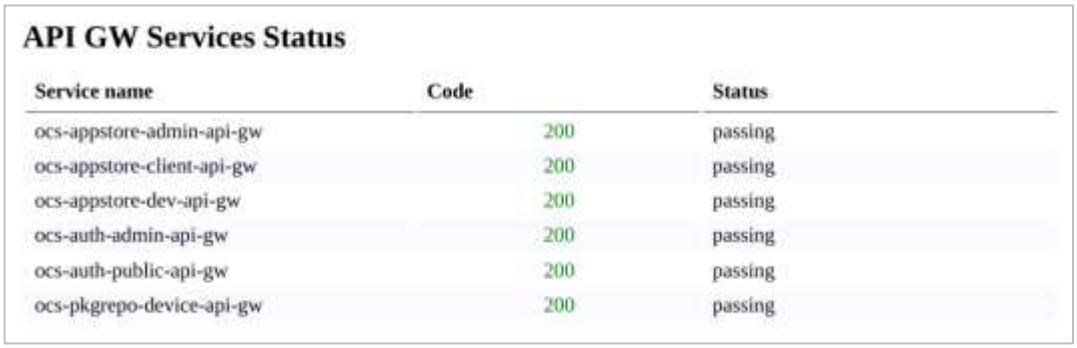


Рисунок 8

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 16).

Таблица 16

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

2.10.2.5. Раздел «Consul Cluster Endpoints Availability»

Раздел содержит информацию о проверке доступности интерфейсных функций системы обнаружения сервисов (Consul). На рисунке (Рисунок 9) приведен пример отображения информации о доступности интерфейсных функций системы обнаружения сервисов.

Рисунок 9

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 17).

Таблица 17

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

2.10.2.6. Раздел «Consul Service Health Check»

Раздел содержит информацию о статусе регистрации сервисов ППО в системе обнаружения сервисов Consul (Рисунок 10).

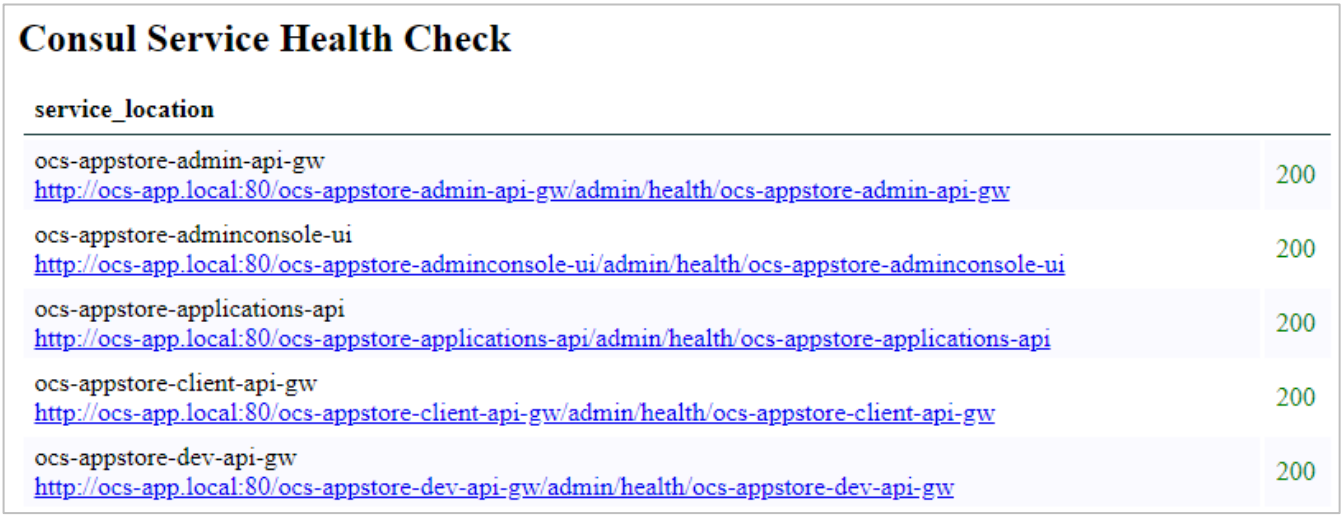


Рисунок 10

Описание назначения столбцов таблицы, а также информация о возможных значениях приведены в таблице (Таблица 18).

Таблица 18

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`.

2.10.2.7. Раздел «Cluster Nodes Reachability»

Раздел содержит информацию о результатах проверки доступности серверов (нод) кластера (Рисунок 11).

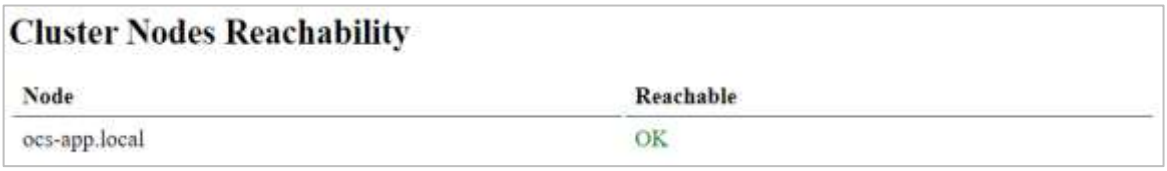


Рисунок 11

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

Таблица 19

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`

2.10.2.8. Раздел «Nginx Service Proxy»

Раздел содержит информацию о проверке конфигурации балансировщика микросервисов Nginx Web Server для каждого сервиса ППО (Рисунок 12).

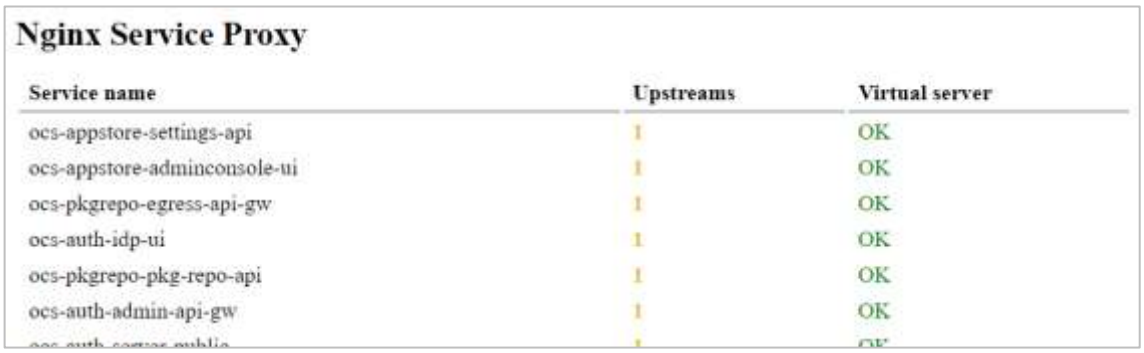


Рисунок 12

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

Таблица 20

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

2.10.2.9. Раздел «Filestorage Configuration»

Раздел содержит информацию о конфигурации файловых хранилищ ПМ и ПООС (Рисунок 13).

Рисунок 13

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. УПРАВЛЕНИЕ КОМПОНЕНТАМИ СРЕДЫ ФУНКЦИОНИРОВАНИЯ, СЕРВИСАМИ, НАСТРОЙКАМИ СЕРВИСОВ И ПОДСИСТЕМ

3.1. Управление компонентами среды функционирования ППО

Управление сервисами 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 2.3).

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 и БД.

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`;

– nats-streaming-server;

– redis;

– ocs-user;

– db.

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 8).

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) -e, --extra-vars

В данном параметре передаются внешние переменные для скриптов развертывания. В ППО используются следующие внешние переменные:

– `pg_slave_recreate=true` - служит для инициализации реплики БД;

– `pg_uninstall_delete_data=true` - служит для удаления данных при удалении СУБД PostgreSQL.

8) `--force-infra-install`

Флаг служит для управления принудительной повторной установки 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) `--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
```

3.2. 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 2.3).

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 и БД.

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);
- `pkgrepo` (для ПООС).

В данном параметре может задаваться список подсистем, например:

```
--subsystems auth,appstore,pkgrepo,emm,mt,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 21).

Таблица 21

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
	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

Значение параметра «--subsystems»	Сервисы (значение параметра «--apps»)
	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-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
ПООС	
pkgrepo	ocs-pkgrepo-pkg-repo-api
	ocs-pkgrepo-device-api-gw
	ocs-pkgrepo-admin-api-gw
	ocs-pkgrepo-egress-api-gw

4) `-A, --action`

Данный параметр задает действие, которое необходимо выполнить. Перечень допустимых действий и соответствующие им значения параметра приведены в таблице (Таблица 22).

Таблица 22

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, 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 8).

Например:

```
--database 12
```

По умолчанию (если параметр не задан) будет использоваться значение, заданное в параметре `pg_version` конфигурационного файла `config/vars/_vars.yml`.

7) `--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 --apps ocs-appstore-applications-api,ocs-appstore-adminconsole --action start
```

3) Получение справочной информации:

```
./deploy-ac.sh --help
```

3.3. Управление настройками сервисов и подсистем ППО

Управление настройками сервисов и подсистем ППО может осуществляться 2 способами.

3.3.1. Способ 1 (рекомендуемый)

3.3.1.1. Задать требуемые значения параметров в конфигурационных файлах сценариев установки ППО и подсистем ППО.

3.3.1.2. 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 3.2.

3.3.2. Способ 2

3.3.2.1. Задать требуемые значения параметров в конфигурационных файлах сервисов и подсистем ППО. Описание параметров конфигурационных файлов сценариев установки подсистем ППО приведено в разделе 9.

3.3.2.2. 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 3.2.

4. РЕЗЕРВНОЕ КОПИРОВАНИЕ

ВНИМАНИЕ! Приведенные ниже имена файлов и каталогов характерны для типового варианта установки ППО и среды функционирования ППО.

4.1. Резервное копирование после установки (обновления) ППО

После успешной установки (обновления) ППО необходимо создать резервную копию каталога `install-<версия ППО>/install-ac/` (`install-<версия ППО>/install-ac-mt/`).

4.2. 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» АДМГ.20134-01 91 02.

5. ОБНОВЛЕНИЕ ППО И ОС АВРОРА

5.1. Порядок обновления

Обновлять ППО до требуемой версии допустимо только с версии ППО, указанной в таблице (Таблица 23).

Таблица 23

Номер новой версии	Перечень версий, с которых допустимо обновление до новой версии
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

* - версии ППО, прошедшие сертификацию во ФСТЭК России.

5.2. 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.

5.2.1. Создать резервную копию данных, ППО и компонентов среды функционирования в соответствии с разделом 3.

5.2.2. Скопировать на управляющую ЭВМ архив с новой версией ППО и распаковать его в соответствии с п. 2.3.6 - 2.3.9.

5.2.3. Обновить на управляющей ЭВМ пакеты в соответствии с п. 2.3.3.

5.2.4. Настроить компоненты среды функционирования ППО и ППО в соответствии с подразделом 2.4.

5.2.5. Выполнить upgrade скрипты.

Для этого необходимо перейти в каталог со сценариями установки новой версии ППО (каталог: `install-<новая версия ППО>/install-ac/`) и выполнить все скрипты `play-upgrade_to_release_<версия ППО>.yml`, версии которых лежат в диапазоне между установленной версий ППО (**не включительно**) и новой версией ППО (**включительно**). Для некоторых версий ППО скрипты могут отсутствовать. Запуск скриптов осуществляется с помощью команды:

```
ansible-playbook -i inventories/hosts.yml -u <имя пользователя>
release_upgrade/play-upgrade_to_release_<версия ППО>.yml -vv --diff
```

Например, для обновления ППО с версии 2.2.2 до версии 2.5.1 необходимо выполнить следующие команды:

```
ansible-playbook -i inventories/hosts.yml -u <имя пользователя>
release_upgrade/play-upgrade_to_release_2.5.0.yml -vv --diff
ansible-playbook -i inventories/hosts.yml -u <имя пользователя>
release_upgrade/play-upgrade_to_release_2.5.1.yml -vv --diff
```

5.2.6. Обновление СУБД PostgreSQL 11/12/13/14 до новой старшей версии[5] (`major version`) необходимо осуществлять в соответствии с ЭД на СУБД.

5.2.7. Установить компоненты среды функционирования в соответствии с п. 2.5.1.

5.2.8. Установить ППО в соответствии с п. 2.5.2.

[5] Согласно спецификации SemVer 2.0.0.

5.2.9. Выполнить `post_upgrade` скрипты.

Для этого необходимо перейти в каталог со сценариями установки новой версии ППО (каталог: `install-<новая версия ППО>/install-ac/`) и выполнить все скрипты `play-post_upgrade_to_release_<версия ППО>.yml`, версии которых лежат в диапазоне между старой версий ППО (**не включительно**) и новой версией ППО (**включительно**). Для некоторых версий ППО скрипты могут отсутствовать.

Запуск скриптов осуществляется с помощью команды:

– скрипт `play-post_upgrade_to_release_3.0.0.yml`:

```
ansible-playbook -i inventories/hosts.yml -u <имя пользователя>
release_upgrade/play-post_upgrade_to_release_3.0.0.yml  -vv --diff --
limit <хост с ПМ>
```

В параметре `limit` необходимо указать имя одного из хостов с установленной ПМ (например, `ocs-app.local`).

– скрипт `play-post_upgrade_to_release_3.1.0.yml`:

```
ansible-playbook -i inventories/hosts.yml -u <имя пользователя>
release_upgrade/play-post_upgrade_to_release_3.1.0.yml -vv --diff
```

5.2.10. 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
```

5.2.11. Оповестить пользователей ППО о необходимости очистить кэш и cookies веб-браузера. Иначе при открытии интерфейса ППО будет ошибка HTTP ERROR 400.

5.3. Обновление ОС Аврора с помощью ПУ

Обновление ОС Аврора выполняется в следующей последовательности:

5.3.1. Обновить сервер приложений ППО в соответствии с подразделом 5.1.

5.3.2. Загрузить в файловое хранилище ПООС пакеты требуемой версии ОС в соответствии с п. 2.8.4.

5.3.3. 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 АДМГ.20134-01 90 01-3.

5.3.4. Убедиться, что после окончания заданного в правиле временного интервала обновления в карточке каждого устройства из тестовой группы отображается требуемая версия ОС Аврора.

5.3.5. Обновить МП ППО на тестовой группе устройств в соответствии с документом АДМГ.20134-01 90 01-7.

5.3.6. Выполнить обновление аналогичным образом для остальных устройств после успешного обновления ОС Аврора и МП на тестовой группе устройств.

6. УДАЛЕНИЕ ППО

Для удаления ППО необходимо выполнить следующие действия:

6.1. Перейти в каталог со сценариями установки ППО.

6.2. 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 3.2.

6.3. 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 3.1.

7. ВАРИАНТЫ УСТАНОВКИ ПСУ

7.1. Установка ПСУ на один сервер (хост) с другими подсистемами ППО

Данный вариант установки ППО осуществляется по умолчанию.

7.2. Установка ПСУ на отдельный сервер (хост)

Задание адресов серверов (имен хостов), на которые будут установлены подсистемы ППО, осуществляется на этапе настройки ППО (пп. 2.4.2.2). Для того, чтобы установить ПСУ на отдельный сервер, необходимо в конфигурационном 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 конфигурационном файле `inventories/hosts.yml` приведено в п. 2.9.13.

7.3. Отдельная установка ПСУ (установка ПБ и ПСУ)

Для отдельной установки ПСУ необходимо выполнить следующую последовательность действий:

7.3.1. Выполнить последовательность действий, предусмотренную п. 2.2 - 2.5.1.

7.3.2. Установить ПБ и ПСУ с помощью команды:

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

7.3.3. Выполнить последовательность действий, предусмотренную п. 2.5.3 - 2.5.5.

8. КОНФИГУРАЦИОННЫЕ ФАЙЛЫ СЦЕНАРИЕВ УСТАНОВКИ СРЕДЫ ФУНКЦИОНИРОВАНИЯ

8.1. Конфигурационные файлы сценариев установки среды функционирования

8.1.1. Инвентарный файл inventories/hosts.yml

В инвентарном файле `inventories/hosts.yml` задаются адреса серверов приложений и серверов БД, на которые будут установлены компоненты среды функционирования. Описание секций инвентарного файла `inventories/hosts.yml` приведено в таблице (Таблица 24).

Таблица 24

Секция конфигурационного файла	Описание
`all.children.ocs.children.app`	Сервера приложений ППО
`all.children.ocs.children.postgresql.children.postgresql_masters`	СУБД Postgres (главный сервер БД)
`all.children.ocs.children.postgresql.children.postgresql_slaves`	Реплика СУБД Postgres (резервный сервер БД)
`all.children.ocs.children.nginx`	Балансировщик микросервисов «Nginx Web Server»
`all.children.ocs.children.consul`	Система обнаружения сервисов «Consul»
`all.children.ocs.children.consul-template`	Средство управления конфигурациями микросервисов «Consul Template»
`all.children.ocs.children.nats_streaming_server`	Сервис гарантированной доставки сообщений «Nats Streaming Server»
`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

Файл сценария установки для установки среды функционирования ППО на 1 сервере с доменным именем `ocs-app.local` имеет следующий вид:

```
all:
  children:
    ocs:
      children:
        app:
          hosts:
            ocs-app.local:
        postgresql:
          children:
            postgresql_masters:
              hosts:
                ocs-app.local:
            postgresql_slaves:
              hosts:
        nginx:
          children:
            app:
          hosts:
        consul:
          children:
            consul_servers:
              children:
                app:
              hosts:
            consul_agents:
        consul_template:
          children:
            app:
        nats_streaming_server:
          children:
                app:
          hosts:
        redis:
          children:
            redis_masters:
              children:
                app:
              hosts:
            sentinel:
              children:
                app:
              hosts:
```

8.1.2. 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: Nats Streaming Server, 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.

8.1.3. 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: Nats Streaming Server, Consul, СУБД Redis и СУБД PostgreSQL.

9. КОНФИГУРАЦИОННЫЕ ФАЙЛЫ ППО (СЦЕНАРИЕВ УСТАНОВКИ ППО)

9.1. 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 14). 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 14

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.

9.2. Общая информация о конфигурационных файлах сценариев установки ППО

Структура конфигурационных файлов сценариев установки ППО в общем виде приведена на рисунке (Рисунок 15). 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 15

Сценарии установки ППО содержат следующие типы конфигурационных файлов:

– конфигурационный 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.

9.2.1. Конфигурационный файл inventories/hosts.yml

Конфигурационный файл inventories/hosts.yml содержит адреса серверов (имена хостов), на которые установлены (будут установлены) компоненты среды функционирования ППО и подсистемы ППО.

Описание параметров конфигурационного файла inventories/hosts.yml приведено в п. 8.1.1.

9.2.2. 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.

9.2.3. 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
```

9.2.4. 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
```

9.2.5. Конфигурационный файл с паролями и токенами компонентов среды функционирования ППО config/secret.yml

В данном конфигурационном файле задаются пароли и токены Nats Streaming Server, 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.

9.2.6. 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
```

9.2.7. 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.

9.2.8. Конфигурационные файлы окружений

В конфигурационных файлах окружения переопределяются параметры конфигурационных файлов, описанных в п. 9.2.1 - 9.2.8 для заданного окружения. Располагаются данные конфигурационные файлы в каталогах `config/environments/<название окружения>/` и `inventories/<название окружения>/`.

Для переопределения параметра необходимо выполнить следующие действия:

– создать в каталоге `inventories/<название окружения>/` конфигурационный 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.

9.2.9. Порядок работы с конфигурационными файлами сценариев установки ППО

Параметры конфигурационных файлов сценариев установки применяются согласно приоритетам, заданным в таблице (Таблица 25).

Таблица 25

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 25), 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 26).

Таблица 26

Термин/ Сокращение	Расшифровка
БД	База данных
ГИС	Государственная информационная система
ИС	Информационные системы
МП	Мобильное приложение
МУ	Мобильное устройство
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»)
СЗИ	Средство защиты информации
СПО	Специальное программное обеспечение
СУБД	Система управления базами данных
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 МУ, на котором функционируют соответствующие компоненты ППО
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
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
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

Термин/ Сокращение	Расшифровка
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
OIDC	OpenID Connect – уровень аутентификации OAuth 2.0, инфраструктуры авторизации. Контролируется OpenID Foundation
RPM-пакет	Файл формата .rpm, позволяющий устанавливать, удалять и обновлять клиентские 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

Термин/ Сокращение	Расшифровка
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
	измененных	замененных	новых	аннулированных					