Руководство по установке продукта Platform V Backend (#BD)

ОГЛАВЛЕНИЕ

Руководство по установке ... 9
Системные требования ... 9
Требования к окружению ... 9
СПО ... 9
Платформенные зависимости ... 11
Выбор способа установки ... 13
Сценарий установки ... 13
Руководство по установке компонента Автонумерация (Autonum) (ANUM) ... 15
Системные требования ... 15
Системное программное обеспечение ... 15
Платформенные зависимости ... 16
Аппаратные требования ... 18
Выбор способа установки ... 19
Подготовка окружения ... 19
Подключение namespace к компоненту Управление политиками ... 19
Выпуск и установка сертификатов для Ingress/Egress ... 19
Настройка OTT ... 20
Инициализация БД ... 21
Подготовка окружения для автоматизированной установки с использованием Deploy Tools ... 23
Установка ... 23
Ручная установка сервиса ... 23
Автоматизированная установка сервиса с использованием Deploy Tools ... 54
Обновление ... 54
Удаление ... 55
Проверка работоспособности ... 55
Через вызов API ... 55
Через UI среды контейнеризации ... 59
Откат ... 59
Часто встречающиеся проблемы и пути их устранения ... 59
Чек-лист валидации установки ... 59
Чек-лист проверки работоспособности интеграции ... 60
Руководство по установке компонента компонента Генерация ID (Sberflake ID) (SNOW) ... 62

Системные требования ... 62
Системное программное обеспечение ... 62
Платформенные зависимости ... 63
Аппаратные требования ... 65
Выбор способа установки ... 66
Подготовка окружения ... 66
Настройка OTT ... 66
Инициализация БД ... 68
Подготовка окружения для автоматизированной установки с использованием Deploy Tools ... 69
Установка ... 70
Ручная установка сервиса ... 70
Автоматизированная установка сервиса с использованием Deploy Tools ... 100
Обновление ... 100
Удаление ... 100
Проверка работоспособности ... 101
Через вызов API ... 101
Через UI среды контейнеризации ... 104
Откат ... 104
Часто встречающиеся проблемы и пути их устранения ... 104
Чек-лист валидации установки ... 105
Чек-лист проверки работоспособности интеграции ... 105
Руководство по установке компонента PACMAN (CFGA) ... 107
Термины и сокращения ... 107
Назначение документа ... 113
Системные требования ... 113
Системное программное обеспечение ... 113
Платформенные зависимости ... 115
Аппаратные требования ... 118
Выбор способа установки ... 119
Состав дистрибутива ... 120
Установка ... 121
Выбор способа установки ... 121
Раздельная установка модулей компонента ... 121

Установка дистрибутива без использования Pipeline DevOps 122
Автоматизированная установка сервиса с использованием Deploy Tools 188
Настройка сертификатов для БД PostgreSQL для работы компонента «Управление конфигурациями» 191
Настройка SSL для Rest сервиса компонента «Управление конфигурациями» 197
Параметры для безопасной передачи паролей 199
Глобальные параметры, используемые при раскатке компонента 200
Настройка подключения к брокерам Kafka 214
Настройка подключения Аудита 214
Настройка подключения Мониторинга 215
Настройка подключения Журналирования 215
Настройка подключения Аутентификации/Авторизации 216
Настройка работы с OTT компонента «Управление параметрами» 224
Настройка подключения Агента конфигурации (для доставки конфигурации в ConfigMap) 236
Особенности старта Приложения-потребителя при подключении Агента конфигурации в виде отдельного модуля 253
Дополнительные настройки работы с компонентом «Управление конфигурациями» 253
Инструкция по сборке из разделенного дистрибутива 253
Обновление 255
Удаление 255
Проверка работоспособности 256
Проверка работоспособности компонента «Управление конфигурациями» 256
Проверка работоспособности компонента «Управление параметрами» 256
Откат 257
Часто встречающиеся проблемы и пути их устранения 257
Часто встречающиеся проблемы и пути их устранения установки компонента «Управление конфигурациями» 257
Часто встречающиеся проблемы и пути их устранения установки компонента «Управление параметрами» 258
Чек-лист валидации установки 260
Чек-лист валидации установки компонента «Управление конфигурациями» 260
Чек-лист валидации установки компонента «Управление параметрами» 260
Руководство по установке компонента Прикладной журнал (APLJ) 261
Термины и определения 261

Системные требования ... 263
Системное программное обеспечение ... 263
Платформенные зависимости ... 265
Аппаратные требования ... 266
Выбор способа установки ... 267
Установка ... 267
Ручная установка сервиса ... 267
Автоматизированная установка сервиса с использованием Deploy Tools ... 275
Обновление ... 276
Проверка работоспособности ... 276
Откат ... 277
Часто встречающиеся проблемы и способы их решения ... 277
Чек-лист валидации установки ... 277
Описание настроек ... 278
Описание настроек в application-journal-config-struct.xml ... 278
Руководство по установке компонента Сервис шардирования приложений (ASPX) ... 288
Системные требования ... 288
Системное программное обеспечение ... 288
Платформенные зависимости ... 289
Аппаратные требования ... 291
ASR (CPU, RAM) ... 291
ASM (CPU, RAM) ... 291
Состав дистрибутива ... 292
Выбор способа установки ... 292
Подготовка окружения ... 293
Предварительная настройка стенда ... 293
Сборка образа контейнера ... 295
Установка ... 297
Схема установки компонент ... 297
Ручная установка ... 298
Автоматизированная установка сервиса с использованием DevOps Tools ... 318
Требования по безопасности ... 318
Обновление ... 322
Проверка работоспособности ... 322

Проверка работоспособности AppShard Manager 322
Проверка работоспособности AppShard Router 323
Чек-лист проверки работоспособности интеграции 325
Проверка работоспособности компонента Журналирование 326
Откат 326
Часто встречающиеся проблемы и пути их устранения 326
Чек-лист валидации установки 327
Чек-лист валидации установки AppShard Manager 327
Чек-лист валидации установки AppShard Router 328
Руководство по установке компонента Сервис межкластерной индексации (CCIX) 330
Список терминов и сокращений 330
Назначение документа 330
Системные требования 330
Системное программное обеспечение 330
Платформенные зависимости 332
Установка 333
Ручная установка компонента 333
Автоматизированная установка компонента с использованием DevOps Tools 338
Обновление 343
Удаление 343
Проверка работоспособности 344
Откат 346
Часто встречающиеся проблемы и пути их устранения 349
Чек-лист валидации установки 349
Руководство по установке One-Time Password (OTP) / OTT (OTTS) 350
Системные требования 351
Системное программное обеспечение 351
Платформенные зависимости 353
Аппаратные требования 355
Установка 356
Сервер OTT 356
Клиент OTT (OTT Sidecar) 365
Установка EJBCA на сервере приложений Wildfly (опционально) 372
Configure EjbcaSSLRealm 379

Проверить статус сервера, удостовериться, что нет ошибок ... 379
Ожидаемый результат ... 379
Reload ... 380
Обновление ... 384
Проверка работоспособности ... 384
Откат ... 384
Часто встречающиеся проблемы и пути их устранения ... 384
Чек-лист валидации установки ... 384
Руководство по установке компонента ЭЦП (криптосервис) (DIGS) ... 385
Системные требования ... 385
Список общих системных требований к ОС и развертыванию ... 385
Платформенные зависимости ... 385
Sizing Guide ... 388
Нагрузка ... 388
Установка ... 389
Руководство по ручной установке ... 389
Проверка работоспособности ... 403
Автоматизированная установка сервиса с использованием Deploy Tools ... 403
Обновление ... 406
Откат ... 406
Часто встречающиеся проблемы и пути их устранения ... 407
Чек-лист валидации установки ... 407
Примеры конфигурации настроек Istio для mtls ... 409
Egress ... 409
Ingress ... 411
Руководство по установке компонента Объединенный сервис авторизации (OCA) (AUTZ) ... 412
Системные требования ... 412
Системное программное обеспечение ... 412
Платформенные зависимости ... 415
Аппаратные требования ... 419
Установка ... 420
Ручная установка ... 420
Автоматизированная установка посредством Platform V DevOps Tools ... 421
Инструкция по развертыванию Platform V Pangolin SE, (далее – Platform V Pangolin SE) 423

Создание тестового пользователя (администратора) в базе данных ... 426
Настройка интеграции с технологическими сервисами и компонентами ... 428
Интеграция с компонентом KeyCloak.SE ... 428
Интеграция с компонентом Управление настройками PACMAN ... 429
Интеграция с компонентом Журналирование продукта Platform V Monitor ... 429
Интеграция с компонентом Прикладной мониторинг продукта Platform V Monitor ... 429
Интеграция с компонентом One-Time Password (OTP) / OTT ... 430
Интеграция с компонентом Аудит продукта Platform V Audit SE ... 434
Обновление ... 435
Проверка работоспособности ... 435
Откат ... 436
Часто встречающиеся проблемы и пути их устранения ... 436
Чек-лист валидации установки ... 436

Руководство по установке

Документ является руководством по установке программного продукта Platform V Backend (#BD)

Процесс установки (развертывания) содержит следующие логические шаги:

1. Получение всего необходимого для развертывания (ИП — функциональная подсистема, приложение (ФП) и скрипты развертывания, конфигурации, документация).
2. Подготовка к восстановлению уже развернутого приложения (опциональный шаг, зависящий от архитектуры ФП).
3. Подготовка среды для развертывания (создание и настройка стенда – все ли папки созданы, права выданы, все ли скрипты на месте и готовы для развертывания).
4. Выполнение развертывания.
5. Проверки после развертывания (проверка успешности развертывания и проверка работоспособности).
6. Если предыдущий шаг не успешен — восстановление приложения на предыдущую версию (опциональный шаг, зависящий от архитектуры ФП).

Некоторые шаги опциональны и могут отключаться в зависимости от среды, на которую происходит развертывание, и в зависимости от требований к конкретной ФП.

Системные требования

В данном разделе обозначен общий вид на системные требования для программного продукта Platform V Backend (#BD). Детальные системные требования перечислены в документации каждого компонента, входящего в состав программного продукта. Комплекс технических средств (далее — КТС) должен обеспечивать работу всех компонентов готового решения.

На выделенных ресурсах устанавливается требуемое программное обеспечение (далее — СПО). Подробнее — Требования к окружению.

Настройки безопасности окружения выбираются, исходя из характера обрабатываемой в каждом сервисе информации и иных требований информационной безопасности (далее по тексту – ИБ), предъявляемых к каждому сервису (ФП).

Требования к окружению

СПО

Информация ниже является обзором требований (подробнее актуальные требования представлены в документации к каждому компоненту).

Код компонента	Имя компонента	ОС Альт 8 СП	Kubernetes	Apache Kafka (опционально)	Apache Zookeeper (опционально)	Kafka (рекомендован Platform V Corax)	PostgreSQL (рекомендован Platform V Pangolin SE)	Red Hat OpenShift (опционально)	OpenJDK (Java SE)	Red Hat Enterprise Linux (опционально)
ANUM	Автонумерация (Autonum)	10.0 и выше	1.0 и выше	-	-	-	4.4, 4.6, 5.1	4.8	8 (1.8), 11	7, 8
APLJ	Прикладной журнал	8 и выше	1.21 и выше	2.2.X, 2.5.X	3.4.13, 4.8.X	2.7.2-v5.1	4.4, 4.6, 5.1	4.8	8 (1.8), 11	7, 8
ASPX	Сервис шардирования приложений	10.0 и выше	1.0 и выше	-	-	-	-	4.8	8 (1.8), 11	-
AUTZ	Объединенный сервис авторизации	10.0 и выше	1.0 и выше	-	-	-	4.4, 4.6, 5.1	4.8	8 (1.8), 11	7, 8
CCIX	Сервис межкластерной индексации	10.0 и выше	1.18 и выше	-	-	-	4.4, 4.6, 5.1	4.8	8 (1.8), 11	7, 8
CCOR	Ядро КБТ (Координатор бизнес-транзакций)	10.0 и выше	1.9 и выше	-	-	-	-	4.8	8 (1.8), 11	-
CFGA	PACMAN	10.0 и выше	1.0 и выше	-	-	-	4.4, 4.6, 5.1	4.8	8 (1.8), 11	7, 8
DIGS	ЭЦП (криптосервис)	10.0 и выше	1.0 и выше	-	-	-	-	4.8	8 (1.8), 11	-
OTTS	One-Time Password	10.0 и выше	1.18 и выше	-	-	-	4.4, 4.6, 5.1	4.8	8 (1.8), 11	7, 8
SNOW	Генерация ID (Sberflake ID)	10.0 и выше	1.0 и выше	-	-	-	4.4, 4.6, 5.1	4.8	8 (1.8), 11	7, 8

Платформенные зависимости

Для настройки, контроля и функционирования компонентов продукта реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код продукта	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Audit SE	AUD	1.1 и выше	AUDT Аудит	опционально	Сервис для аудирования событий
Platform V IAM SE	IAM	1.1 и выше	-	опционально	Сервис для авторизации доступа пользователей на основе проверки ролей и прав доступа, а также на основе атрибутов субъектов и объектов доступа
Platform V Monitor	OPM	2.0	-	опционально	Сервис для хранения лог-файлов
Platform V Pangolin SE	PSQ	4.6.0. и выше	-	опционально	Система управления базами данных, основанная на PostgreSQL

Наименование продукта	Код продукта	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V DevOps Tools	DOT	1.2	-	опционально	Сервис для развертывания и обновления компонентов Платформы и приложений потребителей, для настройки и обслуживания инфраструктуры Платформы
Platform V Synapse Service Mesh	SSM	2.0 и выше	-	опционально	Панель управления, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в среде контейнеризации Kubernetes

Наименование продукта	Код продукта	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Corax	KFK	5.1 и выше	KFKA Kafka Sber Edition	опционально	Программный брокер сообщений, представляющий собой распределенную, отказоустойчивую, реплицированную и легко масштабируемую систему передачи сообщений, рассчитанную на высокую пропускную способность

Выбор способа установки

Процедура развертывания описана в документации каждого компонента, входящего в состав Platform V Backend, установка возможна одним из двух способов:

- ручная установка;
- автоматизированная установка с помощью Platform V DevOps Tools (DOT), а именно – с помощью компонента Deploy Tools (CDJE).

Сценарий установки

Подробная установка конкретного программного компонента Platform V Backend представлена в документации к каждому компоненту.

Установка (сборка) готового решения должна выполняться последовательно в соответствии с общей схемой процесса развертывания в следующем порядке:

1. CFGA. Pacman. Стартовая инсталляция выполняется в *служебном* режиме без использования сервиса Авторизации. Загрузка структур конфигураций и параметров ФП, входящих в состав Platform V Backend;
2. OTTS. One-Time Password (OTP) / OTT;
3. AUTZ. Объединённый сервис авторизации (OCA);
4. APLJ. Прикладной журнал;
5. SNOW. Генератор ID;
6. ANUM. Автонумерация;

7. ASPX. Сервис шардирования приложений;
8. CCIX. Сервис межкластерной индексации;
9. DIGS. ЭЦП (криптосервис);
10. CCOR. Ядро КБТ. КБТ поставляется как базовый образ для сборки образа со сценарием и не устанавливается как отдельный продукт.

Руководство по установке компонента Автонумерация (Autonum) (ANUM)

Системные требования

Настройки безопасности окружения и перечень платформенных (дополнительных внешних) продуктов, используемых для установки, настройки и контроля в конечной информационной системе (далее — ИС), выбираются при разработке конечной ИС, исходя из характера обрабатываемой в ней информации и иных требований информационной безопасности (далее — ИБ), предъявляемых к ней.

Системное программное обеспечение

Ниже представлены категории системного программного обеспечения, которые обязательны для установки, настройки, контроля и функционирования компонента. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО «СберТех» (маркировка «Рекомендовано» в столбце Комментарий). Клиенту необходимо выбрать один из продуктов в каждой категории, исходя из условий использования конечной ИС.

Операционная система

Наименование	Версия	Комментарий
ОС Альт 8 СП	10.0 и выше	Рекомендовано
Red Hat Enterprise Linux	–	Опционально

Среда контейнеризации

Необходимое окружение для разворачивания компонента Автонумерация (Autonum) — Kubernetes версии 1.0 и выше или OpenShift версии 4.2 и выше (далее — среда контейнеризации).

Наименование	Версия	Комментарий
Kubernetes	1.0 и выше	Рекомендовано
Red Hat OpenShift	4.2 и выше	Опционально

Java-машина

Наименование	Версия	Комментарий
OpenJDK	1.8 и выше	Рекомендовано
OracleJDK	—	Опционально

Система управления базами данных (СУБД)

Наименование	Версия	Комментарий
PostgreSQL	11 и выше	Рекомендовано

Правообладателем АО «СберТех» также рекомендован Platform V Pangolin SE, см. раздел «Платформенные зависимости».

Инструмент управления программными проектами

Наименование	Версия	Комментарий
Apache Maven	3.6.1 и выше	Рекомендовано

Сервер приложений

Наименование	Версия	Комментарий
Apache Tomcat	9.0.39 и выше	Рекомендовано

Система для проверки функционирования сервисов (опционально)

Наименование	Версия	Комментарий
SYMON	1.3.2 и выше	Рекомендовано

Сервис интеграции и оркестрации микросервисов в облаке

Наименование	Версия	Комментарий
Istio	1.12 и выше	Рекомендовано

Правообладателем АО «СберТех» также рекомендован Platform V Synapse Service Mesh, см. раздел «Платформенные зависимости».

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Audit SE	AUD	1.1	AUDT Аудит	опционально	Сервис для аудирования событий
Platform V Monitor	OPM	3.27.11	LOGA Журналирование	опционально	Сервис для хранения лог-файлов
Platform V Monitor	OPM	6.0.0-02	MONA Объединенный мониторинг Unimon	опционально	Сервис для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения
Platform V Pangolin SE	PSQ	4.6.0. и выше	–	опционально	Система управления базами данных, основанная на PostgreSQL
Platform V DevOps Tools	DOT	1.2	–	опционально	Сервис для развертывания и обновления компонентов Платформы и приложений потребителей, для настройки и обслуживания инфраструктуры Платформы

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Synapse Service Mesh	SSM	1.12 и выше	POLM Управление политиками	опционально	Панель, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в среде контейнеризации Kubernetes

Примечание:

- **Обязательно** — компонент или продукт необходим для функционирования сервиса.
- **Опционально** — необязательный для функционирования сервиса компонент или продукт, рекомендуется его установка, но допускается использование аналога других производителей.

Аппаратные требования

Для установки компонента/продукта требуется следующая конфигурация аппаратного обеспечения:

- Требования к производительности:

Описание	Параметры
Рекомендуемый минимальный КТС для namespace	8 CPU / 32GB RAM

- Требования к базе данных (далее по тексту — БД):

Описание	Параметры
Минимальные требования к КТС для разворачивания сервиса	2 Core CPU / 8GB RAM / 100GB HDD
Рекомендуемые требования к КТС для разворачивания сервиса	16 Core CPU / 128GB RAM / 1200GB HDD

Выбор способа установки

В зависимости от способа развертывания сервиса воспользуйтесь одной из инструкций:

- целевая ручная установка сервиса;
- опциональная автоматизированная установка сервиса с использованием продукта Platform V DevOps Tools.

Подготовка окружения

Вне зависимости от выбора способа установки выполните шаги, описанные в разделах ниже. Однако для автоматизированной установки сервиса с использованием программного продукта Platform V DevOps Tools дополнительно выполните настройку окружения, описанную в разделе «Подготовка окружения для автоматизированной установки с использованием Deploy Tools».

Подключение namespace к компоненту Управление политиками

Подключение namespace к компоненту Управление политиками (POLM, версия 1.12 и выше) продукта Platform V Synapse Service Mesh (SSM) осуществляется в соответствии с Руководством прикладного разработчика на компонент Управление политиками.

Выпуск и установка сертификатов для Ingress/Egress

Используя шаблоны из дистрибутива /autonum/other/secrets/, выпустите и установите сертификаты Ingress/Egress:

- `istio-egressgateway-ca-certs`;
- `istio-egressgateway-certs`;
- `istio-ingressgateway-ca-certs`;
- `istio-ingressgateway-certs`.

После чего будет создан набор необходимых артефактов, в том числе секреты с сертификатами `cert`, `ca-cert`.

Созданные секреты содержат, в том числе, наименования файлов: `tls.crt`, `tls.key`, `ca-chain.cert.pem` (вся цепочка сертификатов).

В секретах `istio-ingressgateway-ca-certs` и `istio-egressgateway-ca-certs` должен присутствовать один файл `ca-chain.cert.pem` со всей цепочкой сертификатов. Если в секрете указаны два отдельных сертификата, то вызовы через Ingress по HTTPS не будут работать.

Возможен вариант, когда в секретах в файле `ca-chain.cert.pem` содержится вся цепочка сертификатов (т.е. добавлено содержимое из `ca.pem`), но сам файл `ca.pem` не удален. В этом случае все будет работать корректно.

Настройка OTT

Для настройки OTT:

- получите сертификаты приложения, публичный ключ OTT Service и выполните их подключение;
- настройте ConfigMap для OTT-sidecar.

Получение сертификатов приложения, публичного ключа OTT Service и их подключение

Для получения сертификатов приложения, публичного ключа OTT Service и их подключения:

1. При помощи keytool сгенерируйте key-store с ключами (`autonum_cert.p12`) по алгоритму SHA256withECDSA и длиной ключа 256 бит.

Примеры команд

```
keytool -genkey -keyalg EC -sigalg SHA256withECDSA -keystore
autonum.p12 -storetype PKCS12 -keysize 256 -dname
"CN=autonum" -alias autonum keytool -certreq -keyalg EC -
sigalg SHA256withECDSA -keystore autonum.p12 -storetype
PKCS12 -alias autonum > autonum_cert_req.pem
```

2. Отправьте файл `autonum_cert_req.pem` администратору OTT в заявке на генерацию сертификата для модуля. Администратор OTT подписывает/генерирует ключи, высылает их обратно.
3. Импортируйте полученные от администратора OTT ключи в `key-store autonum_cert.p12`.

Примеры команд

```
keytool -import -keystore autonum.p12 -storetype PKCS12 -
file SberbankPlatformCA_EC.pem -alias SberbankPlatformCA_EC
keytool -import -keystore autonum.p12 -storetype PKCS12 -
file autonum.pem -alias autonum
```

4. Создайте два секрета:

- `ott-certs` — с сертификатами (autonum.p12: >- , ott_service_truststore.p12: >-)

  Важно! Чтобы все корректно добавилось, необходимо сконвертировать содержимое файла с ключами в Base64. Далее — полученную строку вставить в yml-файл. Либо выполнить команду:

```
oc create secret generic ott-certs --from-
file=ott_service_truststore.p12=./sigma_ott_trust.p12 --
from-file=autonum.p12=./autonum.p12
```

- `ott-passwords` — с паролями к ключам:
  - `OTT_CERTSTORE_PRIVATE_KEY_PWD` — пароль для чтения ключа;
  - `OTT_CERTSTORE_PWD` — пароль для чтения сертификата;
  - `OTT_TRUST_STORE_PWD` — пароль для чтения ott_service_truststore.p12.

ConfigMap для OTT-sidecar

Для работы OTT-sidecar необходимо наличие в дистрибутиве ConfigMap с настройками сервиса. ConfigMap создается на этапе установки сервиса.

Конфигурирование параметров ConfigMap для OTT-sidecar выполняется в файле `sberflake.istio.all.conf`.

Инициализация БД

Для инициализации БД от ТУЗ администратора БД выполните следующие действия:

1. Очистите namespace согласно действиям, приведенным в разделе «Удаление».
2. Создайте схему в БД:

- для микросервиса Counters:

```
CREATE SCHEMA IF NOT EXISTS autonum_counters AUTHORIZATION
as_admin; ALTER DEFAULT PRIVILEGES for user as_admin IN
SCHEMA autonum_counters GRANT SELECT, INSERT, UPDATE, DELETE
ON TABLES TO autonum_counters; ALTER DEFAULT PRIVILEGES for
user as_admin IN SCHEMA autonum_counters GRANT USAGE, SELECT
ON SEQUENCES to autonum_counters;
```

где `as_admin` — роль, выдаваемая администраторам БД; `autonum_counters` — схема БД микросервиса Counters.

- для микросервиса Patterns:

```
CREATE SCHEMA IF NOT EXISTS autonum_patterns AUTHORIZATION
as_admin; ALTER DEFAULT PRIVILEGES for user as_admin IN
SCHEMA autonum_patterns GRANT SELECT, INSERT, UPDATE, DELETE
ON TABLES TO autonum_patterns; ALTER DEFAULT PRIVILEGES for
user as_admin IN SCHEMA autonum_patterns GRANT USAGE, SELECT
ON SEQUENCES to autonum_patterns;
```

где `autonum_patterns` — схема БД микросервиса Patterns.

3. Разархивируйте postgres-db.zip.
4. Выполните скрипты по созданию табличных пространств из каталога `/package/db/postgres-db/database-scripts`:

- 1-create-folders-for-tablespaces.sh;
- 2-create-tablespaces.sh;
- 3-check-tablespaces-are-created.sh;
- 4-grant-privileges-to-user.sh.

5. Выполните скрипты liquibase согласно `USAGE` (если у скриптов нет права на запуск, то выполните команду `chmod 777 autonum*.*`).
6. Выполните скрипты для хостов с указанием параметров для основной (MAIN) и репликационной (SI) БД:

```
./autonum-counters_update.sh <db_host> <db_port> <db_name>
<db_schema> <db_username> <db_password> ./autonum-
patterns_update.sh <db_host> <db_port> <db_name> <db_schema>
<db_username> <db_password>
```

7. Предоставьте права доступа пользователям autonum_patterns и autonum_counters, выполнив команду:

```
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA
autonum_counters TO autonum_counters;    GRANT ALL
PRIVILEGES ON ALL SEQUENCES IN SCHEMA autonum_patterns TO
autonum_patterns;    GRANT USAGE ON SCHEMA autonum_counters
TO autonum_counters;    GRANT USAGE ON SCHEMA
autonum_patterns TO autonum_patterns;
```

где `ALL` — права на использование (Usage) и создание (Create) объектов в БД.

8. Измените параметр `search_path` для основной и репликационной БД, выполнив команду:

```
ALTER ROLE autonum_counters IN DATABASE <db_name> SET
search_path TO "$user";    ALTER ROLE autonum_patterns IN
DATABASE <db_name> SET search_path TO "$user";
```

где `$user` — имя пользователя БД от соответствующих ТУЗ, указанных в п.5; `<db_name>` — имя основной или репликационной БД для сервисов autonum_counters и autonum_patterns.

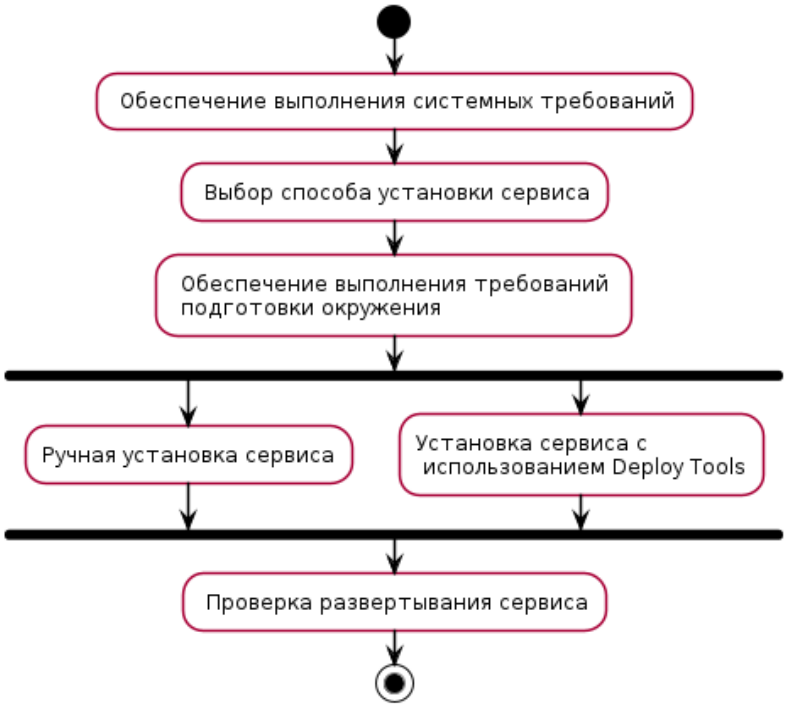
Подготовка окружения для автоматизированной установки с использованием Deploy Tools

Для автоматизированной установки с использованием компонента Deploy Tools (CDJE) продукта Platform V DevOps Tools (DOT) дополнительно требуется:

- наличие у сервиса пространства в репозитории и прав на создание репозиториев в нем;
- наличие у сервиса области в Jenkins с настроенными учетными данными (логин/пароль) для авторизации под ТУЗ, позволяющая загружать артефакты в целевой репозиторий хранилища.

Параметры настройки располагаются в файлах в дистрибутиве по пути: package/conf/config/parameters.

Установка



Ручная установка сервиса

Перед началом установки убедитесь, что выполнена подготовка окружения.

Для ручной установки сервиса Автонумерация выполните следующие действия:

- обновите схему БД;
- настройте конфигурационные файлы.

Обновление схемы БД

Схема данных БД может быть обновлена при условии выполненной первичной инициализации БД.

Для обновления схемы БД до текущей версии выполните следующие действия:

1. Разархивируйте postgres-db.zip.
2. В терминале выполните команду для микросервиса Counters

```
./package/db/postgres-db/nix/autonum-counters_update.sh
<db_host> <db_port> <master|standin>"
```

Пример команды:

```
./package/db/postgres-db/nix/autonum-counters_update.sh
10.53.220.12 10101 master
```

2. В терминале выполните команду для микросервиса Patterns

```
./package/db/postgres-db.zip/nix/autonum-patterns_update.sh
<db_host> <db_port> <master/standin>"
```

Пример команды:

```
./package/db/postgres-db.zip/nix/autonum-patterns_update.sh
10.53.220.12 10201 master
```

Настройка конфигурационных файлов

Для ручной установки сервиса в среде контейнеризации необходимо во всех файлах конфигурации заменить все плейсхолдеры на реальные значения.

Перечень конфигурационных файлов для развертывания в среде контейнеризации Kubernetes:

- /package/conf/k8s/base/istio-configs/ip-service-autonum-counters-db-master/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-counters-db-master/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-counters-db-master/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-counters-db-standin/anum-vs-ip-service.yaml;

- /package/conf/k8s/base/istio-configs/ip-service-autonum-counters-db-standin/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-counters-db-standin/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-patterns-db-master/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-patterns-db-master/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-patterns-db-master/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-patterns-db-standin/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-patterns-db-standin/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-patterns-db-standin/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-1/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-1/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-1/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-2/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-2/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-2/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-3/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-3/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-3/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-4/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-4/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-4/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-5/anum-vs-ip-service.yaml;

- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-5/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-logging-5/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-journal-1/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-journal-1/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-journal-1/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-journal-2/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-journal-2/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-journal-2/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-journal-3/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-journal-3/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-journal-3/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-route-ingress-gateway-mtls-health.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-route-ingress-gateway-db-loader-health.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-gw-ingress-mtls-health-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-ef-ingress-symon-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-ef-configurer-ingress-symon-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-gateway/anum-gw-ingress-mtls-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-gateway/anum-gw-ingress-http-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/dc.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-ingress-ott-gateway-logback.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-ingress-ott-gateway-istio-basic.yaml;

- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-ingress-ott-gateway-env.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/anum-svc-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/anum-gw-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/anum-ef-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-monitoring/anum-vs-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-monitoring/anum-se-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-monitoring/anum-gw-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-monitoring/anum-ef-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-monitoring/anum-dr-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-logger/anum-vs-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-logger/anum-se-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-logger/anum-gw-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-logger/anum-ef-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-logger/anum-dr-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-audit/anum-vs-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-audit/anum-se-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-audit/anum-gw-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-audit/anum-ef-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/https-service-audit/anum-dr-https-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/http-service-monitoring/anum-vs-http-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/http-service-monitoring/anum-se-http-service.yaml;

- /package/conf/k8s/base/openshift/istio-configs/http-service-monitoring/anum-gw-http-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/http-service-logger/anum-vs-http-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/http-service-logger/anum-se-http-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/http-service-logger/anum-gw-http-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/http-service-audit/anum-vs-http-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/http-service- audit/anum-se-http-service.yaml;
- /package/conf/k8s/base/openshift/istio-configs/http-service-audit/anum-gw-http-service.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/dc.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/configmaps/anum-cm-egress-ott-gateway-logback.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-egress-ott-gateway-istio-basic.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-egress-ott-gateway-env.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/anum-svc-egress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-standin/anum-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-standin/anum-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-standin/anum-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-master/anum-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-master/anum-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-master/anum-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-master/anum-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-master/anum-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-master/anum-gw-dns-service.yaml;

- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-standin/anum-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-standin/anum-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-standin/anum-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-1/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-1/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-1/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-2/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-2/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-2/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-3/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-3/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-3/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-4/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-4/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-4/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-5/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-5/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-5/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-1/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-1/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-1/anum-gw-ip-service.yaml;

- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-2/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-2/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-2/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-3/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-3/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-3/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/autonum-patterns/testvs/anum-vs-patterns-service-test.yaml;
- /package/conf/k8s/base/autonum-patterns/testvs/anum-vs-patterns-service-deprecated-test.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-logging-passwords.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-logging-certs.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-journal-passwords.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-journal-certs.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-db-passwords.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-db-certs.yaml;
- /package/conf/k8s/base/autonum-patterns/dc.yaml;
- /package/conf/k8s/base/autonum-patterns/configmaps/anum-cm-patterns-service.yaml;
- /package/conf/k8s/base/autonum-patterns/configmaps/anum-cm-patterns-service-logback.yaml;
- /package/conf/k8s/base/autonum-patterns/configmaps/anum-cm-patterns-service-fluent-bit.yaml;
- /package/conf/k8s/base/autonum-patterns/configmaps/anum-cm-patterns-service-env.yaml;
- /package/conf/k8s/base/autonum-patterns/anum-vs-patterns-service.yaml;
- /package/conf/k8s/base/autonum-patterns/anum-vs-patterns-service-deprecated.yaml;
- /package/conf/k8s/base/autonum-patterns/anum-svc-patterns-service.yaml;
- /package/conf/k8s/base/autonum-patterns/anum-pdb-service-patterns.yaml;
- /package/conf/k8s/base/autonum-counters/testvs/anum-vs-counters-service-test.yaml;
- /package/conf/k8s/base/autonum-counters/testvs/anum-vs-counters-service-deprecated-test.yaml;

- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-logging-passwords.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-logging-certs.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-journal-passwords.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-journal-certs.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-db-passwords.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-db-certs.yaml;
- /package/conf/k8s/base/autonum-counters/dc.yaml;
- /package/conf/k8s/base/autonum-counters/configmaps/anum-cm-counters-service-logback.yaml;
- /package/conf/k8s/base/autonum-counters/configmaps/anum-cm-counters-service-fluent-bit.yaml;
- /package/conf/k8s/base/autonum-counters/configmaps/anum-cm-counters-service-env.yaml;
- /package/conf/k8s/base/autonum-counters /anum-vs-counters-service.yaml;
- /package/conf/k8s/base/autonum-counters/anum-vs-counters-service-deprecated.yaml;
- /package/conf/k8s/base/autonum-counters/anum-svc-counters-service.yaml;
- /package/conf/k8s/base/autonum-counters/anum-pdb-service-counters.yaml;
- /package/conf/k8s/base/autonum-configurer/testvs/anum-vs-configurer-test.yaml;
- /package/conf/k8s/base/autonum-configurer/testvs/anum-vs-configurer-service-deprecated-test.yaml;
- /package/conf/k8s/base/autonum-configurer/secrets/anum-secret-configurer-logging-passwords.yaml;
- /package/conf/k8s/base/autonum-configurer/secrets/anum-secret-configurer-logging-certs.yaml;
- /package/conf/k8s/base/autonum-configurer/secrets/anum-secret-configurer-db-passwords.yaml;
- /package/conf/k8s/base/autonum-configurer/secrets/anum-secret-configurer-db-certs.yaml;
- /package/conf/k8s/base/autonum-configurer/dc.yaml;
- /package/conf/k8s/base/autonum-configurer/configmaps/anum-cm-configurer.yaml;
- /package/conf/k8s/base/autonum-configurer/configmaps/anum-cm-configurer-logback.yaml;
- /package/conf/k8s/base/autonum-configurer/configmaps/anum-cm-configurer-fluent-bit.yaml;
- /package/conf/k8s/base/autonum-configurer/configmaps/anum-cm-configurer-env.yaml;
- /package/conf/k8s/base/autonum-configurer/anum-vs-configurer.yaml;

- /package/conf/k8s/base/autonum-configurer/anum-vs-configurer-service-deprecated.yaml;
- /package/conf/k8s/base/autonum-configurer/anum-svc-configurer.yaml;
- /package/conf/k8s/base/autonum-configurer/anum-pdb-configurer.yaml;
- /package/conf/k8s/base/prometheus/dc.yaml;
- /package/conf/k8s/base/prometheus/configmaps/anum-cm-prometheus.yaml;
- /package/conf/k8s/base/prometheus/anum-se-prometheus-kubeapi-ext.yaml.

Перечень конфигурационных файлов для развертывания в среде контейнеризации OpenShift:

- /package/conf/k8s/base/overrides/openshift/autonum-patterns/testroute/anum-route-patterns-service-http.yaml;
- /package/conf/k8s/base/overrides/openshift/autonum-counters/testroute/anum-route-counters-service-http.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/anum-route-ingress-ott-gateway-db-loader-ott.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/anum-route-ingress-ott-gateway-db-loader-mtls.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/anum-route-ingress-ott-gateway-ott.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/anum-route-ingress-ott-gateway-db-loader.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/anum-route-ingress-ott-gateway-mlts.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/anum-route-ingress-ott-gateway-http.yaml
- /package/conf/k8s/base/overrides/openshift/autonum-configurer/testroute/anum-route-configurer-http.yaml;
- /package/conf/k8s/base/autonum-patterns/overrides/openshift/testroute/anum-route-patterns-service-http.yaml;
- /package/conf/k8s/base/autonum-counters/overrides/openshift/testroute/anum-route-counters-service-http.yaml;
- /package/conf/k8s/base/autonum-configurer/overrides/openshift/testroute/anum-route-configurer-http.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-counters-db-master/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-counters-db-master/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-autonum-counters-db-master/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-counters-db-standin/anum-vs-ip-service.yaml;

- /package/conf/ k8s/base/istio-configs/ip-service-autonum-counters-db-standin/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-counters-db-standin/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-patterns-db-master/anum-vs-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-patterns-db-master/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-patterns-db-master/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-patterns-db-standin/anum-vs-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-patterns-db-standin/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-patterns-db-standin/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-1/anum-vs-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-1/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-1/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-2/anum-vs-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-2/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-2/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-3/anum-vs-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-3/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-3/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-4/anum-vs-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-4/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-4/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-5/anum-vs-ip-service.yaml;

- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-5/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-logging-5/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-journal-1/anum-vs-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-journal-1/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-journal-1/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-journal-2/anum-vs-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-journal-2/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-journal-2/anum-gw-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-journal-3/anum-vs-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-journal-3/anum-se-ip-service.yaml;
- /package/conf/ k8s/base/istio-configs/ip-service-autonum-journal-3/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-route-ingress-gateway-mlts-health.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-route-ingress-gateway-db-loader-health.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-gw-ingress-mtls-health-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-ef-ingress-symon-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/anum-ef-configurer-ingress-symon-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-gateway/anum-gw-ingress-mtls-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-gateway/anum-gw-ingress-http-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/dc.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-ingress-ott-gateway-logback.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-ingress-ott-gateway-istio-basic.yaml;

- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-ingress-ott-gateway-env.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/anum-svc-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/anum-gw-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/anum-ef-ingress-ott-gateway.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-monitoring/anum-vs-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-monitoring/anum-se-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-monitoring/anum-gw-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-monitoring/anum-ef-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-monitoring/anum-dr-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-logger/anum-vs-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-logger/anum-se-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-logger/anum-gw-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-logger/anum-ef-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-logger/anum-dr-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-audit/anum-vs-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-audit/anum-se-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-audit/anum-gw-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-audit/anum-ef-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/https-service-audit/anum-dr-https-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/http-service-monitoring/anum-vs-http-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/http-service-monitoring/anum-se-http-service.yaml;

- /package/conf/ k8s/base/openshift/istio-configs/http-service-monitoring/anum-gw-http-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/http-service-logger/anum-vs-http-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/http-service-logger/anum-se-http-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/http-service-logger/anum-gw-http-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/http-service-audit/anum-vs-http-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/http-service- audit/anum-se-http-service.yaml;
- /package/conf/ k8s/base/openshift/istio-configs/http-service-audit/anum-gw-http-service.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/dc.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/configmaps/anum-cm-egress-ott-gateway-logback.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-egress-ott-gateway-istio-basic.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/anum-cm-egress-ott-gateway-env.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/anum-svc-egress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-standin/anum-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-standin/anum-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-standin/anum-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-master/anum-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-master/anum-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-patterns-db-master/anum-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-master/anum-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-master/anum-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-master/anum-gw-dns-service.yaml;

- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-standin/anum-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-standin/anum-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-counters-db-standin/anum-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-1/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-1/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-1/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-2/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-2/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-2/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-3/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-3/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-3/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-4/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-4/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-4/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-5/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-5/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-logging-5/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-1/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-1/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-1/anum-gw-ip-service.yaml;

- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-2/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-2/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-2/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-3/anum-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-3/anum-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-autonum-journal-3/anum-gw-ip-service.yaml;
- /package/conf/k8s/base/autonum-patterns/testvs/anum-vs-patterns-service-test.yaml;
- /package/conf/k8s/base/autonum-patterns/testvs/anum-vs-patterns-service-deprecated-test.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-logging-passwords.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-logging-certs.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-journal-passwords.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-journal-certs.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-db-passwords.yaml;
- /package/conf/k8s/base/autonum-patterns/secrets/anum-secret-patterns-service-db-certs.yaml;
- /package/conf/k8s/base/autonum-patterns/dc.yaml;
- /package/conf/k8s/base/autonum-patterns/configmaps/anum-cm-patterns-service.yaml;
- /package/conf/k8s/base/autonum-patterns/configmaps/anum-cm-patterns-service-logback.yaml;
- /package/conf/k8s/base/autonum-patterns/configmaps/anum-cm-patterns-service-fluent-bit.yaml;
- /package/conf/k8s/base/autonum-patterns/configmaps/anum-cm-patterns-service-env.yaml;
- /package/conf/k8s/base/autonum-patterns/anum-vs-patterns-service.yaml;
- /package/conf/k8s/base/autonum-patterns/anum-vs-patterns-service-deprecated.yaml;
- /package/conf/k8s/base/autonum-patterns/anum-svc-patterns-service.yaml;
- /package/conf/k8s/base/autonum-patterns/anum-pdb-service-patterns.yaml;
- /package/conf/k8s/base/autonum-counters/testvs/anum-vs-counters-service-test.yaml;
- /package/conf/k8s/base/autonum-counters/testvs/anum-vs-counters-service-deprecated-test.yaml;

- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-logging-passwords.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-logging-certs.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-journal-passwords.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-journal-certs.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-db-passwords.yaml;
- /package/conf/k8s/base/autonum-counters/secrets/anum-secret-counters-service-db-certs.yaml;
- /package/conf/k8s/base/autonum-counters/dc.yaml;
- /package/conf/k8s/base/autonum-counters/configmaps/anum-cm-counters-service-logback.yaml;
- /package/conf/k8s/base/autonum-counters/configmaps/anum-cm-counters-service-fluent-bit.yaml;
- /package/conf/k8s/base/autonum-counters/configmaps/anum-cm-counters-service-env.yaml;
- /package/conf/k8s/base/autonum-counters /anum-vs-counters-service.yaml;
- /package/conf/k8s/base/autonum-counters/anum-vs-counters-service-deprecated.yaml;
- /package/conf/k8s/base/autonum-counters/anum-svc-counters-service.yaml;
- /package/conf/k8s/base/autonum-counters/anum-pdb-service-counters.yaml;
- /package/conf/k8s/base/autonum-configurer/testvs/anum-vs-configurer-test.yaml;
- /package/conf/k8s/base/autonum-configurer/testvs/anum-vs-configurer-service-deprecated-test.yaml;
- /package/conf/k8s/base/autonum-configurer/secrets/anum-secret-configurer-logging-passwords.yaml;
- /package/conf/k8s/base/autonum-configurer/secrets/anum-secret-configurer-logging-certs.yaml;
- /package/conf/k8s/base/autonum-configurer/secrets/anum-secret-configurer-db-passwords.yaml;
- /package/conf/k8s/base/autonum-configurer/secrets/anum-secret-configurer-db-certs.yaml;
- /package/conf/k8s/base/autonum-configurer/dc.yaml;
- /package/conf/k8s/base/autonum-configurer/configmaps/anum-cm-configurer.yaml;
- /package/conf/k8s/base/autonum-configurer/configmaps/anum-cm-configurer-logback.yaml;
- /package/conf/k8s/base/autonum-configurer/configmaps/anum-cm-configurer-fluent-bit.yaml;
- /package/conf/k8s/base/autonum-configurer/configmaps/anum-cm-configurer-env.yaml;
- /package/conf/k8s/base/autonum-configurer/anum-vs-configurer.yaml;

- /package/conf/k8s/base/autonum-configurer/anum-vs-configurer-service-deprecated.yaml;
- /package/conf/k8s/base/autonum-configurer/anum-svc-configurer.yaml;
- /package/conf/k8s/base/autonum-configurer/anum-pdb-configurer.yaml;
- /package/conf/k8s/base/prometheus/dc.yaml;
- /package/conf/k8s/base/ prometheus/configmaps/anum-cm-prometheus.yaml;
- /package/conf/k8s/base/ prometheus/anum-se-prometheus-kubeapi-ext.yaml.

Пример заполнения плейсхолдеров приведен в файле `conf/config/parameters`.

После замены файлов конфигурации необходимо развернуть файлы конфигурации в среде контейнеризации:

1. В панели администратора в правом верхнем углу нажать на кнопку **Import YAML**;
2. в открывшейся окно вставить скопированный конфигурационный файл и нажать **Create**;
3. после создания отобразиться страница конфигурационного файла;
4. повторить действия 1-3 для всех файлов конфигурации.

Настройка интеграции со смежными сервисами (опционально)

Настройка интеграции со смежными сервисами происходит путем конфигурации параметров для конкретного внешнего сервиса в определенных ConfigMaps:

- настройка интеграции с компонентом Аудит продукта Platform V Audit SE;
- настройка интеграции с компонентом Объединенный мониторинг Unimon продукта Platform V Monitor;
- настройка интеграции с компонентом Журналирование продукта Platform V Monitor;
- настройка интеграции с компонентом One-Time Password (OTP) / OTT;
- настройка интеграции с компонентом Прикладной журнал.

Настройка интеграции с компонентом Аудит продукта Platform V Audit

Для настройки интеграции сервиса Автонумерация с сервисом Аудит в папке `conf/config/parameters` сконфигурируйте следующие параметры:

```
# <!- заполните самостоятельно -> - значения параметров
необходимо заполнить самостоятельно перед развертыванием
сервиса ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_PORT_NAME=http-
audit ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_PORT_NUMBER=<!-
заполните самостоятельно ->
ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_TARGETPORT_NUMBER=<!-
заполните самостоятельно ->  AUTONUMPATTERNS_AUDIT_STUB=true
```

```
AUTONUMPATTERNS_AUDIT_RESOURCENAME=<!- заполните
самостоятельно -> AUTONUMPATTERNS_AUDIT_HOST=<!- заполните
самостоятельно ->
AUTONUMPATTERNS_AUDIT_BULKHEAD_MAXWAITDURATION=<!- заполните
самостоятельно ->
AUTONUMPATTERNS_AUDIT_BULKHEAD_MAXCONCURRENTCALLS=<!-
заполните самостоятельно ->  AUTONUMCOUNTERS_AUDIT_STUB=<!-
заполните самостоятельно: true/false -> #
Включение/отключение режима загулшки для сервиса Аудит
AUTONUMCOUNTERS_AUDIT_HOST=<!- заполните самостоятельно -> #
Адрес сервиса Аудит
AUTONUMCOUNTERS_AUDIT_BULKHEAD_MAXWAITDURATION=<!- заполните
самостоятельно -> # Максимальное время ожидания паттерна
bulkhead для Аудита
AUTONUMCOUNTERS_AUDIT_BULKHEAD_MAXCONCURRENTCALLS=<!-
заполните самостоятельно -> # Максимальное количество
одновременных вызовов Аудита
```

Проверку успешности проведения интеграции компонента Автонумерация с внешними сервисами выполняйте в консоли сервиса, с которым выполняется интеграция.

Пример заполнения для интеграции с компонентом Аудит

```
# <!- заполните самостоятельно -> - значения параметров
необходимо заполнить самостоятельно перед развертыванием
сервиса ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_PORT_NAME=http-
audit ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_PORT_NUMBER=8001
ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_TARGETPORT_NUMBER=8001
AUTONUMPATTERNS_AUDIT_STUB=true
AUTONUMPATTERNS_AUDIT_RESOURCENAME=platformv:core:1
AUTONUMPATTERNS_AUDIT_HOST=http://audit2-client-
proxy.apps.dev-gen.xxxxx.xxxx.ru:8080
AUTONUMPATTERNS_AUDIT_BULKHEAD_MAXWAITDURATION=5000ms
AUTONUMPATTERNS_AUDIT_BULKHEAD_MAXCONCURRENTCALLS=5
AUTONUMCOUNTERS_AUDIT_STUB=true # Включение/отключение
режима загулшки для сервиса Аудит
AUTONUMCOUNTERS_AUDIT_HOST=http://audit2-client-
proxy.apps.dev-gen.xxxxx.xxxx.ru:8080 # Адрес сервиса Аудит
AUTONUMCOUNTERS_AUDIT_BULKHEAD_MAXWAITDURATION=5000ms #
Максимальное время ожидания паттерна bulkhead для Аудита
AUTONUMCOUNTERS_AUDIT_BULKHEAD_MAXCONCURRENTCALLS=5
```

Настройка в случае развертывания без компонента Аудит

Для настройки сервиса Автонумерация без интеграции с компонентом Аудит в `conf/config/parameters` сконфигурируйте следующие параметры:

```
AUTONUMCOUNTERS_AUDIT_STUB=false # Отключение режима
загулшки для сервиса Аудит AUTONUMPATTERNS_AUDIT_STUB=false
```

Настройка интеграции с компонентом Объединенный мониторинг Unimon продукта Platform V Monitor

Пререквизиты интеграции с компонентом Объединенный мониторинг Unimon

Пререквизитами настройки интеграции компонента Автонумерация с сервисом Мониторинг является создание в среде контейнеризации Service Account и Role Binding, необходимых для корректного развертывания deployment-config prometheus в среде контейнеризации.

Развертывание конфигурации Prometheus

Если namespace настроен так, что трафик от Prometheus не проходит, то добавьте ServiceEntry:

```
- apiVersion: networking.istio.io/v1alpha3     kind:
ServiceEntry     metadata:       name: prometheus-se
labels:         app: prometheus     spec:       exportTo:
- .       hosts:       - ${APPMON_HOST}       ports:       -
name: http         number: 80         protocol: HTTP       -
name: https         number: 443         protocol: TLS
resolution: DNS
```

И параметр:

```
- name: APPMON_HOST     displayName: "Applied Monitoring
host"     description: "Хост для правила исходящего трафика.
Обязательный."     required: true
```

Пример значения параметра:

```
APPMON_HOST=appmon-ift.apps.dev-gen.ca.xxxxx.ru
```

Настройка интеграции с компонентом Мониторинг

Для настройки интеграции компонента Автонумерация с компонентом Мониторинг в `conf/config/parameters` сконфигурируйте следующие параметры.

```
# <!- заполните самостоятельно -> - значения параметров
необходимо заполнить самостоятельно перед развертыванием
```

```
сервиса  ISTIOCONFIGS_HTTPSERVICE_MONITORING_NAME=monitoring
ISTIOCONFIGS_HTTPSERVICE_MONITORING_HOST=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_INTERNAL=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_GATEWAY=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_TARGET=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_TIMEOUT=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_RETRIES_ATTEMPTS=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_RETRIES_PERTRYTIMEOUT=<!
- заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_NAME=monitoring
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_HOST=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_INTERNAL=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_GATEWAY=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_TARGET=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_TCP_MAXCONNECTIONS=<!- заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_HTTP_MAXPENDREQ=<!- заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_HTTP_MAXREQUESTSPERCONNECTION=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_CONSECUTIVE5XXERRORS=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_INTERVAL=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_BASEEJECTIONTIME=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_TIMEOUT=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_RETRIES_ATTEMPTS=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_RETRIES_PERTRYTIMEOUT=<
!- заполните самостоятельно ->
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_PORT_NAME=http-
monitoring
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_PORT_NUMBER=<!-
заполните самостоятельно ->
```

```
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_TARGETPORT_NUMBER=<!-
заполните самостоятельно ->
```

При необходимости внесения изменений в интеграцию Автонумерация с внешними сервисами внесите соответствующие правки в `conf/config/parameters`.

Пример заполнения conf/config/parameters для интеграции с компонентом Объединенный мониторинг Unimon

```
ISTIOCONFIGS_HTTPSERVICE_MONITORING_NAME=monitoring
ISTIOCONFIGS_HTTPSERVICE_MONITORING_HOST=appmon-
service.apps.dev-gen2.sigma.sbrf.ru
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_INTERNAL=8080
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_GATEWAY=8002
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_TARGET=80
ISTIOCONFIGS_HTTPSERVICE_MONITORING_TIMEOUT=5s
ISTIOCONFIGS_HTTPSERVICE_MONITORING_RETRIES_ATTEMPTS=1
ISTIOCONFIGS_HTTPSERVICE_MONITORING_RETRIES_PERTRYTIMEOUT=5s
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_NAME=monitoring
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_HOST=appmon-
service.apps.dev-gen2.sigma.sbrf.ru
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_INTERNAL=8080
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_GATEWAY=8002
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_TARGET=443
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_TCP_MAXCONNECTIONS=5
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_HTTP_MAXPENDREQ=100
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_HTTP_MAXREQUESTSPERCONNECTION=30
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_CONSECUTIVE5XXERRORS=10
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_INTERVAL=5m
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_BASEEJECTIONTIME=5m
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_TIMEOUT=5s
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_RETRIES_ATTEMPTS=1
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_RETRIES_PERTRYTIMEOUT=5
s  ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_PORT_NAME=http-
monitoring
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_PORT_NUMBER=8002
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_TARGETPORT_NUMBER=8002
```

Настройка интеграции с компонентом Журналирование продукта Platform V Monitor

При ручном развертывании компонента Автонумерация выполните следующие действия:

1. В файле configmap-fluentbit.yml для плейсхолдера {{LOGGER_SERVICE_HOST_NAME}}, {{LOGGER_SERVICE_PORT}} установите соответствующие значения для сервиса Журналирование, развернутого в целевом окружении.
2. Загрузите в среду контейнеризации сконфигурированный файл configmap-fluentbit.yml.

Для настройки интеграции компонента Автонумерация с компонентом Журналирование в conf/config/parameters сконфигурируйте следующие параметры:

```
# <!- заполните самостоятельно -> - значения параметров
необходимо заполнить самостоятельно перед развертыванием
сервиса  SERVICERN=platformv:core:<!- заполните
самостоятельно -> # Resource name из-под которого работает
сервис AUTONUMCONFIGURER_LOGGING_KAFKA_BROKERS=<!- заполните
самостоятельно -> # Адреса серверов Kafka для откидывания
логов AUTONUMCONFIGURER_LOGGING_KAFKA_TOPICS=<!- заполните
самостоятельно -> # Топик для откидывания логов в Kafka
AUTONUMCONFIGURER_LOGGING_KAFKA_SECURITYPROTOCOL=<!-
заполните самостоятельно -> # Протокол Kafka для откидывания
логов  AUTONUMCOUNTERS_LOGGING_KAFKA_BROKERS=<!- заполните
самостоятельно -> # Адреса серверов Kafka для откидывания
логов AUTONUMCOUNTERS_LOGGING_KAFKA_TOPICS=<!- заполните
самостоятельно -> # Топик для откидывания логов в Kafka
AUTONUMCOUNTERS_LOGGING_KAFKA_SECURITYPROTOCOL=<!- заполните
самостоятельно -> # Протокол Kafka для откидывания логов
AUTONUMCOUNTERS_LOGGING_REGISTRY_REPOSITORY
AUTONUMPATTERNS_LOGGING_KAFKA_BROKERS=<!- заполните
самостоятельно -> # Адреса серверов Kafka для откидывания
логов AUTONUMPATTERNS_LOGGING_KAFKA_TOPICS=<!- заполните
самостоятельно -> # Топик для откидывания логов в Kafka
AUTONUMPATTERNS_LOGGING_KAFKA_SECURITYPROTOCOL=<!- заполните
самостоятельно -> | PLAINTEXT # Протокол Kafka для
откидывания логов
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
```

```
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_TARGET=<!-
```

```
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
```

```
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
```

При необходимости внесения изменений в интеграцию Автонумерация с внешними сервисами внесите соответствующие правки в `conf/config/parameters`.

Проверку успешности проведения интеграции Автонумерация с внешними сервисами выполняйте в консоли сервиса, с которым выполняется интеграция.

Пример заполнения conf/config/parameters для интеграции с компонентом Журналирование

```
SERVICERN=platformv:core:1 # Resource name из-под которого
работает сервис
AUTONUMCONFIGURER_LOGGING_KAFKA_BROKERS=10.106.228.11:18082
# Адреса серверов Kafka для откидывания логов
AUTONUMCONFIGURER_LOGGING_KAFKA_TOPICS=default # Топик для
откидывания логов в Kafka
AUTONUMCONFIGURER_LOGGING_KAFKA_SECURITYPROTOCOL=SSL #
Протокол Kafka для откидывания логов
AUTONUMCOUNTERS_LOGGING_KAFKA_BROKERS=10.106.228.11:18082 #
Адреса серверов Kafka для откидывания логов
AUTONUMCOUNTERS_LOGGING_KAFKA_TOPICS=default # Топик для
откидывания логов в Kafka
AUTONUMCOUNTERS_LOGGING_KAFKA_SECURITYPROTOCOL=SSL |
PLAINTEXT # Протокол Kafka для откидывания логов
AUTONUMCOUNTERS_LOGGING_REGISTRY_REPOSITORY
AUTONUMPATTERNS_LOGGING_KAFKA_BROKERS=10.106.228.11:18082 #
Адреса серверов Kafka для откидывания логов
AUTONUMPATTERNS_LOGGING_KAFKA_TOPICS=default # Топик для
откидывания логов в Kafka
AUTONUMPATTERNS_LOGGING_KAFKA_SECURITYPROTOCOL=SSL |
PLAINTEXT # Протокол Kafka для откидывания логов
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_NAME=logging-1 #
```

```
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_IP=10.0.0.1 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_INTERNAL=1808
2 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_GATEWAY=12808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_NAME=logging-1 #
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_HOST=logging-
1.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_INTERNAL=180
82 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_GATEWAY=1280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_NAME=logging-2 #
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_IP=10.0.0.2 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_INTERNAL=2808
2 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_GATEWAY=22808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_NAME=logging-2 #
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_HOST=logging-
2.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_INTERNAL=280
82 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_GATEWAY=2280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_NAME=logging-3 #
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_IP=10.0.0.3 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_INTERNAL=3808
2 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_GATEWAY=32808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_NAME=logging-3 #
```

```
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_HOST=logging-
3.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_INTERNAL=380
82 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_GATEWAY=3280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_NAME=logging-4 #
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_IP=10.0.0.4 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_INTERNAL=4808
2 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_GATEWAY=42808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_NAME=logging-4 #
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_HOST=logging-
4.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_INTERNAL=480
82 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_GATEWAY=4280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_NAME=logging-5 #
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_IP=10.0.0.5 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_INTERNAL=5808
2 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_GATEWAY=52808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_NAME=logging-5 #
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_HOST=logging-
5.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_INTERNAL=580
82 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_GATEWAY=5280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
```

Настройка интеграции с компонентом Прикладной журнал

Для настройки интеграции компонента Автонумерация с компонентом Прикладной журнал:

- Заведите заявку на администраторов компонента Прикладной журнал на стенде на подключение ZONE_ID в конфигураторе со значением `ANUM`.
- Обратитесь к администраторам компонента Прикладной журнал для выпуска сертификатов: `server.keystore.jks` и `trust.jks`.
- Укажите значения вместо плейсхолдеров (плейсхолдеры заключены в двойные фигурные скобки {{ }}).

```
ZONE_ID сервиса регистрируемая в Configurator силами
администраторов Прикладного Журнала AJ_STAND_IN_ZONE_ID=SCHD
URL MMT. Пример значения: stub_url – если включен режим stub
в параметре AJ_STAND_IN_STUB_MODE_FLAG=true, то любое
значение, иначе необходимо указать корректный URL ПЖ,
который установлен на стенде. AJ_MMT_PROXY_URL=<Обязательно
к заполнению>   Время для выполнения запроса на сервер ПЖ в
секундах AJ_MMT_CALL_TIMEOUT=30  Флаг использования заглушки
вместо полноценного Stand-In. true – интеграция с ПЖ
отключена, false – интеграция с ПЖ включена.
AJ_STAND_IN_STUB_MODE_FLAG=true   Сервер Kafka. Если включен
режим stub в параметре AJ_STAND_IN_STUB_MODE_FLAG=true, то
любое значение, иначе необходимо указать корректный сервер
ПЖ, который установлен на стенде.
AJ_KAFKA_BOOTSTRAP_SERVERS=<Обязательно к заполнению>
Протокол взаимодействия с Kafka PLAINTEXT для http endpoint
ПЖ, для https значение SSL AJ_KAFKA_SECURITY_PROTOCOL=SSL
```

Пример заполненного файла `configmap.yml`

```
standin:   cloud:     client:       zoneId:
{{AJ_STAND_IN_ZONE_ID}}       mmtProxyUrl:
{{AJ_MMT_PROXY_URL}}       mmtCallTimeout:
{{AJ_MMT_CALL_TIMEOUT}}       stub:
{{AJ_STAND_IN_STUB_MODE_FLAG}}       heartBeatPeriod: 1000
kafka:         bootstrapServers:
{{AJ_KAFKA_BOOTSTRAP_SERVERS}}         producerConfig:
"[security.protocol]": {{AJ_KAFKA_SECURITY_PROTOCOL}}
"[ssl.key.password]": ${AJ_SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/app/certs/journal/server.keystore.jks
"[ssl.keystore.password]": ${AJ_SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /app/certs/journal/trust.jks
"[ssl.truststore.password]": ${AJ_SSL_TRUSTSTORE_PASSWORD}
```

```
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS          "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
consumerConfig:          "[security.protocol]":
{{AJ_KAFKA_SECURITY_PROTOCOL}}
"[ssl.key.password]": ${AJ_SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/app/certs/journal/server.keystore.jks
"[ssl.keystore.password]": ${AJ_SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /app/certs/journal/trust.jks
"[ssl.truststore.password]": ${AJ_SSL_TRUSTSTORE_PASSWORD}
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS          "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
concurrency: 10       groupId: group_1
```

Настройка интеграции с компонентом One-Time Password (OTP) / OTT

Для настройки интеграции сервиса Автонумерация с OTT в conf/config/parameters сконфигурируйте параметры для микросервисов Counters, Patterns, Configurer:

```
AUTONUMPATTERNS_ISTIO_GATEWAY_NAME=anum-gw-ingress-ott-
gateway AUTONUMPATTERNS_ISTIO_GATEWAY_ROUTES_OTTHOST=<!-
заполните самостоятельно. Например: autonum-dev-
ott.ingress.apps.dev-gen2.sigma.sbrf.ru ->
AUTONUMCONFIGURER_ISTIO_GATEWAY_NAME=anum-gw-ingress-ott-
gateway AUTONUMCONFIGURER_ISTIO_GATEWAY_ROUTES_OTTHOST=<!-
заполните самостоятельно. Например: autonum-configurer-dev-
ott.ingress.apps.dev-gen2.sigma.sbrf.ru ->
AUTONUMCOUNTERS_ISTIO_GATEWAY_NAME=anum-gw-ingress-ott-
gateway AUTONUMCOUNTERS_ISTIO_GATEWAY_ROUTES_OTTHOST=<!-
заполните самостоятельно. Например: autonum-dev-
ott.ingress.apps.dev-gen2.sigma.sbrf.ru ->
ISTIOCONFIGS_OTTINTEGRATION_REGISTRYREPOSITORY=<!- заполните
самостоятельно. Например:
registry.sigma.sbrf.ru/pprb/ci00641491/ci01125613_ott ->
ISTIOCONFIGS_OTTINTEGRATION_OTTSERVICECERTALIAS=<!-
заполните самостоятельно. Например: ott-service ->
ISTIOCONFIGS_OTTINTEGRATION_OTTSERVICEHOSTS=<!- заполните
самостоятельно. Например: 10.53.99.178:8443,10.53.96.30:8443
-> ISTIOCONFIGS_OTTINTEGRATION_OTTSERVICEURL=<!- заполните
самостоятельно. Например: https://10.53.99.178:8443/ott-
service/rest/token ->
ISTIOCONFIGS_OTTINTEGRATION_OTTAUTHZREALM=<!- заполните
```

```
самостоятельно. Например: mmt ->
ISTIOCONFIGS_OTTINTEGRATION_OTTCLIENTCERTALIAS=<!- заполните
самостоятельно. Например: autonum-os ->
ISTIOCONFIGS_OTTINTEGRATION_OTTMODULEID=<!- заполните
самостоятельно. Например: autonum-os ->
ISTIOCONFIGS_OTTINTEGRATION_OTTOPERMODE_EGRESS=<!- заполните
самостоятельно. Например: sign ->
ISTIOCONFIGS_OTTINTEGRATION_OTTOPERMODE_INGRESS=<!-
заполните самостоятельно. Например: validate ->
```

При необходимости внесения изменений в интеграцию компонента Автонумерация с внешними сервисами внесите соответствующие правки в `conf/config/parameters`.

Проверку успешности проведения интеграции компонента Автонумерация с внешними сервисами выполняйте в консоли сервиса, с которым выполняется интеграция.

Настройка интеграции с компонентом Прикладной журнал

Для настройки интеграции компонента Автонумерация с компонентом Прикладной журнал:

- Заведите заявку на администраторов компонента Прикладной журнал на стенде на подключение ZONE_ID в конфигураторе со значением `ANUM`.
- Обратитесь к администраторам компонента Прикладной журнал для выпуска сертификатов: `server.keystore.jks` и `trust.jks`.
- Укажите значения вместо плейсхолдеров (плейсхолдеры заключены в двойные фигурные скобки {{ }}).

```
ZONE_ID сервиса регистрируемая в Configurator силами
администраторов Прикладного Журнала AJ_STAND_IN_ZONE_ID=SCHD
URL MMT. Пример значения: stub_url – если включен режим stub
в параметре AJ_STAND_IN_STUB_MODE_FLAG=true, то любое
значение, иначе необходимо указать корректный URL ПЖ,
который установлен на стенде. AJ_MMT_PROXY_URL=<Обязательно
к заполнению>  Время для выполнения запроса на сервер ПЖ в
секундах AJ_MMT_CALL_TIMEOUT=30  Флаг использования заглушки
вместо полноценного Stand-In. true – интеграция с ПЖ
отключена, false – интеграция с ПЖ включена.
AJ_STAND_IN_STUB_MODE_FLAG=true  Сервер Kafka. Если включен
режим stub в параметре AJ_STAND_IN_STUB_MODE_FLAG=true, то
любое значение, иначе необходимо указать корректный сервер
ПЖ, который установлен на стенде.
AJ_KAFKA_BOOTSTRAP_SERVERS=<Обязательно к заполнению>
```

```
Протокол взаимодействия с Kafka PLAINTEXT для http endpoint
ПЖ, для https значение SSL AJ_KAFKA_SECURITY_PROTOCOL=SSL
```

Пример заполненного файла `configmap.yml`

```
standin:   cloud:     client:       zoneId:
{{AJ_STAND_IN_ZONE_ID}}       mmtProxyUrl:
{{AJ_MMT_PROXY_URL}}       mmtCallTimeout:
{{AJ_MMT_CALL_TIMEOUT}}       stub:
{{AJ_STAND_IN_STUB_MODE_FLAG}}       heartBeatPeriod: 1000
kafka:         bootstrapServers:
{{AJ_KAFKA_BOOTSTRAP_SERVERS}}         producerConfig:
"[security.protocol]": {{AJ_KAFKA_SECURITY_PROTOCOL}}
"[ssl.key.password]": ${AJ_SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/app/certs/journal/server.keystore.jks
"[ssl.keystore.password]": ${AJ_SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /app/certs/journal/trust.jks
"[ssl.truststore.password]": ${AJ_SSL_TRUSTSTORE_PASSWORD}
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS           "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
consumerConfig:           "[security.protocol]":
{{AJ_KAFKA_SECURITY_PROTOCOL}}
"[ssl.key.password]": ${AJ_SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/app/certs/journal/server.keystore.jks
"[ssl.keystore.password]": ${AJ_SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /app/certs/journal/trust.jks
"[ssl.truststore.password]": ${AJ_SSL_TRUSTSTORE_PASSWORD}
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS           "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
concurrency: 10       groupId: group_1
```

Автоматизированная установка сервиса с использованием Deploy Tools

Для установки сервиса с использованием компонента Deploy Tools (CDJE) программного продукта Platform V DevOps Tools (DOT) разверните сервис в инфраструктуре и запустите CD Job по инструкции, приведенной в документации Platform V DevOps Tools (DOT).

Обновление

Специальной процедуры обновления сервиса не предусмотрено.

Для обновления сервиса:

1. Удалите сервис согласно процедуре, описанной в разделе «Удаление».
2. Установите сервиса согласно процедуре, описанной в разделе «Установка».

Удаление

Для удаления компонента Автонумерация очистите namespace, выполнив команду:

```
{ oc get -o name gateway.networking.istio.io; oc get -o name
virtualservice; oc get -o name envoyfilter; oc get -o name
destinationrule.networking.istio.io; oc get -o name
serviceentry; oc get -o name deployment; oc get -o name
configmap; oc get -o name service; oc get -o name route; } |
xargs oc delete
```

Проверка работоспособности

Через вызов API

Для проверки работоспособности компонента Автонумерация отправьте запросы на микросервисы:

- Configurer;
- Counters;
- Patterns;

Проверка работоспособности микросервиса Configurer

Для проверки работоспособности микросервиса Configurer выполните:

1. Запрос на создание сущности `resource`:

```
curl -v -X POST
'[ROUTE]/autonum/v3/rn/admin/configurer/resources' --header
--header 'ott-invoker: invoker' --header 'SUBJECT:
subjectrn' --header 'Content-Type: application/json' --data-
raw '{ "resource_name": "unique_resource_name"}'
```

Ответ:

```
{"message":"resource was created"}
```

2. Запрос на создание шаблона `template`:

```
curl -v -X POST
'[ROUTE]/autonum/v3/rn/admin/configurer/templates' --header
```

```
'ott-invoker: invoker' --header 'ott-hash: 1' --header 'ott-
token: 1' --header 'SUBJECT: subjectrn' --header 'Content-
Type: application/json' --data-raw '{ "template_code":
"template_code_name", "template_name": "qwerty", "bottom":
1, "top": 190, "step": 1, "endless": 0, "range_format":
"SI%date$ty%date$tM%date$tdX01X"}'
```

Ответ:

```
{"message":"Template was created"}
```

3. Запрос на создание pattern:

```
curl -v -X POST
'[ROUTE]/autonum/v3/rn/admin/configurer/patterns' --header
'ott-invoker: invoker' --header 'SUBJECT: subjectrn' --
header 'Content-Type: application/json' --data-raw
'{"template_code": "template_code_name", "pattern_code":
"pattern_code_name", "pattern_name": "Шаблон номера
запроса", "number_format":
"SI%date$ty%date$tM%date$tdX01%[num]$05X", "pattern_desc":
"", "resource_name": "default"}'
```

Ответ:

```
{"message":"Pattern was created"}
```

4. Запрос на удаление сущности `resource`:

```
curl -v -X DELETE
'[ROUTE]/autonum/v3/rn/admin/configurer/resources' --header
'ott-invoker: invoker' --header 'SUBJECT: subjectrn' --
header 'Content-Type: application/json' --data-raw
'{"resource_name": "unique_resource_name"}'
```

Ответ:

```
{"message":"resource was deleted"}
```

Проверка работоспособности микросервиса Counters

Для проверки работоспособности микросервиса Counters выполните:

1. Запрос на создание счетчика:

```
curl -v -X POST '[ROUTE]/autonum/v3/rn/default/counters' --
header 'ott-invoker: invoker' --header 'Content-Type:
application/json' --data-raw '{ "name": "counter_name",
"start": 0, "min": 0, "max": 20000, "step": 1, "cycle":
true, "format": "Account: $_$ $date$"}'
```

Ответ:

```
{"name":"counter_name","start":0,"min":0,"max":20000,"step":
1,"cycle":true,"format":"Account: $_$ $date$"}
```

2. Запрос на получение строки со следующим значением счетчика:

```
curl -v -X POST
'[ROUTE]/autonum/v3/rn/default/counters/counter_name/next' -
-header 'ott-invoker: invoker' --header 'Content-Type:
application/json' --data-raw '{ "date":"11.09.2001"}'
```

Ответ:

```
{"value":"Account: 0 11.09.2001"}
```

3. Запрос на удаление счетчика:

```
curl -v -X DELETE
'[ROUTE]/autonum/v3/rn/default/counters/counter_name' --
header 'ott-invoker: invoker'
```

Ответ:

```
* Connection #0 to host autonumcounters-dev-
http.ingress.apps.dev-gen2.sigma.sbrf.ru left intact
```

Проверка работоспособности микросервиса Patterns

Для проверки работоспособности микросервиса Patterns выполните запросы:

1. Запрос на получение инкремента для используемого шаблона:

```
curl -v -X POST
'[ROUTE]/autonum/v3/rn/default/templates/RequestForCommissio
nPaymentTemplate/next' --header 'Content-Type:
application/json' --header 'ott-invoker: invoker' --data-raw
'{"date": {"type": "Date","format": "dd-MM-yyyy","value":
"01-02-2018"}}'
```

Ответ:

```
{"value":1}
```

2. Запрос на получение массива инкрементов для используемого шаблона:

```
curl -v -X POST
'[ROUTE]/autonum/v3/rn/default/templates/RequestForCommissio
nPaymentTemplate/next/3' --header 'Content-Type:
application/json' --header 'ott-invoker: invoker' --data-raw
'{"date": {"type": "Date","format": "dd-MM-yyyy","value":
"01-02-2018"}}'
```

Ответ:

```
{"values":[2,3,4]}
```

3. Запрос на получение следующего значения для используемого паттерна:

```
curl -v -X POST
'[ROUTE]/autonum/v3/rn/default/patterns/RequestForCommission
PaymentPattern/next' --header 'Content-Type:
application/json' --header 'ott-invoker: invoker' --data-raw
'{"date": {"type": "Date","format": "dd-mm-yyyy","value":
"01-02-2017"}}'
```

Ответ:

```
{"value":"SI170201X0100001"}
```

4. Запрос на получение массива значений для используемого паттерна:

```
curl -v -X POST
'[ROUTE]/autonum/v3/rn/default/patterns/RequestForCommission
```

```
PaymentPattern/next/3' --header 'Content-Type:
application/json' --header 'ott-invoker: invoker' --data-raw
'{"date": {"type": "Date","format": "dd-mm-yyyy","value":
"01-02-2017"}}'
```

Ответ:

```
{"values":["SI170201X0100005","SI170201X0100006","SI170201X0
100007"]}
```

Через UI среды контейнеризации

Проверить успешность установки сервиса можно через UI среды контейнеризации.

Для этого нужно перейти на вкладку **Pods**. Если все Pods (Ingress, Egress, Autonum, Prometheus (опционально)) находятся в статусе `running`, значит сервис успешно установлен.

Откат

Для отката сервиса Автонумерация на предыдущую версию дистрибутива:

1. Удалите все развернутые в окружении артефакты сервиса, кроме БД (и информации в БД).
2. Выполните установку сервиса согласно разделу «Установка».

Часто встречающиеся проблемы и пути их устранения

В журнале инсталлятора встречается ошибка следующего вида:

```
17:00:41 ✗ Error from server (Invalid):
"${ISTIOCONFIGS_DNSSERVICE_AUTONUM_DB_MASTER_NAME}" is
invalid: [metadata.name: Invalid value:
"${ISTIOCONFIGS_DNSSERVICE_AUTONUM_DB_MASTER_NAME}":
```

Путь устранения: задать значение переменной `ISTIOCONFIGS_DNSSERVICE_AUTONUM_DB_MASTER_NAME` в соответствие с конфигурацией к инсталлятору.

Чек-лист валидации установки

1. Проверить журналы событий (log-файлы) инсталлятора на наличие ошибок.
2. Проверить журналы событий (log-файлы) отсека (pod) сервиса в среде контейнеризации на наличие ошибок.

При успешном выполнении всех пунктов можно приступать к использованию сервиса.

Чек-лист проверки работоспособности интеграции

При осуществлении интеграции (опциональных интеграций) сервиса Автонумерация с платформенными компонентами, проверка работоспособности осуществляется согласно следующим инструкциям:

- компонент Объединенный мониторинг Unimon;
- компонент Аудит;
- компонент Журналирование;
- компонент Прикладной журнал.

Проверка работоспособности компонента Объединенный мониторинг Unimon

Для проверки работоспособности компонента Объединенный мониторинг Unimon:

1. В среде контейнеризации перейдите на вкладку **PODs** и выберите pod (anum-configurer, anum-service-counters, anum-service-patterns).
2. Перейдите в терминал и введите команду

```
curl localhost:8081/actuator/prometheus
```

3. Сравните полученное сообщение с отбрасываемыми метриками с событиями мониторинга, приведенными в Руководстве по системному администрированию в разделе «События мониторинга».
4. Перейдите в систему отображения метрик (например, Prometheus), указав необходимую метрику. Например, system_cpu_usage.

Проверка работоспособности компонента Аудит

Для проверки работоспособности компонента Аудит выполните следующие действия:

1. Перейти в АРМ Аудит.
2. Выберите поле **Поиск событий аудита** и на открывшейся странице заполните:
- поля **Период** — необходимым временным периодом (текущая дата);
- поле **Исходный текст** — «anum».
3. Нажмите кнопку **Найти**.
4. Если в результатах поиска присутствуют записи, созданные по времени после запуска сервиса, то проверка выполнена успешно.

Проверка работоспособности компонента Журналирование

Для проверки работоспособности компонента Журналирование выполните следующие действия:

1. Перейти в АРМ Журналирование.

2. Перейти на вкладку **Системный журнал** и на открывшейся странице заполните поля: **Начальная дата**, **Начальное время**, **Конечная дата**, **Конечное время**.
3. Нажмите кнопку **Поиск**.
4. Если в результатах поиска присутствуют записи, созданные по времени после запуска сервиса, то проверка выполнена успешно.

Проверка работоспособности компонента Прикладной журнал

Для проверки работоспособности компонента Прикладной журнал выполните следующие действия:

1. Перейти в АРМ Прикладной журнал.
2. На открывшейся странице **Журнал** выберите в поле **Зона** — «AUTONUMERATION», уберите галочку с чек-бокса **С ошибками репликации** и нажмите кнопку **Найти**.
3. Если в результатах поиска присутствуют записи, созданные по времени после запуска сервиса, то проверка выполнена успешно.

Руководство по установке компонента компонента Генерация ID (Sberflake ID) (SNOW)

Системные требования

Настройки безопасности окружения и перечень платформенных (дополнительных внешних) продуктов, используемых для установки, настройки и контроля в конечной информационной системе (далее — ИС), выбираются при разработке конечной ИС, исходя из характера обрабатываемой в ней информации и иных требований информационной безопасности (далее — ИБ), предъявляемых к ней.

Системное программное обеспечение

Ниже представлены категории системного программного обеспечения, которые обязательны для установки, настройки, контроля и функционирования компонента. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО «СберТех» (маркировка «Рекомендовано» в столбце Комментарий). Клиенту необходимо выбрать один из продуктов в каждой категории, исходя из условий использования конечной ИС.

Операционная система

Наименование	Версия	Комментарий
ОС Альт 8 СП	10.0 и выше	Рекомендовано
Red Hat Enterprise Linux	3.10 и выше	Опционально

Среда контейнеризации

Необходимое окружение для разворачивания — Kubernetes версии 1.0 и выше или OpenShift версии 4.2 и выше (далее — среда контейнеризации).

Наименование	Версия	Комментарий
Kubernetes	1.0 и выше	Рекомендовано
Red Hat OpenShift	4.2 и выше	Опционально

Java-машина

Наименование	Версия	Комментарий
OpenJDK	1.8 и выше	Рекомендовано
OracleJDK	1.8 и выше	Опционально

Система управления базами данных (СУБД)

Наименование	Версия	Комментарий
PostgreSQL	11 и выше	Рекомендовано

Правообладателем АО «СберТех» также рекомендован Platform V Pangolin SE, см. раздел «Платформенные зависимости».

Инструмент управления программными проектами

Наименование	Версия	Комментарий
Apache Maven	3.6.1 и выше	Рекомендовано

Сервер приложений

Наименование	Версия	Комментарий
Apache Tomcat	9.0.39 и выше	Рекомендовано

Система для проверки функционирования сервисов

Наименование	Версия	Комментарий
SYMON	1.3.2 и выше	Рекомендовано (установка необязательна)

Сервис интеграции и оркестрации микросервисов в облаке

Наименование	Версия	Комментарий
Istio	1.12 и выше	Рекомендовано

Правообладателем АО «СберТех» также рекомендован Platform V Synapse Service Mesh, см. раздел «Платформенные зависимости».

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Audit SE	AUD	1.1	AUDT Аудит	опционально	Сервис для аудирования событий
Platform V Monitor	OPM	3.27.11	LOGA Журналирование	опционально	Сервис для хранения лог-файлов
Platform V Monitor	OPM	6.0.0-02	MONA Объединенный мониторинг Unimon	опционально	Сервис для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения
Platform V Pangolin SE	PSQ	4.6.0. и выше	–	опционально	Система управления базами данных, основанная на PostgreSQL
Platform V DevOps Tools	DOT	1.2	–	опционально	Сервис для развертывания и обновления компонентов Платформы и приложений потребителей, для настройки и обслуживания инфраструктуры Платформы

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Synapse Service Mesh	SSM	1.12 и выше	POLM Управление политиками	опционально	Панель управления, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в среде контейнеризации Kubernetes

Примечание:

- **Обязательно** — компонент или продукт необходим для функционирования сервиса.
- **Опционально** — необязательный для функционирования сервиса компонент или продукт, рекомендуется его установка, но допускается использование аналога других производителей.

Аппаратные требования

Для установки компонента/продукта требуется следующая конфигурация аппаратного обеспечения:

- Требования к производительности:

Описание	Параметры
Рекомендуемый минимальный КТС для namespace	8 CPU / 32GB RAM

- Требования к базе данных (далее по тексту — БД):

Описание	Параметры
Минимальные требования к КТС для разворачивания сервиса	2 Core CPU / 8GB RAM / 100GB HDD
Рекомендуемые требования к КТС для разворачивания сервиса	16 Core CPU / 128GB RAM / 1200GB HDD

Выбор способа установки

В зависимости от способа развертывания сервиса воспользуйтесь одной из инструкций:

- целевая ручная установка сервиса;
- опциональная автоматизированная установка сервиса.

Подготовка окружения

Подключение namespace к компоненту Управление политиками

Подключение namespace к компоненту Управление политиками (POLM, версия 1.12) продукта Synapse Servise Mesh (SSM) осуществляется в соответствии с документацией на него.

Выпуск и установка сертификатов для Ingress/Egress

Используя шаблоны из дистрибутива /sberflake/other/secrets/, выпустите и установите сертификаты Istio Ingress/Egress:

- `istio-egressgateway-ca-certs`;
- `istio-egressgateway-certs`;
- `istio-ingressgateway-ca-certs`;
- `istio-ingressgateway-certs`.

После чего будет создан набор необходимых артефактов, в том числе секреты с сертификатами `cert`, `ca-cert`.

Созданные секреты содержат, в том числе, наименования файлов: `tls.crt`, `tls.key`, `ca-chain.cert.pem` (вся цепочка сертификатов).

В секретах `istio-ingressgateway-ca-certs` и `istio-egressgateway-ca-certs` должен присутствовать один файл `ca-chain.cert.pem` со всей цепочкой сертификатов. Если в секрете указаны два отдельных сертификата, то вызовы через Ingress по HTTPS не будут работать.

Возможен вариант, когда в секретах в файле `ca-chain.cert.pem` содержится вся цепочка сертификатов (т.е. добавлено содержимое из `ca.pem`), но сам файл `ca.pem` не удален. В этом случае все будет работать корректно.

Настройка OTT

Для настройки OTT:

- получите сертификаты приложения, публичный ключ OTT Service и выполните их подключение;
- настройте ConfigMap для OTT-sidecar.

Получение сертификатов приложения, публичного ключа OTT Service и их подключение

Для получения сертификатов приложения, публичного ключа OTT Service и их подключения:

1. При помощи keytool сгенерируйте key-store с ключами (`sberflake_cert.p12`) с помощью алгоритма SHA256withECDSA и длиной ключа 256 бит.

Примеры команд

```
keytool -genkey -keyalg EC -sigalg SHA256withECDSA -keystore sberflake.p12 -storetype PKCS12 -keysize 256 -dname "CN=sberflake" -alias sberflake keytool -certreq -keyalg EC -sigalg SHA256withECDSA -keystore sberflake.p12 -storetype PKCS12 -alias sberflake > sberflake_cert_req.pem
```

2. Отправьте файл `sberflake_cert_req.pem` администратору OTT в заявке на генерацию сертификата для модуля. В заявке явно указать, что сертификат должен быть сгенерирован из приложенного запроса по инструкции. Администратор OTT подписывает/генерирует ключи, высылает их обратно.
3. Импортируйте полученные от администратора OTT ключи в `key-store sberflake_cert.p12`.

Примеры команд

```
keytool -import -keystore sberflake.p12 -storetype PKCS12 -file SberbankPlatformCA_EC.pem -alias SberbankPlatformCA_EC
keytool -import -keystore sberflake.p12 -storetype PKCS12 -file sberflake.pem -alias sberflake
```

4. Создать два секрета:

- `ott-certs` — с сертификатами (sberflake.p12: >- , ott_service_truststore.p12: >-)

Важно! Чтобы все корректно добавилось, необходимо сконвертировать содержимое файла с ключами в Base64. Далее — полученную строку вставить в yml-файл. Либо выполнить команду:

```
oc create secret generic ott-certs --from-file=ott_service_truststore.p12=./sigma_ott_trust.p12 --from-file=sberflake.p12=./sberflake.p12
```

- `ott-passwords` — с паролями к ключам:
  - `OTT_CERTSTORE_PRIVATE_KEY_PWD` — пароль для чтения ключа;

- OTT_CERTSTORE_PWD — пароль для чтения сертификата;
- OTT_TRUST_STORE_PWD — пароль для чтения ott_service_truststore.p12.

ConfigMap для OTT-sidecar

Для работы OTT-sidecar необходимо наличие в дистрибутиве ConfigMap с настройками сервиса. Создание ConfigMap выполняется на этапе установки сервиса.

Конфигурирование параметров ConfigMap для OTT-sidecar выполняется в файле conf/config/parameters.

Инициализация БД

Для инициализации БД от ТУЗ администратора БД выполните следующие действия:

1. Очистите namespace согласно действиям, приведенным в разделе «Удаление».
2. Создайте схему в БД:

- для микросервиса Normal:

```
CREATE SCHEMA IF NOT EXISTS sberflake AUTHORIZATION
as_admin; ALTER DEFAULT PRIVILEGES for user as_admin IN
SCHEMA sberflake GRANT SELECT, INSERT, UPDATE, DELETE ON
TABLES TO sberflake; ALTER DEFAULT PRIVILEGES for user
as_admin IN SCHEMA sberflake GRANT USAGE, SELECT ON
SEQUENCES to sberflake;
```

где as_admin — роль, выдаваемая администраторам БД; sberflake — схема БД микросервиса Normal.

- для микросервиса Shared:

```
CREATE SCHEMA IF NOT EXISTS sberflake_shared AUTHORIZATION
as_admin; ALTER DEFAULT PRIVILEGES for user as_admin IN
SCHEMA sberflake_shared GRANT SELECT, INSERT, UPDATE, DELETE
ON TABLES TO sberflake_shared; ALTER DEFAULT PRIVILEGES for
user as_admin IN SCHEMA sberflake_shared GRANT USAGE, SELECT
ON SEQUENCES to sberflake_shared;
```

где sberflake_shared — схема БД микросервиса Shared.

3. Разархивируйте postgres-db.zip.
4. Выполните скрипты по созданию табличных пространств из каталога /package/db/postgres-db/database-scripts:

- 1-create-folders-for-tablespaces.sh;
- 2-create-tablespaces.sh;
- 3-check-tablespaces-are-created.sh;

- 4-grant-privileges-to-user.sh.

5. Выполните скрипты liquibase согласно `USAGE` (если у скриптов нет права на запуск, то выполните команду `chmod 777 sberflake*.*`).
6. Выполните скрипты для хостов с указанием параметров для основной (MAIN) и репликационной (Stand-In) БД:

```
./sberflake_update.sh <db_host> <db_port> <db_name>
<db_schema> <db_username> <db_password>
./sberflake_shared_update.sh <db_host> <db_port> <db_name>
<db_schema> <db_username> <db_password>
```

7. Предоставьте права доступа пользователям sberflake и sberflake_shared, выполнив команду

```
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA sberflake TO
sberflake; GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA
sberflake_shared TO sberflake_shared; GRANT USAGE ON SCHEMA
sberflake TO sberflake; GRANT USAGE ON SCHEMA
sberflake_shared TO sberflake_shared;
```

где `ALL` — права на использование (Usage) и создание (Create) объектов в БД.

8. Измените параметр `search_path` для основной и репликационной БД, выполнив команду:

```
ALTER ROLE sberflake IN DATABASE <db_name> SET search_path
TO "$user";    ALTER ROLE sberflake_shared IN DATABASE
<db_name> SET search_path TO "$user";
```

где `$user` — имя пользователя БД от соотвествующих ТУЗ, указанных в п.5; `<db_name>` — имя основной или репликационной БД для микросервисов Normal, Shared.

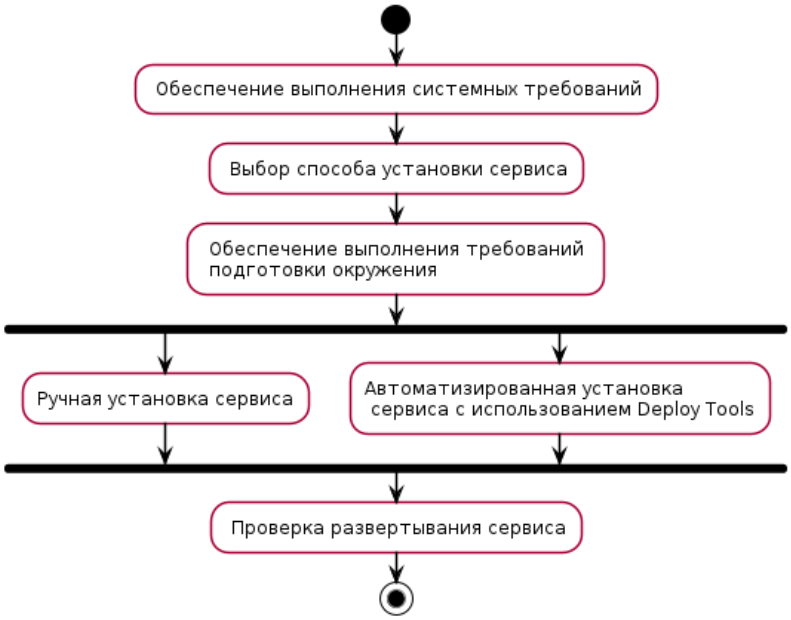
Подготовка окружения для автоматизированной установки с использованием Deploy Tools

Для автоматизированной установки с использованием Deploy Tools дополнительно требуется:

- наличие у сервиса пространства в сервисе централизованного хранения репозиториев исходного кода и прав на создание репозиториев в нем;
- наличие у сервиса области в Jenkins, с настроенными учетными данными (логин/пароль) для авторизации под ТУЗ, позволяющая загружать артефакты в целевой репозиторий хранилища.

Параметры настройки располагаются в файлах в дистрибутиве по пути: package/conf/config/parameters.

Установка



Ручная установка сервиса

Для установки сервиса Генерация ID выполните следующие действия:

- обновите схему БД;
- настройте конфигурационные файлы.

Обновление схемы БД

Схема данных БД может быть обновлена при условии выполненной первичной инициализации БД.

Для обновления схемы БД до текущей версии выполните следующие действия:

1. Заполните конфигурацию подключения к БД, под пользователем-владельцем, в файле (расположен в дистрибутиве): /db/liquibase/nix/liquibase.properties.
2. Разархивируйте postgres-db.zip.
3. В терминале выполните команду для микросервиса Normal

```
./package/db/postgres-db/nix/sberflake_update.sh"
```

3. В терминале выполните команду для микросервиса Shared

```
./package/db/postgres-db/nix/sberflake_shared_update.sh"
```

Настройка конфигурационных файлов

Для ручной установки сервиса в среде контейнеризации необходимо во всех файлах конфигурации заменить все плейсхолдеры на реальные значения.

Перечень конфигурационных файлов для развертывания в Kubernetes:

- /package/conf/k8s/base/sberflake-normal/configmaps/snow-cm-normal-service-logback.yaml;
- /package/conf/k8s/base/sberflake-normal/configmaps/snow-cm-normal-service-fluent-bit.yaml;
- /package/conf/k8s/base/sberflake-normal/configmaps/snow-cm-normal-service-env.yaml;
- /package/conf/k8s/base/sberflake-normal/configmaps/snow-cm-normal-service.yaml;
- /package/conf/k8s/base/sberflake-normal/dc.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-logging-passwords.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-logging-certs.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-db-certs.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-journal-passwords.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-journal-certs.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-db-passwords.yaml;
- /package/conf/k8s/base/sberflake-normal/snow-svc-normal-service.yaml;
- /package/conf/k8s/base/sberflake-configurer/configmaps/snow-cm-configurer-logback.yaml;
- /package/conf/k8s/base/sberflake-configurer/configmaps/snow-cm-configurer-env.yaml;
- /package/conf/k8s/base/sberflake-configurer/configmaps/snow-cm-configurer.yaml;
- /package/conf/k8s/base/sberflake-configurer/configmaps/snow-cm-configurer-fluent-bit.yaml;
- /package/conf/k8s/base/sberflake-configurer/dc.yaml;
- /package/conf/k8s/base/sberflake-configurer/secrets/snow-secret-configurer-db-passwords.yaml;
- /package/conf/k8s/base/sberflake-configurer/secrets/snow-secret-configurer-logging-passwords.yaml;
- /package/conf/k8s/base/sberflake-configurer/secrets/snow-secret-configurer-logging-certs.yaml;
- /package/conf/k8s/base/sberflake-configurer/secrets/snow-secret-configurer-db-certs.yaml;
- /package/conf/k8s/base/sberflake-configurer/snow-svc-configurer.yaml;

- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-master/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-master/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-master/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-1/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-1/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-1/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-5/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-5/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-5/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-2/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-2/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-2/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-3/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-3/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-3/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-4/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-4/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-4/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-1/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-1/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-1/snow-se-dns-service.yaml;

- /package/conf/k8s/base/istio-configs/ingress-gateway/snow-svc-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/snow-cm-ingress-ott-gateway-env.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/snow-cm-ingress-ott-gateway-istio-basic.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/snow-cm-ingress-ott-gateway-logback.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/sberflake-api/snow-vs-sberflake.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/dc.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/snow-gw-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/snow-ef-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configurer-api/snow-vs-configurer.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-route-ingress-gateway-db-loader-health.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-ef-configurer-ingress-symon-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-ef-ingress-symon-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-route-ingress-gateway-mlts-health.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-gw-ingress-mtls-health-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-gateways/snow-gw-ingress-mtls-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-gateways/snow-gw-ingress-http-gateway.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-istio-egressgateway-ca-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-istio-egressgateway-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-ott-passwords-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-istio-ingressgateway-ca-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-ott-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-istio-ingressgateway-certs.yaml;
- /package/conf/k8s/base/istio-configs/http-service-audit/snow-gw-http-service.yaml;

- /package/conf/k8s/base/istio-configs/http-service-audit/snow-vs-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-audit/snow-se-http-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-3/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-3/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-3/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-2/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-2/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-2/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-master/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-master/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-master/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-3/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-3/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-3/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-logger/snow-gw-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-logger/snow-vs-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-logger/snow-se-http-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-2/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-2/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-2/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-standin/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-standin/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-standin/snow-se-ip-service.yaml;

- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-1/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-1/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-1/snow-se-ip-service.yaml;
- /conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-master/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-master/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-master/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-standin/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-standin/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-standin/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-vs-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-ef-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-se-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-dr-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-gw-https-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-2/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-2/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-2/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-5/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-5/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-5/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-logger/snow-vs-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-logger/snow-ef-https-service.yaml;

- /package/conf/k8s/base/istio-configs/https-service-logger/snow-se-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-logger/snow-dr-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-logger/snow-gw-https-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-master/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-master/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-master/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/snow-svc-egress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/configmaps/snow-cm-egress-ott-gateway-logback.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/configmaps/snow-cm-egress-ott-gateway-istio-basic.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/configmaps/snow-cm-egress-ott-gateway-env.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/dc.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-4/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-4/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-4/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-3/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-3/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-3/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-monitoring/snow-gw-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-monitoring/snow-vs-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-monitoring/snow-se-http-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-standin/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-standin/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-standin/snow-se-ip-service.yaml;

- /package/conf/k8s/base/istio-configs/https-service-audit/snow-vs-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-audit/snow-ef-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-audit/snow-se-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-audit/snow-dr-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-audit/snow-gw-https-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-standin/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-standin/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-standin/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-1/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-1/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-1/snow-se-dns-service.yaml;
- /package/conf/k8s/base/prometheus/configmaps/snow-cm-prometheus.yaml;
- /package/conf/k8s/base/prometheus/dc.yaml;
- /package/conf/k8s/base/prometheus/snow-se-prometheus-kubeapi-ext.yaml;
- /package/conf/k8s/base/sberflake-shared/configmaps/snow-cm-shared-service-env.yaml;
- /package/conf/k8s/base/sberflake-shared/configmaps/snow-cm-shared-service-fluent-bit.yaml;
- /package/conf/k8s/base/sberflake-shared/configmaps/snow-cm-shared-service.yaml;
- /package/conf/k8s/base/sberflake-shared/configmaps/snow-cm-shared-service-logback.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-logging-passwords.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-db-passwords.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-journal-certs.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-logging-certs.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-journal-passwords.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-db-certs.yaml;
- /package/conf/k8s/base/sberflake-shared/snow-svc-shared-service.yaml;
- /package/conf/k8s/base/sberflake-shared/withJrpc/dc.yaml;

- /package/conf/k8s/base/sberflake-shared/withJrpc/snow-cm-shared-service-jrpc-config.yaml;
- /package/conf/k8s/base/sberflake-shared/withJrpc/snow-route-shared-service-jrpc-http.yaml;
- /package/conf/k8s/base/sberflake-shared/noJrpc/dc.yaml.

Перечень конфигурационных файлов для развертывания в OpenShift:

- /package/conf/k8s/base/sberflake-normal/overrides/openshift/testroute/snow-route-normal-service-http.yaml;
- /package/conf/k8s/base/sberflake-normal/configmaps/snow-cm-normal-service-logback.yaml;
- /package/conf/k8s/base/sberflake-normal/configmaps/snow-cm-normal-service-fluent-bit.yaml;
- /package/conf/k8s/base/sberflake-normal/configmaps/snow-cm-normal-service-env.yaml;
- /package/conf/k8s/base/sberflake-normal/configmaps/snow-cm-normal-service.yaml;
- /package/conf/k8s/base/sberflake-normal/dc.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-logging-passwords.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-logging-certs.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-db-certs.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-journal-passwords.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-journal-certs.yaml;
- /package/conf/k8s/base/sberflake-normal/secrets/snow-secret-normal-service-db-passwords.yaml;
- /package/conf/k8s/base/sberflake-normal/snow-svc-normal-service.yaml;
- /package/conf/k8s/base/overrides/openshift/sberflake-normal/testroute/snow-route-normal-service-http.yaml;
- /package/conf/k8s/base/overrides/openshift/sberflake-configurer/testroute/snow-route-configurer-http.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/snow-route-ingress-ott-gateway-mlts.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/snow-route-ingress-ott-gateway-http.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/snow-route-ingress-ott-gateway-db-loader-mtls.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/snow-route-ingress-ott-gateway-ott.yaml;

- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/snow-route-ingress-ott-gateway-db-loader.yaml;
- /package/conf/k8s/base/overrides/openshift/istio-configs/ingress-gateway/snow-route-ingress-ott-gateway-db-loader-ott.yaml;
- /package/conf/k8s/base/overrides/openshift/sberflake-shared/testroutes/snow-route-shared-service-http.yaml;
- /package/conf/k8s/base/sberflake-configurer/overrides/openshift/testroute/snow-route-configurer-http.yaml;
- /package/conf/k8s/base/sberflake-configurer/configmaps/snow-cm-configurer-logback.yaml;
- /package/conf/k8s/base/sberflake-configurer/configmaps/snow-cm-configurer-env.yaml;
- /package/conf/k8s/base/sberflake-configurer/configmaps/snow-cm-configurer.yaml;
- /package/conf/k8s/base/sberflake-configurer/configmaps/snow-cm-configurer-fluent-bit.yaml;
- /package/conf/k8s/base/sberflake-configurer/dc.yaml;
- /package/conf/k8s/base/sberflake-configurer/secrets/snow-secret-configurer-db-passwords.yaml;
- /package/conf/k8s/base/sberflake-configurer/secrets/snow-secret-configurer-logging-passwords.yaml;
- /package/conf/k8s/base/sberflake-configurer/secrets/snow-secret-configurer-logging-certs.yaml;
- /package/conf/k8s/base/sberflake-configurer/secrets/snow-secret-configurer-db-certs.yaml;
- /package/conf/k8s/base/sberflake-configurer/snow-svc-configurer.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-master/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-master/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-master/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-1/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-1/snow-gw-ip-service.yaml;
- /package /conf/k8s/base/istio-configs/ip-service-sberflake-journal-1/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-5/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-5/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-5/snow-se-ip-service.yaml;

- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-2/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-2/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-2/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-3/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-3/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-3/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-4/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-4/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-4/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-1/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-1/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-1/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/overrides/openshift/snow-route-ingress-ott-gateway-mlts.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/overrides/openshift/snow-route-ingress-ott-gateway-http.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/overrides/openshift/snow-route-ingress-ott-gateway-db-loader-mtls.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/overrides/openshift/snow-route-ingress-ott-gateway-ott.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/overrides/openshift/snow-route-ingress-ott-gateway-db-loader.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/overrides/openshift/snow-route-ingress-ott-gateway-db-loader-ott.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/snow-svc-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/snow-cm-ingress-ott-gateway-env.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/snow-cm-ingress-ott-gateway-istio-basic.yaml;

- /package/conf/k8s/base/istio-configs/ingress-gateway/configmaps/snow-cm-ingress-ott-gateway-logback.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/sberflake-api/snow-vs-sberflake.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/dc.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/snow-gw-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/snow-ef-ingress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/configurer-api/snow-vs-configurer.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-route-ingress-gateway-db-loader-health.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-ef-configurer-ingress-symon-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-ef-ingress-symon-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-route-ingress-gateway-mlts-health.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-health-gateway/snow-gw-ingress-mtls-health-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-gateways/snow-gw-ingress-mtls-gateway.yaml;
- /package/conf/k8s/base/istio-configs/ingress-gateway/not-ott-gateways/snow-gw-ingress-http-gateway.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-istio-egressgateway-ca-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-istio-egressgateway-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-ott-passwords-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-istio-ingressgateway-ca-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-ott-certs.yaml;
- /package/conf/k8s/base/istio-configs/secrets/snow-secret-istio-ingressgateway-certs.yaml;
- /package/conf/k8s/base/istio-configs/http-service-audit/snow-gw-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-audit/snow-vs-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-audit/snow-se-http-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-3/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-3/snow-gw-dns-service.yaml;

- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-3/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-2/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-2/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-2/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-master/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-master/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-master/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-3/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-3/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-3/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-logger/snow-gw-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-logger/snow-vs-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-logger/snow-se-http-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-2/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-2/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-journal-2/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-standin/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-standin/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-standin/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-1/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-1/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-logging-1/snow-se-ip-service.yaml;

- /conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-master/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-master/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-master/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-standin/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-standin/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-normal-db-standin/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-vs-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-ef-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-se-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-dr-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-monitoring/snow-gw-https-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-2/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-2/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-2/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-5/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-5/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-5/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-logger/snow-vs-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-logger/snow-ef-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-logger/snow-se-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-logger/snow-dr-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-logger/snow-gw-https-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-master/snow-vs-ip-service.yaml;

- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-master/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-normal-db-master/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/snow-svc-egress-ott-gateway.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/configmaps/snow-cm-egress-ott-gateway-logback.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/configmaps/snow-cm-egress-ott-gateway-istio-basic.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/configmaps/snow-cm-egress-ott-gateway-env.yaml;
- /package/conf/k8s/base/istio-configs/egress-gateway/dc.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-4/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-4/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-4/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-3/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-3/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-logging-3/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-monitoring/snow-gw-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-monitoring/snow-vs-http-service.yaml;
- /package/conf/k8s/base/istio-configs/http-service-monitoring/snow-se-http-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-standin/snow-vs-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-standin/snow-gw-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/ip-service-sberflake-shared-db-standin/snow-se-ip-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-audit/snow-vs-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-audit/snow-ef-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-audit/snow-se-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-audit/snow-dr-https-service.yaml;
- /package/conf/k8s/base/istio-configs/https-service-audit/snow-gw-https-service.yaml;

- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-standin/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-standin/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-shared-db-standin/snow-se-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-1/snow-vs-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-1/snow-gw-dns-service.yaml;
- /package/conf/k8s/base/istio-configs/dns-service-sberflake-journal-1/snow-se-dns-service.yaml;
- /package/conf/k8s/base/prometheus/configmaps/snow-cm-prometheus.yaml;
- /package/conf/k8s/base/prometheus/dc.yaml;
- /package/conf/k8s/base/prometheus/snow-se-prometheus-kubeapi-ext.yaml;
- /package/conf/k8s/base/sberflake-shared/overrides/openshift/testroutes/snow-route-shared-service-http.yaml;
- /package/conf/k8s/base/sberflake-shared/configmaps/snow-cm-shared-service-env.yaml;
- /package/conf/k8s/base/sberflake-shared/configmaps/snow-cm-shared-service-fluent-bit.yaml;
- /package/conf/k8s/base/sberflake-shared/configmaps/snow-cm-shared-service.yaml;
- /package/conf/k8s/base/sberflake-shared/configmaps/snow-cm-shared-service-logback.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-logging-passwords.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-db-passwords.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-journal-certs.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-logging-certs.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-journal-passwords.yaml;
- /package/conf/k8s/base/sberflake-shared/secrets/snow-secret-shared-service-db-certs.yaml;
- /package/conf/k8s/base/sberflake-shared/snow-svc-shared-service.yaml;
- /package/conf/k8s/base/sberflake-shared/withJrpc/dc.yaml;
- /package/conf/k8s/base/sberflake-shared/withJrpc/snow-cm-shared-service-jrpc-config.yaml;
- /package/conf/k8s/base/sberflake-shared/withJrpc/snow-route-shared-service-jrpc-http.yaml;

- /package/conf/k8s/base/sberflake-shared/noJrpc/dc.yaml.

Пример в файле `conf/config/parameters`.

После замены файлов конфигурации необходимо развернуть файлы конфигурации в среду контейнеризации:

1. В панели администратора в правом верхнем углу нажать на кнопку **Import YAML**;
2. в открывшейся окно вставить скопированный конфигурационный файл и нажать **Create**;
3. после создания отобразиться страница конфигурационного файла;
4. повторить действия 1-3 для всех файлов конфигурации.

Настройка интеграции со смежными сервисами

Настройка интеграции со смежными сервисами происходит путем конфигурации параметров для конкретного внешнего сервиса в определенных ConfigMaps:

- настройка интеграции с компонентом Аудит продукта Platform V Audit;
- настройка интеграции с компонентом Объединенный мониторинг Unimon продукта Platform V Monitor;
- настройка интеграции с компонентом Журналирование продукта Platform V Monitor;
- настройка интеграции с компонентом One-Time Password (OTP) / OTT;
- настройка интеграции с компонентом Прикладной журнал.

Настройка интеграции с компонентом Аудит продукта Platform V Audit

Для настройки интеграции сервиса Генерация ID с сервисом Аудит в папке `conf/config/parameters` сконфигурируйте следующие параметры:

```
# <!- заполните самостоятельно -> - значения параметров
необходимо заполнить самостоятельно перед развертыванием
сервиса ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_PORT_NAME=http-
audit ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_PORT_NUMBER=<!-
заполните самостоятельно ->
ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_TARGETPORT_NUMBER=<!-
заполните самостоятельно ->  AUTONUMPATTERNS_AUDIT_STUB=true
AUTONUMPATTERNS_AUDIT_RESOURCENAME=<!- заполните
самостоятельно -> AUTONUMPATTERNS_AUDIT_HOST=<!- заполните
самостоятельно ->
AUTONUMPATTERNS_AUDIT_BULKHEAD_MAXWAITDURATION=<!- заполните
самостоятельно ->
AUTONUMPATTERNS_AUDIT_BULKHEAD_MAXCONCURRENTCALLS=<!-
заполните самостоятельно ->  AUTONUMCOUNTERS_AUDIT_STUB=<!-
```

```
заполните самостоятельно: true/false -> #
Включение/отключение режима загулшки для сервиса Аудит
AUTONUMCOUNTERS_AUDIT_HOST=<!- заполните самостоятельно -> #
Адрес сервиса Аудит
AUTONUMCOUNTERS_AUDIT_BULKHEAD_MAXWAITDURATION=<!- заполните
самостоятельно -> # Максимальное время ожидания паттерна
bulkhead для Аудита
AUTONUMCOUNTERS_AUDIT_BULKHEAD_MAXCONCURRENTCALLS=<!-
заполните самостоятельно -> # Максимальное количество
одновременных вызовов Аудита
```

Проверку успешности проведения интеграции сервиса Генерация ID с внешними сервисами выполняйте в консоли сервиса, с которым выполняется интеграция.

Пример заполнения для интеграции с Platform V Audit

```
# <!- заполните самостоятельно -> - значения параметров
необходимо заполнить самостоятельно перед развертыванием
сервиса ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_PORT_NAME=http-
audit ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_PORT_NUMBER=8001
ISTIOCONFIGS_EGRESSGATEWAY_AUDIT_TARGETPORT_NUMBER=8001
AUTONUMPATTERNS_AUDIT_STUB=true
AUTONUMPATTERNS_AUDIT_RESOURCENAME=platformv:core:1
AUTONUMPATTERNS_AUDIT_HOST=http://audit2-client-
proxy.apps.dev-gen.xxxxx.xxxx.ru:8080
AUTONUMPATTERNS_AUDIT_BULKHEAD_MAXWAITDURATION=5000ms
AUTONUMPATTERNS_AUDIT_BULKHEAD_MAXCONCURRENTCALLS=5
AUTONUMCOUNTERS_AUDIT_STUB=true # Включение/отключение
режима загулшки для сервиса Аудит
AUTONUMCOUNTERS_AUDIT_HOST=http://audit2-client-
proxy.apps.dev-gen.xxxxx.xxxx.ru:8080 # Адрес сервиса Аудит
AUTONUMCOUNTERS_AUDIT_BULKHEAD_MAXWAITDURATION=5000ms #
Максимальное время ожидания паттерна bulkhead для Аудита
AUTONUMCOUNTERS_AUDIT_BULKHEAD_MAXCONCURRENTCALLS=5
```

Настройка в случае развертывания без сервиса Platform V Audit

Для настройки сервиса Генерация ID без интеграции сервисом Аудит в `conf/config/parameters` сконфигурируйте следующие параметры:

```
AUTONUMCOUNTERS_AUDIT_STUB=false # Отключение режима
загулшки для сервиса Аудит AUTONUMPATTERNS_AUDIT_STUB=false
```

Настройка интеграции с компонентом Объединенный мониторинг Unimon продукта Platform V Monitor

Пререквизиты интеграции с Platform V Monitor Объединенный мониторинг Unimon

Пререквизитами настройки интеграции сервиса Генерация ID с сервисом Мониторинг является создание в среде контейнеризации Service Account и Role Binding, необходимых для корректного развертывания deployment-config prometheus в среде контейнеризации.

Развертывание конфигурации Prometheus

Если namespace настроен так, что трафик от Prometheus не проходит, то добавьте ServiceEntry:

```
- apiVersion: networking.istio.io/v1alpha3     kind:
ServiceEntry     metadata:       name: prometheus-se
labels:         app: prometheus     spec:       exportTo:
- .       hosts:       - ${APPMON_HOST}       ports:       -
name: http         number: 80         protocol: HTTP       -
name: https         number: 443         protocol: TLS
resolution: DNS
```

И параметр:

```
- name: APPMON_HOST     displayName: "Applied Monitoring
host"     description: "Хост для правила исходящего трафика.
Обязательный."     required: true
```

Пример значения параметра:

```
APPMON_HOST=appmon-ift.apps.dev-gen.ca.xxxxx.ru
```

Настройка интеграции с сервисом Мониторинг

Для настройки интеграции сервиса Генерация ID с сервисом Мониторинг в conf/config/parameters сконфигурируйте следующие параметры.

```
# <!- заполните самостоятельно -> - значения параметров
необходимо заполнить самостоятельно перед развертыванием
сервиса  ISTIOCONFIGS_HTTPSERVICE_MONITORING_NAME=monitoring
ISTIOCONFIGS_HTTPSERVICE_MONITORING_HOST=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_INTERNAL=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_GATEWAY=<!-
заполните самостоятельно ->
```

```
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_TARGET=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_TIMEOUT=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_RETRIES_ATTEMPTS=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSERVICE_MONITORING_RETRIES_PERTRYTIMEOUT=<!
- заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_NAME=monitoring
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_HOST=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_INTERNAL=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_GATEWAY=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_TARGET=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_TCP_MAXCONNECTIONS=<!- заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_HTTP_MAXPENDREQ=<!- заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_HTTP_MAXREQUESTSPERCONNECTION=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_CONSECUTIVE5XXERRORS=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_INTERVAL=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_BASEEJECTIONTIME=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_TIMEOUT=<!- заполните
самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_RETRIES_ATTEMPTS=<!-
заполните самостоятельно ->
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_RETRIES_PERTRYTIMEOUT=<
!- заполните самостоятельно ->
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_PORT_NAME=http-
monitoring
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_PORT_NUMBER=<!-
заполните самостоятельно ->
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_TARGETPORT_NUMBER=<!-
заполните самостоятельно ->
```

При необходимости внесения изменений в интеграцию Генерация ID с внешними сервисами внесите соответствующие правки в `conf/config/parameters`.

Пример заполнения conf/config/parameters для интеграции с продуктом Прикладной мониторинг компонента Platform V Monitor

```
ISTIOCONFIGS_HTTPSERVICE_MONITORING_NAME=monitoring
ISTIOCONFIGS_HTTPSERVICE_MONITORING_HOST=appmon-
service.apps.dev-gen2.sigma.sbrf.ru
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_INTERNAL=8080
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_GATEWAY=8002
ISTIOCONFIGS_HTTPSERVICE_MONITORING_PORTS_TARGET=80
ISTIOCONFIGS_HTTPSERVICE_MONITORING_TIMEOUT=5s
ISTIOCONFIGS_HTTPSERVICE_MONITORING_RETRIES_ATTEMPTS=1
ISTIOCONFIGS_HTTPSERVICE_MONITORING_RETRIES_PERTRYTIMEOUT=5s
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_NAME=monitoring
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_HOST=appmon-
service.apps.dev-gen2.sigma.sbrf.ru
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_INTERNAL=8080
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_GATEWAY=8002
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_PORTS_TARGET=443
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_TCP_MAXCONNECTIONS=5
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_HTTP_MAXPENDREQ=100
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_HTTP_MAXREQUESTSPERCONNECTION=30
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_CONSECUTIVE5XXERRORS=10
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_INTERVAL=5m
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_DESTINATIONRULE_TRAFFIC
POLICY_OUTLIERDETECTION_BASEEJECTIONTIME=5m
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_TIMEOUT=5s
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_RETRIES_ATTEMPTS=1
ISTIOCONFIGS_HTTPSSERVICE_MONITORING_RETRIES_PERTRYTIMEOUT=5
s  ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_PORT_NAME=http-
monitoring
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_PORT_NUMBER=8002
ISTIOCONFIGS_EGRESSGATEWAY_MONITORING_TARGETPORT_NUMBER=8002
```

Настройка интеграции с компонентом Журналирование продукта Platform V Monitor

При ручном развертывании сервиса Генерация ID выполните следующие действия:

1. В файле `configmap-fluentbit.yml` для плейсхолдера `{{LOGGER_SERVICE_HOST_NAME}}`, `{{LOGGER_SERVICE_PORT}}` установите соответствующие значения для сервиса Журналирование, развернутого в целевом окружении.

2. Загрузите в среду контейнеризации сконфигурированный файл `configmap-fluentbit.yml`.

Для настройки интеграции сервиса Генерация ID с сервисом Журналирование в `conf/config/parameters` сконфигурируйте следующие параметры:

```
# <!- заполните самостоятельно -> - значения параметров
необходимо заполнить самостоятельно перед развертыванием
сервиса  SERVICERN=platformv:core:<!- заполните
самостоятельно -> # Resource name из-под которого работает
сервис AUTONUMCONFIGURER_LOGGING_KAFKA_BROKERS=<!- заполните
самостоятельно -> # Адреса серверов Kafka для откидывания
логов AUTONUMCONFIGURER_LOGGING_KAFKA_TOPICS=<!- заполните
самостоятельно -> # Топик для откидывания логов в Kafka
AUTONUMCONFIGURER_LOGGING_KAFKA_SECURITYPROTOCOL=<!-
заполните самостоятельно -> # Протокол Kafka для откидывания
логов   AUTONUMCOUNTERS_LOGGING_KAFKA_BROKERS=<!- заполните
самостоятельно -> # Адреса серверов Kafka для откидывания
логов AUTONUMCOUNTERS_LOGGING_KAFKA_TOPICS=<!- заполните
самостоятельно -> # Топик для откидывания логов в Kafka
AUTONUMCOUNTERS_LOGGING_KAFKA_SECURITYPROTOCOL=<!- заполните
самостоятельно -> # Протокол Kafka для откидывания логов
AUTONUMCOUNTERS_LOGGING_REGISTRY_REPOSITORY
AUTONUMPATTERNS_LOGGING_KAFKA_BROKERS=<!- заполните
самостоятельно -> # Адреса серверов Kafka для откидывания
логов AUTONUMPATTERNS_LOGGING_KAFKA_TOPICS=<!- заполните
самостоятельно -> # Топик для откидывания логов в Kafka
AUTONUMPATTERNS_LOGGING_KAFKA_SECURITYPROTOCOL=<!- заполните
самостоятельно -> | PLAINTEXT # Протокол Kafka для
откидывания логов
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
```

```
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
```

```
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_IP=<!- заполните
самостоятельно -> # IP-адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_NAME=<!- заполните
самостоятельно -> # Имя для Istio конфигурации
```

```
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_HOST=<!- заполните
самостоятельно -> # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_INTERNAL=<!-
заполните самостоятельно -> # Внутренний порт для
подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_GATEWAY=<!-
заполните самостоятельно -> # Порт на Gateway для проброса
соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_TARGET=<!-
заполните самостоятельно -> # Целевой порт ресурса для
подключения
```

При необходимости внесения изменений в интеграцию сервиса Генерация ID с внешними сервисами внесите соответствующие правки в conf/config/parameters.

Проверку успешности проведения интеграции в сервис Генерация ID с внешними сервисами выполняйте в консоли сервиса, с которым выполняется интеграция.

Пример заполнения conf/config/parameters для интеграции с сервисом Журналирование

```
SERVICERN=platformv:core:1 # Resource name из-под которого
работает сервис
AUTONUMCONFIGURER_LOGGING_KAFKA_BROKERS=10.106.228.11:18082
# Адреса серверов Kafka для откидывания логов
AUTONUMCONFIGURER_LOGGING_KAFKA_TOPICS=default # Топик для
откидывания логов в Kafka
AUTONUMCONFIGURER_LOGGING_KAFKA_SECURITYPROTOCOL=SSL #
Протокол Kafka для откидывания логов
AUTONUMCOUNTERS_LOGGING_KAFKA_BROKERS=10.106.228.11:18082 #
Адреса серверов Kafka для откидывания логов
AUTONUMCOUNTERS_LOGGING_KAFKA_TOPICS=default # Топик для
откидывания логов в Kafka
AUTONUMCOUNTERS_LOGGING_KAFKA_SECURITYPROTOCOL=SSL |
PLAINTEXT # Протокол Kafka для откидывания логов
AUTONUMCOUNTERS_LOGGING_REGISTRY_REPOSITORY
AUTONUMPATTERNS_LOGGING_KAFKA_BROKERS=10.106.228.11:18082 #
Адреса серверов Kafka для откидывания логов
AUTONUMPATTERNS_LOGGING_KAFKA_TOPICS=default # Топик для
откидывания логов в Kafka
AUTONUMPATTERNS_LOGGING_KAFKA_SECURITYPROTOCOL=SSL |
PLAINTEXT # Протокол Kafka для откидывания логов
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_NAME=logging-1 #
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_IP=10.0.0.1 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_INTERNAL=1808
2 # Внутренний порт для подключения из сервиса
```

```
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_GATEWAY=12808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_1_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_NAME=logging-1 #
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_HOST=logging-
1.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_INTERNAL=180
82 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_GATEWAY=1280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_1_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_NAME=logging-2 #
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_IP=10.0.0.2 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_INTERNAL=2808
2 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_GATEWAY=22808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_2_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_NAME=logging-2 #
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_HOST=logging-
2.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_INTERNAL=280
82 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_GATEWAY=2280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_2_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_NAME=logging-3 #
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_IP=10.0.0.3 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_INTERNAL=3808
2 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_GATEWAY=32808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_3_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_NAME=logging-3 #
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_HOST=logging-
3.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_INTERNAL=380
82 # Внутренний порт для подключения из сервиса
```

```
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_GATEWAY=3280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_3_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_NAME=logging-4 #
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_IP=10.0.0.4 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_INTERNAL=4808
2 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_GATEWAY=42808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_4_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_NAME=logging-4 #
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_HOST=logging-
4.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_INTERNAL=480
82 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_GATEWAY=4280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_4_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_NAME=logging-5 #
Имя для Istio конфигурации
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_IP=10.0.0.5 # IP-
адрес для подключения
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_INTERNAL=5808
2 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_GATEWAY=52808
# Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_IPSERVICE_AUTONUM_LOGGING_5_PORTS_TARGET=9092 #
Целевой порт ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_NAME=logging-5 #
Имя для Istio конфигурации
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_HOST=logging-
5.stub # Хост ресурса для подключения
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_INTERNAL=580
82 # Внутренний порт для подключения из сервиса
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_GATEWAY=5280
8 # Порт на Gateway для проброса соединения через Egress
ISTIOCONFIGS_DNSSERVICE_AUTONUM_LOGGING_5_PORTS_TARGET=9092
# Целевой порт ресурса для подключения
```

Настройка интеграции с компонентом Прикладной журнал продукта Platform V Data Tools

Для настройки интеграции сервиса Генерация ID с компонентом Прикладной журнал:

- Заведите заявку на администраторов компонента Прикладной журнал на стенде на подключение ZONE_ID в конфигураторе со значением ANUM.
- Обратитесь к администраторам компонента Прикладной журнал для выпуска сертификатов: server.keystore.jks и trust.jks.
- Укажите значения вместо плейсхолдеров (плейсхолдеры заключены в двойные фигурные скобки {{ }}).

```
ZONE_ID сервиса регистрируемая в Configurator силами
администраторов Прикладного Журнала AJ_STAND_IN_ZONE_ID=SCHD
URL MMT. Пример значения: stub_url - если включен режим stub
в параметре AJ_STAND_IN_STUB_MODE_FLAG=true, то любое
значение, иначе необходимо указать корректный URL ПЖ,
который установлен на стенде. AJ_MMT_PROXY_URL=<Обязательно
к заполнению>   Время для выполнения запроса на сервер ПЖ в
секундах AJ_MMT_CALL_TIMEOUT=30  Флаг использования заглушки
вместо полноценного Stand-In. true - интеграция с ПЖ
отключена, false - интеграция с ПЖ включена.
AJ_STAND_IN_STUB_MODE_FLAG=true   Сервер Kafka. Если включен
режим stub в параметре AJ_STAND_IN_STUB_MODE_FLAG=true, то
любое значение, иначе необходимо указать корректный сервер
ПЖ, который установлен на стенде.
AJ_KAFKA_BOOTSTRAP_SERVERS=<Обязательно к заполнению>
Протокол взаимодействия с Kafka PLAINTEXT для http endpoint
ПЖ, для https значение SSL AJ_KAFKA_SECURITY_PROTOCOL=SSL
```

Пример заполненного файла configmap.yml

```
standin:   cloud:     client:       zoneId:
{{AJ_STAND_IN_ZONE_ID}}       mmtProxyUrl:
{{AJ_MMT_PROXY_URL}}       mmtCallTimeout:
{{AJ_MMT_CALL_TIMEOUT}}       stub:
{{AJ_STAND_IN_STUB_MODE_FLAG}}       heartBeatPeriod: 1000
kafka:         bootstrapServers:
{{AJ_KAFKA_BOOTSTRAP_SERVERS}}         producerConfig:
"[security.protocol]": {{AJ_KAFKA_SECURITY_PROTOCOL}}
"[ssl.key.password]": ${AJ_SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/app/certs/journal/server.keystore.jks
"[ssl.keystore.password]": ${AJ_SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /app/certs/journal/trust.jks
"[ssl.truststore.password]": ${AJ_SSL_TRUSTSTORE_PASSWORD}
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS           "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
consumerConfig:           "[security.protocol]":
{{AJ_KAFKA_SECURITY_PROTOCOL}}
```

```
"[ssl.key.password]": ${AJ_SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/app/certs/journal/server.keystore.jks
"[ssl.keystore.password]": ${AJ_SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /app/certs/journal/trust.jks
"[ssl.truststore.password]": ${AJ_SSL_TRUSTSTORE_PASSWORD}
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS          "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
concurrency: 10       groupId: group_1
```

Настройка интеграции с компонентом One-Time Password (OTP) / OTT

Для настройки интеграции сервиса Генерация ID с OTT в `conf/config/parameters` сконфигурируйте параметры для микросервисов Counters, Patterns, Configurer:

```
AUTONUMPATTERNS_ISTIO_GATEWAY_NAME=anum-gw-ingress-ott-
gateway AUTONUMPATTERNS_ISTIO_GATEWAY_ROUTES_OTTHOST=<!-
заполните самостоятельно. Например: autonum-dev-
ott.ingress.apps.dev-gen2.sigma.sbrf.ru ->
AUTONUMCONFIGURER_ISTIO_GATEWAY_NAME=anum-gw-ingress-ott-
gateway AUTONUMCONFIGURER_ISTIO_GATEWAY_ROUTES_OTTHOST=<!-
заполните самостоятельно. Например: autonum-configurer-dev-
ott.ingress.apps.dev-gen2.sigma.sbrf.ru ->
AUTONUMCOUNTERS_ISTIO_GATEWAY_NAME=anum-gw-ingress-ott-
gateway AUTONUMCOUNTERS_ISTIO_GATEWAY_ROUTES_OTTHOST=<!-
заполните самостоятельно. Например: autonum-dev-
ott.ingress.apps.dev-gen2.sigma.sbrf.ru ->
ISTIOCONFIGS_OTTINTEGRATION_REGISTRYREPOSITORY=<!- заполните
самостоятельно. Например:
registry.sigma.sbrf.ru/pprb/ci00641491/ci01125613_ott ->
ISTIOCONFIGS_OTTINTEGRATION_OTTSERVICECERTALIAS=<!-
заполните самостоятельно. Например: ott-service ->
ISTIOCONFIGS_OTTINTEGRATION_OTTSERVICEHOSTS=<!- заполните
самостоятельно. Например: 10.53.99.178:8443,10.53.96.30:8443
-> ISTIOCONFIGS_OTTINTEGRATION_OTTSERVICEURL=<!- заполните
самостоятельно. Например: https://10.53.99.178:8443/ott-
service/rest/token ->
ISTIOCONFIGS_OTTINTEGRATION_OTTAUTHZREALM=<!- заполните
самостоятельно. Например: mmt ->
ISTIOCONFIGS_OTTINTEGRATION_OTTCLIENTCERTALIAS=<!- заполните
самостоятельно. Например: autonum-os ->
ISTIOCONFIGS_OTTINTEGRATION_OTTMODULEID=<!- заполните
самостоятельно. Например: autonum-os ->
ISTIOCONFIGS_OTTINTEGRATION_OTTOPERMODE_EGRESS=<!- заполните
```

```
самостоятельно. Например: sign ->
ISTIOCONFIGS_OTTINTEGRATION_OTTOPERMODE_INGRESS=<!-
заполните самостоятельно. Например: validate ->
```

При необходимости внесения изменений в интеграцию сервиса Генерация ID с внешними сервисами внесите соответствующие правки в `conf/config/parameters`.

Проверку успешности проведения интеграции сервиса Генерация ID с внешними сервисами выполняйте в консоли сервиса, с которым выполняется интеграция.

Настройка интеграции с компонентом Прикладной журнал

Для настройки интеграции сервиса Генерация ID с компонентом Прикладной журнал:

- Заведите заявку на администраторов компонента Прикладной журнал на стенде на подключение ZONE_ID в конфигураторе со значением `ANUM`.
- Обратитесь к администраторам компонента Прикладной журнал для выпуска сертификатов: `server.keystore.jks` и `trust.jks`.
- Укажите значения вместо плейсхолдеров (плейсхолдеры заключены в двойные фигурные скобки {{ }}).

```
ZONE_ID сервиса регистрируемая в Configurator силами
администраторов Прикладного Журнала AJ_STAND_IN_ZONE_ID=SCHD
URL MMT. Пример значения: stub_url - если включен режим stub
в параметре AJ_STAND_IN_STUB_MODE_FLAG=true, то любое
значение, иначе необходимо указать корректный URL ПЖ,
который установлен на стенде. AJ_MMT_PROXY_URL=<Обязательно
к заполнению>  Время для выполнения запроса на сервер ПЖ в
секундах AJ_MMT_CALL_TIMEOUT=30  Флаг использования заглушки
вместо полноценного Stand-In. true - интеграция с ПЖ
отключена, false - интеграция с ПЖ включена.
AJ_STAND_IN_STUB_MODE_FLAG=true  Сервер Kafka. Если включен
режим stub в параметре AJ_STAND_IN_STUB_MODE_FLAG=true, то
любое значение, иначе необходимо указать корректный сервер
ПЖ, который установлен на стенде.
AJ_KAFKA_BOOTSTRAP_SERVERS=<Обязательно к заполнению>
Протокол взаимодействия с Kafka PLAINTEXT для http endpoint
ПЖ, для https значение SSL AJ_KAFKA_SECURITY_PROTOCOL=SSL
```

Пример заполненного файла `configmap.yml`

```
standin:   cloud:     client:       zoneId:
{{AJ_STAND_IN_ZONE_ID}}       mmtProxyUrl:
{{AJ_MMT_PROXY_URL}}       mmtCallTimeout:
```

```
{{AJ_MMT_CALL_TIMEOUT}}       stub:
{{AJ_STAND_IN_STUB_MODE_FLAG}}       heartBeatPeriod: 1000
kafka:         bootstrapServers:
{{AJ_KAFKA_BOOTSTRAP_SERVERS}}         producerConfig:
"[security.protocol]": {{AJ_KAFKA_SECURITY_PROTOCOL}}
"[ssl.key.password]": ${AJ_SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/app/certs/journal/server.keystore.jks
"[ssl.keystore.password]": ${AJ_SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /app/certs/journal/trust.jks
"[ssl.truststore.password]": ${AJ_SSL_TRUSTSTORE_PASSWORD}
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS           "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
consumerConfig:           "[security.protocol]":
{{AJ_KAFKA_SECURITY_PROTOCOL}}
"[ssl.key.password]": ${AJ_SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/app/certs/journal/server.keystore.jks
"[ssl.keystore.password]": ${AJ_SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /app/certs/journal/trust.jks
"[ssl.truststore.password]": ${AJ_SSL_TRUSTSTORE_PASSWORD}
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS           "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
concurrency: 10       groupId: group_1
```

Автоматизированная установка сервиса с использованием Deploy Tools

Для установки сервиса с использованием компонента Deploy Tools (CDJE) программного продукта Platform V DevOps Tools (DOT) разверните сервис в инфраструктуре и запустите CD Job по инструкции, приведенной в документации Platform V DevOps Tools (DOT).

Обновление

Специальной процедуры обновления сервиса не предусмотрено.

Для обновления сервиса:

1. Удалите сервис согласно процедуре, описанной в разделе «Удаление».
2. Установите сервиса согласно процедуре, описанной в разделе «Установка».

Удаление

Очистите namespace, выполнив команду:

```
{ oc get -o name gateway.networking.istio.io; oc get -o name
virtualservice; oc get -o name envoyfilter; oc get -o name
destinationrule.networking.istio.io; oc get -o name
serviceentry; oc get -o name deployment; oc get -o name
configmap; oc get -o name service; oc get -o name route; } |
xargs oc delete
```

Проверка работоспособности

Через вызов API

Для проверки работоспособности сервиса отправьте запросы на микросервисы:

- Configurer;
- Shared;
- Normal.

Проверка работоспособности микросервиса Configurer

Для проверки работоспособности микросервиса Configurer выполните:

1. Запрос на создание resource:

```
curl -v -X POST
'[]/sberflake/v3/rn/admin/configurer/resources' --header
'Content-Type: application/json' --header 'ott-invoker:
invoker' --header 'SUBJECT: subjectrn' --data-raw '{
"resource_name": "temporaryresource"}'
```

Ответ:

```
{"message":"resource was created"}
```

2. Запрос на создание сущности:

```
curl -v -X POST
'[]/sberflake/v3/rn/admin/configurer/entities' --header
'ott-invoker: ott-client-example' --header 'Content-Type:
application/json' --header 'SUBJECT: subjectrn' --data-raw
'{ "name": "johnSnowNewShared", "nodeBits": 12,
"resource_name": "default"}'
```

Ответ:

```
{"message":"Entity was created"}
```

3. Запрос на удаление resource:

```
curl -v -X DELETE
'[]/sberflake/v3/rn/admin/configurer/resources' --header
'Content-Type: application/json' --header 'ott-invoker:
invoker' --header 'SUBJECT: subjectrn' --data-raw '{
"resource_name": "temporaryresource"}'
```

Ответ:

```
{"message":"resource was deleted"}
```

Проверка работоспособности микросервиса Shared

Для проверки работоспособности микросервиса Shared выполните:

1. Запрос на резервирование узла:

```
curl -v -X POST
'[]/sberflake/v3/rn/default/entities/johnSnowNewShared/nodes
' --header 'ott-invoker: ott-client-example' --header
'Content-Type: application/json' --data-raw '{}'
```

Ответ:

```
{"name":"johnSnowNewShared","epoch":"1993-10-
13T03:36:16.391+00:00","nodeBits":12,"sequenceBits":11,"time
stampBits":41,"nodeId":1}
```

2. Запрос на подтверждение использования узла:

```
curl -v -X POST '[]/sberflake/v3/rn/default/heartbeat' --
header 'ott-invoker: ott-client-example' --header 'Content-
Type: application/json' --data-raw '[{ "name":
"johnSnowNewShared", "nodeId": 1 }]'
```

Ответ:

```
[{"name":"johnSnowNewShared","nodeId":1}]
```

Проверка работоспособности микросервиса Normal

Для проверки работоспособности микросервиса Normal выполните:

1. Запрос на создание сущности `snow`:

```
curl -v -X POST '[ROUTE]/sberflake/v1/entities' --header
'ott-invoker: ott-client-example' --header 'Content-Type:
application/json' --data-raw '{"name": "johnSnowNewNormal",
"nodeBits": 10}'
```

Ответ:

```
{"name":"johnSnowNewNormal","timestampBits":41,"epoch":"1993
-10-13T03:36:16.391+00:00","nodeBits":10,"sequenceBits":13}
```

2. Запрос на резервирование узла:

```
curl -v -X POST
'[ROUTE]/sberflake/v1/entities/johnSnowNewNormal/nodes' --
header 'ott-invoker: ott-client-example' --header 'Content-
Type: application/json' --data-raw '{}'
```

Ответ:

```
{"name":"johnSnowNewNormal","timestampBits":41,"epoch":"1993
-10-
13T03:36:16.391+00:00","nodeBits":10,"sequenceBits":13,"node
Id":1}
```

3. Запрос на подтверждение использования узла:

```
curl -v -X POST '[ROUTE]/sberflake/v1/heartbeat' --header
'ott-invoker: ott-client-example' --header 'Content-Type:
application/json' --data-raw '[{ "name":
"johnSnowNewNormal", "nodeId": 1 }]'
```

Ответ:

```
[{"name":"johnSnowNewNormal","nodeId":1}]
```

4. Запрос на удаление узла:

```
curl -v -X DELETE
'[ROUTE]/sberflake/v1/entities/johnSnowNewNormal/nodes/1' --
header 'ott-invoker: ott-client-example'
```

Ответ:

```
{"nodeId":1}
```

5. Запрос на удаление сущности `snow`:

```
curl -v -X DELETE
'[ROUTE]/sberflake/v1/entities/johnSnowNewNormal' --header
'ott-invoker: ott-client-example'
```

Ответ:

```
{"name":"johnSnowNewNormal","timestampBits":41,"epoch":"1993
-10-13T03:36:16.391+00:00","nodeBits":10,"sequenceBits":13}
```

Ответ:

Через UI среды контейнеризации

Проверить успешность установки сервиса можно через UI среды контейнеризации.

Для этого нужно перейти на вкладку **Pods**. Если все Pods (Ingress, Egress, Prometheus (опционально)) находятся в статусе `running`, значит сервис успешно установлен.

Откат

Для отката сервиса Генерация ID на предыдущую версию дистрибутива:

1. Удалите все развернутые в окружении артефакты сервиса, кроме БД (и информации в БД).
2. Выполните установку сервиса согласно разделу «Установка».

Часто встречающиеся проблемы и пути их устранения

В журнале инсталлятора встречается ошибка следующего вида:

```
17:00:41 ✗ Error from server (Invalid):
"${ISTIOCONFIGS_DNSSERVICE_SBERFLAKE_DB_MASTER_NAME}" is
invalid: [metadata.name: Invalid value:
"${ISTIOCONFIGS_DNSSERVICE_SBERFLAKE_DB_MASTER_NAME}":
```

Путь устранения: задать значение переменной `ISTIOCONFIGS_DNSSERVICE_SBERFLAKE_DB_MASTER_NAME` в соответствие с конфигурацией к инсталлятору.

Чек-лист валидации установки

1. Проверить журналы событий (log-файлы) инсталлятора на наличие ошибок.
2. Проверить журналы событий (log-файлы) отсека (pod) сервиса в среде контейнеризации на наличие ошибок.

При успешном выполнении всех пунктов можно приступать к использованию сервиса.

Чек-лист проверки работоспособности интеграции

При осуществлении интеграции (опциональных интеграций) сервиса Генерация ID с платформенными компонентами, проверка работоспособности осуществляется согласно следующим инструкциям:

- компонент Объединенный мониторинг Unimon;
- компонент Аудит;
- компонент Журналирование;
- компонент Прикладной журнал.

Проверка работоспособности компонента Объединенный мониторинг Unimon

Для проверки работоспособности компонента Объединенный мониторинг Unimon:

1. В среде контейнеризации перейдите на вкладку **PODs** и выберите pod (snow-configurer, snow-normal-service, snow-shared-service).
2. Перейдите в терминал и введите команду

```
curl localhost:8081/actuator/prometheus
```

3. Сравните полученное сообщение с отбрасываемыми метриками с событиями мониторинга, приведенными в Руководстве по системному администрированию в разделе «События мониторинга».
4. Перейдите в систему отображения метрик (например, Prometheus), указав необходимую метрику. Например, system_cpu_usage.

Проверка работоспособности компонента Аудит

Для проверки работоспособности компонента Аудит выполните следующие действия:

1. Перейти в АРМ Аудит.
2. Выберите поле **Поиск событий аудита** и на открывшейся странице заполните:

- поля **Период** — необходимым временным периодом (текущая дата);

- поле **Исходный текст** — «snow».

3. Нажмите кнопку **Найти**.
4. Если в результатах поиска присутствуют записи, созданные по времени после запуска сервиса, то проверка выполнена успешно.

Проверка работоспособности компонента Журналирование

Для проверки работоспособности компонента Журналирование выполните следующие действия:

1. Перейти в АРМ Журналирование.
2. Перейти на вкладку **Системный журнал** и на открывшейся странице заполните поля: **Начальная дата**, **Начальное время**, **Конечная дата**, **Конечное время**.
3. Нажмите кнопку **Поиск**.
4. Если в результатах поиска присутствуют записи, созданные по времени после запуска сервиса, то проверка выполнена успешно.

Проверка работоспособности компонента Прикладной журнал

Для проверки работоспособности компонента Прикладной журнал выполните следующие действия:

1. Перейти в АРМ Прикладной журнал.
2. На открывшейся странице **Журнал** выберите в поле **Зона** — «SBERFLAKE», уберите галочку с чек-бокса **С ошибками репликации** и нажмите кнопку **Найти**.
3. Если в результатах поиска присутствуют записи, созданные по времени после запуска сервиса, то проверка выполнена успешно.

Руководство по установке компонента PACMAN (CFGA)

Термины и сокращения

Термин	Определение
Агент компонента «Управление параметрами»	Агент обновления конфигурации приложений-потребителей реализованный в виде sidecar-приложения
Агент конфигурации	Отдельное приложение компонента PACMAN по обновлению конфигурации приложений-потребителей. Устанавливается в Namespace проекта потребителя конфигурации и позволяет обновлять объекты ConfigMap в Namespace проекта в соответствии с настройкой из аннотаций объекта ConfigMap
АРМ компонента «Управление конфигурациями»	Автоматизированное рабочее место администратора компонента «Управление конфигурациями», Rest-микросервис, предоставляющий возможность редактирования конфигурации через консоль Администратора
АРМ компонента «Управление параметрами»	Автоматизированное рабочее место администратора компонента «Управление параметрами», Rest-микросервис, предоставляющий возможность редактирования параметров через консоль Администратора
Артефакт (artifact)	Модель конфигурации компонента «Управление конфигурациями», с которой работает приложение-потребитель
АС	Автоматизированная система
БД	База данных
ИФТ	Стенд интеграционного и функционального тестирования

Термин	Определение
Клиентский Модуль (КМ) компонента «Управление конфигурациями»	Platform-commons – подключаемая библиотека для считывания настроек конфигурации приложения при работе с компонентом «Управление конфигурациями»
Клиентский Модуль (КМ) компонента «Управление параметрами»	Подключаемая библиотека для считывания параметров конфигурации приложения при работе с компонентом «Управление параметрами»
Клиентский Модуль Агента конфигурации	Подключаемая библиотека для обновления конфигурации приложений-потребителей
Ключ конфигурации	Уникальный составной идентификатор, предназначенный для разделения данных в компоненте «Управление параметрами»
Код тенанта/Название проекта	Код группы потребителей в мультитенантной архитектуре
Код тенанта	Код группы потребителей в мультитенантной архитектуре
Комплексный критерий перекрытия (ККП)	Элемент настройки перекрытия версии артефакта конфигурации, представляющий собой совокупность наборов узлов, модулей, групп, описывающих элементный набор конкретной функциональной зоны, в разрезе состава сервисов и инфраструктуры, на которой данные сервисы запускаются
Консоль Администратора	Автоматизированное рабочее место Администратора
Конфигурационный элемент	Параметр или группа параметров артефакта
Конфигурация артефакта	Значения параметров артефакта, соответствующие модели конфигурации
КУ	Клиентский узел на сервере клиентской части

Термин	Определение
ММВ	Стенд для тестирования межмодульного взаимодействия
Модель конфигурации	Описание всех возможных параметров для артефакта определенной версии
НТ	Стенд нагрузочного тестирования
ПО	Программное обеспечение
Приложение-потребитель	Приложение с подключенным КМ, который читает конфигурацию для этого приложения
Приоритетные параметры (ПП)	Это параметры, которые позволяют массово переопределить значение какого-либо параметра у нескольких потребителей, во всех разрезах для конкретного параметра в выбранных ключах конфигураций
ПРОМ	Промышленная среда
ПСИ	Стенд приемо-сдаточных испытаний
Разрез значений	Комбинация атрибутов, которой соответствует значение параметра
Репликация параметров	Механизм синхронизации данных компонента «Управление параметрами» между несколькими БД (репликами)
Семейство БД PostgreSQL	Включает продукты: ванильный PostgreSQL, PostgreSQL SberEdition и Platform V Pangolin SE
Сервис выдачи конфигурации компонента «Управление параметрами»	Rest-микросервис чтения конфигурации, предоставляющий интерфейс, с помощью которого Агент компонента «Управление параметрами» осуществляет чтение полной конфигурации для приложений-потребителей

Термин	Определение
Сервис Импорта компонента «Управление параметрами»	Rest-микросервис групповой вставки параметров, предоставляющий интерфейс, с помощью которого возможно осуществлять групповую вставку параметров и их значений при автоматизированной установке средствами `pipeline`
CT	Стенд разработчиков для развертывания и тестирования своих приложений
СУБД	Система управления базами данных
Тенант/Проект	Запись (порция данных), которая однозначно определяет владельца Ресурсов с целью тарификации использования Ресурсов и возможно, логической изоляции Ресурсов
Тип	Type maven-артефакта
УЦ	Удостоверяющий Центр
API	Application programming interface – программный интерфейс приложения
DEV	Стенд разработчиков для развертывания и тестирования своих приложений
HTTPS	HyperText Transfer Protocol Secure – расширение протокола HTTP для поддержки шифрования в целях повышения безопасности
ID	Идентификатор
Istio	Инструмент обеспечения безопасности, управляемости и мониторинга контейнеров в среде контейнеризации
JDBC (Java Database Connectivity)	Платформенно независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД

Термин	Определение
JKS	Стандартный тип хранилища ключей и сертификатов безопасности в виде файла с расширением jks (java key storage)
JMS	Стандарт промежуточного ПО для рассылки сообщений
JSON	Текстовый формат файла обмена данными, основанный на JavaScript
JVM опция	Параметры запуска java-приложения
JWT	JSON Web Token, это открытый стандарт для создания токенов доступа, основанный на формате JSON. Используется для передачи данных для аутентификации в клиент-серверных приложениях
Kafka	Распределенный программный брокер сообщений
Kubernetes (k8s)	Система с открытым исходным кодом для автоматизации развертывания, масштабирования и управления контейнерными приложениями
Liquibase	Независимая от базы данных библиотека с открытым исходным кодом для отслеживания, управления и применения изменений схемы базы данных
mTLS	Протокол, обеспечивающий взаимную аутентификацию сторон, участвующих в передаче данных по протоколу TLS
OTT	Механизм аутентификации, авторизации и валидации для входящих и для исходящих запросов Приложений на основе токенов авторизации и сертификатов, позволяющих однозначно идентифицировать участников взаимодействия

Термин	Определение
PL	Presentation Layer – презентационный слой приложения
Pod	Объект среды контейнеризации
Rest-сервис компонента «Управление конфигурациями»	Rest-микросервис компонента «Управление конфигурациями» для передачи настроек конфигураций приложению потребителя
Runtime	Режим, позволяющий применить изменения настроек, без перезапуска приложения
SGW	Сетевой шлюз
SLA	Соглашение об уровне обслуживания
SSH	Сетевой протокол прикладного уровня
SSL	Secure Sockets Layer, криптографический протокол для создания безопасной связи передачи данных
TCP	Transmission Control Protocol, протокол управления передачей данных
TLS	Transport layer security – протокол защиты транспортного уровня, криптографический протокол, обеспечивающие защищенную передачу данных между узлами в сети Интернет
UI	User Interface, пользовательский интерфейс
URL	Uniform Resource Locator, унифицированный адрес ресурса
WildFly	Сервер приложений (с открытой лицензией)

Термин	Определение
Xml-файл модели конфигурации	Файл определенного формата, содержащий перечень всех возможных настроек модуля текущей версии, а также ограничения, накладываемые на эти настройки. Имя файла должно быть оформлено по шаблону `{имя_артефакта}-config-struct.xml`

Назначение документа

В данном руководстве приведены инструкции по установке компонента PACMAN. Компонент PACMAN обеспечивает хранение, управление и предоставление по запросу параметров конфигурации библиотек, сервисов Платформы и прикладных приложений, разработанных на Платформе.

PACMAN состоит из компонентов:

- Компонент «Управление конфигурациями». Набор артефактов, выполняющих функцию по управлению конфигурациями приложений-потребителей;
- Компонент «Управление параметрами». Набор артефактов, выполняющих функцию по управлению параметрами приложений-потребителей.

Предназначен для управления общими параметрами Platform V и параметрами подсистем Platform V. Предоставляет администраторам единый универсальный интерфейс для управления настройками, а прикладным проектам — быстрый и надежный интерфейс для получения настроек.

Системные требования

Настройки безопасности окружения и перечень платформенных (дополнительных внешних) продуктов, используемых для установки, настройки и контроля в конечной информационной системе (далее — ИС), выбираются при разработке конечной ИС, исходя из характера обрабатываемой в ней информации и иных требований информационной безопасности (далее — ИБ), предъявляемых к ней.

Системное программное обеспечение

Ниже представлены категории системного программного обеспечения, которые обязательны для установки, настройки, контроля и функционирования компонента. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО «СберТех» (маркировка «Рекомендовано» в столбце Комментарий). Клиенту необходимо выбрать один из продуктов в каждой категории, исходя из условий использования конечной ИС.

Операционная система

Наименование	Версия	Комментарий
ОС «Альт 8 СП»	10.0 и выше	Рекомендовано
Red Hat Enterprise Linux	3.10 и выше	Опционально

Среда контейнеризации

Необходимое окружение для разворачивания компонента — Kubernetes версии 1.0 и выше или OpenShift версии 4.2 и выше (далее — среда контейнеризации).

Наименование	Версия	Комментарий
Kubernetes	1.0 и выше	Рекомендовано
Red Hat OpenShift	4.2 и выше	Опционально

Java-машина

Наименование	Версия	Комментарий
OpenJDK	1.8 и выше	Рекомендовано

Система управления базами данных (СУБД)

Наименование	Версия	Комментарий
PostgreSQL	11.14 и выше	Рекомендовано

Рекомендован Platform V Pangolin SE, см. раздел «Платформенные зависимости».

Браузер

Наименование	Версия	Применение
Яндекс	20 и выше	Рекомендовано
Google Chrome	85 и выше	Опционально

Брокер сообщений (интерфейсный движок)

Наименование	Версия	Применение
Kafka	2.7.0. и выше	Рекомендовано

Рекомендован Platform V Corax, см. раздел «Платформенные зависимости».

Сервер приложений

Наименование	Версия	Применение
WildFly (с открытой		Рекомендовано

лицензией)		
Nginx	1.14.1	Рекомендовано

Сервис интеграции и оркестрации микросервисов в облаке

Наименование	Версия	Комментарий
Istio	1.12 и выше	Рекомендовано

Рекомендован Platform V Synapse Service Mesh, см. раздел «Платформенные зависимости».

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Audit SE	AUD	1.1	AUDT Аудит	опционально	Сервис для аудирования событий
Platform V IAM SE	IAM	1.3	-	опционально	Набор инструментов для управления доступом к информационным ресурсам. Данные инструменты необходимы для аутентификации и авторизации пользователей
Platform V Monitor	OPM	3.27.11	LOGA Журналирование	опционально	Сервис для хранения лог-файлов

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Monitor	OPM	6.0.0-02	MONA Объединенный мониторинг Unimon	опционально	Сервис для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения
Platform V Monitor	OPM		AIOM	опционально	Сервис для обработки данных телеметрии платформы, нахождения взаимосвязей между событиями, выявлению аномального поведения объектов анализа
Platform V DevOps Tools	DOT	1.2	-	опционально	Сервис для развертывания и обновления компонентов Платформы и приложений потребителей, для настройки и обслуживания инфраструктуры Платформы

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Corax	KFK	5.1 и выше	KFKA Kafka Sber Edition	опционально	Программный брокер сообщений, представляющий собой распределенную, отказоустойчивую, реплицированную и легко масштабируемую систему передачи сообщений, рассчитанную на высокую пропускную способность
Platform V Synapse Service Mesh	SSM	1.12 и выше	POLM Управление политиками	Опционально	Панель управления, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в среде контейнеризации Kubernetes

Примечание:

- **Обязательно** — компонент или продукт необходим для функционирования сервиса.
- **Опционально** — необязательный для функционирования сервиса компонент или продукт, рекомендуется его установка, но допускается использование аналога других производителей.

Аппаратные требования

Для установки компонента/продукта требуется следующая конфигурация аппаратного обеспечения:

№	Среда	Состав компонент	Количество pods	Используемая процессорная мощность, Ядра	Память, ГБ
1	DEV, MMB	configurator-server	1	2	4
		configurator-rest	1	2	4
		uko_ufsparams_import_v2	1	2	4
		uko_ufsparams_bh_v2	1	2	4
		uko_ufsparams_ms_v2	1	2	4
		pacman-agent	1	2	4
		pacman-ui	1	2	4
		ingress	1	2	4
		egress	1	2	4
2	ИФТ, СТ, ПСИ	configurator-server	2	4	8
		configurator-rest	2	4	8
		uko_ufsparams_import_v2	2	4	8
		uko_ufsparams_bh_v2	2	4	8
		uko_ufsparams_ms_v2	2	4	8
		pacman-agent	2	4	8
		pacman-ui	2	4	8
		ingress	1	2	4
		egress	1	2	4
3	ПРОМ, НТ	configurator-server	2	4	8
		configurator-rest	2	4	8

№	Среда	Состав компонент	Количество pods	Используемая процессорная мощность, Ядра	Память, ГБ
		uko_ufsparams_import_v2	2	4	8
		uko_ufsparams_bh_v2	2	4	8
		uko_ufsparams_ms_v2	2	4	8
		pacman-agent	2	4	8
		pacman-ui	2	4	8
		ingress	2	4	8
		egress	2	4	8

БД компонента «Управление конфигурациями»:

№	Среда	Используемая процессорная мощность, ядра	Память, Гб	Используемый объем жесткого диска, Гб
1	DEV, ММВ	4	8	300
2	ИФТ, СТ, ПСИ	8	16	300
3	ПРОМ, НТ	8	32	300

БД компонента «Управление параметрами»:

№	Среда	Используемая процессорная мощность	Память, Гб	Используемый объем жесткого диска, Гб
1	DEV, ММВ	4	8	500
2	ИФТ, СТ, ПСИ	8	16	500
3	ПРОМ, НТ	8	32	500

Выбор способа установки

В зависимости от способа развертывания сервиса воспользуйтесь одной из инструкций:

- целевая ручная установка сервиса;
- опциональная автоматизированная установка сервиса с использованием программного продукта Platform V DevOps Tools.

Для установки, настройки и успешной эксплуатации компонента PACMAN требуется соблюдение следующих условий:

- Развернута и настроена среда контейнеризации Kubernetes или Red Hat OpenShift 4+.
- В кластере создан проект (namespace), в котором будет развернута контрольная панель.
- В проекте создана учетная запись с правами на загрузку артефактов.
- В проекте имеются свободные ресурсы по лимитам и запросам не менее, чем зарезервировано в конфигурационных артефактах.

Состав дистрибутива

Компонент дистрибутива	Описание
`./conf/`	Каталог с конфигурационными файлами для автоматизированной установки ПО средствами `pipeline`
`./db/`	Архив со скриптами `liquibase` для развертывания необходимых объектов в БД
`./bh/configurator-server-spring-boot.jar`	Сервис API для взаимодействия с PL (APM компонента «Управление конфигурациями»)
`./bh/configurator-rest.jar`	Rest-микросервис, к которому обращаются Потребители с запросом их конфигурации
`./bh/ufs.sup.manager.jar`	Сервис API для взаимодействия с PL (APM компонента «Управление параметрами»)
`./bh/ufs.sup.service.jar`	Rest-микросервис чтения конфигурации
`./bh/ufs.sup.import.jar`	Rest-микросервис групповой вставки параметров
`./pl/ufs-params-manager-pl.zip`	Архив, содержащий файлы Presentation Layer (workplace) приложения
`./pl/ufs-params-manager-pl-standalone.zip`	Архив, содержащий файлы Presentation Layer (standalone) приложения
`./pl/configurator-ui-static.zip`	Архив, содержащий файлы Presentation Layer приложения

Установка

Выбор способа установки

В зависимости от способа развертывания сервиса воспользуйтесь одной из инструкций:

- целевая ручная установка сервиса;
- опциональная автоматизированная установка сервиса.

Раздельная установка модулей компонента

Компонент PACMAN представляет собой набор модулей, каждый из которых выполняет определенную функцию.

В процессе установки компонента можно установить как все модули, входящие в его состав, так и только часть модулей.

Набор модулей, подлежащих установке, должен быть задан перед установкой комопнента, для этого необходимо:

1. В общих настройках стенда, куда должен установиться компонент PACMAN, в конфигурационном файле `subsystems.json` для компонента добавить блок `app_name`, где перечислить модули, которые необходимо установить.

   Можно подключить следующие модули:

Модуль	Назначение
`configurator_rest`	Rest-микросервис компонента «Управление конфигурациями», к которому обращаются Потребители с запросом их конфигурации
`configurator_server`	Сервис API компонента «Управление конфигурациями» для взаимодействия с PL (АРМ компонента «Управление конфигурациями»)
`pacman_agent`	Общий сервис доставки конфигурации до Потребителя
`pacman_ui`	Модуль, содержащий файлы Presentation Layer приложения
`uko_ufsparams_bh_v2`	Сервис API компонента «Управление параметрами» для взаимодействия с PL (АРМ компонента «Управление параметрами»)

Модуль	Назначение
uko_ufsparams_import_v2	Rest-микросервис компонента «Управление параметрами» импорта параметров
uko_ufsparams_ms_v2	Rest-микросервис компонента «Управление параметрами» чтения конфигурации

Пример:

```
"app_name": [   "configurator_rest",
"configurator_server",   "uko_ufsparams_bh_v2",
"uko_ufsparams_import_v2",   "uko_ufsparams_ms_v2",
"pacman_agent",   "pacman_ui" ]
```

Для текущей версии PACMAN устанавливаются только модули компонента «Управление конфигурациями», в файле необходимо указать следующие модули:

```
"app_name": [   "configurator_server",   "pacman_agent",
"pacman_ui" ]
```

2. Выполнить установку компонента PACMAN.

Установка дистрибутива без использования Pipeline DevOps

Миграция конфигурационных файлов

Перед развертыванием самого приложения необходимо произвести шаблонизацию и миграцию параметров из дистрибутива на среду (параметров среды и параметров программного обеспечения).

Данные для шаблонизации и миграции берутся из папки дистрибутива conf/config/parameters.

Параметры в конфигурационных файлах дистрибутива могут содержать плейсхолдеры на общие параметры установки, конкретные значения которых должны подставляться при установке. Конкретные значения параметров установки хранятся отдельно в файлах стендов, наличие этих данных обеспечивается администраторами стендов, куда производится установка.

```
├── conf │   └── config │       ├── parameters
│       │   └── ufsparams_config.all.conf │       │   ├──
ufsparams_config.istio.all.conf │       │   ├──
ufsparams_config_bh.conf │       │   ├──
```

```
ufsparams_config_configurator.conf |       |   └──
ufsparams_config_configurator_rest.conf |      |   ├──
ufsparams_config_import.conf |       |   ├──
ufsparams_config_ms.conf |       |   └──
ufsparams_config_pacman_agent.conf |       |   └──
ufsparams_config_pacman_ui.conf
```

ufsparams_config_pacman_agent.conf – конфигурационный файл параметров для настройки Агента конфигурации.

```
ufsparams_config_pacman_agent.httpclient.circuitBreaker.enab
le=false
ufsparams_config_pacman_agent.httpclient.circuitBreaker.fail
ureRate.threshold=50
ufsparams_config_pacman_agent.httpclient.circuitBreaker.open
State.duration=60000
ufsparams_config_pacman_agent.httpclient.circuitBreaker.half
Open.requestBuffer=0
ufsparams_config_pacman_agent.httpclient.circuitBreaker.mini
mumCalls.size=10  # Параметры для Kubernetes/OSE контейнеров
# Параметры для автоскейлинга
ufsparams_config_pacman_agent.openshift.autoscaling.min_repl
icas=1  # Размеры лимитов для POD
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.resources.limit
s.cpu=1000m
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.resources.limit
s.memory=1000Mi  # Размеры запросов для POD
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.resources.reque
sts.cpu=500m
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.resources.reque
sts.memory=1000Mi  # Настройки Liveness Probe
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.livenessProbe.i
nitialDelaySeconds=30
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.livenessProbe.p
eriodSeconds=5
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.livenessProbe.t
imeoutSeconds=1
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.livenessProbe.s
uccessThreshold=1
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
```

```
pec.containers.ufsparams_config_pacman_agent.livenessProbe.f
ailureThreshold=3  # Настройки Readiness Probe
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.readinessProbe.
initialDelaySeconds=30
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.readinessProbe.
periodSeconds=1
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.readinessProbe.
timeoutSeconds=1
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.readinessProbe.
successThreshold=1
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.containers.ufsparams_config_pacman_agent.readinessProbe.
failureThreshold=3  # Параметры Агента
ufsparams_config_pacman_agent.spring.active.profiles=PROM
#UNIMON
ufsparams_config_pacman_agent.prometheus.io.scrape='true'
ufsparams_config_pacman_agent.prometheus.io.path='/actuator/
prometheus'
ufsparams_config_pacman_agent.management.server.port=8081
#JAVA опции
ufsparams_config_pacman_agent_ose_configmap_javaArguments=-
Dmodule.id=pacman-agent  #Config for AGENT
ufsparams_config_pacman_agent.configuration.selector='config
urator-yaml'
ufsparams_config_pacman_agent.configuration.managed='true'
ufsparams_config_pacman_agent.configuration.reloader.match='
true'
ufsparams_config_pacman_agent.configuration.reloader.search=
'true'
ufsparams_config_pacman_agent.config_storage.baseurl=${globa
l.pacman_agent.config_storage.baseUrl|https://${global.platf
orm.ingress.route.https|ufsparams-config-
${distrib.release.version}}}
ufsparams_config_pacman_agent.config_storage.http.timeout=10
000 ufsparams_config_pacman_agent.delay=10000
ufsparams_config_pacman_agent.enabled=true  #UFS
ufsparams_config_pacman_agent.platform.environment.product.s
ubsystem=PACMAN_AGENT
ufsparams_config_pacman_agent.platform.environment.product.a
pi-version=1.3
ufsparams_config_pacman_agent.platform.environment.product.d
eploymentUnit=pacman-agent
ufsparams_config_pacman_agent.platform.config.deployment-
unit=pacman-agent
ufsparams_config_pacman_agent.ufs.monitoring.client.subsyste
m=PACMAN_AGENT
```

```
ufsparams_config_pacman_agent.ufs.monitoring.client.title=pa
cman-agent  #ISTIO # Размеры запросов для Istio-sidecar
ufsparams_config_pacman_agent.ose.deployment.metadata.annota
tions.sidecar.istio.io.proxyCPU=200m
ufsparams_config_pacman_agent.ose.deployment.metadata.annota
tions.sidecar.istio.io.proxyMemory=256Mi  # Размеры лимитов
для Istio-sidecar
ufsparams_config_pacman_agent.ose.deployment.metadata.annota
tions.sidecar.istio.io.proxyCPULimit=300m
ufsparams_config_pacman_agent.ose.deployment.metadata.annota
tions.sidecar.istio.io.proxyMemoryLimit=256Mi  # Настройки
podAntiAffinity
ufsparams_config_pacman_agent.ose.feature.SelfAntiAffinity.e
nabled=${global.feature.SelfAntiAffinity.enabled|true}  #
Настройки Graceful Shutdown
ufsparams_config_pacman_agent.ose.deployment.spec.template.s
pec.terminationGracePeriodSeconds=${global.ufs.terminationGr
acePeriodSeconds|30}  # Настройки PodDisruptionBudget
ufsparams_config_pacman_agent.ose.poddisruptionbudget.spec.m
inAvailable=${global.poddisruptionbudget.minAvailable|50%}
# Флаг включения/выключения reloader в pacman-agent
ufsparams_config_pacman_agent.ose.feature.reloader.enabled=f
alse
```

ufsparams_config.all.conf – общие параметры установки контейнеров в Pod.

```
# Параметры удаленного компонента Журналирования
(пробрасываются не через файл параметров, которые
импортируются в компонент «Управление параметрами») ufs-
logger.kafka.topic=${global.platform.logger.kafka.topic|}
ufs-
logger.kafka.bootstrap.servers=${global.platform.logger.kafk
a.bootstrap.servers|} ufs-
logger.kafka.ssl.keystore.location=/etc/config/ssl/mq.jks
ufs-
logger.kafka.ssl.truststore.location=/etc/config/ssl/mq.jks
ufs-
logger.kafka.security.protocol=${global.platform.logger.kafk
a.security.protocol|} ufs.tenant.code={{ TENANT_CODE }} ufs-
logger.parameters.url=${global.ose.platform.ufs.baseurl.logg
er|}/ufs-logger-parameters-ng/parameters
platform.logger.http.host=${global.platform.sgw.route.log.ho
st|}
platform.logger.http.port=${ufsparams_config.ose.istio.egres
s.common.internal-port}  # Kafka Audit ufs-audit-
unsafe.kafka.bootstrap.servers=${global.platform.audit.kafka
.bootstrap.servers|} ufs-audit-
unsafe.kafka.ssl.endpoint.identification.algorithm=${global.
platform.kafka.ssl.endpoint.identification.algorithm|} ufs-
```

```
audit-
unsafe.kafka.security.protocol=${global.platform.kafka.secur
ity.protocol|}  ufs-audit-
unsafe.kafka.producer_factory.delivery.timeout.ms=${global.p
latform.kafka.producer.delivery.timeout.ms|} ufs-audit-
unsafe.kafka.producer_factory.batch.size=${global.platform.k
afka.producer.batch.size|} ufs-audit-
unsafe.kafka.producer_factory.linger.ms=${global.platform.ka
fka.producer.linger.ms|} ufs-audit-
unsafe.kafka.producer_factory.buffer.memory=${global.platfor
m.kafka.producer.buffer.memory|} ufs-audit-
unsafe.kafka.producer_factory.request.timeout.ms=${global.pl
atform.kafka.producer.request.timeout.ms|} ufs-audit-
unsafe.kafka.producer_factory.send.buffer.bytes=${global.pla
tform.kafka.producer.send.buffer.bytes|} ufs-audit-
unsafe.kafka.producer_factory.receive.buffer.bytes=${global.
platform.kafka.producer.receive.buffer.bytes|} ufs-audit-
unsafe.kafka.producer_factory.max.request.size=${global.plat
form.kafka.producer.max.request.size|} ufs-audit-
unsafe.kafka.producer_factory.metadata.max.age.ms=${global.p
latform.kafka.producer.metadata.max.age.ms|} ufs-audit-
unsafe.kafka.producer_factory.connections.max.idle.ms=${glob
al.platform.kafka.producer.connections.max.idle.ms|} ufs-
audit-
unsafe.kafka.producer_factory.max.block.ms=${global.platform
.kafka.producer.max.block.ms|} ufs-audit-
unsafe.kafka.producer_factory.compression.type=${global.plat
form.kafka.producer.compression.type|} ufs-audit-
unsafe.kafka.ssl.enabled.protocols=${global.platform.kafka.s
sl.enabled.protocols|}  # Лимиты (fluent-bit sidecar)
ufsparams_config.ose.deployment.spec.template.spec.container
s.fluent-bit.sidecar.resources.limits.cpu=300m
ufsparams_config.ose.deployment.spec.template.spec.container
s.fluent-bit.sidecar.resources.limits.memory=200Mi
ufsparams_config.ose.deployment.spec.template.spec.container
s.fluent-bit.sidecar.resources.requests.cpu=300m
ufsparams_config.ose.deployment.spec.template.spec.container
s.fluent-bit.sidecar.resources.requests.memory=200Mi  #
Лимиты (reloader sidecar)
ufsparams_config.ose.deployment.spec.template.spec.container
s.reloader.sidecar.resources.limits.cpu=100m
ufsparams_config.ose.deployment.spec.template.spec.container
s.reloader.sidecar.resources.limits.memory=100Mi
ufsparams_config.ose.deployment.spec.template.spec.container
s.reloader.sidecar.resources.requests.cpu=100m
ufsparams_config.ose.deployment.spec.template.spec.container
s.reloader.sidecar.resources.requests.memory=100Mi  #
Настройки окружения
ufsparams_config.pacman.adapters.monitoring.unimon=true
ufsparams_config.pacman.adapters.monitoring.stub=false
```

```
ufsparams_config.pacman.adapters.audit.selector=pprb-2.0
ufsparams_config.pacman.adapters.audit.timeout=5000
ufsparams_config.pacman.adapters.authorization.pprb.enable=f
alse
ufsparams_config.pacman.adapters.authorization.pprb.jwt.enab
le=false
ufsparams_config.pacman.adapters.authorization.pprb.sudir.en
able=false
ufsparams_config.pacman.adapters.authorization.pprb.basic.en
able=false
ufsparams_config.pacman.adapters.authorization.ufs.enable=tr
ue
ufsparams_config.pacman.adapters.authorization.ufs.sudir.ena
ble=true
ufsparams_config.pacman.adapters.authorization.ufs.iam.enabl
e=false
ufsparams_config.pacman.adapters.authorization.ufs.iam.jws_t
oken_header=Authorization
ufsparams_config.pacman.adapters.authorization.ufs.iam.roles
_path=realm_access.roles
ufsparams_config.pacman.adapters.authorization.ufs.iam.publi
ckey_locations=
ufsparams_config.pacman.adapters.authorization.ufs.iam.audie
nce=PlatformUFS
ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl_r
equired=false
ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl_t
ruststore=
ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl_t
ruststore_password=
ufsparams_config.pacman.adapters.authorization.ufs.iam.pools
ize=20
ufsparams_config.pacman.adapters.authorization.ufs.iam.cache
_updatetime=3600
ufsparams_config.pacman.adapters.authorization.ufs.iam.devmo
de=false  # Registry
ufsparams_config.custom.ose.deployment.spec.template.spec.co
ntainers.image.registry=base.sw.sbc.space
ufsparams_config.platform.environment.product.channel=SUPPOR
T ufsparams_config.platform.environment.product.release=1.4
ufsparams_config.ufs.monitoring.client.channel=SUPPORT
ufsparams_config.iam.public.key.filename=${global.iam.public
.key.filename|}  # Авторизация
ufsparams_config.authorization.ufs.baseUrl=${global.ose.plat
form.ufs.baseurl.security|}
```

ufsparams_config.istio.all.conf– общий конфигурационный файл для настройки Istio.

```
# Параметры Istio ufsparams_config.ose.istio.control-plane-
project=${global.ufs.synapse.controlPlane.project}
ufsparams_config.ose.istio.envoy-
image=${global.ufs.synapse.envoy.image|registry.redhat.io/op
enshift-service-mesh/proxyv2-
rhel8@sha256:51d82b560e467ec59a3b6625b04c31b86df5b4c10070a35
1d547cb6cf3f26591}  # Egress
ufsparams_config.ose.istio.egress.common.internal-
port=${global.ose.platform.egress.http.port|}
ufsparams_config.ose.istio.egress.common.nlb_mm-
host=${global.default.nlb_mm.host|}
ufsparams_config.ose.istio.egress.common.nlb_mm-
port=${global.default.nlb_mm.port|}
ufsparams_config.ose.istio.egress.se.tcp.kafkaPort.number=${
global.platform.ose.kafka.ports|}
ufsparams_config.ose.istio.egress.se.tcp.postgreDbPort.numbe
r={{ EFS_PSTGR_DB_PORT }}  # Audit2
ufsparams_config.ose.istio.egress.common.audit2-
host=${global.platform.audit2.host|}
ufsparams_config.ose.istio.egress.common.audit2-
port=${global.platform.audit2.port|}
ufsparams_config.ose.istio.egress.common.audit2-
protocol=${global.platform.audit2.protocol|}  # SPAS
ufsparams_config.ose.istio.egress.common.spas-
host=${global.platform.spas.host|}
ufsparams_config.ose.istio.egress.common.spas-
port=${global.platform.spas.port|}
ufsparams_config.ose.istio.egress.common.spas-
protocol=${global.platform.spas.protocol|}  # CONFIG_STORAGE
ufsparams_config.ose.istio.egress.common.config-storage-
host=${global.platform.config-
storage.host|${ufsparams_config.ose.istio.ingress.route.spec
.host.https.appFQDN}}
ufsparams_config.ose.istio.egress.common.config-storage-
port=${global.platform.config-storage.port|443}
ufsparams_config.ose.istio.egress.common.config-storage-
protocol=${global.platform.config-storage.protocol|HTTPS}  #
Kafka
ufsparams_config.ose.istio.egress.se.tcp.kafkaPort.number=${
global.platform.ose.kafka.ports|}  # Размеры лимитов для
egress
ufsparams_config.ose.istio.egress.deployment.spec.template.s
pec.containers.resources.limits.cpu=700m
ufsparams_config.ose.istio.egress.deployment.spec.template.s
pec.containers.resources.limits.memory=1Gi  # Размеры
запросов для egress
ufsparams_config.ose.istio.egress.deployment.spec.template.s
pec.containers.resources.requests.cpu=700m
ufsparams_config.ose.istio.egress.deployment.spec.template.s
```

```
pec.containers.resources.requests.memory=512Mi
ufsparams_config.ose.istio.egress.deployment.spec.replicas=1
ufsparams_config.ose.istio.egress.deployment.spec.strategy.r
ollingUpdate.maxUnavailable=${global.ufs.strategy.maxUnavail
able|}
ufsparams_config.ose.istio.egress.deployment.spec.strategy.r
ollingUpdate.maxSurge=${global.ufs.strategy.maxSurge|}  #
Размеры лимитов для ingress
ufsparams_config.ose.istio.ingress.deployment.spec.template.
spec.containers.resources.limits.cpu=700m
ufsparams_config.ose.istio.ingress.deployment.spec.template.
spec.containers.resources.limits.memory=1Gi  # Размеры
запросов для ingress
ufsparams_config.ose.istio.ingress.deployment.spec.template.
spec.containers.resources.requests.cpu=700m
ufsparams_config.ose.istio.ingress.deployment.spec.template.
spec.containers.resources.requests.memory=512Mi
ufsparams_config.ose.istio.ingress.deployment.spec.replicas=
1
ufsparams_config.ose.istio.ingress.deployment.spec.strategy.
rollingUpdate.maxUnavailable=${global.ufs.strategy.maxUnavai
lable|}
ufsparams_config.ose.istio.ingress.deployment.spec.strategy.
rollingUpdate.maxSurge=${global.ufs.strategy.maxSurge|}
ufsparams_config.ose.istio.ingress.common.internal-
port=11443  # Routes
ufsparams_config.ose.istio.ingress.route.spec.host.http.appF
QDN=${global.platform.ingress.route.http|ufsparams-config-
${distrib.release.version}-
dev.pacman.${global.multiClusters.openshiftAppsDomain}}
ufsparams_config.ose.istio.ingress.route.spec.host.https.app
FQDN=${global.platform.ingress.route.https|ufsparams-config-
mtls-${distrib.release.version}-
dev.pacman.${global.multiClusters.openshiftAppsDomain}}
ufsparams_config.ose.istio.ingress.route.metadata.annotation
s.io.timeout=180000ms
ufsparams_config.ose.istio.ingress.route.namespace=${global.
multiClusters.openshiftProjectName}  # Gateway
ufsparams_config.ose.istio.ingress.gateway.http=${global.pla
tform.ingress.gateway.http|cfga-gw-ingress-
${distrib.release.version}}
ufsparams_config.ose.istio.ingress.gateway.https=${global.pl
atform.ingress.gateway.https|cfga-gw-mtls-ingress-
${distrib.release.version}}   # OTT parameters (ott-sidecar-
ufs:4.1.18)
ott.ose.istio.deployment.spec.template.spec.containers.ott-
sidecar.image=${global.ott.image.sidecar.version|registry.ca
.sbrf.ru/sigma/pprb/ci00641491/ci01125613_ott/ott-sidecar-
ufs@sha256:49fbbb1c00dd00c72f48ef0518d62a3f18fe21abe223b532b
7398d9254c4b2ee}
```

```
ufsparams_config.ose.istio.egress.database.addresses=${globa
l.ose.istio.egress.database.addresses|}
ufsparams_config.ose.istio.egress.database.endpoints=${globa
l.ose.istio.egress.database.endpoints|}
ufsparams_config.ose.istio.egress.database.host=${global.ose
.istio.egress.database.host|}
ufsparams_config.ose.istio.egress.database.internal-
port=${global.ose.istio.egress.database.internal-port|}
ufsparams_config.ose.istio.egress.database.port=${global.ose
.istio.egress.database.port|}
ufsparams_config.ose.istio.egress.database.protocol=${global
.ose.istio.egress.database.protocol|}  # OTT Egress
(ConfigMap)
ufsparams_config.ose.egress.ott.service.url=${global.ott.ser
vice.url|}
ufsparams_config.ose.egress.ott.service.cert.alias=ott-
service
ufsparams_config.ose.egress.ott.grpc.port=${global.ott.grpc.
port|}
ufsparams_config.ose.egress.ott.certstore.path=/mnt/secrets/
ott.jks
ufsparams_config.ose.egress.ott.client.cert.alias=ci02228938
_efs_ufsparams_config_{{ SEGMENT_ID | lower }}
ufsparams_config.ose.egress.ott.trustStorePath=/mnt/secrets/
ott.jks
ufsparams_config.ose.egress.ott.module.id=ci02228938_efs_ufs
params_config_{{ SEGMENT_ID | lower }}
ufsparams_config.ose.egress.ott.service.hosts=${global.ott.s
ervice.hosts|}  # OTT Ingress (ConfigMap)
ufsparams_config.ose.ingress.ott.service.url=${global.ott.se
rvice.url|}
ufsparams_config.ose.ingress.ott.service.cert.alias=ott-
service
ufsparams_config.ose.ingress.ott.grpc.port=${global.ott.grpc
.port|}
ufsparams_config.ose.ingress.ott.certstore.path=/mnt/secrets
/ott.jks
ufsparams_config.ose.ingress.ott.client.cert.alias=ci0222893
8_efs_ufsparams_config_{{ SEGMENT_ID | lower }}
ufsparams_config.ose.ingress.ott.trustStorePath=/mnt/secrets
/ott.jks
ufsparams_config.ose.ingress.ott.module.id=ci02228938_efs_uf
sparams_config_{{ SEGMENT_ID | lower }}
ufsparams_config.ose.ingress.ott.service.hosts=${global.ott.
service.hosts|}
ufsparams_config.ssl.ose.istio.keyStore.egress.KeyStoreFromF
ile=${ssl.ose.istio.keyStore.egress.KeyStoreFromFile}  #
Выбор типа хранилища для OTT
ufsparams_config.ott.store.type=JKS  # Container params
ufsparams_config.ose.istio.egress.deployment.spec.template.s
```

```
pec.containers.ott.sidecar.resources.limits.cpu=400m
ufsparams_config.ose.istio.egress.deployment.spec.template.s
pec.containers.ott.sidecar.resources.limits.memory=500Mi
ufsparams_config.ose.istio.egress.deployment.spec.template.s
pec.containers.ott.sidecar.resources.requests.cpu=300m
ufsparams_config.ose.istio.egress.deployment.spec.template.s
pec.containers.ott.sidecar.resources.requests.memory=500Mi
ufsparams_config.ose.istio.ingress.deployment.spec.template.
spec.containers.ott.sidecar.resources.limits.cpu=400m
ufsparams_config.ose.istio.ingress.deployment.spec.template.
spec.containers.ott.sidecar.resources.limits.memory=500Mi
ufsparams_config.ose.istio.ingress.deployment.spec.template.
spec.containers.ott.sidecar.resources.requests.cpu=300m
ufsparams_config.ose.istio.ingress.deployment.spec.template.
spec.containers.ott.sidecar.resources.requests.memory=500Mi
ufsparams_config.ose.istio.egress.api-
ose.addresses=${global.ose.istio.egress.api-
ose.addresses|172.30.0.1/32}
ufsparams_config.ose.istio.egress.api-
ose.endpoints=${global.ose.istio.egress.api-
ose.endpoints|172.30.0.1}
ufsparams_config.ose.istio.egress.api-
ose.host=${global.ose.istio.egress.api-ose.host|api-
ose.domain} ufsparams_config.ose.istio.egress.api-
ose.port=${global.ose.istio.egress.api-ose.port|443}
ufsparams_config.ose.istio.egress.api-
ose.protocol=${global.ose.istio.egress.api-ose.protocol|TCP}
ufsparams_config.ose.istio.egress.kafka.bootservers=${global
.ose.istio.egress.kafka.bootservers|}  #istio
ufsparams_config.ssl.ose.istio.keyStore.ingress.CertAlias=${
ssl.ose.istio.keyStore.ingress.CertAlias}
ufsparams_config.ssl.ose.istio.keyStore.ingress.KeyStoreFrom
File=${ssl.ose.istio.keyStore.ingress.KeyStoreFromFile}
ufsparams_config.ssl.ose.istio.keyStore.egress.CertAlias=${s
sl.ose.istio.keyStore.egress.CertAlias}
ufsparams_config.ssl.ose.istio.keyStore.egress.KeyStoreFromF
ile=${ssl.ose.istio.keyStore.egress.KeyStoreFromFile}
ufsparams_config.ssl.ose.istio.keyStore.RootCertAlias=${ssl.
ose.istio.keyStore.RootCertAlias}
ufsparams_config.ssl.ose.istio.keyStore.ingress.password=${s
sl.ose.istio.keyStore.ingress.password}
ufsparams_config.ssl.ose.istio.keyStore.egress.password=${ss
l.ose.istio.keyStore.egress.password}
ufsparams_config.ose.istio.istiod.name=istiod-basic
ufsparams_config.ssl.ose.keyStore.mq.keyStoreFromFile=${ssl.
ose.keyStore.mq.keyStoreFromFile}
ufsparams_config.ose.ott.authz.realm=${global.ose.ott.authz.
realm|ott}  # Настройки podAntiAffinity
ufsparams_config.ose.istio.egress.feature.SelfAntiAffinity.e
nabled=${global.feature.SelfAntiAffinity.enabled|true}
```

```
ufsparams_config.ose.istio.ingress.feature.SelfAntiAffinity.
enabled=${global.feature.SelfAntiAffinity.enabled|true}  #
Настройки Graceful Shutdown
ufsparams_config.ose.istio.egress.deployment.spec.template.s
pec.terminationGracePeriodSeconds=${global.ufs.terminationGr
acePeriodSeconds|30}
ufsparams_config.ose.istio.ingress.deployment.spec.template.
spec.terminationGracePeriodSeconds=${global.ufs.terminationG
racePeriodSeconds|30}  # Настройки PodDisruptionBudget
ufsparams_config.ose.istio.egress.poddisruptionbudget.spec.m
inAvailable=${global.poddisruptionbudget.minAvailable|50%}
ufsparams_config.ose.istio.ingress.poddisruptionbudget.spec.
minAvailable=${global.poddisruptionbudget.minAvailable|50%}
```

ufsparams_config_pacman_ui.conf – конфигурационный файл параметров для настройки UI.

```
# Параметры для контейнеров среды контейнеризации #
Параметры для автоскейлинга
ufsparams_config_pacman_ui.openshift.autoscaling.min_replica
s=1  # Размеры лимитов для пода
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.resources.limits.cpu=
1000m
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.resources.limits.memo
ry=1000Mi  # Размеры запросов для пода
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.resources.requests.cp
u=500m
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.resources.requests.me
mory=1000Mi  # Настройки Liveness Probe
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.livenessProbe.initial
DelaySeconds=30
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.livenessProbe.periodS
econds=5
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.livenessProbe.timeout
Seconds=1
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.livenessProbe.success
Threshold=1
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.livenessProbe.failure
Threshold=3  # Настройки Readiness Probe
```

```
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.readinessProbe.initia
lDelaySeconds=15
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.readinessProbe.period
Seconds=1
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.readinessProbe.timeou
tSeconds=1
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.readinessProbe.succes
sThreshold=1
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.containers.ufsparams_config_pacman_ui.readinessProbe.failur
eThreshold=3  #UNIMON
ufsparams_config_pacman_ui.prometheus.io.scrape='true'
ufsparams_config_pacman_ui.prometheus.io.path='/actuator/pro
metheus'
ufsparams_config_pacman_ui.management.server.port=8081
#ISTIO # Размеры запросов для Istio-sidecar
ufsparams_config_pacman_ui.ose.deployment.metadata.annotatio
ns.sidecar.istio.io.proxyCPU=200m
ufsparams_config_pacman_ui.ose.deployment.metadata.annotatio
ns.sidecar.istio.io.proxyMemory=256Mi  # Размеры лимитов для
Istio-sidecar
ufsparams_config_pacman_ui.ose.deployment.metadata.annotatio
ns.sidecar.istio.io.proxyCPULimit=300m
ufsparams_config_pacman_ui.ose.deployment.metadata.annotatio
ns.sidecar.istio.io.proxyMemoryLimit=256Mi  # Настройки
podAntiAffinity
ufsparams_config_pacman_ui.ose.feature.SelfAntiAffinity.enab
led=${global.feature.SelfAntiAffinity.enabled|true}  #
Настройки Graceful Shutdown
ufsparams_config_pacman_ui.ose.deployment.spec.template.spec
.terminationGracePeriodSeconds=${global.ufs.terminationGrace
PeriodSeconds|30}  # Настройки PodDisruptionBudget
ufsparams_config_pacman_ui.ose.poddisruptionbudget.spec.minA
vailable=${global.poddisruptionbudget.minAvailable|50%}
```

ufsparams_config_configurator.conf – конфигурационный файл параметров для настройки компонента «Управление конфигурациями».

```
# Параметры для контейнеров среды контейнеризации #
Параметры для автоскейлинга
configurator.openshift.autoscaling.min_replicas=1  # Размеры
лимитов для pod
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.resources.limits.cpu=1000m
```

```
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.resources.limits.memory=1000Mi  # Размеры
запросов для pod
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.resources.requests.cpu=500m
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.resources.requests.memory=1000Mi  # Настройки
Liveness Probe
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.livenessProbe.initialDelaySeconds=30
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.livenessProbe.periodSeconds=5
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.livenessProbe.timeoutSeconds=1
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.livenessProbe.successThreshold=1
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.livenessProbe.failureThreshold=3  # Настройки
Readiness Probe
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.readinessProbe.initialDelaySeconds=30
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.readinessProbe.periodSeconds=1
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.readinessProbe.timeoutSeconds=5
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.readinessProbe.successThreshold=1
configurator.ose.deployment.spec.template.spec.containers.co
nfigurator.readinessProbe.failureThreshold=3  # Параметры
Управление конфигурациями
configurator.spring.active.profiles=PROM
ufsparams_config.configurator.core.rest-enabled=false
ufsparams_config.configurator.core.db.url=${jdbc.efs.pstgr.u
rl} configurator.ufs.security.client.mode=PROM
ufsparams_config.configurator.core.kafka.bootstrap.servers=$
{global.platform.kafka.bootstrap.servers}
ufsparams_config.configurator.core.kafka.security.protocol=$
{global.platform.kafka.security.protocol|PLAINTEXT}
ufsparams_config.configurator.core.kafka.configurator.client
.topic-
name=${global.platform.kafka.configurator.client.topic-
name|}
ufsparams_config.configurator.core.kafka.configurator.ssl-
keystore-location=${global.platform.kafka.configurator.ssl-
keystore-location|/etc/config/ssl/mq.jks}
ufsparams_config.configurator.core.kafka.configurator.ssl-
truststore-
location=${global.platform.kafka.configurator.ssl-
truststore-location|/etc/config/ssl/mq.jks}
ufsparams_config.configurator.core.kafka.configurator.client
```

```
.ssl-enabled-
protocols=${global.platform.kafka.ssl.enabled.protocols|TLSv
1.2}
ufsparams_config.configurator.core.kafka.configurator.client
.ssl-
protocol=${global.platform.kafka.ssl.enabled.protocols|TLSv1
.2}
ufsparams_config.configurator.core.kafka.configurator.client
.ssl-keystore-
type=${global.platform.kafka.configurator.client.ssl-
keystore-type|JKS}
ufsparams_config.configurator.core.kafka.configurator.client
.ssl-truststore-
type=${global.platform.kafka.configurator.client.ssl-
truststore-type|JKS}
ufsparams_config.configurator.core.kafka.configurator.client
.ssl-endpoint-identification-
algorithm=${global.platform.kafka.ssl.endpoint.identificatio
n.algorithm|}
ufsparams_config.configurator.core.protectedRules.minLength=
${global.platform.configurator.core.protectedRules.minLength
|8}
ufsparams_config.configurator.core.protectedRules.exclusions
=${global.platform.configurator.core.protectedRules.exclusio
ns|"1234567890"}  # Tomcat settings
ufsparams_config.configurator.server.max-http-header-
size=${global.server.max-http-header-size|32KB}  # Аудит
#'configurator' configurator.core.audit.group=configurator
configurator.core.audit2.baseUrl=${global.platform.pprb.base
url.audit|}  # SPAS configurator.core.clipas.spas-server-
url=${global.platform.spas.clipas.spas-server-url|}  #
UNIMON configurator.prometheus.io.scrape='true'
configurator.prometheus.io.path='/actuator/prometheus'
configurator.management.server.port=8081  # JAVA опции
ufsparams_config_configurator_server_ose_configmap_javaArgum
ents=-Dmodule.id=configurator-module  # Config for AGENT
ufsparams_config.configurator.configuration.selector='config
urator-yaml'
ufsparams_config.configurator.configuration.managed='true'
ufsparams_config.configurator.configuration.reloader.match='
true'
ufsparams_config.configurator.configuration.reloader.search=
'true'  # UFS
configurator.platform.environment.product.subsystem=CONFIGUR
ATOR_UI configurator.platform.environment.product.api-
version=1.2
configurator.platform.environment.product.deploymentUnit=con
figurator-module configurator.platform.config.deployment-
unit=configurator-module
configurator.ufs.monitoring.client.subsystem=CONFIGURATOR_UI
```

```
configurator.ufs.monitoring.client.title=configurator-module
# ISTIO # Размеры запросов для Istio-sidecar
configurator.ose.deployment.metadata.annotations.sidecar.ist
io.io.proxyCPU=200m
configurator.ose.deployment.metadata.annotations.sidecar.ist
io.io.proxyMemory=256Mi  # Размеры лимитов для Istio-sidecar
configurator.ose.deployment.metadata.annotations.sidecar.ist
io.io.proxyCPULimit=300m
configurator.ose.deployment.metadata.annotations.sidecar.ist
io.io.proxyMemoryLimit=256Mi  # Настройки podAntiAffinity
configurator.ose.feature.SelfAntiAffinity.enabled=${global.f
eature.SelfAntiAffinity.enabled|true}  # Настройки Graceful
Shutdown
configurator.ose.deployment.spec.template.spec.terminationGr
acePeriodSeconds=${global.ufs.terminationGracePeriodSeconds|
30}  # Настройки PodDisruptionBudget
configurator.ose.poddisruptionbudget.spec.minAvailable=${glo
bal.poddisruptionbudget.minAvailable|50%}
```

ufsparams_config_configurator_rest.conf – конфигурационный файл параметров для настройки rest-сервиса компонента «Управление конфигурациями».

```
# Параметры для контейнеров среды контейнеризации #
Параметры для автоскейлинга
configurator_rest.openshift.autoscaling.min_replicas=1  #
Размеры лимитов для pod
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.resources.limits.cpu=1000m
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.resources.limits.memory=1000Mi  #
Размеры запросов для pod
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.resources.requests.cpu=500m
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.resources.requests.memory=1000Mi  #
Настройки Liveness Probe
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.livenessProbe.initialDelaySeconds=30
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.livenessProbe.periodSeconds=5
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.livenessProbe.timeoutSeconds=1
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.livenessProbe.successThreshold=1
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.livenessProbe.failureThreshold=3  #
Настройки Readiness Probe
configurator_rest.ose.deployment.spec.template.spec.containe
```

```
rs.configurator_rest.readinessProbe.initialDelaySeconds=30
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.readinessProbe.periodSeconds=1
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.readinessProbe.timeoutSeconds=5
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.readinessProbe.successThreshold=1
configurator_rest.ose.deployment.spec.template.spec.containe
rs.configurator_rest.readinessProbe.failureThreshold=3  #
Параметры Конфигуратора
configurator_rest.spring.active.profiles=PROM
configurator_rest.core.rest-enabled=false
configurator_rest.core.db.url=${jdbc.efs.pstgr.url}
configurator_rest.core.kafka.bootstrap.servers=${global.plat
form.kafka.bootstrap.servers}
configurator_rest.core.kafka.security.protocol=PLAINTEXT
configurator_rest.ufs.security.client.mode=PROM  # Tomcat
settings configurator_rest.server.max-http-header-
size=${global.server.max-http-header-size|32KB}  # Аудит
#'configurator-rest'
configurator_rest.core.audit.group=configurator-rest
configurator_rest.core.audit2.baseUrl=${global.platform.pprb
.baseurl.audit|}  #UNIMON
configurator_rest.prometheus.io.scrape='true'
configurator_rest.prometheus.io.path='/actuator/prometheus'
configurator_rest.management.server.port=8081  #JAVA опции
ufsparams_config_configurator_rest_ose_configmap_javaArgumen
ts=-Dmodule.id=configurator-rest  #Config for AGENT
ufsparams_config.configurator_rest.configuration.selector='c
onfigurator-yaml'
ufsparams_config.configurator_rest.configuration.managed='tr
ue'
ufsparams_config.configurator_rest.configuration.reloader.ma
tch='true'
ufsparams_config.configurator_rest.configuration.reloader.se
arch='true'  #UFS
configurator_rest.platform.environment.product.subsystem=CON
FIGURATOR_REST
configurator_rest.platform.environment.product.api-
version=1.2
configurator_rest.platform.environment.product.deploymentUni
t=configurator-rest
configurator_rest.platform.config.deployment-
unit=configurator-rest
configurator_rest.ufs.monitoring.client.subsystem=CONFIGURAT
OR_REST
configurator_rest.ufs.monitoring.client.title=configurator-
rest
configurator_rest.ssl.keystore.path=/ansible/files/ssl/confi
gurator-rest-keystore.p12
```

```
configurator_rest.ssl.truststore.path=/ansible/files/ssl/con
figurator-rest-truststore.jks  #ISTIO # Размеры запросов для
Istio-sidecar
configurator_rest.ose.deployment.metadata.annotations.sideca
r.istio.io.proxyCPU=200m
configurator_rest.ose.deployment.metadata.annotations.sideca
r.istio.io.proxyMemory=256Mi  # Размеры лимитов для Istio-
sidecar
configurator_rest.ose.deployment.metadata.annotations.sideca
r.istio.io.proxyCPULimit=300m
configurator_rest.ose.deployment.metadata.annotations.sideca
r.istio.io.proxyMemoryLimit=256Mi  # Настройки
podAntiAffinity
configurator_rest.ose.feature.SelfAntiAffinity.enabled=${glo
bal.feature.SelfAntiAffinity.enabled|true}  # Настройки
Graceful Shutdown
configurator_rest.ose.deployment.spec.template.spec.terminat
ionGracePeriodSeconds=${global.ufs.terminationGracePeriodSec
onds|30}  # Настройки PodDisruptionBudget
configurator_rest.ose.poddisruptionbudget.spec.minAvailable=
${global.poddisruptionbudget.minAvailable|50%}
```

- ufsparams_config_bh.conf – конфигурационный файл параметров для настройки APM компонента «Управление параметрами» (ufs-params-manager)).

```
# Компонент «Управление параметрами» ## bh (ufs-params-
manager)  ## Стандартизованные параметры для Kubernetes/OSE
контейнеров ## Параметры для автоскейлинга
ufsparams_config_bh.ose.hpa.max_replicas=1
ufsparams_config_bh.ose.hpa.min_replicas=1
ufsparams_config_bh.ose.hpa.targetCPUUtilizationPercentage=3
0  ## Параметры стратегии развертывания ## Время ожидания
между опросом состояния установки после обновления
ufsparams_config_bh.ose.deployment.spec.strategy.rollingPara
ms.intervalSeconds=${global.ufs.strategy.intervalSeconds|1}
## Максимальное число подов, которые могут быть
запланированы выше исходного количества подов (можно в
процентах, можно явное количество)
ufsparams_config_bh.ose.deployment.spec.strategy.rollingPara
ms.maxSurge=${global.ufs.strategy.maxSurge|25%} ##
Максимальное количество подов, которые могут быть недоступны
во время обновления (можно в процентах, можно явное
количество)
ufsparams_config_bh.ose.deployment.spec.strategy.rollingPara
ms.maxUnavailable=${global.ufs.strategy.maxUnavailable|25%}
## Время ожидания поднятия подов, прежде чем сдаться и
откатится к предыдущей успешной установки
ufsparams_config_bh.ose.deployment.spec.strategy.rollingPara
```

```
ms.timeoutSeconds=${global.ufs.strategy.timeoutSeconds|600}
## Время ожидания между отдельными обновлениями POD
ufsparams_config_bh.ose.deployment.spec.strategy.rollingPara
ms.updatePeriodSeconds=${global.ufs.strategy.updatePeriodSec
onds|1}  ## Размеры хипов и прочие параметры JAVA_OPTS
ufsparams_config_bh.ose.configmap.javaArguments=!-
XX:+UseContainerSupport -XX:MaxRAMPercentage=75.0  ##
Размеры лимитов для pod
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.resources.limits.cpu=2000m
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.resources.limits.memory=4000Mi  ##
Размеры запросов для pod
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.resources.requests.cpu=1000m
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.resources.requests.memory=4000Mi
ufsparams_config_bh.ose.deployment.spec.replicas=1  #
Размеры запросов для Sidecar
ufsparams_config_bh.ose.deployment.metadata.annotations.side
car.istio.io.proxyCPU=200m
ufsparams_config_bh.ose.deployment.metadata.annotations.side
car.istio.io.proxyMemory=256Mi  # Размеры лимитов для
Sidecar
ufsparams_config_bh.ose.deployment.metadata.annotations.side
car.istio.io.proxyCPULimit=300m
ufsparams_config_bh.ose.deployment.metadata.annotations.side
car.istio.io.proxyMemoryLimit=256Mi  ## Параметры Datasource
jdbc_UFSPARAMS_PG_url=${jdbc.efs.pstgr.url}
jdbc_UFSPARAMS_PG_driver=org.postgresql.Driver  # Настройки
Liveness Probe
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.livenessProbe.initialDelaySeconds=6
0
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.livenessProbe.periodSeconds=5
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.livenessProbe.timeoutSeconds=1
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.livenessProbe.successThreshold=1
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.livenessProbe.failureThreshold=3  #
Настройки Readiness Probe
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.readinessProbe.initialDelaySeconds=
60
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.readinessProbe.periodSeconds=1
ufsparams_config_bh.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_bh.readinessProbe.timeoutSeconds=1
```

```
ufsparams_config_bh.ose.deployment.spec.template.spec.containers.ufsparams_config_bh.readinessProbe.successThreshold=1
ufsparams_config_bh.ose.deployment.spec.template.spec.containers.ufsparams_config_bh.readinessProbe.failureThreshold=3
# Update config parameters
ufsparams_config_update_config_attempt_periodSeconds=3
ufsparams_config_update_config_max_attempts=5
ufsparams_config_update_config_connect_timeout=2  # Путь до открытого ключа iam iam.auth.publickey.file=/certs/iam-public-key.pem  # Имя проекта Приложения-потребителя на стенде. Используется DevOps Pipeline. Имя переменной должно быть строго таким – openshift.projectName. # Задается в каждом файле при раздельной установке отдельных компонентов приложения. Используется на среде Dev (в случае, когда не задан параметр в environment.json)
openshift.projectName=dummy  # Параметры для формирования имени json-файла с конфигурацией APM компонента «Управление параметрами» application.deploymentUnit=PACMAN_ADMIN
ufsparams_config_bh_sup2_scopeTemplate={{sup2_scopeTemplate}}  # Tomcat settings ufsparams_config_bh.server.tomcat.max-threads = 200 ufsparams_config_bh.server.tomcat.min-spare-threads = 10 ufsparams_config_bh.server.max-http-header-size=${global.server.max-http-header-size|32KB}  # Аудит #'manager-web' ufsparams_config_bh.audit.group=manager-web
ufsparams_config_bh.audit2.baseUrl=${global.platform.pprb.baseurl.audit|}  # SPAS ufsparams_config_bh.clipas.spas-server-url=${global.platform.spas.clipas.spas-server-url|}
# UNIMON ufsparams_config_bh.prometheus.io.scrape='true'
ufsparams_config_bh.prometheus.io.path='/monitoring/metrics'
ufsparams_config_bh.management.server.port=8081  # Config for AGENT
ufsparams_config_bh.configuration.selector='configurator-yaml' ufsparams_config_bh.configuration.managed='true'
ufsparams_config_bh.configuration.reloader.match='true'
ufsparams_config_bh.configuration.reloader.search='true'  # UFS
ufsparams_config_bh.platform.environment.product.subsystem=PACMAN_ADMIN
ufsparams_config_bh.platform.environment.product.api-version=1.2
ufsparams_config_bh.platform.environment.product.deploymentUnit=PACMAN_ADMIN
ufsparams_config_bh.platform.config.deployment-unit=PACMAN_ADMIN
ufsparams_config_bh.ufs.monitoring.client.subsystem=PACMAN_ADMIN
ufsparams_config_bh.ufs.monitoring.client.title=manager-web
ufsparams_config_bh.spring.active.profiles=PROM  # Настройки podAntiAffinity
ufsparams_config_bh.ose.feature.SelfAntiAffinity.enabled=${g
```

```
lobal.feature.SelfAntiAffinity.enabled|true}  # Настройки
Graceful Shutdown
ufsparams_config_bh.ose.deployment.spec.template.spec.termin
ationGracePeriodSeconds=${global.ufs.terminationGracePeriodS
econds|30}  # Настройки PodDisruptionBudget
ufsparams_config_bh.ose.poddisruptionbudget.spec.minAvailabl
e=${global.poddisruptionbudget.minAvailable|50%}
```

- ufsparams_config_import.conf – конфигурационный файл параметров для настройки сервиса импорта компонента «Управление параметрами».

```
# Стандартизованные параметры для контейнеров # Параметры
для автоскейлинга
ufsparams_config_import.ose.hpa.max_replicas=1
ufsparams_config_import.ose.hpa.min_replicas=1
ufsparams_config_import.ose.hpa.targetCPUUtilizationPercenta
ge=30  # Параметры стратегии развертывания # Время ожидания
между опросом состояния установки после обновления
ufsparams_config_import.ose.deployment.spec.strategy.rolling
Params.intervalSeconds=${global.ufs.strategy.intervalSeconds
|1} # Максимальное число подов, которые могут быть
запланированы выше исходного количества подов (можно в
процентах, можно явное количество)
ufsparams_config_import.ose.deployment.spec.strategy.rolling
Params.maxSurge=${global.ufs.strategy.maxSurge|25%} #
Максимальное количество подов, которые могут быть недоступны
во время обновления (можно в процентах, можно явное
количество)
ufsparams_config_import.ose.deployment.spec.strategy.rolling
Params.maxUnavailable=${global.ufs.strategy.maxUnavailable|2
5%} # Время ожидания поднятия подов, прежде чем сдаться и
откатится к предыдущей успешной установке
ufsparams_config_import.ose.deployment.spec.strategy.rolling
Params.timeoutSeconds=${global.ufs.strategy.timeoutSeconds|6
00} # Время ожидания между отдельными обновлениями POD
ufsparams_config_import.ose.deployment.spec.strategy.rolling
Params.updatePeriodSeconds=${global.ufs.strategy.updatePerio
dSeconds|1}  # Размеры хипов и прочие параметры JAVA_OPTS
ufsparams_config_import.ose.configmap.javaArguments=!-
XX:+UseContainerSupport -XX:MaxRAMPercentage=75.0  # Размеры
лимитов для pod
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.resources.limits.cpu=2000m
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.resources.limits.memory=400
0Mi  # Размеры запросов для pod
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.resources.requests.cpu=1000
```

```
m
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.resources.requests.memory=4
000Mi ufsparams_config_import.ose.deployment.spec.replicas=1
# Размеры запросов для sidecar
ufsparams_config_import.ose.deployment.metadata.annotations.
sidecar.istio.io.proxyCPU=200m
ufsparams_config_import.ose.deployment.metadata.annotations.
sidecar.istio.io.proxyMemory=256Mi  # Размеры лимитов для
sidecar
ufsparams_config_import.ose.deployment.metadata.annotations.
sidecar.istio.io.proxyCPULimit=300m
ufsparams_config_import.ose.deployment.metadata.annotations.
sidecar.istio.io.proxyMemoryLimit=256Mi  # Параметры
Datasource jdbc_UFSPARAMS_PG_url=${jdbc.efs.pstgr.url}
jdbc_UFSPARAMS_PG_driver=org.postgresql.Driver  # Настройки
Liveness Probe
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.livenessProbe.initialDelayS
econds=60
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.livenessProbe.periodSeconds
=5
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.livenessProbe.timeoutSecond
s=1
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.livenessProbe.successThresh
old=1
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.livenessProbe.failureThresh
old=3  # Настройки Readiness Probe
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.readinessProbe.initialDelay
Seconds=60
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.readinessProbe.periodSecond
s=1
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.readinessProbe.timeoutSecon
ds=1
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.readinessProbe.successThres
hold=1
ufsparams_config_import.ose.deployment.spec.template.spec.co
ntainers.ufsparams_config_import.readinessProbe.failureThres
hold=3  # Update config parameters
ufsparams_config_update_config_attempt_periodSeconds=3
ufsparams_config_update_config_max_attempts=5
ufsparams_config_update_config_connect_timeout=2  # Имя
```

```
проекта Приложения-потребителя на стенде. Используется
DevOps Pipeline. Имя переменной должно быть строго таким –
openshift.projectName. # Задается в каждом файле при
раздельной установке отдельных компонентов приложения.
Используется на среде Dev (в случае, когда не задан параметр
в environment.json) openshift.projectName=dummy  # Параметры
для формирования имени json-файла с конфигурацией сервиса
импорта компонента «Управление параметрами»
application.deploymentUnit=PACMAN_IMPORT
ufsparams_config_import_sup2_scopeTemplate={{sup2_scopeTempl
ate}}  # Tomcat settings
ufsparams_config_import.server.tomcat.max-threads = 200
ufsparams_config_import.server.tomcat.min-spare-threads = 10
ufsparams_config_import.server.max-http-header-
size=${global.server.max-http-header-size|32KB}  # Аудит
#'ose-import-web' ufsparams_config_import.audit.group=ose-
import-web
ufsparams_config_import.audit2.baseUrl=${global.platform.ppr
b.baseurl.audit|}  # SPAS
ufsparams_config_import.clipas.spas-server-
url=${global.platform.spas.clipas.spas-server-url|}  #UNIMON
ufsparams_config_import.prometheus.io.scrape='true'
ufsparams_config_import.prometheus.io.path='/monitoring/metr
ics' ufsparams_config_import.management.server.port=8081
#Config for AGENT
ufsparams_config_import.configuration.selector='configurator
-yaml' ufsparams_config_import.configuration.managed='true'
ufsparams_config_import.configuration.reloader.match='true'
ufsparams_config_import.configuration.reloader.search='true'
#UFS
ufsparams_config_import.platform.environment.product.subsyst
em=PACMAN_IMPORT
ufsparams_config_import.platform.environment.product.api-
version=1.2
ufsparams_config_import.platform.environment.product.deploym
entUnit=PACMAN_IMPORT
ufsparams_config_import.platform.config.deployment-
unit=PACMAN_IMPORT
ufsparams_config_import.ufs.monitoring.client.subsystem=PACM
AN_IMPORT
ufsparams_config_import.ufs.monitoring.client.title=ose-
import-web
ufsparams_config_import.spring.active.profiles=PROM  #
Настройки podAntiAffinity
ufsparams_config_import.ose.feature.SelfAntiAffinity.enabled
=${global.feature.SelfAntiAffinity.enabled|true}  #
Настройки Graceful Shutdown
ufsparams_config_import.ose.deployment.spec.template.spec.te
rminationGracePeriodSeconds=${global.ufs.terminationGracePer
iodSeconds|30}  # Настройки PodDisruptionBudget
```

```
ufsparams_config_import.ose.poddisruptionbudget.spec.minAvai
lable=${global.poddisruptionbudget.minAvailable|50%}
```

- ufsparams_config_ms.conf – конфигурационный файл параметров для настройки сервиса доставки конфигураций компонента «Управление параметрами» (ufs-params-service).

```
# Компонент «Управление параметрами» # ms (ufs-params-
service)  # Стандартизованные параметры для контейнеров #
Параметры для автоскейлинга
ufsparams_config_ms.ose.hpa.max_replicas=1
ufsparams_config_ms.ose.hpa.min_replicas=1
ufsparams_config_ms.ose.hpa.targetCPUUtilizationPercentage=3
0  # Параметры стратегии развертывания # Время ожидания
между опросом состояния установки после обновления
ufsparams_config_ms.ose.deployment.spec.strategy.rollingPara
ms.intervalSeconds=${global.ufs.strategy.intervalSeconds|1}
# Максимальное число pods, которые могут быть запланированы
выше исходного количества pods (можно в процентах, можно
явное количество)
ufsparams_config_ms.ose.deployment.spec.strategy.rollingPara
ms.maxSurge=${global.ufs.strategy.maxSurge|25%} #
Максимальное количество pods, которые могут быть недоступны
во время обновления (можно в процентах, можно явное
количество)
ufsparams_config_ms.ose.deployment.spec.strategy.rollingPara
ms.maxUnavailable=${global.ufs.strategy.maxUnavailable|25%}
# Время ожидания поднятия pods, прежде чем сдаться и
откатится к предыдущей успешной установке
ufsparams_config_ms.ose.deployment.spec.strategy.rollingPara
ms.timeoutSeconds=${global.ufs.strategy.timeoutSeconds|600}
# Время ожидания между отдельными обновлениями POD
ufsparams_config_ms.ose.deployment.spec.strategy.rollingPara
ms.updatePeriodSeconds=${global.ufs.strategy.updatePeriodSec
onds|1}  # Размеры хипов и прочие параметры JAVA_OPTS
ufsparams_config_ms.ose.configmap.javaArguments=!-
XX:+UseContainerSupport -XX:MaxRAMPercentage=75.0  # Размеры
лимитов для pod
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.resources.limits.cpu=2000m
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.resources.limits.memory=4000Mi  #
Размеры запросов для pod
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.resources.requests.cpu=1000m
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.resources.requests.memory=4000Mi
ufsparams_config_ms.ose.deployment.spec.replicas=1  #
```

```
Размеры запросов для sidecar
ufsparams_config_ms.ose.deployment.metadata.annotations.side
car.istio.io.proxyCPU=200m
ufsparams_config_ms.ose.deployment.metadata.annotations.side
car.istio.io.proxyMemory=256Mi  # Размеры лимитов для
sidecar
ufsparams_config_ms.ose.deployment.metadata.annotations.side
car.istio.io.proxyCPULimit=300m
ufsparams_config_ms.ose.deployment.metadata.annotations.side
car.istio.io.proxyMemoryLimit=256Mi  # Параметры Datasource
jdbc_UFSPARAMS_PG_url=${jdbc.efs.pstgr.url}
jdbc_UFSPARAMS_PG_driver=org.postgresql.Driver  # Настройки
Liveness Probe
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.livenessProbe.initialDelaySeconds=6
0
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.livenessProbe.periodSeconds=5
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.livenessProbe.timeoutSeconds=1
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.livenessProbe.successThreshold=1
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.livenessProbe.failureThreshold=3  #
Настройки Readiness Probe
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.readinessProbe.initialDelaySeconds=
60
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.readinessProbe.periodSeconds=1
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.readinessProbe.timeoutSeconds=1
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.readinessProbe.successThreshold=1
ufsparams_config_ms.ose.deployment.spec.template.spec.contai
ners.ufsparams_config_ms.readinessProbe.failureThreshold=3
# Имя проекта Приложения-потребителя на стенде. Используется
DevOps Pipeline. Имя переменной должно быть строго таким –
openshift.projectName. # Задается в каждом файле при
раздельной установке отдельных компонентов приложения.
Используется на среде Dev (в случае, когда не задан параметр
в environment.json) openshift.projectName=dummy  # Параметры
для формирования имени json-файла с конфигурацией сервиса
доставки конфигураций компонента «Управление параметрами»
application.deploymentUnit=PACMAN_MS
ufsparams_config_ms_sup2_scopeTemplate={{sup2_scopeTemplate}
}  # Tomcat settings ufsparams_config_ms.server.max-http-
header-size=${global.server.max-http-header-size|32KB}  #
Аудит #'ose-service-web'
ufsparams_config_service.audit.group=ose-service-web
```

```
ufsparams_config_service.audit2.baseUrl=${global.platform.pp
rb.baseurl.audit|}  # SPAS
ufsparams_config_service.clipas.spas-server-
url=${global.platform.spas.clipas.spas-server-url|}  #
UNIMON ufsparams_config_service.prometheus.io.scrape='true'
ufsparams_config_service.prometheus.io.path='/monitoring/met
rics' ufsparams_config_service.management.server.port=8081
# Config for AGENT
ufsparams_config_service.configuration.selector='configurato
r-yaml'
ufsparams_config_service.configuration.managed='true'
ufsparams_config_service.configuration.reloader.match='true'
ufsparams_config_service.configuration.reloader.search='true
'  # UFS
ufsparams_config_service.platform.environment.product.subsys
tem=PACMAN_MS
ufsparams_config_service.platform.environment.product.api-
version=1.2
ufsparams_config_service.platform.environment.product.deploy
mentUnit=PACMAN_MS
ufsparams_config_service.platform.config.deployment-
unit=PACMAN_MS
ufsparams_config_service.ufs.monitoring.client.subsystem=PAC
MAN_MS
ufsparams_config_service.ufs.monitoring.client.title=ose-
service-web
ufsparams_config_service.spring.active.profiles=PROM  #
Настройки podAntiAffinity
ufsparams_config_ms.ose.feature.SelfAntiAffinity.enabled=${g
lobal.feature.SelfAntiAffinity.enabled|true}  # Настройки
Graceful Shutdown
ufsparams_config_ms.ose.deployment.spec.template.spec.termin
ationGracePeriodSeconds=${global.ufs.terminationGracePeriodS
econds|30}  # Настройки PodDisruptionBudget
ufsparams_config_ms.ose.poddisruptionbudget.spec.minAvailabl
e=${global.poddisruptionbudget.minAvailable|50%}
```

Запуск liquibase скриптов. Настройка подключения Базы данных

Принципы работы с объектами БД:

- Набор выделяемых полномочий должен быть минимально необходимым и достаточным для корректной работы сервиса.
- Должно быть исключено использование перекрестных и взаимозависимые полномочий между схемами хранения.
- В каждой инсталляции на уровне отдельного экземпляра БД должен вводиться уникальный идентификатор-суффикс, позволяющий определять принадлежность к инсталляции. Основное назначение уникального идентификатора – это

минимизировать ошибки копирования при ручных операциях во множество различных баз данных.

- Исходя из разного типа нагрузки на ввод-вывод, хранение сегментов таблиц, индексов и больших объектов типа LOB необходимо разделять на отдельные выделенные табличные пространства. Не допускается размещение в одном табличном пространстве содержимого разных типов (например, индексов вперемешку с данными).

Запуск liquibase скриптов компонента «Управление конфигурациями».

Файл дистрибутива для раскатки скриптов: `package/db/pacman.configurator.app.zip`.

Параметры подключения:

- `jdbc URL` берется из глобального параметра `EFS_PSTGR_DB_URL`;
- `jdbcDriver`: `org.postgresql.Driver`;
- схема БД: `schema_name.placeholder` = `cfga_<суффикс_блока>`;
- Префикс для хранения объктов БД: `cfga_`;
- Default Tablespace: `cfga_ts_data`;
- Required Tablespace:
  - `ts_data.placeholder` = `cfga_ts_data`;
  - `ts_idx.placeholder` = `cfga_ts_idx`;
- пользователей и пароли указываются в secret `secret-secret`:
  - пользователь:`jdbc.CONFIGURATOR_PG.user`;
  - пароль: `jdbc.CONFIGURATOR_PG.password`.

Запуск liquibase скриптов компонента «Управление параметрами».

Файл дистрибутива для раскатки скриптов: `package/db/ufs.sup.app.zip`.

Параметры подключения:

- `jdbc URL` берется из глобального параметра `{{ EFS_PSTGR_DB_URL }}`;
- `jdbcDriver`: org.postgresql.Driver;
- схема БД: `schema_name.placeholder` = `cfge_<суффикс_блока>`;
- Префикс для хранения объктов БД: `cfge_`;
- Default Tablespace: `cfge_ts_data`;
- Required Tablespace:
  - `ts_data.placeholder` = `cfge_ts_data`;
  - `ts_idx.placeholder` = `cfge_ts_idx`;
- пользователей и пароли указываются в secret `secret-secret`:
  - пользователь:`jdbc.UFSPARAMS_PG.user`;
  - пароль: `jdbc.UFSPARAMS_PG.password`.

Имя пользователя и схема хранения.

Имя пользователя и схема хранения задаются по шаблону:

```
create user :schema_name.placeholder with encrypted password
:password.placeholder; create schema
:schema_name.placeholder; grant connect on database
:database_name.placeholder to :schema_name.placeholder;
grant all on schema :schema_name.placeholder to
:schema_name.placeholder; alter user
:schema_name.placeholder valid until :time.placeholder;
grant usage on schema :schema_name.placeholder to
:schema_name.placeholder;    create tablespace
:ts_data.placeholder owner :schema_name.placeholder location
:YYY_location.placeholder; create tablespace
:ts_idx.placeholder owner :schema_name.placeholder location
:ZZZ_location.placeholder;
```

Специальная настройка окружения custom_property.conf.yml для обновления БД.

Прикладной `custom_property.conf.yml`:

```
#Определение БД для обновления при инсталляции: только
компонент «Управление параметрами» ("only_sup"), только
компонент «Управление конфигурациями» ("only_cfg") или обе
("both") dbscripts_mode: "only_cfg"
#ufsparams_config_liquidbase_jdbc_cfge_pstgr_url:
jdbc:postgresql://хост:порт/cfge
ufsparams_config_liquidbase_jdbc_cfgp_pstgr_url:
jdbc:postgresql://хост:порт/cfg  configurator_db_schema:
public configurator_db_tablespace_seed: pg_default
configurator_db_index_seed: pg_default
```

, где:

- `dbscripts_mode` – определение базы данных для обновления при инсталляции:
  - `only_sup` – обновлять базу данных только для компонента «Управление параметрами»;
  - `only_cfg` – обновлять базу данных только для компонента «Управление конфигурациями»;
  - `both` – обновлять базы данных для обоих компонентов;
- `ufsparams_config_liquidbase_jdbc_cfge_pstgr_url` – настройка URL для подключения к БД для обновления БД модулей компонента «Управление параметрами»;
- `ufsparams_config_liquidbase_jdbc_cfgp_pstgr_url` – настройка URL для подключения к БД для обновления БД модулей компонента «Управление конфигурациями».

ufsparams_config_liquidbase_jdbc_cfge_pstgr_url и ufsparams_config_liquidbase_jdbc_cfgp_pstgr_url настраиваются в случае, если у вас отличается порт подключения к БД в рамках среды контейнеризации и за его пределами.

Параметры:

- configurator_db_schema – схема;
- configurator_db_tablespace_seed – пространство для размщения таблиц;
- configurator_db_index_seed – пространство для размщения индексов

настраиваются, если устанавливаются модули компонента «Управление конфигурациями» для компонента «Управление конфигурациями» 3G, либо не были выполненны старндарты БД.

Импорт параметров

Импорт параметров происходит из папки дистрибутива:

```
├── conf │   └── data │   │   ├── audit
│   │   │   ├── ufsparams_audit.json            # Файл для
добавления названий подсистем в справочник АРМа компонента
Platform V Audit SE |   │   ├── logger       |   │   │   ├──
sup_logger.json                 # Файл для добавления
названий подсистем в справочник АРМа компонента
журналирования |   │   │   ├── sup_logger_filters.properties
# Файл для добавления параметров (в т.ч. параметров
регулирования уровня логирования для подсистем компонента
«Управление параметрами») в справочник АРМа компонента
журналирования |   │   ├── security       |   │   │   ├──
security-data.xml               # Файл для импорта ролевой
модели в удаленный компонент Авторизации для доступа
пользователей в АРМ компонента «Управление параметрами» |
│   │   ├── securiti-data-spas.xml          # Ролевая модель
компонента «Управление конфигурациями» в SPAS |   │   ├── sm
|   │   │   ├── operation |   │   │   ├── buttons.json
# Файл для импорта настроек для входа в компонент
«Управление параметрами» через единое окно стартового
менеджера |   │   ├── sup2                                #
Файл для импорта параметров в БД компонента «Управление
параметрами» |   │   │   ├── definitions
# Файлы конфигураций (параметров и их значений) |   │   |
│   ├── PACMAN_ADMIN.json          # Файл для импорта
параметров в БД для АРМ компонента «Управление параметрами»
|   │   │   │   ├── PACMAN_IMPORT.json          # Файл для
импорта параметров в БД для сервиса Импорта компонента
«Управление параметрами» |   │   │   ├── PACMAN_MS.json
```

```
# Файл для импорта параметров в БД для сервиса выдачи
конфигурации компонента «Управление параметрами». На текущий
момент не содержит уникальных параметров конфигурирования |
|   |   ├── parameters                    # Фалы для
параметризации плейсхолдеров, используемых в описании
параметров из папки definitions |   |   |   |   ├──
PACMAN_ADMIN.conf           # Файл параметризации для APM
компонента «Управление параметрами» |   |   |   |   ├──
PACMAN_IMPORT.conf          # Файл параметризации для
сервиса Импорта компонента «Управление параметрами» |   |
|   |   ├── PACMAN_MS.conf              # Файл
параметризации для сервиса выдачи конфигурации компонента
«Управление параметрами». На текущий момент не содержит
уникальных параметров конфигурирования |   └── k8s |
└── base |           ├── uko_ufsparams_bh_v2
|           |   ├── configmaps  |           |   |   ├── ufs-
params-manager.yaml    # Файл с конфигурацией APM компонента
«Управление параметрами» для записи в ConfigMap, монтируется
по адресу `/opt/conf/ufsparams/config` |           ├──
uko_ufsparams_import_v2
|           |   ├── configmaps   |           |   |   ├──
ufs-params-import.yaml    # Файл с конфигурацией сервиса
Импорта компонента «Управление параметрами» для записи в
ConfigMap, монтируется по адресу
`/opt/conf/ufsparams/config` |           ├──
uko_ufsparams_ms_v2                                    |
|   ├── configmaps  |           |   |   ├── ufs-params-
service.yaml        # Файл с конфигурацией сервиса выдачи
параметров компонента «Управление параметрами» для записи в
ConfigMap, монтируется по адресу
`/opt/conf/ufsparams/config` |           ├── pacman_agent |
|   ├── configmaps  |           |   |   ├── pacman-
agent.yaml       # Файл с конфигурацией Агента в yaml
формате для записи в ConfigMap, монтируется по адресу
`/config` с именем `cfg` |           ├── configurator_server
|           |   ├── configmaps  |           |   |   ├──
configurator-server.yaml    # Файл с конфигурацией APM
компонента «Управление конфигурациями» для записи в
ConfigMap |           ├── configurator_rest
|           |   ├── configmaps  |           |   |   ├──
configurator-rest.yaml    # Файл с конфигурацией Rest-
сервиса компонента «Управление конфигурациями» для записи в
ConfigMap
```

Примечание: импорт параметров в БД для компонента «Управление параметрами» из папки `sup2` идет с помощью curl. Корректный формат адреса Rest-микросервиса групповой вставки параметров: `https://host:port/ufsparams-import-v2/file/curl/import/`. При стандартном импорте параметров не происходит перезапись данных, только добавление. Для того, чтобы работала перезапись

параметров в БД при импорте, необходимо в environment.json перед установкой выставить параметр `importSupRw: true`.

ConfigMaps с параметрами компонентов для отслеживания Агентом конфигурации (pacman_agent)

Файлы конфигураций помещаются в пространство проекта в виде ConfigMaps. Данные ConfigMaps будут отслеживаться Агентом конфигурации (pacman_agent).

1. ConfigMap `pacman-agent.${distrib.release.version}` с параметрами для Агента конфигурации:

```
apiVersion: v1 kind: ConfigMap metadata:   annotations:
pacman.sbt.ru/id:
'${jenkins_env.fp_artifact_version},${projectName},pacman-
agent'     pacman.sbt.ru/rn: ${ufs.tenant.code}
pacman.sbt.ru/selector:
${ufsparams_config_pacman_agent.configuration.selector}
reloader.stakater.com/match:
${ufsparams_config_pacman_agent.configuration.reloader.match
}   labels:     pacman.sbt.ru/managed:
${ufsparams_config_pacman_agent.configuration.managed}
agent: pacman   name: cfga-cm-pacman-agent-{{
vars['distrib.release.version'] }} data:   pacman-agent-
${projectName}--
${ufsparams_config_pacman_agent_configuration_version}.yml:
|     spring:       jackson:         serialization:
indent_output: true        application:         name: agent-
app       profiles:         active:
${ufsparams_config_pacman_agent.spring.active.profiles}
withoutEndpoints: true      server:       port: 8888
servlet:         context-path: /pacman-agent       ssl:
enabled: false      dependenciesInfoFileName: 'dependencies'
ufs:       platform:         config:           deployment-
unit:
${ufsparams_config_pacman_agent.platform.config.deployment-
unit}           agent:             disabled: true
environment:           product:             subsystem:
${ufsparams_config_pacman_agent.platform.environment.product
.subsystem}             channel:
${ufsparams_config.platform.environment.product.channel}
version: {{ vars['distrib.release.version'] }}
api-version:
${ufsparams_config_pacman_agent.platform.environment.product
.api-version}             release:
${ufsparams_config.platform.environment.product.release}
distrib-version: ${jenkins_env.fp_artifact_version}
deploymentUnit:
```

```
${ufsparams_config_pacman_agent.platform.environment.product
.deploymentUnit}             platform: 'not used'
monitoring:         client:           channel:
${ufsparams_config.ufs.monitoring.client.channel}
subsystem:
${ufsparams_config_pacman_agent.ufs.monitoring.client.subsys
tem}           title:
${ufsparams_config_pacman_agent.ufs.monitoring.client.title}
pacman:       adapters:         monitoring:
unimon:
${ufsparams_config.pacman.adapters.monitoring.unimon}
stub: ${ufsparams_config.pacman.adapters.monitoring.stub}
ru:       sbt:          pacman:           agent:
base:               uri:
${ufsparams_config_pacman_agent.config_storage.baseurl}
poll:               delay:
${ufsparams_config_pacman_agent.delay}             http:
timeout:
${ufsparams_config_pacman_agent.config_storage.http.timeout}
circuitBreaker:                 enable:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.en
able}                 failureRate:
threshold:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.fa
ilureRate.threshold}                openState:
duration:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.op
enState.duration}                halfOpen:
requestBuffer:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.ha
lfOpen.requestBuffer}                minimumCalls:
size:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.mi
nimumCalls.size}      logging:       level:         root:
INFO       file:         path: /fluent-bit/etc/logs
config: /config-log/logback-spring.xml     management:
server:         port:
${ufsparams_config_pacman_agent.management.server.port}
endpoints:         web:           exposure:
include: "health, prometheus"       endpoint:
health:           group:             readiness:
include: "*"             liveness:              include:
"*"
```

2. ConfigMap `ufs-params-manager.${distrib.release.version}` с параметрами для АРМ компонента «Управление параметрами»:

```
apiVersion: v1 kind: ConfigMap metadata:   name: cfga-cm-
ufs-params-manager-{{ vars['distrib.release.version'] }}
annotations:     pacman.sbt.ru/id:
${ufsparams_config_bh.configuration.id}
pacman.sbt.ru/rn: ${ufs.tenant.code}
pacman.sbt.ru/selector:
${ufsparams_config_bh.configuration.selector}
reloader.stakater.com/match:
${ufsparams_config_bh.configuration.reloader.match}
labels:     pacman.sbt.ru/managed:
${ufsparams_config_bh.configuration.managed} data:
manager-web--
${ufsparams_config_bh_configuration_version}.yml: |
spring:       profiles:         active:
${ufsparams_config_bh.spring.active.profiles}       jackson:
serialization:           indent_output: true
default-property-inclusion: non_null       application:
name: manager-web       autoconfigure:         exclude:
ru.sbrf.ufs.platform.audit.AuditAutoConfiguration
server:       ssl:         enabled: false         key-store:
/etc/config/ssl/mq.jks       servlet:         context-path:
'/ufs-params-manager-v2/rest/'       port: '8080'       max-
http-header-size: ${ufsparams_config_bh.server.max-http-
header-size}      environment:       subsystem:
'PACMAN_ADMIN'       channel: 'pacman'     audit2:
url: ${ufsparams_config_bh.audit2.baseUrl}     clipas:
spas-server-url: ${ufsparams_config_bh.clipas.spas-server-
url}      pacman:       adapters:         monitoring:
unimon:
${ufsparams_config.pacman.adapters.monitoring.unimon}
stub: ${ufsparams_config.pacman.adapters.monitoring.stub}
audit:           selector:
${ufsparams_config.pacman.adapters.audit.selector}
timeout: ${ufsparams_config.pacman.adapters.audit.timeout}
group: ${ufsparams_config_bh.audit.group}
authorization:           pprb:             enable:
${ufsparams_config.pacman.adapters.authorization.pprb.enable
}             jwt:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.jwt.en
able}             sudir:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.sudir.
enable}             basic:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.basic.
enable}             tenants: ${ufs.tenant.code}
ufs:             baseUrl:
${ufsparams_config.authorization.ufs.baseUrl}/ufs-security
enable:
${ufsparams_config.pacman.adapters.authorization.ufs.enable}
sudir:               enable:
```

```
${ufsparams_config.pacman.adapters.authorization.ufs.sudir.e
nable}             iam:               enable:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ena
ble}               jws_token_header:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.jws
_token_header}              roles_path:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.rol
es_path}              publickey_locations:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.pub
lickey_locations}              audience:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.aud
ience}              ssl_required:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_required}              ssl_truststore:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_truststore}              ssl_truststore_password:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_truststore_password}              poolsize:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.poo
lsize}              cache_updatetime:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.cac
he_updatetime}              devmode:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.dev
mode}     ru:       sbt:         pacman:           resource:
name: ${ufs.tenant.code}     ufs:       audit-unsafe:
contextProperties:           UFS_SUBSYSTEM_CODE:
${ufsparams_config_bh.platform.environment.product.subsystem
}       platform:         environment:           product:
subsystem:
${ufsparams_config_bh.platform.environment.product.subsystem
}             channel:
${ufsparams_config.platform.environment.product.channel}
version: {{ vars['distrib.release.version'] }}
api-version:
${ufsparams_config_bh.platform.environment.product.api-
version}             release:
${ufsparams_config.platform.environment.product.release}
distrib-version: ${jenkins_env.fp_artifact_version}
deploymentUnit:
${ufsparams_config_bh.platform.environment.product.deploymen
tUnit}         config:           deployment-unit:
${ufsparams_config_bh.platform.config.deployment-unit}
monitoring:         client:           channel:
${ufsparams_config.ufs.monitoring.client.channel}
subsystem:
${ufsparams_config_bh.ufs.monitoring.client.subsystem}
title: ${ufsparams_config_bh.ufs.monitoring.client.title}
healthcheck:         stub-agent-enabled: true
groups:           liveness:
ru.sbrf.ufs.sup.common.healtcheck.DataSourceChecker,
```

```
ru.sbrf.ufs.sup.common.healtcheck.SpringProfileChecker
readiness:
ru.sbrf.ufs.sup.common.healtcheck.DataSourceChecker,
ru.sbrf.ufs.sup.common.healtcheck.SpringProfileChecker
errors:          client:            mode: local
file-name: "admin-errors.json"      logging:       file:
path: /fluent-bit/etc/logs       config: /config-
log/logback-spring.xml      management:       endpoints:
web:           exposure:              include: prometheus
base-path: /           path-mapping:              prometheus:
metrics       server:          port:
${ufsparams_config_bh.management.server.port}
servlet:           context-path: /monitoring
```

3. ConfigMap `ufs-params-service.${distrib.release.version}` с параметрами для Сервиса выдачи конфигурации компонента «Управление параметрами»:

```
apiVersion: v1 kind: ConfigMap metadata:   name: cfga-cm-
ufs-params-service-{{ vars['distrib.release.version'] }}
annotations:     pacman.sbt.ru/id:
${ufsparams_config_service.configuration.id}
pacman.sbt.ru/rn: ${ufs.tenant.code}
pacman.sbt.ru/selector:
${ufsparams_config_service.configuration.selector}
reloader.stakater.com/match:
${ufsparams_config_service.configuration.reloader.match}
labels:     pacman.sbt.ru/managed:
${ufsparams_config_service.configuration.managed} data:
ose-service-web--
${ufsparams_config_service_configuration_version}.yml: |
server:       ssl:         enabled: false         key-store:
/etc/config/ssl/mq.jks       servlet:         context-path:
'/paramsv2'       max-http-header-size:
${ufsparams_config_ms.server.max-http-header-size}
spring:       profiles:         active:
${ufsparams_config_service.spring.active.profiles}
application:         name: ose-service-web
autoconfigure:         exclude:
ru.sbrf.ufs.platform.audit.AuditAutoConfiguration
clipas:       spas-server-url:
${ufsparams_config_service.clipas.spas-server-url}
audit2:       url:
${ufsparams_config_service.audit2.baseUrl}     pacman:
adapters:         monitoring:           unimon:
${ufsparams_config.pacman.adapters.monitoring.unimon}
stub: ${ufsparams_config.pacman.adapters.monitoring.stub}
authorization:           pprb:             enable:
```

```
${ufsparams_config.pacman.adapters.authorization.pprb.enable
}             jwt:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.jwt.en
able}             sudir:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.sudir.
enable}             basic:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.basic.
enable}             tenants: ${ufs.tenant.code}
ufs:             baseUrl:
${ufsparams_config.authorization.ufs.baseUrl}/ufs-security
enable:
${ufsparams_config.pacman.adapters.authorization.ufs.enable}
logging:       file:         path: /fluent-bit/etc/logs
config: /config-log/logback-spring.xml     management:
endpoints:         web:           exposure:
include: prometheus           base-path: /           path-
mapping:             prometheus: metrics       server:
port: ${ufsparams_config_service.management.server.port}
servlet:           context-path: /monitoring     ufs:
audit-unsafe:          contextProperties:
UFS_SUBSYSTEM_CODE:
${ufsparams_config_service.platform.environment.product.subs
ystem}       platform:         environment:
product:             subsystem:
${ufsparams_config_service.platform.environment.product.subs
ystem}             channel:
${ufsparams_config.platform.environment.product.channel}
version: {{ vars['distrib.release.version'] }}
api-version:
${ufsparams_config_service.platform.environment.product.api-
version}             release:
${ufsparams_config.platform.environment.product.release}
distrib-version: ${jenkins_env.fp_artifact_version}
deploymentUnit:
${ufsparams_config_service.platform.environment.product.depl
oymentUnit}         config:           deployment-unit:
${ufsparams_config_service.platform.config.deployment-unit}
monitoring:         client:           channel:
${ufsparams_config.ufs.monitoring.client.channel}
subsystem:
${ufsparams_config_service.ufs.monitoring.client.subsystem}
title:
${ufsparams_config_service.ufs.monitoring.client.title}
```

4. ConfigMap `ufs-params-import.${distrib.release.version}` с параметрами для Сервиса Импорта компонента «Управление параметрами»:

```
apiVersion: v1 kind: ConfigMap metadata:   name: cfga-cm-
ufs-params-import-{{ vars['distrib.release.version'] }}
annotations:     pacman.sbt.ru/id:
'${jenkins_env.fp_artifact_version},,ose-import-web'
pacman.sbt.ru/rn: ${ufs.tenant.code}
pacman.sbt.ru/selector:
${ufsparams_config_import.configuration.selector}
reloader.stakater.com/match:
${ufsparams_config_import.configuration.reloader.match}
labels:     pacman.sbt.ru/managed:
${ufsparams_config_import.configuration.managed} data:
ose-import-web--
${ufsparams_config_import_configuration_version}.yml: |
server:       ssl:         enabled: false         key-store:
/etc/config/ssl/mq.jks       servlet:         context-path:
'/ufsparams-import-v2'       max-http-header-size:
${ufsparams_config_import.server.max-http-header-size}
spring:       profiles:         active:
${ufsparams_config_import.spring.active.profiles}
application:         name: ose-import-web
autoconfigure:         exclude:
ru.sbrf.ufs.platform.audit.AuditAutoConfiguration
clipas:       spas-server-url:
${ufsparams_config_import.clipas.spas-server-url}
audit2:       url: ${ufsparams_config_import.audit2.baseUrl}
pacman:       adapters:         monitoring:
unimon:
${ufsparams_config.pacman.adapters.monitoring.unimon}
stub: ${ufsparams_config.pacman.adapters.monitoring.stub}
audit:           selector:
${ufsparams_config.pacman.adapters.audit.selector}
timeout: ${ufsparams_config.pacman.adapters.audit.timeout}
group: ${ufsparams_config_import.audit.group}
authorization:           pprb:             enable:
${ufsparams_config.pacman.adapters.authorization.pprb.enable
}             jwt:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.jwt.en
able}             sudir:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.sudir.
enable}             basic:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.basic.
enable}             tenants: ${ufs.tenant.code}
ufs:             baseUrl:
${ufsparams_config.authorization.ufs.baseUrl}/ufs-security
enable:
${ufsparams_config.pacman.adapters.authorization.ufs.enable}
ru:       sbt:         pacman:           resource:
name: ${ufs.tenant.code}     logging:       file:
path: /fluent-bit/etc/logs       config: /config-
```

```
log/logback-spring.xml     management:       endpoints:
web:           exposure:             include: prometheus
base-path: /          path-mapping:             prometheus:
metrics       server:         port:
${ufsparams_config_import.management.server.port}
servlet:           context-path: /monitoring     ufs:
audit-unsafe:         contextProperties:
UFS_SUBSYSTEM_CODE:
${ufsparams_config_import.platform.environment.product.subsy
stem}       platform:         environment:
product:             subsystem:
${ufsparams_config_import.platform.environment.product.subsy
stem}             channel:
${ufsparams_config.platform.environment.product.channel}
version: {{ vars['distrib.release.version'] }}
api-version:
${ufsparams_config_import.platform.environment.product.api-
version}             release:
${ufsparams_config.platform.environment.product.release}
distrib-version: ${jenkins_env.fp_artifact_version}
deploymentUnit:
${ufsparams_config_import.platform.environment.product.deplo
ymentUnit}         config:           deployment-unit:
${ufsparams_config_import.platform.config.deployment-unit}
monitoring:          client:           channel:
${ufsparams_config.ufs.monitoring.client.channel}
subsystem:
${ufsparams_config_import.ufs.monitoring.client.subsystem}
title:
${ufsparams_config_import.ufs.monitoring.client.title}
```

5. ConfigMap `configurator-server.${distrib.release.version}` с параметрами для АРМ компонента «Управление конфигурациями»:

```
apiVersion: v1 kind: ConfigMap metadata:   annotations:
pacman.sbt.ru/id:
'${jenkins_env.fp_artifact_version},,configurator'
pacman.sbt.ru/rn: ${ufs.tenant.code}
pacman.sbt.ru/selector:
${ufsparams_config.configurator.configuration.selector}
reloader.stakater.com/match:
${ufsparams_config.configurator.configuration.reloader.match
}   labels:     pacman.sbt.ru/managed:
${ufsparams_config.configurator.configuration.managed}
name: cfga-cm-configurator-server-{{
vars['distrib.release.version'] }} data:   configurator-
module--${jenkins_env.fp_artifact_version}.yml: |
spring:       application:          name: configurator-server
```

```
profiles:         active:
${configurator.spring.active.profiles}       flyway:
enabled: false       thymeleaf.cache: false       servlet:
multipart:           max-file-size: 20MB           max-
request-size: 20MB       autoconfigure:         exclude:
ru.sbrf.ufs.platform.audit.AuditAutoConfiguration       jpa:
hibernate:           naming:             implicit-strategy:
org.hibernate.boot.model.naming.ImplicitNamingStrategyLegacy
JpaImpl             physical-strategy:
org.hibernate.boot.model.naming.PhysicalNamingStrategyStanda
rdImpl     configurator:       protectedRules:
minLength:
${ufsparams_config.configurator.core.protectedRules.minLengt
h}         exclusions:
${ufsparams_config.configurator.core.protectedRules.exclusio
ns}       ui:         roles:           group-levels1-plus1-
enable: true           group-levels1-plus2-enable: true
session-timeout-minutes: 30         active-directory-auth:
false       core:         rest-enabled:
${ufsparams_config.configurator.core.rest-enabled}
db:           url:
${ufsparams_config.configurator.core.db.url}         kafka:
bootstrap-servers:
${ufsparams_config.configurator.core.kafka.bootstrap.servers
}           security-protocol:
${ufsparams_config.configurator.core.kafka.security.protocol
}           ssl-keystore-location:
${ufsparams_config.configurator.core.kafka.configurator.ssl-
keystore-location}           ssl-truststore-location:
${ufsparams_config.configurator.core.kafka.configurator.ssl-
truststore-location}           topic-name:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.topic-name}           ssl-enabled-protocols:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-enabled-protocols}           ssl-protocol:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-protocol}           ssl-keystore-type:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-keystore-type}           ssl-truststore-type:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-truststore-type}           ssl-endpoint-
identification-algorithm:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-endpoint-identification-algorithm}     server:
port: 8080       servlet:         context-path:
/configurator-ui       max-http-header-size:
${ufsparams_config.configurator.server.max-http-header-size}
ssl:         enabled: false     audit2:       url:
${configurator.core.audit2.baseUrl}     pacman:
adapters:         monitoring:           unimon:
```

```
${ufsparams_config.pacman.adapters.monitoring.unimon}
stub: ${ufsparams_config.pacman.adapters.monitoring.stub}
audit:           selector:
${ufsparams_config.pacman.adapters.audit.selector}
timeout: ${ufsparams_config.pacman.adapters.audit.timeout}
group: ${configurator.core.audit.group}
authorization:           pprb:             enable:
${ufsparams_config.pacman.adapters.authorization.pprb.enable
}             jwt:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.jwt.en
able}             sudir:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.sudir.
enable}             basic:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.basic.
enable}             tenants: ${ufs.tenant.code}
ufs:             baseUrl:
${ufsparams_config.authorization.ufs.baseUrl}/ufs-security
enable:
${ufsparams_config.pacman.adapters.authorization.ufs.enable}
sudir:               enable:
${ufsparams_config.pacman.adapters.authorization.ufs.sudir.e
nable}             iam:               enable:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ena
ble}               jws_token_header:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.jws
_token_header}               roles_path:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.rol
es_path}               publickey_locations:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.pub
lickey_locations}               audience:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.aud
ience}               ssl_required:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_required}               ssl_truststore:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_truststore}               ssl_truststore_password:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_truststore_password}               poolsize:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.poo
lsize}               cache_updatetime:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.cac
he_updatetime}               devmode:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.dev
mode}     ru:       sbt:         pacman:           resource:
name: ${ufs.tenant.code}     clipas:       spas-server-url:
${configurator.core.clipas.spas-server-url}     ufs:
logger:         client:           mode: LOCAL
security:         client:           mode:
${configurator.ufs.security.client.mode}           local-
file-privileges-path: /config/security-permissions.json
```

```
local-file-filters-path: /config/security-filters.json
local-files-attributes-path: /config/security-
attributes.json           local-file-user-info-path:
/config/security-user-info.json           local-file-tenant-
code-path: /config/security-tenants.json       platform:
environment:           product:             subsystem:
${configurator.platform.environment.product.subsystem}
channel:
${ufsparams_config.platform.environment.product.channel}
version: {{ vars['distrib.release.version'] }}
api-version:
${configurator.platform.environment.product.api-version}
release:
${ufsparams_config.platform.environment.product.release}
distrib-version: ${jenkins_env.fp_artifact_version}
deploymentUnit:
${configurator.platform.environment.product.deploymentUnit}
config:           deployment-unit:
${configurator.platform.environment.product.deploymentUnit}
monitoring:         client:           channel:
${ufsparams_config.ufs.monitoring.client.channel}
subsystem: ${configurator.ufs.monitoring.client.subsystem}
title: ${configurator.ufs.monitoring.client.title}
tenant:         code: ${ufs.tenant.code}     platform:
http:         timeout: 300000     logging:       file:
path: /fluent-bit/etc/logs       config: /config-
log/logback-spring.xml       level:         org:
springframework:             boot:
autoconfigure:                 security:
servlet: ERROR     management:       server:         port:
${configurator.management.server.port}       endpoints:
web:           exposure:             include: "health,
prometheus"       endpoint:         health:           group:
readiness:               include: "*"             liveness:
include: "*"
```

5. ConfigMap `configurator-rest.${distrib.release.version}` с параметрами для Rest-сервиса компонента «Управление конфигурациями»:

```
apiVersion: v1 kind: ConfigMap metadata:   annotations:
pacman.sbt.ru/id:
${ufsparams_config.configurator_rest.configuration.id}
pacman.sbt.ru/rn: ${ufs.tenant.code}
pacman.sbt.ru/selector:
${ufsparams_config.configurator_rest.configuration.selector}
reloader.stakater.com/match:
${ufsparams_config.configurator_rest.configuration.reloader.
match}   labels:     pacman.sbt.ru/managed:
```

```
${ufsparams_config.configurator_rest.configuration.managed}
name: cfga-cm-configurator-rest-{{
vars['distrib.release.version'] }} data:   configurator-
rest--${jenkins_env.fp_artifact_version}.yml: |     spring:
application:         name: configurator-rest       profiles:
active: ${configurator_rest.spring.active.profiles}
autoconfigure:         exclude:
ru.sbrf.ufs.platform.audit.AuditAutoConfiguration
server:       servlet:         context-path: /configurator-
rest       max-http-header-size:
${configurator_rest.server.max-http-header-size}       port:
8443       ssl:         client-auth: need         enabled:
true         key-store: /ssl-certs/key-store.p12
protocol: TLS         trust-store: /ssl-certs/trustStore.jks
audit2:       url: ${configurator_rest.core.audit2.baseUrl}
pacman:       adapters:         monitoring:
unimon:
${ufsparams_config.pacman.adapters.monitoring.unimon}
stub: ${ufsparams_config.pacman.adapters.monitoring.stub}
audit:           selector:
${ufsparams_config.pacman.adapters.audit.selector}
timeout: ${ufsparams_config.pacman.adapters.audit.timeout}
group: ${configurator_rest.core.audit.group}     ru:
sbt:         pacman:           resource:             name:
${ufs.tenant.code}     ufs:       platform:
environment:           product:             subsystem:
${configurator_rest.platform.environment.product.subsystem}
channel:
${ufsparams_config.platform.environment.product.channel}
version: {{ vars['distrib.release.version'] }}
api-version:
${configurator_rest.platform.environment.product.api-
version}             release:
${ufsparams_config.platform.environment.product.release}
distrib-version: ${jenkins_env.fp_artifact_version}
deploymentUnit:
${configurator_rest.platform.environment.product.deploymentU
nit}         config:           deployment-unit:
${configurator_rest.platform.config.deployment-unit}
monitoring:         client:           channel:
${ufsparams_config.ufs.monitoring.client.channel}
subsystem:
${configurator_rest.ufs.monitoring.client.subsystem}
title: ${configurator_rest.ufs.monitoring.client.title}
config-store:       jdbc_url:
${configurator_rest.core.db.url}     configurator-rest:
db_pool:         acquire_increment: 1         max_idle_time:
10         checkout_timeout: 120000
initial_pool_size: 1         idle_connection_test_period: 60
min_pool_size: 0         max_statements: 50         retry-
```

```
delay: 2000         retry-attempts: 10
max_pool_size: 2         unreturnedConnectionTimeout: 180
logging:       file:         path: /fluent-bit/etc/logs
config: /config-log/logback-spring.xml     management:
server:         port:
${configurator_rest.management.server.port}       endpoints:
web:           exposure:             include: "health,
prometheus"       endpoint:         health:           group:
readiness:               include: "*"             liveness:
include: "*"
```

[Файлы с параметрами Приложений для импорта в БД компонета «Управление параметрами»]

1. Параметры для АРМ компонента «Управление параметрами»:

Расположение файлов в дистрибутиве:

```
├── conf │   └── data │       ├── sup2 │       │   ├──
definitions │       │   │   ├── PACMAN_ADMIN.json │       │
├── parameters │       │   │   ├── PACMAN_ADMIN.conf
```

Конфигурационные файлы:

- config.json:

```
{   "version": "1.0",   "parameters": [     {       "name":
"ufsparams.auth.cache.ttl",       "description": "TTL кеша
авторизации.",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.ufsparams.auth.cache.ttl}"           ]
}       ]     },     {       "name":
"parameter.export.http.timeout",       "description":
"Платформенный таймаут на запрос экспорта конфигурации для
сравнения (/parameter/data/export)",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.parameter.export.http.timeout}"           ]
}       ]     },     {       "name":
"configuration.compare.http.timeout",       "description":
"Платформенный таймаут на запрос сравнения конфигурации
(/configuration/compare)",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "LONG",       "bundle": [         {
```

```
"path": [],           "values": [
"${pacman_admin.configuration.compare.http.timeout}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.bootstrap.servers
",       "description": "Адреса серверов подключения к Kafka
при отправке сообщения репликации",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.kafka.bootstrap.servers}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.client.id",
"description": "Идентификатор клиента Kafka при подключении
для отправки сообщения репликации",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.kafka.client.id}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.acks",
"description": "Тип подтверждений при подключении к Kafka
для отправки сообщений репликации",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.kafka.acks}"           ]
}       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.batch.size",
"description": "Количество сообщений в пакете при
подключении к Kafka для отправки сообщений репликации",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.kafka.batch.size}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.topic.name",
"description": "Имя топика Kafka для подключения при
отправке сообщений репликации",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.kafka.topic.name}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.additional.parame
ters",       "description": "Дополнительные параметры, в
формате ключ значение при подключении к Kafka для отправки
сообщений репликации",       "isList": true,       "roles":
[         "ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
```

```
"path": [],           "values": []         }       ]     },
{       "name":
"ufsparams.replication.intseg.destination.kafka.bootstrap.se
rvers",       "description": "Адреса серверов подключения к
Kafka для получения сообщения репликации",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.destination.kafka.bootstrap.serv
ers}"           ]         }       ]     },     {
"name":
"ufsparams.replication.intseg.destination.kafka.client.id",
"description": "Идентификатор клиента для подключения к
Kafka при получении сообщения репликации",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.destination.kafka.client.id}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.destination.kafka.topic.name",
"description": "Имя топика Kafka для подключения при
получении сообщения репликации",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.destination.kafka.topic.name}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.destination.kafka.additional.p
arameters",       "description": "Дополнительные параметры,
в формате ключ значение при подключении к Kafka для
получения сообщения репликации",       "isList": true,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": []         }       ]     },
{       "name":
"ufsparams.replication.intseg.source.enabled",
"description": "Включение/Отключение функционала отправки
сообщения репликации в Kafka (применяется при старте
приложения)",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.enabled}"           ]
}       ]     },     {       "name":
"ufsparams.replication.intseg.source.skip",
"description": "Включение/Отключение режима пропуска
отправки сообщений репликации (применяется в рантайме)",
"isList": false,       "roles": [
```

```
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.skip}"           ]
}       ]     },     {       "name":
"ufsparams.replication.intseg.destination.enabled",
"description": "Включение/Отключение функционала получения
сообщения репликации из Kafka (применяется при старте
приложения)",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.destination.enabled}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.destination.skip",
"description": "Включение/Отключение режима пропуска приемки
сообщений репликации (применяется в рантайме)",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.destination.skip}"           ]
}       ]     },     {       "name":
"ufsparams.replication.source.ssl.enabled",
"description": "Включение/Отключение функционала отправки
сообщения репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.ssl.enabled}"           ]
}       ]     },     {       "name":
"ufsparams.replication.source.kafka.ssl.store.name",
"description": "Имя хранилища для отправки сообщения
репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.kafka.ssl.store.name}"
]         }       ]     },     {       "name":
"ufsparams.replication.source.kafka.ssl.truststore.name",
"description": "Имя доверенного хранилища для отправки
сообщения репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.kafka.ssl.truststore.name
}"           ]         }       ]     },     {       "name":
"ufsparams.replication.source.kafka.ssl.store.password.varia
ble.name",       "description": "Имя переменной среды с
указанием хранилища паролей для отправки сообщения
```

```
репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.kafka.ssl.store.password.
variable.name}"           ]         }       ]     },     {
"name":
"ufsparams.replication.source.kafka.ssl.truststore.password.
variable.name",       "description": "Имя переменной среды с
указанием доверенного хранилища паролей для отправки
сообщения репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.source.kafka.ssl.truststore.pass
word.variable.name}"           ]         }       ]     },
{       "name":
"ufsparams.replication.destination.ssl.enabled",
"description": "Включение/Отключение функционала получения
сообщения репликации из Kafka по SSL",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.destination.ssl.enabled}"
]         }       ]     },     {       "name":
"ufsparams.replication.destination.kafka.ssl.store.name",
"description": "Имя хранилища для получения сообщения
репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.destination.kafka.ssl.store.name
}"           ]         }       ]     },     {       "name":
"ufsparams.replication.destination.kafka.ssl.truststore.name
",       "description": "Имя доверенного хранилища для
получения сообщения репликации в Kafka по SSL",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.destination.kafka.ssl.truststore
.name}"           ]         }       ]     },     {
"name":
"ufsparams.replication.destination.kafka.ssl.store.password.
variable.name",       "description": "Имя переменной среды с
указанием хранилища паролей для получения сообщения
репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
```

```
"path": [],           "values": [
"${pacman_admin.replication.destination.kafka.ssl.store.pass
word.variable.name}"           ]         }       ]     },
{       "name":
"ufsparams.replication.destination.kafka.ssl.truststore.pass
word.variable.name",       "description": "Имя переменной
среды с указанием доверенного хранилища паролей для
получения сообщения репликации в Kafka по SSL",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.replication.destination.kafka.ssl.truststore
.password.variable.name}"           ]         }       ]
},     {       "name": "ufsparams.mass.operations.enabled",
"description": "Включение/отключение массовых операций с
параметрами",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_admin.ufsparams.mass.operations.enabled}"
]         }       ]     },     {       "name":
"ufsparams.delete.configuration.enabled",
"description": "Включение/отключение возможности удаления
конфигураций",       "isList": false,       "type":
"BOOLEAN",       "bundle": [         {           "path": [],
"values": [
"${pacman_admin.ufsparams.delete.configuration.enabled}"
]         }       ]     }   ] }
```

- config.conf:

```
pacman_admin.replication.source.kafka.bootstrap.servers=${gl
obal.platform.kafka.bootstrap.servers}
pacman_admin.replication.source.kafka.client.id=PACMAN_ADMIN
pacman_admin.replication.source.kafka.acks=all
pacman_admin.replication.source.kafka.batch.size=1
pacman_admin.replication.source.kafka.topic.name=sup-
replication pacman_admin.replication.source.enabled=false
pacman_admin.replication.source.skip=false
pacman_admin.replication.source.ssl.enabled=false
pacman_admin.replication.source.kafka.ssl.store.name=mq.jks
pacman_admin.replication.source.kafka.ssl.truststore.name=mq
.jks
pacman_admin.replication.source.kafka.ssl.store.password.var
iable.name=ssl.ose.keyStore.mq.password
pacman_admin.replication.source.kafka.ssl.truststore.passwor
d.variable.name=ssl.ose.keyStore.mq.password
pacman_admin.replication.destination.kafka.bootstrap.servers
```

```
=${global.platform.kafka.bootstrap.servers}
pacman_admin.replication.destination.kafka.client.id=PACMAN_
ADMIN
pacman_admin.replication.destination.kafka.topic.name=sup-
replication
pacman_admin.replication.destination.enabled=false
pacman_admin.replication.destination.skip=false
pacman_admin.replication.destination.ssl.enabled=false
pacman_admin.replication.destination.kafka.ssl.store.name=mq
.jks
pacman_admin.replication.destination.kafka.ssl.truststore.na
me=mq.jks
pacman_admin.replication.destination.kafka.ssl.store.passwor
d.variable.name=ssl.ose.keyStore.mq.password
pacman_admin.replication.destination.kafka.ssl.truststore.pa
ssword.variable.name=ssl.ose.keyStore.mq.password
pacman_admin.parameter.export.http.timeout=-1
pacman_admin.configuration.compare.http.timeout=-1
pacman_admin.ufsparams.auth.cache.ttl=10
pacman_admin.ufsparams.mass.operations.enabled=true
pacman_admin.ufsparams.delete.configuration.enabled=false
```

2. Параметры для Сервиса Импорта конфигураций компонента «Управление параметрами»:

Расположение файлов в дистрибутиве:

```
├── conf │   └── data │       ├── sup2 │       │   ├──
definitions │       │   │   ├── PACMAN_IMPORT.json  │
│   ├── parameters │       │   │   ├── PACMAN_IMPORT.conf
```

Конфигурационные файлы:

- config.json:

```
{   "version": "1.0",   "parameters": [     {       "name":
"platform.http.timeout",       "description": "Таймаут
обработки HTTP запроса",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.platform.http.timeout}"           ]
}       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.bootstrap.servers
",       "description": "Адреса серверов подключения к Kafka
при отправке сообщения репликации",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
```

```
"path": [],           "values": [
"${pacman_import.replication.source.kafka.bootstrap.servers}
"          ]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.client.id",
"description": "Идентификатор клиента Kafka при подключении
для отправки сообщения репликации",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.client.id}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.acks",
"description": "Тип подтверждений при подключении к Kafka
для отправки сообщений репликации",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.acks}"           ]
}       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.batch.size",
"description": "Количество сообщений в пакете при
подключении к Kafka для отправки сообщений репликации",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.batch.size}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.topic.name",
"description": "Тип подтверждений при подключении к Kafka
для отправки сообщений репликации",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.topic.name}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.additional.parame
ters",       "description": "Дополнительные параметры, в
формате ключ значение при подключении к Kafka для отправки
сообщений репликации",       "isList": true,       "roles":
[         "ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": []         }       ]     },
{       "name":
"ufsparams.replication.intseg.source.enabled",
"description": "Включение/Отключение функционала отправки
сообщения репликации в Kafka (применяется при старте
приложения)",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
```

```
"path": [],           "values": [
"${pacman_import.replication.source.enabled}"           ]
}       ]     },     {       "name":
"ufsparams.replication.intseg.source.skip",
"description": "Включение/Отключение режима пропуска
отправки сообщений репликации (применяется в рантайме)",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.skip}"           ]
}       ]     },     {       "name":
"ufsparams.replication.intseg.source.ssl.enabled",
"description": "Включение/Отключение функционала отправки
сообщения репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.ssl.enabled}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.ssl.store.name",
"description": "Имя хранилища для отправки сообщения
репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.ssl.store.name}"
]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.ssl.truststore.na
me",       "description": "Имя доверенного хранилища для
отправки сообщения репликации в Kafka по SSL",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.ssl.truststore.nam
e}"           ]         }       ]     },     {       "name":
"ufsparams.replication.intseg.source.kafka.ssl.store.passwor
d.variable.name",       "description": "Имя переменной среды
с указанием хранилища паролей для отправки сообщения
репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.ssl.store.password
.variable.name}"           ]         }       ]     },     {
"name":
"ufsparams.replication.intseg.source.kafka.ssl.truststore.pa
ssword.variable.name",       "description": "Имя переменной
среды с указанием доверенного хранилища паролей для отправки
```

```
сообщения репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.ssl.truststore.pas
sword.variable.name}"           ]         }       ]     },
{       "name": "ufsparams.replication.source.ssl.enabled",
"description": "Включение/Отключение функционала отправки
сообщения репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.ssl.enabled}"
]         }       ]     },     {       "name":
"ufsparams.replication.source.kafka.ssl.store.name",
"description": "Имя хранилища для отправки сообщения
репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.ssl.store.name}"
]         }       ]     },     {       "name":
"ufsparams.replication.source.kafka.ssl.truststore.name",
"description": "Имя доверенного хранилища для отправки
сообщения репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.ssl.truststore.nam
e}"           ]         }       ]     },     {       "name":
"ufsparams.replication.source.kafka.ssl.store.password.varia
ble.name",       "description": "Имя переменной среды с
указанием хранилища паролей для отправки сообщения
репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.ssl.store.password
.variable.name}"           ]         }       ]     },     {
"name":
"ufsparams.replication.source.kafka.ssl.truststore.password.
variable.name",       "description": "Имя переменной среды с
указанием доверенного хранилища паролей для отправки
сообщения репликации в Kafka по SSL",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${pacman_import.replication.source.kafka.ssl.truststore.pas
sword.variable.name}"           ]         }       ]     }
] }
```

- config.conf:

```
pacman_import.upload.tempDir=/var/tmp/import
pacman_import.platform.http.timeout=300000
pacman_import.replication.source.kafka.bootstrap.servers=${g
lobal.platform.kafka.bootstrap.servers}
pacman_import.replication.source.kafka.client.id=PACMAN_IMPO
RT pacman_import.replication.source.kafka.acks=all
pacman_import.replication.source.kafka.batch.size=1
pacman_import.replication.source.kafka.topic.name=sup-
replication pacman_import.replication.source.enabled=false
pacman_import.replication.source.skip=false
pacman_import.replication.source.ssl.enabled=false
pacman_import.replication.source.kafka.ssl.store.name=mq.jks
pacman_import.replication.source.kafka.ssl.truststore.name=m
q.jks
pacman_import.replication.source.kafka.ssl.store.password.va
riable.name=ssl.ose.keyStore.mq.password
pacman_import.replication.source.kafka.ssl.truststore.passwo
rd.variable.name=ssl.ose.keyStore.mq.password
```

3. Параметры для Sidecar-Агента компонента «Управление параметрами»:

Конфигурационные файлы:

- config.json:

```
{   "version": "1.0",   "parameters": [     {       "name":
"ufsparams.agent.baseUrl.openshift",       "description":
"Base url for IAG-mm",       "isList": false,       "roles":
[         "ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_baseUrl_insecure}"           ]         }
]     },     {       "name": "ufsparams.agent.baseUrl",
"description": "Base url for IAG-mm (например, для WAS)",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_baseUrl_secure}"           ]         }
]     },     {       "name": "ufsparams.agent.delay",
"description": "Interval between configuration refresh
checks",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_delay}"           ]         }       ]
```

```
},     {       "name": "healthcheck.agent.url",
"description": "Base url for Health check agent",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_healthcheck_agent_url}"           ]         }
]     },     {       "name":
"healthcheck.registration.attempt.period",
"description": "Timeout to next attempt to registration in
HealthCheck Agent",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_healthcheck_registration_attempt_period}"
]         }       ]     },     {       "name":
"ufsparams.agent.max.file.size",       "description":
"Максимальный размер файла конфигурации",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_max_file_size}"           ]         }
]     },     {       "name":
"ufsparams.agent.min.file.size",       "description":
"Минимальный размер файла конфигурации",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_min_file_size}"           ]         }
]     },     {       "name":
"ufsparams.agent.ignore.file.pattern",       "description":
"Паттерн, по которому Агент не будет отслеживать файл для
обновления из БД",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_ignore_file_pattern}"           ]
}       ]     },     {       "name":
"ufsparams.agent.httpclient.circuitBreaker.enable",
"description": "Признак работы механизма circuit breaker.
Если указано false, то http вызов происходит без
декорирования механизмом circuit breaker.",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_enable}"
]         }       ]     },     {       "name":
```

```
"ufsparams.agent.httpclient.circuitBreaker.failureRate.thres
hold",       "description": "Процент отказов в потоке
запросов. После превышения данного процента поток будет
прерван (переход в режим OPEN).",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_failureRate_thr
eshold}"           ]         }       ]     },     {
"name":
"ufsparams.agent.httpclient.circuitBreaker.openState.duratio
n",       "description": "Время ожидания перед переходом от
режима работы OPEN к HALF_OPEN",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_openState_durat
ion}"           ]         }       ]     },     {
"name":
"ufsparams.agent.httpclient.circuitBreaker.halfOpen.requestB
uffer",       "description": "Размер буфера запросов в
режиме HALF_OPEN. После заполнения буфера высчитывается
процент отказов или процент медленных запросов для принятия
решения о повторном прерывании потока (переход в режим OPEN)
или о переходе в обычный режим работы потока запросов
(переход в режим CLOSED). По умолчанию значение для данной
настройки \"0\" и в этом случае реальный размер буфера
запрос в режиме HALF_OPEN равен размеру буфера запросов в
режиме CLOSED.",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_halfOpen_reques
tBuffer}"           ]         }       ]     },     {
"name":
"ufsparams.agent.httpclient.circuitBreaker.minimumCalls.size
",       "description": "Минимальное количество вызовов,
которое требуется (за период скользящего окна), прежде чем
CircuitBreaker сможет вычислить частоту ошибок или медленную
скорость вызова. По умолчанию значение для данной настройки
\"0\" и в этом случае Минимальное количество вызовов равно
размеру буфера запросов в режиме CLOSED.",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_minimumCalls_si
ze}"           ]         }       ]     },     {
"name": "ufs.logger.kafka.topic",       "description": "",
"isList": false,       "roles": [
```

```
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_logger_kafka_topic}"           ]
}       ]     },     {       "name":
"ufs.logger.kafka.bootstrap.servers",       "description":
"",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_logger_kafka_bootstrap_servers}"
]         }       ]     },     {       "name":
"ufs.logger.parameters.url",       "description": "",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_logger_baseUrl}"           ]         }
]     },     {       "name":
"ufs.logger.kafka.security.protocol",       "description":
"",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_logger_kafka_security_protocol}"
]         }       ]     },     {       "name":
"logger.ufs.client_module.kafka.ssl.store_name",
"description": "",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_logger_kafka_ssl_storeName}"           ]
}       ]     }   ] }
```

- config.conf:

```
ufsparams_agent_baseUrl_insecure=${global.ose.params.baseUrl
.stringToBind}paramsv2/
ufsparams_agent_baseUrl_secure=${global.params.baseUrl.strin
gToBind}paramsv2/ ufsparams_agent_delay=10000
ufsparams_healthcheck_agent_url=${global.healthcheck.agent.u
rl}/ ufsparams_healthcheck_registration_attempt_period=30000
ufsparams_agent_max_file_size=104857600
ufsparams_agent_min_file_size=50
ufsparams_agent_ignore_file_pattern=glob:**sup2_common.json
ufsparams_agent_httpclient_circuitBreaker_enable=false
ufsparams_agent_httpclient_circuitBreaker_failureRate_thresh
old=50
ufsparams_agent_httpclient_circuitBreaker_openState_duration
```

```
=60000
ufsparams_agent_httpclient_circuitBreaker_halfOpen_requestBu
ffer=0
ufsparams_agent_httpclient_circuitBreaker_minimumCalls_size=
10
ufsparams_agent_logger_baseUrl=!${global.platform.ufs.baseur
l.logger}/ufs-logger-parameters/parameters
ufsparams_agent_logger_kafka_topic=${global.platform.logger.
kafka.topic}
ufsparams_agent_logger_kafka_bootstrap_servers=${global.plat
form.logger.kafka.bootstrap.servers}
ufsparams_agent_logger_kafka_ssl_storeName=${global.wmqcf.Lo
ggerCF.ssl.config}
ufsparams_agent_logger_kafka_security_protocol=${global.plat
form.logger.kafka.security.protocol}
```

4. Параметры Клиентского Модуля компонента «Управление параметрами»:

Файлы располагаются в дистрибутиве самой библиотеки Клиентского Модуля компонента «Управление параметрами».

При использовании Maven-plugin данные параметры записываются в общий файл параметров Приложения-потребителя.

Конфигурационные файлы:

- config.json:

```
{   "version": "1.0",   "parameters": [     {       "name":
"ufs.tenant.code",       "description": "Код тенанта",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_module_tenant_code}"           ]         }
]     },     {       "name":
"ufsparams.module.monitoring.enabled",       "description":
"Включение отправки метрики мониторинга при запросе
параметра. (true - метрики отправляются, false - метрики не
отправляются)",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_module_monitoring_enabled}"           ]
}       ]     }   ] }
```

- config.conf:

```
ufsparams_module_tenant_code={{ TENANT_CODE }}
ufsparams_module_monitoring_enabled=false
```

5. Параметры Клиентского Модуля Агента компонента «Управление параметрами»:

Файлы располагаются в дистрибутиве самой библиотеки Модуля Агента конфигурации.

При использовании Maven-plugin данные параметры записываются в общий файл параметров Приложения-потребителя.

Конфигурационные файлы:

- config.json:

```
{   "version": "1.0",   "parameters": [     {       "name":
"ufsparams.agent.baseUrl.openshift",       "description":
"Base url for IAG-mm",       "isList": false,       "roles":
[         "ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_baseUrl_insecure}"           ]         }
]     },     {       "name": "ufsparams.agent.baseUrl",
"description": "Base url for IAG-mm (например, для WAS)",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_baseUrl_secure}"           ]         }
]     },     {       "name": "ufsparams.agent.delay",
"description": "Interval between configuration refresh
checks",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_delay}"           ]         }       ]
},     {       "name": "ufsparams.agent.max.file.size",
"description": "Максимальный размер файла конфигурации",
"isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_max_file_size}"           ]         }
]     },     {       "name":
"ufsparams.agent.min.file.size",       "description":
"Минимальный размер файла конфигурации",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
```

```
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_min_file_size}"           ]         }
]     },     {       "name":
"ufsparams.agent.ignore.file.pattern",       "description":
"Паттерн, по которому Агент не будет отслеживать файл для
обновления из БД",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "STRING",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_ignore_file_pattern}"           ]
}       ]     },     {       "name":
"ufsparams.agent.httpclient.circuitBreaker.enable",
"description": "Признак работы механизма circuit breaker.
Если указано false, то http вызов происходит без
декорирования механизмом circuit breaker.",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "BOOLEAN",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_enable}"
]         }       ]     },     {       "name":
"ufsparams.agent.httpclient.circuitBreaker.failureRate.thres
hold",       "description": "Процент отказов в потоке
запросов. После превышения данного процента поток будет
прерван (переход в режим OPEN).",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_failureRate_thr
eshold}"           ]         }       ]     },     {
"name":
"ufsparams.agent.httpclient.circuitBreaker.openState.duratio
n",       "description": "Время ожидания перед переходом от
режима работы OPEN к HALF_OPEN",       "isList": false,
"roles": [         "ManageParameters.ParameterRole.System"
],       "type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_openState_durat
ion}"           ]         }       ]     },     {
"name":
"ufsparams.agent.httpclient.circuitBreaker.halfOpen.requestB
uffer",       "description": "Размер буфера запросов в
режиме HALF_OPEN. После заполнения буфера высчитывается
процент отказов или процент медленных запросов для принятия
решения о повторном прерывании потока (переход в режим OPEN)
или о переходе в обычный режим работы потока запросов
(переход в режим CLOSED). По умолчанию значение для данной
настройки \"0\" и в этом случае реальный размер буфера
запрос в режиме HALF_OPEN равен размеру буфера запросов в
```

```
режиме CLOSED.",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_halfOpen_reques
tBuffer}"           ]         }       ]     },     {
"name":
"ufsparams.agent.httpclient.circuitBreaker.minimumCalls.size
",       "description": "Минимальное количество вызовов,
которое требуется (в режиме CLOSED), прежде чем
CircuitBreaker сможет вычислить частоту ошибок. По умолчанию
значение для данной настройки \"10\".",       "isList":
false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_httpclient_circuitBreaker_minimumCalls_si
ze}"           ]         }       ]     },     {
"name": "ufsparams.agent.startup.synchronization.timeout",
"description": "",       "isList": false,       "roles": [
"ManageParameters.ParameterRole.System"       ],
"type": "LONG",       "bundle": [         {
"path": [],           "values": [
"${ufsparams_agent_startup_synchronization_timeout}"
]         }       ]     }   ] }
```

- config.conf:

```
ufsparams_agent_baseUrl_insecure=${global.ose.params.baseUrl
.stringToBind}paramsv2/
ufsparams_agent_baseUrl_secure=${global.params.baseUrl.strin
gToBind}paramsv2/ ufsparams_agent_delay=10000
ufsparams_agent_max_file_size=104857600
ufsparams_agent_min_file_size=50
ufsparams_agent_ignore_file_pattern=glob:**sup2_common.json
ufsparams_agent_httpclient_circuitBreaker_enable=false
ufsparams_agent_httpclient_circuitBreaker_failureRate_thresh
old=50
ufsparams_agent_httpclient_circuitBreaker_openState_duration
=60000
ufsparams_agent_httpclient_circuitBreaker_halfOpen_requestBu
ffer=0
ufsparams_agent_httpclient_circuitBreaker_minimumCalls_size=
10 ufsparams_agent_startup_synchronization_timeout=3000
```

Установка PL-компонентов на группу серверов NGINX

- `./pl/configurator-ui-static.zip` – презентационный слой приложения для доступа к АРМ компонента «Управление конфигурациями» через удаленный компонент «Стартовый менеджер» `{{ nginx_iag_home_path }}/ufs-sm`.
- `./pl/ufs-params-manager-pl.zip` – презентационный слой приложения для доступа к АРМ «Управление параметрами» через удаленный компонент «Стартовый менеджер» `{{ nginx_iag_home_path }}/ufs-sm`;
- `./pl/ufs-params-manager-pl-standalone.zip` – презентационный слой standalone приложения для доступа АРМ «Управление параметрами» по прямой ссылке `{{nginx_iag_home_path }}/ufs-config-manager/pacman/sup`.

Установка ingress/egress

Все входящие и исходящие взаимодействия с установленным приложениями должны происходить через Istio (Synapse):

- Egress – управляет исходящими соединениями;
- Ingress – управляет входящими соединениями.

Перед установкой самого приложения необходимо установить Egress/Ingress. Все дальнейшие обращения к Приложению будут происходить через Route Ingress'a, который в свою очередь будет маршрутизировать запрос на сервис Приложения.

Конфигурационные файлы установки Egress/Ingress:

```
├── conf │   └── k8s │       └── base │       │   ├── istio
# Каталог с конфигурационными элементами istio для работы в
среде Kubernetes │       │   │   ├── config
# Каталог, содержащий настройки для компонентов Istio
(egress/ingress) │       │   │   │   ├── egress
# Настройки Gateway, отвечающий за исходящие соединения │
│   │   │   │   ├── dsr-database.yaml
# Файл с настройками DestinationRule, отвечает за политики
для конечного endpoint'а подключения к БД │       │   │   │
│   ├── dsr-egress-mtls-audit.yaml                # Файл с
настройками DestinationRule, отвечает за политики для
конечного endpoint'а сервиса аудита │       │   │   │   │
├── dsr-egress-mtls-iag.yaml                  # Файл с
настройками DestinationRule, отвечает за политики для
конечного endpoint'а │       │   │   │   │   ├── dsr-egress-
mtls-sgw.yaml             # Файл с настройками
DestinationRule, отвечает за политики для конечного
endpoint'а │       │   │   │   │   ├── dsr-egress-mtls-
spas.yaml                 # Файл с настройками
DestinationRule, отвечает за политики для конечного
```

```
endpoint'a сервиса SPAS │       │   │   │   │   ├── dsr-
kafka-adapter.yaml                    # Файл с настройками
DestinationRule, отвечает за политики для VirtualService для
Kafka │       │   │   │   │   ├── gw-database.yaml
# Файл с настройками Gateway, открывает порт на самом
Egress/Ingress, для приема трафика │       │   │   │   │
├── gw-egress-mtls.yaml        # Файл с настройками Gateway,
открывает порт на самом Egress/Ingress, для приема трафика │
│   │   │   │   ├── gw-kafka-adapter.yaml      # Файл с
настройками Gateway, открывает порт на самом Egress/Ingress,
для приема трафика │       │   │   │   │   ├── istio-basic-
cm.yaml        # Файл с настройками Istio │       │   │   │
│   ├── ott-egress-auth-filter.yaml   # Файл с настройками
│       │   │   │   │   ├── ott-egress-cm.yaml            #
Файл с настройками  │       │   │   │   │   ├── ott-jour-
mon-cm.yaml          # Файл с настройками  │       │   │   │
│   ├── se-database.yaml                                 # Файл
с настройками ServiceEntry, содержит информацию с
разрешенными хост:порт для подключения к БД │       │   │
│   │   ├── se-egress-mtls-audit.yaml                    #
Файл с настройками ServiceEntry, содержит информацию с
разрешенными для исходящих соединений по mtls хост:порт для
аудита │       │   │   │   │   ├── se-egress-mtls-spas.yaml
# Файл с настройками ServiceEntry, содержит информацию с
разрешенными для исходящих соединений по mtls хост:порт для
SPAS │       │   │   │   │   ├── se-egress-mtls.yaml
# Файл с настройками ServiceEntry, содержит информацию с
разрешенными хост:порт для подключения к БД │       │   │
│   │   ├── se-egress-tcp.yaml                           #
Файл с настройками ServiceEntry, содержит информацию с
разрешенными для исходящих соединений по tcp хост:порт │
│   │   │   │   ├── se-kafka.yaml
# Файл с настройками ServiceEntry, содержит информацию с
разрешенными хост:порт подключения к Kafka │       │   │   │
│   ├── vs-database.yaml                          # Файл с
настройками VirtualService для БД, содержит правила для
дальнейшей балансировки трафика, после принятия запроса
Gateway'ем │       │   │   │   │   ├── vs-egress-mtls-
audit.yaml               # Файл с настройками
VirtualService для аудита, содержит правила для дальнейшей
балансировки трафика, после принятия запроса Gateway'ем │
│   │   │   │   ├── vs-egress-mtls-iag.yaml
# Файл с настройками VirtualService  │       │   │   │   │
├── vs-egress-mtls-sgw.yaml                  # Файл с
настройками VirtualService  │       │   │   │   │   ├── vs-
egress-mtls-spas.yaml                # Файл с настройками
VirtualService для SPAS, содержит правила для дальнейшей
балансировки трафика, после принятия запроса Gateway'ем │
│   │   │   │   ├── vs-kafka-adapter.yaml
# Файл с настройками VirtualService для Kafka, содержит
```

```
правила для дальнейшей балансировки трафика, после принятия
запроса Gateway'ем |       |   |   |   └── vs-ott-
egress.yaml                        # Файл с настройками
VirtualService  |       |   |   |   └── ingress
# Gateway отвечающий за входящие соединения |       |   |
|       ├── gw-ingress-mtls.yaml |       |   |   |       ├──
gw-ingress.yaml   |       |   |   |       ├── ott-ingress-
auth-filter.yaml   |       |   |   |       ├── ott-ingress-
cm.yaml    |       |   |   |       ├── skip-ext-authz.yaml
|       |   |   |       ├── vs-config-storage-ingress-
mtls.yaml   |       |   |   └── deployments
# Настройки, необходимые для развертывания компонентов Istio
|       |   |   |   ├── egress                   # Каталог с
набором настроек для развертывания egress'a |       |   |
|   |   ├── dc-egress-mtls.yaml |       |   |   |   |   └──
svc-egress-mtls.yaml |       |   |   |   |   └── pdb-
egress.yaml     # Настройки для PodDisruptionBudget |
|   |   |   └── ingress                  # Каталог с набором
настроек для развертывания ingress |       |   |   |
├── dc-ingress-mtls.yaml |       |   |   |       ├── pdb-
ingress.yaml    # Настройки для PodDisruptionBudget |
|   |   |       ├── route-ingress-configurator-rest-
mtls.yaml |       |   |   |       ├── route-ingress-
mtls.yaml |       |   |   |       ├── route-ingress.yaml |
|   |   |       └── svc-ingress.yaml |       |   |   └──
secrets                    |       |   |       ├── ott-
certs.yaml                    # Файл хранения сертификата для
OTT |       |   |       ├── egress-ca-jks.yaml |       |   |
├── egress-jks.yaml |       |   |       ├── ingress-ca-
jks.yaml |       |   |       ├── ingress-jks.yaml |       |
|       ├── configuration-rest-ssl-certs.yaml |       |
configurator_server          # Директория с настройками
сборки контейнера и развертывания приложения компонента
«Управление конфигурациями». Имя директории соответствует
имени application из distrib.yaml |       |   ├── istio
# Дополнительные настройки istio для приложения компонента
«Управление конфигурациями» |       |   |   ├── config
|       |   |   |   ├── ingress   |       |   |   |   |
├── vs-configurator-ingress.yaml  # Файл описания
дополнительных политик маршрутизации трафика с gateway по
URL на нужный сервис  |       |   uko_ufsparams_bh_v2
# Директория с настройками сборки контейнера и развертывания
приложения сервиса АРМ «Управление параметрами» в
Kubernetes. Имя директории соответствует имени application
из distrib.yaml |       |   ├── istio                    #
Дополнительные настройки istio для приложения АРМ
«Управление параметрами» |       |   |   ├── config
|       |   |   |   ├── ingress   |       |   |   |   |
├── vs-ufsparams-bh-ingress.yaml         |       |
uko_ufsparams_import_v2      # Директория с настройками
```

```
сборки контейнера и развертывания приложения сервиса Импорта
«Управление параметрами» в Kubernetes. Имя директории
соответствует имени application из distrib.yaml │       │
├── istio                    # Дополнительные настройки
istio для приложения Сервис Импорта «Управление параметрами»
│       │   │   ├── config          │       │   │   │   ├──
ingress   │       │   │   │   │   ├── vs-ufsparams-import-
ingress.yaml   # Файл описания дополнительных политик
маршрутизации трафика с gateway по URL на нужный сервис │
│   uko_ufsparams_ms_v2          # Директория с настройками
сборки контейнера и развертывания приложения сервиса выдачи
параметров «Управление параметрами» в Kubernetes. Имя
директории соответствует имени application из distrib.yaml │
│   ├── istio                    # Дополнительные настройки
istio для приложения сервиса выдачи параметров «Управление
параметрами» │       │   │   ├── config    │       │   │   │
├── ingress   │       │   │   │   │   ├── vs-ufsparams-ms-
ingress.yaml    # Файл описания дополнительных политик
маршрутизации трафика с gateway по URL на нужный сервис
│       │   pacman_agent                # Директория с
настройками сборки контейнера и развертывания Агента
конфигурации. Имя директории соответствует имени application
из distrib.yaml │       │   ├── istio                    #
Дополнительные настройки istio для Агента конфигурации │
│   │   ├── config                │       │   │   │   ├──
egress   │       │   │   │   │   ├── dsr-egress-mtls-config-
storage.yaml │       │   │   │   │   ├── gw-egress-mtls-
config-storage.yaml  │       │   │   │   │   ├── se-api-
ose.yaml │       │   │   │   ├── se-egress-mtls-config-
storage.yaml │       │   │   │   │   ├── svc-egress-mtls-
config-storage.yaml │       │   │   │   │   ├── vs-egress-
mtls-config-storage.yaml │       │   pacman_ui
│       │   ├── istio                    │       │   │   ├──
config                │       │   │   │   ├── ingress   │
│   │   │   │   ├── vs-ingress-ui.yaml │       overrides │
└── openshift │           └── istio   # Каталог с
конфигурационными элементами istio для работы в среде
OpenShift │               └── deployments │
└── ingress │                       ├── route-ingress-
configurator-rest-mtls.yaml │                       ├──
route-ingress-mtls.yaml │                   ├── route-
ingress.yaml
```

Конфигурационные файлы для Fluent bit'ов

```
├── conf │   └── k8s │       └── base │           ├──
configmaps                      # Каталог для загрузки
не параметризуемых файлов │               ├── pacman-fluent-
```

```
bit.yaml            # Файл с настройками fluent bit    │
└── pacman-logback-spring.yaml         # Конфигурация
журналирования
```

Установка pod

Необходимо установить или обновить DeploymentConfigs:

Для компонента «Управление параметрами»:

- ufs-params-service-unver – микросервис доставки конфигурации;
- ufs-params-manager-unver – АРМ Администрирования;
- ufs-params-import-unver – сервис импорта;
- pacman-agent – Агент обновления конфигурации;
- pacman_ui – UI для АРМ Администрирования.

Для компонента «Управление конфигурациями»:

- configurator_server – АРМ Администрирования;
- configurator_rest – микросервис доставки конфигурации;
- pacman_agent – Агент обновления конфигурации;
- pacman_ui – UI для АРМ Администрирования.

Папка хранения конфигураций: `Ks8/base/<app_name>/dc.yaml`.

```
├── conf │   └── k8s │       └── base
# Каталог для установки компонентов в среде Kubernetes │
├── istio                        # Каталог с
конфигурационными элементами istio │           ├──
configurator_server          # Директория с настройками
сборки контнейнера и развертывания приложения компонента
«Управление конфигурациями». Имя директории соответствует
имени application из distrib.yaml │           │   ├──
configmaps               # Configmaps для приложения  │
│   │   └── configurator-server.yaml   │           │   ├──
dc.yaml                  # DeploymentConfig для Kubernetes.
Описывает установку POD'а с приложением │           │   ├──
svc.yaml                 # Конфигурация сервиса в Kubernetes
(привязка имени к установленным POD'ам, чтобы интерфейсы
приложения в этих POD'ах были доступны по этому имени
сервиса) │           │   ├── pdb-configurator-server.yaml
Конфигурация для PodDisruptionBudget │           ├──
configurator_rest            # Директория с настройками
сборки контейнера и развертывания приложения компонента
«Управление конфигурациями» в Kubernetes. Имя директории
соответствует имени application из distrib.yaml │
│   ├── configmaps              # Configmaps для приложения
│           │   │   └── configurator-rest.yaml     │
```

```
│   ├── dc.yaml                  # DeploymentConfig для
Kubernetes. Описывает установку POD'а с приложением │
│   ├── svc.yaml                 # Конфигурация сервиса в
Kubernetes (привязка имени к установленным POD'ам, чтобы
интерфейсы приложения в этих POD'ах были доступны по этому
имени сервиса) │           │   ├── pdb-configurator-
rest.yaml │           ├── pacman_agent                 #
Директория с настройками сборки контейнера и развертывания
приложения. Имя директории соответствует имени application
из distrib.yaml │           │   ├── configmaps
# Configmaps для приложения  │           │   │   └── pacman-
agent.yaml     │           │   ├── dc.yaml                 #
DeploymentConfig для Kubernetes. Описывает установку POD'а с
приложением. │           │   ├── svc.yaml                #
Конфигурация сервиса в Kubernetes (привязка имени к
задеплоенным POD'ам, чтобы интерфейсы приложения в этих
POD'ах были доступны по этому имени сервиса) │           │
├── pdb-pacman-agent.yaml │           ├── pacman_ui
# Директория с настройками сборки контейнера и развертывания
приложения. Имя директории соответствует имени application
из distrib.yaml           │           │   ├── dc.yaml
│           │   ├── svc.yaml  │           │   ├── pdb-
pacman-ui.yaml │           ├── uko_ufsparams_bh_v2
# Директория с настройками сборки контейнера и развертывания
приложения сервиса АРМ «Управление параметрами» в
Kubernetes. Имя директории соответствует имени application
из distrib.yaml │           │   ├── istio
# Дополнительные настройки istio для приложения АРМ
«Управление параметрами» │           │   ├── configmaps
# Configmaps для приложения  │           │   │   └── ufs-
params-config-bh-iam-publickey.yaml │           │   │   └──
ufs-params-manager.yaml │           │   ├── dc.yaml
# DeploymentConfig для Kubernetes. Описывает установку POD'а
с приложением.  │           │   ├── svc.yaml
# Конфигурация сервиса в Kubernetes (привязка имени к
установленным POD'ам, чтобы интерфейсы приложения в этих
POD'ах были доступны по этому имени сервиса). │           │
├── pdb-manager.yaml │           ├── uko_ufsparams_import_v2
# Директория с настройками сборки контейнера и развертывания
приложения сервиса Импорта «Управление параметрами» в
Kubernetes. Имя директории соответствует имени application
из distrib.yaml │           │   ├── istio
# Дополнительные настройки istio для приложения Сервис
Импорта «Управление параметрами» │           │   ├──
configmaps              # Configmaps для приложения   │
│   │   └── ufs-params-import.yaml    │           │   ├──
dc.yaml                # DeploymentConfig для Kubernetes.
Описывает установку POD'а с приложением. │           │   ├──
svc.yaml               # Конфигурация сервиса в Kubernetes
(привязка имени к установленным POD'ам, чтобы интерфейсы
```

```
приложения в этих POD'ах были доступны по этому имени
сервиса) |           |   ├── pdb-import.yaml |           ├──
uko_ufsparams_ms_v2         # Директория с настройками
сборки контейнера и развертывания приложения сервиса выдачи
параметров «Управление параметрами» в Kubernetes. Имя
директории соответствует имени application из distrib.yaml |
|   ├── istio                   # Дополнительные настройки
istio для приложения сервиса выдачи параметров «Управление
параметрами» |           |   ├── configmaps            |
|   |   └── ufs-params-service.yaml |           |   ├──
dc.yaml                 # DeploymentConfig для Kubernetes.
Описывает установку POD'а с приложением. |           |   ├──
svc.yaml                # Конфигурация сервиса в Kubernetes
(привязка имени к установленным POD'ам, чтобы интерфейсы
приложения в этих POD'ах были доступны по этому имени
сервиса) |           |   ├── pdb-service.yaml |
overrides               Каталог для установки компонентов в
среде OSE |       └── openshift |           └──
configurator_server          # Директория с настройками
сборки контейнера и развертывания приложения компонента
«Управление конфигурациями». Имя директории соответствует
имени application из distrib.yaml |           |   ├──
dc.yaml                  # DeploymentConfig для OSE.
Описывает установку POD'а с приложением |           ├──
configurator_rest            # Директория с настройками
сборки контейнера и развертывания приложения компонента
«Управление конфигурациями» в Kubernetes. Имя директории
соответствует имени application из distrib.yaml |
|   ├── dc.yaml                  # DeploymentConfig для OSE.
Описывает установку POD'а с приложением |           ├──
pacman_agent                 # Директория с настройками
сборки контейнера и развертывания приложения. Имя директории
соответствует имени application из distrib.yaml |
|   ├── dc.yaml                # DeploymentConfig для OSE.
Описывает установку POD'а с приложением. |           ├──
pacman_ui                   # Директория с настройками
сборки контейнера и развертывания приложения. Имя директории
соответствует имени application из distrib.yaml |
|   ├── dc.yaml                   |           ├──
uko_ufsparams_bh_v2          # Директория с настройками
сборки контейнера и развертывания приложения сервиса APM
«Управление параметрами» в OSE. Имя директории соответствует
имени application из distrib.yaml |           |   ├──
dc.yaml                  # DeploymentConfig для OSE.
Описывает установку POD'а с приложением.  |           ├──
uko_ufsparams_import_v2      # Директория с настройками
сборки контейнера и развертывания приложения сервиса Импорта
«Управление параметрами» в OSE. Имя директории соответствует
имени application из distrib.yaml |           |   ├──
dc.yaml                 # DeploymentConfig для OSE.
```

```
Описывает установку POD'a с приложением. |           ├──
uko_ufsparams_ms_v2          # Директория с настройками
сборки контейнера и развертывания приложения сервиса выдачи
параметров «Управление параметрами» в OSE. Имя директории
соответствует имени application из distrib.yaml |
|   ├── dc.yaml                  # DeploymentConfig для OSE.
Описывает установку POD'a с приложением.
```

Автоматизированная установка сервиса с использованием Deploy Tools

Для установки сервиса с использованием Deploy Tools (CDJE) продукта Platform V DevOps Tools (DOT) разверните сервис в инфраструктуре и запустите CD Job по инструкции, приведенной в Руководстве по установке Platform V Backend .

Стендозависимые параметры описаны в данном разделе ниже по тексту.

Первичная установка компонента «Управление параметрами» с помощью Pipeline DevOps

Перед установкой программного обеспечения с помощью Pipeline DevOps необходимо выполнить следующие действия:

- проверить наличие нужных плейбуков в списке сборки в Jenkins;
- проверить настройки Kafka.

При установке ПО с помощью Pipeline DevOps необходимо выполнить следующие действия при запуске Job раскатки:

1. Выбрать SUBSYSTEM: `UFSPARAMS_CONFIG`.
2. Выбрать DISTRIB_VERSION: `{build-version}`.
3. Выбрать Версию платформы: необходимо уточнить версию у администратора.
4. Выбрать playbook `IMPORT_SUP_PARAMS`.
5. Нажать кнопку **Собрать**.
6. После завершения выполнения сборки необходимо проверить, что лог установки не содержит ошибок.
7. Выполнить стандартную установку дистрибутива (см. пункт «Стандартная установка дистрибутива»).
8. После установки приложения в среду контейнеризации следует проверить, что все контейнеры в POD имеют статус `Ready`.

   Примечание: если все компоненты (в т.ч. внешних потребителей) устанавливаются в один (коммунальный) Namespace в среде контейнеризации, то при такой схеме установки не используйте плейбук `OPENSHIFT_INGRESS_EGRESS_DEPLOY`.

Установка компонента «Управление параметрами» с помощью Pipeline DevOps

При установке ПО с помощью Pipeline DevOps необходимо выполнить следующие действия при запуске Job раскатки:

1. Выбрать SUBSYSTEM: `UFSPARAMS_CONFIG`.
2. Выбрать DISTRIB_VERSION: `{build-version}`.
3. Выбрать Версию платформы: необходимо уточнить версию у администратора.
4. Выбрать playbooks: `MIGRATION_FP_CONF`,`FP_CONF_CHECK`,`DB_UPDATE`,`IMPORT_SUP_PARAMS`,`IMPORT_SECURITY_PARAMS`,`IMPORT_AUDIT_PARAMS`,`IMPORT_LOGGER_PARAMS`,`IMPORT_SM_PARAMS`,`NGINX_DEPLOY`,`NGINX_MM_DEPLOY`,`NGINX_II_DEPLOY`,`OPENSHIFT_DEPLOY`,`OPENSHIFT_INGRESS_EGRESS_DEPLOY`.
5. Нажать кнопку **Собрать**.
6. После завершения выполнения сборки необходимо проверить, что лог установки не содержит ошибок.
7. После установки приложения в среду контейнеризации следует проверить, что все контейнеры в pod имеют статус `Ready`.

В состав дистрибутива входят конфигурационные файлы с рекомендуемыми значениями параметров для настройки компонента, их изменение может нарушить безопасность кмпонента.

Предварительная проверка доступности инфраструктуры, возможности шаблонизации конфигурационных файлов перед непосредственно установкой компонента.

В `distrib.yml` дистрибутива компонента добавлен параметр `default_playbooks`, содержащий набор playbooks для осуществления предварительной проверки (PreCheck) доступности инфраструктуры, возможности шаблонизации конфигурационных файлов перед установкой самого компонента.

```
"playbooks_default":   "default":
["MIGRATION_FP_CONF","FP_CONF_CHECK","DB_UPDATE","IMPORT_SUP
_PARAMS","IMPORT_SECURITY_PARAMS","IMPORT_AUDIT_PARAMS","IMP
ORT_LOGGER_PARAMS","IMPORT_SM_PARAMS","NGINX_DEPLOY","NGINX_
MM_DEPLOY","NGINX_II_DEPLOY","OPENSHIFT_DEPLOY","OPENSHIFT_I
NGRESS_EGRESS_DEPLOY"]
```

Автоматический запуск предварительной проверки регулируется параметром `forcePreCheck` в конфигурации среды, куда устанавливается компонент (`environment.json`). По умолчанию параметр включен.

Набор плейбуков для выполнения PreCheck определяется следующим образом:

- Если выбран только плейбук FP_CONF_CHECK, то выполняется PreCheck списка плейбуков из default_playbooks указанный в distrib.yml в составе дистрибутива.
- Если выбраны плейбуки установки вручную, выполняется PreCheck выбранных плейбуков (для которых есть функциональность проверки).

При включенном forcePreCheck шаблонизация конфигурационных файлов падает и установка завершается с ошибкой при наличии не раскрытых параметров.

Импорт параметров Агента компонента «Управление параметрами».

Для корректной работы Агента компонента «Управление параметрами» необходимо импортировать его параметры. Для этого перед установкой приложения Потребителя обязательно нужно выполнить playbook IMPORT_ALL_PARAMS для версии агента из глобальной переменной global.ufs.config.agent.image-version или актуальной версии с помощью job раскатки с помощью Pipeline DevOps:

1. Выбрать SUBSYSTEM: UFSPARAMS_AGENT.
2. Выбрать DISTRIB_VERSION: {build-version}.
3. Выбрать Версию платформы: необходимо уточнить версию у администратора.
4. Выбрать Playbook IMPORT_ALL_PARAMS.
5. Нажать кнопку Собрать.
6. После завершения выполнения сборки необходимо проверить, что лог установки не содержит ошибок.

   Примечание: если все компоненты (в т.ч. внешних потребителей) устанавливаются в один (коммунальный) Namespace в среде контейнеризации, то при такой схеме раскатки не использовать плейбук OPENSHIFT_INGRESS_EGRESS_DEPLOY.

Установка компонента «Управление конфигурациями» с помощью Pipeline DevOps

1. Выбрать SUBSYSTEM: UFSPARAMS_CONFIG.
2. Выбрать DISTRIB_VERSION: {build-version}.
3. Выбрать Версию платформы: необходимо уточнить версию у администратора.
4. Выбрать playbook SHIFT_OK.
5. Нажать кнопку **Собрать**.
6. После завершения выполнения сборки необходимо проверить, что лог установки не содержит ошибок.

В состав дистрибутива входят конфигурационные файлы с рекомендуемыми значениями параметров для настройки компонента, их изменение может нарушить безопасность компонента.

Настройка сертификатов для БД PostgreSQL для работы компонента «Управление конфигурациями»

Используется двухсторонняя аутентификация с использованием протокола `TLSv1.2` при подключении серверной и клиентских частей компонента «Управление конфигурациями» к БД PostgreSQL.

При включении аутентификации для новых модулей по сертификатам на БД PostgreSQL уже существующие модули продолжат работу по логину/паролю. Для подключения новым клиентам сервером компонента «Управление конфигурациями» в БД будут созданы новые пользователи, у которых будет настроена аутентификация исключительно по сертификатам.

> **Примечание**. После настройки узлов на подключение с аутентификацией по сертификатам потребуется их перезапуск для применения настроек.

Подготовительные действия

1.Выпуск сертификатов.

До проведения работ должны быть выпущены сертификаты с определенными требованиями:

- Необходимо выпустить 3 набора сертификатов и ключей:
  - Для сервера PostgreSQL: `server.crt` – сертификат сервера БД, `server.key` – приватный ключ сертификата сервера БД (в формате `pem`), `issuer.crt` – промежуточный сертификат, `rootCA.crt` – корневой сертификат.
  - Для компонента «Управление конфигурациями» `configurator_ssl.crt` – сертификат клиента БД компонента «Управление конфигурациями», `configurator_ssl.pk8` – приватный ключ клиента БД компонента «Управление конфигурациями» (в формате `der`).
  - Для клиентских модулей `client_config_reader_ssl.crt` – клиентский сертификат модуля, `client_config_reader_ssl.pk8` – приватный ключ клиентского сертификата модуля (в формате `der`).
- Требования к сертификатам:
  - Сертификаты `rootCA.crt` и `issuer.crt`полученные от УЦ должны быть выпущены в формате `v3_ca`. Например, для выпуска `issuer.crt` с помощью `openssl` должны быть использована следующая команда:

```
openssl x509 -req -in issuer.csr -text -days 3560 -
extfile /etc/pki/tls/openssl.cnf -extensions v3_ca -CA
rootCA.crt -CAkey rootCA.key -CAcreateserial -out
issuer.crt
```

где: issuer.csr – это файл запроса на выпуск сертификата, rootCA.key – файл ключа корневого сертификата.

- Необходимо выпускать ключи и сертификаты, используя openssl версии не ниже той, которая используется в БД. Например, в PostgreSQL версии 9.6.10 используется OpenSSL 1.0.2k-fips, и он не принимает сертификаты, выпущенные OpenSSL 1.0.1e.
- Требования к CN: атрибут сертификата CN (Common Name) для сервера PostgreSQL должен совпадать с доменным именем сервера. Для клиентских сертификатов (компонента «Управление конфигурациями» и модулей) CN должен совпадать с логином пользователя в БД.

Запустить следующие команды:

```
openssl req -new -nodes -text -out server.csr -keyout
server.key -subj "/C=RU/ST=NSO/L=NSK/OU=POSTGRES
SERVER/O=TEMPO/CN=<доменное имя сервера>"
```

где, <доменное имя сервера> — доменное имя сервера БД, которое необходимо заменить на реальное имя. Полученный файл server.csr необходимо передать в УЦ и получить подписанный сертификат server.crt.

```
openssl req -new -nodes -text -out
client_config_reader_ssl.csr -keyout
client_config_reader_ssl.key -subj
"/C=RU/ST=NSO/L=NSK/OU=CLIENT1/O=TEMPO/CN=config_reader
_ssl"
```

Полученный файл client_config_reader_ssl.csr необходимо передать в УЦ и получить подписанный сертификат client_config_reader_ssl.crt.

Переконвертировать файл ключа в der формат:

```
openssl pkcs8 -topk8 -in client_config_reader_ssl.key -
out client_config_reader_ssl.pk8 -outform der -nocrypt
openssl req -new -nodes -text -out configurator_ssl.csr
-keyout configurator_ssl.key -subj
"/C=RU/ST=NSO/L=NSK/OU=CLIENT2/O=TEMPO/CN=configurator_
ssl"
```

Полученный файл configurator_ssl.csr необходимо передать в УЦ и получить подписанный сертификат configurator_ssl.crt. Переконвертировать файл ключа в der формат:

```
openssl pkcs8 -topk8 -in configurator_ssl.key -out
configurator_ssl.pk8 -outform der -nocrypt
```

2.Тиражирование сертификатов на узлы.

- На узле с сервером PostgreSQL необходимо создать каталог /var/lib/pgsql/ssl/, скопировать в него файлы server.crt, server.key, rootCA.crt.

  Выполнить:

  ```
  chmod 600 server.key
  ```

  В конец файла server.crt дописать содержимое файла issuer.crt.

- На узлах с Envelope.

  В домашних каталогах пользователей, под которым будет запускаться Envelope, создать каталог .postgresql. Переименовать файлы client_config_reader_ssl.crt в postgresql.crt, client_config_reader_ssl.pk8 в postgresql.pk8, rootCA.crt в root.crt.
  В каталог поместить файлы:

  - postgresql.crt – сертификат с публичным ключом агента. В конец файла дописать содержимое файла issuer.crt;
  - root.crt – корневой сертификат;
  - postgresql.pk8 – приватный ключ клиента.

  Выполнить:

  ```
  chmod 600 server.key
  ```

- В Namespace сервера компонента «Управление конфигурациями» в secret поместить файлы:
  - postgresql.crt – сертификат с публичным ключом агента. В конец файла дописать содержимое файла issuer.crt;
  - root.crt – корневой сертификат;
  - postgresql.pk8– приватный ключ клиента.

Secret должен быть примонтирован к pod сервера компонента «Управление конфигурациями» ${полный путь}.

Добавить в окружение pod следующие настройки:

1. configurator.core.db.ssl=true;
2. configurator.core.db.sslrootcert=${полный путь}/root.crt;
3. configurator.core.db.sslcert=${полный путь}/postgresql.crt;
4. configurator.core.db.ssl.key=${полный путь}/postgresql.pk8;
5. configurator.core.db.username=${login};
6. configurator.core.db.url=${строка подключения к серверу базы данных в виде jdbc:postgresql://${server.name}:5432/cfg}?sslmode=verify-full;
7. configurator.core.db.sslpassword=«» – должен быть пустым, либо должен быть указан пароль, заданный при получении ключа.

Выпуск сертификатов

Перед установкой сервера компонента «Управление конфигурациями» на основе выпущенных сертификатов и ключей для сервера и клиента компонента «Управление конфигурациями» подготовить следующие хранилища ключей jks:

- Хранилище корневых сертификатов trust.jks для сервера компонента «Управление конфигурациями».
- Хранилище с личным сертификатом key_server.jks для сервера компонента «Управление конфигурациями». Пароль от личного сертификата должен совпадать с паролем хранилища server.jks. Параметр CN (Common Name — общепринятое имя) личного сертификата должен соответствовать имени FQDN (Fully Qualified Domain Name — полное имя домена) сервера компонента «Управление конфигурациями».

Для создания хранилища ключей JKS выполните следующие действия:

1.Убедитесь, что у вас есть набор файлов сертификатов и ключей:

- <Имя_сертификата>.cer — сертификат;
- <Имя_ключа>.key — закрытый ключ;
- <Имя_промежуточного_сертификата>.cer — промежуточный сертификат, которым подписан сертификат;
- <Имя_корневого_сертификата>.cer — корневой сертификат, которым подписан промежуточный сертификат.

Обращаем ваше внимание на то, что все сертификаты должны быть в формате DER.

2.Создайте промежуточный файл хранилища в формате PKCS 12 с помощью набора инструментов OpenSSL, для этого:

a. Сохраните файл сертификата и файл ключа в формате PEM, выполнив команду в окне терминала или окне командной строки:

```
openssl x509 -in <файл_сертификата_в_формате_DER> -inform
DER -out <конечное_имя_сертификата_PEM> -outform PEM
```

Например, выполните:

```
openssl x509 -in katun-hq.<домен>.csr_cert.cer -inform DER -
out katun-hq.<домен>_PEM.crt -outform PEM
```

Файл katun-hq.<домен>.csr_cert.cer сертификата сервера КУ будет преобразован в формат PEM и сохранен под именем katun-hq.<домен>_PEM.crt.

Файл хранилища в формате PKCS12 является шифрованным контейнером и позволяет хранить в одном файле сертификат и закрытый ключ. Сертификат и закрытый ключ имеют разный формат. Закрытый ключ имеет формат контейнера Privacy Enhanced Mail (PEM).

Пример заголовка содержимого файла ключа, открытого в текстовом редакторе:

```
-----BEGIN RSA PRIVATE KEY----- Proc-Type: 4,ENCRYPTED DEK-
Info: AES-128-CBC,213210CF6D48B9DEAB7BB00D12FAE159
McAf8OxQvZZIqng7BtbgX6N0uyC+bwTiAuk…
```

Пример заголовка содержимого преобразованного файла в формате PEM: файл сертификата имеет двоичный формат DER, поэтому для хранения файла сертификата в одном файле с файлом ключа, файл сертификата следует преобразовать в формат PEM.

```
-----BEGIN CERTIFICATE----- MIIFwjCCA6qgAwIBAgIKEamwlgAAAA…
```

b. Экспортируйте преобразованный файл сертификата и файл закрытого ключа в файл хранилища формата PKCS12, выполнив команду:

```
openssl pkcs12 -export -in <файл_сертификата> -inkey
<файл_закрытого_ключа> -out <имя_файла_хранилища>.p12 -name
<имя_псевдонима_внутри_хранилища>
```

Например, выполните:

```
openssl pkcs12 -export -in katun-hq.<домен>_PEM.crt -inkey
katun-hq.<домен>.key -out hq-serv.p12 -name hq
```

Установите пароль хранилища PKCS12, введите: hyperic1.

В текущем каталоге будет создан защищенный паролем файл hq-serv.p12 с записью (alias) hq, внутри которого будут сохранены файл сертификата и файл закрытого ключа сервера.

3.Создайте файл хранилища JKS, с которым будет работать сервер компонента «Управление конфигурациями», для этого:

a. Импортируйте корневой сертификат.

Если имя файла сертификата содержит пробелы, заключайте имя в кавычки.

```
keytool -import -trustcacerts -alias root -file
"<файл_корневого_сертификата>" -keystore
<имя_файла_хранилища>.jks
```

b. Установите пароль хранилища JKS, введите: hyperic1.

В текущем каталоге будет создан защищенный паролем файл hq-serv.jks с записью (alias) root, внутри которой будет сохранен файл корневого сертификата Sberbank Test Root CA.cer.

c. Импортируйте промежуточный сертификат.

```
keytool -import -trustcacerts -alias issuing -file
"<файл_промежуточного_сертификата>" -keystore
<имя_файла_хранилища>.jks
```

4.Импортируйте сертификат и ключ из хранилища в формате PKCS12 в хранилище в формате JKS, выполнив:

```
keytool -importkeystore -srckeystore
<имя_файла_хранилища>.p12 -srcstoretype pkcs12 -srcalias
<имя_псевдонима_внутри_хранилища> -destkeystore
<имя_файла_хранилища>.jks -deststoretype jks -destalias
<новый_псевдоним_внутри_хранилища>
```

5.Введите пароль.

Файл хранилища сертификатов сервера компонента «Управление конфигурациями» создан в формате JKS и доступен.

Рекомендации по настройке пользователей для доступа к БД компонента «Управление конфигурациями»

Рекомендуется заводить разные учетные записи пользователей с разным набором прав доступа к БД для Клиентского Модуля компонента «Управление конфигурациями» и сервиса предоставления конфигурации компонента «Управление конфигурациями».

Пользователь для доступа к БД для Клиентского модуля компонента Управление конфигурациями должен обладать правом только на чтение данных. Пользователь для доступа к БД для Сервиса предоставления конфигурации компонента Управление конфигурациями может обладать правами на чтение, и на изменение данных.

Настройка SSL для Rest сервиса компонента «Управление конфигурациями»

Общее: создать хранилище JKS для цепочки доверенных сертификатов.

Настройка серверной части (модуль configurator-rest):

1. Выпустить ключ и серверный сертификат, который либо в секции CN, либо в секции SubjectAlternativeNames содержит URL по шаблону `ingressgateway-mtls-ufsparams-configurator-rest-${distrib.release.version}.${projectName}.${appsDomain}`, где `${distrib.release.version}`, `${projectName}` и `${appsDomain}` глобальные переменные. Это URL, по которому к сервису будут обращаться клиенты.
2. Создать хранилище PKCS12 для приватного ключа и сертификата сервиса.
3. Добавить пути к хранилищам в конфигурацию сервиса:
   - server:
     - ssl:
       - `key-store`: ${хранилище с серверным сертификатом};
       - `key-store-password`: ${пароль от хранилища с серверным сертификатом};
       - `trust-store`: ${хранилище с серверным доверенными сертификатами};
       - `trust-store-password`: ${пароль от хранилища с доверенными сертификатами}.
4. Тело хранилища с приватным ключом в формате `base64` разместить в файле secret `configuration-rest-ssl-certs.yaml` с наименованием `configuration-rest-ssl-certs` в область `data` под наименованием `key-store.p12`.

5. Пароль от хранилища с приватным ключом в формате base64 разместить в файле secret configuration-rest-passwords.yaml с наименованием configuration-rest-passwords в область data под наименованием CONFIGURATION_REST_PRIVATE_KEY_PWD.
6. Создать хранилище JKS для доверительных сертификатов.
7. Тело хранилища с доверительными сертификатами в формате base64 разместить в файле secret configuration-rest-ssl-certs.yaml с наименованием configuration-rest-ssl-certs в область data под наименованием trustStore.jks.
8. Пароль от хранилища с доверительными сертификатами в формате base64 разместить в файле secret configuration-rest-passwords.yaml с наименованием configuration-rest-passwords в область data под наименованием CONFIGURATION_REST_TRUST_STORE_PWD.
9. Секреты разместить в namespace проекта.

Настройка клиентской части (клиент на Wildfly):

1. Выпустить клиентский сертификат, содержащий в секции CN некоторый идентификатор, который будет использоваться в интерфейсе компонента «Управление конфигурациями» для указания сертификатов, которым разрешен доступ к артефакту.
2. Создать хранилище PKCS12 для приватного ключа и сертификата.
3. Настроить параметры config-store на Wildfly, используя vaults если нужно:
   - <property name="config-store@configuration-loader-mode" value="HTTP"/>;
   - <property name="config-store@rest.hostname" value="https://${hostname}/"/>;
   - <property name="config-store@rest.keyStoreType" value="PKCS12"/>;
   - <property name="config-store@rest.keyStore" value="${хранилище с клиентским сертификатом}"/>;
   - <property name="config-store@rest.keyStorePassword" value="${пароль от хранилища с клиентским сертификатом}"/>;
   - <property name="config-store@rest.trustStoreType" value="JKS"/>;
   - <property name="config-store@rest.trustStore" value="${хранилище с серверным доверенными сертификатами}"/>;
   - <property name="config-store@rest.trustStorePassword" value="${пароль от хранилища с доверенными сертификатами}"/>.

Параметры для безопасной передачи паролей

В конфигурационный `distrib.yml` добавляется область `secrets`, в которую размещаются параметры для передачи паролей:

```
istio-secrets: ["ssl.ose.istio.keyStore.egress.password",
"ssl.ose.keyStore.mq.password"]  secrets: [
"jdbc.UFSPARAMS_PG.password",
"jdbc.UFSPARAMS_PG.user",
"jdbc.CONFIGURATOR_PG.user",
"jdbc.CONFIGURATOR_PG.password",            "server.ssl.key-
store-password",            "server.ssl.trust-store-
password",            "ssl.ose.keyStore.mq.password",
"ufs.logger.kafka.ssl.truststore.password",
"ufs.logger.kafka.ssl.keystore.password",
"clipas.secret-key",            "configurator.core.crypto-
password" ]
```

Secrets

№	Параметр	Описание
1	jdbc.UFSPARAMS_PG.password	Пароль для подключения к БД компонента «Управление параметрами»
2	jdbc.UFSPARAMS_PG.user	Логин пользователя для подключения к БД компонента «Управление параметрами»
3	jdbc.CONFIGURATOR_PG.user	Пароль для подключения к БД компонента «Управление конфигурациями»
4	jdbc.CONFIGURATOR_PG.password	Логин для подключения к БД компонента «Управление конфигурациями»
5	server.ssl.key-store-password	Пароль от хранилища ключа для настройки mTLS REST сервиса компонента «Управление конфигурациями»

№	Параметр	Описание
6	server.ssl.trust-store-password	Пароль от хранилища доверенных сертификатов для настройки mTLS REST сервиса компонента «Управление конфигурациями»
7	ssl.ose.keyStore.mq.password	Пароль от хранилища ключа для настройки mTLS к Kafka
8	ufs.logger.kafka.ssl.truststore.password	Пароль от доверенных сертификатов для настройки mTLS к Kafka Logger
9	ufs.logger.kafka.ssl.keystore.password	Пароль от хранилища ключа для настройки mTLS к Kafka Logger
10	clipas.secret-key	Секрет для настройки удаленного компонента «Авторизации»(SPAS)
11	configurator.core.crypto-password	Секрет для настройки компонента «Управление конфигурациями»
12	ssl.ose.istio.keyStore.egress.password	Параметр с паролями от хранилищ и сертификата
13	ssl.ose.keyStore.mq.password	

Глобальные параметры, используемые при раскатке компонента

№	Параметр	Описание
1	`${appsDomain}`	API кластера
2	`${distrib.release.version}`	Версия релиза дистрибутива
3	`${global.default.nlb_mm.host}`	Настройка сетевого шлюза, хост

№	Параметр	Описание
4	${global.default.nlb_mm.port}	Настройка сетевого шлюза, порт
5	${global.healthcheck.agent.url}	URL для Агента healthcheck
6	${global.jdbc.hikari.aliveBypassWindowMs}	Настройка Hikari DataSource
7	${global.jdbc.hikari.housekeeping.periodMs}	Настройка Hikari DataSource
8	${global.jdbc.oracle.spring.datasource.driver-class-name}	Настройка DataSource
9	${global.jdbc.oracle.spring.datasource.hikari.connection-init-sql}	Настройка Hikari DataSource
10	${global.jdbc.oracle.spring.datasource.hikari.data-source-properties.oracle.jdbc.ReadTimeout}	Настройка Hikari DataSource
11	${global.jdbc.spring.datasource.hikari.connection-test-query}	Настройка Hikari DataSource
12	${global.jdbc.spring.datasource.hikari.connection-timeout}	Настройка Hikari DataSource
13	${global.jdbc.spring.datasource.hikari.idle-timeout}	Настройка Hikari DataSource
14	${global.jdbc.spring.datasource.hikari.initializationFailTimeout}	Настройка Hikari DataSource

№	Параметр	Описание
15	${global.jdbc.spring.datasource.hikari.leak-detection-threshold}	Настройка Hikari DataSource
16	${global.jdbc.spring.datasource.hikari.max-lifetime}	Настройка Hikari DataSource
17	${global.jdbc.spring.datasource.hikari.maximum-pool-size}	Настройка Hikari DataSource
18	${global.jdbc.spring.datasource.hikari.minimum-idle}	Настройка Hikari DataSource
19	${global.jdbc.spring.datasource.hikari.validation-timeout}	Настройка Hikari DataSource
20	${global.kafka.monitoring.bootstrap.servers}	Настройка Kafka для Мониторинга
21	${global.kafka.monitoring.filter.topic}	Настройка Kafka для Мониторинга
22	${global.kafka.monitoring.metadata.topic}	Настройка Kafka для Мониторинга
23	${global.kafka.monitoring.metrics.topic}	Настройка Kafka для Мониторинга
24	${global.kafka.monitoring.settings.topic}	Настройка Kafka для Мониторинга
25	${global.kafka.monitoring.ssl.store.name}	Настройка Kafka для Мониторинга
26	${global.monitoring.jms.enabled "или" true}	Настройка включения JMS для Мониторинга

№	Параметр	Описание
27	${global.multiClusters.openshiftNewRoute}	Переменная для идентификации ссылок доступа к кластеру в среде контейнеризации
28	${global.ose.params.baseUrl.stringToBind}	Base url для компонента «Управление параметрами»
29	${global.ose.platform.egress.http.port}	Настройка Istio. Порт для общения Приложения с egress (рекомендованное значение: 7070).
30	${global.ose.platform.ufs.baseurl.logger}	Base url для сервиса фильтрации журналирования
31	${global.ott.grpc.port}	Порт OTT
32	${global.ott.image.sidecar.version}	Версия образа sidecar OTT
33	${global.ott.service.hosts}	Настройка хостов серверов OTT
34	${global.ott.service.url}	Настройка сервиса OTT
35	${global.ott.trust.store.path}	Путь до хранилища доверенных сертификатов
36	${global.params.baseUrl.stringToBind}	Base url для компонента «Управление параметрами»
37	${global.platform.audit.kafka.bootstrap.servers}	Настройка Аудит
38	${global.platform.kafka.consumer.fetch.max.bytes}	Настройка получателя сообщений через Kafka
39	${global.platform.kafka.consumer.fetch.max.wait.ms}	Настройка получателя сообщений через Kafka

№	Параметр	Описание
40	${global.platform.kafka.consumer.fetch.min.bytes}	Настройка получателя сообщений через Kafka
41	${global.platform.kafka.consumer.heartbeat.interval.ms}	Настройка получателя сообщений через Kafka
42	${global.platform.kafka.consumer.max.partition.fetch.bytes}	Настройка получателя сообщений через Kafka
43	${global.platform.kafka.consumer.max.poll.interval.ms}	Настройка получателя сообщений через Kafka
44	${global.platform.kafka.consumer.receive.buffer.bytes}	Настройка получателя сообщений через Kafka
45	${global.platform.kafka.consumer.request.timeout.ms}	Настройка получателя сообщений через Kafka
46	${global.platform.kafka.consumer.send.buffer.bytes}	Настройка получателя сообщений через Kafka
47	${global.platform.kafka.consumer.session.timeout.ms}	Настройка получателя сообщений через Kafka
48	${global.platform.kafka.producer.batch.size}	Настройка отправителя сообщений через Kafka
49	${global.platform.kafka.producer.buffer.memory}	Настройка отправителя сообщений через Kafka
50	${global.platform.kafka.producer.compression.type}	Настройка отправителя сообщений через Kafka

№	Параметр	Описание
51	${global.platform.kafka.producer.connections.max.idle.ms}	Настройка отправителя сообщений через Kafka
52	${global.platform.kafka.producer.delivery.timeout.ms}	Настройка отправителя сообщений через Kafka
53	${global.platform.kafka.producer.linger.ms}	Настройка отправителя сообщений через Kafka
54	${global.platform.kafka.producer.max.block.ms}	Настройка отправителя сообщений через Kafka
55	${global.platform.kafka.producer.max.request.size}	Настройка отправителя сообщений через Kafka
56	${global.platform.kafka.producer.metadata.max.age.ms}	Настройка отправителя сообщений через Kafka
57	${global.platform.kafka.producer.receive.buffer.bytes}	Настройка отправителя сообщений через Kafka
58	${global.platform.kafka.producer.request.timeout.ms}	Настройка отправителя сообщений через Kafka
59	${global.platform.kafka.producer.send.buffer.bytes}	Настройка отправителя сообщений через Kafka
60	${global.platform.kafka.security.protocol}	Настройка протокола безопасности для подключения к Kafka
61	${global.platform.kafka.ssl.enabled.protocols}	Настройка включения SSL для подключения к Kafka

№	Параметр	Описание
62	${global.platform.kafka.ssl.endpoint.identification.algorithm}	Настройка алгоритма идентификации при подключении к Kafka
63	${global.platform.logger.kafka.bootstrap.servers}	Bootstrap настройка серверов Kafka сервисов журналирования
64	${global.platform.logger.kafka.security.protocol}	Настройка протокола безопасности для подключения к Kafka журналирования
65	${global.platform.logger.kafka.topic}	Настройка топика журналирования
66	${global.platform.ose.kafka.ports}	Адреса портов Kafka для работы приложения через Istio (рекомендованное значение: 9092)
67	${global.platform.sgw.cache.port}	Настройка SGW
68	${global.platform.sgw.config.port}	Настройка SGW
69	${global.platform.sgw.log.port}	Настройка SGW
70	${global.platform.sgw.route.cache.host}	Настройка SGW
71	${global.platform.sgw.route.config.host}	Настройка SGW. Шлюз, на который нужно ходить с MTLS
72	${global.platform.sgw.route.log.host}	Настройка SGW
73	${global.platform.sgw.route.session.host}	Настройка SGW

№	Параметр	Описание
74	${global.platform.sgw.session.port}	Настройка SGW
75	${global.platform.sgw.route.core.host}	Настройка SGW
76	${global.platform.sgw.core.port}	Настройка SGW
77	${global.platform.sgw.core.protocol}	Настройка SGW
78	${global.platform.ingress.route.https}	HTTPS ссылка на Istio Ingress
79	${global.platform.ingress.route.http}	HTTP ссылка на Istio Ingress
80	${global.platform.audit2.host}	Настройка Audit2
81	${global.platform.audit2.port}	Настройка Audit2
82	${global.platform.audit2.protocol}	Настройка Audit2
83	${global.platform.audit2.internal.port}	Настройка Audit2
84	${global.platform.ufs.baseurl.logger}	Base url до удаленного компонента журналирования
85	${global.queue.AuditSendQueue.baseQueueName}	Наименование очереди MQ аудита
86	${global.queue.monitoring.baseQueueName}	Наименование очереди MQ мониторинга для событий
87	${global.queue.monitoring.persistence}	Свойство постоянного хранения в очереди MQ мониторинга для событий

№	Параметр	Описание
88	${global.queue.monitoringMeta.baseQueueName}	Наименование очереди MQ мониторинга для метаданных
89	${global.queue.monitoringMeta.persistence}	Свойство постоянного хранения в очереди MQ мониторинга для метаданных
90	${global.topic.monitoringFilter.baseQueueName}	Наименование очереди MQ мониторинга для фильтра
91	${global.topic.monitoringFilter.persistence}	Свойство постоянного хранения в очереди MQ мониторинга для фильтра
92	${global.ufs-logger.queue.ufsIntegrationJournalListLoggerQueue.baseQueueName}	Наименование очереди MQ журналирования для интеграционного журнала
93	${global.ufs-logger.queue.ufsSystemJournalListLoggerQueue.baseQueueName}	Наименование очереди MQ журналировния для системного журнала
94	${global.ufs-logger.queue.ufsUserJournalListLoggerQueue.baseQueueName}	Наименование очереди MQ журналировfния для пользовательского журнала
95	${global.ufs.strategy.intervalSeconds}	Время ожидания между опросом состояния установки после обновления
96	${global.ufs.strategy.maxSurge}	Максимальное число pod, которые могут быть запланированы выше исходного количества POD (можно в процентах, можно явное количество)

№	Параметр	Описание
97	`${global.ufs.strategy.maxUnavailable}`	Максимальное количество pod, которые могут быть недоступны во время обновления (можно в процентах, можно явное количество)
98	`${global.ufs.strategy.timeoutSeconds}`	Время ожидания поднятия pod, прежде чем откатиться к предыдущей успешной установке
99	`${global.ufs.strategy.updatePeriodSeconds}`	Время ожидания между отдельными обновлениями Pod
100	`${global.ufs.synapse.controlPlane.project}`	Имя проекта, где развернут control-plane istio. Должно соответствовать значению maistra.io/member-of (в консоли среды контейнеризации: Networking → Network Policies → istio mesh)
101	`${global.ufs.synapse.envoy.image}`	Ссылка на базовый имедж envoy
102	`${global.wm.monitoring.isGrowable}`	Настройки мониторинга
103	`${global.wm.monitoring.maxThreads}`	Настройки мониторинга
104	`${global.wm.monitoring.minThreads}`	Настройки мониторинга
105	`${global.wm.monitoring.numAlarmThreads}`	Настройки мониторинга
106	`${jdbc.efs.url}`	Строка подключения к БД

№	Параметр	Описание
107	${jenkins_env.fp_artifact_version}	Метка с номером версии дистрибутива в секторе конфигурации развертывания
108	${jenkins_env.fp_image_hash}	Уникальный hash образов устанавливаемых сервисов
109	${jenkinsDeployId}	Метка с номером идентификатора задания Jenkins в секторе конфигурации развертывания
110	${projectName}	Имя проекта
111	${ssl.ose.istio.keyStore.egress.CertAlias}	Alias сертификата для исходящих запросов istio
112	${ssl.ose.istio.keyStore.egress.KeyStoreFromFile}	Файл с ключами и сертификата для исходящих запросов istio
113	${ssl.ose.istio.keyStore.ingress.CertAlias}	Alias сертификата для входящих запросов istio
114	${ssl.ose.istio.keyStore.ingress.KeyStoreFromFile}	Файл с ключами и сертификата для входящих запросов istio
115	${ssl.ose.istio.keyStore.RootCertAlias}	Alias рутового сертификата для istio
116	{{ TENANT_CODE }}	Код тенанта
117	${global.platform.spas.clipas.spas-server-url}	Настройка для подключения удаленного компонента Авторизации
118	${global.ose.platform.ufs.baseurl.security}	Настройка для подключения Аудит

№	Параметр	Описание
119	${global.platform.kafka.configurator.client.topic-name}	Настройка для подключения Клиента компонента «Управление конфигурациями» к топику компонента «Управление конфигурациями»
120	${global.platform.audit2.host}	Настройка для подключения Аудит
121	${global.platform.audit2.port}	Настройка для подключения Аудит
122	${global.platform.audit2.protocol}	Настройка для подключения Аудит
123	${global.platform.spas.host}	Настройка для подключения удаленного компонента Объединенный сервис авторизации (ОСА)
124	${global.platform.spas.port}	Настройка для подключения удаленного компонента Объединенный сервис авторизации (ОСА)
125	${global.platform.spas.protocol}	Настройка для подключения удаленного компонента Объединенный сервис авторизации (ОСА)
126	${global.platform.fluent-bit.sidecar.image-version}	Настройка для подключения Fluent bit
127	${global.platform.fluent-bit.limits.cpu}	Настройка для подключения Fluent bit
128	${global.platform.fluent-bit.limits.memory}	Настройка для подключения Fluent bit
129	${global.platform.fluent-bit.requests.cpu}	Настройка для подключения Fluent bit

№	Параметр	Описание
130	${global.platform.fluent-bit.requests.memory}	Настройка для подключения Fluent bit
131	${global.pacman.adapters.monitoring.unimon}	Настройка адаптера для подключения к удаленному компоненту Объединенный мониторинг Unimon (MONA) для работы компонента «Управление конфигурациями»
132	${global.pacman.adapters.monitoring.stub}	Настройка адаптера для подключения к удаленному компоненту Объединенный мониторинг Unimon (MONA) для работы компонента «Управление конфигурациями»
133	${global.pacman.adapters.audit.pprb}	Настройка адаптера для подключения к удаленному компоненту Аудит продукта Platform V Audit SE для работы компонента «Управление конфигурациями»
134	${global.pacman.adapters.audit.ufs}	Настройка адаптера для подключения к удаленному компоненту Аудит продукта Platform V Audit SE для работы компонента «Управление конфигурациями»
135	${global.pacman.adapters.authorization.pprb.enable}	Настройка адаптера для подключения к удаленному компоненту Объединенный сервис авторизации (ОСА) для работы компонента «Управление конфигурациями»

№	Параметр	Описание
136	${global.pacman.adapters.authorization.pprb.jwt.enable}	Настройка адаптера для подключения к удаленному компоненту «Авторизации» для работы компонента «Управление конфигурациями»
137	${global.pacman.adapters.authorization.pprb.sudir.enable}	Настройка адаптера для подключения к удаленному компоненту «Авторизации» для работы компонента «Управление конфигурациями»
138	${global.platform.kafka.configurator.ssl-keystore-location}	Настройка для подключения Клиента компонента «Управление конфигурациями» к Kafka по SSL
139	${global.platform.kafka.configurator.ssl-truststore-location}	Настройка для подключения Клиента компонента «Управление конфигурациями» к Kafka по SSL
140	${global.platform.kafka.configurator.client.ssl-keystore-type}	Настройка для подключения Клиента компонента «Управление конфигурациями» к Kafka по SSL
141	${global.platform.kafka.configurator.client.ssl-truststore-type}	Настройка для подключения Клиента компонента «Управление конфигурациями» к Kafka по SSL
142	${global.platform.kafka.ssl.endpoint.identification.algorithm}	Настройка для подключения Клиента компонента «Управление конфигурациями» к Kafka по SSL

Настройка подключения к брокерам Kafka

В ConfigMaps сервера `configurator-server.*` настраиваются следующие настройки для подключения к Kafka:

```
configurator:       core:         kafka:
bootstrap-servers:
${ufsparams_config.configurator.core.kafka.bootstrap.servers
}           security-protocol:
${ufsparams_config.configurator.core.kafka.security.protocol
}           ssl-keystore-location:
${ufsparams_config.configurator.core.kafka.configurator.ssl-
keystore-location}           ssl-truststore-location:
${ufsparams_config.configurator.core.kafka.configurator.ssl-
truststore-location}           topic-name:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.topic-name}           ssl-enabled-protocols:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-enabled-protocols}           ssl-protocol:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-protocol}           ssl-keystore-type:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-keystore-type}           ssl-truststore-type:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-truststore-type}           ssl-endpoint-
identification-algorithm:
${ufsparams_config.configurator.core.kafka.configurator.clie
nt.ssl-endpoint-identification-algorithm}
```

Список брокеров настраиваются в глобальном параметре `global.platform.kafka.bootstrap.servers`.

Хранилище сертификатов для подключения к брокерам по SSL размещаются в файле `mk.jks` и монитруется по адресу `/etc/config/ssl/`.

```
...   volumeMounts:     - mountPath: /etc/config/ssl/
name: ufs-mq-jks-vol       readOnly: true ...   volumes:
- name: ufs-mq-jks-vol       secret:         defaultMode:
420         secretName: mq.jks
```

Настройка подключения Аудита

Настройка адаптера подключения к удаленному компоненту Аудит продукта Platform V Audit SE осуществляется через файл `application.yaml`.

```
spring:   autoconfigure:     exclude:
<домен>.ufs.platform.audit.AuditAutoConfiguration  pacman:
adapters:      audit: // переключатель для адаптера Аудита
selector: ${ufsparams_config.pacman.adapters.audit.selector}
timeout: ${ufsparams_config.pacman.adapters.audit.timeout}
group: ${configurator.core.audit.group}  audit2:   url:
${configurator_rest.core.audit2.baseUrl}
```

Возможные значения `selector`:

- `pprb-1.0` – запись событий аудита для компонента «Управление конфигурациями», версия с передачей информации об измененных значениях параметров;
- `pprb-2.0` – запись событий аудита для компонента «Управление конфигурациями», версия с передачей информации о старых и измененных значениях параметров;
- `log` – «заглушка» для записи событий аудита в локальный лог, без передачи в удаленный компонент;
- `ufs` – запись событий аудита для компонента «Управление параметрами».

Настройка подключения Мониторинга

Настройка адаптера подключения к удаленному компоненту Объединенный мониторинг Unimon продукта Platform V Monitor осуществляется через файл `application.yaml`.

```
pacman:   adapters:     monitoring:       unimon:
${ufsparams_config.pacman.adapters.monitoring.unimon}
stub: ${ufsparams_config.pacman.adapters.monitoring.stub} //
заглушка для DEV режима
```

Настройка подключения Журналирования

Настройка файла записи логов для удаленного компонента Журналирования осуществляется через файл `application.yaml`.

```
logging:   file:     path: /fluent-bit/etc/logs   config:
/config-log/logback-spring.xml
```

Настойки берутся из файла `logback-spring.xml`, в котором так же указывается уровень логирования.

Настройка подключения Аутентификации/Авторизации для работы компонента «Управление конфигурациями»

Настройка адаптера подключения к продукту Platform V IAM SE осуществляется через файл `configurator-server.yaml`.

При настройках PROM режима включить адаптер `pprb`, у него может быть авторизация с использованием JWT или с интеграцией с удаленным компонентом IAM Proxy продукта Platform V IAM SE):

```
pacman:   adapters:       authorization:           pprb:
enable:
${ufsparams_config.pacman.adapters.authorization.pprb.enable
}             jwt:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.jwt.en
able}             sudir:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.sudir.
enable}             basic:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.basic.
enable}             tenants: ${ufs.tenant.code}
ufs:             baseUrl:
${ufsparams_config.authorization.ufs.baseUrl}/ufs-security
enable:
${ufsparams_config.pacman.adapters.authorization.ufs.enable}
sudir:               enable:
${ufsparams_config.pacman.adapters.authorization.ufs.sudir.e
nable}             iam:               enable:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ena
ble}               jws_token_header:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.jws
_token_header}               roles_path:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.rol
es_path}               publickey_locations:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.pub
lickey_locations}               audience:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.aud
ience}               ssl_required:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_required}               ssl_truststore:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_truststore}               ssl_truststore_password:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_truststore_password}               poolsize:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.poo
lsize}               cache_updatetime:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.cac
```

```
he_updatetime}                devmode:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.dev
mode}
```

Настройки PROM режима:

```
spring:   profiles:     active:
${configurator.spring.active.profiles}
```

Если профиль был DEV, то включится аутентификация BASIC.

При включении профиля AUTH_DEV включает аварийный режим emergency, в данном режиме пользователь авторизуется с супер правами.

URL до удаленного компонента Авторизации (OCA):

```
clipas:   spas-server-url: ${configurator.core.clipas.spas-
server-url}
```

Настройка `clipas.secret-key` для секретного ключа удаленного компонента Авторизации в secret.

Загрузка ролевой модели.

Первоначальная ролевая модель доступа пользователей в АРМ компонента «Управление конфигурациями» загружается из файла дистрибутива `securiti-data-spas.xml` в удаленный компонент Авторизации при установке с помощью инструментов pipeline DevOps.

Расположение файла с ролевой моделью в дистрибутиве:

```
├── conf │   └── data │       └── spas
│            └── securiti-data-spas.xml      # Ролевая модель
компонента «Управление конфигурациями» в удаленный компонент
«Авторизации»
```

Файл импортируется в удаленный компонент авторизации продукта Platform V IAM SE автоматически средствами DevOps pipeline в процессе развертывания приложения-потребителя.

Удаленный компонент авторизации при запросе возвращает привилегии аутентифицированного пользователя.

Дополнительно загрузить модель `securiti-data-spas.xml` в удаленный компонент Авторизации можно через UI. Подробнее о работе с удаленным

компонентом «Авторизации» можно узнать в руководстве оператора в разделе «Использование приложения оператором».

Далее необходимо распределить соответствующие роли пользователям через интерфейс удаленного компонента Авторизации.

Параметра настройки ролевой модели из файла `securiti-data-spas.xml`:

```
<?xml version="1.0" encoding="UTF-8"?> <module
moduleVersion="3.3.0" roleModelVersion="1.0"
name="configurator-module"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://AccessSystem/namespace"
xsi:schemaLocation="http://AccessSystem/namespace">     <!--
С версии 7.1.0 -->     <trustedModules>
<trustedModule name="azimuth-forms"/>     </trustedModules>
<nodes>         <node display-name="Управление
конфигурациями">             <permission
id="Configurator.ArtifactManager.ImportModel"
display-name="Импорт моделей артефактов"
type="USER_RIGHT"/>             <permission
id="Configurator.ArtifactManager.ImportProperties"
display-name="Импорт настроек из файла"
type="USER_RIGHT"/>             <permission
id="Configurator.ArtifactManager.Export.Properties"
display-name="Экспорт настроек в файл properties"
type="USER_RIGHT"/>             <permission
id="Configurator.ArtifactManager.Export.Yaml"
display-name="Экспорт настроек в файл yaml"
type="USER_RIGHT"/>             <permission
id="Configurator.ArtifactManager.Delete"
display-name="Удаление артефакта"
type="USER_RIGHT"/>             <permission
id="Configurator.ArtifactManager.Edit"
display-name="Редактирование настроек конфигураций
артефакта"                          type="USER_RIGHT"/>
<permission id="Configurator.ArtifactManager.Edit.All"
display-name="Редактирование настроек конфигураций всех
артефактов"                          type="USER_RIGHT"/>
<permission id="Configurator.ArtifactManager.View"
display-name="Просмотр конфигураций артефакта"
type="USER_RIGHT"/>             <permission
id="Configurator.ArtifactManager.View.All"
display-name="Просмотр конфигураций всех артефактов"
type="USER_RIGHT"/>             <permission
id="Configurator.Complex.View"
display-name="Просмотр всех ККП"
type="USER_RIGHT"/>             <permission
id="Configurator.Complex.CreateComplex"
```

```
display-name="Создание ККП"
type="USER_RIGHT"/>             <permission
id="Configurator.Complex.DeleteComplex"
display-name="Удаление ККП"
type="USER_RIGHT"/>             <permission
id="Configurator.Complex.EditComplex"
display-name="Настройка ККП"
type="USER_RIGHT"/>             <permission
id="Configurator.RoleModel.Export"
display-name="Выгрузка ролевой модели"
type="USER_RIGHT"/>         </node>     </nodes>
<roles>         <role code="EFS_APPLICATION_ADMIN"
description="Группа администраторов" name="Группа
администраторов">             <permission-ref
id="Configurator.Complex.View"/>             <permission-ref
id="Configurator.Complex.CreateComplex"/>
<permission-ref id="Configurator.Complex.DeleteComplex"/>
<permission-ref id="Configurator.Complex.EditComplex"/>
<permission-ref
id="Configurator.ArtifactManager.ImportModel"/>
<permission-ref id="Configurator.ArtifactManager.View"/>
<permission-ref id="Configurator.ArtifactManager.View.All"/>
<permission-ref
id="Configurator.ArtifactManager.ImportProperties"/>
<permission-ref
id="Configurator.ArtifactManager.Export.Properties"/>
<permission-ref
id="Configurator.ArtifactManager.Export.Yaml"/>
<permission-ref id="Configurator.ArtifactManager.Edit"/>
<permission-ref id="Configurator.ArtifactManager.Edit.All"/>
<permission-ref id="Configurator.ArtifactManager.Delete"/>
</role>          <role code="ROLE_REVISOR"
description="Группа ревизоров" name="Группа ревизоров">
<permission-ref id="Configurator.Complex.View"/>
<permission-ref id="Configurator.ArtifactManager.View"/>
<permission-ref id="Configurator.ArtifactManager.View.All"/>
</role>          <role code="ROLE_IMPORTER"
description="Группа пользователей имморта конфигураций"
name="Группа пользователей имморта конфигураций">
<permission-ref
id="Configurator.ArtifactManager.ImportModel"/>
<permission-ref
id="Configurator.ArtifactManager.ImportProperties"/>
</role>          <role code="ROLE_MODEL_EXPORTER"
description="Администратор выгрузки ролевой модели"
name="Администратор выгрузки ролевой модели">
<permission-ref id="Configurator.RoleModel.Export"/>
</role>     </roles>  </module>
```

Выгрузка ролевой модели для новых артефактов.

После загрузки модели нового артефакта конфигурации необходимо выдать права пользователю на работу с данным артефактом.

Для этого необходимо выгрузить модель авторизации из компонента «Управление конфигурациями» и загрузить в удаленной компонент авторизации.

Для этого необходимо:

1. Зайти в АРМ компонента «Управление конфигурациями» пользователем с ролью `ROLE_MODEL_EXPORTER` (роль должна содержать необходимую привилегию доступа –`Configurator.RoleModel.Export`).
2. В браузере перейти по: `http://хост:порт/ufs-config-manager/pacman/configurator/cfg/api/spasRoleModel`.
3. Сохранить выгруженную модель в файл .xml (в файле будет содержаться сгенерированная роль вида `role code="Configurator.ArtifactManager.<artefact>"` с набором привилегий для доступа к новому артефакту).
4. Загрузить модель в удаленный компонент Авторизации через UI или через API для автоматической загрузки модели.
5. Выдать пользователю роль доступа к загруженному артефакту.

Настройка подключения Аутентификации/Авторизации для работы компонента «Управление параметрами»

Настройка адаптера подключения в удаленному компоненту Авторизации продукта Platform V IAM SE осуществляется через файл `ufs-params-manager.yaml`.

При настройках PROM режима включить адаптер `ufs`:

```
pacman:       adapters:         authorization:
pprb:             enable:
${ufsparams_config.pacman.adapters.authorization.pprb.enable
}             jwt:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.jwt.en
able}             sudir:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.sudir.
enable}             basic:               enable:
${ufsparams_config.pacman.adapters.authorization.pprb.basic.
enable}             tenants: ${ufs.tenant.code}
ufs:            baseUrl:
${ufsparams_config.authorization.ufs.baseUrl}/ufs-security
enable:
${ufsparams_config.pacman.adapters.authorization.ufs.enable}
sudir:              enable:
${ufsparams_config.pacman.adapters.authorization.ufs.sudir.e
nable}             iam:               enable:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ena
```

```
ble}                jws_token_header:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.jws
_token_header}                roles_path:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.rol
es_path}                publickey_locations:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.pub
lickey_locations}                audience:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.aud
ience}                ssl_required:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_required}                ssl_truststore:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_truststore}                ssl_truststore_password:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.ssl
_truststore_password}                poolsize:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.poo
lsize}                cache_updatetime:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.cac
he_updatetime}                devmode:
${ufsparams_config.pacman.adapters.authorization.ufs.iam.dev
mode}
```

Загрузка ролевой модели.

Ролевая модель находится в дистрибутиве:

```
├── conf │   └── data │       └── security
# Каталог, с настройками безопасности │           └──
security-data.xml           # Ролевая модель компонента
«Управление параметрами», «Управление конфигурациями» в
удаленный компонент «Авторизации»
```

Параметры настройки ролевой модели указаны в файле `security-data.xml`.

Файл импортируется в удаленный компонент Авторизации продукта Platform V IAM SE автоматически средствами DevOps pipeline в процессе развертывания приложения-потребителя. Удаленный компонент Авторизации при запросе возвращает привилегии аутентифицированного пользователя.

Подробнее о работе с удаленным компонентом Авторизации можно узнать в руководстве оператора в разделе «Использование приложения оператором».

Параметра настройки ролевой модели из файла `security-data.xml`:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>      <resource code="ManageParameters"
name="Управление параметрами"
subsystem="PARAMS">          <resource
```

```
code="ManageParameters.Services"
name="Сервисы СУП"                    subsystem="PARAMS">
<action code="ManageParameters.Services.Access"
name="Доступ к сервисам СУП"
category="ADMIN"/>         </resource>          <resource
code="ManageParameters.ParameterRole"
name="Доступ к параметрам, заведенным под определенной
ролью"                   subsystem="PARAMS">
<action code="ManageParameters.ParameterRole.System"
name="Доступ к системным параметрам"
category="ADMIN"/>         </resource>          <resource
code="ManageParameters.StartManager"
name="Управление плитками в Стартовом Менеджере"
subsystem="PARAMS">              <action
code="ManageParameters.StartManager.ViewPanel"
name="Доступ к плитке СУП"
category="ADMIN"/>         </resource>          <resource
code="ManageParameters.Parameters"
name="Управление параметрами"
subsystem="PARAMS">              <action
code="ManageParameters.Parameters.View"
name="Просмотр параметров и значений, экспорт"
category="ADMIN"/>             <action
code="ManageParameters.Parameters.View.All"
name="Просмотр всех параметров и значений, экспорт"
category="ADMIN"/>             <action
code="ManageParameters.Parameters.Edit"
name="Добавление, редактирование, удаление параметров и
значений"                     category="ADMIN"/>
<action code="ManageParameters.Parameters.Edit.All"
name="Добавление, редактирование, удаление всех параметров и
значений"                     category="ADMIN"/>
</resource>      </resource>      <role
code="UfsParamsSuperAdminRole_${tenant_code}"
name="Роль Прикладного Администратора платформенного сектора
СУП тенанта ${tenant_code}"           category="ADMIN"
subsystem="PARAMS">         <needRoleSUDIR
name="EFS_APPLICATION_ADMIN"/>         <permission>
<action-ref code="ManageParameters.Services.Access"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.StartManager.ViewPanel"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.Parameters.View"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.Parameters.View.All"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
```

```
code="ManageParameters.Parameters.Edit"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.Parameters.Edit.All"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.ParameterRole.System"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManagePermissions.ManageLayouts.MainSupportLayout"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManagePermissions.ManageLayouts.ComponentSupportLayout
"/>             <channel-ref code="SUPPORT"/>
</permission>         <permission>             <action-ref
code="ManagePermissions.ManageApplication.OperationCatalogAp
p"/>             <channel-ref code="SUPPORT"/>
</permission>         <permission>             <action-ref
code="ManagePermissions.ManageApplication.DashboardApp"/>
<channel-ref code="SUPPORT"/>         </permission>
</role>      <role
code="UfsParamsEmergencyAdminRole_${tenant_code}"
name="Роль Экстренного Администратора платформенного сектора
СУП тенанта ${tenant_code}"           category="ADMIN"
subsystem="PARAMS">         <needRoleSUDIR
name="EFS_EMERGENCY_ADMIN"/>         <permission>
<action-ref code="ManageParameters.Services.Access"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.StartManager.ViewPanel"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.Parameters.View"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.Parameters.View.All"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.Parameters.Edit"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.Parameters.Edit.All"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManageParameters.ParameterRole.System"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
code="ManagePermissions.ManageLayouts.MainSupportLayout"/>
<channel-ref code="SUPPORT"/>         </permission>
<permission>             <action-ref
```

```
code="ManagePermissions.ManageLayouts.ComponentSupportLayout
"/>              <channel-ref code="SUPPORT"/>
</permission>          <permission>              <action-ref
code="ManagePermissions.ManageApplication.OperationCatalogAp
p"/>              <channel-ref code="SUPPORT"/>
</permission>          <permission>              <action-ref
code="ManagePermissions.ManageApplication.DashboardApp"/>
<channel-ref code="SUPPORT"/>          </permission>
</role>  </task>
```

Настройка работы с OTT компонента «Управление параметрами»

Данная версия компонента устанавливается без OTT sidecar. Ниже приведена справочная информация для настройки подключения OTT sidecar в pods Egress/Ingress при необходимости.

1. `global.ott.ose_deploy` - глобальный параметр, отвечающий за включение/отключение установки с OTT Sidecar.
   - `true` - установка приложения должна производиться с OTT Sidecar;
   - `false` - установка приложения должна производиться без OTT Sidecar.

   По умолчанию компонент устанавливается без OTT, параметр `global.ott.ose_deploy` установлен в значение `false`.
2. Параметры для настройки подключения по OTT располагаются в конфигурационном файле `ufsparams_config.istio.all.conf`. Таблица с параметрами и их описанием приведена выше в документе.
3. ConfigMap с настройками OTT Sidecar:

- `ott-egress-cm-unver` настраивается в файле `ott-egress-cm.yaml` дистрибутива;
- `ott-ingress-cm-unver` настраивается в файле `ott-ingress-cm.yaml`дистрибутива.

ConfigMap для Egress `ott-egress-cm-unver`:

```
OTT_OPER_MODE: sign
#<mode> = "sign" для egress GW, "validate" - для ingress GW
OTT_CERTSTORE_TYPE: ${ufsparams_config.ott.store.type}
#JKS                                OTT_SERVICE_CERT_ALIAS:
${ufsparams_config.ose.egress.ott.service.cert.alias} #ott-
service   OTT_AUTHZ_REALM: ott   OTT_GRPC_PORT:
${ufsparams_config.ose.egress.ott.grpc.port}
#${global.ott.grpc.port}   OTT_SERVICE_URL:
${ufsparams_config.ose.egress.ott.service.url}
#Шаблон URL сервиса OTT, при балансировке значения будут
подставляться из OTT_SERVICE_HOSTS   OTT_CERTSTORE_PATH:
${ufsparams_config.ose.egress.ott.certstore.path}
```

```
#Путь к keystore с сертификатом и секретным ключом
приложения - /mnt/secrets/ott.jks   OTT_CLIENT_CERT_ALIAS:
${ufsparams_config.ose.egress.ott.client.cert.alias}
#Сертификат в нижнем регистре -
ci02228938_efs_ufsparams_config_{{ SEGMENT_ID | lower }}
OTT_TRUST_STORE_PATH:
${ufsparams_config.ose.egress.ott.trustStorePath}
#Путь к trust store с сертификатом сервиса OTT -
/mnt/secrets/ott.jks   OTT_AUTHZ_VERSION: '1.0'
#Версия метода авторизации   OTT_CLIENT_MMT-COMPATIBILITY-
MODE: 'false'                                    #Режим
совместимости с MMT 3 поколения, включается для
взаимодейтсвия с ППРБv3, ожидаемая структура path http-
запроса: /<receiver_id>/<api.name>/<api.method>  или
/<api.name>/<api.method>   OTT_MODULE_ID:
${ufsparams_config.ose.egress.ott.module.id}
#Идентификатор приложения в OTT с учетом регистра -
ci02228938_efs_ufsparams_config_{{ SEGMENT_ID | lower }}
OTT_CLIENT_DEFAULT_REALM: "mmt"   OTT_SERVICE_HOSTS:
${ufsparams_config.ose.egress.ott.service.hosts}
#Список серверов:портов сервиса OTT для балансировки
нагрузки, или аналогичный список nginx OTT (на НТ/ПСИ/Пром)
- ${global.ott.service.hosts}
OTT_ANONYMOUS_REQUESTS_ENABLED: 'true'
OTT_CLIENT_MMT_RESOURCE_ATTRID:
'urn:sbrf:names:pprb:1.0:api:interface:fullname'
OTT_CLIENT_MMT_ACTION_ATTRID:
'urn:sbrf:names:pprb:1.0:action:id'
OTT_APPLICATION_ATTRIBUTE_ID:
'urn:sbrf:names:pprb:1.0:module:id'
```

ConfigMap для Ingress `ott-ingress-cm-unver`:

```
OTT_MODULE_ID: ${ufsparams_config.ose.ingress.ott.module.id}
#Идентификатор приложения в OTT с учетом регистра -
ci02228938_efs_ufsparams_config_{{ SEGMENT_ID | lower }}
OTT_CLIENT_CERT_ALIAS:
${ufsparams_config.ose.ingress.ott.client.cert.alias}
#Сертификат в нижнем регистре -
ci02228938_efs_ufsparams_config_{{ SEGMENT_ID | lower }}
OTT_OPER_MODE: validate
#<mode> = "sign" для egress GW, "validate" - для ingress GW
OTT_CERTSTORE_PATH:
${ufsparams_config.ose.ingress.ott.certstore.path}
#Путь к keystore с сертификатом и секретным ключом
приложения - /mnt/secrets/ott.jks   OTT_SERVICE_HOSTS:
${ufsparams_config.ose.ingress.ott.service.hosts}
#Список серверов:портов сервиса OTT для балансировки
нагрузки, или аналогичный список nginx OTT (на НТ/ПСИ/Пром)
```

```
- ${global.ott.service.hosts}   OTT_TRUST_STORE_PATH:
${ufsparams_config.ose.ingress.ott.trustStorePath}
#Путь к trust store с сертификатом сервиса OTT -
/mnt/secrets/ott.jks   OTT_ANONYMOUS_REQUESTS_ENABLED:
'true'   OTT_CLIENT_MMT-COMPATIBILITY-MODE: 'false'
#Режим совместимости с MMT 3 поколения, включается для
взаимодейтсвия с ППРБv3, ожидаемая структура path http-
запроса: /<receiver_id>/<api.name>/<api.method>  или
/<api.name>/<api.method>   OTT_GRPC_PORT:
${ufsparams_config.ose.ingress.ott.grpc.port}
#${global.ott.grpc.port}   OTT_SERVICE_URL:
${ufsparams_config.ose.ingress.ott.service.url}
#Шаблон URL сервиса OTT, при балансировке значения будут
подставляться из OTT_SERVICE_HOSTS   OTT_CERTSTORE_TYPE:
${ufsparams_config.ott.store.type}
#JKS   OTT_SERVICE_CERT_ALIAS:
${ufsparams_config.ose.ingress.ott.service.cert.alias}
#ott-service   OTT_AUTHZ_VERSION: '1.0'
#Версия метода авторизации   OTT_AUTHZ_REALM: ott
OTT_CLIENT_DEFAULT_REALM: "mmt"
OTT_CLIENT_MMT_RESOURCE_ATTRID:
'urn:sbrf:names:pprb:1.0:api:interface:fullname'
OTT_CLIENT_MMT_ACTION_ATTRID:
'urn:sbrf:names:pprb:1.0:action:id'
OTT_APPLICATION_ATTRIBUTE_ID:
'urn:sbrf:names:pprb:1.0:module:id'
```

ConfigMap для конфигурирования подключения к удаленному сервису журналирования и мониторинга для Sidecar OTT `ott-jm-cm.${distrib.release.version}`:

```
apiVersion: "v1" kind: "ConfigMap" metadata:   name: ott-jm-
cm.${distrib.release.version} data:   ufs-
monitoring.kafka.bootstrap.servers:
'${global.kafka.monitoring.bootstrap.servers|}'   ufs-
monitoring.kafka.topic.metrics:
'${global.kafka.monitoring.metrics.topic|}'   ufs-
monitoring.kafka.topic.settings:
'${global.kafka.monitoring.settings.topic|}'   ufs-
monitoring.kafka.topic.metadata:
'${global.kafka.monitoring.metadata.topic|}'   ufs-
monitoring.kafka.topic.filter:
'${global.kafka.monitoring.filter.topic|}'   ufs-
monitoring.kafka.security.protocol:
'${global.platform.kafka.security.protocol|}'   ufs-
monitoring.kafka.ssl.endpoint.identification.algorithm:
'${global.platform.kafka.ssl.endpoint.identification.algorit
hm|}'   ufs-monitoring.kafka.ssl.enabled.protocols:
```

```
'${global.platform.kafka.ssl.enabled.protocols|}'   ufs-
monitoring.kafka.producer_factory.delivery.timeout.ms:
'${global.platform.kafka.producer.delivery.timeout.ms|}'
ufs-monitoring.kafka.producer_factory.batch.size:
'${global.platform.kafka.producer.batch.size|}'   ufs-
monitoring.kafka.producer_factory.linger.ms:
'${global.platform.kafka.producer.linger.ms|}'   ufs-
monitoring.kafka.producer_factory.buffer.memory:
'${global.platform.kafka.producer.buffer.memory|}'   ufs-
monitoring.kafka.producer_factory.request.timeout.ms:
'${global.platform.kafka.producer.request.timeout.ms|}'
ufs-monitoring.kafka.producer_factory.send.buffer.bytes:
'${global.platform.kafka.producer.send.buffer.bytes|}'
ufs-monitoring.kafka.producer_factory.receive.buffer.bytes:
'${global.platform.kafka.producer.receive.buffer.bytes|}'
ufs-monitoring.kafka.producer_factory.max.request.size:
'${global.platform.kafka.producer.max.request.size|}'   ufs-
monitoring.kafka.producer_factory.metadata.max.age.ms:
'${global.platform.kafka.producer.metadata.max.age.ms|}'
ufs-
monitoring.kafka.producer_factory.connections.max.idle.ms:
'${global.platform.kafka.producer.connections.max.idle.ms|}'
ufs-monitoring.kafka.producer_factory.max.block.ms:
'${global.platform.kafka.producer.max.block.ms|}'   ufs-
monitoring.kafka.producer_factory.compression.type:
'${global.platform.kafka.producer.compression.type|}'   ufs-
monitoring.kafka.consumer_factory.session.timeout.ms:
'${global.platform.kafka.consumer.session.timeout.ms|}'
ufs-monitoring.kafka.consumer_factory.fetch.max.bytes:
'${global.platform.kafka.consumer.fetch.max.bytes|}'   ufs-
monitoring.kafka.consumer_factory.fetch.max.wait.ms:
'${global.platform.kafka.consumer.fetch.max.wait.ms|}'
ufs-monitoring.kafka.consumer_factory.fetch.min.bytes:
'${global.platform.kafka.consumer.fetch.min.bytes|}'   ufs-
monitoring.kafka.consumer_factory.max.poll.interval.ms:
'${global.platform.kafka.consumer.max.poll.interval.ms|}'
ufs-monitoring.kafka.consumer_factory.heartbeat.interval.ms:
'${global.platform.kafka.consumer.heartbeat.interval.ms|}'
ufs-
monitoring.kafka.consumer_factory.max.partition.fetch.bytes:
'${global.platform.kafka.consumer.max.partition.fetch.bytes|
}'   ufs-
monitoring.kafka.consumer_factory.send.buffer.bytes:
'${global.platform.kafka.consumer.send.buffer.bytes|}'
ufs-monitoring.kafka.consumer_factory.receive.buffer.bytes:
'${global.platform.kafka.consumer.receive.buffer.bytes|}'
ufs-monitoring.kafka.consumer_factory.request.timeout.ms:
'${global.platform.kafka.consumer.request.timeout.ms|}'
ufs-monitoring.validation-jms-config: 'true'   ufs-
monitoring.infra.metrics.period: '20s'   ufs-
```

```
monitoring.jms.enabled:
'${global.monitoring.jms.enabled|true|}'   ufs-
monitoring.queue.monitoringCF.metrics:
'${global.queue.monitoring.baseQueueName|}'   ufs-
monitoring.queue.monitoringCF.metadata:
'${global.queue.monitoringMeta.baseQueueName|}'   ufs-
monitoring.topic.monitoringTCF.filters:
'${global.topic.monitoringFilter.baseQueueName|}'   ufs-
monitoring.wmqcf.monitoringCF.qmgrName:
'${global.wmqcf.monitoringCF.qmgrName|}'   ufs-
monitoring.wmqcf.monitoringCF.qmgrHostname:
'${global.wmqcf.monitoringCF.qmgrHostname|}'   ufs-
monitoring.wmqcf.monitoringCF.qmgrPortNumber:
'${global.wmqcf.monitoringCF.qmgrPortNumber|}'   ufs-
monitoring.wmqcf.monitoringCF.qmgrSvrconnChannel:
'${global.wmqcf.monitoringCF.qmgrSvrconnChannel|}'   ufs-
monitoring.wmqcf.monitoringCF.maxConnections:
'${global.wmqcf.monitoringCF.maxConnections|}'   ufs-
monitoring.wmqcf.monitoringCF.sess.maxConnections:
'${global.wmqcf.monitoringCF.sess.maxConnections|}'   ufs-
monitoring.wmqcf.monitoringCF.connectionTimeout:
'${global.wmqcf.connectionTimeout|}'   ufs-
monitoring.wmqcf.monitoringCF.sslCipherSuite:
'${global.wmqcf.monitoringCF.ose.ssl.enabledCiphers|}'
ufs-monitoring.wmqcf.monitoringCF.agedTimeout:
'${global.wmqcf.agedTimeout|}'   ufs-
monitoring.wmqcf.monitoringCF.reapTime:
'${global.wmqcf.reapTime|}'   ufs-
monitoring.wmqcf.monitoringCF.unusedTimeout:
'${global.wmqcf.unusedTimeout|}'   ufs-
monitoring.wmqcf.monitoringCF.clientReconnectOptions:
'${global.wmqcf.clientReconnectOptions|}'   ufs-
monitoring.wmqcf.monitoringCF.clientReconnectTimeout:
'${global.wmqcf.clientReconnectTimeout|}'   ufs-
monitoring.wmqcf.monitoringCF.stuckTimerTime:
'${global.wmqcf.stuckTimerTime|}'   ufs-
monitoring.wmqcf.monitoringCF.stuckTime:
'${global.wmqcf.stuckTime|}'   ufs-
monitoring.wmqcf.monitoringCF.stuckThreshold:
'${global.wmqcf.stuckThreshold|}'   ufs-
monitoring.wmqcf.monitoringCF.surgeThreshold:
'${global.wmqcf.surgeThreshold|}'   ufs-
monitoring.wmqcf.monitoringCF.surgeCreationInterval:
'${global.wmqcf.surgeCreationInterval|}'   ufs-
monitoring.wmqcf.monitoringCF.reconnectOnException: 'true'
ufs-monitoring.wmqcf.monitoringTCF.qmgrName:
'${global.wmqcf.monitoringTCF.qmgrName|}'   ufs-
monitoring.wmqcf.monitoringTCF.qmgrHostname:
'${global.wmqcf.monitoringTCF.qmgrHostname|}'   ufs-
monitoring.wmqcf.monitoringTCF.qmgrPortNumber:
```

```
'${global.wmqcf.monitoringTCF.qmgrPortNumber|}'   ufs-
monitoring.wmqcf.monitoringTCF.qmgrSvrconnChannel:
'${global.wmqcf.monitoringTCF.qmgrSvrconnChannel|}'   ufs-
monitoring.wmqcf.monitoringTCF.maxConnections:
'${global.wmqcf.monitoringTCF.maxConnections|}'   ufs-
monitoring.wmqcf.monitoringTCF.sess.maxConnections:
'${global.wmqcf.monitoringTCF.sess.maxConnections|}'   ufs-
monitoring.wmqcf.monitoringTCF.connectionTimeout:
'${global.wmqcf.connectionTimeout|}'   ufs-
monitoring.wmqcf.monitoringTCF.agedTimeout:
'${global.wmqcf.agedTimeout|}'   ufs-
monitoring.wmqcf.monitoringTCF.sslCipherSuite:
'${global.wmqcf.monitoringTCF.ose.ssl.enabledCiphers|}'
ufs-monitoring.wmqcf.monitoringTCF.reapTime:
'${global.wmqcf.reapTime|}'   ufs-
monitoring.wmqcf.monitoringTCF.unusedTimeout:
'${global.wmqcf.unusedTimeout|}'   ufs-
monitoring.wmqcf.monitoringTCF.clientReconnectOptions:
'${global.wmqcf.clientReconnectOptions|}'   ufs-
monitoring.wmqcf.monitoringTCF.clientReconnectTimeout:
'${global.wmqcf.clientReconnectTimeout|}'   ufs-
monitoring.wmqcf.monitoringTCF.stuckTimerTime:
'${global.wmqcf.stuckTimerTime|}'   ufs-
monitoring.wmqcf.monitoringTCF.stuckTime:
'${global.wmqcf.stuckTime|}'   ufs-
monitoring.wmqcf.monitoringTCF.stuckThreshold:
'${global.wmqcf.stuckThreshold|}'   ufs-
monitoring.wmqcf.monitoringTCF.surgeThreshold:
'${global.wmqcf.surgeThreshold|}'   ufs-
monitoring.wmqcf.monitoringTCF.surgeCreationInterval:
'${global.wmqcf.surgeCreationInterval|}'   ufs-
monitoring.wmqcf.monitoringTCF.reconnectOnException: 'true'
ufs-logger.kafka.bootstrap.servers:
'${global.platform.logger.kafka.bootstrap.servers|}'   ufs-
logger.kafka.security.protocol:
'${global.platform.logger.kafka.security.protocol|}'   ufs-
logger.kafka.ssl.enabled.protocols:
'${global.platform.kafka.ssl.enabled.protocols|}'   ufs-
logger.kafka.topic: '${global.platform.logger.kafka.topic|}'
ufs-logger.wmqcf.loggerCF.qmgrName:
'${global.wmqcf.LoggerCF.qmgrName|}'   ufs-
logger.wmqcf.loggerCF.qmgrHostname:
'${global.wmqcf.LoggerCF.qmgrHostname|}'   ufs-
logger.wmqcf.loggerCF.qmgrPortNumber:
'${global.wmqcf.LoggerCF.qmgrPortNumber|}'   ufs-
logger.wmqcf.loggerCF.qmgrSvrconnChannel:
'${global.wmqcf.LoggerCF.qmgrSvrconnChannel|}'   ufs-
logger.wmqcf.loggerCF.sslCipherSuite:
'${global.wmqcf.LoggerCF.ose.ssl.enabledCiphers|}'   ufs-
logger.queue.loggerCF.system: '${global.ufs-
```

```
logger.queue.ufsSystemJournalListLoggerQueue.baseQueueName|}
'   ufs-logger.queue.loggerCF.integration: '${global.ufs-
logger.queue.ufsIntegrationJournalListLoggerQueue.baseQueueN
ame|}'   ufs-logger.queue.loggerCF.user: '${global.ufs-
logger.queue.ufsUserJournalListLoggerQueue.baseQueueName|}'
ufs-logger.parameters.url: 'http://egressgateway-svc-pacman-
${distrib.release.version}.${projectName}.svc:7071/ufs-
logger-parameters-ng/parameters'
```

4.Настройка EnvoyFilter.

- для Egress `ott-egress-auth-filter`:

```
{% if global.ott.ose_deploy == true %} kind: EnvoyFilter
apiVersion: networking.istio.io/v1alpha3 metadata:   name:
ott-egress-auth-filter-${distrib.release.version} spec:
workloadLabels:     app: istio-egressgateway-pacman-
${distrib.release.version}     projectName: ${projectName}
configPatches:     - applyTo: HTTP_FILTER       match:
context: GATEWAY         listener:           filterChain:
filter:               name: envoy.http_connection_manager
portNumber:
${ufsparams_config.ose.istio.egress.common.internal-port}
patch:         operation: INSERT_BEFORE         value:
name: envoy.ext_authz           config:
failure_mode_allow: false             grpc_service:
google_grpc:                  stat_prefix: ext_authz
target_uri: 'unix:/mnt/ott-uds-socket/ott.socket'
timeout: 2s             with_request_body:
allow_partial_message: false
max_request_bytes: 2097152 {% endif %}
```

- для Ingress `ott-ingress-auth-filter`:

```
{% if global.ott.ose_deploy == true %} kind: EnvoyFilter
apiVersion: networking.istio.io/v1alpha3 metadata:   name:
ott-ingress-auth-filter-${distrib.release.version} spec:
workloadSelector:     app: istio-ingressgateway-pacman-
${distrib.release.version}     projectName: ${projectName}
configPatches:     - applyTo: HTTP_FILTER       match:
context: GATEWAY         listener:           filterChain:
filter:               name: envoy.http_connection_manager
portNumber:
${ufsparams_config.ose.istio.ingress.common.internal-port}
patch:         operation: INSERT_BEFORE         value:
name: envoy.ext_authz           config:
failure_mode_allow: false             grpc_service:
```

```
google_grpc:                    stat_prefix: ext_authz
target_uri: 'unix:/mnt/ott-uds-socket/ott.socket'
timeout: 2s                 with_request_body:
allow_partial_message: false
max_request_bytes: 2097152 {% endif %}
```

5.В качестве механизма авторизации у Envoy sidecar прокси указаны настройки для передачи заголовка ufs-release.

В настройках VirtualService исходящих запросов (файл `vs-egress-mtls-sgw.yaml`):

```
apiVersion: networking.istio.io/v1alpha3 kind:
VirtualService ... spec:     ...       route:         -
destination:              ...           headers:
request:              set:                ufs-release:
R20.1
```

6.Для перенаправления трафика через egress с использованием OTT созданы VS и GW.

VS:

```
apiVersion: networking.istio.io/v1alpha3 kind:
VirtualService metadata:   name: egressgateway-vs-ott-unver
spec:   exportTo:     - .   gateways:     - egressgateway-
ott-unver   hosts:     - egressgateway-svc-pacman-
${distrib.release.version}.${projectName}.svc   http:     -
match:         - gateways:             - egressgateway-ott-
unver           port: 7071       rewrite:         authority:
${global.platform.sgw.route.log.host|}       route:
- destination:             host: egressgateway-svc-pacman-
${distrib.release.version}             port:
number: ${ufsparams_config.ose.istio.egress.common.internal-
port}           weight: 100
```

GW:

```
apiVersion: networking.istio.io/v1alpha3 kind: Gateway
metadata:   name: egressgateway-ott-unver spec:   selector:
istio: egressgateway-ufsparams-unver   servers:     - hosts:
- egressgateway-svc-ufsparams-unver.${projectName}.svc
port:         name: http4         number: 7071
protocol: HTTP
```

Добавлен порт в service egress :

```
kind: Service apiVersion: v1 metadata:   name:
egressgateway-svc-pacman-${distrib.release.version} ...
spec:   ports:                              # Это
минимальный список портов нужных самому Egress. ...     -
name: http       protocol: TCP       port: 7071
targetPort: 7071 ...
```

7.Доработана конфигурация ingress для приема запросов с использованием OTT.

```
apiVersion: networking.istio.io/v1alpha3 kind:
VirtualService metadata:   name: ingress-vs-mtls-pacman-
${distrib.release.version} spec:   ...   http:     - match:
- uri:             prefix: /ufsparams-import-v2/
route:         - destination:             host: ufs-params-
import-unver             port:               number: 8080
- match:         - uri:             prefix:
/healthcheck/service-healthcheck/UFSPARAMS_IMPORT_20
name: ufsparams_import_healthcheck       route:         -
destination:             host: ufs-params-import-unver     #
Хост назначения запроса (service POD'a);             port:
number: 8080     - match:         - uri:             prefix:
/ufs-params-manager-v2/       route:         - destination:
host: ufs-params-manager-unver      # Хост назначения
запроса (service POD'a);             port:
number: 8080     - match:         - uri:             prefix:
/healthcheck/service-healthcheck/UFSPARAMS_ADMIN_20
name: ufsparams_admin_healthcheck       route:         -
destination:             host: ufs-params-manager-unver    #
Хост назначения запроса (service POD'a);             port:
number: 8080
```

8.Развертывание Sidecar OTT.

Для развертывания Sidecar OTT добавлены новые разделы в Deployment ingress и egress.

a.Добавлены новые volumes в Deployment ingress и egress на уровне deployment.spec.template.spec.volumes:

```
kind: Deployment apiVersion: apps/v1 .... spec:   ...
template:     ...     spec:       volumes:         ...
- name: ott-certs           secret:             secretName:
ott-certs.${distrib.release.version}
defaultMode: 420         - name: ott-uds-socket
emptyDir:             medium: Memory         - name: ufs-mq-
jks-vol           secret:             defaultMode: 420
```

```
secretName: mq.jks.${distrib.release.version}         -
name: ott-disable-fs-logging           emptyDir: {}
```

b.Добавлены новые volumeMounts в Deployment ingress и egress на уровне deployment.spec.template.spec.containers.volumeMounts.

```
kind: Deployment apiVersion: apps/v1 .... spec:   ...
template:     ...     spec:       containers:           ...
volumeMounts:             - name: ott-uds-socket
mountPath: /mnt/ott-uds-socket
```

c.В Deployment ingress и egress на уровне deployment.spec.template.spec.containers добавлена конфигурация для развертывания Sidecar OTT.

```
kind: Deployment apiVersion: apps/v1 .... spec:   ...
template:     ...     spec:       containers: {% if
global.ott.ose_deploy == true %}         - name: ott-sidecar
image:
'${ott.ose.istio.deployment.spec.template.spec.containers.ot
t-sidecar.image}'           env:              - name:
SPRING_PROFILES_ACTIVE               value: PROM
- name: OTT_CERTSTORE_PRIVATE_KEY_PWD
valueFrom:                  secretKeyRef:
name: istio-secret-ufsparams-
config.${distrib.release.version}                   key:
ssl.ose.istio.keyStore.egress.password             - name:
OTT_CERTSTORE_PWD               valueFrom:
secretKeyRef:                   name: istio-secret-
ufsparams-config.${distrib.release.version}
key: ssl.ose.istio.keyStore.egress.password             -
name: OTT_TRUST_STORE_PWD               valueFrom:
secretKeyRef:                   name: istio-secret-
ufsparams-config.${distrib.release.version}
key: ssl.ose.istio.keyStore.egress.password
envFrom:             - configMapRef:                  name:
ott-egress-cm.unver             - configMapRef:
name: ott-jm-cm.${distrib.release.version}             -
secretRef:                 name: istio-secret-ufsparams-
config.${distrib.release.version}           resources:
limits:               cpu:
${ufsparams_config.ose.istio.egress.deployment.spec.template
.spec.containers.ott.sidecar.resources.limits.cpu}
memory:
${ufsparams_config.ose.istio.egress.deployment.spec.template
.spec.containers.ott.sidecar.resources.limits.memory}
requests:               cpu:
${ufsparams_config.ose.istio.egress.deployment.spec.template
```

```
.spec.containers.ott.sidecar.resources.requests.cpu}
memory:
${ufsparams_config.ose.istio.egress.deployment.spec.template
.spec.containers.ott.sidecar.resources.requests.memory}
readinessProbe:              httpGet:                path:
/actuator/health               port: 8090
scheme: HTTP             initialDelaySeconds: 3
timeoutSeconds: 10             periodSeconds: 1
successThreshold: 1             failureThreshold: 30
volumeMounts:             - name: ott-certs
readOnly: true               mountPath: /mnt/secrets
- name: ott-uds-socket                mountPath: /mnt/ott-
uds-socket             - name: ufs-mq-jks-vol
mountPath: '/etc/config/ssl/'               readOnly: true
- name: ott-disable-fs-logging               mountPath:
'/home/jboss/logs'               readOnly: true
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File           imagePullPolicy:
Always {% endif %} ...
```

9.Настройка паролей для хранилищ сертификатов.

Параметр с паролями от хранилищ и сертификата добавлен в distrib.yaml дистрибутива.

```
istio-secrets: [
"ssl.ose.istio.keyStore.egress.password",
"ssl.ose.keyStore.mq.password" ]
```

В раздел `Secrets` в secret `istio-secret-ufsparams-config.${distrib.release.version}` добавляется пароль от хранилищ доверительных сертификатов и ключей для работы с OTT из параметра `ssl.ose.istio.keyStore.egress.password`.

В Environment Variables OTT Sidecar монтируются 3 новые переменные окружения, содержащие пароли для аутентификации/авторизации OTT Sidecar:

- OTT_CERTSTORE_PRIVATE_KEY_PWD;
- OTT_CERTSTORE_PWD;
- OTT_TRUST_STORE_PWD.

Значение переменных `secret: istio-secret-ufsparams-config.${distrib.release.version}/ssl.ose.istio.keyStore.egress.password`.

10.Конфигурация для установки `secrets` с сертификатами для OTT.

В раздел Secrets из дистрибутива размещается конфигурационный файл ott-certs.${distrib.release.version}, который содержит хранилище сертификата:

```
kind: Secret apiVersion: v1 metadata:   name: ott-
certs.${distrib.release.version} type: Opaque data:
ott.jks: >-     {{ lookup('binto64',
configCommon.currentName + '/' + ottCertPath |
default(vars['ssl.ose.istio.keyStore.egress.KeyStoreFromFile
'])) }}
```

Все сертификаты находятся в хранилище сертификатов egress/ingress на стенде в репозитории common.

11.Настройка уровней логирования.

Способ записи логов и уровни логирования настраиваются через ConfigMap ott-jm-cm.${distrib.release.version}.

Изменить уровень логирования для sidecar OTT возможно через APM Журналирования.

12.Инструкция для получения сертификата OTT для стендов тестирования.

Для OTT должны быть использованы индивидуальные сертификаты для каждого сервиса и полигона. Сертификаты подписываются удостоверяющим центром. Все сертификаты находятся в хранилище сертификатов egress на стенде в репозитории common. За выпуск и хранение индивидуальных сертификатов для ПРОМ среды отвечают сотрудники сопровождения.

Для выпуска сертификата OTT необходимо:

1. Провести генерацию ключевой пары:

   ```
   keytool -genkey -keyalg EC -sigalg SHA256withECDSA -
   keystore ${module_id}_{{ SEGMENT_ID }}.jks -storetype JKS -
   keysize 256 -dname "CN=${module_id}_{{ SEGMENT_ID }}" -
   alias ${module_id}__{{ SEGMENT_ID }}
   ```

2. Создать запрос на сертификат:

   ```
   keytool -certreq -keyalg EC -sigalg SHA256withECDSA -
   keystore ${module_id}_{{ SEGMENT_ID }}.jks -storetype JKS -
   alias ${module_id}_{{ SEGMENT_ID }} > ${module_id}__{{
   SEGMENT_ID }}_cert_req.pem
   ```

3. Подписать сертификаты с помощью Удостоверяющего Центра. Получить корневые сертификаты.
4. Полученные сертификаты импортировать в ранее созданное хранилище:

```
keytool -import -keystore ${module_id}_{{ SEGMENT_ID }}.jks
-storetype JKS -file SberbankPlatformCA_EC.pem -alias
SberbankPlatformCA_EC

keytool -import -keystore ${module_id}_{{ SEGMENT_ID }}.jks
-storetype JKS -file ${module_id}_{{ SEGMENT_ID }}.pem -
alias ${module_id}_{{ SEGMENT_ID }}
```

5. Разместить сертификат и private_key в хранилище Egress.

Рекомендуется использовать версию JDK не ниже 1.8 (содержит `keytool`), `<module-id>_{{ SEGMENT_ID }}` указывать в нижнем регистре.

Настройка подключения Агента конфигурации (для доставки конфигурации в ConfigMap)

Описание работы Агента

Агент конфигурации (Configuration Agent) работает, как с компонентом «Управление параметрами», так и с компонентом «Управление конфигурациями».

Configuration Agent устанавливается в Namespace проекта потребителя конфигурации в отдельный pod и позволяет обновлять объекты ConfigMap в Namespace проекта в соответствии с настройкой из аннотаций объекта ConfigMap. Так же в его функции входит отслеживание состояния ConfigMap, связанного с изменением конфигурации приложения, и отправка уведомления в API конфигурации о наличии конфликта (ручного изменения конфигурации). Отслеживание изменений происходит по проверке hash code данных объекта ConfigMap.

Для работы с Агентом конфигурации по обновлению ConfigMap, необходимо:

1. Наличие ConfigMap с параметрами Приложения определенного вида, данный ConfigMap будет отслеживаться Агентом конфигурации.
2. Для КМ компонента «Управление параметрами» необходимо настроить папку для вычитывания файла конфигурации.
3. Настроить ролевую модель для Агента конфигурации, установить Агента, либо проверить, что Агент конфигурации установлен в ваш NameSpace в отдельный pod.

Для того, чтобы Агент отслеживал и обновлял ConfigMap, необходимо, чтобы ConfigMap с конфигурацией был определенного вида.

API Сервиса получения конфигурации предназначено для предоставления конфигурации по запросу Configuration Agent.

Подробнее о макете и правильности заполнения данных ConfigMap для работы с Агентом конфигурации, настройках самого Агента конфигурации, настройках маршрутизации передачи запросов обновления конфигурации написано «в

Руководстве разработчика» в пункте «Настройки для работы с Агентом конфигурации (PACMAN Агент)».

Установка Агента конфигурации

Для установки Агента конфигурации необходимо выполнить следующие шаги:

1.Создать Service Account `config-manager`.

Configuration Agent должен иметь Service Account с ролью, позволяющей читать и обновлять объекты ConfigMap в Namespace, в который он установлен.

Скрипт для создания Service Account через импорт yaml-файла:

```
kind: ServiceAccount apiVersion: v1 metadata:   name:
config-manager imagePullSecrets:   - name: registry-pull
```

2.Создать Role `configmap-role`.

Скрипт для создания Role для работы с ConfigMap, через импорт yaml-файла:

```
# configmap-role.yaml kind: Role apiVersion:
rbac.authorization.k8s.io/v1 metadata:   name: configmap-
role rules:   - verbs:       - get       - watch       -
list       - update       - patch     apiGroups:       - ''
resources:       - configmaps     resourceNames:       -
'my-config'
```

При необходимости можно ограничить доступ к ресурсам не только по типам, но и по именам configmap через `resourceNames`.

Роль дает права на управление объектом ConfigMap для доставки конфигурации.

- `get`, `list`, `watch`: разрешения предназначены для чтения и отслеживания объекта;
- `update`, `patch`: разрешения предназначены для обновления существующего объекта, в соответствии с выбранным способом доставки конфигурации.

3.Добавить RoleBinding между аккаунтом и ролью.

Скрипт для создания RoleBinding связки ServiceAccount с Role, через импорт yaml-файла:

```
kind: RoleBinding apiVersion: rbac.authorization.k8s.io/v1
metadata:   name: config-manager-configmap-role subjects:
- kind: ServiceAccount     name: config-manager
namespace: <namespace> roleRef:   apiGroup:
```

```
rbac.authorization.k8s.io   kind: Role   name: configmap-
role
```

4.Убедиться, что Service Account `config-manager` может скачать образ Агента конфигурации.

5.Установить Агента конфигурации вручную или с помощью инструментов DevOps.

Ручная установка Агента

Данные предоставлены с учетом раскатки Агента «руками», без использования Pipeline DevOps, при этом необходимо предварительно подставить конкретные данные, вместо параметризованных значений. Параметризованные значения указаны в файлах в пункте «Миграция конфигурационных файлов».

1.Добавить ConfigMaps и secrets из файлов.

a.Добавить ConfigMap `pacman-agent.yaml` из дистрибутива в директории `conf/k8s/base/pacman_agent/configmaps/`:

```
apiVersion: v1 kind: ConfigMap metadata:   annotations:
pacman.sbt.ru/id:
'${jenkins_env.fp_artifact_version},${projectName},pacman-
agent'     pacman.sbt.ru/rn: ${ufs.tenant.code}
pacman.sbt.ru/selector:
${ufsparams_config_pacman_agent.configuration.selector}
reloader.stakater.com/match:
${ufsparams_config_pacman_agent.configuration.reloader.match
}   labels:     pacman.sbt.ru/managed:
${ufsparams_config_pacman_agent.configuration.managed}
agent: pacman   name: cfga-cm-pacman-agent-{{
vars['distrib.release.version'] }} data:   pacman-agent-
${projectName}--${jenkins_env.fp_artifact_version}.yml: |
spring:       jackson:         serialization:
indent_output: true        application:         name: agent-
app       profiles:         active:
${ufsparams_config_pacman_agent.spring.active.profiles}
withoutEndpoints: true      server:       port: 8888
servlet:         context-path: /pacman-agent       ssl:
enabled: false      dependenciesInfoFileName: 'dependencies'
ufs:       platform:         config:           deployment-
unit:
${ufsparams_config_pacman_agent.platform.config.deployment-
unit}           agent:             disabled: true
environment:           product:             subsystem:
${ufsparams_config_pacman_agent.platform.environment.product
.subsystem}             channel:
${ufsparams_config.platform.environment.product.channel}
```

```
version: {{ vars['distrib.release.version'] }}
api-version:
${ufsparams_config_pacman_agent.platform.environment.product
.api-version}             release:
${ufsparams_config.platform.environment.product.release}
distrib-version: ${jenkins_env.fp_artifact_version}
deploymentUnit:
${ufsparams_config_pacman_agent.platform.environment.product
.deploymentUnit}             platform: 'not used'
monitoring:         client:           channel:
${ufsparams_config.ufs.monitoring.client.channel}
subsystem:
${ufsparams_config_pacman_agent.ufs.monitoring.client.subsys
tem}           title:
${ufsparams_config_pacman_agent.ufs.monitoring.client.title}
pacman:       adapters:         monitoring:
unimon:
${ufsparams_config.pacman.adapters.monitoring.unimon}
stub: ${ufsparams_config.pacman.adapters.monitoring.stub}
ru:       sbt:          pacman:           agent:
base:               uri:
${ufsparams_config_pacman_agent.config_storage.baseurl}
poll:               delay:
${ufsparams_config_pacman_agent.delay}             http:
timeout:
${ufsparams_config_pacman_agent.config_storage.http.timeout}
circuitBreaker:                 enable:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.en
able}                 failureRate:
threshold:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.fa
ilureRate.threshold}                openState:
duration:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.op
enState.duration}                 halfOpen:
requestBuffer:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.ha
lfOpen.requestBuffer}                minimumCalls:
size:
${ufsparams_config_pacman_agent.httpclient.circuitBreaker.mi
nimumCalls.size}      logging:       level:         root:
INFO       file:         path: /fluent-bit/etc/logs
config: /config-log/logback-spring.xml     management:
server:         port:
${ufsparams_config_pacman_agent.management.server.port}
endpoints:         web:           exposure:
include: "health, prometheus"       endpoint:
health:           group:             readiness:
include: "*"             liveness:              include:
"*"
```

б.Добавить ConfigMap `pacman-logback-spring.yaml`:

```
apiVersion: "v1" kind: "ConfigMap" metadata:   name: cfga-
cm-logback-spring-{{ vars['distrib.release.version'] }}
data:   logback-spring.xml: |-     <?xml version="1.0"
encoding="UTF-8"?>     <configuration>
<springProperty scope="context" name="LOG_PATH"
source="logging.file.path"/>          <appender name="JSON"
class="ch.qos.logback.core.rolling.RollingFileAppender">
<file>{{ '$'+'{'+'LOG_PATH' + '}' }}/log.json</file>
<encoder
class="net.logstash.logback.encoder.LoggingEventCompositeJso
nEncoder">                 <providers>
<timestamp>
<fieldName>serverEventDatetime</fieldName>
<pattern>[UNIX_TIMESTAMP_AS_NUMBER]</pattern>
</timestamp>                      <logLevel>
<fieldName>levelStr</fieldName>
</logLevel>                     <logLevelValue>
<fieldName>levelInt</fieldName>
</logLevelValue>                        <loggerName>
<fieldName>loggerName</fieldName>
</loggerName>                        <threadName>
<fieldName>threadName</fieldName>
</threadName>                        <callerData>
<classFieldName>callerClass</classFieldName>
<methodFieldName>callerMethod</methodFieldName>
<lineFieldName>callerLine</lineFieldName>
<fileFieldName>callerFile</fileFieldName>
</callerData>                        <message/>
<stackTrace>
<fieldName>stackTrace</fieldName>
<throwableConverter
class="net.logstash.logback.stacktrace.ShortenedThrowableCon
verter">
<maxDepthPerThrowable>30</maxDepthPerThrowable>
<maxLength>4096</maxLength>
<rootCauseFirst>true</rootCauseFirst>
</throwableConverter>                        </stackTrace>
<pattern>
<omitEmptyFields>true</omitEmptyFields>
<pattern>                              {
"journalType": "%mdc{journalType}"
}                          </pattern>
</pattern>                          <nestedField>
<fieldName>mdc</fieldName>
<providers>                                <mdc/>
</providers>                        </nestedField>
</providers>              </encoder>
```

```
<rollingPolicy
class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPo
licy">                <fileNamePattern>{{
'$'+'{'+'LOG_PATH' + '}' }}/log_%d{yyyy-MM-
dd}_%i.json</fileNamePattern>
<maxFileSize>5MB</maxFileSize>
<maxHistory>5</maxHistory>
<totalSizeCap>50MB</totalSizeCap>
</rollingPolicy>         </appender>          <appender
name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
<encoder>                <pattern>%yellow(%d{HH:mm:ss.SSS})
%highlight([%-5level]) %cyan([%-36.36logger{36}])
%magenta([%thread]) - %highlight(%msg%n)</pattern>
</encoder>         </appender>          <root level="info">
<appender-ref ref="STDOUT"/>            <appender-ref
ref="JSON"/>         </root>     </configuration>
```

c.Добавить Secret `mq-jks.yaml`:

```
# Сертификаты для подключения # путь к хранилищу {% set
jks_file = configCommon.currentName + '/' + (mqKeyStoreFile
|
default(lookup("vars","ssl.ose.keyStore.mq.keyStoreFromFile"
))) %} # имя переменной с паролем от хранилища {% set
dec_password_vairable = mqKeyStorePass |
default("ssl.ose.keyStore.mq.password") %} # имя переменной
с паролем которым зашифруем итоговое хранилище {% set
enc_password_vairable = "ssl.ose.keyStore.mq.password" %} #
алиас сертификата {% set cert_alias = mqCertAlias |
default(lookup("vars","ssl.ose.keyStore.mq.CertAlias")) %} #
алиас приватного ключа сертификата {% set private_key_alias
= mqPrivateKeyAlias |
default(lookup("vars","ssl.ose.keyStore.mq.CertAlias")) %} #
алиас root сертификата {% set root_cert_alias =
mqRootCertAlias |
default(lookup("vars","ssl.ose.istio.keyStore.RootCertAlias"
)) %} {% set crt = lookup('jks', jks_file,
password_variable_name=dec_password_vairable,
alias=cert_alias, type="certificate") %} {% set key =
lookup('jks', jks_file,
password_variable_name=dec_password_vairable,
alias=private_key_alias, type="key") %} {% set root_crt =
lookup('jks', jks_file,
password_variable_name=dec_password_vairable,
alias=root_cert_alias, type="certificate") %} kind: Secret
apiVersion: v1 metadata:   name: cfga-secret-mq.jks.{{
vars['distrib.release.version'] }} type: Opaque data:
mq.jks: >-     {{ lookup('pack_jks', alias=cert_alias,
root_alias=root_cert_alias,
```

```
password_variable_name=enc_password_vairable, cert_64=crt,
key_64=key, root_64=root_crt) }}
```

2.Создать VirtualService с помощью файла `vs-egress-mtls-config-storage.yaml` (расположение файла в дистрибутиве: `conf/k8s/base/istio/config/egress`).

При этом необходимо заменить `egressgateway-svc-pacman-${distrib.release.version}` на имя сервиса Egress.

```
apiVersion: networking.istio.io/v1alpha3 kind:
VirtualService metadata:   name: egressgateway-mtls-pacman-
config-storage-${distrib.release.version} spec:   exportTo:
- .   gateways:     - egressgateway-mtls-pacman-
${distrib.release.version}     - mesh   hosts:     -
${ufsparams_config.ose.istio.egress.common.config-storage-
host}   http:     - match:         - gateways:             -
mesh           port:
${ufsparams_config.ose.istio.egress.common.internal-port}
rewrite:         authority:
${ufsparams_config.ose.istio.egress.common.config-storage-
host}       route:         - destination:             host:
egressgateway-svc-pacman-${distrib.release.version}
port:               number:
${ufsparams_config.ose.istio.egress.common.internal-port}
- match:         - gateways:             - egressgateway-
mtls-pacman-${distrib.release.version}           port:
${ufsparams_config.ose.istio.egress.common.internal-port}
rewrite:         authority:
${ufsparams_config.ose.istio.egress.common.config-storage-
host}       route:         - destination:             host:
${ufsparams_config.ose.istio.egress.common.config-storage-
host}             port:               number:
${ufsparams_config.ose.istio.egress.common.config-storage-
port}
```

3.Добавить остальные настройки istio из файлов:

a. `se-egress-mtls-config-storage.yaml` (расположение файла в дистрибутиве: `conf/k8s/base/istio/config/egress`):

```
apiVersion: networking.istio.io/v1alpha3 kind: ServiceEntry
metadata:   name: egressgateway-mtls-pacman-config-storage-
${distrib.release.version} spec:   exportTo:     - .
hosts:     -
${ufsparams_config.ose.istio.egress.common.config-storage-
host}   location: MESH_EXTERNAL   ports:     - name: port-
```

```
config-storage-external       number:
${ufsparams_config.ose.istio.egress.common.config-storage-
port}       protocol:
${ufsparams_config.ose.istio.egress.common.config-storage-
protocol}     - name: http-default-config-storage
number: ${ufsparams_config.ose.istio.egress.common.internal-
port}       protocol: HTTP   resolution: DNS
```

b. se-api-ose.yaml (расположение файла в дистрибутиве: conf/k8s/base/istio/config/egress):

```
apiVersion: networking.istio.io/v1alpha3 kind: ServiceEntry
metadata:   name: api-ose-${distrib.release.version} spec:
addresses:     - ${ufsparams_config.ose.istio.egress.api-
ose.addresses}   endpoints:     - address:
${ufsparams_config.ose.istio.egress.api-ose.endpoints}
exportTo:     - .   hosts:     -
${ufsparams_config.ose.istio.egress.api-ose.host}
location: MESH_EXTERNAL   ports:     - name: api-tcp
number: ${ufsparams_config.ose.istio.egress.api-ose.port}
protocol: ${ufsparams_config.ose.istio.egress.api-
ose.protocol}   resolution: STATIC
```

c. gw-egress-mtls.yaml (расположение файла в дистрибутиве: conf/k8s/base/istio/config/egress):

```
apiVersion: networking.istio.io/v1alpha3 kind: Gateway
metadata:   name: egressgateway-mtls-pacman-
${distrib.release.version} spec:   selector:     istio:
egressgateway-pacman-${distrib.release.version}   servers:
- hosts:         - '*' # Необходимо указать host Сервиса
доставки конфигурации соответствует значению
${ufsparams_config.ose.istio.egress.common.config-storage-
host}       port:         name: http-default         number:
${ufsparams_config.ose.istio.egress.common.internal-port}
protocol: HTTP
```

4.Установить Агента конфигурации с помощью файла dc.yaml (расположение файла в дистрибутиве: conf/k8s/base/pacman_agent):

```
apiVersion: apps/v1 kind: Deployment metadata:   name:
pacman-agent-{{ vars['distrib.release.version'] }}
annotations:     description: Pacman Agent   labels:
app: cfga-pacman-agent-{{ vars['distrib.release.version'] }}
agent: pacman spec:   replicas:
```

```
${ufsparams_config_pacman_agent.openshift.autoscaling.min_re
plicas}   revisionHistoryLimit: 2   selector:
matchLabels:       app: pacman-agent-{{
vars['distrib.release.version'] }}   strategy:     type:
RollingUpdate     rollingUpdate:       maxUnavailable: 25%
maxSurge: 25%   template:     metadata:       name: pacman-
agent-{{ vars['distrib.release.version'] }}
annotations:         description: pacman-agent service
sidecar.istio.io/inject: "true"
sidecar.istio.io/proxyCPU:
${ufsparams_config_pacman_agent.ose.deployment.metadata.anno
tations.sidecar.istio.io.proxyCPU}
sidecar.istio.io/proxyMemory:
${ufsparams_config_pacman_agent.ose.deployment.metadata.anno
tations.sidecar.istio.io.proxyMemory}
sidecar.istio.io/proxyCPULimit:
${ufsparams_config_pacman_agent.ose.deployment.metadata.anno
tations.sidecar.istio.io.proxyCPULimit}
sidecar.istio.io/proxyMemoryLimit:
${ufsparams_config_pacman_agent.ose.deployment.metadata.anno
tations.sidecar.istio.io.proxyMemoryLimit}
reloader.stakater.com/search:
${ufsparams_config_pacman_agent.configuration.reloader.searc
h}       labels:         app: pacman-agent-{{
vars['distrib.release.version'] }}         distrib-version:
${jenkins_env.fp_artifact_version}         jenkinsDeployId:
'${jenkinsDeployId}'         deploymentconfig: cfga-pacman-
agent-{{ vars['distrib.release.version'] }}     spec:
containers:         - name: pacman-agent-{{
vars['distrib.release.version'] }}           image:
${ufsparams_config.custom.ose.deployment.spec.template.spec.
containers.image.registry}/efs/ci00428440_efs/ci02228938_ufs
params_config/pacman_agent@${jenkins_env.fp_image_hash}
imagePullPolicy: Always           livenessProbe:
failureThreshold:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.livenessProbe
.failureThreshold}             httpGet:               path:
/actuator/health/liveness               port:
${ufsparams_config_pacman_agent.management.server.port}
scheme: HTTP             initialDelaySeconds:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.livenessProbe
.initialDelaySeconds}             periodSeconds:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.livenessProbe
.periodSeconds}             successThreshold:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.livenessProbe
.successThreshold}             timeoutSeconds:
```

```
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.livenessProbe
.timeoutSeconds}           readinessProbe:
failureThreshold:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.readinessProb
e.failureThreshold}             httpGet:               path:
/actuator/health/readiness               port:
${ufsparams_config_pacman_agent.management.server.port}
scheme: HTTP             initialDelaySeconds:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.readinessProb
e.initialDelaySeconds}             periodSeconds:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.readinessProb
e.periodSeconds}             successThreshold:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.readinessProb
e.successThreshold}             timeoutSeconds:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.readinessProb
e.timeoutSeconds}           envFrom:             -
secretRef:                 name: secret-{{ vars['fpi_name']
| replace('_', '-') | lower }}.{{
vars['distrib.release.version'] }}             -
configMapRef:                name:
ufsparams.config.pacman.agent.conf.{{
vars['distrib.release.version'] }}           env:
- name: POD_NAMESPACE               valueFrom:
fieldRef:                  fieldPath: metadata.namespace
- name: POD_NAME               valueFrom:
fieldRef:                  fieldPath: metadata.name
- name: spring.config.location               value:
/config/pacman-agent-${projectName}--
${jenkins_env.fp_artifact_version}.yml
volumeMounts:             - name: logsshare
mountPath: "/fluent-bit/etc/logs"             - name: ufs-
mq-jks-vol               readOnly: true
mountPath: /etc/config/ssl/             - name: cfg
readOnly: true               mountPath: /config
- name: log               readOnly: true
mountPath: /config-log           ports:             - name:
http               containerPort: 8080             - name:
https               containerPort: 8443             - name:
actuator               containerPort:
${ufsparams_config_pacman_agent.management.server.port}
resources:             limits:               cpu:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.resources.lim
its.cpu}               memory:
```

```
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.resources.lim
its.memory}             requests:               cpu:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.resources.req
uests.cpu}              memory:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.containers.ufsparams_config_pacman_agent.resources.req
uests.memory}           terminationMessagePath:
/dev/termination-log         - name: fluent-bit-sidecar
image:
${ufsparams_config.custom.ose.deployment.spec.template.spec.
containers.image.registry}/efs/ci02587203/ci02809205_tengri/
fluent-
bit@sha256:04da83ed2af92600f7e0b4055d707c038c6e549ed14dba637
2b0a2a50ddae32d           resources:             limits:
memory:
${ufsparams_config.ose.deployment.spec.template.spec.contain
ers.fluent-bit.sidecar.resources.limits.memory}
cpu:
${ufsparams_config.ose.deployment.spec.template.spec.contain
ers.fluent-bit.sidecar.resources.limits.cpu}
requests:               memory:
${ufsparams_config.ose.deployment.spec.template.spec.contain
ers.fluent-bit.sidecar.resources.requests.memory}
cpu:
${ufsparams_config.ose.deployment.spec.template.spec.contain
ers.fluent-bit.sidecar.resources.requests.cpu}
env:             - name: service               value:
pacman-agent             - name: cluster
value: cluster             - name: moduleId
value: pacman-agent             - name: moduleVersion
value: latest            - name: nodeId
value: pacman-agent             - name: standId
value: dev            - name: zoneId               value:
default            - name: pod               valueFrom:
fieldRef:                  apiVersion: v1
fieldPath: metadata.name             - name: namespace
valueFrom:                fieldRef:
apiVersion: v1                  fieldPath:
metadata.namespace            - name:
ssl.ose.keyStore.mq.password               valueFrom:
secretKeyRef:                   key:
ssl.ose.keyStore.mq.password                    name: secret-
{{ vars['fpi_name'] | replace('_', '-') | lower }}.{{
vars['distrib.release.version'] }}          volumeMounts:
- name: logsshare               mountPath: "/fluent-
bit/etc/logs"             - name: fluent-bit
mountPath: "/fluent-bit/etc"            - name: logger-
kafka-cert-vol               readOnly: true
```

```
mountPath: /etc/config/ssl {% if
ufsparams_config_pacman_agent.ose.feature.reloader.enabled
== true %}         - name: reloader-sidecar           image:
${ufsparams_config.custom.ose.deployment.spec.template.spec.
containers.image.registry}/efs/ci00428440_efs/ci02228938_ufs
params_config/reloader@sha256:8c51bd7bcc50ab5e79343601589ee3
63435a15dca4768620cece818101d7d632           resources:
limits:               memory:
${ufsparams_config.ose.deployment.spec.template.spec.contain
ers.reloader.sidecar.resources.limits.memory}
cpu:
${ufsparams_config.ose.deployment.spec.template.spec.contain
ers.reloader.sidecar.resources.limits.cpu}
requests:               memory:
${ufsparams_config.ose.deployment.spec.template.spec.contain
ers.reloader.sidecar.resources.requests.memory}
cpu:
${ufsparams_config.ose.deployment.spec.template.spec.contain
ers.reloader.sidecar.resources.requests.cpu}           env:
- name: KUBERNETES_NAMESPACE               valueFrom:
fieldRef:                   apiVersion: v1
fieldPath: metadata.namespace           lifecycle:
postStart:               httpGet:                path:
/healthz/ready                port: 15021 {% endif %} {% if
ufsparams_config_pacman_agent.ose.feature.SelfAntiAffinity.e
nabled == true %}       affinity:         podAntiAffinity:
preferredDuringSchedulingIgnoredDuringExecution:
- weight: 100               podAffinityTerm:
labelSelector:                    matchExpressions:
- key: app                        operator: In
values:                          - pacman-agent-{{
vars['distrib.release.version'] }}
topologyKey: "kubernetes.io/hostname" {% endif %}
dnsPolicy: ClusterFirst       restartPolicy: Always
securityContext:         readOnlyRootFilesystem: true
terminationGracePeriodSeconds:
${ufsparams_config_pacman_agent.ose.deployment.spec.template
.spec.terminationGracePeriodSeconds}       volumes:
- name: ufs-mq-jks-vol           secret:
defaultMode: 288             secretName: cfga-secret-
mq.jks.{{ vars['distrib.release.version'] }}         - name:
ufsparams-config-volume           emptyDir: {}         -
name: cfg           configMap:             name: cfga-cm-
pacman-agent-{{ vars['distrib.release.version'] }}
defaultMode: 288         - name: log           configMap:
name: cfga-cm-logback-spring-{{
vars['distrib.release.version'] }}            defaultMode:
288         - name: logsshare           emptyDir: { }
- name: fluent-bit           configMap:
defaultMode: 288             name: cfga-cm-fluent-bit-{{
```

```
vars['distrib.release.version'] }}          - name: logger-
kafka-cert-vol           secret:             defaultMode:
288             secretName: cfga-secret-logger-kafka-cert.{{
vars['distrib.release.version'] }}       serviceAccount:
config-manager       serviceAccountName: config-manager
```

[Установка Агента конфигурации с помощью инструментов DevOps (стабильно работает с версии pipeline release/D-01.038.856-432)]

1.Каждый компонент, использующий Агента, должен параметризировать метки своих Deployment egress и ingress:

Пример `package/conf/config/<fp_name>.conf`:

```
# ...
agent.ose.istio.all.common.ingressAppSelector=<application_i
ngress_deployment_name>
agent.ose.istio.all.common.egressAppSelector=<application_eg
ress_deployment_name>
agent.ose.istio.all.common.ingressIstioSelector=<application
_ingress_deployment_name>
agent.ose.istio.all.common.egressIstioSelector=<application_
egress_deployment_name>    # ...
```

В Deployment egress и ingress должны быть ссылки на эти параметры.

Пример `package/conf/k8s/base/istio/deployments/egress/egress-dc.yaml`:

```
kind: Deployment    apiVersion: apps/v1    metadata:      #
...      labels:        app: {{ lookup('vars',
'agent.ose.istio.all.common.egressAppSelector',
default='<application_egress_deployment_name>') }}
istio: {{ lookup('vars',
'agent.ose.istio.all.common.egressIstioSelector',
default='<application_egress_deployment_name>') }}        #
...    spec:      selector:        matchLabels:
app: {{ lookup('vars',
'agent.ose.istio.all.common.egressAppSelector',
default='<application_egress_deployment_name>') }}
istio: {{ lookup('vars',
'agent.ose.istio.all.common.egressIstioSelector',
default='<application_egress_deployment_name>') }}    # ...
```

2.В репозитории с глобальными параметрами стенда, куда будет установлено Приложение-потребителя, в конфигурационном файле `environment.json` установить `useMavenForDownload` и `enableComponentsUI` в значение `true`.

3.В репозитории с глобальными параметрами стенда, куда будет установлено Приложение-потребителя, в конфигурационном файле `susbsystems.json` для компонента добавить секцию `agents`:

```
"agents": { // <- эта часть json файла определяет Агентов
для Приложения-потребителя (можно указать несколько агентов
через запятую)       "PACMAN_AGENT": {         "groupId":
"Nexus_PROD",         "artifactId":
"CI02228938_AS_EFS_UFSPARAMS_CONFIG",         "version":
"<версия дистрибутива компонента PACMAN>",
"fpi_name": "UFSPARAMS_CONFIG",         "include": [
"package/conf/config/parameters/ufsparams_config.all.conf"
"package/conf/config/parameters/ufsparams_config.istio.all.c
onf"
"package/conf/config/parameters/ufsparams_config_pacman_agen
t.conf"           "package/conf/k8s/base/pacman_agent/**/*",
"package/conf/overrides/openshift/pacman_agent/**/*",
"package/conf/k8s/base/configmaps/pacman-fluent-bit.yaml",
"package/conf/k8s/base/configmaps/pacman-logback-
spring.yaml",
"package/conf/k8s/base/secrets/pacman-logger-kafka-
cert.yaml"         ]       }     }
```

4.Произвести реконфигурацию джобы установки компонента (запустить задание на установку без параметров). После реконфигурации в секции `COMPONENTS` будет объявлен Агент конфигурации, помимо основного компонента.

5.Выбрать оба компонента (Агент и PACMAN), а затем произвести миграцию конфигурационных файлов в репозиторий компонента PACMAN (галочка `MIGRATION_FP_CONF`). После миграции помимо конфигурационных файлов компонента PACMAN, в репозитории появятся конфигурационные файлы Агента конфигурации. На этом этапе необходимо их заполнить/сконфигурировать под потребности потребителя.

6.Установить компонент вместе с Агентом конфигурации: аналогично предыдущему шагу выбрать оба компонента (Агент и PACMAN) и выбрать необходимые playbooks по установке. После этих действий Агент будет установлен вместе с компонентом PACMAN.

Настройка плавного перезапуска pod потребителя после изменения конфигурации.

Для возможности плавного перезапуска после изменения конфигурации, требуется права на обновление объектов Deployment или DeploymentConfig.

Роль для работы с ConfigMap.

```
# agent-role.yaml apiVersion: rbac.authorization.k8s.io/v1
kind: Role metadata:   name: agent-role   namespace:
NAMESPACE rules:   - apiGroups:       - ""     resources:
- configmaps     verbs:       - list       - get       -
watch       - update   - apiGroups:       - "apps"
resources:       - deployments       - daemonsets       -
statefulsets     verbs:       - list       - get       -
update       - patch
```

Роль дает права на управление Deployment для осуществления плавного перезапуска pod (rollout):

- `get`, `list`, `watch`: разрешения предназначены для чтения и отслеживания состояния объекта;
- `update`, `patch`: разрешения предназначены для внесения изменения, необходимого для старта плавного перезапуска.

Для настройки автоматического перезапуска pod Приложения-потребителя при изменении Configmap необходимо добавить специальную метку в Deployment.

```
apiVersion: v1 kind: Deployment metadata:   annotations:
reloader.stakater.com/search: true spec:   template:
```

Так же необходимо в Configmap установить метку `reloader.stakater.com/match:` со значением `true`.

```
apiVersion: v1 kind: ConfigMap ...   labels:
pacman.sbt.ru/managed: true    annotations:     ...
reloader.stakater.com/match: true # Метка обновления
Deployment при обновлении ConfigMap
```

Проверка корректности установки Агента конфигурации, создание тестовой ConfigMap.

1. Убедиться, что pod Агента конфигурации поднялся, имеет статус `Ready`.
2. Заменив все `${namespace}` на namespace, в котором подключается Агент, создать тестовую ConfigMap `pacman-agent-test` с помощью файла:

config-map-agent-test.yaml:

```
kind: ConfigMap apiVersion: v1 metadata:   name: pacman-
agent-test.unver   annotations:   pacman.sbt.ru/id:
',${namespace},pacman-agent-test'   pacman.sbt.ru/rn: {{
TENANT_CODE }}   pacman.sbt.ru/selector: configurator-
yaml   labels:   pacman.sbt.ru/managed: 'true' data:
pacman-agent-test--${namespace}.yml: |   group:
inner_group[TEST_GROUP]:           prop1: 1
prop1: 'abc'       inner_group:           prop1: 2
prop1: 'def'
```

3. Через некоторое время (около 10 секунд) убедиться, что в ConfigMap pacman-agent-test появилась заполненная аннотация pacman.sbt.ru/hashcode.
4. Зайти в UI компонента «Управление конфигурациями». Открыть список версий артефакта pacman-agent-test, найти перекрытие с параметрами Узел: ${namespace}, Модуль: pacman-agent-test для версии: ${namespace}.
5. Поменять значение какой-либо настройки.
6. Через некоторое время (около 10 секунд) убедиться, что ConfigMap pacman-agent-test получила новое значение настройки.

Подготовка ACL политик авторизации запросов для работы Агента конфигурации

Для того, чтобы запросы от Агент конфигурации (PACMAN Агент) по обновлению ConfigMap, находящихся в Namespace Потребителя, проходили авторизованными через OTT, необходимо подготовить и установить набор политик авторизации на основании access-листа. Если для OTT не прописаны политики для запросов, то по умолчанию доступ разрешен всем запросам. При наличии прописанных политик запрос, попадающий в access-лист получает по умолчанию разрешение Permit, а запрос, не находящийся в access-листе получает запрет Deny.

1.Для Агента конфигурации, находящегося в Namespace Потребителя необходимо подготовить политики авторизации для всех ConfigMap из Namespace Потребителя.

Определить ConfigMap, которые обновляет Агент конфигурации можно по метке labels: pacman.sbt.ru/managed: true.

2.Политики должны соответствовать следующему формату:

```
{"aclVersions":[     {          "id": %ACL_ID%,
"version": %ACL_VERSION%,          "test": true,
"ACL":     [                     {                  "to":
%RESOURCE_ID%,                    "from":             [
```

```
%CONSUMERS_LIST%              ],                  "rule":
"Permit"             }          ]      } ]}
```

Все значения переменных ниже должны быть заданы в двойных кавычках.

- %ACL_ID% – уникальный идентификатор ACL, для API в 4 поколении необходимо использовать URL-safe идентификатор (шаблон для валидации ^[a-zA-Z0-9!$&'()*+,;=:@~_.-]*$).
- %ACL_VERSION% – версия ACL в формате X.Y.Z.
- %RESOURCE_ID% – идентификатор API, доступ к которому ограничиваем с помощью данного ACL.
- %CONSUMERS_LIST% – список через запятую, идентификатор сервисов потребителей, которым разрешается вызов API, указывается информация о системе-отправителе запроса (её OTT идентификатор), можно получить из http-заголовка ott-invoker протокола OTT (значение параметра OTT_MODULE_ID ConfigMap для OTT sidecar)

Для каждого ConfigMap, который должен обновлять Агент конфигурации необходимо подготовить по 2 ACL политики вида:

a.

```
...        "ACL": [{                  "to":
"/configuration/v2/<значение аннотации
«pacman.sbt.ru/selector»>/rn/<значение аннотации
«pacman.sbt.ru/rn»>/configs/<значение аннотации
«pacman.sbt.ru/id»>», ...
```

b.

```
...        "ACL": [{                  "to":
"/configuration/v2/<значение аннотации
«pacman.sbt.ru/selector»>/rn/<значение аннотации
«pacman.sbt.ru/rn»>/configs/<значение аннотации
«pacman.sbt.ru/id»>:status», ...
```

Пример для API V2

```
{"aclVersions":[     {         "id": "pacman_ott",
"version": "1.1.1",         "test": true,         "ACL":
[             {                 "to":
"/configuration/v2/configurator-
yaml/rn/SECTOR_ID/configs/,,configurator",
"from":             [
```

```
"ci02228938_efs_ufsparams_config_segment_id"            ],
"rule": "Permit"            },            {
"to": "/configuration/v2/configurator-
yaml/rn/SECTOR_ID/configs/,,configurator:status",
"from":            [
"ci02228938_efs_ufsparams_config_segment_id"            ],
"rule": "Permit"            }        ]     } ]}
```

3.Дальнейшая настройка политик авторизации по подготовленному json-файлу осуществляется согласно принятым на стороне заказчика регламентам.

Особенности старта Приложения-потребителя при подключении Агента конфигурации в виде отдельного модуля

Контейнер с Istio Sidecar в Namespace приложения-потребителя должен стартовать перед основным контейнером. В противном случае, будет невозможно обратиться через Istio Proxy к сервису конфигурации для получения актуальных значений параметров конфигурации.

Дополнительные настройки работы с компонентом «Управление конфигурациями»

В файлах `conf/k8s/base/configurator_server/entryPoint.sh` и `conf/k8s/base/configurator_rest/entryPoint.sh` добавлена директива для дополнительной конфигурации запуска приложений, позволяющая добавлять параметры необходимые для инициализации приложения:

- `export JAVA_OPTS="$JAVA_OPTS $ufsparams_config_configurator_server_ose_configmap_javaArguments";`
- `export JAVA_OPTS="$JAVA_OPTS $ufsparams_config_configurator_rest_ose_configmap_javaArguments".`

Инструкция по сборке из разделенного дистрибутива

Pipeline repack

Данный pipeline предназначен для сборки уже загруженных в Nexus-public дистрибутивов из owned- и party- части. Ниже описана инструкция по созданию и использованию job на основе данного pipeline.

Подготовка репозитория

Импортировать в хостинг репозитория исходного кода на основе Git содержимое архива license-resolver или иным образом провести перенос репозитория license-resolver в целевой хостинг репозитория исходного кода.

Создание job

Чтобы создать job, должна быть отдельная папка в Jenkins, в которой будут настроены credentials для вашей технической учетной записи:

- с типом SSH username with private key;
- с типом Username with password.

В данной папке Jenkins необходимо выполнить шаги:

1. Нажать кнопку **New item**, появится страница создания job (проекта).
2. Ввести имя **job**, выбрать тип **Pipeline**.
3. Нажать кнопку **OK**, откроется страница конфигурирования job.
4. В самом низу в разделе **Pipeline** для поля **definition** выбрать вариант **Pipeline script from scm**.
5. Указать в открывшейся форме настройки репозитория адрес pipeline **repack**.
6. В поле **Credentials** указать тот, что с типом **SSH username with private key**.
7. В поле **Script path** указать **repack.groovy**.
8. Нажать кнопку **SAVE**, job создана. Чтобы закончить формирование job, необходимо запустить ее один раз без параметров.

Использование job

Job использует следующие параметры:

- `ARTIFACT_ID`: artifactId дистрибутива, который требуется разделить;
- `VERSION`: версия дистрибутива (загруженного в nexus), которую необходимо разделить;
- `GET_CREDS`: Credentials для технической учетной записи с правами доступа в nexus, где лежит разделенный дистрибутив, на чтение;
- `PUSH_CREDS`: Credentials для технической учетной записи с правами доступа в nexus на запись;
- `DOWNLOAD_URL`: адрес репозитория, в котором лежит разделенный дистрибутив;
- `UPLOAD_URL`: адрес репозитория, в котором нужно разместить объединенный дистрибутив;
- `DOCKER_TOKEN`: credentials для доступа в docker-registry;
- `REGISTRY_URL`: адрес docker-registry;
- `IMAGE_PATH`: расположение готового образа в docker-registry;
- `BASE_IMAGE_URL`: расположение базового образа в docker-registry.

После запуска job выполнит следующие операции:

1. Скачает разделенный дистрибутив.
2. Объединит дистрибутив и выложит его в nexus-public.
3. Соберет Docker-образ и выложит в registry.

Обновление

1. Обновление компонента PACMAN представляет собой установку новой версии релиза через Jenkins.
2. Установку новой версии компонента PACMAN необходимо осуществить согласно описанным шагам в пункте **Установка дистрибутива** текущего документа. Дополнительных настроек не требуется.

Миграция данных для работы в мультитенантной архитектуре для потребителей компонента «Управление конфигурациями», работающих на Wildfly.

1.Миграция данных БД.

Все необходимые скрипты миграции данных БД для работы в мультитенантной архитектуре расположены в основных liquibase скриптах. Для миграции достаточно прокатить основные liquibase скрипты, подробнее в пункте «Запуск liquibase скриптов».

2.Для работы потребителей в мультитенантной архитектуре необходимо в файле настроек wildfly `standalone.xml`, завести параметр:

```
<system-properties>        <property
name="isolation_project" value="название проекта"> </system-
properties>
```

, который должен содержать код (название) проекта.

3.Распределение артефактов по проектам мультитенантной архитектуры.

После установки приложения, для распределения артефактов по проектам в UI, необходимо:

- Выбрать список проектов, к которому будет относится артефакт;
- Осуществить импорт модели;
- Экспортировать нужные настройки.

Удаление

Для удаления всех компонентов PACMAN необходимо:

1. Удалить конфигурационные файлы из репозитория стенда, где производится откат на предудущую версию компонента:
   - ufsparams_config.all.conf;
   - ufsparams_config.istio.all.conf;
   - ufsparams_config_configurator.conf;
   - ufsparams_config_configurator_rest.conf;
   - ufsparams_config_pacman_agent.conf;
   - ufsparams_config_pacman_ui.conf;

- ufsparams_config.agent-sidecar.all.conf (при наличии);
- ufsparams_config_bh.conf;
- ufsparams_config_import.conf;
- ufsparams_config_ms.conf.

2. Удалить PL-компоненты и настройки с группы серверов NGINX.
3. Удалить все ресурсы компонента «Управление конфигурациями» и компонента «Управление параметрами» из Namespace в среде контейнеризации:
   - ConfigMaps;
   - Secrets;
   - Deployment Configs;
   - Services;
   - Pod Ingress/Egress;
   - Pod Агента конфигурации компонента «Управление конфигурациями» (при наличии);
   - Pod APM компонента «Управление конфигурациями» (при наличии);
   - Pod Rest-сервиса компонента «Управление конфигурациями» (при наличии);
   - Pod UI компонента «Управление конфигурациями» (при наличии);
   - Pod Сервиса Импорта компонента «Управление параметрами» (при наличии);
   - Pod APM компонента «Управление параметрами» (при наличии);
   - Pod Сервиса доставки конфигурации компонента «Управление параметрами» (при наличии).

Проверка работоспособности

Проверка работоспособности компонента «Управление конфигурациями»

1. С помощью утилиты curl выполнить следующие проверки корректности установки на сервер Nginx:

   `curl -X POST https://<host:port>/ufs-config-manager/pacman/configurator/cfg/healthcheck` для проверки сервиса API для взаимодействия с PL.

2. Провести проверку работоспособности Агента конфигурации. Подробнее шаги описаны в пункте **Установка Агента конфигурации (Configuration Agent) – Проверка корректности установки, создание тестовой ConfigMap**.

Проверка работоспособности компонента «Управление параметрами»

1. С помощью утилиты curl выполнить следующие проверки корректности установки на сервера приложений, в ответе обратить внимание на параметры: `success: true`, `version: {build-version}`:
   - `curl -X GET http(s)://<host:port>/config-v2/environment/product`;

- `curl -X GET http(s)://<host:port>/ufsparams-import-v2/environment/product`;
- `curl -X GET https://<host:port>/ufs-params-manager-v2/rest/environment/product`.

2. С помощью утилиты curl выполнить следующие проверки корректности установки на Nginx, в ответе обратить внимание на параметры: `success: true`, `body: OK`:
   - `curl -X POST http://<host:port>/paramsv3/healthcheck` для проверки rest-микросервиса чтения конфигурации;
   - `curl -X POST http://<host:port>/ufsparams-import-v3/healthcheck` для проверки rest-микросервиса групповой вставки параметров;
   - `curl -X POST http://<host:port>/ufs-config-manager/pacman/rest/healthcheck` для проверки сервиса API для взаимодействия с PL.
3. Проверка корректности настройки OTT. В Pod Egress и Ingress в логах OTT Sidecar проверяется наличие от потребителей OTT-заголовков:
   - `ott-invoker` со значением сертификата Приложения (на Ingress должен быть, как заголовок нашего приложения `ott-invoker=ci02228938_as_efs_ufsparams_config`, так и заголовки других, обращающихся к нам, Приложений. На Egress должен быть заголовок нашего приложения `ott-invoker=ci02228938_as_efs_ufsparams_config`);
   - `ott-hash`;
   - `ott-token-valid` со значением `true`.

Откат

Для отката на предыдущую версию необходимо:

1. Удалить все компоненты PACMAN согласно пункту Удаление .
2. Установить предыдущую стабильную версию в соответствии с инструкцией, пункт Установка текущего документа.

Часто встречающиеся проблемы и пути их устранения

Часто встречающиеся проблемы и пути их устранения установки компонента «Управление конфигурациями»

1. Приложение установлено, но есть проблемы с заведением пользователей и выдачей им прав.

   Решение:

   а. Проверить наличие подключенного удаленного компонента аутентификации.

b. Проверить в БД записи в таблицах `um_subject` – должен быть вписан логин в столбец `name` в нижнем регистре, столбец `active` должен быть `true`. Должна быть заполнена табличка `um_role_type`. При импорте артефакта должны в табличке `um_role` создаваться записи с именами:

- роль Администратора конфигураций для артефакта `<имя артефакта>`;
- роль Ревизора конфигураций для артефакта `<имя артефакта>` с корректным указанием типа роли и `ID artefact` артефакта в соответствующих столбцах.

c. Проверить, что приложение запущено не в `emergency` режиме.

2. Пользователю выдали права на артефакт, но он не может его редактировать.

   Решение:

   a. Проверить в БД записи в таблицах `um_role`, найти пользователя в таблице `um_subject` и убедиться, что эти роли «приписаны» этому пользователю через таблицы `um_role_group_map` (приписка ролей к группе ролей) и `um_role_group_subject_map` (приписка групп ролей к пользователям).

   b. При наличии возможности удалить артефакт и импортировать его заново, после чего перевыдать права на артефакт.

3. Не работает обновление настроек в runtime режиме.

   Решение: проверить правильность подключения runtime режима по инструкции.

4. Проблемы с импортом модели, редактированием настроек.

   Решение:

   a. Проверить наличие прав на артефакт.

   b. Проверить валидность `xml` файла модели, или `property` файла настроек.

Часто встречающиеся проблемы и пути их устранения установки компонента «Управление параметрами»

1. После установки параметры конфигурации не вычитываются Клиентским модулем у Потребителя, хотя файл с конфигурацией есть на хосте Потребителя и параметр присутствует.

   Проблема может возникнуть из-за некорректного именования файла конфигурации: `DeploymentUnit` (`DU`) в имени файла не соответствует `DeploymentUnit`, указанному в Приложении в теге `core:environment` или в `application.yaml`. Клиентский модуль компонента «Управление параметрами» вычитывает `DU` из тега `core:environment` и пытается найти файл по первой части имени файла (до .__) по маске `${application.deploymentUnit}.__${sup2_scopeTemplate}.json`.

2. Обновляем данные в АРМ компонента «Управление параметрами», но КМ вычитывает старые данные.

   В конфигурационных файлах среды контейнеризации `conf/config/parameters/<fp_name>.ufsparams-agent-sidecar.conf` и `conf/config/parameters/<FP_Name>.conf` указаны разные значения env переменной `USER_INSTALL_ROOT`. Или же, данное значение переопределено системной переменной `ufsparams.config.directory`.

3. Ошибка вида: `ERROR [configurationTaskScheduler-1] r.s.u.p.a.t.LoggingPathHandlerDecorator - Unexpected error during processing file /opt/conf/ufsparams/config/UFSPARAMS_AGENT_20.__emp.dict.sb.sk-UFSPARAMS_AGENT_20.json: [java.lang.IllegalArgumentException: Remote config service base url is not set!] java.lang.IllegalArgumentException: Remote config service base url is not set!`.

   Решение: нет нужного файла конфигурации Агента. Для Агента необходимо сделать импорт параметров. В Jenkins выбрать `UFSPARAMS_AGENT` и запустить с плейбуком `IMPORT_SUP_PARAMS`.

4. Проблемы импорта параметров в БД компонента «Управление параметрами».

   a. Импорт параметров завершен успешно, но в базе нет нужных данных.

   Решение: проблема может возникнуть, если использовать неправильный url обращения к сервису Импорта параметров в БД. Правильный адрес: `https://host:port/ufsparams-import-v2/file/curl/import/`.

   b. Ошибки вида: `[Import sup2] ERROR [ERROR]: Ошибка в списке комплектов значений параметра [ru.ufc.cache.db.card_way.time_to_live.hard] по индексу [0]. Ошибка в списке значений по индексу [0]. Значение [3.1536e10] не является числом`.

   Решение: все значения в БД хранятся в виде строк, поэтому в файле импорта значения параметров должны быть заключены в кавычки.

   c. При установке Приложения ранее импортированный параметр не обновляется.

   Решение: при последующих установках ранее записанные параметры/значения параметров не обновляются, только добавляются новые. Чтобы перезаписать значение параметра необходимо его удалить через АРМ компонента «Управление параметрами» и пролить заново.

5. Проблемы настройки OTT. Настройка фильтра авторизации Egress OTT, файл `ott-egress-auth-filter.yaml`.

При первоначальной настройке фильтра в pod Egress в логах OTT Sidecar не было видно исходящих запросов.

После изменения порта на `7070` и добавлении

```
...   workloadSelector:      labels:          app: istio-egressgateway-ufsparams-unver
```

работа потребителей всего кластера была прервана.

Решение: корректная настройка фильтра на Egress/Ingress по инструкции.

Чек-лист валидации установки

Чек-лист валидации установки компонента «Управление конфигурациями»

1. Проверить, что в среде контейнеризации все контейнеры компонента «Управление конфигурациями» в pod имеют статус `Ready`.
2. Проверить, что существует схема БД `cfga`.
3. Проверить, что для проекта создались объекты приложений на основании файлов конфигурации: DeploymentConfig, ConfigMap, Service, Route, pod.
4. Проверить, что для проекта создались объекты istio: ingress/egress на основании файлов конфигурации: Deployment, VirtualService, ServiceEntry, DestinationRule, Gateway, ConfigMap.
5. Проверить логи на отсутствие ошибок.

Чек-лист валидации установки компонента «Управление параметрами»

1. Проверить, что в среде контейнеризации все контейнеры компонента «Управление параметрами» в pod имеют статус `Ready`.
2. Проверить, что существует схема БД `cfge` и прошел импорт параметров Приложения.
3. Проверить, что для проекта в среде контейнеризации создались объекты приложений на основании файлов конфигурации: DeploymentConfig, ConfigMap, Service, Route, pod.
4. Проверить, что для проекта в среде контейнеризации создались объекты istio: ingress/egress на основании файлов конфигурации: Deployment, VirtualService, ServiceEntry, DestinationRule, Gateway, ConfigMap.
5. Проверить, что все компоненты отвечают по `/environment/product`.
6. Проверить логи на отсутствие ошибок.
7. Проверить наличие заголовков Ingress/Egress.

Руководство по установке компонента Прикладной журнал (APLJ)

Термины и определения

Термин	Значение
АРМ	Автоматизированное Рабочее Место, компонент ПЖ, предоставляющий графический интерфейс пользователя
ПО	Программное Обеспечение
HMAC	hash-based message authentication code, код аутентификации сообщений, использующий хеш-функции с ключом
ACL	Access Control List, список управления доступом, определяющий, кто или что может получать доступ к объекту (программе, процессу или файлу), и какие именно операции разрешено или запрещено выполнять субъекту
GUI	Graphical User Interface, графический интерфейс пользователя
СУБД	Система управления базами данных, совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных
АС	Автоматизированная система
БД	База данных
ПЖ	Компонент «Прикладной Журнал»
Модуль Platform V	АС или часть АС, реализованная на платформе Platform V и выделенная в отдельный функциональный модуль

ПФ	Прикладная Фабрика. Клиентский по отношению к Platform V Archiving сервис, поставляющий свои объекты в формате ПЖ для их репликации в ФД (ОД)
СП	Сервер приложений WildFly
ТС	Технологический сервис. Сервис Platform V, выполняющий выделенный набор задач, общих для ПФ
ФД	Фабрика Данных. Функциональная подсистема Корпоративной Аналитической Платформы, которая работает с хранилищем данных (ОД)
Stand-In	Особый режим работы некоторых АС с ограниченной функциональностью, который предназначен для обеспечения работы клиентов Банка и смежных систем в следующих случаях: при системных сбоях АС; при технологических работах в АС; в процессе установки плановых обновлений прикладного ПО
StandBy	При отказе основной (Primary) базы идет автоматическое переключение на дублирующую базу (StandBy)
IAM	Platfrom V IAM SE
КМД	Корпоративная модель данных
АРМ	Консоль управления, через которую происходит переход в функциональный Stand-In
КПЖ	Клиенты ПЖ, использующие Stand-In-client для перехода в Stand-In
BSSI	Stand-In Base Service, сервис заливки и сверки БД пользователя
SSL	Secure Sockets Layer, криптографический протокол, который подразумевает более безопасную связь

Pod	Kubernetes Pod — это абстрактный объект Kubernetes, представляющий собой группу из одного или нескольких контейнеров приложения (например, Docker или rkt) и совместно используемых ресурсов для этих контейнеров
EDM	сервис выгрузки, отправляет журналы по ММТ
LCK	Lock, тип блокировки, отправляемый в ПЖ, для блокировки записи и перехода в Stand-In
УПС	Умеющие Переходить в Stand-In, обозначение серврера, умеющего переходить в Stand-In
ЦОД	Центр Обработки Данных
jks	Java KeyStore, хранилище ключей в виде базы данных, представленное классом KeyStore (java.security.KeyStore)
TPS	Transactions Per Second, количество транзакций, обрабатываемых за одну секунду
Bean	JavaBeans, Java-классы, написанные по определенным правилам. Используются для объединения нескольких объектов в один bean, для удобной передачи данных

Системные требования

Настройки безопасности окружения и перечень платформенных (дополнительных внешних) продуктов, используемых для установки, настройки и контроля в конечной информационной системе (далее — ИС), выбираются при разработке конечной ИС, исходя из характера обрабатываемой в ней информации и иных требований информационной безопасности (далее — ИБ), предъявляемых к ней.

Системное программное обеспечение

Ниже представлены категории системного программного обеспечения, которые обязательны для установки, настройки, контроля и функционирования компонента. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО

«СберТех» (маркировка «Рекомендовано» в столбце Комментарий). Клиенту необходимо выбрать один из продуктов в каждой категории, исходя из условий использования конечной ИС.

Операционная система

Наименование	Версия	Комментарий
ОС Альт 8 СП	8 и выше	Рекомендовано
Red Hat Enterprise Linux	7.8 и выше	Опционально

Среда контейнеризации

Наименование	Версия	Комментарий
Kubernetes	1.21 и выше	Рекомендовано
Red Hat OpenShift	2.0 и выше	Опционально

Java-машина

Наименование	Версия	Комментарий
OpenJDK	8 и выше	Рекомендовано
OracleJDK	8 и выше	Опционально

Система управления базами данных (СУБД)

Наименование	Версия	Комментарий
PostgreSQL	11 и выше	Рекомендовано
Oracle Database	19	Опционально

Рекомендован Platform V Pangolin SE, см. раздел «Платформенные зависимости».

Браузер

Наименование	Версия	Применение
Yandex Browser	19.10.3 и выше	Рекомендовано
Браузер на основе Chromium		Опционально

Брокер сообщений (интерфейсный движок)

Наименование	Версия	Рекомендовано/опционально
Kafka	2.7.0. и выше	Рекомендовано

Рекомендован Platform V Corax, см. раздел «Платформенные зависимости».

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Audit SE	AUD	2.1 и выше	AUDT Аудит	опционально	Сервис для аудирования событий
Platform V IAM SE	IAM	1.2 и выше	-	опционально	Набор инструментов для управления доступом к информационным ресурсам. Данные инструменты необходимы для аутентификации и авторизации пользователей
Platform V Monitor	OPM	3.0 и выше	MONA Объединенный мониторинг Unimon	опционально	Сервис для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения
Platform V Pangolin SE	PSQ	4.6.0 и выше	-	опционально	Система управления базами данных, основанная на PostgreSQL

Platform V DevOps Tools	DOT	1.2	-	опционально	Сервис для развертывания и обновления компонентов Платформы и приложений потребителей, для настройки и обслуживания инфраструктуры Платформы

Примечание:

- **Обязательно** — компонент или продукт необходим для функционирования сервиса.
- **Опционально** — необязательный для функционирования сервиса компонент или продукт, рекомендуется его установка, но допускается использование аналога других производителей.

Аппаратные требования

Для установки компонента/продукта требуется конфигурация аппаратного обеспечения, указанная в таблице ниже.

Назначение	Технические требования	Комментарий	Рекомендации по отказоустойчивости
Сервер СУБД	CPU: 4*88 ядер; RAM: 2048 ГБ; HDD: 3600 ГБ	Хранение настроек, сообщений и статуса репликации	Рекомендуется использовать отказоустойчивый кластер (Oracle ADG или Pangolin Patroni)
Кластер Kafka	CPU: 2*28 ядер; RAM: 512 ГБ; HDD: 1000 ГБ	Передача сообщений между ПЖ и потребителем	Рекомендуется использовать георасределённый кластер, `replicationFactor = 3`

Назначение	Технические требования	Комментарий	Рекомендации по отказоустойчивости
Namespace в среде контейнеризации			

Выбор способа установки

В зависимости от способа развертывания сервиса воспользуйтесь одной из инструкций:

- целевая ручная установка сервиса;
- опциональная автоматизированная установка сервиса с использованием программного продукта Platform V DevOps Tools.

Установка

Ручная установка сервиса

Прикладной Журнал является набором модулей, интегрирующих работу БД и Kafka. Для работы Прикладного Журнала необходимо подготовить и настроить базу данных и брокер Kafka.

Подготовка базы данных ПЖ

Для работы Прикладного Журнала необходимо предварительно настроить базу данных. Данное руководство не рассматривает установку PostgreSQL, только его настройку.

Для настройки базы данных:

1. Подготовка схемы в зависимости от используемой СУБД:
   - Platform V Pangolin SE: Выполните скрипт `POSTGRESQL\TABLESPACESANDUSERS.sql`, который находится в дистрибутиве. При выполнении скрипта автоматически создаются пользователи, а также все необходимые таблицы и схемы.
   - Oracle: Выполните скрипт `DB\TABLESPACESANDUSERS.sql`, который находится в дистрибутиве. При выполнении скрипта автоматически создаются пользователи, а также все необходимые таблицы и схемы.
2. В `liquibase.properties` внесите информацию о базе данных (URL и имя пользователя).
3. Запустите файл `dbupdate.bat` для ОС Windows или `dbupdate.shl` для ОС Linux. После запуска `dbupdate` будут автоматически созданы/обновлены все необходимые для работы таблицы.

Использование SSL для доступа к БД не является обязательным для ПЖ, но может быть необходимо по требованиям вашей безопасности.

Подготовка Kafka

ПЖ автоматически создает нужные топики при старте.

Использование SSL для доступа к Kafka не является обязательным для ПЖ, но может быть необходимо по требованиям вашей безопасности.

Настройки Kafka идентичны для ПЖ третьего и четвертого поколений.

Для отключения автоматического создания топиков Kafka для брокеров укажите настройку `auto.create.topics.enable = false`.

Требования по предоставлению прав доступа к топикам Kafka

Выдайте сертификату для Kafka ПЖ разрешения, указанные в таблице ниже, где:

- `\{ZONE\}` – зона, в которой будет работать фабрика;
- `\{DATA\_TYPE\}` – тип вектора;
- `\{PLUGIN\}` – плагин выгрузки.

Имя топика	Субъект	Привилегия	Пример
`journal\_request\_topic\_\{ZONE\}\_\{DATA_TYPE\}`	Клиент (применитель)	Чтение	`journal\_request\_topic\_CA1\_CDM\_CV`
`journal\_response\_topic\_\{ZONE\}\_\{DATA\_TYPE\}\_\{PLUGIN\}`	Клиент (применитель)	Запись	`journal\_response\_topic\_CA1\_CDM\_CV\_EXPORT\_FD`
`journal\_error\_request\_topic\_\{ZONE\}\_\{DATA\_TYPE\}\_\{PLUGIN\}`	Клиент (применитель)	Запись/Чтение	`journal\_error\_request\_topic\_CA1\_CDM\_CV\_EXPORT\_FD`
`journal\_sicl\_siservice`	Клиент (все)	Запись	`journal\_sicl\_siservice`
`aj\_standin\_status\_event\_compact`	Клиент (все)	Чтение	`aj\_standin\_status\_event\_compact`
`journal\_writer\_topic\_\{ZONE\}`	Клиент (бизнес-логика)	Запись	`journal\_writer\_topic\_CA1`

cloud_client_event		Чтение	cloud_client_event
journal_acc_sync_topic_\{ZONE\}		Запись	journal_acc_sync_topic_CA1

Для просмотра списка серверов и топиков Kafka в АРМ ПЖ откройте вкладку **Мониторинг**.

Настройка в OpenShift

Для работы Stand-In клиента необходимо добавить в ConfigMap конфигурацию по следующему примеру:

```
standin:   cloud:     client:       zoneId: Название зоны
stub: false       heartBeatPeriod: 1000       kafka:
bootstrapServers: адреса серверов Кафки ПЖ через запятую
producerConfig:           "[security.protocol]": SSL
"[ssl.key.password]": ${SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/opt/keystore/kafka/server.keystore.jks
"[ssl.keystore.password]": ${SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /opt/keystore/kafka/trust.jks
"[ssl.truststore.password]": ${SSL_TRUSTSTORE_PASSWORD}
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS           "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
consumerConfig:           "[security.protocol]": SSL
"[ssl.key.password]": ${SSL_KEY_PASSWORD}
"[ssl.keystore.location]":
/opt/keystore/kafka/server.keystore.jks
"[ssl.keystore.password]": ${SSL_KEYSTORE_PASSWORD}
"[ssl.truststore.location]": /opt/keystore/kafka/trust.jks
"[ssl.truststore.password]": ${SSL_TRUSTSTORE_PASSWORD}
"[ssl.keystore.type]": JKS
"[ssl.truststore.type]": JKS           "[ssl.protocol]": TLS
"[ssl.enabled.protocols]": TLSv1.2
"[ssl.endpoint.identification.algorithm]": ~
concurrency: 10       groupId: group_1
```

Не рекомендуется перенаправлять трафик на Egress. Исключение составляют:

Формулировка	Исключение

Все входящие запросы (в проект) должны проходить через router и ingress-proxy. Все исходящие запросы (из проекта) должны проходить через egress-proxy. При этом для входящих/исходящих взаимодействий должен использоваться mTLS 1.2+	Только для http-вызовов, пока не будет внедрена система ротации сертификатов
На ingress/egress proxy должны быть реализованы механизмы аутентификации и авторизации (как для входящих, так и для исходящих запросов)	Только для http-вызовов, пока не будет внедрена система ротации сертификатов

Для настройки интеграции через EGRESS в ISTIO следует обратиться к поддержке разработки Synapse.

Установка в Kubernetes

В данном разделе описывается ручная установка в Kubernetes. Перед началом ручной установки заполните в `.yaml` файлах параметры, описанные в `package/config/parameters/aplj.all.conf`.

Установка journal-arm-cloud

Конифигурационные файлы находятся в директории `package/conf/openshift/journal-arm-cloud`:

1. Создайте ConfigMap командой:

   ```
   kubectl apply -f application.property.yaml
   ```

   Создаст - `journal-arm-cloud-config`.

2. Создайте Deployment командой:

   ```
   kubectl apply -f dc.yaml
   ```

   Создаст - `journal-arm-cloud-deployment`.

3. Создайте Service командой:

   ```
   kubectl apply -f service.yaml
   ```

Создаст - `journal-arm-cloud-service`.

Установка journal-writer-cloud

Конифигурационные файлы находятся в директории `package/conf/openshift/journal-writer-cloud`:

1. Создайте ConfigMap командой:

```
kubectl apply -f application.property.yaml
```

   Создаст - `journal-writer-cloud-config`.

2. Создайте Deployment командой:

```
kubectl aply -f dc.yaml
```

   Создаст - `journal-writer-cloud-deployment`.

Установка сервиса в OpenShift

Модули ПЖ для OpenShift располагаются в директории `openshift` дистрибутива ПЖ.

Настройка основных приложений ARM и Writer

Для настройки необходимо:

1. Создать Image Stream для модулей APM и Writer.
2. Создать ConfigMap для OpenShift.
3. Создать DeploymentConfig.
4. Создать Service.
5. Создать Route.

Создание Image Stream

Для создания Image Stream для модулей APM и Writer выполните в консоли команды (с заменой комментариев на соответствующие значения):

- для APM:

```
oc process -f <имя yml-файла, содержащего конфигурацию
ImageStream для APM>.yml \ -p DOCKER_REGISTRY="<имя
докер-реестра>" \ -p DOCKER_REPO="<имя репозитория в
докер-реестре>" \  | oc create -f -
```

Создаст Image Stream:

```
- standin-arm-image-stream
```

- для WRITER:

```
oc process -f imagestream-application-journal-writer-
image-stream.yml \ -p DOCKER_REGISTRY="<имя докер-
реестра>" \ -p DOCKER_REPO="<имя репозитория в докер-
реестре>" \  | oc create -f -
```

Создаст Image Stream

```
- application-journal-writer-image-stream
```

Создание ConfigMap

Для создания ConfigMap:

1. Создайте секрет с сертификатами до Audit и до Kafka ПЖ:

```
oc create secret generic kafka-secret \     --from-
file=keystore=client.keystore \     --from-
file=truststore=client.truststore \     --from-
file=audit_keystore=client.keystore \     --from-
file=audit_truststore=client.truststore
```

2. Создайте секрет с паролями
   - SECURITY_SECRET_KEY - секрет для подключения к SUDIR;
   - DB_PASSWORD - пароль от БД ПЖ;
   - SSL_KEY_PASSWORD - пароль от ключа Kafka ПЖ;
   - SSL_KEYSTORE_PASSWORD - пароль от keystore Kafka ПЖ;
   - SSL_TRUSTSTORE_PASSWORD - пароль от truststore Kafka ПЖ;
   - audit2.proxy.url - URL для подключения к Platform V Audit;
   - ssl.truststore.password - пароль от truststore Kafka Platform V Audit;
   - ssl.keystore.password - пароль от keystore Kafka Platform V Audit;

- ssl.key.password - пароль ключа Kafka Platform V Audit;

```
oc create secret generic application-journal-secret  \
--from-literal=SECURITY_SECRET_KEY=AAAaaaQB \     --from-
literal=DB_PASSWORD=EVENT_JOURNAL2 \     --from-
literal=SSL_KEY_PASSWORD=module \     --from-
literal=SSL_KEYSTORE_PASSWORD=module \     --from-
literal=SSL_TRUSTSTORE_PASSWORD=module \     --from-
literal=audit2.proxy.url=your.server.name:<port> \     --
from-literal=ssl.truststore.password=module \     --from-
literal=ssl.keystore.password=module \     --from-
literal=ssl.key.password=module
```

3. Выполните команды:
   - для APM:

     ```
     oc process -f config-map-template.yml \     -p
     MODULE_ID=standin-arm \     -p
     DB_URL="jdbc:oracle:thin:@your.server.name:<port>" \
     -p ZOOKEEPER="your.server.name:<port>" \     -p
     KAFKA_BOOTSTRAPSERVERS="your.server.name:<port>" \
     -p SECURITY_PROTOCOL="SSL" \     | oc create -f -
     ```

     Создаст

     ```
     - standin-arm-config
     ```

   - для Writer:

     ```
     oc process -f config-map-template.yml \        -p
     MODULE_ID=application-journal-writer \        -p
     DB_URL="jdbc:oracle:thin:@your.server.name:<port>" \
     -p ZOOKEEPER="your.server.name:<port>" \        -p
     KAFKA_BOOTSTRAPSERVERS="your.server.name:<port>" \
     -p SECURITY_PROTOCOL="SSL" \        | oc create -f -
     ```

     Создаст

     ```
     - application-journal-writer-config
     ```

- Описание параметров:
  - MODULE_ID - параметр изменять не нужно;
  - DB_URL - подключение до БД ПЖ;

- ZOOKEEPER - подключение до zookeeper ПЖ;
- KAFKA_BOOTSTRAPSERVERS - подключение до Kafka ПЖ;
- SECURITY_PROTOCOL - протокол подключения до Kafka ПЖ SSL или PLAINTEXT.

Создание DeploymentConfig

Для создания DeploymentConfig выполните:

- для ARM:

```
oc process -f deployment-config-arm.yml \ | oc create -f
-
```

Создаст

```
- standin-arm-deployment-config
```

- для WRITER:

```
oc process -f deployment-config-writer.yml \ | oc create
-f -
```

Создаст

```
- application-journal-writer-deployment-config
```

Создание Service

Для создания Service для ARM выполните:

```
oc process -f service-arm.yml \ | oc create -f -
```

Создаст

```
- standin-arm-service
```

Создание Route

Для создания Route для ARM выполните:

```
oc process -f route-arm.yml \ | oc create -f -
```

Создаст

```
- standin-arm-route
```

Автоматизированная установка сервиса с использованием Deploy Tools

Для установки сервиса с использованием компонента Deploy Tools (CDJE) программного продукта Platform V DevOps Tools (DOT) разверните сервис в инфраструктуре и запустите CD Job по инструкции, приведенной в документации Platform V DevOps Tools (DOT).

Требования к окружению:

- наличие у сервиса пространства в сервисе централизованного хранения репозиториев исходного кода и прав на создание репозиториев в нем;
- наличие у сервиса области в Jenkins, с настроенными учетными данными (логин/пароль) для авторизации под технологической учетной записью (ТУЗ), позволяющая загружать артефакты в целевой репозиторий хранилища.

Перед установкой компонента «Прикладной Журнал»:

- создайте репозиторий, куда будет распакован дистрибутив;
- добавьте в файл subsystem.json GAV параметры загружаемого дистрибутива.

Запустите ansible playbook MIGRATION_FP_CONF инструмента Deploy Tools, который поместит дистрибутив в подготовленный репозиторий.

Параметры компонента содержатся в:

- каталоге `package/conf/config/parameters`: основные параметры компонента;
- все каталоги `secrets` внутри каталога `package/conf/k8s` — параметры для установки через Kubernetes или OpenShift;
- файле `package/distrib.yaml`.

Перед запуском установки:

- проверить, что в COMMON заданы все глобальные параметры, которые указаны в параметрах компонента;
- изменить значения по умолчанию, если требуется;

- в файле `_passwords.conf` заполнить параметры, содержащиеся в блоке `secrets` в файле `package/distrib.yaml`.

Обновление

Обновление Прикладного Журнала состоит из двух этапов:

1. Подготовительные действия:
   1. Заполните параметр `url` в файле `liquibase.properties`, указав путь до БД Прикладного Журнала.
   2. Запустите консоль и выполните скрипт `dbupdate.bat` или `dbupdate.shl`.
   3. Введите пароль пользователя `EVENT_JOURNAL2`.
   4. Дождитесь выполнения скриптов, должна появиться надпись:

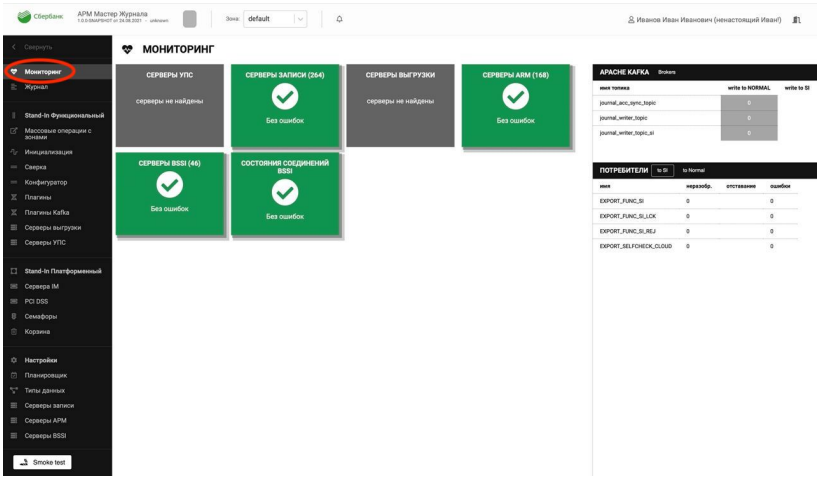
```
Liquibase: Update has been successful.
```

2. Установка новой версии Прикладного Журнала.

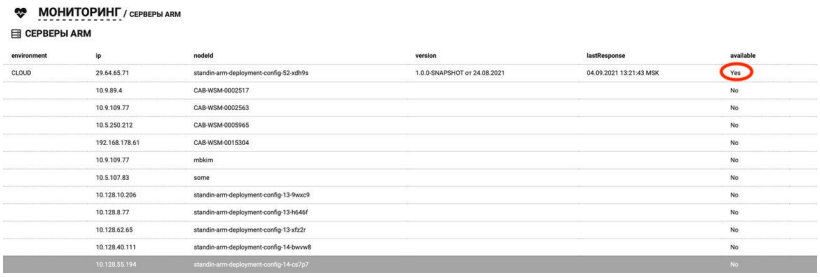
Проверка работоспособности

Для проверки работоспособности модулей необходимо:

1. Переключиться на вкладку **Мониторинг** в UI АРМ.



2. Выбрать необходимый модуль и убедиться в том, что в графе «available» интересующих серверов стоит статус «Yes».



Откат

База данных имеет обратную совместимость.

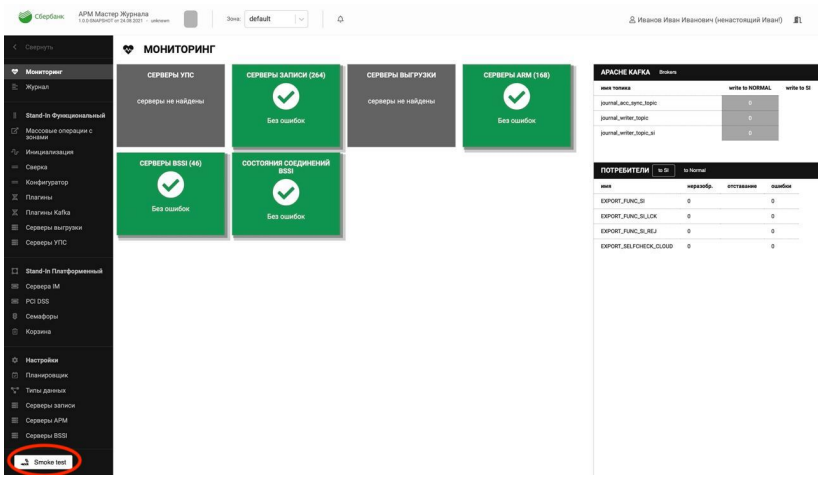
Для отката на предыдущую версию Прикладного Журнала необходимо запустить в среде контейнеризации модули Прикладного Журнала предыдущей версии .

Часто встречающиеся проблемы и способы их решения

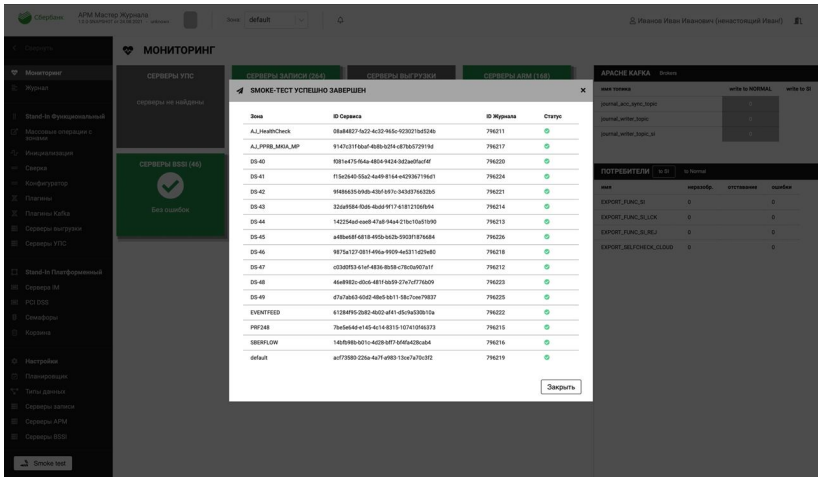
Ошибка	Способ решения
Ошибка выполнения liquibase-скриптов	1. Посмотреть лог liquibase на предмет ошибок; 2. Проверить подключение к БД; 3. Проверить привилегии пользователя; 4. Произвести правки для устранения найденных ошибок (описание ошибок см. в документации liquibase)
Ошибка установки в среде контейнеризации	1. Посмотреть лог POD на предмет ошибок; 2. Произвести правки для устранения найденных ошибок (описание ошибок см. в документации выбранной среды контейнеризации)

Чек-лист валидации установки

Валидация установки осуществляется при помощи Smoke Test. Для этого нажмите на кнопку **Smoke Test** в нижней части интерфейса:



Затем дождитесь результатов и убедитесь, что все сервисы работают в штатном режиме:



Описание настроек

Описание настроек в application-journal-config-struct.xml

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
числовой	bssi_oracle.db.min_pool_size	20	необязательный	Минимальное количество подключений к БД ПЖ от модуля BSSI
числовой	bssi_oracle.db.max_pool_size	100	необязательный	Максимальное количество подключений к БД ПЖ от модуля BSSI
числовой	bssi_oracle.db.connection_test_period	30	необязательный	Периодичность проверки подключений от модуля BSSI к БД ПЖ
числовой	arm_db_min_pool_size	20	необязательный	Минимальное количество подключений к БД ПЖ от модуля АРМ
числовой	arm_db_max_pool_size	100	необязательный	Максимальное количество подключений к БД ПЖ от модуля АРМ

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
числовой	arm_db_connection_test_period	30	необязательный	Периодичность проверки подключений от модуля АРМ к БД ПЖ
числовой	writer_db_min_pool_size	50	необязательный	Минимальное количество подключений к БД ПЖ от модуля WRITER
числовой	writer_db_max_pool_size	100	необязательный	Максимальное количество подключений к БД ПЖ от модуля WRITER
числовой	writer_db_connection_test_period	30	необязательный	Периодичность проверки подключений от модуля WRITER к БД ПЖ
числовой	sicl_refresh_state_period	5000	необязательный	Периодичность оправки heartbeat клиентами ПЖ
числовой	export_db_min_pool_size	50	необязательный	Минимальное количество подключений к БД ПЖ от модуля EDM
числовой	export_db_max_pool_size	100	необязательный	Максимальное количество подключений к БД ПЖ от модуля EDM
числовой	export_db_connection_test_period	30	необязательный	Периодичность проверки подключений от модуля EDM к БД ПЖ

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
числовой	export_max_threads	200	необязательный	Максимальное количество потоков выгрузки на EDM
числовой	zone_monitor_max_threads	10	необязательный	Количество вспомогательных потоков в модулях APM, WRITER, EDM, BSSI
числовой	standin.validate.duration.hours	24	необязательный	Разница в дате между первичной и последующими сверками
перечисление	enviroment	NORMAL	необязательный. Возможные варианты значений: NORMAL, STAND-IN	Окружение платформенного Stand-In
логический	gather_stats_db_executions.enabled	true	необязательный	Собирать ли внутри приклада статистику работы методов обращающихся к БД ПЖ
числовой	gather_stats_db_executions.print_period	10	необязательный	Период вывода в логи статистики работы методов обращающихся к БД ПЖ в минутах
строковый	export_salt	protected=true	необязательный	Соль для контента журналов
строковый	arm_audit.client	sbt-client	необязательный	Идентификатор клиента для Азимута

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
строковый	arm_audit.ticket	application-journal-ticket	необязательный	Идентификатор тикета для Азимута
числовой	aj_monitor.scheduler.interval	5000	необязательный	Интервал планировщика сбора системной информации для мониторинга ПЖ (в мс)
строковый	zookeeper.connection	обязательный		Подключение к Zookeeper ПЖ с общими настройками
числовой	zookeeper.session.timeout.ms	30000	необязательный	Таймаут сессии к Zookeeper ПЖ
числовой	zookeeper.connection.timeout.ms	15000	необязательный	Таймаут подключения к Zookeeper ПЖ
числовой	zookeeper.base.sleep.time.ms	1000	необязательный	Время между повторными попытками запросов в Zookeeper ПЖ
числовой	zookeeper.max.retries	3	необязательный	Количество повторных попыток записи в Zookeeper ПЖ
строковый	zookeeper.ssl.key.store.location		необязательный	Key store сертификата для подключения к Zookeeper ПЖ
строковый	zookeeper.ssl.key.store.password		необязательный	Пароль для key store сертификата для поключения к Zookeeper ПЖ

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
строковый	zookeeper.ssl.trust.store.location		необязательный	Trust store сертификата для подключения к Zookeeper ПЖ
строковый	zookeeper.ssl.trust.store.password		необязательный	Пароль trust store для подключения к Zookeeper ПЖ

Следующих параметров может быть несколько для каждой зоны:

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
строковый	zoneId			Идентификаторы зоны Stand-In через запятую
логический	default			Если стоит true, в этой БД будут храниться настройки и журналы без зоны, через этот экземпляр Kafka будет передаваться heartbit. Может быть true только у одной зоны
строковый	db_url			URL БД ПЖ
строковый	db_user			Пользователь БД ПЖ
строковый	db_password	protected=true		Пароль БД ПЖ

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
логический	switchPlugins	true	необязательный	Флаг переключения плагинов на всех зонах в группе. Если true, то при переходе в Платформенный Stand-In у плагинов сменится активный контур
строковый	db_schema			Схема в БД ПЖ
строковый	kafka.bootstrap.servers			Подключение к Kafka ПЖ
enumProperty	kafka.producer.compression.type	snappy	необязательный	Тип компрессии сообщений используемый при взаимодействие с Kafka ПЖ
числовой	kafka.consumer.concurrency	20	необязательный	Количество потоков, вычитывающих журналы, на модуле WRITER
строковый	group.id			Префикс group_id, с которым ПЖ подключается к Kafka
числовой	db.connect.read.timeout	660000	необязательный	Таймаут на подключение или чтение на уровне jdbc драйвера, в мс

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
строковый	db_keyStore		необязательный	Расположение хранилища ключей для подключения к БД ПЖ
строковый	db_keyStorePassword		необязательный	Пароль хранилища ключей для подключения к БД ПЖ
строковый	db_keyStoreType	JKS	необязательный	Тип хранилища ключей для подключения к БД ПЖ
строковый	db_trustStore		необязательный	Расположение хранилища сертификатов для подключения к БД ПЖ
строковый	db_trustStorePassword		необязательный	Пароль хранилища сертификатов для подключения к БД ПЖ
строковый	db_trustStoreType	JKS	необязательный	Тип хранилища сертификатов для подключения к БД ПЖ
строковый	db_cipherSuites	SSL_RSA_WITH_3DES_EDE_CBC_SHA	необязательный	Алгоритм шифрования для подключения к БД ПЖ
строковый	db_sslVersion	1.2	необязательный	Версия TLS, используемая для подключения к БД ПЖ

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
строковый	db_postgresSslCert	./postgresClient.crt	необязательный	Расположение сертификата шифрования при использовании в качестве БД ПЖ PostgreSQL
строковый	db_postgresSslKey	./postgresClient.pk8	необязательный	Ключ сертификата при использовании в качестве БД ПЖ PostgreSQL
строковый	db_postgresSslRootCert	./postgresRoot.crt	необязательный	Расположения корневого сертификата при использовании в качестве БД ПЖ PostgreSQL
перечисление	db_postgresSslMod	disable	необязательный	Возможные варианты значений: disable; allow; prefer; require; verify-ca; verify-full SSL Mode при использовании в качестве БД ПЖ PostgreSQL
логический	db_postgresSsl	false	необязательный	Включить подключения к БД PostgreSQL по TLS
строковый	kafka.producer.ssl.key.password	protected=true	необязательный	Пароль сертификата для Kafka ПЖ

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
строковый	kafka.producer.ssl.keystore.location		необязательный	Расположение key store для Kafka ПЖ
строковый	kafka.producer.ssl.keystore.password	protected=true	Пароль для key store для Kafka ПЖ	
строковый	kafka.producer.ssl.truststore.location		необязательный	Расположение trust store для Kafka ПЖ
строковый	kafka.producer.ssl.truststore.password	protected=true	необязательный	Пароль для trust store для Kafka ПЖ
строковый	kafka.consumer.ssl.key.password	protected=true	необязательный	Пароль сертификата для Kafka ПЖ
строковый	kafka.consumer.ssl.keystore.location		необязательный	Расположения key store для Kafka ПЖ
строковый	kafka.consumer.ssl.keystore.password	protected=true	Пароль для key store для Kafka ПЖ	
строковый	kafka.consumer.ssl.truststore.location		необязательный	Расположение trust store для Kafka ПЖ
строковый	kafka.consumer.ssl.truststore.password	protected=true	необязательный	Пароль для trust store для Kafka ПЖ
строковый	zookeeper.connection		обязательный	Подключение к Zookeeper ПЖ с зональными настройками
числовой	zookeeper.session.timeout.ms	30000	необязательный	Таймаут сессии к Zookeeper ПЖ

Тип	Наименование	Значение по умолчанию	Обязательность	Описание
числовой	zookeeper.connection.timeout.ms	15000	необязательный	Таймаут подключения к Zookeeper ПЖ
числовой	zookeeper.base.sleep.time.ms	1000	необязательный	Время между повторными попытками запросов в Zookeeper ПЖ
числовой	zookeeper.max.retries	3	необязательный	Количество ретраев записи в Zookeeper ПЖ
строковый	zookeeper.ssl.key.store.location		необязательный	Key store сертификата для подключения к Zookeeper ПЖ
строковый	zookeeper.ssl.key.store.password		необязательный	Пароль для key store сертификата для поключения к Zookeeper ПЖ
строковый	zookeeper.ssl.trust.store.location		необязательный	Trust store сертификата для подключения к Zookeeper ПЖ
строковый	zookeeper.ssl.trust.store.password		необязательный	Пароль trust store для подключения к Zookeeper ПЖ

Руководство по установке компонента Сервис шардирования приложений (ASPX)

Системные требования

Настройки безопасности окружения и перечень платформенных (дополнительных внешних) продуктов, используемых для установки, настройки и контроля в конечной информационной системе (далее — ИС), выбираются при разработке конечной ИС, исходя из характера обрабатываемой в ней информации и иных требований информационной безопасности (далее — ИБ), предъявляемых к ней.

Системное программное обеспечение

Ниже представлены категории системного программного обеспечения, которые обязательны для установки, настройки, контроля и функционирования компонента. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО «СберТех» (маркировка «Рекомендовано» в столбце Комментарий). Клиенту необходимо выбрать один из продуктов в каждой категории, исходя из условий использования конечной ИС.

Операционная система

Наименование	Версия	Комментарий
ОС Альт 8 СП	10.0 и выше	Рекомендовано
Red Hat Enterprise Linux	3.10 и выше	Опционально

Java-машина

Наименование	Версия	Комментарий
OpenJDK	1.8 и выше	Рекомендовано
OracleJDK	1.8 и выше	Опционально

Система мониторинга сборки

Наименование	Версия	Применение	Комментарий

Наименование	Версия	Применение	Комментарий
Prometheus	2.31 и выше	Рекомендовано	Программное обеспечение для анализа численных метрик работы сценария.
Grafana	2.5.0 и выше	Рекомендовано	Программное обеспечение для визуализации численных метрик работы сценария (предоставленных, например, Prometheus).

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Monitor	OPM	D-02.000.00-77 и выше	LOGA Журналирование	опционально	Сервис для хранения лог-файлов

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Monitor	OPM	6.0.0-02	MONA Объединенный мониторинг Unimon	опционально	Сервис для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения
Platform V DevOps Tools	DOT	1.2	-	опционально	Сервис для развертывания и обновления компонентов Платформы и приложений потребителей, для настройки и обслуживания инфраструктуры Платформы
Platform V Synapse Service Mesh	SSM	1.12 и выше	IGEG Граничный прокси	опционально	Сервис для обеспечения управляемого вызова интеграционных сервисов прикладной части в проекте Kubernetes

Примечание:

- **Обязательно** — компонент или продукт необходим для функционирования сервиса.
- **Опционально** — необязательный для функционирования сервиса компонент или продукт, рекомендуется его установка, но допускается использование аналога других производителей.

Аппаратные требования

Для установки компонента требуется следующая конфигурация аппаратного обеспечения:

ASR (CPU, RAM)

- Квота на проект: 32, 48Gi.
- Количество pod на один элемент развертывания ASR: 4
- Характеристики элемента развертывания ASR:
  - fort-asr: 1900m, 1500Mi
- Характеристики Sidecar-контейнеров элемента развертывания ASR:
  - fluent-bit: 200m, 512Mi
  - istio: 200m, 1600Mi
- Количество pod на один элемент развертывания ingress: 4
- Характеристики элемента развертывания ingress:
  - istio-proxy: 1600m, 1500Mi
- Характеристики Sidecar-контейнеров элемента развертывания ingress:
  - ott: 800m, 2000Mi
- Количество pod на один элемент развертывания egress: 4
- Характеристики элемента развертывания egress:
  - istio-proxy: 1600m, 1500Mi
- Характеристики Sidecar-контейнеров элемента развертывания egress:
  - ott: 800m, 2000Mi
- Квота на pod с учетом sidecar-контейнеров:
  - fort-asr: 2300m, 3600 Mi;
  - ingress: 2400m, 3500Mi.
  - egress: 2400m, 3500Mi

ASM (CPU, RAM)

- Квота на проект: 16, 24Gi.
- Количество pod на один элемент развертывания ASM: 3
- Характеристики элемента развертывания ASM:
  - appshard-manager: 1800m, 2500Mi
- Характеристики Sidecar-контейнеров элемента развертывания ASM:
  - fluent-bit: 100m, 512Mi
  - istio: 200m, 512Mi
- Количество pod на один элемент развертывания ingress: 3
- Характеристики элемента развертывания ingress:
  - istio-proxy: 800m, 1500Mi
- Характеристики Sidecar-контейнеров элемента развертывания ingress:
  - ott: 800m, 2000Mi
- Количество pod на один элемент развертывания egress: 3
- Характеристики элемента развертывания egress:
  - istio-proxy: 800m, 1500Mi
- Квота на pod с учетом sidecar-контейнеров:
  - fort-asr: 2100m, 3600 Mi;
  - ingress: 1600m, 3500Mi.
  - egress: 800m, 1500Mi

Состав дистрибутива

Компонент дистрибутива	Описание
documentation	Каталог с документацией в формате MarkDown
package/bh	Исполняемые файлы приложений appshard-manager, apphard-router
package/conf	Конфигурация развертывания
package/conf/config/parameters	Файлы с параметрами развертывания
package/conf/k8s/base	Манифесты развертывания для рекомендуемого стека
package/conf/k8s/base/asmx	Манифесты развертывания appshard-manager
package/conf/k8s/base/asmx/dc.yaml	DeploymentConfig для appshard-manager
package/conf/k8s/base/asmx/configmaps	Ресурсы типа ConfigMap для appshard-manager
package/conf/k8s/base/asmx/istio	Манифесты и конфигурации istio
package/conf/k8s/base/asrx	Манифесты развертывания appshard-router
package/conf/k8s/base/asrx/dc.yaml	DeploymentConfig для appshard-router
package/conf/k8s/base/asrx/configmaps	Ресурсы типа ConfigMap для appshard-router
package/conf/k8s/base/asrx/istio	Манифесты и конфигурации istio
package/conf/k8s/overrides/openshift	Альтернативные ресурсы для установки на OpenShift
package/docker/asmx	Файлы сборки docker-образа appshard-manager
package/docker/asrx	Файлы сборки docker-образа appshard-router

Выбор способа установки

В зависимости от способа развертывания сервиса воспользуйтесь одной из инструкций:

- целевая ручная установка сервиса;
- опциональная автоматизированная установка сервиса с использованием программного продукта Platform V DevOps Tools.

Установка Application Sharding (appsharding) состоит из следующих шагов:

1. Подготовка окружения (предварительная настройка стенда).
2. Установка и настройка AppShard Manager.
3. Установка и настройка AppShard Router.

Последовательность установки AppShard Manager и AppShard Router не имеет значения.

Подготовка окружения

Предварительная настройка стенда

Перед началом установки необходимо в кластере kubernetes выполнить ряд действий, которые не могут быть выполнены в рамках основного конвейера установки (pipeline) и требуют расширенных прав и выполняются администраторами kubernetes:

1. **Загрузка файлов CRD**. Для корректной работы Application Sharding загрузите файлы Custom Resource Definition (CRD), на основе которых будут формироваться Custom Resource (CR), используемые для конфигурирования прикладного ПО.

Custom Resource Definition расположены в дистрибутиве Application Sharding по следующему пути:

```
<директория с дистрибутивом>/   documentation/installation-
guide/manifest/crd/AppShardApi.yaml - CRD для CR AppShardApi
с черновой конфигурацией APIs.   documentation/installation-
guide/manifest/crd/AppShardEntry.yaml - CRD для CR
AppShardEntry.yaml с черновой конфигурацией Entries
documentation/installation-
guide/manifest/crd/AppShardApiReference.yaml - CRD для CR
AppShardApiReference.yaml с чистовой конфигурацией.
```

2. **Создание кластерной роли pipeline-app-shard-resources-role**. Для создания Custom Resources (CR) AppShardApi и AppShardEntry в проектах поставщиков сервисов создайте специальную роль и привяжите ее к Service Account jenkins (п.3).

Манифест для создания кластерной роли расположен в дистрибутиве по следующему пути:

```
<директория с дистрибутивом>/   documentation/installation-
guide/manifest/roles/pipeline-app-shard-resources-role.yaml
- роль для Service Account jenkins.
```

3. **Привязка кластерного типа для роли pipeline-app-shard-resources-role**. Привяжите кластерный тип (ClusterRoleBinding) для роли pipeline-app-shard-resources-role к Service Account jenkins.

4. **Создание ServiceAccount appshard-manager**. Для запуска приложения Application Sharding Manager в проекте, где установлен Application Sharding Manager, создайте отдельный ServiceAccount appshard-manager.

Манифест для создания Service Account расположен по следующему пути:

```
<директория с дистрибутивом>/ documentation/installation-guide/manifest/service-account/asm-service-account.yml - создание Service Account
```

5. **Создание кластерной роли app-shard-resources-role**. Для доступа к специальным ресурсам кластера kubernetes создайте кластерную роль app-shard-resources-role.

Манифест для создания кластерной роли расположен по следующему пути:

```
<директория с дистрибутивом>/  documentation/installation-guide/manifest/asm/roles/app-shard-resources-role.yaml - роль для Service Account appshard-manager
```

6. **Создание кластерной роли hazelcast-cluster-role**. Для работы распределенного хранилища кластера Hazelcast создайте кластерную роль hazelcast-cluster-role.

Манифест для создания кластерной роли расположен по следующему пути:

```
<директория с дистрибутивом>/ documentation/installation-guide/manifest/roles/hazelcast-cluster-role.yaml - роль для Service Account appshard-manager
```

7. **Привязка кластерного типа для ролей app-shard-resources-role и hazelcast-cluster-role**. Выполните привязку кластерного типа (ClusterRoleBinding) ролей app-shard-resources-role и hazelcast-cluster-role к Service Account appshard-manager.
8. **Создание роли app-shard-resources-role-admin**. Создайте роль для просмотра ресурсов AppShardApi, AppShardEntry, AppShardApiReference.

Манифест для создания кластерной роли расположен по следующему пути:

```
<директория с дистрибутивом>/  documentation/installation-guide/manifest/roles/app-shard-resources-role-admin.yaml - роль для прикладных администраторов
```

9. **Создание роли app-shard-resources-role-admin-test**. Создайте роль для просмотра и редактирования ресурсов AppShardApi, AppShardEntry, AppShardApiReference в тестовых средах.

Манифест для создания кластерной роли расположен по следующему пути:

```
<директория с дистрибутивом>/   documentation/installation-guide/manifest/roles/app-shard-resources-role-admin-test.yaml - роль для прикладных администраторов в тестовых средах
```

Сборка образа контейнера

Ниже представлено описание шагов сборки образа контейнера. Настройка окружения при создании образа контейнера происходит автоматически.

Таблица 1. Шаги сборки образа контейнера

№	Шаг	Описание	Вход	Выход
1	Подготовка правил сборки	Создание Jenkinsfile и Dockerfile. Размещение их в сервисе централизованного хранения репозиториев исходного кода	Задание на разработку или актуализацию скриптов развертывания и (или) файлов с настройками	Правила сборки контейнера (Jenkinsfile, Dockerfile)
2	Передача правил в сборщик	Загрузка в Jenkins правил сборки из сервиса централизованного хранения репозиториев исходного кода. Трансляция правил в сборщик	Правила сборки контейнера (Jenkinsfile, Dockerfile). Запрос на создание новой версии образа контейнера	Задачи по сборке

№	Шаг	Описание	Вход	Выход
3	Сборка исходного кода	Дистрибутив может быть взят из репозитория Nexus-Public или собран на месте непосредственно из исходников. Возможен вариант, когда одновременно используется дистрибутив и сборка кода	Исходный код	Артефакт сборки кода (байт-код, машинный код и т. д.)
4	Определение маршрута	Jenkins Job определяет тип дистрибутива и (или) артефакта сборки кода и маршрутизирует конвейер	Дистрибутив и (или) артефакт сборки кода	Маршрут (Dev или Release)
5	Создание образа контейнера	Запускается сборка образа контейнера. Применение базового образа. Настройка окружения в соответствии с файлом Dockerfile. Инсталляция дистрибутива и (или) артефакта сборки кода	Дистрибутив и (или) артефакт сборки кода. Базовый образ. Задачи по сборке. Маршрут	Образ контейнера

№	Шаг	Описание	Вход	Выход
6	Публикация образа контейнера в репозитории	Полученный файл образа контейнера размещается в репозитории Nexus-Public (Dev или Release). Выход из данного процесса сопоставляется с выходом из этапа *публикация дистрибутива* и входом на этап *непрерывной поставки*	Образ контейнера	Образ контейнера, опубликованный в репозитории Nexus-Public (Dev или Release). Контрольная сумма образа контейнера, записанная в Nexus-Public

Установка

Схема установки компонент

Компонент AppShard Manager (ASM) предназначен для управления компонентом AppShard Router.

Компонент AppShard Router (ASR) предназначен для маршрутизации запросов в определенную шарду. Вычисление маршрута производится на основании конфигурации по различным стратегиям, описанным в этой конфигурации.

Для обеспечения взаимодействия с сервисами необходимо наличие Istio INGRESS и Istio EGRESS на базе Istio Service Mesh.

AppShard Router и AppShard Manager устанавливаются каждый в своем пространстве. Более детально схема установки изображена на рисунке 1.

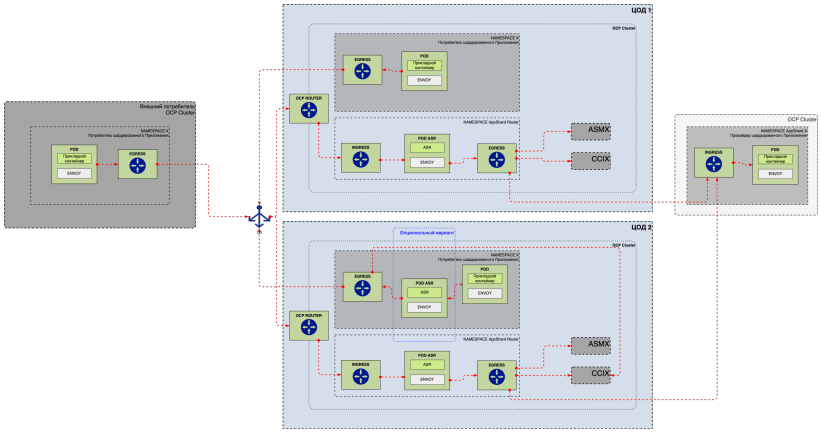


Рисунок 1. Схема установки компонент

Ручная установка

Установка AppShard Manager в среду управления контейнерами Kubernetes

Для установки AppShard Manager в среду управления контейнерами Kubernetes необходимо создать ресурсы с использованием шаблонов, указанных в соответствующих директориях, в зависимости от требуемой конфигурации.

Также необходимо задать значения параметров, используемых в данных шаблонах.

Список параметров и их значения приведены в таблице 2.

Список директорий с шаблонами ресурсов AppShard Manager

```
<директория с дистрибутивом>/  package/k8s/base/asm
#артефакты для установки приложения appshard-manager
package/k8s/base/asm/configmaps
#конфигурация для приложения appshard-manager
package/k8s/base/asm/istio/deployments/ingress
#артефакты для установки ingress
package/k8s/base/asm/istio/deployments/egress
#артефакты для установки egress
package/k8s/base/istio/config/ingress
#конфигурация установки ingress
package/k8s/base/istio/config/egress
#конфигурация установки egress
```

Список параметров AppShard Manager

Таблица 2. Параметры AppShard Manager

Параметр	Описание	Значение по умолчанию
asmx.asm.deployment.container.asm.limits.cpu	предельное значение CPU для контейнера ASM	300m
asmx.asm.deployment.container.asm.limits.memory	предельное количество памяти для контейнера ASM	1Gi
asmx.asm.container.metrics.port	номер порта prometheus для передачи внутренних (прикладных) метрик ASM	8081
asmx.asm.container.port	номер порта для соединения ASR с ASM	8091
asmx.asm.deployment.container.asm.requests.cpu	запрашиваемое значение CPU для контейнера ASM	300m
asmx.asm.deployment.container.asm.requests.memory	запрашиваемое количество памяти для контейнера ASM	1Gi
asmx.asm.group	имя группы ASM	none
asmx.asm.deployment.replicas.count	количество реплик ASM	3
asmx.component.name	имя (префикс) ресурсов ASM	appshard-manager
registry.host	адрес хранилища Docker-образов	registry.sigma.sbrf.ru
istio.egress.component.name	имя (префикс) Istio egress ASM	egress
istio.egress.selector	имя селектора Istio egress ASM	asm-egress-istio-selector
asmx.asm.deployment.container.fluent.bit.container.name	имя контейнера fluent-bit	fluent-bit-sidecar

Параметр	Описание	Значение по умолчанию
asmx.hazelcast.init.time.out	время в секундах, выделяемое для соединения реплик ASM в кластер Hazelcast. Если по истечении этого времени кластер Hazelcast не был сформирован, liveness/readiness проверки (probes) ASM будут отвечать отрицательно	240
asmx.hazelcast.retry.count	максимальное количество попыток реплик ASM соединиться в кластер Hazelcast	2000
asmx.hazelcast.service.name	имя сервиса, обеспечивающего работоспособность Hazelcast	asm-hazelcast
istio.ingress.component.name	имя (префикс) Istio ingress ASM	ingress
istio.ingress.selector	имя селектора Istio ingress ASM	asm-ingress-istio-selector
istio.control.plane.namespace	имя проекта, в котором находится IstioControlPlane	значение по умолчанию отсутствует, необходимо определить
istio.deployment.ingress.container.limits.cpu	предельное значение CPU для контейнера istio-proxy	200m
istio.deployment.ingress.container.limits.memory	предельное количество памяти для контейнера istio-proxy	512Mi
istio.deployment.ingress.container.requests.cpu	запрашиваемое значение CPU для контейнера istio-proxy	100m

Параметр	Описание	Значение по умолчанию
istio.deployment.ingress.container.requests.memory	запрашиваемое количество памяти для контейнера istio-proxy	256Mi
istio.deployment.egress.container.limits.cpu	предельное значение CPU для контейнера istio-proxy	200m
istio.deployment.egress.container.limits.memory	предельное количество памяти для контейнера istio-proxy	512Mi
istio.deployment.egress.container.requests.cpu	запрашиваемое значение CPU для контейнера istio-proxy	100m
istio.deployment.egress.container.requests.memory	запрашиваемое количество памяти для контейнера istio-proxy	256Mi
istio.deployment.egress.replicas.count	количество реплик Istio egress ASM	1
istio.deployment.ingress.replicas.count	количество реплик Istio ingress ASM	1
istio.egress.mtls.ca.certificate.path	путь в файловой системе до корневого сертификата egress	/etc/istio/egressgateway-ca-certs/ca.pem
istio.egress.mtls.certificate.path	путь в файловой системе до сертификата egress удостоверяющего центра	/etc/istio/egressgateway-certs/tls.pem
istio.egress.mtls.private.key.path	путь в файловой системе до приватного ключа сертификата egress	/etc/istio/egressgateway-certs/tls.key
istio.asm.ingress.host	адрес Istio ingress ASM	значение по умолчанию отсутствует, необходимо определить

Параметр	Описание	Значение по умолчанию
istio.ingress.mtls.ca.certificate.path	путь в файловой системе до корневого сертификата ingress	/etc/istio/ingressgateway-ca-certs/ca.pem
istio.ingress.mtls.certificate.path	путь в файловой системе до сертификата ingress удостоверяющего центра	/etc/istio/ingressgateway-certs/tls.pem
istio.ingress.mtls.private.key.path	путь в файловой системе до приватного ключа сертификата ingress	/etc/istio/ingressgateway-certs/tls.key
istio.ingress.secret.ca	имя секрета, содержащего корневой сертификат	istio-outer-mtls-ca
istio.egress.secret.client	имя секрета, содержащего клиентский сертификат	istio-outer-mtls-client
istio.ingress.secret.server	имя секрета, содержащего сертификат удостоверяющего центра	istio-outer-mtls
istio.envoy.image	путь до Docker-образа istio-proxy (вместе с адресом Docker-registry)	значение по умолчанию отсутствует, необходимо определить
asmx.istio.sidecar.inject	логическая переменная, отвечающая за подключение к Istio Service Mesh	true
asmx.asm.deployment.env.istio.version	используемая версия Istio	v1beta1
asmx.kube.api.ip	адрес для доступа к K8s API	172.30.0.1
asmx.kube.api.port	порт для доступа к K8s API	443

Параметр	Описание	Значение по умолчанию
asmx.asm.deployment.container.asm.liveness.probe.failure.threshold	количество повторных попыток получить положительную liveness пробу перед тем, как попытаться запустить приложение снова	10
asmx.asm.deployment.container.asm.liveness.probe.initial.delay.seconds	количество секунд от старта контейнера до начала запроса liveness проб	15
asmx.asm.deployment.container.asm.liveness.probe.period.seconds	длительность времени (в секундах) между двумя последовательными запросами liveness проб	30
asmx.asm.deployment.container.asm.liveness.probe.success.threshold	минимальное количество последовательных проверок liveness проб, чтобы проба считалась успешной после неудачной	1
asmx.asm.deployment.container.asm.liveness.probe.timeout.seconds	количество секунд ожидания liveness проверки (probe)	2
istio.ott.config.map.data.module.id	идентификатор модуля ASM в настройках компонента One-Time Password (OTP) / OTT (OTTS)	appshard-manager-cloud
asmx.namespace	имя проекта ASM	значение по умолчанию отсутствует, необходимо определить

Параметр	Описание	Значение по умолчанию
asmx.ott.certstore.path	путь в файловой системе до хранилища сертификатов (CertStore) OTT	/usr/config/ott/cert.p12
istio.ott.client.hosts	адреса серверов OTT	значение по умолчанию отсутствует, необходимо определить
istio.ott.http.port	порт доступа OTT	8086
istio.ott.service.url	URL вида https://ott-server-host:port/ott-service/rest/token, где ott-server-host:port - один из хостов из параметра OTT_SERVICE_HOSTS	значение по умолчанию отсутствует, необходимо определить
istio.deployment.ott.sidecar.limits.cpu	предельное значение CPU для контейнера ott-sidecar	200m
istio.deployment.ott.sidecar.requests.cpu	запрашиваемое значение CPU для контейнера ott-sidecar	100m
istio.ott-sidecar.image.path	путь до Docker-образа ott-sidecar (вместе с адресом Docker-registry)	значение по умолчанию отсутствует, необходимо определить
istio.deployment.ott.sidecar.limits.memory	предельное количество памяти для контейнера ott-sidecar	400Mi
istio.deployment.ott.sidecar.requests.memory	запрашиваемое количество памяти для контейнера ott-sidecar	200Mi

Параметр	Описание	Значение по умолчанию
asmx.ott.trust.store.path	путь в файловой системе до хранилища криптографических публичных и приватных ключей и сертификатов (TrustStore) OTT	/usr/config/ott/trust.p12
asmx.asm.deployment.container.asm.readiness.probe.failure.threshold	количество повторных попыток получить положительную readiness пробу перед тем, как POD будет помечен, как недоступный (unready)	4
asmx.asm.deployment.container.asm.readiness.probe.initial.delay.seconds	количество секунд от старта контейнера до начала запроса readiness проб	15
asmx.asm.deployment.container.asm.readiness.probe.period.seconds	длительность времени (в секундах) между двумя последовательными запросами readiness проб	30
asmx.asm.deployment.container.asm.readiness.probe.success.threshold	минимальное количество последовательных проверок readiness проб, чтобы проба считалась успешной после неудачной	1
asmx.asm.deployment.container.asm.readiness.probe.timeout.seconds	количество секунд ожидания readiness проверки (probe)	2

Установка AppShard Router в среду управления контейнерами Kubernetes

Для установки AppShard Router в среду управления контейнерами Kubernetes необходимо создать ресурсы с использованием шаблонов, указанных в соответствующих директориях, в зависимости от требуемой конфигурации.

Также необходимо задать значения параметров, используемых в данных шаблонах.

Список параметров и их значения приведены в таблице 3.

Список директорий с шаблонами ресурсов AppShard Router

```
<директория с дистрибутивом>/  package/k8s/base/asr
#артефакты для установки приложения fort-asr
package/k8s/base/asr/configmaps
#конфигурация для приложения fort-asr
package/k8s/base/asr/istio/deployments/ingress
#артефакты для установки ingress
package/k8s/base/asr/istio/deployments/egress
#артефакты для установки egress
package/k8s/base/istio/config/ingress
#конфигурация установки ingress
package/k8s/base/istio/config/egress
#конфигурация установки egress
```

Список параметров AppShard Router

Таблица 3. Параметры AppShard Router

Параметр	Описание	Значение по умолчанию
asrx.asm.host	адрес ASM	значение по умолчанию отсутствует, необходимо определить
asrx.asm.istio.asm.name	имя для создания компонентов istio для подключения к ASM	asm
asrx.role.binding.asm.service.account.namespace	имя проекта ASM	значение по умолчанию отсутствует, необходимо определить
asrx.asm.port	порт ASM	8091
asrx.asr.deployment.container.asr.limits.cpu	предельное значение CPU для контейнера ASR	800m

Параметр	Описание	Значение по умолчанию
asrx.asr.deployment.container.asr.limits.memory	предельное количество памяти для контейнера ASR	1536Mi
asrx.asr.container.metrics.port	номер порта prometheus для передачи внутренних (прикладных) метрик ASR	8060
asrx.asr.deployment.container.asr.requests.cpu	запрашиваемое значение CPU для контейнера ASR	600m
asrx.asr.deployment.container.asr.requests.memory	запрашиваемое количество памяти для контейнера ASR	1024Mi
asrx.asr.container.server.port	номер порта сервера ASR	8080
asrx.asr.deployment.replicas.count	количество реплик ASR	2
asrx.component.name	имя (префикс) ресурсов ASR	asr-fort
registry.host	адрес хранилища Docker-образов	значение по умолчанию отсутствует, необходимо определить
asrx.egress.component.name	имя (префикс) Istio egress ASR	egress
asrx.egress.istio.selector	имя селектора Istio egress ASR	asr-egress-istio-selector
asrx.egress.service.port.number	порт сервиса istio egress	8701
asrx.egress.service.port.name	имя порта сервиса istio egress	http-8701
asrx.fluent.bit.config.map.name	имя ConfigMap настроек fluent-bit	fluent-bit-config
asrx.asr.deployment.container.fluent.bit.container.name	имя контейнера fluent-bit	fluent-bit-sidecar

Параметр	Описание	Значение по умолчанию
asrx.asr.deployment.container.fluent.bit.image.path	путь до Docker-образа fluent-bit (внутри Docker-registry)	значение по умолчанию отсутствует, необходимо определить
asrx.asr.deployment.container.fluent.bit.limits.cpu	предельное значение CPU для контейнера fluent-bit	200m
asrx.asr.deployment.container.fluent.bit.limits.memory	предельное количество памяти для контейнера fluent-bit	32Mi
asrx.asr.deployment.container.fluent.bit.requests.cpu	запрашиваемое значение CPU для контейнера fluent-bit	100m
asrx.asr.deployment.container.fluent.bit.requests.memory	запрашиваемое количество памяти для контейнера fluent-bit	16Mi
asrx.ingress.component.name	имя (префикс) Istio ingress ASR	ingress
asrx.ingress.istio.selector	имя селектора Istio ingress ASR	asr-ingress-istio-selector
asrx.istio.control.plane.namespace	имя проекта, в котором находится IstioControlPlane	значение по умолчанию отсутствует, необходимо определить
asrx.istio.deployment.ingress.container.limits.cpu	предельное значение CPU для контейнера istio-proxy	200m
asrx.istio.deployment.ingress.container.limits.memory	предельное количество памяти для контейнера istio-proxy	512Mi
asrx.istio.deployment.ingress.container.requests.cpu	запрашиваемое значение CPU для контейнера istio-proxy	100m
asrx.istio.deployment.ingress.container.requests.memory	запрашиваемое количество памяти для контейнера istio-proxy	256Mi

Параметр	Описание	Значение по умолчанию
asrx.istio.deployment.egress.container.limits.cpu	предельное значение CPU для контейнера istio-proxy	200m
asrx.istio.deployment.egress.container.limits.memory	предельное количество памяти для контейнера istio-proxy	512Mi
asrx.istio.deployment.egress.container.requests.cpu	запрашиваемое значение CPU для контейнера istio-proxy	100m
asrx.istio.deployment.egress.container.requests.memory	запрашиваемое количество памяти для контейнера istio-proxy	256Mi
asrx.istio.deployment.egress.replicas.count	количество реплик Istio egress ASR	1
asrx.istio.deployment.ingress.replicas.count	количество реплик Istio ingress ASR	1
asrx.istio.sidecar.limits.cpu	предельное значение CPU для контейнера istio-sidecar	200m
asrx.istio.sidecar.limits.memory	предельное количество памяти для контейнера istio-sidecar	512Mi
asrx.istio.sidecar.requests.cpu	запрашиваемое значение CPU для контейнера istio-sidecar	100m
asrx.istio.sidecar.requests.memory	запрашиваемое количество памяти для контейнера istio-sidecar	256Mi
asrx.istio.egress.mtls.ca.certificate.path	путь в файловой системе до корневого сертификата egress	/etc/istio/egressgateway-ca-certs/ca.pem

Параметр	Описание	Значение по умолчанию
asrx.istio.egress.mtls.certificate.path	путь в файловой системе до сертификата egress удостоверяющего центра	/etc/istio/egressgateway-certs/tls.pem
asrx.istio.egress.mtls.private.key.path	путь в файловой системе до приватного ключа сертификата egress	/etc/istio/egressgateway-certs/tls.key
istio.asr.ingress.host	адрес Istio ingress ASR для входящих запросов	значение по умолчанию отсутствует, необходимо определить
istio.asr.ingress.https.port	порт istio ingress	2443
asrx.istio.ingress.mtls.ca.certificate.path	путь в файловой системе до корневого сертификата ingress	/etc/istio/ingressgateway-ca-certs/ca.pem
asrx.istio.ingress.mtls.certificate.path	путь в файловой системе до сертификата ingress удостоверяющего центра	/etc/istio/ingressgateway-certs/tls.pem
asrx.istio.ingress.mtls.private.key.path	путь в файловой системе до приватного ключа сертификата ingress	/etc/istio/ingressgateway-certs/tls.key
asrx.istio.mtls.secret.ca	имя секрета, содержащего корневой сертификат	istio-outer-mtls-ca
asrx.istio.mtls.secret.client	имя секрета, содержащего клиентский сертификат	istio-outer-mtls-client
asrx.istio.mtls.secret.server	имя секрета, содержащего сертификат удостоверяющего центра	istio-outer-mtls
asrx.istio.proxy.image	путь до Docker-образа istio-proxy (вместе с адресом Docker-registry)	значение по умолчанию отсутствует, необходимо определить

Параметр	Описание	Значение по умолчанию
asrx.istio.sidecar.inject	логическая переменная, отвечающая за подключение к Istio Service Mesh	true
asrx.asr.deployment.container.asr.liveness.probe.failure.threshold	количество повторных попыток получить положительную liveness пробу перед тем, как попытаться запустить приложение снова	10
asrx.asr.deployment.container.asr.liveness.probe.initial.delay.seconds	количество секунд от старта контейнера до начала запроса liveness проб	15
asrx.asr.deployment.container.asr.liveness.probe.period.seconds	длительность времени (в секундах) между двумя последовательными запросами liveness проб	30
asrx.asr.deployment.container.asr.liveness.probe.success.threshold	минимальное количество последовательных проверок liveness проб, чтобы проба считалась успешной после неудачной	1
asrx.asr.deployment.container.asr.liveness.probe.timeout.seconds	количество секунд ожидания liveness проверки (probe)	2
asrx.logger.component.name	имя (префикс) ресурсов журналирования ASR	logger
asrx.logger.endpoint.host	адрес сервиса приема событий журналирования	значение по умолчанию отсутствует, необходимо определить
asrx.logger.endpoint.port	порт сервиса приема событий журналирования	8703

Параметр	Описание	Значение по умолчанию
asrx.ott.config.map.data.module.id	идентификатор модуля ASR в настройках OTT	appshard-router-cloud
asrx.namespace	имя проекта ASR	значение по умолчанию отсутствует, необходимо определить
asrx.ott.certstore.path	путь в файловой системе до хранилища сертификатов (CertStore) компонента One-Time Password (OTP) / OTT (OTTS)	/usr/config/ott/cert.p12
asrx.ott.client.hosts	адреса серверов OTT	значение по умолчанию отсутствует, необходимо определить
asrx.ott.http.port	порт доступа OTT	8086
asrx.ott.service.url	URL вида https://ott-server-host:port/ott-service/rest/token, где ott-server-host:port - один из хостов из параметра OTT_SERVICE_HOSTS	значение по умолчанию отсутствует, необходимо определить
asrx.istio.deployment.ott.sidecar.limits.cpu	предельное значение CPU для контейнера ott-sidecar	200m
asrx.istio.deployment.ott.sidecar.requests.cpu	запрашиваемое значение CPU для контейнера ott-sidecar	100m
asrx.ott.sidecar.image	путь до Docker-образа ott-sidecar (вместе с адресом Docker-registry)	значение по умолчанию отсутствует, необходимо определить

Параметр	Описание	Значение по умолчанию
asrx.istio.deployment.ott.sidecar.limits.memory	предельное количество памяти для контейнера ott-sidecar	400Mi
asrx.istio.deployment.ott.sidecar.requests.memory	запрашиваемое количество памяти для контейнера ott-sidecar	200Mi
asrx.ott.trust.store.path	путь в файловой системе до хранилища криптографических публичных и приватных ключей и сертификатов (TrustStore) OTT	/usr/config/ott/trust.p12
asrx.asr.deployment.container.asr.readiness.probe.failure.threshold	количество повторных попыток получить положительную readiness пробу перед тем, как POD будет помечен, как недоступный (unready)	4
asrx.asr.deployment.container.asr.readiness.probe.initial.delay.seconds	количество секунд от старта контейнера до начала запроса readiness проб	15
asrx.asr.deployment.container.asr.readiness.probe.period.seconds	длительность времени (в секундах) между двумя последовательными запросами readiness проб	30
asrx.asr.deployment.container.asr.readiness.probe.success.threshold	минимальное количество последовательных проверок readiness проб, чтобы проба считалась успешной после неудачной	1

Параметр	Описание	Значение по умолчанию
asrx.asr.deployment.container.asr.readiness.probe.timeout.seconds	количество секунд ожидания readiness проверки (probe)	2
asrx.asr.deployment.container.asr.graceful.shutdown.timeout	количество секунд ожидания начала graceful	7

Установка в среду управления контейнерами OpenShift

В среде управления контейнерами OpenShift вместо ресурсов типа Ingress используются ресурсы типа Route, поэтому для установки в среду управления контейнерами OpenShift необходимо воспользоваться манифестами для создания ресурсов типа Route, расположенными по следующему пути:

- AppShard Manager:

```
<директория с дистрибутивом>/
package/k8s/overrides/openshift/asm/istio/deployments/ingres
s      #манифест для создания Route для приложения appshard-
manager
```

- AppShard Router:

```
<директория с дистрибутивом>/
package/k8s/overrides/openshift/asr/istio/deployments/ingres
s      #манифест для создания Route для приложения fort-asr
```

Выпуск и установка сертификатов для Ingress/Egress

Используя шаблоны из дистрибутива, выпустите и установите сертификаты Istio Ingress/Egress. Шаблоны – см. разделы Список директорий с шаблонами ресурсов AppShard Manager и Список директорий с шаблонами ресурсов AppShard Router соответственно.

После чего будет создан набор необходимых артефактов, в том числе секреты с сертификатами cert, ca-cert.

Созданные секреты содержат, в том числе, наименования файлов: tls.crt, tls.key, ca-chain.cert.pem (вся цепочка сертификатов).

В секретах istio-ingressgateway-ca-certs и istio-egressgateway-ca-certs должен присутствовать один файл ca-chain.cert.pem со всей цепочкой сертификатов. Если в

секрете указаны два отдельных сертификата, то вызовы через Ingress по HTTPS не будут работать.

Возможен вариант, когда в секретах в файле ca-chain.cert.pem содержится вся цепочка сертификатов (т.е. добавлено содержимое из ca.pem), но сам файл ca.pem не удален. В этом случае все будет работать корректно.

Установка и параметры настройки компонента Журналирование (LOGA)

Для хранения лог-файлов выполните установку компонента Журналирование (LOGA) продукта Platform V Monitor (OPM), далее – сервис Журналирование. Установка выполняется автоматически при разворачивании AppShard Manager и AppShard Router в виде sidecar-контейнера fluent-bit в deployment приложения.

Для настройки сервиса Журналирование воспользуйтесь таблицами ниже, в которых описаны параметры для работы fluent-bit.

Таблица 4. Параметры настройки сервиса Журналирование при установке приложения AppShard Manager

Параметр	Описание	Значение по умолчанию
asmx.asm.deployment.container.fluent.bit.image.path	путь до Docker-образа fluent-bit (внутри Docker-registry)	значение по умолчанию отсутствует, необходимо определить
asmx.asm.deployment.container.fluent.bit.limits.cpu	предельное значение CPU для контейнера fluent-bit	200m
asmx.asm.deployment.container.fluent.bit.limits.memory	предельное количество памяти для контейнера fluent-bit	32Mi
asmx.asm.deployment.container.fluent.bit.requests.cpu	запрашиваемое значение CPU для контейнера fluent-bit	100m
asmx.asm.deployment.container.fluent.bit.requests.memory	запрашиваемое количество памяти для контейнера fluent-bit	16Mi
asmx.logback.config.map.name	имя ConfigMap настроек журналирования	logback-config
istio.egress.logger.component.name	имя (префикс) ресурсов журналирования ASM	logger

Параметр	Описание	Значение по умолчанию
istio.egress.logger.endpoint.host	адрес сервиса приема событий журналирования	значение по умолчанию отсутствует, необходимо определить
istio.egress.logger.endpoint.port	порт сервиса приема событий журналирования	8703
asmx.fluent.bit.config.map.data.stand	имя стенда (для журналирования)	appshard-manager
asmx.fluent.bit.config.map.data.stand.id	идентификатор стенда (для журналирования)	appshard-manager
asmx.fluent.bit.sidecar.env.zone.id	идентификатор зоны (для журналирования)	default
asmx.fluent.bit.config.map.name	имя ConfigMap настроек fluent-bit	fluent-bit-config
asmx.fluent.bit.config.map.data.cluster	имя кластера	значение по умолчанию отсутствует, необходимо определить

Таблица 5. Параметры настройки сервиса Журналирование при установке приложения AppShard Router

Параметр	Описание	Значение по умолчанию
asrx.asr.deployment.container.fluent.bit.image.path	путь до Docker-образа fluent-bit (внутри Docker-registry)	значение по умолчанию отсутствует, необходимо определить
asrx.asr.deployment.container.fluent.bit.limits.cpu	предельное значение CPU для контейнера fluent-bit	200m
asrx.asr.deployment.container.fluent.bit.limits.memory	предельное количество памяти для контейнера fluent-bit	32Mi
asrx.asr.deployment.container.fluent.bit.requests.cpu	запрашиваемое значение CPU для контейнера fluent-bit	100m

Параметр	Описание	Значение по умолчанию
asrx.asr.deployment.container.fluent.bit.requests.memory	запрашиваемое количество памяти для контейнера fluent-bit	16Mi
asrx.logger.component.name	имя (префикс) ресурсов журналирования ASM	logger
asrx.logger.endpoint.host	адрес сервиса приема событий журналирования	значение по умолчанию отсутствует, необходимо определить
asrx.logger.endpoint.port	порт сервиса приема событий журналирования	8703
asrx.fluent.bit.config.map.data.stand	имя стенда (для журналирования)	asr-fort
asrx.fluent.bit.config.map.data.stand.id	идентификатор стенда (для журналирования)	asr-fort
asrx.fluent.bit.sidecar.env.zone.id	идентификатор зоны (для журналирования)	default
asrx.fluent.bit.config.map.data.cluster	имя кластера	значение по умолчанию отсутствует, необходимо определить

Установка и параметры настройки компонента Объединенный мониторинг Unimon (MONA)

Для сбора и публикации метрик используется компонент Объединенный мониторинг Unimon (MONA) продукта Platform V Monitor (OPM), далее – сервис Объединенный мониторинг Unimon. Для установки необходимо:

1. Скачать дистрибутив сервиса Объединенный мониторинг Unimon;
2. Выполнить его установку в проект приложения AppShard Manager;
3. Выполнить его установку в проект приложения AppShard Router;
4. Выполнить настройку приложения unimon-agent для сбора метрик;
5. Выполнить настройку unimon-sender для публикации метрик.

Более подробно об установке и настройке сервиса Объединенный мониторинг Unimon читайте в документе *Руководство по установке* сервиса Объединенный мониторинг Unimon.

Автоматизированная установка сервиса с использованием DevOps Tools

Для установки сервиса с использованием компонента Deploy Tools (CDJE) программного продукта Platform V DevOps Tools (DOT) разверните сервис в инфраструктуре и запустите CD Job по инструкции, приведенной в документации Platform V DevOps Tools (DOT).

Параметры для установки описаны в разделах ручной установки в таблицах 2, 3, 4 и 5.

Требования по безопасности

AppShard Manager работает под специальной сервисной учетной записью, которой необходимы привилегии на получение и редактирование ресурсов k8s.

Для этого необходимо предоставить сервисной учетной записи следующие привилегии, описанные в таблице 6.

Таблица 6. Перечень привилегий для Service Account appshard-manager

Api Groups (группа ресурсов)	Resource (ресурсы Kubernetes или Openshift)	Verbs (привилегии)	Назначение verb (назначение привилегий)
appshard.platform-v	endpoints	get list watch create update patch	используются для механизма выбора лидера, который базируется на идее взятия блокировок через endpoints. Выбор лидера среди AppShard Manager позволяет уйти от возможности параллельной записи ресурсов в k8s различными инстансами AppShard Manager, что может привести к несогласованности публикуемых данных
appshard.platform-v	appshardapis appshardentries	get list watch	appshardapis и appshardentries являются внутренними ресурсами Appsharding и используются для хранения непровалидированной информации об API и шардах

Api Groups (группа ресурсов)	Resource (ресурсы Kubernetes или Openshift)	Verbs (привилегии)	Назначение verb (назначение привилегий)
appshard.platform-v	appshardapis/status appshardentries/status	create delete get patch update	используется для возможности работы со статусами внутренних ресурсов Appsharding, необходимых для отслеживания информации о валидации на стороне AppShard Manager
appshard.platform-v	appshardapirefs	create delete deletecollection get list patch update watch	appshardapirefs является внутренним ресурсом Appsharding и используется для хранения провалидированной и сгруппированной информации об API и адресах шард. Данным ресурсом должен иметь права управлять только AppShard Manager, иначе внешнее редактирование данного ресурса может сломать работу Appsharding

Api Groups (группа ресурсов)	Resource (ресурсы Kubernetes или Openshift)	Verbs (привилегии)	Назначение verb (назначение привилегий)
appshard.platform-v	events	create delete deletecollection get list patch update watch	используется для возможности публикации AppShard Manager событийных записей (logs) в хранилище kubernetes, на которые в дальнейшем могут опираться администраторы стенда для отслеживания состояния системы Appsharding
appshard.platform-v	endpoints pods nodes services	get list	необходимо для работы библиотеки распределенного хранилища данных Hazelcast

Api Groups (группа ресурсов)	Resource (ресурсы Kubernetes или Openshift)	Verbs (привилегии)	Назначение verb (назначение привилегий)
networking.istio.io	virtualservices gateways serviceentries destinationrules	get create update delete list	для предоставления AppShard Manager возможности публикации и управления Istio конфигурациями маршрутизации до адресов шард необходимо выдать привилегии на Service Account AppShard Manager в namespace AppShard Router

Для функционирования сервиса используется ролевая модель на уровне среды контейнеризации. Примеры ролей (с типовыми наборами привилегий) можно посмотреть по следующему пути в дистрибутиве: /package/conf/asm/roles.

Обновление

Для обновления компонент AppShard Manager и AppShard Router на текущую версию необходимо:

1. Остановить запущенные отсеки (pods) приложений AppShard Manager, AppShard Router, Ingress, Egress, unimon-agent, unimon-sender.
2. Удалить ресурсы типа Deployment, ConfigMap, Secret, Route, Ingress, Egress, VirtualService, ServiceEntry, DestinationRule, Gateway, Service, созданных при установке предыдущей версии.
3. Установить текущую версию приложения в соответствии с настоящим руководством по установке.

Проверка работоспособности

Чтобы проверить работоспособность сервиса шардирования приложений, необходимо проверить работоспособность AppShard Manager и AppShard Router.

Проверка работоспособности AppShard Manager

AppShard Manager успешно установлен, если:

1. Все отсеки (pods) поднялись, т.е. все отсеки (pods) находятся в статусе **running**.
2. Все отсеки (pods) по liveness (работоспособности) и readiness (готовности) проверкам (probes) не перезапускаются через время, указанное в свойстве hazelcast.init.timeout, и параметры (probes) liveness и readiness имеют значения true.
3. В записях (logs) AppShard Manager есть строка Asm context was initialized.
4. В записях (logs) AppShard Manager есть строка New subscriber: {asrPodName}, значит к AppShard Manager подключился AppShard Router, имя отсека (pod) которого указано в скобках.

Проверка работоспособности AppShard Router

AppShard Router успешно установлен, если:

1. Все отсеки (pods) поднялись, т.е. все отсеки (pods) находятся в статусе **running**.
2. Все отсеки (pods) по liveness (работоспособности) и readiness (готовности) проверкам (probes) не перезапускаются, параметры (probes) liveness и readiness имеют значения true.
3. В процессе запуска AppShard Router запустилось несколько подсистем AppShard Router в разных процессах:

	процесс proxy (выполняет проксирование)	**процесс мониторинга (аккумулирует в себе метрики и логи разных процессов)**	**процесс роутера (выполняет поиск шарды по стратегии шардирования)**
о запуске указывают записи (logs):	<14> 2021-10-28T12:43:02+00:00 info asr-fort-1-fmqkg fort/proxy 1 listen: 0.0.0.0:8080	<14> 2021-10-28T12:43:02+00:00 info asr-fort-1-fmqkg fort/monitoring 1 listen: 127.0.0.1:1514	<14> 2021-10-28T12:43:02+00:00 info asr-fort-1-fmqkg fort/router 1 listen: 0.0.0.0:9051
	<12> 2021-10-28T12:43:33+00:00 warning asr-fort-1-fmqkg fort/proxy 9 config: read file '/opt/shared_config/mime_types_json/mime_types.json'	<14> 2021-10-28T12:43:02+00:00 info asr-fort-1-fmqkg fort/monitoring 1 2 sensor(s) configured	<14> 2021-10-28T12:43:33+00:00 info asr-fort-1-fmqkg fort/proxy 9 router: enabled

	процесс proxy (выполняет проксирование)	процесс мониторинга (аккумулирует в себе метрики и логи разных процессов)	процесс роутера (выполняет поиск шарды по стратегии шардирования)
	<12> 2021-10-28T12:43:33+00:00 warning asr-fort-1-fmqkg fort/proxy 9 config: read file '/opt/fort/config/include.json'	<14> 2021-10-28T12:43:02+00:00 info asr-fort-1-fmqkg fort/monitoring 14 transport(s) configured	–
	<12> 2021-10-28T12:43:33+00:00 warning asr-fort-1-fmqkg fort/proxy 9 config: read file '/opt/shared_config/fort_proxy_config/fort-proxy-config.json'	<14> 2021-10-28T12:43:33+00:00 info asr-fort-1-fmqkg fort/proxy 9 monitoring: enabled	–
	<14> 2021-10-28T12:43:34+00:00 warning asr-fort-1-fmqkg fort/proxy 9 proxy: enabled	–	–

Количество подобных записей (logs) соответствует количеству запущенных слушателей (listener).

Номер порта, на котором поднимается listener, указывается в конфигурационном файле fort-proxy-config.json, который расположен в configmap asr-fort-config.

4. После запуска AppShard Router подключился по протоколу gRPC к AppShard Manager. Если в процессе подключения возникают ошибки, выводится запись (log):

```
<11> 2021-10-28T12:26:11+00:00 error asr-fort-1-9zvv7
fort/router 11  grpc client: gRPC connection error has
occurred, <error code>, <error text>
```

Попытки подключения повторяются каждые 5 секунд. Если AppShard Router не может подключиться к AppShard Manager, такая запись будет появляться каждые 5 секунд.

Если AppShard Router подключился к AppShard Manager и в ответ получил конфигурацию шардирования, то отобразится запись:

```
<14> 2021-10-28T12:43:33+00:00 info asr-fort-1-fmqkg
fort/router 10  start create new APP_SHARD_CONFIG      <14>
2021-10-28T12:43:33+00:00 info asr-fort-1-fmqkg fort/router
10  get 63 api elements                    <14> 2021-10-
28T12:43:33+00:00 info asr-fort-1-fmqkg fort/router 10
appShardConfig was created  <14> 2021-10-28T12:43:33+00:00
info asr-fort-1-fmqkg fort/router 10  end create new
APP_SHARD_CONFIG  <14> 2021-10-28T12:43:33+00:00 info asr-
fort-1-fmqkg fort/router 10  shards in proxy was updated
```

Процесс proxy с новой конфигурацией будет перезапущен, о чем сообщит запись (log) о запуске процесса proxy, указанная выше.

5. Ошибок с уровнем логирования error в журналах записей (logs) не наблюдается.

Чек-лист проверки работоспособности интеграции

При осуществлении интеграции (опциональных интеграций) Сервиса шардирования приложений с платформенными компонентами проверка работоспособности осуществляется согласно следующим инструкциям:

- компонент Объединенный мониторинг Unimon;
- компонент Журналирование.

Проверка работоспособности компонента Объединенный мониторинг Unimon

Для проверки подключения компонента Объединенный мониторинг Unimon убедитесь, что компоненты мониторинга установлены и работают, что в записях (logs) есть сообщения об успешной установке сессии, есть подключение к kafka, метрики доступны у приложения по определенному порту:

1. В среде контейнеризации перейдите на вкладку **PODs** и выберите pod.
2. Перейдите в терминал и введите команду

```
curl localhost:8081/actuator/prometheus
```

3. Сравните полученное сообщение с отбрасываемыми метриками с событиями мониторинга, приведенными в «Руководстве по системному администрированию» в разделе «События мониторинга».

4. Перейдите в систему отображения метрик (например, Prometheus), укажите необходимую метрику.

Проверка работоспособности компонента Журналирование

Для проверки подключения компонента Журналирование убедитесь, что есть fluent-bit sidecar и у него нет ошибок публикации журналов. Дополнительно:

1. Перейти в АРМ Журналирование.
2. Перейти на вкладку **Системный журнал** и на открывшейся странице заполните поля: **Начальная дата**, **Начальное время**, **Конечная дата**, **Конечное время**.
3. Нажмите кнопку **Поиск**.
4. Если в результатах поиска присутствуют записи, созданные по времени после запуска сервиса, то проверка выполнена успешно.

Откат

Для отката к стабильной версии сервиса шардирования приложений необходимо:

1. Остановить запущенные отсеки (pods) приложений AppShard Manager, AppShard Router, Ingress, Egress, unimon-agent, unimon-sender.
2. Удалить ресурсы типа Deployment, ConfigMap, Secret, Route, Ingress, Egress, VirtualService, ServiceEntry, DestinationRule, Gateway, Service, созданные при установке предыдущей версии.
3. Установить последнюю стабильную версию приложения в соответствии с документом *Руководством по установке* соответствующей версии.

Часто встречающиеся проблемы и пути их устранения

Таблица 7. Часто встречающиеся проблемы и пути их устранения

Проблема	Причины	Пути устранения
Перезапуск pod	Ошибка при поднятии gRPC-сервера	Проверить наличие ресурса Service с именем appshard-manager в namespace ASM
Перезапуск pod	Недостаточно привилегий либо привязка роли выполнена не на уровне кластера (не сделан ClusterRoleBinding)	Привязать роль с набором необходимых привилегий (см. раздел *Требования по безопасности*), убедиться, что для привязки роли используется ресурс с типом ClusterRoleBinding

Проблема	Причины	Пути устранения
Перезапуск pod через время, указанное в значении параметра `hazelcast.init.timeout` в environment DeploymentConfig ASM	Инстансы ASM не сформировали кластер Hazelcast	Проверить наличие ресурса Service с именем asm-hazelcast. Привязать роль с набором необходимых привилегий (см. раздел *Требования по безопасности*), убедиться, что для привязки роли используется ресурс с типом ClusterRoleBinding

Чек-лист валидации установки

Чек-лист валидации установки AppShard Manager

1. Необходимо убедиться, что в среде управлениями контейнерами были установлены следующие артефакты для AppShard Manager:

```
DeploymentConfig (${COMPONENT_NAME}) Service
(${COMPONENT_NAME}) Service (${HAZELCAST_SERVICE}) ConfigMap
(${FLUENT_BIT_CONFIG_MAP_NAME}) ConfigMap
(${LOGBACK_CONFIG_MAP_NAME}) ConfigMap (${COMPONENT_NAME}-
monitoring) ServiceAccount (appshard-manager) - опционально
Deployment (${EGRESS_COMPONENT_NAME}) Service
(${EGRESS_COMPONENT_NAME}) DestinationRule
(${LOGGER_COMPONENT_NAME}) DestinationRule
(${LOGGER_COMPONENT_NAME}-virtual) Gateway
(${LOGGER_COMPONENT_NAME}) VirtualService
(${LOGGER_COMPONENT_NAME}) ServiceEntry
(${LOGGER_COMPONENT_NAME}) Deployment
(${INGRESS_COMPONENT_NAME}) Service
(${INGRESS_COMPONENT_NAME}) ServiceEntry (kube-api)
DestinationRule (enable-mtls) Route
(${INGRESS_COMPONENT_NAME}) Gateway
(${INGRESS_COMPONENT_NAME}) VirtualService
(${INGRESS_COMPONENT_NAME})
```

Если производилась установка с параметрами по умолчанию, то их значения будут следующими:

Таблица 8. Значения параметров по умолчанию для AppShard Manager

Параметр	Значение
COMPONENT_NAME	appshard-manager
HAZELCAST_SERVICE	asm-hazelcast
FLUENT_BIT_CONFIG_MAP_NAME	fluent-bit-config
LOGBACK_CONFIG_MAP_NAME	logback-config
EGRESS_COMPONENT_NAME	egress
LOGGER_COMPONENT_NAME	logger
INGRESS_COMPONENT_NAME	ingress

2. В случае установки компонента AppShard Manager с компонентом One-Time Password (OTP) / OTT (OTTS), среди установленных артефактов необходимо проверить наличие следующих:

```
EnvoyFilter (${LOGGER_COMPONENT_NAME}) ConfigMap
(${EGRESS_COMPONENT_NAME}-ott-config)  ConfigMap
(${EGRESS_COMPONENT_NAME}-ott-logback)
```

Чек-лист валидации установки AppShard Router

1. Необходимо убедиться, что в среде управлениями контейнерами были установлены следующие артефакты для AppShard Router:

```
Service (${COMPONENT_NAME}) Service
(${EGRESS_COMPONENT_NAME}) Service
(${INGRESS_COMPONENT_NAME}) ConfigMap
(${FLUENT_BIT_CONFIG_MAP_NAME}) ConfigMap
(${COMPONENT_NAME}-monitoring) ConfigMap (${COMPONENT_NAME}-
config) Deployment (${EGRESS_COMPONENT_NAME}) Deployment
(${INGRESS_COMPONENT_NAME}) DestinationRule
(${ASM_ISTIO_COMPONENT_NAME}-${EGRESS_COMPONENT_NAME})
DestinationRule (${ASM_ISTIO_COMPONENT_NAME}-
${EGRESS_COMPONENT_NAME}-virtual) DestinationRule
(${LOGGER_COMPONENT_NAME}) DestinationRule
(${LOGGER_COMPONENT_NAME}-virtual) DestinationRule
(${INGRESS_COMPONENT_NAME}) Gateway
(${ASM_ISTIO_COMPONENT_NAME}) Gateway
(${LOGGER_COMPONENT_NAME}) Gateway
(${INGRESS_COMPONENT_NAME}) ServiceEntry
(${ASM_ISTIO_COMPONENT_NAME}) ServiceEntry
```

```
(${LOGGER_COMPONENT_NAME}) VirtualService
(${ASM_ISTIO_COMPONENT_NAME}) VirtualService
(${LOGGER_COMPONENT_NAME})  VirtualService
(${INGRESS_COMPONENT_NAME}) Route
(${INGRESS_COMPONENT_NAME}) Role (${COMPONENT_NAME}-istio-
role) RoleBinding (${COMPONENT_NAME}-istio-role-binding)
```

Если производилась установка с параметрами по умолчанию, то их значения будут следующими:

Таблица 9. Значения параметров по умолчанию для AppShard Router

Параметр	Значение
COMPONENT_NAME	asr-fort
EGRESS_COMPONENT_NAME	egress
INGRESS_COMPONENT_NAME	ingress
FLUENT_BIT_CONFIG_MAP_NAME	fluent-bit-config
LOGGER_COMPONENT_NAME	logger
ASM_ISTIO_COMPONENT_NAME	asm

1. В случае установки компонента AppShard Router с компонентом One-Time Password (OTP) / OTT (OTTS) среди установленных артефактов необходимо проверить наличие следующих:

```
EnvoyFilter (${LOGGER_COMPONENT_NAME}) EnvoyFilter
(${INGRESS_COMPONENT_NAME}) EnvoyFilter
(${EGRESS_COMPONENT_NAME}) ConfigMap
(${EGRESS_COMPONENT_NAME}-ott-config)  ConfigMap
(${EGRESS_COMPONENT_NAME}-ott-logback) ConfigMap
(${INGRESS_COMPONENT_NAME}-ott-config) ConfigMap
(${INGRESS_COMPONENT_NAME}-ott-logback)
```

Руководство по установке компонента Сервис межкластерной индексации (CCIX)

Список терминов и сокращений

Термин	Определение
CCI	Сервис межкластерной индексации
K8s	Kubernetes
OTT	Компонент One-Time Password (OTP) / OTT (OTTS)
БД	База данных
ПЖ	Компонент Прикладной журнал (APLJ)
ПО	Программное обеспечение
СУБД	Система управления базами данных
ТС	Технологический сервис

Назначение документа

Данный документ содержит информацию о создании инфраструктуры и порядке установки и настройки компонентов CCIX. Документ предназначен для администраторов, выполняющих установку, настройку и сопровождение CCIX.

Системные требования

Настройки безопасности окружения и перечень платформенных (дополнительных внешних) продуктов, используемых для установки, настройки и контроля в конечной информационной системе (далее — ИС), выбираются при разработке конечной ИС, исходя из характера обрабатываемой в ней информации и иных требований информационной безопасности (далее — ИБ), предъявляемых к ней.

Системное программное обеспечение

Ниже представлены категории системного программного обеспечения, которые обязательны для установки, настройки, контроля и функционирования компонента. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО «СберТех» (маркировка «Рекомендовано» в столбце Комментарий). Клиенту необходимо выбрать один из продуктов в каждой категории, исходя из условий использования конечной ИС.

Операционная система

Наименование	Версия	Комментарий
ОС Альт 8 СП	10.0 и выше	Рекомендовано
Red Hat Enterprise Linux	7 и выше	Опционально

Среда контейнеризации

Необходимое окружение для разворачивания компонента — Kubernetes версии 1.18 и выше или OpenShift версии 4.2 и выше (далее — среда контейнеризации).

Наименование	Версия	Комментарий
Kubernetes	1.18 и выше	Рекомендовано
Red Hat OpenShift	4.2 и выше	Опционально

Java-машина

Наименование	Версия	Комментарий
OpenJDK	1.8 и выше	Рекомендовано
OracleJDK	8 и выше	Опционально

Система управления базами данных (СУБД)

Наименование	Версия	Комментарий
PostgreSQL	11 и выше	Рекомендовано
Oracle Database	19c и выше	Опционально

Правообладателем АО «СберТех» также рекомендован Platform V Pangolin SE, см. раздел «Платформенные зависимости».

Сервис интеграции и оркестрации микросервисов в облаке

Наименование	Версия	Комментарий
Istio	1.12 и выше	Рекомендовано

Правообладателем АО «СберТех» также рекомендован Platform V Synapse Service Mesh, см. раздел «Платформенные зависимости».

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Monitor	OPM	3.27.11	LOGA Журналирование	опционально	Сервис для хранения лог-файлов
Platform V Monitor	OPM	6.0.0-02	MONA Объединенный мониторинг Unimon	опционально	Сервис для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения
Platform V Pangolin SE	PSQ	4.6.0. и выше	-	опционально	Система управления базами данных, основанная на PostgreSQL
Platform V DevOps Tools	DOT	1.2	-	опционально	Сервис для развертывания и обновления компонентов Платформы и приложений потребителей, для настройки и обслуживания инфраструктуры Платформы

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Synapse Service Mesh	SSM	1.12 и выше	POLM Управление политиками	опционально	Панель управления, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в среде контейнеризации Kubernetes

Примечание:

- **Обязательно** — компонент или продукт необходим для функционирования сервиса.
- **Опционально** — необязательный для функционирования сервиса компонент или продукт, рекомендуется его установка, но допускается использование аналога других производителей.

Установка

Ручная установка компонента

Подготовка окружения

Настройка namespace в среде контейнеризации

Настройка namespace в среде контейнеризации состоит из следующих шагов:

1. Подключение Istio.
2. Выпуск и установка сертификатов Istio для Ingress/Egress.
3. Настройка OTT.

Подключение Istio

Подключение Istio осуществляется в соответствии с документацией на него.

Выпуск и установка сертификатов Istio для Ingress/Egress

Используя шаблоны из дистрибутива /ccix/other/secrets/, выпустите и установите сертификаты Istio Ingress/Egress:

- `istio-egressgateway-ca-certs`;
- `istio-egressgateway-certs`;
- `istio-ingressgateway-ca-certs`;
- `istio-ingressgateway-certs`.

После чего будет создан набор необходимых артефактов, в том числе секреты с сертификатами `cert`, `ca-cert`.

Созданные секреты содержат наименования файлов: `tls.crt`, `tls.key`, `ca-chain.cert.pem` (вся цепочка сертификатов).

В секретах `istio-ingressgateway-ca-certs` и `istio-egressgateway-ca-certs` должен присутствовать один файл `ca-chain.cert.pem` со всей цепочкой сертификатов. Если в секрете указаны два отдельных сертификата, то вызовы через Ingress по HTTPS не будут работать.

Возможен вариант, когда в секретах в файле `ca-chain.cert.pem` содержится вся цепочка сертификатов (т.е. добавлено содержимое из `ca.pem`), но сам файл `ca.pem` не удален. В этом случае все будет работать корректно.

Настройка OTT

Для настройки OTT:

- получите сертификаты приложения, публичный ключ OTT Service и выполните их подключение;
- настройте ConfigMap для OTT-sidecar.

Получение сертификатов, публичного ключа OTT Service и их подключение

Для получения сертификатов, публичного ключа OTT Service и их подключения:

1. При помощи keytool сгенерируйте key-store с ключами (`ccix_cert.p12`) с помощью алгоритма SHA256withECDSA и длиной ключа 256 бит.

Примеры команд

```
keytool -genkey -keyalg EC -sigalg SHA256withECDSA -keystore
ccix.p12 -storetype PKCS12 -keysize 256 -dname "CN=ccix" -
alias ccix keytool -certreq -keyalg EC -sigalg
```

```
SHA256withECDSA -keystore ccix.p12 -storetype PKCS12 -alias
ccix > ccix_cert_req.pem
```

2. Отправьте файл `ccix_cert_req.pem` администратору OTT в заявке на генерацию сертификата для модуля. В заявке явно указать, что сертификат должен быть сгенерирован из приложенного запроса по инструкции. Администратор OTT подписывает/генерирует ключи, высылает их обратно.
3. Импортируйте полученные от администратора OTT ключи в `key-store ccix_cert.p12`.

Примеры команд

```
keytool -import -keystore ccix.p12 -storetype PKCS12 -file
SberbankPlatformCA_EC.pem -alias SberbankPlatformCA_EC
keytool -import -keystore ccix.p12 -storetype PKCS12 -file
ccix.pem -alias ccix
```

4. Создать два секрета:

- `ott-certs` — с сертификатами (ccix.p12: >- , ott_service_truststore.p12: >-)

Важно! Чтобы все корректно добавилось, необходимо сконвертировать содержимое файла с ключами в Base64. Далее — полученную строку вставить в yml-файл. Либо выполнить команду:

```
oc create secret generic ott-certs --from-
file=ott_service_truststore.p12=./sigma_ott_trust.p12 --
from-file=ccix.p12=./ccix.p12
```

- `ott-passwords` — с паролями к ключам:
  - `OTT_CERTSTORE_PRIVATE_KEY_PWD` — пароль для чтения ключа;
  - `OTT_CERTSTORE_PWD` — пароль для чтения сертификата;
  - `OTT_TRUST_STORE_PWD` — пароль для чтения ott_service_truststore.p12.

Запуск liquibase скриптов. Настройка подключения Базы данных

Принципы работы с объектами БД:

- Набор выделяемых полномочий должен быть минимально необходимым и достаточным для корректной работы компонента.
- Должно быть исключено использование перекрестных и взаимозависимые полномочий между схемами хранения.
- В каждой инсталляции на уровне отдельного экземпляра БД должен вводиться уникальный идентификатор-суффикс, позволяющий определять принадлежность к инсталляции. Основное назначение уникального идентификатора – это

минимизировать ошибки копирования при ручных операциях во множество различных баз данных.

- Исходя из разного типа нагрузки на ввод-вывод, хранение сегментов таблиц, индексов и больших объектов типа LOB необходимо разделять на отдельные выделенные табличные пространства. Не допускается размещение в одном табличном пространстве содержимого разных типов (например, индексов вперемешку с данными).

Запуск liquibase скриптов компонента.

Файл дистрибутива для раскатки скриптов: `package/db/ccix-postgres-db.zip`.

Параметры подключения:

- `jdbc URL` берется из глобального параметра `EFS_PSTGR_DB_URL`;
- `jdbcDriver`: `org.postgresql.Driver`;
- Префикс для хранения объктов БД: `ccix_`;
- Default Tablespace: `ccix_ts_data`;
- Required Tablespace:
  - `ts_data.placeholder` = `ccix_ts_data`;
  - `ts_idx.placeholder` = `ccix_ts_idx`;
- пользователей и пароли указываются в secret `secret-secret`:
  - пользователь: `jdbc.ccix_postgres.user`;
  - пароль: `jdbc.ccix_postgres.password`.

Настройка конфигурационных файлов

Для ручной установки компонента в среде контейнеризации необходимо во всех файлах конфигурации заменить все плейсхолдеры на реальные значения.

Перечень конфигурационных файлов для развертывания в Kubernetes:

```
└── k8s ├── base │   ├── cci-proxy │   │   ├── configmaps │
│   │   └── cm-cci-proxy-webserver.yml │   │   ├── dc.yml │
│   ├── istio │   │   │   ├── config │   │   │   │   ├──
egress │   │   │   │   └── ingress │   │   │   └──
deployments │   │   │       ├── egress │   │   │       │
├── dr-proxy.yml │   │   │       │   ├── dr-virtual-
proxy.yml │   │   │       │   ├── gw-proxy.yml │   │   │
│   ├── se-proxy.yml │   │   │       │   └── vs-proxy.yml │
│   │       └── ingress │   │   ├── pdb.yml │   │   ├──
route.yml │   │   └── svc.yml │   ├── cci-server │   │   ├──
configmaps │   │   │   ├── cm-cci-server-webserver.yml │   │
│   ├── cm-db.yml │   │   │   ├── cm-java-props.yml │   │
```

```
│   ├── cm-monitoring.yml │   │   │   └── cm-ott.yml │   │
├── dc.yml │   │   ├── istio │   │   │   ├── config │   │
│   │   ├── egress │   │   │   │   └── ingress │   │   │
└── deployments │   │   │       ├── egress │   │   │
└── ingress │   │   ├── pdb.yml │   │   ├── secrets │   │
│   ├── secret-db-certs.yaml │   │   │   └── secret-db-
passwords.yaml │   │   └── svc.yml │   ├── configmaps │   │
├── cm-fluent-bit.yml │   │   └── cm-logback.yml │   ├──
istio │   │   ├── config │   │   │   ├── egress │   │   │
│   ├── cm-egress-ott-logback.yml │   │   │   │   ├── cm-
egress-ott-settings.yml │   │   │   │   └── cm-egress.yml │
│   │   └── ingress │   │   │       ├── cm-ingress-ott-
logback.yml │   │   │       ├── cm-ingress-ott-settings.yml
│   │   │       ├── cm-ingress-ott.yml │   │   │       └──
cm-ingress.yml │   │   ├── deployments │   │   │   ├──
egress │   │   │   │   ├── dc-egress-ott.yml │   │   │   │
├── dc-egress.yml │   │   │   │   ├── dr-logger-mtls.yml │
│   │   │   ├── ef-logger-mtls.yml │   │   │   │   ├── gw-
db-main-standby.yml │   │   │   │   ├── gw-db-main.yml │   │
│   ├── gw-db-si-standby.yml │   │   │   │   ├── gw-db-
si.yml │   │   │   │   ├── gw-egress.yml │   │   │   │   ├──
gw-logger-mtls.yml │   │   │   │   ├── gw-logger.yml │   │
│   ├── se-db-main-standby.yml │   │   │   │   ├── se-
db-main.yml │   │   │   │   ├── se-db-si-standby.yml │   │
│   ├── se-db-si.yml │   │   │   │   ├── se-kube-api-
service-entry.yml │   │   │   │   ├── se-logger-mtls.yml │
│   │   ├── se-logger.yml │   │   │   │   ├── vs-db-
main-standby.yml │   │   │   │   ├── vs-db-main.yml │   │
│   ├── vs-db-si-standby.yml │   │   │   │   ├── vs-db-
si.yml │   │   │   │   ├── vs-logger-mtls.yml │   │   │   │
└── vs-logger.yml │   │   │   └── ingress │   │   │
├── dc-ingress-ott.yml │   │   │       ├── dc-ingress.yml │
│   │       ├── ef-ingress-ott.yml │   │   │       ├── gw-
ingress-ott.yml │   │   │       ├── gw-ingress.yml │   │   │
├── route-ingress-ott.yml │   │   │       ├── route-
ingress.yml │   │   │       ├── svc-ingress-ott.yml │   │
│       └── svc-ingress.yml │   │   └── secrets │   │
├── egress-ca-jks.yaml │   │       ├── egress-jks.yaml │   │
├── ingress-ca-jks.yaml │   │       ├── ingress-jks.yaml │
│       └── ott-certs.yaml │   └── secrets └── overrides └──
openshift ├── cci-proxy │   ├── dc.yml │   └── route.yml ├──
cci-server │   └── dc.yml └── istio └── deployments └──
ingress ├── route-ingress-ott.yml └── route-ingress.yml
```

После замены файлов конфигурации необходимо развернуть файлы конфигурации в среду контейнеризации:

1. В панели администратора в правом верхнем углу нажать на кнопку **Import YAML**;

2. в открывшейся окно вставить скопированный конфигурационный файл и нажать **Create**;
3. после создания отобразиться страница конфигурационного файла;
4. повторить действия 1-3 для всех файлов конфигурации.

Автоматизированная установка компонента с использованием DevOps Tools

Для установки сервиса с использованием компонента Deploy Tools (CDJE) программного продукта Platform V DevOps Tools (DOT) разверните сервис в инфраструктуре и запустите CD Job по инструкции, приведенной в документации Platform V DevOps Tools (DOT).

Требования к окружению:

- наличие у компонента пространства в сервисе централизованного хранения репозиториев исходного кода и прав на создание репозиториев в нем;
- наличие у компонента области в Jenkins, с настроенными учетными данными (логин/пароль) для авторизации под технологической учетной записью (ТУЗ), позволяющая загружать артефакты в целевой репозиторий хранилища.

Параметры настройки

Параметры в конфигурационных файлах дистрибутива могут содержать плейсхолдеры на общие параметры установки, конкретные значения которых задаются администратором для каждого конкретного стенда. Конкретные значения параметров установки хранятся отдельно в файлах стендов, наличие этих данных обеспечивается администраторами стендов, куда производится установка.

```
├── conf │   └── config │       ├── parameters
│       │   ├── ccix.all.conf │       │   ├── ccix.cci-
proxy.conf │       │   ├── ccix.cci-proxy.istio.conf │
│   ├── ccix.cci-server.conf │       │   └──
ccix.istio.all.conf
```

`ccix.all.conf` – общие параметры установки.

```
# ---------- COMMON ---------- #
docker.registry.host=${global.registry.url|dzo.sw.sbc.space}
ccix.namespace=${global.multiCluster.openshiftProjectName}
ccix.domain=${global.multiCluster.openshiftAppsDomain}
ccix.cluster=${global.multiClusters.openshiftAppsDomain}
ccix.java.module.id=cci-server-cloud ccix.java.xms=1g
ccix.java.xmx=3g ccix.java.metaspace.size=1g
ccix.java.max.metaspace.size=3g  # ---------- LOGGER -------
--- # ccix.logback.config.map.name=logback-config
ccix.fluent.bit.config.map.name=fluent-bit-config
```

```
ccix.fluent.bit.sidecar.env.zone.id=default  # ----------
OTT ---------- #
ccix.ott.sidecar.image=${global.ott.image.sidecar.version}
ccix.ott.client.hosts=${global.ott.service.hosts}
ccix.ott.service.url=${global.ott.service.url}
ccix.ott.grpc.port=${global.ott.grpc.port}
ccix.ott.certstore.path=${global.ott.keyStore.keystore.keySt
oreFromFile}
ccix.ott.trust.store.path=${global.ott.keyStore.truststore.k
eyStoreFromFile}
ccix.ott.authorization.enabled=${global.cci.ott.authorizatio
n.enabled|true} ccix.ott.client.tls='TLSv1.2'
ccix.ott.auth.realm=${global.ose.ott.authz.realm}
ccix.ott.token.type.attribute.id='urn:sbrf:names:pprb:1.0:to
ken:type'
ccix.ott.module.attribute.id='urn:sbrf:names:pprb:1.0:module
:id'  ccix.ott.sidecar.limits.cpu=400m
ccix.ott.sidecar.limits.memory=500Mi
ccix.ott.sidecar.requests.cpu=300m
ccix.ott.sidecar.requests.memory=500Mi
```

`ccix.cci-proxy.conf` – параметры установки клиентской части.

```
ccix.cci-proxy.module.id=cci-proxy ccix.cci-
proxy.node.id=cci-proxy ccix.cci-proxy.image=cci-proxy
ccix.cci-proxy.component.name=cci-proxy ccix.cci-
proxy.container.name=cci-proxy ccix.cci-
proxy.image.version=DISTRIB_VERSION ccix.cci-
proxy.webserver.min.spare.threads=10 ccix.cci-
proxy.webserver.max.threads=200 ccix.cci-
proxy.services.configmap.name=cci-proxy-services ccix.cci-
proxy.indexes.configmap.name=cci-proxy-indexes  ###
RESOURCES ### выставлены минимальные значения, необходима
корректировка по результатам НТ ccix.cci-
proxy.deployment.container.cci-proxy.limits.cpu=300m
ccix.cci-proxy.deployment.container.cci-
proxy.limits.memory=1Gi ccix.cci-
proxy.deployment.container.cci-proxy.requests.cpu=300m
ccix.cci-proxy.deployment.container.cci-
proxy.requests.memory=1Gi ccix.cci-
proxy.poddisruptionbudget.spec.min.available=${global.poddis
ruptionbudget.minAvailable|50%}  ### FLUENT_BIT ###
ccix.cci-
proxy.deployment.container.fluent.bit.image.path=${global.pl
atform.fluent-bit.sidecar.image-version} ccix.cci-
proxy.deployment.container.fluent.bit.container.name=fluent-
bit-sidecar ccix.cci-
proxy.deployment.container.fluent.bit.limits.cpu=${global.pl
atform.fluent-bit.limits.cpu} ccix.cci-
```

```
proxy.deployment.container.fluent.bit.limits.memory=${global
.platform.fluent-bit.limits.memory} ccix.cci-
proxy.deployment.container.fluent.bit.requests.cpu=${global.
platform.fluent-bit.requests.cpu} ccix.cci-
proxy.deployment.container.fluent.bit.requests.memory=${glob
al.platform.fluent-bit.requests.memory}
```

`ccix.cci-server.conf` – параметры установки серверной части.

```
# ---------- CCI SERVER ---------- # ccix.cci-
server.instance.name.prefix=cci-server ccix.cci-
server.instance.qualifier=default ccix.cci-
server.image.version=DISTRIB_VERSION ccix.cci-
server.image.container.name=cci-server ccix.cci-
server.image=cci-server ccix.cci-
server.indexes.config.map=cci-indexes ccix.cci-
server.index.config.path=/config/index/index-config.yml
ccix.cci-server.db.config.path=/config/db/db-config.yml
ccix.cci-server.replicas.count=1 ccix.cci-
server.healthcheck.index.name='__INDEX_FOR_CCI_TEST__'
ccix.cci-
server.monitoring.config.path=/config/monitoring/monitoring-
config.yml  # ---------- WEB-SERVER ---------- # ccix.cci-
server.webserver.config.path=/config/webserver/webserver-
config.yml ccix.cci-server.webserver.max.threads=200
ccix.cci-server.webserver.min.spare.threads=10  # ----------
MAIN DATABASE ---------- # ccix.cci-
server.db.main.jdbc.url=${jdbc.cci_postgres.url.main}
ccix.cci-server.db.main.ip=${jdbc.cci_postgres.ip.main}
ccix.cci-
server.db.main.ip.standby=${jdbc.cci_postgres.ip.main.standb
y} ccix.cci-
server.db.main.port=${jdbc.cci_postgres.port.main} ccix.cci-
server.db.main.port.standby=${jdbc.cci_postgres.port.main.st
andby} ccix.cci-server.db.main.virtual.port=8701 ccix.cci-
server.db.main.virtual.port.standby=8701 ccix.cci-
server.db.main.user=${DB_USER_MAIN} ccix.cci-
server.db.main.password=${DB_PASSWORD_MAIN} ccix.cci-
server.db.main.ssl.enabled=false ccix.cci-
server.db.main.ssl.keystore.password=${DB_MAIN_SSL_KEYSTORE_
PWD} ccix.cci-
server.db.main.ssl.truststore.password=${DB_MAIN_SSL_TRUSTST
ORE_PWD} ccix.cci-
server.db.main.schema.owner=${jdbc.cci_postgres.schema.main}
ccix.cci-server.db.main.asquire.increment=5 ccix.cci-
server.db.main.asquire.retry.attempts=5 ccix.cci-
server.db.main.asquire.retry.delay=5001 ccix.cci-
server.db.main.checkout.timeout=30000 ccix.cci-
server.db.main.idle.connection.test.period=6 ccix.cci-
```

```
server.db.main.idle.update.test.period=120001 ccix.cci-
server.db.main.idle.time.max=181 ccix.cci-
server.db.main.connection.age.max=0 ccix.cci-
server.db.main.pool.size.initial=18 ccix.cci-
server.db.main.pool.size.max=18 ccix.cci-
server.db.main.pool.size.min=18  # ---------- SI DATABASE --
-------- # ccix.cci-
server.db.si.jdbc.url=${jdbc.cci_postgres.url.standin}
ccix.cci-server.db.si.ip=${jdbc.cci_postgres.ip.standin}
ccix.cci-
server.db.si.ip.standby=${jdbc.cci_postgres.ip.standin.stand
by} ccix.cci-
server.db.si.port=${jdbc.cci_postgres.port.standin}
ccix.cci-
server.db.si.port.standby=${jdbc.cci_postgres.port.standin.s
tandby} ccix.cci-server.db.si.virtual.port=8701 ccix.cci-
server.db.si.virtual.port.standby=8701 ccix.cci-
server.db.si.user=${DB_USER_SI} ccix.cci-
server.db.si.password=${DB_PASSWORD_SI} ccix.cci-
server.db.si.ssl.enabled=false ccix.cci-
server.db.si.ssl.keystore.password=${DB_SI_SSL_KEYSTORE_PWD}
ccix.cci-
server.db.si.ssl.truststore.password=${DB_SI_SSL_TRUSTSTORE_
PWD} ccix.cci-
server.db.si.schema.owner=${jdbc.cci_postgres.schema.standin
} ccix.cci-server.db.si.asquire.increment=5 ccix.cci-
server.db.si.asquire.retry.attempts=5 ccix.cci-
server.db.si.asquire.retry.delay=5001 ccix.cci-
server.db.si.checkout.timeout=30000 ccix.cci-
server.db.si.idle.connection.test.period=6 ccix.cci-
server.db.si.idle.update.test.period=120001 ccix.cci-
server.db.si.idle.time.max=181 ccix.cci-
server.db.si.connection.age.max=0 ccix.cci-
server.db.si.pool.size.initial=18 ccix.cci-
server.db.si.pool.size.max=18 ccix.cci-
server.db.si.pool.size.min=18  # ----------
LIVENESS_READINESSTHEUS ---------- # ccix.cci-
server.deployment.container.cci-
server.liveness.probe.initial.delay.seconds=30 ccix.cci-
server.deployment.container.cci-
server.liveness.probe.period.seconds=3 ccix.cci-
server.deployment.container.cci-
server.liveness.probe.failure.threshold=3 ccix.cci-
server.deployment.container.cci-
server.liveness.probe.success.threshold=1 ccix.cci-
server.deployment.container.cci-
server.liveness.probe.timeout.seconds=3 ccix.cci-
server.deployment.container.cci-
server.readiness.probe.initial.delay.seconds=20 ccix.cci-
server.deployment.container.cci-
```

```
server.readiness.probe.period.seconds=5 ccix.cci-
server.deployment.container.cci-
server.readiness.probe.failure.threshold=3 ccix.cci-
server.deployment.container.cci-
server.readiness.probe.success.threshold=1 ccix.cci-
server.deployment.container.cci-
server.readiness.probe.timeout.seconds=5  # ----------
RESOURCES ---------- # ccix.cci-
server.deployment.container.cci-server.limits.cpu=2
ccix.cci-server.deployment.container.cci-
server.limits.memory=2Gi ccix.cci-
server.deployment.container.cci-server.requests.cpu=300m
ccix.cci-server.deployment.container.cci-
server.requests.memory=1Gi ccix.cci-
server.poddisruptionbudget.spec.min.available=${global.poddi
sruptionbudget.minAvailable|50%}  # ---------- FLUENT_BIT --
-------- # ccix.cci-
server.deployment.container.fluent.bit.image.path=${global.p
latform.fluent-bit.sidecar.image-version} ccix.cci-
server.deployment.container.fluent.bit.container.name=fluent
-bit-sidecar ccix.cci-
server.deployment.container.fluent.bit.limits.cpu=${global.p
latform.fluent-bit.limits.cpu} ccix.cci-
server.deployment.container.fluent.bit.limits.memory=${globa
l.platform.fluent-bit.limits.memory} ccix.cci-
server.deployment.container.fluent.bit.requests.cpu=${global
.platform.fluent-bit.requests.cpu} ccix.cci-
server.deployment.container.fluent.bit.requests.memory=${glo
bal.platform.fluent-bit.requests.memory}
```

ccix.cci-proxy.istio.conf– конфигурационный файл для настройки Istio клиентской части.

```
ccix.cci-proxy.istio.ingress.host=${global.cci.ingress.host}
ccix.cci-proxy.istio.ingress.virtual.port=8704 ccix.cci-
proxy.istio.ingress.url.prefix=${global.cci.ingress.url.pref
ix|https://}
```

ccix.cci-server.istio.conf– конфигурационный файл для настройки Istio серверной части.

```
# ---------- COMMON ---------- #
ccix.istio.control.plane.namespace=${global.ufs.synapse.cont
rolPlane.project}
ccix.istio.ingress.host=${global.cci.istio.ingress.host}
ccix.istio.ingress.component.name.envoy.image=${global.ufs.s
ynapse.envoy.image}  # ---------- INGRESS ---------- #
```

```
ccix.istio.ingress.component.name=ingressgw-
${global.ufs.synapse.controlPlane.project}
ccix.istio.ingress.mtls.ca.certificate.path='/etc/istio/ingr
essgateway-ca-certs'
ccix.istio.ingress.mtls.server.certificate.path='/etc/istio/
ingressgateway-certs' ccix.istio.ingress.port.number=443
ccix.istio.ingress.envoy.limits.cpu=450m
ccix.istio.ingress.envoy.limits.memory=800Mi
ccix.istio.ingress.envoy.requests.cpu=350m
ccix.istio.ingress.envoy.requests.memory=600Mi  # ----------
EGRESS ---------- #
ccix.istio.egress.component.name=egressgateway-
${global.multiCluster.openshiftProjectName}
ccix.istio.egress.mtls.ca.certificate.path='/etc/istio/egres
sgateway-ca-certs'
ccix.istio.egress.mtls.server.certificate.path='/etc/istio/e
gressgateway-certs'  ccix.istio.egress.envoy.limits.cpu=200m
ccix.istio.egress.envoy.limits.memory=512Mi
ccix.istio.egress.envoy.requests.cpu=100m
ccix.istio.egress.envoy.requests.memory=256Mi  # ----------
LOGGER ---------- #
ccix.logger.endpoint.host=${global.cci.logger.endpoint.host}
ccix.logger.endpoint.port=${global.cci.logger.endpoint.port|
443} ccix.logger.http.port.number=80
ccix.logger.endpoint.virtual.port=443
ccix.logger.endpoint.uri=${global.cci.logger.endpoint.uri|v1
/events} ccix.logger.component.name=logger  # ----------
KUBE_API ---------- #
ccix.istio.egress.kube.api.ip=${global.cci.istio.egress.kube
.api.ip}
ccix.istio.egress.kube.api.port=${global.cci.istio.egress.ku
be.api.port}
```

Обновление

Обновление компонента осуществляется путём установки новой версии компонента. Предварительное удаление текущей версии компонента не требуется.

Удаление

Для удаления компонента CCIX необходимо выполнить следующие шаги:

1. Остановить deployment CCIX.
2. Удалить deployment CCIX.
3. Удалить связанные ConfigMap, Gateway, VirtualService, Service имена которых начинаются с $-$
4. Удалить связанные секреты, в случае если они не используются другими модулями в namespace:

```
certs:     istio-egressgateway-ca-certs     istio-
egressgateway-certs     istio-ingressgateway-ca-certs
istio-ingressgateway-certs     ott-certs     secret:
cci-server-one-db-secret-main.yml     cci-server-one-db-
secret-si.yml     ott-pwd.yml
```

5. Удалить ott-envoy-filter.
6. Остановить Ingress и Egress, в случае если они не используются другими модулями в namespace.
7. Удалить deployment Ingress, Egress, связанных ConfigMap, Gateway, VirtualService имена которых начинаются с $, $*.
8. Удалить ConfigMap egress-ott-logback, egress-ott-settings, istio-basic-$, *istio-basic-$*, ingress-ott-logback, ingress-ott-settings.
9. Удалить ConfigMap $, *$*, ott.
10. Удалить табличное пространства CCIX в основной и SI БД.

```
* Значения параметров ccix.cci-server.instance.name.prefix,
ccix.cci-server.instance.qualifier,
ccix.istio.ingress.component.name,
ccix.istio.egress.component.name,
ccix.fluent.bit.config.map.name,
ccix.logback.config.map.name задаются в файле конфигурации
переменных окружения при установке модуля.
```

Проверка работоспособности

Сценарий проверки работоспособности:

№ п/п	Действие	Результат
1	Проверить, что тестовый индекс настроен на зону default	Для тестового индекса _INDEX_FOR_CCI_TEST_ в конфигурации параметр zone установлен в значение default
2	Выполнить тестовый запрос get	По логам серверной части убедиться, что запрос пришел на серверную часть в зоне default

№ п/п	Действие	Результат
3	Настроить тестовый индекс на зону CCIZONE_DEMO	Для тестового индекса `_INDEX_FOR_CCI_TEST_` в конфигурации параметр zone установлен в значение CCIZONE_DEMO
4	Выполнить тестовый запрос put	По логам серверной части убедиться, что запрос пришел на серверную часть в зоне `CCIZONE_DEMO`. Был отправлен запрос в ПЖ и после прохождения цикла репликации журнал был обработан серверной частью в зоне `CCIZONE_DEMO`
5	Выполнить тестовый запрос delete	По логам серверной части убедиться, что запрос пришел на серверную часть в зоне `CCIZONE_DEMO`. Был отправлен запрос в ПЖ и после прохождения цикла репликации журнал был обработан серверной частью в зоне `CCIZONE_DEMO`

№ п/п	Действие	Результат
6	Проверка корректности настройки OTT	В Pod Egress и Ingress в логах OTT Sidecar проверяется наличие от потребителей OTT-заголовков: ott-invoker со значением сертификата Приложения (на Ingress должен быть, как заголовок CCIX, так и заголовки потребителей, обращающихся к компоненту. На Egress должен быть заголовок CCIX; ott-hash; ott-token-valid со значением true.
7	Проверка корректности настройки OTT (негатив)	Необходимо выполнить запрос без OTT-токена. В случае, если OTT корректно настроен и включен, ответом на запрос будет ошибка вида "403 OTT Exception"
8	Проверка корректности настройки TLS	Необходимо выполнить запрос с корретными сертификатами.

Откат

Откат осуществляется путем установки предыдущей версии компонента.:

№ п/п	Действие	Описание	Влияние на основную зону
1	Остановить клиентскую часть	Выполнить остановку серверов приложений на всех серверах фабрик	да (15 мин)

№ п/п	Действие	Описание	Влияние на основную зону
2	Остановить серверную часть	Выполнить остановку серверов приложений, на которых установлена серверная часть CCIX как глобального, так и локального инстанса	да (15 мин)
3	Создать конфигурацию для загрузки данных	Используя пакет db/migrate из дистрибутива CCI создать конфигурацию для загрузки данных из локального в глобальный инстанс CCI	да (1 мин)

№ п/п	Действие	Описание	Влияние на основную зону
4	Через пакет migration выполнить загрузку данных из локального инстанса CCIX в глобальный инстанс CCIX	В случае, если при включении локального инстанса CCIX в него успели добавиться какие-либо записи, то при откате необходимо предусмотреть перенос этих новых данных обратно в глобальный инстанс CCIX. Для переноса данных из локального инстанса CCI в глобальный инстанс CCI необходимо воспользоваться пакетом миграции из дистрибутива ТС Межкластерной индексации	да (зависит от объема данных)
5	Перевести работу с индексом в глобальный инстанс CCIX	В конфигурации для индекса, обслуживание которого необходимо перенести в глобальный инстанс установить параметр zone в значение default	да (5 мин)

№ п/п	Действие	Описание	Влияние на основную зону
6	Запустить серверную часть CCIX	Запустить все серверные части CCIX глобального инстанса	да (10 мин)
7	Запустить клиентскую часть CCIX	Выполнить запуск серверов приложений на всех серверах фабрик	да (30 мин)

Часто встречающиеся проблемы и пути их устранения

Часто встречающиеся проблемы не зафиксированы.

Чек-лист валидации установки

1. В разделе Workloads/Deployments развернут ingress.
2. В разделе Workloads/Deployments развернут egress.
3. В разделе Networking/Routes создан ресурс типа Route, указывающий на сервис Ingress.
4. В разделе Workloads/Config Maps присутствуют необходимые файлы настроек.
5. В разделе Workloads/Deployment присутствуют deployments для cci-server, unimon-agent, unimon-sender. Каждый из них имеет необходимое количество реплик, заданных в конфигурации развёртывания и все pod'ы имеют статус «Запущен».

Сокращение	Расшифровка
БД	База данных
ММТ	Межмодульный транспорт
УЦ	Удостоверяющий центр
CA	англ. Certification authority – центр сертификации или удостоверяющий центр
CPU	англ. Central Processing Unit – центральное обрабатывающее устройство, процессор
DNS	англ. Domain Name System – система доменных имён
Egress / Ingress GW	Egress Gateway – выходной шлюз, Ingress Gateway – входной шлюз
EJBCA	англ. Enterprise Java Beans Certificate Authority – центр сертификации, построенный с использованием технологии Java
JBoss CLI	англ. JBoss Command line interface – интерфейс командной строки сервера приложений JBoss
JCE	англ. Java Cryptography Extension – криптографическое расширение Java
JDBC	англ. Java DataBase Connectivity – соединение с базами данных на Java
JDK	англ. Java Development Kit – комплект разработчика приложений на языке Java
JKS	англ. Java Keystore – хранилище криптографических публичных и приватных ключей и сертификатов на основе Java
OTT	One-Time Token
PEM-формат	англ. Privacy Enhanced Mail – формат файла-контейнера, содержащего криптографический сертификат
RAM	англ. Random Access Memory – память с произвольным доступом, оперативная память

Сокращение	Расшифровка
SQL	англ. Structured Query Language – язык структурированных запросов, применяемый для создания, модификации и управления данными в реляционной базе данных
SSL	англ. Secure Sockets Layer — уровень защищённых сокетов – криптографический протокол, который подразумевает более безопасную связь
TLS	англ. Transport layer security — протокол защиты транспортного уровня
TLSv1	TLS версии 1.0
TLSv1.1	TLS версии 1.1
TLSv1.2	TLS версии 1.2
Transact-SQL	процедурное расширение языка SQL
UDS	англ. Unix domain socket – сокет межпроцессного взаимодействия
URL	англ. Uniform Resource Locator – унифицированный указатель ресурса
Web-UI	англ. Web user interface – пользовательский веб-интерфейс

Системные требования

Настройки безопасности окружения и перечень платформенных (дополнительных внешних) продуктов, используемых для установки, настройки и контроля в конечной информационной системе (далее — ИС), выбираются при разработке конечной ИС, исходя из характера обрабатываемой в ней информации и иных требований информационной безопасности (далее — ИБ), предъявляемых к ней.

Системное программное обеспечение

Ниже представлены категории системного программного обеспечения, которые обязательны для установки, настройки, контроля и функционирования компонента. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО «СберТех» (маркировка «Рекомендовано» в столбце Комментарий). Клиенту необходимо выбрать один из продуктов в каждой категории, исходя из условий использования конечной ИС.

Среда контейнеризации

Необходимое окружение для разворачивания компонента — Kubernetes версии 1.18 и выше или OpenShift версии 4.5.13 и выше (далее — среда контейнеризации).

Наименование	Версия	Комментарий
Kubernetes	1.18 и выше	Рекомендовано
Red Hat OpenShift	4.5.13 и выше	Опционально

Средство контейнеризации

Наименование	Версия	Комментарий
Docker CE		Рекомендовано

Удостоверяющий центр EJBCA (опциональный внешний компонент)

Наименование	Версия	Применение
EJBCA Community	6.10	Опционально

Для установки, настройки, контроля и функционирования УЦ обязательны следующие категории системного программного обеспечения:

Операционная система (для EJBCA)

Наименование	Версия	Комментарий
ОС Альт 8 СП	10.0 и выше	Рекомендовано
Red Hat Enterprise Linux	7 и выше	Опционально

Java-машина (для EJBCA)

Наименование	Версия	Комментарий
OpenJDK	1.8 и выше	Рекомендовано
OracleJDK	8 и выше	Опционально

Система управления базами данных (СУБД) (для EJBCA)

Наименование	Версия	Комментарий
PostgreSQL	9.2 и выше	Рекомендовано
Oracle Database	12c и выше	Опционально
Oracle GoldenGate	12c и выше	Опционально

Правообладателем АО «СберТех» также рекомендован Platform V Pangolin SE (PSQ) версии 4.6.0. и выше в качестве системы управления базами данных, основанной на PostgreSQL.

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание	Допустимая альтернатива
Platform V Audit SE	AUD	1.1	AUDT Аудит	опционально	Сервис для аудирования событий	Отсутствует, без компонента AUDT сбор событий аудита невозможен
Platform V Monitor	OPM	3.27.11	LOGA Журналирование	опционально	Сервис для хранения лог-файлов	Любой сервис сбора логов, совместимый с fluent-bit, например: Elasticsearch, InfluxDB.
Platform V Monitor	OPM	6.0.0-02	MONA Объединенный мониторинг Unimon	опционально	Сервис для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения	Prometheus 2.21.0 и выше

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание	Допустимая альтернатива
Platform V DevOps Tools	DOT	1.2	CDJE Deploy tools	опционально	Сервис для развертывания и обновления компонентов Платформы и приложений потребителей, для настройки и обслуживания инфраструктуры Платформы	Ручное развертывание
Platform V Synapse Service Mesh	SSM	1.12 и выше	IGEG Граничный прокси	опционально	Сервис для обеспечения управляемого вызова интеграционных сервисов прикладной части	Istio proxy 1.4 и выше

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание	Допустимая альтернатива
Platform V Synapse Service Mesh	SSM	1.12 и выше	POLM Управление политиками	опционально	Сервис автоматизированной рассылки конфигураций политик управления трафиком, политик безопасности и прочих управляющих команд для сервисов «Граничный прокси» (IGEG) и «Сервисный прокси» (SVPX)	Istio control plane 1.4 и выше
Platform V Synapse Service Mesh	SSM	1.12 и выше	SVPX Сервисный прокси	опционально	Сервис для предоставления базовых интеграционных операций прикладной части интеграционного сервиса	Istio proxy 1.4 и выше

Примечание:

- **Обязательно** — компонент или продукт необходим для функционирования сервиса.
- **Опционально** — необязательный для функционирования сервиса компонент или продукт, рекомендуется его установка, но допускается использование аналога других производителей.

Аппаратные требования

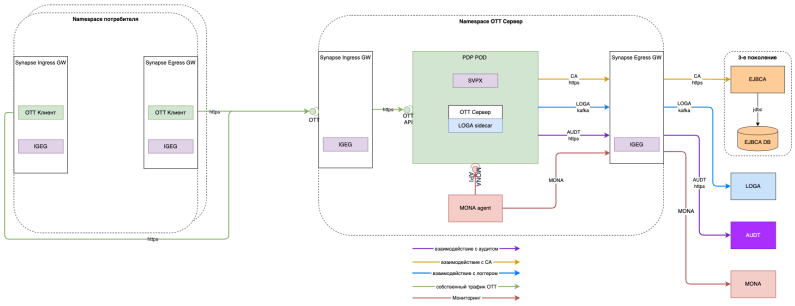
Для установки компонента/продукта требуется следующая конфигурация аппаратного обеспечения:

Назначение	Ресурсы
Namespace в кластере Kubernetes	CPU: 8 ядер; RAM: 32 ГБ
Квоты OTT Sidecar	CPU-limit: 400m; Memory-limit: 500Mi; EphemeralStorage-limit: 1Gi; CPU-request: 300m; Memory-request: 500Mi

Установка

В разделе по шагам описан процесс установки программного компонента и всех входящих в его состав компонентов, включая рекомендованные настройки.

Компонентная диаграмма сервиса OTT



Сервер OTT

В разделе описаны процедуры развертывания компонентов сервиса в облачной инфраструктуре.

Подготовка окружения

Создание хранилища с прикладным сертификатом OTT сервера

На сервере WildFly с EJBCA (EJBCA master) необходимо выполнить следующие действия:

1. Перейти в директорию установки EJBCA Web Service CLI `<ejbca_home>/dist/ejbca-ws-cli/` (далее директория: `<ejbca_ws_cli_home>`)
2. Настроить приложение EJBCA Web Service CLI:
   - Открыть на изменение конфигурационный файл `<ejbca_ws_cli_home>/ejbcawsracli.properties`
   - Задать параметры:
     - `ejbcawsracli.url`: URL для подключения к веб-сервисам EJBCA (https:// :8444/ejbca/ejbcaws/ejbcaws)
     - `ejbcawsracli.keystore.path`: Путь к keystore, содержащему secret key администратора EJBCA
     - `ejbcawsracli.keystore.password`: Пароль от keystore, указанного в ejbcawsracli.keystore.path

3. Выполнить регистрацию приложения OTT сервер и получить его приватный ключ:
   - Перейти в директорию EJBCA Web Service CLI <ejbca_ws_cli_home>
   - Выполнить команды регистрации модуля:

```
# Create/update module ./ejbcawsracli.sh edituser
ott-service <p12_password> false "cn=ott-service" NULL
NULL SberbankPlatformCA_EC 1 P12 NEW PPRBModule SBERBANK
# Upload module secret key ./ejbcawsracli.sh
pkcs12req ott-service <p12_password> secp256r1 ECDSA NONE
```

   <p12_password> – пароль для создаваемого хранилища .p12 После выполнения указанных команд в директории EJBCA Web Service CLI <ejbca_ws_cli_home> появится хранилище ott-service.p12

4. Переложить сгенерированный secret key из PKCS12 в JKS хранилище, выполнив команду:

```
keytool -importkeystore -v -srckeystore ott-service.p12 -
srcstoretype PKCS12 -srcstorepass <p12_password> -
destkeystore ott-service.jks -deststoretype JKS -
deststorepass <jks_password>
```

<p12_password> – пароль для хранилища .p12 <jks_password> - пароль для хранилища jks После выполнения команды в директории EJBCA Web Service CLI <ejbca_ws_cli_home> появится хранилище ott-service.jks

Установка хранилища с прикладным сертификатом OTT сервера

Используя шаблон ott-srv-certs (/package/conf/k8s/base/ott-srv-app/secrets/otts-secret-certs.yaml) из дистрибутива установите сертификат сервера OTT, полученный при выполнении пункта **Создание хранилища с прикладным сертификатом OTT сервера**.

После чего будет создан необходимый секрет с хранилищем прикладного сертификата OTT сервера ott-service.jks.

Подготовка окружения в среде контейнеризации Kubernetes

Выпуск и установка сертификатов для Ingress/Egress

Используя шаблоны из дистрибутива /package/conf/k8s/base/istio/secrets/, выпустите и установите сертификаты Istio Ingress/Egress:

- egressgateway-ca-certs;
- egressgateway-certs;

- istio-ingressgateway-ca-certs;
- istio-ingressgateway-certs.

После чего будет создан набор необходимых артефактов, в том числе секреты с сертификатами cert, ca-cert.

Созданные секреты содержат, в том числе, наименования файлов: tls.crt, tls.key, ca-chain.cert.pem (вся цепочка сертификатов).

В секретах istio-ingressgateway-ca-certs и istio-egressgateway-ca-certs должен присутствовать один файл ca-chain.cert.pem со всей цепочкой сертификатов. Если в секрете указаны два отдельных сертификата, то вызовы через Ingress по HTTPS не будут работать.

Подготовка окружения в среде контейнеризации OpenShift

Подключение Контрольной панели Synapse Service Mesh

Подключение Контрольной панели Synapse Service Mesh (компонент POLM) осуществляется в соответствии с документацией на него.

Выпуск и установка сертификатов для Ingress/Egress

Необходимо выпустить и установить сертификаты Synapse Service Mesh Ingress/Egress согласно инструкции, приведенной в разделе Подготовка окружения в среде контейнеризации Kubernetes.

Подготовка окружения для ручной установки

Для формирования файла с паролями для OTT сервера, выполните следующие действия:

1. Создать файл ott-service-passwords.yml с содержимым:

```
kind: Secret apiVersion: v1 metadata: name: secret-
otts.unver data: otts.ott-srv-app.ejbca.ks.password:
<password> otts.ott-srv-app.integrity.ks.password:
<password> otts.ott-srv-app.integrity.ts.password:
<password> type: Opaque
```

2. В созданном файле задать актуальные значения параметров:

Параметр	Описание
otts.ott-srv-app.ejbca.ks.password	Пароль от keyStore ott-service.jks
otts.ott-srv-app.integrity.ks.password	Пароль от keyStore модуля контроля целостности политик авторизации (В данном дистрибутиве не используется)

otts.ott-srv-app.integrity.ts.password	Пароль от trustStore модуля контроля целостности политик авторизации (В данном дистрибутиве не используется)

Все пароли необходимо предварительно закодировать в base64, например, так:

```
echo -n "my_password" | base64
```

Подготовка окружения для автоматизированной установки с использованием Deploy Tools

Для автоматизированной установки с использованием Deploy Tools дополнительно требуется:

- наличие пространства в сервисе централизованного хранения репозиториев исходного кода и прав на создание репозиториев в нем;
- наличие области в Jenkins с настроенными учетными данными (логин/пароль) для авторизации под технологической учетной записью (ТУЗ), позволяющей загружать артефакты в целевой репозиторий хранилища.

Параметры настройки располагаются в файлах в дистрибутиве по пути: `/package/conf/config/parameters/`.

Ручная установка сервера OTT

Для ручной установки сервера OTT в среде контейнеризации необходимо во всех файлах конфигурации заменить все плейсхолдеры на реальные значения. Значение сначала необходимо определить в параметрах, расположенных в файлах настроек в дистрибутиве по пути: /package/conf/config/parameters/. Затем указанные значение параметров перенести в конфигурационные файлы.

Для настройки интеграций с внешними сервисами задать следующие параметры:

Внешний сервис	Файл настроек	Параметр	Описание
Audit (AUDT)	ott-srv-app.conf	otts.ott-srv-app.ose.configmap.data.application.yaml.audit-stub	Режим заглушки для интеграции с сервисом Аудита (компонент AUDT). Если включен, события аудита не передаются
Audit (AUDT)	ott-srv-app.conf	otts.ott-srv-app.ose.configmap.data.application.yaml.audit.url	URL для подключения к сервису Аудита

Внешний сервис	Файл настроек	Параметр	Описание
Audit (AUDT)	ott-srv-app.istio.conf	otts.ose.istio.egress.audit.common.external-audit-host	hostname сервиса Аудита
Audit (AUDT)	ott-srv-app.istio.conf	otts.ose.istio.egress.audit.common.external-audit-port	порт сервиса Аудита
Журналирование (LOGA)	ott-srv-app.conf	otts.fluent-bit.ose.configmap.data.fluent-bit.conf.output.brokers	список серверов (с указанием портов) kafka для сервиса Журналирования через запятую вида host1:port1,..,hostN:portN
Журналирование (LOGA)	ott-srv-app.conf	otts.fluent-bit.ose.configmap.data.fluent-bit.conf.output.topics	название топика kafka для сервиса Журналирования
Журналирование (LOGA)	ott-srv-app.istio.conf	otts.ott-srv-app.ose.istio.egress.ulogger.common.external-logger-hosts	список серверов kafka для сервиса Журналирования через запятую вида host1,..,hostN
УЦ	ott-srv-app.conf	otts.ott-srv-app.ose.configmap.data.application.yaml.key-service-url	URL УЦ
УЦ	ott-srv-app.istio.conf	otts.ott-srv-app.ose.istio.egress.ejbca.common.external-ejbca-host	hostname УЦ
УЦ	ott-srv-app.istio.conf	otts.ott-srv-app.ose.istio.egress.ejbca.common.external-ejbca-port	порт УЦ

Перечень конфигурационных файлов для развертывания в Kubernetes:

/package/conf/k8s/base/istio/config/egress/otts-dr-audit.yaml
/package/conf/k8s/base/istio/config/egress/otts-dr-ejbca.yaml
/package/conf/k8s/base/istio/config/egress/otts-gw-audit.yaml
/package/conf/k8s/base/istio/config/egress/otts-gw-ejbca.yaml

/package/conf/k8s/base/istio/config/egress/otts-gw-ulogger.yaml
/package/conf/k8s/base/istio/config/egress/otts-se-audit.yaml
/package/conf/k8s/base/istio/config/egress/otts-se-ejbca.yaml
/package/conf/k8s/base/istio/config/egress/otts-se-ulogger.yaml
/package/conf/k8s/base/istio/config/egress/otts-vs-audit.yaml
/package/conf/k8s/base/istio/config/egress/otts-vs-ejbca.yaml
/package/conf/k8s/base/istio/config/egress/otts-vs-ulogger.yaml
/package/conf/k8s/base/istio/config/ingress/otts-dr-ott-srv-app.yaml
/package/conf/k8s/base/istio/config/ingress/otts-gw-ott-srv-app.yaml
/package/conf/k8s/base/istio/config/ingress/otts-rt-ott-srv-app.yaml
/package/conf/k8s/base/istio/config/ingress/otts-vs-ott-srv-app.yaml
/package/conf/k8s/base/istio/deployments/egress/dc.yaml
/package/conf/k8s/base/istio/deployments/egress/otts-svc-audit.yaml
/package/conf/k8s/base/istio/deployments/egress/otts-svc-ejbca.yaml
/package/conf/k8s/base/istio/deployments/egress/otts-svc-ulogger.yaml
/package/conf/k8s/base/istio/deployments/ingress/dc.yaml
/package/conf/k8s/base/istio/deployments/ingress/otts-svc-ott-srv-app-ingress.yaml
/package/conf/k8s/base/istio/deployments/ingress/otts-svc-ott-srv-app.yaml
/package/conf/k8s/base/istio/secrets/otts-secret-istio-egressgateway-ca-certs.yaml
/package/conf/k8s/base/istio/secrets/otts-secret-istio-ingressgateway-ca-certs.yaml
/package/conf/k8s/base/istio/secrets/otts-secret-istio-egressgateway-certs.yaml
/package/conf/k8s/base/istio/secrets/otts-secret-istio-ingressgateway-certs.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-fluent.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-istio-basic.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-ott-srv-app-configuration-options.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-ott-srv-app-configuration.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-ott-srv-app-logback.yaml
/package/conf/k8s/base/ott-srv-app/dc.yaml
/package/conf/k8s/base/ott-srv-app/secrets/otts-secret-certs.yaml

Перечень конфигурационных файлов для развертывания в OpenShift:

/package/conf/k8s/base/istio/config/egress/otts-dr-audit.yaml
/package/conf/k8s/base/istio/config/egress/otts-dr-ejbca.yaml
/package/conf/k8s/base/istio/config/egress/otts-gw-audit.yaml
/package/conf/k8s/base/istio/config/egress/otts-gw-ejbca.yaml
/package/conf/k8s/base/istio/config/egress/otts-gw-ulogger.yaml
/package/conf/k8s/base/istio/config/egress/otts-se-audit.yaml
/package/conf/k8s/base/istio/config/egress/otts-se-ejbca.yaml
/package/conf/k8s/base/istio/config/egress/otts-se-ulogger.yaml
/package/conf/k8s/base/istio/config/egress/otts-vs-audit.yaml
/package/conf/k8s/base/istio/config/egress/otts-vs-ejbca.yaml
/package/conf/k8s/base/istio/config/egress/otts-vs-ulogger.yaml
/package/conf/k8s/base/istio/config/ingress/otts-dr-ott-srv-app.yaml
/package/conf/k8s/base/istio/config/ingress/otts-gw-ott-srv-app.yaml
/package/conf/k8s/base/istio/config/ingress/otts-vs-ott-srv-app.yaml
/package/conf/k8s/base/istio/deployments/egress/dc.yaml
/package/conf/k8s/base/istio/deployments/egress/otts-svc-audit.yaml
/package/conf/k8s/base/istio/deployments/egress/otts-svc-ejbca.yaml
/package/conf/k8s/base/istio/deployments/egress/otts-svc-ulogger.yaml

/package/conf/k8s/base/istio/deployments/ingress/dc.yaml
/package/conf/k8s/base/istio/deployments/ingress/otts-svc-ott-srv-app-ingress.yaml
/package/conf/k8s/base/istio/deployments/ingress/otts-svc-ott-srv-app.yaml
/package/conf/k8s/base/istio/secrets/otts-secret-istio-egressgateway-ca-certs.yaml
/package/conf/k8s/base/istio/secrets/otts-secret-istio-ingressgateway-ca-certs.yaml
/package/conf/k8s/base/istio/secrets/otts-secret-istio-egressgateway-certs.yaml
/package/conf/k8s/base/istio/secrets/otts-secret-istio-ingressgateway-certs.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-fluent.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-istio-basic.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-ott-srv-app-configuration-options.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-ott-srv-app-configuration.yaml
/package/conf/k8s/base/ott-srv-app/configmaps/otts-cm-ott-srv-app-logback.yaml
/package/conf/k8s/base/ott-srv-app/dc.yaml
/package/conf/k8s/base/ott-srv-app/secrets/otts-secret-certs.yaml
/package/conf/k8s/overrides/openshift/ott-srv-app/istio/config/ingress/otts-rt-ott-srv-app.yaml

После замены файлов конфигурации необходимо развернуть файлы конфигурации в среду контейнеризации:

1. В панели администратора в правом верхнем углу нажать на кнопку **Import YAML**;
2. В открывшейся окно вставить скопированный конфигурационный файл и нажать **Create**;
3. После создания отобразится страница конфигурационного файла;
4. Повторить действия 1–3 для всех файлов конфигурации.
5. Повторить действия 1–3 для файла с паролями для OTT сервера ott-service-passwords.yml, полученного в результате выполнения пункта **Формирование файла с паролями для OTT сервера**.

Установка и настройка Объединенного мониторинга Unimon (MONA) осуществляется согласно документации по данному сервису.

Автоматизированная установка сервера OTT с использованием Deploy Tools

Для установки сервиса с использованием компонента Deploy Tools (CDJE) программного продукта Platform V DevOps Tools (DOT) разверните сервис в инфраструктуре и запустите CD Job по инструкции, приведенной в документации Platform V DevOps Tools (DOT).

Список артефактов сервера OTT

Тип артефакта	Имя артефакта	Описание
ConfigMap	configmap-ott-service-config-fluent	ConfigMap с настройками компонента fluent-bit

Тип артефакта	Имя артефакта	Описание
ConfigMap	configmap-ott-service-configuration	ConfigMap с настройками OTT Сервера
ConfigMap	configmap-ott-service-configuration-options	ConfigMap с дополнительными настройками сервера OTT
ConfigMap	configmap-ott-service-logback-configuration	ConfigMap с настроками логирования
ConfigMap	istio-basic	ConfigMap с настройками istio
Deployment	deployment-egress	Deployment Egress GW (компонент IGEG)
Deployment	deployment-ingress	Deployment Ingress GW (компонент IGEG)
Deployment	deployment-ott-service	Deployment сервера OTT
DestinationRule	destinationrule-service-deployment-ott-service	DestinationRule для сервера OTT
DestinationRule	external-host-audit-dr	DestinationRule для сервиса Аудита (компонент AUDT)
DestinationRule	external-host-ejbca-dr	DestinationRule УЦ EJBCA
Gateway	egressgateway-audit-gw	Gateway для сервиса Аудита (компонент AUDT)
Gateway	egressgateway-ejbca-gw	Gateway для УЦ EJBCA
Gateway	egressgateway-ulogger-gw-*	Gateway для сервиса Журналирования (компонент LOGA)
Gateway	gateway-deployment-ingressgateway-ott-service	Gateway для сервера OTT
Route	route-service-deployment-ingressgateway-ott-service-https	Route сервера OTT
Ingress	ott-service-ingress	Ingress сервера OTT

Тип артефакта	Имя артефакта	Описание
Secret	istio-ingressgateway-ca-certs	Secret для Ingress GW (компонент IGEG)
Secret	istio-ingressgateway-certs	Secret для Ingress GW (компонент IGEG)
Secret	egressgateway-ca-certs	Secret для Egress GW (компонент IGEG)
Secret	egressgateway-certs	Secret для Egress GW (компонент IGEG)
Secret	ott-srv-certs	Secret с сертификатми для сервера OTT
Secret	ott-service-passwords	Secret с паролями для сервера OTT
Service	service-deployment-egressgateway-audit-ott-service	Service egressgateway для сервиса Аудита (компонент AUDT)
Service	service-deployment-egressgateway-ejbca-ott-service	Service egressgateway для УЦ EJBCA
Service	service-deployment-egressgateway-ulogger-ott-service	Service egressgateway для сервиса Журналирования (компонент LOGA)
Service	service-deployment-ingressgateway-ott-service	Service ingressgateway для сервера OTT
Service	service-deployment-ott-service	Service для сервера OTT
ServiceEntry	external-host-audit-se	ServiceEntry для сервиса Аудита (компонент AUDT)
ServiceEntry	external-host-ejbca-se	ServiceEntry для УЦ EJBCA
ServiceEntry	external-host-ulogger-se-*	ServiceEntry для сервиса Журналирования (компонент LOGA)
VirtualService	external-host-audit-vs	VirtualService для сервиса Аудита (компонент AUDT)
VirtualService	external-host-ejbca-vs	VirtualService для УЦ EJBCA

Тип артефакта	Имя артефакта	Описание
VirtualService	external-host-ulogger-vs-*	VirtualService для сервиса Журналирования (компонент LOGA)
VirtualService	virtualservice-service-deployment-ott-service	VirtualService для сервера OTT

Клиент OTT (OTT Sidecar)

Конфигурирование клиента OTT (sidecar контейнер)

- Получение сертификата приложения.

С целью исключения необходимости передачи секретов между разными командами сопровождения при настройке клиента OTT в облачной инфраструктуре предлагается выполнять по следующему сценарию:

1. Администратор прикладного модуля: выполняет генерацию ключевой пары в p12-контейнере

```
keytool -genkey -keyalg EC -sigalg SHA256withECDSA -keystore ${module_id}.p12 -storetype PKCS12 -keysize 256 -dname "CN=${module_id}" -alias ${module_id}
```

2. Администратор прикладного модуля: формирует запрос на сертификат

```
keytool -certreq -keyalg EC -sigalg SHA256withECDSA -keystore ${module_id}.p12 -storetype PKCS12 -alias ${module_id} > ${module_id}_cert_req.pem
```

В результате выполнения данной команды создастся файл CSR `{module_id}_cert_req.pem`, который необходимо передать Администратору OTT в заявке на генерацию сертификата для модуля. В заявке явно указать, что сертификат должен быть сгенерирован из приложенного запроса по данной инструкции.

3. Администратор OTT: выписывает сертификат
4. Администратор прикладного модуля: импортирует сертификат УЦ OTT и сертификат модуля в keystore, созданный на 1 шаге (строго в указанном порядке)

```
keytool -import -keystore ${module_id}.p12 -storetype PKCS12 -file SberbankPlatformCA_EC.pem -alias SberbankPlatformCA_EC
keytool -import -keystore ${module_id}.p12 -storetype PKCS12 -file ${module_id}.pem -alias ${module_id}
```

- Получение сертификата публичного ключа OTT Service.

```
# Download OTT Service certificate curl -X GET "https://<ott-service-host>:8442/ejbca/publicweb/webdist/certdist?cmd=lastcert&subject=cn%3Dott-service" --insecure > ott-service.crt # Import OTT Service certificate into PKCS12 store keytool -importcert -file ott-service.crt -keystore ott_truststore.p12 -storetype PKCS12 -storepass <password> -alias "ott-service" -noprompt
```

- Подключение хранилищ, полученных на шаге 1 и 2 в Secrets: Хранилища p12 необходимо подключить с типом Opaque. Файлы с сертификатами нужно преобразовать в base64 строку с помощью bash-команды base64 или аналогом, удалить переносы строк. При загрузке файла через веб-интерфейс Kubernetes преобразование может выполняться некорректно. Например, для подключения truststore и ключа прикладного модуля:

```
apiVersion: v1 data: ott_service_truststore.p12: >-
...base64store... <module-id>.p12: >-
...base64store... kind: Secret metadata: name:
<module-id>-ott-certs namespace: <namespace> type:
Opaque
```

Сертификаты для формирования Truststore необходимо запросить у команды администрирования OTT.

```
apiVersion: v1 data: OTT_CERTSTORE_PRIVATE_KEY_PWD:
<password> OTT_CERTSTORE_PWD: <password>
OTT_TRUST_STORE_PWD: <password> kind: Secret
metadata: name: <module-id>-ott-passwords
namespace: <namespace> type: Opaque
```

- OTT_TRUST_STORE_PWD – пароль от контейнера с сертификатом OTT Сервиса.
- OTT_CERTSTORE_PWD – пароль от контейнера с сертификатом модуля.
- OTT_CERTSTORE_PRIVATE_KEY_PWD – пароль для извлечения секретного ключа модуля (=OTT_CERTSTORE_PWD).

Не рекомендуется создавать секрет с паролями через CLI в Windows, могут быть проблемы с кодировкой. Надежнее создать секрет с паролями через Web-UI Kubernetes

- Конфигурирование

Загрузить ConfigMap с настройками OTT Sidecar:

```
apiVersion: v1 kind: ConfigMap metadata: name:
<module-id>-ott-settings namespace: <namespace>
```

```
data:        OTT_AUTHZ_REALM: mmt        OTT_CERTSTORE_PATH:
/mnt/secrets/<module-id>.p12             OTT_CERTSTORE_TYPE:
PKCS12         OTT_CLIENT_CERT_ALIAS: <module-id> (в нижнем
регистре)                    OTT_GRPC_PORT:   /mnt/ott-uds-
socket/ott.socket       OTT_MODULE_ID: <module-id> (с учетом
регистра)               OTT_SERVICE_CERT_ALIAS:  ott-service
OTT_SERVICE_HOSTS:  >-            ott-server-host1:port,ott-
server-host2:port           OTT_SERVICE_URL:  'https://<ott-
server-host1>:<port>/ott-service/rest/token'   #шаблон   URL
сервиса    OTT,   нужно   вписать   реальный   host:port,   при
балансировке    значения    будут    подставляться    из
OTT_SERVICE_HOSTS                       OTT_TRUST_STORE_PATH:
/mnt/secrets/ott_service_truststore.p12       OTT_OPER_MODE:
Значение - "sign" (для egress), "validate" - (для ingress)
OTT_AUTHZ_VERSION:   '1.0'   #версия   метода   авторизации,
актуально    для    версии    образа    4.0.4    и    выше
OTT_CLIENT_MMT-COMPATIBILITY-MODE:      'false'       #режим
совместимости с MMT 3 поколения, актуально для версии образа
4.0.4 и выше
```

Загрузить ConfigMap с настройками журналирования OTT.

```
kind: ConfigMap     apiVersion: v1     metadata:       name:
ott-logback           data:                  logback.xml:   |
<configuration    scan="true"    scanPeriod="60    seconds">
<appender                                      name="STDOUT"
class="ch.qos.logback.core.ConsoleAppender">
<encoder>                              <pattern>%X{ott.req.id}
%d{HH:mm:ss.SSS}   [%-5level]   [%logger{36}]   [%thread]   -
%msg%n</pattern>             </encoder>          </appender>
<appender                                       name="FILE"
class="ch.qos.logback.core.FileAppender">
<file>/tmp/OttTest.log</file>
<append>false</append>                                  <encoder>
<pattern>%X{ott.req.id}      %d{HH:mm:ss.SSS}      [%-5level]
[%logger{36}]       [%thread]       -       %msg%n</pattern>
</encoder>          </appender>          <root level="debug">
<appender-ref  ref="STDOUT"/>                   <appender-ref
ref="FILE"/>                 </root>                 <logger
name="com.sbt.ott"  level="debug"/>                 <logger
name="com.sbt.ott.api.rest" level="debug"/>         <logger
name="com.sbt.core"  level="debug"/>                <logger
name="com.fasterxml" level="debug"/>     </configuration>
```

Задать переменную окружения, определяющую путь к файлу `logback.xml:JAVA_TOOL_OPTIONS=-Dlogback.configurationFile=/tmp/ott-logback/logback.xml`.

После отладки рекомендуется переключить уровень `<root level="info">`, отдельные логеры так же в "info", либо удалить.

Необходимо подключить в Deployment UDS сокет как EmptyDir в режиме Memory, пример в блоке ниже.

Конфиг logback и остальные секреты так же в volumes:

```
volumes:          - name: ott-uds-socket            emptyDir:
medium: Memory        - name: ott-logback          configMap:
name: ott-logback            defaultMode: 420        - name:
ott-certs           secret:             secretName: ott-certs
defaultMode: 420
```

Ранее созданные секреты и конфиги, а также UDS сокет, подключить к образу OTT сайдкар:

```
volumeMounts:        - name: ott-certs          readOnly: true
mountPath:  /mnt/secrets              -  name:  ott-uds-socket
mountPath:  /mnt/ott-uds-socket           -  name:  ott-logback
readOnly: true        mountPath: /tmp/ott-logback
```

Пример Deployment с образами istio-proxyv2 и ott-sidecar:

```
kind: Deployment apiVersion: apps/v1 metadata:    name: ott-
egressgateway       labels:           app:  ott-egressgateway
istio:  ott-egressgateway spec:     replicas:  1     selector:
matchLabels:        app: ott-egressgateway        istio: ott-
egressgateway     template:        metadata:           labels:
app:  ott-egressgateway              istio:  ott-egressgateway
maistra-control-plane:       ci01994970-edevgen-control-panel
annotations:                sidecar.istio.io/inject:  'false'
spec:        restartPolicy: Always        serviceAccountName:
ott-egressgateway-service-account              schedulerName:
default-scheduler          affinity:            nodeAffinity:
requiredDuringSchedulingIgnoredDuringExecution:
nodeSelectorTerms:                          -  matchExpressions:
- key: beta.kubernetes.io/arch                        operator:
In                     values:                          - amd64
-  ppc64le                                              -  s390x
preferredDuringSchedulingIgnoredDuringExecution:
-   weight:   2                                     preference:
matchExpressions:                                        -  key:
beta.kubernetes.io/arch                            operator: In
values:                          - amd64              - weight:
2                                                   preference:
matchExpressions:                                        -  key:
beta.kubernetes.io/arch                            operator: In
```

```
values:                              - ppc64le                -
weight:   2                                          preference:
matchExpressions:                                        -  key:
beta.kubernetes.io/arch                           operator: In
values:                                                  -  s390x
terminationGracePeriodSeconds: 30        securityContext: {}
containers:             - resources:                   limits:
cpu: 700m                memory: 800Mi              requests:
cpu:  700m                                     memory:  800Mi
readinessProbe:               httpGet:                  path:
/healthz/ready                                    port:  15020
scheme:  HTTP                          initialDelaySeconds:  1
timeoutSeconds:  1                           periodSeconds:  2
successThreshold:  1                     failureThreshold:  30
terminationMessagePath: /dev/termination-log            name:
istio-proxy              env:                - name: POD_NAME
valueFrom:                                          fieldRef:
apiVersion: v1                        fieldPath: metadata.name
-  name:  POD_NAMESPACE                             valueFrom:
fieldRef:                                      apiVersion:  v1
fieldPath:  metadata.namespace                        -  name:
INSTANCE_IP                                        valueFrom:
fieldRef:                                      apiVersion:  v1
fieldPath:  status.podIP                     -  name:  HOST_IP
valueFrom:                                          fieldRef:
apiVersion: v1                        fieldPath: status.hostIP
-  name:  ISTIO_META_POD_NAME                       valueFrom:
fieldRef:                                      apiVersion:  v1
fieldPath:  metadata.name                             -  name:
ISTIO_META_CONFIG_NAMESPACE                        valueFrom:
fieldRef:                                      apiVersion:  v1
fieldPath:  metadata.namespace                        -  name:
ISTIO_META_ROUTER_MODE                        value:  sni-dnat
ports:                              -  name:  http-envoy-prom
containerPort:  15090                           protocol:  TCP
imagePullPolicy:  IfNotPresent                   volumeMounts:
-  name:  istio-certs                          readOnly:  true
mountPath: /etc/certs                  - name: egressgateway-
certs                readOnly: true                mountPath:
/etc/istio/egressgateway-certs                        -  name:
egressgateway-ca-certs                         readOnly:  true
mountPath: /etc/istio/egressgateway-ca-certs                -
name:  mtls-client-certs                       readOnly:  true
mountPath: /etc/istio/mtls-egress-certs               - name:
mtls-client-ca-certs                           readOnly:  true
mountPath:  /etc/istio/mtls-egress-ca-certs                 -
name: ott-uds-socket                  mountPath: /mnt/ott-uds-
socket                        terminationMessagePolicy:  File
image:                                                     >-
ci01994970/ci02320655_synapse_security_dev/proxyv2-
```

```
rhel8:1.1.1                 args:                       - proxy
-  router                   -  '--domain'                    -
$(POD_NAMESPACE).svc.cluster.local                      -  '--
log_output_level=default:info'                          -  '--
drainDuration'                  -  45s                  -  '--
parentShutdownDuration'             -  1m0s             -  '--
connectTimeout'                 -  10s                  -  '--
serviceCluster'                          -  ott-egressgateway
-  '--zipkinAddress'                    -  'zipkin.ci01994970-
edevgen-control-panel:9411'             -  '--proxyAdminPort'
-  '15000'             -  '--statusPort'             -  '15020'
-  '--controlPlaneAuthPolicy'                           -  NONE
-  '--discoveryAddress'                              -  'istio-
pilot.ci01994970-edevgen-control-panel:15010'              -
name:   ott-sidecar                                 image:   >-
ci00641491/ci01125613_ott/ott-client-
api@sha256:764085215941ecb7e90429326113fe63b78b024e8595ac324
9c5d88d128574ac           envFrom:              - secretRef:
name: <module-id>-ott-passwords               - configMapRef:
name:  <module-id>-ott-settings                  resources:
limits:               cpu: 700m               memory: 800Mi
requests:                   cpu: 700m                  memory:
800Mi           startupProbe:                       httpGet:
path: /actuator/health/startup                         port:
8090                                    failureThreshold:  30
periodSeconds:   10                              livenessProbe:
httpGet:                                                path:
/actuator/health/liveness                          port: 8090
initialDelaySeconds: 0                       timeoutSeconds: 2
failureThreshold: 3                          periodSeconds: 10
readinessProbe:                                      httpGet:
path: /actuator/health/readiness                       port:
8090                                   initialDelaySeconds:  5
timeoutSeconds: 1                          successThreshold: 1
failureThreshold: 3                           periodSeconds: 5
volumeMounts:                              -  name:  ott-certs
readOnly:  true                         mountPath:  /mnt/secrets
-  name:  ott-logback                             readOnly:  true
mountPath:  /tmp/ott-logback                  - name: ott-uds-
socket                        mountPath:  /mnt/ott-uds-socket
terminationMessagePath:                   /dev/termination-log
terminationMessagePolicy:  File                imagePullPolicy:
IfNotPresent                serviceAccount:  ott-egressgateway-
service-account         volumes:            - name: istio-certs
secret:                   secretName:  istio.ott-egressgateway-
service-account                                 defaultMode:  420
optional:  true                   -  name:  egressgateway-certs
secret:                  secretName:  istio-egressgateway-certs
defaultMode:  420               optional:  true           - name:
egressgateway-ca-certs                                    secret:
```

```
secretName:                    istio-egressgateway-ca-certs
defaultMode: 420             optional: true          - name:
mtls-client-certs           secret:              secretName:
ingress-client-cert                          defaultMode:  420
optional:  true               -  name:  mtls-client-ca-certs
secret:                secretName: istio-ingressgateway-ca-
certs              defaultMode: 420               optional:
true           - name: ott-logback               configMap:
name: ott-logback               defaultMode: 420          -
name: ott-certs            secret:               secretName:
<module-id>-ott-certs             defaultMode: 420         -
name: ott-uds-socket           emptyDir:              medium:
Memory       dnsPolicy: ClusterFirst   strategy:      type:
RollingUpdate      rollingUpdate:        maxUnavailable: 25%
maxSurge: 25%
```

Подключите фильтр авторизации, который будет вызывать OTT:

```
kind: EnvoyFilter apiVersion: networking.istio.io/v1alpha3
metadata:          name:    ott-egress-authz-filter    spec:
workloadSelector:     labels:       app: ott-egressgateway #
Соответствует метке вашего деплоймента egress шлюза, то есть
применяется только туда     configPatches:        - applyTo:
HTTP_FILTER           match:                context: GATEWAY
listener:              filterChain:                 filter:
name: envoy.http_connection_manager              portNumber:
11443 # Указанный порт соответствует порту, определенному
ресурсом   networking.istio.io/gateway - вы создадли его,
когда настраивали маршрутизацию через егресс         patch:
operation: INSERT_BEFORE           value:             name:
envoy.sbt_authz                                    config:
authorize_only_request: false # Флаг того, что авторизуются
как     HTTP     Request,     так     и     HTTP     Response
failure_mode_allow:    false    #    Запрещает    трафик    при
недоступности сервера авторизации              grpc_service:
google_grpc:                        stat_prefix: sbt_authz
target_uri: 'unix:/mnt/ott-uds-socket/ott.socket' # путь к
файлу   соответствует   точке   монтирования   в   деплойменте
timeout: 2s # значение as is на фактическую производительно
влияние не оказывает но необходимо для корректного запуска,
может    быть    скорректировано    по    результатам    HT
sidecar_role: EGRESS # Флаг того, что фильтр устанавливается
на  egress  proxy,  для  ingress  -  INGRESS  соответственно
with_request_body: # Параметр, указывающий фильтру, что в
запрос к сервису авторизации также нужно добавлять тело HTTP
запроса,          если          оно          присутствует;
allow_partial_message:                                 false
max_request_bytes:  100000  #  данный  параметр  ограничит
максимальный размер запроса. в слуяае превышения запрос не
```

```
пройдет авторизацию или проверку целостности - указывайте его с запасом                pack_as_bytes: false # Флаг того, что тело запроса передается в виде строки в сервис авторизации, в случае необходимости поддержки бинарного тела - true;
```

Создайте ServiceEntry в вашем проекте для того, чтобы клиент OTT мог связаться с сервером:

```
apiVersion: networking.istio.io/v1alpha3 kind: ServiceEntry metadata:     name: ott-server spec:     addresses:     - $(SERVER_IP) # уточните адреса серверов у администраторов OTT    exportTo:    - .    hosts:    - localhost    location: MESH_EXTERNAL    ports:    - name: tcp-8080      number: 8080 protocol: TCP
```

Установка EJBCA на сервере приложений Wildfly (опционально)

В разделе описаны процедуры развертывания компонентов сервиса.

База данных

1. Установить БД, поддерживающую Transact-SQL.
2. Создать схему БД:

```
CREATE USER OTT IDENTIFIED BY "OTT" DEFAULTTABLESPACE "OTT" TEMPORARY TABLESPACE "TEMP"; GRANT CREATE SESSION TO OTT; GRANT CREATE TABLE TO OTT;  GRANT UNLIMITED TABLESPACE TO OTT;COMMIT;
```

Серверы приложений

1. [Администратор OTT] – Установить **JDK** (1.8 Oracle HotSpot, min 1.8.0_162 + пакет политик Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files for JDK/JRE) + **Wildfly** + **Конфигуратор Hyperic** + **seap-lib** + **seap-lib-deps** + **Журналирование**.
2. [Администратор OTT] – Подготовка к установке EJBCA.
3. [Администратор Wildfly] – Установить JDBC драйвер.
4. [Администратор Wildfly] – Создать DataSource и настроить Wildfly Remoting для EJBCA.
5. [Администратор Wildfly] – Установить EJBCA.
6. [Администратор Wildfly] – Настроить Security Realm, интерфейсы и системные свойства для EJBCA.
7. [Администратор Wildfly] – Выполнить первичное конфигурирование EJBCA.
8. [Администратор OTT] – Сгенерировать ключевую пару для модуля OTT.

Подготовка к установке EJBCA

Решение подразумевает использование 2-х standalone серверов Wildfly для EJBCA и одной общей БД. При этом установка приложения EJBCA для серверов Wildfly отличается и выполняется последовательно. Инструкции необходимо выполнить один раз только на 1 сервере wildfly (EJBCA master host).

1. Убедиться, что утилита **keytool** доступна и для нее создана ссылка **/usr/bin/keytool**.
2. Выполнить установку **ant**и **unzip**.
3. Распаковать дистрибутив **EJBCA**, например в каталог /opt/:

```
unzip  <tmp>/other/ejbca/cesecore-6.10.1.4.zip  -d  /opt/
/<tmp>
```

Далее каталог, в который была выполнена распаковка EJBCA, будет указываться как **<ejbca_home>**

4. В файле **<ejbca_home>/conf/database.properties** задать параметры подключения к **БД:**

Параметр	Описание
database.url	jdbc URL для подключения к БД
database.username	логин пользователя для подключения к БД
database.password	пароль пользователя для подключения к БД

5. В файле **<ejbca_home>/conf/ejbca.properties** в параметре **appserver.home** задать домашнюю директорию **wildfly**.
6. В файле **<ejbca_home>/conf/cesecore.properties** в параметре **ca.keystorepass** задать pin для создаваемого CA.

   Примечание: избегайте в пароле специальных символов, которые нужно экранировать, таких как $. Ant может обрабатывать их некорректно.

7. В файле **<ejbca_home>/conf/web.properties** задать параметры:

Параметр	Описание
java.trustpassword	Password for java **trust keystore** (p12/truststore.jks) (*Придумать)*
superadmin.password	The password used to protect the generated **super administrator P12 keystore** (to be imported in browser) (*Придумать)*
httpsserver.password	The password used to protect the web server's **SSL keystore** (*Придумать)*

Параметр	Описание
httpsserver.hostname	DNS имя балансировщика для OTT. Если его нет, указатьDNS имя сервера, на котором производится установка
httpsserver.an	*Необязательный параметр* The Subject Alternative Name (X.509 extension) of the SSL server certificate. Используется для конфигурации с 2-мя серверами без балансировщика. В этом случае в параметре **httpsserver.hostname** задаётся hostname 1-го сервера Wildfly, а в параметре **httpsserver.an**указывается 2 значения через запятую.

8. Временно дать на все содержимое каталога **<ejbca_home>** полномочия на **чтение/запись/выполнение** для пользователя, из-под которого работает wildfly. Например, так:

```
chmod -R 777 <ejbca_home>
```

Установка JDBC драйвера

Все шаги выполняются на всех серверах OTT текущего контура. Подразумевается, что команда выполняется из пути $JBOSS_HOME (домашняя директория WIldfly).

1. Устанавливаем JDBC-драйвер PostgreSQL как модуль Wildfly (от пользователя с правами чтения/записи в директории Wildfly!)

```
bin/jboss-cli.sh -c --command="module add --
name=org.postgresql --resources=/tmp/postgresql-
42.2.16.jar --
dependencies=javax.api,javax.transaction.api"
```

2. Регистрируем JDBC-драйвер PostgreSQL в Wildfly

```
bin/jboss-cli.sh -c --
command="/subsystem=datasources/jdbc-
driver=postgres:add(driver-name="postgres",driver-module-
name="org.postgresql",driver-class-
name=org.postgresql.Driver)"
```

DataSource и Wildfly Remoting для EJBCA

Выполнить предустановочное конфигурирование Wildfly для EJBCA:

1. Подключиться к JBoss CLI

```
cd <wf_home>/bin/ ./jboss-cli.sh -c
```

2. Выполнить команды, предварительно задав необходимые параметры подключения к БД (запросить у администратора ОТТ):

- <jdbc_connection_url> – jdbc URL для подключения к БД
- <db_user> – логин пользователя для подключения к БД
- <db_password> – пароль пользователя для подключения к БД

```
Значение параметра driver-name должно соответствовать имени, под которым был установлен jdbc драйвер. Посмотреть список развернутых приложений можно в jboss-cli с помощью команды: deployment-info ### Configure Data Source data-source add --name=ejbcads --driver-name="postgres" --connection-url="<jdbc_connection_url>" --jndi-name="java:/EjbcaDS" --use-ccm=true --driver-class="org.postgresql.Driver" --user-name="<db_user>" --password="<db_password>" --validate-on-match=true --background-validation=false --prepared-statements-cache-size=50 --share-prepared-statements=true --min-pool-size=5 --max-pool-size=150 --pool-prefill=true --transaction-isolation=TRANSACTION_READ_COMMITTED --check-valid-connection-sql="select 1 from dual" data-source enable --name=ejbcads ### Configure Wildfly Remoting (для WF10) /subsystem=remoting/http-connector=http-remoting-connector:remove /subsystem=remoting/http-connector=http-remoting-connector:add(connector-ref="remoting", security-realm="ApplicationRealm") /socket-binding-group=standard-sockets/socket-binding=remoting:add(port="4447") /subsystem=undertow/server=default-server/http-listener=remoting:add(socket-binding=remoting) ### Configure Wildfly Remoting (для WF15) #/subsystem=remoting/http-connector=http-remoting-connector:remove #/subsystem=remoting/http-connector=http-remoting-connector:add(connector-ref="remoting",security-realm="ApplicationRealm") #/socket-binding-group=standard-sockets/socket-binding=remoting:add(port="4447",interface="management") #/subsystem=undertow/server=default-server/http-listener=remoting:add(socket-binding=remoting,enable-http2=true) ### Configure logging /subsystem=logging/logger=org.ejbca:add /subsystem=logging/logger=org.ejbca:write-attribute(name=level, value=INFO)
```

```
/subsystem=logging/logger=org.cesecore:add
/subsystem=logging/logger=org.cesecore:write-
attribute(name=level, value=INFO) ### Проверить статус
сервера, удостовериться, что нет ошибок :read-
attribute(name=server-state) ### Ожидаемый результат #
"outcome" => "success", # "result" => "reload-required",
# "response-headers" => {"process-state" => "reload-
required"} ### Reload server if (outcome == success &&
result == reload-required) of :read-attribute(name=server-
state) :reload end-if
```

3. Дождаться завершения процесса перезагрузки (для проверки доступности сервера можно в jboss-cli выполнить команду: read-attribute(name=server-state)) и проверить конфигурацию:

```
/subsystem=datasources/data-source=ejbcads:read-
attribute(name="enabled") ### Ожидаемый результат #
"outcome" => "success", # "result" => true
```

Установка EJBCA

ВАЖНО Решение подразумевает использование 2-х standalone серверов Wildfly для EJBCA и одной общей БД PostgreSQL. При этом установка приложения EJBCA для серверов Wildfly отличается и выполняется последовательно. Сначала для 1-го сервера выполняется инструкция EJBCA master, затем для 2-го сервера - EJBCA Slave

EJBCA Master

Подключиться к JBoss CLI

```
cd <wf_home>/bin/ ./jboss-cli.sh -c
```

1. Выполнить процедуру развертывания приложения EJBCA в рабочем окружении:

```
# deploy JDBC-driver deploy --disabled
<ejbca_home>/dist/ejbca.ear deploy --name=ejbca.ear # check
deployments deployment-info # exit q
```

2. Выполнить конфигурирование EJBCA и первичную генерацию ключей:

```
cd <ejbca_home>/ ant runinstall
```

3. Выполнить копирование в *readonly директорию* сгенерированных хранилищ сертификатов: `<ejbca_home>/p12/tomcat.jks` (хранилище с секретным ключом EJBCA) и `<ejbca_home>/p12/truststore.jks` (хранилище с доверенными сертификатами для EJBCA).
4. Задать параметр `owner` для каталога `<ejbca_home>/p12/`, указать учетную запись администратора OTT, например так:

```
sudo chown -R wildflylog:wildflylog <ejbca_home>/p12/
```

EJBCA Slave

1. Выполнить копирование ключей, сгенерированных ранее с 1-го сервера на второй, сохранить тот же путь *readonly директории*.
2. Выполнить процедуру развертывания ejbca.ear, скопировав его с 1-го сервера из <ejbca_home>/dist/ejbca.ear.
3. Убедиться, что приложение успешно развернулось в рабочем окружении.

Security Realm для EJBCA

Выполнить постустановочное конфигурирование Wildfly для EJBCA:

1. Подключиться к JBoss CLI:

```
cd <wf_home>/bin/ ./jboss-cli.sh -c
```

Выполнить команды (без изменения):

```
###Update interfaces if (outcome != success) of
/interface=http:read-resource /interface=http:add(inet-
address="0.0.0.0") end-if if (outcome != success || result
!= "0.0.0.0") of /interface=http:read-attribute(name=inet-
address) /interface=http:write-attribute(name=inet-
address,value="0.0.0.0") end-if if (outcome != success) of
/interface=httpspub:read-resource
/interface=httpspub:add(inet-address="0.0.0.0") end-if if
(outcome != success) of /interface=httpspriv:read-resource
/interface=httpspriv:add(inet-address="0.0.0.0") end-if ###
Remove unused if (outcome == success) of
/subsystem=undertow/server=default-server/http-
listener=default:read-resource
/subsystem=undertow/server=default-server/http-
listener=default:remove end-if ### Update socket-bindings #
http if (outcome != success) of /socket-binding-
group=standard-sockets/socket-binding=http:read-resource
/socket-binding-group=standard-sockets/socket-
```

```
binding=http:add(port=8080,interface="http")    end-if    if
(outcome != success || result != 8080) of /socket-binding-
group=standard-sockets/socket-binding=http:read-
attribute(name=port)          /socket-binding-group=standard-
sockets/socket-binding=http:write-
attribute(name=port,value=8080)   end-if   if   (outcome   !=
success   ||   result   !=   "http")   of   /socket-binding-
group=standard-sockets/socket-binding=http:read-
attribute(name=interface)     /socket-binding-group=standard-
sockets/socket-binding=http:write-
attribute(name=interface,value="http") end-if # httpspriv if
(outcome   !=   success)   of   /socket-binding-group=standard-
sockets/socket-binding=httpspriv:read-resource      /socket-
binding-group=standard-sockets/socket-
binding=httpspriv:add(port=8444,interface="httpspriv")   end-
if if (outcome != success || result != 8444) of /socket-
binding-group=standard-sockets/socket-
binding=httpspriv:read-attribute(name=port) /socket-binding-
group=standard-sockets/socket-binding=httpspriv:write-
attribute(name=port,value=8444)   end-if   if   (outcome   !=
success   ||   result   !=   "httpspriv")   of   /socket-binding-
group=standard-sockets/socket-binding=httpspriv:read-
attribute(name=interface)     /socket-binding-group=standard-
sockets/socket-binding=httpspriv:write-
attribute(name=interface,value="httpspriv")     end-if     #
httpspub   if   (outcome   !=   success)   of   /socket-binding-
group=standard-sockets/socket-binding=httpspub:read-resource
/socket-binding-group=standard-sockets/socket-
binding=httpspub:add(port=8442, interface="httpspub") end-if
if (outcome != success || result != 8442) of /socket-
binding-group=standard-sockets/socket-binding=httpspub:read-
attribute(name=port)          /socket-binding-group=standard-
sockets/socket-binding=httpspub:write-
attribute(name=port,value=8442)   end-if   if   (outcome   !=
success   ||   result   !=   "httpspub")   of   /socket-binding-
group=standard-sockets/socket-binding=httpspub:read-
attribute(name=interface)     /socket-binding-group=standard-
sockets/socket-binding=httpspub:write-
attribute(name=interface,value="httpspub")   end-if       ###
Проверить статус сервера, удостовериться, что нет ошибок
:read-attribute(name=server-state) ### Ожидаемый результат #
"outcome" => "success", #     "result" => "reload-required",
#       "response-headers"   =>   {"process-state"   =>   "reload-
required"}    ### Reload if (outcome == success && result ==
reload-required)     of     :read-attribute(name=server-state)
:reload end-if
```

2. Дождаться завершения процесса перезагрузки (для проверки доступности сервера можно в jboss-cli выполнить команду: read-attribute(name=server-state)) и проверить конфигурацию:

```
/socket-binding-group=standard-sockets/socket-
binding=httpspriv:read-resource ### Ожидаемый результат #
"outcome" => "success", # "result" => { # "client-
mappings" => undefined, # "fixed-port" => false, #
"interface" => "httpspriv", # "multicast-address" =>
undefined, # "multicast-port" => undefined, #
"name" => "httpspriv", # "port" => 8444 # }
/socket-binding-group=standard-sockets/socket-
binding=httpspub:read-resource ### Ожидаемый результат #
"outcome" => "success", # "result" => { # "client-
mappings" => undefined, # "fixed-port" => false, #
"interface" => "httpspub", # "multicast-address" =>
undefined, # "multicast-port" => undefined, #
"name" => "httpspub", # "port" => 8442 # }
```

3. Выполнить конфигурирование Security Realm в JBoss CLI, предварительно заполнить параметры для хранилищ сертификатов (запросить у администратора ОТТ):

- <ejbca_keystore> – путь к хранилищу сертификатов keystore EJBCA, расположен в *readonly директории* (см. Установка EJBCA);
- <ejbca_truststore> – путь к хранилищу сертификатов truststore EJBCA, расположен в *readonly директории* (см. Установка EJBCA);
- <httpsserver.password> – пароль для доступа к хранилищу сертификатов keystore EJBCA;
- <httpsserver.hostname> – псевдоним (alias) ключа в keystore EJBCA;
- <java.trustpassword> – пароль для доступа к хранилищу сертификатов truststore EJBCA.

Configure EjbcaSSLRealm

if (outcome != success) of /core-service=management/security-realm=EjbcaSSLRealm:read-resource /core-service=management/security-realm=EjbcaSSLRealm:add() end-if if (outcome != success) of /core-service=management/security-realm=EjbcaSSLRealm/server-identity=ssl:read-resource /core-service=management/security-realm=EjbcaSSLRealm/server-identity=ssl:add(keystore-path="<ejbca_keystore>", keystore-password="<httpsserver.password>", alias="<httpsserver.hostname>") end-if if (outcome != success) of /core-service=management/security-realm=EjbcaSSLRealm/authentication=truststore:read-resource /core-service=management/security-realm=EjbcaSSLRealm/authentication=truststore:add(keystore-path="<ejbca_truststore>", keystore-password="<java.trustpassword>") end-if

Проверить статус сервера, удостовериться, что нет ошибок

:read-attribute(name=server-state)

Ожидаемый результат

"outcome" => "success",

"result" => "reload-required",

"response-headers" => {"process-state" => "reload-required"}

Reload

if (outcome == success && result == reload-required) of :read-attribute(name=server-state) :reload end-if

4. Дождаться завершения процесса перезагрузки (для проверки доступности сервера можно в jboss-cli выполнить команду: read-attribute(name=server-state)) и проверить конфигурацию:

```
/core-service=management/security-realm=EjbcaSSLRealm:read-
resource ### Ожидаемый результат # "outcome" =>
"success", # "result" => { # "map-groups-to-roles"
=> true, # "authentication" => {"truststore" =>
undefined}, # "authorization" => undefined, #
"plug-in" => undefined, # "server-identity" => {"ssl"
=> undefined} # } /core-service=management/security-
realm=EjbcaSSLRealm/server-identity=ssl:read-resource ###
Ожидаемый результат # "outcome" => "success", #
"result" => { # "alias" => "ott-dev-web-1", #
"enabled-cipher-suites" => undefined, # "enabled-
protocols" => [ # "TLSv1", #
"TLSv1.1", # "TLSv1.2" # ], #
"generate-self-signed-certificate-host" => undefined, #
"key-password" => undefined, # "keystore-password" =>
"tivoli1", # "keystore-path" =>
"/opt/wildfly/standalone/configuration/keystore/keystore.jks
", # "keystore-provider" => "JKS", #
"keystore-relative-to" => undefined, # "protocol" =>
"TLS" # } /core-service=management/security-
realm=EjbcaSSLRealm/authentication=truststore:read-resource
### Ожидаемый результат # "outcome" => "success", #
"result" => { # "keystore-password" => "tivoli1", #
"keystore-path" =>
"/opt/wildfly/standalone/configuration/keystore/truststore.j
ks", # "keystore-provider" => "JKS", #
"keystore-relative-to" => undefined # }
```

5. Выполнить финальные шаги конфигурирования (без изменения):

```
### Configure http-listener if (outcome != success) of
/subsystem=undertow/server=default-server/http-
listener=http:read-resource
/subsystem=undertow/server=default-server/http-
listener=http:add(socket-binding="http",redirect-
socket="httpspriv") end-if if (outcome != success || result
!= "http") of /subsystem=undertow/server=default-
```

```
server/http-listener=http:read-attribute(name=socket-
binding)      /subsystem=undertow/server=default-server/http-
listener=http:write-attribute(name=socket-binding,
value="http") end-if if (outcome != success || result !=
"httpspriv")      of      /subsystem=undertow/server=default-
server/http-listener=http:read-attribute(name=redirect-
socket)      /subsystem=undertow/server=default-server/http-
listener=http:write-attribute(name=redirect-socket,
value="httpspriv") end-if   ### Configure https-listener #
httpspriv      if      (outcome      !=      success)      of
/subsystem=undertow/server=default-server/https-
listener=httpspriv:read-resource
/subsystem=undertow/server=default-server/https-
listener=httpspriv:add(socket-binding=httpspriv,security-
realm="EjbcaSSLRealm",verify-client=REQUIRED,max-
parameters=2048) end-if if (outcome != success || result !=
httpspriv)      of      /subsystem=undertow/server=default-
server/https-listener=httpspriv:read-attribute(name=socket-
binding)     /subsystem=undertow/server=default-server/https-
listener=httpspriv:write-attribute(name=socket-
binding,value=httpspriv) end-if if (outcome != success ||
result           !=           "EjbcaSSLRealm")           of
/subsystem=undertow/server=default-server/https-
listener=httpspriv:read-attribute(name=security-realm)
/subsystem=undertow/server=default-server/https-
listener=httpspriv:write-attribute(name=security-
realm,value="EjbcaSSLRealm") end-if if (outcome != success
||          result          !=          REQUIRED)          of
/subsystem=undertow/server=default-server/https-
listener=httpspriv:read-attribute(name=verify-client)
/subsystem=undertow/server=default-server/https-
listener=httpspriv:write-attribute(name=verify-
client,value=REQUIRED) end-if if (outcome != success ||
result   !=   2048)   of   /subsystem=undertow/server=default-
server/https-listener=httpspriv:read-attribute(name=max-
parameters) /subsystem=undertow/server=default-server/https-
listener=httpspriv:write-attribute(name=max-parameters,
value=2048) end-if # httpspub if (outcome != success) of
/subsystem=undertow/server=default-server/https-
listener=httpspub:read-resource
/subsystem=undertow/server=default-server/https-
listener=httpspub:add(socket-binding=httpspub,security-
realm="EjbcaSSLRealm",max-parameters=2048)      end-if      if
(outcome   !=   success   ||   result   !=   httpspub)   of
/subsystem=undertow/server=default-server/https-
listener=httpspub:read-attribute(name=socket-binding)
/subsystem=undertow/server=default-server/https-
listener=httpspub:write-attribute(name=socket-binding,
value=httpspub) end-if  if (outcome != success || result !=
"EjbcaSSLRealm")   of   /subsystem=undertow/server=default-
```

```
server/https-listener=httpspub:read-attribute(name=security-
realm)       /subsystem=undertow/server=default-server/https-
listener=httpspub:write-attribute(name=security-realm,
value="EjbcaSSLRealm") end-if   if (outcome != success ||
result   !=   2048)   of   /subsystem=undertow/server=default-
server/https-listener=httpspub:read-attribute(name=max-
parameters) /subsystem=undertow/server=default-server/https-
listener=httpspub:write-attribute(name=max-parameters,
value=2048) end-if  ### Update system properties if (outcome
!=             success)             of             /system-
property=org.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_S
LASH:read-resource                                    /system-
property=org.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_S
LASH:add(value=true)  end-if  if  (outcome  !=  success)  of
/system-
property=org.apache.catalina.connector.CoyoteAdapter.ALLOW_B
ACKSLASH:read-resource                                /system-
property=org.apache.catalina.connector.CoyoteAdapter.ALLOW_B
ACKSLASH:add(value=true) end-if  if (outcome != success) of
/system-
property=org.apache.catalina.connector.URI_ENCODING:read-
resource                                              /system-
property=org.apache.catalina.connector.URI_ENCODING:add(valu
e="UTF-8")  end-if   if  (outcome  !=  success)  of  /system-
property=org.apache.catalina.connector.USE_BODY_ENCODING_FOR
_QUERY_STRING:read-resource                           /system-
property=org.apache.catalina.connector.USE_BODY_ENCODING_FOR
_QUERY_STRING:add(value=true) end-if  ### Update webservices
if (outcome != success || result != jbossws.undefined.host)
of     /subsystem=webservices:read-attribute(name=wsdl-host)
/subsystem=webservices:write-attribute(name=wsdl-host,
value=jbossws.undefined.host) end-if if (outcome != success
||   result   !=   true)   of   /subsystem=webservices:read-
attribute(name=modify-wsdl-address)
/subsystem=webservices:write-attribute(name=modify-wsdl-
address, value=true) end-if  ### Проверить статус сервера,
удостовериться, что нет ошибок :read-attribute(name=server-
state) ### Ожидаемый результат #    "outcome" => "success",
#    "result" => "reload-required", #    "response-headers"
=> {"process-state" => "reload-required"} ### Reload if
(outcome == success && result == reload-required) of :read-
attribute(name=server-state) :reload end-if
```

6. Дождаться завершения процесса перезагрузки (для проверки доступности сервера можно в jboss-cli выполнить команду: read-attribute(name=server-state)) и проверить конфигурацию:

```
/subsystem=undertow/server=default-server/https-
listener=httpspriv:read-resource  ### Ожидаемый результат #
```

```
"outcome" => "success", # "result" => { # ... #
"enabled" => true, # ... # "security-realm" =>
"EjbcaSSLRealm", # ... # "socket-binding" =>
"httpspriv", # ... # }
/subsystem=undertow/server=default-server/https-
listener=httpspub:read-resource ### Ожидаемый результат #
"outcome" => "success", # "result" => { # ... #
"enabled" => true, # ... # "security-realm" =>
"EjbcaSSLRealm", # ... # "socket-binding" =>
"httpspub", # ... # }
```

Первичное конфигурирование EJBCA

Все шаги выполняются только на 1-м сервере OTT

1. Создать в EJBCA новый Certificate Authority:

```
cd <ejbca_home>/bin/ ./ejbca.sh ca init PlatformCA_EC
"CN=PlatformCA_EC" soft null prime256v1 ECDSA 36500 --policy
null SHA256withECDSA
```

2. Выполнить импорт профилей EJBCA для OTT:

```
cd <ejbca_home>/bin/ ./ejbca.sh ca importprofiles
<tmp>/other/ejbca/ejbca_profiles
```

3. Создать в EJBCA роль **DevopsAdminRole**:

```
cd <ejbca_home>/bin/ ./ejbca.sh roles addrole
DevopsAdminRole ./ejbca.sh roles changerule DevopsAdminRole
/administrator/ ACCEPT ./ejbca.sh roles changerule
DevopsAdminRole /ca_functionality/create_certificate/ ACCEPT
./ejbca.sh roles changerule DevopsAdminRole
/ca/PlatformCA_EC/ ACCEPT ./ejbca.sh roles changerule
DevopsAdminRole /endentityprofilesrules/PPRBModule/ ACCEPT
./ejbca.sh roles changerule DevopsAdminRole
/ra_functionality/ ACCEPT ./ejbca.sh roles changerule
DevopsAdminRole
/endentityprofilesrules/PPRBModule/edit_end_entity/ DECLINE
./ejbca.sh roles changerule DevopsAdminRole
/endentityprofilesrules/PPRBModule/delete_end_entity/
DECLINE ./ejbca.sh roles changerule DevopsAdminRole
/endentityprofilesrules/PPRBModule/revoke_end_entity/
DECLINE
```

Обновление

Специальной процедуры обновления не предусмотрено. Выполните установку новых версий артефактов OTT согласно разделу **Установка**.

Проверка работоспособности

Проверка работоспособности осуществляется в продукте Platform V Monitor компонент MONA:

- метрика OTT Sidecar: ott-service-client.HealthCheckSuccess должна иметь значение больше или равно 1
- метрика OTT Сервер: ott-service.health.summary должна иметь значение равное 1
- метрика EJBCA: health.database.main.summary должна иметь значение равное 1

Откат

Для отката к предыдущей версии OTT необходимо удалить все развернутые в среде контейнеризации артефакты OTT, после чего выполнить установку предыдущей стабильной версии в соответствии с данным руководством (см. раздел **Установка**).

Часто встречающиеся проблемы и пути их устранения

1. Если в процессе выполнения команд в CLI EJBCA возникает сообщение *не удается соединиться с инстансом ejbca*, убедитесь в том, что модуль ejbca развернут строго под именем ejbca.ear
2. При запуске сервера OTT возникают ошибки с текстом:

```
EJBCA/OTT Database health check error!
```

```
EJBCA health check response code: 401, response body:
<html><head><title>Error</title></head><body>ERROR      :
Healthcheck request received from an non authorized IP
```

Решение: необходимо обновить приложение EJBCA.

Чек-лист валидации установки

1. OTT Sidecar стартует.
2. В компоненте Журналирование (LOGA) продукта Platform V Monitor (OPM) для OTT Sidecar видны сообщения "Токен X успешно получен с сервера".
3. Метрика EJBCA: health.database.main.summary должна иметь значение равное 1. Проверка работоспособности осуществляется в компоненте Объединенный мониторинг Unimon (MONA) продукта Platform V Monitor (OPM).

Руководство по установке компонента ЭЦП (криптосервис) (DIGS)

Системные требования

Список общих системных требований к ОС и развертыванию

Наименование	Внешняя/Внутренняя	Назначение
ОС Альт 8 СП 10.0 и выше или RedHat Enterprise Linux 3.10 и выше	Внешняя	Операционная система
Kubernetes 1.0 и выше или OpenShift 4.2 и выше	Внешняя	Среда управления контейнеризированными приложениями
OpenJDK 11	Внешняя	Инструмент разработки
КриптоПро JCP 2.0	Внешняя	Библиотека СКЗИ, криптопровайдер

Подробнее – см. документ Руководство прикладного разработчика.

Платформенные зависимости

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Audit SE	AUD	1.1	AUDT Аудит	опционально	Сервис для аудирования событий
Platform V Monitor	OPM	3.27.11	LOGA Журналирование	опционально	Сервис для хранения лог-файлов
Platform V Monitor	OPM	6.0.0-02	MONA Объединенный мониторинг Unimon	опционально	Сервис для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Synapse Service Mesh	SSM	1.12 и выше	POLM Управление политиками	опционально	Панель управления, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в среде контейнеризации Kubernetes

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Synapse Service Mesh	SSM	1.12 и выше	SVPX Сервисный прокси	опционально	Компонент Сервисный прокси из состава продукта Platform V Synapse Service Mesh (далее Synapse) предназначен для предоставления базовых интеграционных операций прикладной части интеграционного сервиса. Сервисный прокси используется для маршрутизации и обеспечения безопасности трафика между приложениями.

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Описание
Platform V Synapse Service Mesh	SSM	1.12 и выше	IGEG Граничный прокси	опционально	Граничные прокси выполняют роль управляемого маршрутизатора запросов (трафика), приходящих и исходящих из проекта одной системы. Запросы могут приходить как из других проектов Kubernetes, так и из-за пределов кластера

Sizing Guide

Минимальный размер сервера:

CPU	RAM (Гб)	HDD (Гб)
4	24	100

Нагрузка

Показатели тестов для определения производительности при указанных аппаратных характеристиках физического сервера:

Тест	tps(операций в секунду)
Тест определения максимальной производительности	42
Тест подтверждения максимальной нагрузки	36

Установка

Криптосервис разворачивается в любой среде контейнеризации, удовлетворяющей системным требованиям.

В параметре

```
namespace: <имя_проекта_в_среде_контейнеризации>
```

необходимо указать используемую среду контейнеризации.

Руководство по ручной установке

Подготовка

Создание объектов Secrets

Secrets для ЖАМ

В случае необходимости настройки SSL при подключении к сервисам ЖАМ необходимо создать *один secret в проекте с хранилищем сертификатов* и *один secret с паролем от данного хранилища*. Для всех трех интеграций должно использоваться одно хранилище в формате `jks`. Данное хранилище должно называться `mq.jks`.

1. Пример yaml-файла для создания mq.jks secret:

```
apiVersion: v1 kind: Secret metadata:   name: mq.jks
namespace: <имя_проекта_в_среде_контейнеризации> type:
Opaque data:   mq.jks: {value}
```

2. Пример yaml-файла для создания secret с паролем от mq.jks:

```
apiVersion: v1 kind: Secret metadata:   name: digs-service-
secrets   namespace: <имя_проекта_в_среде_контейнеризации>
type: Opaque data:   ssl.ose.keyStore.mq.password: {value}
```

Secret для хранения приватных ключей и сертификатов, созданных по ГОСТ

Для хранения ключей и сертификатов, выпущенных по ГОСТ, используется хранилище сертификатов типа HDImageStore, реализованное КриптоПро.

Внимание! В сервисе предусмотрено использование только одного хранилища данного типа.

Хранилище HDImageStore представляет собой директорию в файловой системе, где каждому приватному ключу соответствует своя директория с определенным набором файлов, которые создаются в результате генерации ключевой пары и выпуска сертификата. Потребитель должен предоставить свои ключи в виде некоторой директории, соответствующей хранилищу типа HDImageStore.

Для первоначальной загрузки ключей необходимо содержимое этой директории заархивировать в файл типа tar с именем **keys.tar**.

Secret для архива (keys.tar):

```
apiVersion: v1 kind: Secret metadata:     name: digs-jcp-
keys     namespace: <имя_проекта_в_среде_контейнеризации>
type: Opaque data:     keys.tar: {value}
```

Чтобы получить `{value}`, откройте файл **keys.tar** в Notepad++, выделите все строки, нажмите **Plugins → MIME Tools → Base64 Encode**. Получившуюся строку вставьте вместо `{value}`.

Внимание! При дальнейшей эксплуатации для добавления новых ключей потребителя требуется добавлять в архив keys.tar новые директории с ключами и сертификатами и пересоздавать данный secret. Данная операция потребует повторного развертывания POD.

Secret сервиса (digs-service-secrets) и потребителя (digs-consumer-secrets)

1. Пример yaml-файла digs-service-secrets:

```
apiVersion: v1 kind: Secret metadata:   name: digs-service-
secrets   namespace: <имя_проекта_в_контейнеризации> type:
Opaque data:   jcp-trust-stores-secret:
ssl.ose.keyStore.mq.password:
```

2. Пример yaml-файла digs-consumer-secrets:

```
apiVersion: v1 kind: Secret metadata:   name: digs-consumer-
secrets   namespace: type: Opaque data:   sign-keystore-
private-key-password-:
```

Secrets для хранилища доверенных сертификатов

Для хранения доверенных сертификатов, используемых при проверке подписи, выполненной в соответствии с ГОСТ, применяется хранилище сертификатов типа CertStore, реализованного КриптоПро.

Внимание! В сервисе предусмотрено использование только одного хранилища данного типа.

В зависимости от требований и решаемых задач, владелец сервиса сам определяет источник получения хранилища доверенных сертификатов. Принимается решение о том, каким конкретно удостоверяющим центрам можно доверять. Имя файла хранилища доверенных сертификатов должно быть **trustStore.keystore**. Для загрузки хранилища доверенных сертификатов требуется создать два secrets: *для файла хранилища* и *для пароля от него*.

1. Secret для хранилища доверенных сертификатов

```
apiVersion: v1 kind: Secret metadata:     name: digs-jcp-
trustore     namespace: <имя_проекта_в_среде
контейнеризации> type: Opaque data:     trustStore.keystore:
{value}
```

Чтобы получить `{value}`, откройте файл **trustStore.keystore** в Notepad++, выделите все строки, нажмите **Plugins → MIME Tools → Base64 Encode**. Получившуюся строку вставьте вместо `{value}`.

2. Secret для пароля от хранилища

```
apiVersion: v1 kind: Secret metadata:     name: digs-
service-secrets     namespace:
<имя_проекта_в_среде_контейнеризации> type: Opaque data:
jcp-trust-stores-secret: {value}
```

Чтобы получить `{value}`, напишите пароль в Notepad++, выделите все строки, нажмите **Plugins → MIME Tools → Base64 Encode**. Получившуюся строку вставьте вместо `{value}`.

Secret для лицензионных ключей КриптоПРО (Опционально)

```
apiVersion: v1 kind: Secret metadata:   name: digs-service-
secrets   namespace: <имя_проекта_в_среде_контейнеризации>
type: Opaque data:   jcplicense0:   jcplicense1:
```

Важно! Используемые для POD ресурсы должны соответствовать применяемой лицензии. Например, если применяется лицензия на право использования

службы СКЗИ КриптоПро JCP версии 2.0 на одном сервере (POD) с одним ядром, то необходимо указать не более 1000m для параметра containers.resources.limits.cpu контейнера ufs-sign-application. В противном случае возможны нарушения лицензионных соглашений.

Добавление настроек потребителя

Настройки потребителей необходимо хранить в виде secrets и configMaps. Псевдоним (alias) потребителя определяет набор настроек связанных с хранилищами ключей и сертификатов. Один потребитель может иметь несколько для работы с разными ключами и сертификатами.

Для ключей, выпущенных по ГОСТ, при добавлении нового потребителя или нового псевдонима, требуется выполнить следующие действия:

1. Добавить новый ключ в хранилище путем добавления его в **keys.tar** (необходимо пересоздать secrets).
2. Добавить yml файл в configMap **digs-consumer.conf** параметры для потребителя:

- sign.keystore.type.<alias_потребителя>=JCP_HD_IMAGE_STORE – тип хранилища (Возможно только такой тип для ГОСТ ключей).
- sign.keystore.private.key.alias.<alias_потебителя>=<alias_приватного_ключа> – псевдоним приватного ключа. Данный alias может не совпадать с alias потребителя. Потребитель при генерации ключей должен задать alias. Значение для данного параметра должен предоставить потребитель при передаче ключей.
- sign.keystore.cert.alias.<alias_потребителя>=<alias_сертификата> – псевдоним сертификата в хранилище сертификатов. Аналогично псевдониму приватного ключа.
- sign.truststore.type.<alias_потребителя>=JCP_CERT_STORE – тип хранилища доверенных сертификатов (Пока другое значения для данного параметра не предусмотрено).

Пример yml файла configMap с настройками потребителя:

```
apiVersion: v1 kind: ConfigMap metadata:     name: digs-
consumer.conf     namespace:
<имя_проекта_в_среде_контейнеризации>     data:
sign.keystore.cert.alias.<alias_потребителя>:
<alias_сертификата>
sign.keystore.private.key.alias.<alias_потребителя>:
<alias_приватного_ключа>
sign.keystore.type.<alias_потребителя>: JCP_HD_IMAGE_STORE
sign.truststore.type.<alias_потребителя>: JCP_CERT_STORE
```

3. Добавить secrets потребителя в **digs-consumer-secrets** путем добавления в поле **data** своих значений:

Пример yml файла:

```
data:     sign-keystore-private-key-password-:
```

Конфигурирование

Пример конфигурационного файла сервиса:

```
ufs:   sign:     client:       bicrypt-check-sign-type: cuks
bicrypt-sign-type: cuks       cms-check-sign-type: cuks
cms-cloud-sign-type: cuks       cms-cloud-adding-sign-type:
cuks       cms-tech-sign-type: cuks       encryption-type:
rsa       decryption-type: rsa       hash-type:
jca_cryptopro       jws-check-sign-type: rsa       raw-
check-sign-type: jca_cryptopro       smev2-check-sign-type:
jca_cryptopro       smev2-sp-sign-type: jca_cryptopro
smev2-ov-sign-type: jca_cryptopro       smev3-check-sign-
type: jca_cryptopro       smev3-sign-type: jca_cryptopro
xades-enhance-type: jca_cryptopro       xades-check-sign-
type: jca_cryptopro       transsign-check-sign-type:
infocrypt       wss-sign-type: jca_cryptopro       wss-
check-sign-type: jca_cryptopro    audit:     writer: STUB
testOnly: true
```

Настройка стендозависимых параметров сервиса

При установке сервиса с помощью **SynapseInataller** пайплайна необходимо указывать репозиторий, где хранятся стендозависимые параметры.

Для сервиса необходимо создать в этом репозитории **ENV** файл со стендозависимыми параметрами.

Настройка конфигурационных файлов для интеграции с Аудитом

Дополнительно, при необходимости, в Config map **digs-service.conf** можно задать значения для следующих стендозависимых параметров Аудита. Данные настройки валидные для компонента Аудит (AUDT) продукта Platform V Audit (AUD).

Пример заполнения параметров аудита:

```
ufs-audit.kafka.ssl.endpoint.identification.algorithm: https
ufs-audit.kafka.producer_factory.batch.size: '262144'  ufs-
audit.kafka.security.protocol: PLAINTEXT  ufs-
```

```
audit.kafka.producer_factory.delivery.timeout.ms: '120000'
ufs-sign-application.queue.auditInQueue.persistence: NON
ufs-audit.kafka.producer_factory.buffer.memory: '33554432'
ufs-audit.asyncConfig.queueSize: '1000'  ufs-
audit.kafka.producer_factory.linger.ms: '0'  ufs-sign-
application.wmqcf.auditCF.ssl.config: SSLConf  ufs-sign-
application.wmqcf.auditCF.sess.maxConnections: '300'  ufs-
audit.kafka.bootstrap.servers: '<IP-значение>'  ufs-
audit.kafka.producer_factory.compression.type: none  ufs-
sign-application.queue.auditInQueue.baseQueueName:
UFS.AUDIT.RESPONSE  ufs-sign-
application.object.mqQueue_auditInQueue: 'on'  ufs-
audit.writer: KAFKA  ufs-sign-
application.object.wmqcf_auditCF: 'on'  ufs-sign-
application.wmqcf.auditCF.qmgrSvrconnChannel: channel_ukofl
ufs-audit.kafka.producer_factory.connections.max.idle.ms:
'540000'  ufs.audit.kafka.trustStore.location:
/etc/config/ssl/mq.jks  ufs-
audit.kafka.producer_factory.receive.buffer.bytes: '32768'
ufs-audit.kafka.producer_factory.metadata.max.age.ms:
'300000'  ufs-sign-application.wmqcf.auditCF.maxConnections:
'300'  ufs-sign-application.wmqcf.auditCF.qmgrPortNumber:
'1418'  ufs-sign-application.wmqcf.auditCF.qmgrName:
manager_UKO_B2  ufs-
audit.kafka.producer_factory.max.request.size: '1048576'
ufs-audit.asyncConfig.threadPoolSize: '25'
ufs.audit.kafka.trustStore.type: JKS  ufs-
audit.kafka.producer_factory.send.buffer.bytes: '131072'
ufs-sign-application.wmqcf.auditCF.ose.ssl.enabledCiphers:
''  ufs.audit.kafka.keyStore.location:
/etc/config/ssl/mq.jks  ufs-sign-
application.wmqcf.auditCF.ssl.enabled: 'false'  ufs-
audit.kafka.producer_factory.max.block.ms: 60000'  ufs-
audit.kafka.producer_factory.request.timeout.ms: '30000'
ufs-sign-application.wmqcf.auditCF.qmgrHostname: tklis-
efs000917.vm.mos.cloud.sbrf.ru
ufs.audit.kafka.keyStore.type: JKS
```

Настройка конфигурационных файлов для интеграции с Прикладным Мониторингом

Пример возможной рекомендованной конфигурации yml-файла:

```
monitoring:     client:     channel: ${значение}
distributiveVersion: ${значение}     version: ${значение}
subsystem: ${значение}     title: значение     unit:
${значение}     enabledEndpoint: true     pathToEndpoint:
/metrics     enabledHttpServer: true     port: 8081
```

Ключевые парамеры pathToEndpoint и port, остальные могут отличаться в зависимости он настроек ФП Потребителя.

Настройка конфигурационных файлов для интеграции с Журналированием

Логи отправляются средствами сборщика FluentBit, для корректной работы которого рекомендуется настроить конфигурацию в yml-файле следующим образом:

```
[SERVICE]              Flush        1              Daemon
Off              Parsers_File /fluent-bit/etc/parsers.conf
[INPUT]              Name tail              Path
/app/logs/*.json             Mem_Buf_Limit 5MB
Skip_Long_Lines On             Parser custom          #
Plachholder'ы в секции [FILTER] подставляются значениями из
переменных среды с соответствующими наименованиями
[FILTER]             Name modify             Match *
Add cluster ${cluster}             Add namespace
${namespace}             Add service ${service}
Add pod ${pod}             Add moduleId ${moduleId}
Add moduleVersion ${moduleVersion}             Add nodeId
${nodeId}             Add zoneId ${zoneId}             Add
standId ${standId}          [OUTPUT]             Name http
Match *             # Hostname сервиса приема событий
# Например: logger-endpoint-demo-http-ci00641491-idevlg-
pprbtsloggertest.apps.test-ose.ca.sbrf.ru             Host
${loggingServiceHostname}             # Порт сервиса приема
событий 80 для http и 443 для https             Port
${loggingServicePort}             # Метод сервиса приема
событий             URI /v1/events             # Параметры
для защищеннного соединения             #tls On/Off
#tls.verify On/Off             #tls.ca_file /fluent-
bit/ssl/certname.pem             Format json
json_date_key timestamp          [OUTPUT]             Name
stdout             Match *       parsers.conf: |-
[PARSER]             Name custom             Format json
Time_Key timestamp             Time_Keep On
[PARSER]             Name loggermod_head             Format
regex             Regex ^(?<time>[^ ]* {1,2}[^ ]*)
\[(?<threadName>[^ ]*)\] \[(?<levelStr>[^ ]*)\]
\((?<loggerName>[^ ]*)\)
\[(?<callerClass>[^:]*)\:\:(?<callerMethod>[^:]*)\:(?<caller
Line>[^ ]*)\](\s+mdc:)(?<mdc>[^|]*)(\|\s+)(?<message>.*)
Time_Key time             Time_Format %Y-%m-%d %H:%M:%S,%L
Time_Keep On          [PARSER]             Name loggermod
Format regex             Regex (?m-ix)^(?<time>[^ ]* {1,2}[^
]*) \[(?<threadName>[^ ]*)\] \[(?<levelStr>[^ ]*)\]
\((?<loggerName>[^ ]*)\)
\[(?<callerClass>[^:]*)\:\:(?<callerMethod>[^:]*)\:(?<caller
```

```
Line>[^ ]*)\](\s+mdc:)(?<mdc>[^|]*)(\|\s+)(?<message>.*)
Time_Key time                Time_Format %Y-%m-%d %H:%M:%S,%L
Time_Keep On
```

Обратите внимание, что конфигурация конкретной ФП может существенно отличаться от примера.

Настройка конфигурационных файлов для сервиса

В Configmap **digs-service.conf** необходимо задать значения для следующих стендозависимых параметров:

data.ufs-sign-application.spring.active.profile

Для стендов DEV, CT необходимо указывать значение DEV. Таким образом, будет применена конфигурация на заглушках и не потребуется реальная интеграция с сервисами Аудит, Прикладной мониторинг, Журналирование.

Установка

Используемые библиотеки (Собственные):

- DevCommon.groovy Для подключения. В pipeline добавляется:

```
dev = new ru.sbt.synapse.DevCommon(this, env, job_props,
env.SkipInput)
```

- Nexus.groovy Для подключения. В pipeline добавляется:

```
nexus = new ru.sbt.synapse.Nexus(this, env, job_props,
runtime_props)
```

Параметры запуска:

Параметры среды	Обязательность	Значения	Описание
DistribName	-	String	Дистрибутив с конфигами для загрузки в среду контейнеризации. Список тянется напрямую из Nexus Public или Nexus.

Параметры среды	Обязательность	Значения	Описание
ClusterName	+	String	Кластер для установки
URL	+	String	URL кластера среды контейнеризации. Заполняется автоматом на основании ClusterName
projectsConfigGitUrlSources	+	String	Адрес репозитория стендозависимых параметров. Заполняется автоматом на основании ClusterName
projectsConfigGitBranchSources	+	String	Ветка репозитория стендозависимых параметров.

Параметры установки	Обязательность	Значения	Описание
nameOfProject	+	String	Проект в среде контейнеризации

Параметры установки	Обязательность	Значения	Описание
projectsConfigResourcesList	-	String	Список ресурсов, разделенных ',' или ';', из каталога проекта репозитория с конфигами для установки. Формат заполнения параметра: **path/file.yaml,path2/file2.yaml,path3/** Пример adapter-erib/<среда контейнеризации>/adapter-erib.yml,other/devops-test.yaml,test.yaml,folder-srv-client/,*folder-srv-client/* Можно указывать маску, например если написать */gateway*, то будут поставлены все конфиги с этой маской в пути или названии. Если пуст, будут загружены только зависимости для Template.

Параметры установки	Обязательность	Значения	Описание
deleteResources	-	String	Ресурсы для удаления, разделенных ',' или ';'. Пример ввода:**deployment.apps/bsp-egressgateway;service/bsp-egressgateway**

Управление режимами	Обязательность	Значения	Описание
needProjectConfigUpload (DEPRECATED)	+	String	Нужно ли выгружать конфиги проекта из репозитория стендозависимых параметров **projectsConfigGitUrlSources** (Содержимое поставки пункт Стендозависимые конфиги)
needProjectEnvUpload (DEPRECATED)	+	String	Требуется ли применять **env** файлы из репозитория

Управление режимами	Обязательность	Значения	Описание
needUploadSecret	+	true/false	Требуется ли загрузка secrets.Для расшифровки должен быть задан параметр **AnsibleVaultCredentialsId**, содержащий файл с паролем,либо использован параметр **needInputVault**
needCanaryUpload	-	true/false	Включение режима канареечной установки. Используется для загрузки новой версии приложения при наличии старой. DevOps добавит новую версию в VS и DR, которые уже загружены в проект из старой версии. В конце пайплайна будет предоставлена возможность задать вес для новой версии
needUploadVSandDR	-	true/false	Требуется ли загружать из темплейтов **VirtualService** и **DestinationRule** при канареечном режиме установки

Управление режимами	Обязательность	Значения	Описание
devMode	-	true/false	Включение **develop-mode** для канареечного режима установки (needCanaryUpload), при котором в проекте автоматически на новую версию приложения переключится 100% нагрузки
needOCValidate	+	true/false	Добавление ключа **--validate=true** при загрузке в OS конфигов. Валидация конфигов по схеме OS
needInputOSLogin	+	true/false	Нужна ли авторизация по логину/паролю (вводится во время работы пайплайна динамически, маскируется)
needToUpdateVSandDR	-	true/false	Если вам необходимо перезалить VS и/или DR в проекте, в котором они уже есть. Опция необходима в связи с ограничениями опеншифта на перезаписывание VS и DR.

Настройки джоба:

Параметр	Обязательность	Значения	Описание
JenkinsSlaveNode	+		Метка Jenkins для выбора slave
propertiesID	+		ID Config файла добавленного в настройки пространства Jenkins
needInputVault	+	true/false	Нужно ли вводить пароль от Vault для дешифрации secrets
needPreCheck	-	true/false	Добавление ключа **--dry-run=true** при загрузке в OS конфигов. Конфиги не применятся, покажется что изменяется.
needReplaceNamespace	-	true/false	Добавление ключа **--namespace** при загрузке в OS конфигов. Позволяет выгружать конфиги из другого проекта.

Внимание! Полный список обязательных параметров смотрите в исходном коде, в блоке stage ('Проверка наличия обязательных параметров').

Настройки ConfigFile:

Параметр	Обязательность	Значения	Описание

Параметр	Обязательность	Значения	Описание
needInputVault	+	true/false	Нужно ли вводить пароль от Vault

Запустить джоб

Как запустить джоб, см. в документации Platform V DevOps Tools (DOT).

Проверка работоспособности

Проверка работоспособности аппаратной части Сервиса (HelthCheck) выполняется средствами среды контейнеризации. Статус POD должен быть Running. Проверка доступности методов потребителю выполняется путём вызова этих методов – API Потребителя.

```
startupProbe:                  failureThreshold: 3
initialDelaySeconds: 80                periodSeconds: 5
successThreshold: 1                timeoutSeconds: 10
httpGet:                   path: /digs-
application/actuator/health                    port: 8080
scheme: HTTP livenessProbe:
failureThreshold: 3                initialDelaySeconds: 80
periodSeconds: 5               successThreshold: 1
timeoutSeconds: 10               httpGet:
path: /digs-application/actuator/health/liveness
port: 8080                   scheme: HTTP
```

Автоматизированная установка сервиса с использованием Deploy Tools

Для установки сервиса с использованием компонента Deploy Tools (CDJE) программного продукта Platform V DevOps Tools (DOT) разверните сервис в инфраструктуре и запустите CD Job по инструкции, приведенной в документации Platform V DevOps Tools (DOT).

Требования к окружению:

- наличие у сервиса пространства в сервисе централизованного хранения репозиториев исходного кода и прав на создание репозиториев в нем;
- наличие у сервиса области в Jenkins, с настроенными учетными данными (логин/пароль) для авторизации под технологической учетной записью (ТУЗ), позволяющая загружать артефакты в целевой репозиторий хранилища.

Параметры настройки

Пример конфигурационного файла

```
kind: ConfigMap apiVersion: v1 metadata: name: digs-
consumer.${distrib.release.version} labels: stand: ufs-sign-
application data: digs.keystore.cert.alias.kmsTest1:
'2012256' digs.keystore.cert.alias.kmsTest2: ufssign
digs.keystore.cert.alias.kmsTest3: cpro2012
digs.keystore.cert.alias.kmsTest4: esia
digs.keystore.password.kmsTest2: qwe123
digs.keystore.path.kmsTest2: /opt/ufs-sign-
application/megacuks/megacuks.p12
digs.keystore.private.key.alias.kmsTest1: '2012256'
digs.keystore.private.key.alias.kmsTest2: ufssign
digs.keystore.private.key.alias.kmsTest3: cpro2012
digs.keystore.type.kmsTest1: JCP_HD_IMAGE_STORE
digs.keystore.type.kmsTest2: PKCS12
digs.keystore.type.kmsTest3: JCP_HD_IMAGE_STORE
digs.keystore.type.kmsTest4: JCP_CERT_STORE
digs.megacuks.common.cipherSuites:
'TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384'
digs.megacuks.common.timeout: '60000'
digs.megacuks.common.url: ${digs.cuks.CryptoService.url}
digs.megacuks.kmsTest2.calculateHash.serviceName:
CalcHash_EFSConfServ
digs.megacuks.kmsTest2.checkSignBicrypt.serviceName:
CheckBicrypt_EFSConfServ
digs.megacuks.kmsTest2.checkSignCms.serviceName:
CheckDocSign_EFSConfServ
digs.megacuks.kmsTest2.checkSignSmev2.serviceName:
CheckDocSign_EFSConfServ.SMEV
digs.megacuks.kmsTest2.checkSignSmev3.serviceName:
CheckDocSign_EFSConfServ.SMEV
digs.megacuks.kmsTest2.signBicrypt.serviceName:
SignBicrypt_EFSConfServ
digs.megacuks.kmsTest2.signCmsCloud.serviceName:
SignCloudBy_EFSConfServ
digs.megacuks.kmsTest2.signCmsTech.attachedData.serviceName:
SignCMS_EFSConfServA
digs.megacuks.kmsTest2.signCmsTech.detachedData.serviceName:
SignCMS_EFSConfServD
digs.megacuks.kmsTest2.signCmsTech.detachedHash.serviceName:
SignCMS_EFSConfServD
digs.megacuks.kmsTest2.signSmev2.serviceName:
Dsign_EFSConfServ.SMEV
digs.megacuks.kmsTest2.signSmev3.serviceName:
Dsign_EFSConfServ.SMEV digs.megacuks.kmsTest2.systemId:
EFSConfServ digs.megacuks.mock.calculateHash.serviceName:
'#%MBGN_calculateHash_MND%#'
digs.megacuks.mock.checkSignBicrypt.serviceName:
'#%MBGN_checkSignBicrypt_MND%#'
```

```
digs.megacuks.mock.checkSignCms.serviceName:
'#%MBGN_checkSignCms_MND%#'
digs.megacuks.mock.checkSignSmev2.serviceName:
'#%MBGN_checkSignSmev2_MND%#'
digs.megacuks.mock.checkSignSmev3.serviceName:
'#%checkSignSmev3_MND%#'
digs.megacuks.mock.signBicrypt.serviceName:
'#%MBGN_signBicrypt_MND%#'
digs.megacuks.mock.signCmsCloud.serviceName:
'#%MBGN_signCmsCloud_MND%#'
digs.megacuks.mock.signCmsTech.attachedData.serviceName:
'#%MBGN_signCmsTech_MND%#'
digs.megacuks.mock.signCmsTech.detachedData.serviceName:
'#%MBGN_signCmsTech_MND%#'
digs.megacuks.mock.signCmsTech.detachedHash.serviceName:
'#%MBGN_signCmsTech_MND%#'
digs.megacuks.mock.signSmev2.serviceName:
'#%MBGN_signSmev2_MND%#'
digs.megacuks.mock.signSmev3.serviceName:
'#%MBGN_signSmev3_MND%#' digs.megacuks.mock.systemId: mock
digs.truststore.path.kmsTest2: /opt/ufs-sign-
application/megacuks/megacuks.p12
digs.truststore.type.kmsTest1: JCP_CERT_STORE
digs.truststore.type.kmsTest2: PKCS12
digs.truststore.type.kmsTest3: JCP_CERT_STORE
digs.truststore.type.kmsTest4: JCP_CERT_STORE
digs.transsign.common.serverTimeout: '5000'
digs.transsign.common.serverIp: ${digs.transsign.serverIp}
digs.transsign.common.waitTimeout: '0'
digs.transsign.common.port: '18092'
```

Пример параметров для стендов

```
##############################################################
#######################################
###################### СПРАВОЧНИК ДЛЯ СТЕНДА
#####################################################
##############################################################
######################################## # -----------------
--------------------------------------------------------------
------------------- # АС ЕФС # ----------------------------
--------------------------------------------------------------
--------- # Суфикс для подключения к БД (например, "_bf1")
DB_SCHEMA_SUFFIX: "_1"  # Переменная, которая используется
для задания краткого кода блока. Более подробно описано в
реестре сегментов и блоков ЕФС (например, "BF1"): #
https://sbtatlas.sigma.sbrf.ru/wiki/pages/viewpage.action?pa
geId=90873258 BLOCK_ID: "1" # Домен ЕФС (alpha, sigma,
```

```
pcidss) domain: "cloud" # id сегмента SEGMENT_ID: "FIS" # id
ЦОД (сокращенный код, согласно Реестру сегментов и блоков
ЕФС) datacenter_name: "acod" # переменная для топиков Kafka
INNER_ZONE: "{{ SEGMENT_ID }}" # DEPRECATED тип стенда (DEV,
MMV, IFT, PSI, PROM...) STAND_TYPE: ""  #### # ------------
------------------------------------------------------------
----------------------- ## Переменные для среды ## # ------
------------------------------------------------------------
------------------------------ # Группы, описанные в
distrib.yml, должны строго быть заполнены в inventory # #
если true - будет ошибка, если false - warning # # по-
умолчанию  должно быть true # deploy_groups_strict: true   #
Максимальное время ожидания выполнения команд для
интерепретаторов Shell / Bash shell_max_wait: 1200  #
Максимальное время ожидания выполнения SQL скриптов в
секундах db_update_wait: 10000  # Таймаут, в течение
которого будет происходить периодическая проверка
доступности REST сервисов. # Смок тест может завершиться
раньше таймаута, если все REST сервисы доступны и ответ от
REST сервисов соответствует регулярным выражениям, #
заданным в параметрах searchPattern, errorPattern
smoke_timeout: 300 # ------------------------------------
------------------------------------------------------------
## Настройки проксирования/SSL nginx ##  ### NGINX SGW ###
#Путь к каталогу NGINX SGW nginx_sgw_path: "/opt/nginx-sgw"
#Путь к конфигурационным файлам nginx_sgw_conf_path: "{{
nginx_sgw_path }}/conf" #Путь к статике nginx_sgw_home_path:
"{{ nginx_sgw_path }}/static" #Путь к логам
nginx_sgw_logs_path: "{{ nginx_sgw_path }}/logs" #Путь к
каталогу с бэкапом конфигурационных файлов
nginx_sgw_backup_path: "{{ nginx_sgw_path }}/backup"
#Параметры для установки мастера-ССД на VM CORESUP-17460
was_timeout_lock: 60 was_stop_servers_timeout: 60  #Список
профилей для подключения VM к Unimon_agent для сбора метрик.
Параметры профилей заполняются в конфигах в репозитории
IstioDictionary на среде unimon_agent_vm_profiles:
"sdsmaster"  #Список профилей для ЭП. Параметры профилей
заполняются в репозитории ЭП digs_profiles: ""
```

Обновление

Для обновления Сервиса необходимо выполнить инструкцию по установке, используя дистрибутив с последней версией сервиса. Ручное удаление старой версии не требуется.

Откат

Для отката Сервиса к предыдущей версии необходимо выполнить инструкцию по установке, используя дистрибутив с нужной версией из репозитория Nexus-Public или Nexus. Ручное удаление старой версии не требуется.

Часто встречающиеся проблемы и пути их устранения

Часто встречающихся проблем не выявлено.

Чек-лист валидации установки

№	Тип	Блок	Шаг
1	Пререквизит	Подготовка namespace в среде контейнеризации	Получение доступ в namespace среды контейнеризации, Выполнение подготовки namespace (настройка Istio)
2	Пререквизит	Любое git-совместимое хранилище, рекомендуется использовать GitLab CE	Создание репозитория в любом git-совместимом хранилище для хранения проекта.
3	Пререквизит	Jenkins Job	Создание своего инстанса Jenkins Job, Выполнение проверки доступа от Jenkins Job до Nexus Public или Nexus, namespace, Настройка доступа в namespace среде контейнеризации

№	Тип	Блок	Шаг
4	Пререквизит	Подготовка конфигурации развертывания	Выполнение подготовки файловой структуры конфигурации, Заполнение файлов конфигурации, Создание secrets в репозитории конфигурации, Создание ServiceAccount для ТУЗ, скачивающего Docker-образа в namespace среды контейнеризации
5	Основные работы	Развертывание продукта	Выполнение запуска Jenkins Job: Заполнение файлов конфигурации среды контейнеризации на основе конфигурации в репозитории любом git-совместимом хранилище (рекомендуется использовать GitLab CE). Создание secrets в среде контейнеризации, Загрузка файлов конфигурации, отвечающих за старт приложения

№	Тип	Блок	Шаг
6	Пререквизит	Актуализация настроек интеграции с внешними сервисами	Выпуск/загрузка сертификатов доступа к смежным сервисам в случае использования HTTPS, Настройка интеграции с используемыми смежным сервисами

Примеры конфигурации настроек Istio для mtls

Egress

Пример файла *dsr-egress-mtls.yaml*

```
apiVersion: networking.istio.io/v1alpha3 kind:
DestinationRule metadata:   name: egress-dsr-mtls-ufs-sign-
application-${distrib.release.version} spec:   exportTo:
- .   host: ${ufs-sign-application.default.nlb_mm.host}
trafficPolicy:     loadBalancer:       simple: ROUND_ROBIN
portLevelSettings:       - port:           number: ${ufs-
sign-application.default.nlb_mm.port}         tls:
caCertificates: /etc/istio/egressgateway-ca-certs/root.crt
clientCertificate: /etc/istio/egressgateway-certs/tls.crt
mode: MUTUAL           privateKey: /etc/istio/egressgateway-
certs/tls.key
```

Пример файла *gw-egress-mtls.yaml*

```
apiVersion: networking.istio.io/v1alpha3 kind: Gateway
metadata:   name: egressgateway-gw-mtls-ufs-sign-
application-${distrib.release.version} spec:   selector:
istio: egressgateway-ufs-sign-application-
${distrib.release.version}   servers:     - hosts:         -
'*'       port:         name: int-http         number:
${ufs-sign-application.ose.platform.egress.http.port}
protocol: HTTP
```

Пример файла *se-egress-http.yaml*

```
apiVersion: networking.istio.io/v1alpha3 kind: ServiceEntry
metadata:   name: egressgateway-ufs-sign-application-mtls-
iag-${distrib.release.version} spec:   exportTo:     - .
hosts:     - ${ufs-sign-application.default.nlb_mm.host}
- ${ufs-sign-application.openshift.egress.OCSP.host}     -
${ufs-sign-
application.openshift.egress.CloudSignServer.host}
location: MESH_EXTERNAL   ports:     - name: http1
number: ${ufs-sign-
application.ose.platform.egress.http.port}       protocol:
HTTP     - name: https1       number: ${ufs-sign-
application.default.nlb_mm.port}       protocol: HTTPS     -
name: OCSP       number: ${ufs-sign-
application.openshift.egress.OCSP.port}       protocol:
${ufs-sign-application.openshift.egress.OCSP.protocol}     -
name: CloudSignServer       number: ${ufs-sign-
application.openshift.egress.CloudSignServer.port}
protocol: ${ufs-sign-
application.openshift.egress.CloudSignServer.protocol}
resolution: DNS
```

Пример файла *se-security-tcp.yaml*

```
apiVersion: networking.istio.io/v1alpha3 kind: ServiceEntry
metadata:   name: egress-se-ufs-sign-application-
${distrib.release.version}   spec:   exportTo:     - .
hosts:   location: MESH_EXTERNAL   ports:     - name:
FPSUPort       number: ${ufs-sign-
application.openshift.egress.FPSU.port}       protocol:
${ufs-sign-application.openshift.egress.FPSU.protocol}     -
name: auditCF       number: ${ufs-sign-
application.wmqcf.auditCF.qmgrPortNumber}       protocol:
TCP     - name: loggerCf       number: ${ufs.logger.cf.port}
protocol: TCP     - name: kafkaBootstrap       number:
${ufs-sign-application.openshift.egress.ose.kafka.ports}
protocol: TCP
```

Пример файла *vs-egress-mtls.yaml*

```
apiVersion: networking.istio.io/v1alpha3 kind:
VirtualService metadata:   name: egressgateway-vs-mtls-ufs-
sign-application-${distrib.release.version} spec:
exportTo:     - .   gateways:     - egressgateway-gw-mtls-
ufs-sign-application-${distrib.release.version}     - mesh
hosts:     - ${ufs-sign-application.default.nlb_mm.host}
```

```
http:     - match:         - gateways:              - mesh
port: ${ufs-sign-application.ose.platform.egress.http.port}
route:         - destination:            host:
egressgateway-ufs-sign-application-
${distrib.release.version}            port:
number: ${ufs-sign-
application.ose.platform.egress.http.port}          weight:
100     - match:         - gateways:             -
egressgateway-gw-mtls-ufs-sign-application-
${distrib.release.version}           port: ${ufs-sign-
application.ose.platform.egress.http.port}       route:
- destination:             host: ${ufs-sign-
application.default.nlb_mm.host}             port:
number: ${ufs-sign-application.default.nlb_mm.port}
weight: 100
```

Ingress

Пример файла *gw-ingress-security.yaml*

```
apiVersion: networking.istio.io/v1alpha3 kind: Gateway
metadata:   name: ingress-gw-ufs-sign-application-
${distrib.release.version} spec:   selector:     istio:
ingressgateway-ufs-sign-application-
${distrib.release.version}   servers:     - hosts:         -
ingressgateway.${distrib.release.version}.${istio.ingress.ro
ute.url}       port:         name: http         number: 8080
protocol: HTTP
```

Пример файла *vs-ingress-security.yaml*

```
apiVersion: networking.istio.io/v1alpha3 kind:
VirtualService metadata:   name: ingress-vs-ufs-sign-
application-${distrib.release.version} spec:   exportTo:
- .   gateways:     - ingress-gw-ufs-sign-application-
${distrib.release.version}   hosts:     -
ingressgateway.${distrib.release.version}.${istio.ingress.ro
ute.url}   http:     - match:         - uri:
prefix: /ufs-sign-application/       route:         -
destination:             host: ufs-sign-application-
${distrib.release.version}             port:
number: 8080     - match:         - uri:             prefix:
/healthcheck/       route:         - destination:
host: ufs-sign-application-${distrib.release.version}
port:               number: 8080
```

Руководство по установке компонента Объединенный сервис авторизации (OCA) (AUTZ)

В руководстве приведены инструкции по установке компонента Объединенный сервис авторизации (OCA) (AUTZ).

Системные требования

Настройки безопасности окружения и перечень платформенных (дополнительных внешних) продуктов, используемых для установки, настройки и контроля в конечной информационной системе (далее — ИС), выбираются при разработке конечной ИС, исходя из характера обрабатываемой в ней информации и иных требований информационной безопасности (далее — ИБ), предъявляемых к ней.

Системное программное обеспечение

Ниже представлены категории системного программного обеспечения, которые обязательны для установки, настройки, контроля и функционирования компонента. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО «СберТех» (маркировка «Рекомендовано» в столбце Комментарий). Клиенту необходимо выбрать один из продуктов в каждой категории, исходя из условий использования конечной ИС.

Операционная система

Наименование	**Версия**	**Комментарий**
ОС Альт 8 СП	8 и выше	Рекомендовано
Red Hat Enterprise Linux	4.5.1 и выше	Опционально

Среда контейнеризации

Необходимое окружение для разворачивания компонента — Kubernetes версии 1.19.15 и выше или OpenShift версии 4.5.13 и выше (далее — среда контейнеризации).

Наименование	**Версия**	**Комментарий**

Наименование	Версия	Комментарий
Kubernetes	1.19.15 и выше	Рекомендовано
Red Hat OpenShift	4.5.13 и выше	Опционально

Инструмент контейнеризации

Наименование	Версия	Комментарий
Docker CE	20.10.8	Рекомендовано

Java-машина

Наименование	Версия	Комментарий
OpenJDK	8 и выше	Рекомендовано

Система управления базами данных (СУБД)

Наименование	Версия	Комментарий
PostgreSQL	11 и выше	Рекомендовано

Правообладателем АО «СберТех» также рекомендована СУБД, основанная на PostgreSQL, – Platform V Pangolin SE, см. раздел «Платформенные зависимости».

Браузер

Наименование	Версия	Комментарий
Яндекс	19.1.3.281	Рекомендовано
Google Chrome	92.0 и выше	Опционально
Safari	15.5	Опционально

Инструмент управления проектом

Наименование	Версия	Комментарий
Apache Maven	3.6.1	Рекомендовано

Система мониторинга сборки

Система для анализа численных метрик:

Наименование	Версия	Комментарий
Prometheus	2.31 и выше	Рекомендовано

Система для визуализации численных метрик (предоставленных, например, Prometheus):

Наименование	Версия	Комментарий
Grafana	2.5.0 и выше	Рекомендовано

Сервис интеграции и оркестрации микросервисов в облаке

Наименование	Версия	Комментарий
Istio	1.12 и выше	Рекомендовано

Правообладателем АО «СберТех» также рекомендован сервис интеграции и оркестрации микросервисов в облаке, снованный на Istio, – Platform V Synapse Service Mesh, см. раздел «Платформенные зависимости».

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Назначение/выполняемые функции
Platform V Audit SE	AUD	1.1	AUDT Аудит	опционально	Сервис выполняет функцию аудирования событий
Platform V Monitor	OPM	3.27.11	LOGA Журналирование	опционально	Сервис выполняет функцию хранения лог-файлов

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Назначение/выполняемые функции
Platform V Monitor	OPM	6.0.0-02	MONA Объединенный мониторинг Unimon	опционально	Сервис выполняет функцию сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения
Platform V Pangolin SE	PSQ	4.6 и выше	PSQL Platform V Pangolin	опционально	Система выполняет функцию управления базами данных, которая основана на PostgreSQL
Platform V Synapse Service Mesh	SSM	1.12 и выше	POLM Управление политиками	опционально	Панель управления с открытым исходным кодом выполняет функции для взаимодействий, мониторинга и обеспечения безопасности контейнеров в среде контейнеризации Kubernetes
Platform V Synapse Service Mesh	SSM	1.12 и выше	SVPX Сервисный прокси	опционально	Для выполнения функции доверенных взаимодействий внутри проекта (неймспейса) граничный прокси устанавливает mTLS соединение с сервисными прокси, входящими в состав

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Назначение/выполняемые функции
					подов приложений
Platform V Synapse Service Mesh	SSM	1.12 и выше	IGEG Граничный прокси	опционально	Компонент граничный прокси выполняет функцию аутентификации входящих запросов в рамках двусторонней аутентификации в рамках mTLS
Platform V IAM SE	IAM	1.3	KCSE KeyCloak.SE	опционально	IDP – провайдер, выполняющий функции управления доступа пользователей/клиентов (приложений), а также функции аутентификации/авторизации с помощью различных протоколов, таких как OAuth2.0, OIDC, SCIM
Platform V IAM SE	IAM	1.3	AUTH IAM Proxy	опционально	Выполняет функции аутентификации/авторизации запросов. Реализует Policy Enforcement Point (PEP). Взаимодействует с KCSE/AUTZ или другими провайдерами аутентификации/авторизации

Наименование продукта	Код	Версия	Код и наименование компонента	Обязательность установки	Назначение/выполняемые функции
Platform V DevOps Tools	DOT	1.2	CIJE, Build tools	опционально	Сервис представляет собой библиотеку Jenkins и предназначен для сборки дистрибутивов командами разработки
Platform V DevOps Tools	DOT	1.2	CDJE, Deploy tools	опционально	Сервис представляет собой Jenkins библиотеку, предназначенную для автоматизации развертывания и автоматической установки технологических сервисов и бизнес-приложений на тестовых и промышленных стендах (в средах DEV и ПРОМ), а также для выполнения обновлений установленных технологических сервисов и бизнес-приложений

Примечание:

- **Обязательно** — компонент или продукт необходим для функционирования сервиса.
- **Опционально** — необязательный для функционирования сервиса компонент или продукт, рекомендуется его установка, но допускается использование аналога других производителей.

Аппаратные требования

В таблице ниже указаны размерность компонента ОСА по стендам:

Стенд	Наименование позиции	Количество серверов (шт)	CPU	RAM (Гб)	Комментарий	Итого CPU	Итого RAM
DEV	АРМ администратора ufs-security-manager	1	1	3	ufs-security-manager	1	3
DEV	загрузчик ролевой модели ufs-security-importer	1	1	3	ufs-security-importer	1	3
DEV	Сервис авторизации ufs-security-bh	1	1	3	ufs-security-bh	1	3
ИФТ	АРМ администратора ufs-security-manager	2	2	4	ufs-security-manager	4	8
ИФТ	загрузчик ролевой модели ufs-security-importer	2	2	4	ufs-security-importer	4	8
ИФТ	Сервис авторизации ufs-security-bh	2	2	4	ufs-security-bh	4	8
НТ	АРМ администратора ufs-security-manager	1	4	16	ufs-security-manager	4	16

Стенд	Наименование позиции	Количество серверов (шт)	CPU	RAM (Гб)	Комментарий	Итого CPU	Итого RAM
НТ	загрузчик ролевой модели ufs-security-importer	2	4	8	ufs-security-importer	8	16
НТ	Сервис авторизации ufs-security-bh	4	4	16	ufs-security-bh	16	64
ПРОМ	АРМ администратора ufs-security-manager	4	4	16	ufs-security-manager	16	64
ПРОМ	загрузчик ролевой модели ufs-security-importer	4	4	16	ufs-security-importer	16	64
ПРОМ	Сервис авторизации ufs-security-bh	4	4	16	ufs-security-bh	16	64

Установка

Ручная установка

1. Распаковать дистрибутив компонента ОСА
2. Выполнить скрипты в командной оболочке Linux (Bash):

- скрипт ./package/conf/k8s/base/ufs-security/docker-entrypoint.sh (Сервис авторизации);
- скрипт ./package/conf/k8s/base/ufs-security-import/docker-entrypoint.sh (Импорт сервиса авторизации);
- скрипт ./package/conf/k8s/base/ufs-security-manager/docker-entrypoint.sh (АРМ администратора авторизации).

Пример выполнения через команду JAVA в командной оболочке Linux (Bash):

```
java \
-Djavax.net.ssl.keyStore="/etc/config/ssl/mq.jks" \
-Djavax.net.ssl.keyStorePassword="${mq_jksPassword}" \
-Djavax.net.ssl.trustStore="/etc/config/ssl/mq.jks" \
-Djavax.net.ssl.trustStorePassword="${mq_jksPassword}" \
-Dspas.identity.basicAuthUser="${spas_identity_basicAuthUser}" \
-Dspas.identity.basicAuthPas="${spas_identity_basicAuthPas}" \
-Dspas.rest.secretKey="${spas_rest_secretKey}" \
-Dspas.rest.platformAuthTruststorePath="/etc/config/ssl/mq.jks" \
-Dspas.rest.platformAuthTruststorePas="${mq_jksPassword}" \
-Dcom.ibm.mq.cfg.useIBMCipherMappings="false" \
-Dufsparams.config.directory="/opt/conf/ufsparams/config" \
-Dspring.profiles.active=PROM,COMMON \
-Dhealthcheck.offline=true \
-Dufs.database.type=postgres \
-Xms2000m -Xmx2200m \
--add-opens jdk.management/com.sun.management.internal=ALL-UNNAMED \
-jar /opt/ufs-security-import/ufs-security-import.jar
```

Этот документ содержит названия переменных, которые одинаково применимы для различных сред контейнеризации, указанных в системных требованиях руководства по установке.

Автоматизированная установка посредством Platform V DevOps Tools

Для установки дистрибутива:

1. В web-интерфейсе компонента Deploy tools (CDJE) Platform V DevOps Tools (DOT) и нажмите кнопку **Собрать с параметрами**.
2. На открывшейся странице с параметрами для сборки выберите:

- DISTRIB_VERSION: D-05.000.00-<последняя версия>;

- Выбрать OSE_CLUSTERS: <необходимый>;
- Выбрать версию платформы: <актуальную>;
- Выбрать сценарии запуска: CLEANUP_FP_CONFIG, MIGRATION_FP_CONF, FP_CONF_CHECK, DB_UPDATE, IMPORT_ALL_PARAMS, NGINX_DEPLOY, NGINX_MM_DEPLOY, NGINX_II_DEPLOY, WMQ_UPDATE_FP, OPENSHIFT_DEPLOY, MQ -- WMQ_UPDATE_FP, OPENSHIFT_INGRESS_EGRESS_DEPLOY;

3. Нажать кнопку **Собрать**
4. После завершения процесса сборки проверьте:

- DB_UPDATE — проверить, что создались и заполнились таблицы в БД на основании скриптов из папки package/db дистрибутива;
- NGINX_DEPLOY — проверить, что создались файлы locations, upstreams и routing в папке /opt/nginx-iag/conf на сервере nginx_ui в соответствии с файлами конфигурации nginx-iag-routing.json.j2, nginx-iag-services.json.j2, nginx-iag-nodes.json.j2. Также необходимо проверить, что разархивирован PL в папку /u01/nginx/static/support_workplace на сервере nginx_ui из папки package/pl дистрибутива;
- NGINX_II_DEPLOY — проверить, что создались файлы locations, upstreams и routing в папке /opt/nginx-iag/conf на сервере nginx_ii в соответствии с файлами конфигурации nginx-iag-routing.json.j2, nginx-iag-services.json.j2, nginx-iag-nodes.json.j2;
- NGINX_MM_DEPLOY — проверить, что создались файлы locations, upstreams и routing в папке /opt/nginx-iag/conf на сервере nginx_mm в соответствии с файлами конфигурации nginx-iag-routing.json.j2, nginx-iag-services.json.j2, nginx-iag-nodes.json.j2;
- OPENSHIFT_INGRESS_EGRESS_DEPLOY — проверить, что для проекта в среде контейнеризации создались объекты istio ingress/egress на основании файлов конфигурации: DestinationRule, ServiceEntry, Gateway, VirtualService, Deployment, Service, Route, Pod;
- OPENSHIFT_DEPLOY — проверить, что для проекта в среде контейнеризации создались объекты приложений на основании файлов конфигурации: DeploymentConfig, Service, Route, HorizontalAutoscaler, Pod;
- NGINX_EAG_UPLOAD — проверить, что в Реестре сервисов создалась конфигурация для внешнего шлюза в соответствии с файлом nginx-eag.json.j2;
- IMPORT_ALL_PARAMS — проверить, что в АРМ Администратора компонентов Управление настройками приложений PACMAN, ОСА, Журналирование, Аудит – создались соответствующие объекты в соответствии с файлами конфигурации из папки package/conf/data дистрибутива.

Справочная информация: компонент ОСА поддерживает работу в режиме интеграции с компонентом IAM Proxy продукта Platform V IAM SE (далее — IAM Proxy). Подробнее можно ознакомиться в документации IAM Proxy в разделе «Руководство разработчика».

Инструкция по развертыванию Platform V Pangolin SE, (далее – Platform V Pangolin SE)

Важная информация: в целях минимизации выдачи полномочий, дополнительного вмешательства администраторов в настройки БД и изоляции сервисов в рамках собственной схемы хранения требуется изменить 2-а параметра. Скрипты сервиса в части БД должны развертываться под владельцем схемы хранения сервиса. Для этого в файле distrib.yml указывается выражение lookup ('vars', 'переменная из passwords.conf'):

Пример:
username: "{{ lookup('vars', 'jdbc.UFS_SECURITY.user') }}"
password: "{{ lookup('vars', 'jdbc.UFS_SECURITY.password') }}"

Порядок установки:

1. Установить Platform V Pangolin SE:

Версия БД	Версия JDK	Версия
Platform V Pangolin SE	8 и выше	4.6. и выше

2. Установить версию Platform V Pangolin SE и уровень расширений:
   2.1. При инсталляции экземпляра БД следует указывать кодировку en_US.UTF-8 character set.
   2.2. Версия БД должна содержать следующие установленные расширения:

Наименование	Версия
pg_cron	1.2 и выше
pgcrypto	1.3 и выше

   2.3. Выдать гранты, как обходной вариант до реализации тикета:

```
-- Создание расширений
CREATE EXTENSION IF NOT EXISTS pgcrypto SCHEMA public;
CREATE EXTENSION IF NOT EXISTS pg_cron SCHEMA public;
-- Изменение search_path:
ALTER DATABASE <database_name> SET search_path="$user",public;
-- Грант
GRANT ALL ON ALL FUNCTIONS IN SCHEMA public TO ufs_security_<суффикс_блока>;
```

   2.4. Использовать версию Platform V Pangolin SE – 4.6 и выше:

Название	Версия
Ядро PostgreSQL	11.10
pgBouncer	1.14
patroni	1.6.4
confd	0.16.0
Расширения	см. ниже Список расширений

Агент DataProtector	10.60

Список расширений:

№	Имя	Версия
1	plperlu	1.0
2	cube	1.4
3	hstore_plperlu	1.0
4	hstore_plpythonu	1.0
5	pltcl	1.0
6	hstore_plperl	1.0
7	jsonb_plpythonu	1.0
8	plpythonu	1.0
9	ltree	1.1
10	file_fdw	1.0
11	pgrowlocks	1.2
12	insert_username	1.0
13	dblink	1.2
14	refint	1.0
15	jsonb_plperlu	1.0
16	sslinfo	1.2
17	moddatetime	1.0
18	btree_gist	1.5
19	tablefunc	1.0
20	bloom	1.0
21	uuid-ossp	1.1
22	pg_stat_statements	1.6
23	unaccent	1.1
24	pg_trgm	1.4
25	btree_gin	1.3
26	ltree_plpython2u	1.0
27	dict_xsyn	1.0
28	pg_cron	1.2
29	pg_freespacemap	1.2
30	hstore_plpython2u	1.0
31	pltclu	1.0
32	ltree_plpython3u	1.0
33	jsonb_plpython2u	1.0
34	dict_int	1.0

№	Имя	Версия
35	fuzzystrmatch	1.1
36	postgres_fdw	1.0
37	intagg	1.1
38	plperl	1.0
39	jsonb_plpython3u	1.0
40	pg_visibility	1.2
41	citext	1.5
42	pgcrypto	1.3
43	tsm_system_rows	1.0
44	tsm_system_time	1.0
45	timescaledb	2.1.0-dev
46	hstore_plpython3u	1.0
47	pg_repack	1.4.5
48	isn	1.2
49	pgstattuple	1.5
50	ltree_plpythonu	1.0
51	seg	1.3
52	autoinc	1.0
53	pg_prewarm	1.2
54	timetravel	1.0
55	tcn	1.0
56	xml2	1.1
57	pageinspect	1.7
58	amcheck	1.1
59	adminpack	2.0
60	pg_pathman	1.5
61	lo	1.1
62	pg_buffercache	1.3
63	plpgsql	1.0
64	jsonb_plperl	1.0
65	plpython2u	1.0
66	earthdistance	1.1
67	hstore	1.5
68	pgse_backup	1.1
69	intarray	1.2
70	pgaudit (включен в ядро)	1.3

3. Установить AUTZ [Объединенный сервис авторизации (OCA)]:

Код сервиса	Компонент	Схема хранения	Префикс для хранения объктов БД	Default Tablespace	Required Tablespace
AUTZ	OCA	ufs_security	sec_	ufs_ts_sec_data	ufs_ts_sec_data ufs_ts_sec_idx

schema_name.placeholder=ufs_security_<суффикс_блока>;
ts_data.placeholder=ufs_ts_sec_data;
ts_idx.placeholder=ufs_ts_sec_idx;

create user :schema_name.placeholder with encrypted password '<password>';
create schema :schema_name.placeholder;
grant connect on database <database_name> to :schema_name.placeholder;
grant all on schema :schema_name.placeholder to :schema_name.placeholder;
alter user :schema_name.placeholder VALID UNTIL 'INFINITY';
grant usage on schema :schema_name.placeholder to :schema_name.placeholder;

create tablespace :ts_data.placeholder owner :schema_name.placeholder location 'YYY';
create tablespace :ts_idx.placeholder owner :schema_name.placeholder location 'ZZZ';

Создание тестового пользователя (администратора) в базе данных

Для того чтобы появился тестовый пользователь для входа в АРМ ОСА необходимо выполнить следующие действия:

1. Выполните установку дистрибутива сервиса, чтобы пролить Liquibase-скрипты;
2. Запустите импорт ролевой модели сервиса авторизации в сервис импорта ролевой модели;
3. Выполните тестовый скрипт для создания пользователя:

Без тенанта:

```
DO
$$
  DECLARE
    userId varchar(36);
    loginP varchar := 'test_admin_s';
  BEGIN
    insert into sec_user (id, login, category_code, locked, first_name, last_name, middle_name, password,
                          pass_exp, salt, is_tech, tenant_code, user_type)
```

```
    values (random_uuid(), loginP, 'ADMIN', false, 'admin', 'admin', 'admin', '123456',
            false, '123456', false, null, 1)
    on conflict (login) do nothing;

    select id into userId from sec_user where login = loginP and tenant_code is null;

    insert into sec_user_role (id, user_id, role_id)
    select random_uuid(), userId, ro.id from sec_role ro join sec_role_sudir_roles srsr on
ro.id = srsr.role_id
    where (srsr.role_sudir_name = 'EFS_APPLICATION_ADMIN' or
srsr.role_sudir_name = 'EFS_AUTHORIZATION_ADMIN')
      and ro.tenant_code is null
    on conflict (role_id, user_id) do nothing;
  END;
$$
```

С тенантом:

```
DO
$$
  DECLARE
    userId varchar(36);
    loginP varchar := 'test_admin';
    tenantP varchar := 'test';
  BEGIN
    insert into sec_user (id, login, category_code, locked, first_name, last_name,
middle_name, password,
                          pass_exp, salt, is_tech, tenant_code, user_type)
    values (random_uuid(), loginP, 'ADMIN', false, 'admin', 'admin', 'admin', '123456',
            false, '123456', false, tenantP, 1)
    on conflict (login,tenant_code) do nothing;

    select id into userId from sec_user where login = loginP and tenant_code = tenantP;

    insert into sec_user_role (id, user_id, role_id)
    select random_uuid(), userId, ro.id from sec_role ro join sec_role_sudir_roles srsr on
ro.id = srsr.role_id
    where (srsr.role_sudir_name = 'EFS_APPLICATION_ADMIN' or
srsr.role_sudir_name = 'EFS_AUTHORIZATION_ADMIN')
      and ro.tenant_code = tenantP
    on conflict (role_id, user_id) do nothing;
  END;
$$
```

Настройка интеграции с технологическими сервисами и компонентами

Интеграция с компонентом KeyCloak.SE

Параметры компонента ОСА для интеграции с компонентом KeyCloak.SE продукта Platform V IAM SE (далее — KeyCloak.SE) начинаются со spas.rest.

Параметры настройки:

Параметр	Тип	Описание	Пример значения
secretKeyValidationEnable	boolean	Вкл/выкл проверку подписи запросов секретным ключем	true
secretKey	String	Секретный ключ для проверки подписи запросов	123456
sudirEnable	boolean	Вкл/выкл интеграция с KeyCloak.SE	
sudirLogonUrl	String	URL-для редиректа на внешнюю аутентификацию	
sudirLogoutUrl	String	URL-для редиректа на сервис logOut внешней аутентификации	
sudirConfirmUrl	String	URL-для редиректа на сервис подтверждения внешней аутентификации	
passMaxDuration	int	Время жизни пароля (дней)	40
platformAuthEnable	boolean	Включение аутентификации по JWT-токену KeyCloak.SE	
platformAuthSslRequired	boolean	Включение обязательной проверки сертификата KeyCloak.SE	true
platformAuthServerUrl	String	URL KeyCloak.SE, используется для валидации токена	
platformAuthLoginUrl	String	URL страницы входа сервера KeyCloak.SE, используется для переадресации пользователя на страницу входа	
platformAuthLogoutUrl	String	URL страницы выхода сервера KeyCloak.SE, используется для выхода пользователя	
platformAuthClientId	String	Идентификатор клиента в KeyCloak.SE для ОСА	osa
platformAuthRealmName	String	Имя используемого реалма	
platformAuthAudience	String	Audience для проверки потребителей JWT, если не задано не проверяется	
platformAuthCacheUpdateTime	long	Интервал обновление кэша в ОСА, используемого при подключении к сервису аутентификации (ключи, сертификаты) в секундах	3600
platformAuthConnectionPoolSize	int	Размер пула подключений	20
platformAuthTruststorePath	String	Путь к truststore для проверки сертификата сервиса аутентификации (путь абсолютный).Если TrustStore не указан будет использоваться системный - javax.net.ssl.trustStore	
platformAuthTruststorePas	char[]	Пароль к truststore, используемого для проверки сертификата сервиса аутентификации	

При включенном режиме интеграции:

- аутентификация пользователя осуществляется с помощью внешнего сервиса аутентификации (KeyCloak.SE) (логин передается в HTTP-заголовке IV-USER);
- добавление и изменение пользователей осуществляется администраторами во внешнем сервисе аутентификации;
- Сервис управления УЗ доступен по ссылке следующего типа: https://Доменное_имя:8443/spas/sudir/integration/GenericAccountManagement2?wsdl
- Сервис выгрузки сопутствующих данных доступен по ссылке следующего типа: https://Доменное_имя:8443/spas/sudir/integration/GenericAccountManagement2?wsdl
- Операции прикладных модулей, связанные с взаимодействием с сервисом авторизации ОСА (например, проверка прав), выполняются одинаково, независимо от режима интеграции.

Интеграция с компонентом Управление настройками PACMAN

Интеграция происходит через подключение Maven-зависимости (ru.sbrf.ufs.platform:ufs-platform-config-spring-boot-starter:jar:7.0.24.9:compile):

```
<dependency>
   <groupId>ru.sbrf.ufs.platform</groupId>
   <artifactId>ufs-platform-config-spring-boot-starter</artifactId>
   <version>${lib.versions.config}</version>
</dependency>
 private static Boolean getBooleanValue(ExtendedConfigService config,
                                        ConfigRequest configRequest, Boolean defaultValue) {
   return ExtendedConfigService.getParameters(configRequest)
           .getOne(configRequest)
           .getBoolean()
           .or(defaultValue);
}
```

Интеграция с компонентом Журналирование продукта Platform V Monitor

События системного журнала отправляются и находятся в компоненте Журналирование продукта Platform V Monitor.

Необходимо определить глобальные переменные для стенда в файле ufs-security.all.conf (/config/parameters/ufs-security.all.conf) для **Logger** ufs-security.logger.http.host=$ ufs-security.logger.http.port=$

Fluentbit images fluent-bit.sidecar.image-version=$

Интеграция с компонентом Прикладной мониторинг продукта Platform V Monitor

События мониторинга отправляются и находятся в компоненте Объединенный мониторинг Unimon продукта Platform V Monitor.

Интеграция происходит через подключение Maven-зависимости:

```
 <dependency>
```

```
    <groupId>com.sbt.opsmon.unimon</groupId>
    <artifactId>ufs-monitoring-adapter-starter</artifactId>
    <version>${lib.versions.monitoring-client}</version>
  </dependency>
```

и настройки конфигурационного файла application.yaml

```
management:
  endpoints:
    web:
      exposure:
        include: prometheus
      base-path: /
      path-mapping:
        prometheus: metrics
  server:
    port: 8081
    base-path: /monitoring
ufs:
  monitoring:
    client:
      channel: SUPPORT
      subsystem: AUTHORIZATION
      title: Объединённый сервис авторизации
      enabledEndpoint: false
      period: 5s
      channel: ${channel}
      distributiveVersion: ${distrib.version}
      version: 2.0.0
      subsystem: ${subsystem}
      title: ФП Сервис авторизации (БТС)
      thread-name-prefix: ${deploymentUnit}-thread
      unit: ${deploymentUnit}
```

Интеграция с компонентом One-Time Password (OTP) / OTT

Для настройки OTT:

1. Получите сертификаты приложения, публичный ключ OTT Service и выполните их подключение.

   Примечание: генерирование и импортирование сертификата осущевляет адамнистартор стенда.

2. Настройте ConfigMap для OTT-sidecar.

Для работы OTT-sidecar необходимо наличие в дистрибутиве ConfigMap с настройками сервиса. Создание ConfigMap выполняется на этапе установки сервиса.

Sidecar OTT

Egress sidecar

```
- name: ott-sidecar image: '${ott.ose.istio.deployment.spec.template.spec.containers.ott-sidecar.image}' env: - name: SPRING_PROFILES_ACTIVE value: PROM - name: OTT_CERTSTORE_PRIVATE_KEY_PWD valueFrom: secretKeyRef: name: istio-secret-ufs-security.${distrib.release.version} key: ufs-security.ssl.ose.istio.keyStore.egress.password - name: OTT_CERTSTORE_PWD valueFrom: secretKeyRef: name: istio-secret-ufs-security.${distrib.release.version} key: ufs-security.ssl.ose.istio.keyStore.egress.password - name: OTT_TRUST_STORE_PWD valueFrom: secretKeyRef: name: istio-secret-ufs-security.${distrib.release.version} key: ufs-security.ssl.ose.istio.keyStore.egress.password envFrom: - configMapRef: name: ott-egress-cm.${distrib.release.version} - secretRef: name: istio-secret-ufs-security.${distrib.release.version} resources: limits: cpu: ${ott.ose.istio.egress.deployment.spec.template.spec.containers.ott-sidecar.resources.limits.cpu} memory: ${ott.ose.istio.egress.deployment.spec.template.spec.containers.ott-sidecar.resources.limits.memory} requests: cpu: ${ott.ose.istio.egress.deployment.spec.template.spec.containers.ott-sidecar.resources.requests.cpu} memory: ${ott.ose.istio.egress.deployment.spec.template.spec.containers.ott-sidecar.resources.requests.memory} volumeMounts: - name: ott-certs readOnly: true mountPath: /mnt/secrets - name: ott-uds-socket mountPath: /mnt/ott-uds-socket - name: ufs-mq-jks-vol mountPath: '/etc/config/ssl/' readOnly: true terminationMessagePath: /dev/termination-log terminationMessagePolicy: File imagePullPolicy: Always
```

Ingress sidecar

```
- name: ott-sidecar image: '${ott.ose.istio.deployment.spec.template.spec.containers.ott-sidecar.image}' env: - name: SPRING_PROFILES_ACTIVE value: PROM - name: OTT_CERTSTORE_PRIVATE_KEY_PWD valueFrom: secretKeyRef: name: istio-secret-ufs-security.${distrib.release.version} key: ufs-security.ssl.ose.istio.keyStore.ingress.password - name: OTT_CERTSTORE_PWD valueFrom: secretKeyRef: name: istio-secret-ufs-security.${distrib.release.version} key: ufs-security.ssl.ose.istio.keyStore.ingress.password - name: OTT_TRUST_STORE_PWD valueFrom: secretKeyRef: name: istio-secret-ufs-security.${distrib.release.version} key: ufs-
```

```
security.ssl.ose.istio.keyStore.ingress.password envFrom: - configMapRef: name: ott-ingress-cm.${distrib.release.version} - secretRef: name: istio-secret-ufs-security.${distrib.release.version} resources: limits: cpu: ${ott.ose.istio.ingress.deployment.spec.template.spec.containers.ott-sidecar.resources.limits.cpu} memory: ${ott.ose.istio.ingress.deployment.spec.template.spec.containers.ott-sidecar.resources.limits.memory} requests: cpu: ${ott.ose.istio.ingress.deployment.spec.template.spec.containers.ott-sidecar.resources.requests.cpu} memory: ${ott.ose.istio.ingress.deployment.spec.template.spec.containers.ott-sidecar.resources.requests.memory} volumeMounts: - name: ott-certs readOnly: true mountPath: /mnt/secrets - name: ott-uds-socket mountPath: /mnt/ott-uds-socket - name: ufs-mq-jks-vol mountPath: '/etc/config/ssl/' readOnly: true terminationMessagePath: /dev/termination-log terminationMessagePolicy: File imagePullPolicy: Always
```

Конфигурация sidecar OTT

Ingress

```
apiVersion: "v1" kind: "ConfigMap" metadata: name: ott-ingress-cm.${distrib.release.version} data: OTT_OPER_MODE: validate OTT_CERTSTORE_TYPE: ${ufs-security.ott.store.type} OTT_SERVICE_CERT_ALIAS: ${ufs-security.ose.ingress.ott.service.cert.alias} OTT_SERVICE_HOSTS: ${ufs-security.ose.ingress.ott.service.hosts} OTT_AUTHZ_REALM: mmt OTT_GRPC_PORT: ${ufs-security.ose.ingress.ott.grpc.port} OTT_SERVICE_URL: ${ufs-security.ose.ingress.ott.service.url} OTT_CERTSTORE_PATH: ${ufs-security.ose.ingress.ott.certstore.path} OTT_CLIENT_CERT_ALIAS: ${ufs-security.ose.ingress.ott.client.cert.alias} OTT_TRUST_STORE_PATH: ${ufs-security.ose.ingress.ott.trustStorePath} OTT_AUTHZ_VERSION: '1.0' OTT_CLIENT_MMT-COMPATIBILITY-MODE: 'false' OTT_MODULE_ID: ${ufs-security.ose.ingress.ott.module.id} OTT_CLIENT_DEFAULT_REALM: mmt OTT_ANONYMOUS_REQUESTS_ENABLED: 'true' OTT_CLIENT_MMT_RESOURCE_ATTRID: 'urn:sbrf:names:pprb:1.0:api:interface:fullname' OTT_CLIENT_MMT_ACTION_ATTRID: 'urn:sbrf:names:pprb:1.0:action:id' OTT_APPLICATION_ATTRIBUTE_ID: 'urn:sbrf:names:pprb:1.0:module:id'
```

Filter Ingress

```
kind: EnvoyFilter apiVersion: networking.istio.io/v1alpha3 metadata: name: ott-ingress-auth-filter spec: workloadLabels: app: istio-ingressgateway-ufs-security-${distrib.release.version}-${stand_Id} projectName: ${projectName} configPatches: - applyTo: HTTP_FILTER match: context: GATEWAY listener: filterChain: filter: name: envoy.http_connection_manager
```

portNumber: 8090 patch: operation: INSERT_BEFORE value: name: envoy.ext_authz config: failure_mode_allow: false grpc_service: google_grpc: stat_prefix: ext_authz target_uri: 'unix:/mnt/ott-uds-socket/ott.socket' timeout: 2s with_request_body: allow_partial_message: false max_request_bytes: 2097152 name: envoy.ext_authz - applyTo: HTTP_FILTER match: context: GATEWAY listener: filterChain: filter: name: envoy.http_connection_manager portNumber: 12443 patch: operation: INSERT_BEFORE value: name: envoy.ext_authz config: failure_mode_allow: false grpc_service: google_grpc: stat_prefix: ext_authz target_uri: 'unix:/mnt/ott-uds-socket/ott.socket' timeout: 2s with_request_body: allow_partial_message: false max_request_bytes: 2097152 name: envoy.ext_authz

Egress

apiVersion: "v1" kind: "ConfigMap" metadata: name: ott-egress-cm.${distrib.release.version} data: OTT_OPER_MODE: sign OTT_CERTSTORE_TYPE: ${ufs-security.ott.store.type} OTT_SERVICE_CERT_ALIAS: ${ufs-security.ose.egress.ott.service.cert.alias} OTT_AUTHZ_REALM: mmt OTT_GRPC_PORT: ${ufs-security.ose.egress.ott.grpc.port} OTT_SERVICE_URL: ${ufs-security.ose.egress.ott.service.url} OTT_CERTSTORE_PATH: ${ufs-security.ose.egress.ott.certstore.path} OTT_CLIENT_CERT_ALIAS: ${ufs-security.ose.egress.ott.client.cert.alias} OTT_TRUST_STORE_PATH: ${ufs-security.ose.egress.ott.trustStorePath} OTT_AUTHZ_VERSION: '1.0' OTT_CLIENT_MMT-COMPATIBILITY-MODE: 'false' OTT_MODULE_ID: ${ufs-security.ose.egress.ott.module.id} OTT_CLIENT_DEFAULT_REALM: mmt OTT_SERVICE_HOSTS: ${ufs-security.ose.egress.ott.service.hosts} OTT_ANONYMOUS_REQUESTS_ENABLED: 'true' OTT_CLIENT_MMT_RESOURCE_ATTRID: 'urn:sbrf:names:pprb:1.0:api:interface:fullname' OTT_CLIENT_MMT_ACTION_ATTRID: 'urn:sbrf:names:pprb:1.0:action:id' OTT_APPLICATION_ATTRIBUTE_ID: 'urn:sbrf:names:pprb:1.0:module:id'

Filter Egress

kind: EnvoyFilter apiVersion: networking.istio.io/v1alpha3 metadata: name: ott-egress-auth-filter spec: workloadLabels: app: istio-egressgateway-ufs-security-${distrib.release.version}-${stand_Id} projectName: ${projectName} configPatches: - applyTo: HTTP_FILTER match: context: GATEWAY listener: filterChain: filter: name: envoy.http_connection_manager portNumber: 7071 patch: operation: INSERT_BEFORE value: name: envoy.ext_authz config: failure_mode_allow: false grpc_service: google_grpc: stat_prefix: ext_authz target_uri: 'unix:/mnt/ott-uds-socket/ott.socket' timeout: 2s with_request_body: allow_partial_message: false max_request_bytes: 2097152

```
@Override
@SuppressWarnings("unchecked")
public Object invoke(MethodInvocation methodInvocation) throws Throwable {
   Method method = methodInvocation.getMethod();
   Audit annotation = MethodInvocationUtils.findAnnotation(methodInvocation,
Audit.class);
   if (annotation != null) {
      AuditEventCreator auditEventCreator =
applicationContext.getBean(annotation.value());
      AuditEvent auditEvent = null;
      int counter = 0;
      if (method.getParameterTypes().length == 0) {
         auditEvent = auditEventCreator.createAuditEvent(null);
      }
      for (Annotation[] annotationsP : method.getParameterAnnotations()) {
         for (Annotation anAnnotationsP : annotationsP) {
            if (AuditParam.class.isInstance(anAnnotationsP)) {
               try {
                  auditEvent =
auditEventCreator.createAuditEvent(methodInvocation.getArguments()[counter]);
                  break;
               } catch (Exception e) {
                  LOGGER.error("Audit event error in {}", annotation.value(), e);
               }
            }
         }
         counter++;
      }
      try {
         Object result = methodInvocation.proceed();
         if (auditEvent != null) {
            String auditContextUUID = auditService.asyncSend(auditEvent);
            if
(START_IMPORT_EXPORT_EVENT_NAME.equals(annotation.value().getName()) ||
START_IMPORT_DICT_EVENT_NAME.equals(annotation.value().getName())) {
               //TODO разобраться зачем сюда сетается auditContextUUID
               //environment.getRequestContext().setAttribute("AuditContextUUID",
auditContextUUID);
            }
         }
        return result;
      } catch (Exception ex) {
```

```
            if (auditEvent != null) {
                setAuditEventSuccessFalse(auditEvent);
                auditService.asyncSend(auditEvent);
            }
            throw ex;
        }
    } else {
        return methodInvocation.proceed();
    }
}
```

Обновление

По умолчанию применяется стратегия Rolling. По ней pod старой версии приложения меняются на pod новой версии последовательно, одним за другим. Причем система следит, чтобы всегда оставался хотя бы один работающий pod.

При установке дистрибутива с новой версии все отсутствующие параметры, необходимые для работы приложения, будут установлены из дистрибутива. При этом старые настройки не обязательны к удалению, поскольку они не влияют на работу новой версии компонента ОСА.

Проверка работоспособности

Для проверки корректности процесса функционирования приложения необходимо перейти по ссылке url...//ufs-security-manager/rest/v2/environment/product,

Если контекст приложения поднялся правильно, то в открытом окне должна отобразится запись, содержащая подстроку следующего формата:

{"success":true,"body":{"subsystem":"AUTHORIZATION","channel":"SUPPORT","deploymentUnit":"ufs-security-manager","version":"1","distribVersion":"D-05.000.00-*","platform":"7.2.2.4-beta01","sessionId":"sJrBOuMuQSGuuG_GDIvIqoSTjjUF9UTwKOsNroNIzjz3nEHihOUTq7LaSRc7U96p","serverIp":"29.64.165.83","release":"R20.1"}}

Для проверки корректности процесса функционирования приложения необходимо в ответе обратить внимание только на параметры:
"success":true;
"subsystem":"AUTHORIZATION";
"channel":"SUPPORT";
"deploymentUnit":"<ufs-security-manager>";

"distribVersion":"D-05.000.00-<последняя версия>"

Откат

Если при установке сборки на стенд произошли ошибки и установка завершилась не успешно, необходимо произвести откат к версии, которая установлена в ПРОМе.

В Jenkins нажмите кнопку **Собрать с параметрами**, выбрав:

- DISTRIB_VERSION: D-05.000.00-<последняя версия на ПРОМе>;
- Выбрать OSE_CLUSTERS: <необходимый>;
- Выбрать версию платформы: <актуальную>;
- Выбрать сценарии запуска: CLEANUP_FP_CONFIG, MIGRATION_FP_CONF, FP_CONF_CHECK, DB_UPDATE, IMPORT_ALL_PARAMS, NGINX_DEPLOY, NGINX_MM_DEPLOY, NGINX_II_DEPLOY, WMQ_UPDATE_FP, OPENSHIFT_DEPLOY, OPENSHIFT_INGRESS_EGRESS_DEPLOY, WAS_RUN_AUTOTEST.

  Откат БД не требуется, поскольку скрипты БД поддерживают обратную совместимость в пределах одной версии. Все данные, накопленные в процессе работы в новой версии, останутся и будут доступны также и в предыдущей версии.

  Примечание:

  При установке дистрибутива с предыдущей версией значения настроек, отредактированные администратором в процессе эксплуатации, могут быть утеряны. В этом случае после установки может потребоваться настройка значений параметров.

Часто встречающиеся проблемы и пути их устранения

Узкое место	Комментарий	Предложения по оптимизации
Пул подключений к базе данных JDBC	Пул постепенно утилизируется на 100%, после чего растет время отклика.	Установить min значение в 10, max в значение 100 на всех серверах ВН для всех источников данных.
Нехватка количества портов для подключения к серверам серверной части	Ошибка nginx: 99: Cannot assign requested address	Увеличить диапазон портов: cat /proc/sys/net/ipv4/ip_local_port_r ange 15000 64000 Настроить тайм-ауты на nginx.
Утилизация очередей для подключения к компоненту журналирование	Снижение производительности из-за полной утилизации подключений к очереди Журналирование Журналирование (LOGA) продукта Platform V Monitor (OPM)	Увеличить пулл коннектов к очереди MQ Журналирование (LOGA) продукта Platform V Monitor (OPM) с 10 на 60.

Чек-лист валидации установки

После завершения процесса сборки проверьте:

- создались и заполнились таблицы в базе данных на основании скриптов из папки package/db дистрибутива;
- наличие файлов locations, upstreams и routing в папке /opt/nginx-iag/conf на сервере nginx_ui в соответствии с файлами конфигурации nginx-iag-routing.json.j2, nginx-iag-services.json.j2, nginx-iag-nodes.json.j2. Также необходимо проверить, что разархивирован PL в папку /u01/nginx/static/support_workplace на сервере nginx_ui из папки package/pl дистрибутива;
- наличие файлов locations, upstreams и routing в папке /opt/nginx-iag/conf на сервере nginx_ii в соответствии с файлами конфигурации nginx-iag-routing.json.j2, nginx-iag-services.json.j2, nginx-iag-nodes.json.j2;
- наличие файлов locations, upstreams и routing в папке /opt/nginx-iag/conf на сервере nginx_mm в соответствии с файлами конфигурации nginx-iag-routing.json.j2, nginx-iag-services.json.j2, nginx-iag-nodes.json.j2;
- для проекта в Kubernetes создались объекты istio ingress/egress на основании файлов конфигурации: DestinationRule, ServiceEntry, Gateway, VirtualService, Deployment, Service, Route, Pod;
- для проекта в Kubernetes создались объекты приложений на основании файлов конфигурации: DeploymentConfig, Service, Route, HorizontalAutoscaler, Pod;
- в Реестре сервисов создалась конфигурация для внешнего шлюза в соответствии с файлом nginx-eag.json.j2;
- в АРМ Администратора компонентов Управление настройками приложения PACMAN, ОСА, компонента Журналирование продукта Platform V Monitor, компонента Аудит продукта Platform V Audit создались соответствующие объекты в соответствии с файлами конфигурации из папки package/conf/data дистрибутива.