Руководство по установке

Продукта Platform V IAM SE (IAM)

ОГЛАВЛЕНИЕ

Руководство по установке

Перечень стороннего ПО, необходимого для установки и функционирования Platform V IAM SE (далее так же «IAM») приведен в документации на соответствующие компоненты IAM.

Состав комплекса технических средств

Развертывание на виртуальных машинах и в среде контейнеризации

Компонент	Тип приложения	Рекомендуемая конфигурация	Минимальное количество
Proxy	Proxy-сервер (nginx)	Виртуальный сервер (4-ядерный CPU архитектура x86, 24 Гбайт RAM, 150 Гбайт HDD)	2
Proxy	Сервер обработки событий аудита (syslog-ng)	Виртуальный сервер (4-ядерный CPU архитектура x86, 24 Гбайт RAM, 150 Гбайт HDD)	2
Proxy	RDS для управления Proxy	Виртуальный сервер (4-ядерный CPU архитектура x86, 24 Гбайт RAM, 150 Гбайт HDD)	2
Proxy	RDS-Server	Виртуальный сервер (4-ядерный CPU архитектура x86, 24 Гбайт RAM, 150 Гбайт HDD)	2
KCSE	Сервер аутентификации (KeyCloak, OIDC, 2FA, биометрия)	Виртуальный сервер (8-ядерный CPU архитектура x86, 32 Гбайт RAM, 200 Гбайт HDD)	2
KCSE	БД (Postgresql для KeyCloak)	Виртуальный сервер (8-ядерный CPU архитектура x86, 32 Гбайт RAM, 200 Гбайт HDD)	2
KCSE	Оркестратор для БД (patroni Postgresql)	Виртуальный сервер (2-ядерный CPU архитектура x86, 8 Гбайт RAM, 150 Гбайт HDD)	1
OCA	Среда исполнения	Среда контейнеризации. Квота (16-ядерный CPU архитектура x86, 20 Гбайт RAM)	-
OCA	СУБД	Сервер БД (4-ядерный CPU архитектура x86, 32 Гбайт RAM, 200 Гбайт HDD)	1

Необходимо установить два балансировщика нагрузки (балансировка по двум DNS-именам, на https-порт на два разных порта если используется один сервер для обоих БН).

Адреса балансировщиков нагрузки должны быть доступны из браузера пользователей UI-сервиса.

Сайзинг производится в зависимости от среднего количества одновременных обращений в секунду (приблизительно можно рассчитать Proxy-серверов - tps/800 , Серверов аутентификации - tps/200).

Примечание: приведенные данные носят рекомендательный характер.

Компонент	Тип приложения	Рекомендуемая конфигурация	Минимальное количество
Proxy	Proxy-сервер (nginx) Среда контейнеризации Квота (16-ядерный CPU архитектура x86, 20 Гбайт RAM)	-	-
KCSE	Сервер аутентификации (KeyCloak, OIDC, 2FA, биометрия)	Среда контейнеризации Квота (16-ядерный CPU архитектура x86, 20 Гбайт RAM)	-
KCSE	БД (Postgresql для KeyCloak)	Виртуальный сервер (8-ядерный CPU архитектура x86, 32 Гбайт RAM, 200 Гбайт HDD)	2
KCSE	Оркестратор для БД (patroni Postgresql)	Виртуальный сервер (2-ядерный CPU архитектура x86, 8 Гбайт RAM, 150 Гбайт HDD)	1
OCA	Среда исполнения	Среда контейнеризации Квота (16-ядерный CPU архитектура x86, 20 Гбайт RAM)	-
OCA	СУБД	Сервер БД (4-ядерный CPU архитектура x86, 32 Гбайт RAM, 200 Гбайт HDD)	1

Необходимо установить два балансировщика нагрузки (балансировка по двум DNS-именам, на https-порт на два разных порта если используется один сервер для обоих БН).

Состав дистрибутива

Дистрибутив продукта Platform V IAM SE состоит из дистрибутивов его компонент, каждому из которых соответствует определенный префикс в наименовании:

- AUTH - компонент «IAM Proxy» (далее – IAM Proxy)
- AUTZ - компонент «Объединенный сервис авторизации» (далее – OCA)
- KCSE - компонент «KeyCloak.SE» (далее – KeyCloak.SE)

Развертывание продукта Platform V IAM SE

Для развертывания необходимо распаковать дистрибутив distrib-iam_1_3_full-12-owned-distrib и установить поочередно его компоненты:

1. KeyCloak.SE
2. IAM Proxy
3. Объединенный сервис авторизации (OCA)

Для аутентификации пользователей в клиентском приложении можно использовать библиотеки Фильтра аутентификации, которые размещены в дистрибутиве в виде JAR-файлов.

Развертывание компонента IAM Proxy

1. Распаковать дистрибутив auth-[version]-distrib.zip
2. Заполнить профиль деплоя и конфигурационные файлы (пример есть в дистрибутиве)
3. Выполнить ansible playbook подготовки (deploy/ansible/platformauth-system-prepare.yml)
4. Выполнить ansible playbook установки (deploy/ansible/platformauth-deploy-playbook.yml.)

Детальное описание приведено в руководстве по установке компонента IAM Proxy.

Развертывание компонента Объединенный сервис авторизации (ОСА)

1. Распаковать 3 дистрибутива 1) autz-[version]-backend-distrib.zip 2) autz-[version]-frontend-distrib.zip 3) autz-[version]_gostech-distrib.zip
2. Выполнить скрипты в терминале

- скрипт ./package/conf/k8s/base/ufs-security/docker-entrypoint.sh (CA);
- скрипт ./package/conf/k8s/base/ufs-security-import/docker-entrypoint.sh (Import);
- скрипт ./package/conf/k8s/base/ufs-security-manager/docker-entrypoint.sh (APM авторизации).

Детальное описание приведено в руководстве по установке компонента ОСА

Развертывание компонента KeyCloak.SE

1. Распаковать дистрибутив kcse-[version]-distrib.zip
2. Выполнить скрипт в терминале с необходимыми параметрами ./bin/standalone.sh

Детальное описание приведено в руководстве по установке компонента KeyCloak.SE.

Термины и определения

Наименование	Описание	Код компонента
IAM Proxy	Компонент IAM Proxy продукта Platform V IAM SE	AUTH
KeyCloak.SE	Компонент KeyCloak.SE продукта Platform V IAM SE	KCSE
PACMAN	Компонент PACMAN продукта Platform V Backend (используется опционально)	CFGA
Аудит	Компонент Аудит продукта Platfrom V Audit SE (используется опционально)	AUDT
Объединенный сервис авторизации	Компонент Объединенный сервис авторизации (ОСА) продукта Platform V IAM SE	AUTZ
Прикладной журнал	Компонент Прикладной журнал продукта Platform V Backend (используется опционально)	APLJ

Руководство по установке компонента IAM Proxy (AUTH)

Системные требования

№	Тип ПО	Полное наименование ПО	Версия ПО
1	Операционная система	Linux (рекомендована ОС Альт 8 СП)	8
2	Среда контейнеризации	Kubernetes (K8S)	1.23
3	Java-машина	Open JDK	8 (1.8), 11
4	Сервер приложений	WildFly	15 и выше;

Для установки IAM Proxy требуется предварительно установить следующие компоненты (количество с георезервированием по двум зонам):

Тип приложения	Рекомендуемая конфигурация	Минимальная конфигурация	Минимальное количество экземпляров в PROD
Обязательные приложения			
Proxy (nginx)	VM x86 (4/24/150)	VM x86 (1/2/70)	2
RDS-клиент для управления Proxy (java, WildFly 15)	VM x86 (4/24/150)	VM x86 (1/2/70)	2
RDS-Server	VM x86 (4/24/150)	VM x86 (1/4/150)	2
Сервер обработки логов (syslog-ng)	VM x86 (4/24/150)	VM x86 (1/2/70)	2

Необязательные приложения			
PACMAN	VM x86 (4/24/150)	VM x86 (1/8/150)	2
Прикладной журнал	VM x86 (4/24/150)	VM x86 (1/8/150)	2
Аудит	VM x86 (4/24/150)	VM x86 (1/8/150)	4

Минимально возможная конфигурация:

Тип приложения	Рекомендуемая конфигурация	Минимальное количество
Proxy-сервер (nginx); Сервер обработки событий аудита (syslog-ng); RDS для управления Proxy (java, WildFly 15); Сервер аутентификации (KeyCloak, OIDC, 2FA, биометрия); БД (Postgresql для KeyCloak)	Виртуальный сервер (4-ядерный CPU архитектура x86, 16 Гбайт RAM, 50 Гбайт HDD)	1

Минимальные размеры разделов жесткого диска:

Раздел	Для разработки	Для прома
/usr	6 ГБ	6 ГБ
/opt	8 ГБ	32 ГБ
/root	8 ГБ	8 ГБ
/home	4 ГБ	4 ГБ
/tmp	1 ГБ	2 ГБ
/var	4 ГБ	8 ГБ

Так же для функционирования программного обеспечения необходимо наличие web-браузера (Яндекс.Браузер)

Подготовка к установке

Установка сервиса состоит из следующих основных этапов:

- Подготовка окружения;
- Создание или изменение профиля развертывания (или деплоя);
- Подготовка и заполнение конфигурационных артефактов;

- Развертывание джобов на сервере CI Jenkins;
- Непосредственно выполнение развертывания;
- Проверка результатов развертывания.

Подготовка окружения

Подготовка дистрибутива

Получите файл архива, содержащий дистрибутив. Текущая поставка предусматривает несколько файлов, поставки дистрибутива:

- Поставка полного дистрибутива AUTH, включая зависимые модули KCSE (далее, extra дистрибутив)
- Поставка оригинальных частей дистрибутива AUTH, включая зависимости на KCSE, (далее, owned дистрибутив)
- Поставка зависимостей дистрибутива AUTH, включая зависимости на KCSE(далее, party дистрибутив)
- Поставка компонента AUTH, не включающего зависимые модули KCSE (далее, proxy дистрибутив)

Перед установкой, необходимо получить интересующие вас дистрибутивы и разархивировать их в локальный каталог. В случае, получения party и owned дистрибутивов после получения их необходимо предварительно "склеить" специальным механизмом, использование которого подробно описано в разделе Объединение разделенного дистрибутива

Расположение и запуск плейбука

Все основные манипуляции по развертыванию закодированы в ansible-playbook, описание которого находится в файле дистрибутива ansible/platformauth-deploy-playbook.yml

Версия дистрибутива сервиса указывается при запуске ansible-playbook через переменную `deploy_platformauth_version`

Пример запуска playbook

```
ansible-playbook platformauth-deploy-playbook.yml -i
inventories/DevBarrier/hosts --extra-vars
"deploy_platformauth_version=1.0.1 tmp_clear=True"`
```

Клонирование директории развертывания

Рекомендуется использовать систему версионного контроля кода для хранения конфигурации развертывания. Если вы уже поместили подготовленные файлы конфигурации в такой системе, то выполните клонирование репозитория на локальную машину для внесения правок.

Пример команды клонирования репозитория, которую необходимо выполнить команду в терминале:

```
git clone
ssh://git@gitlab.mycompany.ru:7999/Project/CI_platformauth.
git  git checkout develop
```

Если Вы не храните конфигурационные файлы к системе версионного контроля, то используйте пример из дистрибутива.

Подготовка окружения

На этом шаге приведено описание процесса заполнения конфигурационных файлов на примере стенда под названием StendX Здесь и далее под профилем подразумевается совокупность элементов конфигурации и окружения, свойственную определенному стенду.

1. Скопировать профиль DevBarrier из дистрибутива как шаблонный для нового профиля StendX ansible/inventories/DevBarrier -> ansible/inventories/StendX

Создание или изменение профиля развертывания(или деплоя)

Внесение изменений в конфигурационные файлы профиля стенда

Необходимо внести изменения в параметры профиля StendX. В данной документации приводится описание и назначение определенных параметров. Поставляемые YAML-файл могут содержать дополнительные комментарии к заполняемым параметрам. Например, дополнительные примеры и уточнения.

Заполнение файла ansible/inventories/StendX/hosts

Здесь необходимо задать IP-адреса серверов стенда

Параметр	Описание
keycloak_geo1, keycloak_geo2	KeyCloak, модуль аутентификации (keycloak_geo1 min 1 ip)
proxy_geo1, proxy_geo2	Proxy, модуль проксирования (proxy_geo1 min 1 ip)
syslogng_geo1, syslogng_geo2	Syslog-ng, модуль событий аудита (syslogng_geo1 min 1 ip, syslogng_geo* max 1 ip)
loadbalancer_for_proxy	soft-балансировщик для Proxy, обычно необходим на стендах тестирования, и может быть заменен на hw-балансировщик (min 0 ip, max 1 ip)
loadbalancer_for_keycloak	soft-балансировщик для KeyCloak, обычно необходим на стендах тестирования, и может быть заменен на hw-балансировщик (min 0 ip, max 1 ip)
rds_server_geo1, rds_server_geo2	Route Discovery Service, модуль управления маршрутами для прокси (min 0 ip)

Заполнение файла ansible/inventories/StendX/group_vars/all/main.yml

Здесь описываются основные параметры развертываемого сервиса

Параметр	Описание	Пример
nexus_repo_url	задать репозиторий в котором размещен дистрибутив сервиса	пример для Nexus - https://nexus.my.company.ru/nexus/content/repositories/Nexus_PROD)
nexus_artifact_id	задать имя артефакта дистрибутива	для Nexus - CI01871802_PLATFORMAUTH
nexus_classifier	задать класс артефакта (не обязателен, задается значение distrib при получении дистрибутива из Nexus	
nexus_url_username	nexus_url_password - задать под кем аутентифицироваться в Nexus (если данные конфиденциальны, вынести их в файл ansible/inventories/StendX/group_vars/all/vault_main_decrypted.yml)	
ansible_user	задать пользователя, под которым будут выполняться задачи по установке на целевых серверах	
stend_type	задать тип стенда. Допустимы следующие значения: dev,ift, psi, prom, nt (default psi). dev – стенд разработки, ift – интеграционно-функциональное тестирование, psi – приемно-сдаточные испытания, prom – промышленный стенд nt – стенд нагрузочного тестирования. (default psi)	
stend_abbr	префикс для технических названий (например используется в шаблоне DNS-имён)	
stend_name	понятное название стенда для отображения в UI	
keycloak_dnsname	DNS-имя балансировщика для фронта KeyCloak. Данный параметр	

	вычисляется по шаблону (platformauth-stendx.my.company.ru, для пром platformauth.my.company.ru), и менять в нем может понадобиться только суффикс “.my.company.ru”. Для PCI DSS например это будет .pcidss.my.company.ru	
proxy_dnsname	DNS-имя балансировщика для фронта Proxy. Данный параметр вычисляется по шаблону (platform-stendx.my.company.ru, для пром platform.my.company.ru), и менять в нем может понадобиться только суффикс “.my.company.ru”. Для PCI DSS например это будет .pcidss.my.company.ru	
proxy_system_user	логин системного пользователя, под которым будут работать процессы прокси и балансировщика	
proxy_use_configuration_from_rds	Получать конфигурацию ответвлений из RouteDiscoveryService (установка rds-client рядом с прокси)	
proxy_oidc_client_id	id клиента/системы на провайдере Open ID Connect, и в случае использования Keycloak обычно равно "PlatformAuth-Proxy"	
proxy_oidc_client_secret	опциональный, пароль для аутентификации на провайдере Open ID Connect, и в случае настройки Keycloak данным деплоем этот параметр не задается и будет получен автоматически деплоем из Keycloak. [MISSING IMAGE: ,] Влияет на безопасность	
rds_server_urls	указывается список URL rds-серверов через ";" по которым отдается активная конфигурация, используется клиентская failover-балансировка	
rds_client_keyAli	алиас клиентского сертификата для	

as	rds-client из файла proxy-server.p12 (из rds_client_keyStore)	
audit2_options	параметры отправки событий в Аудит (java опции)	Пример задания audit2_options: audit2_options: - { name: "zookeeper.connecting.string", value: "10.1.2.160:2181,10.1.2.22:2181" } - { name: "kafka.producer.bootstrap.servers", value: "10.1.2.52:19092,10.1.2.56:19092" } - { name: "kafka.producer.acks", value: "1" } - { name: "buffer.maxSize", value: "1000000" } - { name: "buffer.directory", value: "/tmp/audit2" }

Заполнение раздела с описанием параметров ответвлений (junction) в proxy_jct_list

В этой группе параметров необходимо задать описание параметров маршрутизации запросов на защищаемые сервисы, на который/которые будет осуществляться проксирование. Данный раздел - это список с произвольным количеством элементов/объектов, где каждый элемент описывает параметры проксирования на конкретный back-end (указываются все back-end с front-UI, имеющиеся на стенде).

Параметр	Описание	Пример
junctionName	задаем понятное описание для отображения ответвления сервиса на тех.странице IAM Proxy	
junctionPoint	корневой контекст запросов, по которому будетопределяться принадлежность запроса к конкретной подсистеме/back-end, и вкакой back-end будет проксироваться запрос	

indexUrl	url относительно корня на back-end, по которому осуществляется основной вход в UI подсистемы. Данный параметр используется только для формирования ссылки на тех.странице IAM Proxy, и никак не влияет на функционал проксирования!	
transparent	“прозрачность” url (необязателен, default False). True - при проксировании запросы будут проходить без изменения URL. URL введенный в адресной строке браузера будет совпадать с URL который придет в HTTP-запросе на backend(на сервер приложения). False - значение из junctionPoint будет вырезано из URL запросов, и вставлено в URL-ы в контента ответов.	Пример (когда transparent=false): Запрос - Browser (https://my.com/jct-point/myindex.html) -> IAM.Proxy -> BackEnd (https://my.com/myindex.html) Ответ - Browser (html:href: https://my.com/jct-point/mysubpage.html) <- IAM.Proxy <- BackEnd (html:href: https://my.com/mysubpage.html)
https, True	используется SSL на серверах back-end (необязателен,default True)	
sslCommonName	шаблон/значение имени из CN сертификата backend-серверов, используется при соединении с back-end по HTTPS. Значение "*" - отключает проверку SSL. (необязателен, default .mycompany.ru)	
serverAddresses	список серверов (сервер:порт), на которые будут проксироваться/балансироваться запросы для данного контекста junctionPoint.	Пример: ["sbt-oabs-1144.delta.mycompany.ru:8080" , "sbt-oabs-1145.delta.mycompany.ru:8080"]
applyJctRequestFilter	указываются дополнительные опции или конфигурационные файлы, применяемые к данному junctionPoint, которые необходимо применить к данному контексту . Варианты значений опций можно посмотреть в разделе настройки через Platform V Backend.	"common/rds-ssl-sni-on.server.conf" - включаем передачу имени сервера из sslCommonName через SNI

authorizeByRoleTemplate	Шаблон (регулярное выражение) авторизации ролей пользователя. Пул ролей по которым осуществляется доступ к junction'у формируется из ID токена общими ролями (роли Realm'а. Атрибуты: id_token.realm_access.roles или id_token.roles или id_token.groups) и ролью client'а (атрибут:id_token.resource_access[текущий client].roles)	EFS_.*

Пример заполненного раздела для нескольких ответвлений:

```
proxy_jct_list:   - junctionName: Мое приложение
junctionPoint: /my-app     indexUrl: /my-app/start-page
sslCommonName: ".my.server.ru" # шаблон имени в SAN
сертификата backend-серверов     #https: False # default
True     transparent: True     serverAddresses: [
"node1.my.server.ru:443" , "node2.my.server.ru:8443" ]
applyJctRequestFilter: "common/rds-set-header-host-to-
backend.location.conf"   - junctionName: ФП СПАС
junctionPoint: /spas-dev     indexUrl: /spas/admin/index
sslCommonName: "*" # шаблон имени из CN сертификата
backend-серверов (default .sbrf.ru)     #https: False #
default True     #transparent: False # default False
serverAddresses: [ "127.0.0.1:8443" ]
```

В параметре `applyJctRequestFilter` можно задать набор из следующих значений (через запятую):

- `common/rds-set-header-host-to-backend.location.conf` - переопределение заголовка Host в сторону бэкенда с указанием первого сервера из пула балансировки (необходимо при проксировании в сторону среды контейнеризации);
- `common/rds-ssl-sni-on.server.conf` - разрешаем передачу имени сервера по SNI, fqdn-имя сервера обязательно задается в proxy_ssl_name;
- `common/set-authz-by-role-admin.location.conf` - проксировать только в случае наличия роли функционального администратора IAM Proxy;
- `common/rds-auth-in-esia.location.conf` - выбрать автоматически поставщика аутентификации с именем `esia`;
- `common/rds-opts-cloudera.location.conf` - опции под работу UI Cloudera на непрозрачном ответвлении (опционально);
- `common/rds-proxy-buffering-off.location.conf` - отключение буферизации на ответвлении, что может потребоваться например для работы SSE.

Для значений параметров применяются следующие ограничения:

- при использовании DNS-имён в serverAddresses они должны успешно разрешаться в IP на DNS-сервере, который используется на IAM Proxy (иначе конфигурация не будет применена);
- applyJctRequestFilter должен содержать пути к существующим файлам на IAM Proxy;
- при `https = true` необходимо обеспечить наличие сертификатов ЦС в TrustStore IAM Proxy;
- параметр `junctionPoint` должен быть уникален и не должен заканчиваться на "/";
- в случае наличия у всех запросов на приложение одного базового корневого контекста, рекомендуется использовать `transparent = true`;
- использовать в `applyJctRequestFilter` опции `common/rds-set-header-host-to-backend.location.conf` и\или `common/rds-ssl-sni-on.server.conf` при проксировании в `k8s\OS`.

Заполнение файла ansible/inventories/StendX/group_vars/keycloak.yml

В этом файле задаются параметры настройки KeyCloak.SE, модуль аутентификации

- `keycloak_db_address, keycloak_db_name, keycloak_db_user_name` - задать реквизиты подключения к БД KeyCloak.SE (чистая БД, при отсутствии таблиц они автоматически будут созданы). В keycloak_db_address можно указать несколько серверов (пример "10.1.1.104:5432,10.2.1.105:5432"). В keycloak_db_name можно указать доп.параметры для jdbc драйвера (пример "keycloak?targetServerType=master&prepareThreshold=0").
- `keycloak_system_user` - логин системного пользователя, под которым будет работать служба keycloak
- `keycloak_force_install - True` - принудительная полная установка KeyCloak.SE, с предварительным удалением всех старых каталогов/файлов KeyCloak.SE на целевом сервере (устанавливается в True при проблемах первоначальной установки, а после успешной установки необходимо выставить в False, чтобы при следующем обновлении случайно не потерять файлы размещенные вручную)
- `keycloak_alternative_redirect_uri` - задаем дополнительные разрешенные redirect_uri (обычно не требуется задавать, если не используются дополнительные dns-имена до прокси), https://* - разрешает перенаправить на любые хосты (на промышленную среду так делать нельзя).
- `keycloak_deploy_custom_modules` - установить доп.модули на сервер (указывается имя файла из дистрибутива по пути \keycloak\config\deployments-custom)

  Пример `["custom1.jar","custom2.ear"]`

- `keycloak_undeploy_modules` - удалить модули если они есть на сервере (указывается имя файла, может потребоваться при отключении функционала или для удаления устаревших модулей).

  Пример `["old1.jar","deprecated2.ear"]`

- `proxy_oidc_client_jwt_signed_cert` - сертификат для аутентификации на OIDC-ендпоинтах методом "Signed Jwt". Если определен этот параметр, то метод аутентификации будет выставлен в "Signed Jwt"(и будет использован сертификат, вместо client_secret). Значение задается текстом из открытой части сертификата прокси\oidc-клиента, текст между строками "---BEGIN CERTIFICAT---" и "---END CERTIFICATE---".

  Пример значения:
  "MIIKmDCCCICgAwIBAgITGAAAAAS5RQdwBbAYQAAAAAAABDANBgkqhkiG9w0BAQsF ...
  JB9bF2BQ=="

- `wsSyncSpas` - опциональный, настройки для синхронизации\получения справочников из Объединенного сервиса авторизации

- `url` - ендпоинт по которому опубликован SOAP интерфейс Объединенного сервиса авторизации
- `user, password` - логин/пароль для доступа к SOAP интерфейсу (параметр, влияющий на безопасность)
- `roleOwnerType` - куда сохранять роли полученные из Объединенного сервиса авторизации
- (CLIENT, REALM)
- `roleOwnerName` - имя существующего клиента для сохранения ролей (используется при roleOwnerType = CLIENT)

  Пример: "PlatformAuth-Proxy"

- `scheduleTrigger` - опциональный, задать периодический запуск синхронизации

  Пример: "0 0 4 ? * *" запускать по расписанию каждый день в 4 утра

- keycloak
  - `smtpServer` - задать параметры подключения к почтовому серверу по SMTP и пользователя, которые будут использоваться в KeyCloak.SE при отсылке уведомлений по эл.почте
  - `userRealmOptions.afterCreateUserSendEmail` - задать True/False. True - при создании польз. отправлять уведомление по email и одноразовый временный токен на смену пароля
  - `userRealmOptions.actionTokenGeneratedByAdminLifespan` - задать время жизни токена на смену пароля (в часах)
  - `userRealmOptions.ssoSessionIdleTimeout` - опциональный, время жизни сессии KeyCloak.SE по не активности (в минутах)
  - `userRealmOptions.displayName` , `userRealmOptions.displayNameHtml` - отображаемое название сервиса на форме входа
  - `soapApi.verifyCN` - SOAP API по управлению УЗтребует аутентификации по клиентскому сертификату, и в данном параметре указывается список допустимых CN клиентского сертификата (при mTLS) через "|", "*" - отключить проверку
  - `soapApi.rolesFilter` - фильтр по скоупу ролей, подпадающих под синхронизацию через API. Можно указать несколько префиксов ролей через запятую. Если в имени есть "/" то, считается что это роль клиента (пример, фильтр "PlatformAuth-Proxy/" подходит под любую роль клиента PlatformAuth-Proxy).

    Пример: "platformauth, EFS, PlatformAuth-Proxy/"

Заполнение файла ansible/inventories/StendX/group_vars/proxy.yml

Заполнение параметров IAM Proxy - модуль проксирования

- `proxy_dns_servers` - Dns-сервера(через пробел) актуальные для текущего стенда, используемые для резолва имен из модулей прокси
- `proxy_autoload_trusted_certificates` - выставить необходимость автоматически загружать trusted-сертификаты с HTTPS-серверов back-end при деплое. При False нужно вручную для всех серверов back-end размещать их сертификаты в папке ansible/inventories/StendX/files с префиксом “trusted_” и расширением “crt.pem” (сертификаты д.б. в формате PEM)
- `proxy_session_idletime` - время таймаута сессии прокси по не активности (в секундах) - устанавливаем значение в 30 минут
- `proxy_session_check_addr` - включаем привязку сессии к IP (True/False, default False)
- `proxy_to_backend_access_token` - опциональный, True - передавать в бэк access_token вместо id_token

- `proxy_mtls_key_file` - файл ключа сертификата прокси, для организации mTLS между прокси и проксируемой подсистемой/backend. Не обязательный.
- `proxy_mtls_cert_file` - файл сертификата прокси, для организации mTLS между прокси и проксируемой подсистемой/backend. Не обязательный
- `proxy_session_secret: "{{ vault_proxy_session_secret }}"`- опциональный, время жизни сессии по не активности в секундах
- `proxy_jct_ssl_name: ".ru"` - по умолчанию при проверке сертификата на проксируемом сервере считаем валидные CN/SAN(из сертификатов бэков) с таким доменом/host
- `proxy_session_secret: "{{ vault_proxy_session_secret }}"`- опциональный, время жизни сессии по не активности в секундах
- `proxy_session_check_addr: True` - опциональный, умолчание False, привязка сессии прокси к клиентскому IP
- `proxy_to_backend_access_token: True` - опциональный, True - передавать в бэк access_token вместо id_token
- `proxy_jct_ssl_name: ".ru"` - по умолчанию при проверке сертификата на проксируемом сервере считаем валидные CN\SAN(из сертификатов бэков) с таким доменом/host
- `proxy_to_syslog_server: "{{ syslogng_host ~ ':' ~ syslogng_from_nginx_port }}"` - удаленное логгирование событий из Nginx
- `proxy_support_isam_headers` - опциональный, default True, True - добавлять в запросы http-заголовки аналогично ISAM/WebSeal (iv-user, iv-groups, iv-remote-address). Функциональность доступна с версии (IAM Proxy) 4.2.2.

  Параметры `authz_spas` указываются если необходимо использовать функционал авторизации по URL на разрешениях которые предоставляет Объединенный сервис авторизации.

  `authz_spas_url: "https://10.116.18.10:8443/spas/rest"` - опциональный, url для вызова API Объединенного сервиса авторизации

- `authz_spas_secret: 123456 - опциональный, секрет для вызова API`
  - `authz_spas_ticket_lifetime: 3600` - опциональный, частота обновления тикета, в секундах
  - `authz_spas_ticket_failed_lifetime: 5` - опциональный, частота получения тикета, если ранее попытка была неуспешной, в секундах
  - `authz_spas_rights_lifetime: 60` - опциональный, частота обновления полномочий из Объединенного сервиса авторизации, в секундах
  - `authz_spas_rights_failed_lifetime: 5` - опциональный, частота обновления полномочий из Объединенного сервиса авторизации, если ранее попытка была неуспешной, в секундах
  - `authz_spas_ssl_verify: False` - опциональный, проверять сертификат на ендпоинте authz_spas_url

- `oidc_discovery_url: ""` - опциональный, задание URL метаданных OIDC IDP
- `oidc_logout_uri: ""` - опциональный, задание URL на который делать редирект при logout
- `oidc_use_idp_provider: "esia"` - опциональный, вход на KeyCloak.SE через заранее указанного внешнего провайдера
- `oidc_scope: "groups"` - опциональный, задание дополнительных скоупов OIDC при аутентификации
- `oidc_ssl_verify: False` - опциональный, проверять сертификат на эндпоинте OIDC

- `oidc_use_client_cert: True` - опциональный, использовать клиентский сертификат на эндпоинтах OIDC
- `oidc_host_gray: "1.1.1.1:8443"` - использовать для подключения к OIDC IDP отдельный ip:port, а не тот который в URL из $oidc_discovery_url (может потребоваться при необходимости использовании серых адресов IDP)
- `oidc_host_gray: "{{ groups['keycloak'] | first }}:{{ keycloak_https_port }}"` - использовать для подключения к OIDC IDP отдельный ip:port, а не тот который в URL из $oidc_discovery_url (может потребоваться при необходимости использовании серых адресов IDP)
- `oidc_post_logon_by_token_call_uri: "{{ keycloak_base_url }}/auth/realms/PlatformAuth/refreshEsiaSession"` - вызвать эндпоинт на IDP после восстановления по токену сессии на прокси

Заполнение файла ansible/inventories/StendX/group_vars/syslogng.yml

Необходимо задать параметры модуля событий аудита (Syslog-ng)

- `syslogng_audit2_options` - параметры для отправки событий в Аудит
- `syslogng_system_user` - логин системного пользователя, под которым будут работать процессы syslog-ng

Заполнение файла ansible/inventories/StendX/group_vars/rds_server.yml

Задать параметры модуля RDS-Server из профиля, для веток указанных ниже, передаются в конфигурацию приложения(конфигурация springboot application) **полностью** как будет задано в профиле (корневой узел rds не передается), и их состав не ограничен деплоем.

  - rds.logging
  - rds.configStore
  - rds.audit
  - rds.standin
  - rds.server
- `rds.configStore.config-store-jdbc-login` - логин к БД Platform V Configuration
- `rds.configStore.config-store-jdbc-url` - url для подключения к БД Platform V Configuration (пример `jdbc:postgresql://10.1.2.3:5432/config`)
- `rds.configStore.configStoreCryptoPas` - пароль для шифрования в Platform V Configuration (параметр, влияющий на безопасность)
- `rds.audit.kafka-servers` - сервера для подключения к kafka Аудит (пример 10.1.2.48:9092,10.1.2.179:9092)
- `rds.audit.mockMode` - true/false, включение режима заглушки Аудит
- `rds.audit.runWithoutAudit` - true/false, true - продолжить работу приложения при проблемах подключения к Аудит
- `rds.standin.cloud.client` - параметры КМ Standin для получения состояния платформенного Standin в Прикладном Журнале
- `rds.server.port` - порт на котором будет поднят сервер по https
- `rds.server.http.enable` - true/false, true - включить работу по http
- `rds.server.http.port` - порт на котором будет поднят сервер по http
- `rds.server.ssl.client-auth` - none/need, использовать аутентификацию по клиентскому сертификату при обращении к API

Примечание: Список параметров деплоя указанный выше **не исчерпывающий**, указаны параметры на которые стоит обратить внимание. Более полный список параметров можно найти в демо-профиле деплоя, и/или в документации к конкретному функционалу.

Заполнение конфиденциальных параметров, влияющих на безопасность

- Параметры задать в ansible/inventories/StendX/group_vars/all/vault_main_decrypted.yml (игнорируется git, и хранится только локально)

  Список приведен для примера, и может быть как расширен, так и уменьшен по необходимости. Все переменные `vault_*` используются исключительно на уровне yaml-файлов ansible профиля деплоя.

  - `vault_ansible_ssh_pass` - пароль пользователя которым заходим при деплое по ssh
  - `vault_keycloak_admin_password` - пароль тех.админа (используется только при деплое)
  - `vault_keycloak_db_password` - пароль к БД KeyCloak.SE (параметр, влияющий на безопасность)
  - `vault_keycloak_keystore_password` - пароль к базе сертификатов/ключей keycloak (файл keycloak-keystore.p12)
  - `vault_keycloak_smtpServer_user_password` - пароль пользователя под которым отправляются email-уведомления
  - `vault_proxy_session_secret` - секрет используемый для шифрования сессии nginx (длинна д.б. > 100 символом, и НЕ должно содержать символов `" $`)
  - `vault_rds_config_store_jdbc_pas` - пароль к БД Platform V Configuration
- Зашифровать файл с помощью ansible-vault, для этого:

  Ниже только рекомендация, но необязательный подход к шифрованию секретов. Допустимо шифровать как файлы целиком, так и переменные по отдельности, используя отдельно утилиту ansible-vault.

  - записать vault-пароль в ansible/inventories/StendX/group_vars/all/vault-password.txt (игнорируется git, и хранится только локально)
  - скопировать файлы из ansible/inventories/StendX/group_vars/all/ на ПК с установленным ansible(linux)
  - выполнить команду из каталога с файлами `chmod a+x encrypt_vaults.sh && ./encrypt_vaults.sh`
  - в результате файл vault_main_decrypted.yml будет зашифрован и записан в vault_main_encrypted.yml
  - скопировать файл vault_main_encrypted.yml в каталог профиля ansible/inventories/StendX/group_vars/all/

Заполнение файлов-секретов

- Файлы размещаются в каталоге ansible/inventories/StendX/files/ . Описание файлов:
  - `keycloak-keystore.p12` - база сертификатов/ключей KeyCloak.SE в которой содержится сертификат для организации https на KeyCloak.SE (см. ниже описание выпуска сертификатов в ЦС)
  - `proxy-server.crt.pem` - открытый ключ для организации https на прокси
  - `proxy-server.key.pem` - закрытый ключ для организации https на прокси

    - `proxy-server.p12` - из этой базы rds-client берет клиентский сертификат для mTLS при подключении по https к rds-server (обычно там тот же сертификат, что и в `proxy-server.*.pem`, но в формате PKCS12)
    - `syslogng-server.crt.pem` - открытый ключ для организации tls+syslog KeyCloak.SE ->syslog-ng
    - `syslogng-server.key.pem` - закрытый ключ для организации tls+syslog KeyCloak.SE ->syslog-ng
    - `rds-server-keystore.p12` - база сертификатов/ключей KeyCloak.SE в которой содержится сертификат для организации https на rds-server
    - `trusted_ca_*.cer` - доверенные центры сертификации в формате DER, которые выдали сертификаты нам и/или смежным серверам (это как минимум корневой и промежуточный ЦС)
    - `trusted_ca_*.crt.pem` - доверенные центры сертификации в формате PEM, которые выдали сертификаты нам и/или смежным серверам (это как минимум корневой и промежуточный ЦС)
    - `trusted_keycloak_*.cer` - доверенные центры сертификации в формате DER, для аутентификации на keycloak по клиентскому сертификату (нужен, если `keycloak_funcadmin_required_auth_cert`: True)
    - `client_trusted_chain.crt.pem` - доверенные центры сертификации в формате PEM, для аутентификации на proxy по клиентскому сертификату (нужен, если задан `proxy_mtls_front_verify_dn_regex`)
    - `keycloak/*` - файлы и подкаталоги из этого каталога будут доставлены до серверов KeyCloak.SE, в каталог `/opt/keycloak/`(переменная keycloak_home). Файлы `*.j2` будут обработаны как шаблоны, и попадут на сервера без расширения `.j2`.
    - `keycloak/post-deploy/*.sh` - файлы из этого каталога будут запущены на серверах keycloak в конце деплоя (если расширение будет `.sh.j2`, то предварительно файл будет обработан как шаблон, и `.j2` будет отброшено)
    - `proxy/*` - файлы и подкаталоги из этого каталога будут доставлены до серверов proxy, в каталог `/usr/local/openresty/`. Файлы `*.j2` будут обработаны как шаблоны, и попадут на сервера без расширения j2.
- Для конфиденциальных файлов добавляется расширение ".decrypted" (эти файлы игнорируется git, и хранятся только локально), они будут позже шифроваться.
- Зашифровать конфиденциальные файлы с помощью ansible-vault, для этого можно использовать такой подход:
    - записать vault-пароль в ansible/inventories/StendX/files/vault-password.txt (игнорируется git, и хранится только локально) (параметр, влияющий на безопасность)
    - скопировать файлы из ansible/inventories/StendX/files/ на ПК с установленным ansible(linux)
    - выполнить команду из каталога с файлами chmod a+x encrypt_vaults.sh && ./encrypt_vaults.sh
    - в результате файлы file_name.xxxxx.decrypted будут зашифрованы и записаны в file_name.xxxxx
    - скопировать зашифрованные файлы file_name.xxxxx в каталог профиля ansible/inventories/StendX/files/

Сохранение профиля в GIT

После внесения всех изменений, наш каталог с профилем сохраняется в GIT, и размещается под версионный контроль. Сохранение лучше делать в отдельной ветке, соответствующей типу среды или конкретному стенду.

```
git branch StendX git checkout StendX git commit -a -m
"init" git push origin
```

Примечание: Для хранения профилей развертывания и установки, использование версионного хранилища необязательно. Данный подход приведен как рекомендация.

Выпуск сертификатов

Для работы HTTPS по фронтовым dns-именам потребуются сертификаты, выпущенные Центром Сертификации. Так же потребуются сертификаты для межмодульного взаимодействия по TLS.
Создайте сертификаты на используемые в сервисе DNS-имена, с использованием доверенных ЦС:
platform-stendx.my.company.ru (значение из параметра proxy_dnsname, для пром platform.my.company.ru)
platformauth-stendx.my.company.ru (значение из параметра keycloak_dnsname, для пром platformauth.my.company.ru)
platformauth-syslogng-stendx.my.company.ru(обмен по TLS между Keycloak и Syslog-ng, для пром platformauth-syslogng.my.company.ru)

PS: если ранее в параметрах keycloak_dnsname, proxy_dnsname менялся домен ".my.company.ru" на какой то другой (например на ".pcidss.my.company.ru"), то здесь это тоже нужно учесть/поменять.

Для каждого сертификата нужно будет в альтернативных dns-псевдонимах(SAN) указать основное dns-имя(с доменом и без), и желательно(но не обязательно) указать IP всех конечных серверов включая балансировщик

Регистрация DNS имен в службе DNS

Доменные имена прокси и провайдера идентификаии platform-stendx.my.company.ru , platformauth-stendx.my.company.ru необходимо зарегистрировать в вашем DNS, указав для них IP-адреса балансировщиков для прокси и провайдера идентификации(keycloak) соответственно. Для корректного функционирования решения достаточно заведения в DNS записи с типом A. В

Выполнение скриптов предустановки на серверах

Все действия, требующие прав суперпользователя(root) вынесены в отдельные SH-скрипты, которые необходимо предварительно выполнить на серверах, предназначенных для развертывания.

Скрипты включены в дистрибутив (`doc\system-prepare\*`):

- `prep-deployers.sh` - выполняется на всех серверах в первую очередь. В значении скрипта "p=q1w2e3r4!" необходимо задать пароль для ssh-пользователя(deployer) под которым будет производиться деплой (значение из переменной профиля vault_ansible_ssh_pass). Так же пароль можно указать при запуске prep-deployers.sh как первый параметр. (параметр, влияющий на безопасность)
- `prep-keycloak.sh` - выполняется на всех серверах группы keycloak. В значении скрипта 'dnsname="platformauth-ift.pcidss.my.company.ru"' указать dns-имя из параметра keycloak_dnsname (параметр профиля деплоя, который задали выше по тексту).
- `prep-nginx.sh` - выполняется на всех серверах группы nginx

- `prep-rds-client.sh` - выполняется на всех серверах группы nginx, при использовании конфигурирования через rds-server и Platform V Configuration
- `prep-rds-server.sh` - выполняется на всех серверах группы rds-server, при использовании конфигурирования через rds-server и Platform V Configuration
- `prep-syslog-ng.sh` - выполняется на всех серверах группы syslogng
- `prep-lb-nginx.sh` - выполняется на всех серверах группы loadbalancer
- `unprep-all.sh` - удалить все установленные ранее компоненты сервиса, и все преднастройки
- `system-prepare.zip` - содержит все файлы данного каталога и необходимые для установки пакеты (архивируется каталог doc\system-prepare, в который добавляются файлы из proxy\platformauth-proxy.zip\packages* и syslog-ng\platformauth-syslog-ng.zip\packages*)

При установке на Alt Linux необходимо использовать скрипты с расширением .alt.sh (при отсутствии в дистрибутиве скрипта под конкретную ОС .alt.sh, используется общий .sh).

Пример установки скриптов:

```
# На сервера из списка(на все сервера сервиса) скопировать из дистрибутива файл system-prepare.zip и из каталога с архивом # выполнить следующие команды (под пользователем root): yum -y install unzip unzip system-prepare.zip -d /tmp && cd /tmp/system-prepare chmod a+x *.sh ./prep-deployers.sh  # сервера keycloak: # На данных серверах выполнить ./prep-keycloak.sh  # сервера nginx, syslog-ng: # На данных серверах выполнить ./prep-nginx.sh ./prep-rds-client.sh ./prep-syslog-ng
```

Запуск развертывания сервиса на стенд

Запуск желательно производить через задачу Jenkins, с параметрами стенда и версией дистрибутива. Пример простого запуска из командной строки

```
cd ansible ansible-playbook platformauth-deploy-playbook.yml -i inventories/StendX/hosts --vault-password-file ~/vault-password.txt --extra-vars "deploy_platformauth_version=D-01.000.03-1.0.1 tmp_clear=True"
```

Загрузка ролевой модели для Объединенного сервиса авторизации

Для того чтобы назначать пользователю роли (в частности роль функционального администратора сервиса аутентификации) необходимо добавить их в ролевую модель Объединенного сервиса авторизации, для этого:

- создать в Объединенном сервисе авторизации модуль с именем `platformauth`
- импортировать в модуль ролевую модель из файла (файл из дистрибутива `keycloak\config\roleModel_platformauth.xml`)

Полный переход на работу через сервис аутентификации

Для завершения перехода на работу пользователей только через сервис аутентификации, нужно настроить в Platform V Configuration параметр для Объединенного сервиса авторизации

`auth_proxy_url=https://platform-stendx.my.company.ru/spas-st/spas/admin/`

А при работе с платформой далее использовать ссылки, использующие прокси сервиса аутентификации, или использующие OIDC-провайдер сервиса аутентификации.

Объединение разделенного дистрибутива

Если потребуется восстановить дистрибутив из разделенного на owned и 3rd-party части, необходимо использовать джобу в jenkins.

Данная джоба загружает из Nexus-public (далее - Nexus)(\ две части (owned и 3rd-party), объединяет их, и "заливает" объединенный дистрибутив в Nexus.

Файл ее конфигурации находится в дистрибутиве по пути: **deploy/tools/import-job-to-jenkins/config_Inject3rdParty_Job.xml**

Параметры джобы:

Параметр	Описание
NEXUS_DOWNLOAD_CREDS	Credentials для скачивания разделенного дистрибутив
OWNED_PART_DOWNLOAD_URL	полный URL к owned-части разделенного дистрибутива
NEXUS_UPLOAD_CREDS	Credentials для закачивания восстановленного дистрибутива
NEXUS_UPLOAD_URL	корневой урл NEXUS, куда закачиваем дистрибутив (часть до groupId)
UPLOAD_ARTIFACT_GROUP_ID	groupId восстановленного дистрибутива
UPLOAD_ARTIFACT_ID	artifactId восстановленного дистрибутива

UPLOAD_DISTRIB_VERSION	Версия восстановленного дистрибутива
UPLOAD_CLASSIFIER	классификатор восстановленного дистрибутива
CLEANUP	Очистить директорию после выполнения джобы

Описание работы джобы:

1. По переданному урл (OWNED_PART_DOWNLOAD_URL), в коде джобы за счет автоподмены подстроки "owned-distrib" на "party-distrib", и, затем "party-distrib" на "owned.pom" произойдет скачивание owned-, party- и pom-файлов.
2. Запустится процесс слияния owned-части и party-части дистрибутива средствами утилиты inject3rdParty. Далее запустится процесс дополнительного слияния owned-части и party-части дистрибутива с помощью отдельного sh-скрипта.
3. Деплой восстановленного дистрибутива по указанному пути.

Параметры влияющие на безопасность отмечены одноименным названием "(параметры, влияющие на безопасность)"

Обновление

Миграция с предыдущих версий

В случае если руководство по миграции отсутствует, то необходимо повторно запустить развертывание с указанием более новой версии. В случае отсутствия информации по версии, считается что дополнительных действий для перехода на эту версию не требуется.

Миграция на версии 4.1.* с версий 4.0.1 и 4.0.5

Миграция компонент KeyCloak.SE

Миграция данных БД KeyCloak.SE не выполняется. Последовательность действий:

- Выполнить резервное копирование БД
- Выполнить резервное каталога с текущей версии KeyCloak.SE на сервере
- Выполнить скрипт подготовки KeyCloak.SE из дистрибутива (prep-keycloak.sh)
- Выполнить проверку и убедиться, что по умолчанию на сервере используется OpenJDK 11, с помощью команды:

```
java -version
```

- добавить\заменить в профиле деплоя параметры (в keycloak.yml):

```
keycloak_home: "/opt/keycloak" keycloak_extract_distrib:
"{{ keycloak_home }}" # default "/opt" для старой версии и
ванильного дистрибутива keycloak keycloak_force_install:
True # принудительная полная установка keycloak, с
предварительным удалением всех старых каталогов\файлов
keycloak_deploy_run_sync_users: False # запуск
синхронизации пользователей при деплое (запуск будет при
наличии изменений в конфигурации)
keycloak_deploy_run_sync_data: True # запуск синхронизации
справочников при деплое (запуск будет при наличии изменений
в конфигурации) keycloak_events_to_syslog: True # отправка
событий аудита в syslog-ng keycloak_events_to_audit2: False
# отправка событий аудита в Аудит
keycloak_use_policy_unique_characters: True # использовать
парольную политику "новый пароль должен отличаться от
старого на X символов" keycloak_use_esia: True #
использовать аутентификацию в ЕСИА keycloak_use_scim: False
# публиковать SCIM API keycloak_metrics_enabled: True #
публиковать метрики prometheus на https://<keycloak-
host>:9993/metrics , https://<keycloak-
host>/auth/realms/<realm>/metrics
keycloak_undeploy_modules: [] #deploy_keycloak_version:
"16.0.0" # Закоментировать
```

- заменить имена файлов модулей в параметре `keycloak_deploy_custom_modules`(в keycloak.yml) на новые:
  - platformauth-keycloak-identity-adapters.war -> kcse-keycloak-identity-adapters.jar
  - platformauth-esia-idp.jar -> kcse-esia-idp.jar
  - platformauth-crypto-pro.war -> kcse-crypto-pro.jar
  - platformauth-extended-idp.jar -> kcse-extended-idp.jar
  - platformauth-audit-sender.war -> kcse-audit-sender.jar
- добавить при необходимости деплой модулей через параметр keycloak_deploy_custom_modules или оставить его пустым `keycloak_deploy_custom_modules: []`. Ниже пример добавления модуля КриптоПРО

```
keycloak_deploy_custom_modules:   - kcse-crypto-pro.jar #
при использовании КриптоПРО для работы с ГОСТ-подписью
```

Обновление и удаление ответвлений

Работа с ответвлениями представляет собой процессы связанные с их созданием (описано выше) изменением и удалением. Для решения задачи по изменению ответвлений, например, в случае отсутствия интеграции с RDS-Server'ом, необходимо:

1. Изменить эти параметры в файле `ansible/inventories/StendX/group_vars/all/main.yml`, в теге `proxy_jct_list` (детальное описание см. в разделе по созданию профиля стенда).
2. Запустить установку/обновление сервиса с помощью деплоя, описанную в пункте "Установка сервиса".

Для удаления ответвления необходимо:

1. В файле `ansible/inventories/StendX/group_vars/all/main.yml`, в теге `proxy_jct_list` найти необходимое ответвление (junction) и удалить его вместе с внутренним содержимым.
2. Запустить установку/обновление сервиса с помощью деплоя, описанную в пункте "Установка сервиса".

Удаление

Чтобы удалить сервис с сервера можно использовать скрипт удаления из дистрибутива `doc/system-prepare/unprep-all.sh`, который запускается с правами root.

Проверка работоспособности

Для проверки работоспособности IAM Proxy необходимо

1. Открыть браузер и перейти по ссылке: `≥iamproxyhosts/`
2. Пройти аутентификацию у провайдера аутентификации (пароль/сертификат)

Отображение страницы браузера с настроенными эндпойнтами (звездное небо) свидетельствует об успешной аутентификации.

Откат

Вариант 1

Произвести деплой с указанием старой версии дистрибутива

Вариант 2

Для проведения процедуры отката необходимо в директории пользователя сервиса:

1. Удалить каталоги `/usr/local/openresty/nginx/conf`, `/usr/local/openresty/nginx/html`, `/usr/local/openresty/nginx/logs`, `/usr/local/openresty/site`
2. Перейти в папку `/home/nginx`
3. Выбрать файл с резервной копией (backup), и восстановить из архива удаленные выше каталоги

Важно! Не перезаписывать бинарники при распаковке резервной копии, в противном случае могут возникнуть проблемы с правами пользователей.

```
Справочно: деплой создает резервную копию 1 раз в день.
```

Чек-лист валидации установки

Для проверки установки необходимо выполнить следующие шаги:

1. Перейти в файл main.yml
2. Поменять параметры сервера в rds-server, подождать 10 сек.
3. Перейти на сервер прокси и убедится что конфигурация обновилась

Руководство по установке компонента Объединенный сервис авторизации (OCA) (AUTZ)

Системные требования

Для работы компонента Объединенный сервис авторизации (OCA) продукта Platform V IAM SE (далее — OCA) требуется:

- Операционная система: Linux (рекомендована ОС «Альт 8 СП»);
- Среда контейнеризации: рекомендован Kubernetes (K8S) - 1.23; (опционально может быть использован Red Hat OpenShift;
- Java OpenJDK – 11 и выше;
- СУБД: PostgreSQL (рекомендован Platform V Pangolin SE)

В таблице ниже указаны размерность компонента OCA по стендам:

Стенд	Наименование позиции	Кол-во 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-secutity-bh	1	1	3	ufs-secutity-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-secutity-bh	2	2	4	ufs-secutity-bh	4	8
НТ	АРМ администратора ufs-security-manager	1	4	16	ufs-security-manager	4	16
НТ	загрузчик ролевой модели ufs-security-importer	2	4	8	ufs-security-importer	8	16
НТ	Сервис авторизации ufs-secutity-bh	4	4	16	ufs-secutity-bh	16	64
ПРОМ	АРМ администратора ufs-security-manager	4	4	16	ufs-security-manager	16	64
ПРОМ	загрузчик ролевой модели ufs-security-	4	4	16	ufs-security-importer	16	64

	importer						
ПРОМ	Сервис авторизации ufs-secutity-bh	4	4	16	ufs-secutity-bh	16	64

Перечень доступных внешних продуктов для интеграционных взаимодействий:

№	Наименование ПО	Код ПО	Код компонента	Предустановка обязательна или опциональна
1	Компонент «Журналирование» продукта Platform V Monitor»	OPM	LOGA	Опционально
2	Компонент «Аудит» продукта Platform V Audit SE	AUD	AUDT	Опционально
3	Компонент «PACMAN» продукта Platform V Backend»	#BH	CFGA	Опционально
4	Компонент «Прикладной мониторинг» продукта Platform V Monitor»	OPM	MONA	Опционально
5	Компонент «One-Time Password (OTP) / OTT» - продукта Platform V Backend» (далее «OTT»)	#BH	OTTS	Опционально
6	Компонент «Сессионные данные» продукта Platform V Frontend Std»	#FS	SUSD	Опционально
7	Компонент «Стартовой менеджер» продукта Platform V Frontend Std»	#FS	SMGX	Опционально

Установка

Этот документ содержит названия переменных, которые одинаково применимы для различных сред контейнеризации, указанных в системных требованиях руководства по установке.

Для установки дистрибутива:

1. Откройте Jenkins 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 базе данных 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 Kubernetes создались объекты istio ingress/egress на основании файлов конфигурации: DestinationRule, ServiceEntry, Gateway, VirtualService, Deployment, Service, Route, Pod;
   - OPENSHIFT_DEPLOY — проверить, что для проекта в Kubernetes 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 APM Администратора компонентов PACMAN продукта Platform V Backend, компонента Стартовый менеджер продукта Platform V Frontend Std, OCA, компонента Журналирование продукта Platform V Monitor, компонента Аудит продукта Platform V Audit SE создались соответствующие объекты в соответствии с файлами конфигурации из папки package/conf/data дистрибутива.

Справочная информация: компонент OCA 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
Platform V Pangolin SE	11 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 :

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
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 [Объединённый 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 OCA для интеграции с компонентом KeyCloak.SE продукта Platform V IAM (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 OCA	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 PACMAN и продукта Platform V Backend

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 (далее - OTT)

Для настройки OTT:

1. получите сертификаты приложения, публичный ключ OTT Service и выполните их подключение.

   Примечание: генерирование и импортирование сертификата осущевляет адаминистартор 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
```

Интеграция с компонентом Аудит продукта Platform V Audit SE

```
@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();
    }
}
```

Интеграция с компонентом Стартовый менеджер продукта Platform V Frontend Std

```
[
    {
        "id": null,
        "permission": "ManagePermissions.StartManager.ViewPanel",
        "channelName": "SUPPORT",
        "channel": {
            "id": 0,
            "name": "SUPPORT"
        },
        "name": "SM_URL_UFS_SECURITY",
        "subsystem": "ФП Сервис Авторизации",
        "title": "Авторизация",
        "background": "#009F00",
        "keyWords": "авторизация;роль;роли;разрешение;разрешения;группа;группы;динамическая
группа;динамические группы;фильтр;фильтры;атрибут сессии, атрибуты сессии;",
        "keyWordSynonyms": "",
        "shortcut": "",
        "weight": 100,
```

```
    "path": "",
    "path2": "",
    "url": "security.js",
    "subSystemType": "INSTEAD",
    "supParamId": "",
    "needIdentification": false,
    "isAvailableInStandIn": true,
    "isPilotZone": false,
    "type": {
      "id": 0,
      "name": "HIGH_ORDER_OPER"
    },
    "pilotZone": false,
    "availableInStandIn": true,
    "isBlocked": false,
    "operationZone": "PILOT_AND_PROM"
  }
]
```


Обновление

При установке дистрибутива с предыдущей версией все отсутствующие параметры, необходимые для работы приложения, будут установлены из дистрибутива. При этом новые настройки не обязательны к удалению, поскольку они не влияют на работу предыдущей 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 ВН	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 ФП Журналирование	Увеличить пулл коннектов к очереди MQ ФП Журналирование 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 PACMAN продукта Platform V Backend, компонента Стартовый менеджер продукта Platform V Frontend Std, OCA, компонента Журналирование продукта Platform V Monitor, компонента Аудит продукта Platform V Audit SE создались соответствующие объекты в соответствии с файлами конфигурации из папки package/conf/data дистрибутива.

Руководство по установке компонента KeyCloak.SE (KCSE)

Системные требования

№	Тип ПО	Полное наименование ПО	Версия ПО
1	Операционная система	Linux (рекомендована ОС Альт 8 СП)	8
2	Средство контейнеризации	Kubernetes (K8S)	1.23
3	Java-машина	Open JDK	Любая актуальная
4	СУБД	PostgreSQL (рекомендован Platform V Pangolin SE)	Любая актуальная
5	Сервер приложений	WildFly	Любая актуальная

Список ПО, необходимого для развертывания компонента KeyCloak.SE продукта Platform V IAM SE на локальном компьютере

Наименование ПО	Обязательность	Что делает
Docker Desktop	Да	Выполняет запуск контейнера компонента KeyCloak.SE и Platform V Pangolin SE
Браузер (любой из перечисленных): • Яндекс Браузер 22+; • MozillaFirefox 15+; • GoogleChrome 21+; • либо любой другой идентичный браузер.	Да	Выполняет функцию визуального отображения интерфейса компонента KeyCloak.SE

Список ПО, необходимого для развертывания компонента KeyCloak.SE продукта Platform V IAM SE в среде контейнеризации с использованием Jenkins

Наименование ПО	Обязательность	Что делает
Jenkins	Да	Выполняет сборку ПО
Среда контейнеризации Kubernetes 1.23	Да	Осуществляет оркестрацию контейнеров (POD), в которых запущен компонент KeyCloak.SE и все необходимо ПО для его работы.
PostgreSQL (рекомендован Platform V Pangolin SE)	Да	СУБД
Ansible	Да	Необходим для Template-processing параметров yml-файлов среды контейнеризации

Системные требования к POD в среде контейнеризации и системе, на которой развернут продукт Platform V Pangolin SE, которую использует компонент KeyCloak.SE продукта Platform V IAM SE.

Требования к конфигурации POD прописываются интервально, т.е. указывается минимальный размер озу/цпу и максимальный. В рамках требований указан минимальный размер.

Для проверки всего функционала желательно разворачивать минимум 2 POD.

Тип стенда	DEV	ИФТ
Конфигурация в среде контейнеризации (в рамках одного POD)	-resources: limits: cpu: "2" memory: 4Gi requests: cpu: 300m memory: 2Gi	-resources: limits: cpu: "2" memory: 4Gi requests: cpu: 300m memory: 2Gi
На чем хостится БД	CPU: 8 ядер MEMORY : 16 Gi ПЗУ: 150Гигов.	CPU: 8 ядер MEMORY : 16 Gi ПЗУ: 150Гигов.

Установка

Локальная установка с помощью docker - образа

Локальная установка с помощью docker - образа применяется для тестирования функционала на локальном компьютере.

Для этого необходимы следующие условия:

1. Доступ к docker - хабу, на котором расположен дистрибутив компонента KeyCloak.SE
2. Доступ к docker - хабу, на котором расположен дистрибутив продукта Platform V Pangolin SE
3. Иметь установленный Docker на компьютере, на котором будет осуществляться поднятие контейнеров

Для того чтобы поднять docker - контейнеры с компонентом KeyCloak.SE продукта Platform V IAM SE необходимо:

1. Создать docker - compose файл, в соответствии, с необходимыми требованиями:
   1.1. Пример базового docker - compose файла:

```
version: '3'  services:   postgres:     image: postgres     volumes:
- ~/postgres-3:/var/lib/postgresql/data     environment:
POSTGRES_DB: keycloak       POSTGRES_USER: keycloak
POSTGRES_PASSWORD: password     ports:     - 5432:5432    keycloak:
image:
dzo.sw.sbc.space/sbt_dev/ci90000020_kcse_dev/keycloak.se_with_module
s:1.003.001 #Подставить ссылку на образ Keycloak.SE (будет
предоставлена при передаче дистрибутива)     container_name:
keycloak     environment:       JAVA_OPTS: "-
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=*:5004
```

```
-Xms128m -Xmx128m -Djava.security.egd=file:/dev/./urandom"
#INFO - default value       KEYCLOAK_LOGLEVEL: "INFO"       #set
welcome theme       KEYCLOAK_WELCOME_THEME: "platform-v"       #INFO
- default value       ROOT_LOGLEVEL: "INFO"
PROMETHEUS_PUSHGATEWAY_ADDRESS: "localhost:9091/metrics"        ##
Параметры для аудита. Используются для модулей kcse-logging.
AUDIT_PROPERTY_NAMES:
"kafka.producer.bootstrap.servers,buffer.maxSize,mockMode"
AUDIT_BOOTSTRAP_SERVER: "localhost:9092"       AUDIT_BUFFER :
"10000"       AUDIT_MOCKMODE: "true"       AUDIT_MODE: "async"
AUDIT_NODE: "testnode.platformid"       AUDIT_MODULE: "PLID"
AUDIT_SOURCE_SYSTEM: "PlatformId"       AUDIT_SECURITY_PROTOCOL:
"SSL"       AUDIT_KEYSTORE_PASSWORD: "zaq12345678"
AUDIT_TRUSTSTORE_PASSWORD: "zaq12345678"        ## Настройки для
подключения к базе данных       DB_VENDOR: POSTGRES       DB_ADDR:
postgres       DB_DATABASE: keycloak       DB_USER: keycloak
DB_SCHEMA: public       DB_PASSWORD: password        ## Логин и
пароль администратора Keycloak       KEYCLOAK_USER: admin
KEYCLOAK_PASSWORD: admin        ## Настройка модуля kcse-sbersms
### Адрес для отправки SMS (Интеграция с @900)
SMS_SERVICE_ADDR:
https://ift.apim2.sberbank.ru:8443/prod/at900/message       ###
Адрес для аутентификации перед отправкой SMS (Интеграция с @900)
SMS_AUTH_SERVICE_ADDR:
https://ift.apim2.sberbank.ru:8443/prod/tokens/v2/oauth       ###
Client Id и Client Secret клиента, который будет отвечать за работу
с @900       SMS_AUTH_CLIENT_ID: 1045461f-257e-49aa-9297-
5722ec645238       SMS_AUTH_CLIENT_SECRET:
wF7wP3kU5mK1xD6qI5sI1hU2cS1jY7nH7dB5aP3cG0gF8dB0hF
SMS_AUTH_SCOPE: https://api.sberbank.ru/sendSMS       ## Количество
цифр в SMS-коде       SMS_COUNT_DIGIT: 6       SMS_SUBSYSTEM_CODE:
sberapi.sberId       SMS_OPERATION_NAME: SendClientNotificationsRq
SMS_SC_NAME: urn:sbrfsystems:99-sberapi       SMS_SP_NAME:
urn:sbrfsystems:99-900       SMS_SERVICE_NAME:
SrvSendClientNotification       SMS_SERVICE_NAMESPACE:
srv://sbg.sbr/ucn       SMS_SERVICE_VERSION: 002       SMS_MSG_TYPE:
"SberIdLoginConfirmed"       SMS_VERSION_ID: "2.0"
SMS_BUSINESS_PARAM_NAME: "code"       SMS_TECH_PARAM_NAME:
subSystemCode        ## Настройки для исходящего http трафика
OUTGOING_KEYSTORE_PASSWORD: "zaq12345678"
OUTGOING_KEY_PASSWORD: "zaq12345678"       X509_CA_BUNDLE:
/etc/x509/https/rootCA.crt        ## Настройки для модуля kcse-
syslog       SYSLOG_HOST: "10.53.4.53"       SYSLOG_PORT: 6914
SYSLOG_BY_SSL: "true"       SYSLOG_RFC_PROTOCOL: 5424
SYSLOG_BACKUP_HOST: "10.53.4.53"       SYSLOG_BACKUP_PORT: 515
SYSLOG_BACKUP_BY_SSL: "false"       SYSLOG_BACKUP_RFC_PROTOCOL: 5424
## Настройки для модуля kcse-period-of-use-account       ###
Параметр PASSWORD_POLICY_PERIOD_OF_USE предназначен для установки.
```

Если данный параметр присутствует, то модуль устанавливается.
PASSWORD_POLICY_PERIOD_OF_USE: "true"   PERIOD_OF_USAGE_ACCOUNT:
60   ## Настройки для модуля kcse-password-policy   ###
Параметр PASSWORD_POLICY предназначен для установки. Если данный
параметр присутствует, то модуль устанавливается.
PASSWORD_POLICY: "true"   ## Настройки для модуля kcse-
forbidden-sequence-password   ### Параметр
PASSWORD_POLICY_FORBIDDEN_SEQ предназначен для установки. Если
данный параметр присутствует, то модуль устанавливается.
PASSWORD_POLICY_FORBIDDEN_SEQ: "true"   ## Настройки для модуля
kcse-policy-unique-characters   ### Параметр
PASSWORD_POLICY_UNIQUE_CHARACTER предназначен для установки. Если
данный параметр присутствует, то модуль устанавливается.
PASSWORD_POLICY_UNIQUE_CHARACTER: "true"   ## Настройки для
модуля kcse-keycloak-user-no-activity   ### Параметр
PASSWORD_POLICY_USER_NO_ACTIVITY предназначен для установки. Если
данный параметр присутствует, то модуль устанавливается.
PASSWORD_POLICY_USER_NO_ACTIVITY: "true"   ## Настройки для
модуля kcse-audit-sender   ### Параметр AUDIT_SENDER
предназначен для установки. Если данный параметр присутствует, то
модуль устанавливается.   AUDIT_SENDER: "true"   ##
Настройки для модуля kcse-cryptp-pro   ### Параметр CRYPTO_PRO
предназначен для установки. Если данный параметр присутствует, то
модуль устанавливается.   CRYPTO_PRO: "true"   ## Настройки
для модуля kcse-keycloak-rest-module   ### Параметр REST_MODULE
предназначен для установки. Если данный параметр присутствует, то
модуль устанавливается.   REST_MODULE: "true"   ##
Настройки для модуля kcse-keycloak-code-authenticator   ###
Параметр CODE_AUTH предназначен для установки. Если данный параметр
присутствует, то модуль устанавливается.   CODE_AUTH: "true"
Настройки для модуля kcse-esia-idp   ### Параметр ESIA
предназначен для установки. Если данный параметр присутствует, то
модуль устанавливается.   ESIA: "true"   ## Настройки для
модуля kcse-keycloak-identity-adapters   ### Параметр
IDENTITY_ADAPTERS предназначен для установки. Если данный параметр
присутствует, то модуль устанавливается.   IDENTITY_ADAPTERS:
"true"   ## Настройки для модуля kcse-keycloak-attribute-
manager   ### Параметр ATTRIBUTE_MANAGER предназначен для
установки. Если данный параметр присутствует, то модуль
устанавливается.   ATTRIBUTE_MANAGER: "true"   ## Настройки
для модуля kcse-extended-idp   ### Параметр EXTENDED_IDP
предназначен для установки. Если данный параметр присутствует, то
модуль устанавливается.   EXTENDED_IDP: "true"   ##
Настройки для модуля kcse-russian-idp   ### Параметр RUSSIAN_IDP
предназначен для установки. Если данный параметр присутствует, то
модуль устанавливается.   RUSSIAN_IDP: "true"   ##
Настройки для модуля kcse-scim-adapters   ### Параметр
SCIM_ADAPTER предназначен для установки. Если данный параметр

```
присутствует, то модуль устанавливается.       SCIM_ADAPTER: "true"
volumes:       - logAccept:/opt/jboss/keycloak/standalone/log
- ./certsAT900:/opt/jboss/certs       - ./rootCA:/etc/x509/https
- ./esia:/opt/jboss/keycloak/standalone/configuration/ssl/
logging: #      driver: fluentd       options:         tag:
"docker.{{.ID}}"     ports:       - 8080:8080       - 9100:9100
depends_on:       - postgres  volumes:   logAccept:
```

Название параметра	Назначение	Пример	Допустимые значения	Значение по умолчанию
Параметры для подключения к БД (базе данных)				
DB_VENDOR	Тип базы данных	POSTGRES	postgres, mysql, mariadb, mssql, h2	h2
DB_ADDR	Адрес базы данных	postgres	Имя контейнера или IP-адрес	h2 (Если не задана переменная DB_VENDOR)
DB_PORT (Опционально)	Порт (будет использоваться в случае, если порт не задан в параметре DB_ADDR)	5432	Числовые	5432 (Если DB_VENDOR равно POSTGRES)
DB_DATABASE	Название базы данных	keycloak	-	-
DB_USER	Имя пользователя для доступа к БД	keycloak	-	-
DB_SCHEMA	Пространство имён	public	-	-
DB_PASSWORD	Пароль для доступа к БД	password	-	-
Базовые настройки KeyCloak.SE				
KEYCLOAK_USER (Опционально)	Имя пользователя с админскими правами. Если KEYCLOAK_USER и KEYCLOAK_PASSWORD не заданы KeyCloak.SE предложит создать данного пользователя при первом входе.	admin	-	-

KEYCLOAK_PASSWORD (Опционально)	Пароль пользователя с админскими правами. Если KEYCLOAK_USER и KEYCLOAK_PASSWORD не заданы KeyCloak.SE предложит создать данного пользователя при первом входе.	admin	-	-
KEYCLOAK_LOGLEVEL	Уровень логирования	INFO	DEBUG, INFO, WARNING, SEVERE, OFF	-
KEYCLOAK_WELCOME_THEME (Опционально)	Тема приветствия KeyCloak.SE	platform-v	platform-v, keycloak	keycloak
KEYCLOAK_DEFAULT_THEME (Опционально)	Тема KeyCloak.SE по умолчанию	platform-v	platform-v, keycloak	keycloak
Identity Provider (Поставщики аутентификации)				
RUSSIAN_IDP (Включает модуль)	Настройка для модуля kcse-russian-idp	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
ESIA (Включает модуль)	Настройка для модуля kcse-esia-idp	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
X509_ESIA_BUNDLE	Настройка для модуля kcse-esia-idp. Путь до сертификата ESIA в дистрибутиве	/etc/x509/https/trusted_keycloak_esia_portal.crt.pem	Строка	-
Журналирование и аудит				
AUDIT_SENDER_REST	Настройки для модуля kcse-audit-sender-rest	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
AUDIT_SENDER	Настройки для модуля kcse-audit-sender	"true"	Любое значение - вкл/ Параметр не задан - выкл	-

AUDIT_PROPERTY_NAMES	Настройки для модуля kcse-logging.	"kafka.producer.bootstrap.servers,buffer.maxSize,mockMode"	-	-
AUDIT_BOOTSTRAP_SERVER (Включает модуль)	Настройки для модуля kcse-logging.	"localhost:9092"	-	-
AUDIT_BUFFER	Настройки для модуля kcse-logging.	"10000"	-	-
AUDIT_MOCKMODE	Настройки для модуля kcse-logging.	"true"	-	-
AUDIT_MODE	Настройки для модуля kcse-logging.	"async"	-	-
AUDIT_NODE	Настройки для модуля kcse-logging.	"testnode.platformid"	-	-
AUDIT_MODULE	Настройки для модуля kcse-logging.	"PLID"	-	-
AUDIT_SOURCE_SYSTEM	Настройки для модуля kcse-logging.	"PlatformId"	-	-
AUDIT_SECURITY_PROTOCOL	Настройки для модуля kcse-logging.	"SSL"	-	-
AUDIT_KEYSTORE_PASSWORD	Настройки для модуля kcse-logging.	"zaq12345678"	-	-
AUDIT_TRUSTSTORE_PASSWORD	Настройки для модуля kcse-logging.	"zaq12345678"	-	
SHOW_EVENTS_CONFIG_TAB	Настройки для модуля kcse-keycloak-rest-module. Позволяет конфигурировать вкладку событий.	"false"	-	-
Параметры, которым нужно присвоить любое значение для того, чтобы прогнать cli-скрипты для конкретного модуля				
SCIM_ADAPTER	Настройка для модуля kcse-scim-adapters	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
EXTENDED_IDP	Настройка для модуля kcse-extended-idp	"true"	Любое значение - вкл/ Параметр не задан - выкл	-

ATTRIBUTE_MANAGER	Настройка для модуля kcse-keycloak-attribute-manager	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
IDENTITY_ADAPTERS	Настройка для модуля kcse-keycloak-identity-adapters	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
CODE_AUTH	Настройка для модуля kcse-keycloak-code-authenticator	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
REST_MODULE	Настройки для модуля kcse-keycloak-rest-module	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
CRYPTO_PRO	Настройки для модуля kcse-cryptp-pro	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
Парольные политики				
PASSWORD_POLICY_USER_NO_ACTIVITY	Настройки для модуля kcse-keycloak-user-no-activity	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
PASSWORD_POLICY_UNIQUE_CHARACTER	Настройки для модуля kcse-policy-unique-characters	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
PASSWORD_POLICY_FORBIDDEN_SEQ	Настройки для модуля kcse-forbidden-sequence-password	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
PASSWORD_POLICY	Настройки для модуля kcse-password-policy	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
PASSWORD_POLICY_PERIOD_OF_USE	Настройки для модуля kcse-password-policy.	"true"	Любое значение - вкл/ Параметр не задан - выкл	-
PERIOD_OF_USAGE_ACCOUNT	Настройки для модуля kcse-password-policy. Запускать Scheduler, отвечающий за сброс сессий у пользователей с истекшим сроком использования аккаута, каждые N минут.	60	Числовые	60

2. Авторизоваться в docker registry (необходим доступ к директории в docker registry, где находится образ) при помощи команды:
    - `docker login -u $your_username $your_registry`
    - Ввести пароль
3. Запустить docker-compose файл в командной строке по команде:
    - `docker login -u $your_username $your_registry`
4. В случае успешного поднятия, вы увидите лог действий, находящийся в командной строке с запущенным контейнером
5. Для проверки того, что контейнер запущен, перейдите по адресу http://localhost:18080 (порт был задан ранее в docker-compose файле)
6. Вам откроется консоль администратора
7. Для входа в консоль администратора используется login/password администратора (указаны в docker-compose файле)
8. Для выключения docker - контейнера необходимо прописать в терминале

```
docker-compose down
```

Установка в среде контейнеризации

Использование среды контейнеризации для оркестрации компонента KeyCloak.SE продукта Platform V IAM SE позволяет гибко управлять конфигурацией, производить масштабирование в зависимости от нагрузки, а также устанавливать сопутствующие приложения.

1. В директории **keycloak/extensions/kcse-distribution/src/main/** находится ряд yml-файлов, для деплоя в среде контейнеризации. Необходимо заполнить параметры всех файлов (secret.yml, fluentd-config-map.yml, deploymentConfig.yml, headless-service.yml, service.yml, route.yml, registrySecret.yml) путём template-processing в ansible, либо непосредственным указанием значений параметров в этих файлах исходя из окружающей среды установки.

    1.1. Например, ниже приведен файл *DeploymentConfig* с указанными параметризированными свойствами для ansible:

```
apiVersion: v1   kind: DeploymentConfig   metadata:     name:
keycloak-app   spec:     replicas: {{ application.replicas }}
revisionHistoryLimit: 10     selector:       deploymentconfig:
keycloak-app     strategy:       activeDeadlineSeconds: 21600
resources: {}       rollingParams:         intervalSeconds: 1
maxSurge: 25%         maxUnavailable: 25%         timeoutSeconds:
600         updatePeriodSeconds: 1       type: Rolling     template:
metadata:         creationTimestamp: null         labels:
name: frontend           deploymentconfig: keycloak-app       spec:
containers:           - env:               - name: KEYCLOAK_LOGLEVEL
value: "{{ loglevel.keycloak }}"              - name: ROOT_LOGLEVEL
value: "{{ loglevel.root }}"   {%if audit.enabled %}              -
name: AUDIT_PROPERTY_NAMES               value: "{{
audit.propertyNames }}"              - name: AUDIT_BOOTSTRAP_SERVER
value: "{{ audit.bootstrapServer }}"              - name:
AUDIT_BUFFER               value: "{{ audit.buffer }}"
```

```
- name: AUDIT_MOCKMODE                value: "{{ audit.mockmode }}"
- name: AUDIT_MODE                  value: "{{ audit.mode }}"
- name: AUDIT_NODE                  value: "{{ audit.node }}"
- name: AUDIT_MODULE                  value: "{{ audit.module }}"
- name: AUDIT_SOURCE_SYSTEM                value: "{{
audit.sourceSystem }}"               - name: AUDIT_SECURITY_PROTOCOL
value: "{{ audit.securityProtocol }}"              - name:
AUDIT_TRUSTSTORE_LOCATION                value: "{{
audit.truststoreLocation }}"              - name:
AUDIT_KEYSTORE_LOCATION                value: "{{
audit.keystoreLocation }}"              - name:
AUDIT_TRUSTSTORE_PASSWORD                value: "{{
audit.truststorePassword }}"              - name:
AUDIT_KEYSTORE_PASSWORD                value: "{{
audit.keystorePassword }}"   {% endif %}              - name:
KEYCLOAK_USER                value: admin              - name:
KEYCLOAK_PASSWORD                valueFrom:
secretKeyRef:                    key: application-password
name: {{ application.name }}              - name: DB_USER
value: {{ db.user }}              - name: DB_PASSWORD
valueFrom:                   secretKeyRef:                    key:
db-password                   name: {{ application.name }}
- name: DB_ADDR                value: {{ db.server }}
- name: DB_VENDOR                value: POSTGRES               -
name: DB_PORT                value: "{{ db.port }}"               -
name: DB_DATABASE                value: {{ db.name }}
- name: JGROUPS_DISCOVERY_PROTOCOL                value:
dns.DNS_PING               - name: TZ                value:
Europe/Moscow               - name: JGROUPS_DISCOVERY_PROPERTIES
value: >-                   dns_query={{ application.jgroups.query
}}               - name: JAVA_OPTS                value: {{
application.javaOptions }}              - name:
CACHE_OWNERS_AUTH_SESSIONS_COUNT                value: "{{
application.jgroups.owners }}"              - name:
JGROUPS_TRANSPORT_STACK                value: "tcp"   {%if srp.url
is defined  %}               - name: CSA_ADDR                value:
"{{ srp.url }}"   {% endif %}   {%if adminPort is defined %}
- name: ADMIN_PORT                value: "{{ adminPort }}"   {%
endif %}   {%if nmt is defined %}              - name:
SMS_SERVICE_ADDR                value: "{{ nmt.smsService }}"
- name: SMS_SUBSYSTEM_CODE                value: "{{
nmt.subsystemCode }}"              - name: SMS_OPERATION_NAME
value: "{{ nmt.operationName }}"              - name:  SMS_SC_NAME
value: "{{ nmt.scName }}"              - name:  SMS_SP_NAME
value: "{{ nmt.spName }}"              - name:  SMS_SERVICE_NAME
value: "{{ nmt.serviceName }}"              - name:
SMS_SERVICE_NAMESPACE                value: "{{
nmt.serviceNamespace }}"              - name:  SMS_SERVICE_VERSION
```

```
value: "{{ nmt.serviceVersion }}"              - name:
SMS_AUTH_SERVICE_ADDR                  value: "{{
nmt.smsAuthServiceAddr }}"               - name:  SMS_AUTH_CLIENT_ID
value: "{{ nmt.clientId }}"               - name:
SMS_AUTH_CLIENT_SECRET                  valueFrom:
secretKeyRef:                       key: client-secret
name: {{ application.name }}               - name:  SMS_AUTH_SCOPE
value: "{{ nmt.authScope }}"               - name: SMS_COUNT_DIGIT
value: "{{ nmt.smsCountDigit }}"              - name: SMS_MSG_TYPE
value: "{{ nmt.smsMsgType }}"              - name: SMS_VERSION_ID
value: "{{ nmt.versionId }}"               - name:
SMS_BUSINESS_PARAM_NAME                  value: "{{
nmt.businessParamName }}"               - name: SMS_TECH_PARAM_NAME
value: "{{ nmt.techParamName }}"   {% endif %}               - name:
OUTGOING_KEYSTORE_PASSWORD                  valueFrom:
secretKeyRef:                       key: outgoing-key-password
name: {{ application.name }}               - name:
OUTGOING_KEY_PASSWORD                  valueFrom:
secretKeyRef:                       key: outgoing-key-password
name: {{ application.name }}               - name: X509_CA_BUNDLE
value: /etc/x509/https/rootCA.crt             image: >-
${docker.registry.url}             imagePullPolicy: Always
readinessProbe:               httpGet:                 path: /auth/
port: 8443                  scheme: HTTPS
initialDelaySeconds: {{
application.probes.readiness.initialDelaySeconds}}
timeoutSeconds: {{ application.probes.readiness.timeoutSeconds}}
periodSeconds: {{ application.probes.readiness.periodSeconds}}
successThreshold: 1               failureThreshold: 3
name: keycloak-app             livenessProbe:               httpGet:
path: /auth/                  port: 8443                 scheme:
HTTPS               initialDelaySeconds: {{
application.probes.liveness.initialDelaySeconds}}
timeoutSeconds: {{ application.probes.liveness.timeoutSeconds}}
periodSeconds: {{ application.probes.liveness.periodSeconds}}
successThreshold: 1               failureThreshold: 3
ports:               - containerPort: 8443                 protocol:
TCP             resources:               limits:
cpu: "{{ application.resources.limits.cpu }}"
memory: "{{ application.resources.limits.memory }}"
requests:                 cpu: "{{
application.resources.requests.cpu }}"                 memory: "{{
application.resources.requests.memory }}"
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File             volumeMounts:
- name: outgoing-cert                 readOnly: true
mountPath: /opt/jboss/certs               - name: certs
readOnly: true                mountPath: /etc/x509/https
```

```
- name: shared-logs                mountPath:
/opt/jboss/keycloak/standalone/log/   {%if sidecar.fluentd.endpoint
is defined %}          - name: fluentd            image: {{
sidecar.fluentd.image }}            imagePullPolicy: Always
resources:              limits:                cpu: "{{
sidecar.fluentd.resources.limits.cpu }}"               memory: "{{
sidecar.fluentd.resources.limits.memory }}"              requests:
cpu: "{{ sidecar.fluentd.resources.requests.cpu }}"
memory: "{{ sidecar.fluentd.resources.requests.memory }}"
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File            volumeMounts:
- name: fluentd-root-config               mountPath: /fluentd/etc/
readOnly: true              - name: shared-logs
mountPath: /opt/jboss/keycloak/standalone/log/              - name:
certs                readOnly: true                mountPath:
/certs-for-rest/   {% endif %}   {%if
sidecar.fluentbit.endpoint.host is defined %}          - name:
fluentbit            image: {{ sidecar.fluentbit.image }}
imagePullPolicy: Always            resources:              limits:
cpu: "{{ sidecar.fluentbit.resources.limits.cpu }}"
memory: "{{ sidecar.fluentbit.resources.limits.memory }}"
requests:               cpu: "{{
sidecar.fluentbit.resources.requests.cpu }}"               memory:
"{{ sidecar.fluentbit.resources.requests.memory }}"
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File            volumeMounts:
- name: fluentbit-root-config              mountPath: /fluent-
bit/etc/               readOnly: true              - name:
shared-logs              mountPath: /keycloak/log/
- name: certs              readOnly: true
mountPath: /certs-for-rest/   {% endif %}        dnsPolicy:
ClusterFirst        restartPolicy: Always        schedulerName:
default-scheduler        securityContext: {}
serviceAccount: {{ serviceAccount }}        serviceAccountName: {{
serviceAccount }}        terminationGracePeriodSeconds: 30
volumes:          - name: outgoing-cert            secret:
secretName: {{ application.name }}            items:
- key: outgoing.keystore                path: outgoing.keystore
defaultMode: 444          - name: certs            secret:
secretName: {{ application.name }}            items:
- key: tls.key                path: tls.key              -
key: tls.crt                path: tls.crt              - key:
rootCA.crt                path: rootCA.crt
defaultMode: 444   {%if sidecar.fluentd.endpoint is defined %}
- name: fluentd-root-config            configMap:
name: fluentd-config            items:              - key:
fluent.conf                path: fluent.conf   {% endif %}   {%if
sidecar.fluentbit.endpoint.host is defined %}          - name:
```

```
fluentbit-root-config             configMap:                name:
fluentd-config               items:                  - key: fluent-
bit.conf                   path: fluent-bit.conf                -
key: parser_java.conf                    path: parser_java.conf
- key: stream_processor.conf                   path:
stream_processor.conf   {% endif %}          - name: shared-logs
emptyDir: {}     test: false     triggers:       - type:
ConfigChange
```

2. После параметризации свойств этих документов необходимо запустить процедуру развертывания (Deployment) в среде контейнеризации.
3. После запуска POD необходимо проверить работоспособность приложения согласно url, прописанному в Routes для данного приложения.
4. Вам откроется консоль администратора

Чек лист валидации установки в среде контейнеризации

- Во всех yml-файлах директории **keycloak/extensions/kcse-distribution/src/main/** были заполнены параметризированные параметры;
- После параметризации свойств этих документов запущен *DeploymentConfig* в среде контейнеризации;
- По url, прописанному в Routes для данного приложения, открывается консоль администратора.

Установка Standalone

Автономный режим работы полезен только в том случае, если вы хотите запустить один и только один экземпляр сервера KeyCloak.SE. Он непригоден для кластерных развертываний, и все кэши являются нераспределенными и только локальными. Не рекомендуется использовать автономный режим в рабочей среде, так как у вас будет одна точка отказа. Если ваш сервер автономного режима выйдет из строя, пользователи не смогут войти в систему. Этот режим действительно полезен только для тест-драйва и игры с функциями KeyCloak.SE.

Основная часть этого раздела посвящена настройке аспектов KeyCloak.SE на уровне инфраструктуры. Эти аспекты настраиваются в конфигурационном файле, специфичном для сервера приложений, производной которого является KeyCloak.SE. В режиме автономной работы этот файл находится в .../standalone/configuration/standalone.xml. Этот файл также используется для настройки неинфраструктурного уровня, специфичного для компонентов KeyCloak.SE.

1. Для установки компонента KeyCloak.SE продукта Platform V IAM SE Standalone необходимо после загрузки автономного сервера с модулями через терминал запустить ./standalone.sh
    - `$ .../bin/standalone.sh`
2. Перейти по адресу http://localhost:8080/auth/
3. Вам откроется консоль администратора

Чек лист валидации установки Standalone

- Загружен корректный дистрибутив компонента KeyCloak.SE продукта Platform V IAM SE с модулями

- Консоль администратора открывается

Структура каталога дистрибутива

Разберем назначение некоторых каталогов:

- **bin/** - Содержит различные сценарии либо для загрузки сервера, либо для выполнения других действий управления на сервере.
- **docs/** - Содержит документацию компонента KeyCloak.SE продукта Platform V IAM SE, включающую в себя следующие разделы: Руководство системного администрирования, Руководство оператора, Руководство по установке, Руководство по безопасности, Руководство прикладного разработчика, Детальная архитектура Программа и методика испытаний, Документация REST API, Примечания к релизу. Документация содержится в формате markdown.
- **domain/** - Содержит свойства, настройки для ролей, пользователей и т.д. для различных realm.
- **locales/** - Содержит интернационализированные свойства отображения в интерфейсе компонента KeyCloak.SE продукта Platform V IAM SE.
- **modules/** - Содержит как зависимости и артефакты которые нужны для корректной работы Keycloak.SE (библиотеки), так и модули компонента KeyCloak.SE продукта Platform V IAM SE.
- **standalone/** - Содержит сервисные конфигурационные файлы запускаемого проекта, логи.
- **themas/** - Содержит все html, таблицы стилей, файлы JavaScript и изображения, используемые для отображения всех экранов пользовательского интерфейса, выводимых сервером.

Настройка сети

KeyCloak.SE может работать из коробки с некоторыми сетевыми ограничениями. Например, все сетевые конечные точки привязываются к localhost, поэтому сервер авторизации можно использовать только на одной локальной машине. Для HTTP-соединений он не использует порты по умолчанию, такие как 80 и 443. HTTPS/SSL не настраивается из коробки, а без этого KeyCloak.SE имеет множество уязвимостей в безопасности. Наконец, KeyCloak.SE может часто нуждаться в безопасных SSL и HTTPS соединениях с внешними серверами, и поэтому ему необходимо настроить хранилище доверия, чтобы конечные точки могли быть проверены правильно.

Обновление

Обновление осуществляется путем получения новой версии от команды разработки.

При получении обновленной версии от команды разработки (ссылка на образ) возможно будет необходимо обновить/актуализировать список параметров в DeploymentConfig.

Удаление

Удаление осуществляется путём удаления всех объектов в среде контейнеризации в директории, указанной в разделе "Установка в среде контейнеризации".

Откат

Откат на предыдущую версию не предусмотрен

Часто встречающиеся проблемы и пути их устранения

При получении обновленной версии от команды разработки (ссылка на образ) возможно будет необходимо обновить/актуализировать список параметров в DeploymentConfig.

При возникновении ошибок при установке в среде контйнеризации нужно локализовать конкретный POD и просмотреть его events и logs. Если в логах ошибка отсутствует, то необходимо повысить уровень логирования, например поставить уровень DEBUG.

Если, при попытке установки с помощью docker-образа, не получается авторизоваться через docker login, то обратитесь к системному администратору.