Platform V Addresses (FIA)

Руководство по установке

ОГЛАВЛЕНИЕ

Руководство по установке

О документе

Документ содержит сведения о порядке установки и первоначальной настройки продукта Platform V Addresses. Также в документе приводятся требования к программному и аппаратному обеспечению, описание дистрибутива и параметров компонента управления параметрами Platform V Configuration (CFGE).

Документ предназначен для администратора АС, выполняющего установку, обновление и первоначальную настройку продукта Platform V Addresses.

Основные понятия

В таблице приведены основные аббревиатуры и сокращения:

Аббревиатура, сокращение	Определение
АС	Автоматизированная система
БД	База данных
ОС	Операционная система
ПК	Персональный компьютер
ПО	Программное обеспечение
СУБД	Система управления базами данных
ФИАС	Федеральная информационная адресная система
ФНС	Федеральная налоговая служба
ФП	Функциональная подсистема
ЦСПС	Центр сопровождения прикладных сервисов
API	Application programming interface. Набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом)
BH	Business hub
CPU	Central processing unit. Центральное обрабатывающее устройство
JVM	Java Virtual Machine. Виртуальная машина Java (OpenJDK)
mTLS	Mutual Transport Layer Security. Протокол защиты транспортного уровня
PL	Презентационный слой
RAM	Random Access Memory. Оперативная память

В таблице приведены основные термины и определения:

Термин	Определение

Платформа	Набор продуктов Platform V, правообладателем которых является АО “СберТех”. Перечень таких продуктов обозначен в документации на конкретный продукт
Продукт	Platform V Addresses
Хранилище S3	Облачное объектное хранилище, совместимое с протоколом S3
Kafka	Компонент Kafka Sber Edition продукта Platform V Corax (KFKA)
OpenSearch	Пакет поиска и аналитики с открытым исходным кодом, лицензированный Apache 2.0, который упрощает сбор, поиск, визуализацию и анализ данных
S3	Тип файлового ресурса

Системные требования

Требования к программному обеспечению

Необходимо наличие следующего ПО:

- Среда управления контейнерами Kubernetes v.1.19.15 и выше (или OpenShift 4.5.13 и выше).
- Виртуальная машина Java для работы Java-приложений Java OpenJDK версии 11.
- Инфраструктура для обеспечения взаимодействия между сервисами – Istio (Platform V Synapse Service Mesh версии 1.12 и выше).
- СУБД – PostgreSQL (рекомендован Platform V Pangolin SE v.4.6.0 и выше).
- Утилита для удаленной настройки сервисов при осуществлении ручной установки (например, Ansible).
- Интерфейс командной строки для управления Kubernetes при осуществлении ручной установки – Kubectl версии, соответствующей версии Kubernetes. Является составной частью Kubernetes, устанавливается на рабочей станции.

Требования к аппаратным ресурсам

Имеются следующие требования:

Компонент	CPU (количество ядер)	RAM (Гб)	Диск (ГБ)	SAN (ГБ) или RDM (ГБ)	Среда развертывания
addresses-manager	4	4			Kubernetes или OpenShift
addresses-parser	8	16	10		Kubernetes или OpenShift
addresses-adapter	2	4	1		Kubernetes или OpenShift

addresses-finder	2	4	1		Kubernetes или OpenShift
Хранилище S3				SAN 200	CEPH
СУБД	Согласно документации производителя	Согласно документации производителя	0,01		Отдельный виртуальный или физический сервер с архитектурой x86
Kafka					Отдельный виртуальный или физический сервер с архитектурой x86

Сервер приложений

Для обеспечения корректной работы продукта Platform V Addresses необходимо предоставить проект в среде контейнеризации с параметрами:

```
{
"cpu": "16",
"mem": "28Gi"
}
```

Параметры для обеспечения работы каждого компонента продукта описаны в таблице:

Компонент	CPU limit	Mem limit (Мб)	CPU request	Mem request (Мб)
address-manager	3	3128	3	3128
address-parser-manager	800m	4200	600m	4096
address-adapter	800m	5200	600m	5096
address-finder	800m	1200	600m	1200
fluent-bit-sidecar	200	256	100	128
istio-proxy-sidecar	500m	1024	500m	1024

Рабочее место пользователя

Для обеспечения корректной работы пользователя с продуктом на машине пользователя должны быть установлены:

- Операционная система – Альт 8 СП, Windows 7 и выше (32/64-разрядная), Ubuntu 12.04 LTS и выше (32/64-разрядная), Mac OS X 10.6 и выше (32/64-разрядная).
- Веб-браузер – Яндекс.Браузер или Internet Explorer (Edge) 11 и выше.

Требования к среде установки

Сервис не является автономным и нуждается в интеграции микросервисов (дополнительных настроек к программно-аппаратным средствам не требуется).

Интеграция со следующими сервисами опциональна:

- Компонент Управление параметрами продукта Platform V Configuration (CFGE) – осуществляет централизованное управление параметрами продукта. Некоторые параметры применяют во время работы приложения (например, healthcheck.cascadeprotection.cpu-threshold, healthcheck.cascadeprotection.ram-threshold – пороговые значения утилизации CPU и RAM, при превышении которых не будут обрабатываться клиентские запросы; addresses.scheduler.cron.full, addresses.scheduler.cron.delta – параметры графика запуска полного и частичного импорта данных из fias.nalog.ru), другие параметры применяются при запуске.
- Компонент Журналирование продукта Platform V Monitor (LOGE) – предоставляет инструменты для вывода в журнал событий, производимых в ходе работы продукта (инфо-сообщения, ошибки).
- Компонент Прикладной мониторинг продукта Platform V Monitor (MONE) – сбор метрик производительности работы продукта.
- Компонент Аудит Platform V Audit (AUDE) – контроль за работой пользователя с продуктом.
- Компоненты Аутентификация и IAM Proxy продукта Platform V IAM (AUTH, ATHE).
- Сервис файловой передачи (Synapse) продукта Platform V Synapse File Exchange (FTRF) – требуется для загрузки данных с сайта ФИР.
- Компонент Внутренний шлюз продукта Platform V API Management (IAGW) – требуется опционально (зависит от варианта окружения).

Интеграция со следующими сервисами обязательна:

- Продукт Platform V Synapse Service Mesh – требуется с целью осуществления функции маршрутизации и создания защищенного соединения.
- Компонент Platform V Kafka Sber Edition продукта Platform V Corax.
- Хранилище S3 – требуется для хранения загруженных данных.
- Продукт OpenSearch (или же Platform V Index Search (IDXS)) – является хранилищем адресных объектов и обеспечивает их сохранение при импорте из ресурсов, предоставляемых ФИАС. Platform V Addresses обращается к хранилищу за требуемыми данными.

Описание версий компонентов окружения, при которых работоспособность продукта протестирована, приведено в разделе «Варианты работы в различном окружении» документа «Детальная архитектура».

Установка

До осуществления установки продукта Platform V Addresses со стороны администратора стенда требуется выпустить сертификат. Сертификат требуется для обеспечения защищенного взаимодействия продукта по протоколу mTLS с иными компонентами в среде контейнеризации через Platform V Synapse Service Mesh (см. раздел «Сценарии администрирования» документа «Руководство системного администратора»).

Перечень сторонних компонентов, интеграция с которыми необходима для успешного функционирования продукта, приведен в разделе «Требования к среде установки». О недоступности любого из перечисленных компонентов во время установки продукта будет сообщено в результатах работы CI/CD по установке продукта.

Интеграционные параметры для инструментов мониторинга, аудита, сервиса управления параметрами прописаны в конфигурационном файле addresses.all.conf, входящему в состав дистрибутива продукта. Агент HealthCheck реализован на уровне кода (класс `FiasSuggestionHealthCheck`) и также содержится в поставке дистрибутива, поэтому никаких дополнительных настроек осуществлять не требуется. Файлы инструкций, включающие в себя описание создаваемых объектов для Kafka, находятся в каталоге дистрибутива *\package\conf\config\definitions*.

Подключение компонента Сервис файловой передачи (Synapse) продукта Platform V Synapse File Exchange осуществляется в соответствии с разделом «Подключение и конфигурирование» документа «Руководство прикладного разработчика» данного компонента.

Описание механизма аутентификации приведено в разделе “Доступ к приложению” документа “Руководство оператора”.

Описание параметров метамодели событий аудита приведено в разделе «Аудит» документа «Руководство по безопасности».

Состав дистрибутива

Дистрибутив сервиса состоит из следующих каталогов:

- **\package\bh** – исполняемые jar-файлы в среде JVM (adaptere.jar, finder.jar, loader.jar, parser-manager.jar);
- **\package\conf** – конфигурационные файлы продукта Platform V Addresses;
- **\package\conf\config\definitions** – файлы инструкций, включающие в себя описание создаваемых объектов для Kafka;
- **\package\conf\config\parameters** – файлы конфигурационных параметров общие для всех подов оркестратора (Kubernetes или OpenShift) addresses.all.conf, а также в частности для каждого пода (addresses-adapter.conf, addresses-finder.conf и т.д.);
- **\package\conf\config\data** – каталоги, которые содержат файлы импорта данных для платформенных компонентов Аутентификация и Авторизация Platform V IAM, компонент Журналирование Platform V Monitor;
- **\package\conf\config\global**, **\package\conf\config\iag** – каталоги с настройками для внутреннего шлюза IAG (компонент Внутренний шлюз продукта Platform V API Management);
- **\package\conf\inventory** – файлы конфигурации серверов;
- **\package\conf\k8s\base** – каталог содержит общие файлы настроек: docker, istio, configmaps, secrets, для оркестраторов контейнеров Kubernetes или OpenShift, а также deployment configs только для Kubernetes;
- **\package\conf\k8s\overrides\openshift** – каталог содержит файлы настроек deployment configs для оркестратора контейнеров OpenShift (аналогичные для Kubernetes описывались в предыдущем пункте и находятся уровнем выше);
- **\package\conf\templates** – файлы настроек шлюза;
- **\package\distrib.yml** – файл, описывающий расположение и состав дистрибутива;
- **\package\pipeline.yml** – файлы с настройками для системы CI/CD по установке сервиса;
- **\package\db** – скрипты (Liquibase) для инициализации объектов базы данных;

- **\package\documentation** – документация продукта Platform V Addresses.

Порядок установки и первоначальной настройки при ручной установке

В данном разделе описан ручной способ установки продукта. Данный способ установки является целевым (основным) вариантом установки.

Продукт не нуждается в специальной настройке, влияющей на безопасность данных. Настройки безопасности осуществляются за пределами конфигурации сервиса (см. описание механизмов безопасности в документе «Руководство по безопасности»).

Все настраиваемые параметры продукта описаны в разделе «Настраиваемые параметры» документа «Руководство по системному администрированию».

Для ручной установки выполните следующие действия:

1. Скачайте дистрибутив на персональный компьютер.

2. Разархивируйте дистрибутив.

3. Подготовьте директорию.

   Подготовьте yaml-манифесты с secres. Добавьте к именам secres окончание .unver. Создайте yaml-манифест с именем *secret-addresses.unver* и поместите все параметры для _password.conf в него.

   Пример:

```
kind: Secret
apiVersion: v1
metadata:
name: secret-addresses.unver
type: Opaque
data:
DB_CURRATE_MAIN_LOGIN: <base64 enc >
DB_CURRATE_MAIN_PASSWORD:  <base64 enc >
# База данных CR SI
DB_CURRATE_STANDIN_LOGIN:  <base64 enc >
DB_CURRATE_STANDIN_PASSWORD:  <base64 enc >
#
DB_EKPIT_MAIN_LOGIN:  <base64 enc >
DB_EKPIT_MAIN_PASSWORD:  <base64 enc >
DB_EKPIT_STANDIN_LOGIN:  <base64 enc >
DB_EKPIT_STANDIN_PASSWORD:  <base64 enc >
#
MESSENGER_KAFKA_SSL_KEY_STORE_PASSWORD:  <base64 enc >
MESSENGER_KAFKA_SSL_TRUST_STORE_PASSWORD:  <base64 enc >
# пароль от закрытого ключа
MESSENGER_KAFKA_SSL_PRIVATE_KEY_PASSWORD:  <base64 enc >
#
OTT_CERTSTORE_PWD:  <base64 enc >
OTT_TRUST_STORE_PWD:  <base64 enc >
```

```
    # пароль от закрытого ключа
    OTT_CERTSTORE_PRIVATE_KEY_PWD:  <base64 enc >
    #
    AJ_KEYSTORE_PASSWORD:  <base64 enc >
    AJ_TRUSTORE_PASSWORD:  <base64 enc >
    # пароль от закрытого ключа
    AJ_KEY_PRIVATE_KEY_PASSWORD: <base64 enc >
```

4. Установите все secret командой:

   ```
   kubectl apply -f <имя файла манифеста> -n <пространство имен для установки>
   ```

   В результате будут созданы все secres из блока *extra-secrets* в *distib.yaml* и *secret-addresses.unver*.

5. Шаблонизируйте вручную yaml-манифесты из дистрибутива.

6. Перейдите в директорию *package/conf/k8s/base*. В текущей директории расположены поддиректории, отвечающие за каждый сервис. В директории *istio* расположены манифесты для настройки сервисной сети.

Для того, чтобы шаблонизировать манифесты в каждой поддиректории, воспользуйтесь любым инструментом (например, Ansible). Пример Ansible для шаблонизации:

```
--
- hosts: 127.0.0.1_
  connection: local
  tasks:
   - name: "template"
     ansible.builtin.template:
     src : package/conf/k8s/base/{вставить_необходимый_микросервис}/dc.yaml
     dest: ./<Директория для хранения итоговых манифестов>
```

Далее вызовите playbook и передайте параметры соответствующему сервису:

```
ansible-playbook ./pipline.yaml --extra-vars "@package/conf/config/parameters/{в
ставить_необходимый_микросервис}.conf" --extra-vars @/package/conf/config/parame
ters/addresses.istio..all.conf
```

Результат работы ansible-playbook:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: addresses-adapter-${distrib.release.version}
    group: addresses
    unimonId: addresses-adapter-unimon-id
  name: addresses-adapter-${distrib.release.version}
spec:
  replicas: ${addresses-adapter.ose.deployment.spec.replicas}
  selector:
```

```
    matchLabels:
      app: addresses-adapter-${distrib.release.version}
      deploymentconfig: addresses-adapter-${distrib.release.version}
  strategy:
    activeDeadlineSeconds: 21600
    resources: {}
    rollingParams:
      intervalSeconds: ${addresses.ose.deployment.spec.strategy.rollingParams.in
tervalSeconds}
      maxSurge: ${addresses.ose.deployment.spec.strategy.rollingParams.maxSurge}
      maxUnavailable: ${addresses.ose.deployment.spec.strategy.rollingParams.max
Unavailable}
      timeoutSeconds: ${addresses.ose.deployment.spec.strategy.rollingParams.tim
eoutSeconds}
      updatePeriodSeconds: ${addresses.ose.deployment.spec.strategy.rollingParam
s.updatePeriodSeconds}
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: addresses-adapter-${distrib.release.version}
        deploymentconfig: addresses-adapter-${distrib.release.version}
      annotations:
        haproxy.router.openshift.io/timeout: 180s
        sidecar.istio.io/inject: 'true'
        sidecar.istio.io/proxyCPU: ${addresses-adapter.ose.deployment.metadata.a
nnotations.sidecar.istio.io.proxyCPU}
        sidecar.istio.io/proxyMemory: ${addresses-adapter.ose.deployment.metadat
a.annotations.sidecar.istio.io.proxyMemory}
        sidecar.istio.io/proxyCPULimit: ${addresses-adapter.ose.deployment.metad
ata.annotations.sidecar.istio.io.proxyCPULimit}
        sidecar.istio.io/proxyMemoryLimit: ${addresses-adapter.ose.deployment.me
tadata.annotations.sidecar.istio.io.proxyMemoryLimit}
    spec:
      containers:
        - image: {{registry}}{{registry_path}}/fias/addresses-adapter@${jenkins_
env.fp_image_hash}
          name: addresses-adapter-${distrib.release.version}
          envFrom:
            - configMapRef:
                name: addresses-adapter.conf.${distrib.release.version}
            - configMapRef:
                name: addresses.all.conf.${distrib.release.version}
            - secretRef:
                name: secret-secret.${distrib.release.version}
          env:
            - name: ufs.tenant.code
              value: {{ TENANT_CODE }}
            - name: USER_INSTALL_ROOT
              value: /opt/conf
          imagePullPolicy: Always
          ports:
```

```
            - containerPort: 8080
              protocol: TCP
            - containerPort: 8081
              protocol: TCP
          livenessProbe:
            httpGet:
              path: ${addresses-adapter.ose.deployment.spec.template.spec.contai
ners.addresses-adapter.livenessProbe.httpGet.path}
              port: ${addresses-adapter.ose.deployment.spec.template.spec.contai
ners.addresses-adapter.livenessProbe.httpGet.port}
              scheme: ${addresses-adapter.ose.deployment.spec.template.spec.cont
ainers.addresses-adapter.livenessProbe.httpGet.scheme}
            failureThreshold: ${addresses-adapter.ose.deployment.spec.template.s
pec.containers.addresses-adapter.livenessProbe.failureThreshold}
            initialDelaySeconds: ${addresses-adapter.ose.deployment.spec.templat
e.spec.containers.addresses-adapter.livenessProbe.initialDelaySeconds}
            periodSeconds: ${addresses-adapter.ose.deployment.spec.template.spec
.containers.addresses-adapter.livenessProbe.periodSeconds}
            successThreshold: ${addresses-adapter.ose.deployment.spec.template.s
pec.containers.addresses-adapter.livenessProbe.successThreshold}
            timeoutSeconds: ${addresses-adapter.ose.deployment.spec.template.spe
c.containers.addresses-adapter.livenessProbe.timeoutSeconds}
          readinessProbe:
            httpGet:
              path: ${addresses-adapter.ose.deployment.spec.template.spec.contai
ners.addresses-adapter.readinessProbe.httpGet.path}
              port: ${addresses-adapter.ose.deployment.spec.template.spec.contai
ners.addresses-adapter.readinessProbe.httpGet.port}
              scheme: ${addresses-adapter.ose.deployment.spec.template.spec.cont
ainers.addresses-adapter.readinessProbe.httpGet.scheme}
            failureThreshold: ${addresses-adapter.ose.deployment.spec.template.s
pec.containers.addresses-adapter.readinessProbe.failureThreshold}
            initialDelaySeconds: ${addresses-adapter.ose.deployment.spec.templat
e.spec.containers.addresses-adapter.readinessProbe.initialDelaySeconds}
            periodSeconds: ${addresses-adapter.ose.deployment.spec.template.spec
.containers.addresses-adapter.readinessProbe.periodSeconds}
            successThreshold: ${addresses-adapter.ose.deployment.spec.template.s
pec.containers.addresses-adapter.readinessProbe.successThreshold}
            timeoutSeconds: ${addresses-adapter.ose.deployment.spec.template.spe
c.containers.addresses-adapter.readinessProbe.timeoutSeconds}
          resources:
            limits:
              cpu: ${addresses-adapter.ose.deployment.spec.template.spec.contain
ers.addresses-adapter.resources.limits.cpu}
              memory: ${addresses-adapter.ose.deployment.spec.template.spec.cont
ainers.addresses-adapter.resources.limits.memory}
            requests:
              cpu: ${addresses-adapter.ose.deployment.spec.template.spec.contain
ers.addresses-adapter.resources.requests.cpu}
              memory: ${addresses-adapter.ose.deployment.spec.template.spec.cont
ainers.addresses-adapter.resources.requests.memory}
          terminationMessagePath: /dev/termination-log
```

```
          volumeMounts:
            - mountPath: /etc/config/ssl/
              name: ufs-mq-jks-vol
            - mountPath: /opt/conf/ufsparams/config
              name: ufsparams-config-volume
            - mountPath: /tmp/ufsparams/config
              name: addresses-adapter-config-cm
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      securityContext: {}
      terminationGracePeriodSeconds: 30
      volumes:
        - name: ufs-mq-jks-vol
          secret:
            defaultMode: 0440
            secretName: mq.jks
        - name: ufsparams-config-volume
          emptyDir: {}
        - name: addresses-adapter-config-cm
          configMap:
            name: addresses-adapter-config-cm.${distrib.release.version}
  triggers:
    - type: ConfigChange
```

Далее проверьте этот манифест и замените строки, содержащие *${}*или *{{}}*. Так как efc pipline поддерживает различные механизмы шаблонизации, то могут быть переменные типа .$, которые не были заменены.

Значение текущей переменной замените на unver, выполнив команду:

```
Kubectl apply -f <ваш обработанный манифест > -n <пространство имен в k8s>
```

Для успешного разворачивания любого сервиса разверните все манифесты в директории сервиса.

Порядок установки и первоначальной настройки при использовании инструментов CI/CD

В данном разделе описана установка дистрибутива командой сопровождения при помощи средств CI/CD.

Версия продукта выпускается под Kubernetes (или же OpenShift) и устанавливается с помощью подготовленных Job системы CI/CD Jenkins.

Продукт не нуждается в специальной настройке, влияющей на безопасность данных. Настройки безопасности осуществляются за пределами конфигурации сервиса (см. описание механизмов безопасности в документе «Руководство по безопасности»).

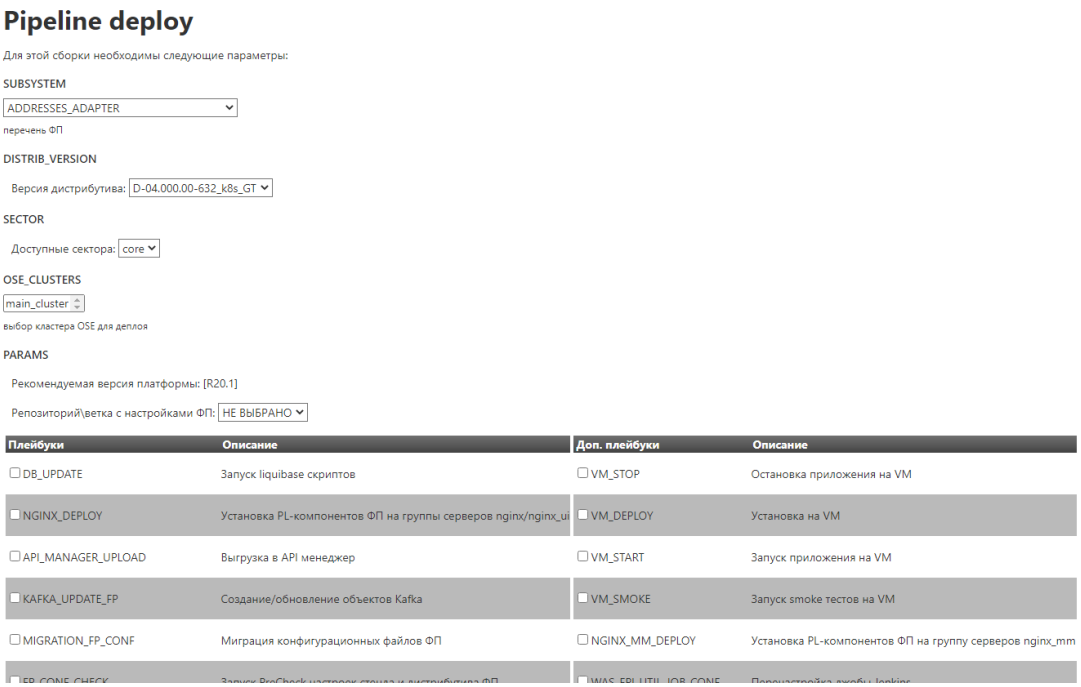
Все настраиваемые параметры продукта описаны в разделе «Настраиваемые параметры» документа «Руководство по системному администрированию».

Для установки и первоначальной настройки продукта Platform V Addresses через Pipeline нужно выполнить следующие действия:

1. Открыть приложение **Jenkins**.
2. В каталоге выбрать нужную job и в меню слева выбрать раздел **Собрать с параметрами**.
3. На открывшейся странице выполнить следующие настройки:
   - **SUBSYSTEM** – указать компонент, который необходимо установить: ADDRESSES_FINDER, ADDRESSES_PARSER_MANAGER, ADDRESSES_MANAGER, ADDRESSES_ADAPTER.
   - **DISTRIB_VERSION** – указать версию дистрибутива.
   - **OSE_CLUSTER** – указать кластер среды контейнеризации (например, main_cluster).
   - **PARAMS** – установить дополнительные параметры: **BRANCH:R20.1**, **OPENSHIFT_DEPLOY:true** и т.д.

   Установить флаги для playbook, которые должны будут выполнены при установке сервиса:

Название	Описание
CLEANUP_FP_CONFIG	Очистка конфигурационного репозитория ФП
MIGRATION_FP_CONF	Миграция новых конфигурационных файлов ФП
FP_CONF_CHECK	Проверка конфигурационных файлов ФП
DB_UPDATE	Запуск Liquibase-скриптов
NGINX_DEPLOY	Установка PL-компонентов ФП на группы серверов nginx
API_MANAGER_UPLOAD	Загрузка настроек nginx в api manager
UPDATE_KAFKA_FP	Добавление топиков Kafka
OPENSHIFT_DEPLOY	Установка в Kubernetes или OpenShift (название плейбука одинаково для обоих оркестраторов)
OPENSHIFT_INGRESS_EGRESS_DEPLOY	Деплой ingress/egress в Kubernetes или OpenShift (название плейбука одинаково для обоих оркестраторов)
IMPORT_ALL_PARAMS	Импорт параметров сервисов

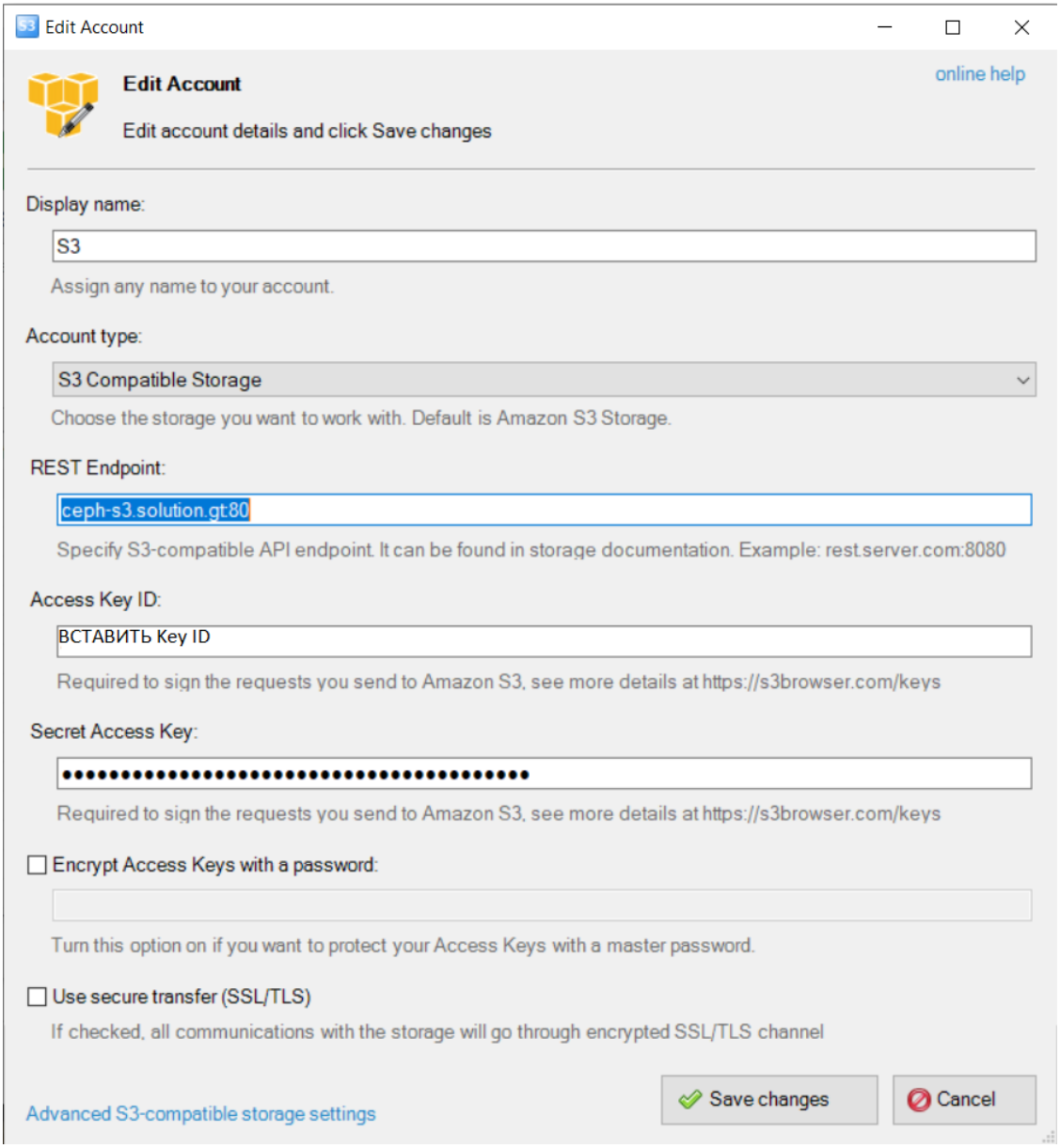


4. Для запуска процесса установки нажать кнопку **Запустить**.

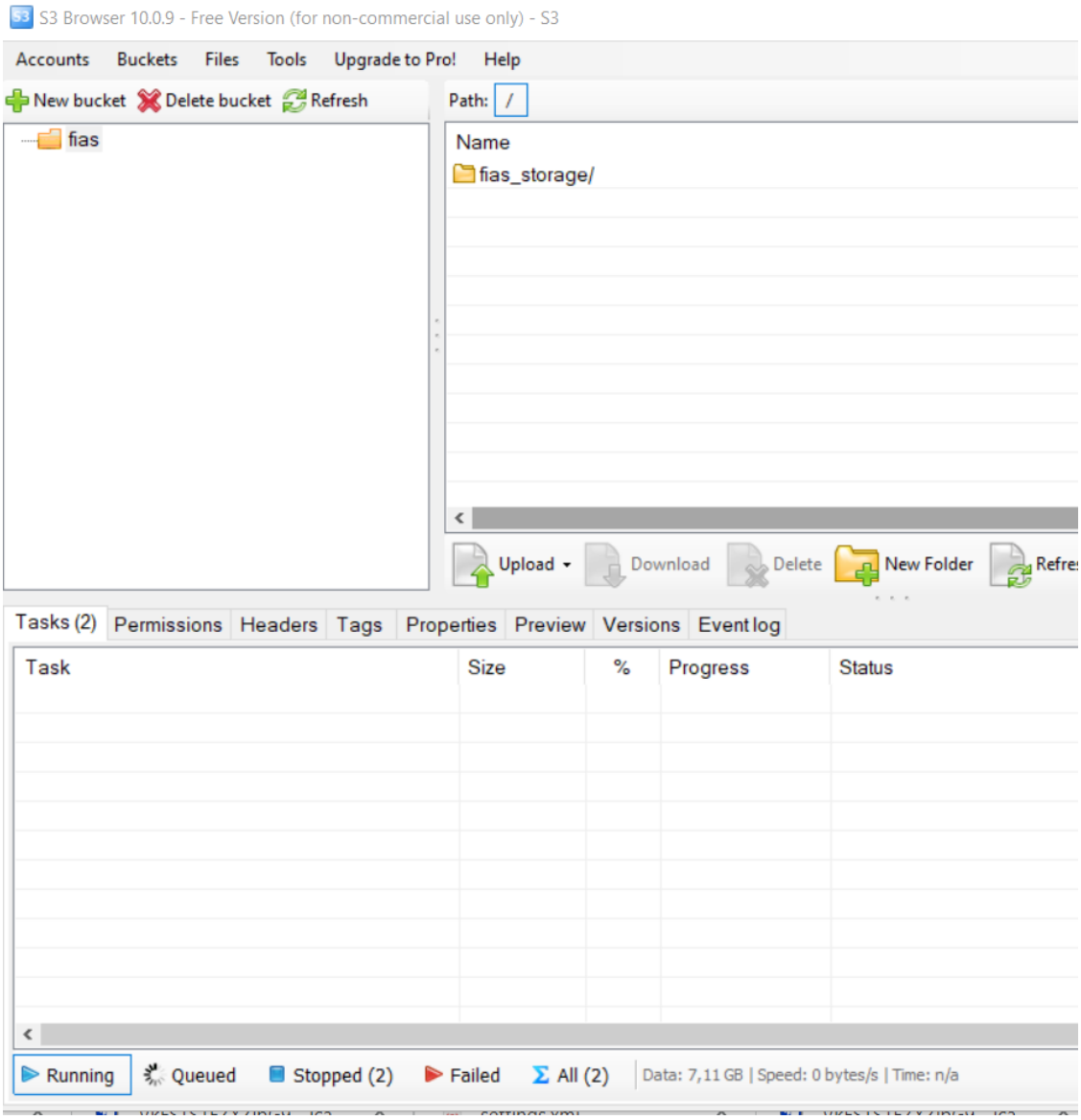
При развертывании из компонента Управление параметрами Platform V Configuration определяются несколько baseurl:

- OpenSearch (или Platform V Index Search).
- Хранилища S3.

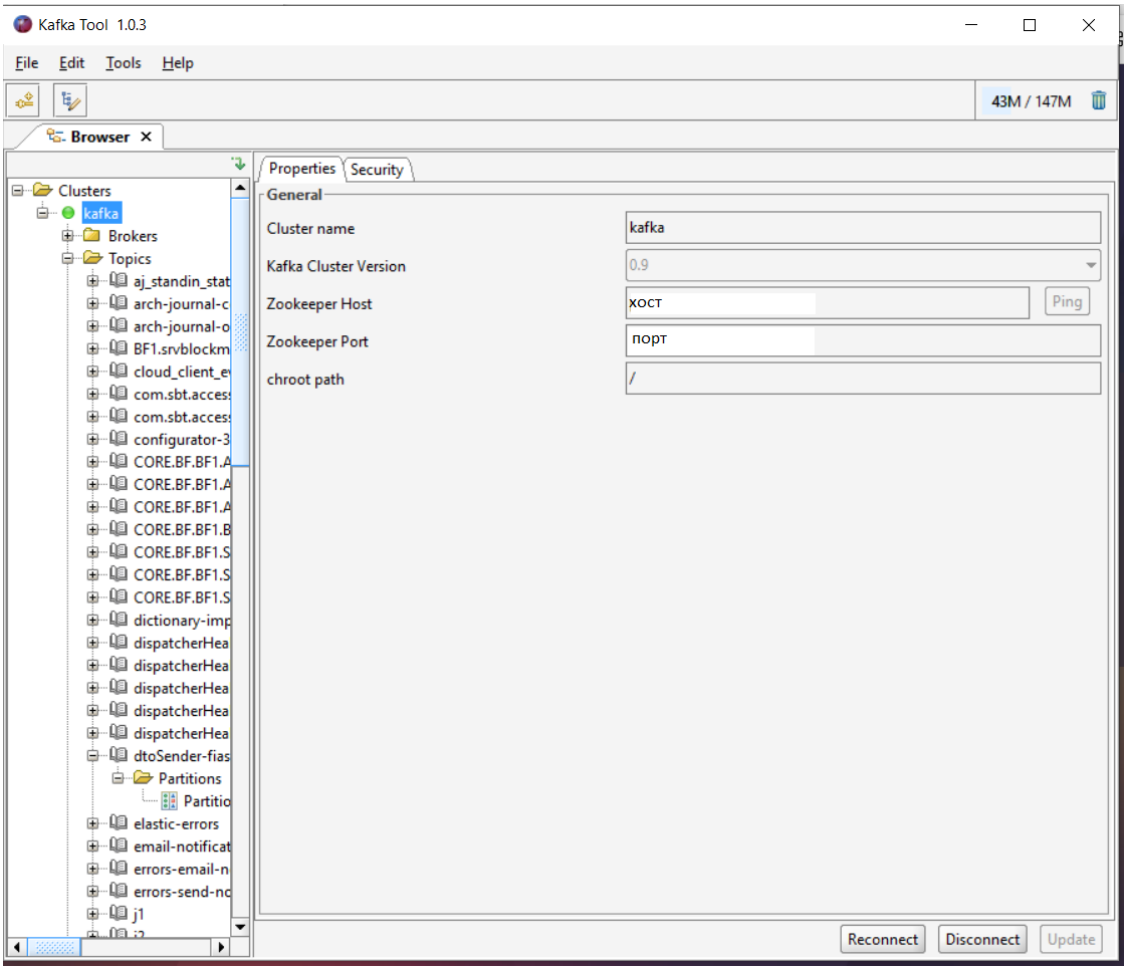
Подключение к хранилищу S3:



После успешного подключения появится главное меню:



Подключение к Kafka:



Проверка корректности установки

Для проверки корректности установки необходимо выполнить следующие действия:

1. Убедиться, что задача установки завершилась со статусом «Успешно».
2. Зайти в web-панель администратора среды контейнеризации и убедиться, что все POD обновлены и работоспособны (имеют статус «ready» для всех контейнеров, входящих в POD).
3. Для проверки работоспособности ВН необходимо пройти по URL:
   - для addresses-finder: http://:/addresses-finder/healthcheck;
   - для addresses-manager: http://:/addresses-manager/healthcheck;
   - для addresses-parser: http://:/addresses-parser/healthcheck;
   - для addresses-adapter: http://:/addresses-adapter/healthcheck. При успехе должен вернуться ответ следующего вида: *{“success”:true,“body”:Status UP}*.

Проверка выполнения плейбука DB_UPDATE:

1. Убедиться, что плейбук завершился со статусом «Успешно».
2. Подключиться к БД, для которой были выполнены скрипты Liquibase.

3. Убедиться, что были созданы следующие таблицы: fias_sync_parser, fias_sync_adapter, fias_loader_history_upload, fias_loader_history_download.

Проверка выполнения плейбука NGINX_DEPLOY:

1. Зайти на шлюз IAG_MM и убедиться, что для модуля addresses-finder была настроена переадресация.
2. Зайти на шлюз IAG_UI и убедиться, что для модуля addresses-manager была настроена переадресация.

Проверка выполнения плейбука API_MANAGER_UPLOAD: зайти в арм api-manager и убедиться, что для модулей addresses-finder и addresses-manager были созданы конфигурации.

Проверка выполнения плейбука UPDATE_KAFKA_FP: подключиться к стендовой kafka и убедиться, что были созданы топики: dtoSender-fiasAdapter, dtoSender-fiasLoader.

Обновление

Обновление при ручном способе установки

Обновление осуществляется также, как первоначальная установка: запускается задача установки с параметром `updateMode` равным *fullInstall*. В процедуру установки включен процесс обновления базы данных с помощью утилиты Liquibase.

Установку новой версии продукта необходимо осуществлять согласно описанным шагам в разделе «Порядок установки и первоначальной настройки при ручной установке».

Дополнительных настроек не требуется.

Обновление при использовании инструментов CI/CD

Версия устанавливается с помощью подготовленных Job системы CI/CD Jenkins.

Обновление заключается в:

1. Установке версии в среде контейнеризации (процесс установки см. в разделе «Установка»).
2. Переключении в inventory трафика на Kubernetes или OpenShift. Данное действие осуществляется системным администратором в файле inventory дистрибутива.

Перед обновлением продукта в первую очередь обеспечивается обновление БД с применением скриптов Liquibase. Обновление БД осуществляется силами администраторов стенда.

При обновлении компонентов продукта в среде контейнеризации рекомендуется использовать стратегию Rolling.

Проверка работоспособности

После установки сервиса в оркестратор за его работоспособностью и возможностью обрабатывать запросы будет следить сам оркестратор через liveness и readiness пробы, которые разработаны с использованием сервиса HealthCheck.

Также можно вручную проверить работоспособность сервиса, отправив REST-запрос и получив в ответе состояние сервиса.

GET-запрос на примере компонента addresses-manager: *http://localhost:8080/addresses-manager/healthcheck*. Ответ сервиса: *Status UP*.

Также можно получить мета-информацию (версию, подсистему, ip-адрес) о сервисе, отправив следующий GET-запрос (пример для addresses-finder): *http://localhost:8080/addresses-finder/environment/product*.

Ответ сервиса:

{“success”:true,“body”:{“subsystem”:“ADDRESSES”,“channel”:“SUPPORT”,“deploymentUnit”:“addresses-finder”,“version”:“”,“distribVersion”:“”,“platform”:“”,“serverIp”:“”,“release”:“”}}.

Откат

Откат производится аналогично установке новой версии (см. раздел «Порядок установки и первоначальной настройки»). При этом необходимо указать требуемую версию дистрибутива в выпадающем списке параметра **DISTRIB_VERSION** Pipeline deploy Jenkins.

Часто встречающиеся проблемы и пути их устранения

На текущий момент не имеется информации о встречающихся проблемах.

Чек-лист валидации установки

Выполнены условия:

1. Проверено наличие хранилища S3.
2. Имеются все необходимые смежные компоненты, описанные в разделе «Требования к среде установки».
3. Пришел успешный ответ от /environment/product (см. описание в разделе «Проверка работоспособности»).
4. Пришел успешный ответ от healthcheck (см. описание в разделе «Проверка работоспособности»).