



**Руководство по установке**

**Продукта Platform V Synapse Enterprise Integration (SEI)**

## ОГЛАВЛЕНИЕ

Руководство по установке компонента JDBC Gateway (JDBC).....	4
Термины и определения.....	4
Назначение документа .....	5
Системные требования.....	6
Установка .....	7
Обновление.....	7
Удаление .....	8
Проверка работоспособности .....	10
Откат.....	11
Часто встречающиеся проблемы и пути их устранения .....	12
Чек-лист валидации установки .....	14
Руководство по установке компонента Kafka Gateway (KFGT).....	15
Термины и определения.....	15
Системные требования.....	16
Установка .....	17
Чек-лист валидации установки .....	18
Обновление.....	19
Откат.....	20
Проверка работоспособности .....	21
Часто встречающиеся проблемы и пути их решения .....	22

Руководство по установке компонента Unimapper (UNIM) .....	24
Термины и определения .....	24
Описание .....	25
Системные требования.....	26
Минимальные ресурсы .....	27
Требования к КТС .....	28
Установка .....	28
Обновление .....	34
Проверка работоспособности .....	34
Откат.....	35
Часто встречающиеся проблемы и пути их устранения .....	35
Чек-лист валидации установки .....	36

# Руководство по установке компонента JDBC Gateway (JDBC)

## Термины и определения

Термин/Аббревиатура	Определение
Деплоймент	Файл, содержащий набор инструкций для запуска приложения в Kubernetes
Под	Абстрактный объект, набор контейнеров в архитектуре Kubernetes
Liveness-проба	Программный зонд для определения живучести контейнера
Лимит	Максимальный объем ресурсов, который выдается на работу сервиса
Реквест	Объем ресурсов, который необходим для запуска приложения

Термин/Аббревиатура	Определение
БД	База данных
URL	Uniform Resource Locator, унифицированный указатель ресурса
CPU	Central Processing Unit, центральный процессор
RAM	Random Access Memory, оперативная память

## Назначение документа

Настоящий документ содержит руководство по установке Компонент «JDBC Gateway» программного продукта Platform V Synapse Enterprise Integration (далее так же «Шлюз БД»), который предоставляет облачным приложениям возможность вызова процедур, хранящихся во внешней БД.

### Важно!

Компонент «Шлюз БД» поставляется в виде собранного Docker-образа, размещаемого в целевом Docker-репозитории, и предназначен для использования в составе прикладных интеграционных решений, разрабатываемых продуктовыми командами.

Для этого команды, реализующие прикладную интеграцию, выпускают дистрибутив, содержащий набор конфигурационных артефактов Kubernetes, обеспечивающих развертывание и настройку отдельного экземпляра Шлюза БД под требования этой интеграции.

Компонент «Шлюз БД» не предназначен для самостоятельной эксплуатации вне рамок прикладных интеграционных решений.

## Системные требования

Для установки компонента «Шлюз БД» в составе прикладного дистрибутива требуется соблюдение следующих условий:

- Развернут и настроен кластер Kubernetes в соответствии с требованиями, предъявляемыми к Платформе.
- В кластере создан проект (namespace), в котором будет развернут шлюз.
- В проекте создана учетная запись с правами на загрузку артефактов.
- Docker-образ шлюза размещен в целевом Docker-репозитории, ссылка на который указана в конфигурационном артефакте Деплоймента.
- В проект добавлен Secret для загрузки Docker-образов из целевого Docker-репозитория.
- Подготовлен комплект настроенных конфигурационных артефактов.
- Сервера БД настроены и готовы к работе, к ним настроен доступ со стороны кластера Kubernetes.
- В БД созданы необходимые процедуры и к ним предоставлен соответствующий доступ.
- В проекте имеются свободные ресурсы по лимитам и реквестам: не менее, чем зарезервировано в конфигурационных артефактах.
- Для установки с использованием консоли на рабочем месте должен быть установлен клиент Kubernetes.

Репозиторий размещения Docker-образа Шлюза БД:

Минимальные ресурсы, необходимые для запуска компонента «Шлюз БД»:

Контейнер	CPU Request, millicores	Memory Request, mebibytes	CPU Limit, millicores	Memory Limit, mebibytes
Приложение	1000	1500	1000	1500

Перечень программных продуктов, используемых при установке:

Наименование	Функции
Kubernetes Client	Подключение к проекту Kubernetes, загрузка артефактов конфигурации в проект Kubernetes в консольном режиме
Kubernetes Web Console	Подключение к проекту Kubernetes, загрузка артефактов конфигурации в проект Kubernetes в интерактивном режиме

При необходимости использования компонента «Сервисный прокси» (SVPX) продукта «Platform V Synapse Service Mesh» (SSM), информацию о минимальных ресурсах, необходимых для его запуска, следует уточнить в документации на этот компонент.

## Установка

1. Войдите в учетную запись необходимого проекта:  
`kubect1 login --server=<url кластера> kubect1 project <имя проекта>`
2. Установите компоненты, необходимые для работы адаптера (`oc apply -f <имя компонента>`):
  - Deployment;
  - Service;
  - ServiceEntry с точками подключения к серверам БД;
  - ConfigMap с настройками шлюза;
  - Secret с хранилищем сертификатов и настройками подключения к БД (пароли от хранилищ, тип шифрования и т. д.).

## Обновление

1. Перед обновлением требуется сохранить бэкап предыдущей конфигурации:

```
kubect1 get deployment <deployment-name> > depl.yml
```

```
kubectl get service <service-name> > svc.yml
kubectl get se <serviceentry-name> > se.yml
kubectl get configmap <configmap-name> > cm.yml
kubectl get secret <keystore-secret-name> > keystore.yml
kubectl get secret <secret-name> > secret.yml
```

2. Удалить все предыдущие ресурсы шлюза:

```
kubectl delete deployment <deployment-name>
kubectl delete service <service-name>
kubectl delete se <serviceentry-name>
kubectl get configmap <configmap-name>
kubectl delete secret <keystore-secret-name>
kubectl delete secret <secret-name>
```

3. Провести установку по инструкции из раздела **Установка**.

## Удаление

1. Остановите шлюз:

**С использованием веб-интерфейса**

Шаг	Действия	Описание
Остановка шлюза	<ol style="list-style-type: none"><li>1. В меню выберите пункт <b>Workload -&gt; Deployments</b>.</li><li>2. На странице найдите нужный Деплоймент (при необходимости воспользуйтесь поиском по имени).</li><li>3. Перейдите по ссылке в наименовании на вкладку <b>Detail</b>.</li><li>4. Стрелкой вниз (<b>Decrease the pod count</b>) уменьшите количества Подов шлюза до 0.</li></ol>	-



### С использованием консоли

Шаг	Действия	Описание
Остановка шлюза	В консоли выполните команду <code>kubectl scale --replicas=0 deployment/&lt;имя Деплоймента&gt;</code>	-

1. Удалите артефакты действующей версии:

### С использованием веб-интерфейса

Шаг	Действия	Описание
Удаление Деплоймента	<ol style="list-style-type: none"><li>1. В меню выберите пункт <b>Workload -&gt; Deployments</b>.</li><li>2. На странице найдите нужный Деплоймент (при необходимости воспользуйтесь поиском по имени).</li><li>3. Перейдите по ссылке в наименовании на вкладку <b>Detail</b>.</li><li>4. Раскройте меню <b>Action</b> в правом верхнем углу окна, выберите пункт <b>Delete Deployment</b>.</li><li>5. В появившемся окне подтвердите действие.</li></ol>	-
Удаление конфигурации	<ol style="list-style-type: none"><li>1. В меню выберите пункт <b>Workload -&gt; Config Maps</b>.</li><li>2. На странице найдите нужный артефакт (при необходимости воспользуйтесь поиском по имени).</li></ol>	Если конфигурация

Шаг	Действия	Описание
	3. Перейдите по ссылке в наименовании на вкладку <b>Detail</b> . 4. Раскройте меню <b>Action</b> в правом верхнем углу окна, выберите пункт <b>Delete Config Map</b> . 5. В появившемся окне подтвердите действие.	была изменена

### *С использованием консоли*

Шаг	Действия	Описание
Удаление Деплоймента	В консоли выполните команду: <code>kubectl delete deployment &lt;имя Деплоймента&gt;</code>	-
Удаление конфигурации	В консоли выполните команду: <code>kubectl delete configmap &lt;имя config map&gt;</code>	Если была изменена конфигурация

## Проверка работоспособности

### *С использованием веб-интерфейса*

Шаг	Действие
Проверить работоспособность	<ol style="list-style-type: none"> <li>1. В меню выберите пункт <b>Workload</b> → <b>Pods</b>.</li> <li>2. На странице найдите нужный Под шлюза.</li> <li>3. Перейдите по ссылке в имени на вкладку <b>Detail</b>.</li> <li>4. Перейдите на вкладку <b>Terminal</b>.</li> <li>5. В терминале Пода выполните команду: <code>curl localhost:&lt;portnum&gt;/actuator/health</code>, где &lt;portnum&gt; — номер порта, указанный в параметре <code>server/port</code> конфигурации шлюза. В ответном сообщении должна быть строка: <code>"ping":{"status":"UP"}</code></li> </ol>

### **С использованием консоли**

Шаг	Действие
Проверить работоспособность	<ol style="list-style-type: none"> <li>1. В консоли задайте команду: <code>kubectl port-forward pod/&lt;имя пода&gt; &lt;portnum&gt;:&lt;portnum&gt;</code>, где &lt;portnum&gt; — номер порта, указанный в параметре <code>server/port</code> конфигурации шлюза.</li> <li>2. Запустите еще одно окно консоли и в нем выполните команду: <code>curl localhost:&lt;portnum&gt;/actuator/health</code>. В ответном сообщении должна быть строка: <code>"ping":{"status":"UP"}</code>.</li> <li>3. Завершите перенаправление портов нажатием <b>Ctrl+C</b></li> </ol>

## **Откат**

1. Удалить все предыдущие ресурсы адаптера:

```
kubectl delete deployment <deployment-name>
kubectl delete service <service-name>
kubectl delete se <serviceentry-name>
kubectl delete configmap <configmap-name>
kubectl delete secret <keystore-secret-name>
kubectl delete secret <secret-name>
```

2. Провести установку бэкапа по инструкции из раздела **Установка**.

## Часто встречающиеся проблемы и пути их устранения

Проблема	Причина	Решение
Ошибка при подключении к проекту	Пользователю не предоставлен доступ в проект	Запросите доступ к проекту
Ошибка при подключении к проекту	Нет физического доступа к кластеру	Зарегистрируйте обращение в поддержку для восстановления доступа
Ошибка при загрузке артефактов	У пользователя, от имени которого производится загрузка, отсутствуют необходимые права	Проверьте и скорректируйте настройку прав доступа пользователя

Проблема	Причина	Решение
Ошибка при загрузке Docker-образа из репозитория	Отсутствуют права на загрузку образа из репозитория	Запросите права
Ошибка при загрузке Docker-образа из репозитория	Недоступен репозиторий	Зарегистрируйте обращение в поддержку для восстановления доступа
Ошибка при загрузке Docker-образа из репозитория	Неверная ссылка на имя образа Docker-контейнера в Деплойменте шлюза	Проверьте ссылку, при необходимости скорректируйте Деплоймент
Отсутствует подключение к БД	Установленная configmap является некорректной	Проверьте конфигурационные файлы, найдите и устранили опечатки
Не видны необходимые процедуры	Установленная configmap является некорректной	Проверьте конфигурационные файлы, найдите и устранили опечатки
Ошибка TCPS-подключения	Загруженные Secret некорректны	Проверьте конфигурационные файлы, найдите и устранили опечатки

# Чек-лист валидации установки

Для проверки валидности установки проверьте наличие всех установленных компонентов:

```
kubectl get deployment <deployment-name>  
kubectl get service <service-name>  
kubectl get se <serviceentry-name>  
kubectl get configmap <configmap-name>  
kubectl get secret <keystore-secret-name>  
kubectl get secret <secret-name>
```

Все компоненты должны присутствовать в проекте и должна отсутствовать ошибка <<Error from server (NotFound)>>.

# Руководство по установке компонента Kafka Gateway (KFGT)

## Термины и определения

Термин/аббревиатура	Определение
CPU	Central Processing Unit, центральный процессор
gRPC	Высокопроизводительный фреймворк, разработанный компанией Google для вызова удаленных процедур (RPC)
Под	Абстрактный объект, набор контейнеров в архитектуре Kubernetes
Лимит	Максимальный объем ресурсов, который выдается на работу сервиса
Реквест	Объем ресурсов, который необходим для запуска приложения

Термин/аббревиатура	Определение
URL	Uniform Resource Locator, унифицированный указатель ресурса

### Важно!

Программный компонент Kafka Gateway Продукта Platform V Synapse Enterprise Integration (далее — Шлюз Kafka) поставляется в виде собранного Docker-образа, размещаемого в целевом Docker-репозитории, и предназначается для использования в составе прикладных интеграционных решений, разрабатываемых продуктовыми командами.

Для этого команды, реализующие прикладную интеграцию, выпускают дистрибутив, содержащий набор конфигурационных артефактов Kubernetes или Red Hat OpenShift, обеспечивающих развертывание и настройку отдельного экземпляра Шлюза Kafka под требования этой интеграции.

Самостоятельно, вне рамок прикладных интеграционных решений, Шлюз Kafka не эксплуатируется.

## Системные требования

Для установки Шлюза Kafka в составе прикладного дистрибутива должно быть выполнено следующее:

1. Развернут и настроен кластер Kubernetes 1.17+ или Red Hat OpenShift 4.5.13+ в соответствии с требованиями, предъявляемыми к Платформе.
2. В кластере создан проект (namespace), в котором будет разворачиваться шлюз.
3. В проекте создана учетная запись с правами на загрузку артефактов.
4. Docker-образ шлюза размещен в целевом Docker-репозитории по ссылке, указанной в конфигурационном артефакте Deployment.
5. В проект добавлен Secret для загрузки Docker-образов из целевого Docker-репозитория.
6. Подготовлен комплект настроенных конфигурационных артефактов.
7. Брокеры Kafka настроены и готовы к работе, есть доступ к ним со стороны кластера Kubernetes или Red Hat OpenShift.
8. Созданы целевые топики сообщений, необходимый доступ к ним предоставлен.
9. В проекте имеются свободные ресурсы по лимитам и реквестам не менее, чем зарезервировано в конфигурационных артефактах.
10. При установке с использованием консоли, на рабочем месте должен быть установлен клиент Kubernetes или Red Hat OpenShift.



Минимальные ресурсы, необходимые для запуска Шлюза Kafka

Контейнер	CPU Request	Memory Request	CPU Limit	Memory Limit
Приложение	400m	512Mi	400m	512Mi

Перечень программных продуктов, используемых при установке:

Наименование	Функции
Kubernetes или Red Hat OpenShift Client	Подключение к проекту Kubernetes или Red Hat OpenShift, загрузка артефактов конфигурации в проект в консольном режиме
Kubernetes или Red Hat OpenShift Web Console	Подключение к проекту Kubernetes или Red Hat OpenShift, загрузка артефактов конфигурации в проект в интерактивном режиме

При опциональном использовании совместно со Шлюзом Kafka Сайджара компонента Сервисный прокси (входящего в состав продукта «Platform V Synapse Service Mesh») минимальные ресурсы, необходимые для запуска, следует уточнить в соответствующей документации используемого компонента.

## Установка

Для установки Шлюза Kafka:

1. Войдите в необходимый проект:

```
kubectl login --server=<url кластера>  
kubectl project <имя проекта>
```

1. Установите необходимые для работы адаптера компоненты:

```
kubectl apply -f <имя компонента>
```

Список компонентов:

- Deployment;
- Service;
- ServiceEntry с точками подключения к брокерам Kafka;
- ConfigMap с настройками шлюза;
- Secret с хранилищем сертификатов;
- Secret с паролями.

## Чек-лист валидации установки

Для проверки валидности установки проверьте наличие всех установленных компонентов:

```
kubectl get deployment <deployment-name>  
kubectl get service <service-name>  
kubectl get se <serviceentry-name>  
kubectl get configmap <configmap-name>  
kubectl get secret <keystore-secret-name>  
kubectl get secret <secret-name>
```

Все компоненты должны присутствовать в проекте и должна отсутствовать ошибка "Error from server (NotFound)".

Инструкция по проверке работоспособности Шлюза Kafka представлена в разделе **Проверка работоспособности**.

# Обновление

1. Перед обновлением требуется сохранить бэкап предыдущей конфигурации:

```
kubectl get deployment <deployment-name> > depl.yml  
kubectl get service <service-name> > svc.yml  
kubectl get se <serviceentry-name> > se.yml  
kubectl get configmap <configmap-name> > cm.yml  
kubectl get secret <keystore-secret-name> > keystore.yml  
kubectl get secret <secret-name> > secret.yml
```

1. Удалить все предыдущие ресурсы адаптера:

```
kubectl delete deployment <deployment-name>  
kubectl delete service <service-name>  
kubectl delete se <serviceentry-name>  
kubectl get configmap <configmap-name>  
kubectl delete secret <keystore-secret-name>  
kubectl delete secret <secret-name>
```

1. Провести установку по инструкции из раздела *Установка*.

Если требуется обновление без недоступности, то необходимо проводить канареечную установку, исходя из таблицы:

<b>Решение по повышению отказоустойчивости Kafka-адаптера</b>	<b>Тип воздействия</b>	<b>Тип воздействия</b>
<b>Режим работы Kafka адаптера</b>	<b>Вывод плеча кластера Synapse из под нагрузки</b>	<b>Обновление адаптера</b>

Решение по повышению отказоустойчивости Kafka-адаптера	Тип воздействия	Тип воздействия
Запросы в адаптер поступают по gRPC, адаптер обрабатывает их и пересылает в топик	Решается переводом нагрузки на балансировщике	Решается канареечной установкой. Нагрузка на новые Поды заводится стандартно с помощью задачи Jenkins
Адаптер читает сообщения из топика, обрабатывает их и пересылает по gRPC	Решается путем остановки Подов адаптера на плече, с которого нужно снять нагрузку	Решается канареечной установкой. Нагрузка начинает идти через новый Под после его старта. Поды со старой версией нужно остановить

## Откат

1. Удалить все предыдущие ресурсы адаптера:

```
kubectl delete deployment <deployment-name>
kubectl delete service <service-name>
kubectl delete se <serviceentry-name>
kubectl delete configmap <configmap-name>
kubectl delete secret <keystore-secret-name>
kubectl delete secret <secret-name>
```

2. Провести установку бэкапа по инструкции из раздела *Установка*.

## Проверка работоспособности

### С использованием веб-интерфейса

Шаг	Действие	Примечание
Проверить работоспособность	В меню выбрать пункт Workload/Pods. На странице найти нужный pod шлюза. Пройти по ссылке в имени на вкладку Detail, перейти на вкладку Terminal. В терминале пода выполнить команду: curl localhost:<portnum>/actuator/health В выдаче команды должна быть строка: "ping":{"status":"UP"}	<portnum> - номер порта указанный в параметре server/port конфигурации шлюза

### С использованием консоли

Шаг	Действие	Примечание
Проверить работоспособность	В консоли выполнить команду: kubectl port-forward pod/<имя Пода> <portnum>:<portnum> Запустить еще одно окно консоли, в нем выполнить команду: curl localhost:<portnum>/actuator/health В выдаче команды должна быть строка: "ping":{"status":"UP"} Завершить форвардинг портов нажатием Ctrl+C	<portnum> - номер порта указанный в параметре server/port конфигурации шлюза

## Часто встречающиеся проблемы и пути их решения

Проблема	Причина	Решение
Ошибка при подключении к проекту	Пользователю не предоставлен доступ в проект	Запросить доступ к проекту
Ошибка при подключении к проекту	Нет физического доступа к кластеру	Зарегистрировать обращение в поддержку для восстановления доступа
Ошибка при загрузке артефактов	У пользователя, под которым производится загрузка, отсутствуют необходимые права	Проверить и скорректировать настройку прав доступа пользователю
Ошибка при получении Docker-образа из репозитория	Отсутствуют права на Pull образа из репозитория	Запросить права
Ошибка при получении Docker-образа из репозитория	Недоступен репозиторий	Зарегистрировать обращение в поддержку для восстановления доступа
Ошибка при получении Docker-образа из репозитория	Неверная ссылка на имя образа Docker-контейнера в Деплойменте шлюза	Проверить ссылку, при необходимости скорректировать Деплоймент

Проблема	Причина	Решение
Отсутствует подключение к Kafka	Установленная configmap является некорректной	Проверить конфигурационные файлы, найти и устранить опечатки в имени топика, хоста и/или порта вызываемого сервиса и/или брокера Kafka
Отсутствует подключение к Kafka	Установленная configmap является некорректной	Проверить конфигурационные файлы, найти и устранить опечатки в имени топика, хоста и/или порта вызываемого сервиса и/или брокера Kafka
Отсутствует подключение к Kafka	Проверить конфигурационные файлы, найти и устранить опечатки в имени топика, хоста и/или порта вызываемого сервиса и/или брокера Kafka	

Чтение журналов всех контейнеров Пода помогает найти причину и решение большинства проблем при установке.

# Руководство по установке компонента Unimapper (UNIM)

## Термины и определения

Аббревиатура/сокращение	Описание
HTTP	Расширение протокола HTTP для поддержки шифрования в целях повышения безопасности
JSON	JavaScript Object Notation — текстовый формат обмена данными, основанный на JavaScript
POST	Метод запроса, поддерживаемый HTTP-протоколом, при котором веб-сервер принимает данные, заключенные в тело сообщения, для хранения
XML	eXtensible Markup Language — расширяемый язык разметки
XSLT	eXtensible Stylesheet Language Transformations — язык преобразования XML-документов



Аббревиатура/сокращение	Описание
Platform V Synapse Service Mesh	Программный продукт на базе Istio SE, обеспечивающий возможность создания сервисной сети поверх Платформенной в Kubernetes
Deployment/Деплоймент	Набор инструкций для запуска приложения в Kubernetes
Unimapper / UNIM	Компонент Unimapper продукта Platform V Synapse Enterprise Integration
Сервисный прокси / SVPX	Компонент Сервисный прокси продукта Platform V Synapse Service Mesh
Граничный прокси / IGEG	Компонент Граничный прокси продукта Platform V Synapse Service Mesh

## Описание

Данное руководство предназначено для пользователей, устанавливающих компонент Unimapper.

## Системные требования

Для функционирования компонента UNIM предъявляются требования по наличию следующего программного обеспечения:

Наименование ПО	Версия ПО	Назначение ПО
Kubernetes	1.19 и выше	Платформа оркестрации приложений с средствами автоматизации и управления на основе политик

Для возможности работы компоненты UNIM рекомендуется развернуть следующие компоненты Platform V

Наименование компонента Platform V	Код компонента Platform V	Назначение компонента Platform V
Сервисный прокси	SVPX	Сервисный прокси из состава продукта Platform V Synapse Service Mesh (далее Synapse) предназначен для предоставления базовых интеграционных операций прикладной части интеграционного сценария-сервиса. Сервисный прокси используется для маршрутизации и обеспечения безопасности трафика между

Наименование компонента Platform V	Код компонента Platform V	Назначение компонента Platform V
		приложениями.

Компонент UNIM также может работать без привязки к компонентам Platform V

Перед установкой проверьте соблюдение следующих условий:

- Развернутый и настроенный кластер Платформы (Kubernetes 1.19 и выше)
- В кластере создан проект (namespace) для развертывания контрольной панели Platform V Synapse Service Mesh платформы (Istio SE)
- ОС: Linux (Рекомендован Альт 8СП)
- В проекте создана учетная запись с правами на загрузку артефактов (администратор проекта).
- Свободные в проекте ресурсы, по лимитам и реквестам по размеру равные или превышающие зарезервированные в конфигурационных артефактах.

## Минимальные ресурсы

Минимальные ресурсы, необходимые для компонента UNIM:

Сервис	CPU	Memory	REQUESTS_CPU	REQUESTS_MEMORY
unimapper	200m	200Mb	200m	200m

Сервис	CPU	Memory	REQUESTS_CPU	REQUESTS_MEMORY
istio-proxy	200m	200m	200m	200m

Количество задействованных реплик компонента UNIM зависит от объема принимаемых данных.

## Требования к КТС

### Требования к техническим компонентам

Требования к КТС — это размер ресурсов CPU, памяти, выделяемых в Kubernetes. Их значение вычисляется для конкретного разворачиваемого инстанса в зависимости от профиля нагрузки.

### Принципы размещения сервиса на КТС

Компонент UNIM разворачивается в виде контейнера в отдельном Деплойменте.

**Специфические технологические решения:** Отсутствуют.

## Установка

### Состав дистрибутива

Дистрибутив содержит файлы:

- Deployment.yaml (unimapper)

- ConfigMap.yaml (unimapper-cm)
- Service.yaml (unimapper-svc)

Перед установкой конфигураций необходимо пройти авторизацию в кластере Kubernetes и сгенерировать токен.

1. Зайдите в командную оболочку.

2. Создайте файл конфигурации для kubectl

```
kubectl config set-credentials логин/хост:порт --username=логин --password=пароль
```

```
kubectl config set-cluster хост:порт --insecure-skip-tls-verify=true --server=https://хост:порт
```

```
kubectl config set-context неймспейс/хост:порт/логин --user=логин/хост:порт --namespace=неймспейс --cluster=хост:порт
```

```
kubectl config use-context неймспейс/хост:порт/логин
```

1. Настройте конфигурационный файл config.env для развёртывания шаблонов:

```
PROJECT_NAME="test" # имя проекта
APP_NAME="unimapper" # имя сервиса преобразователя
APP_RCV_NAME="test-srv" # имя сервиса назначения
APP_PORT="8787"
REPOSITORY="registry.ru/unimapper-go-dev:0.1.0.5" # дистрибутив преобразователя
KUBERNETES_HOST="sber.ru" # основной хост Kubernetes
```

1. Создайте скрипт:

Скрипт для создания и обновления ресурсов deploy.sh

```
#!/bin/bash
FILES=configs/*.yaml

cat config.env | sed -e 's!^s/!; s!=/!; s!$/!>' > sed-tmp
for f in $FILES
do
  echo "Applying $f file..."
  sed -f sed-tmp $f | kubectl apply -f -
done
rm sed-tmp
```

Скрипт для удаления ресурсов delete.sh

```
#!/bin/bash
```

```
FILES=configs/*.yaml

for f in $FILES
do
  echo "Deleting $f file..."
  kubectl delete -f $f
done
```

Для запуска конфигурации используется файл: deploy.sh.

Для удаления конфигурации используется файл: delete.sh.

Файлы в поставке:

### **service-unimapper.yaml**

```
kind: Service
apiVersion: v1
metadata:
  name: unimapper-svc
spec:
  selector:
    app: unimapper
  ports:
    - name: http
      protocol: TCP
      port: 8787
      targetPort: 8787
```

### **deployment-unimapper.yaml**

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: unimapper
spec:
  selector:
    matchLabels:
      app: unimapper
  replicas: 1
```

```
template:
  metadata:
    labels:
      app: unimapper
    annotations:
      sidecar.istio.io/inject: "true"
  spec:
    volumes:
      - name: app-config
        configMap:
          name: unimapper-cm
          defaultMode: 420
    containers:
      - name: unimapper
        image: {registry}/unimapper-go-dev@{version}
        ports:
          - containerPort: 8787
            protocol: TCP
        env:
          - name: PROJECT_NAME
            valueFrom:
              fieldRef:
                apiVersion: v1
                fieldPath: metadata.namespace
        envFrom:
          - configMapRef:
              name: unimapper-cm
        volumeMounts:
          - name: app-config
            readOnly: true
            mountPath: /schemas
        resources:
          limits:
            cpu: 200m
            memory: 200Mi
          requests:
            cpu: 150m
            memory: 100Mi
```

## Схемы трансформации (ConfigMap)

- `json-request.json` — отвечает за трансформацию тела запроса из сервиса потребителя;
- `json-response.json` — отвечает за трансформацию ответа от сервиса назначения.

Для XML-преобразований аналогично.

Добавление и редактирование существующих схем происходит в ConfigMap сервиса. При добавлении схемы трансформации необходимо указать имя файла схемы и содержимое. Также, указать адрес монтируемого файла в Deployment и тип запроса (JSON\_REQUEST, JSON\_RESPONSE, XSL\_REQUEST, XSL\_RESPONSE).

### **configmap-unimapper.yaml**

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: unimapper-cm
  labels:
    app: unimapper
data:
  TRANSFORM_RESPONSE: "true"
  LOG_LEVEL: "debug"
  JSON_REQUEST: "/schemas/json-request.json"
  JSON_RESPONSE: "/schemas/json-response.json"
  XSL_REQUEST: "/schemas/xml-request.xsl"
  XSL_RESPONSE: "/schemas/xml-response.xsl"
  json-request.json: |-
    [
      {
        "operation": "shift",
        "spec": {
          "configs.types": "configs[*].@type",
          "configs.updated": "configs[*].last_updated",
          "configs.resources": "configs[*].bootstrap.dynamic_resources"
        }
      }
    ]
  xml-request.xsl: |-
    <?xml version="1.0" encoding="UTF-8"?>
    <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
      <xsl:template match="/">
        <OperationTRANSFORMED>
```



```

        <RqUIDTRANSFORMED>
            <xsl:value-of select="Operation/RqUID"/>
        </RqUIDTRANSFORMED>
        <RqTm>
            <xsl:value-of select="Operation/RqTm"/>
        </RqTm>
        <OperUIDTRANSFORMED>
            <xsl:value-of select="Operation/OperUID"/>
        </OperUIDTRANSFORMED>
        <Date>
            <xsl:value-of select="Operation/DateFrom"/>
        </Date>
    </OperationTRANSFORMED>
</xsl:template>
</xsl:stylesheet>

```

json-response.json: |-

```

[
  {
    "operation": "shift",
    "spec": {
      "types": "configs.types",
      "updated": "configs.updated"
    }
  }
]

```

xml-response.xml: |-

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <Operation-RESPONSE>
      <RqUID-RESPONSE>
        <xsl:value-of select="OperationTRANSFORMED/RqUIDTRANSFORMED"/>
      </RqUID-RESPONSE>
      <RqTm>
        <xsl:value-of select="OperationTRANSFORMED/RqTm"/>
      </RqTm>
      <OperUID-RESPONSE>
        <xsl:value-of select="OperationTRANSFORMED/OperUIDTRANSFORMED"/>
      </OperUID-RESPONSE>
    </Operation-RESPONSE>
  </template>
</stylesheet>

```

```
</xsl:template>  
</xsl:stylesheet>
```

## Обновление

Для обновления версии компонента Unimapper необходимо произвести разворачивание всех артефактов новой версии командой `kubectl apply -f <имя файла>`

## Проверка работоспособности

1. Убедитесь что выставленное количество подов Unimapper запущено (по умолчанию: 1)
2. Выполните проверку лога istio-proxu на отсутствие ошибок
3. Выполните проверку лога Unimapper на отсутствие ошибок
4. Выполните http запрос методом POST с обязательным путем в URL, по которому будет выполнена выборка необходимой схемы трансформации. Пример запроса из сервиса:

```
curl -X POST 'localhost:8787/xml' \  
--header 'method: POST' \  
--header 'Content-Type: application/xml' \  
--data-raw '<?xml version="1.0" encoding="UTF-8"?>  
<OperationName>  
<RqUID>fff</RqUID>  
<RqTm>2020-09-02T08:30:35</RqTm>  
<OperUID>1111</OperUID>  
<SPName>TEST</SPName>  
</OperationName>  
' -i
```

## Откат

Для отката Unimapper к предыдущей версии необходимо произвести разворачивание всех артефактов с использованием дистрибутива предыдущей версии.

## Часто встречающиеся проблемы и пути их устранения

Описание ошибки	Устранение проблемы
Ошибка от конечного сервиса	Возникает, если конечный сервис не доступен и не может принять запрос. Необходимо проверить наличие доступа до сервиса из любого контейнера проекта. Например, выполните в терминале контейнера команду <code>curl &lt;redirect_address&gt;</code> .
Включение отладки журналирования	Для включения отладки журналирования добавьте в ConfigMap сервиса env переменную LOG_LEVEL с значением debug. Для выключения удалите эту переменную.
Ошибка загрузки	Показывает ошибку в загруженной схеме формата XSLT. Необходимо провести анализ схемы на наличие ошибок.

Описание ошибки	Устранение проблемы
xsltproc	
Сервис недоступен	Необходимо проверить настройку окружения и маршрутизации IGEG.

## Чек-лист валидации установки

После установки выполните проверку указанную в пункте "Проверка работоспособности"