EVD Platform V Synapse Event Transfer Service

EVTD Сервис передачи сообщений

Руководство по установке

ОГЛАВЛЕНИЕ

Руководство по установке

Термины и определения

Термин/Аббревиатура	Определение
EVTD	Четырехбуквенный код программного компонента. Event Transfer Service — сервис передачи событий из состава программного продукта Platform V Synapse Event Transfer Service, основанный на технологиях Apache Kafka
DNS (Domain Name System)	Система доменных имен
DN/DName (Distinguished Name)	Уникальное имя сертификата, должно быть уникальным в пределах дерева. В DName описывается содержимое атрибутов в дереве (так называемый путь навигации), требуемое для доступа к конкретной записи ИЛИ базовой (стартовой) записи поиска. DName состоит из серии RDN (Relative Distinguished Names, относительных уникальных имен), определяемых путем перемещения вверх по дереву в направлении его корневой записи (суффикса или базовой записи), и записываемых слева направо
SSH	Сетевой протокол прикладного уровня, позволяющий производить удаленное управление операционной системой и туннелирование TCP-соединений
URL (Uniform Resource Locator)	Унифицированный указатель ресурса
REST	Архитектурный стиль взаимодействия компонентов

Термин/Аббревиатура	Определение
	распределенного приложения в сети
Гб (Гигабайт)	Единица измерения количества информации
ОС	Операционная система

Системные требования

Пререквизиты установки

- Ansible 2.9.
- Узлы EVTD обязательно проверить на возможность подключения на порт 9093 иных узлов продукта EVTD.
- Необходим доступ с узлов продукта EVTD до узлов Zookeeper продукта EVTD по порту 2181.
- Обеспечено разрешение имен хостов по IP в рамках всех хостов EVTD (DNS или записи в ./etc/hosts).

При использовании Jenkins (опциональный способ), дополнительно:

- Должен быть доступ в Jenkins и созданы необходимые сущности в нем.
- Все узлы сервиса EVTD должны быть доступны для вызова со стороны Jenkins.
- Должен быть доступ в BitBucket и создан в нем проект для помещения ролей и inventory.
- Должен быть доступ в Nexus и туда должен быть помещен дистрибутив EVTD.

Требования к серверам

ZOOKEEPER EVTD

(может располагаться на серверах EVTD, обязательно нечетное количество запущенных экземпляров)

- От 3-х серверов 2 ядра / 4 Гб RAM / 100 ГБ HDD или 4 ядра /8 Гб RAM /100 ГБ HDD.
- ОС Linux с версией kernel не ниже 3.10.0-327.
- На сервере установлены : java 8/java 11, unzip.

EVTD node

- От 3-х серверов 4/8/ от 150Гб или 8/16/ от 150Гб.
- Требуемый объем HDD зависит от нагрузки, рассчитывается предварительно.
- ОС Linux с версией kernel не ниже 3.10.0-327.
- На сервере установлены дистрибутивы: java 8/java 11, unzip.

Настройка серверов ZOOKEEPER EVTD

- Создать пользователя *kafka*, выдать пользователю права *sudoedit* для создания сервисов и права для управления сервисами:

```
kafka (ALL) NOPASSWD: /bin/systemctl start zookeeper
kafka (ALL) NOPASSWD: /bin/systemctl stop zookeeper
kafka (ALL) NOPASSWD: /bin/systemctl status zookeeper
kafka (ALL) NOPASSWD: /bin/systemctl restart zookeeper
kafka (ALL) NOPASSWD: /bin/systemctl enable zookeeper
kafka (ALL) NOPASSWD: /bin/systemctl disable zookeeper
kafka (ALL) NOPASSWD: /bin/systemctl daemon-reload
```

- Создать разделы на диске:

`/opt/Apache/` - 10Гб, владелец kafka:kafka; `/zookeeper/` - 50Гб, владелец kafka:kafka.

Настройка серверов EVTD

Подготовка окружения

Установить в настройки ядра Linux следующие значения:

```
net.ipv4.tcp_syn_retries = 3 <br>
net.ipv4.tcp_synack_retries = 3 <br>
net.ipv4.tcp_keepalive_time=60 <br>
net.ipv4.tcp_keepalive_probes=5 <br>
net.ipv4.tcp_keepalive_intvl=1 <br>
net.ipv4.tcp_retries2=3 <br>
```

где

- `tcp_syn_retries` - количество попыток передачи SYN-пакета при установлении нового соединения;
- `tcp_synack_retries` - количество попыток передачи SYN,ACK-пакета в ответ на SYN-запрос. Другими словами, максимальное число попыток установить пассивное TCP-соединение, инициированное другим хостом;
- `tcp_keepalive_time` - переменная определяет, как часто, в секундах, следует проверять соединение, если оно давно не используется;
- `tcp_keepalive_probes` - переменная определяет количество попыток проверки жизнеспособности прежде, чем будет принято решении о разрыве соединения;
- `tcp_keepalive_intvl` - переменная определяет интервал, в секундах, проверки жизнеспособность сокета. Это значение учитывается при подсчете времени, которое должно пройти перед тем как соединение будет разорвано;
- `tcp_retries2` - максимальное количество попыток повторной передачи пакетов, до того, как соединение будет считаться разорванным.

Пример

```
echo "3"    > /proc/sys/net/ipv4/tcp_syn_retries
echo "3"    > /proc/sys/net/ipv4/tcp_synack_retries
echo "60"   > /proc/sys/net/ipv4/tcp_keepalive_time
echo "5"    > /proc/sys/net/ipv4/tcp_keepalive_probes
echo "1"    > /proc/sys/net/ipv4/tcp_keepalive_intvl
echo "3"    > /proc/sys/net/ipv4/tcp_retries2
```

Создать пользователя* kafka*, выдать пользователю права* sudoedit *для создания сервисов и права для управления сервисами

```
kafka (ALL) NOPASSWD: /bin/systemctl start kafka
kafka (ALL) NOPASSWD: /bin/systemctl stop kafka
kafka (ALL) NOPASSWD: /bin/systemctl status kafka
kafka (ALL) NOPASSWD: /bin/systemctl restart kafka
kafka (ALL) NOPASSWD: /bin/systemctl enable kafka
kafka (ALL) NOPASSWD: /bin/systemctl disable kafka
kafka (ALL) NOPASSWD: /bin/systemctl daemon-reload
```

Создать разделы на диске

`/opt/Apache/` - 10Гб, владелец kafka:kafka; `/KAFKADATA/` - от 100Гб, владелец kafka:kafka (объем зависит от нагрузки, рассчитывается предварительно).

Создать JKS хранилище с сертификатом для брокеров, подписанное электронной подписью, выданной Удостоверяющим центром (далее - УЦ), доверенным для всех клиентов.

***Увеличить лимит дескрипторов для пользователя* kafka**

В файле `/etc/security/limits.conf` для пользователя *kafka* прописать:
```
nofile 128000
nproc 16384
```

Создание JKS хранилища и сертификатов

Для создания сертификата используется утилита **keytool.exe** из состава JDK. Для получения сертификата нужно:

1. Создать хранилище ключей и сертификатов:
   1.1. `keytool -genkey -keyalg RSA -alias Test -keystore [путь и имя файла с хранилищем ключей и сертификатов] -storepass [пароль для хранилища ключей и сертификатов] -validity 1440 -keysize 2048 -dname CN=[по правилам описанным ниже],OU=00CA,O=Org,L=Moscow,ST=Moscow,C=RU`
   Например: `keytool -genkey -keyalg RSA -alias ks -keystore D:\ks.jks -storepass 23101989 -validity 1440 -keysize 2048 -dname CN=00CA0001P.TestProducer.zzzz,OU=00CA,O=Org,L=Moscow,ST=Moscow,C=RU.`
   1.2. `keytool -certreq -alias Test -keyalg RSA -file [путь и имя файла с запросом на сертификат] -keystore [путь и имя файла с хранилищем ключей и сертификатов, созданного на шаге 1]`
2. Создать запрос на сертификат.
   Например: `keytool -certreq -alias ks -keyalg RSA -file D:\testProducer.csr -keystore D:\ks.jks`.
3. Отправить запрос на сертификат в УЦ.
4. Импортировать сертификаты в хранилище ключей.
   Полученные от УЦ файл с сертификатом и файлы с корневыми сертификатами необходимо импортировать в хранилище ключей и сертификатов при помощи утилиты **keytool.exe**.
   4.1. Первым необходимо импортировать корневой сертификат:
   `keytool -import -alias ks1 -file [путь и имя файла с корневым сертификатом, полученным от УЦ] -keystore [путь и имя файла с хранилищем ключей и сертификатов, созданного на шаге 1]`
   4.2. Вторым необходимо импортировать сертификат УЦ:

```
keytool -import -alias ks2 -file [путь и имя файла с сертификатом УЦ] -keystore
[путь и имя файла с хранилищем ключей и сертификатов, созданного на шаге 1]
```

4.3. Последним импортируется TLS-сертификат:

```
keytool -import -alias cmks -file [путь и имя файла с TLS-сертификатом,
полученным от УЦ] -keystore [путь и имя файла с хранилищем ключей и
сертификатов, созданного на шаге 1]
```

Формирование DN сертификата

CN = 00ZZ0001M.minitoringsystem.segment.contour.AS (пример)

Рекомендуется заполнять следующим образом:

- код ЦА - «00CA»;
- порядковый номер ключа (4 цифры) – до появления централизованной системы управления сертификатами не заполняется;
- тип ключа:
    - P - Producer (Продьюсер)
    - C - Consumer (Консьюмер)
    - S - Support monitoring (Инженер мониторинга)
    - M - Monitoring System (Система мониторинга)
    - A - Access manager (Администратор доступа)
    - E - Maintenance Engineer (Инженер сопровождения)
    - B - Broker (Брокер)
    - I - InfoSec Admin (Администратор безопасности)

    Для одного бизнес-сервиса допускается сертификат с несколькими ролями. Например сертификат системы, которая является одновременно поставщиком и потребителем событий, должен иметь тип ключа «PC» в DN сертификата.
- логин учетной записи пользователя;
- сетевой сегмент;
- тип (контур) кластера (только для брокера и администрирования, **роли producer и consumer не должны включать данный раздел**);

OU = OrganizationalUnitName

O = Organization

L = LocalityName

ST = StateOrProvinceName

C = CountryName

Установка

Версия дистрибутива

В корневом каталоге дистрибутива находится файл **EVTD-<номер дистрибутива>.info**, в котором содержится информация о версии дистрибутива. При установке EVTD на сервер данный файл появится в корне директории установки продукта. Содержимое файла **EVTD-<номер дистрибутива>.info**:

```
Info: <Наименование продукта>
```

```
Version: <Версия дистрибутива>
Date: <Дата сборки дистрибутива>
```

Например, файл **EVTD-D-01.001.00-00.info** имеет следующее содержимое:

```
Info: EVTD Сервис передачи событий
Version: D-01.001.00-00
Date: 2021-12-23 14:05
```

Ручной способ установки

1. Проверить, что на сервер, с которого будет производиться установка, установлен Ansible и с него доступны все узлы сервиса EVTD.
2. Распаковать дистрибутив и поместить содержимое директории *modules* в *scripts/Ansible*.
3. Все дальнейшие операции производить из директории *scripts/Ansible*.
4. Создать свой inventory (например, *ID*). Для этого создать директорию *ID* в папке *inventories*.
5. Создать структуру файлов по примеру, указанному в таблице.

Файл конфигурации	**Секция**	**Параметры**	**Описание значения**
`group_vars/all/vars.yml`		`ansible_user:` `ansible_port:`	Пользователь и порт для SSH-соединения ansible
		`zk_port_list:`	Перечень узлов zookeeper с портом Пример: `my-host1.org:2181,my-host2.org:2181`
	`kafka:jmx_access_roles:`	`password:`	Пароль к JMX-endpoint
	`kafka:audit`	`url:`	URL для подключения к REST-endpoint Platform V Audit SE. Если аудит не требуется, то секция kafka:audit не

Файл конфигурации	Секция	Параметры	Описание значения
			укзазывается
	kafka	superUser:	Distinguished Name (далее - DN) сертификата узла EVTD. DN задается без пробелов после запятых, разделяющих поля сертификата (При отсутствии этого поля значение будет рассчитано автоматически в процессе установки)
	kafka	xms: xmx:	xms: 256m # начальный heap size xmx: 4G # максимальный heap size
	zookeeper:	xms: xmx:	xms: 256m # начальный heap size xmx: 4G # максимальный heap size
	zookeeper:jmx_access_roles:	password:	Пароль к JMX-endpoint
group_vars/all/va		jks_password:	Токен vault пароля от jks-

Файл конфигурации	Секция	Параметры	Описание значения
ult.yml			хранилища узла EVTD
		ansible_ssh_pass:	Токен vault для пароля доступа SSH к узлам EVTD
		vault_password_encoder_secret:	Токен vault для ключа кодирования паролей в конфигурационных файлах
inventory	[kafka]		Перечень хостов брокеров EVTD в виде записей: <fqdn сервера> advertised_host =<fqdn для подключения извне> Пример: [kafka] my-host1.org advertised_host =my-host1.org my-host2.org advertised_host =my-host2.org my-host3.org advertised_host =my-host3.org
	[zookeeper]		Перечень хостов zookeeper EVTD в виде записей: <fqdn сервера> advertised_host =<fqdn для

Файл конфигурации	Секция	Параметры	Описание значения
			подключения извне> Пример: [zookeeper] my-host1.org advertised_host=my-host1.org my-host2.org advertised_host=my-host2.org my-host3.org advertised_host=my-host3.org
Папка ssl			Поместить jks-хранилище сертификата узла EVTD

Важно: Все параметры inventory аннотированы, приведенные выше параметры являются теми, на которые требуется обратить внимание (стендозависимые). В случае необходимости кастомизации рекомендуется читать аннотации параметров.

Использование vault для шифрования паролей

При хранении сенситивной информации в Git ее рекомендуется шифровать. Для этого можно использовать утилиту **ansible-vault** (идет в комплекте с пакетом ansible).

Для шифрования пароля следует выполнить команду на сервере, с которого производится развертывание EVTD:

```
ansible-vault encrypt_string -n jks_password 'ENCRYPT_STRING'
```

где ENCRYPT_STRING - строка, которую необходимо зашифровать, а jks_password - имя переменной.

При запросе вводим пароль для шифрования, а на выходе получаем то, что нужно занести в inventory:

```
jks_password: !vault |
  $ANSIBLE_VAULT;1.1;AES256
  30323632346331616266363234303338663965366539343535353133626165316564633237626536
  39323338313537393561353764633233633261333333338340a336338623837303937393538313939
  37626531383432366662303466363761616566393638306564623661323133356133613863313032
  39666535316436316 60a666136623361613863643137396663653363316139316566393366653838
  3039
```

Аналогично, возможно шифрование файлов:

```
ansible-vault encrypt <имя файла>
```

Ручное шифрование паролей в конфигурационных файлах

Для шифрования паролей используется утилита **password-encrypt-cli-1.3.jar** в составе дистрибутива EVTD и пакет java, установленный на сервере.

Для шифрования вызывается команда:

```
java -jar password-encrypt-cli-1.3.jar --key <ключ> --password <пароль>
```

В результате выполнения команды будет выведен зашифрованный пароль:

```
Encrypted password: <зашифрованный пароль>
```

где <ключ> - содержимое файла *encrypt.pass*, путь к которому указывается в *encoding_configs/security.encoding.key*.

Запуск установки

Запустить установку командой:

```
ansible-playbook -i inventories/<ID>/inventory zk_and_kafka.yml --ask-vault-pass
```
где `ID` - имя недавно созданного inventory.

Установка будет производиться на все хосты из inventory. Для ограничения списка узлов используем команду:

```
ansible-playbook -i inventories/<ID>/inventory zk_and_kafka.yml --ask-vault-pass -l <узлы через запятую без пробелов>
```

Установка с помощью Jenkins (опциональный способ)

1. Распаковать дистрибутив и поместить содержимое папки *scripts* в BitBucket.
2. Создать и настроить inventory (см. раздел *Ручной способ установки* выше) и поместить изменения в BitBucket.
3. В Jenkins создать Jenkins Pipeline с получением скриптов развертывания из BitBucket.
    - Pipepline script from SCM
        - SCM - GIT
        - repository url - ссылка на репозиторий, куда поместили скрипты.
        - Выбираем или добавляем учетные данные для доступа к BitBucket
    - Script_path - относительный путь *SYN_custom.groovy*. Убедиться, что не стоит галочка *Lightweight checkout*.
4. Сохранить получившийся Jenkins Pipeline и запустить его.
5. Проверить, что после запуска подгрузились дополнительные параметры.
6. При необходимости, поменять для параметров значения по умолчанию. Например, изменить имя используемых *credentials*.

Запуск установки

При запуске задания Jenkins по установке в параметрах выбирать нужный inventory, playbook `zk_and_kafka.yml`, а в поле `customURL` указать ссылку на дистрибутив.

Предоставление прав на кластере для администратора доступа

По завершению установки под пользователем *kafka* на сервере EVTD создать пользователя с правами администратора доступа, который в дальнейшем будет отвечать за выдачу прав пользователям:

```
/opt/Apache/kafka/bin/kafka-acls.sh --bootstrap-server `hostname -f`:9093 --allow-principal
User:"<DN сертификата администратора доступа из папки ssl_admin>" --operation CREATE --
cluster --add --command-config /opt/Apache/kafka/config/producer.properties

/opt/Apache/kafka/bin/kafka-acls.sh --bootstrap-server `hostname -f`:9093 --allow-principal
User:"<DN сертификата администратора доступа из папки ssl_admin>" --operation ALTER --
cluster --topic "*" --add --command-config /opt/Apache/kafka/config/producer.properties

/opt/Apache/kafka/bin/kafka-acls.sh --bootstrap-server `hostname -f`:9093 --allow-principal
User:"<DN сертификата администратора доступа из папки ssl_admin>" --operation DESCRIBE --
cluster --group "*" --topic "*" --add --command-config
/opt/Apache/kafka/config/producer.properties

/opt/Apache/kafka/bin/kafka-acls.sh --bootstrap-server `hostname -f`:9093 --allow-principal
User:"<DN сертификата администратора доступа из папки ssl_admin>" --operation
DESCRIBECONFIGS  --topic "*" --add --command-config
/opt/Apache/kafka/config/producer.properties

/opt/Apache/kafka/bin/kafka-acls.sh --bootstrap-server `hostname -f`:9093 --allow-principal
User:"<DN сертификата администратора доступа из папки ssl_admin>" --operation ALTERCONFIGS
--topic "*" --add --command-config /opt/Apache/kafka/config/producer.properties
```

Настройка сервисов

ZooKeeper

На серверах ZooKeeper EVTD:

- создать сервисы для перезапуска процесса Zookeeper с помощью команды:

```
ansible-playbook -i inventories/<ID>/inventory zk_kafka_service.yml --ask-vault-pass
```

где `ID` - имя недавно созданного inventory.

Kafka

На серверах EVTD:

- создать сервисы для перезапуска процесса EVTD с помощью команды:

```
ansible-playbook -i inventories/<ID>/inventory zk_kafka_service.yml --ask-vault-pass
```

где `ID` - имя недавно созданного inventory.

Обновление

Перед обновлением рекомендуется сделать резервную копию текущей версии.

Поузловое обновление кластера через установку новой версии дистрибутива:

1. Проверить, что в *inventory* в **vars.yaml** установлено *cleanData = false*.
2. Произвести установку EVTD.
    - При ручной установке - выполнить команду:
3. `ansible-playbook -i inventories/<ID>/inventory zk_and_kafka.yml --ask-vault-pass -l <узел для обновления>`
    - При использовании Jenkins - запустить задание Jenkins по установке с параметрами:
        - выбрать нужный контур;
        - playbook - *zk_and_kafka.yml*;
        - *customURL* - указать ссылку на дистрибутив;
        - *only_on_host* - отметить галочками нужные хост(ы) из списка (список соответствует выбранному inventory), если необходимо перезапустить компоненты только для данного хоста(ов) выбранного кластера;
        - *install_all_hosts* - выбрать параметр, если необходимо перезапустить все хосты из данного inventory.

    Если не выбран ни один из параметров *only_on_host*, *install_all_hosts* выполнение задания Jenkins прервется с ошибкой *Не выбраны хосты*.
4. Убедиться, что в лог-файлах узла присутствует запись о старте сервера. Для этого на узле выполнить `cat /opt/Apace/kafka/logs/server.log | grep "started (kafka.server.KafkaServer)"` и убедиться, что вхождение строки есть;
5. Убедится в отсутствии ошибок в **zookeeper.log**.
6. Перейти к следующему узлу.

Удаление

Процесс удаления отсутствует.

Проверка работоспособности

Скрипты установки автоматически по завершению проверяют корректность и успешность проведенных действий.

При возникновении ошибки при ручной установке: обработка скриптом остановится, в консоль будет выведен текст ошибки.

При возникновении ошибки при автоматической установке: Jenkins Build завершится с ошибкой, Console Output будет содержать сообщение об ошибке.

Откат

Общий подход

Понодное обновление кластера EVTD производится через установку старой версии дистрибутива. Подробно описано в данном руководстве в разделе «Установка».

При наличии созданного ранее бэкапа, можно восстановиться из него, запустив установку с тегом *backup_restore*.

Часто встречающиеся проблемы и пути их устранения

Проблема	Причина	Исправление
Отсутствует связь между брокерами в кластере	* При конфигурировании, DN сертификата в `kafka.superUser` был задан не корректно * Отсутствует физический доступ между узлами по порту 9093	* DN сертификата должен задаваться без пробелов после запятых, разделяющих поля. Необходимо исправить настройки и произвести установку повторно. * Открыть доступ между узлами по порту 9093

Чек-лист валидации установки

- Убедиться, что в лог-файлах узлов EVTD присутствует запись о старте сервера. Для этого на узле выполнить `cat /opt/Apace/kafka/logs/server.log | grep "started (kafka.server.KafkaServer)"` и убедиться, что вхождение строки есть.
- Убедиться, что сервис zookeeper работает корректно. Для этого выполнить команду `telnet <адрес узла zookeeper> 2181`. В появившемся приглашении ввода ввести ruok и нажать Enter. Корректно работающий узел ответит imok и закроет соединение.
- Проверить работу сервисов ZooKeeper. Для этого требуется зайти на каждый из серверов ZooKeeper EVTD и выполнить команду: `systemctl list-unit-files | grep -e zookeeper`. Проверить содержимое конфигурации сервиса командой `cat /etc/systemd/system/zookeeper.service`. В частности, проверить, что исполняемые команды в параметрах ExecStart и ExecStop ссылаются на существующие файлы.
- Проверить работу сервисов Kafka. Для этого требуется зайти на каждый из серверов EVTD и выполнить команду: `systemctl list-unit-files | grep -e kafka`. Проверить содержимое конфигурации сервиса командой `cat /etc/systemd/system/kafka.service`. В частности, проверить, что исполняемые команды в параметрах ExecStart и ExecStop ссылаются на существующие файлы.