EVP Platform V Synapse Streaming Event Processing

EVTP Сервис обработки сообщений

Руководство по установке

ОГЛАВЛЕНИЕ

Руководство по установке

Термины и определения

Термин/Аббревиатура	Определение
EVTP	Четырехбуквенный код программного компонента. Streaming Event Processing — сервис обработки событий, из состава программного продукта Platform V Synapse Streaming Event Processing, основанный на технологиях Apache Flink
Ansible	Система управления конфигурациями, написанная на языке программирования Python, с использованием декларативного языка разметки для описания конфигураций
BitBucket	Веб-сервис для хостинга проектов и их совместной разработки
DNS	Domain Name System, служба доменных имен
Jenkins	Программная система для обеспечения процесса непрерывной интеграции программного обеспечения
Kafka	Apache Kafka
Nexus	Хранилище репозиториев и артефактов Nexus
Zookeeper	Сервис, используемый кластером для обеспечения координации между узлами и поддержки общих данных

Термин/Аббревиатура	Определение
	с помощью надежных методов синхронизации
inventory	Хранилище конфигурационных файлов и настроек

Системные требования

- Предоставлен доступ от узлов компонента EVTP до узлов Zookeeper компонента EVTP по порту 2181.
- Обеспечено разрешение имен хостов по IP в рамках всех хостов EVTP (DNS или записи в `/etc/hosts`).
- Ansible версии 2.9 и выше.

При использовании Jenkins (опциональный способ) дополнительно требуется:

- предоставить доступ в Jenkins и создать необходимые сущности;
- Предоставить доступ в BitBucket и создать проект для размещения ролей и inventory.
- Предоставить доступ в Nexus и разместить дистрибутив EVTP.
- все узлы сервиса EVTP должны быть доступны для вызова со стороны Jenkins.

Требования к серверам

ZOOKEEPER EVTP

ZOOKEEPER может располагаться на серверах EVTP, обязательно нечетное количество запущенных экземпляров.

Требуется от трех серверов с конфигурацией сервера:

- CPU минимально — 2 ядра, оптимально — 4 ядра;
- RAM 4 Гбайт;
- HDD 100 ГБайт;
- Linux с kernel не ниже 3.10.0-327;
- Java версии 11 и выше;
- unzip.

EVTP node

Требуется от двух серверов с конфигурацией сервера:

- CPU минимально — 4 ядра, оптимально — 8 ядер;
- RAM минимально — 8 Гбайт, оптимально — 16 Гбайт;
- HDD 150 ГБайт;
- Linux с kernel не ниже 3.10.0-327;
- Java версии 11 и выше;
- unzip;

- подмонтированный NFS от 100 Гбайт (директория /Flink пользователь flink:flink) или выделенный S3/HDFS для режима высокой доступности и хранения состояния обработчиков

Настройка серверов ZOOKEEPER EVTP

1. Создайте пользователя **flink**.
2. Предоставьте пользователю права `sudoedit` для создания сервисов и права для управления сервисами:
    - `flink (ALL) NOPASSWD: /bin/systemctl start zookeeper;`
    - `flink (ALL) NOPASSWD: /bin/systemctl stop zookeeper;`
    - `flink (ALL) NOPASSWD: /bin/systemctl status zookeeper;`
    - `flink (ALL) NOPASSWD: /bin/systemctl restart zookeeper;`
    - `flink (ALL) NOPASSWD: /bin/systemctl enable zookeeper;`
    - `flink (ALL) NOPASSWD: /bin/systemctl disable zookeeper;`
    - `flink (ALL) NOPASSWD: /bin/systemctl daemon-reload.`
3. Создайте разделы на диске.
    - `/opt/Apache/` — 10 Гбайт, владелец `flink:flink`;
    - `/data` — 50 Гбайт, владелец `flink:flink`.

Настройка серверов EVTP

1. Создайте пользователя **flink**.
2. Предоставьте пользователю права `sudoedit` для создания сервисов и права для управления сервисами.

Для серверов с Jobmanager:

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

Для серверов с Taskmanager:

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

1. Создайте разделы на диске:
    - `/opt/Apache/` — 20 Гбайт, владелец `flink:flink`;
    - `/flink-logs/` — 100 Гбайт, владелец `flink:flink`.
2. Создайте JKS-хранилище с клиент-серверным сертификатом, подписанное УЦ, доверенным для всех клиентов.
3. В файле `/etc/security/limits.conf` для пользователя flink увеличьте лимит файловых дескрипторов.

```
nofile 128000
nproc 16384
```

Создание служебных топиков

Создайте служебные топики `audit-buffer-flink` при использовании буфера отправки событий аудита в Apache Kafka.

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:

- `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.`
- `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]`

1. Создать запрос на сертификат.
   Например: `keytool -certreq -alias ks -keyalg RSA -file D:\testProducer.csr -keystore D:\ks.jks`.
2. Отправить запрос на сертификат в УЦ.
3. 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 (логин ТУЗ или КЭ для систем и УЗ для пользователей);
- 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 **evtp-release.info**, в котором содержится информация о версии дистрибутива. При установке EVTP на сервер данный файл появится в корне директории установки продукта. Содержимое файла **evtp-release.info**:

```
Info: <Наименование продукта>
Version: <Версия дистрибутива>
Build date: <Дата сборки дистрибутива>
```

Например:

```
Info: EVTP Сервис потоковой обработки событий
Version: D-01.001.00-00
Build 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 EVTP.
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
	flink:high_availibility	zookeeper:	Перечень узлов zookeeper сервиса EVTP с портом. Пример: 172.31.12.11:2181, 172.31.12.12:2181
	flink	taskManager_xmx	Максимальный размер heap Task Manager
	flink	numerOfTaskSlots	Число слотов под задачи Task Manager
	flink	customSystemProps:	Задаются настройки подключения к S3 для хранения чек-поинтов и размер хранилища метаданных Task Manager в памяти, можно также задать иные параметры конфигурации flink runtime: - { key: 'taskmanager.memory.jvm-metaspace.size', value: '1024m' } - { key: 'classloader.resolve-order', value: 'parent-first' } - { key: 's3.endpoint', value: 'https://test.test.ru' } #адрес S3-endpoint - { key: 's3.access-key', value: '71e641c9-9e28-4299-

Файл конфигурации	Секция	Параметры	Описание значения
			a4ce-6427c37afaff' } #ключ доступа до S3 - { key: 's3.secret-key', value: vault_s3_key } #секретный ключ - { key: 'high-availability.storageDir', value: 's3://synapse/flink-ha' } #точка хранения данных - { key: 'state.checkpoints.dir', value: 's3://synapse/flink-ha/default-checkpoints' } #точка хранения чек-поинтов - { key: 'state.savepoints.dir', value: 's3://synapse/flink-ha/default-savepoints' } #точка хранения savepoint-ов
group_vars/all/vault.yml		jks_password:	Токен vault пароля от jks-хранилища узла EVTP
		ansible_ssh_pass:	Токен vault для пароля доступа SSH к узлам EVTP (от ТУЗ flink)
		vault_s3_key:	Токен vault для закрытого ключа соединения с S3
inventory	[flink_jobmanager]		Перечень хостов JM EVTP в виде записей: <IP-адрес или fqdn сервера>

Файл конфигурации	Секция	Параметры	Описание значения
			advertised_host=<IP-адрес или fqdn сервера> Пример: [flink_jobmanager] 172.28.0.72 advertised_host=172.28.0.72 172.28.0.222 advertised_host=172.28.0.222
	[flink_taskmanager]		Перечень хостов TM EVTP в виде записей: <IP-адрес или fqdn сервера> advertised_host=<IP-адрес или fqdn сервера> пример: [flink_taskmanager] 172.28.0.72 advertised_host=172.28.0.72 172.28.0.222 advertised_host=172.28.0.222
Папка ssl			Поместить JKS-хранилище сертификата узла EVTP

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

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 выполните команду:

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

где:

- `ENCRYPT_STRING` — строка, которую необходимо зашифровать;
- `jks_password` — имя переменной.

При запросе введите пароль для шифрования. Полученные сведения внесите в inventory:

```
jks_password: !vault |
  $ANSIBLE_VAULT;1.1;AES256
  303236323463316162663632343033386639653665393435353531336261653165646332376265 36
  393233383135373935613537646332336332613333333838340a336338623837303937393538313939
  376265313834323666623034663637616165663936383065646236613231333561336138633130 32
  396665353164363166 0a666136623361613863643137396663653363316139316566393366653838
```

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 EVTP и пакет java, установленный на сервере.

Для шифрования паролей вызывается команда на сервере, с которого производится развертывание EVTP:

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

В результате выполнения команды будет выведен зашифрованный пароль:

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

Запуск установки

Запустить установку командой:

```
ansible-playbook -i inventories/<ID>/inventory flink.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 flink.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, где размещены скрипты, выберите или добавьте учетные данные для доступа к BitBucket;
    - Script_path — относительный путь `SYN_custom.groovy`. Убедитесь, что не установлен флажок **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 `flink.yml`, а в поле `customURL` указать ссылку на дистрибутив.

Настройка транспорта

После завершения установки EVTP на серверах в папке `/opt/Apache/flink/mapping` создайте файл `defaults.json` с настройками транспорта (при установке с помощью Jenkins файл создается автоматически):

```
{
    "defaults": {
        "flinkInstalldir": "/opt/Apache/flink",
        "kafka": {
            "bootstrap.servers": "<список узлов EVTD>",
            "secret": "/opt/Apache/flink/conf/encrypt.pass",
            "security.protocol": "SSL",
            "ssl.endpoint.identification.algorithm": "",
            "ssl.key.password": "",
            "ssl.keystore.location": "/opt/Apache/flink/ssl/flink.jks",
            "ssl.keystore.password": "",
            "ssl.truststore.location": "/opt/Apache/flink/ssl/flink.jks",
            "ssl.truststore.password": "",
            "type": "kafka"
        }
    }
}
```

Обновление

Поузловое обновление кластера производится через установку новой версии дистрибутива.

1. Выполните установку EVTP.
    - При ручной установке:
2. `ansible-playbook -i inventories/defaults/inventory flink.yml --ask-vault-pass -l <узел для обновления>`
    - При использовании Jenkins - запустить задание Jenkins по установке с параметрами:
    - выбрать нужный контур;
    - playbook - *flink.yml*;
    - *customURL* - указать ссылку на дистрибутив;
    - *only_on_host* - указать хост из inventory, на котором будет производиться обновление.
3. Повторите действия на следующем узле.

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

Понодное обновление кластера выполняется путем установки старой версии дистрибутива EVTP. 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

Не выявлено.

Чек-лист валидации установки

- Проверьте работу APM управления Apache Flink:
  . на узле Job Manager EVTP в файле `/opt/Apache/flink/conf/flint-conf.yaml` найдите значение параметра `rest.bind-port`;
  i. в браузере перейдите на страницу `http://<адрес узла JM EVTP>:<порт из параметра rest.bind-port>`. Откроется пользовательский интерфейс управления.
- Убедитесь, что сервис Zookeeper работает корректно:
  . Выполните команду:
  i. `telnet <адрес узла zookeeper> 2181`
  ii. Введите `ruok` и нажмите клавишу **Enter**. Корректно работающий узел ответит `imok` и закроет соединение.
- Проверьте работу сервисов ZooKeeper:
  . на каждом из серверов ZooKeeper EVTP выполните команду:
  i. `systemctl list-unit-files | grep -e zookeeper`
  ii. проверьте содержимое конфигурации сервиса с помощью команды:
  iii. `cat /etc/systemd/system/zookeeper.service`
  В частности, проверьте, что исполняемые команды в параметрах `ExecStart` и `ExecStop` ссылаются на существующие файлы.
- Проверьте работу сервисов Kafka — на каждом из серверов EVTP выполните команду:
- `systemctl list-unit-files | grep -e flink`
- Проверьте содержимое конфигурации сервиса с помощью команд:
- `cat /etc/systemd/system/flink_jobmanager.service`
- `cat /etc/systemd/system/flink_taskmanager.service`

В частности, проверьте, что исполняемые команды в параметрах `ExecStart` и `ExecStop` ссылаются на существующие файлы.