Инструкция по настройке CICD для NiFi.

Table of Contents

1 Для деплоя схем в nifi: 4

2 Для разработки кастомных процессоров: 6

Есть два способа: для деплоя схем в nifi и для разработки кастомных процессоров.Для деплоя схем в nifi:

1. Копируем в свой репозиторий ,gitlam-ci.yaml.
2. Меняем в этом файле параметры: REGISTRY\_NAME - название схемы в нифи-реджистри в бакете dev, которую необходимо задеплоить; VERSION - версия схемы в нифи-реджистри, которую необходимо задеплоить; PROPERTIES\_STRING - список сенситив пропертей, которые необходимо ввести в процессоры при деплое.

Что нужно знать:

* Все недостающие кастомные nar файлы будут скачиваться на сервера из нексуса. Соответственно, если они есть в схеме, они должны быть выложены в нексус (см. "Для разработки кастомных процессоров" ниже)
* Все секреты в виде файлов должны выглядеть следующим образом: /opt/nifi/secrets/PROJECT/SECRET , где PROJECT - название группы в GitLab, из которой выполняется деплой, а SECRET - название секрета в вашем пространстве в vault с ключом PASSWORD. Они будут размещены автоматически, в них будет записано содержимое значения по ключу PASSWORD. Исключение составляют файлы \*.keytab: содержимое таких файлов в vault в поле PASSWORD должно быть закодированном base64 формате.
* Пути до секретов, используемых в схемах, в тех местах, в которых они указываются, должны указываться полностью (/opt/nifi/secrets/PROJECT/SECRET). Если секрет будет разбит на части (например, начало в variable, а продолжение дописано в проперти процессора), CICD некорректно его обработает, что может прервать процесс внедрения схемы до устранения таких составных путей. Допускается заведение variable с полным путём до секрета и дальнейшее использование её в пропертях процессоров и контроллер сервисах.
* Формат PROPERTIES\_STRING следующий: "Название процессора,Название проперти,Название секрета в vault" . При необходимости ввести несколько секретов, эту строчку можно продолжать: "proc1,prop1,secret1,proc2,prop2,secret2,...". Проперти обазательно должны быть sensitive. В секрете должна быть пара ключ-значение с ключом PASSWORD, содержимое значения такой пары будет записано в проперти. Название всех указанных процессоров proc1, proc2, ... должны быть уникальными в схеме.
* Если необходимых nar файлов в нексусе или секретов в vault не будет обнаружено, CD упадёт с ошибкой.
* Все этапы запускаются в ручном режиме для большего контроля.
* Переменные с адресом NiFi, в который будет производится деплой, и с названиями серверов (записанных через пробел), на которые будут размещаться секреты, называются NIFI\_HOST и NIFI\_HOSTS, соответственно. Если вам по какой-то причине потребовалось деплоить схемы на другой кластер NiFi (относительно того, на который производились деплои раньше), необходимо повторно завести задачу, а также в настройках репозитория указать новые переменные NIFI\_HOST и NIFI\_HOSTS (Settings→CI/CD→Expand Variables→Add variable). Полный список серверов, относящихся к кластеру можно посмотреть в списке экземпляров NIFI
* При каждом запуске deploy на целевой процесс группе будет вызван Start для всех её процессоров и Enable для всех контроллер сервисов. Из этого следует, что в схеме не должно быть лишних, неиспользуемых контроллер сервисов, а так же все процессоры, которые должны работать, НЕ будут в статусе disabled на момент сохранения схемы в nifi-registry: все процессоры, сохранённые в статусе disabled, будут перенесены в прод в этом статусе и НЕ будут запущены.
* Для удаления схемы, которая больше не нужна, достаточно задеплоить её с параметром VERSION: "0" (ноль). **При удалении схемы все её очереди очищаются, данные, находящиеся в них, будут удалены. Если они не хранятся где-то ещё, то безвозвратно.**
* Так как nifi не позволяет удалить очередь, в которой находятся флоуфайлы, обновление схемы, включающее удаление такой очереди, выполнено **не** будет.
* CICD состоит из 3 **независимых** этапов. 1 "copy nars to nifi libs" (Nars) спуливает из нексуса кастомные процессоры схемы и выкладывает их на сервера кластера nifi. 2 "move secrets from vault to nifi(cephfs)" (Secrets) достаёт из волта секреты, которые требуются в схеме в виде файлов, и размещает их на сервера кластера nifi. 3 "deploy-nifi-pipeline" (Deploy) выполняет непосредственное изменение nifi схемы (деплой новой схемы\обновление существующей\удаление схемы). Если для деплоя схемы не требуется изменять nar файлы и\или добавлять новые файловые секреты, то соответствующие этапы запускать нет необходимости.

# Для разработки кастомных процессоров:

1. Копируем в свой репозиторий ,gitlam-ci.yaml
2. В pom.xml добавляем следующий блок:

|  |
| --- |
| <parent>  <groupId>ru.beeline</groupId>  <artifactId>nexus\_nifi\_pom</artifactId>  <version>1.0.0</version>  </parent> |

Для размещения артефакта в нексус после тестирования необходимо создать тэг release-VERSION, где VERSION - версия артефакта.