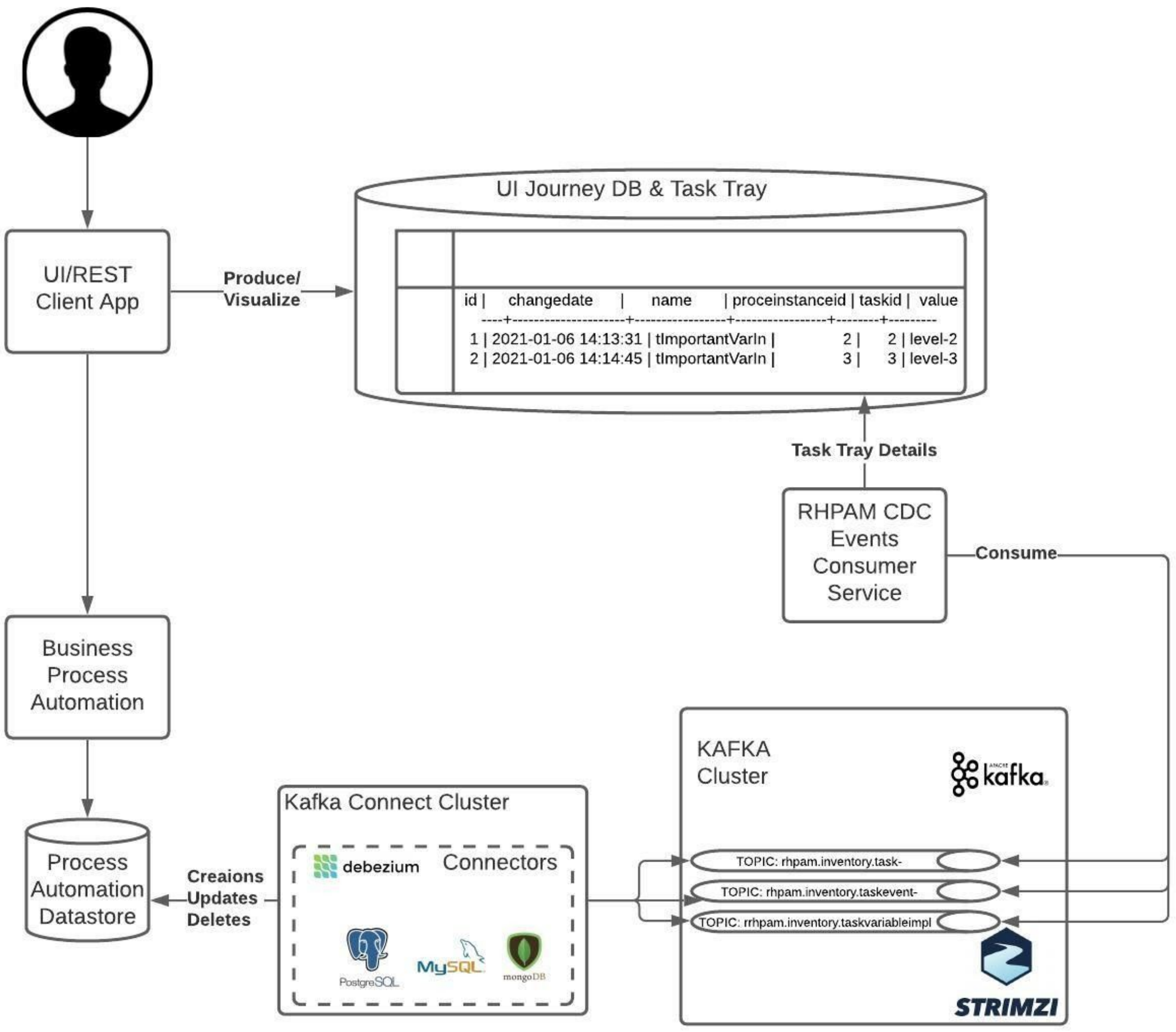
RHPAM Task Variable Events CDC to Kafka

The architecture of this option for extracting and monitoring of Task and Variable details is depicted below:


UI Journey DB & Task Tray
UI/REST Client App
Produce/ Visualize
Task Tray Details
RHPAM CDC Events Consumer Service
Consume
Business Process Automation
Process Automation Datastore
Creaions Updates Deletes
Kafka Connect Cluster
debezium
Connectors
PostgreSQL
MySQL
mongoDB
KAFKA Cluster
kafka
TOPIC: rhpam.inventory.task-
TOPIC: rhpam.inventory.taskevent-
TOPIC: rrhpam.inventory.taskvariableimpl
STRIMZI

```
id |     changedate       |      name        | proceinstanceid | taskid |  value
----+--------------------+----------------+-----------------+--------+---------
  1 | 2021-01-06 14:13:31 | tImportantVarIn |               2 |      2 | level-2
  2 | 2021-01-06 14:14:45 | tImportantVarIn |               3 |      3 | level-3
```

Resources

- CDC pipeline with Red Hat AMQ Streams and Red Hat Fuse
- Debezium on Openshift Cheatsheet
- Quarkus Cheatsheet

Prepare Environment

Prerequisite is access to OCP Cluster with capability to

a) Install Strimzi/AMQ Streams operator (TBD CRD to do that? Otherwise from console)
b) Create KAFKA CRD to create the KAFKA Cluster (see Appendix B - Kafka Cluster Setup)
c) Create KAFKA AMQ Streams/KAFKA Connect (see Appendix B - Kafka Connect with Debezium plugins IMAGE Creation and Deployment)

Prepare Applications

Create and Deploy KIE Server (Spring Boot Based) Service

1. Build the KJAR (in .m2 or MAVEN Artifact Repository)

https://github.com/skoussou/JBossAutomationPlayground/tree/master/example-kjars/simple-process-kjar

2. Build and Deploy KIE Server Service based on KJAR
   a. Utilize Debezium Based MYSQL Database rather than OCP 8.x database
      i. With OCP 8.x DB the GLOBAL_VARIABLES table is missing so the following didn't work with debezium

```
DATABASE 8:  oc new-app --template=mysql-ephemeral -p DATABASE_SERVICE_NAME=pam-mysql
-p MYSQL_USER=jbpm -p MYSQL_PASSWORD=jbpm -p MYSQL_ROOT_PASSWORD=root -p
MYSQL_DATABASE=jbpm
```

      ii. DATABASE 5.7 with DEBEZIUM based image (Used and works)

```
oc new-app --name=dbz-14-pam-mysql debezium/example-mysql:1.4
-e=MYSQL_ROOT_PASSWORD=debezium  -e=MYSQL_USER=jbpm -e=MYSQL_PASSWORD=jbpm
```

      iii. With DEBEZIUM Image and modified to use Mysql 8 (TESTED no problem ie. use DEBEZIUM DB)

Gunnar said: It Would be interesting to see how things look if you use version 8.x for that example image

https://github.com/debezium/docker-images/blob/master/examples/mysql/1.3/Dockerfile

i.e. deriving that from 8 instead of 5.7 we probably should update the image anyways
./build-debezium.sh 1.3

```
1. ./build-debezium.sh 1.3
2. docker image tag debezium/example-mysql:1.3
   default-route-openshift-image-registry.apps.cluster-demo-d3f8.demo-d3f8.example
   .opentlc.com/dev-demo/dbz-example-mysql:1.3-8.0
3. docker login -u `oc whoami` -p `oc whoami -t`
   default-route-openshift-image-registry.apps.cluster-demo-d3f8.demo-d3f8.example
   .opentlc.com
4. docker push
   default-route-openshift-image-registry.apps.cluster-demo-d3f8.demo-d3f8.example
   .opentlc.com/dev-demo/dbz-example-mysql:1.3-8.0
5. oc new-app
   --docker-image=image-registry.openshift-image-registry.svc:5000/dev-demo/dbz-ex
```

```
ample-mysql:1.3-8.0  --name=dbz-13-80-pam-mysql -e=MYSQL_ROOT_PASSWORD=debezium
-e=MYSQL_USER=jbpm -e=MYSQL_PASSWORD=jbpm -e=MYSQL_DATABASE=jbpm -l
app=dbz-mysql-example-13-80
```

b. Using SB RHPAM based on repo https://github.com/skoussou/springboot-business-app configure the mysql DB above details in application-openshift.properties and the KJAR details in business-application-service.xml and then
c. Deploy it

```
mvn clean package -DskipTests=true -P openshift -Dmaven.artifact.threads=50 -s
~/.m2/settings.xml

mvn oc:deploy -Djkube.namespace=dev-demo -DskipTests=true -P openshift
-Dmaven.artifact.threads=50 -s ~/.m2/settings.xml
```

d. Create process and tasks content

```
curl -u user:user -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' -d '{ "taskOwner" : "user", "pImporantVar" : "level-2"}'
'http://business-application-service-dev-demo.apps.cluster-demo-d3f8.demo-d3f8.examp
le.opentlc.com/rest/server/containers/simple-process-kjar-1.0.8/processes/ht-basics.
simple-ht/instances'
```

e. This will create db events from RHPAM.

Create and Deploy KAFKA CONNECT/DEBEZIUM Connector (CONFIGURATION/USAGE)

Inspecting Kafka Connect Service Debezium Connector

1. Choose the kafka connect service by running

```
oc get svc -l app.kubernetes.io/name=kafka-connect -o json | jq -r '.items[] |
.metadata.name'
```

2. Export the following environment properties

```
export DEBEZIUM_CONNECT_SVC=debezium-connect-connect-api
export CONNECTOR=rhpam-connector
```

3. Check the available connector plugins:

GET /connector-plugins	check the available connector plugins

```
oc exec -i events-cluster-kafka-0 -- curl -X GET -H "Accept:application/json" -H
"Content-Type:application/json" http://$DEBEZIUM_CONNECT_SVC:8083/connector-plugins
|jq
 [
  {
    "class": "io.debezium.connector.mongodb.MongoDbConnector",
    "type": "source",
    "version": "1.3.1.Final"
  },
  {
    "class": "io.debezium.connector.mysql.MySqlConnector",
    "type": "source",
    "version": "1.3.1.Final"
  },
  {
    "class": "io.debezium.connector.postgresql.PostgresConnector",
    "type": "source",
    "version": "1.3.1.Final"
  },
  {
    "class": "org.apache.kafka.connect.file.FileStreamSinkConnector",
    "type": "sink",
    "version": "2.5.0.redhat-00003"
  },
  {
    "class": "org.apache.kafka.connect.file.FileStreamSourceConnector",
    "type": "source",
    "version": "2.5.0.redhat-00003"
  },
  {
    "class": "org.apache.kafka.connect.mirror.MirrorCheckpointConnector",
    "type": "source",
    "version": "1"
  },
  {
    "class": "org.apache.kafka.connect.mirror.MirrorHeartbeatConnector",
    "type": "source",
    "version": "1"
  },
  {
    "class": "org.apache.kafka.connect.mirror.MirrorSourceConnector",
    "type": "source",
    "version": "1"
  }
]
```

4. Get all connectors:

GET /connectors Get a list of active connectors
* request:

```
oc exec -i events-cluster-kafka-0 -- curl -X GET \
-H "Accept:application/json" \
-H "Content-Type:application/json" \
http://$DEBEZIUM_CONNECT_SVC:8083/connectors
```

* response:

```
HTTP/1.1 200 OK
Accept:application/json
["inventory-connector"]
```

Create Debezium Connector

A. Create Debezium Connector - Using RESTful API (Issue with AMQ Streams Operator)

The following worked but the AMQ Streams/Strimzi operator kept on deleting the resource so we went with option B. Create Debezium Connector - Using CR (Custom Resource)

** request:

```
        oc exec -i events-cluster-kafka-0 -- curl -X POST \
        -H "Accept:application/json" \
        -H "Content-Type:application/json" \
        http://$DEBEZIUM_CONNECT_SVC:8083/connectors -d @- <<'EOF'
        {
                "name": "rhpam-connector",
                "config": {
                        "connector.class": "io.debezium.connector.mysql.MySqlConnector",
                        "tasks.max": "1",
                        "database.hostname": "pam-mysql",
                        "database.port": "3306",
                        "database.user": "root",
                        "database.password": "",
                        "database.server.id": "184054",
                        "database.server.name": "processes",
                        "database.include.list": "jbpm",
                        "table.include.list" : "jbpm.Task, jbpm.TaskEvent, jbpm.TaskEvent",
                        "database.history.kafka.bootstrap.servers":
"events-cluster-kafka-bootstrap:9092",
                        "database.history.kafka.topic": "schema-changes.processes"
                        "transforms": "route",
                              "transforms.route.type":
"org.apache.kafka.connect.transforms.RegexRouter",
                              "transforms.route.regex": "([^.]+)\\.([^.]+)\\.([^.]+)",
                              "transforms.route.replacement": "$3"
                    }
          }
          EOF

oc exec -i events-cluster-kafka-0 -- curl -X POST \
-H "Accept:application/json" \
-H "Content-Type:application/json" \
http://$DEBEZIUM_CONNECT_SVC:8083/connectors --data-binary @- << EOF
{
"name": "rhpam-connector",
"config": {
"connector.class": "io.debezium.connector.mysql.MySqlConnector",
"tasks.max": "1",
"database.hostname": "pam-mysql",
"database.port": "3306",
"database.user": "user",
"database.password": "password",
"database.server.id": "184054",
"database.server.name": "processes",
"database.include.list": "jbpm",
"table.include.list" : "jbpm.Task, jbpm.TaskEvent, jbpm.TaskEvent",
"database.history.kafka.bootstrap.servers": "events-cluster-kafka-bootstrap:9092",
"database.history.kafka.topic": "schema-changes.processes",
"transforms": "route",
"transforms.route.type": "io.debezium.transforms.ByLogicalTableRouter",
"transforms.route..topic.regex": "*",
"transforms.route.topic.replacement": "task_all_events"
}
}
EOF



oc exec -i events-cluster-kafka-0 -- curl -X POST -H "Accept:application/json" -H
"Content-Type:application/json" http://$DEBEZIUM_CONNECT_SVC:8083/connectors
--data-binary @- << EOF
{
"name": "rhpam3-connector",
"config": {
```

```
"connector.class": "io.debezium.connector.mysql.MySqlConnector",
"tasks.max": "1",
"database.hostname": "dbz-14-pam-mysql",
"database.port": "3306",
"database.user": "root",
"database.password": "debezium",
"database.server.id": "3184054",
"database.server.name": "rhpam3",
"database.include.list": "inventory",
"table.include.list" : "inventory.Task, inventory.TaskEvent",
"database.history.kafka.bootstrap.servers": "events-cluster-kafka-bootstrap:9092",
"database.history.kafka.topic": "schema-changes.rhpam3"
}
}
EOF


** response:

{"name":"rhpam-connector","config":{"connector.class":"io.debezium.connector.mysql.MySqlC
onnector","tasks.max":"1","database.hostname":"dbz-14-pam-mysql","database.port":"3306","
database.user":"root","database.password":"debezium","database.server.id":"184054","databa
se.server.name":"processes","database.include.list":"inventory","table.include.list":"inventory.T
ask, inventory.TaskEvent,
inventory.TaskEvent","database.history.kafka.bootstrap.servers":"events-cluster-kafka-bootstr
ap:9092","database.history.kafka.topic":"schema-changes.processes","transforms":"route","tra
nsforms.route.type":"io.debezium.transforms.ByLogicalTableRo100  1560  100   786  100
774   1336   1316 --:--:-- --:--:-- --:--:--
1339:"task_all_events","name":"rhpam-connector"},"tasks":[],"type":"source"}
```

*WARNING: * A Problem occurred with the connector being removed ... Possible reason "suppose that's the Strimzi operator battling against a resource created via REST"

B. Create Debezium Connector - Using CR (Custom Resource)

See Appendix B - Kafka Connect with Debezium plugins IMAGE Creation and Deployment

```
oc apply -f - << EOF
```

```
apiVersion: kafka.strimzi.io/v1alpha1
kind: KafkaConnector
metadata:
  name: rhpam-connector
  namespace: dev-demo
  labels:
    strimzi.io/cluster: debezium-connect
    app: rhpam
spec:
  class: io.debezium.connector.mysql.MySqlConnector
  tasksMax: 1
  config:
    database.hostname: 172.30.88.1
    database.port: 3306
    database.user: root
    database.password: debezium
    database.server.id: 184054
    database.server.name: rhpam
    database.include.list: jbpm
    table.include.list: 'jbpm.Task,jbpm.TaskEvent,jbpm.TaskVariableImpl'
    database.history.kafka.bootstrap.servers: events-cluster-kafka-bootstrap:9092
    database.history.kafka.topic: schema-changes.rhpam
EOF
```

From Operator Console

```
apiVersion: kafka.strimzi.io/v1alpha1
kind: KafkaConnector
metadata:
  name: rhpam-connector
  namespace: dev-demo
  labels:
    strimzi.io/cluster: debezium-connect
    app: rhpam
spec:
  class: io.debezium.connector.mysql.MySqlConnector
  tasksMax: 1
  config:
    database.hostname: dbz-14-pam-mysql
    database.port: 3306
    database.user: root
    database.password: debezium
    database.server.id: 184054
    database.server.name: rhpam
    database.include.list: inventory
    table.include.list:
'inventory.Task,inventory.TaskEvent,inventory.TaskVariableImpl'
    database.history.kafka.bootstrap.servers: events-cluster-kafka-bootstrap:9092
    database.history.kafka.topic: schema-changes.rhpam
    key.converter.schemas.enable: false
    value.converter.schemas.enable: false
```

- Test the connector by creating 2 consumers to show the Change Event Messages published on the Kafka Topics
- Find KafkaTopics

```
$ oc get KafkaTopic
NAME                                                                          CLUSTER          PARTITIONS
REPLICATION FACTOR
connect-cluster-configs                                                       events-cluster   1            3
connect-cluster-offsets                                                       events-cluster   25           3
connect-cluster-status                                                        events-cluster   5            3
consumer-offsets---84e7a678d08f4bd226872e5cdd4eb527fadc1c6a                   events-cluster   50           3
rhpam                                                                         events-cluster   1            1
rhpam.inventory.task---f68d02765129d9af74d459c93d1cd20f9660c6d7               events-cluster   1            1
rhpam.inventory.taskevent---8d8523294316cd052c7becae4e9f8e9c20c73254          events-cluster   1            1
rhpam.inventory.taskvariableimpl---29472ca4ef1328558f839e9a94f2c4bdc248ce12   events-cluster   1            1
rhpam.jbpm.task---bc221859bfa76b6c8ce81b8762f02087406def5                     events-cluster   1            1
rhpam.jbpm.taskevent---828bc7d928f361cad2e60dc68b28a37eed460c0c               events-cluster   1            1
rhpam.jbpm.taskvariableimpl---89ddfbf83791abf955d1516daede574cf1ffc3d8        events-cluster   1            1
schema-changes.rhpam                                                          events-cluster   1            1
```

- Enter one of the KAFKA pods (*oc rsh events-cluster-kafka-0*) and list the kafka topics

```
$ ./kafka-topics.sh --bootstrap-server localhost:9092 --list
__consumer_offsets
connect-cluster-configs
connect-cluster-offsets
connect-cluster-status
rhpam
rhpam.inventory.Task
rhpam.inventory.TaskEvent
rhpam.inventory.TaskVariableImpl
rhpam.jbpm.Task
rhpam.jbpm.TaskEvent
rhpam.jbpm.TaskVariableImpl
rhpam6.jbpm.Task
schema-changes.rhpam
```

- Check the messages published per table in each topic

POD	Table	Command
oc rsh events-cluster-kafka-0	`Task`	`./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic` **`rhpam.inventory.TaskEvent`** `--from-beginning`
oc rsh events-cluster-kafka-1	`TaskEvent`	`./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic` **`rhpam.inventory.Task`** `--from-beginning`

oc rsh events-cluster-kafka-2	`TaskVariableImpl`	`./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic` **`rhpam.inventory.TaskVariableImpl`** `--from-beginning`

- Delete via KafkaTopic CRD any non required topics (KafkaConnector deletion does not remove them)
  - `oc delete KafkaTopic rhpam.inventory.Task`
  - `oc delete KafkaTopic rhpam.inventory.TaskEvent`
  - `oc delete KafkaTopic rhpam.inventory.TaskVariableImpl`
  -

Creating Consumer of CDC Kafka Messages & Storage in DB

Code at: rhpam-cdc-service/ GIT Repository

Create App DB to store APP view of TaskVariable Events

- PostgreSQL Template Details

```
$ oc process --parameters -n openshift postgresql-persistent


NAME                     DESCRIPTION                                                  GENERATOR         VALUE
MEMORY_LIMIT             Maximum amount of memory the container can use.                                512Mi
NAMESPACE                The OpenShift Namespace where the ImageStream resides.                           openshift
DATABASE_SERVICE_NAME    The name of the OpenShift Service exposed for the database.
postgresql
POSTGRESQL_USER          Username for PostgreSQL user that will be used
                         for accessing the database.                         expression
user[A-Z0-9]{3}
POSTGRESQL_PASSWORD      Password for the PostgreSQL connection user.
                                                                             Expression
[a-zA-Z0-9]{16}
POSTGRESQL_DATABASE      Name of the PostgreSQL database accessed.                                       sampledb
VOLUME_CAPACITY          Volume space available for data, e.g. 512Mi, 2Gi.                               1Gi
POSTGRESQL_VERSION       Version of PostgreSQL image to be used                                          10-el8
```

- Create PostgreSQL POD from template

```
oc new-app --template=postgresql-persistent -p
DATABASE_SERVICE_NAME=taskdetails-postgresql -p
POSTGRESQL_USER=postgresrhpamuser -p POSTGRESQL_PASSWORD=postgresrhpampwd -p
POSTGRESQL_DATABASE=taskdetails -l app=task-details-db
```

- Check PSQL Setup

Enter POD	oc rsh <`taskdetails-postgresql Pod name>`
Authenticate to DB	psql -U postgresrhpamuser -W postgresrhpampwd -d taskdetails

Check DBs

```
taskdetails=> \l
                                       List of databases
    Name     |       Owner       | Encoding |  Collate   |   Ctype    |
Access privileges
-------------+-------------------+----------+------------+------------+-
----------------------
 postgres    | postgres          | UTF8     | en_US.utf8 | en_US.utf8 |
 taskdetails | postgresrhpamuser | UTF8     | en_US.utf8 | en_US.utf8 |
 template0   | postgres          | UTF8     | en_US.utf8 | en_US.utf8 |
=c/postgres        +
             |                   |          |            |            |
postgres=CTc/postgres
 template1   | postgres          | UTF8     | en_US.utf8 | en_US.utf8 |
=c/postgres        +
             |                   |          |            |            |
postgres=CTc/postgres
(4 rows)
```

Configure & Deploy Quarkus CDC Cosumer App

1. Configure App https://github.com/skoussou/cdc-playground/tree/main/rhpam-cdc-service

src/main/resources/application.properties

```
quarkus.datasource.url=jdbc:postgresql://taskdetails-postgresql:5432/taskdetails?cur
rentSchema=public
quarkus.datasource.username=postgresrhpamuser
quarkus.datasource.password=postgresrhpampwd
quarkus.hibernate-orm.database.generation=drop-and-create
quarkus.hibernate-orm.dialect=org.hibernate.dialect.PostgreSQLDialect
quarkus.hibernate-orm.log.sql=true

mp.messaging.incoming.taskdetails.connector=smallrye-kafka
#mp.messaging.incoming.orders.topic=Order.events
#mp.messaging.incoming.orders.bootstrap.servers=kafka:9092
mp.messaging.incoming.taskdetails.topic=rhpam.jbpm.TaskVariableImpl
mp.messaging.incoming.taskdetails.bootstrap.servers=events-cluster-kafka-bootstrap:9
092

mp.messaging.incoming.taskdetails.group.id=taskdetails-service
mp.messaging.incoming.taskdetails.key.deserializer=org.apache.kafka.common.serializa
```

```
tion.StringDeserializer
mp.messaging.incoming.taskdetails.value.deserializer=org.apache.kafka.common.seriali
zation.StringDeserializer
...
```

2. Deploy

```
 mvn  clean package -Dquarkus.kubernetes.deploy=true
-Dquarkus.openshift.expose=true -Dquarkus.kubernetes-client.trust-certs=true
```

3. Check DB Table Relations created

```
taskdetails=> \dt
                  List of relations
 Schema |      Name       | Type  |       Owner
--------+-----------------+-------+-------------------
 public | consumedmessage | table | postgresrhpamuser
 public | shipment        | table | postgresrhpamuser
 public | taskvariables   | table | postgresrhpamuser
```

4. Check consumed TaskVariableImpl events from Topic **`rhpam.jbpm.TaskVariableImpl`** result in App DB Entries

```
curl -u user:user -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' -d '{ "taskOwner" : "user", "pImporantVar" : "level-2"}'
'http://business-application-service-dev-demo.apps.cluster-demo-d3f8.demo-d3f8.examp
le.opentlc.com/rest/server/containers/simple-process-kjar-1.0.8/processes/ht-basics.
simple-ht/instances'
```

```
taskdetails=> select * from taskvariables;
 id |     changedate      |      name       | proceinstanceid | taskid |  value
----+---------------------+-----------------+-----------------+--------+---------
  1 | 2021-01-06 14:13:31 | tImportantVarIn |               2 |      2 | level-2
```

```
curl -u user:user -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' -d '{ "taskOwner" : "user", "pImporantVar" : "level-3"}'
'http://business-application-service-dev-demo.apps.cluster-demo-d3f8.demo-d3f8.examp
le.opentlc.com/rest/server/containers/simple-process-kjar-1.0.8/processes/ht-basics.
simple-ht/instances'
```

```
taskdetails=> select * from taskvariables;
 id |     changedate      |      name       | proceinstanceid | taskid |  value
----+---------------------+-----------------+-----------------+--------+---------
  1 | 2021-01-06 14:13:31 | tImportantVarIn |               2 |      2 | level-2
  2 | 2021-01-06 14:14:45 | tImportantVarIn |               3 |      3 | level-3
```

5.

```
io.deb.exa.out.shi.fac.KafkaEventConsumer] (Thread-4) Kafka message with key =
{"schema":{"type":"struct","fields":[{"type":"int64","optional":false,"field":"
id"}],"optional":false,"name":"rhpam6.inventory.TaskVariableImpl.Key"},"payload
":{"id":2}} arrived
2020-12-11 10:26:58,790 INFO  [io.deb.exa.out.shi.fac.KafkaEventConsumer]
(Thread-4) Kafka message with payload =
{"schema":{"type":"struct","fields":[{"type":"struct","fields":[{"type":"int64"
,"optional":false,"field":"id"},{"type":"int64","optional":true,"name":"io.debe
zium.time.Timestamp","version":1,"field":"modificationDate"},{"type":"string","
optional":true,"field":"name"},{"type":"string","optional":true,"field":"proces
sId"},{"type":"int64","optional":true,"field":"processInstanceId"},{"type":"int
64","optional":true,"field":"taskId"},{"type":"int32","optional":true,"field":"
type"},{"type":"string","optional":true,"field":"value"}],"optional":true,"name
":"rhpam6.inventory.TaskVariableImpl.Value","field":"before"},{"type":"struct",
"fields":[{"type":"int64","optional":false,"field":"id"},{"type":"int64","optio
nal":true,"name":"io.debezium.time.Timestamp","version":1,"field":"modification
Date"},{"type":"string","optional":true,"field":"name"},{"type":"string","optio
nal":true,"field":"processId"},{"type":"int64","optional":true,"field":"process
InstanceId"},{"type":"int64","optional":true,"field":"taskId"},{"type":"int32",
"optional":true,"field":"type"},{"type":"string","optional":true,"field":"value
"}],"optional":true,"name":"rhpam6.inventory.TaskVariableImpl.Value","field":"a
fter"},{"type":"struct","fields":[{"type":"string","optional":false,"field":"ve
rsion"},{"type":"string","optional":false,"field":"connector"},{"type":"string"
,"optional":false,"field":"name"},{"type":"int64","optional":false,"field":"ts_
ms"},{"type":"string","optional":true,"name":"io.debezium.data.Enum","version":
1,"parameters":{"allowed":"true,last,false"},"default":"false","field":"snapsho
t"},{"type":"string","optional":false,"field":"db"},{"type":"string","optional"
:true,"field":"table"},{"type":"int64","optional":false,"field":"server_id"},{"
type":"string","optional":true,"field":"gtid"},{"type":"string","optional":fals
e,"field":"file"},{"type":"int64","optional":false,"field":"pos"},{"type":"int3
2","optional":false,"field":"row"},{"type":"int64","optional":true,"field":"thr
ead"},{"type":"string","optional":true,"field":"query"}],"optional":false,"name
":"io.debezium.connector.mysql.Source","field":"source"},{"type":"string","opti
onal":false,"field":"op"},{"type":"int64","optional":true,"field":"ts_ms"},{"ty
pe":"struct","fields":[{"type":"string","optional":false,"field":"id"},{"type":
"int64","optional":false,"field":"total_order"},{"type":"int64","optional":fals
e,"field":"data_collection_order"}],"optional":true,"field":"transaction"}],"op
tional":false,"name":"rhpam6.inventory.TaskVariableImpl.Envelope"},"payload":{"
before":null,"after":{"id":2,"modificationDate":1607680532000,"name":"tImportan
tVarIn","processId":"ht-basics.simple-ht","processInstanceId":2,"taskId":2,"typ
e":0,"value":"Level-0"},"source":{"version":"1.3.1.Final","connector":"mysql","
name":"rhpam6","ts_ms":1607680532000,"snapshot":"false","db":"inventory","table
":"TaskVariableImpl","server_id":223344,"gtid":null,"file":"mysql-bin.000003","
pos":123187,"row":0,"thread":null,"query":null},"op":"c","ts_ms":1607680532198,
"transaction":null}} arrived
```

APPENDIX A - RHPAM TABLES STATES

TASK CREATION

```
MariaDB [rhpam79]> select parentProcessInstanceId, processName, status from
ProcessInstanceLog;
+-------------------------+-------------+--------+
| parentProcessInstanceId | processName | status |
+-------------------------+-------------+--------+
|                      -1 | simple-ht   |      1 |
+-------------------------+-------------+--------+

MariaDB [rhpam79]> select * from VariableInstanceLog;
+----+---------------------+-------------------------+----------+--------------------+------
-------------+----------+-------------+--------------------+
| id | log_date            | externalId              | oldValue | processId          |
processInstanceId | value    | variableId   | variableInstanceId |
+----+---------------------+-------------------------+----------+--------------------+------
-------------+----------+-------------+--------------------+
|  1 | 2020-11-24 10:30:27 | ht-basics_1.0.0-SNAPSHOT |          | ht-basics.simple-ht |
1 | pamAdmin | taskOwner    | taskOwner          |
|  2 | 2020-11-24 10:30:27 | ht-basics_1.0.0-SNAPSHOT |          | ht-basics.simple-ht |
1 | Level-3  | pImporantVar | pImporantVar       |
|  3 | 2020-11-24 10:30:27 | ht-basics_1.0.0-SNAPSHOT |          | ht-basics.simple-ht |
1 | pamAdmin | initiator    | initiator          |
+----+---------------------+-------------------------+----------+--------------------+------
-------------+----------+-------------+--------------------+

MariaDB [rhpam79]> select id, formName, previousStatus, status, taskType, processInstanceId,
actualOwner_id from Task where processInstanceId=1;
+----+----------+----------------+----------+----------+-------------------+----------------+
| id | formName | previousStatus | status   | taskType | processInstanceId | actualOwner_id |
+----+----------+----------------+----------+----------+-------------------+----------------+
|  1 | Task     |              0 | Reserved | NULL     |                 1 | pamAdmin       |
+----+----------+----------------+----------+----------+-------------------+----------------+
MariaDB [rhpam79]> select * from TaskEvent where processInstanceId=1;
+----+----------------+---------------------+---------+-------------------+-------------+-----
---+-----------+---------------------+---------+------------+
| id | correlationKey | logTime             | message | processInstanceId | processType |
taskId | type      | userId              | OPTLOCK | workItemId |
+----+----------------+---------------------+---------+-------------------+-------------+-----
---+-----------+---------------------+---------+------------+
|  1 | NULL           | 2020-11-24 10:30:27 | NULL    |                 1 |        NULL |
1 | ADDED     | ht-basics.simple-ht |       0 |          1 |
|  2 | NULL           | 2020-11-24 10:30:27 | NULL    |                 1 |        NULL |
1 | ACTIVATED | pamAdmin            |       0 |          1 |
+----+----------------+---------------------+---------+-------------------+-------------+-----
---+-----------+---------------------+---------+------------+

MariaDB [rhpam79]> select * from TaskVariableImpl;
+----+---------------------+-----------------+---------------------+-------------------+------
--+------+---------+
| id | modificationDate    | name            | processId           | processInstanceId |
taskId | type | value   |
+----+---------------------+-----------------+---------------------+-------------------+------
--+------+---------+
|  1 | 2020-11-24 10:30:27 | tImportantVarIn | ht-basics.simple-ht |                 1 |
1 |    0 | Level-3 |
+----+---------------------+-----------------+---------------------+-------------------+------
--+------+---------+

MariaDB [rhpam79]> select * from Content;
+----+----------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
-----------------------------------+
| id | content
|
+----+----------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
-----------------------------------+
|  1 | �� z  +
, J Horg.drools.core.marshalling.impl.SerializablePlaceholderResolverStrategy�e�� sr
java.util.ArrayListx����a� I sizexp   w   t falsetpamAdmint Taskt Taskt Level-3xRk

       Skippable �

ActorId �

TaskName �

NodeName �

tImportantVarIn �              |
+----+----------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
-----------------------------------+


MariaDB [rhpam79]> select * from BAMTaskSummary;
+----+---------------------+----------+---------+-------------------+-----------+----------+--
------+----------+----------+---------+
| pk | createdDate         | duration | endDate | processInstanceId | startDate | status   |
taskId | taskName | userId   | OPTLOCK |
+----+---------------------+----------+---------+------------------+----------+----------+--
------+----------+----------+---------+
|  1 | 2020-11-24 10:30:27 |     NULL | NULL    |                1 | NULL      | Reserved |
1 | Task     | pamAdmin |       0 |
+----+---------------------+----------+---------+------------------+----------+----------+--
------+----------+----------+---------+

MariaDB [rhpam79]> select createdOn, activationTime, lastModificationDate, taskId,
actualOwner, status, workItemId, processId, processInstanceId, ProcessSessionId from
AuditTaskImpl;
+---------------------+---------------------+----------------------+--------+-------------+---
-------+------------+---------------------+-------------------+------------------+
| createdOn           | activationTime      | lastModificationDate | taskId | actualOwner |
status   | workItemId | processId           | processInstanceId | ProcessSessionId |
+---------------------+---------------------+----------------------+--------+-------------+---
-------+------------+---------------------+-------------------+------------------+
| 2020-11-24 10:30:27 | 2020-11-24 10:30:27 | 2020-11-24 10:30:27  |      1 | pamAdmin    |
Reserved |          1 | ht-basics.simple-ht |                 1 |                1 |
+---------------------+---------------------+----------------------+--------+-------------+---
-------+------------+---------------------+-------------------+------------------+
```

TASK COMPLETION

```
MariaDB [rhpam79]> select parentProcessInstanceId, processName, status from
ProcessInstanceLog;
+-------------------------+------------+--------+
| parentProcessInstanceId | processName | status |
+-------------------------+------------+--------+
|                      -1 | simple-ht   |      2 |
+-------------------------+------------+--------+

MariaDB [rhpam79]> select * from VariableInstanceLog;
+----+---------------------+---------------------------+----------+---------------------+------
-------------+----------+-------------+-------------------+
| id | log_date            | externalId                | oldValue | processId           |
processInstanceId | value    | variableId    | variableInstanceId |
+----+---------------------+---------------------------+----------+---------------------+------
-------------+----------+-------------+-------------------+
|  1 | 2020-11-24 10:30:27 | ht-basics_1.0.0-SNAPSHOT |          | ht-basics.simple-ht |
1 | pamAdmin | taskOwner    | taskOwner          |
|  2 | 2020-11-24 10:30:27 | ht-basics_1.0.0-SNAPSHOT |          | ht-basics.simple-ht |
1 | Level-3   | pImporantVar | pImporantVar       |
|  3 | 2020-11-24 10:30:27 | ht-basics_1.0.0-SNAPSHOT |          | ht-basics.simple-ht |
1 | pamAdmin | initiator    | initiator          |
|  4 | 2020-11-24 10:50:22 | ht-basics_1.0.0-SNAPSHOT | Level-3  | ht-basics.simple-ht |
1 | Level-5   | pImporantVar | pImporantVar       |
+----+---------------------+---------------------------+----------+---------------------+------
-------------+----------+-------------+-------------------+

MariaDB [rhpam79]> select id, formName, previousStatus, status, taskType, processInstanceId,
actualOwner_id from Task where processInstanceId=1;
Empty set (0.001 sec)

MariaDB [rhpam79]> select * from TaskEvent where processInstanceId=1;
+----+----------------+---------------------+--------------------------+-------------------+--
-----------+--------+-----------+--------------------+---------+------------+
| id | correlationKey | logTime             | message                  | processInstanceId |
processType | taskId | type      | userId             | OPTLOCK | workItemId |
+----+----------------+---------------------+-------------------------+-------------------+--
-----------+--------+-----------+--------------------+---------+------------+
|  1 | NULL           | 2020-11-24 10:30:27 | NULL                    |                 1 |
NULL |      1 | ADDED     | ht-basics.simple-ht |       0 |          1 |
|  2 | NULL           | 2020-11-24 10:30:27 | NULL                    |                 1 |
NULL |      1 | ACTIVATED | pamAdmin            |       0 |          1 |
|  3 | NULL           | 2020-11-24 10:50:11 | NULL                    |                 1 |
NULL |      1 | STARTED   | pamAdmin            |       0 |          1 |
|  4 | NULL           | 2020-11-24 10:50:22 | Task output data updated |                 1 |
NULL |      1 | UPDATED   | pamAdmin            |       0 |          1 |
|  5 | NULL           | 2020-11-24 10:50:22 | NULL                    |                 1 |
NULL |      1 | COMPLETED | pamAdmin            |       0 |          1 |
+----+----------------+---------------------+-------------------------+-------------------+--
-----------+--------+-----------+--------------------+---------+------------+

MariaDB [rhpam79]> select * from TaskVariableImpl;
+----+---------------------+------------------+---------------------+-------------------+-----
---+------+---------+
| id | modificationDate    | name             | processId           | processInstanceId |
taskId | type | value   |
+----+---------------------+------------------+---------------------+-------------------+-----
---+------+---------+
|  1 | 2020-11-24 10:30:27 | tImportantVarIn  | ht-basics.simple-ht |                 1 |
1 |    0 | Level-3 |
|  2 | 2020-11-24 10:50:22 | tImportantVarOut | ht-basics.simple-ht |                 1 |
1 |    1 | Level-5 |
+----+---------------------+------------------+---------------------+-------------------+-----
---+------+---------+

MariaDB [rhpam79]> select * from Content;

MariaDB [rhpam79]> select * from BAMTaskSummary;
+----+---------------------+----------+---------------------+-------------------+-------------
--------+-----------+--------+----------+----------+---------+
| pk | createdDate         | duration | endDate             | processInstanceId | startDate
| status    | taskId | taskName | userId   | OPTLOCK |
+----+---------------------+----------+---------------------+-------------------+-------------
--------+-----------+--------+----------+----------+---------+
|  1 | 2020-11-24 10:30:27 |    11102 | 2020-11-24 10:50:22 |                 1 | 2020-11-24
10:50:11 | Completed |      1 | Task     | pamAdmin |       2 |
+----+---------------------+----------+---------------------+-------------------+-------------
--------+-----------+--------+----------+----------+---------+

MariaDB [rhpam79]> select createdOn, activationTime, lastModificationDate, taskId,
actualOwner, status, workItemId, processId, processInstanceId, ProcessSessionId from
AuditTaskImpl;
+---------------------+---------------------+----------------------+--------+-------------+---
--------+------------+---------------------+-------------------+------------------+
| createdOn           | activationTime      | lastModificationDate | taskId | actualOwner |
status    | workItemId | processId           | processInstanceId | ProcessSessionId |
+---------------------+---------------------+----------------------+--------+-------------+---
--------+------------+---------------------+-------------------+------------------+
| 2020-11-24 10:30:27 | 2020-11-24 10:30:27 | 2020-11-24 10:50:22  |      1 | pamAdmin    |
Completed |          1 | ht-basics.simple-ht |                 1 |                1 |
+---------------------+---------------------+----------------------+--------+-------------+---
--------+------------+---------------------+-------------------+------------------+
```

APPENDIX B - CRDs for OCP CDC Setup

Appendix B - Kafka Cluster Setup

AMQ Streams Kafka CRD

```
apiVersion: kafka.strimzi.io/v1beta1
kind: Kafka
metadata:
 name: events-cluster
 namespace: dev-demo
spec:
 kafka:
   config:
     offsets.topic.replication.factor: 3
     transaction.state.log.min.isr: 2
     transaction.state.log.replication.factor: 3
     log.message.format.version: '2.6'
   version: 2.6.0
   listeners:
     - name: plain
       port: 9092
```

```yaml
      tls: false
      type: internal
    - name: tls
      port: 9093
      tls: true
      type: internal
  replicas: 3
  storage:
    type: ephemeral
entityOperator:
  topicOperator: {}
  userOperator: {}
zookeeper:
  replicas: 3
  storage:
    type: ephemeral
```

Appendix B - Kafka Connect with Debezium plugins IMAGE Creation and Deployment

Preparation

```
export IMG_NAME="debezium-connect"
export DEBEZIUM_VERSION=1.3.1.Final

mkdir -p plugins && cd plugins && \
for PLUGIN in {mongodb,mysql,postgres}; do \
curl
https://repo1.maven.org/maven2/io/debezium/debezium-connector-$PLUGIN/$DEBEZIUM_VERS
ION/debezium-connector-$PLUGIN-$DEBEZIUM_VERSION-plugin.tar.gz | tar xz; \
done
```

https://access.redhat.com/documentation/en-us/red_hat_amq/7.7/html/using_amq_streams_on_openshift/getting-started-str#using-kafka-connect-with-plug-ins-str

AMQ Streams - Product Based Image (See Red Hat Container Catalogue)

```
cat <<EOF > Dockerfile
FROM registry.redhat.io/amq7/amq-streams-kafka-26-rhel7:1.6.0
```

```
USER root:root
COPY ./plugins/ /opt/kafka/plugins/
USER 1001
EOF
```

Strimzi - Community Based

```
cat <<EOF > Dockerfile
FROM strimzi/kafka:0.20.0-kafka-2.6.0
USER root:root
COPY ./plugins/ /opt/kafka/plugins/
USER 1001
EOF
```

Build

```
oc new-build --binary --name=$IMG_NAME -l app=$IMG_NAME
oc patch bc/$IMG_NAME -p
'{"spec":{"strategy":{"dockerStrategy":{"dockerfilePath":"Dockerfile"}}}}'
oc start-build $IMG_NAME --from-dir=. --follow  --follow --loglevel=8
--build-loglevel=8
```

KAFKA CONNECT/DEBEZIUM CREATION

```
oc create -f - <<EOF
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaConnect
metadata:
  name: debezium-connect
  annotations:
    strimzi.io/use-connector-resources: "true"
spec:
  replicas: 1
  version: latest
  image:
"image-registry.openshift-image-registry.svc:5000/dev-demo/debezium-connect"
  bootstrapServers: events-cluster-kafka-bootstrap:9093
  tls:
    trustedCertificates:
      - secretName: events-cluster-cluster-ca-cert
        certificate: ca.crt
EOF
```

Read "Kafka Connect configuration for multiple instances" if multiple KAFKA Connect in place and change the default configuration of the following config properties:

```
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaConnect
metadata:
  name: my-connect
spec:
  # ...
  config:
    group.id: connect-cluster (1)
    offset.storage.topic: connect-cluster-offsets (2)
    config.storage.topic: connect-cluster-configs (3)
    status.storage.topic: connect-cluster-status  (4)
    # ...
# ...
```

1. Kafka Connect cluster group that the instance belongs to.
2. Kafka topic that stores connector offsets.
3. Kafka topic that stores connector and task status configurations.
4. Kafka topic that stores connector and task status updates.

NOTE — Values for the three topics must be the same for all Kafka Connect instances with the same group.id.

Unless you change the default settings, each Kafka Connect instance connecting to the same Kafka cluster is deployed with the same values. What happens, in effect, is all instances are coupled to run in a cluster and use the same topics.
If multiple Kafka Connect clusters try to use the same topics, Kafka Connect will not work as expected and generate errors.
If you wish to run multiple Kafka Connect instances, change the values of these properties for each instance.