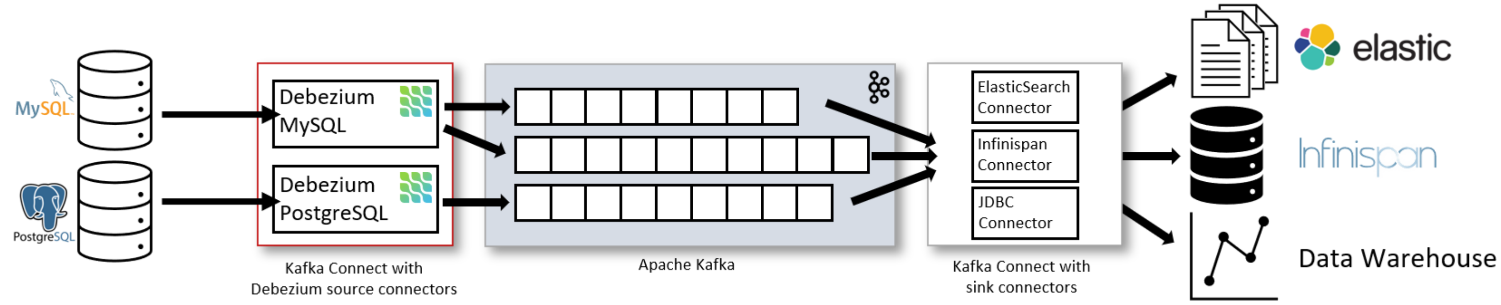
Debezium on OpenShift

This cheat sheet covers how to deploy/create/run/update a Debezium Connector on OpenShift.

WHAT'S DEBEZIUM ?

Debezium is an distributed open-source platform for change data capture. Start it up, point it at your databases, and your apps can start responding to all of the inserts, updates, and deletes that other apps commit to your databases. Debezium is durable and fast, so your apps can respond quickly and never miss an event, even when things go wrong.



DEPLOYMENT

Debezium is based on Apache Kafka and Kafka Connect, and can be run on Kubernetes and OpenShift via the Strimzi project. Strimzi provides a set of operators and container images for running Kafka on Kubernetes and OpenShift.

Deploy Kafka & Kafka Connect

```
oc new-project myproject
# install the Strimzi operator
oc apply -f https://github.com/strimzi/strimzi-kafka-operator/releases/download/0.19.0/strimzi-cluster-operator-0.19.0.yaml
# Deploy a single node Kafka broker
oc apply -f https://github.com/strimzi/strimzi-kafka-operator/raw/0.19.0/examples/kafka/kafka-persistent-single.yaml
# Deploy a single instance of Kafka Connect with no plug-in installed
oc apply -f https://github.com/strimzi/strimzi-kafka-operator/raw/0.19.0/examples/connect/kafka-connect-s2i-single-node-kafka.yaml
```

Extend Kafka Connect with Debezium Binaries:

Source-to-Image (S2I):

```
export DEBEZIUM_VERSION=1.2.0.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_VERSION/debezium-connector-$PLUGIN-$DEBEZIUM_VERSION-plugin.tar.gz | tar xz; \
done && \
oc start-build my-connect-cluster-connect --from-dir=. --follow && \
cd .. && rm -rf plugins
```

Docker:

```
export IMG_NAME="debezium-connect"
export DEBEZIUM_VERSION=1.2.0.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_VERSION/debezium-connector-$PLUGIN-$DEBEZIUM_VERSION-plugin.tar.gz | tar xz; \
done
cd ..
cat <<EOF > Dockerfile
FROM strimzi/kafka:0.19.0-kafka-2.5.0
USER root:root
COPY ./plugins/ /opt/kafka/plugins/
USER 1001
EOF

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

oc create -f - <<EOF
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaConnect
metadata:
  name: $IMG_NAME
  annotations:
    strimzi.io/use-connector-resources: "true"
spec:
  replicas: 1
  version: 2.5.0
  image: "image-registry.openshift-image-registry.svc:5000/myproject/$IMG_NAME"
  bootstrapServers: my-cluster-kafka-bootstrap:9093
  tls:
    trustedCertificates:
      - secretName: my-cluster-cluster-ca-cert
        certificate: ca.crt
EOF
rm -rf plugins && rm Dockerfile
```

COMMANDS

All of Debezium's connectors are Kafka Connector source connectors, and as such they can be deployed and managed using the Kafka Connect service. A Kafka Connect service has a RESTful API for managing and deploying connectors; the service can be clustered and will automatically distribute the connectors across the cluster, e.g. ensuring that the connector will be seamlessly restarted after a node failure.

```
export DEBEZIUM_CONNECT_SVC=my-connect-cluster-connect-api
# choose the kafka connect service by running oc get svc -l app.kubernetes.io/name=kafka-connect -o json | jq -r '.items[] | .metadata.name'
export CONNECTOR=inventory-connector
```

Check the available connector plugins:

GET /connector-plugins	check the available connector plugins

request:

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

response:

```
HTTP/1.1 200 OK
Accept:application/json
{"class":"io.debezium.connector.mongodb.MongoDbConnector","type":"source","version":"1.2.0.Final"},
{"class":"io.debezium.connector.mysql.MySqlConnector","type":"source","version":"1.2.0.Final"},
{"class":"io.debezium.connector.postgresql.PostgresConnector","type":"source","version":"1.2.0.Final"},
{"class":"org.apache.kafka.connect.file.FileStreamSinkConnector","type":"sink","version":"2.5.0"},
{"class":"org.apache.kafka.connect.file.FileStreamSourceConnector","type":"source","version":"2.5.0"},
{"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"}
```

Get all connectors:

GET /connectors Get a list of active connectors

request:

```
oc exec -i my-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

Using RESTful API

POST /connectors Create a new Debezium connector

request:

```
oc exec -i my-cluster-kafka-0 -- curl -X POST \
   -H "Accept:application/json" \
   -H "Content-Type:application/json" \
   http://$DEBEZIUM_CONNECT_SVC:8083/connectors -d @- <<'EOF'
{
   "name": "inventory-connector",
   "config": {
      "connector.class": "io.debezium.connector.mysql.MySqlConnector",
      "tasks.max": "1",
      "database.hostname": "mysql",
      "database.port": "3306",
      "database.user": "debezium",
      "database.password": "dbz",
      "database.server.id": "184054",
      "database.server.name": "dbserver",
      "database.whitelist": "inventory",
      "database.history.kafka.bootstrap.servers": "my-cluster-kafka-bootstrap:9092",
        "database.history.kafka.topic": "schema-changes.inventory"
    }
}
EOF
```

response:

```
HTTP/1.1 201 Created
Accept:application/json
{"name":"inventory-connector","config":
{"connector.class":"io.debezium.connector.mysql.MySqlConnector","tasks.max":"1","database.hostname":"mysql","database.port":"3306","database.
user":"debezium","database.password":"dbz","database.server.id":"184054","database.server.name":"dbserver","database.whitelist":"inventory","dat
abase.history.kafka.bootstrap.servers":"my-cluster-kafka-bootstrap:9092","database.history.kafka.topic":"schema-
changes.inventory","name":"inventory-connector"},"tasks":[{"connector":"inventory-connector","task":0}],"type":"source"}
```

Using CR (Custom Resource)

If use-connector-resources is enabled for your Kafka Connect resource, you can create the connector instance by creating a specific custom resource:

```
oc apply -f - << EOF
apiVersion: kafka.strimzi.io/v1alpha1
kind: KafkaConnector
metadata:
  name: $CONNECTOR
  namespace: myproject
  labels:
    strimzi.io/cluster: my-connect-cluster
spec:
  class: io.debezium.connector.mysql.MySqlConnector
  tasksMax: 1
  config:
    database.hostname: mysql
    database.port: 3306
    database.user: debezium
    database.password: dbz
    database.server.id: 184054
    database.server.name: dbserver
    database.whitelist: inventory
    database.history.kafka.bootstrap.servers: my-cluster-kafka-bootstrap:9092
    database.history.kafka.topic: schema-changes.inventory
EOF
```

Tip Enable use-connector-resources to instantiate Kafka connectors through specific custom resources: oc annotate kafkaconnects2is my-connect-cluster strimzi.io/use-connector-resources=true

Note

`oc get kctr --selector strimzi.io/cluster=my-connect-cluster -o name`

Check that the resource was created

`oc get kctr/inventory-connector -o yaml | yq read - status`

Check the status of the Debezium Connector from the resource

`oc apply kctr/inventory-connector` or `oc edit kctr/inventory-connector`

Update the Debezium connector CR

`oc delete kctr/inventory-connector`

delete the Debezium connector CR

Get connector configuration

GET /connectors/(string:name)/config	Get the configuration for the connector.

request:

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

response:

```
HTTP/1.1 200 OK
Accept:application/json
{"connector.class":"io.debezium.connector.mysql.MySqlConnector","database.user":"debezium","database.server.id":"184054","database.hostname"
:"mysql","tasks.max":"1","database.history.kafka.bootstrap.servers":"my-cluster-kafka-bootstrap:9092","database.history.kafka.topic":"schema-
changes.inventory","database.password":"dbz","name":"inventory-
connector","database.server.name":"dbserver","database.whitelist":"inventory","database.port":"3306"}
```

Check connector status

GET /connectors/(string:name)/status	Get current status of the connector.

request:

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

response:

```
HTTP/1.1 200 OK
Accept:application/json
{"name":"inventory-connector","connector":{"state":"RUNNING","worker_id":"10.131.0.49:8083"},"tasks":
[{"id":0,"state":"RUNNING","worker_id":"10.131.0.49:8083"}],"type":"source"}
```

Update connector

PUT /connectors/(string:name)/config	Create a new connector using the given configuration, or update the configuration for an existing connector..

request:

```
oc exec -i my-cluster-kafka-0 -- curl -i -X PUT -H "Accept:application/json" -H "Content-Type:application/json"
http://$DEBEZIUM_CONNECT_SVC:8083/connectors/$CONNECTOR/config/ -d @- <<'EOF'
{
      "connector.class": "io.debezium.connector.mysql.MySqlConnector",
      "tasks.max": "1",
      "database.hostname": "mysql",
      "database.port": "3306",
      "database.user": "debezium",
      "database.password": "dbz",
      "database.server.id": "184054",
      "database.server.name": "dbserver",
      "database.whitelist": "inventory",
      "database.history.kafka.bootstrap.servers": "my-cluster-kafka-bootstrap:9092",
      "database.history.kafka.topic": "schema-changes.inventory",
      "include.schema.changes": "false"
  }
}
EOF
```

response:

```
HTTP/1.1 200 OK
Accept:application/json
{"name":"inventory-connector","config":
{"connector.class":"io.debezium.connector.mysql.MySqlConnector","tasks.max":"1","database.hostname":"mysql","database.port":"3306","database.
user":"debezium","database.password":"dbz","database.server.id":"184054","database.server.name":"dbserver","database.whitelist":"inventory","dat
abase.history.kafka.bootstrap.servers":"my-cluster-kafka-bootstrap:9092","database.history.kafka.topic":"schema-
changes.inventory","include.schema.changes":"false","name":"inventory-connector"},"tasks":[{"connector":ta not shown]
"inventory-connector","task":0}],"type":"source"}
```

Restart connector

POST /connectors/(string:name)/restart	Restart the connector.

request:

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

response:

```
HTTP/1.1 204 No Content
Accept:application/json
```

Pause connector

PUT /connectors/(string:name)/pause	Pause the connector and its tasks.

request:

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

response:

```
HTTP/1.1 202 Accepted
Accept:application/json
```

Resume a paused connector

PUT /connectors/(string:name)/resume — Resume a paused connector or do nothing if the connector is not paused.

request:

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

response:

```
HTTP/1.1 202 Accepted
Accept:application/json
```

Delete a connector

DELETE /connectors/(string:name)/ — Delete a connector.

request:

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

response:

```
HTTP/1.1 204 No Content
Accept:application/json
```

LOGS

Change the log level to trace of io.debezium as follows:

```
export KAFKA_CONNECT_POD=my-connect-cluster-connect-2-hns52
oc exec -it $KAFKA_CONNECT_POD -- curl -s -X PUT -H "Content-Type:application/json"  http://localhost:8083/admin/loggers/io.debezium -d '{"level": "TRACE"}'
```

Revert the log level back to INFO as follows:

```
export KAFKA_CONNECT_POD=my-connect-cluster-connect-2-hns52
oc exec -it $KAFKA_CONNECT_POD -- curl -s -X PUT -H "Content-Type:application/json"  http://localhost:8083/admin/loggers/io.debezium -d '{"level": "INFO"}'
```