Kafka Connect

docker start $(docker ps -a -q --filter "status=exited")

docker rm $(docker ps -a -q) —> how to kill all the containers

cd com.docker.devenvironments.code.configuration

curl -i -X POST -H “Accept:application/json” -H “Content-Type:application/json” <http://localhost:8083/connectors/> –data “@debezium.json”

curl -i -X POST -H “Accept:application/json” -H “Content-Type:application/json” [127.0.0.1:8083/connectors/](http://localhost:8083/connectors/) –data “@debezium.json”

docker-compose exec postgres bash -c 'psql -U $POSTGRES\_USER $POSTGRES\_DB-c "select \* from pg\_catalog.pg\_tables"'

export DEBEZIUM\_VERSION=1.8

docker-compose up -d

curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json" http://localhost:8083/connectors/ -d @source.json

curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json" 0.0.0.0:8083/connectors/ -d @source.json

curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json" 172.18.0.7:8083/connectors/ -d @source.json

How to connect to postregs

docker-compose exec postgres bash -c 'psql -U $POSTGRES\_USER $POSTGRES\_DB'

docker-compose exec mysql bash -c 'mysql -u $MYSQL\_USER -p$MYSQL\_PASSWORD'

docker-compose exec mysql bash -c 'mysql -u $MYSQL\_USER -p$MYSQL\_PASSWORD inventory -e'

mysql bash -c 'mysql -u **$MYSQL\_USER -p$MYSQL\_PASSWORD**

docker exec -it a11fb8d22d84 mysql -umysqluser -pmysqlpw login to mysql

docker exec -it mysql-1 /bin/bash

curl -i -X GET -H "Accept:application/json" localhost:8083/connectors/inventory-connector

docker run -it --rm --name watcher --link zookeeper:zookeeper --link kafka:kafka quay.io/debezium/kafka:1.9 watch-topic -a -k dbserver1.inventory.customers

http://localhost:8083/connectors/

{

"name": "jdbc\_source\_mysql\_01",

"config": {

"connector.class": "io.debezium.connector.mysql.MySqlConnector",

"database.hostname": "mysql",

"database.port": "3306",

"database.user": "debezium",

"database.password": "dbz",

"database.server.id": "42",

"database.allowPublicKeyRetrieval":"true",

"database.server.name": "asgard",

"table.whitelist": "demo.customers",

"database.history.kafka.bootstrap.servers": "kafka:29092",

"database.history.kafka.topic": "asgard.dbhistory.demo" ,

"include.schema.changes": "true"

}

}

docker-compose -f docker-compose.yaml exec mysql bash -c 'mysql -u $MYSQL\_USER -p$MYSQL\_PASSWORD inventory -e "select \* from customers"'

docker-compose -f docker-compose.yaml exec postgres bash -c 'psql -U $POSTGRES\_USER $POSTGRES\_DB -c "select \* from customers"'

docker-compose -f docker-compose.yaml exec mysql bash -c 'mysql -u $MYSQL\_USER -p$MYSQL\_PASSWORD inventory'

curl -s XGET http:localhost:8083/connector-plugin| jq '.[].class'

[localhost:8083/connectors](http://localhost:8083/connectors)

curl -X POST -H "Content-Type: application/json" --data

'{"name": "distributed-sql-sink",

"config":

{"connector.class":"io.confluent.connect.jdbc.JdbcSinkConnector",

"tasks.max":"1",

"topics":"psqlproducts",

"connection.url":"jdbc:sqlserver://host:1433;database=testdb;username=user;password=password", "mode":"insert",

"table.name.format":"products",

"Pk.mode":"record\_value",

"pk\_fields":"record\_value","pk.fields":"prod\_id","auto.create": "true" ,

"auto.evolve": "true"}}'

<http://localhost:8083/connectors>

Steps :

Docker login

Enter docker username and password

Docker-compose up -d

psql -U docker -d exampledb -W

Create table student(id integer primary key,name varchar);

Alter table public.student REPLICA IDENTITY FULL

curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json"<http://localhost:8083/connectors/> --data "@debezium.json"

docker run --tty --network kafkaconnect55555\_default confluentinc/cp-kafkacat kafkacat -b kafka:9092 -C -s key=s -s value=avro -r [http://schema-registry:8081](http://schema-registry:8081/) -t postgres.public.student

sh kafka-topics --list --zookeeper localhost:2181

curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json"<http://localhost:8083/connectors/> --data "@debezium.json"

curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json" http://localhost:8083/connectors/ --data "@SimpleOracleCDCLocal.json"

docker run --tty --network kafka\_connect\_default confluentinc/cp-kafkacat kafkacat -b kafka:9092 -C -s key=s -s value=avro -r [http://schema-registry:8081](http://schema-registry:8081/) -t postgres.public.student

You17:43

<https://github.com/debezium/debezium-examples/tree/main/unwrap-smt>

<https://github.com/debezium/debezium-examples/blob/main/unwrap-smt/docker-compose-jdbc.yaml>

You17:47

debezium/connect-jdbc-es:${DEBEZIUM\_VERSION}

You17:50

<https://suchit-g.medium.com/mysql-to-postgresql-using-kafka-connect-ec074ddc9209>

iNSERT INTO STUDENT VALUES(1,'KULDEEP');

docker run --tty --network kafkaconnect55555\_default confluentinc/cp-kafkacat kafkacat -b kafka:9092 -C -s key=s -s value=avro -r http://schema-registry:8081 -t postgres.public.student

Kafka on windows

Download kafka → extract the folder → rename the folder to kafka → copy the folder and place it in c drive

Goto config folder → change the dir to your folder eg new folder where you wants the logs to be created

→ goto kafka folder and run cmd → .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties

→ run this command and run zookeper

kafka-server-start.bat .\config\server.properties

zookeeper-server-start.bat ../../config/zookeeper.properties

D:\kafka\_2.12-3.2.0\bin\windows\zookeeper-server-start.bat D:\kafka\_2.12-3.2.0\config\zookeeper.properties

Oracle

Oracle database curl command

name=oracle-logminer-connector

connector.class=com.ecer.kafka.connect.oracle.OracleSourceConnector

db.name.alias=test

tasks.max=1

topic=cdctest

db.name=testdb

db.hostname=10.1.X.X

db.port=1521

db.user=kminer

db.user.password=kminerpass

db.fetch.size=1

table.whitelist=TEST.\*,TEST2.TABLE2

parse.dml.data=true

reset.offset=true

start.scn=

multitenant=false

su - oracle

How to download oracle image

F:\11111\KafkaConnect8888>docker login container-registry.oracle.com

Username: yadavkuldeep5810@gmail.com

Password:

Login Succeeded

How to run an image

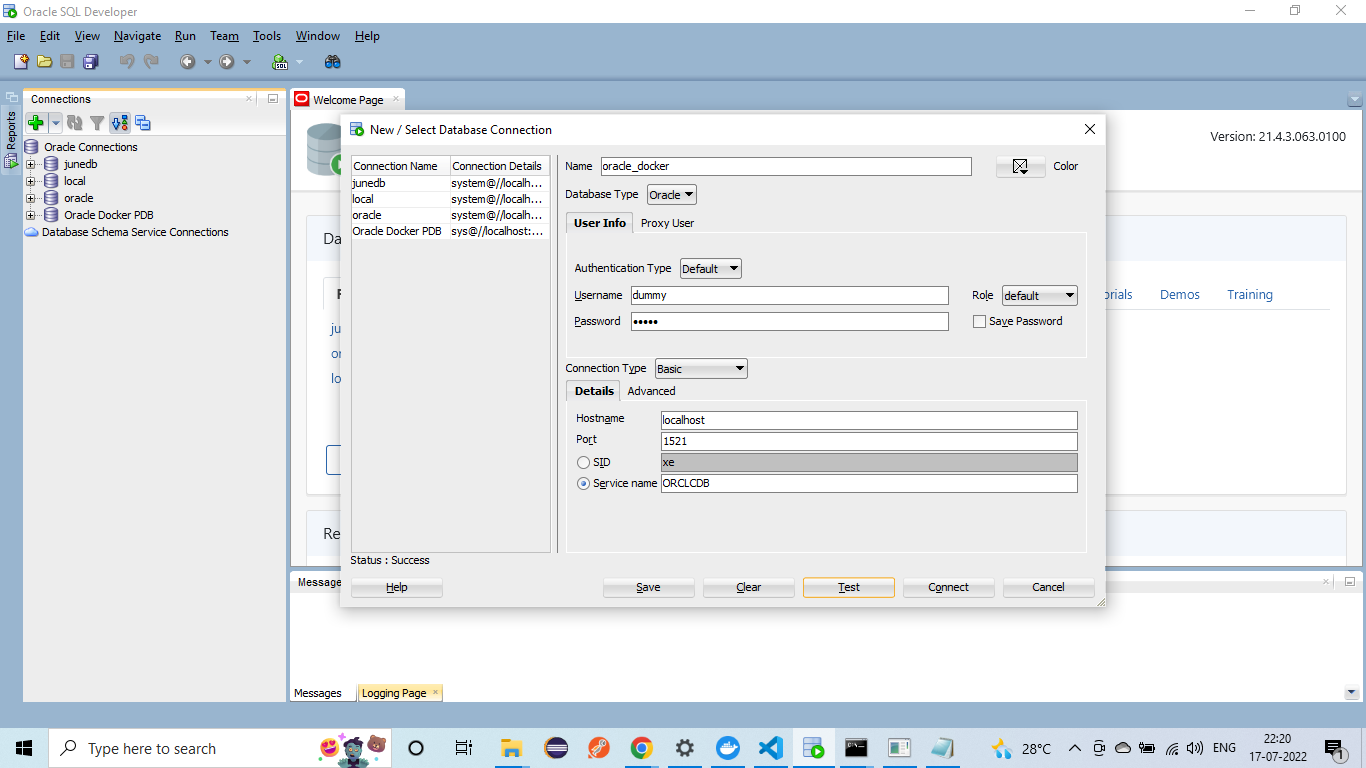
docker run -d -p 1521:1521 --name 085b78f09b56 container-registry.oracle.com/database/enterprise:latest

docker exec -it 7b249dbd6af2 bash -c "source /home/oracle/.bashrc; sqlplus /nolog"

docker logs -f kafkaconnect8888-oracle-db-1

Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production

Version 21.3.0.0.0



1. Log into Docker hub (in order to access oracle repository)

docker login

2. Download image

docker pull store/oracle/database-enterprise:12.2.0.1

3. Run image

docker run -d -p 1521:1521 --name youtubeoracle store/oracle/database-enterprise:12.2.0.1

4. Connect to container

docker exec -it youtubeoracle bash -c "source /home/oracle/.bashrc; sqlplus /nolog"

5. Copy below script to open SQL shell

connect sys as sysdba;

-- Here enter the password as **'Oradoc\_db1'**

alter session set "\_ORACLE\_SCRIPT"=true;

create user dummy identified by dummy;

GRANT ALL PRIVILEGES TO dummy;

4. Configure Aqua Studio or Sql Developer

Username: youtube

Password: youtube

Hostname: localhost

Port: 1521

Service name: ORCLCDB.localdomain

select user from dual;

connect dummy;

# **How to Find Out Your Oracle Database Name**

SQL> select name from V$database;

NAME

---------

XE

# [**Forgot Oracle username and password, how to retrieve?**](https://stackoverflow.com/questions/35199084/forgot-oracle-username-and-password-how-to-retrieve)

select username,password from dba\_users;

# [**Get list of all tables in Oracle?**](https://stackoverflow.com/questions/205736/get-list-of-all-tables-in-oracle)

select \* from student;

select \* from all tables;;

select table\_name from all\_tables ;

SELECT table\_name FROM user\_tables;

create table student(id integer primary key,name varchar2(10));

insert into student values(1,'MAX');

How to select current user from oracle

select user from dual;

docker exec -it 20bb486194f5 bash -c "source /home/oracle/.bashrc; sqlplus /nolog"

docker exec -it 4ada51b11dcc bash -c "source /home/oracle/.bashrc; sqlplus /nolog"

kafka-topics --bootstrap-server localhost:9092 --create --partitions 1 --replication-factor 1 --topic ORCLCDB.dummy.student

docker-compose exec kafka-connect kafka-avro-console-consumer --bootstrap-server kafka:29092 --property schema.registry.url="http://schema-registry:8081" --topic ORCLCDB.C\_\_dummy.student --from-beginning

curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json" http://localhost:8083/connectors/ --data "@SimpleOracleCDCLocal.json"

curl -X POST http://localhost:8083/connectors -H "Content-Type: application/json" -d '{

"name": "Oracle",

"config": {

"connector.class": "io.confluent.connect.jdbc.JdbcSourceConnector",

"connection.url": "jdbc:mysql://mysql:3306/demo",

"connection.user": "connect\_user",

"connection.password": "asgard",

"topic.prefix": "mysql-01-",

"mode":"bulk"

}

}'

curl -X POST http://localhost:8083/connectors -H "Content-Type: application/json" -d '{

"name": "Oracle",

"config": {

"name": "inventory-connector",

"config": {

"connector.class" : "io.debezium.connector.oracle.OracleConnector",

"tasks.max" : "1",

"database.server.name" : "server1",

"database.hostname" : "localhost",

"database.port" : "1521",

"database.user" : "dummy",

"database.password" : "dummy",

"database.dbname" : "ORCLCDB",

"database.pdb.name" : "ORCLPDB1",

"database.out.server.name" : "dbzxout",

"database.history.kafka.bootstrap.servers" : "kafka:9092",

"database.history.kafka.topic": "schema-changes.inventory"

}

}

{

"name": "inventory-connector",

"config": {

"name": "inventory-connector",

"config": {

"connector.class" : "io.debezium.connector.oracle.OracleConnector",

"tasks.max" : "1",

"database.server.name" : "server1",

"database.hostname" : "localhost",

"database.port" : "1521",

"database.user" : "dummy",

"database.password" : "dummy",

"database.dbname" : "ORCLCDB",

"database.pdb.name" : "ORCLPDB1",

"database.out.server.name" : "dbzxout",

"database.history.kafka.bootstrap.servers" : "kafka:9092",

"database.history.kafka.topic": "schema-changes.inventory"

}

}}

Root cause :-

This lines are the most important from your log:

java.util.concurrent.ExecutionException: org.apache.kafka.connect.errors.ConnectException: Failed to find any class that implements Connector and which name matches io.confluent.connect.jdbc.JdbcSinkConnector, available connectors are:...

It seems, that you didn't install kafka-connect-jdbc connector

[Check your plugin.path property in etc/schema-registry/connect-avro-standalone.properties](https://github.com/confluentinc/schema-registry/blob/5.1.0-post/config/connect-avro-standalone.properties#L41-L49) and ensure that the line for plugin.path is uncommented.

If not using Confluent Platform, you will need to create under that plugin.path directory, another directory for the jdbc plugin: ex. kafka-connect-jdbc and put all needed jars there ex. kafka-connect-jdbc-5.1.0.jar, its dependencies, and your jdbc drivers.

More details can be found: <https://docs.confluent.io/current/connect/userguide.html#installing-plugins>

F:\11111\KafkaConnect9999>curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json" http://localhost:8083/connectors/ --data "@SimpleOracleCDCLocal.json"

HTTP/1.1 500 Internal Server Error

Date: Mon, 18 Jul 2022 14:10:11 GMT

Content-Type: application/json

Content-Length: 2975

Server: Jetty(9.4.33.v20201020)

{"error\_code":500,"message":"Failed to find any class that implements Connector and which name matches io.confluent.connect.oracle.cdc.OracleCdcSourceConnector, available connectors are: PluginDesc{klass=class org.apache.kafka.connect.file.FileStreamSinkConnector, name='org.apache.kafka.connect.file.FileStreamSinkConnector', version='6.1.0-ccs', encodedVersion=6.1.0-ccs, type=sink, typeName='sink', location='file:/usr/share/java/kafka/'}, PluginDesc{klass=class org.apache.kafka.connect.file.FileStreamSourceConnector, name='org.apache.kafka.connect.file.FileStreamSourceConnector', version='6.1.0-ccs', encodedVersion=6.1.0-ccs, type=source, typeName='source', location='file:/usr/share/java/kafka/'}, PluginDesc{klass=class org.apache.kafka.connect.mirror.MirrorCheckpointConnector, name='org.apache.kafka.connect.mirror.MirrorCheckpointConnector', version='1', encodedVersion=1, type=source, typeName='source', location='file:/usr/share/java/kafka/'}, PluginDesc{klass=class org.apache.kafka.connect.mirror.MirrorHeartbeatConnector, name='org.apache.kafka.connect.mirror.MirrorHeartbeatConnector', version='1', encodedVersion=1, type=source, typeName='source', location='file:/usr/share/java/kafka/'}, PluginDesc{klass=class org.apache.kafka.connect.mirror.MirrorSourceConnector, name='org.apache.kafka.connect.mirror.MirrorSourceConnector', version='1', encodedVersion=1, type=source, typeName='source', location='file:/usr/share/java/kafka/'}, PluginDesc{klass=class org.apache.kafka.connect.tools.MockConnector, name='org.apache.kafka.connect.tools.MockConnector', version='6.1.0-ccs', encodedVersion=6.1.0-ccs, type=connector, typeName='connector', location='file:/usr/share/java/acl/'}, PluginDesc{klass=class org.apache.kafka.connect.tools.MockSinkConnector, name='org.apache.kafka.connect.tools.MockSinkConnector', version='6.1.0-ccs', encodedVersion=6.1.0-ccs, type=sink, typeName='sink', location='file:/usr/share/java/acl/'}, PluginDesc{klass=class org.apache.kafka.connect.tools.MockSourceConnector, name='org.apache.kafka.connect.tools.MockSourceConnector', version='6.1.0-ccs', encodedVersion=6.1.0-ccs, type=source, typeName='source', location='file:/usr/share/java/acl/'}, PluginDesc{klass=class org.apache.kafka.connect.tools.SchemaSourceConnector, name='org.apache.kafka.connect.tools.SchemaSourceConnector', version='6.1.0-ccs', encodedVersion=6.1.0-ccs, type=source, typeName='source', location='file:/usr/share/java/acl/'}, PluginDesc{klass=class org.apache.kafka.connect.tools.VerifiableSinkConnector, name='org.apache.kafka.connect.tools.VerifiableSinkConnector', version='6.1.0-ccs', encodedVersion=6.1.0-ccs, type=source, typeName='source', location='file:/usr/share/java/acl/'}, PluginDesc{klass=class org.apache.kafka.connect.tools.VerifiableSourceConnector, name='org.apache.kafka.connect.tools.VerifiableSourceConnector', version='6.1.0-ccs', encodedVersion=6.1.0-ccs, type=source, typeName='source', location='file:/usr/share/java/acl/'}"}

F:\11111\KafkaConnect9999>plugin.path=/usr/local/share/java,/usr/local/share/kafka/plugins,/opt/connector

'plugin.path' is not recognized as an internal or external command,

CONNECT\_PLUGIN\_PATH: '/usr/share/java,/etc/kafka-connect/jars'

kafka-connect\_fronting: image: confluentinc/cp-kafka-connect container\_name: kafka-connect\_fronting hostname: connect\_fronting depends\_on: - zookeeper\_fronting - kafka\_fronting - schema-registry\_fronting ports: - "8083:8083" volumes: - ./jars:/etc/kafka-connect/jars/ environment: CONNECT\_BOOTSTRAP\_SERVERS: 'kafka\_fronting:29092' CONNECT\_REST\_ADVERTISED\_HOST\_NAME: connect CONNECT\_REST\_PORT: 8083 CONNECT\_GROUP\_ID: compose-connect-group CONNECT\_CONFIG\_STORAGE\_TOPIC: docker-connect-configs CONNECT\_CONFIG\_STORAGE\_REPLICATION\_FACTOR: 1 CONNECT\_OFFSET\_FLUSH\_INTERVAL\_MS: 10000 CONNECT\_OFFSET\_STORAGE\_TOPIC: docker-connect-offsets CONNECT\_OFFSET\_STORAGE\_REPLICATION\_FACTOR: 1 CONNECT\_STATUS\_STORAGE\_TOPIC: docker-connect-status CONNECT\_STATUS\_STORAGE\_REPLICATION\_FACTOR: 1 CONNECT\_KEY\_CONVERTER: io.confluent.connect.avro.AvroConverter CONNECT\_KEY\_CONVERTER\_SCHEMA\_REGISTRY\_URL: 'http://schema-registry\_fronting:8081' CONNECT\_VALUE\_CONVERTER: io.confluent.connect.avro.AvroConverter CONNECT\_VALUE\_CONVERTER\_SCHEMA\_REGISTRY\_URL: 'http://schema-registry\_fronting:8081' CONNECT\_INTERNAL\_KEY\_CONVERTER: org.apache.kafka.connect.json.JsonConverter CONNECT\_INTERNAL\_VALUE\_CONVERTER: org.apache.kafka.connect.json.JsonConverter CONNECT\_ZOOKEEPER\_CONNECT: 'zookeeper\_fronting:32181' CONNECT\_PLUGIN\_PATH: '/etc/kafka-connect/jars'