UI repo - <https://gitlab.com/preciselydata/data-integration/connect-portal-ui>

Docker initial setup repo - <https://gitlab.com/preciselydata/data-integration/devops/connect-infra/-/tree/master/>

https://sso.precisely.services/

[connect\_user\_1@mailinator.com](mailto:connect_user_1@mailinator.com) / Password@1234

if build getting failed, follow below steps,

1. Cmd: npm run check:saas
2. Cmd: npm run test-watch
3. Cmd: npm run lint

designer | desinger

neo | neo

operator | operator

**New Data Connection:**

Instance name : ORCLPDB1

Pluggable database: true

C##CDBREP/cdbrep

SYS/Oradoc\_db1

dbtester/dbsystest

host.docker.internal

If your code not working in browser, then use do docker setup again and authenticate again here

http://host.docker.internal:8080/auth

and

command to clear all volumes folder CMd: docker system prune -f --volumes –all

command to run connect-infra cmd: docker-compose -p connect-infra up -d

command to down connect-infra volums cmd: docker-compose down -v

command to up connect-infra volums cmd: docker-compose up -d

command to load docker connect-infra : docker-compose -p connect-infra up -d

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

# offset explorer for zookeeper and kafka setup

cmd path : C:\apps\kafka\_2.13-3.1.0\bin\windows>

zookeeper cmd : zookeeper-server-start.bat ..\..\config\zookeeper.properties

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





data page password: SysAdmin1

# Kafka setup

File name - C:\Precisely work space\Database create images

Open cmd prompt -> num cmd: docker compose up -d

I wil create database in docker

DB2z -> data connection adding -> C:\PreciselyConnect\ConnectCDCV58\kernel -> wrapper file -> add two changes -> C:/cust-dir/Db2Drivers/Db2Drivers/\*

# KafKA Server details { "name": "hadoop11", "hostname": "hadoop11-master.winkerb2.org", "port": 6667, "zookeeper": "hadoop11-master.winkerb2.org:2181" },

Metabase : SAASREP1/ SAASREP1

# DB2z

"db2z\_accessMethod": "JDBC",

"db2z\_connection\_description": "Db2z Connection",

"db2zConnectionName": "Db2zConn",

"db2z\_database": "zos1.syncsort.com",

"db2z\_dbusername": "WWCDMC",

"db2z\_dbpassword": "sdmxats1",

"parameters.DB2Z\_DATABASE\_NAME": "ZOS2DBCG",

"parameters.DB2Z\_PORT\_NUMBER":"5031",

"parameters.DB2Z\_CAPTURE\_SERVER":"zos1.syncsort.com",

"parameters.DB2Z\_CAPTURE\_PORT":"2637",

"parameters.DB2Z\_CAPTURE\_PRIVATE\_KEY":"C:\\\\PreciselyConnect\\\\ConnectCDCV58\\\\kernel\\\\id\_nacl",

# DataCconnection(IBMi)

{  
"name": "LTIAS08",  
"type": "IBM i",  
"accessMethod": "JDBC",  
"hostname": "10.22.27.34",  
"libraryName": "SHARE55GA",  
"user": "qarep1",  
"password": "rpuser1",  
"replicationUser": "qarep1",  
"replicationPassword": "rpuser1",  
"adminUser": "qarep1",  
"adminPassword": "rpuser1"  
}

**Or**

LTIAS08  
LTIAS08.visionsolutions.com  
10.22.27.34  
Installation lib: SHARE5805 / SHARE55GA  
qarep1/rpuser1

Or

LTIAS08.visionsolutions.com

SHARE55GA

test user: SAASUSR1 / SAASUSR1

replication user: SAASREP1 / SAASREP1

# Oracle Testers Details

Pluggable (checkbox)-

Root user -

C##CDBREP | cdbrep

test user -

sys | Oradoc\_db1

version - 12c & above

ip - 172.30.6.172

port 1521

instance name - orclpdb1

Metabase creation -

admin credentials -

sys | Oradoc\_db1

# Meta base

Here is the info for an Oracle server that's up and running, please make sure that when creating a metabase that you change the size from 200MB to either 10 or 20MB so it does not run out of space.

{  
"name": "sitarc",  
"accessMethod": "JDBC",  
"hostname": "10.2.66.210",  
"port": 1521,  
"instanceName": "orcl",  
"version": 11,  
"user": "rpuserm",  
"password": "rpuserm",  
"adminUser": "SYS",  
"adminPassword": "SysAdm",  
"pluggable": false  
},

Graphical user interface, text, application

Description automatically generated

# Application KT setup

Graphical user interface, application

Description automatically generated

Graphical user interface, application, website

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Select -> create new topic - >

Remove nodemodules easiy **cmd**: RMDIR /Q/S node\_modules

Graphical user interface, text, application, email

Description automatically generated

# **Setup**

1. Get all git access below links  
   <https://gitlab.com/preciselydata/data-integration>  
   connect-portal-ui  
   connect-portal  
   registry.gitlab.com/preciselydata/data-integration/devops/devops-tools/keycloak-internal  
   registry.gitlab.com/preciselydata/cloud/di-poc/grpc-prototype  
   registry.gitlab.com/preciselydata/data-integration/connect-portal  
   registry.gitlab.com/preciselydata/data-integration/connect-portal-uiregistry.gitlab.com/preciselydata/data-integration/connect-hub/agent  
   registry.gitlab.com/preciselydata/data-integration/connect-hub/agent  
   jfrog.precisely.engineering/docker-virtual/mongo
2. First need gitlab – create like this <https://jira.syncsort.com/browse/DEVOPS-976>
3. Create JIRA request in - <https://jira.syncsort.com/plugins/servlet/theme/portal/1> or check with Rajeev for gitlab request for permission into gitlab
4. Once permission approved raise RAM upgradation into <https://it.precisely.com/>
5. Install docker from open source, don’t install from company portal. And
6. Created file .wslconfig.txt not sure on this – this file helps not use total RAM from VDI. And open that file and paste below text   
   **[wsl2]  
   memory=8GB # Limits VM memory in WSL 2 up to 8GB  
   processors=4 # Makes the WSL 2 VM use two virtual processors  
   swap=1GB # For Swapping**
7. Setup SSh key also from local machine to gitlab account
8. Now clone - <https://gitlab.com/preciselydata/data-integration/devops/connect-infra/-/tree/masterhttps://gitlab.com/preciselydata/data-integration/devops/connect-infra/-/tree/master>
9. Follow read me file on that clone, run all commands except last command - docker-compose down -v
10. Now open docker, it will creates some servers - > click on Graphical user interface, application

    Description automatically generated
11. If any errors on connect-hub-agent – take that particular url and run in browers for authentication with ID: neo and password; neo  
    EG:
12. Once server is running in docker, open VS code and create file any name loggin.log file like this then open terminal in VS code and run below command   
    cmd: docker logs connect-hub-agent > grepTest.log
13. Then finally open <http://localhost:4200/> - this is now not required -

# Register docker again steps

1. Open docker and delete all ‘connect-insfra’-> click on delte – left hand side
2. And dlete all things in volume and images also
3. Then take latest in connect-infra and do modification in In file -> dmxmgr.properties -> change line#10 to false

And again run command : docker-compose -p connect-infra up -d

1. Run above command

# Run local host with 4200:

* Go to <http://localhost:8080/auth/> -> admin/admin -> clients(left side menu) -> [OIDC-Connect](http://localhost:8080/auth/admin/master/console/#/realms/Precisely/clients/28502fbe-2124-4992-8fc6-82ff8f91bc59) -> add new ‘http://localhost:4200/\*’ -> save.

# Latest CDC listner changes

First download link for emini folder fun -> run.exe

-remove first and then install

-> install lo only two changes, 4 check box lo 3 select cheskovali

-> install lo 6-7 dropdowns vasthadi, dantlo oracle downdown tharuvath ikoti undali dani varku eddaina lasts number select cheskovali

-> last install ki velipovali malla middle lo omini licence aduguthundi, adi path ivali

CDC listner stop chesi below stop follow avail.

Malla C:\Windows\System32\config\systemprofile\.cdc - > e folder lo all files delete cheyali.