Connect to Mongodb server:

6/11 **docker exec -it 7fa5e286b664 bash**

**Installed nodejs and mongoose in** 7fa5e286b664 **docker container:**

1. Install java in this container:
2. [root@7fa5e286b664 /]# yum install -y java

Java -version is "1.8.0\_171"

1. [root@7fa5e286b664 /]# yum install -y gcc-c++ make

Loaded plugins: ovl, ulninfo

Package gcc-c++-4.8.5-28.0.1.el7\_5.1.x86\_64 already installed and latest version

Package 1:make-3.82-23.el7.x86\_64 already installed and latest version

Nothing to do

1. [root@7fa5e286b664 /]# curl -sL https://rpm.nodesource.com/setup\_10.x | sudo -E bash -

bash: line 29: which: command not found

## Installing the NodeSource Node.js 10.x repo...

## Inspecting system...

+ rpm -q --whatprovides redhat-release || rpm -q --whatprovides centos-release || rpm -q --whatprovides cloudlinux-release || rpm -q --whatprovides sl-release

+ uname -m

## Confirming "el7-x86\_64" is supported...

+ curl -sLf -o /dev/null 'https://rpm.nodesource.com/pub\_10.x/el/7/x86\_64/nodesource-release-el7-1.noarch.rpm'

## Downloading release setup RPM...

+ mktemp

+ curl -sL -o '/tmp/tmp.hlc68qDkU8' 'https://rpm.nodesource.com/pub\_10.x/el/7/x86\_64/nodesource-release-el7-1.noarch.rpm'

## Installing release setup RPM...

+ rpm -i --nosignature --force '/tmp/tmp.hlc68qDkU8'

## Cleaning up...

+ rm -f '/tmp/tmp.hlc68qDkU8'

## Checking for existing installations...

+ rpm -qa 'node|npm' | grep -v nodesource

## Run `sudo yum install -y nodejs` to install Node.js 10.x and npm.

## You may also need development tools to build native addons:

sudo yum install gcc-c++ make

## To install the Yarn package manager, run:

curl -sL https://dl.yarnpkg.com/rpm/yarn.repo | sudo tee /etc/yum.repos.d/yarn.repo

sudo yum install yarn

1. [root@7fa5e286b664 /]# yum install nodejs

Loaded plugins: ovl, ulninfo

nodesource | 2.5 kB 00:00:00

nodesource/x86\_64/primary\_db | 11 kB 00:00:01

Resolving Dependencies

--> Running transaction check

---> Package nodejs.x86\_64 2:10.4.0-1nodesource will be installed

--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================================================

Package Arch Version Repository Size

=====================================================================================================================

Installing:

nodejs x86\_64 2:10.4.0-1nodesource nodesource 17 M

Transaction Summary

=====================================================================================================================

Install 1 Package

Total download size: 17 M

Installed size: 51 M

Is this ok [y/d/N]: y

Downloading packages:

warning: /var/cache/yum/x86\_64/7Server/nodesource/packages/nodejs-10.4.0-1nodesource.x86\_64.rpm: Header V4 RSA/SHA256 Signature, key ID 34fa74dd: NOKEYMB 00:00:00 ETA

Public key for nodejs-10.4.0-1nodesource.x86\_64.rpm is not installed

nodejs-10.4.0-1nodesource.x86\_64.rpm | 17 MB 00:00:07

Retrieving key from file:///etc/pki/rpm-gpg/NODESOURCE-GPG-SIGNING-KEY-EL

Importing GPG key 0x34FA74DD:

Userid : "NodeSource <gpg-rpm@nodesource.com>"

Fingerprint: 2e55 207a 95d9 944b 0cc9 3261 5ddb e8d4 34fa 74dd

Package : nodesource-release-el7-1.noarch (installed)

From : /etc/pki/rpm-gpg/NODESOURCE-GPG-SIGNING-KEY-EL

Is this ok [y/N]: y

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Warning: RPMDB altered outside of yum.

Installing : 2:nodejs-10.4.0-1nodesource.x86\_64 1/1

Verifying : 2:nodejs-10.4.0-1nodesource.x86\_64 1/1

Installed:

nodejs.x86\_64 2:10.4.0-1nodesource

Complete!

1. **Check node and npm version**

[root@7fa5e286b664 software]# **node -v**

v10.4.0

[root@7fa5e286b664 software]# **npm -v**

6.1.0

1. **Create folder:**

[root@7fa5e286b664 /]# mkdir mongoose\_basics

1. **Initialize npm**

Go to mongoose\_basics

[root@7fa5e286b664 mongoose\_basics]# **npm –init**

1. **Install mongoose**

[root@7fa5e286b664 mongoose\_basics]# **npm install mongoose**

npm WARN saveError ENOENT: no such file or directory, open '/mongoose\_basics/package.json'

npm WARN enoent ENOENT: no such file or directory, open '/mongoose\_basics/package.json'

npm WARN mongoose\_basics No description

npm WARN mongoose\_basics No repository field.

npm WARN mongoose\_basics No README data

npm WARN mongoose\_basics No license field.

+ mongoose@5.1.4

updated 1 package and audited 59 packages in 2.629s

found 0 vulnerabilities

1. **set up the connection to Mongodb:**
2. **start Mongod first**:
3. **go to mongoose\_basics folder and vim file:**

[root@7fa5e286b664 mongoose\_basics]# vim connection.js

*var mongoose = require ('mongoose');*

*mongoose.connect('mongodb://localhost/mydb');*

*var connection = mongoose.connection;*

*connection.on('connected',function(){*

*console.log('connected to db');*

*});*

*connection.on('disconnected',function(){*

*console.log('disconnected to db');*

*});*

*connection.on('error',function(error){*

*console.log('db connection error', error);*

*});*

*process.on('SIGINT', function() {*

*connection.close(function(){*

*console.log('db connection closed due to process termination');*

*process.exit(0);*

*});*

*});*

*module.exports = connection;*

1. from another terminal to run

[root@7fa5e286b664 mongoose\_basics]#node connection.js

Can see “connected to db” opening….

If ^Cdisconnected to db, can see:

db connection closed due to process termination

1. **Create Demo Web Server (Optional)**

**This is an optional step. If you want to test your node.js install. Let’s create a web server with “Welcome Node.js” text. Create a file demo\_server.js**

[root@7fa5e286b664 mongoose\_basics]# **vim demo\_server.js**

***var http = require('http');***

***http.createServer(function (req, res) {***

***res.writeHead(200, {'Content-Type': 'text/plain'});***

***res.end('Welcome Node.js');***

***}).listen(3001, "127.0.0.1");***

***console.log('Server running at*** [***http://127.0.0.1:3001/***](http://127.0.0.1:3001/)***');***

1. **start the web server using the command.**

**[root@7fa5e286b664 mongoose\_basics]# node --inspect demo\_server.js**

**Debugger listening on ws://127.0.0.1:9229/36a5521b-29c1-4d77-bc2d-5116cf965938**

**For help, see: https://nodejs.org/en/docs/inspector**

**Server running at** [**http://127.0.0.1:3001/**](http://127.0.0.1:3001/)

[root@7fa5e286b664 mongoose\_basics]# npm install mongoose --save

npm WARN saveError ENOENT: no such file or directory, open '/mongoose\_basics/package.json'

npm notice created a lockfile as package-lock.json. You should commit this file.

npm WARN enoent ENOENT: no such file or directory, open '/mongoose\_basics/package.json'

npm WARN mongoose\_basics No description

npm WARN mongoose\_basics No repository field.

npm WARN mongoose\_basics No README data

npm WARN mongoose\_basics No license field.

+ mongoose@5.1.4

added 20 packages from 15 contributors and audited 23 packages in 12.864s

found 0 vulnerabilities

6/6

MongoDB provides the mongoimport utility that can be used to import JSON, CSV, or TSV files into a MongoDB database.

Connect to Mongodb server:

**docker exec -it 7fa5e286b664 bash**

Install Mongodb to docker containe in Linux:

First way: use a current available docker container and transfer mongodb installation file there:

1. Download “ mongodb-linux-x86\_64-rhel70-3.6.5.tgz” from …to local machine, and then pscp to docker server:

C:\software pscp mongodb-linux-x86\_64-rhel70-3.6.5.tgz [gjiang@10.103.157.117://home/gjiang/software](mailto:gjiang@10.103.157.117://home/gjiang/software)

1. Cp the download file to **7fa5e286b664** docker container: root account

[gjiang@ip-10-103-157-48 ~]$ docker cp mongodb-linux-x86\_64-rhel70-3.6.5.tgz **7fa5e286b664**:/software/

confluent-oss-4.1.1-2.11.zip

1. access to the container:

[gjiang@ip-10-103-157-48 ~]$ **docker exec -it 7fa5e286b664 bash**

**4. access to /software folder and extract the files:**

[root@7fa5e286b664 software]# tar zxvf mongodb-linux-x86\_64-rhel70-3.6.5.tgz to got a new folder: **“mongodb-linux-x86\_64-rhel70-3.6.5”, in this folder can see**

[root@7fa5e286b664 software]# tar zxvf mongodb-linux-x86\_64-rhel70-3.6.5.tgz

mongodb-linux-x86\_64-rhel70-3.6.5/README

mongodb-linux-x86\_64-rhel70-3.6.5/THIRD-PARTY-NOTICES

mongodb-linux-x86\_64-rhel70-3.6.5/MPL-2

mongodb-linux-x86\_64-rhel70-3.6.5/GNU-AGPL-3.0

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongodump

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongorestore

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongoexport

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongoimport

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongostat

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongotop

mongodb-linux-x86\_64-rhel70-3.6.5/bin/bsondump

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongofiles

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongoreplay

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongoperf

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongod

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongos

mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongo

mongodb-linux-x86\_64-rhel70-3.6.5/bin/install\_compass

5. create a new folder for mongodb:

[root@7fa5e286b664 db]# mkdir -p mongodb and data, db and log folders under mongodb folder

6. copy or cut the extracted folders to the new directory: mongodb:

[root@7fa5e286b664 db]# mv -R -n mongodb-linux-x86\_64-rhel70-3.6.5 /mongodb

7**. go to /mongodb/bin and start mongodb:**

[root@7fa5e286b664 bin]# ./mongod -dbpath /mongodb/data/db

*2018-06-07T15:34:48.447+0000 I CONTROL [initandlisten] MongoDB starting : pid=313 port=27017 dbpath=/mongodb/data/db 64-bit host=7fa5e286b664*

*2018-06-07T15:34:48.447+0000 I CONTROL [initandlisten] db version v3.6.5*

*2018-06-07T15:34:48.448+0000 I CONTROL [initandlisten] git version: a20ecd3e3a174162052ff99913bc2ca9a839d618*

*2018-06-07T15:34:48.448+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1e-fips 11 Feb 2013*

*2018-06-07T15:34:48.448+0000 I CONTROL [initandlisten] allocator: tcmalloc*

*2018-06-07T15:34:48.448+0000 I CONTROL [initandlisten] modules: none*

*2018-06-07T15:34:48.448+0000 I CONTROL [initandlisten] build environment:*

*2018-06-07T15:34:48.448+0000 I CONTROL [initandlisten] distmod: rhel70*

*2018-06-07T15:34:48.448+0000 I CONTROL [initandlisten] distarch: x86\_64*

*2018-06-07T15:34:48.448+0000 I CONTROL [initandlisten] target\_arch: x86\_64*

*2018-06-07T15:34:48.448+0000 I CONTROL [initandlisten] options: { storage: { dbPath: "/mongodb/data/db" } }*

*2018-06-07T15:34:48.457+0000 I STORAGE [initandlisten] wiredtiger\_open config: create,cache\_size=7052M,session\_max=20000,eviction=(threads\_min=4,threads\_max=4),config\_base=false,statistics=(fast),cache\_cursors=false,log=(enabled=true,archive=true,path=journal,compressor=snappy),file\_manager=(close\_idle\_time=100000),statistics\_log=(wait=0),verbose=(recovery\_progress),*

*2018-06-07T15:34:49.280+0000 I STORAGE [initandlisten] WiredTiger message [1528385689:280931][313:0x7f49f5757b00], txn-recover: Set global recovery timestamp: 0*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten]*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* WARNING: Access control is not enabled for the database.*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* Read and write access to data and configuration is unrestricted.*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* WARNING: You are running this process as the root user, which is not recommended.*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten]*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* WARNING: This server is bound to localhost.*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* Remote systems will be unable to connect to this server.*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* Start the server with --bind\_ip <address> to specify which IP*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* addresses it should serve responses from, or with --bind\_ip\_all to*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* bind to all interfaces. If this behavior is desired, start the*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* server with --bind\_ip 127.0.0.1 to disable this warning.*

*2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten]*

*2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten]*

*2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten] \*\* WARNING: /sys/kernel/mm/transparent\_hugepage/enabled is 'always'.*

*2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten] \*\* We suggest setting it to 'never'*

*2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten]*

*2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten] \*\* WARNING: /sys/kernel/mm/transparent\_hugepage/defrag is 'always'.*

*2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten] \*\* We suggest setting it to 'never'*

*2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten]*

*2018-06-07T15:34:49.322+0000 I STORAGE [initandlisten] createCollection: admin.system.version with provided UUID: ad47efed-8f34-44e3-9913-f330423ba9e6*

*2018-06-07T15:34:49.351+0000 I COMMAND [initandlisten] setting featureCompatibilityVersion to 3.6*

*2018-06-07T15:34:49.356+0000 I STORAGE [initandlisten] createCollection: local.startup\_log with generated UUID: 9fe1c34a-82a8-47f1-ba32-8184dcb37a54*

*2018-06-07T15:34:49.373+0000 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory '/mongodb/data/db/diagnostic.data'*

*2018-06-07T15:34:49.373+0000 I NETWORK [initandlisten] waiting for connections on port 27017*

From above info, can see the mongodb started successfully

8. Open another terminal to start Mongodb command line or shell:

[root@7fa5e286b664 bin]# ./mongo

MongoDB shell **version v3.6.5**

connecting to: mongodb**://127.0.0.1:27017**

MongoDB server version: 3.6.5

Welcome to the MongoDB shell.

For interactive help, type "help".

For more comprehensive documentation, see

http://docs.mongodb.org/

Questions? Try the support group

http://groups.google.com/group/mongodb-user

Server has startup warnings:

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten]

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* WARNING: Access control is not enabled for the database.

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* Read and write access to data and configuration is unrestricted.

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* WARNING: You are running this process as the root user, which is not recommended.

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten]

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* WARNING: This server is bound to localhost.

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* Remote systems will be unable to connect to this server.

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* Start the server with --bind\_ip <address> to specify which IP

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* addresses it should serve responses from, or with --bind\_ip\_all to

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* bind to all interfaces. If this behavior is desired, start the

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten] \*\* server with --bind\_ip 127.0.0.1 to disable this warning.

2018-06-07T15:34:49.321+0000 I CONTROL [initandlisten]

2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten]

2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten] \*\* WARNING: /sys/kernel/mm/transparent\_hugepage/enabled is 'always'.

2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten] \*\* We suggest setting it to 'never'

2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten]

2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten] \*\* WARNING: /sys/kernel/mm/transparent\_hugepage/defrag is 'always'.

2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten] \*\* We suggest setting it to 'never'

2018-06-07T15:34:49.322+0000 I CONTROL [initandlisten]

**> show dbs**

admin 0.000GB

config 0.000GB

local 0.000GB

9. create a database:

> **use mydb**

switched to db mydb

> **db**

mydb

10. create a user:

> db.createUser ({user: "gjiang", pwd: "Passw0rd", roles:["readWrite","dbAdmin"]});

Successfully added user: { "user" : "gjiang", "roles" : [ "readWrite", "dbAdmin" ] }

11. create a collection:

> db.createCollection('customers');

{ "ok" : 1 }

> show collections

Customers

Second way, create a new docker with Mongodb image: but doesn’t work:

See reference: <https://tech.oeru.org/installing-mongodb-docker-ubuntu-linux-1404>

Installing Docker (since I have had Docker platform, I just open my default Linux Ducker container server: /home/gjiang

1. Installing Mongodb , Directly run pull mongo command, and it took couple minutes

[gjiang@ip-10-103-157-48 ~]$ **docker pull mongo**

*Using default tag: latest*

*Trying to pull repository registry.access.redhat.com/mongo ...*

*Trying to pull repository docker.io/library/mongo ...*

*latest: Pulling from docker.io/library/mongo*

*4d0d76e05f3c: Pull complete*

*d3e8f2a80d9c: Pull complete*

*fa1dfbce34f0: Pull complete*

*9c46dfce21d0: Pull complete*

*5108b7b81738: Pull complete*

*f96cba0bb653: Pull complete*

*11707d2d22ac: Pull complete*

*8e8c0aaadd20: Pull complete*

*23c7bba6fd7d: Pull complete*

*3f745addabe8: Pull complete*

*Digest: sha256:3a09cd85fb4e76f1d5832f9ea1d4e7481f76e807389b7d8ea6ac4d4ba96f83e5*

1. **Run and start mongodb:**

[gjiang@ip-10-103-157-48 ~]$ docker run --name mongodb --restart unless-stopped -v /home/gjiang/mongo/data/db:/data/db -v /home/gjiang/mongo/backups:/backups -d mongo --smallfiles

d8e94c4679eb6e7c4ee936097c97926a2f3133d134d136026e5e66df60761659

log into the mongo-db-container container shell:

docker exec -it mongo-db-container /bin/bash

**check mongodb docker is running:**

[gjiang@ip-10-103-157-48 ~]$ docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

d8e94c4679eb mongo "docker-entrypoint.sh" 22 seconds ago Restarting (1) 3 seconds ago 27017/tcp mongodb

----------------------------------

Run JSON data **import** in Linux:

copied a JSON file from website: [https://github.com/ozlerhakan/mongodb-json-files/blob/master/datasets/students.json](https://webmail.psi-it.com/owa/redir.aspx?C=oxD_FF4iszq1hbPUPDGWNm6XhI-MZJ7nVJ5lrkYUjswk9RYlVs3VCA..&URL=https%3a%2f%2fgithub.com%2fozlerhakan%2fmongodb-json-files%2fblob%2fmaster%2fdatasets%2fstudents.json)

to 7fa5e286b664 container and in /mongodb/testing folder

[root@7fa5e286b664 testing]# /mongodb/mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongoimport --db mydb --collection students --file students\_import.json

Very quickly to get the 200 data import

2018-06-08T15:36:42.559+0000 connected to: localhost

2018-06-08T15:36:42.587+0000 imported 200 documents

Run CSV data **import** in Linux:

Copy excel csv data to Linx vi file, not delimited

testing]# /mongodb/mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongoimport --db mydb --collection people --type csv --file people\_csv\_import.csv –headerline

2018-06-08T19:22:34.269+0000 connected to: localhost

2018-06-08T19:22:34.307+0000 imported 500 documents

[root@7fa5e286b664 testing]# /mongodb/mongodb-linux-x86\_64-rhel70-3.6.5/bin/mongoimport --db mydb --collection month --type csv --file month\_csv\_import.csv --headerline

2018-06-08T19:50:33.588+0000 connected to: localhost

2018-06-08T19:50:33.607+0000 imported 60 documents

6/11/18

Install Node.js and Mongoose into Linux docker contaioner:

Got error, cannot get npm command!

Website:

<https://code.tutsplus.com/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527>

download Latest LTS Version: **8.11.2** (includes npm 5.6.0) **Linux Binaries (x86/x64)**

from CMD:

C:\software>pscp node-v8.11.2-linux-x64.tar.xz [gjiang@10.103.157.117://home/gjiang/software](mailto:gjiang@10.103.157.117://home/gjiang/software)

Move file from the server to docker container:

[gjiang@ip-10-103-157-48 software]$ docker cp node-v8.11.2-linux-x64.tar.xz 7fa5e286b664:/software/

Extract tar.xz file:

tar -xf node-v8.11.2-linux-x64.tar.xz

6/20/18

Download and install Confluent JDBC Connector to the Docker container (that has had MongoDB installed):

Follow the instruction:

[**https://docs.confluent.io/current/quickstart/cos-quickstart.html**](https://docs.confluent.io/current/quickstart/cos-quickstart.html)

1. Download “confluent-oss-4.1.1-2.11.zip” from <https://www.confluent.io/confirmation/>

to local machine, and then pscp to docker server:

C:\software pscp confluent-oss-4.1.1-2.11.zip [gjiang@10.103.157.117://home/gjiang/software](mailto:gjiang@10.103.157.117://home/gjiang/software)

1. Cp the download file to **7fa5e286b664** docker container: root account

[gjiang@ip-10-103-157-48 ~]$ docker cp confluent-oss-4.1.1-2.11.zip **7fa5e286b664**:/software/

1. access to the container:

[gjiang@ip-10-103-157-48 ~]$ **docker exec -it 7fa5e286b664 bash**

1. access /software folder and extract the files:

[root@7fa5e286b664 software]# unzip confluent-oss-4.1.1-2.11.zip

from bin folder, can see [root@7fa5e286b664 confluent-4.1.1]# ls

README bin etc lib share src , **Kafka installation are included already**, so don’t need to install kafka again

Java 1.7 and 1.8 are supported in this version of Confluent Platform. This docker container has Java 1.8

1. set up the path in the profile

export PATH=/mongodb/mongodb-linux-x86\_64-rhel70-3.6.5/bin:/confluent-4.1.1/bin:$PATH

export MONGODB=/mongodb/mongodb-linux-x86\_64-rhel70-3.6.5/bin

1. need to install curl and which before start confluent

yum install curl yum install which

1. run confluent successfully:

[root@7fa5e286b664 bin]# /confluent-4.1.1/bin/confluent start

Using CONFLUENT\_CURRENT: /tmp/confluent.NFBX0ijJ

Starting zookeeper

zookeeper is [UP]

Starting kafka

kafka is [UP]

Starting schema-registry

schema-registry is [UP]

Starting kafka-rest

kafka-rest is [UP]

Starting connect

connect is [UP]

Starting ksql-server

ksql-server is [UP]

for stop:

[root@7fa5e286b664 bin]# /confluent-4.1.1/bin/confluent stop

[root@7fa5e286b664 bin]# /confluent-4.1.1/bin/confluent status

Check each status:

[root@7fa5e286b664 bin]# confluent status zookeeper

zookeeper is [DOWN]

when confluent starts, zookeeper, kafka, schema-registry, kafka-rest, ksql and connect are started same time!

1. **Start topic**:

[root@7fa5e286b664 bin]# /confluent-4.1.1/bin/kafka-**topics** --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic users

Created topic "users".

[root@7fa5e286b664 bin]# /confluent-4.1.1/bin/kafka-**topics** --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic users2

Created topic "users2".

1. **Start KSQL**

[root@7fa5e286b664 bin]# ksql

===========================================

= \_ \_\_ \_\_\_\_\_ \_\_\_\_ \_ =

= | |/ // \_\_\_\_|/ \_\_ \| | =

= | ' /| (\_\_\_ | | | | | =

= | < \\_\_\_ \| | | | | =

= | . \ \_\_\_\_) | |\_\_| | |\_\_\_\_ =

= |\_|\\_\\_\_\_\_\_/ \\_\_\_\\_\\_\_\_\_\_\_| =

= =

= Streaming SQL Engine for Apache Kafka? =

===========================================

ksql> help

ksql> server

http://localhost:8088

ksql> version

Version: 4.1.1

1. Set up **JDBC Source Connector**

<https://docs.confluent.io/current/connect/connect-jdbc/docs/source_connector.html?_ga=2.127427593.568445286.1529330498-2038198777.1529330498>

1. Check which connector in confluent:

[root@7fa5e286b664 bin]# confluent list connectors

Bundled Predefined Connectors (edit configuration under etc/):

elasticsearch-sink

file-source

file-sink

jdbc-source

jdbc-sink

hdfs-sink

s3-sink

1. Run command from /confluent-4.1.1/bin to load JDBC

[root@7fa5e286b664 bin]# **confluent load jdbc-source**

{"name":"jdbc-source","config":{"connector.class":"io.confluent.connect.jdbc.JdbcSourceConnector","tasks.max":"1","connection.url":"jdbc:sqlite:test.db","mode":"incrementing","incrementing.column.name":"id","topic.prefix":"test-sqlite-jdbc-","name":"jdbc-source"},"tasks":[],"type":null}

Can see an empty test.db is generated, but it is for sqlite

]# /confluent-4.1.1/bin/kafka-**topics** --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic users

1. **For Oracle JDBC connection, to use the Kafka JDBC connector, need to make available the relevant JDBC driver for source database, such as oracle JDBC driver**

<https://www.confluent.io/blog/simplest-useful-kafka-connect-data-pipeline-world-thereabouts-part-1/>

<http://www.oracle.com/technetwork/database/application-development/jdbc/downloads/index.html>

1. Download oracle jdbc driver, ojdbc8-full.tar:

<http://www.oracle.com/technetwork/database/features/jdbc/jdbc-ucp-122-3110062.html>

1. Pscp > docer server > container > /confluent-4.1.1/share/java/kafka-connect-jdbc/

tar zxvf ojdbc8-full.tar.gz and the cope ojdbc8.tar to **/confluent-4.1.1/share/java/kafka-connect-jdbc/**

1. **Create a properties file called “source-quickstart-oracle.properties” for the source connector:**

**Go to /confluent-4.1.1/etc/kafka-connect-jdbc**

**[root@7fa5e286b664 kafka-connect-jdbc]# vi source-quickstart-oracle.properties**

*name=test-oracle-jdbc-autoincrement*

*connector.class=io.confluent.connect.jdbc.JdbcSourceConnector*

*tasks.max=1*

*connection.password = examplePassword*

*connection.url = jdbc:oracle:thin:@example.oracle.server.com:1521/ExampleServiceName*

*connection.user = exampleUser*

*table.whitelist=USERS*

*mode=timestamp+incrementing*

*incrementing.column.name=ID*

*timestamp.column.name=MODIFIED*

*topic.prefix=test-oracle-jdbc-*

**can user ps -ef | grep xxxxx to check zookeeper, kafka and registration status**

1. When run above , zookeeper, schema\_registry and kafka server started already, on a new terminal **run the Kafka Connector:**

[root@7fa5e286b664 confluent-4.1.1]# ./bin/connect-standalone ./etc/schema-registry/connect-avro-standalone.properties ./etc/kafka-connect-jdbc/source-quickstart-oracle.properties

1. **Open another terminal to run a Consumer.**

[root@7fa5e286b664 confluent-4.1.1]# ./bin/kafka-avro-console-consumer --new-consumer --bootstrap-server localhost:9092 --topic test-oracle-jdbc-USERS --from-beginning

Got the message, but not stop running:

The --new-consumer option is deprecated and will be removed in a future major release.The new consumer is used by default if the --bootstrap-server option is provided.

Connection URL: jdbc:oracle:thin:@**ContainerIPAddress:1521**:orcl11g?

# Kafka Connect MongoDB Sink

<https://www.confluent.io/connector/kafka-connect-mongodb-sink/>

<https://github.com/RADAR-base/MongoDb-Sink-Connector> 🡨 following this instruction !

1. Download latest version, **kafka-connect-mongodb-sink-0.2.2**:

<https://github.com/RADAR-base/MongoDb-Sink-Connector/releases>

1. Transfer to docker container

[gjiang@ip-10-103-157-48 software]$ docker cp kafka-connect-mongodb-sink-0.2.2.jar 7fa5e286b664:/software/

1. Copy to confluent java folder where jar files are held.

[root@7fa5e286b664 software]# cp **kafka-connect-mongodb-sink-0.2.2.jar** /confluent-4.1.1/share/java/

And export CLASSPATH for this jar file and put it to .bash\_profile file, need to run it first every time.

[root@7fa5e286b664 java]# export CLASSPATH=/confluent-4.1.1/share/java/kafka-connect-mongodb-sink-0.2.2.jar

1. **Create “sink-connect-mongodb.properties**” file in /confluent-4.1.1/etc:

[root@7fa5e286b664 etc]# more sink-connect-mongodb.properties

# Kafka consumer configuration

name=kafka-connector-mongodb-sink

# Kafka connector configuration

connector.class=org.radarcns.connect.mongodb.MongoDbSinkConnector

tasks.max=1

# Topics that will be consumed

**topics=users**

# MongoDB server

mongo.host=localhost

mongo.port=27017

# MongoDB configuration

mongo.username=

mongo.password=

**mongo.database=mydb**

# Collection name for putting data into the MongoDB database. The {$topic} token will be replaced

# by the Kafka topic name.

#mongo.collection.format={$topic}

# Factory class to do the actual record conversion

#record.converter.class=org.radarcns.connect.mongodb.serialization.RecordConverterFactory

1. If use this command “/confluent-4.1.1/bin/confluent start” to start zookeeper-server, kafka-server, schema-registry and kafka-rest, and run command below, will get connection error!
2. Run the MongoDB-Sink-Connector

./bin/connect-standalone ./etc/schema-registry/connect-avro-standalone.properties ./etc/sink-connect-mongodb.properties

at org.apache.kafka.connect.cli.ConnectStandalone.main(ConnectStandalone.java:95)

[2018-06-22 19:59:27,149] ERROR Stopping after connector error *(org.apache.kafka.connect.cli.ConnectStandalone:113)*

*org.apache.kafka.connect.errors.ConnectException: Unable to start REST server*

*at org.apache.kafka.connect.runtime.rest.RestServer.start(RestServer.java:204))*

*...*

*[2018-06-22 19:59:27,150] INFO Kafka Connect stopping (org.apache.kafka.connect.runtime.Connect:65)*

*[2018-06-22 19:59:27,150] INFO Stopping REST server (org.apache.kafka.connect.runtime.rest.RestServer:211)*

*[2018-06-22 19:59:27,152] INFO Stopped http\_8083@d08edc5{HTTP/1.1}{0.0.0.0:8083} (org.eclipse.jetty.server.ServerConnector:306)*

*[2018-06-22 19:59:27,162] INFO Stopped o.e.j.s.ServletContextHandler@234cff57{/,null,UNAVAILABLE} (org.eclipse.jetty.server.handler.ContextHandler:865)*

*[2018-06-22 19:59:27,165] INFO REST server stopped (org.apache.kafka.connect.runtime.rest.RestServer:222)*

*[2018-06-22 19:59:27,165] INFO Herder stopping (org.apache.kafka.connect.runtime.standalone.StandaloneHerder:77)*

*[2018-06-22 19:59:27,165] INFO Worker stopping (org.apache.kafka.connect.runtime.Worker:151)*

*[2018-06-22 19:59:27,166] INFO Stopped FileOffsetBackingStore (org.apache.kafka.connect.storage.FileOffsetBackingStore:67)*

*[2018-06-22 19:59:27,171] INFO Worker stopped (org.apache.kafka.connect.runtime.Worker:172)*

*[2018-06-22 19:59:27,172] INFO Herder stopped (org.apache.kafka.connect.runtime.standalone.StandaloneHerder:87)*

*[2018-06-22 19:59:27,173] INFO Kafka Connect stopped (org.apache.kafka.connect.runtime.Connect:70)*

1. Have to start zookeeper, kafka server, schema registry and kafka rest separately:

# Start zookeeper

[root@7fa5e286b664 confluent-4.1.1]# ./bin/zookeeper-server-start ./etc/kafka/zookeeper.properties

# Start kafka-broker

[root@7fa5e286b664 confluent-4.1.1]# ./bin/kafka-server-start ./etc/kafka/server.properties

# Start schema-registry

[root@7fa5e286b664 confluent-4.1.1]# ./bin/schema-registry-start ./etc/schema-registry/schema-registry.properties

# Start kafka-rest

[root@7fa5e286b664 confluent-4.1.1]# ./bin/kafka-rest-start ./etc/kafka-rest/kafka-rest.properties

1. **Run connect-mongodb-sink-connector (need to run .bash\_profile to set CLASSPATH first)**

[root@7fa5e286b664 confluent-4.1.1]# ./bin/connect-standalone ./etc/schema-registry/connect-avro-standalone.properties ./etc/sink-connect-mongodb.properties

Above running is about Worker (connect-standalone) and Plugin (plug.in in sink-connect-mongodb.properties file)?

**Need to modify /confluent-4.1.1/etc/schema-registry/connect-avro-standalone.properties for plug.in**

**Default in connect-avro-standalone.properties is plugin.path=share/java. Should change to**

**plugin.path=/confluent-4.1.1/share/java/kafka-connect-mongodb-sink-0.2.2.jar**

**or**

**plugin.path=/confluent-4.1.1/share/java/**

*[2018-06-25 15:03:25,420]* ***ERROR WorkerSinkTask{id=kafka-connector-mongodb-sink-0****}* ***Task threw an uncaught and unrecoverable exception (org.apache.kafka.connect.runtime.WorkerTask:172)***

*java.lang.NoClassDefFoundError: com/mongodb/MongoException*

*at org.radarcns.connect.mongodb.MongoDbSinkTask.createMongoDbWriter(MongoDbSinkTask.java:114)*

*at org.radarcns.connect.mongodb.MongoDbSinkTask.start(MongoDbSinkTask.java:95)*

*at org.apache.kafka.connect.runtime.WorkerSinkTask.initializeAndStart(WorkerSinkTask.java:281)*

*at org.apache.kafka.connect.runtime.WorkerSinkTask.execute(WorkerSinkTask.java:170)*

*at org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.java:170)*

*at org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:214)*

*at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)*

*at java.util.concurrent.FutureTask.run(FutureTask.java:266)*

*at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)*

*at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)*

*at java.lang.Thread.run(Thread.java:748)*

*Caused by: java.lang.ClassNotFoundException: com.mongodb.MongoException*

*at java.net.URLClassLoader.findClass(URLClassLoader.java:381)*

*at java.lang.ClassLoader.loadClass(ClassLoader.java:424)*

*at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:349)*

*at java.lang.ClassLoader.loadClass(ClassLoader.java:357)*

*... 11 more*

*[2018-06-25 15:03:25,421] ERROR WorkerSinkTask{id=kafka-connector-mongodb-sink-0} Task is being killed and will not recover until manually restarted (org.apache.kafka.connect.runtime.WorkerTask:173)*

*[2018-06-25 15:03:25,421] INFO Stopping MongoDBSinkTask (org.radarcns.connect.mongodb.MongoDbSinkTask:163)*

*[2018-06-25 15:03:25,421] INFO Stopped MongoDBSinkTask (org.radarcns.connect.mongodb.MongoDbSinkTask:185)*

For some reason, schema-registry stop running, and rerun again

1. Stream sample data into **Kafka topics** in sink-connect-mongodb.properties

* Can see in /confluent-4.1.1/etc/sink-connect-mongodb.properties file, **topics=avrotest**
* Run this command:

[root@7fa5e286b664 confluent-4.1.1]# curl -X POST -H "Content-Type: application/vnd.kafka.avro.v2+json" \

-H "Accept: application/vnd.kafka.v2+json" \

--data '{"value\_schema": "{\"type\": \"record\", \"name\": \"User\", \"fields\": [{\"name\": \"name\", \"type\": \"string\"}]}", "records": [{"value": {"name": "testUser"}}]}' \

"http://localhost:8082/topics/avrotest "

And got the following response (that is right!):

**{"offsets":[{"partition":0,"offset":0,"error\_code":null,"error":null}],"key\_schema\_id":null,"value\_schema\_id":21}**

1. Check mongodb db, but can not see “avrotest” collection in mongo:

> show collections

customers

month

month1

people

students

users

1. Check kafka topic, can see topics that were created , but can not be found them in MongoDB:

[root@7fa5e286b664 bin]# ./kafka-topics --list --zookeeper localhost:2181

\_\_confluent.support.metrics

\_\_consumer\_offsets

\_schemas

avrotest

test-oracle-jdbc-USERS

users

That means No connection between Kafka and Mongo:

………………………….

[2018-06-25 15:03:25,419] INFO 0 have been processed (org.radarcns.connect.mongodb.MongoDbSinkTask:57)

[2018-06-25 15:03:25,420] **ERROR WorkerSinkTask{id=kafka-connector-mongodb-sink-0} Task threw an uncaught and unrecoverable exception (org.apache.kafka.connect.runtime.WorkerTask:172)**

java.lang.NoClassDefFoundError: com/mongodb/MongoException

at org.radarcns.connect.mongodb.MongoDbSinkTask.createMongoDbWriter(MongoDbSinkTask.java:114)

at org.radarcns.connect.mongodb.MongoDbSinkTask.start(MongoDbSinkTask.java:95)

at org.apache.kafka.connect.runtime.WorkerSinkTask.initializeAndStart(WorkerSinkTask.java:281)

at org.apache.kafka.connect.runtime.WorkerSinkTask.execute(WorkerSinkTask.java:170)

at org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.java:170)

at org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:214)

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)

at java.util.concurrent.FutureTask.run(FutureTask.java:266)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)

at java.lang.Thread.run(Thread.java:748)

Caused by: java.lang.ClassNotFoundException: com.mongodb.MongoException

at java.net.URLClassLoader.findClass(URLClassLoader.java:381)

at java.lang.ClassLoader.loadClass(ClassLoader.java:424)

at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:349)

at java.lang.ClassLoader.loadClass(ClassLoader.java:357)

... 11 more

[2018-06-25 15:03:25,421] ERROR WorkerSinkTask{id=kafka-connector-mongodb-sink-0} Task is being killed and will not recover until manually restarted (org.apache.kafka.connect.runtime.WorkerTask:173)

[2018-06-25 15:03:25,421] INFO Stopping MongoDBSinkTask (org.radarcns.connect.mongodb.MongoDbSinkTask:163)

[2018-06-25 15:03:25,421] INFO Stopped MongoDBSinkTask (org.radarcns.connect.mongodb.MongoDbSinkTask:185)

Reference:

<https://github.com/hpgrahsl/kafka-connect-mongodb/issues/36>

<https://docs.confluent.io/current/connect/userguide.html#installing-plugins>

set message to

<https://github.com/hpgrahsl/kafka-connect-mongodb/issues/new>

and Robin in Slack:

<https://confluentcommunity.slack.com/messages/C49L0V3L7/convo/C49L0V3L7-1529438474.000328/>

Hello,

Thank you for your article and help! I followed your instruction to set up the connection between Kafka and MongoDB (sink), After I run ./bin/connect-standalone ./etc/schema-registry/connect-avro-standalone.properties ./etc/sink-connect-mongodb.properties:", I run curl......, I can get the correct response," {"offsets":[{"partition":0,"offset":0,"error\_code":null,"error":null}],"key\_schema\_id":null,"value\_schema\_id":21}" and see "avrotest in kafka topic, however I got error below and cannot find “avrotest” collection in mongo:

......

[2018-06-25 15:03:25,419] INFO 0 have been processed (org.radarcns.connect.mongodb.MongoDbSinkTask:57)

[2018-06-25 15:03:25,420] \*\*ERROR WorkerSinkTask{id=kafka-connector-mongodb-sink-0} Task threw an uncaught and unrecoverable exception (org.apache.kafka.connect.runtime.WorkerTask:172)

java.lang.NoClassDefFoundError: com/mongodb/MongoException\*\*

at org.radarcns.connect.mongodb.MongoDbSinkTask.createMongoDbWriter(MongoDbSinkTask.java:114)

at org.radarcns.connect.mongodb.MongoDbSinkTask.start(MongoDbSinkTask.java:95)

at org.apache.kafka.connect.runtime.WorkerSinkTask.initializeAndStart(WorkerSinkTask.java:281)

at org.apache.kafka.connect.runtime.WorkerSinkTask.execute(WorkerSinkTask.java:170)

at org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.java:170)

at org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:214)

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)

at java.util.concurrent.FutureTask.run(FutureTask.java:266)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)

at java.lang.Thread.run(Thread.java:748)

Caused by: java.lang.ClassNotFoundException: com.mongodb.MongoException

at java.net.URLClassLoader.findClass(URLClassLoader.java:381)

at java.lang.ClassLoader.loadClass(ClassLoader.java:424)

at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:349)

at java.lang.ClassLoader.loadClass(ClassLoader.java:357)

... 11 more

[2018-06-25 15:03:25,421] ERROR WorkerSinkTask{id=kafka-connector-mongodb-sink-0} Task is being killed and will not recover until manually restarted (org.apache.kafka.connect.runtime.WorkerTask:173)

[2018-06-25 15:03:25,421] INFO Stopping MongoDBSinkTask (org.radarcns.connect.mongodb.MongoDbSinkTask:163)

[2018-06-25 15:03:25,421] INFO Stopped MongoDBSinkTask (org.radarcns.connect.mongodb.MongoDbSinkTask:185).

I know the connection between Kafka and MongoDB is failed, but cannot figure out why. Can you help to give some suggestions? Do I need to set up plugin, configuration, workers.....specifically before running the connect-standalone? what parts did I miss?

Thank you very much for your time and help! my email is gjiang@psi-it.com

gw

Check log files in /tmp/confluent.NFBX0ijJ

1. Create plugin folder:

Mkdir /usr/local/share/kafka/plugins

And set plugin path:

plugin.path**=**/usr/local/share/kafka/plugins

install Kafka Connect plugins using the plugin path as described above.??

A Kafka Connect plugin is:

1. an uber JAR containing all of the classfiles for the plugin and its third-party dependencies in a single JAR file; or
2. a directory on the file system that contains the JAR files for the plugin and its third-party dependencies.

All of the classes that implement or are used by a connector are defined in a **connector plugin**.

Connectors and tasks are logical units of work and must be scheduled to execute in a process. Kafka Connect calls these processes **workers** and has two types of workers: **standalone** and **distributed**.