Notes on How to Deploy Proposal Manager - on CentOS 7

1. MySQL database server

(1)_ Install and Start — MySQL server

```
# wget https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm
# rpm –ivh mysql80-community-release-el7-3.noarch.rpm
# yum install mysql-server
# systemctl start mysqld
# systemctl status mysqld
# grep 'temporary password'  /var/log/mysqld.log
# mysql -u root -p  (input the temporary password, and set a new one)
```

(2)_ Create MySQL database tables

```
# mysql -u root -p
# (input your own new possword)
mysql> (indicating MySQL is OK, ready to create tables)
```

(3)_ Connecting to Node Express server

modify table — reviewers with columns as id /ldap_username /first_name /last_name /email /phone_number /created_at /updated_at /user_status

```
var con = mysql.createConnection({
        host: "127.0.0.1",
        user: "root"
        password: "Boston4now&"
        database: "vaProject"
})

===> This is for MacOS connection ===
var con = mysql.createConnection({
        host: "127.0.0.1",
        user: "VA_Boston2020",
         password: "Boston4now",
         database: "Proposals_MRS"
 });
===> This is for Docker connection ===
  var con = mysql.createConnection({
         host: "192.168.1.10",
         user: "hengzhi2",
         password: "123456",
         database: "proposal_database",
         port: "3306"
});
```

(4)_ Create 3 tables with the schema (copied from Sequel Pro, 6-12-2020)

```
CREATE TABLE `reviewers` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `ldap_username` char(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT '',
  `first_name` char(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT '',
  `last_name` char(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT '',
  `email` char(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL,
  `phone_number` bigint(50) DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `updated_at` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
  `user_status` char(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT 'regular',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=40 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `proposals` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `title` varchar(250) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT '',
  `VA_sponsor` char(100) DEFAULT NULL,
  `support` char(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL,
  `project_presenter` char(50) DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `updated_at` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
  `stage` int(11) DEFAULT NULL,
  `status` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT NULL,
  `deleted` char(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT 'false',
  `cycle` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `id` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2019284 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `reviewdata` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `proposal_id` int(11) unsigned NOT NULL,
  `reviewer_id` int(11) unsigned NOT NULL,
  `business_score` int(11) DEFAULT NULL,
  `business_comms` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
  `feasibility_score` int(11) DEFAULT NULL,
  `feasibility_comms` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
  `resources_score` int(11) DEFAULT NULL,
  `resources_comms` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
  `commitment_score` int(11) DEFAULT NULL,
  `commitment_comms` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
  `constraints_score` int(11) DEFAULT NULL,
  `constraints_comms` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
  `overall_comms` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `updated_at` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
  `submitted` char(25) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT 'false',
  PRIMARY KEY (`id`),
  KEY `FK_reviewdata_proposals` (`proposal_id`),
  KEY `FK_reviewdata_reviewers` (`reviewer_id`),
  CONSTRAINT `FK_reviewdata_proposals` FOREIGN KEY (`proposal_id`) REFERENCES `proposals` (`id`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `FK_reviewdata_reviewers` FOREIGN KEY (`reviewer_id`) REFERENCES `reviewers` (`id`) ON
DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=112 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

2. Run and test on VM CentOS 7

(1)_ Frontend — React — $ npm start
(2)_ Backend — Node Express — $ npm start
(3)_ Test the at Frontend — CRUD, Scoring, Report, Search (OR/AND)

3. OpenLDAP server connecting to Node Express server

(1)_ # systemctl status slapd __ SEE the Daemon is active (running)
(2)_ # **service httpd start**
(3)_ # netstat -antup l grep 389 __ SEE the tcp /tcp6 LISTEN at port 389
(4)_ Run LAM on CentOS 7 __ in order to CRUD users/groups easily
cd /var/www/html/
httpd -t __ SEE syntax ok, ignore other info
ifconfig __ SEE enp03: inet 192.168.1.27 (last one changes)
IP address 192.168.1.27 to launch LAM (LAM login password as laptop)

(5)_ $ npm install ldapjs --save

(6)_ at Node app.js, adding the following code:

- $ npm install ldapjs —save
- at Node app.js, adding the following code:

```
var ldap = require('ldapjs');
// —— create a client ——binding ——
function authenticateDN(username, password){
    var client = ldap.createClient({
        url: 'ldap://127.0.0.1:389'
    });
//__ netstat -antup l grep 192.168 to SEE the current PORT
    client.bind(username, password,
        function(err){
            if(err){
                authenObj.passLDAP = false;
                console.log("Error in new LDAP connection: " + err);
            } else {
                authenObj.passLDAP = true;
                console.log("LDAP connection: Success! username: " + username);

                var opts = {
                  // filter: '(objectClass=*)',              // ==> ALL
                  // filter: '(|(uid=2)(sn=Jhon)(uid-3))',   // ==> OR
                  // filter: '(&(uid=2)(sn=Jhon))'           // ==> AND
                  // filter: '(uidNumber=2008)',
```

```
        // filter: '(uid=hwang)',              // ==> LDAP username
         filter: '(sn=wang)',
         scope: 'sub',
         attributes: ['sn']
       };

    client.search('ou=People', opts, function(err, res) {
       if(err){
           console.log("Error in Search: " + err);
       } else {
           res.on('searchEntry', function(entry) {
              console.log('entry: ' + JSON.stringify(entry.object));
            });
            res.on('searchReference', function(referral) {
              console.log('referral: ' + referral.uris.join());
            });
            res.on('error', function(err) {
              console.error('error: ' + err.message);
            });
            res.on('end', function(result) {
              console.log('status: ' + result.status);
            });
       }
    });

exports.loginValidation = function (req, res) {
   // after integrating this web portal into VA intranet dashboard, it is supposed
   // to decode SSO(single sign-on) token and get LDAP username to replace the
next "req.body.username";
   // "Proposal Manager" requires LDAP "username" only, no password, no LDAP
authen for one more time;
   // add OpenLDAP search() with "req.body.username" and "req.body.password"
to get  the DN for the next
  // ------------------ Search function ----------------------
   authenObj.username = req.body.username;
    authenObj.password = req.body.password;
          //__ username is the user “dn” in the next
    authenticateDN("cn=admin,dc=openldap,dc=local","516405");
}
```

4. Docker — Dockerfile and docker-compose (local at MacOS)

(1)_ Define a volume (not using local folder) for mysql-image data persisting

```
$ docker volume create proposal_data_vol
$ docker run -d -v proposal_data_vol:/var/lib/mysql -p 3600:3600
```

(2)_ Define a Docker network (with a name)

```
$ docker network create proposal_net
```

(3a)_ run mysql-Image to create a Container that can store database

```
$ docker run
        --name newContainerName
        --net proposal_net
            -v proposal_data_vol:/var/lib/mysql
            -d -p 3600:3600 mysqlImageName
# docker exec -it newContainerName bash
/# mysql -u root -p 123
```

(3b)_ the above 1-2-3 can be replaced by a Dockerfile as the following

```
==> create a Dockerfile for Official mysql Image
FROM mysql:latest
ENV MYSQL_ROOT_PASSWORD 123          ==> SEE above assigned
ENV MYSQL_DATAVBASE dockerdb         ==> to create a NEW database
ENV MYSQL_USER hengzhi               ==> to create a NEW user
ENV MYSQL_PASSWORD 123456            ==> to create a NEW password
### ADD whzscripts.sql  /docker-entrypoint-initdb.d
        ==> import my scripts;
EXPOSE 3306
————————————————————————————————————
$ docker build -t hengzhi2020/proposal_db_img .
$ docker inspect hengzhi2020/proposal_db_img
$ docker run --name proposal_db_cont --net proposal_net -v proposal_data_vol:/
var/lib/mysql -d -p 3306:3306 hengzhi2020/proposal_db_img
$ docker exec -it proposal_db_cont bash
/# (sign for inside the container) mysql -uroot -p123
```

(4)_ after creating a Docker Container by official mysql_Image with a “volume”, which can store the database data/tables for future use after the container stops. And now can can create tables as it is working for regular MySQL database server, SEE above 1(4) —“create 3 tables … ”.

It is suggested to use GUI tool to create and update tables with Sequel Pro at MacOs, and MySQL Workbench at Linux or Windows. The mysql database connection information is

```
host: “192.168.1.10",       ==> SEE by using $ ifconfig | grep 168
user: "hengzhi2",
password: "123456",
database: "proposal_database",
port: "3306"
```

(5)_ MySQL (after version 8.0.0) password Issue — default using SHA-2

It is suggested to use 3 CMD to resume previous regular password

```
mysql>  CREATE USER 'hengzhi2'@'%' IDENTIFIED WITH mysql_native_password BY ‘123456’;
        ==> BE CAREFUL OF your copy —> maybe: lost a single-quotation-mark !
mysql> GRANT ALL PRIVILEGES ON *.* TO ‘hengzhi2’@'%';
        ==> BE CAREFUL OF your copy —> maybe: lost a single-quotation-mark !
mysql> flush privileges;
```

(6)_ Backend: Node Express Server

==> create a Dockerfile for Backend

```
FROM node:11.10.0

RUN mkdir -p /app
WORKDIR /app

COPY package.json /app
COPY package-lock.json /app

RUN npm install
COPY . /app

EXPOSE 8000

CMD ["npm", “start"]
```

==> modify the connection to Docker mysql Container

```
var con = mysql.createConnection({
  host: “192.168.1.10",       ==> SEE by using $ ifconfig | grep 168
  user: "hengzhi2",
  password: "123456",
  database: "proposal_database",
```

```
        port: "3306"
    });
```

```
$ docker build -t hengzhi2020/proposal_node .
$ docker run --name propopsal_node --net proposal_net -v
proposal_data_vol -it -p 8000:8000 hengzhi2020/proposal_node
```

==> If no conflict, Node Express server will be running at terminal

(7)_ Frontend: React, CSS - SemanticUI

==> create a Dockerfile for Frontend

```
FROM node:11.10.0

RUN mkdir -p /app
WORKDIR /app

COPY package.json /app
COPY package-lock.json /app

RUN npm install
COPY . /app

EXPOSE 3000

CMD ["npm", “start”]
```

```
$ docker build -t hengzhi2020/proposal_react .
$ docker run --name propopsal_react --net proposal_net -it -p
3000:3000 hengzhi2020/proposal_react
```

---

5. miscellaneous — additional notes

(1)_ The login interface requires “user name”and “password”, but it does not check the password at the backend; “user name” should be the same as LDAP login userName; supposed the user has already passed LDAP authentication, and the web portal fetch the LDAP userName, which will be the reviewers’ user name; If LDAP can not search out this user name, there will be an alert window;

(2)_ It is a MUST that the reviewers table should store all the reviewers’ username firstly, because (a) only listed reviewers can access this web portal; (b) when the Admin (super user) creates a proposal, it will automatically give assignment to all the reviewers;

(3)_ By default, all the reviewers are “regular”, except “admin” (super user, has the privilege to CRUD, designated by developer’s hard coding on the db table).