rpb-doc-internal Documentation

Release 1

Wahyu Wijaya Hadiwikarta

Apr 23, 2018

CONTENTS:

CHAPTER

ONE

OPENCLINICA

This article documents the installation of OpenClinica. OpenClinica is an open source software to create electronic CRF (Case Report Form). In RadPlanBio, it is one of the main components.

To begin with, login as root (`sudo su`). Perform `yum update`. Install some support packages such as yum-plugin-remove-with-leaves, links, bash-completion, net-tools, unzip, wget, vim, mlocate, epel-release, lsof, bzip2 and gcc.

Prior to installation, create an install folder to store the required .rpm and .war files.

```
mkdir /usr/local/oc
mkdir /usr/local/oc/install
```

1.1 Configurations for OpenClinica

OS	Init	Container, JDK	Database
CentOS 7	`systemd`	Tomcat 7, JDK 7	PostgreSQL 8.4

1.2 List of required files

Tomcat 7	apache-tomcat-7.0.84.tar.gz
JDK 7	jdk-7u80-linux-x64.rpm
PostgreSQL 8.4	postgresql84-8.4.22-1PGDG.rhel6.x86_64.rpm postgresql84-libs-8.4.22-1PGDG.rhel6.x86_64.rpm postgresql84-server-8.4.22-1PGDG.rhel6.x86_64.rpm postgresql84-contrib-8.4.22-1PGDG.rhel6.x86_64.rpm postgresql84-docs-8.4.22-1PGDG.rhel6.x86_64.rpm

1.3 JDK

Remove OpenJDK if originally instaled on the system. Install Oracle jdk

```
cd /usr/local/oc/install
yum install jdk-7u80-linux-x64.rpm
ln -s /usr/java/default /usr/local/java
```

1.4 Container

Install container Tomcat 7

```
cd /usr/local/oc/install
mkdir /usr/tomcat
tar xvzf apache-tomcat-7.0.84.tar.gz
mv apache-tomcat-7.0.84 /usr/tomcat/.
ln -s /usr/tomcat/apache-tomcat-7.0.84 /usr/tomcat/latest
ln -s /usr/tomcat/latest /usr/tomcat/default
ln -s /usr/tomcat/default /usr/local/tomcat
```

Create user for tomcat service

```
groupadd tomcat1
useradd -g tomcat1 tomcat1
chown -Rf tomcat1.tomcat1 /usr/tomcat/apache-tomcat-7.0.84
chown -Rf tomcat1.tomcat1 /usr/tomcat/latest/
chown -Rf tomcat1.tomcat1 /usr/tomcat/default/
chown -Rf tomcat1.tomcat1 /usr/local/tomcat/
```

Create .service file for init

```
vim /lib/systemd/system/tomcat.service
```

```
[Unit]
Description=Tomcat version 7.0.85
Documentation=https://tomcat.apache.org/download-70.cgi
After=syslog.target
After=network.target

[Service]
Type=forking
Restart=always

User=tomcat1
Group=tomcat1

Environment=JAVA_HOME=/usr/local/java
Environment=CATALINA_HOME=/usr/local/tomcat
Environment='JAVA_OPTS=-Xms128m -Xmx512m -XX:PermSize=128m'

ExecStart=/usr/local/tomcat/bin/startup.sh
ExecStop=/usr/local/tomcat/bin/shutdown.sh
SuccessExitStatus=143

TimeoutSec=0

[Install]
WantedBy=multi-user.target
```

Enable and start the service

```
systemctl enable tomcat.service
systemctl start tomcat.service
systemctl status tomcat.service
```

1.5 Database

To install PostgreSQL 8.4, the rpms are needed to be downloaded first

```
cd /usr/local/oc/install/
wget https://yum.postgresql.org/8.4/redhat/rhel-6-x86_64/postgresql84-libs-8.4.22-
↪1PGDG.rhel6.x86_64.rpm
wget https://yum.postgresql.org/8.4/redhat/rhel-6-x86_64/postgresql84-8.4.22-1PGDG.
↪rhel6.x86_64.rpm
wget https://yum.postgresql.org/8.4/redhat/rhel-6-x86_64/postgresql84-server-8.4.22-
↪1PGDG.rhel6.x86_64.rpm
wget https://yum.postgresql.org/8.4/redhat/rhel-6-x86_64/postgresql84-contrib-8.4.22-
↪1PGDG.rhel6.x86_64.rpm
wget https://yum.postgresql.org/8.4/redhat/rhel-6-x86_64/postgresql84-docs-8.4.22-
↪1PGDG.rhel6.x86_64.rpm
```

Install the rpms

```
yum install postgresql84-libs-8.4.22-1PGDG.rhel6.x86_64.rpm
yum install postgresql84-8.4.22-1PGDG.rhel6.x86_64.rpm
yum install postgresql84-server-8.4.22-1PGDG.rhel6.x86_64.rpm
yum install uuid
yum install postgresql84-contrib-8.4.22-1PGDG.rhel6.x86_64.rpm
yum install postgresql84-docs-8.4.22-1PGDG.rhel6.x86_64.rpm
```

Initialize database for postgres

```
/etc/init.d/postgresql-8.4 initdb
```

Edit `postgresql.conf`

```
cp /var/lib/pgsql/8.4/data/postgresql.conf /var/lib/pgsql/8.4/data/postgresql.conf.BAK
vim /var/lib/pgsql/8.4/data/postgresql.conf
```

```
listen addresses = '*'
port = 5432
```

Create symlink to `/opt/PostgreSQL`

```
mkdir /opt/PostgreSQL
ln -s /usr/pgsql-8.4 /opt/PostgreSQL/8.4
ln -s /var/lib/pgsql/8.4/data /opt/PostgreSQL/8.4/data
```

Prior to running the database, create an init .service file (non-optimized)

```
rm /etc/init.d/postgresql-8.4
vim /lib/systemd/system/postgresql-8.4.service
```

```
[Unit]
Description=PostgreSQL 8.4 database server
Documentation=https://www.postgresql.org/docs/8.4/static/index.html
After=syslog.target
After=network.target

[Service]
Type=forking
```

(continues on next page)

(continued from previous page)

```
Restart=always

User=postgres
Group=postgres

OOMScoreAdjust=-1000
Environment=PG_OOM_ADJUST_FILE=/proc/self/oom_score_adj
Environment=PG_OOM_ADJUST_VALUE=0

Environment=PGDATA=/var/lib/pgsql/8.4/data/

ExecStart=/usr/pgsql-8.4/bin/pg_ctl start -D ${PGDATA} -s -w -t 300
ExecStop=/usr/pgsql-8.4/bin/pg_ctl stop -D ${PGDATA} -s -m fast
ExecReload=/usr/pgsql-8.4/bin/pg_ctl reload -D ${PGDATA} -s

TimeoutSec=0

[Install]
WantedBy=multi-user.target
```

Enable and start the service

```
systemctl enable postgresql-8.4.service
systemctl start postgresql-8.4.service
systemctl status postgresql-8.4.service
```

Assign password to user postgres

```
su postgres
psql

ALTER USER postgres PASSWORD 'xxxxxxxxx';
\q

exit
```

Edit authentication method to access database by editing `pg_hba.conf`

```
cp /var/lib/pgsql/8.4/data/pg_hba.conf /var/lib/pgsql/8.4/data/pg_hba.conf.BAK
vim /var/lib/pgsql/8.4/data/pg_hba.conf
```

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD
local   all             all                                     md5
host    all             all             127.0.0.1/32            md5
```

Setup database for OpenClinica

```
psql -U postgres -c "CREATE ROLE clinica LOGIN ENCRYPTED PASSWORD 'xxxxxxxxx'␣
↪SUPERUSER NOINHERIT NOCREATEDB NOCREATEROLE"
psql -U postgres -c "CREATE DATABASE openclinica WITH ENCODING='UTF8' OWNER=clinica"
psql -U postgres

ALTER USER clinica WITH PASSWORD 'xxxxxxxxx';
\q
```

1.6 OpenClinica

We go to the install folder and unzip OpenClinica war file

```
systemctl stop tomcat.service
```

```
cd /usr/local/oc/install
unzip OpenClinica.war -d OpenClinica
mv OpenClinica /usr/local/tomcat/webapps
```

Edit `datainfo.properties` file

```
vim /usr/local/tomcat/webapps/OpenClinica/WEB-INF/classes/datainfo.properties
```

```
dbType=postgres
dbUser=clinica
dbPass=xxxxxxxxx
db=openclinica
dbPort=5432
dbHost=localhost

filePath=${catalina.home}/${WEBAPP.lower}.data/

sysURL=http://localhost:8080/${WEBAPP}/MainMenu

log.dir=${catalina.home}/logs/openclinica
logLocation=local

logLevel=info
syslog.host=localhost
syslog.port=514
```

Start tomcat and depends on the `filePath` parameter, a folder will be created that contains a new `datainfo.properties`. Subsequent changes on the settings must be performed on this file.

```
systemctl start tomcat.service
```

Test connection to OpenClinica

```
links 127.0.0.1:8080/OpenClinica/MainMenu
```

1.7 OpenClinica Web Service

The procedure is the same like installing OpenClinica

We go to the install folder and unzip OpenClinica war file

```
systemctl stop tomcat.service
```

```
cd /usr/local/oc/install
unzip OpenClinica-ws.war -d OpenClinica-ws
mv OpenClinica-ws /usr/local/tomcat/webapps
```

Edit `datainfo.properties` file

```
vim /usr/local/tomcat/webapps/OpenClinica-ws/WEB-INF/classes/datainfo.properties
```

```
dbType=postgres
dbUser=clinica
dbPass=xxxxxxxxx
db=openclinica
dbPort=5432
dbHost=localhost

filePath=${catalina.home}/${WEBAPP.lower}.data/

sysURL=http://localhost:8080/${WEBAPP}/MainMenu

log.dir=${catalina.home}/logs/openclinica
logLocation=local

logLevel=info
syslog.host=localhost
syslog.port=514
```

Start Tomcat and another folder will be created too with a new `datainfo.properties` file.

```
systemctl start tomcat.service
```

Test connection to OpenClinica Web Service

```
links 127.0.0.1:8080/OpenClinica-ws/MainMenu
```

CHAPTER

TWO

PACS CONQUEST

This article documents the installation of PACS Conquest. Conquest is an open source project for PACS. In the RadPlanBio project, Conquest is used as the DICOM server component.

To begin with, login as root (`sudo su`). Perform `yum update`. Install some support packages such as yum-plugin-remove-with-leaves, links, bash-completion, net-tools, unzip, wget, vim, mlocate, epel-release, lsof, bzip2 and gcc.

2.1 Configurations for DICOM server

OS	Init	Server	Database
CentOS 7	`systemd`	Apache2	PostgreSQL 9.5

2.2 Database

For database PostgreSQL 9.5, the installation steps is as follows

```
yum install https://download.postgresql.org/pub/repos/yum/9.5/redhat/rhel-7-x86_64/
↪pgdg-centos95-9.5-3.noarch.rpm
yum install postgresql95 postgresql95-server postgresql95-contrib postgresql95-docs
/usr/pgsql-9.5/bin/postgresql95-setup initdb
```

Edit `postgresql.conf`

```
cp /var/lib/pgsql/9.5/data/postgresql.conf /var/lib/pgsql/9.5/data/postgresql.conf.BAK
vim /var/lib/pgsql/9.5/data/postgresql.conf
```

```
listen addresses = '*'
port = 5432
```

Enable init script

```
systemctl enable postgresql-9.5.service
systemctl start postgresql-9.5
systemctl status postgresql-9.5.service
```

Create symlink to `/opt/PostgreSQL`

```
mkdir /opt/PostgreSQL
ln -s /usr/pgsql-9.5 /opt/PostgreSQL/9.5
ln -s /var/lib/pgsql/9.5/data /opt/PostgreSQL/9.5/data
```

Assign password to user postgres

```
su postgres
psql

ALTER USER postgres PASSWORD 'xxxxxxxxx';
\q

exit
```

Edit authentication method to access database by editing `pg_hba.conf`

```
cp /var/lib/pgsql/9.5/data/pg_hba.conf /var/lib/pgsql/9.5/data/pg_hba.conf.BAK
vim /var/lib/pgsql/9.5/data/pg_hba.conf
```

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD
local   all             all                                     md5
host    all             all             127.0.0.1/32            md5
```

Setup the database for Conquest

```
psql -U postgres -c "CREATE ROLE conquest LOGIN ENCRYPTED PASSWORD 'xxxxxxxxx'␣
↪SUPERUSER NOINHERIT NOCREATEDB NOCREATEROLE"
psql -U postgres -c "CREATE DATABASE conquest WITH ENCODING='UTF8' OWNER=conquest"
psql -U postgres

ALTER USER conquest WITH PASSWORD 'xxxxxxxxx';
\q
```

Restart PostgreSQL 9.5 service

```
systemctl restart postgresql-9.5.service
systemctl status postgresql-9.5.service
```

2.3 Web Server

Install Apache2 via yum

```
yum install httpd
rm -f /etc/httpd/conf.d/welcome.conf
```

Edit `httpd.conf`

```
cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.BAK
vim /etc/httpd/conf/httpd.conf
```

```
ServerAdmin admin@domain.de
ServerName hostname.domain.de:80

<IfModule dir_module>
  DirectoryIndex index.html index.htm
</IfModule>
```

Test syntax and enable init service

```
apachectl configtest
systemctl enable httpd.service
systemctl start httpd.service
systemctl status httpd.service
```

Enable CGI and manage SELinux*

```
yum install policycoreutils-python
setsebool -P httpd_enable_cgi 1
semanage fcontext -a -t httpd_sys_script_exec_t /var/www/cgi-bin
restorecon -Rv /var/www/cgi-bin
systemctl restart httpd.service
systemctl status httpd.service
```

Additionally if wanting to test CGI with perl or ruby script

```
yum install perl perl-CGI ruby
```

2.4 Compiling the DICOM server

Start by installing required packages

```
yum install gcc-c++ gcc-c++-sh-linux-gnu clang
```

Create an install folder

```
mkdir /usr/local/pacs/install
```

Download dicomserver1419b.zip from https://ingenium.home.xs4all.nl/dicom.html and save to install folder.

Unzip the zip file

```
unzip /usr/local/pacs/install/dicomserver1419b.zip -d /usr/local/pacs/dicomserver1419b
```

Create symlink to `/opt`

```
ln -s /usr/local/pacs/dicomserver1419b/distribution /opt/conquest-14-19b
```

Create folder for incoming DICOM data

```
mkdir /opt/conquest-14-19b/data/incoming
```

Create user and group `conquest`

```
groupadd conquest
useradd -g conquest conquest
```

Remove shell login for user `conquest`

```
vim /etc/passwd
```

```
conquest:x:1000:1000::/home/conquest:/bin/false
```

Change ownership and set permission

```
chown -R conquest:conquest /usr/local/pacs/dicomserver1419b
chown -R conquest:conquest /opt/conquest-14-19b
chmod g+w /opt/conquest-14-19b/data/incoming
```

Prepare for compiling source code

```
ln -s /usr/pgsql-9.5/lib/libpq.so.5 /usr/lib/libpq.so
mkdir /usr/local/man
mkdir /usr/local/man/man1
```

Edit the `total.cpp` file to move `aaac.cxx` on top `qrsop.cxx`. Uncomment the line below `aarj.cxx` so `aaac.cxx` will be read first sequentially and will not be compiled twice. This is because the function min() that is required to compile `qrsop.cxx` is defined in `aaac.cxx`.

```
cd /opt/conquest-14-19b/src/dgate/src
vim total.cpp
```

```
#include "aaac.cxx"
#include "qrsop.cxx"

#include "aarj.cxx"
//#include "aaac.cxx"
```

Edit/create new compile script

```
cd /opt/conquest-14-19b
cp maklinux maklinux.BAK
chmod 755 maklinux
vim maklinux
```

```
#!/bin/bash

SRC=./src/dgate;
CONF=./linux/conf;
LINUX=./linux;

chmod 777 src/dgate/jpeg-6c/configure
cd src/dgate/jpeg-6c
./configure
make
sudo make install
cd ../../..

export LD_LIBRARY_PATH="/usr/pgsql-9.5/lib/";
gcc -o $SRC/lua.o -c $SRC/lua_5.1.5/all.c -I$SRC/lua_5.1.5 -DLUA_USE_DLOPEN -DLUA_USE_
↪POSIX;
g++ -std=c++11 -o $SRC/charls.o -c $SRC/charls/all.cpp -I$SRC/charls
gcc -o $SRC/openjpeg.o -c $SRC/openjpeg/all.c -I$SRC/openjpeg

g++ -std=c++11 -I/usr/pgsql-9.5/include/ -DUNIX -DNATIVE_ENDIAN=1 -DHAVE_LIBJPEG -
↪DPOSTGRES -DHAVE_LIBCHARLS -DHAVE_LIBOPENJPEG2 -Wno-write-strings $SRC/lua.o $SRC/
↪charls.o $SRC/openjpeg.o -o dgate -lpthread -ldl -I$SRC/src $SRC/src/total.cpp -I
↪$SRC/dicomlib -L/usr/pgsql-9.5/lib/ -lpq  -ljpeg -I$SRC/jpeg-6c -L$SRC/jpeg-6c -I
↪$SRC/lua_5.1.5 -I$SRC/openjpeg -I$SRC/charls -Wno-multichar;

rm $SRC/lua.o;
```

(continues on next page)

(continued from previous page)

```
rm $SRC/charls.o;
rm $SRC/openjpeg.o;

pkill -9 dgate;
sleep 0.2s;

cp $CONF/dicom.ini.postgres dicom.ini;
cp $CONF/dicom.sql.postgres dicom.sql;

cp $LINUX/acrnema.map acrnema.map;
cp $LINUX/dgatesop.lst dgatesop.lst;

cp /opt/conquest-14-19b/dgate /var/www/cgi-bin/dgate ;
cp /opt/conquest-14-19b/dicom.sql /var/www/cgi-bin/dicom.sql ;
cp /opt/conquest-14-19b/acrnema.map /var/www/cgi-bin/acrnema.map ;

cp -r /opt/conquest-14-19b/webserver/cgi-bin/* /var/www/cgi-bin;
cp -r /opt/conquest-14-19b/webserver/cgi-bin/.lua /var/www/cgi-bin;
cp -r /opt/conquest-14-19b/webserver/cgi-bin/.lua.linux /var/www/cgi-bin;

cp /var/www/cgi-bin/dicom.ini.linux /var/www/cgi-bin/dicom.ini;
cp /var/www/cgi-bin/newweb/dicom.ini.linux /var/www/cgi-bin/newweb/dicom.ini;
cp /var/www/cgi-bin/.lua.linux /var/www/cgi-bin/.lua;

cp /opt/conquest-14-19b/dgate /var/www/cgi-bin/newweb/dgate ;
cp /opt/conquest-14-19b/acrnema.map /var/www/cgi-bin/newweb/acrnema.map ;

cp -r /opt/conquest-14-19b/webserver/htdocs/* /var/www;
cp -r /opt/conquest-14-19b/webserver/htdocs/* /var/www/html;

mkdir /opt/conquest-14-19b/logs
chown -R conquest:conquest /opt/conquest-14-19b/
```

Run the compile script

```
/opt/conquest-14-19b/maklinux
```

If everything is right, on CentOS 7, compilation will run smooth with some warnings that can be ignored.

2.5 Setting the DICOM server

After finish with installation of the DICOM server, the next step is to have the correct settings.

File	Path
dicom.ini	/opt/conquest-14-19b/dicom.ini /var/www/cgi-bin/dicom.ini /var/www/cgi-bin/newweb/dicom.ini
acrnema.map	/opt/conquest-14-19b/acrnema.map /var/www/cgi-bin/acrnema.map /var/www/cgi-bin/newweb/acrnema.map

2.5.1 Edit dicom.ini

```
vim /opt/conquest-14-19b/dicom.ini
```

```
[sscscp]
MicroPACS               = sscscp

# Network configuration: server name and TCP/IP port#
MyACRNema               = NNNNNNNN
TCPPort                 = 5678

# Host for postgres or mysql only, name, username and password for database
SQLHost                 = localhost
SQLServer               = conquest
Username                = conquest
Password                = xxxxxxxxx
PostGres                = 1
MySQL                   = 0
SQLite                  = 0
DoubleBackSlashToDB     = 1
UseEscapeStringConstants = 1

# Configure server
ImportExportDragAndDrop  = 1
ZipTime                 = 05:
UIDPrefix               = 99999.99999
EnableComputedFields    = 1

FileNameSyntax          = 4

# Configuration of compression for incoming images and archival
DroppedFileCompression   = un
IncomingCompression      = un
ArchiveCompression       = as

# For debug information
PACSName                = NNNNNNNN
OperatorConsole         = 127.0.0.1
DebugLevel              = 0

# Configuration of disk(s) to store images
MAGDeviceFullThreshold   = 30
MAGDevices              = 1
MAGDevice0              = /opt/conquest-14-19b/data/

# Files to store logs
StatusLog = /opt/conquest-14-19b/logs/serverstatus.log
TroubleLog = /opt/conquest-14-19b/logs/pacstrouble.log
UserLog = /opt/conquest-14-19b/logs/pacsuser.log
```

```
vim /var/www/cgi-bin/dicom.ini
```

```
[sscscp]
MicroPACS               = sscscp

# database layout (copy dicom.sql to the web server script directory or point to the␣
↪one in your dicom server directory)
```

(continues on next page)

(continued from previous page)

```
kFactorFile              = /opt/conquest-14-19b/dicom.sql

# gives access to the SQL server of the DICOM server
# use of independent database is also allowed (depends on scripts used)

SQLHost                  = localhost
SQLServer                = conquest
Username                 = conquest
Password                 = xxxxxxxxx
PostGres                 = 1
MySQL                    = 0
SQLite                   = 0
DoubleBackSlashToDB      = 1
UseEscapeStringConstants = 1

# gives access to all DICOM servers known in acrnema.map

ACRNemaMap               = /opt/conquest-14-19b/acrnema.map
Dictionary               = /opt/conquest-14-19b/dgate.dic
SOPClassList             = /opt/conquest-14-19b/dgatesop.lst

# default IP address and port of DICOM server (may be non-local, web pages empty if␣
↪wrong)

WebServerFor             = 127.0.0.1
TCPPort                  = 5678

# AE title: only used if web client originates queries or moves

MyACRNema                = NNNNNNNN

# path to script engine: ocx will not download images if wrong - shows as black␣
↪square with controls

WebScriptAddress         = http://127.0.0.1/cgi-bin/dgate

# if set to 1 (default), the web user cannot edit databases and (in future) other␣
↪things
# webpush enables push of data to other servers

WebReadonly              = 0
WebPush                  = 1
```

```
vim /var/www/cgi-bin/newweb/dicom.ini
```

```
[sscscp]
MicroPACS                = sscscp
ACRNemaMap               = acrnema.map
Dictionary               = dgate.dic
WebServerFor             = 127.0.0.1
TCPPort                  = 5678
```

2.5.2 Edit acrnema.map

For `acrnema.map`, every files are identical in the content and should contain the correct AE name i.e. NNNNNNNN

```
NNNNNNNN                127.0.0.1       5678            un
```

2.6 Running the DICOM server

Upon completion of the installation which includes compilation and settings, the database will need to be initialized for the DICOM server.

```
/opt/conquest-14-19b/dgate -v -r
```

Create init script for dgate service

```
vim /lib/systemd/system/dgate.service
```

```
[Unit]
Description=Conquest DICOM Server dgate
Documentation=https://ingenium.home.xs4all.nl/dicom.htm
After=syslog.target
After=network.target

[Service]
Type=simple

User=conquest
Group=conquest

ExecStart=/opt/conquest-14-19b/dgate -v -L/opt/conquest-14-19b/logs/serverstatus.log

[Install]
WantedBy=multi-user.target
```

Enable and start the service

```
systemctl enable dgate.service
systemctl start dgate.service
systemctl status dgate.service
```

Now DICOM server is online and ready to receive connection. Set SELinux to Permissive before testing the web service.

```
vim /etc/selinux/config
```

```
SELINUX=permissive
SELINUXTYPE=targeted
```

Open the web service with any browser e.g. links to localhost

```
links http://127.0.0.1/cgi-bin/dgate?mode=top
links http://127.0.0.1/cgi-bin/newweb/dgate
```