

```

#Sample Evergreen Install Script
#Used by PINES staff to install test servers for bug
squashing activities
#Replace XXXX with local info
#!/bin/bash

OSRF_GIT_URL=git://git.evergreen-ils.org/OpenSRF.git
OSRF_GIT_BRANCH=main
EG_GIT_URL=git://git.evergreen-ils.org/Evergreen/Ever
green.git
EG_GIT_BRANCH=main
DB_USER=evergreen
DB_PASS=evergreen
DB_HOST=localhost
DB_PORT=XXXX
DB_NAME=evergreen
ADMIN_USER=admin
ADMIN_PASS=demo123
OSRF_REPODIR=$(echo $OSRF_GIT_URL | awk -F/ '{print
$NF}' | sed -e 's/.git//')
EG_REPODIR=$(echo $EG_GIT_URL | awk -F/ '{print $NF}'
| sed -e 's/.git//')
DISTRONAME="$(lsb_release -si | tr [:upper:]
[:lower:])-${lsb_release -sc | tr [:upper:]
[:lower:]}"

Usage () {
    echo "$0 [-m] [-o] [-h]"
    echo "-m install Evergreen Main"
    echo "-o also install OpenSRF"
    echo "-h show this message"
    echo
}

while getopts moh OPTIONS
do case "$OPTIONS" in
    m) MAIN="1";;
    o) INSTALL_OSRF="1";;
    h) Usage ; exit 1;;
    [?]) Usage ; exit 2;;
    esac
done

#stop opensrf
StopOpenSRF () {
systemctl stop opensrf
}

#install opensrf

```

```

CreateOsrifUser () {
if [ ! $(grep opensrf /etc/passwd) ]; then
    useradd -m -s /bin/bash opensrf
fi
}

UpdateOsrifGit () {
if [ ! -e /home/opensrf/$OSRF_REPODIR ]; then
    su - opensrf -c "git clone $OSRF_GIT_URL"
fi
su - opensrf -c "cd $OSRF_REPODIR && git checkout
$OSRF_GIT_BRANCH"
cd /home/opensrf/$OSRF_REPODIR
git pull
}

InstallOpenSRF () {
cd /home/opensrf/$OSRF_REPODIR
apt update
make -f src/extras/Makefile.install $DISTRONAME
su - opensrf -c "cd $OSRF_REPODIR && autoreconf -i &&
./configure --prefix=/openils
--sysconfdir=/openils/conf --with-websockets-port=443
&& make"
make install
chown -R opensrf:opensrf /openils
}

#install evergreen
UpdateEgGit () {
if [ ! -e /home/opensrf/$OSRF_REPODIR ]; then
    su - opensrf -c "git clone $EG_GIT_URL"
fi
su - opensrf -c "cd $EG_REPODIR && git checkout
$EG_GIT_BRANCH"
cd /home/opensrf/$EG_REPODIR
git pull
}

InstallEG () {
cd /home/opensrf/$EG_REPODIR
make -f Open-ILS/src/extras/Makefile.install
$DISTRONAME
make -f Open-ILS/src/extras/Makefile.install
${DISTRONAME}-developer
# build AngularJS
su - opensrf -c "cd
$EG_REPODIR/Open-ILS/web/js/ui/default/staff && npm
install && npm run build-prod &&
CHROME_BIN=/usr/bin/chromium-browser npm run test"
# build Angular

```

```

su - opensrf -c "cd $EG_REPODIR/Open-ILS/src/eg2 &&
npm install && ng build --configuration production &&
npm run test"
# install bootstrap deps
su - opensrf -c "cd
$EG_REPODIR/Open-ILS/web/opac/deps && npm install"
# build & install EG
su - opensrf -c "cd $EG_REPODIR &&
PATH=$PATH:/openils/bin autoreconf -i &&
PATH=$PATH:/openils/bin ./configure --prefix=/openils
--sysconfdir=/openils/conf && make"
make install
chown -R opensrf:opensrf /openils
chown -R opensrf:opensrf /home/opensrf
}

#replace database
KillDBConns () {
read -r -d '' KILL_Q <<EOF
SELECT
    pg_terminate_backend(pid)
FROM
    pg_stat_activity
WHERE
    -- don't kill my own connection!
    pid <> pg_backend_pid()
    -- don't kill the connections to other databases
    AND datname = 'evergreen'
;
EOF
echo $KILL_Q
psql -U evergreen -c "$KILL_Q"
}

InstallDB () {
cd /home/opensrf/$EG_REPODIR && \
perl Open-ILS/src/support-scripts/eg_db_config \
--update-config \
--service all \
--create-database \
--create-schema \
--create-offline \
--user $DB_USER \
--password $DB_PASS \
--hostname $DB_HOST \
--port $DB_PORT \
--database $DB_NAME \
--admin-user $ADMIN_USER \
--admin-pass $ADMIN_PASS \
--load-all-sample
}

```

```
#start it all back up
StartOpenSRF () {
systemctl start opensrf
systemctl restart apache2 nginx websocketd-osrf
}

InstallMain () {
StopOpenSRF
if [ ! -z $INSTALL_OSRF ]; then
    CreateOsrfsUser
    UpdateOsrfsGit
    InstallOpenSRF
fi
UpdateEgGit
InstallEG
KillDBConns
InstallDB
StartOpenSRF
}

Install () {
StopOpenSRF
if [ ! -z $INSTALL_OSRF ]; then
    InstallOpenSRF
fi
InstallEG
KillDBConns
InstallDB
StartOpenSRF
}

if [ -z $MAIN ]; then
    Install
else
    InstallMain
fi
```