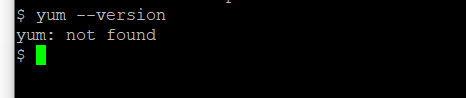
**Steps to setup git and Jenkins on IBMi**

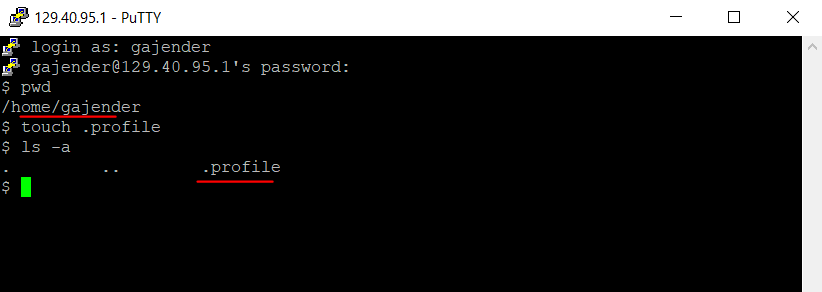
Using SSH connection is a preferred way to communicate with PASE rather that QP2TERM, but for this basic steps you can use wither. I have used PuTTY for SSH connection.

To get started with open source you should have yum package manager in you IBMi

1. **Check if YUM package manager is installed or not.**

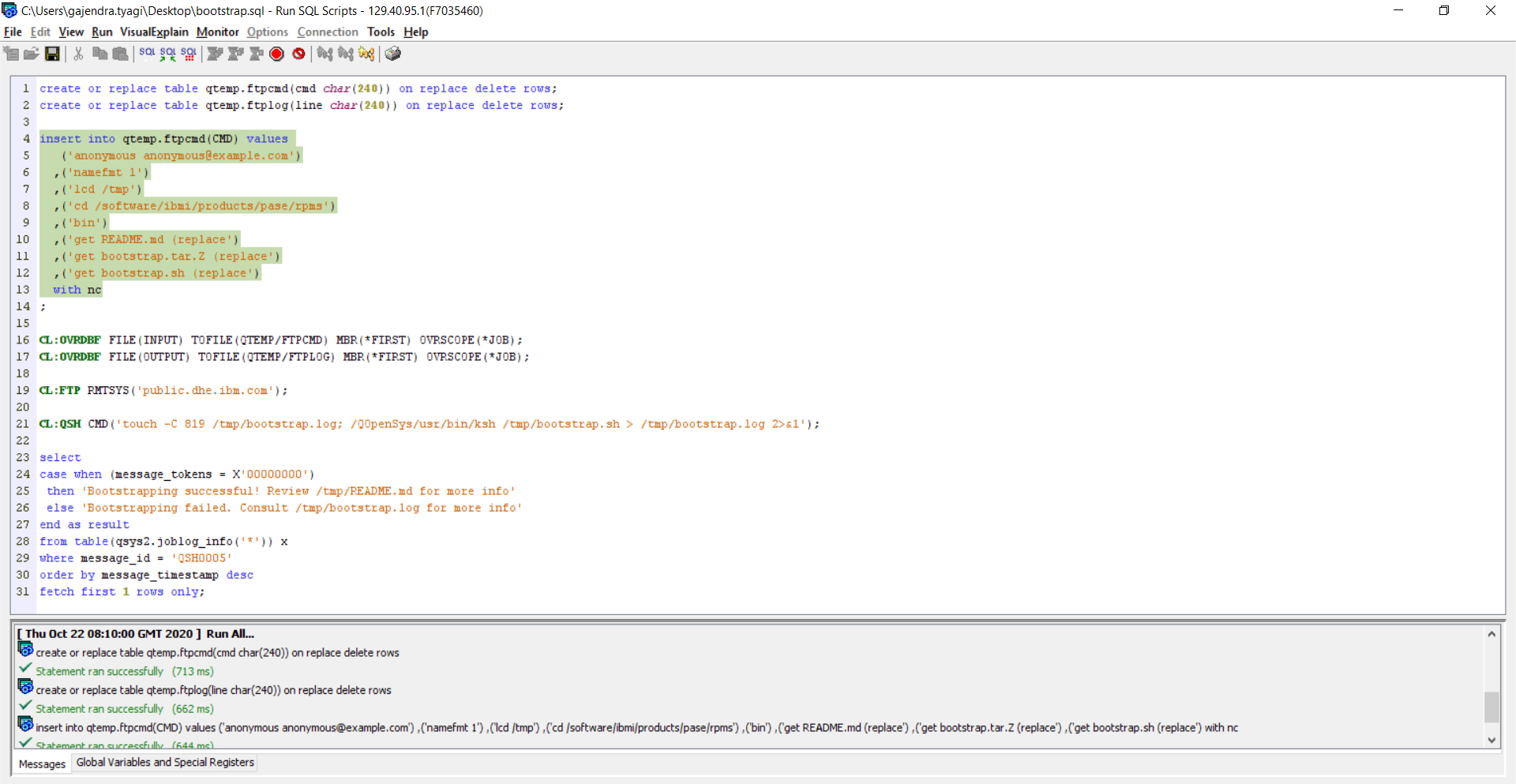


1. **Check if the home directory has a folder with your profile name, if not then create one with `mkdir userProfile` and create a hidden file in it with command `touch .profile` (dot profile).**

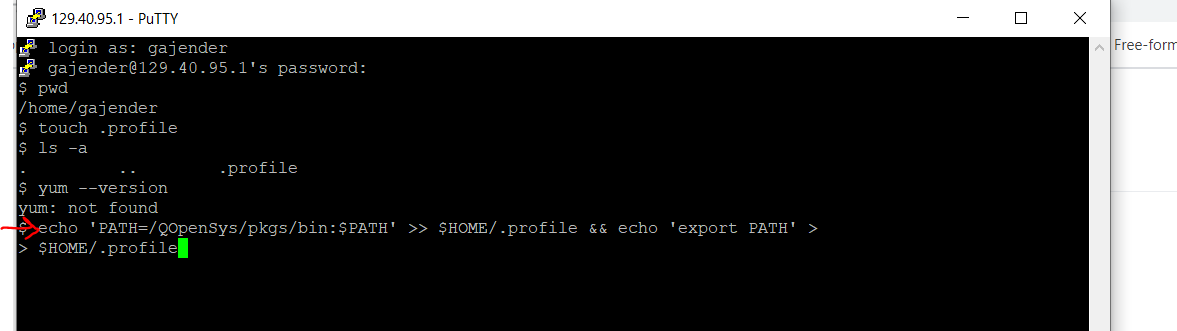


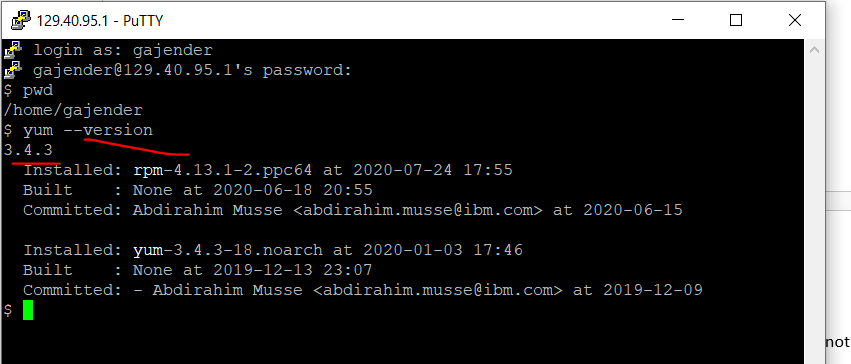
1. **Use the following link if you don’t have yum setup already**

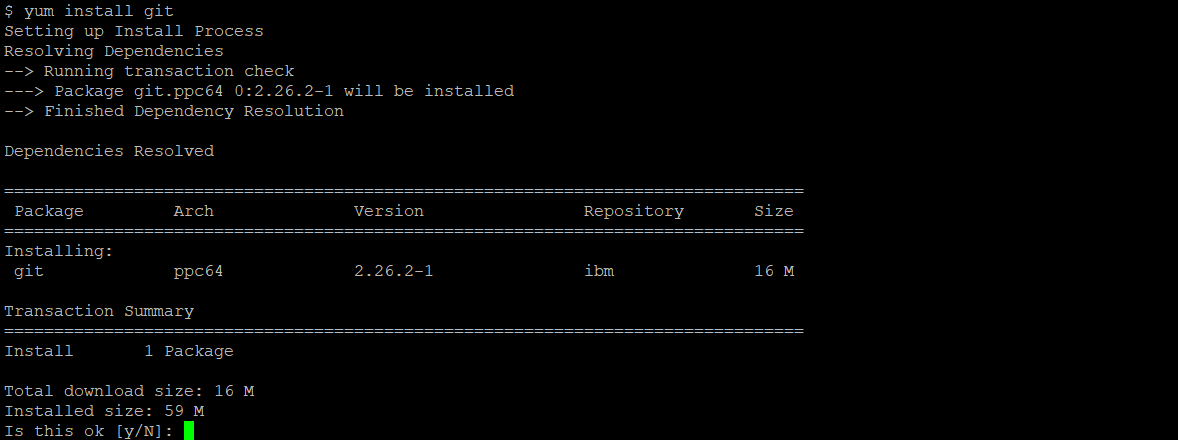
<https://github.com/Programmersio-IBMi/OSSonIBMi>

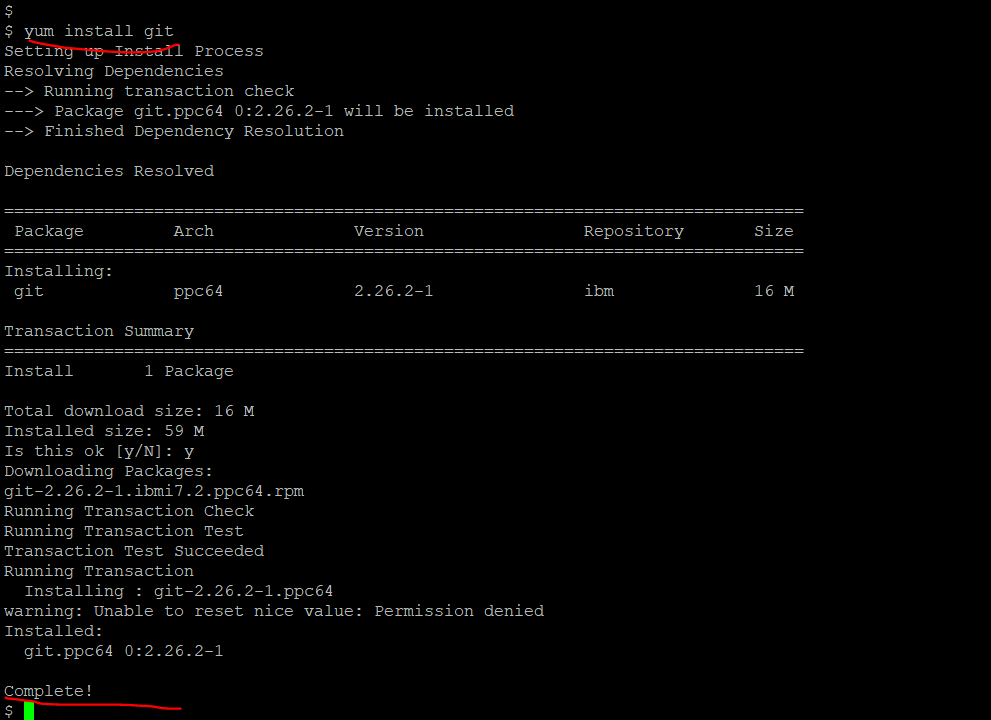
1. Execute bootstap.sql from acs. Even if it shows fail at the end do not panic it ran fine.   
     
   
2. Run this command after acs has ran.

* echo 'PATH=/QOpenSys/pkgs/bin:$PATH' >> $HOME/.profile && echo 'export PATH' >> $HOME/.profile

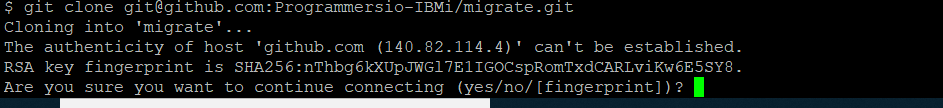
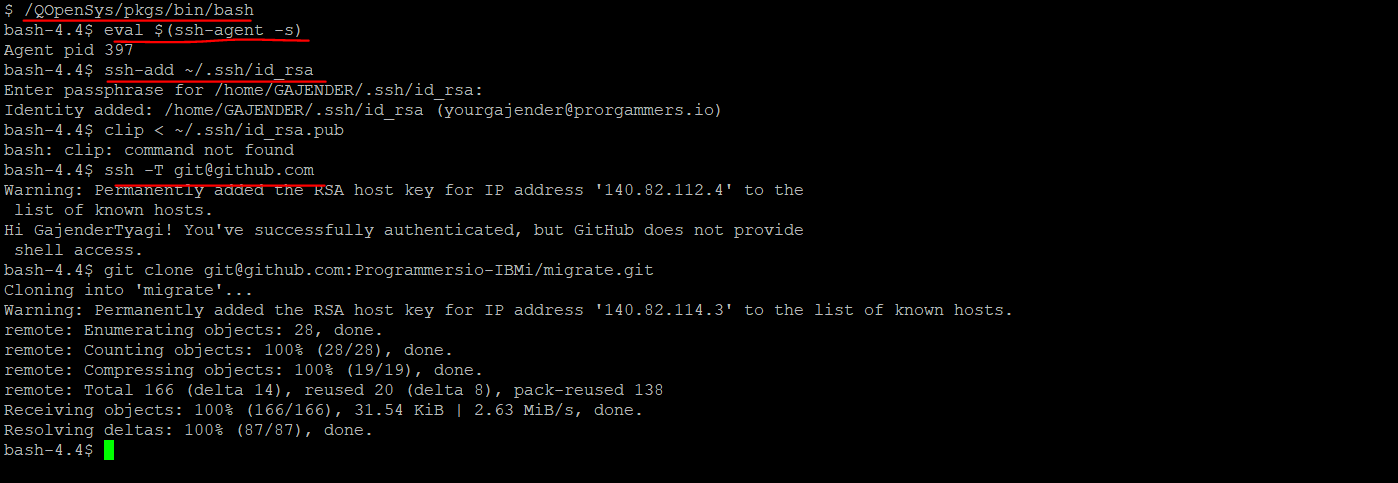


1. 
2. Some YUM commands you must know <https://access.redhat.com/sites/default/files/attachments/rh_yum_cheatsheet_1214_jcs_print-1.pdf>
3. **Install git using YUM**







1. **Now git can be used locally on IBMi but it can be used on IFS objects only**, git cannot track native sources from libraries. All sources have to be moved to ifs in a structured manner.
2. **To use remote repository like from GitHub we need to have SSH key setup which can be done following these steps -**https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent   
     
   Without SSH setup you won’t be able to clone remote repositories.  
     
   
3. **If you are using SSH connection then run this command first :**   
   /QOpenSys/pkgs/bin/bash
4. **If you have followed all steps properly then you are good to go.**   
     
   
5. **Setup Jenkins, reference link** : <https://github.com/worksofliam/blog/issues/43>
6. `wget <http://mirrors.jenkins.io/war-stable/latest/jenkins.war>`
7. **Once Jenkins is installed its time to run it.**

java -jar jenkins.war --httpPort=9090

1. **Use above command to run initially and do one time setup.**
2. **Complete the initial setup.**
3. **Once the initial setup is done we now should have a batch job running Jenkins so that we can monitor that.**  
     
   a. Move to the path where you have Jenkins setup   
     
   b. Create a shell file and paste the below content

**export** PATH=/QOpenSys/pkgs/bin:/home/gajender/.ssh:$PATH;

**export** TERM=xterm;

**export** SHELL=/QOpenSys/usr/bin/bsh;

**export** SHLVL=1;

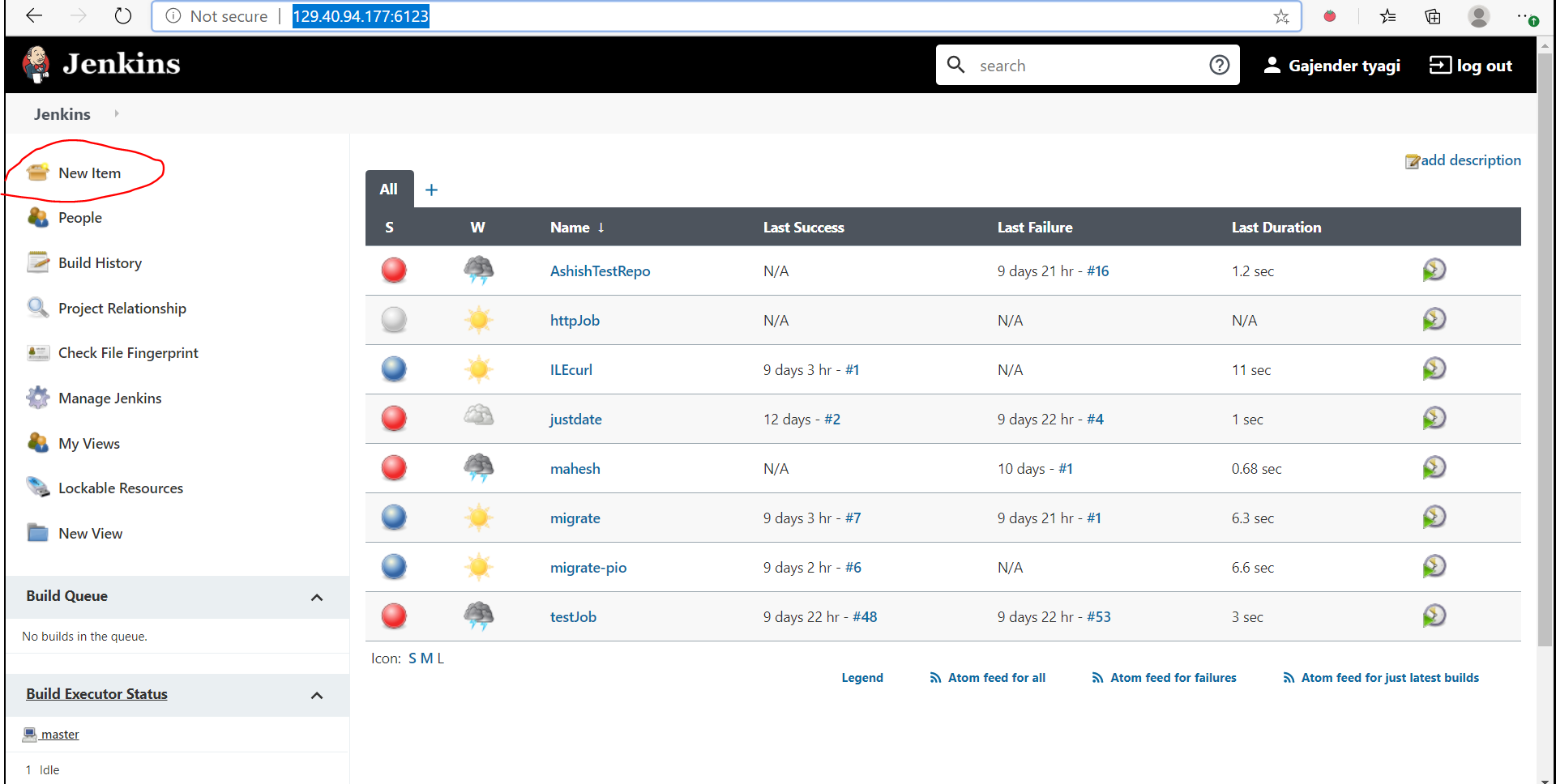
cd /home/gajender/jenkins;

java -jar jenkins.war --httpPort=6123

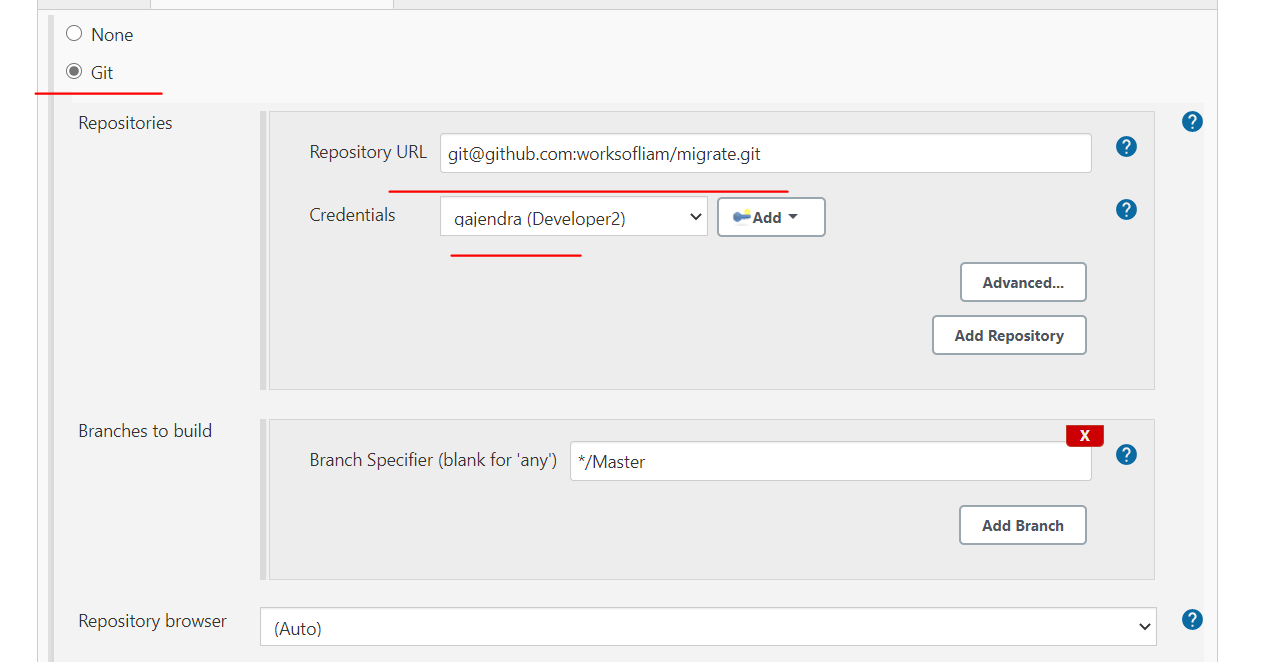
change user Gajender with your user

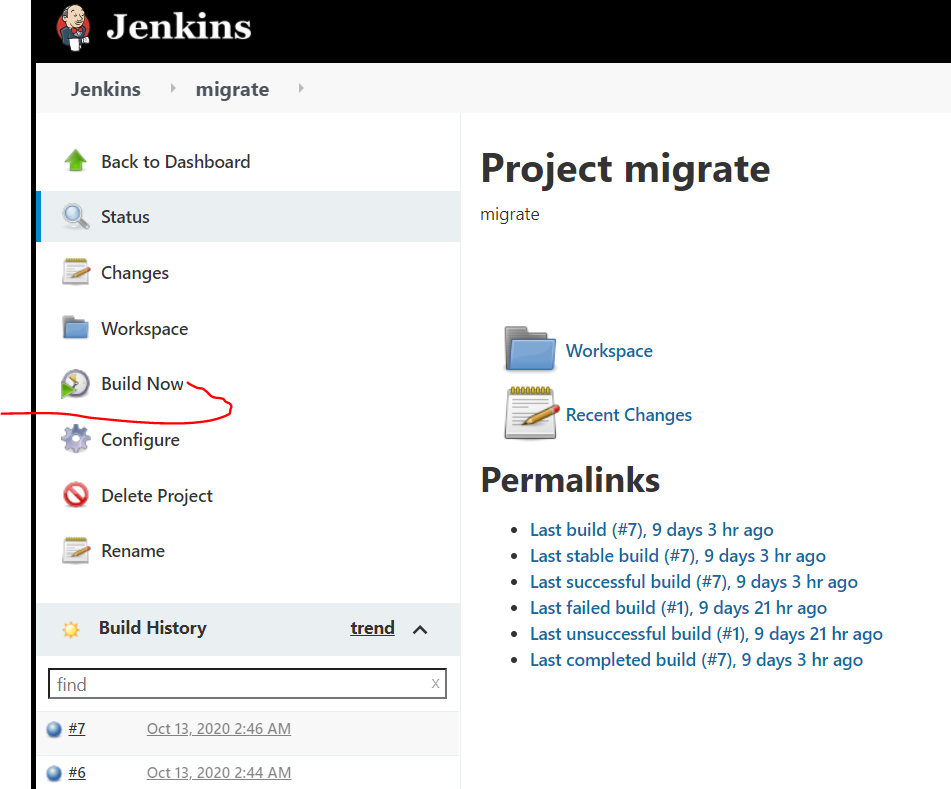
c. Save this file and run the following command from 5250 session   
QSH CMD('/QOpenSys/pkgs/bin/bash /home/gajender/jenkins/runJenkins.sh')

1. **New Item is used to setup a job to perform the build process.**



1. **Inside the item config you will mention the repo link and the user with correct authority.**



1. 
2. 
3. MAKEFILE lives in projects repository and the structure can be as shown below, MAKEFILE is configured as per requirement.

**A standard MAKEFILE**

SMY\_ENV=DEV

DBGVIEW=\*SOURCE

SHELL=/QOpenSys/usr/bin/qsh

BIN\_LIB=PROFSRVDEV

DTA\_LIB=$(BIN\_LIB)

SRC\_LIB=$(BIN\_LIB)

MBR\_PGM=QRPGLESRC

MBR\_DDS=QDDSSRC

LIBLIST=$(BIN\_LIB)  ASODTAPDEV AAPDTANEWT APODTANEWT AARDTANEWT CASPGM AWODTANEWT DSTPGMSMY APOPGMSMY AARPGMSMY AWOPGMSMY PUIDEV PJSDEV

all: rpdlssps.pgm bidyspsls.pgm bil12spsls.pgm rpl12spsls.pgm biwkspsls.pgm rpwkspsls.pgm

*# Daily sales person sales*

bidyspsls.pgm: bidyspsls.sqlrpglemod dlsspsrcd.sqlrpglemod

bidyspsls.sqlrpglemod: bidyspslsd.dspf

rpdlssps.pgm: rpdlssps.sqlrpglemod dlsspsrcd.sqlrpglemod

rpdlssps.sqlrpglemod: rpdlsspsp.prtf asoscmv1

asoscmv1:

    -db2 -t -f ./buildSrc/asoscmv1.sql $(DTA\_LIB) | /usr/bin/iconv -t 819 -f 37;

    system "CPYFRMSTMF FROMSTMF('./buildSrc/asoscmv1.sql') TOMBR('/QSYS.lib/$(SRC\_LIB).lib/$(MBR\_DDS).file/asoscmv1.mbr') MBROPT(\*replace)"

%.rpgle:

    liblist -d PROFSRVDEV; \

    liblist -al $(LIBLIST);\

    system "CRTRPGMOD MODULE($(BIN\_LIB)/$\*) SRCSTMF('./src/qrpglesrc/$\*.rpgle') DBGVIEW($(DBGVIEW)) REPLACE(\*YES)"

    system "CPYFRMSTMF FROMSTMF('./src/qrpglesrc/$\*.rpgle') TOMBR('/QSYS.lib/$(SRC\_LIB).lib/$(MBR\_PGM).file/$\*.mbr') MBROPT(\*replace)"

%.c:

    system "CRTCMOD MODULE($(BIN\_LIB)/$\*) SRCSTMF('./src/$\*.c') DBGVIEW($(DBGVIEW)) REPLACE(\*YES)"

%.clle:

*#Can't compile CL from IFS on all OS versions... < 7.1*

    -system -q "CRTSRCPF FILE($(BIN\_LIB)/$(MBR\_PGM)) RCDLEN(112)"

    system "CPYFRMSTMF FROMSTMF('./src/$\*.clle') TOMBR('/QSYS.lib/$(BIN\_LIB).lib/$(MBR\_PGM).file/$\*.mbr') MBROPT(\*replace)"

    system "CRTCLMOD MODULE($(BIN\_LIB)/$\*) SRCFILE($(BIN\_LIB)/$(MBR\_PGM)) DBGVIEW($(DBGVIEW))"

%.dspf:

    liblist -d PROFSRVDEV; \

    liblist -al $(LIBLIST); \

    system "CPYFRMSTMF FROMSTMF('./src/qddssrc/$\*.dspf') TOMBR('/QSYS.lib/$(SRC\_LIB).lib/$(MBR\_DDS).file/$\*.mbr') MBROPT(\*replace)"

    system "CRTDSPF FILE($(BIN\_LIB)/$\*) SRCFILE($(SRC\_LIB)/$(MBR\_DDS)) SRCMBR($\*) REPLACE(\*YES)"

%.prtf:

    liblist -d PROFSRVDEV; \

    liblist -al $(LIBLIST); \

    system "CPYFRMSTMF FROMSTMF('./src/qddssrc/$\*.prtf') TOMBR('/QSYS.lib/$(SRC\_LIB).lib/$(MBR\_DDS).file/$\*.mbr') MBROPT(\*replace)"

    system "CRTPRTF FILE($(BIN\_LIB)/$\*) SRCFILE($(SRC\_LIB)/$(MBR\_DDS)) SRCMBR($\*) DEVTYPE(\*AFPDS) IGCDTA(\*N) IGCEXNCHR(\*N) IGCCHRRTT(\*N) IGCCPI(\*N) IGCSOSI(\*N) IGCCDEFNT(\*SYSVAL) REPLACE(\*YES) "

%.pgm:

    liblist -d PROFSRVDEV; \

    liblist -d PUIDEV;\

    liblist -al $(LIBLIST);\

    system "CRTPGM PGM($(BIN\_LIB)/$\*) MODULE($(patsubst %,$(BIN\_LIB)/%,$(basename $^))) ENTMOD($(BIN\_LIB)/$\*) REPLACE(\*YES)"

%.sqlrpglemod:

    liblist -d PROFSRVDEV;\

    liblist -d PUIDEV;\

    liblist -al $(LIBLIST);\

    system "CRTSQLRPGI OBJ($(BIN\_LIB)/$\*) SRCSTMF('./src/qrpglesrc/$\*.sqlrpglemod') OBJTYPE(\*MODULE) REPLACE(\*YES)"

    system "CPYFRMSTMF FROMSTMF('./src/qrpglesrc/$\*.sqlrpglemod') TOMBR('/QSYS.lib/$(SRC\_LIB).lib/$(MBR\_PGM).file/$\*.mbr') MBROPT(\*replace)"

%.sqlrpgle:

    liblist -d PROFSRVDEV;\

    liblist -d PUIDEV;\

    liblist -al $(LIBLIST);\

    system "CRTSQLRPGI OBJ($(BIN\_LIB)/$\*) SRCSTMF('./src/qrpglesrc/$\*.sqlrpgle') OBJTYPE(\*PGM) REPLACE(\*YES)"

    system "CPYFRMSTMF FROMSTMF('./src/qrpglesrc/$\*.sqlrpgle') TOMBR('/QSYS.lib/$(SRC\_LIB).lib/$(MBR\_PGM).file/$\*.mbr') MBROPT(\*replace)"

%.sql:

    sed -i.bak "s/DTA\_LIB/$(DTA\_LIB)/g" ./$\*.sql

    system "RUNSQLSTM SRCSTMF('./src/$\*.sql')"

all:

    @echo "Build finished!"

**EXECUTE shell script as used in Jenkins**

*#!/QOpenSys/usr/bin/qsh*

gmake -s >> gmakeOutput.txt;

exitCode="$?";

/usr/bin/iconv -t UTF-8 -f 819 gmakeOutput.txt > converted.txt;

/QOpenSys/usr/bin/sh ;

tail -20000 converted.txt | /usr/bin/iconv -t 37 -f 819;

rm gmakeOutput.txt;

rm converted.txt;

**if** test $exitCode -gt 0

**then**

    false

**else**

    true

**fi**