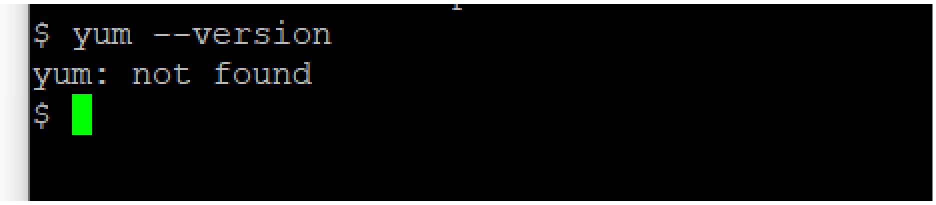
Steps to setup git and Jenkins on IBMi

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.

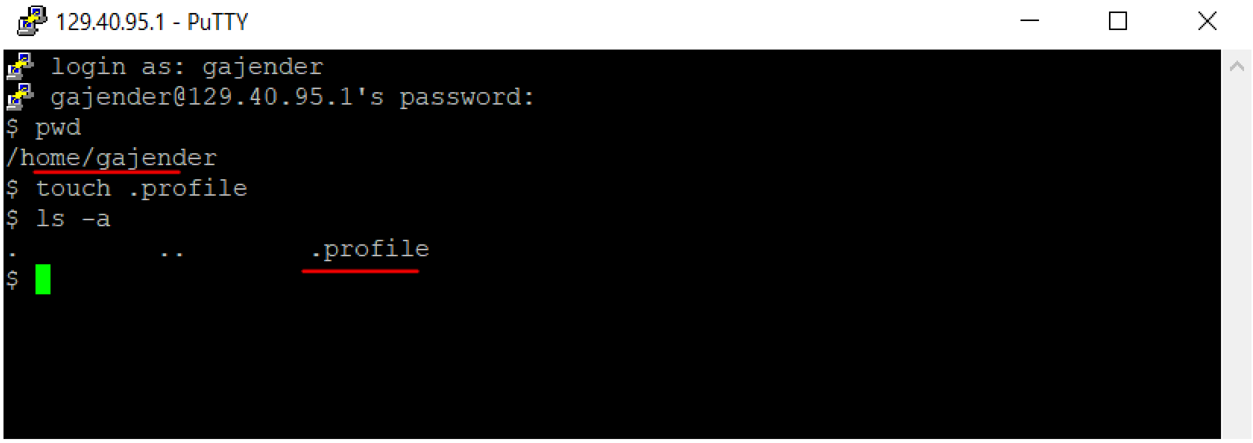
To get started with open source you should have yum package manager in you IBMi

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



```
$ yum --version
yum: not found
$
```

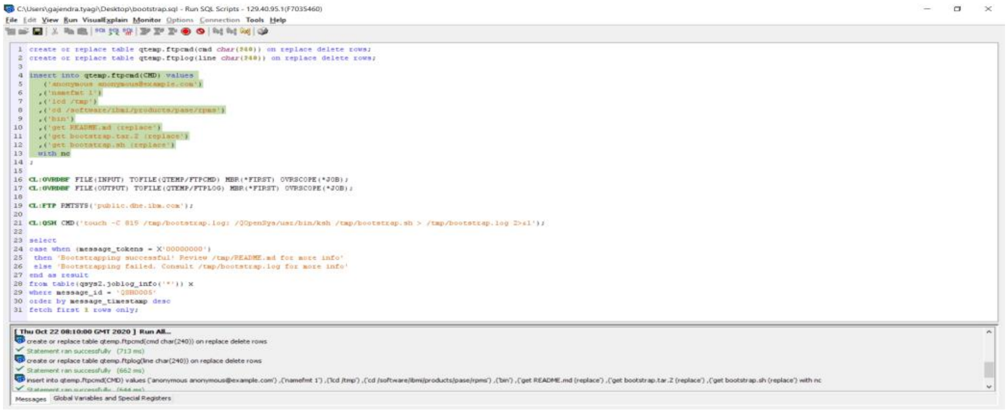
2. **Check if the home directory has a folder with your profile name, if not then create one with `mkdir userProfile` and create a hidden file in it with command `touch .profile` (dot profile).**



```
login as: gajender
gajender@129.40.95.1's password:
$ pwd
/home/gajender
$ touch .profile
$ ls -a
.          ..          .profile
$
```

3. **Use the following link if you don't have yum setup already**
   https://github.com/Programmersio-IBMi/OSSonIBMi

a. Execute bootstap.sql from acs. Even if it shows fail at the end do not panic it ran fine.



C:\Users\gajendra.tyagi\Desktop\bootstrap.sql - Run SQL Scripts - 129.40.95.1(F7035460)

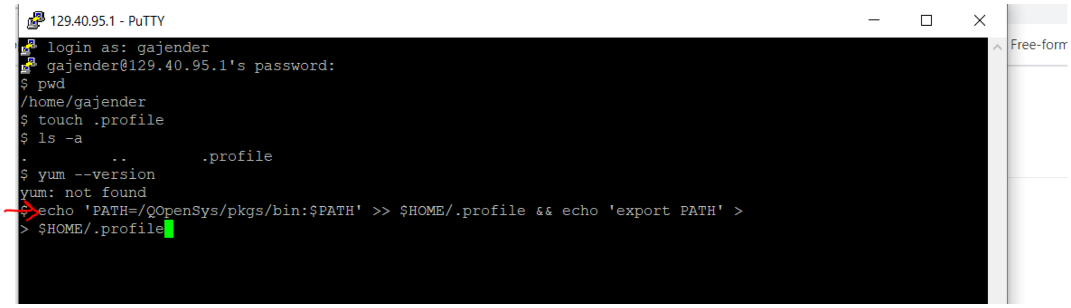
File Edit View Run VisualExplain Monitor Options Connection Tools Help

```
create or replace table qtemp.ftpcmd(cmd char(240)) on replace delete rows;
create or replace table qtemp.ftplog(line char(240)) on replace delete rows;

insert into qtemp.ftpcmd(CMD) values
    ('anonymous anonymous@example.com')
  ,('namefmt 1')
  ,('lcd /tmp')
  ,('cd /software/ibmi/products/pase/rpms')
  ,('bin')
  ,('get README.md (replace')
  ,('get bootstrap.tar.Z (replace')
  ,('get bootstrap.sh (replace')
  with nc
;

CL:OVRDBF FILE(INPUT) TOFILE(QTEMP/FTPCMD) MBR(*FIRST) OVRSCOPE(*JOB);
CL:OVRDBF FILE(OUTPUT) TOFILE(QTEMP/FTPLOG) MBR(*FIRST) OVRSCOPE(*JOB);

CL:FTP RMTSYS('public.dhe.ibm.com');

CL:QSH CMD('touch -C 819 /tmp/bootstrap.log; /QOpenSys/usr/bin/ksh /tmp/bootstrap.sh > /tmp/bootstrap.log 2>&1');

select
case when (message_tokens = X'00000000')
 then 'Bootstrapping successful! Review /tmp/README.md for more info'
 else 'Bootstrapping failed. Consult /tmp/bootstrap.log for more info'
end as result
from table(qsys2.joblog_info('*')) x
where message_id = 'QSH0005'
order by message_timestamp desc
fetch first 1 rows only;
```

```
[ Thu Oct 22 08:10:00 GMT 2020 ] Run All...
create or replace table qtemp.ftpcmd(cmd char(240)) on replace delete rows
Statement ran successfully   (713 ms)
create or replace table qtemp.ftplog(line char(240)) on replace delete rows
Statement ran successfully   (662 ms)
insert into qtemp.ftpcmd(CMD) values ('anonymous anonymous@example.com') ,('namefmt 1') ,('lcd /tmp') ,('cd /software/ibmi/products/pase/rpms') ,('bin') ,('get README.md (replace') ,('get bootstrap.tar.Z (replace') ,('get bootstrap.sh (replace') with nc
Statement ran successfully   (644 ms)
```

b. Run this command after acs has ran.

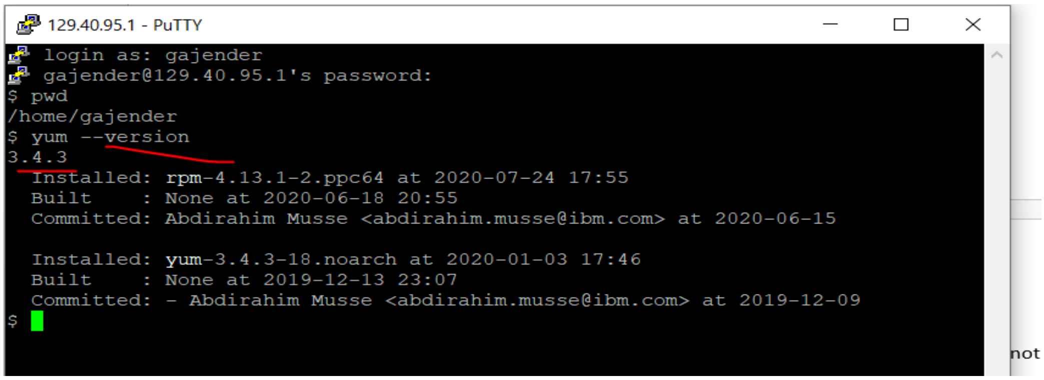
- `echo 'PATH=/QOpenSys/pkgs/bin:$PATH' >> $HOME/.profile && echo 'export PATH' >> $HOME/.profile`



129.40.95.1 - PuTTY

```
login as: gajender
gajender@129.40.95.1's password:
$ pwd
/home/gajender
$ touch .profile
$ ls -a
.           ..          .profile
$ yum --version
yum: not found
$ echo 'PATH=/QOpenSys/pkgs/bin:$PATH' >> $HOME/.profile && echo 'export PATH' >
> $HOME/.profile
```

c.

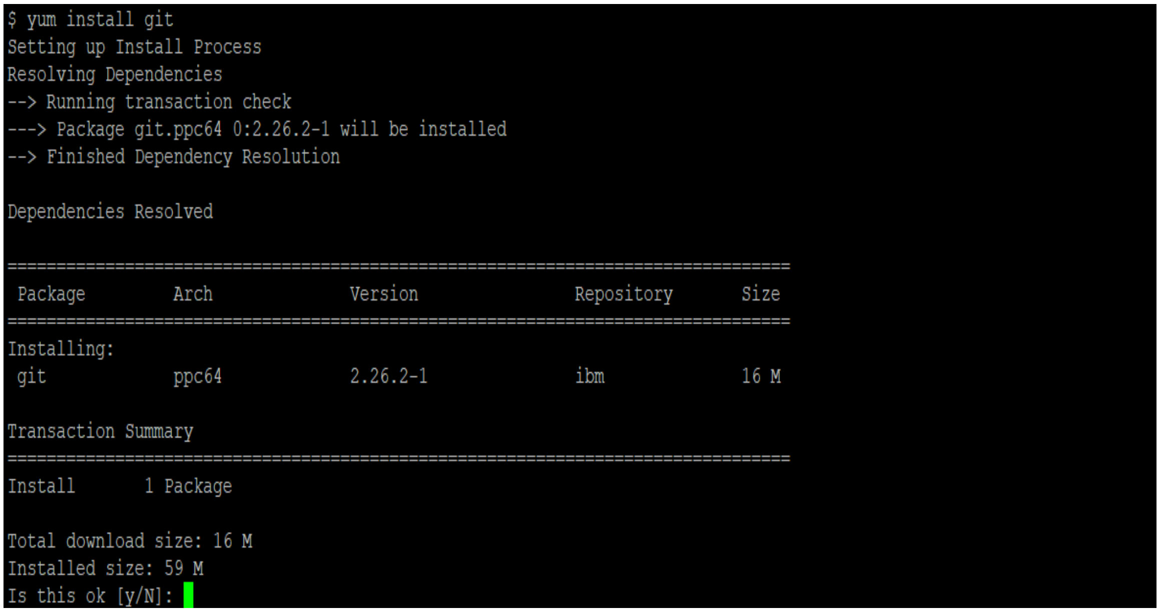


129.40.95.1 - PuTTY

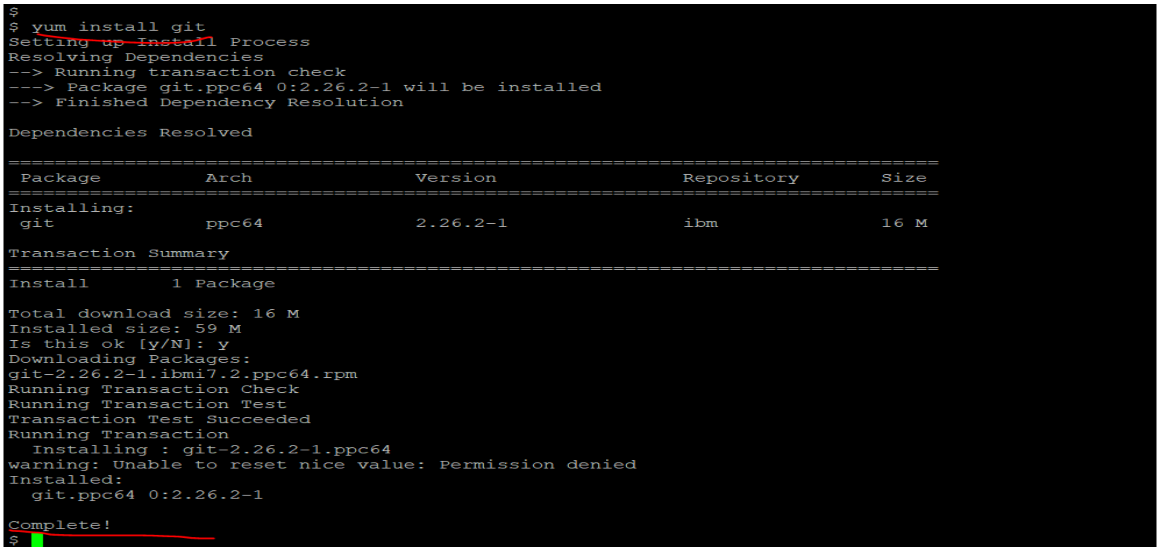
```
login as: gajender
gajender@129.40.95.1's password:
$ pwd
/home/gajender
$ yum --version
3.4.3
  Installed: rpm-4.13.1-2.ppc64 at 2020-07-24 17:55
  Built    : None at 2020-06-18 20:55
  Committed: Abdirahim Musse <abdirahim.musse@ibm.com> at 2020-06-15

  Installed: yum-3.4.3-18.noarch at 2020-01-03 17:46
  Built    : None at 2019-12-13 23:07
  Committed: - Abdirahim Musse <abdirahim.musse@ibm.com> at 2019-12-09
$
```

d. Some YUM commands you must know https://access.redhat.com/sites/default/files/attachments/rh_yum_cheatsheet_1214_jcs_print-1.pdf

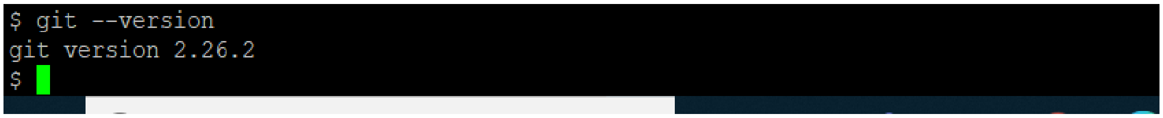
4. Install git using YUM



```
$ yum install git
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package git.ppc64 0:2.26.2-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package         Arch              Version                Repository      Size
================================================================================
Installing:
 git             ppc64             2.26.2-1               ibm             16 M

Transaction Summary
================================================================================
Install       1 Package

Total download size: 16 M
Installed size: 59 M
Is this ok [y/N]:
```



```
$
$ yum install git
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package git.ppc64 0:2.26.2-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package         Arch              Version                Repository      Size
================================================================================
Installing:
 git             ppc64             2.26.2-1               ibm             16 M

Transaction Summary
================================================================================
Install       1 Package

Total download size: 16 M
Installed size: 59 M
Is this ok [y/N]: y
Downloading Packages:
git-2.26.2-1.ibmi7.2.ppc64.rpm
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : git-2.26.2-1.ppc64
warning: Unable to reset nice value: Permission denied
Installed:
  git.ppc64 0:2.26.2-1

Complete!
$
```

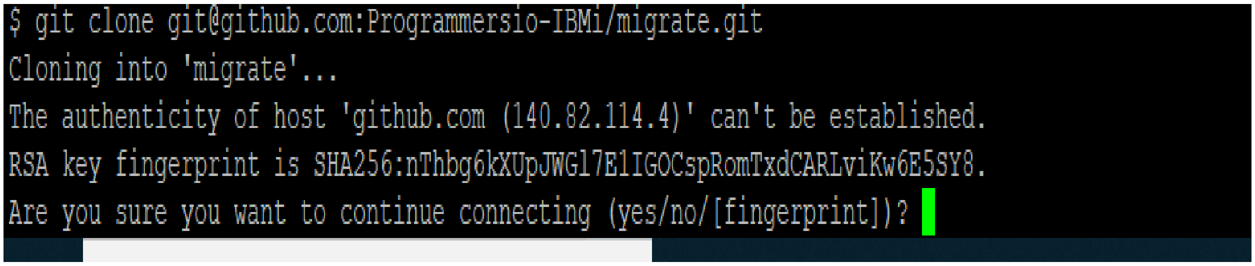


```
$ git --version
git version 2.26.2
$
```

5. **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.

6. **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

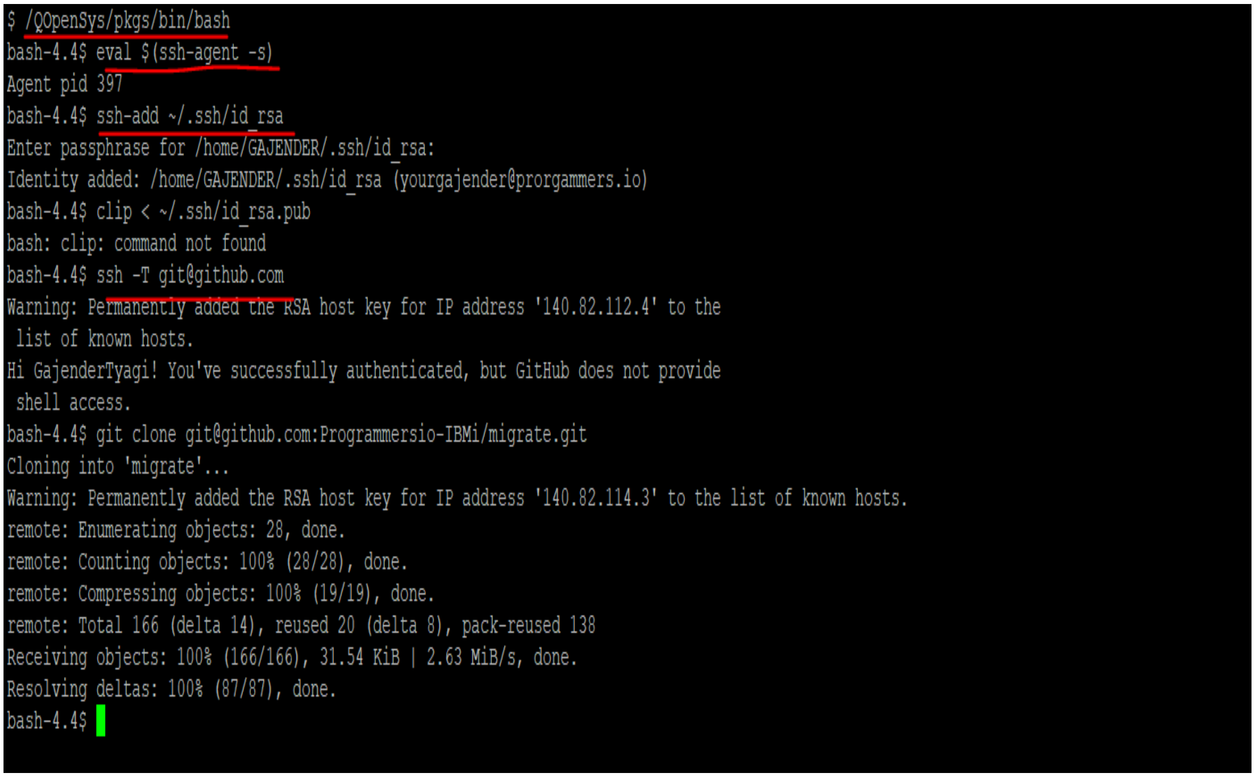
   Without SSH setup you won’t be able to clone remote repositories.



```
$ git clone git@github.com:Programmersio-IBMi/migrate.git
Cloning into 'migrate'...
The authenticity of host 'github.com (140.82.114.4)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

7. **If you are using SSH connection then run this command first :**
   /QOpenSys/pkgs/bin/bash

8. **If you have followed all steps properly then you are good to go.**



```
$ /QOpenSys/pkgs/bin/bash
bash-4.4$ eval $(ssh-agent -s)
Agent pid 397
bash-4.4$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/GAJENDER/.ssh/id_rsa:
Identity added: /home/GAJENDER/.ssh/id_rsa (yourgajender@prorgammers.io)
bash-4.4$ clip < ~/.ssh/id_rsa.pub
bash: clip: command not found
bash-4.4$ ssh -T git@github.com
Warning: Permanently added the RSA host key for IP address '140.82.112.4' to the
 list of known hosts.
Hi GajenderTyagi! You've successfully authenticated, but GitHub does not provide
 shell access.
bash-4.4$ git clone git@github.com:Programmersio-IBMi/migrate.git
Cloning into 'migrate'...
Warning: Permanently added the RSA host key for IP address '140.82.114.3' to the list of known hosts.
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 166 (delta 14), reused 20 (delta 8), pack-reused 138
Receiving objects: 100% (166/166), 31.54 KiB | 2.63 MiB/s, done.
Resolving deltas: 100% (87/87), done.
bash-4.4$
```

9. **Setup Jenkins, reference link** : https://github.com/worksofliam/blog/issues/43

10. `` `wget http://mirrors.jenkins.io/war-stable/latest/jenkins.war` ``

11. **Once Jenkins is installed its time to run it.**

```
java -jar jenkins.war --httpPort=9090
```

12. **Use above command to run initially and do one time setup.**

13. **Complete the initial setup.**

14. **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

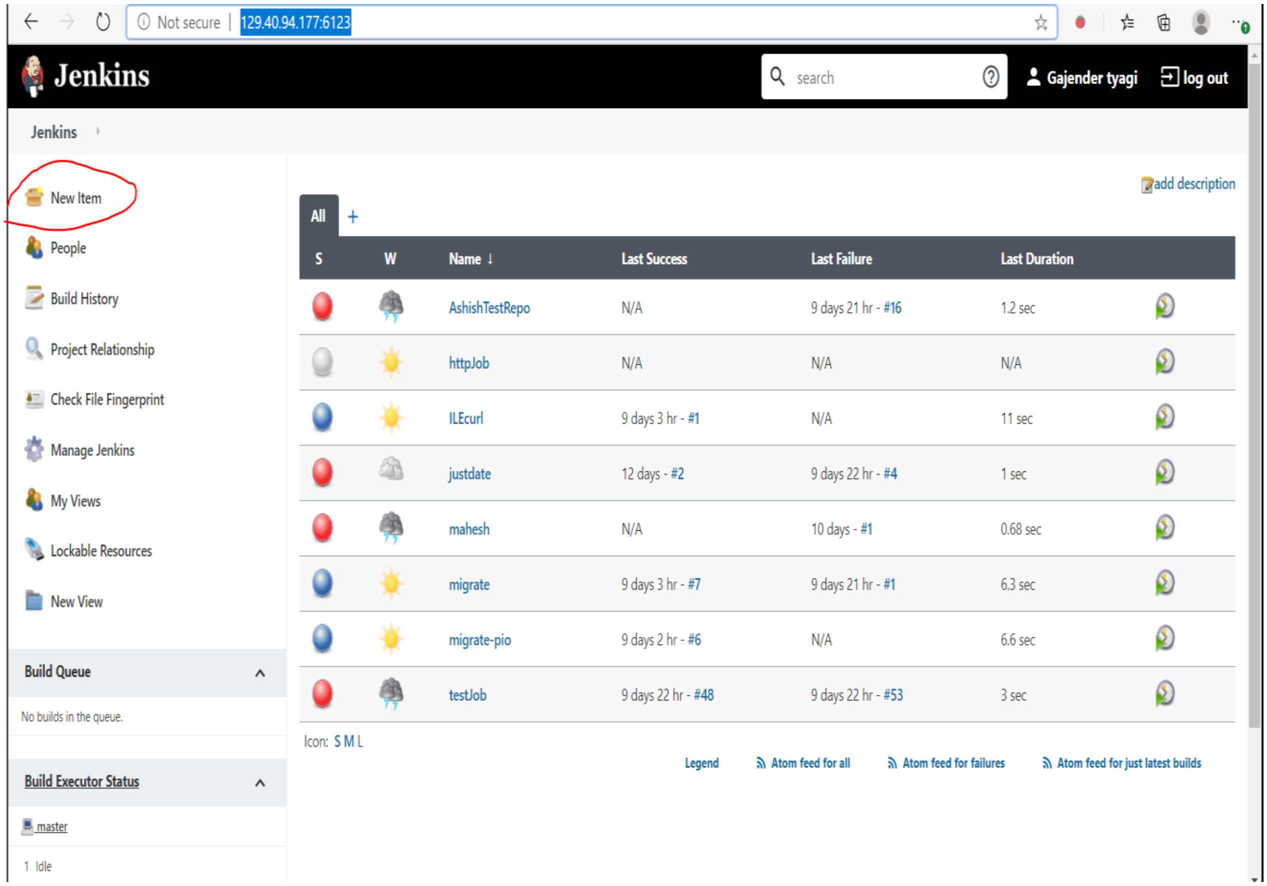
    b. Create a shell file and paste the below content

```
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')

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



No builds in the queue.

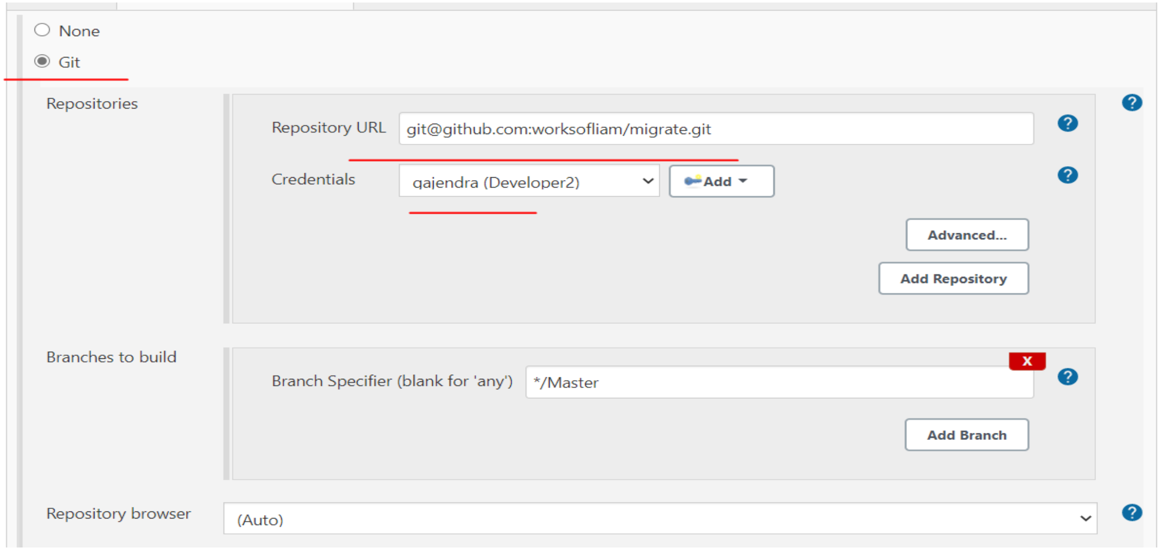
Build Executor Status

master

1 Idle

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☁	AshishTestRepo	N/A	9 days 21 hr - #16	1.2 sec
○	☀	httpJob	N/A	N/A	N/A
●	☀	ILEcurl	9 days 3 hr - #1	N/A	11 sec
●	☁	justdate	12 days - #2	9 days 22 hr - #4	1 sec
●	☁	mahesh	N/A	10 days - #1	0.68 sec
●	☀	migrate	9 days 3 hr - #7	9 days 21 hr - #1	6.3 sec
●	☀	migrate-pio	9 days 2 hr - #6	N/A	6.6 sec
●	☁	testJob	9 days 22 hr - #48	9 days 22 hr - #53	3 sec

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



○ None

● Git

Repositories

Repository URL: git@github.com:worksofliam/migrate.git

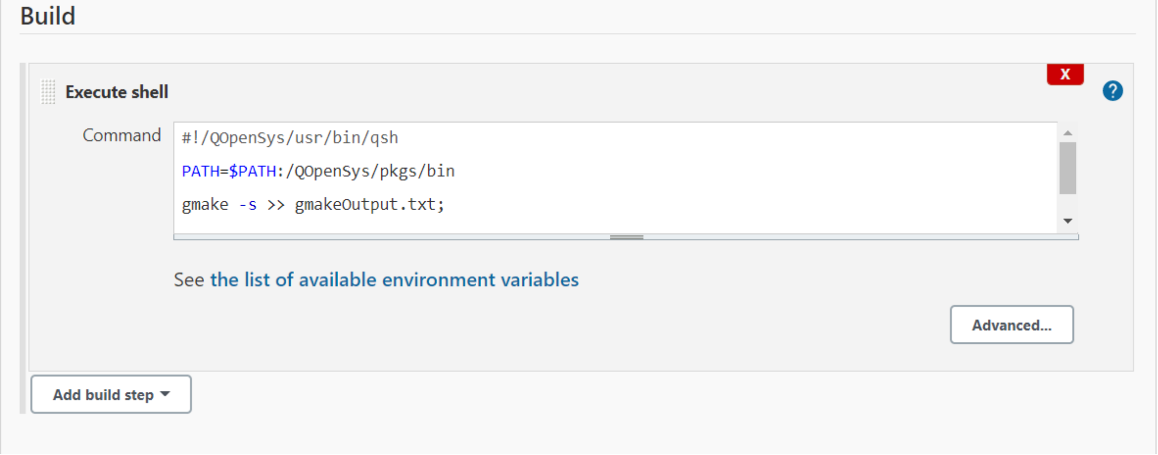
Credentials: gajendra (Developer2)

Branches to build

Branch Specifier (blank for 'any'): */Master

Repository browser: (Auto)

17.

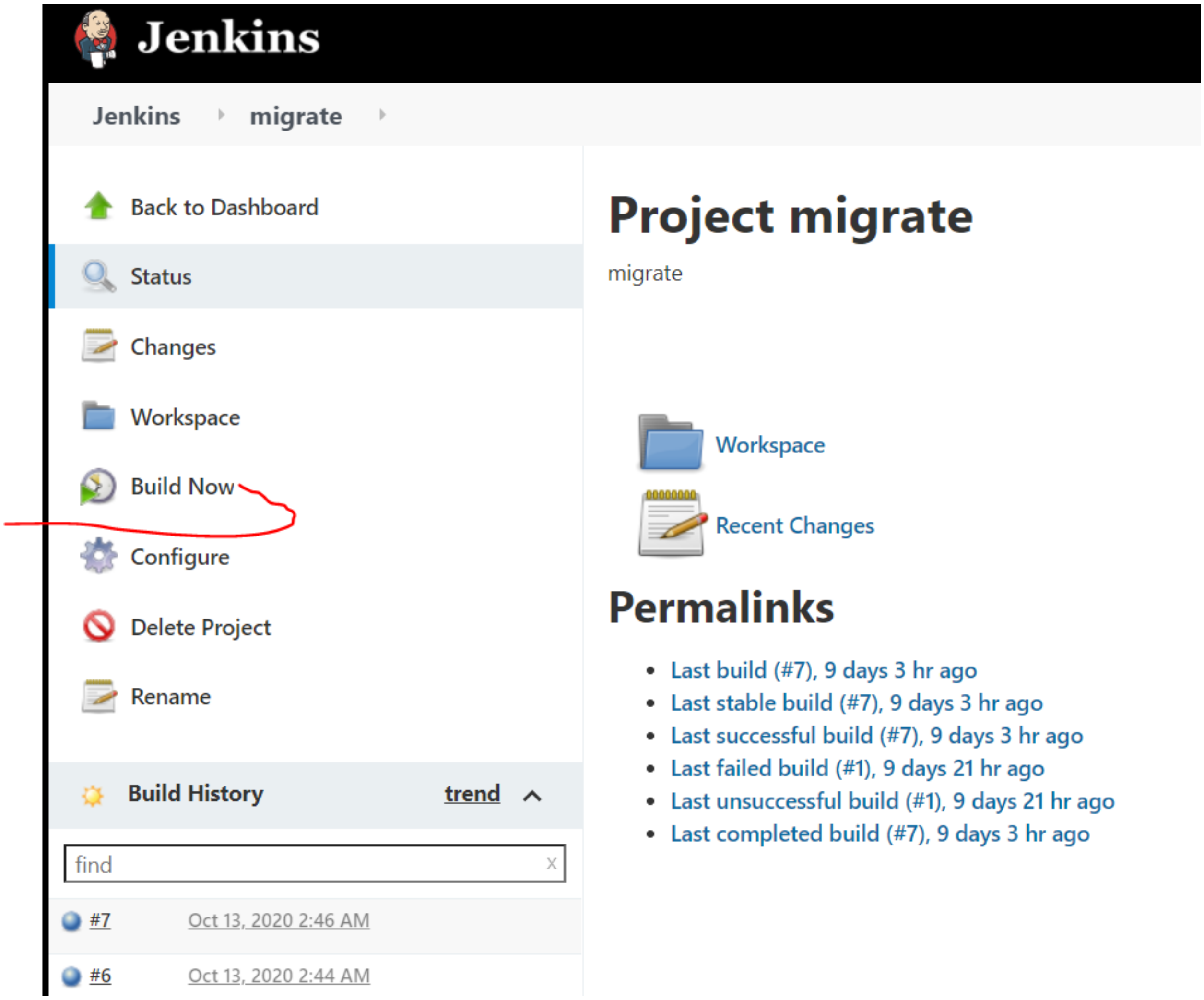


Command

```
#!/QOpenSys/usr/bin/qsh
PATH=$PATH:/QOpenSys/pkgs/bin
gmake -s >> gmakeOutput.txt;
```

See the list of available environment variables

18.



Project migrate

migrate

Permalinks

- Last build (#7), 9 days 3 hr ago
- Last stable build (#7), 9 days 3 hr ago
- Last successful build (#7), 9 days 3 hr ago
- Last failed build (#1), 9 days 21 hr ago
- Last unsuccessful build (#1), 9 days 21 hr ago
- Last completed build (#7), 9 days 3 hr ago

#7 Oct 13, 2020 2:46 AM

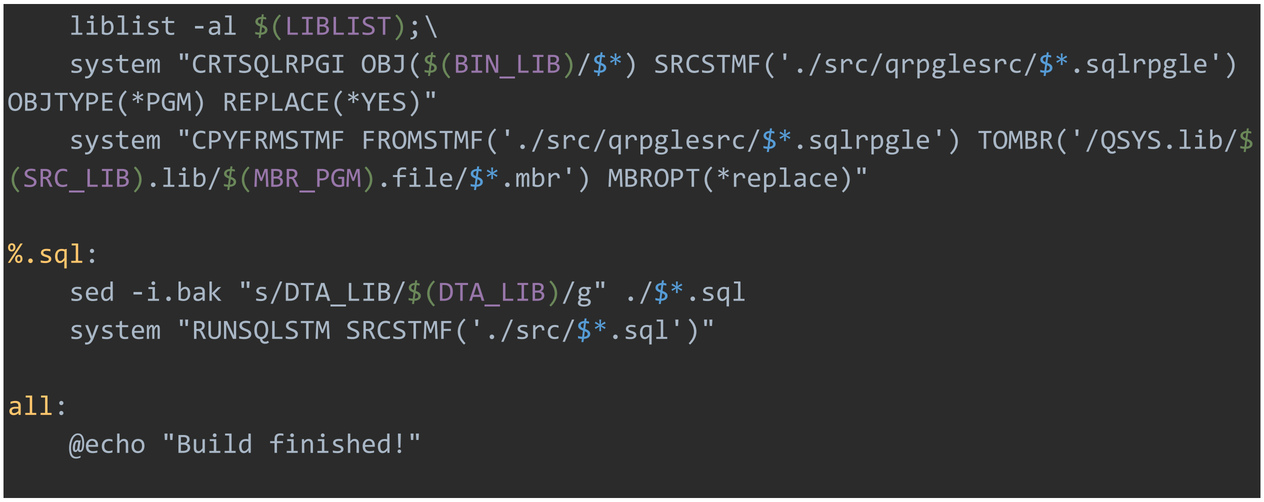
#6 Oct 13, 2020 2:44 AM

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

A standard MAKEFILE

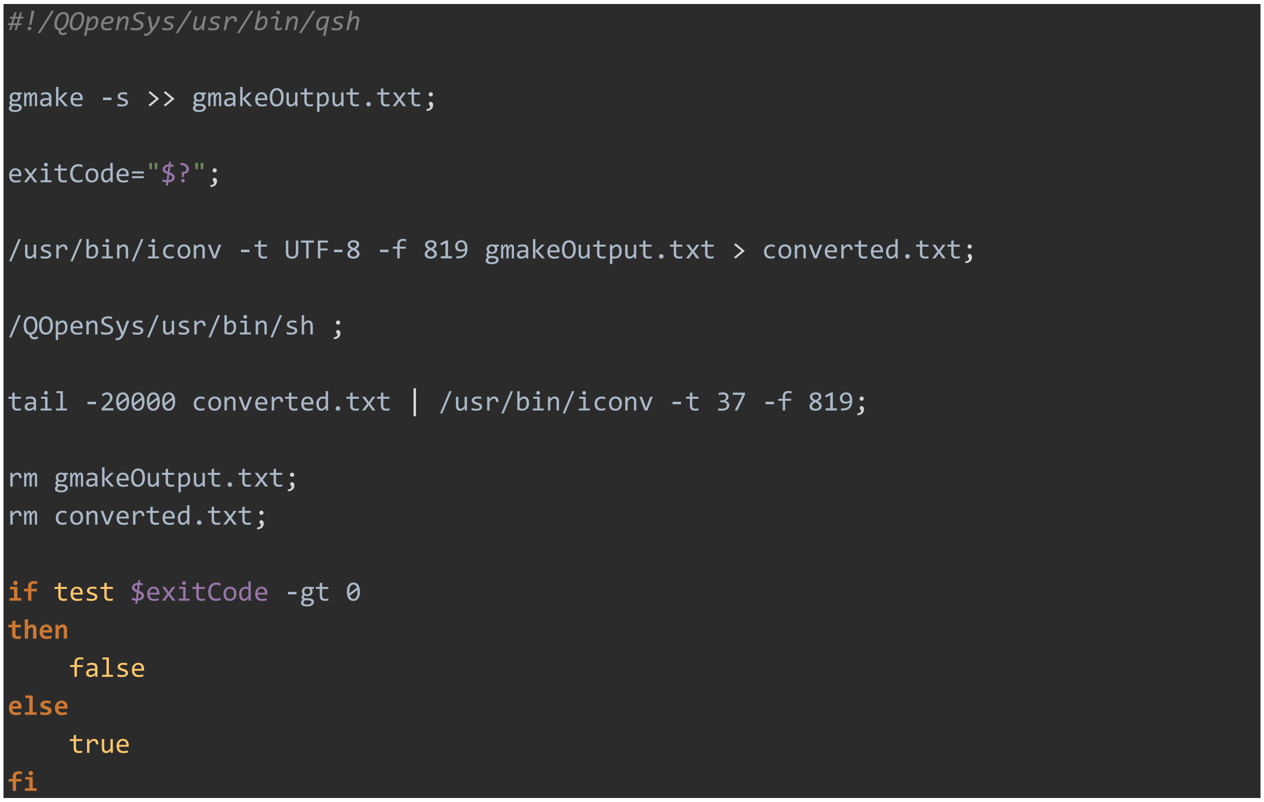
```
SMY_ENV=DEV

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
```

Repo Path: https://public.dhe.ibm.com/software/ibmi/products/pase/rpms/