2017

Gerrit Code Review

DOCUMENT HISTORY

Created - 06/15/2016 - KLohar, CI / CD Team

Updated - 07/12/2016 – KLohar, CI-CD Team

Updated - 01/06/2017 – KLohar, CI-CD Team

Updated – 6/30/2017 – vchinth, CI-CD Team

Updated – 8/23/2017 – sindira1, CI-CD Team

Contents

DOCUMENT HISTORY 2

Gerrit Code Review Pre-requisites (setting up Gerrit) 3

First install below mentioned two software 3

1. Git (v2.6.4 or higher) -- https://git-scm.com/downloads 3
2. Tortoise Git -- https://tortoisegit.org/download/ 3

Setting up SSH Keys & Environment variable 3

3. Generate SSH keys – using Git Bash application 3
4. Add key (contents of `id_rsa.pub`) to settings → SSH Public keys on Gerrit 5
5. Add below mentioned environment variable to Local Laptop/Desktop 5
6. Add a file named – config inside .ssh dir (which was created during above step) 5

7. Please run below 1-time two commands on Git Bash prompt, to set global variables… there will be no output for those commands 5
8. Test ssh connection to PROD Gerrit Server using below command 6
9. set up system wide variable – for using Git Tortoise 6
SSH URL from Gerrit 8
Clone Repository using Tortoise Git 9
10. Just got to your local Git directory, copy the ssh url from Gerrit 9
Cloning using command line 10

Gerrit Code Review Pre-requisites (setting up Gerrit)

First install below mentioned two software.

1. Git (v2.6.4 or higher) -- https://git-scm.com/downloads
2. Tortoise Git -- https://tortoisegit.org/download/

Setting up SSH Keys & Environment variable

3. Generate SSH keys – using Git Bash application

```
KLohar@BLRKEC94578L MINGW64 ~/.ssh
$ cd
$ pwd
/c/Users/KLohar
```

Commands:

```
mkdir .ssh
```

```
cd .ssh
ssh-keygen -t rsa
```

You will need to provide inputs here
Don't type anything Just hit enter for all the below questions

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/vchinth/.ssh/id_rsa): |
```

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

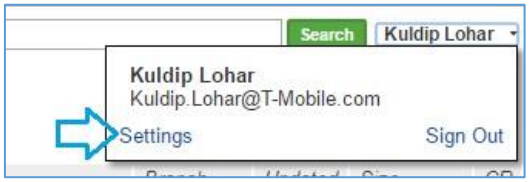
```
KLohar@BLRKEC94578L MINGW64 ~/.ssh
$ ls -ltr total
31
-rw-r--r-- 1 KLohar 1049089  401 Jun 13 11:45 id_rsa.pub
-rw-r--r-- 1 KLohar 1049089 1679 Jun 13 11:45 id_rsa

KLohar@BLRKEC94578L MINGW64 ~/.ssh
```

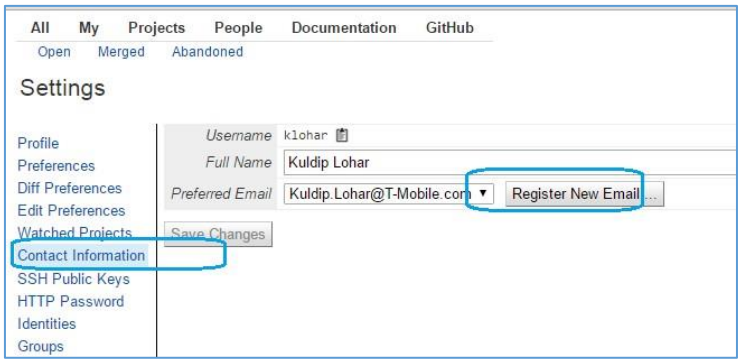
1) `id_rsa.pub` → key file; copy content to a notepad

2) Log on to Gerrit - https://prdcidb001.unix.gsm1900.org:8443

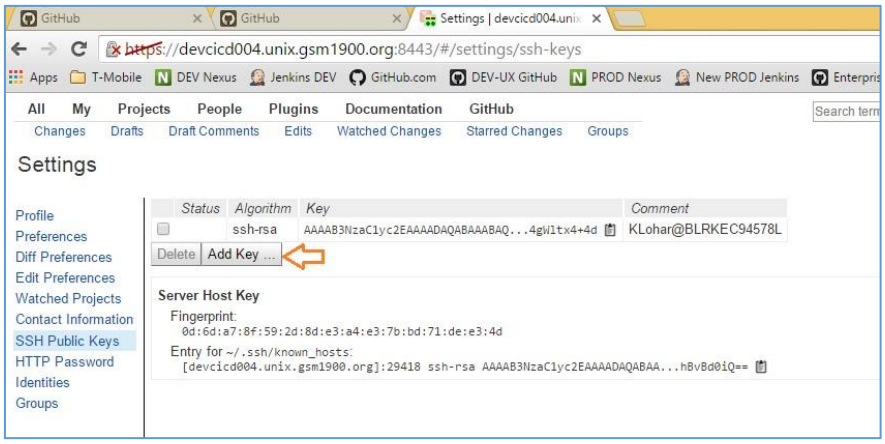
   Click: Authorize GitHub profile access

3) After first login – please register your email address





4. Add key (contents of `id_rsa.pub`) to settings → SSH Public keys on Gerrit



5. Add below mentioned environment variable to Local Laptop/Desktop

System → Advanced System Settings → Environment Variables → System Variables → New

Variable Name: GIT_SSH_COMMAND

Variable Value: ssh -o KexAlgorithms=+diffie-hellman-group1-sha1

6. Add a file named – config inside .ssh dir (which was created during above step)

Add below contents to config file (Type **vi config** in git bash)

Please copy the same as shown below in 2 separate lines

```
Host prdcidb001.unix.gsm1900.org
    KexAlgorithms +diffie-hellman-group1-sha1
```

*For MAC Users:

Mac acts like a linux box so they can set up environment variable for "GIT_SSH_COMMAND" by adding ~/.profile and exporting the variable and its value there. Once it is added .profile should be sourced.

p.s. config folder inside .ssh is not needed for mac.

7. Please run below 1-time two commands on Git Bash prompt, to set global variables… there will be no output for those commands

`git config --global user.name 'klohar'` – please use your NT-ID `git config --global user.email 'Kuldip.Lohar@T-Mobile.com'` – please use your NT-ID

```
KLOHAR@USABOS133784L MINGW64 ~
$ git config --global user.name 'klohar'

KLOHAR@USABOS133784L MINGW64 ~
$ git config --global user.email 'Kuldip.Lohar@T-Mobile.com'

KLOHAR@USABOS133784L MINGW64 ~
$
```

8. Test ssh connection to PROD Gerrit Server using below command

```
ssh -T klohar@prdcidb001.unix.gsm1900.org -p 29418
```

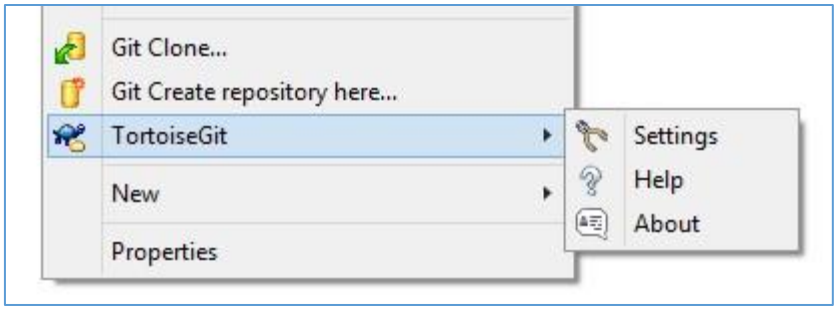
Note username in the above command is case sensitive

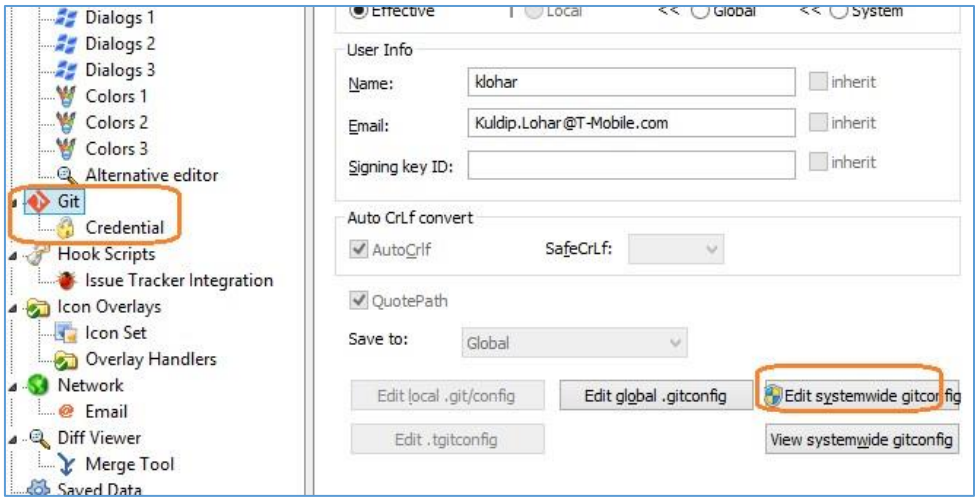
you will see the message as like below

```
KLohar@BLRKEC94578L MINGW64 ~/.ssh (master)
$ ssh -T klohar@prdcidb001.unix.gsm1900.org -p 29418

  ****    Welcome to Gerrit Code Review    ****

  Hi Kuldip Lohar, you have successfully connected over SSH.

  Unfortunately, interactive shells are disabled.
  To clone a hosted Git repository, use:

  git clone ssh://klohar@prdcidb001.unix.gsm1900.org:29418/REPOSITORY_NAME.git


KLohar@BLRKEC94578L MINGW64 ~/.ssh (master)
```

9. set up system wide variable – for using Git Tortoise

TortoiseGit → Settings → Git





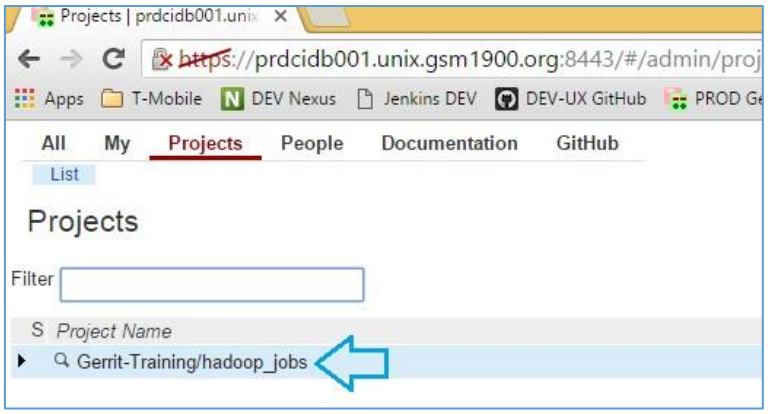
Please add below line to **Config** file

sslverify = false

```
[color]
  diff = auto
  status = auto
  branch = auto
  interactive = true
[pack]
[help]
  format = html
[http]
  sslCAInfo = C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
  sslverify = false
[diff "astextplain"]
  textconv = astextplain
[rebase]
  autosquash = true
```

SSH URL from Gerrit

Projects → List → select your Repository to clone

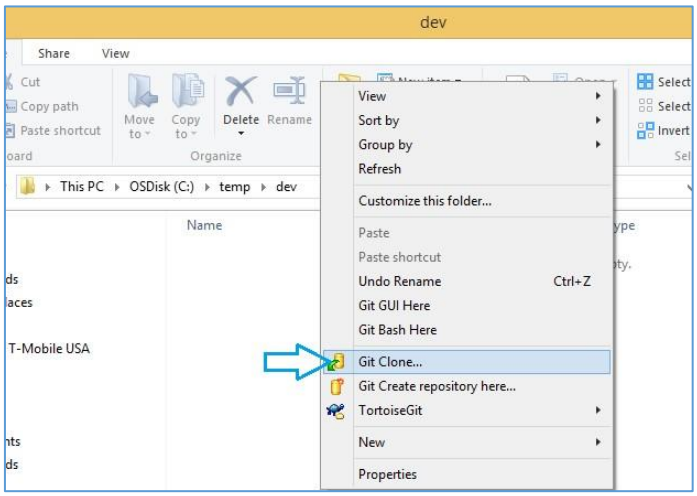


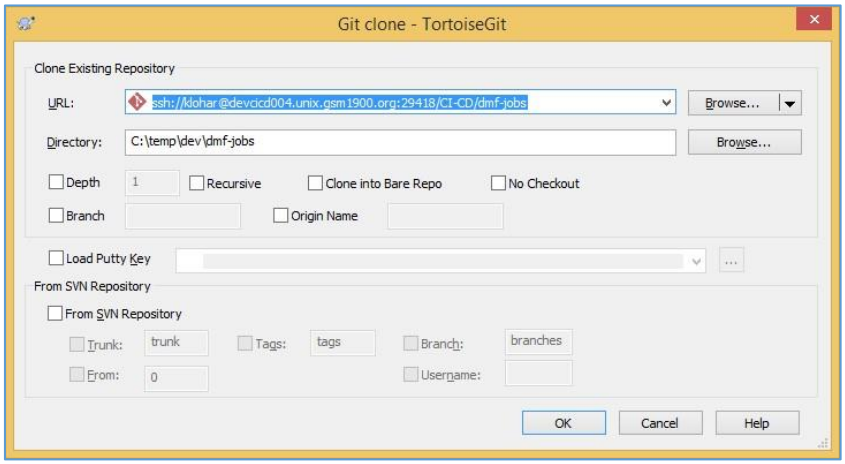
Preferred to use "clone with ssh" option to copy URL, which will be used to Clone the Repository.

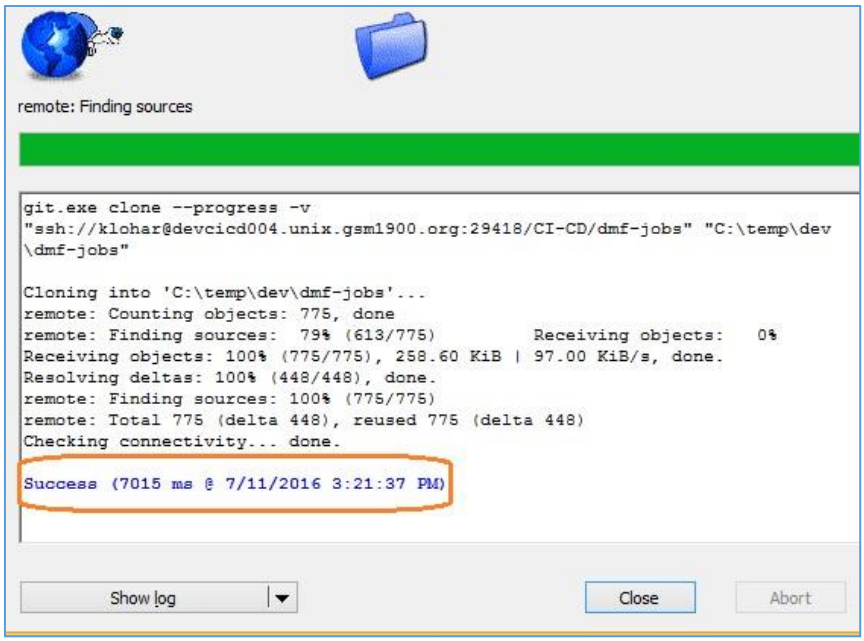
Clone Repository using Tortoise Git

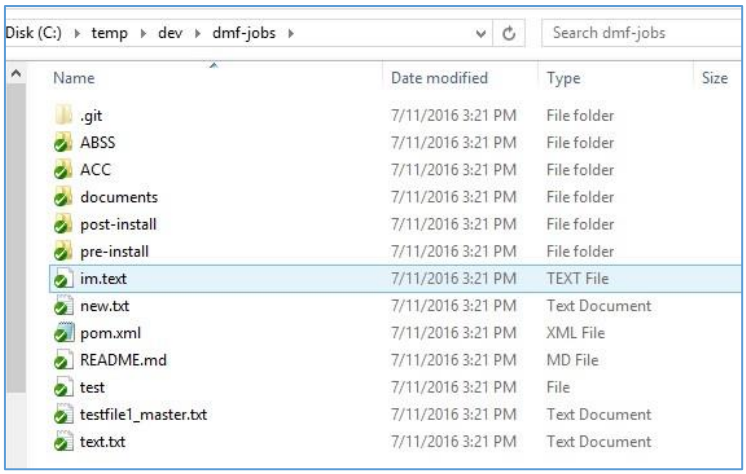
10. Just got to your local Git directory, copy the ssh url from Gerrit

Right Click → Git Clone



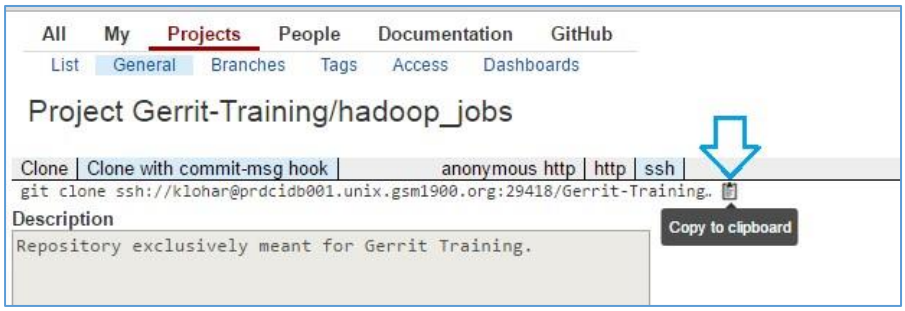






Cloning using command line

Same as like GIT commands, once you clone Repository… you will see screen like below




```
KLohar@BLRKEC94578L MINGW64 /c/temp/klohar
$ git clone ssh://klohar@prdcidb001.unix.gsm1900.org:29418/Gerrit-Training/hadoop_jobs && scp -p -P 29418 klohar@prdcidb001.u
nix.gsm1900.org:hooks/commit-msg hadoop_jobs/.git/hooks/
Cloning into 'hadoop_jobs'...
remote: Counting objects: 2472, done
remote: Finding sources: 100% (2472/2472)
remote: Total 2472 (delta 1978), reused 2470 (delta 1978)
Receiving objects: 100% (2472/2472), 4.33 MiB | 164.00 KiB/s, done.
Resolving deltas: 100% (1978/1978), done.
Checking connectivity... done.
Checking out files: 100% (4260/4260), done.
commit-msg                                                    100% 4662     4.6KB/s   00:00

KLohar@BLRKEC94578L MINGW64 /c/temp/klohar
```

```
KLohar@BLRKEC94578L MINGW64 /c/temp/klohar
$ cd hadoop_jobs/

KLohar@BLRKEC94578L MINGW64 /c/temp/klohar/hadoop_jobs (master)
$ ls -ltr
total 37
-rw-r--r-- 1 KLohar 1049089 95 Jul 12 16:00 README.md
drwxr-xr-x 1 KLohar 1049089  0 Jul 12 16:00 configs/
drwxr-xr-x 1 KLohar 1049089  0 Jul 12 16:00 ddl/
drwxr-xr-x 1 KLohar 1049089  0 Jul 12 16:00 documents/
drwxr-xr-x 1 KLohar 1049089  0 Jul 12 16:00 jobs/
drwxr-xr-x 1 KLohar 1049089  0 Jul 12 16:00 scheduler/
drwxr-xr-x 1 KLohar 1049089  0 Jul 12 16:00 views/
drwxr-xr-x 1 KLohar 1049089  0 Jul 12 16:00 scripts/

KLohar@BLRKEC94578L MINGW64 /c/temp/klohar/hadoop_jobs (master)
```