Gitlab usage

1. 安裝 git

```
root@ruby-desktop:/home/ruby# apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra libslp1
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  git-man liberror-perl
```

2. 設定 username & email

```
root@ruby-desktop:/home/ruby#
root@ruby-desktop:/home/ruby# git config --global user.name "Ruby Liu"
```

```
root@ruby-desktop:/home/ruby#
root@ruby-desktop:/home/ruby# git config --global user.email "Ruby_Liu@asrockrack.com"
```

```
root@ruby-desktop:/home/ruby#
root@ruby-desktop:/home/ruby# git config --list
user.name=Ruby Liu
user.email=Ruby_Liu@asrockrack.com
```

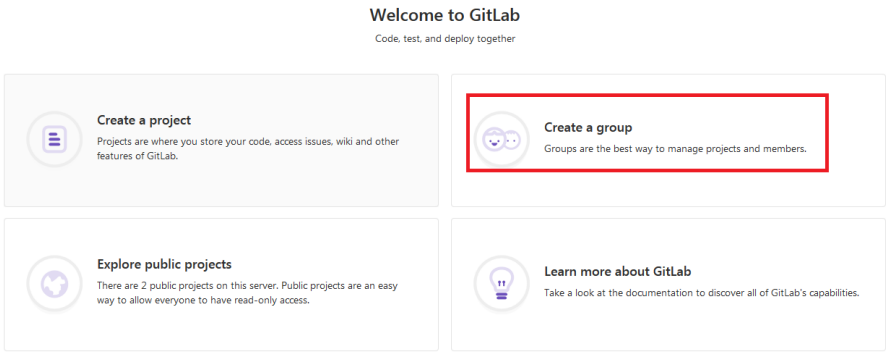
3. Create group & project

Gitlab: http://192.168.36.8/

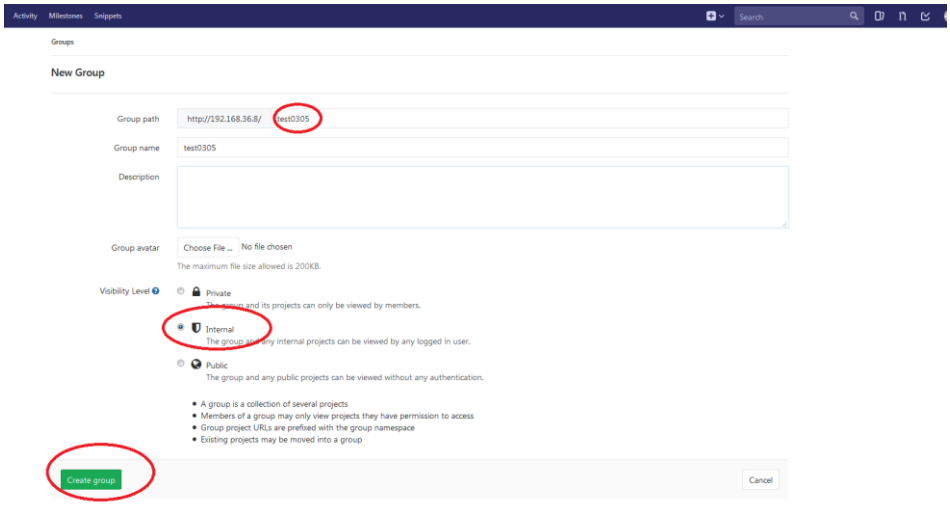
目錄架構如下:

- Channel projects: BMC/Chip/platforms/平台名稱/案子名稱
  Ex. BMC/AST2400/platforms/grantley/EP2C612D16HM/
- ODM projects: /BMC/Chip/customers/客戶名稱/案子名稱
  Ex. BMC/AST2500/customers/Intel/CRB-PwP/

因為是在遠端儲存,所以要在 server 上建 project,可以集中放在同一個資料夾，要建資料夾就要選取 ***Create group***

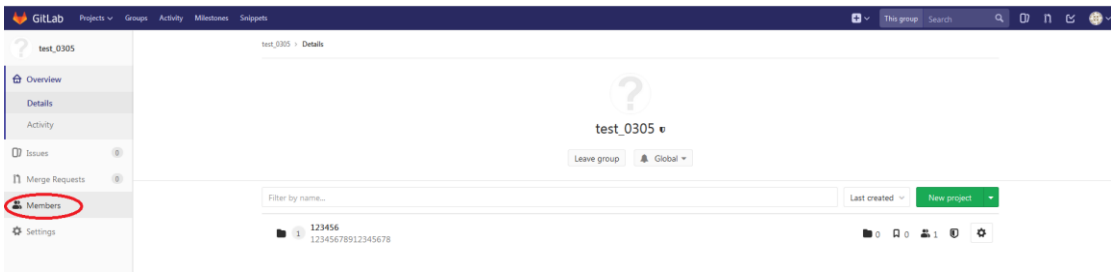


填資料夾名稱與 Visibility level(要選 Internal)

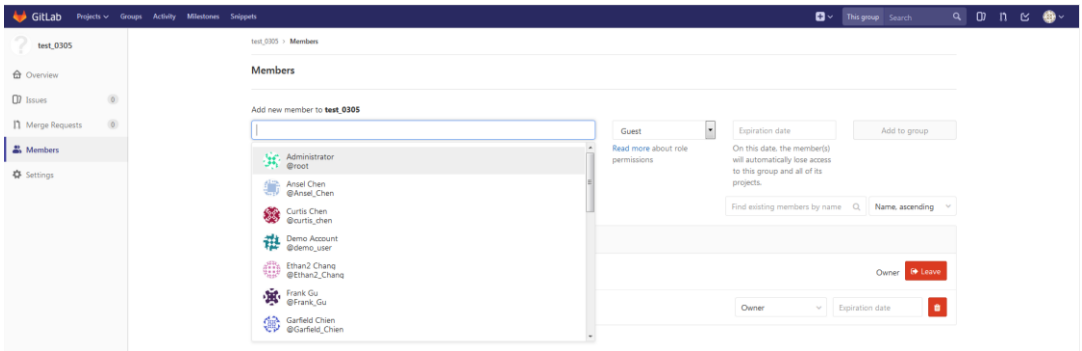


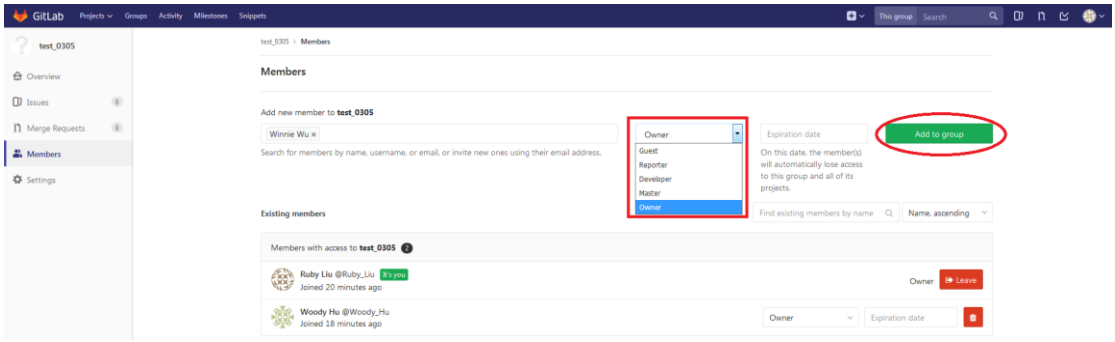
進入資料夾後需要將其他人加入 menbers,其他人才能在資料夾下建 project

點進去資料夾,選取右側的 Members

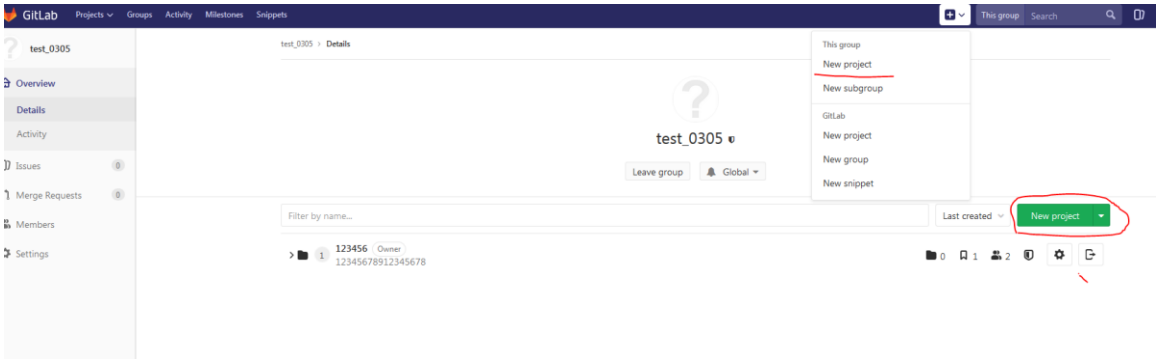


選取要加入的成員，並設定權限(Owner 可以建立 sub folder(group) & project ,Master 可以建 project)

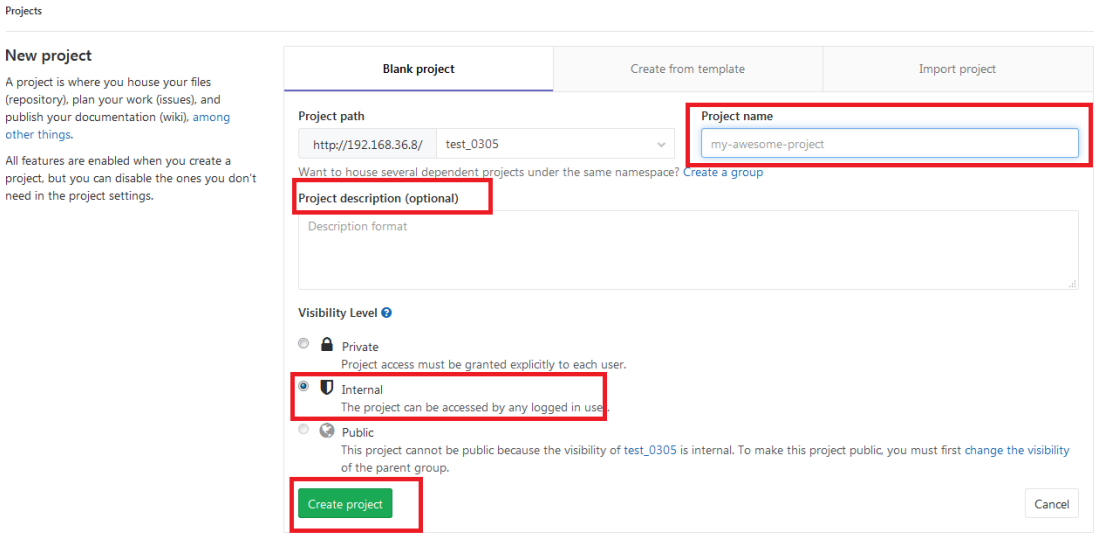




當你成為別的資料夾的 member 後 就可以建 project,點選該 group,就可以看到新增 project 的選項



點選後，填入 project name & description & visibility level(Internal), 就可建 project.



4. 把 local 端的 project(package)上傳(第一次上傳,全部的 package)
到 local 端電腦上的資料夾內輸入"***git init***", 會多一個叫.git 的資料夾

```
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00#
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git init
Initialized empty Git repository in /home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00/.git/
```

然後用 ***git add .*** 來加入所有的檔案(也可以一個一個檔案下去加 但是東西太

多還是就先這樣吧),這個動作是要把檔案加入 stage. 加入 stage 才能 commit & push, 另因為是遠端的 repository,所以要加入遠端 URL，位置如下圖.

,輸入: *git remote add <url name> <URL>* ,url name 就是這個 URL 的代名詞 (可自由更換,例如本篇為 4u100_test,從遠端 clone 時預設 url name 為 origin)

```
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git add .
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git remote add 4u100_test http://192
.168.36.8/test_0305/123456/test_project_0307.git
```

再來就是 commit, 因為是第一次上傳, 輸入*"git commit –m <commit message>"*, -m 後面應該是接這次 commit 的名稱. 如果沒打-m, 就會跳出一個編輯的畫面要你填 message(本篇先不談), "initial commit"就是這次 commit 的 description,故可視狀況自行修改,例如"4u100_l0.07.0"

```
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git commit -m "Initial connmmit"
[master (root-commit) bc6689a] Initial commit
 572 files changed, 3980 insertions(+)
 create mode 100755 configs/.MLK.PRJ.swp
 create mode 100755 configs/.svn/entries
 create mode 100755 configs/.svn/prop-base/ast2400evb.MAP.svn-base
 create mode 100755 configs/.svn/text-base/ast2400evb.MAP.svn-base
 create mode 100755 configs/.svn/text-base/ast2400evb.PRJ.svn-base
 create mode 100755 configs/.svn/text-base/ast2400evb_ker_3.MAP.svn-base
```

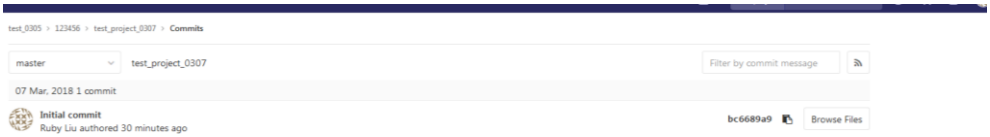
Commit 結束後，就是用 push 去真正上傳檔案到 server, 輸入*"git push –u <url name> master"*,會叫你輸入帳密,然後就開始上傳.

```
fatal: Authentication failed for 'http://192.168.36.8/test_0305/123456/test_project_0307.git/'
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git push -u 4u100_test master
Username for 'http://192.168.36.8': ruby_liu@asrockrack.com
Password for 'http://ruby_liu@asrockrack.com@192.168.36.8':
Counting objects: 579, done.
Delta compression using up to 4 threads.
Compressing objects:  98% (567/578)
```

傳完的結果:

```
Counting objects: 579, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (578/578), done.
Writing objects: 100% (579/579), 1.05 GiB | 32.29 MiB/s, done.
Total 579 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To http://192.168.36.8/test_0305/123456/test_project_0307.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from 4u100_test.
```

上面的 command 是將 package 傳到了 project 內叫做 master 的 branch, branch 詳細內容這邊也先不談,完成後就可在 server 上找到下面的 commit 紀錄,也就是完成了第一次上傳.



5. 修改 package 後,第 n 次上傳

假設這一次更新只有某幾包 package 被 repack, 輸入 *"git status"*可以看到如下面圖片內容,顯示了幾個 modified package,這個時候 commit 會被駁回,要先 add 這些 package 到 stage

用 git add 加入後,再用 git status 可以看到 modified 的部分已經可以 commit

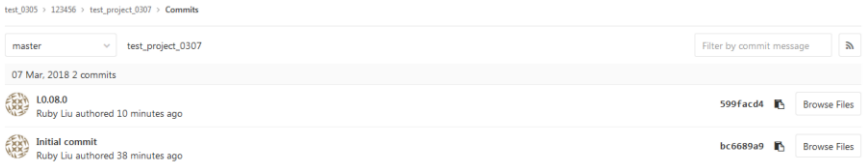
```
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git add packages/libipmimsghndlr-3.8
1.0-src.spx
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git add packages/libipmipar_ast2400e
vb_4u100-1.0.0-ARM-AST2400-AST2400EVB-AMI-src.spx
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git add packages/libmodhapi-2.28.0-s
rc.spx
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git status
On branch master
Your branch is up-to-date with '4u100_test/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   packages/libipmimsghndlr-3.81.0-src.spx
        modified:   packages/libipmipar_ast2400evb_4u100-1.0.0-ARM-AST2400-AST2400EVB-AMI-src.spx
        modified:   packages/libmodhapi-2.28.0-src.spx
```

Commit 並 push

```
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git commit -m "L0.08.0"
[master 599facd] L0.08.0
 3 files changed, 0 insertions(+), 0 deletions(-)
 rewrite packages/libipmimsghndlr-3.81.0-src.spx (95%)
 rewrite packages/libipmipar_ast2400evb_4u100-1.0.0-ARM-AST2400-AST2400EVB-AMI-src.spx (92%)
 rewrite packages/libmodhapi-2.28.0-src.spx (98%)
root@ruby-desktop:/home/ruby/FilesIn1T/home/ruby_m2/code_379/4U100_L0.07.00# git push -u 4u100_test master
Username for 'http://192.168.36.8': ruby_liu@asrockrack.com
Password for 'http://ruby_liu@asrockrack.com@192.168.36.8':
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 767.72 KiB | 0 bytes/s, done.
Total 6 (delta 1), reused 0 (delta 0)
To http://192.168.36.8/test_0305/123456/test_project_0307.git
   bc6689a..599facd  master -> master
Branch master set up to track remote branch master from 4u100_test.
```

就可以看到新的 commit 在上面了

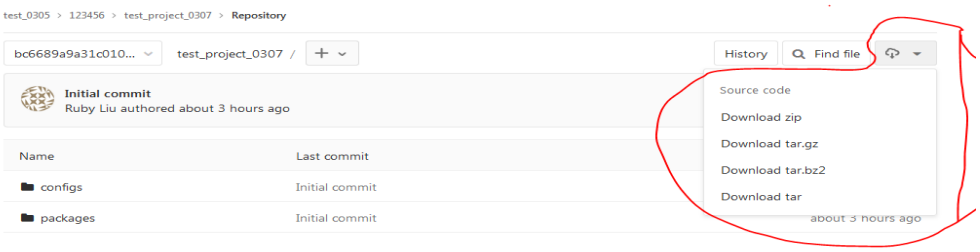


6. 下載 project,

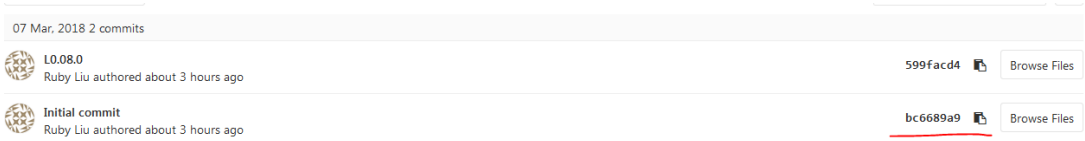
下載最新版: *"git clone <URL> <new folder>"*,

```
root@ruby-desktop:/home/ruby/FilesInIT/home/ruby_m2/code_379# git clone http://192.168.36.8/test_0305/123456/test_project_0307.git 4U100_L0.08.0_CLONE
Cloning into '4U100_L0.08.0_CLONE'...
Username for 'http://192.168.36.8': ruby_liu@asrockrack.com
Password for 'http://ruby_liu@asrockrack.com@192.168.36.8':
remote: Counting objects: 585, done.
remote: Compressing objects: 100% (581/581), done.
remote: Total 585 (delta 4), reused 578 (delta 3)
Receiving objects: 100% (585/585), 1.05 GiB | 109.50 MiB/s, done.
Resolving deltas: 100% (4/4), done.
Checking connectivity... done.
```

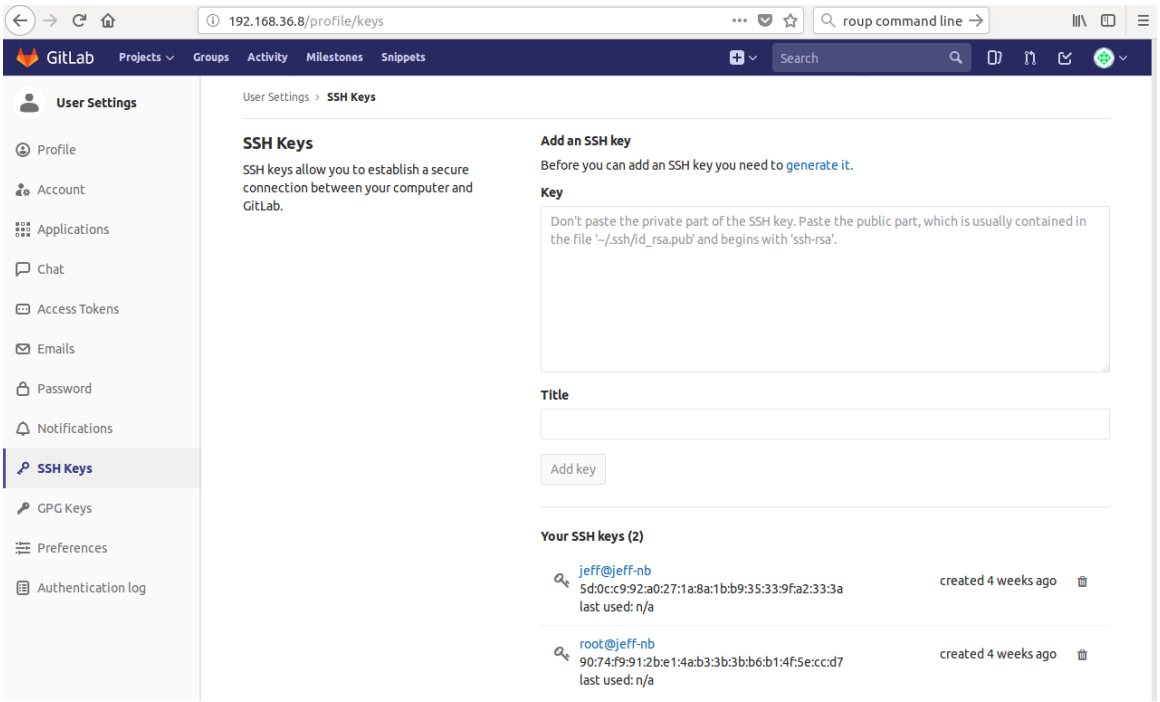
7. 下載特定版(非最新)，進入網頁點選特定 commit，可以下載壓縮檔



或者，先 clone 下最新版,用"*git checkout <commit version>*" 去更新(好像會把沒有 commit 上去的修改全蓋掉,要用 checkout 下特定版請小心),就會蓋掉成該版本



8. 可以上傳 ssh key，這樣 local commit 時就不用輸入 username, password 了.



9. 如果有 OEM special 版本要處理，可以使用 branch 指令，請參考 Git_Branch.doc

10. 參考

因為 git 指令太多了,附上幾個連結請參考,或許會有更好的做法

https://gogojimmy.net/2012/01/17/how-to-use-git-1-git-basic/

https://gitbook.tw/chapters/github/push-to-github.html

https://dotblogs.com.tw/wasichris/2016/05/07/231839