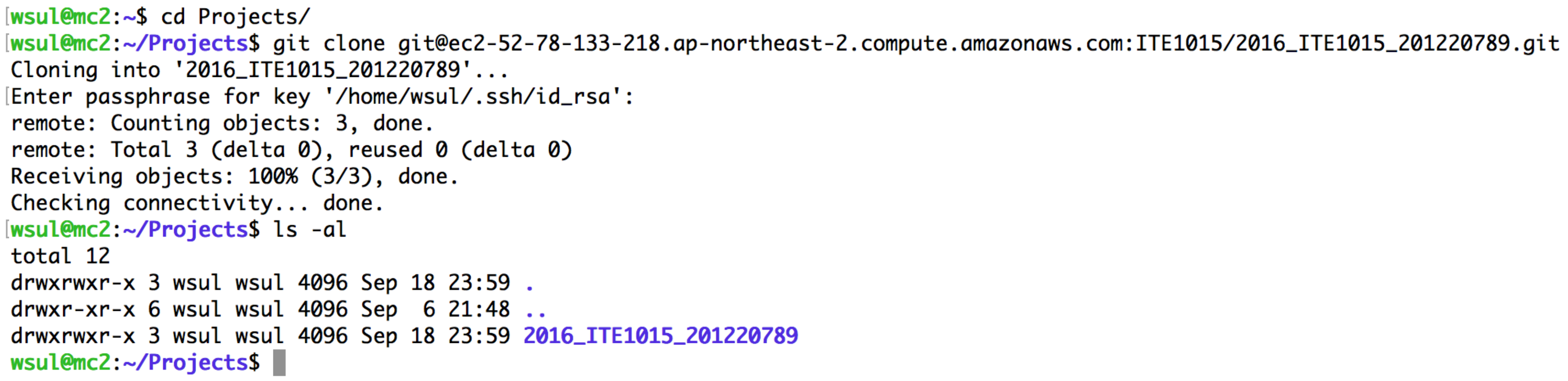
Homework Tutorial

1. 과제용 directory 준비하기

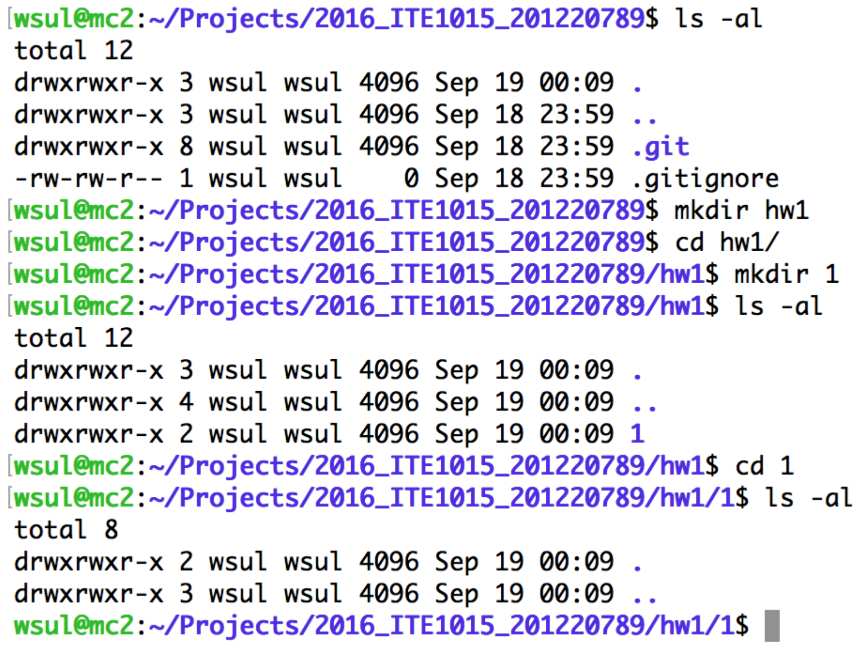
- gitlab으로 부터 개별과제용 프로젝트 내려받기
  - 개인별 과제 프로젝트가 gitlab 서버에 준비되어 있음
  - git clone 명령을 통해 gitlab의 remote repo. → local repo. 내려받음
  - 아래 예시 참고



```
wsul@mc2:~$ cd Projects/
wsul@mc2:~/Projects$ git clone git@ec2-52-78-133-218.ap-northeast-2.compute.amazonaws.com:ITE1015/2016_ITE1015_201220789.git
Cloning into '2016_ITE1015_201220789'...
Enter passphrase for key '/home/wsul/.ssh/id_rsa':
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.
Checking connectivity... done.
wsul@mc2:~/Projects$ ls -al
total 12
drwxrwxr-x 3 wsul wsul 4096 Sep 18 23:59 .
drwxr-xr-x 6 wsul wsul 4096 Sep  6 21:48 ..
drwxrwxr-x 3 wsul wsul 4096 Sep 18 23:59 2016_ITE1015_201220789
wsul@mc2:~/Projects$
```

2. 과제별 directory 생성

- 과제회차별로 directory를 만들고 과제번호로 구분
  - 예) hw1의 1번 과제 – hello world 출력



```
wsul@mc2:~/Projects/2016_ITE1015_201220789$ ls -al
total 12
drwxrwxr-x 3 wsul wsul 4096 Sep 19 00:09 .
drwxrwxr-x 3 wsul wsul 4096 Sep 18 23:59 ..
drwxrwxr-x 8 wsul wsul 4096 Sep 18 23:59 .git
-rw-rw-r-- 1 wsul wsul    0 Sep 18 23:59 .gitignore
wsul@mc2:~/Projects/2016_ITE1015_201220789$ mkdir hw1
wsul@mc2:~/Projects/2016_ITE1015_201220789$ cd hw1/
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1$ mkdir 1
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1$ ls -al
total 12
drwxrwxr-x 3 wsul wsul 4096 Sep 19 00:09 .
drwxrwxr-x 4 wsul wsul 4096 Sep 19 00:09 ..
drwxrwxr-x 2 wsul wsul 4096 Sep 19 00:09 1
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1$ cd 1
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ ls -al
total 8
drwxrwxr-x 2 wsul wsul 4096 Sep 19 00:09 .
drwxrwxr-x 3 wsul wsul 4096 Sep 19 00:09 ..
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$
```

3. main 함수 만들기

- 뼈대 소스코드 작성

```
$vi hello.c
```

```
#include “stdio.h”

int main(void) {

    return 0;

}
```

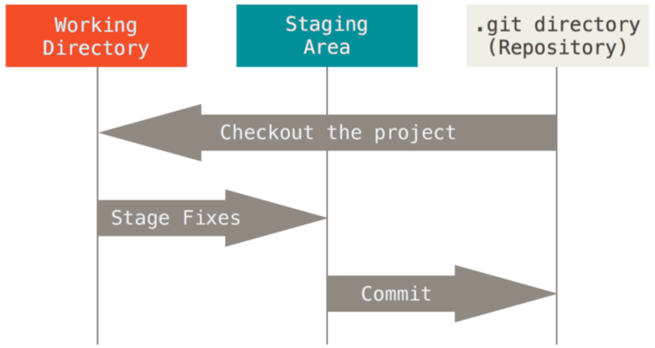
- Makefile 작성

```
$vi Makefile
```

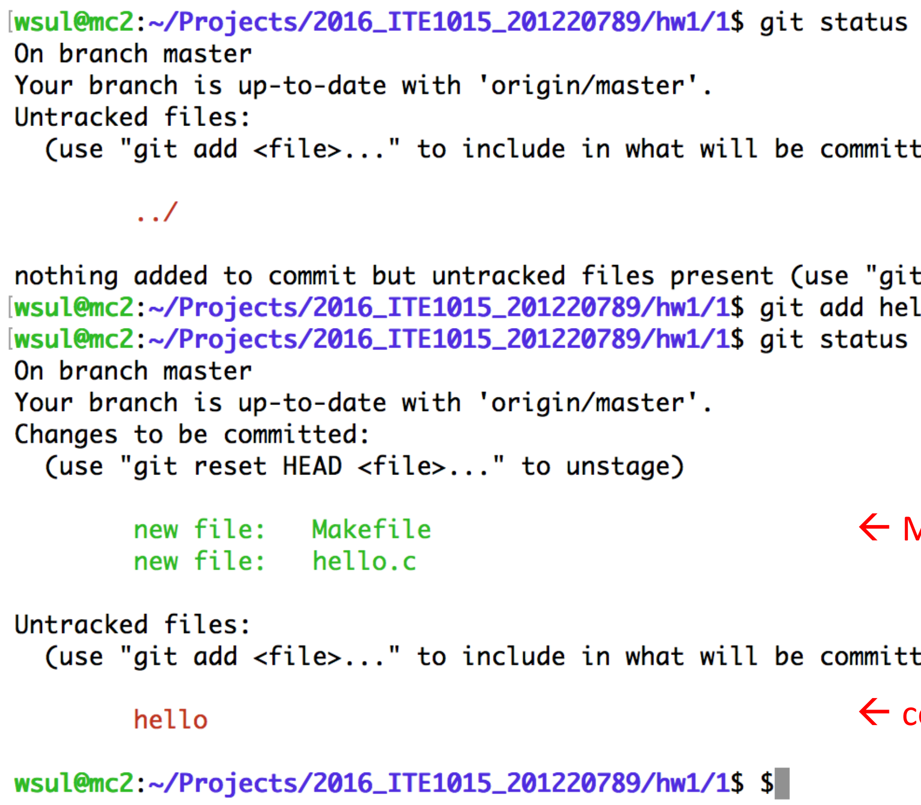
```
all: hello

hello: hello.c
	gcc hello.c -o hello
```

4. initial commit

- make로 빌드가 되는 것을 확인 후 initial commit
  - git add
    - 새로 작성한 파일들은 아직 version관리 되지 않음
    - git add 명령은 해당 파일을 staging함
  - git commit
    - staging 파일을 commit 시킴
    - 새로운 히스토리 로그가 남음 (git log로 확인)



4. initial commit - git add 예시

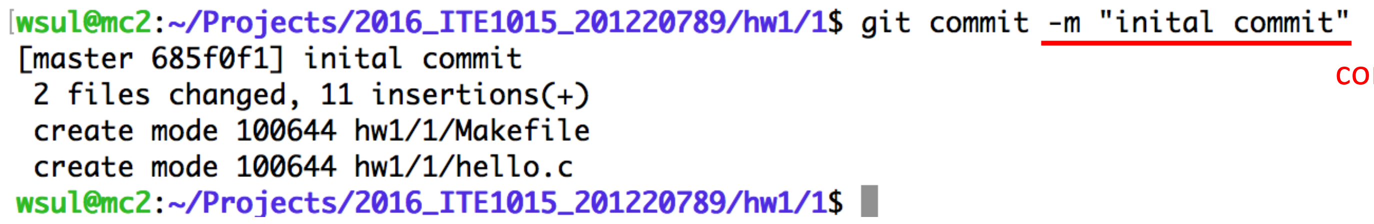


```
[wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        ../

nothing added to commit but untracked files present (use "git add" to track)
[wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git add hello.c Makefile
[wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   Makefile
        new file:   hello.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        hello

wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ $
```

← Makefile, hello.c 두 파일이 staging 됨

← compile된 실행파일은 version 관리 하지 않음

4. initial commit - git commit 예시



```
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git commit -m "inital commit"
[master 685f0f1] inital commit
 2 files changed, 11 insertions(+)
 create mode 100644 hw1/1/Makefile
 create mode 100644 hw1/1/hello.c
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$
```

commit 메시지 남기기

5. 소스코드 수정하기 - 1

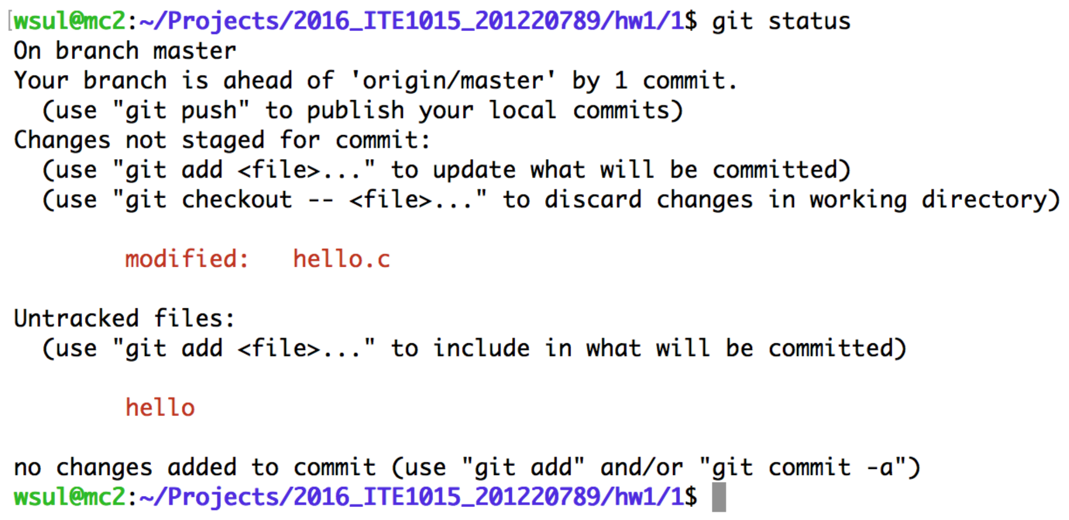
- 뼈대로 만든 hello.c에 내용추가하기

```
$vi hello.c
```

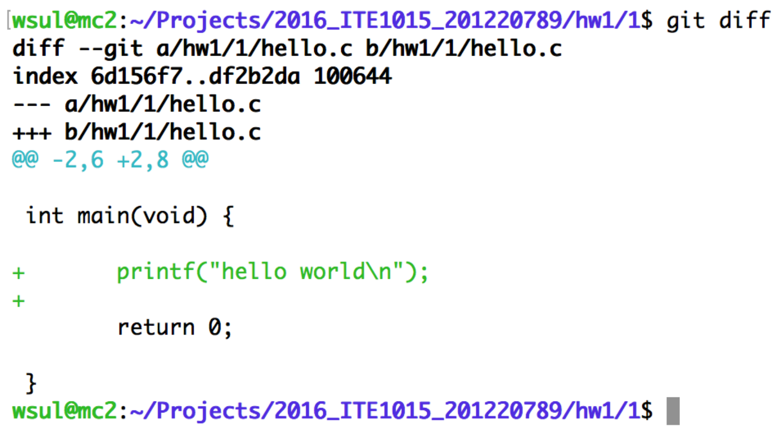
```
#include “stdio.h”

int main(void) {

    printf(“hello world\n”);
    return 0;

}
```

5. 소스코드 수정하기 - 2

- 뼈대로 만든 hello.c에 내용추가하기
- git status 및 git diff 명령으로 변경된 내용 확인하기



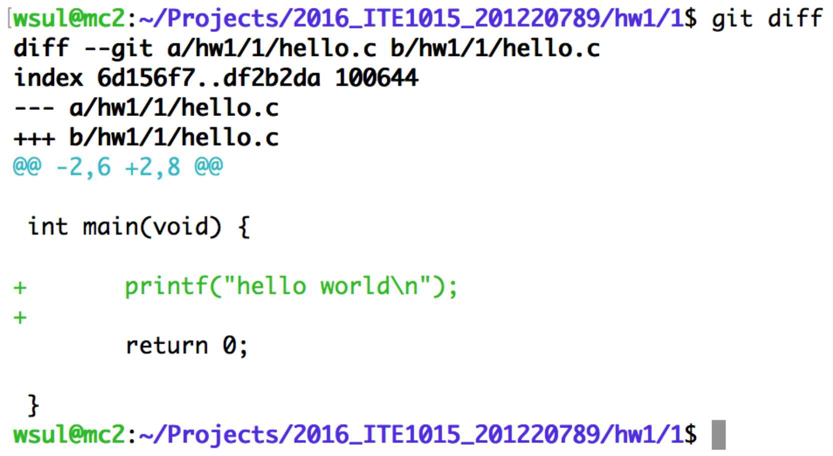
```
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   hello.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        hello

no changes added to commit (use "git add" and/or "git commit -a")
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$
```



```
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git diff
diff --git a/hw1/1/hello.c b/hw1/1/hello.c
index 6d156f7..df2b2da 100644
--- a/hw1/1/hello.c
+++ b/hw1/1/hello.c
@@ -2,6 +2,8 @@

 int main(void) {

+       printf("hello world\n");
+
        return 0;

 }
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$
```

5. 소스코드 수정하기 - 3

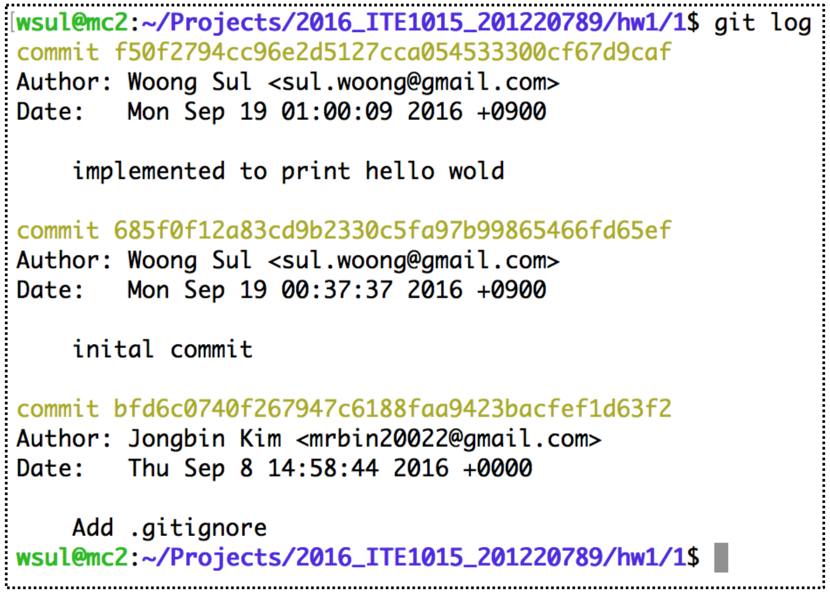
- 뼈대로 만든 hello.c에 내용추가하기
- git status 및 git diff 명령으로 변경된 내용 확인하기
- git commit 으로 수정된 내용 반영하기



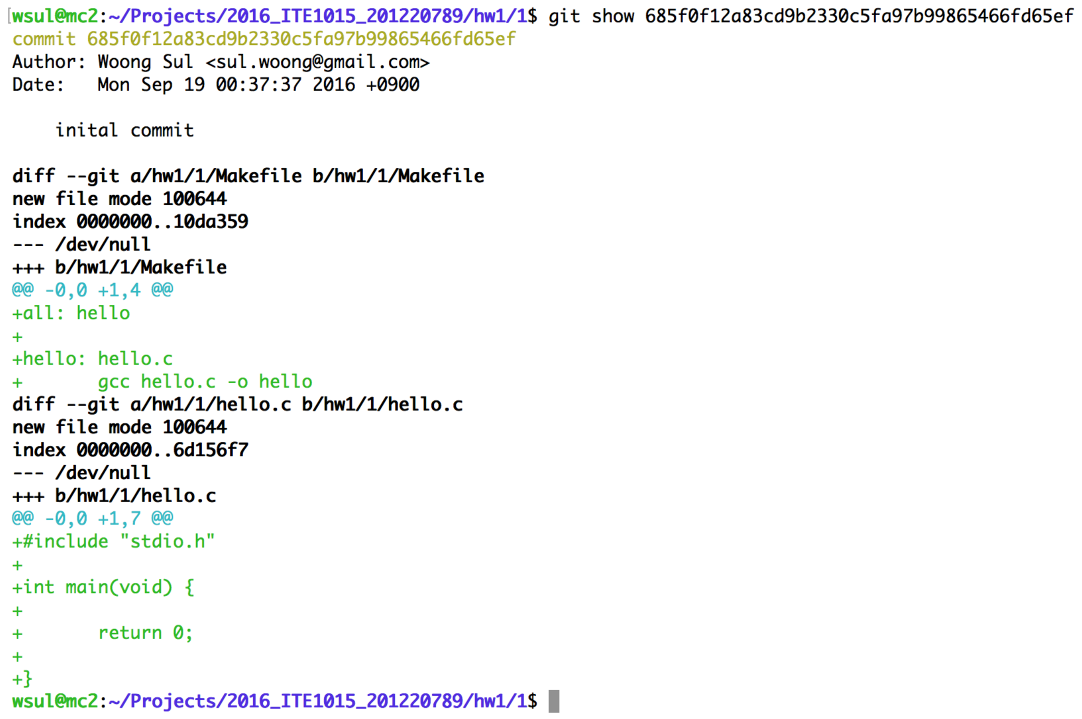
```
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git diff
diff --git a/hw1/1/hello.c b/hw1/1/hello.c
index 6d156f7..df2b2da 100644
--- a/hw1/1/hello.c
+++ b/hw1/1/hello.c
@@ -2,6 +2,8 @@

 int main(void) {

+        printf("hello world\n");
+
         return 0;

 }
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ 
```

6. 소스코드 수정이력 조회

- git log 명령을 통해 commit 시점확인
- git show {commit} 명령은 해당 commit의 내용출력

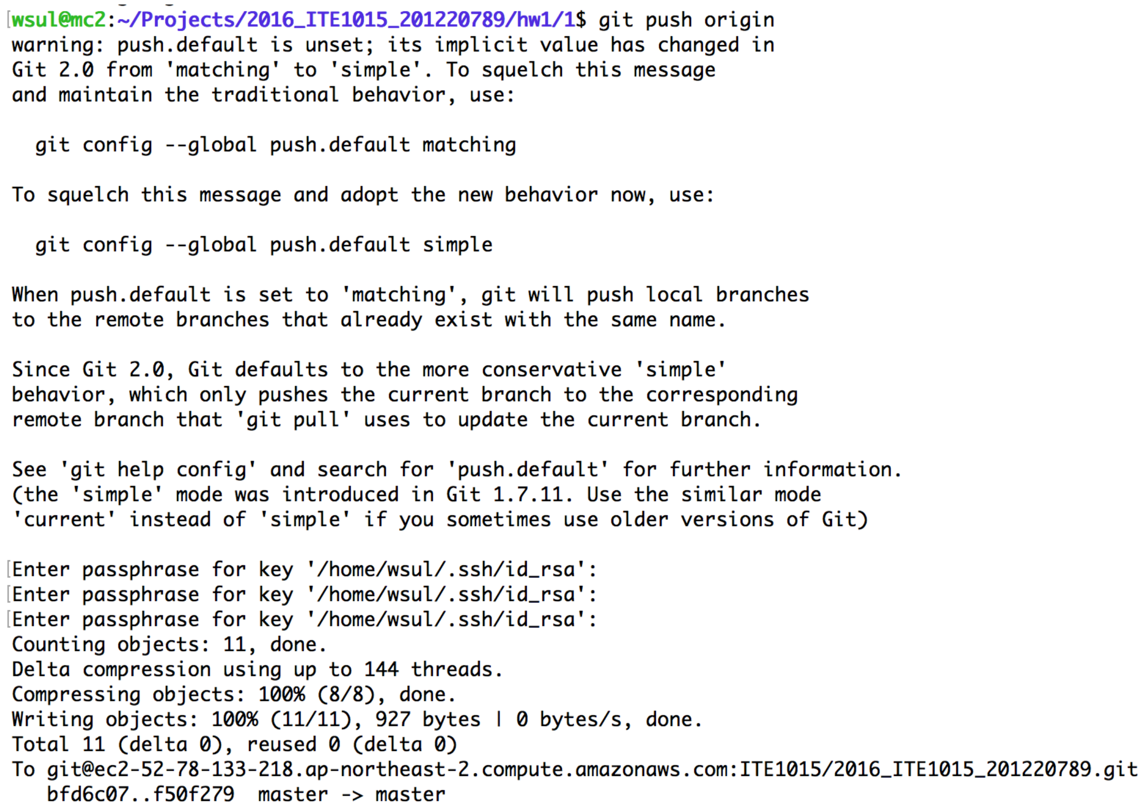


```
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git log
commit f50f2794cc96e2d5127cca054533300cf67d9caf
Author: Woong Sul <sul.woong@gmail.com>
Date:   Mon Sep 19 01:00:09 2016 +0900

    implemented to print hello wold

commit 685f0f12a83cd9b2330c5fa97b99865466fd65ef
Author: Woong Sul <sul.woong@gmail.com>
Date:   Mon Sep 19 00:37:37 2016 +0900

    inital commit

commit bfd6c0740f267947c6188faa9423bacfef1d63f2
Author: Jongbin Kim <mrbin20022@gmail.com>
Date:   Thu Sep 8 14:58:44 2016 +0000

    Add .gitignore
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$
```



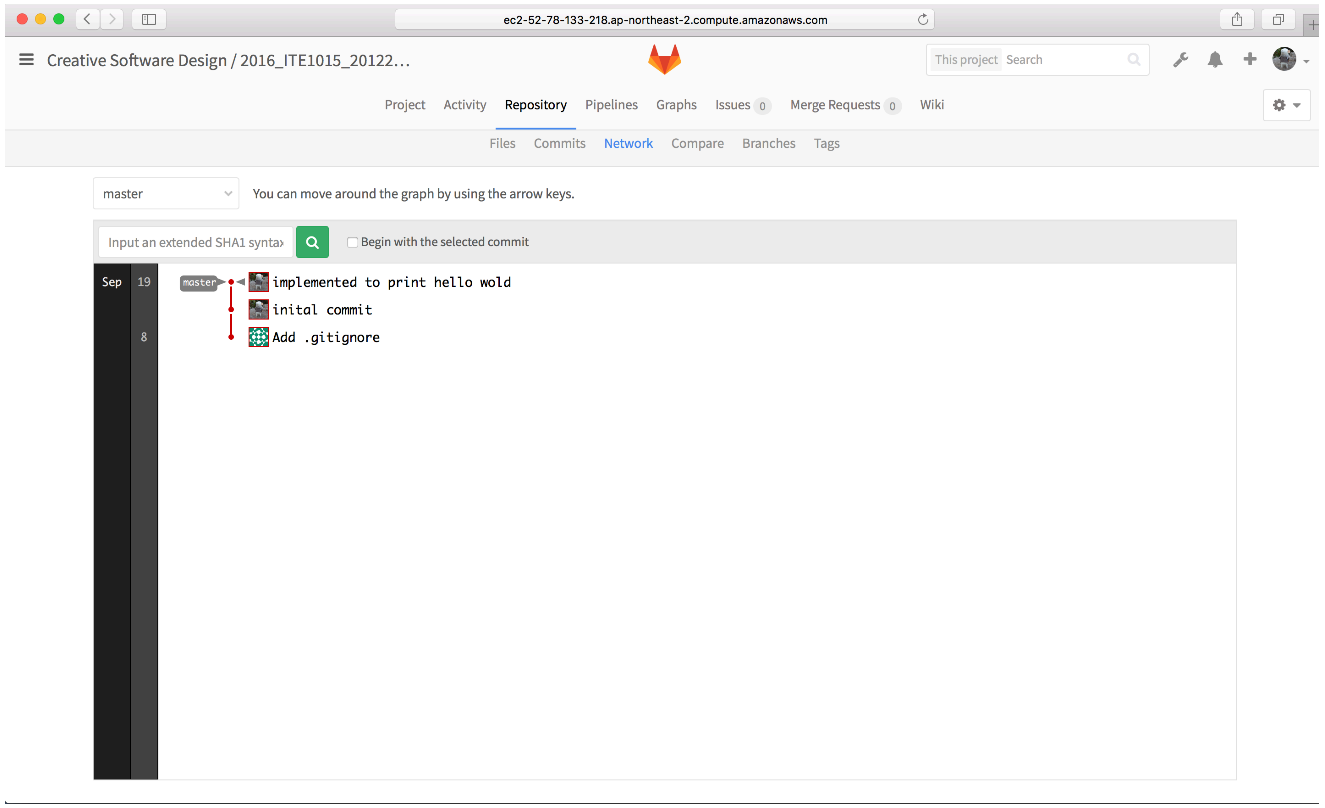
```
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git show 685f0f12a83cd9b2330c5fa97b99865466fd65ef
commit 685f0f12a83cd9b2330c5fa97b99865466fd65ef
Author: Woong Sul <sul.woong@gmail.com>
Date:   Mon Sep 19 00:37:37 2016 +0900

    inital commit

diff --git a/hw1/1/Makefile b/hw1/1/Makefile
new file mode 100644
index 0000000..10da359
--- /dev/null
+++ b/hw1/1/Makefile
@@ -0,0 +1,4 @@
+all: hello
+
+hello: hello.c
+       gcc hello.c -o hello
diff --git a/hw1/1/hello.c b/hw1/1/hello.c
new file mode 100644
index 0000000..6d156f7
--- /dev/null
+++ b/hw1/1/hello.c
@@ -0,0 +1,7 @@
+#include "stdio.h"
+
+int main(void) {
+
+       return 0;
+
+}
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$
```

7. gitlab에 push하기



```
wsul@mc2:~/Projects/2016_ITE1015_201220789/hw1/1$ git push origin
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Enter passphrase for key '/home/wsul/.ssh/id_rsa':
Enter passphrase for key '/home/wsul/.ssh/id_rsa':
Enter passphrase for key '/home/wsul/.ssh/id_rsa':
Counting objects: 11, done.
Delta compression using up to 144 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 927 bytes | 0 bytes/s, done.
Total 11 (delta 0), reused 0 (delta 0)
To git@ec2-52-78-133-218.ap-northeast-2.compute.amazonaws.com:ITE1015/2016_ITE1015_201220789.git
   bfd6c07..f50f279  master -> master
```

8. gitlab에서 commit 확인


ec2-52-78-133-218.ap-northeast-2.compute.amazonaws.com
Creative Software Design / 2016_ITE1015_20122...
This project Search
Project
Activity
Repository
Pipelines
Graphs
Issues 0
Merge Requests 0
Wiki
Files
Commits
Network
Compare
Branches
Tags
master
Input an extended SHA1 syntax
Begin with the selected commit
Sep 19
8
master

You can move around the graph by using the arrow keys.

```
implemented to print hello wold
inital commit
Add .gitignore
```

Thank You