1. **GITHUB**

**Share repository**

1. Create repository on Github
2. Setting
3. Manage Access
4. Invite a collaborator
5. Write email and send

**Download Repository**

**open the terminal**

1. **cd workspace/src**
2. **git clone link the repository**

**Upload Repository for the first time**

**Open terminal, go to the folder that we want upload in this case the folder of name test**

1. **cd workspace/src /test**
2. git init
3. git add .
4. git config user.email "[nabihandres@gmail.com](mailto:nabihandres@gmail.com)"
5. git commit -m "first commit"
6. git remote add origin <https://github.com/XXXXXXX.git>
7. git push -u origin master
8. username
9. password

Note: if create submodules in the repository it mean the repository previous you make git init and have the folder inside, we must delete this folder with ctrl+h in the folder to upload and delete the folder git init and repeat the steps

**Upload Repository for Second or more time**

**Open terminal, go to the folder that we want upload for second time in this case the folder of name test**

1. cd workspace/scr /test
2. git add .
3. git config user.email "[xxxx@gmail.com](mailto:xxxx@gmail.com)"
4. git commit -m "second commit"
5. git push -u origin master
6. username
7. password

**Updated the repository**

**Note:** when is shared repo,If the github repo is pushed with new changes and your local repo doesn't have that changes this error occurs for this you could use:

1. git pull --rebase origin master

2 ) git push origin master

1. **C++**

**Run code C++ in ROS**

**Create a workspace**

1. mkdir -p ~/catkin\_ws/src
2. cd ~/catkin\_ws/
3. catkin\_make

**Create a package**

1. catkin\_ws/src
2. catkin\_create\_pkg test std\_msgs rospy roscpp tf
3. catkin\_make

Note: Inside the folder test now are two folders include, src and **CMakeLists.txt** and **package.xml** files

**Code C++ in this case** helloworld

Dowland code in in Github

1. git clone <https://github.com/emontero1991/tutorial-c-.git>

**Modify the Cmakelist of the folder where is the code in cpp**

catkin\_package(

INCLUDE\_DIRS include

LIBRARIES test

)

add\_executable(helloworld src/ helloworld.cpp)

add\_dependencies(helloworld ${${PROJECT\_NAME}\_EXPORTED\_TARGETS}

${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(helloworld

${catkin\_LIBRARIES}

)

**Open new terminal and run roscore**

1. roscore

**Open second terminal**

1. catkin\_make
2. source devel/setup.bash
3. rosrun test helloworld

**Use Eigen Library in ROS**

**tutorial**

<https://www.youtube.com/watch?v=jna5J3HsJI0>

https://blog.naver.com/tmdals727/221374583966