- How do run this project in my own Ubuntu machine?

  1. Launch Project 1, then in Vocareum click Actions>Download Starter code. This will download all the files you need to make the project run locally in your computer.
  2. **IGNORE** the 2 files "get_free_port.py" and "setup_project1.sh". You do not need these in your local machine
  3. The downloaded files are structured as a catkin workspace. You can either use this structure directly (as downloaded) and build the workspace using the "catkin_make" command or use whatever catkin workspace you already had, and just copy the packages (the 2 folders called "project1_solution" and "two_int_talker" inside the src folder) inside your own src folder.
  4. Once you have a catkin workspace with the 2 packages inside the src folder, you are ready to work on your project without having to make any changes in any of the files.
  5. NOTE: You can source both your ROS distribution and your catking workspace automatically everytime you open up a terminal automatically by editing the ~/.bashrc file in your home directory. For example if your ROS distribution is Indigo, and your catkin workspace is called "robotics_ws" (and is located in your home directory) then you can add the following at the end of your .bashrc file:

     ```
     source /opt/ros/kinetic/setup.bash
     echo "ROS Kinetic was sourced"
     source ~/robotics_ws/devel/setup.bash
     echo "robotics_ws workspace was sourced"
     ```

     This way every time you open up a terminal, you will already have your workspace sourced, such that ROS will have knowledge of the packages there.
  6. To run the project, open up a terminal and fire up a roscore (just type "roscore"). Before moving forward, if you haven't followed the instructions on step 5, you will need to source ROS and the catking workspace every time you open a new terminal. On another 2 separate terminals you need to run the scripts in each package: "`rosrun two_int_talker two_int_talker.py`" and "`rosrun project1_solution solution.py`". At this point, behind the scenes the two scripts are running, hence they are subscribing and publishing to their own respective topics. You can open a new terminal and start listening to the topics using the rostopic echo /name_of_the_topic command.

- I am publishing the right sum between the two integers, yet the grader is timing out, why ???

As per Project 1 instructions, you should be publishing your result in a message of type std_msgs/Int16. **DO NOT USE A CUSTOM MESSAGE OR ANY OTHER KIND OF MESSAGE**. The grader cannot possible account for all possible custom messages you come up with, it is specifically listening for a message of type std_msgs/Int16.
Do not confuse the type of message with the message content. A message can contain several fields defined with any primitive data type (uint8, string, float32, etc). The std_msgs library just contains message definitions that encapsulate each one of these primitive data types.