Sources:

1. <http://wiki.ros.org/ROS/Tutorials>

2. <https://www.theconstructsim.com/>

We can publish to a ros topic or we can create a service which just sends command without topic for any specific functionality.  
ROS

1. Packages:  
   These contain software organisation of ROS code. Each package can contain libraries.
2. Manifests:  
   Kinda description of the package. Version and all.

**Creating Packages:**

These are the steps which go into creating ad building a package. I just skimmed over them as they are not needed much.

If we need to change any package it has the commands to do that.

Content

1. [What makes up a catkin Package?](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.What_makes_up_a_catkin_Package.3F)
2. [Packages in a catkin Workspace](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.Packages_in_a_catkin_Workspace)
3. [Creating a catkin Package](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.Creating_a_catkin_Package)
4. [Building a catkin workspace and sourcing the setup file](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.Building_a_catkin_workspace_and_sourcing_the_setup_file)
5. [package dependencies](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.package_dependencies)
   1. [First-order dependencies](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.First-order_dependencies)
   2. [Indirect dependencies](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.Indirect_dependencies)
6. [Customizing Your Package](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.Customizing_Your_Package)
   1. [Customizing the package.xml](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.Customizing_the_package.xml)
      1. [description tag](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.description_tag)
      2. [maintainer tags](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.maintainer_tags)
      3. [license tags](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.license_tags)
      4. [dependencies tags](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.dependencies_tags)
      5. [Final package.xml](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.Final_package.xml)
   2. [Customizing the CMakeLists.txt](http://wiki.ros.org/ROS/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.Customizing_the_CMakeLists.txt)

**About build and devel:**

The build folder is the default location of the [build space](http://wiki.ros.org/catkin/workspaces#Build_Space) and is where cmake and make are called to configure and build your packages. The devel folder is the default location of the [devel space](http://wiki.ros.org/catkin/workspaces#Development_.28Devel.29_Space), which is where your executables and libraries go before you install your packages.

**Nodes and writing nodes:**

<http://wiki.ros.org/Nodes>

Remapping arguments can be passed to any node and use the syntax name:=new\_name. For example, to configure the talker node to publish to /wg/chatter instead of chatter:

rosrun rospy\_tutorials talker chatter:=/wg/chatter

**RQT graph:**

This page has all the information about the rqt graph and how to access that.

<http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics>

Commands:

1. Rospack: To get information about any package. Like rospack find <package name>
2. roscd : to change directory in ros. Going to some other package  
   Note that [roscd](http://wiki.ros.org/roscd), like other ROS tools, will *only* find ROS packages that are within the directories listed in your [ROS\_PACKAGE\_PATH](http://wiki.ros.org/ROS/EnvironmentVariables#ROS_PACKAGE_PATH). To see what is in your [ROS\_PACKAGE\_PATH](http://wiki.ros.org/ROS/EnvironmentVariables#ROS_PACKAGE_PATH), type:
3. Rosls: like ls
4. Use tab to complete some directory names or paths.
5. Source command:  
   When you "source" something in bash, it will execute each line of the file as is it were typed into the current shell. This is contrasted to actually executing the script, which will be run in a new shell.  
   The setup.bash file is merely adding environment variables to your path to allow ROS to function.  
   You can add the source setup.bash command to your ~/.bashrc file, so that it will be executed every time that you open a new shell.