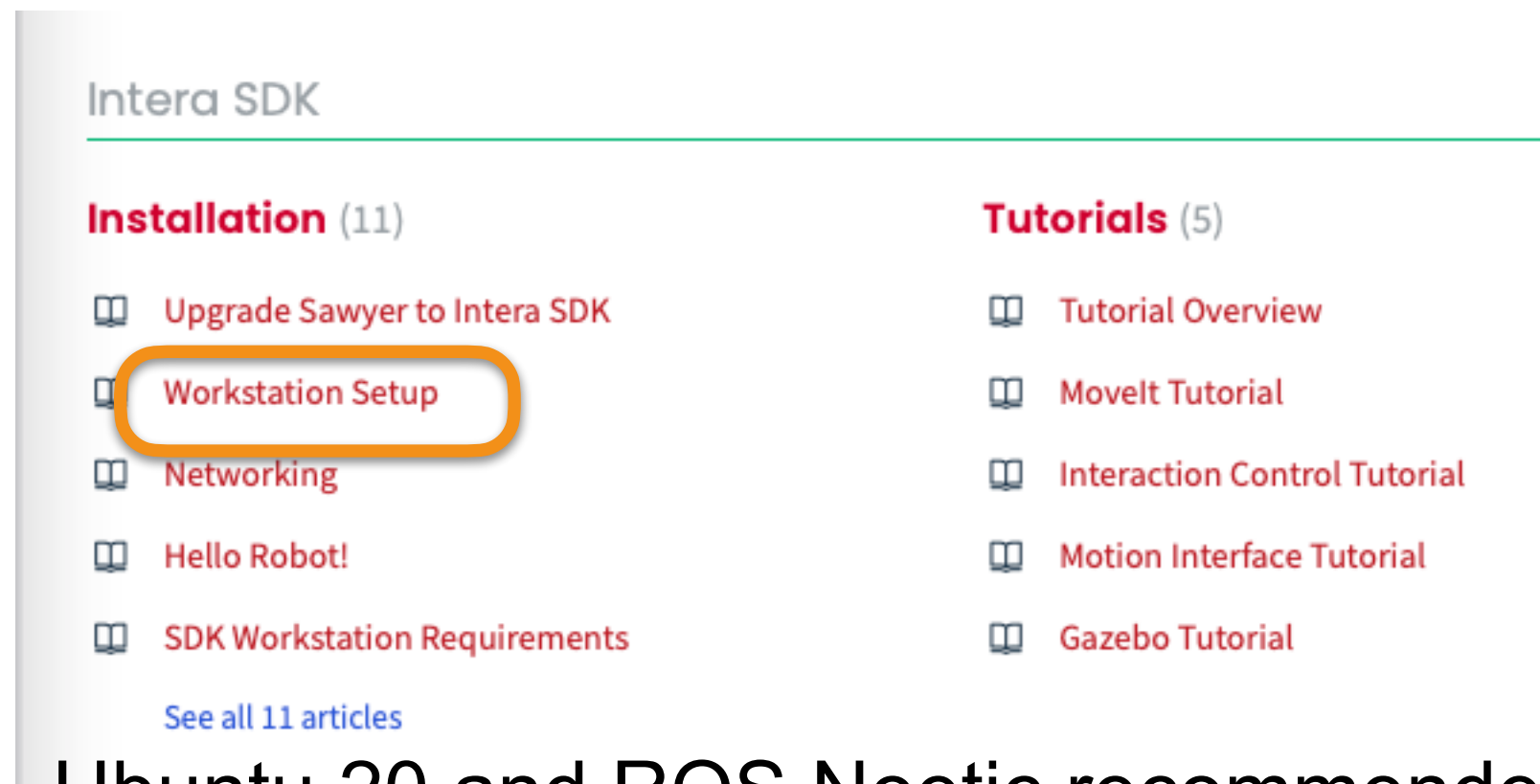


Sawyer with ROS1 / ROS1 with Sawyer

- Content
 - Setting up Workstation for Sawyer (on your computer*)
 - ROS1
 - Turtlesim
 - 'Hello World' (with Python and Sawyer)
 - MoveIt!
- Literature/Sources/Inspiration
 - <https://support.rethinkrobotics.com/support/solutions>
 - <https://ubuntu.com>
 - <https://www.ros.org>
- To Do
 - Set up workstation (own computer)



Ubuntu 20 and ROS Noetic recommended



For ROS boot Sawyer into SDK mode:

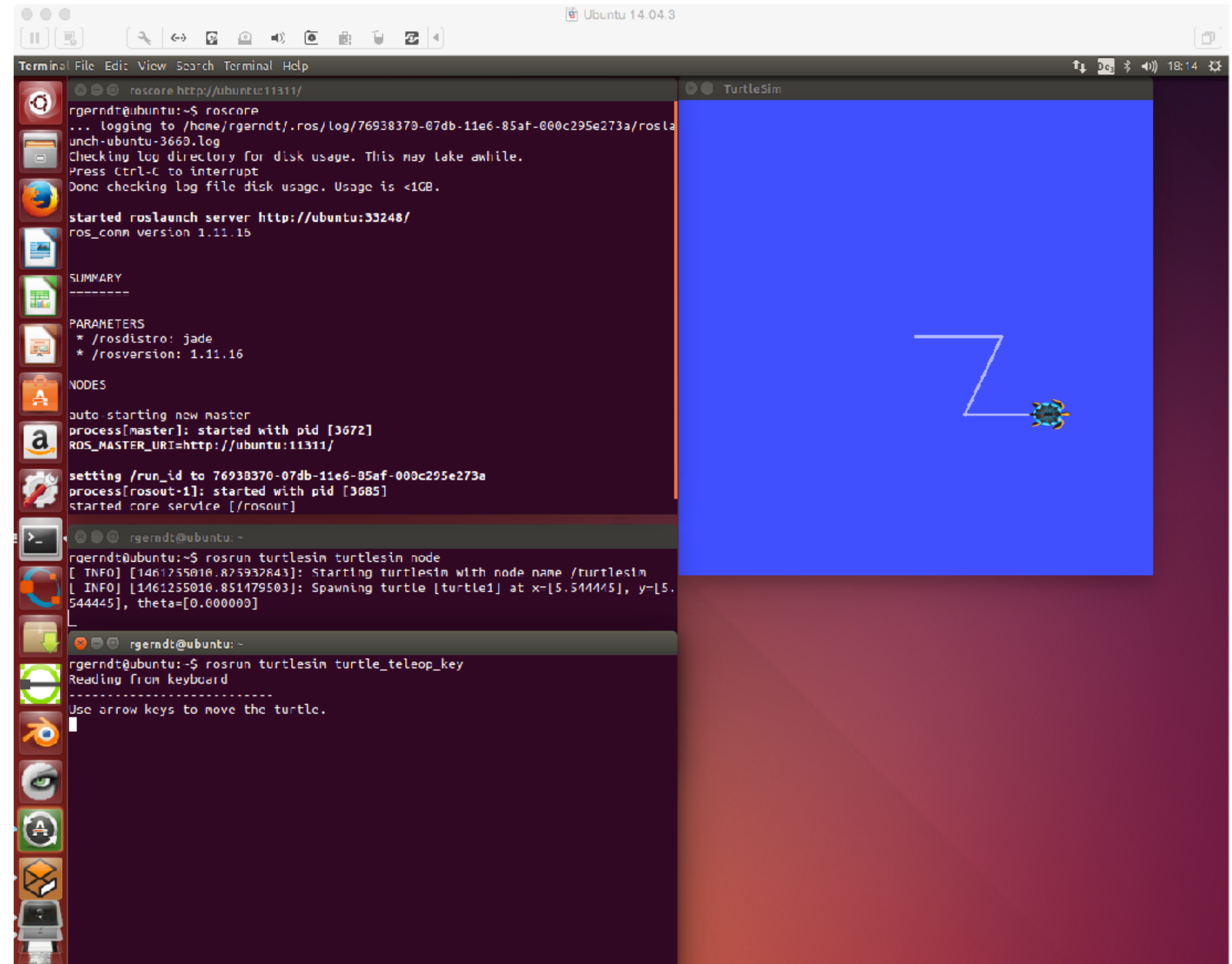
- Keep hitting CTRL+F on the keyboard during startup
 - Select 'Intera-SDK'
 - Reboot
- (For Intera5 select 'Intera')

*consider installing a VirtualBox (www.virtualbox.org)



ROS Run&Smoke Test - Turtlesim and Teleop

- 3 terminals
 - **roscore**
 - **roslaunch turtlesim turtlesim_node**
 - **roslaunch turtlesim turtle_teleop_key**
- Control turtle with keys in teleop terminal



- All ROS software in packages
 - e.g. ,turtlesim‘
- package is collection of files
 - e.g. ,turtlesim_node‘,...
- Publisher - Subscriber
 - nodes publish messages on a topic
 - nodes subscribe to a topic
- Instructions
 - rospack list
 - rospack find *package-name* (e.g. turtlesim)
 - rosls *package-name*
 - roscd *package-name*

- Central ROS Core
 - handling communication
- Distributed Nodes
 - independently running programs
- Instructions
 - `roslaunch package-name executable-name`
 - e.g. `roslaunch turtlesim turtlesim_node`
 - `roslaunch list`
 - `roslaunch info node-name` (see list)
 - `roslaunch kill node-name`

- Related instructions
 - rqt_graph
 - rostopic list
 - rostopic echo *topic-name*
 - e.g. rostopic echo /turtle1/cmd_vel
 - rostopic info *topic-name*
 - rostopic {hz, bw} *topic-name* ({speed, bandwidth})
 - rostopic info *topic-name* (outputs message type)
 - rosmmsg show *message-type-name*

Multiple Instances

- Four separate terminals, roscore running
 - `roslaunch turtlesim turtlesim_node __name:=A`
 - `roslaunch turtlesim turtlesim_node __name:=B`
 - `roslaunch turtlesim turtle_teleop_key __name:=C`
 - `roslaunch turtlesim turtle_teleop_key __name:=D`
 - ,name' parameter overwrites the default name

- Sanity checks
 - roswtf
- Publishing messages from the command line for debugging
 - `rostopic pub -r rate-in-hz topic-name message-type message-content`
 - e.g. `rostopic pub -r 1 /turtle1/cmd_vel geometry_msgs/Twist ,[0, 0, 0]' ,[0, 0, 1]'`

catkin - ROS Build System

- Setting up a workspace
 - `mkdir -p ~/catkin_ws/src`
 - `cd ~/catkin_ws/src`
 - `catkin_init_workspace`
- Build package
 - `cd ~/catkin_ws`
 - `catkin_make`

Creating an ROS package

- change to source directory
 - `cd ~/catkin_ws/src`
- create hello_world package
 - `catkin_create_pkg hello_world std_msgs rospy`
- change to hello_world package/folder
- create folder ,scripts' and change working directory
 - `mkdir scripts`
 - `cd scripts`



Hello_world_publisher.py

```
#!/usr/bin/env python
# license removed for brevity
import rospy
from std_msgs.msg import String

def talker():
    pub = rospy.Publisher('hello_pub', String, queue_size=10)
    rospy.init_node('hello_world_publisher', anonymous=True)
    r = rospy.Rate(10) # 10hz
    while not rospy.is_shutdown():
        str = "hello world %s"%rospy.get_time()
        rospy.loginfo(str)
        pub.publish(str)
        r.sleep()

if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException: pass
```

import rospy

publisher object ,hello_pub', type string, queue 10
name assignment
publishing rate of 10 Hz

Hello_world_subscriber.py

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import String

def callback(data):
    rospy.loginfo(rospy.get_caller_id()+"I heard %s",data.data)

def listener():

    # in ROS, nodes are unique named. If two nodes with the same
    # node are launched, the previous one is kicked off. The
    # anonymous=True flag means that rospy will choose a unique
    # name for our 'listener' node so that multiple listeners can
    # run simultaenously.
    rospy.init_node('hello_world_subscriber', anonymous=True)

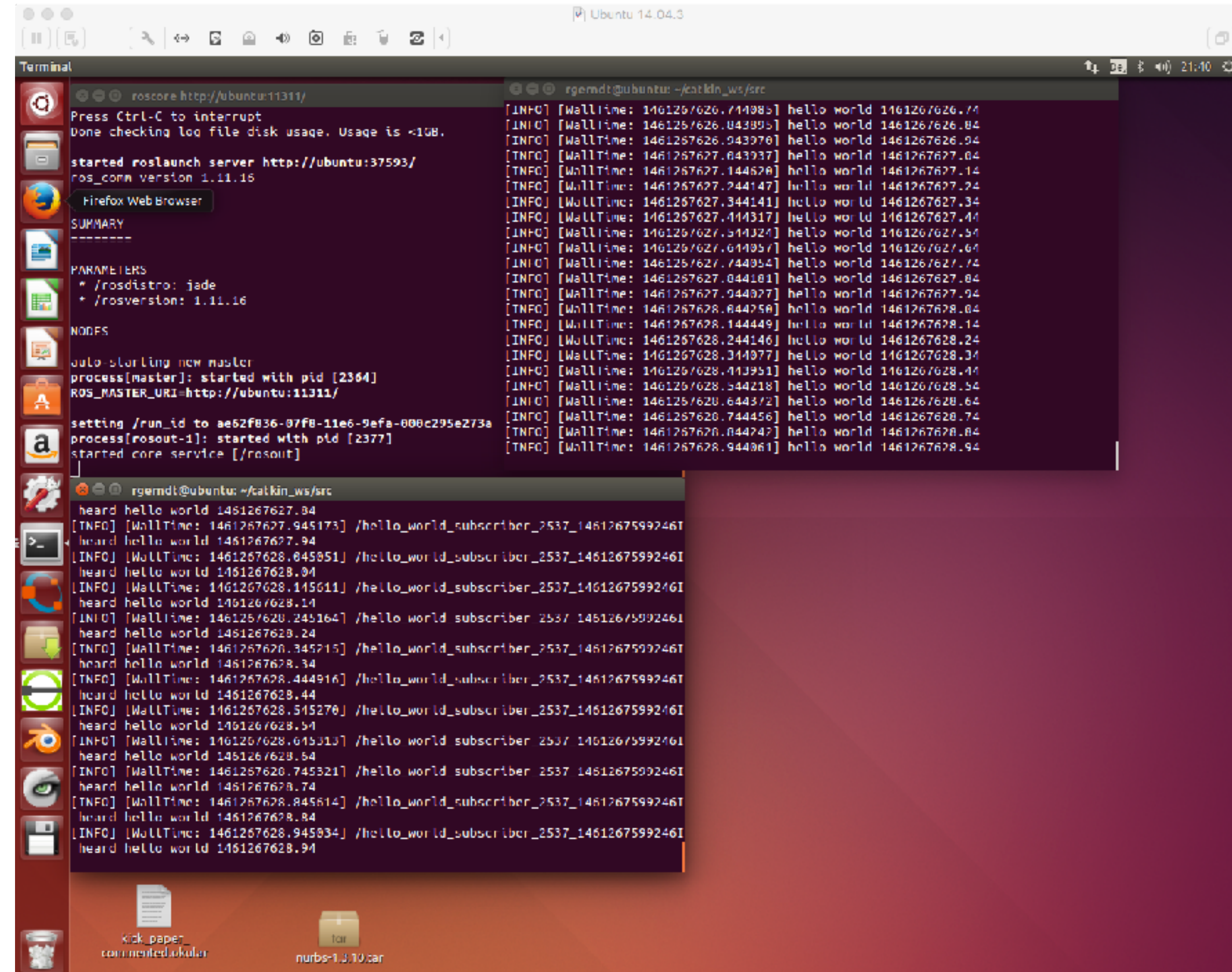
    rospy.Subscriber("hello_pub", String, callback)

    # spin() simply keeps python from exiting until this node is stopped
    rospy.spin()

if __name__ == '__main__':
    listener()
```


Run

- make files executable
 - `chmod +x hello_world_publisher.py`
 - `chmod +x hello_world_subscriber.py`
- build
 - `cd ~/catkin_ws`
 - `catkin_make`
- possibly add path
 - `echo „source ~/catkin_ws/devel/setup.bash“ >> ~/.bashrc`
 - `source ~/.bashrc` (depending on shell, e.g. `.profile` on mac)
- run (in individual terminals)
 - `roscore`
 - `roslaunch hello_world hello_world_publisher.py`
 - `roslaunch hello_world hello_world_subscriber.py`
- try
 - `rqt_graph`



```

Terminal
rgermdt@ubuntu: ~/catkin_ws/src
roscore http://ubuntu:11311/
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:37593/
ros_comm version 1.11.16

SUMMARY
-----
PARAMETERS
 * /roscore: jade
 * /rosversion: 1.11.16

NODES
outgoing new master
process[roscore-1]: started with pid [2354]
ROS_MASTER_URI=http://ubuntu:11311/

setting /run_id to ae52f836-87f8-11e6-9efa-000c295e273a
process[roscore-1]: started with pid [2377]
started core service [/roscout]

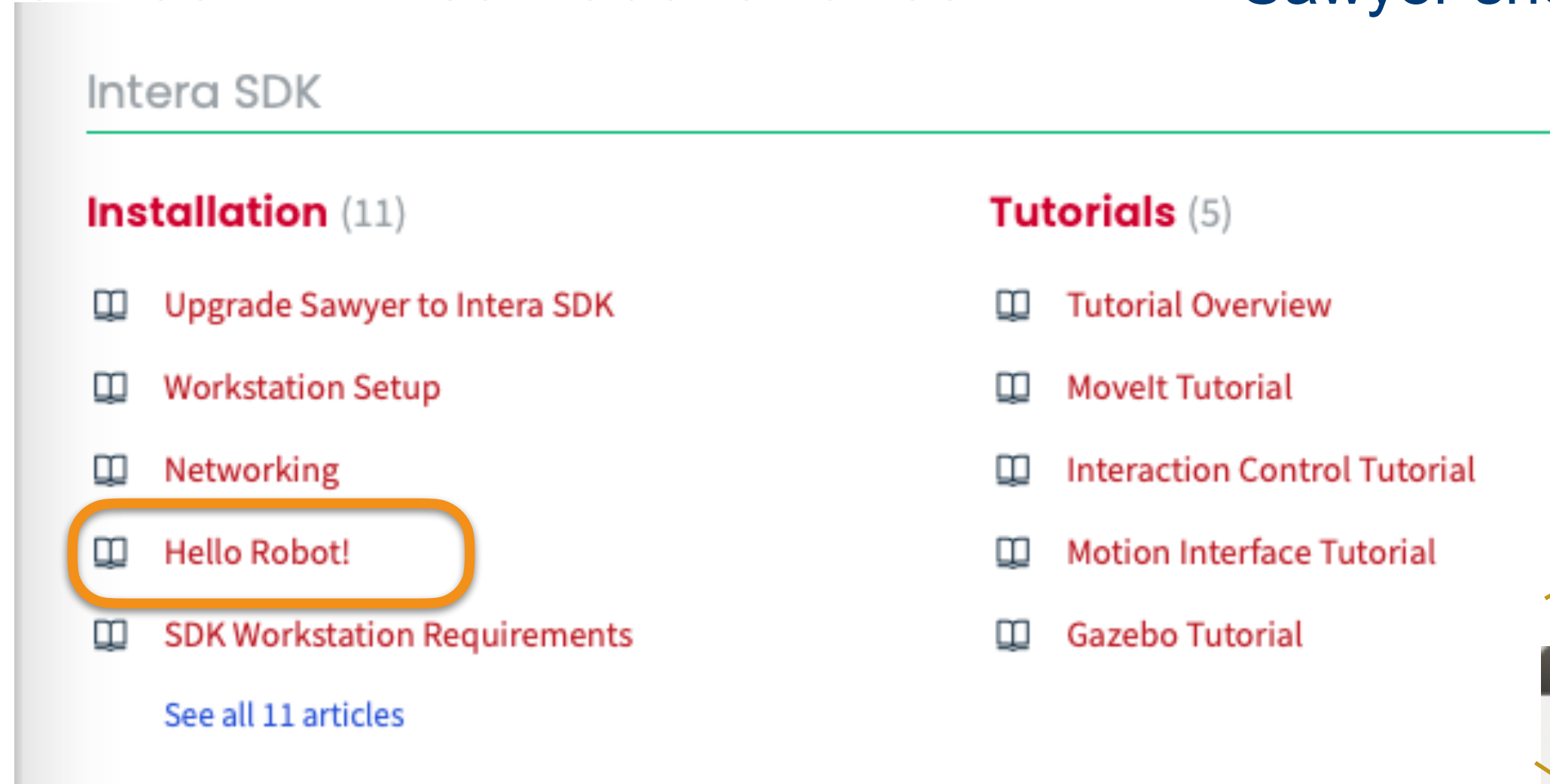
rgermdt@ubuntu: ~/catkin_ws/src
heard hello world 1461267627.84
[INFO] [WallTime: 1461267627.945173] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267627.94
[INFO] [WallTime: 1461267628.045051] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.04
[INFO] [WallTime: 1461267628.145011] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.14
[INFO] [WallTime: 1461267628.245164] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.24
[INFO] [WallTime: 1461267628.345215] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.34
[INFO] [WallTime: 1461267628.444916] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.44
[INFO] [WallTime: 1461267628.545270] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.54
[INFO] [WallTime: 1461267628.645313] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.64
[INFO] [WallTime: 1461267628.745321] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.74
[INFO] [WallTime: 1461267628.845614] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.84
[INFO] [WallTime: 1461267628.945834] /hello_world_subscriber_2537_14612675992461
heard hello world 1461267628.94

```

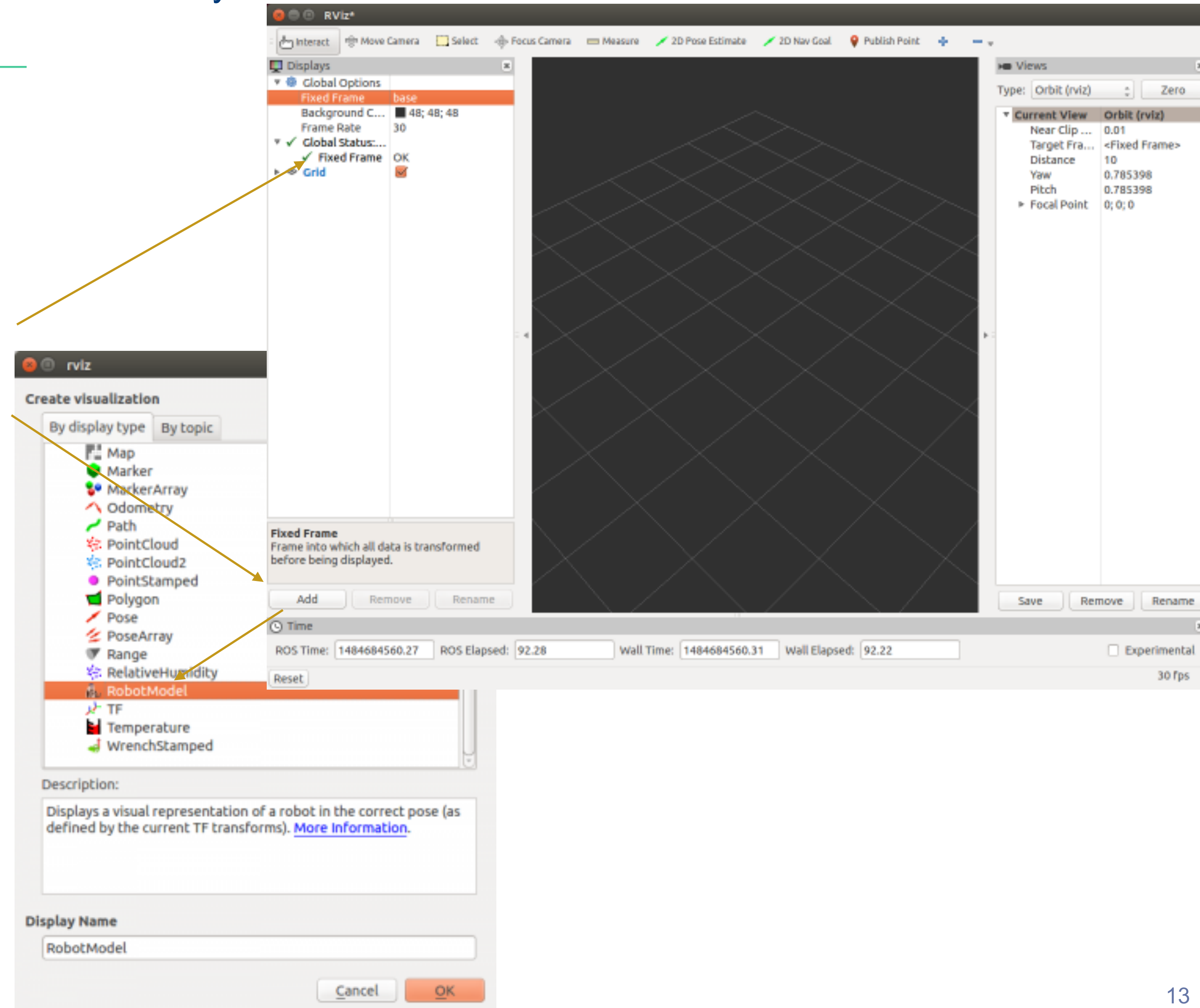

Continue with Sawyer

- Connect with real robot and test

Sawyer should be 'sawyer.local'



- Check out RVIZ
 - Start RVIZ
 - \$ rosrun rviz rviz
- Configure RVIZ
 - Set Fixed Frame as '/base'
 - Add 'Robot Model' to RVIZ





Continue with Sawyer

- Check out 'MoveIt! (<https://moveit.ros.org>)

Intera SDK

Installation (11)

- Upgrade Sawyer to Intera SDK
- Workstation Setup
- Networking
- Hello Robot!
- SDK Workstation Requirements

[See all 11 articles](#)

Tutorials (5)

- Tutorial Overview
- MoveIt Tutorial**
- Interaction Control Tutorial
- Motion Interface T
- Gazebo Tutorial

Displays

- Global Options
- Global Status: Ok
- Grid
- MotionPlanning
 - Status: OK
 - Robot Description
 - Planning Scene Topic
 - Scene Geometry
 - Scene Robot
 - Planning Request
 - Planning Metrics
 - Planned Path

robot_description
move_group/moveit...

Motion Planning

Context Planning Manipulation Scene Objects Stored Scenes Stored States Status

Planning Library
OMPL unspecified Publish Current Scene

Warehouse
Host: 127.0.0.1 Port: 13829 Connect

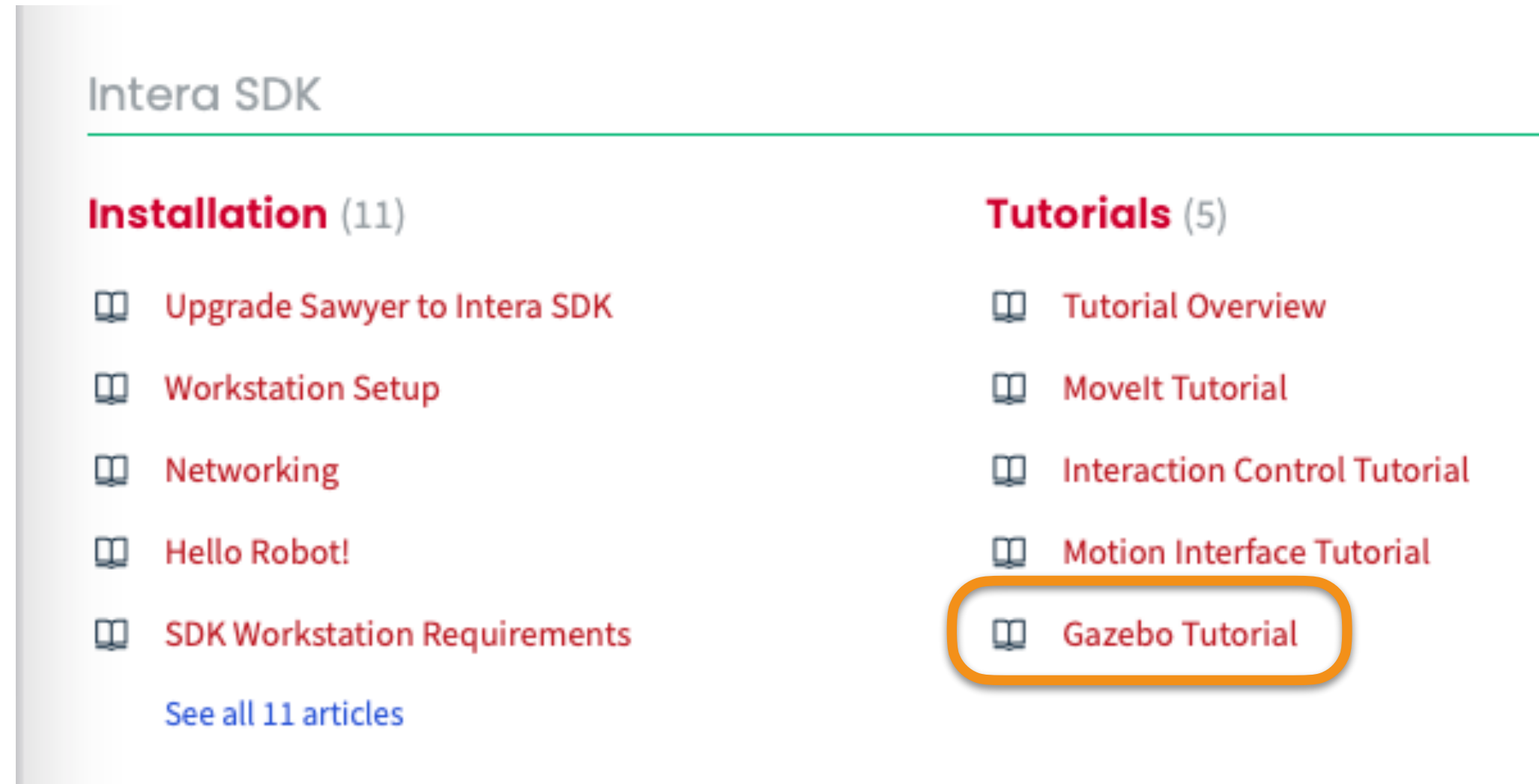
Kinematics
☒ Use Collision Aware IK
☐ Allow Approximate IK Solutions



Gazebo Simulation for Sawyer (no sawyer required)

- Visit <https://support.rethinkrobotics.com/support/solutions/articles/80000980381-gazebo-tutorial>

-



Sawyer Task



- Implement 'some (kinematical) function'
- Run on
 - Gazebo
 - Sawyer
- Compare
- Possibly try with a simple 'teleop'

