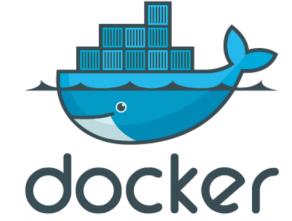


Autonomous Systems

Lab 3

Christoph Killing

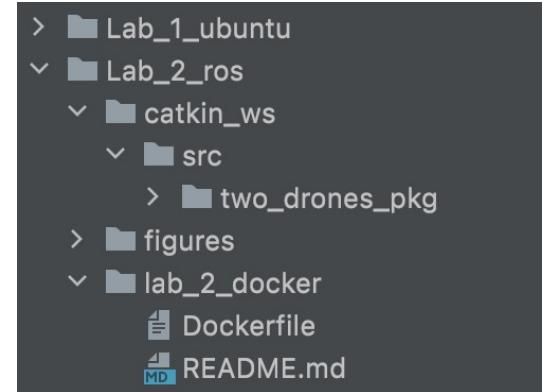


Docker – What and Why?

- open platform for developing, shipping, and running applications.
- enables you to **separate your applications from your infrastructure**
- provides the ability to package and run an application in a **loosely isolated environment** called a **container**
- Containers are lightweight and **contain everything needed to run the application**
 - you do not need to rely on what is currently installed on the host!

Docker – nothing to be scared of

- Write your code as usual
- Dockerfile provides compile instructions for your code



Dockerfile Example

basic ROS docker instructions

A screenshot of a GitHub code viewer showing a Dockerfile. The Dockerfile contains 38 lines of code. Red arrows point from the left side of the image to specific lines in the Dockerfile, corresponding to the following tasks:

- Line 1: FROM ros:kinetic-robot
- Line 6: RUN apt-get update && \
apt-get install -y sudo ninja-build python-catkin-tools python-wstool python-rosdep ros-kinetic-abseil-cpp ros-kinetic-catkin ros-kinetic-map-server ros
- Line 10: RUN /bin/bash -c 'source /opt/ros/kinetic/setup.bash'
- Line 14: RUN git clone https://github.com/Roboy/cartographer_ros.git
#WORKDIR /home/ros/src
- Line 17: RUN cd cartographer_ros/ && \
git checkout roboy
- Line 22: WORKDIR /home/ros
RUN wstool init src && \
wstool merge -t src https://raw.githubusercontent.com/Roboy/cartographer_ros/roboy/cartographer_ros.rosinstall && \
wstool update -t src
- Line 26: RUN src/cartographer/scripts/install_proto3.sh && \
rosdep update && \
rosdep install --from-paths src --ignore-src --rosdistro=\${ROS_DISTRO} -y
- Line 31: RUN catkin config --extend /opt/ros/kinetic
RUN catkin build
RUN /bin/bash -c 'source devel/setup.bash'
- Line 36: RUN cd src/cartographer_ros/ && \
git checkout roboy && \
git pull

The GitHub interface shows 38 lines (29 sloc) and 1.14 KB of code. There are also standard GitHub navigation buttons like Raw, Blame, and copy/paste.

```
38 lines (29 sloc) | 1.14 KB
Raw Blame ⌂ ⌄ ⌅ ⌁
```

```
1 FROM ros:kinetic-robot
2
3 WORKDIR /home/ros
4
5 # Install system libraries
6 RUN apt-get update && \
7     apt-get install -y sudo ninja-build python-catkin-tools python-wstool python-rosdep ros-kinetic-abseil-cpp ros-kinetic-catkin ros-kinetic-map-server ros
8
9 # Initialize ROS
10 RUN /bin/bash -c 'source /opt/ros/kinetic/setup.bash'
11
12 # Copy autonomous driving code
13 WORKDIR /home/ros/src
14 RUN git clone https://github.com/Roboy/cartographer_ros.git
15 #WORKDIR /home/ros/src
16
17 RUN cd cartographer_ros/ && \
18     git checkout roboy
19
20 # Compile cartographer
21 WORKDIR /home/ros
22 RUN wstool init src && \
23     wstool merge -t src https://raw.githubusercontent.com/Roboy/cartographer_ros/roboy/cartographer_ros.rosinstall && \
24     wstool update -t src
25
26 RUN src/cartographer/scripts/install_proto3.sh && \
27     rosdep update && \
28     rosdep install --from-paths src --ignore-src --rosdistro=${ROS_DISTRO} -y
29
30 # Build catkin workspace
31 RUN catkin config --extend /opt/ros/kinetic
32 RUN catkin build
33 RUN /bin/bash -c 'source devel/setup.bash'
34
35 # Update Cartographer_ROS roboy
36 RUN cd src/cartographer_ros/ && \
37     git checkout roboy && \
38     git pull
```

install packages

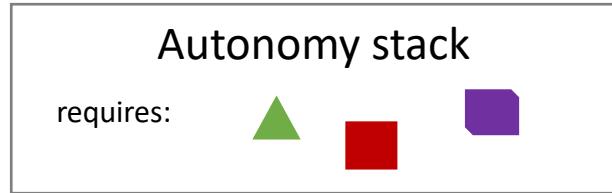
initialize ROS

clone git

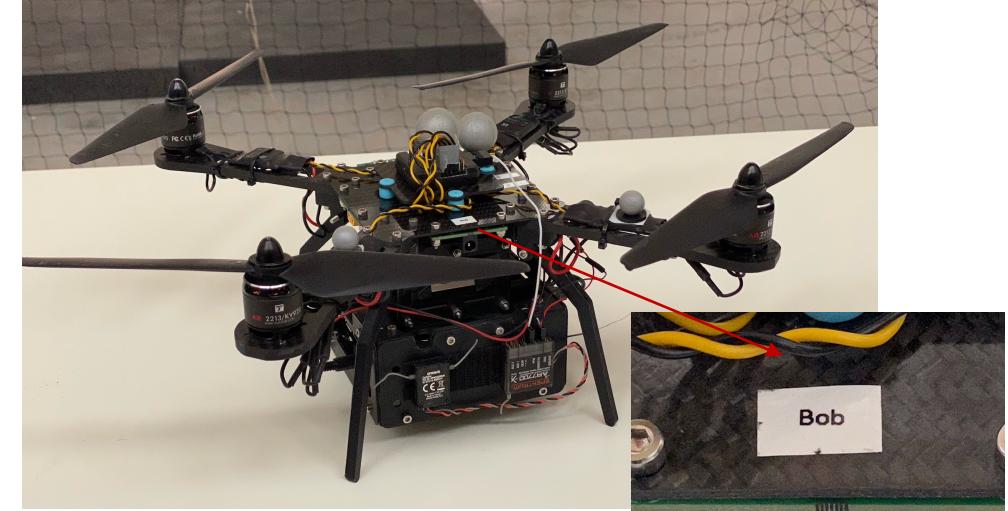
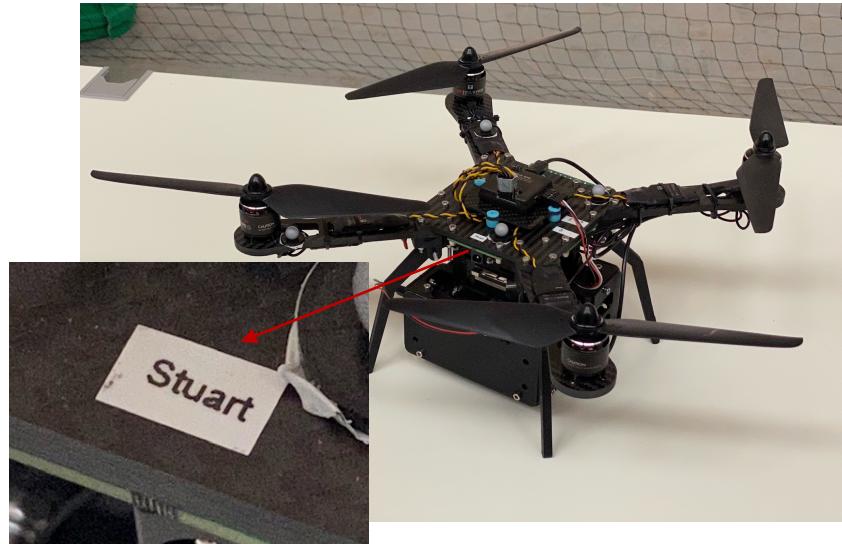
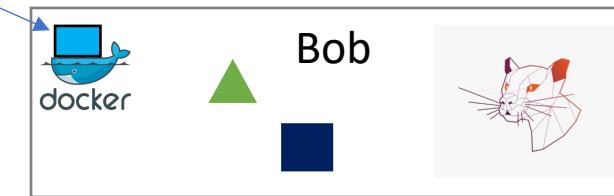
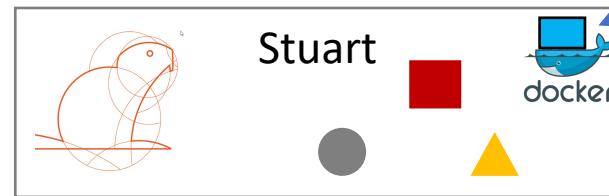
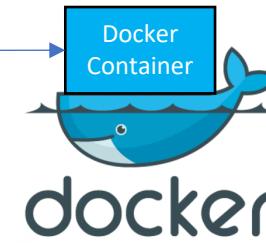
git internals

build the workspace

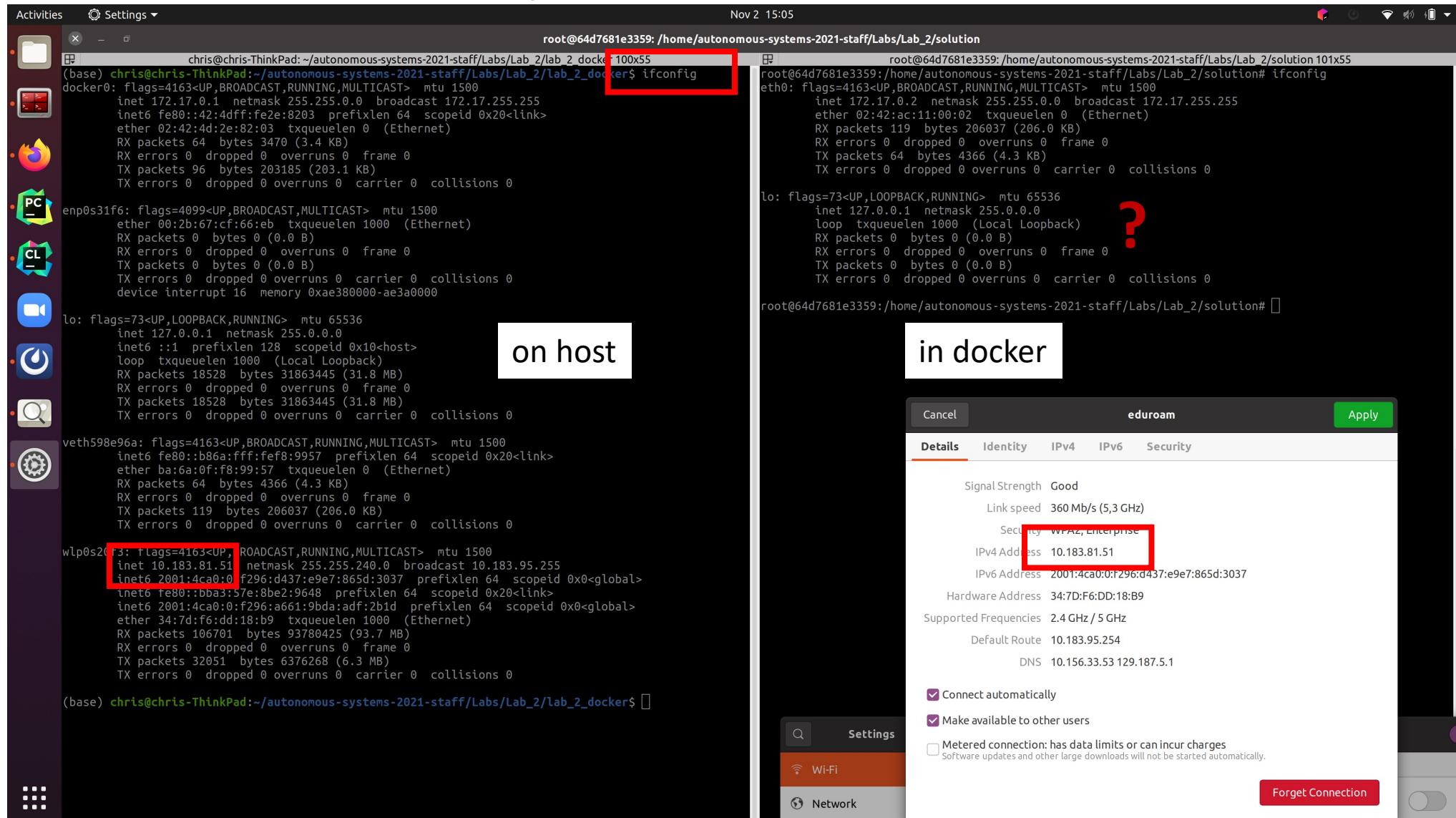
Deploying software through Docker



Dockerfile



Docker – an independent machine?

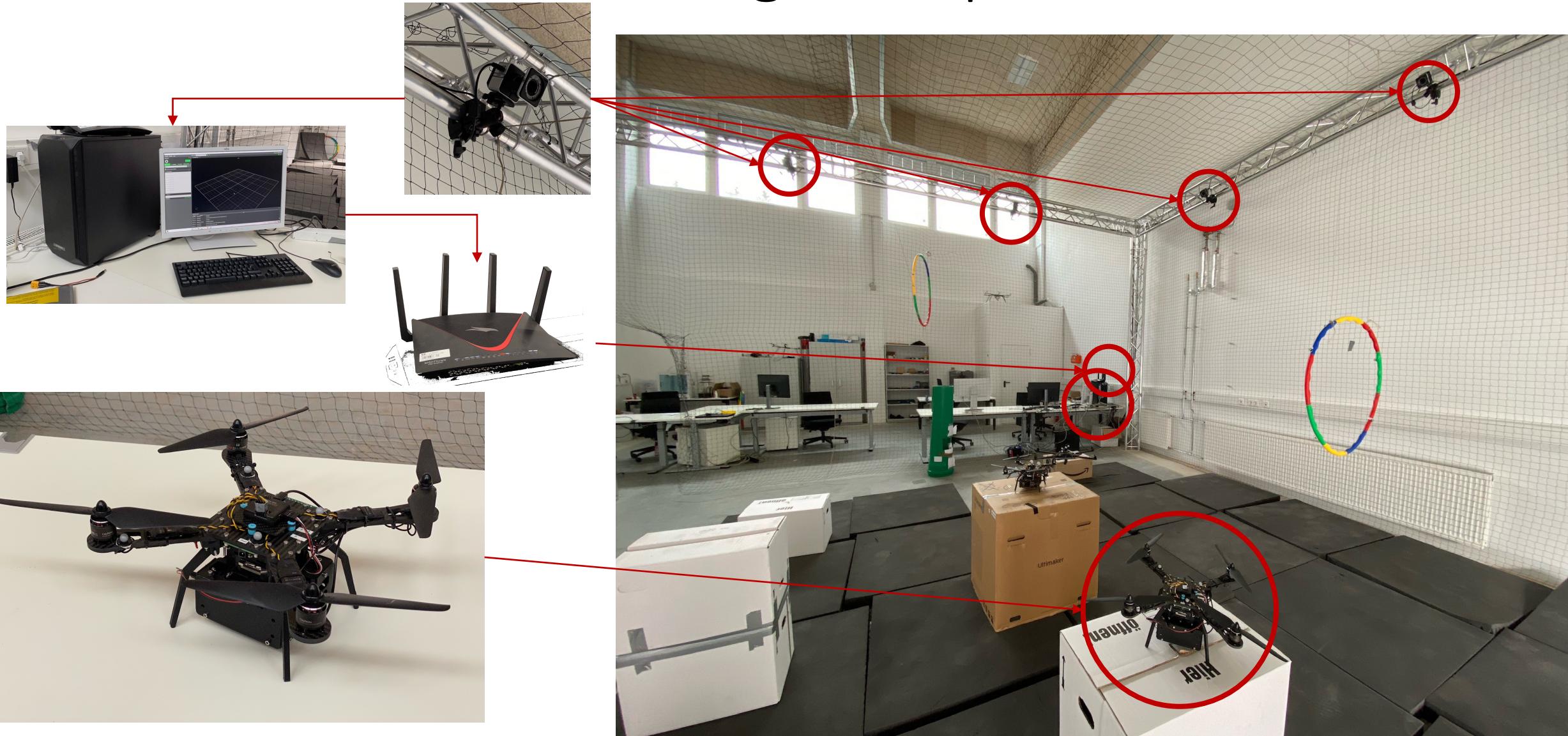


ROS Network

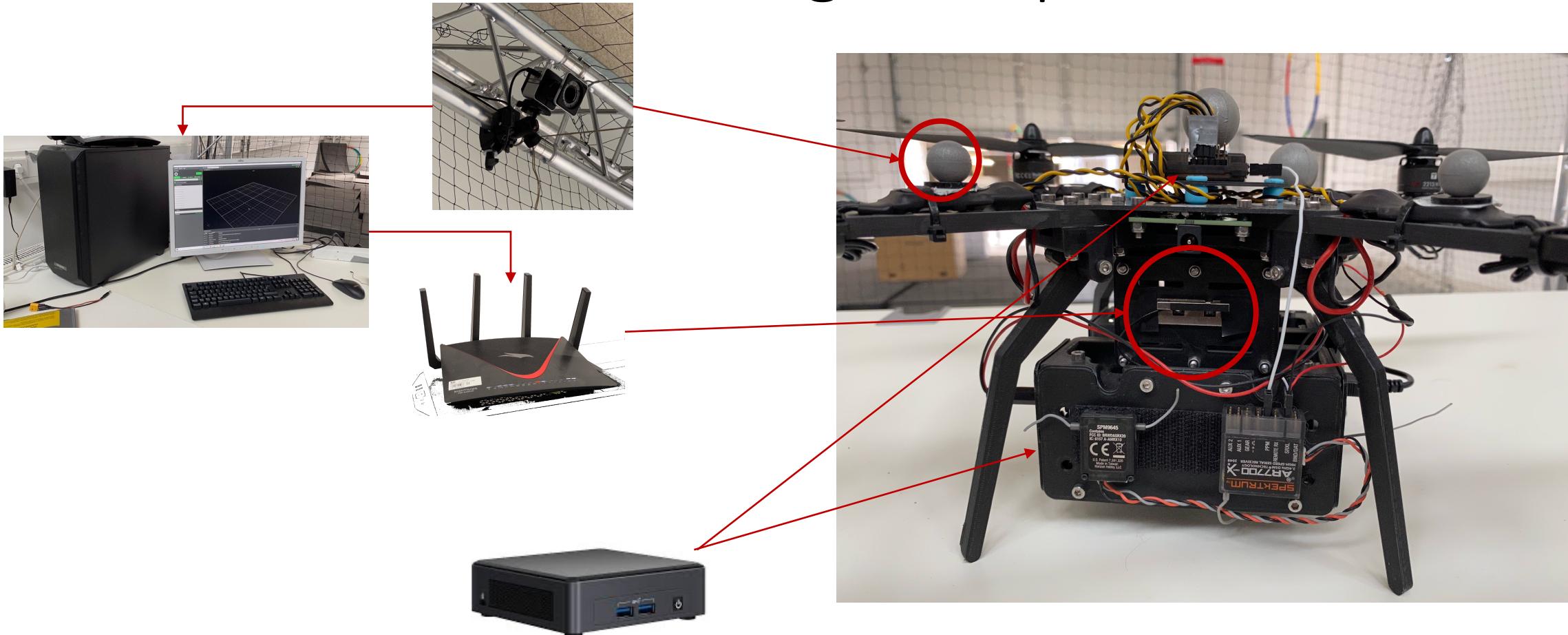
- ROS uses the network to ship data around
 - i.e. TCP, the Transmission Control Protocol
- Simple example:
 - publisher gives the subscriber the **IP address** and **port** on which to call connect
 - subscriber **creates a TCP/IP socket** to the specified address and port
 - nodes exchange a Connection Header
 - includes information like the **MD5 sum of the message type** and the **name of the topic**
 - then the **publisher begins sending serialized message data** directly over the socket.

More info: <http://wiki.ros.org/ROS/TCPROS>

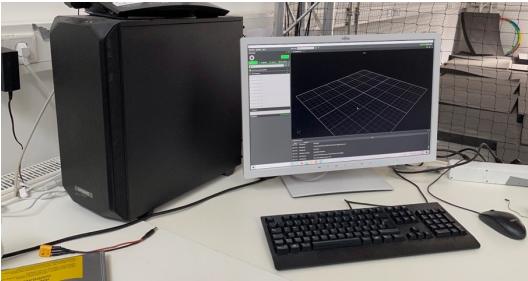
A visual ROS networking example



A visual ROS networking example



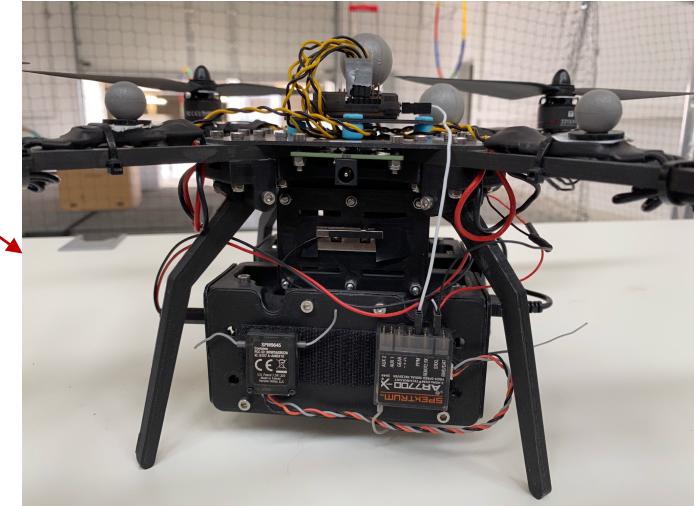
A visual example



HOST_IP



REMOTE_IP



```
export ROS_MASTER_URI=http://HOST_IP:11311/  
export ROS_HOSTNAME=HOST_IP  
export ROS_IP=HOST_IP
```

```
export ROS_MASTER_URI=http://HOST_IP:11311/  
export ROS_HOSTNAME=REMOTE_IP  
export ROS_IP=REMOTE_IP
```

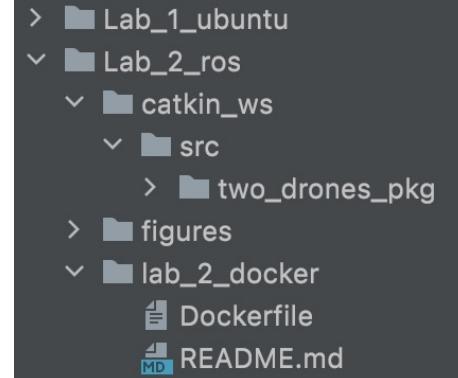
ROS

roscore

vicon node
• advertises /state

states estimation node
• subscribes to /state

Docker – Setup



Setup

Build Image

```
sudo docker build -t cartographer-devel . --no-cache
```

Create Container

```
sudo docker run -it -d --network=host --name cartographer-devel cartographer-devel:latest bash
```

Docker – Usage

Booting

Startup

To start the container:

```
sudo docker start cartographer-devel
```

Interaction

To enter a docker shell:

```
sudo docker exec -it cartographer-devel bash
```

Docker – where to start?

- OSRF Docker Repo: https://github.com/osrf/docker_images
- Previous work: https://github.com/Roboy/autonomous_driving/tree/master/dockers/cartographer-devel



The screenshot shows a GitHub repository page for the file `Dockerfile` in the `df19ab7d59` branch of the `docker_images / ros / melodic / ubuntu / bionic / desktop-full` directory. The page includes navigation buttons for "Go to file" and "History". Below the header, there's a commit message from `ruffsl` that says "Revert LABELS in ROS images" and indicates the latest commit was made on 17 Nov 2020. It also shows 3 contributors. The code editor displays the Dockerfile content:

```
9 lines (7 sloc) | 326 Bytes
Raw Blame ⚡ 🖍️ 🗑️

1 # This is an auto generated Dockerfile for ros:desktop-full
2 # generated from docker_images/create_ros_image.Dockerfile.em
3 FROM osrf/ros:melodic-desktop-bionic
4
5 # install ros packages
6 RUN apt-get update && apt-get install -y --no-install-recommends \
7     ros-melodic-desktop-full=1.4.1-0* \
8     && rm -rf /var/lib/apt/lists/*
9
```

Pitfalls

- Docker has no GUI -> X-server errors
- ROS is network based
 - You will have a host-machine which runs the Docker container
 - Need for communication between them
- GitLab access using ssh
 - on docker build, use `--build-arg ssh_prv_key="$(cat ~/.ssh/id_rsa)"` to copy your ssh key
 - in your Dockerfile, use `ARG ssh_prv_key` to accept your key

Resources

- SSH Keys: <https://docs.gitlab.com/ee/ssh/>
- Docker Install: <https://docs.docker.com/engine/install/ubuntu/>
- ROS Documentation: <http://wiki.ros.org/docker/Tutorials/Docker>
- Docker Documentation: https://hub.docker.com/_/ros

Homework 3 (due Nov 17 2021, 23:59)

- Dockerize Homework 2
 - make sure your **Docker pulls your github repo**, do not copy/past data into it
 - for this, assume your **ssh key is located at `~/.ssh/id_rsa`** and load it in the docker build process using the `--build-arg ssh_prv_key="$(cat ~/.ssh/id_rsa)"` flag
 - properly **document** what you did and **how to run it** (benchmark is our previous work): https://github.com/Roboy/autonomous_driving/tree/master/dockers/cartographer-devel
 - how many terminals do we need?
 - what should we put into each terminal?
 - commands should be on copy-paste level
- note that all further submissions will be dockerized
- submission is required