

Docker containers in Windows

This is a quick guide that will lead you through the steps to successfully launch and interact with a simulation of a TIAGo or TIAGo++ robot in a Windows PC. This is done using the docker containers of the TIAGo development PCs that are made available upon your purchase of the robot.

It is worth mentioning that docker runs natively in Linux, and it can be solely run in either Windows or Mac via virtual machines.

Installation

Two publicly available resources have to be installed in your Windows system:

1. **Docker Desktop**, an easy to install application that enables you to manage your containers. [Download it here](#).
2. **Xming X server** for Windows, an X11 display server that will allow you to visualize, in particular the RViZ robot visualization and tools and the Gazebo simulation of the robot. [Download it here](#).

Generic procedure

To be able to use these tools, you have to always make sure that

1. The Docker Desktop is running. To do this just double click the desktop icon

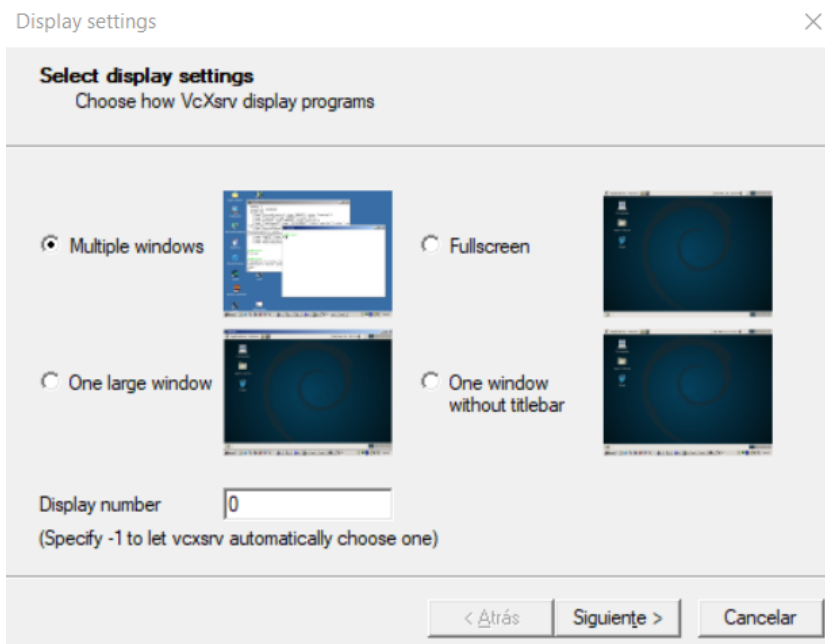


2. The display server is running. This has to be done only once per session. To do this, first double click the XLauncher desktop icon

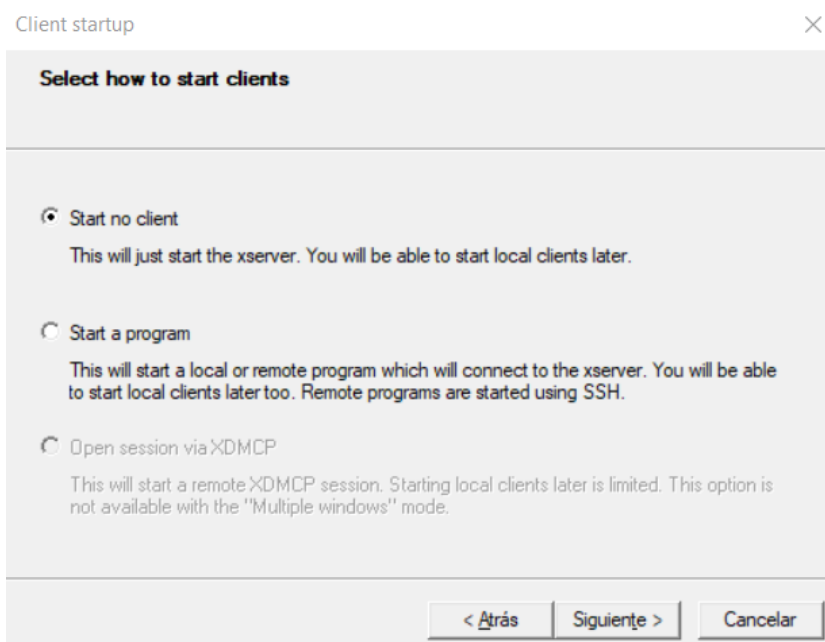


This will lead you to the following dialogue:

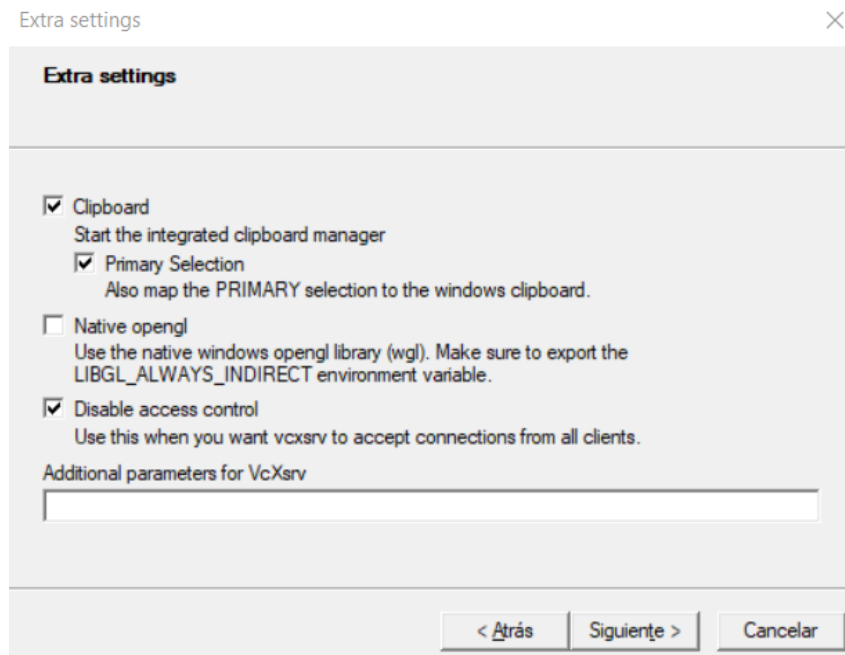
- a. First choose “Multiple windows” and set to 0 the “Display number”



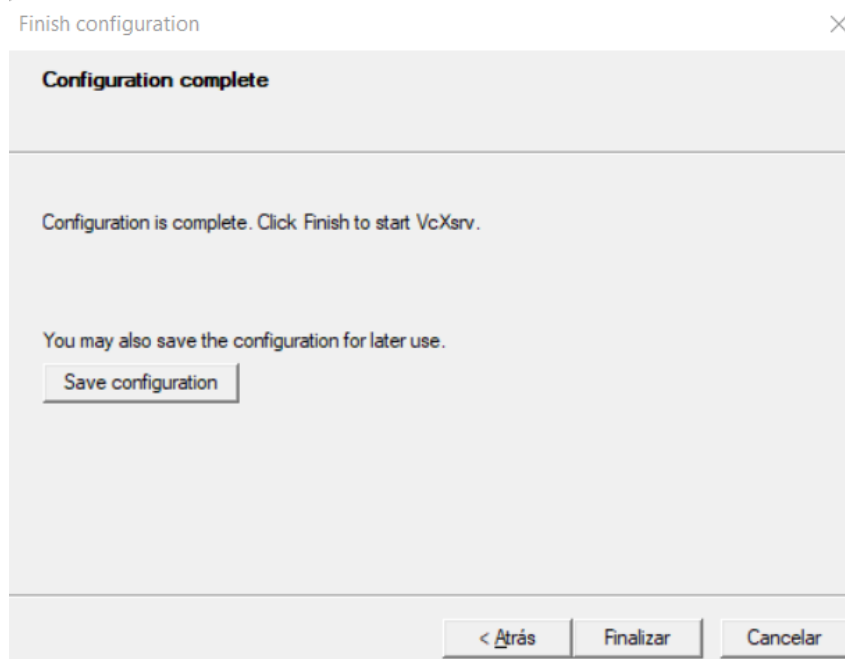
- b. Then, set “Start no client” in the second screen



- c. In the third screen, click “Clipboard” as Primary Selection and “Disable access control”, while leaving “Native opengl” unclicked



- d. And just click “Finalize” in the last screen

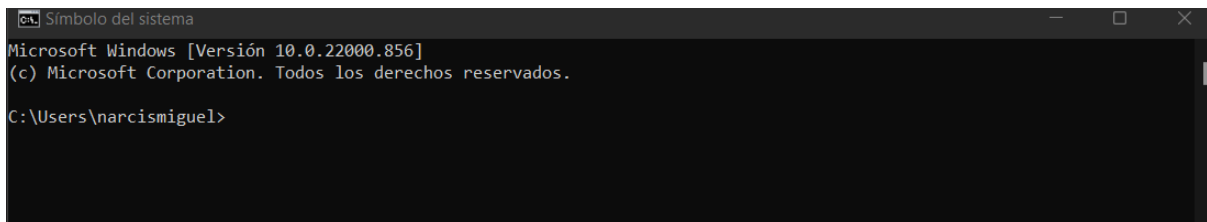


Pulling the container

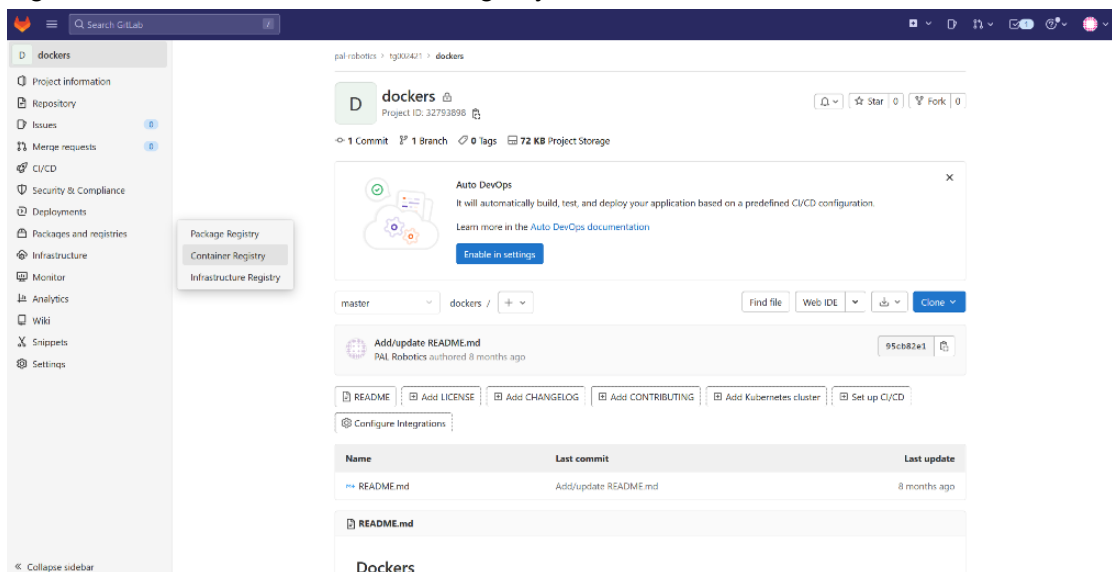
In order to pull the container, you must have an account in gitlab.com. The technical team of PAL Robotics will create and share with you access to a repository in a group for your organization where you will be able to pull the development PC containers from.

This operation has to be done only once every time the development PC container is updated.

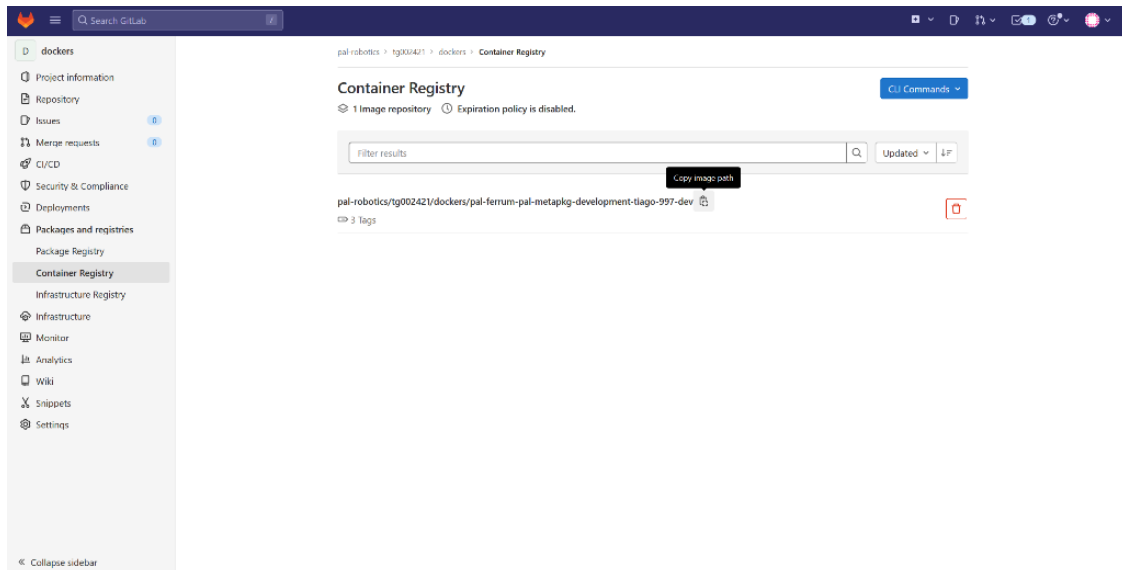
With the Docker desktop running, access the command line tool of windows



1. Login to the docker daemon of gitlab.com by doing `docker login registry.gitlab.com`
And enter your username and password of your gitlab.com account.
2. In the shared repository of dockers, on the left hand side, go to “Packages and registries”, and choose “Container registry”



3. Click the copy icon to copy the image path



4. Back in the command line tool, type `docker pull <copied Image Path>`

```

C:\> Símbolo del sistema

Microsoft Windows [Versión 10.0.22000.856]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\narcismiguel>docker login registry.gitlab.com
Authenticating with existing credentials...
Login Succeeded

C:\Users\narcismiguel>docker pull registry.gitlab.com/pal-robotics/tg002421/dockers/pal-ferrum-pal-metapkg-development-tiago-997-dev
Using default tag: latest
latest: Pulling from pal-robotics/tg002421/dockers/pal-ferrum-pal-metapkg-development-tiago-997-dev
5d9821c94847: Already exists
a610eae58dfc: Already exists
a40e0eb9f140: Already exists
0656e15ffcec: Pull complete
56abdd813bf1: Pull complete
Digest: sha256:5ff759b3a48b97e844a6a89f72bbcf189a090b6e318c2a354982bc5cd7cfb8ef
Status: Downloaded newer image for registry.gitlab.com/pal-robotics/tg002421/dockers/pal-ferrum-pal-metapkg-development-tiago-997-dev:latest
registry.gitlab.com/pal-robotics/tg002421/dockers/pal-ferrum-pal-metapkg-development-tiago-997-dev:latest

C:\Users\narcismiguel>

```

5. Now you can check that this is available by doing *docker images*

```
C:\Users\narcismiguel>docker images
REPOSITORY                                IMAGE ID      CREATED      SIZE      TAG
registry.gitlab.com/pal-robotics/touchlab/dockers/pal-ferrum-pal-metapkg-development-tiago-dual-touchlab-dev  3eb97fd2a409  46 hours ago  6.15GB    latest
ubuntu                                    2dc39ba059dc  12 days ago  77.8MB    latest
registry.gitlab.com/pal-robotics/tg002421/dockers/pal-ferrum-pal-metapkg-development-tiago-997-dev  0292aff1c270  4 weeks ago  4.17GB    latest
docker101tutorial                        f35ff86719a9  7 weeks ago  28.9MB    latest
alpine/git                               22a2874e1112  8 weeks ago  39.5MB    latest
registry.gitlab.com/pal-robotics/cyberselves/dockers/pal-ferrum-pal-metapkg-development-tiago-dual-171-dev  d65bc6887437  2 months ago  6.14GB    latest
nvidia/cuda                              d65bc6887437  2 months ago  6.14GB    latest
se-ubuntu20.04                          a347d6e418ed  3 months ago  126MB     11.4.2-ba
C:\Users\narcismiguel>
```

Running the container

Make sure the two steps in the “Generic Procedure” section have been done. Enter the command line tool of Windows, and run

```
docker run -it --name docker_test -e DISPLAY=host.docker.internal:0  
registry.gitlab.com/pal-robotics/tg002421/dockers/pal-ferrum-pal-metapkg-development-tiago  
-997-dev:latest
```

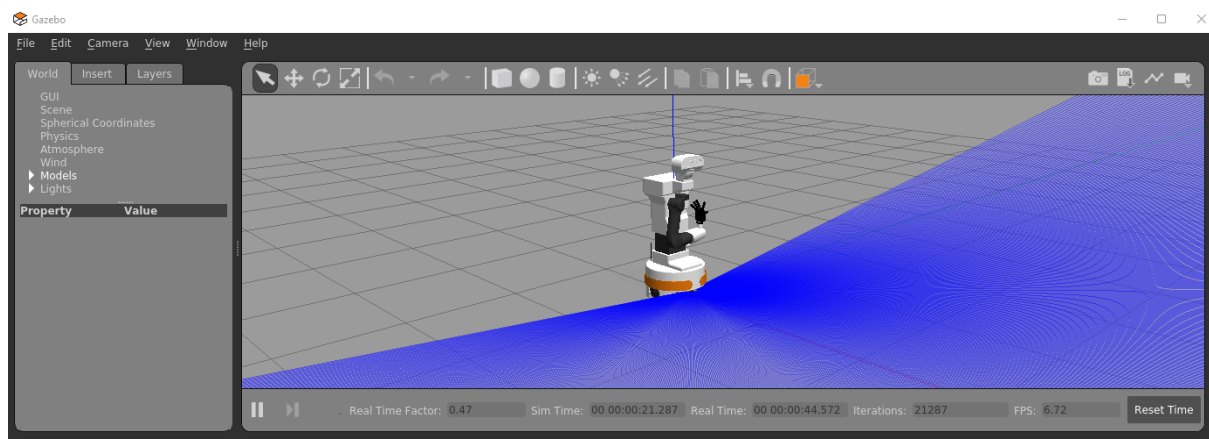
1. *docker run* is the command to run a container of an image
2. *-it* refers that this is done in interactive mode
3. *--name docker_test* prescribes a unique name to this container that will allow you to re-open it later on if you made any change. This is optional, and in case you do not prescribe any name a random one will be assigned.
4. *-e DISPLAY=host.docker.internal:0* exports the display so that you can have a graphical interface when running the docker, and
5. *registry.gitlab.com/pal-robotics/tg002421/dockers/pal-ferrum-pal-metapkg-development-tiago-997-dev:latest* is the name of the image, followed by *:* and by the tag that you want to run. The same image can have different tags

This will prompt you to a Linux terminal.

```
C:\Users\narcismiguel>docker run -it --name docker_test -e DISPLAY=host.docker.internal:0 registry.gitlab.com/pal-roboti  
cs/tg002421/dockers/pal-ferrum-pal-metapkg-development-tiago-997-dev:latest  
Associating docker user user to host's user with UID 1000  
usermod: no changes  
Changing group name to host's  
usermod: no changes  
user@b9cad2ce58e7:/$
```

To use any ROS-related software, you have to source it first, by doing
source /opt/pal/ferrum/setup.bash

After this, you can run, for instance, the TIAGo simulation by doing
roslaunch tiago_gazebo tiago_gazebo.launch



Terminator is installed in the image also, so if you do
terminator -u &

A terminator terminal will appear, where you can do the usual Ctrl+E or Ctrl+O to split the terminal, Ctrl+X to maximize, etc.

Running the container with NVIDIA GPU

This option requires you to have the last version of either Windows 10 or Windows 11. Also needs to have WSL2 (Windows Subsystem Linux, v2).

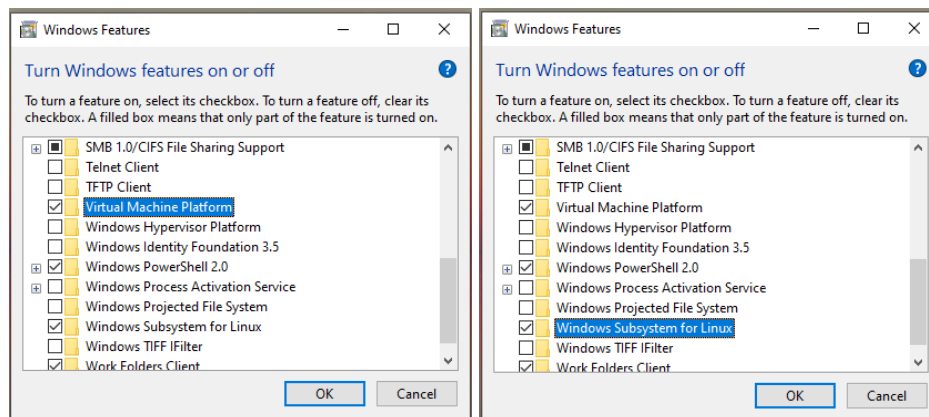
Note: This only works for dockers of Ubuntu 20.04. In PAL Robotics, these are labelled with the keyword *gallium*.

Prepare your machine

In order to use GPU acceleration you have to make sure that the following has been done

1. Update Nvidia drivers, from [this link](#).
2. Enable Windows features:

The *Virtual Machine Platform* and *Windows Subsystem for Linux* should be on.



Export the docker container a

The goal here is to export the container in such a way that it can be run from wsl directly. To follow this step you should have pulled a gallium (Ubuntu 20.04) docker.

1. Open a command shell and run the docker container
`docker run --name pal_dock registry.gitlab.com/pal-robotics/cesi/dockers/developer-gallium-tiago-dual`
This will start the docker, not in interactive mode
2. Export the docker to a file
`docker export -o gallium.tar.gz pal_dock`
3. Import it to wsl as distributions of the subsystem
`wsl --import gallium .\gallium .\gallium.tar.gz`

Note: This step may take a long time

After this, if you do

```
wsl -l -v
```

You will see something like

```
C:\Users\narcismiguel>wsl -l -v
```

NAME	STATE	VERSION
* docker-desktop	Running	2
developer-gallium-tiago-dual	Running	2
docker-desktop-data	Running	2

Where one of the subsystem distros will be called 'gallium'

To run it, do

```
wsl -u user -d gallium
```

Once in, you have to export the display. Do

1. `cd`
2. `echo "export DISPLAY=$(ip route list default | awk '{print $3}'):0" >> .bashrc`
3. `source .bashrc`

And now you can open new terminals using terminator, and even run the TIAGo dual simulation:

4. `terminator -u`
5. `source /opt/pal/gallium/setup.bash`
6. `roslaunch tiago_dual_gazebo tiago_dual_gazebo.launch`