

Edge Insights for Autonomous Mobile Robots (EI for AMR) Robot Get Started Guide

Contents

Chapter 1: Edge Insights for Autonomous Mobile Robots Robot Kit

Requirements	3
Step 1: Prepare the Target System	4
Step 2 (Optional) Intel® GPU Support	4
Step 3: Download and Copy the edge_insights_for_amr.zip File to the Target System	5
Step 4: Extract the Edge Insights for Autonomous Mobile Robots Software	6
Step 5: Install the Edge Insights for Autonomous Mobile Robots Software	7
Step 6. Run a ROS 2 Sample Application in the Docker* Container	12
Deepen Your Knowledge	14
Troubleshooting	14
Terminology	18
Notices and Disclaimers	20

Edge Insights for Autonomous Mobile Robots Robot Kit

1

The steps in this guide explain how to install the robot kit and run a sample application. For robot orchestration, go to: [Get Started Guide for Robot Orchestration](#).

For more information about complex scenarios, advanced features, and debugging, see the [Developer Guide](#).

The currently supported versions are:

- Base OS: Ubuntu* 20.04 LTS
- ROS 2 with data distribution service: Foxy
- OpenVINO™: 2021.4
- Intel® oneAPI Base Toolkit: 2021.4
- Intel® RealSense™ SDK: v2.50
- Simulation: Gazebo* v11.8.1 + Agile Robotics for Industrial Automation Competition (ARIAC) world

EI for AMR is delivered as a compressed .zip file that is compatible with the operating system you selected during the download. The .zip contains a binary executable file, a manifest file that lists the modules to be installed, and a configuration file (config.ini).

See [Troubleshooting](#) if you run into problems installing the software.

Requirements

In addition to the [Product Download](#), you must meet the following requirements.

Target System for the Robot Base Kit

- Intel® CPU Processors:
 - Intel® Atom® processor with Intel® SSE4.1 support
 - Intel® Pentium® processor N4200/5, N3350/5, N3450/5 with Intel® HD Graphics
 - 6th Generation or newer Intel® Core™ processors
- 8 GB RAM
- 64 GB hard drive
- Intel® RealSense™ camera D435i
- Accelerator: Intel® Movidius™ Myriad™ X VPU (optional)
- [IOT Ubuntu* Desktop 20.04](#)
- Slamtec* RPLIDAR A3 2D LIDAR (optional)

NOTE

Intel does not recommend running simulations, like Gazebo*, on a robot.
Intel does not recommend compilation on robots with 8 GB of RAM.
For development systems and running simulations, see the requirements below.

Target System for Development and Simulations

- Intel® CPU Processors:
 - 11th Generation Intel® Core™ processors with integrated GPU Intel® Iris® Xe® Graphics or Intel® UHD Graphics

- 10th Generation Intel® Core™ processors with integrated GPU Intel® UHD Graphics
- 16 GB RAM
- 128 GB hard drive
- Intel® RealSense™ camera D435i
- Accelerator: Intel® Movidius™ Myriad™ X VPU (optional)
- [IOT Ubuntu* Desktop 20.04](#)
- Slamtec* RPLIDAR A3 2D LIDAR (optional)

Knowledge/Experience

- You are familiar with executing Linux* commands.
- You have basic Docker* experience.
- ROS 1 or ROS 2 background recommended.

Step 1: Prepare the Target System

Make sure your target system has a fresh installation of Ubuntu* Linux* that corresponds to the version of Edge Insights for Autonomous Mobile Robots (EI for AMR) that you downloaded. If you need help installing Ubuntu*, follow these steps:

1. Download Ubuntu's [Ubuntu for Intel IoT platforms](#) ISO file to your developer workstation.
2. Create a bootable flash drive using an imaging application, such as Startup Disk Creator, available on Ubuntu*.
3. After flashing the USB drive, power off your target system, insert the USB drive, and power on the target system.
If the target system doesn't boot from the USB drive, change the boot priority in the system BIOS.
4. Follow the prompts to install Ubuntu* with the default configurations. For detailed instructions, see the [Ubuntu* guide](#).

NOTE When creating your login details, do not use `eiforamr` as a username because it is used in the Automated Mobile Robots Docker images. If the system has this username, it will crash when it tries to open AMR Docker images using this command: `./run_interactive_docker.sh eiforamr-full-flavour-sdk:<TAG> eiforamr`

5. Power down your target system and remove the USB drive.
6. Power up the target system. You see Ubuntu* Desktop is successfully installed.

Step 2 (Optional) Intel® GPU Support

If your system has an Intel® GPU, you can improve the simulation experience by making configuration changes. GPU support can speed up larger Gazebo* simulations noticeably.

To enable GPU support, perform the following steps:

1. Check if your system has a GPU from Intel®.

```
sudo lshw -c video |grep driver
```

Examples of output for an Intel® Iris® Xe Graphics GPU:

```
configuration: depth=32 *driver=i915* latency=0 mode=1280x1024 visual=truecolor xres=1280 yres=1024
```

2. Check that the drivers are working and enabled:

```
sudo apt install mesa-utils  
glxinfo| grep -i "opengl renderer"
```

Examples of output:

- a. For 10th generation Core microprocessors, Comet Lake GPU:
OpenGL renderer string: Mesa Intel(R) UHD Graphics 630 (CML GT2)
 - b. For 11th generation Core microprocessors, Tiger Lake GPU:
OpenGL renderer string: Mesa Intel(R) Xe Graphics (TGL GT2)
 - c. For GPU support missing: (pure slow software rendering)
OpenGL renderer string: llvmpipe (LLVM 12.0.0m 256 bits)
3. If the system reports llvmpipe instead of "UHD Graphics" or "Xe Graphics" then:
- a. Check your connection type:

```
echo $DISPLAY
```

If it is not ":0" then you may be using a VNC.
Check below for working configurations.

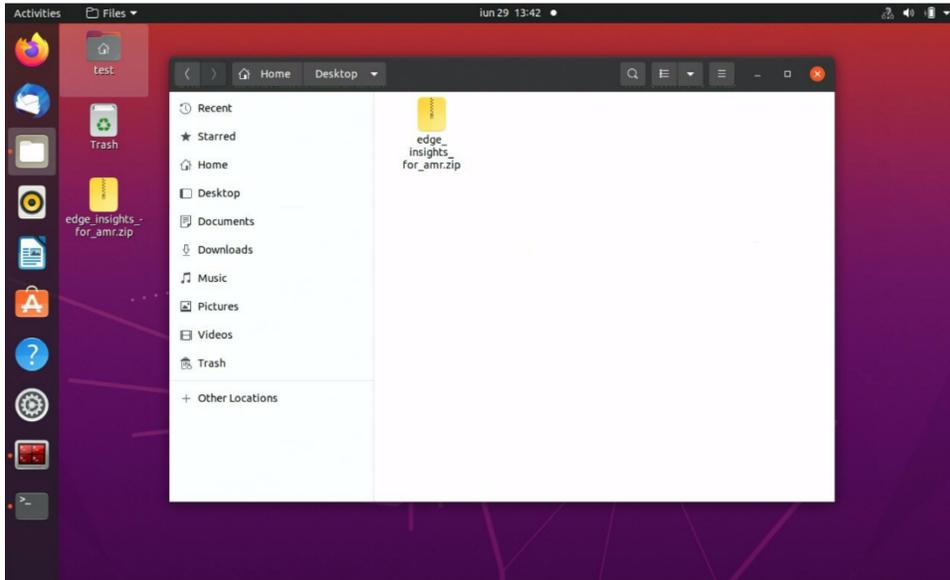
- b. Software and Hardware environments found to be working:
 - gnome desktop
 - direct access to the system via a monitor
 - remote access using KVM over IP
 - remote access using x11vnc, without any headless configurations
- c. Software and Hardware environments found not to be working:
 - XFCE desktop, Wayland desktop
 - headless x11vnc config
 - tigerVNC
- d. If the system has integrated GPU but it is still not recognized, try:
 - Connect a monitor (or graphical display hardware).
 - Remove any headless config (if existing):

```
sudo rm -rf /etc/X11/xorg.conf
sudo rm -rf /usr/share/X11/xorg.conf.d/xorg.conf
sudo mv ~/.Xauthority ~/.Xauthority.backup
sudo apt install --reinstall xserver-xorg-video-intel
sudo apt-get install --reinstall xserver-xorg
sudo apt-get install --reinstall x11vnc
sudo systemctl enable x11vnc
sudo apt install --reinstall gdm3 ubuntu-desktop gnome-shell
sudo systemctl reboot
```

Step 3: Download and Copy the edge_insights_for_amr.zip File to the Target System

1. Download the latest release.
 - a. Go to the [Product Download](#) page.
 - b. Select one of the use case bundles:
 - Robot Complete Kit (contains the Robot Base Kit plus the images needed for simulation, tools, and development)
 - Robot Base Kit (optimized, minimal file size for installation on a robot)
 - UP Xtreme i11 Robotic Kit (contains only the Docker* images needed for the UP Xtreme i11 Robotic Kit, no development images.)
 - c. Verify the components to be downloaded for your selected bundle by checking the **Available with your selection** section. If incorrect components are listed, refresh the webpage, and re-select the bundle you want to download.
 - d. Click **Download Recommended Configuration**.

2. Copy `edge_insights_for_amr.zip` from the developer workstation to the Home directory on your target system. You can use a USB flash drive to copy the file.

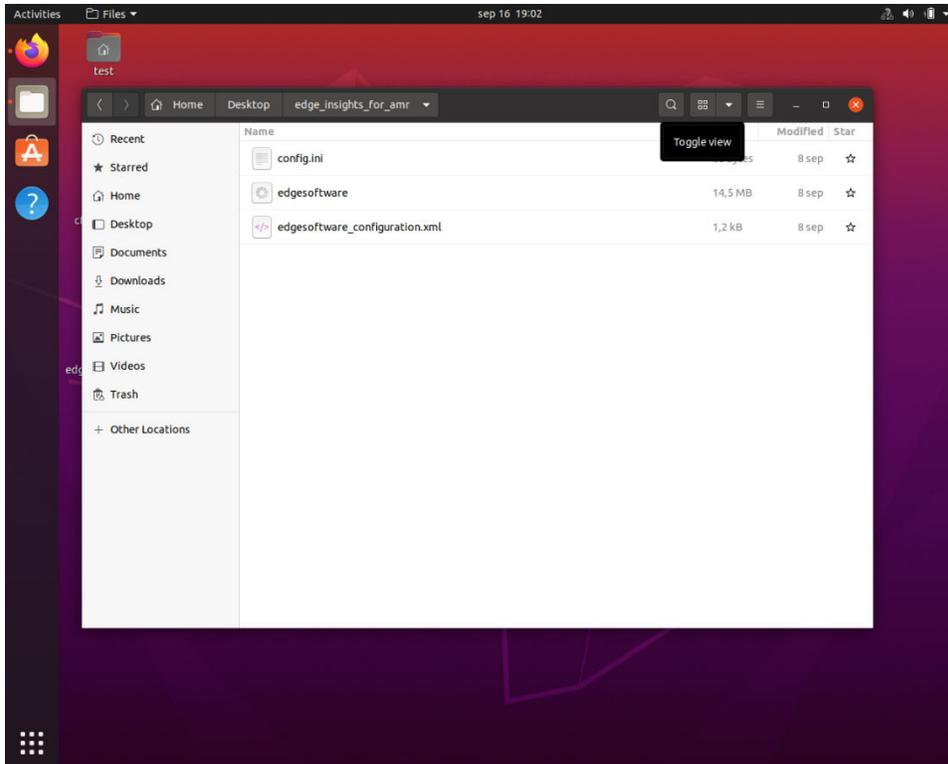


Step 4: Extract the Edge Insights for Autonomous Mobile Robots Software

You need to be on the target system to complete these steps.

1. Make sure you have a working Internet connection.
2. Open a new terminal.
3. Extract the package:

```
unzip edge_insights_for_amr.zip
```



Step 5: Install the Edge Insights for Autonomous Mobile Robots Software

This step runs a script that downloads components and installs Edge Insights for Autonomous Mobile Robots on your target system. The software installation takes about 10 minutes. The completion time depends on your target system and Internet connection.

During the installation, you are prompted to enter your product key. This key is in the email you received from the Intel® Registration Center. Contact [Support Forum](#) if you do not have this email message.

1. If a proxy is required to connect to the Internet, add the proxy settings as described below, updating `<http_proxy>` and `<https_proxy>` to your actual proxy.

- a. Add proxies in `/etc/apt/apt.conf.d/proxy.conf`:

```
echo 'Acquire::http::proxy "http://<http_proxy>:port";' | sudo tee -a /etc/apt/apt.conf.d/proxy.conf
echo 'Acquire::https::proxy "http://<https_proxy>:port";' | sudo tee -a /etc/apt/apt.conf.d/proxy.conf
```

- b. Change the environment.

- Change the environment for all users on the system:

```
sudo su
echo 'export http_proxy="http://<http_proxy>:port"' >> /etc/environment
echo 'export https_proxy="http://<https_proxy>:port"' >> /etc/environment
echo 'export ftp_proxy="http://<ftp_proxy>:port"' >> /etc/environment
echo 'export no_proxy="<no_proxy>"' >> /etc/environment
exit
source /etc/environment
```

NOTE These steps are needed only once per host. They do not have to be done for different users or different logins of the same user.

- Change the environment for the current user only:

```
echo 'export http_proxy="http://<http_proxy>:port"' >> ~/.bashrc
echo 'export https_proxy="http://<https_proxy>:port"' >> ~/.bashrc
echo 'export ftp_proxy="http://<ftp_proxy>:port"' >> ~/.bashrc
echo 'export no_proxy="<no_proxy>"' >> ~/.bashrc
source ~/.bashrc
```

Read and follow best practices as described in this [Linux* wiki about environment variables](#).

Edit the `/etc/sudoers` file with visudo:

```
sudo visudo
# Add after other lines that add Defaults:
Defaults env_keep += "ftp_proxy http_proxy https_proxy no_proxy"
```

2. Run the following commands to go to the directory, change permission of the executable `edgesoftware` file, and install the package:

```
cd edge_insights_for_amr
chmod 775 edgesoftware
sudo groupadd docker
sudo usermod -aG docker $USER
newgrp docker
./edgesoftware install
```

NOTE If the installer is blocked at checking the Internet connection, refer to [Troubleshooting](#). If you encounter any Docker* pull-related issues during the installation process, refer to [Troubleshooting](#).

3. Type the product key at the prompt:

NOTE The Product Key is displayed in the download page.

```

Activities  Terminal  set 14 13:23
intel@edgesoftware: ~/Intel/edge_insights_for_amr
intel@edgesoftware:~/intel/edge_insights_for_amr$ ll
total 14172
drwxrwxr-x 2 test test 4096 set 14 13:02 ./
drwxr-xr-x 3 test test 4096 set 14 12:26 ../
-rw-rw-r-- 1 test test 88 set 14 12:26 config.ini
-rwxrwxr-x 1 test test 14477024 set 14 12:26 edgesoftware*
-rw-rw-r-- 1 test test 1177 set 14 12:26 edgesoftware_configuration.xml
-rwxrwxr-x 1 test test 15386 set 14 12:26 install_prerequisites.sh*
intel@edgesoftware:~/intel/edge_insights_for_amr$ sudo ./edgesoftware install
Please enter the Product Key. The Product Key is contained in the email you received from
fc60-41f6-a251-b529f6dd557a
Starting the setup...
ESB CLI version: 2021.2
Target OS: Ubuntu 20.04
Python version: 3.8.10
Checking Internet connection
Connected to the Internet
Validating product key
Successfully validated Product Key
Checking for prerequisites
All dependencies met
-----SYSTEM INFO-----
Package Name: Edge Insights for Autonomous Mobile Robots 2021.3
Product Name: Intel Corporation Tiger Lake Client Platform
CPU SKU: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
Memory Size: 16 GB
Operating System: Ubuntu 20.04.2 LTS
Kernel Version: 5.11.0-34-generic
Accelerator: None
CPU Utilization: 1.8%
Available Disk Space: 335 GB
Starting installation
Downloading modules...
Downloading component esb_common
Module validation passed for 6022bd8ccc7449002afdbedd
Successfully downloaded module esb_common
Downloading component Docker_Community_Edition_CE
Module validation passed for 605cab935a4b53002c272678
Successfully downloaded module Docker_Community_Edition_CE
Downloading component AMR_containers
Module validation passed for 612f6fa5fc7d8e002b3f357d
Successfully downloaded module AMR_containers
Downloading component FastMapping
Module validation passed for 612f6f5bfc7d8e002b3f357b
Successfully downloaded module FastMapping
Downloading component Wandering
Module validation passed for 612f6f3ffc7d8e002b3f3579
Successfully downloaded module Wandering
Downloading component AMR_Simulations
Module validation passed for 61376241fc7d8e002b3f35a5
Successfully downloaded module AMR_Simulations
Downloading modules completed...

```

- Based on components selected and system configuration, you might be prompted for additional actions. For example, if your system is behind a proxy, you are asked to enter proxy settings.

When the installation is complete, you see the message `Installation of package complete` and the installation status for each module.

```
Activities Terminal set 14 13:41 intel@edgesoftware: ~/intel/edge_insights_for_amr

CPU Utilization: 1.8%
Available Disk Space: 335 GB
Starting installation
Downloading modules...
Downloading component esb_common
Module validation passed for 6022bd8ccc7449002afdbedd
Successfully downloaded module esb_common
Downloading component Docker_Community_Edition_CE
Module validation passed for 605cab935a4b53002c272678
Successfully downloaded module Docker_Community_Edition_CE
Downloading component AMR_containers
Module validation passed for 612f6fa5fc7d8e002b3f357d
Successfully downloaded module AMR_containers
Downloading component FastMapping
Module validation passed for 612f6f5bfc7d8e002b3f357b
Successfully downloaded module FastMapping
Downloading component Wandering
Module validation passed for 612f6f3ffc7d8e002b3f3579
Successfully downloaded module Wandering
Downloading component AMR_Simulations
Module validation passed for 61376241fc7d8e002b3f35a5
Successfully downloaded module AMR_Simulations
Downloading modules completed...
Installing shared module 'esb_common'
Unzipping the shared module 'esb_common'...
running install
running bdist_egg
running egg_info
creating esb_common.egg-info
writing esb_common.egg-info/PKG-INFO
writing dependency_links to esb_common.egg-info/dependency_links.txt
writing top-level names to esb_common.egg-info/top_level.txt
writing manifest file 'esb_common.egg-info/SOURCES.txt'
reading manifest file 'esb_common.egg-info/SOURCES.txt'
writing manifest file 'esb_common.egg-info/SOURCES.txt'
installing library code to build/bdist.linux-x86_64/egg
running install_lib
warning: install_lib: 'build/lib' does not exist -- no Python modules to install

creating build
creating build/bdist.linux-x86_64
creating build/bdist.linux-x86_64/egg
creating build/bdist.linux-x86_64/egg/EGG-INFO
copying esb_common.egg-info/PKG-INFO -> build/bdist.linux-x86_64/egg/EGG-INFO
copying esb_common.egg-info/SOURCES.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
copying esb_common.egg-info/dependency_links.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
copying esb_common.egg-info/top_level.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
zip_safe flag not set; analyzing archive contents...
creating dist
creating 'dist/esb_common-0.1-py3.8.egg' and adding 'build/bdist.linux-x86_64/egg' to it
removing 'build/bdist.linux-x86_64/egg' (and everything under it)
Processing esb_common-0.1-py3.8.egg
Removing /usr/local/lib/python3.8/dist-packages/esb_common-0.1-py3.8.egg
```

```
Activities Terminal sep 23 15:29 intel@edgesoftware: ~/edge_insights_for_amr

creating esb_common.egg-info
writing esb_common.egg-info/PKG-INFO
writing dependency_links to esb_common.egg-info/dependency_links.txt
writing top-level names to esb_common.egg-info/top_level.txt
writing manifest file 'esb_common.egg-info/SOURCES.txt'
reading manifest file 'esb_common.egg-info/SOURCES.txt'
writing manifest file 'esb_common.egg-info/SOURCES.txt'
installing library code to build/bdist.linux-x86_64/egg
running install_lib
warning: install_lib: 'build/lib' does not exist -- no Python modules to install

creating build
creating build/bdist.linux-x86_64
creating build/bdist.linux-x86_64/egg
creating build/bdist.linux-x86_64/egg/EGG-INFO
```

5. Verify that all Docker* images were downloaded:

```
docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
amr-nav2 2022.2 d8b47598f285 22 hours ago 4.15GB
eiforamr-full-flavour-sdk 2022.2 add353b53ad6 22 hours ago 31.2GB
amr-object-detection 2022.2 dd8c57f22507 23 hours ago 10.1GB
.
.
```

NOTE Installation failure logs are available at: `/var/log/esb-cli/Edge_Insights_for_Autonomous_Mobile_Robots_<version>/<Component_name>/install.log`

Step 6. Run a ROS 2 Sample Application in the Docker* Container

This step tests your install.

After this step, to deepen your knowledge about complex scenarios, advanced features, and debugging, see the [Developer Guide](#).

1. Go to the `AMR_containers` folder:

```
cd <edge_insights_for_amr_path>/Edge_Insights_for_Autonomous_Mobile_Robots_*/AMR_containers
```

2. Prepare the environment setup:

```
source 01_docker_sdk_env/docker_compose/05_tutorials/config/docker_compose.source
export CONTAINER_BASE_PATH=`pwd`
export ROS_DOMAIN_ID=12
```

3. Run an automated yml file that opens a ROS 2 sample application inside the EI for AMR Docker* container.

```
CHOOSE_USER=eiforamr docker-compose -f 01_docker_sdk_env/docker_compose/05_tutorials/
turtlesim.demo.yml up
```

4. Go to **Plugins > Services > Service Caller**: Choose to move turtle1 by choosing (from the Service drop-down list) `/turtle1/teleport_absolute` and make sure you changed x and y coordinates for the original values. Press **Call**. The turtle should move. Close the service caller window by pressing **x**. Then type `Ctrl-c`.

- Run this command in another terminal:

```
CHOOSE_USER=eiforamr docker-compose -f 01_docker_sdk_env/docker_compose/05_tutorials/turtlesim.demo.yml down
```

6. For an explanation of what happened, open the yml file:
 - The first 23 lines are from the EI for AMR infrastructure.
 - Line 26 starts the `turtlesim` ROS 2 node.
 - Line 31 starts the `rqt` so that the turtle can be controlled.

Deepen Your Knowledge

For more information about complex scenarios, advanced features, and debugging, see the [Developer Guide](#).

Troubleshooting

If you're unable to resolve your issues, go to the [Support Forum](#).

Installation Failure Due to bad system date

Make sure you that your system has the correct date:

```
date
```

If the date is incorrect, contact your local support team for help setting the correct date and time.

Proxies Error

If you set up proxies using `export`, `sudo` does not know about them. You must use `sudo -E ./edgesoftware` to install the package:

```
sudo -E ./edgesoftware
```

Installation Failure Due to Lost Internet Connection

Make sure you have an active Internet connection during the full installation. If you lose Internet connectivity at any time, the installation might fail.

Ubuntu* Errors

Make sure you are using a fresh Ubuntu* installation. Earlier software, especially Docker* and Docker* Compose, can cause issues.

Rate Limit Error

For the following error message, go to [Docker* hub download rate limit](#).

```
docker: Error response from daemon: toomanyrequests: You have reached your pull rate limit.
```

Credential Errors

Example credential errors:

```
Error saving credentials: error storing credentials - err: exit status 1, out: `Error calling StartServiceByName for org.freedesktop.secrets: Timeout was reached`
```

```
ERROR: rpc error: code = Unknown desc = error getting credentials - err: exit status 1, out:
`Error calling StartServiceByName for org.freedesktop.sec
rets: Timeout was reached`
```

To fix, use the working solution from this [thread](#):

```
sudo apt update
sudo apt -V install gnupg2 pass
```

https Communication Errors

For errors related to https communication, make sure that DNS filtering is not filtering out sites. For example, the edgessoftware installer uses the site ip-api.com to get geographic locations. If this site is filtered out by DNS filtering, you get this error:

```
requests.exceptions.ConnectionError: HTTPConnectionPool(host='ip-api.com', port=80): Max retries
exceeded with url: /json/ (Caused by NewConnectionError('<urllib3.connection.HTTPConnection
object at 0x7f8ffc530400>: Failed to establish a new connection: [Errno -2] Name or service not
known'))
```

To fix, do not block ip-api.com through DNS filtering or firewalls.

Installation as Different Users Error

If EI for AMR was installed on the same system using a different user than the current user, you may encounter an error when doing Step 7: Install the Edge Insights for Autonomous Mobile Robots Software.

The error is similar to:

```
Successfully installed eiforamr-full-flavour-sdk took 32 minutes 49.44 seconds
Traceback (most recent call last):
File "PyInstaller/loader/pyimod04_ctypes.py", line 54, in __init__
File "ctypes/__init__.py", line 373, in __init__
OSError: /opt/intel/lanternrocksdk-linux-3.0.90/native/lib/libstdc++/libstdc++.so: cannot open
shared object file: Permission denied
The above exception was the direct cause of the following exception:
Traceback (most recent call last):
File "edgesoftware/edgesoftware.py", line 536, in <module>
File "click/core.py", line 1128, in __call__
File "click/core.py", line 1053, in main
File "click/core.py", line 1659, in invoke
File "click/core.py", line 1395, in invoke
File "click/core.py", line 754, in invoke
File "edgesoftware/edgesoftware.py", line 101, in install
File "edgesoftware/functions.py", line 850, in setup_start
File "edgesoftware/functions.py", line 742, in run_installation
File "edgesoftware/common/utils.py", line 1052, in send_LR_data
File "edgesoftware/common/utils.py", line 2490, in import_LR_helper
File "ctypes/__init__.py", line 451, in LoadLibrary
File "PyInstaller/loader/pyimod04_ctypes.py", line 56, in __init__
pyimod04_ctypes.PyInstallerImportError: Failed to load dynlib/dll '/opt/intel/lanternrocksdk-
linux-3.0.90/native/lib/libstdc++/libstdc++.so'. Most probably this dynlib/dll was not found
when the application was frozen.
[3897] Failed to execute script 'edgesoftware' due to unhandled exception!
```

The installation process will complete even if you encounter this issue.

To avoid this error, change /opt/intel ownership to the current user as shown below:

```
sudo chown -R intelamr:intelamr /opt/intel/
```

NOTE In the above example, the current user is `intelamr` and its group is `intelamr`. Modify the command with your user and group details.

Downloading Incorrect Components Error

There is a known issue with the list of images displayed in **Available with your selection** being different than expected.

If you select the **Server Complete Kit** and then select the **Robot Complete Kit**, the displayed text is for the **Robot and Sever Complete Kit**. Also, the incorrect components you see in the **Available with your selection** section are downloaded when you install the release.

To fix this, refresh the webpage, and re-select the bundle you want to download.

This guide lists common troubleshooting tips for robot tutorials.

Permission Denied Error

For a permission denied error when running a script:

```
$ ./run_interactive_docker.sh eiforamr-full-flavour-sdk:<TAG> eiforamr
bash: ./run_interactive_docker.sh Permission denied
```

Give executable permission to the script:

```
chmod 755 run_interactive_docker.sh
```

DISPLAY Environment Variable Error

For errors related to the `DISPLAY` environment variable when trying to open the Docker container or a GUI application, enter the command:

```
echo $DISPLAY
```

If this variable is empty, it will cause issues when opening applications that need GUI.

The most common solution is to give it the `0:0` value:

```
export DISPLAY= "0:0"
```

If the connection with the system is via VNC, `DISPLAY` should be already set.

If it is not, find out the value of `DISPLAY` set by `vncserver` and then set the correct value:

For example:

```
ps ax |grep vncserver
/usr/bin/Xtigervnc :42 -desktop ....
/usr/bin/perl /usr/bin/vncserver -localhost no -geometry 1920x1000 -depth 24 :42

export DISPLAY= ":42"
```

Use `ROS_DOMAIN_ID` to Avoid Interference in ROS Messages

A typical method to demonstrate a use case requires you to start a container (or group of containers) and exchange ROS messages between various ROS nodes. However, interference from other ROS nodes can disrupt the whole process. For example, you might receive ROS messages from unknown nodes that are not intended for the demo use case. These other nodes could be on the same host machine or on other host machines within the local network. In this scenario, it can be difficult to debug and resolve the interference.

You can avoid this by declaring `ROS_DOMAIN_ID` as a fixed numeric value per use case, under the following conditions:

- The `ROS_DOMAIN_ID` should be same for all containers launched for a particular use case.
- The `ROS_DOMAIN_ID` should be an integer between 0 and 101.
- After launching the container, you can declare it with:

```
export ROS_DOMAIN_ID=<value>
```

For more information, go to: [ROS_DOMAIN_ID](#)

To add the `ROS_DOMAIN_ID`, you can choose any of the below options.

1. Add it in the `common.yml` file for all containers:

```
# In file 01_docker_sdk_env/docker_compose/common/common.yml
# ROS_DOMAIN_ID can be added that applies to all use cases
services:
common:
  environment:
    ROS_DOMAIN_ID: <choose ID>
```

2. Add it in the `.env` file for all containers:

```
# In file 01_docker_sdk_env/docker_compose/01_amr/.env
# add below line and provide ROS_DOMAIN_ID
ROS_DOMAIN_ID=<choose ID>
```

3. Add it in the specific `yml` file for a specific use case for specific targets:

```
# In the below example, ROS_DOMAIN_ID is added in ros-base target
# For any use case where this target is used, the ROS_DOMAIN_ID will be set to the given value.

services:

  ros-base:
    image: ${REPO_URL}amr-ubuntu2004-ros2-foxy-ros-base:${DOCKER_TAG:-latest}
    container_name: ${CONTAINER_NAME_PREFIX:-amr-sdk-}ros-base
    environment:
      ROS_DOMAIN_ID: <choose ID>
    env_file:
      - ./env
    extends:
```

4. Add it in the specific `yml` file in the `command:` section and apply only after launching the containers:

```
# In file 01_docker_sdk_env/docker_compose/05_tutorials/
fleet_mngmnt_with_low_battery.up.tutorial.yml
# In the below example, ROS_DOMAIN_ID is set to 58
# You may change it to any new value as per use case requirement.
services:

battery_bridge:
  image: ${REPO_URL}amr-ubuntu2004-ros2-foxy-battery_bridge:${DOCKER_TAG:-latest}
  container_name: ${CONTAINER_NAME_PREFIX:-amr-sdk-}battery_bridge
  extends:
  file: ../01_amr/amr-sdk.all.yml
  service: ros-base
  volumes:
    - /dev/battery_bridge:/dev/battery_bridge:rw
  build:
    target: battery_bridge
  network_mode: host
  restart: "no"
  command:
    - |
      source ros_entrypoint.sh
```

```
source battery-bridge/src/prebuilt_battery_bridge/local_setup.bash
export ROS_DOMAIN_ID=58
sleep 5
ros2 run battery_pkg battery_bridge
```

5. Add it while running a container using the `run_interactive_docker.sh` script:

```
# by adding env parameter, ROS_DOMAIN_ID can be exported inside container:
./run_interactive_docker.sh <image name> <user> --extra_params "-e ROS_DOMAIN_ID=<choose ID>"
```

NOTE You can use any number between 0 and 101 (inclusive), to set `ROS_DOMAIN_ID`, as long as it is not used by a different ROS system.

Be aware that you can also use these options to modify other environment variables.

System HOME Directory Issues

If your test system uses `$HOME` mounted in remote volumes, for example, in a network file system (NFS), you may encounter the error below when you try to run a Docker* image using the `./run_interactive_docker.sh` script:

```
docker: Error response from daemon: error while creating mount source path '/nfs/site/home/<user>': mkdir /nfs/site/home/<user>: file exists.
```

To avoid this, before you run a Docker* image, create a new directory in `/tmp` (or any locally mounted volume), and set `$HOME` to the new path:

```
mkdir /tmp/tmp_home
export HOME=/tmp/tmp_home
./run_interactive_docker.sh eiforamr-full-flavour-sdk:<release_tag> eiforamr
```

Terminology

Term	Description
ADBSCAN	Adaptive Density-Based Spatial Clustering of Applications with Noise
AOTA	Application Over the Air
ARIAC	Agile Robotics for Industrial Automation Competition
CNDA	Corporate Non-Disclosure Agreement
CPU	Central Processing Unit
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DI	Device Initialization Protocol
DL	Deep Learning
DMS	Device Management Service
DPC++	Data Parallel C++

Term	Description
DRM	Deterministic Road Map
EI for AMR	Edge Insights for Autonomous Mobile Robots
DPC++	Data Parallel C++
EOF	end-of-file
FDO	FIDO Device Onboard
FIDO	Fast IDentity Online
FLANN	Fast Library for Approximate Nearest Neighbors
FM	Fast Mapping
GEAR	Gazebo Environment for Agile Robotics
GPU	Graphics Processor Unit
GSLAM	General Simultaneous Localization and Mapping
GUI	Graphical User Interface
IDE	Integrated Development Environment
IE	Inference Engine
IMU	Inertial Measurement Unit
IPU	Image Processing Unit
ITS	Intelligent sampling and Two-way Search
KVM	Kernel-based Virtual Machine
LIDAR	Light Detection and Ranging
MQTT	Message Queuing Telemetry Transport
NFS	Network File System
NN	Neural Network
OSRF	Open Source Robotics Foundation
OTA	Over-The-Air
PCL	Point Cloud Library
PRM	Probabilistic Road Map
RDC	Resource and Documentation Center
RGBD	Red, Green, Blue plus Depth
ROS	Robot Operating System
RPLIDAR	360-degree 2D LIDAR solution developed by SLAMTEC
RPM	Red Hat* Package Manager
RTAB-Map	Real-Time Appearance-Based Mapping

Term	Description
RV	Rendezvous
SDK	Software Development Kit
SDO	Intel® Secure Device Onboard (Intel® SDO)
SLAM	Simultaneous Localization And Mapping
SOTA	Software Over the Air
SSD	Single-Shot multibox Detection
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TMI	Test Module Interface
UEFI	Unified Extensible Firmware Interface
VNC	Virtual Network Computing
vSLAM	Visual Simultaneous Localization and Mapping

Notices and Disclaimers

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All product plans and roadmaps are subject to change without notice.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Your costs and results may vary.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

This software and the related documents are Intel copyrighted materials, and your use of them is governed by the express license under which they were provided to you (License). Unless the License provides otherwise, you may not use, modify, copy, publish, distribute, disclose or transmit this software or the related documents without Intel's prior written permission.

This software and the related documents are provided as is, with no express or implied warranties, other than those that are expressly stated in the License.

