ARNEIS

The B-AROL-O Team

Jan 22, 2024

HOME

1 This is ARNEIS **3**
1.1 ARNEIS in a nutshell . . . 3
1.2 ARNEIS Project sources . . . 3
1.3 How to stay in touch . . . 3

2 Documentation **5**
2.1 ARNEIS - Acronyms . . . 5
2.2 ARNEIS - Bibliography . . . 6
2.3 ARNEIS - Social Media Presence . . . 9
2.4 ARNEIS - System Integration Rel. 1.0 . . . 17

3 Architecture **19**
3.1 ARNEIS - Specification . . . 19
3.2 ARNEIS - System Architecture . . . 21
3.3 ARNEIS - Software Architecture . . . 23

4 How-to Guides **25**
4.1 HOWTO Configure arneis.readthedocs.io . . . 26
4.2 HOWTO Create a landing page on Netlify . . . 35
4.3 HOWTO Configure Google Analytics . . . 42
4.4 HOWTO Install Eclipse Capella . . . 51
4.5 HOWTO Mermaid utilization and code examples . . . 56
4.6 HOWTO Use rclone with Cubbit DS3 . . . 57
4.7 HOWTO Install DepthAI on a Raspberry Pi 4B for the ARNEIS project . . . 67
4.8 HOWTO Install DepthAI Demo v3.0.9 on Windows 10 . . . 70
4.9 HOWTO Train a NN for Object Detection and create a deployable model for the OAK-D-Lite . . . 72
4.10 HOWTO Control a LEGO Powered Up Hub for the ARNEIS Project . . . 77
4.11 HOWTO Install Raspberry Pi OS 64-bit for the ARNEIS project . . . 82
4.12 HOWTO Configure a Static IP Address on Linux . . . 100
4.13 HOWTO Prepare a Raspberry Pi 4B for the ARNEIS project . . . 102
4.14 HOWTO Install MicroPython on an Arduino Nano 33 BLE . . . 113
4.15 HOWTO Getting started with Foundries.io . . . 120
4.16 HOWTO Testing Arduino Portenta X8 . . . 137
4.17 HOWTO Create a VM on Azure for the ARNEIS project . . . 146
4.18 HOWTO Install a K3s cluster for the ARNEIS project . . . 151

5 LEGO® **179**
5.1 LEGO Set 42100 (Liebherr R 9800 Excavator) . . . 180
5.2 Unboxing LEGO® Set 42100 . . . 181
5.3 LEGO® MOC studies . . . 237

5.4 OAK-D-Lite LEGO® mechanical adapter 240
5.5 LEGO® MOCs Work in Progress 245
5.6 LEGO® ARNEIS Project 250

6 Indices and tables **253**

The **ARNEIS** (short for `A` utomated `R` ecognizer, `N` etwork - `E` nabled, `I` tem `S` orter) is the winner of the Popular Vote of the OpenCV Spatial AI Contest sponsored by Intel® and Microsoft Azure.

CHAPTER

ONE

THIS IS ARNEIS

1.1 ARNEIS in a nutshell

ARNEIS aims at reproducing in scale a packaging machine for the Industry-4.0.

This goal is achieved by means of a combination of:

- An OAK-D-Lite Spatial AI camera
- A lot of LEGO® Technic parts
- … and plenty of Open Source software

If you want to know some history about this project please watch the webinar we gave on 2022-03-17:

1.2 ARNEIS Project sources

The source files for the software programs, the LEGO® MOC as well as the documentation site are maintained in the GitHub repository at github.com/B-AROL-O/ARNEIS.

Whenever the `main` branch of the git repository is updated, this site will be updated accordingly.

1.3 How to stay in touch

The ARNEIS project roadmap is maintained on GitHub.

Gianpaolo Macario publishes regular updates of the ARNEIS project on his personal blog.

You may also follow twitter.com/baroloteam to get notified about the progress of the project.

Please report bugs and feature requests on https://github.com/B-AROL-O/ARNEIS/issues, or DM the B-AROL-O Team on Twitter about security issues or other non-public topics.

CHAPTER

TWO

DOCUMENTATION

Here you find reference information used in redaction of the this document, and, if you want, some useful links following the story of the project by social-media publication.

- *ARNEIS - Acronyms*
- *ARNEIS - Bibliography*
- *ARNEIS - Social Media Presence*
- *ARNEIS - System Integration Rel. 1.0*

2.1 ARNEIS - Acronyms

- **AI**: Artificial Intelligence
- **ARNEIS**: Automated Recognizer, Network-Enabled, Items Sorter. See https://github.com/B-AROL-O/ARNEIS
- **B-AROL-O**: See https://github.com/B-AROL-O/
- **BLE**: Bluetooth Low Energy. See https://en.wikipedia.org/wiki/Bluetooth_Low_Energy
- **BT**: Bluetooth®. See https://www.bluetooth.com/
- **CLI**: Command-Line Interface
- **CPU**: Central Processing Unit
- **DNS**: Domain Name System. See https://en.wikipedia.org/wiki/Domain_Name_System
- **FPU**: Floating Point Unit
- **HTTP**: Hypertext Transfer Protocol. See https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- **HW**: Hardware
- **JS**: JavaScript. See https://www.javascript.com/
- **JSON**: JavaScript Object Notation. See https://www.json.org/
- **JWT**: JSON Web Tokens. See https://jwt.io/
- **K3S**: Lightweight Kubernetes. See https://k3s.io/
- **K8S**: Kubernetes. See https://kubernetes.io/
- **K9S**: See https://k9scli.io/
- **LED**: Light Emitting Diode

- **LEGO®**: See https://www.lego.com/
- **LUG**: LEGO® User Group
- **MOC**: My Own Creation
- **MQTT**: See https://mqtt.org/
- **NN**: Neural Network
- **OAK**: OpenCV AI Kit
- **PDF**: Portable Document Format. See https://en.wikipedia.org/wiki/PDF
- **SSH**: Secure Shell Protocol. See https://en.wikipedia.org/wiki/Secure_Shell
- **SW**: Software
- **UML**: Unified Modeling Language. See http://uml.org/
- **UI**: User Interface
- **USB**: Universal Serial Bus. See https://en.wikipedia.org/wiki/USB

2.2 ARNEIS - Bibliography

2.2.1 Computer Vision and AI

Articles and books

- Book: Artificial Intelligence for IoT Cookbook by Michael Roshak. Packt Publishing, March 2021
- Book: OpenCV 4 for Secret Agents - Second Edition by Joseph Howse. Packt Publishing, April 2019
- What the FAQ are AI, ANNs, ML, DL, and DNNs?

Learning Resources

- LearnOpenCV.com hosted by Dr. Satya Mallick
- GitHub repository with C++ and Python examples from LearnOpenCV.com: https://github.com/spmallick/learnopencv

Webinars

- OpenCV Weekly Webinar hosted by Dr. Satya Mallick and Phil Nelson
-

2.2.2 Containers and Kubernetes

- Book: The Kubernetes Bible by Nassim Kebbani, Piotr Tylenda, Russ McKendrick. Packt Publishing, February 2022

2.2.3 LEGO®

- Book: Build and Code Creative Robots with LEGO BOOST by Ashwin Shah. Packt Publishing, November 2021
- Glossario by Italian LEGO® Users Group
- LEGO® Customer Service - Device Check - provided by lego.com
- LEGO® Powered UP Official Homepage
- LEGO® Powered Up basic tutorials - 01 - Introduction & history - RacingBrick, 2021-01-11
- LEGO® Powered Up Connector - Marek's microsite, 2022-09-26
- LEGO® Powered Up simple code block guide - RackingBrick, 2020-04-29
- LEGO® Powered Up extended code block guide - RackingBrick, 2020-04-29
- LEGO® Set 42100 (Liebherr R 9800 Excavator) includes 2x LEGO® Technic® Bluetooth Hub (LEGO® Part bb0961c01).
- Powered Up - A tear down… - Eurobricks Forum, 2018-06-07
- Programming the Lego NXT using Mathematica with Math4NXT by Denis Cousineau
- Smart Robotics with LEGO MINDSTORMS Robot Inventor by Aaron Maurer. Packt Publishing, May 2021

Controlling LEGO via Bluetooth

- **BrickNil**: A Python async library for PoweredUp/Boost
  - GitHub repository: https://virantha.github.io/bricknil
  - Discussion on Eurobricks.com: https://www.eurobricks.com/forum/index.php?/forums/topic/170945-bricknil-a-python-async-library-for-poweredupboost/
- Controlling the Lego Technic Hub (Hub 2) via Bluetooth (btle) using a Raspberry Pi 4B
- Article: Hack Lego Boost with Raspberry Pi - The MagPi Magazine, 2019-04-01
- Tutorial: How to export/import a project in the Powered Up app on Android - RacingBrick, 2020-01-22
- Article: How to Set Up Bluetooth on the Raspberry Pi for a LEGO@reg; Spike - Jason Jurotich on Medium.com, 2020-05-23
- Lego Boost Roboter mit Python steuern - Reutlingen University (language: German; format: PDF, 21 pages)
- **legoBTLE4PI**: A school project that shows how to control a Lego Jeep with Bluetooth on the Raspberrypi 4B.
  - GitHub repository: https://github.com/DietrichChristopeit/legoBTLE4PI
- LEGO Wireless Protocol 3.0.00
- **pyb00st**: Python for LEGO BOOST
  - GitHub repository: https://github.com/JorgePe/pyb00st
- **pylgbst**: Python library to interact with Move Hub / PoweredUp Hubs
  - GitHub repository: https://github.com/undera/pylgbst

  - LibHunt: Pylgbst Alternatives: https://www.libhunt.com/r/pylgbst
- Reverse Engineering the LEGO BOOST Move Hub: https://github.com/JorgePe/BOOSTreveng
- Thread on StackExchange.com: Which programming environments or APIs are available for the LEGO BOOST?

Marketplaces for LEGO® Parts

- BrickEconomy: https://www.brickeconomy.com/
- BrickLink: https://www.bricklink.com/
  - BrickLink API guide
  - Welcome to BrickLink Store API
  - How To: Use the BrickLink API - Episode 1: Creating a CSV Using the API - YouTube Video (37:31) by Just a Brick in The Bucket, 2020-10-05
  - bricklink_api (Python)
  - BSX file handler: BSX file (BrickStock, BrickStore) read/write and conversion with Python

Pybricks

- Pybricks is Python coding for smart LEGO® hubs. Run MicroPython scripts directly on the hub, and get full control of your motors and sensors. Pybricks runs on LEGO® BOOST, City, Technic, MINDSTORMS®, and SPIKE®. You can code using Windows, Mac, Linux, Chromebook, and Android.
- Pybricks Support
  - The future of BLE, Bluetooth, and USB - Issue #262 on pybricks/support
  - Troubleshooting Bluetooth in Pybricks Code - Discussion #270 on pybricks/support

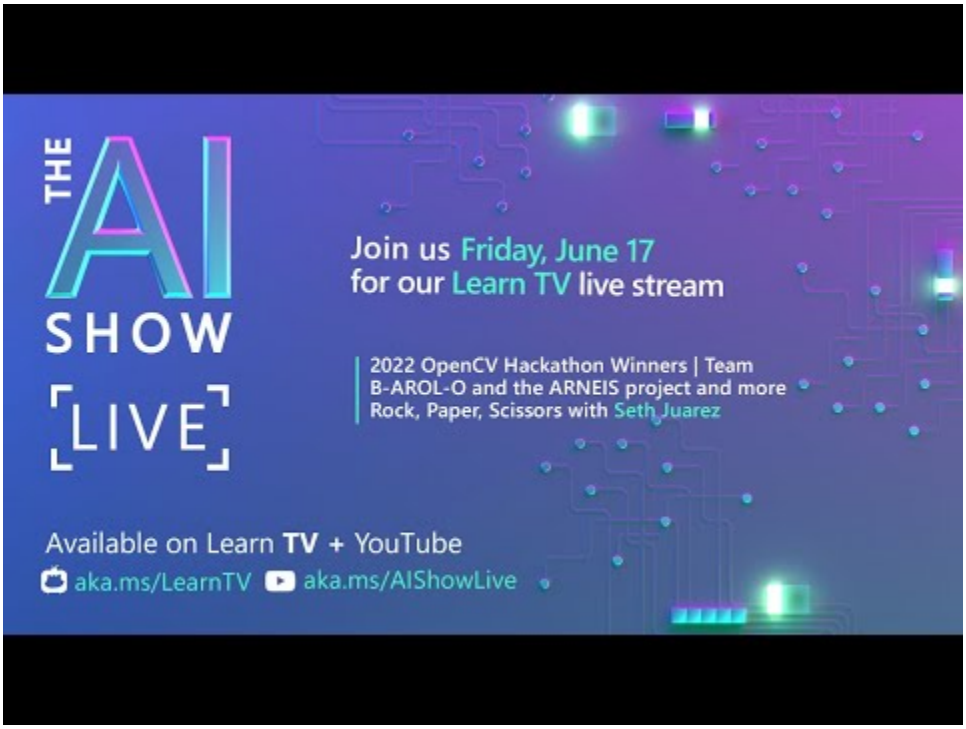
2.2.4 Python and Machine Learning

- Book: Advanced Deep Learning with Python by Ivan Vasilev. Packt Publishing, December 2019
- Free course: Intro to Machine Learning using Microsoft Azure by Udacity in collaboration with Microsoft Azure
- Book: Learn Python Programming - Third Edition by Fabrizio Romano, Heinrich Kruger. Packt Publishing, October 2021
- Tutorial: MicroPython: Program ESP32/ESP8266 using Mu Editor by Random Nerd Tutorials
- Book: Mobile Deep Learning with TensorFlow Lite, ML Kit and Flutter by Anubhav Singh, Rimjhim Bhadani. Packt Publishing, April 2020
- Book: Practical Python Programming for IoT by Gary Smart. Packt Publishing, November 2020
- Book: The Machine Learning Solutions Architect Handbook by David Ping. Packt Publishing, January 2022

2.3 ARNEIS - Social Media Presence

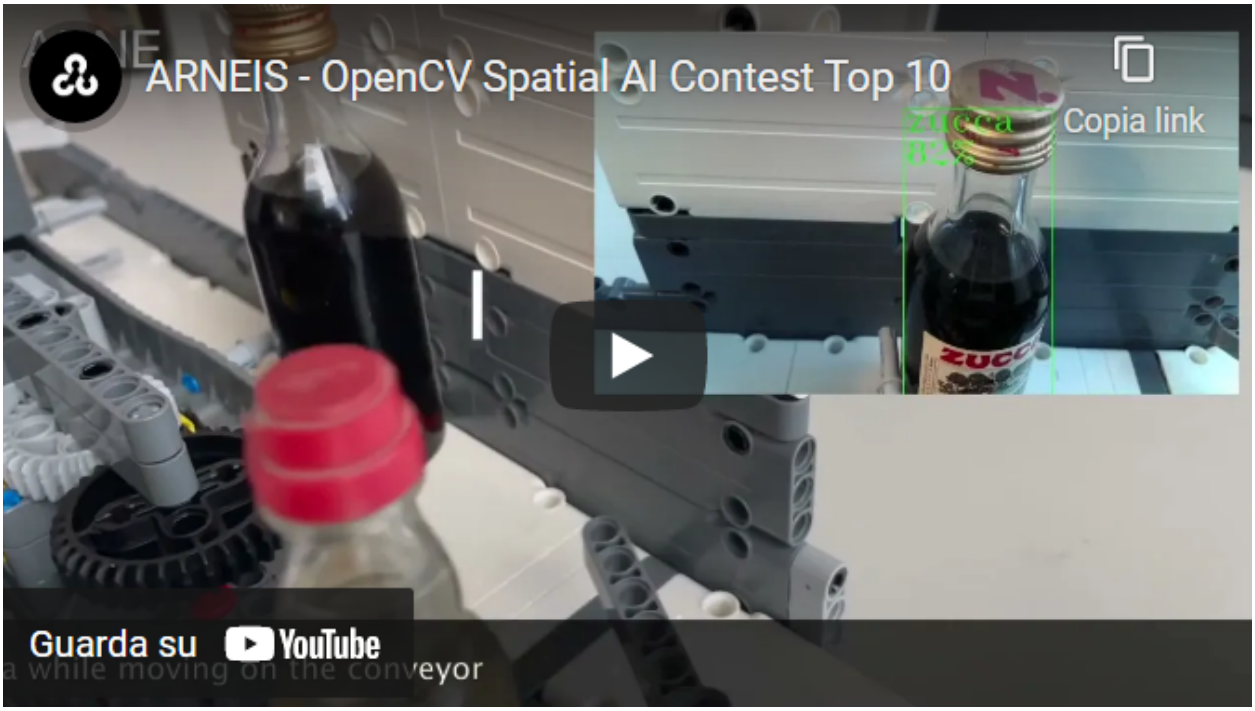
2.3.1 AI Show Live - Episode 59, 2022-06-17

Seth Juarez interview to B-AROL-O Team starts at 13:11



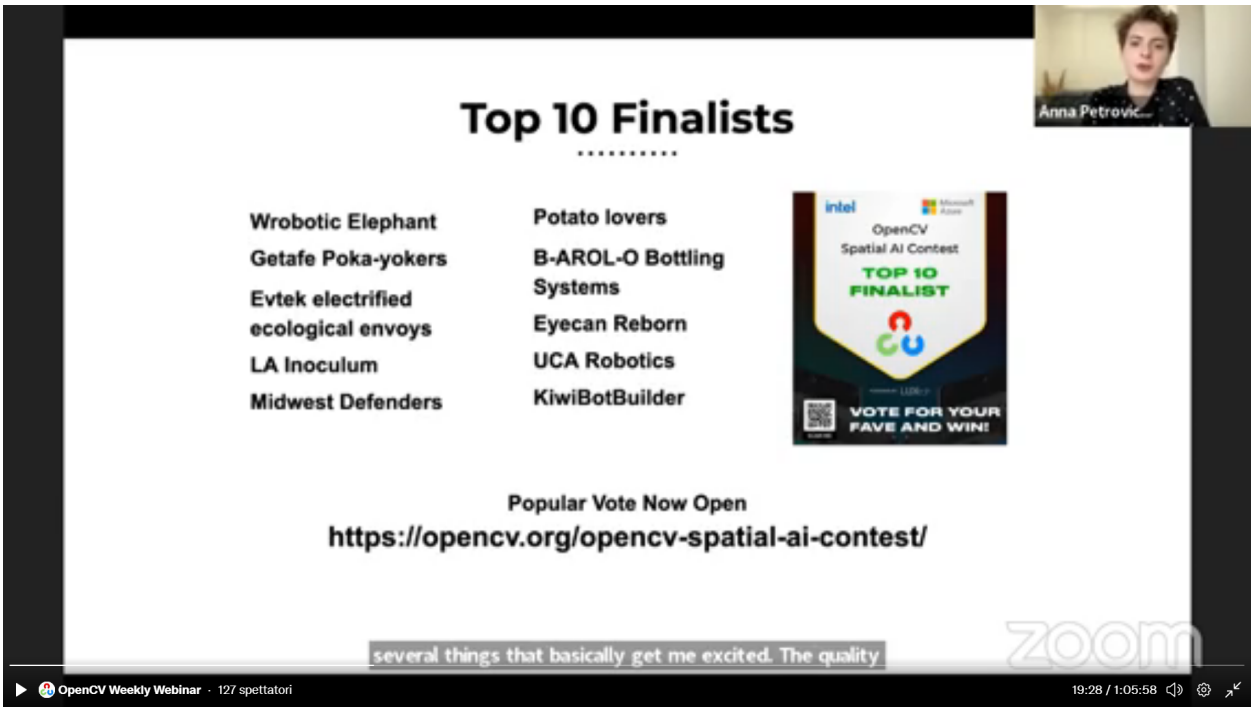
2.3.2 ARNEIS - OpenCV Spatial AI Contest Top 10, 2022-04-21

This is the ARNEIS elevator pitch video re-published on 2022-04-21 by OpenCV AI for the OpenCV Spatial AI Contest Popular Vote:



2.3.3 OpenCV Spatial AI Contest Top 10 Finalists, 2022-04-21

During the OpenCV Weekly Webinar Episode 54 our team was appointed Top 10 Finalist of the OpenCV Spatial AI Contest sponsored by Intel and Microsoft. Click on the image to watch the full announcement:

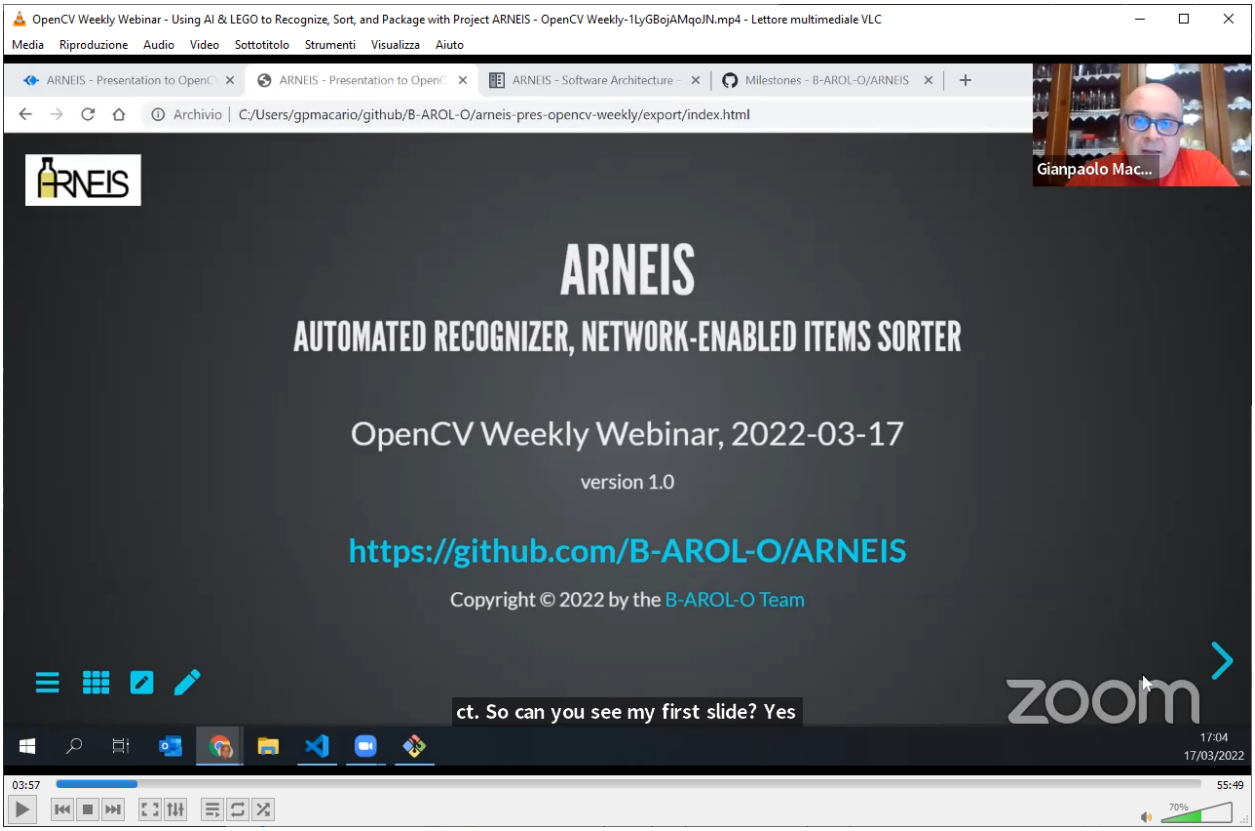


2.3.4 ARNEIS elevator pitch, 2022-04-05

This short video (3:48) was prepared as part of the final submission to the OpenCV Spatial AI Contest:

2.3.5 B-AROL-O team interview, 2022-03-17

The B-AROL-O Team was invited OpenCV Weekly Webinar Episode 49 to present ARNEIS, and explain the vision, the motivations and the achievements after the first three months of development:



2.3.6 Social media content

This section contains a collection of links to the public posts, tweets and videos produced by the B-AROL-O team during the OpenCV Spatial AI Contest for documenting the progress, the failures and the successes of the project.

We deliberately omitted any material produced by third-parties whom anyway are grateful to because they helped us giving exposure to the project, and in some cases stimulate external contributions. Most support was given us in terms of "Likes", "Retweets" or other ways, and even the simply posting of a link to our project page at https://arneis.readthedocs.io was helpful.

We believe we have already expressed our gratitude to all the people who provided us with code contributions, support and constructive feedbacks; if not please accept our apologies.

ARNEIS Weekly Reports (from Gianpaolo blog)

Here is a list of weekly posts made by Gianpaolo on his personal blog to document the progress of the ARNEIS project during the OpenCV Spatial AI Competition:

Date	Coverage	Summary
2022-05-23	dev-cw20	Few days left for voting usThe ARNEIS project hits release 1.0Not attending Expo Mattoncino 6
2022-05-16	dev-cw19	Update 3D model of the ARNEIS conveyorProgress with testing Foundries.ioTroubleshooting Bluetooth on Portenta
2022-05-09	dev-cw18	B-AROL-O and ARNEIS logos deposited to ProofyKeep on voting us in the OpenCV public pollPreparation to Expo Mattoncino 6
2022-05-02	dev-cw17	We have a new contributor to ARNEISAnother contributor to ARNEISGetting started with Foundries.io
2022-04-25	dev-cw16	ARNEIS Top 10 Finalist of the OpenCV Spatial AI ContestPreparing for ARNEIS Release 1.0More progress with the Arduino Portenta X8
2022-04-18	dev-cw15	Scaling up the Kubernetes clusterTesting software stack for Raspberry Pi OS 64-bitFirst tests with Arduino Portenta X8
2022-04-11	dev-cw14	Meeting in Arduino with Massimo BanziARNEIS project is getting knownSan Mauro Brick Show 2022
2022-04-04	dev-cw13	ARNEIS submitted to the OpenCV Spatial AI ContestARNEIS OpenCV submission videoPreparing for San Mauro Bricks 2022
2022-03-28	dev-cw12	Get inspiration from the Arduino WeekARNEIS System Integration in progressPreparing the final submission to the OpenCV Spatial AI Contest
2022-03-21	dev-cw11	B-AROL-O Team at the OpenCV Weekly WebinarWorking on a synthetic dataset for training the ARNEIS NNBuilt a second ARNEIS MOC
2022-03-14	dev-cw10	New release of the LEGO® design for ARNEISProgress in controlling the LEGO® Powered Up devicesMulti-agent support to the Kubernetes cluster
2022-03-07	dev-cw09	We got a logo for the ARNEIS ProjectReleased code to pretty-print LEGO BOMARNEIS will be showcased at San Mauro Brick Show 2022
2022-02-28	dev-cw08	Talks with Piemonte Bricks LUGThe MOC is taking shapeFirst attempts at programming LEGO® Technic® Bluetooth Hubs
2022-02-21	dev-cw07	First assembly of the ARNEIS Conveyor with LEGO®Our OpenCV Weekly Webinar postponedFirst release of the ARNEIS Software Architecture
2022-02-14	dev-cw06	Our LEGO 42100 Set should eventually arriveFirst shot at ARNEIS System ArchitectureAnd the Kubernetes cluster is alive
2022-02-07	dev-cw05	OAK-D-Lite LEGO® mechanical adapterMotorized Bottle Conveyor (2022-02-06)Steps for training our custom Neural Network
2022-01-31	dev-cw04	First rendering of the ARNEIS Bottle ConveyorProblems in sourcing LEGO® Set 42100First attempts at recognizing mignon bottles with the OAK-D-Lite
2022-01-24	dev-cw03	Enhancements to the B-AROL-O logoAnnouncing arneis.readthedocs.io
2022-01-17	dev-cw02	Welcome to our first contributors
2022-01-10	dev-cw01	Announcing regular weekly posts until end of Phase 2
2021-12-18	Announcement	ARNEIS project selected finalist at the OpenCV Spatial AI Contest

Twitter

NOTE: Search for `#ARNEIS` and `#OAKDLiteContest` to find other tweets on the same subject:

1. 2021-12-18 11:28: https://twitter.com/gpmacario/status/1472151778678517760
2. 2021-12-17 15:21: https://twitter.com/avaresi/status/1472210479212867595
3. 2022-01-10 19:49: https://twitter.com/gpmacario/status/1480612900502917122
4. 2022-01-15 10:48: https://twitter.com/baroloteam/status/1482288465941254144
5. 2022-01-16 22:53: https://twitter.com/gpmacario/status/1482833459253063680
6. 2022-01-25 18:27: https://twitter.com/avaresi/status/1486027840533499904
7. 2022-01-26 18:07: https://twitter.com/baroloteam/status/1486204084776583178
8. 2022-01-30 09:16: https://twitter.com/baroloteam/status/1487701128628383751
9. 2022-01-30 14:24: https://twitter.com/baroloteam/status/1487778753027715073
10. 2022-01-30 14:27: https://twitter.com/gpmacario/status/1487779578118615044
11. 2022-01-31 13:55: https://twitter.com/gpmacario/status/1488133932101033988
12. 2022-01-31 14:04: https://twitter.com/baroloteam/status/1488136200372338692
13. 2022-02-06 13:38: https://twitter.com/avaresi/status/1490303925400084484
14. 2022-02-06 21:37: https://twitter.com/baroloteam/status/1490424433865117700
15. 2022-02-07 13:54: https://twitter.com/gpmacario/status/1490670294536699907
16. 2022-02-07 13:57: https://twitter.com/baroloteam/status/1490671045715574788
17. 2022-02-10 23:00: https://twitter.com/avaresi/status/1491894970063761416
18. 2022-02-14 13:46: https://twitter.com/gpmacario/status/1493204971302567936
19. 2022-02-14 13:47: https://twitter.com/baroloteam/status/1493205240539197444
20. 2022-02-22 14:59: https://twitter.com/gpmacario/status/1495941221847646214
21. 2022-02-22 15:28: https://twitter.com/baroloteam/status/1495948692645421056
22. 2022-02-23 21:34: https://twitter.com/baroloteam/status/1496584284739674116
23. 2022-02-24 05:45: https://twitter.com/baroloteam/status/1496707728798011392
24. 2022-02-28 18:56: https://twitter.com/gpmacario/status/1498356421435482115
25. 2022-02-28 18:58: https://twitter.com/baroloteam/status/1498356996180955154
26. 2022-03-08 13:29: https://twitter.com/gpmacario/status/1501173253271961602
27. 2022-03-14 14:41: https://twitter.com/gpmacario/status/1503365803848175618
28. 2022-03-14 14:43: https://twitter.com/baroloteam/status/1503366271118807045
29. 2022-03-15 21:17: https://twitter.com/opencvweekly/status/1503827692205998080
30. 2022-03-16 06:35: https://twitter.com/gpmacario/status/1503968248945160199
31. 2022-03-16 16:01: https://twitter.com/baroloteam/status/1504110505941741568
32. 2022-03-18 18:01: https://twitter.com/baroloteam/status/1504865479290241026
33. 2022-03-18 18:08: https://twitter.com/baroloteam/status/1504867311194456072
34. 2022-03-26 18:57: https://twitter.com/baroloteam/status/1507778881658540038

35. 2022-03-27 01:43: https://twitter.com/gpmacario/status/1507881065033175041
36. 2022-03-28 21:13: https://twitter.com/gpmacario/status/1508522598665236481
37. 2022-03-28 22:03: https://twitter.com/gpmacario/status/1508535208638926851
38. 2022-03-30 08:33: https://twitter.com/baroloteam/status/1509056192055689218
39. 2022-04-04 19:05: https://twitter.com/gpmacario/status/1511027081861566464
40. 2022-04-05 12:47: https://twitter.com/baroloteam/status/1511113288377151491
41. 2022-04-12 13:44: https://twitter.com/baroloteam/status/1513844561545768965
42. 2022-04-17 21:13: https://twitter.com/gpmacario/status/1515770531731095562
43. 2022-04-17 21:15: https://twitter.com/baroloteam/status/1515770869410324481
44. 2022-04-18 16:08: https://twitter.com/gpmacario/status/1516056138168643595
45. 2022-04-18 16:13: https://twitter.com/baroloteam/status/1516057455175909385
46. 2022-04-21 07:48: https://twitter.com/gpmacario/status/1517017417762889729
47. 2022-04-21 17:56: https://twitter.com/baroloteam/status/1517170479102054405
48. 2022-04-21 17:59: https://twitter.com/gpmacario/status/1517171276615503875
49. 2022-04-21 18:00: https://twitter.com/opencvweekly/status/1517171312807989249
50. 2022-05-02 20:59: https://twitter.com/gpmacario/status/1521202789535436802
51. 2022-05-02 22:09: https://twitter.com/baroloteam/status/1521220425623257088
52. 2022-05-09 07:34: https://twitter.com/gpmacario/status/1523536748235870208
53. 2022-05-09 07:42: https://twitter.com/baroloteam/status/1523538811321430016
54. 2022-05-16 13:42: https://twitter.com/gpmacario/status/1526166010964885505
55. 2022-05-16 13:52: https://twitter.com/baroloteam/status/1526168831021965314
56. 2022-05-22 11:42: https://twitter.com/baroloteam/status/1528310312591581184
57. 2022-05-22 11:46: https://twitter.com/gpmacario/status/1528311412312924160
58. 2022-05-24 17:26: https://twitter.com/gpmacario/status/1529121730475790336
59. 2022-05-24 17:33: https://twitter.com/baroloteam/status/1529123465097363457

LinkedIn

NOTE: Search for `#ARNEIS` and/or `#OAKDLiteContest` to find other posts on the same subject:

1. 2022-01-17: Gianluca on ARNEIS project journal - dev-cw02: https://www.linkedin.com/posts/gianluca-teti_arneis-project-journal-dev-cw02-activity-6889286643065393152-J2YP
2. 2022-01-24: Gianluca on ARNEIS project journal - dev-cw03: https://www.linkedin.com/posts/gianluca-teti_arneis-project-journal-dev-cw03-activity-6891458849144930304-yKTn
3. 2022-02-02: Gianluca on ARNEIS project journal - dev-cw04: https://www.linkedin.com/posts/gianluca-teti_github-b-arol-oarneis-automated-recognizer-activity-6894740891420938240-i1Ay
4. 2022-02-07: Gianluca on ARNEIS project journal - dev-cw05: https://www.linkedin.com/posts/gianluca-teti_arneis-project-journal-dev-cw05-activity-6896872473116123137-YzWT
5. 2022-02-28: Gianluca on ARNEIS project journal - dev-cw08: https://www.linkedin.com/posts/gianluca-teti_arneis-project-journal-dev-cw08-activity-6904516778966228992-EM8P

6. 2022-03-07: Gianpaolo on ARNEIS project journal - dev-cw09: https://www.linkedin.com/feed/update/urn:li:activity:6906939286512631808/
7. 2022-03-17: Using AI & LEGO to Recognize, Sort, and Package - OpenCV Weekly - Event by OpenCV: https://www.linkedin.com/video/event/urn:li:ugcPost:6909587011267891201/
8. 2022-04-04: Gianpaolo on ARNEIS project journal - dev-cw13: https://www.linkedin.com/feed/update/urn:li:activity:6916879749529108480/
9. 2022-04-05: Comment to post by Satya Mallick: https://www.linkedin.com/feed/update/urn:li:activity:6916649664603717632?commentUrn=urn%3Ali%3Acomment%3A%28activity%3A6916649664603717632%2C6916963029045628928%29
10. 2022-04-12: Article by AROL S.p.A. on LinkedIn: https://www.linkedin.com/pulse/arneis-automated-recognizer-network-enabled-items-/
11. 2022-04-12: Post by AROL Sp.A. with link to article: https://www.linkedin.com/feed/update/urn:li:activity:6919549840301404160/
12. 2022-04-13: Gianpaolo repost of article by AROL S.p.A.: https://www.linkedin.com/feed/update/urn:li:activity:6919910108042792960/
13. 2022-04-14: Pietro repost of article by AROL S.p.A.: https://www.linkedin.com/feed/update/urn:li:activity:6919938297855098880/
14. 2022-04-15: Post by Alessandro soliciting votes for "Team B-AROL-O" at #OAKDLiteContest: https://www.linkedin.com/posts/alessandrovaresi_oakdlitecontest-arneis-activity-6928303739082530816-X9Kx
15. 2022-04-16: Serena Bisacca repost of post by Alessandro: https://www.linkedin.com/posts/serena-bisacca_oakdlitecontest-arneis-activity-6928556282975006720-jHxW
16. 2022-04-15: Gianpaolo repost of post by Alessandro: https://www.linkedin.com/posts/gmacario_oakdlitecontest-arneis-activity-6929081218374197248-HaD5

YouTube

Those are mainly the videos made by Alessandro:

1. 2022-01-22: Bottle conveyor 20220122 (https://youtu.be/BFKyx1COfqk)
2. 2022-02-10: ARNEIS project by B-AROL-O team - WIP (https://youtu.be/7qxbT31U5dE)
3. 2022-02-14: OpenCV Spatial AI Contest - NeuralNetwork 01 (https://youtu.be/fuWxTJHCN7g)
4. 2022-02-23: OpenCV Spatial AI Contest - NeuralNetwork 02 (https://youtu.be/q7ccqtx44ko)
5. 2022-03-20: ARNEIS LEGO MOC History (https://youtu.be/S-DiK0UgNBY)
6. 2022-03-29: Using AI & LEGO to Recognize, Sort, and Package Bottles - OpenCV Weekly Ep 49 - 3/17/22 (https://youtu.be/nwAOgkqVJMo)
7. 2022-03-31: ARNEIS Conveyor 22.04 (https://youtu.be/4_–99dwwpc)
8. 2022-04-05: ARNEIS submission Video (https://youtu.be/qHFRRHWtTqY)
9. 2022-04-21: ARNEIS - OpenCV Spatial AI Contest Top 10 (https://youtu.be/mijyQ9Kjg9Y)
10. 2022-04-26: Announcing The Winners of OpenCV Spatial AI Contest - OpenCV Weekly Webinar Ep. 54 - 4/21/22 (https://www.youtube.com/watch?v=NXhtmPpLEbc%C3%AC)
11. 2022-06-17: AI Show Live - Episode 59 - 2022 OpenCV Hackathon Winners and More! (https://www.youtube.com/watch?v=cl1sDvzCD9c)

Facebook

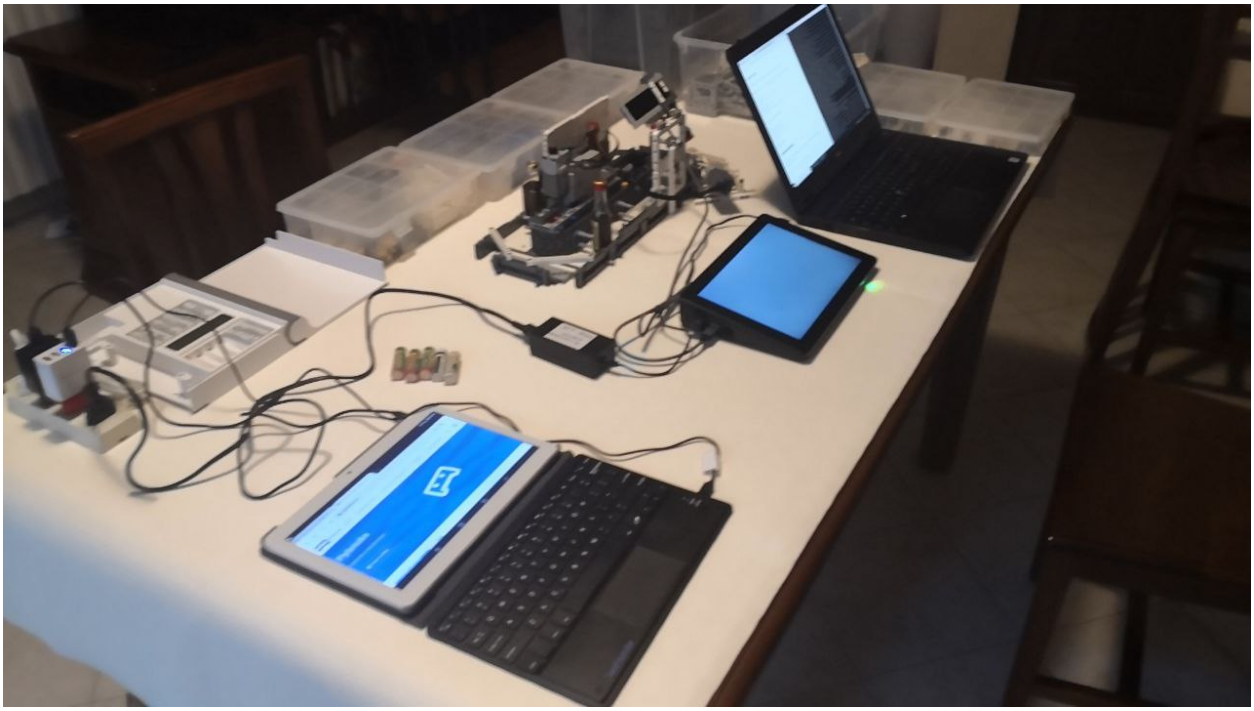
NOTE: Search for `#ARNEIS` and `#OAKDLiteContest` to find other posts on the same subject:

1. 2021-12-24 15:45: https://www.facebook.com/gianluca.teti/posts/10226595044611171
2. 2022-01-10 15:19: https://www.facebook.com/alessandro.varesi.67/posts/5107573495960647
3. 2022-01-10 20:13: https://www.facebook.com/gianluca.teti/posts/10226682797124929
4. 2022-01-10 21:45: https://www.facebook.com/gianpaolo.macario/posts/10158724207284542
5. 2022-02-08 18:44: https://www.facebook.com/1498552959/posts/10226836812855226/
6. 2022-02-19 09:34: https://www.facebook.com/gianpaolo.macario/posts/10158786608679542
7. 2022-02-23 07:04: https://www.facebook.com/gianpaolo.macario/posts/10158792878009542
8. 2022-03-05 09:59: https://www.facebook.com/photo/?fbid=10158808808314542
9. 2022-03-18 19:06: https://www.facebook.com/photo/?fbid=10158828954349542
10. 2022-03-19 14:04: https://www.facebook.com/photo/?fbid=10158829940794542
11. 2022-03-21 08:19: https://www.facebook.com/photo/?fbid=10158833031559542
12. 2022-04-09 12:41: https://www.facebook.com/gianluca.teti/posts/10227135788729436
13. 2022-04-09 16:34: https://www.facebook.com/piemontebricksLUG/photos/pcb.3545752858882361/3545747368882910
14. 2022-04-09 22:42: https://www.facebook.com/gianpaolo.macario/posts/10158862528084542
15. 2022-04-10 11:12: https://www.facebook.com/groups/custombricks1.8/posts/1432016813930408/
16. 2022-04-21 17:58: https://www.facebook.com/gianluca.teti/posts/10227196925897827
17. 2022-04-21 19:15: https://www.facebook.com/alessandro.varesi.67/posts/5421054011279259
18. 2022-04-21 19:48: https://www.facebook.com/caterina.piovano.5/posts/5521491711227835
19. 2022-04-21 20:16: https://www.facebook.com/franco.bianco.10/posts/7378613172180228
20. 2022-04-21 20:19: https://www.facebook.com/gianpaolo.macario/posts/10158879595084542
21. 2022-04-22 09:43: https://www.facebook.com/pierfrancesco.cerruti.3/posts/5128096707277605
22. 2022-04-23 08:34: https://www.facebook.com/groups/custombricks1.8/posts/1440654476399975/
23. 2022-04-23 13:23: https://www.facebook.com/groups/ohmybrick/posts/1890882524456117/
24. 2022-05-03 20:49: https://www.facebook.com/gianluca.teti/posts/10227263306237294
25. 2022-05-03 21:00: https://www.facebook.com/gianpaolo.macario/posts/10158898859124542

2.4 ARNEIS - System Integration Rel. 1.0

WORK IN PROGRESS

TODO



CHAPTER

THREE

ARCHITECTURE

Learn more about the ARNEIS project architecture.

- *ARNEIS - Specification*
- *ARNEIS - System Architecture*
- *ARNEIS - Software Architecture*

3.1 ARNEIS - Specification

3.1.1 Introduction

This document contains the Functional and Non-Functional requirements for the ARNEIS project.

Copyright and License

This document is Copyright (c) 2021-2024 by the B-AROL-O Team.

Reference documents

- ARNEIS: Automated Recognizer, Network-Enabled, Items Sorter - Project Proposal submitted to the OpenCV Spatial AI Contest, 2021-11-20 (PDF, 5 pages)

Definitions, acronyms and abbreviations

Please refer to https://arneis.readthedocs.io/en/latest/acronyms.html for the general acronyms used in the ARNEIS project.

- **NFR**: Non-Functional Requirement
- **OpenCV**: See https://opencv.org/
- **SysML**: Systems Modeling Language. See http://www.sysml.org/
- **UC**: Use case. See https://en.wikipedia.org/wiki/Use_case

3.1.2 Open questions

- Q1: How many types of bottles should be recognized?
  - A1: First hypothesis: At least 5 different types

3.1.3 Use cases

- UC1: End users should login to an internet site where they can submit an order (i.e. a case with 1xA, 2xB)
  - Result: the case is assembled taking bottles from the warehouse

3.1.4 Objectives

- Minimize the number of motors
- (NFR) Create a fun and "brand-izeable" installation making high help of sounds, videos, lights, etc

3.1.5 Design options (to be investigated)

- Rotating table
- Option for the container: something like the mignon 3x container

3.1.6 Design constraints

- The whole installation should fit on a 200 x 90 cm table
- The installation should be easy to unassemble and reassemble (to make it easy to transport it in the carry-on luggage)
- The installation should be powered up from one single phase 100-240Vac power supply, max 400W
- The installation should only use radio frequencies in the ISM band (i.e. 2.54 GHz)
- The mechanical design should be based of only official LEGO parts
- If no suitable LEGO parts can be identified from the official LEGO catalog, a limited number of 3D-printed parts are allowed. Do not use non-official LEGO products (this may hamper the possibility to feature the MOC in official LEGO fests in the future)
- Additionally, beyond the OAK-D-LITE device provided by the organizers, only widely available, off-the shelf electronic devices (such as smart phones, tablets, Raspberry Pi, USB cables etc) should be required
- The site for submitting orders should be usable from a mobile phone or tablet (should decide on minimum requirements, but ideally the site is responsive and renders beautifully on a phone)

3.2 ARNEIS - System Architecture

3.2.1 Reference Documents

- ARNEIS Project Proposal as submitted for the Phase 1 of the OpenCV Spatial AI Contest - B-AROL-O Bottling Systems team, 2021-11-20 (PDF, 5 pages)
- OMG Systems Modeling Language (OMG SysML™) Tutorial - September, 2009 (PDF, 132 pages)

3.2.2 Main mechatronic components

NOTE: In this list we deliberately omitted functions and services provided on the cloud, which are clearly available through the internet connection.

Client devices and HMI

- **Client1**: Generic Android phone
  - Reference model: Huwaei Mate 10 Pro
  - Connected to internet via Wi-Fi through AP1 (or mobile network)
  - Connected to RPI4 via Wi-Fi through AP1
- **Client2**: Generic Android tablet
  - Reference model: Lenovo TB-X606F
  - Connected to internet via Wi-Fi through AP1
  - Connected to RPI4 via Wi-Fi through AP1
- **Client3**: Generic iOS phone
  - Reference model: iPhone SE
  - Connected to internet via Wi-Fi through AP1 (or mobile network)
  - Connected to RPI4 via Wi-Fi through AP1

CPU and network devices

- **RPI4**: Raspberry Pi 4. Acting as main controller and gateway to cameras, sensors and actuators, as well as edge server to communicate with the services on the cloud
  - USB0: USB 3.0 Port #1. Connected to CAM1 via USB-A to USB-C cable
  - USB1: USB 3.0 Port #2. TODO
  - USB2: USB 2.0 Port #1. Connect to a USB keyboard (for debug purposes)
  - USB3: USB 2.0 Port #2. Connect to a USB mouse (for debug purposes)
  - MicroHDMI0: Connect to a HDMI display using a MicroHDMI-to-HDMI cable (for debug purposes)
  - Acting as BLE central
  - Connected to AP1 using wired Ethernet through a RJ-45 cable. In case wired Ethernet is not available, connect to AP1 through Wi-Fi
  - Services: SSH, http, others?

- **AP1**: Access Point and internet router
  - Reference model: Linksys WRT54GL
- **HUB1**: Electric Battery Box Powered Up Bluetooth Hub with Dark Bluish Gray Bottom
  - Connected to RPI4 via Bluetooth (acting as BLE peripheral)
- **HUB2**: Electric Battery Box Powered Up Bluetooth Hub with Dark Bluish Gray Bottom
  - Connected to RPI4 via Bluetooth (acting as BLE peripheral)
- (TBV) **HUB3**: Mindstorms NXT Control Hub
  - Connected to RPI4 using a USB-B to USB-A cable
- (TBV) **SBRICK1**: Sbrick Plus
  - Connected to RPI4 via Bluetooth (acting as BLE peripheral)
- (TBV) **SBRICK2**: Sbrick Plus
  - Connected to RPI4 via Bluetooth (acting as BLE peripheral)
- (TBV) **PMC1**: Arduino Portenta Machine Control
  - Connected to RPI4 via USB (or BLE? Or Ethernet?)
- (TBV) **NICLA1**: Arduino Nicla Sense ME
  - Connected to RPI4 via Bluetooth (acting as BLE peripheral)

Computer Vision devices

- **CAM1**: OAK-D-Lite (Auto Focus)
  - Connected via a USB-A to USB-C cable (USB 3.0 or later) to RPI4
- (TBV) **CAM2**: OAK-D-Lite (Auto Focus)
- (TBV) **CAM3**: OAK-D-Lite (Auto Focus)
- (TBV) **CAM4**: OAK-D-Lite (Auto Focus)

Actuators

- **M1**: Electric, Motor Powered Up, XL
  - Purpose: TODO
  - Connected to HUB1 through its Power Up cable
- **M2**: Electric, Motor Powered Up, XL
  - Purpose: TODO
  - Connected to HUB1 through its Power Up cable
- **M3**: Electric, Motor Powered Up, XL
  - Purpose: TODO
  - Connected to HUB1 through its Power Up cable
- **M4**: Electric, Motor Powered Up, L
  - Purpose: TODO

  - Connected to HUB1 through its Power Up cable
- **M5**: Electric, Motor Powered Up, L
  - Purpose: TODO
  - Connected to HUB2 through its Power Up cable
- **M6**: Electric, Motor Powered Up, L
  - Purpose: TODO
  - Connected to HUB2 through its Power Up cable
- **M7**: Electric, Motor Powered Up, L
  - Purpose: TODO
  - Connected to HUB2 through its Power Up cable
- (TBV) **EJCT1**: Ejector TODO
  - Purpose: TODO
  - Mechanically connected to (TBV) Mx
- (TBV) **EJCT2**: Ejector TODO
  - Purpose: TODO
  - Mechanically connected to (TBV) Mx
- (TBV) **EJCT3**: Ejector TODO
  - Purpose: TODO
  - Mechanically connected to (TBV) Mx
- TODO: LEGO Power Functions motor/switches/lights controlled via SBrick?
- TODO: LEGO Mindstorms NXT motors/switches/lights controlled via Mindstorms CPU?

Misc

- TODO: Lights?
- TODO: Physical push buttons/encoders/etc?

3.3 ARNEIS - Software Architecture

This document details the planned Software Architecture that will be developed for the ARNEIS project.

The software architecture of the ARNEIS project will be based on microservices running on the ARNEIS Kubernetes cluster.

3.3.1 Architecture of the ARNEIS cluster

The ARNEIS cluster is based on Kubernetes.

For simplicity we chose to deploy the ARNEIS cluster using K3s.

K3s (or "Lightweight Kubernetes") is a simplified installation of the Kubernetes distribution built for IoT and Edge computing.

K3s is an Open Source project started and maintained by Rancher.com (now part of SUSE).

Planned topology of the ARNEIS K3s cluster

Server Node[s]

1. VM arneis-vm01
2. (optional) 2x additional VM to provide High Availability?

Agent Nodes

1. The same host acting as main K3s server (primary agent)
2. VM arneis-vm02 on the cloud
3. RPi4 on each instance of ARNEIS edge system

3.3.2 Services to be deployed on the ARNEIS cluster

The following services are expected to be deployed on the ARNEIS cluster:

- Terminate HTTPS (Traefik Proxy instance built into K3s)
- Web server for ARNEIS documentation (replica of https://arneis.readthedocs.io)
- Database for ARNEIS backend service (alternative: MongoDB instance at https://www.mongodb.com/cloud/atlas/)
- ARNEIS backend service (Node.js + Koa - possibly based on some publicly available boilerplate)
- ARNEIS customer frontend (static site developed in Next.js)
- Service running on the RPi4 for controlling the OAK-D-Lite (Python3?)
- Service running on the RPi4 for controlling the LEGO® Technic® Bluetooth Hub (Python3?)

CHAPTER

FOUR

HOW-TO GUIDES

These guides will help walk you through specific use cases the team has gone through during the project realization.

- **For documentation authors:**
  - *HOWTO Configure arneis.readthedocs.io*
  - *HOWTO Create a landing page on Netlify*
  - *HOWTO Configure Google Analytics*
  - *HOWTO Install Eclipse Capella*
  - *HOWTO Mermaid utilization and code examples*
  - *HOWTO Use rclone with Cubbit DS3*
- **Learning AI and related tools:**
  - *HOWTO Install DepthAI on a Raspberry Pi 4B for the ARNEIS project*
  - *HOWTO Install DepthAI Demo v3.0.9 on Windows 10*
  - *HOWTO Train a NN for Object Detection and create a deployable model for the OAK-D-Lite*
- **For Embedded electronics enthusiast:**
  - *HOWTO Control a LEGO Powered Up Hub for the ARNEIS Project*
  - *HOWTO Install Raspberry Pi OS 64-bit for the ARNEIS project*
  - *HOWTO Configure a Static IP Address on Linux*
  - *HOWTO Prepare a Raspberry Pi 4B for the ARNEIS project*
  - *HOWTO Install MicroPython on an Arduino Nano 33 BLE*
  - *HOWTO Getting started with Foundries.io*
  - *HOWTO Testing Arduino Portenta X8*
- **For system integrators:**
  - *HOWTO Create a VM on Azure for the ARNEIS project*
  - *HOWTO Install a K3s cluster for the ARNEIS project*

4.1 HOWTO Configure arneis.readthedocs.io

4.1.1 Introduction

This document explains the steps performed for preparing the automated deployment of site https://arneis.readthedocs.io/ based on the contents of GitHub repository https://github.com/B-AROL-O/ARNEIS.

4.1.2 References

- https://readthedocs.org/
- https://docs.readthedocs.io/
- https://www.sphinx-doc.org/

4.1.3 Step-by-step instructions

Create a feature branch

Logged as gmacario@gmpowerhorse

```
cd ~/github/B-AROL-O/ARNEIS/
git checkout main
git pull --all --prune
```

Create a feature branch

```
git checkout -b feat/issue-29-readthedocs
```

Create initial Sphinx configuration

Follow instructions at https://docs.readthedocs.io/en/stable/intro/getting-started-with-sphinx.html

Create initial Sphinx configuration

```
cd docs
python3 -m venv .venv
source .venv/bin/activate
pip3 install sphinx
sphinx-quickstart
```

Fill in the required information:

```
(.venv) gmacario@gmpowerhorse:~/github/B-AROL-O/ARNEIS/docs (feat/issue-29-readthedocs)$␣
↪sphinx-quickstart
Welcome to the Sphinx 4.4.0 quickstart utility.

Please enter values for the following settings (just press Enter to
accept a default value, if one is given in brackets).

Selected root path: .
```

(continues on next page)

(continued from previous page)

```
You have two options for placing the build directory for Sphinx output.
Either, you use a directory "_build" within the root path, or you separate
"source" and "build" directories within the root path.
> Separate source and build directories (y/n) [n]:

The project name will occur in several places in the built documentation.
> Project name: ARNEIS
> Author name(s): The B-AROL-O Team
> Project release []:

If the documents are to be written in a language other than English,
you can select a language here by its language code. Sphinx will then
translate text that it generates into that language.

For a list of supported codes, see
https://www.sphinx-doc.org/en/master/usage/configuration.html#confval-language.
> Project language [en]:

Creating file /home/gmacario/github/B-AROL-O/ARNEIS/docs/conf.py.
Creating file /home/gmacario/github/B-AROL-O/ARNEIS/docs/index.rst.
Creating file /home/gmacario/github/B-AROL-O/ARNEIS/docs/Makefile.
Creating file /home/gmacario/github/B-AROL-O/ARNEIS/docs/make.bat.

Finished: An initial directory structure has been created.

You should now populate your master file /home/gmacario/github/B-AROL-O/ARNEIS/docs/
↪index.rst and create other documentation
source files. Use the Makefile to build the docs, like so:
   make builder
where "builder" is one of the supported builders, e.g. html, latex or linkcheck.

(.venv) gmacario@gmpowerhorse:~/github/B-AROL-O/ARNEIS/docs (feat/issue-29-readthedocs)*$
```

Locally build the static HTML site

Run the following command:

```
make html
```

Result:

```
(.venv) gmacario@gmpowerhorse:~/github/B-AROL-O/ARNEIS/docs (feat/issue-29-readthedocs)*
↪$ make html
Running Sphinx v4.4.0
making output directory... done
building [mo]: targets for 0 po files that are out of date
building [html]: targets for 1 source files that are out of date
updating environment: [new config] 1 added, 0 changed, 0 removed
reading sources... [100%] index
looking for now-outdated files... none found
```

(continues on next page)

(continued from previous page)

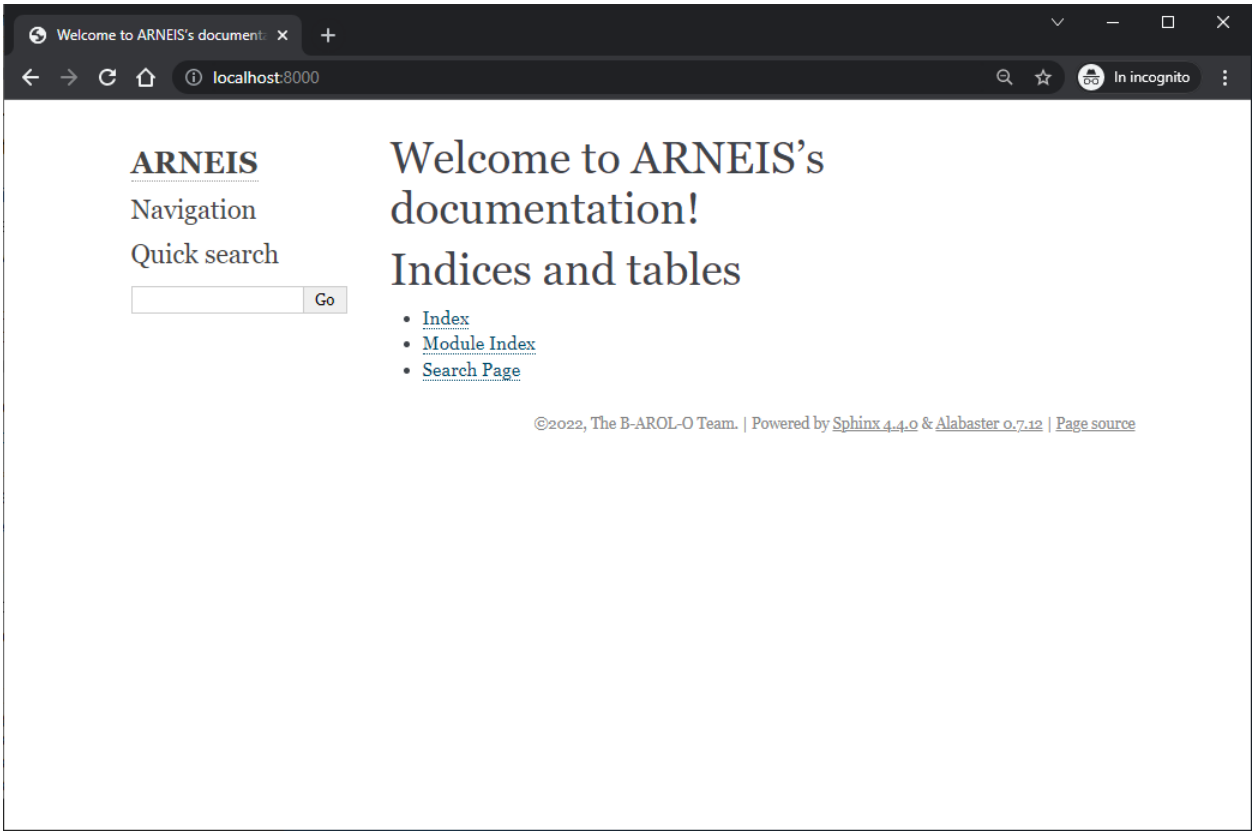
```
pickling environment... done
checking consistency... done
preparing documents... done
writing output... [100%] index
generating indices... genindex done
writing additional pages... search done
copying static files... done
copying extra files... done
dumping search index in English (code: en)... done
dumping object inventory... done
build succeeded.

The HTML pages are in _build/html.
(.venv) gmacario@gmpowerhorse:~/github/B-AROL-O/ARNEIS/docs (feat/issue-29-readthedocs)*$
```

Test the generated website

Run a simple http server to make the generated pages available on the network:

```
cd ~/github/B-AROL-O/ARNEIS/docs/
python3 -m http://server --directory _build/html/
```

then if you open `http://localhost:8000/` from your browser, the home page of the generated website should be displayed.



All the http requests will also be logged on the terminal where you launched the http server:

```
(.venv) gmacario@gmpowerhorse:~/github/B-AROL-O/ARNEIS/docs (feat/docsite-add-links)*$␣
↪python3 -m http.server --directory _build/html/
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [22/Jan/2022 07:48:24] "GET / HTTP/1.1" 304 -
127.0.0.1 - - [22/Jan/2022 07:48:24] "GET /_static/pygments.css HTTP/1.1" 304 -
127.0.0.1 - - [22/Jan/2022 07:48:24] "GET /_static/css/theme.css HTTP/1.1" 304 -
127.0.0.1 - - [22/Jan/2022 07:48:24] "GET /_static/documentation_options.js HTTP/1.1"␣
↪304 -
127.0.0.1 - - [22/Jan/2022 07:48:24] "GET /_static/js/theme.js HTTP/1.1" 304 -
127.0.0.1 - - [22/Jan/2022 07:48:25] "GET /_static/css/fonts/lato-normal.woff2?
↪bd03a2cc277bbbc338d464e679fe9942 HTTP/1.1" 304 -
127.0.0.1 - - [22/Jan/2022 07:48:25] "GET /_static/css/fonts/fontawesome-webfont.woff2?
↪af7ae505a9eed503f8b8e6982036873e HTTP/1.1" 304 -
127.0.0.1 - - [22/Jan/2022 07:48:25] "GET /_static/css/fonts/lato-bold.woff2?
↪cccb897485813c7c256901dbca54ecf2 HTTP/1.1" 304 -
127.0.0.1 - - [22/Jan/2022 07:48:25] "GET /_static/css/fonts/Roboto-Slab-Bold.woff2?
↪9984f4a9bda09be08e83f2506954adbe HTTP/1.1" 304 -
127.0.0.1 - - [22/Jan/2022 07:48:32] "GET /genindex.html HTTP/1.1" 200 -
127.0.0.1 - - [22/Jan/2022 07:48:36] "GET /index.html HTTP/1.1" 200 -
```

Add the Read the Docs Sphinx Theme

Reference: https://sphinx-rtd-theme.readthedocs.io/

Make sure the following package is included in `docs/requirements.txt`

```
sphinx_rtd_theme==1.0.0
```

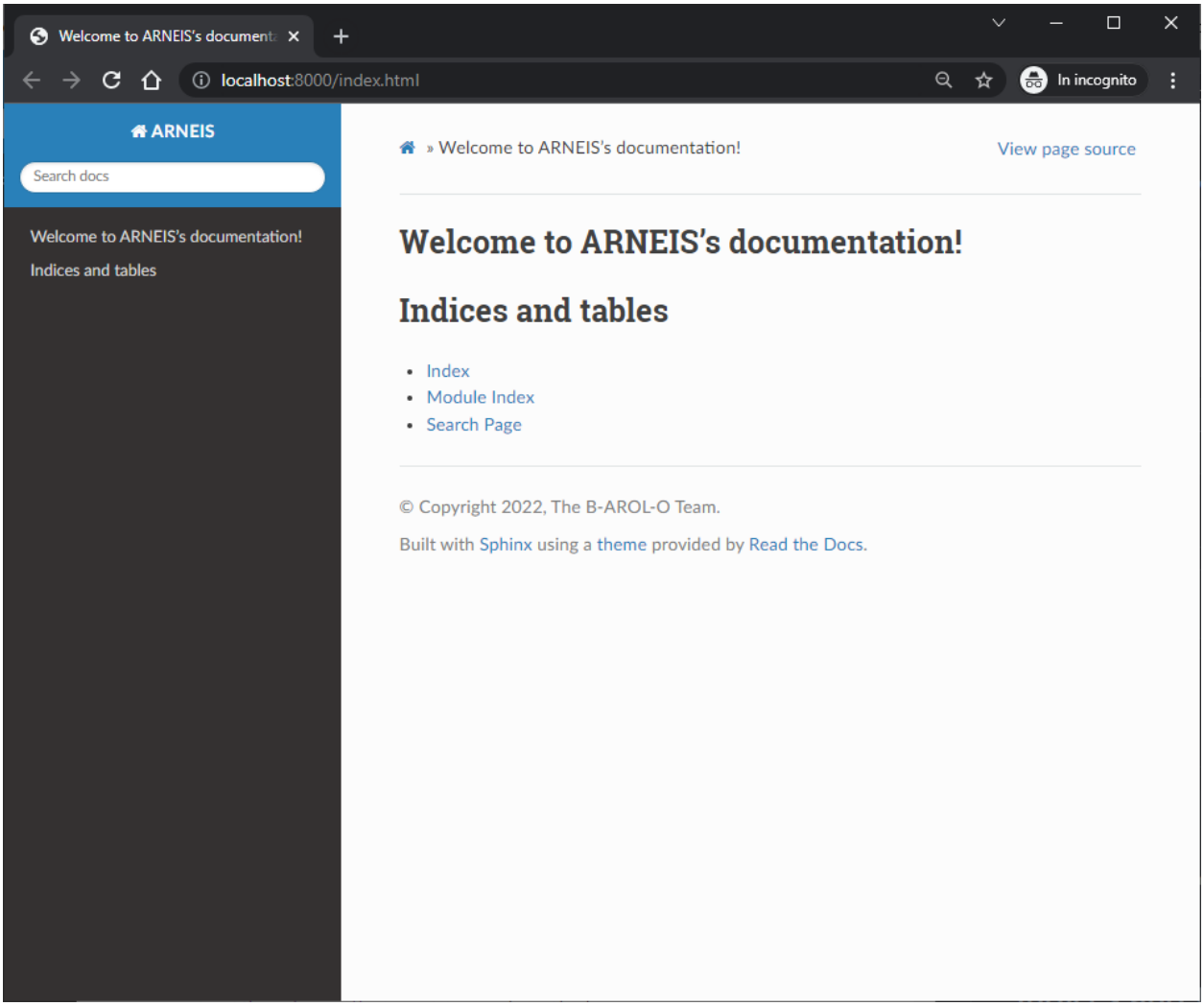
Add the following configuration to `docs/conf.py`

```
extensions = ['sphinx_rtd_theme']

html_theme = 'sphinx_rtd_theme'
```

Test as usual with the following commands

```
cd ~/github/B-AROL-O/ARNEIS/docs/
python3 -m http.server --directory _build/html/
```

then open `http://localhost:8000/` from your browser. Verify that the Read the Docs Sphinx Theme has been applied to the website:



Adding support for Markdown

Reference: https://www.sphinx-doc.org/en/master/usage/markdown.html

```
cd ~/github/B-AROL-O/ARNEIS/docs
pip3 install --upgrade myst-parser
```

Add *myst_parser* to the list of configured extensions

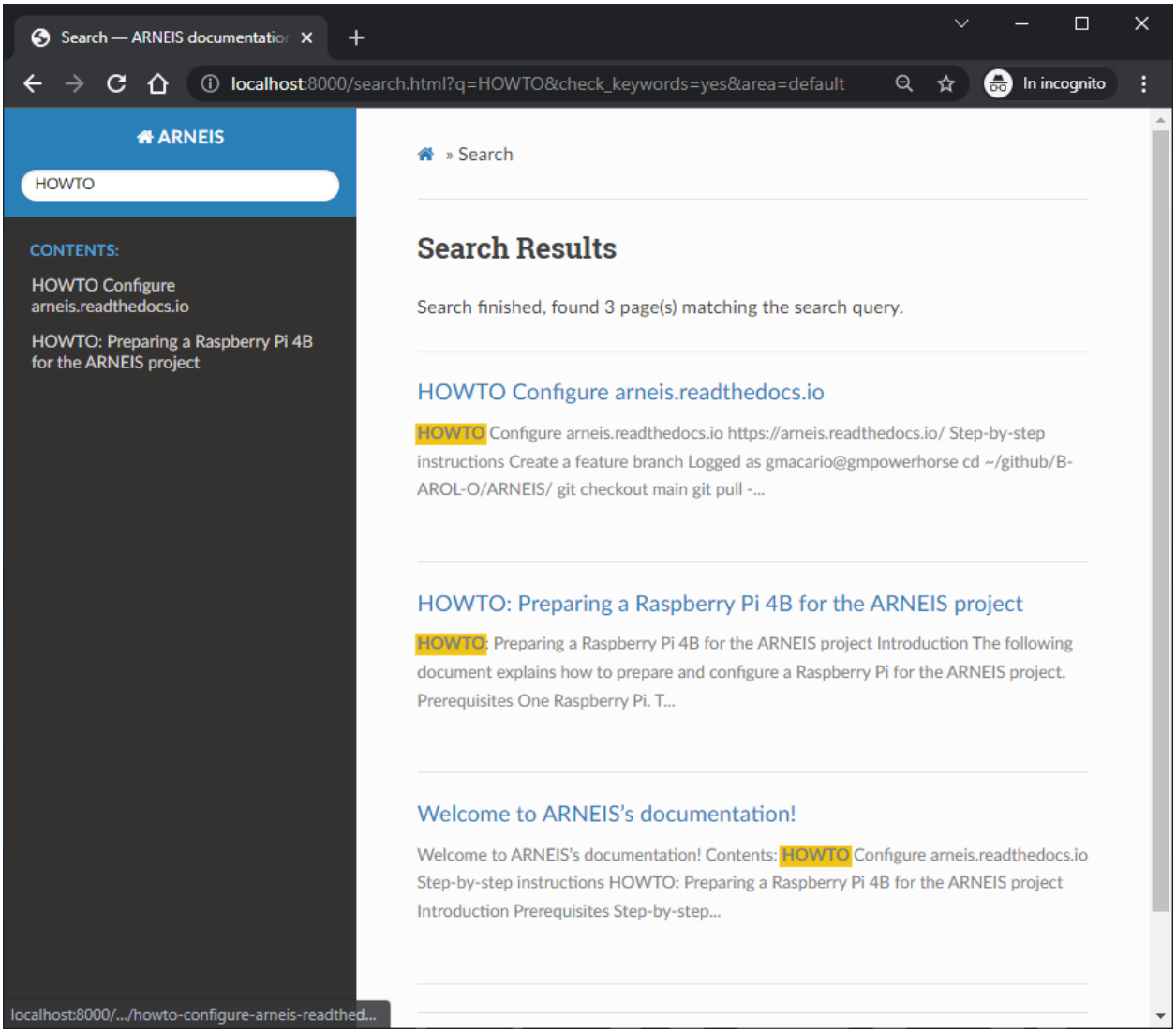
```
extensions = [
    'myst_parser'
]
```

Test as usual with the following commands

```
cd ~/github/B-AROL-O/ARNEIS/docs/
python3 -m http.server --directory _build/html/
```

then open `http://localhost:8000/` from your browser.

Type "HOWTO" into the "Search docs" input field and verify that the Markdown pages have been included as well



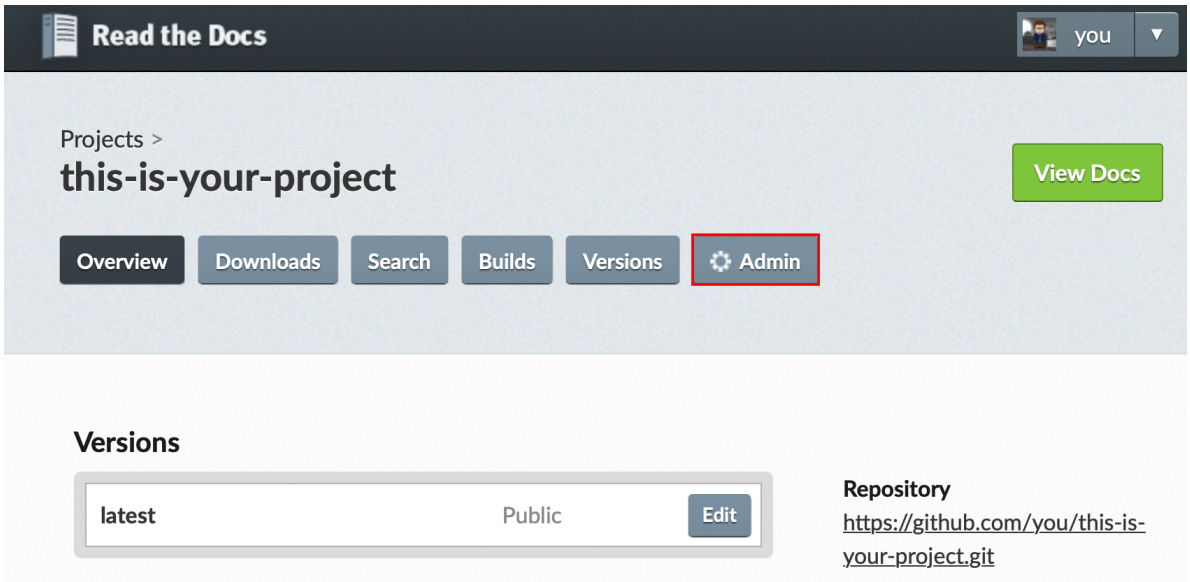
Implement preview for Pull Requests

References:

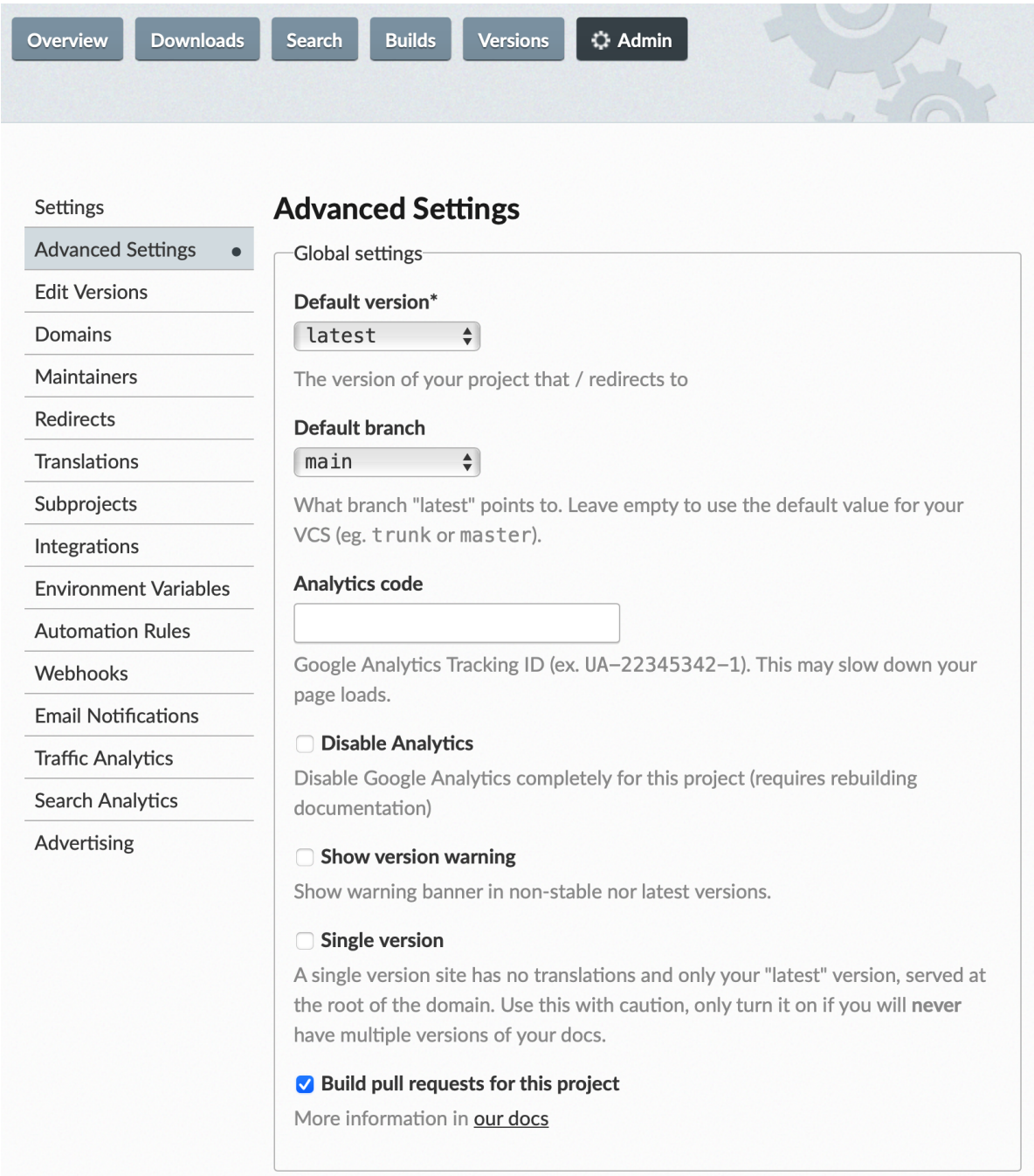
- https://github.com/B-AROL-O/ARNEIS/issues/85
- Preview Documentation from Pull Requests

In order to enable the Pull Request preview feature you should execute the following steps:

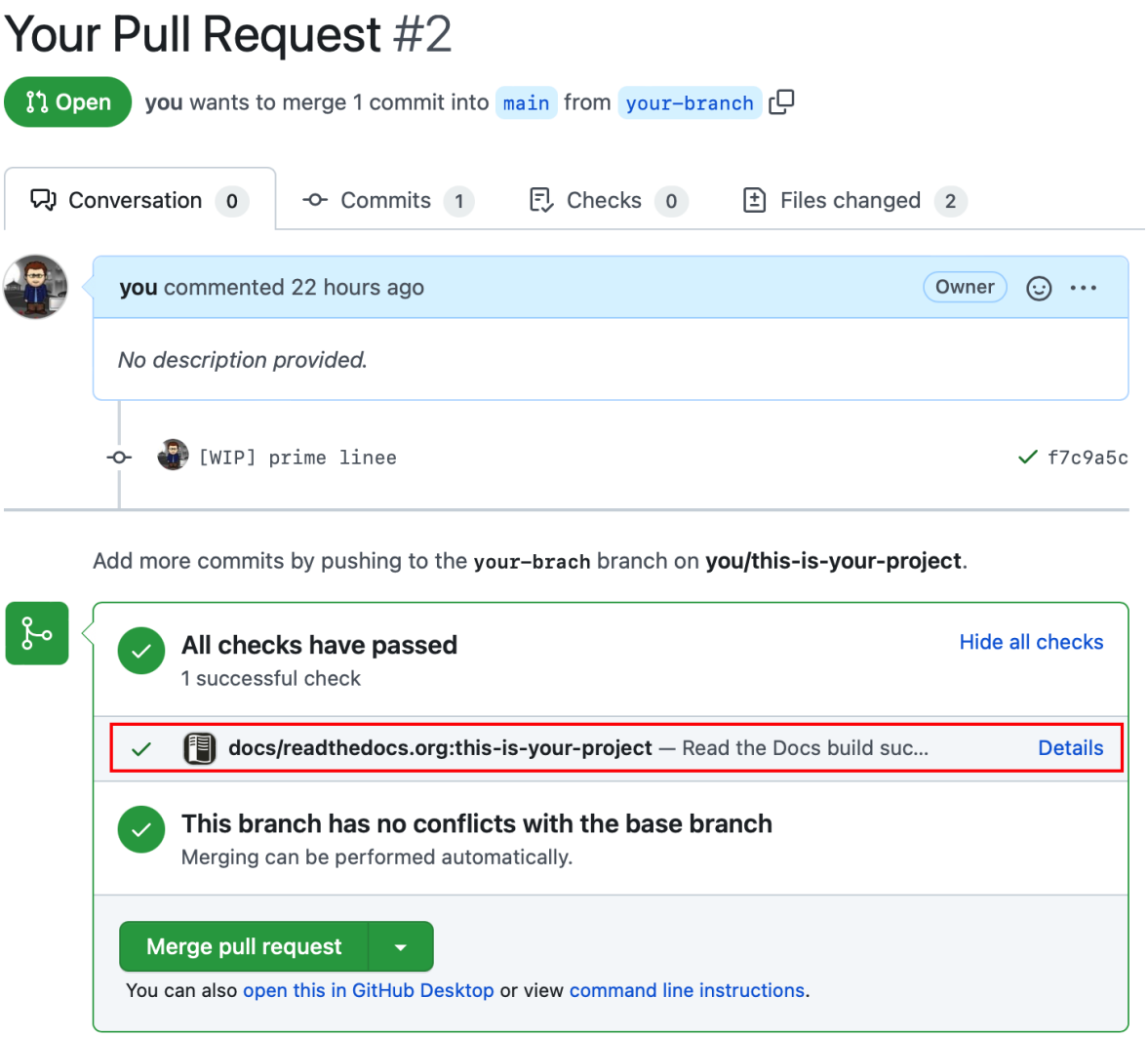
1. In your *Read the Docs* project page select the `Admin` menu



2. In `Advanced Settings` select **Build pull requests for this project**



3. That's all. For each pull request an automatic check which builds the document on *Read The Docs* is added. You can check the result of this compilation in the `Detail` information for this check.



Removing the Ads

References:

- https://docs.readthedocs.io/en/stable/advertising/ethical-advertising.html

There are multiple ways to opt out of the advertising on Read the Docs.

1. You can go completely ad-free by becoming a Gold member or a Supporter. Additionally, Gold members may remove advertising from their projects for all visitors.
2. You can opt out of seeing paid advertisements on documentation pages:
   - Go to the drop down user menu in the top right of the Read the Docs dashboard and clicking **Settings** (https://readthedocs.org/accounts/edit/).
   - On the **Advertising** tab, you can deselect **See paid advertising**.

   You will still see community ads for open source projects and conferences.
3. Project owners can also opt out of paid advertisements for their projects. You can change these options:
   - Go to your **project** page (`/projects/<slug>/`)

- Go to **Admin** > **Advertising**
- Change your advertising settings

4. If you are part of a company that uses Read the Docs to host documentation for a commercial product, Read the Docs for Business offers a completely ad-free experience, additional build resources, and other great features like CDN support and private documentation.
5. If you would like to completely remove advertising from your open source project, but our commercial plans don't seem like the right fit, you may get in touc with ReadTheDocs.org to discuss alternatives to advertising.

4.2 HOWTO Create a landing page on Netlify

4.2.1 Introduction

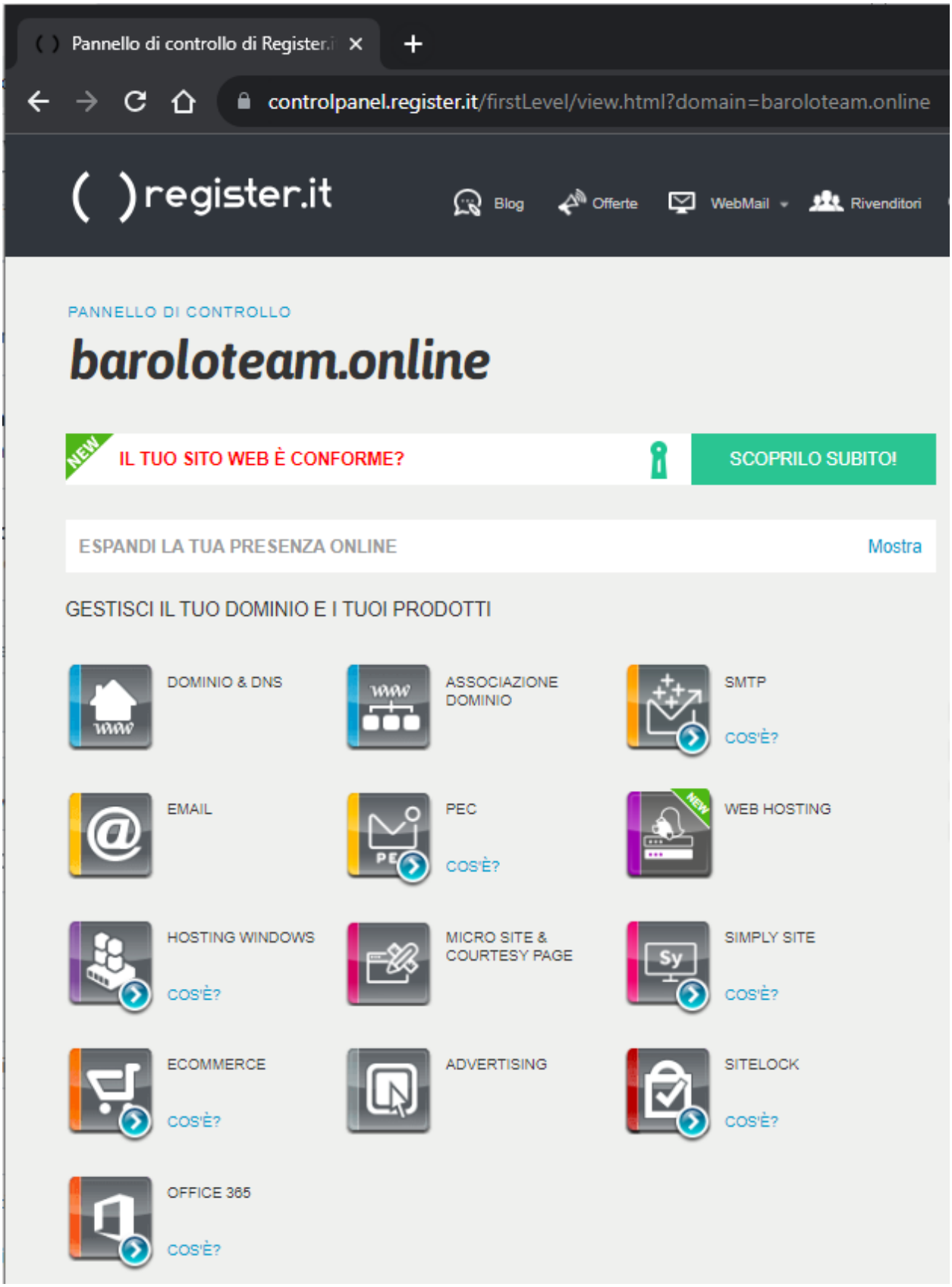
This document is a detailed walkthrough of the steps performed to create a landing page on https://baroloteam.online.

4.2.2 Step-by-step instructions

Register a domain name

Choose your preferred domain registrar and buy an available domain.

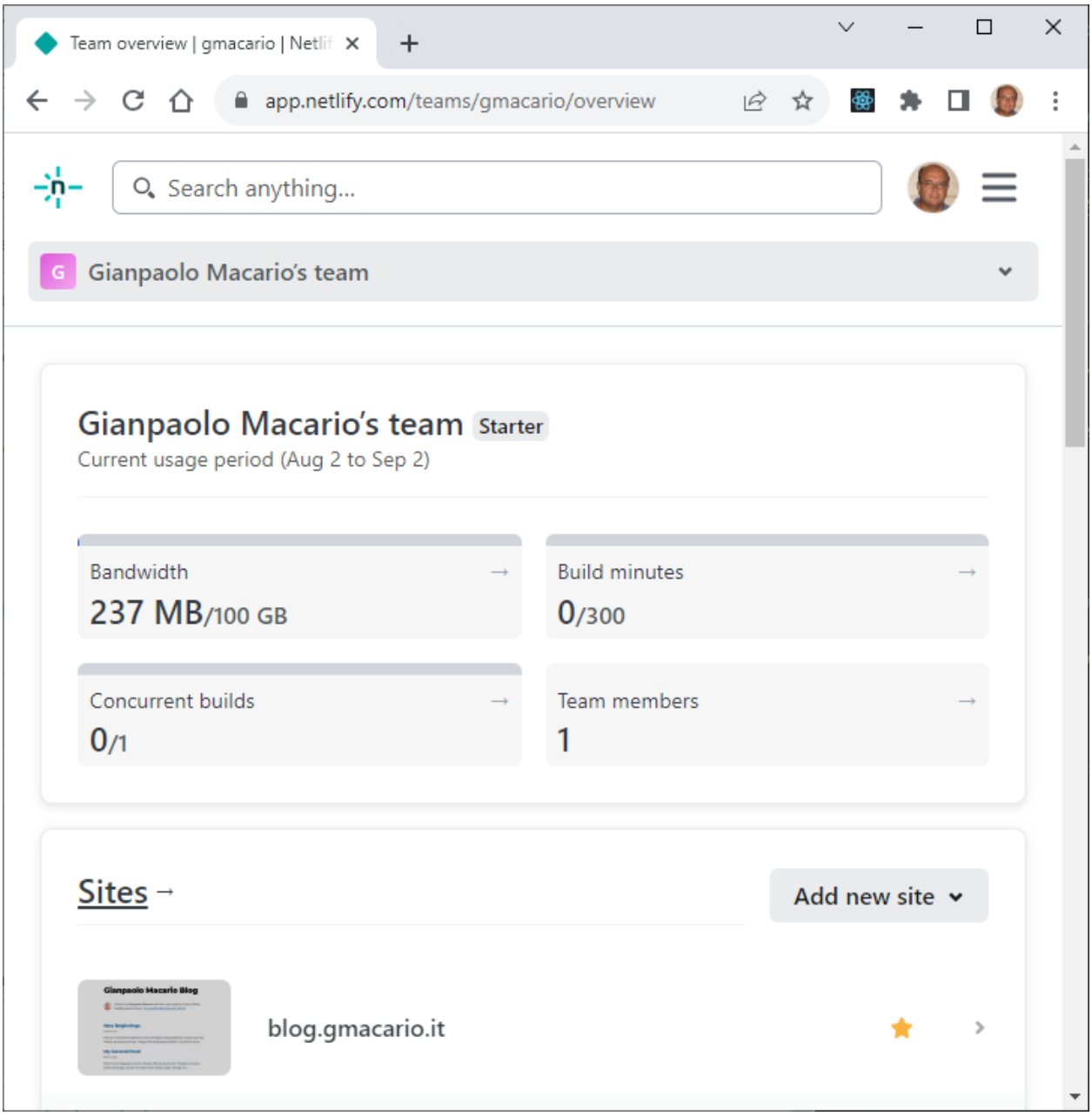
In our example I logged into https://register.it and purchased `baroloteam.online`.


Pannello di controllo di Register.i
controlpanel.register.it/firstLevel/view.html?domain=baroloteam.online
( )register.it
Blog
Offerte
WebMail
Rivenditori
PANNELLO DI CONTROLLO
baroloteam.online
NEW
IL TUO SITO WEB È CONFORME?
SCOPRILO SUBITO!
ESPANDI LA TUA PRESENZA ONLINE
Mostra
GESTISCI IL TUO DOMINIO E I TUOI PRODOTTI
DOMINIO & DNS
ASSOCIAZIONE DOMINIO
SMTP
COS'È?
EMAIL
PEC
COS'È?
NEW
WEB HOSTING
HOSTING WINDOWS
COS'È?
MICRO SITE & COURTESY PAGE
SIMPLY SITE
COS'È?
ECOMMERCE
COS'È?
ADVERTISING
SITELOCK
COS'È?
OFFICE 365
COS'È?

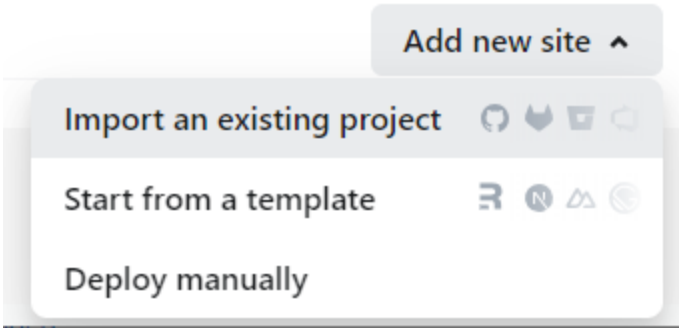
Create a project on Netlify

Login to https://www.netlify.com/ (or Sign up if you do not have an account yet).

You should be redirected to this page:



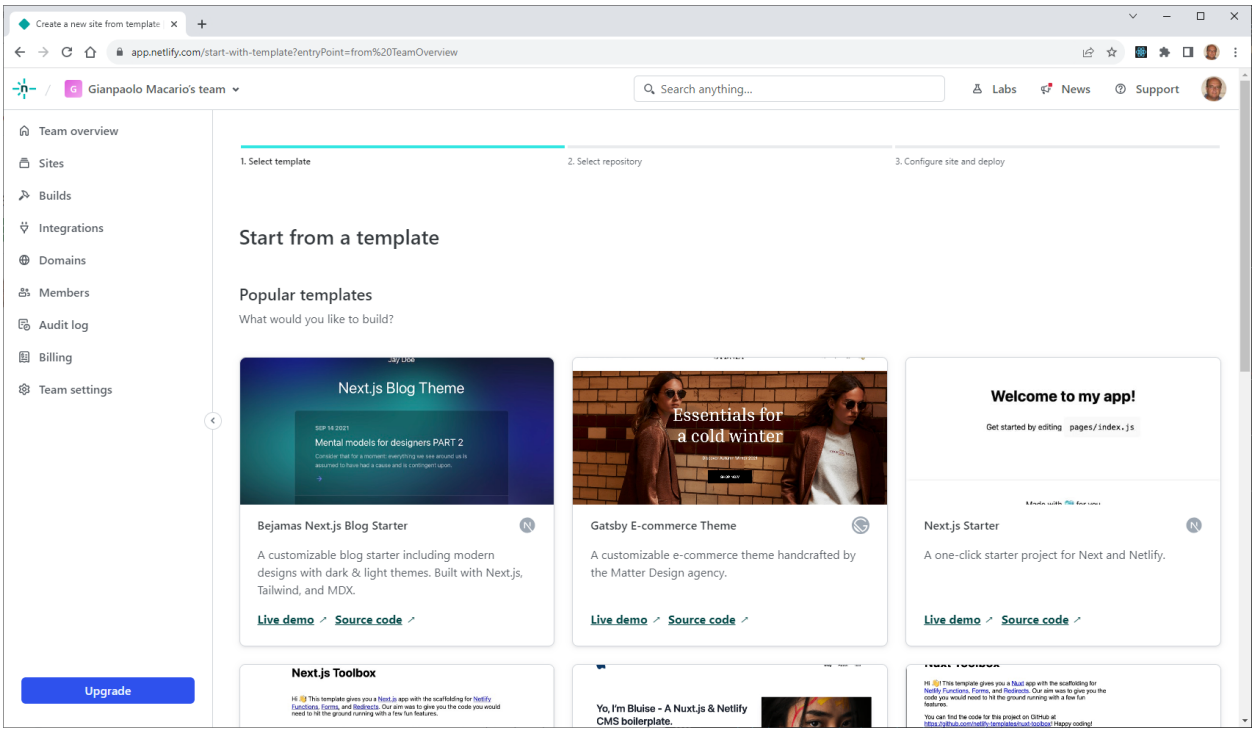
In the "Sites" panel, click **Add a new site**



If you already have a source project select the first option and provide the requested information.

In our case baroloteam.online#1 was still open so I chose **Start from a template** instead.

A few templates will be proposed which you can start from:



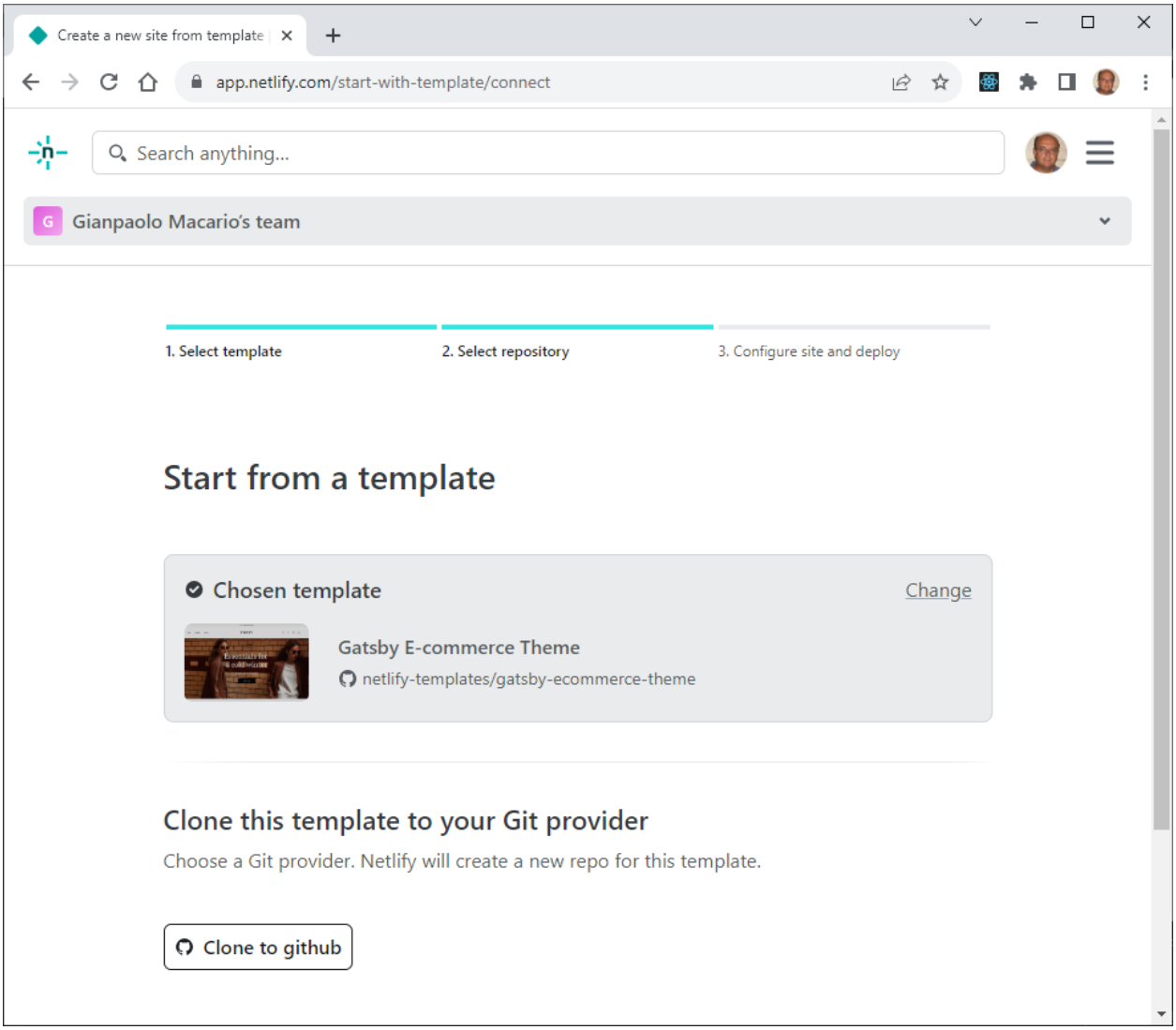
Choose one proposed template, or click **Browse more templates** to expand the list.

If you are still not satisfied, Google is your friend :wink:

You may also revert your previous choice and click the "Import from a Git repository" link.

In our example, I chose the "Gatsby E-commerce Theme": Live demo, Source code.

You will be presented with the following choices



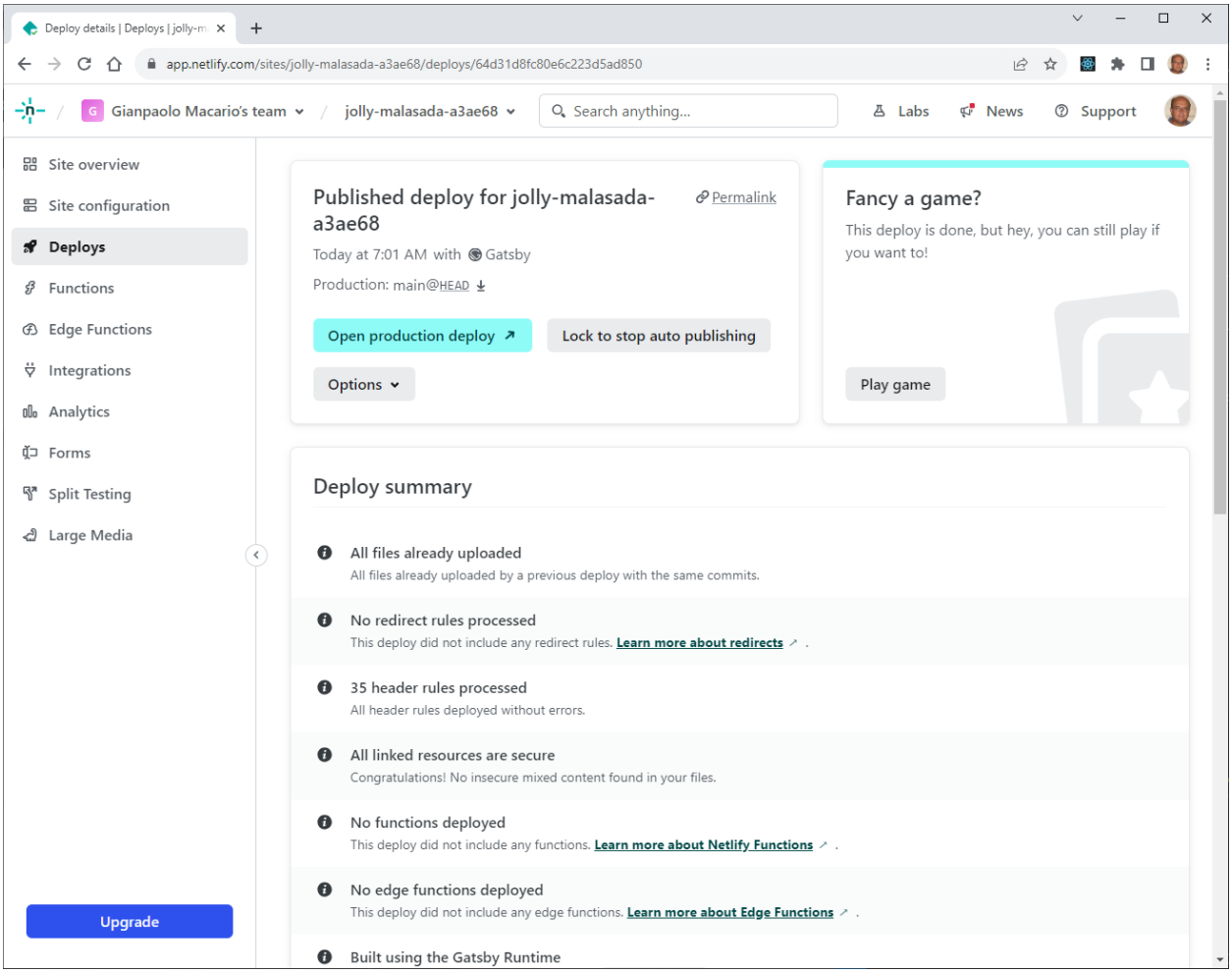
Click **Clone to GitHub**, then in section "Set up your site"

- Team: (accept default)
- Repository settings:
  - Repository name: gatsby-ecommerce-theme
  - Repository visibility: Public

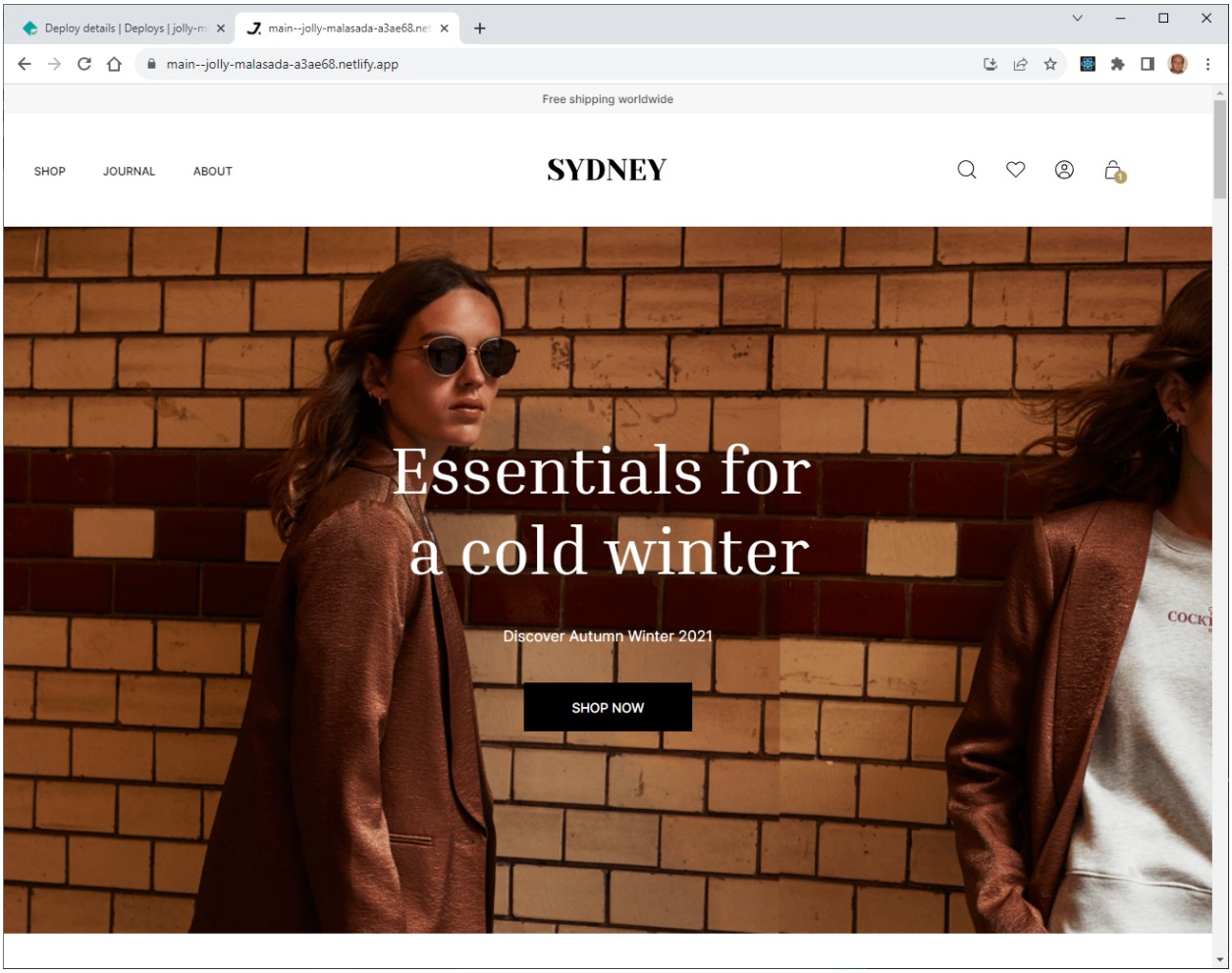
then click **Deploy site**.

This command will create a new public repository https://github.com/gmacario/gatsby-ecommerce-theme on GitHub, then will start building the site.

If the build is successful you should be presented a page like this:



Click on the **Open production deploy** link and browse the site which has been generated.



Apply a template to a source repository

We can easily apply template "gatsby-ecommerce-theme" to the source repository "baroloteam.online" using the following shell commands

```
# Clone source repository
mkdir -p ~/github/B-AROL-O
cd ~/github/B-AROL-O
git clone https://github.com/B-AROL-O/baroloteam.online

# Add a git remote for the template repository
cd ~/github/B-AROL-O/baroloteam.online
git remote add template https://github.com/netlify-templates/gatsby-ecommerce-theme
git fetch --all --prune

# Create a feature branch
git checkout main
git pull --all --prune
git checkout -b feat/apply-gatsby-template
git merge --allow-unrelated-histories template/main
```

The last command may return errors, such as

```
gianpaolo.macario@HW2457 MINGW64 ~/github/B-AROL-O/baroloteam.online (feat/apply-gatsby-
↪template)
$ git merge --allow-unrelated-histories template/main
Auto-merging README.md
CONFLICT (add/add): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

gianpaolo.macario@HW2457 MINGW64 ~/github/B-AROL-O/baroloteam.online (feat/apply-gatsby-
↪template|MERGING)
$
```

If so, inspect the problems with `git status` then

```
git status
vi README.md # Reconcile merge conflicts
git add -A
git commit -sm "Manually merge with template/main"
```

Update site contents

You may update the contents of the site by pushing new commits on the main branch of the source repository.

Configure Google Analytics

Please refer to *HOWTO Configure Google Analytics* for details.

TODO

Deploy to official URL

TODO

4.3 HOWTO Configure Google Analytics

4.3.1 Introduction

This document explains how to configure Google Analytics 4 and integrate it into your existing web site, for instance the one created following the *HOWTO Create a Landing Page on Netlify*.

4.3.2 Reference documents

A Founders Guide to Google Analytics: Part 1

In this session we'll cover the basics of Google Analytics 4, including how to surface AI driven insights for your customers and how to create and use reports

Presenters:

- Ainhoa Ybanez - Advertising Solutions Architect, Mobile Apps, Google

- Grant Kemp - Advertising Solutions Architect, Mobile Apps, Google

A Founders Guide to Google Analytics: Part 2

In this follow up session, we'll explore workspaces in GA4 and you'll learn how to build custom tables and visualizations of your data

Presenters:

- Ainhoa Ybanez - Advertising Solutions Architect, Mobile Apps, Google
- Grant Kemp - Advertising Solutions Architect, Mobile Apps, Google

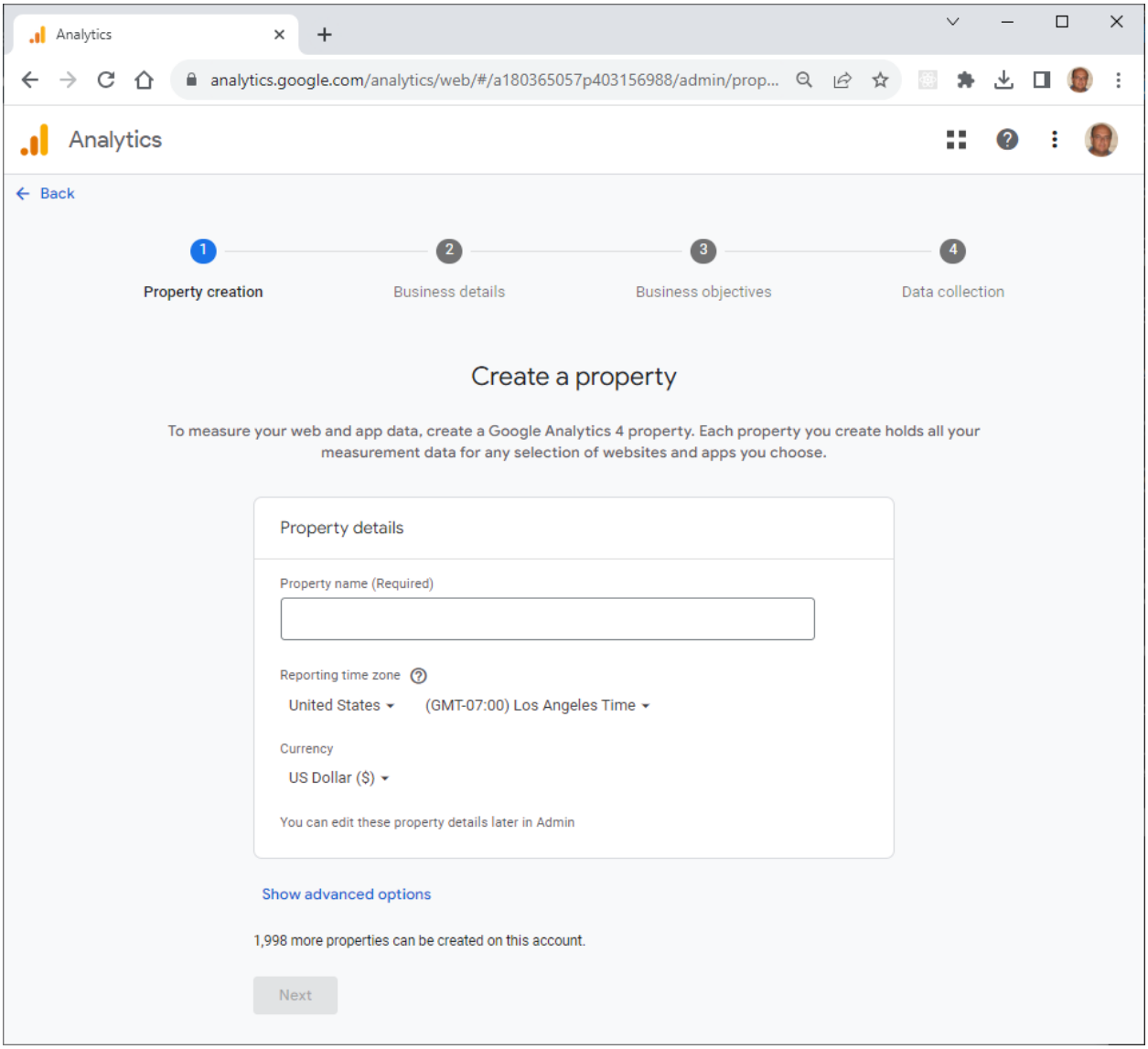
4.3.3 Step-by-step instructions

Create a new GA4 stream on Google Analytics

Login to https://analytics.google.com/

Create a new Analytics Account, or choose an existing one.

Click **Admin**, then **+ Create Property**. The following page should be displayed:

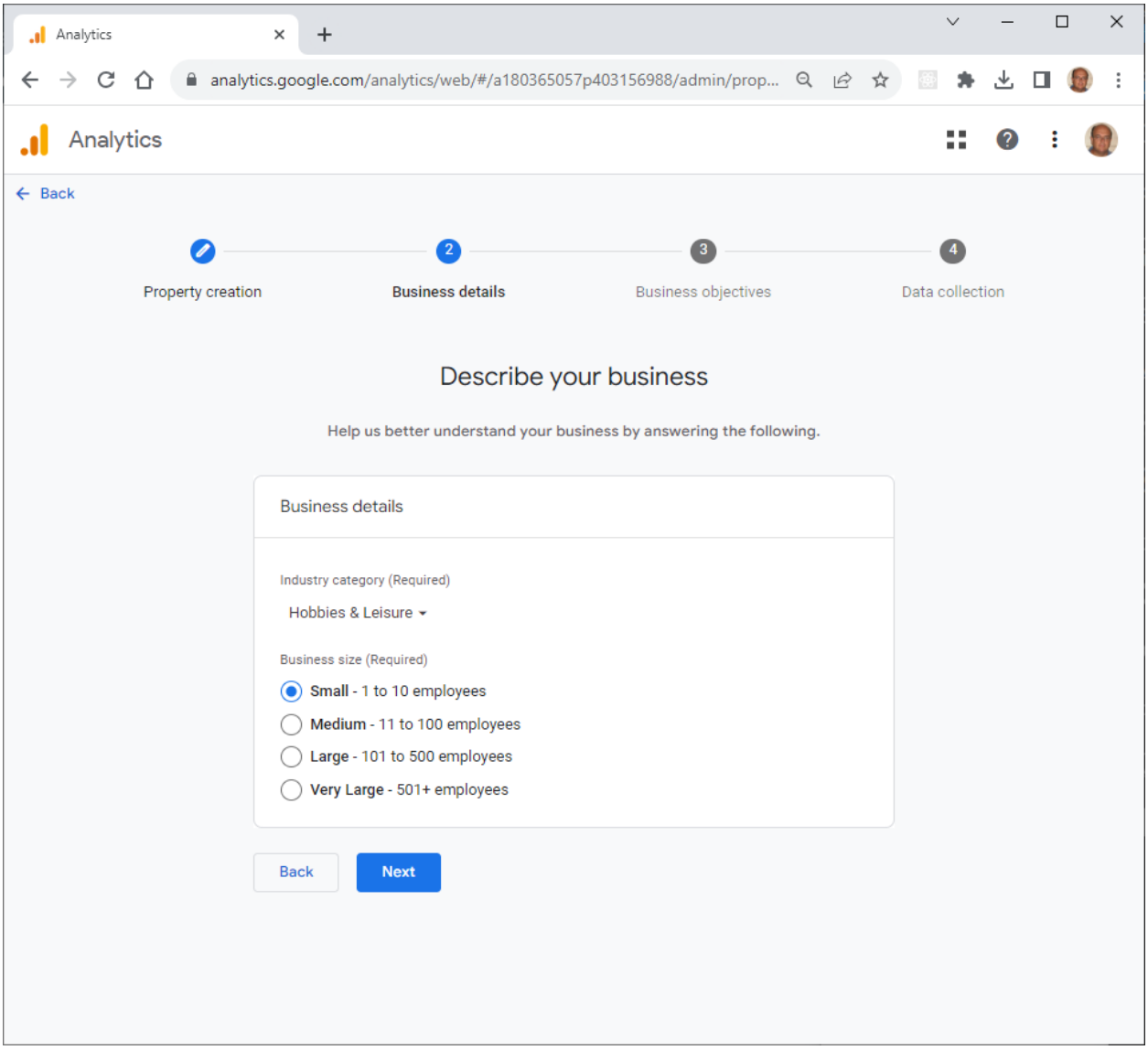


Fill in the Property details. In our example:

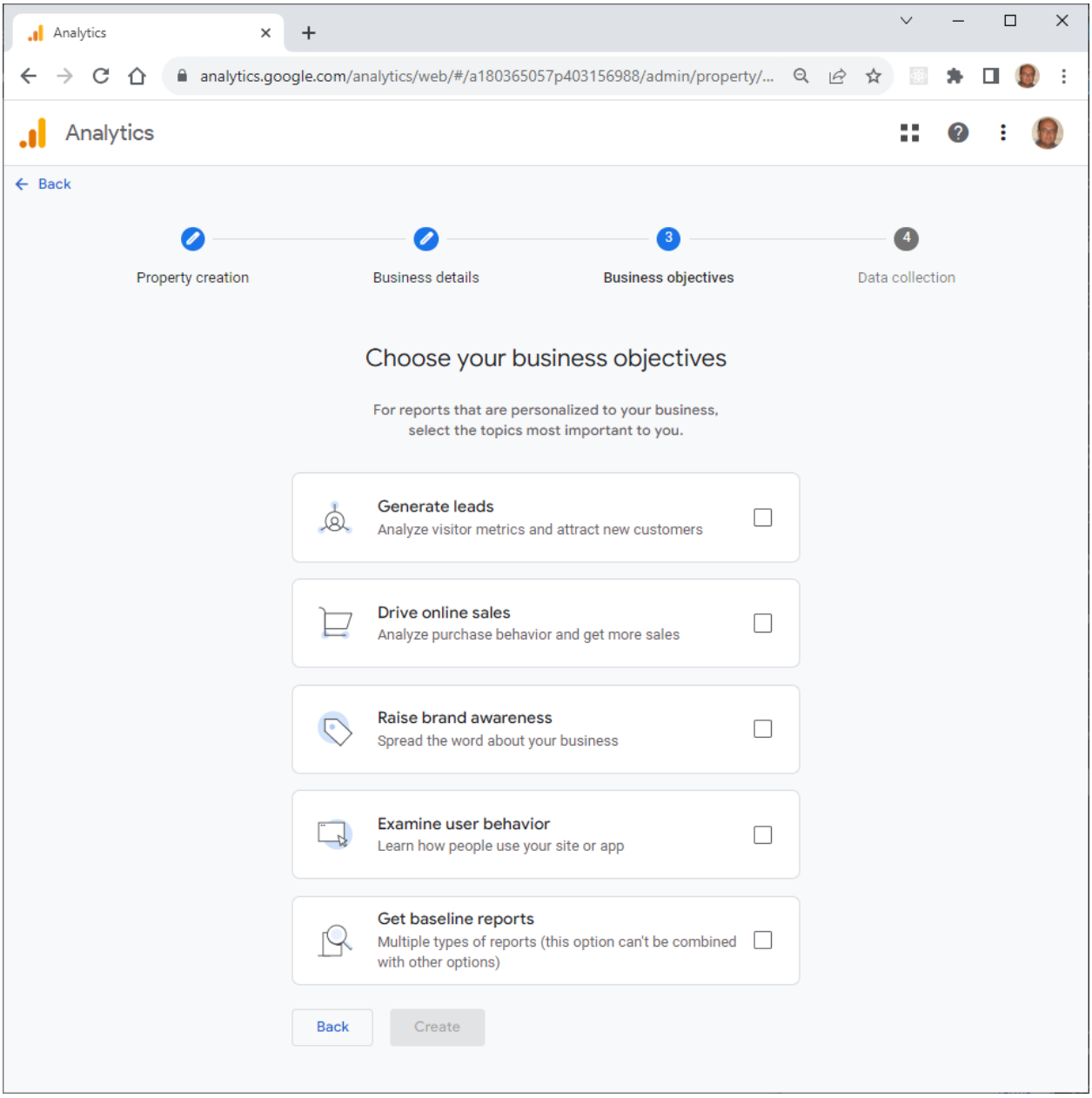
- Property name (Required): `baroloteam`
- Reporting time zone: Italy
- Currency: Euro (€)

then click **Next** and fill in our Business details. In our example:

- Industry category (Required): Hobbies & Leisure
- Business size (Required): Small - 1 to 10 employees



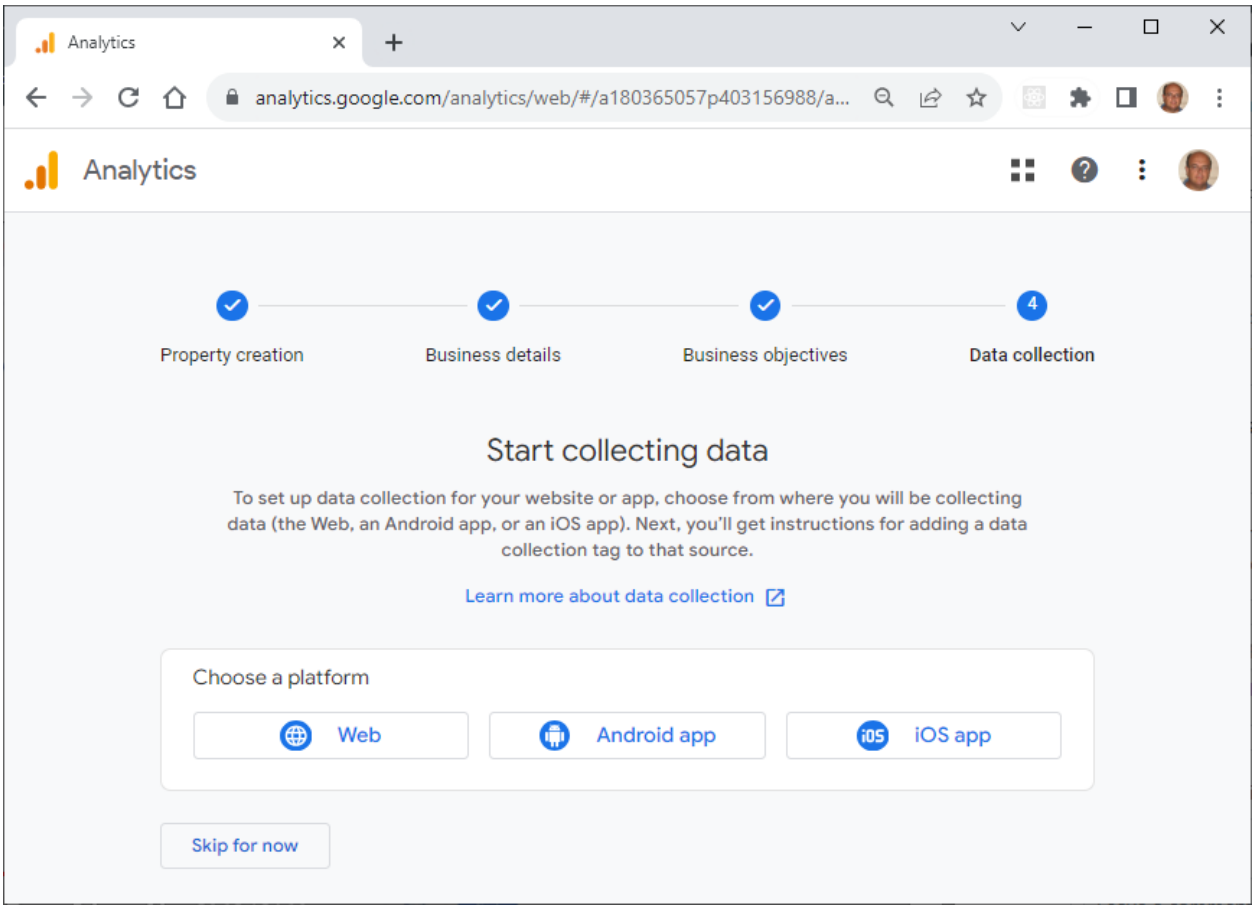
Click **Next**, then choose our business objectives.



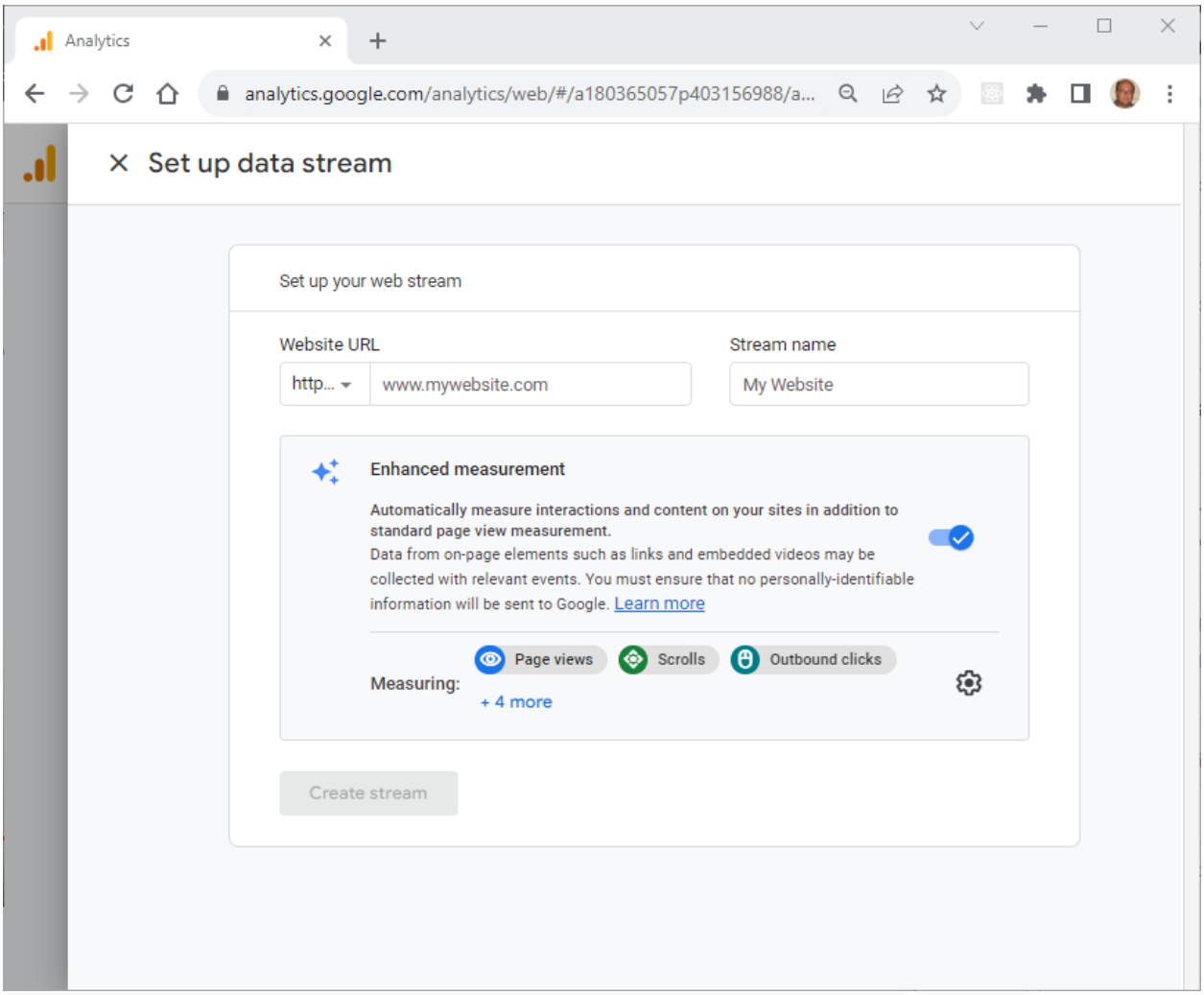
In our example:

- [x] **Generate leads**
- [ ] **Drive online sales**
- [x] **Raise brand awareness**
- [x] **Examine user behavior**
- [ ] **Get baseline reports**

When you are done, click **Create**. The wizard will show the following page:



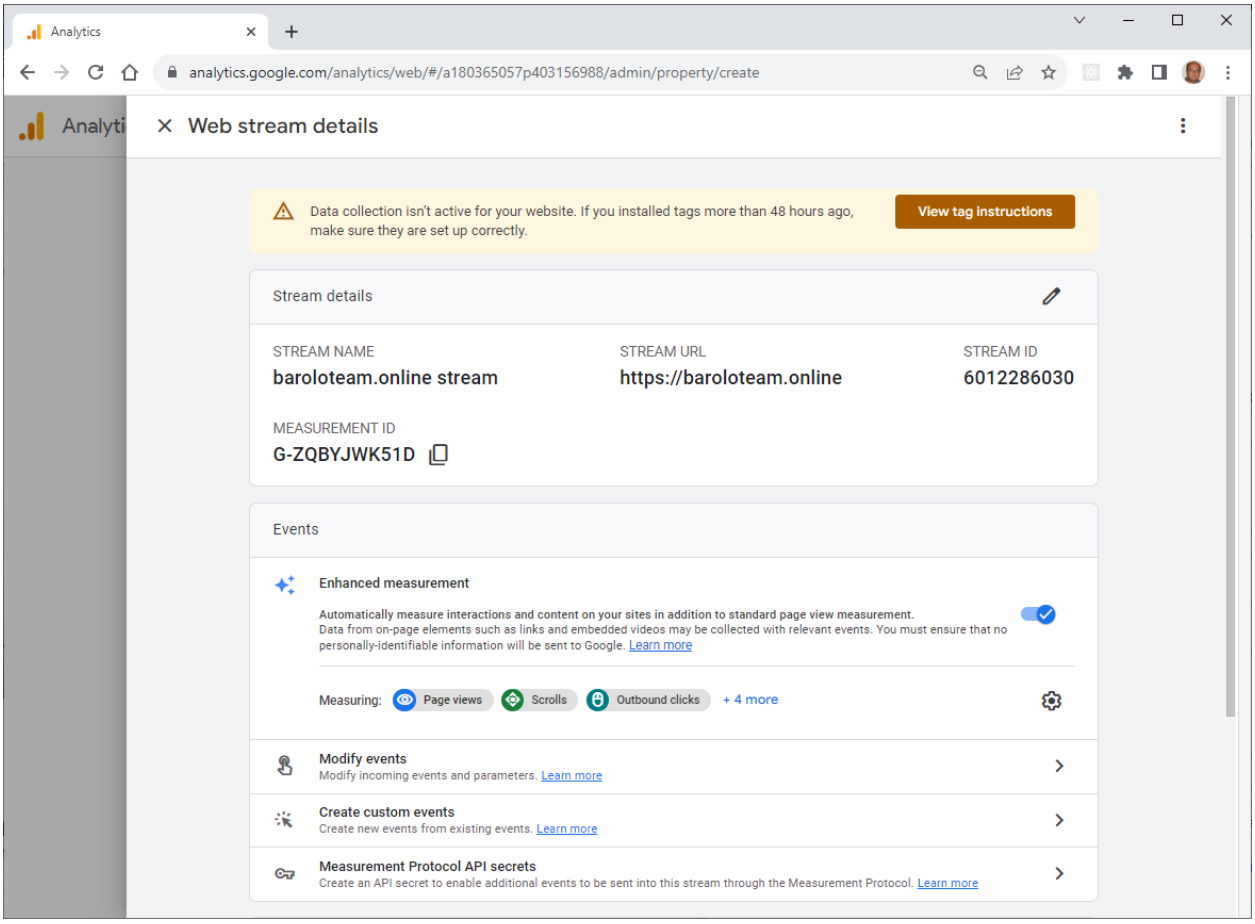
In our example we only choose “Web”. We should now set up our web stream.



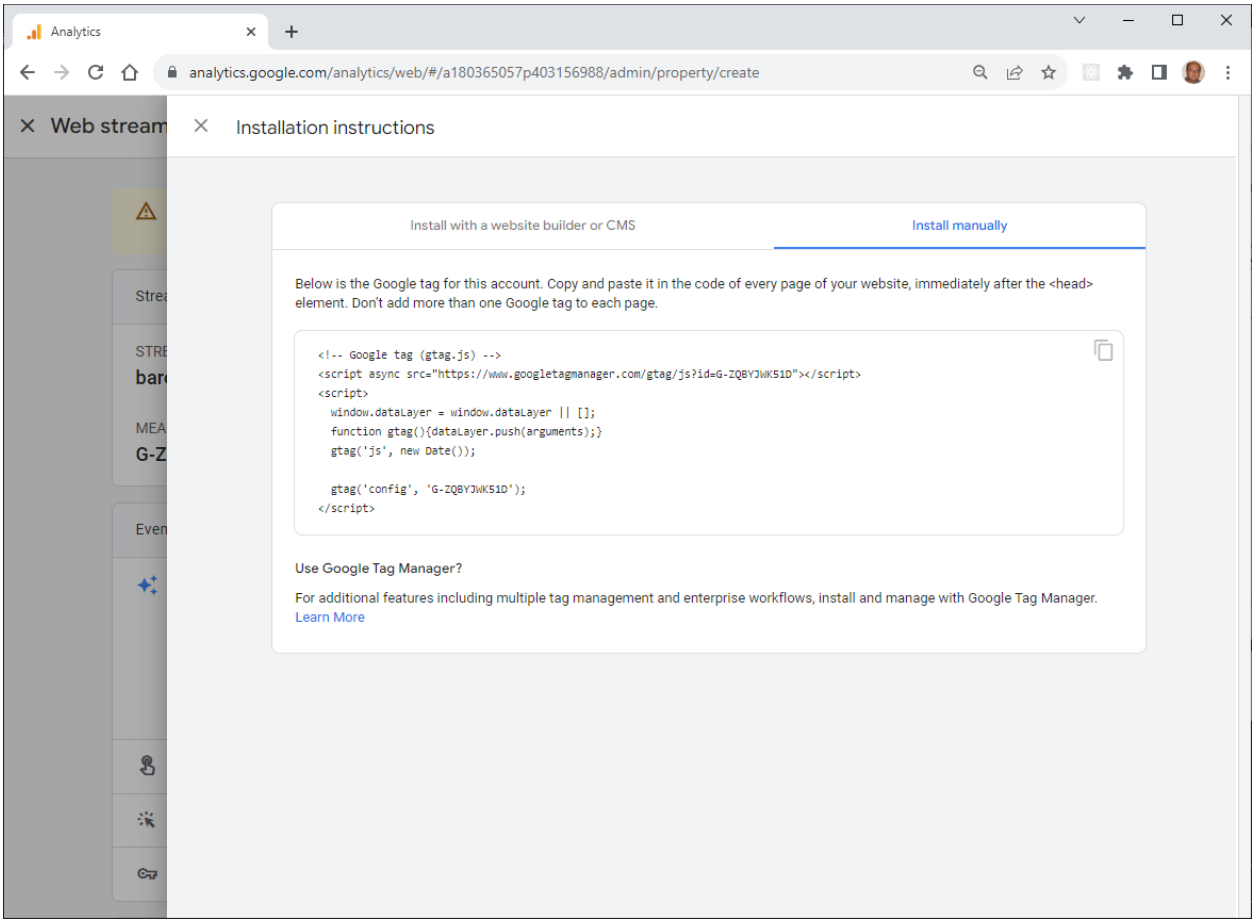
Fill in the required information - in our example:

- Website URL: https://baroloteam.online
- Stream name: baroloteam.online stream
- Enhanced measurements: (leave defaults)

then click **Create stream**. After a few seconds the stream details will be displayed:



Click **View tag instructions** to display code snippet to be injected into your website. In our example:



Inject GA4 scripts into a Gatsby site

Reference: https://www.gatsbyjs.com/docs/how-to/testing/ab-testing-with-google-analytics-and-netlify/

Edit file `src/components/header.js`

NOTE: For a site based on the https://github.com/netlify-templates/gatsby-ecommerce-theme template the file is called `src/components/Header/Header.js`.

TODO

TODO

4.4 HOWTO Install Eclipse Capella

4.4.1 Executive Summary

This document explains how to install Eclipse Capella™ - an Open Source solution for Model-Based Systems Engineering.

4.4.2 Introduction to Eclipse Capella™

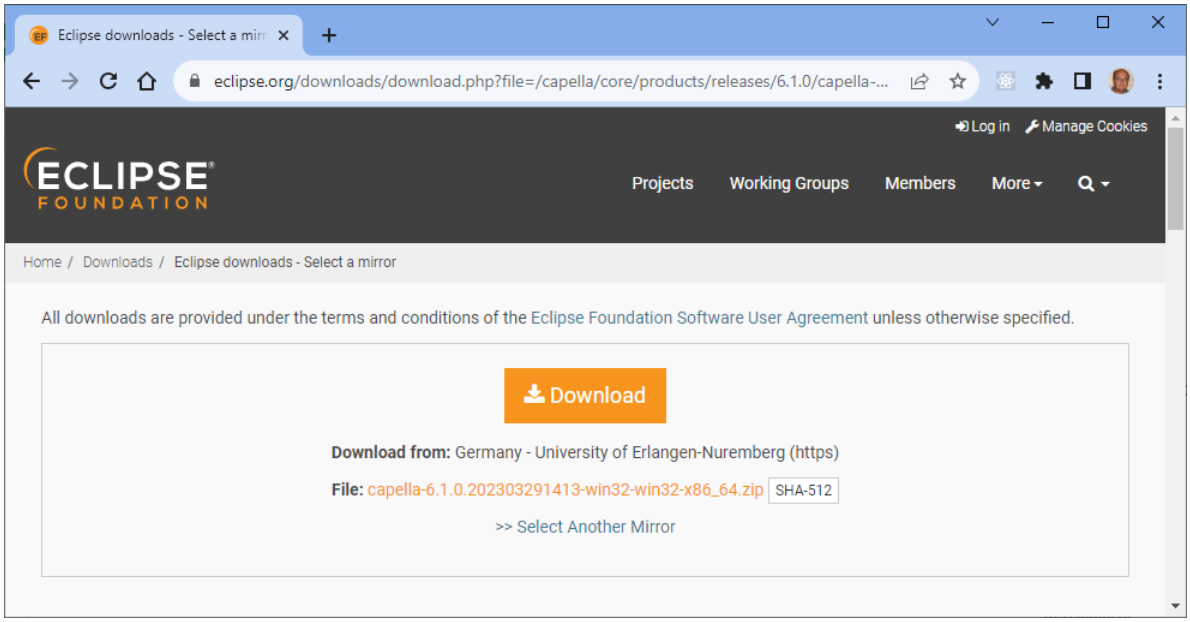
Eclipse Capella™ is a comprehensive, extensible, and field-proven MBSE tool and method to design systems architecture successfully.

References

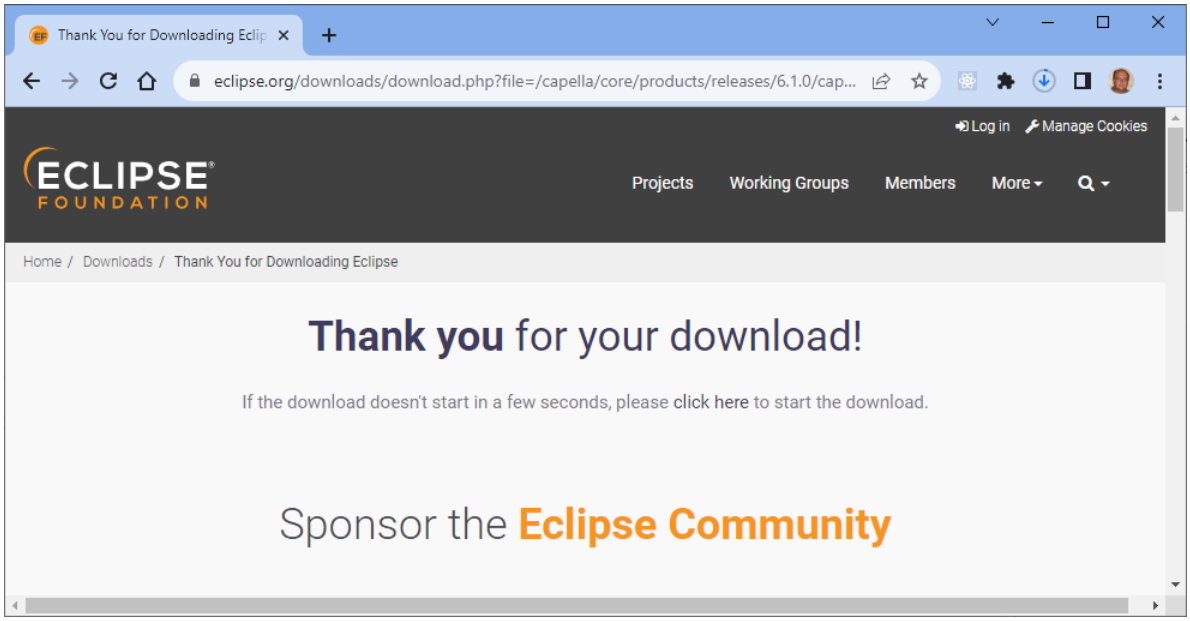
- https://www.eclipse.org/capella/

4.4.3 Install Capella (on MS Windows)

1. From your browser open https://www.eclipse.org/capella/download.html
2. Click "GET CAPELLA 6.1 FOR WINDOWS 64-BIT".



You may select another mirror if you like. Otherwise, click “Download”.



You may also support the Eclipse Community by donating some money.

- Result: file `capella-6.1.0.202303291413-win32-win32-x86_64.zip` will be downloaded

3. Extract the file from the ZIP archive.

```
mkdir /c/opt/capella
cd /c/opt/capella
unzip $HOME/Downloads/capella-*.zip
ls -la
```

Result:

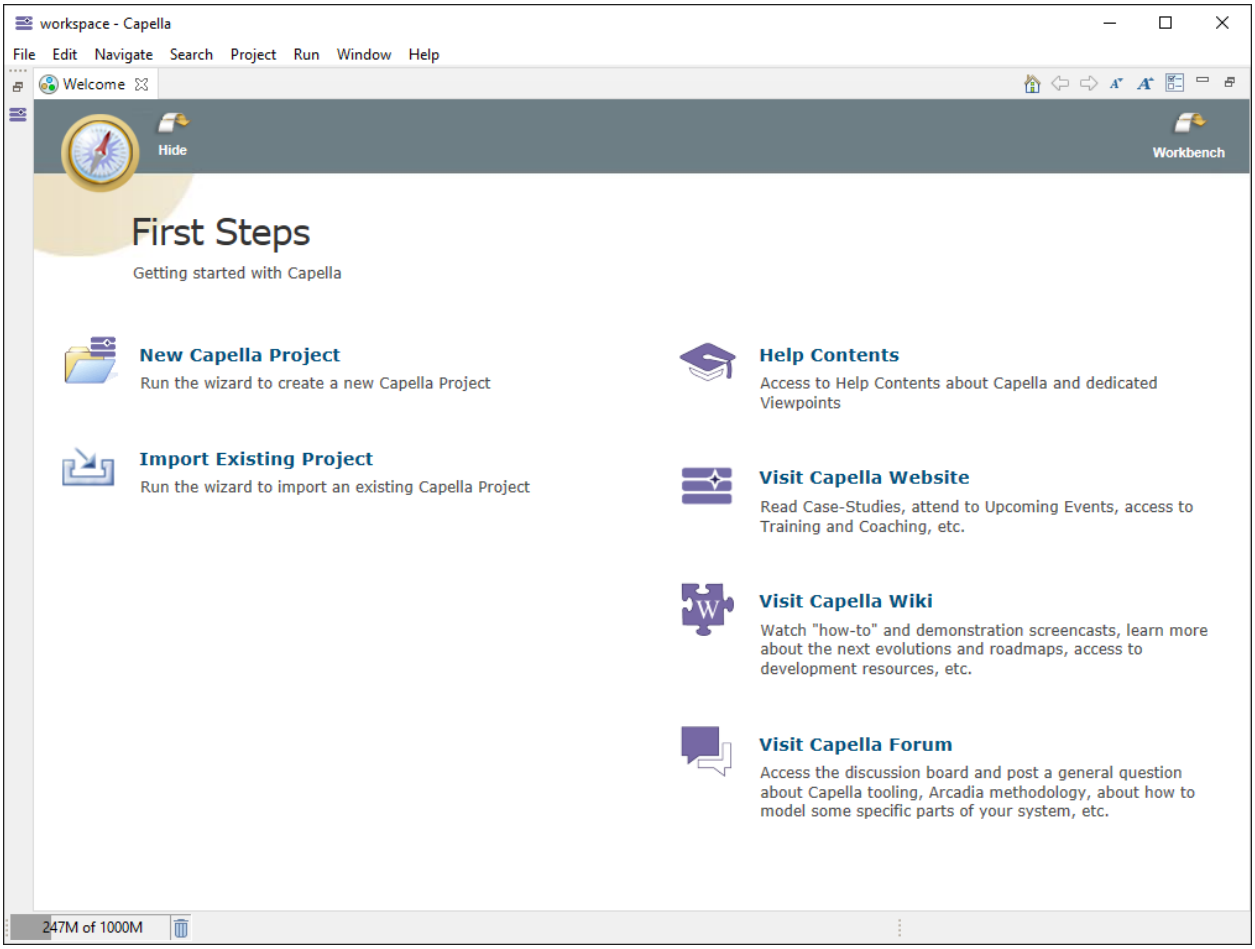
```
gianpaolo.macario@HW2457 MINGW64 /c/opt/capella
$ ls -la *
capella:
total 2226
drwxr-xr-x 1 gianpaolo.macario 1049089      0 Oct  9 13:20 ./
drwxr-xr-x 1 gianpaolo.macario 1049089      0 Oct  9 13:21 ../
-rw-r--r-- 1 gianpaolo.macario 1049089     61 Jun 11  2021 .eclipseproduct
-rw-r--r-- 1 gianpaolo.macario 1049089 446509 Mar 29  2023 artifacts.xml
-rwxr-xr-x 1 gianpaolo.macario 1049089 426152 Mar 29  2023 capella.exe*
-rw-r--r-- 1 gianpaolo.macario 1049089    304 Mar 29  2023 capella.ini
-rwxr-xr-x 1 gianpaolo.macario 1049089 131240 Mar 29  2023 capellac.exe*
drwxr-xr-x 1 gianpaolo.macario 1049089      0 Oct  9 13:20 configuration/
drwxr-xr-x 1 gianpaolo.macario 1049089      0 Oct  9 13:20 dropins/
-rw-r--r-- 1 gianpaolo.macario 1049089  16536 Jun 29  2022 epl-v10.html
drwxr-xr-x 1 gianpaolo.macario 1049089      0 Oct  9 13:20 features/
drwxr-xr-x 1 gianpaolo.macario 1049089      0 Oct  9 13:20 jre/
-rw-r--r-- 1 gianpaolo.macario 1049089   9230 Jun 29  2022 notice.html
drwxr-xr-x 1 gianpaolo.macario 1049089      0 Oct  9 13:20 p2/
drwxr-xr-x 1 gianpaolo.macario 1049089      0 Oct  9 13:21 plugins/
drwxr-xr-x 1 gianpaolo.macario 1049089      0 Oct  9 13:21 readme/

samples:
total 3008
drwxr-xr-x 1 gianpaolo.macario 1049089       0 Oct  9 13:21 ./
drwxr-xr-x 1 gianpaolo.macario 1049089       0 Oct  9 13:21 ../
-rw-r--r-- 1 gianpaolo.macario 1049089 3077889 Mar 29  2023 IFE_samplemodel.zip

gianpaolo.macario@HW2457 MINGW64 /c/opt/capella
$
```

You can now launch Capella with the following command-line:

```
/c/opt/capella/capella/capella.exe
```

The Capella splash screen should be displayed.

Once you confirm the workspace directory, after a few seconds Capella main windows should eventually be displayed:



4.4.4 Install Capella (on Ubuntu 64-bit)

1. Go to Capella - Downloads
   - Look at the section "Other Platforms" > "Linux".
   - If you have an Intel CPU, choose "64-bit"; for ARM 64-bit, choose "aarch64".
   - Result: file `capella-6.1.0.202303291413-linux-gtk-x86_64.tar.gz` will be downloaded
2. Extract the file from the tar archive:

```
sudo mkdir -p /opt/capella
sudo tar -C /opt/capella -xvzf capella-*.tar.gz
```

3. You can launch Capella with the following command-line

```
/opt/capella/capella/capella
```

4.4.5 Install Capella (on macOS)

1. Go to Capella - Downloads and choose the file right for your operating system:
   - If you have an Intel Mac, choose "64-bit"; otherwise, choose "aarch64".
2. Extract the file from the tar archive.
3. Drag the application into the Applications folder.
4. Launch terminal.
5. Copy and paste the following command:

```
xattr -d com.apple.quarantine /Applications/Capella.app
```

Then, the program is ready to be used.

4.5 HOWTO Mermaid utilization and code examples

Mermaid is a useful tool to create diagrams and charts directly using Markdown. The good part about it is that those diagrams can be easily modified just by changing a code displayed before the image, however we can't just move the image wherever we want to. For this reason the Mermaid Live Editor is better in case we are just interested in the results.

In order to create a Mermaid graph:

- Use a *code block* tagged with language `mermaid`.
- write the code of the diagram (the diagram syntax can be find here, where all the kinds of diagram are stated with examples and codes)
- as a result, the graph will appear. There are a lot of configurable settings like the background or the color of the boxes. Everything can be found in their site.

4.5.1 Example

```
graph TD;
    A-->B;
    A-->C;
    B-->D;
    C-->D;
```

NOTE: be careful, the background of the diagram depends on the style chosen on GitHub. So if you have the dark theme, it will be dark, making the diagram not understandable. They are trying to implement a new feature to solve this problem but for now I suggest you to use the theme dark in the code by using `%%{init:{'theme':'dark'}}%%` at the beginning of the code (on their site it is well explained and there are multiple examples).

4.5.2 Other links

- https://ardalis.com/github-diagrams-with-mermaid/
- https://github.com/mermaid-js/mermaid
- https://github.com/github/roadmap/issues/372
- https://mermaid-js.github.io/
- https://www.youtube.com/watch?v=-HUwt8dF4X8

4.6 HOWTO Use rclone with Cubbit DS3

4.6.1 Executive Summary

This guide explains how to use the rclone tool to access contents and backup data inside a bucket on Cubbit DS3 Object Storage.

4.6.2 Reference Documents

- Rclone landing page
- GitHub: rclone/rclone
- Cubbit docs > Integrations > Rclone

4.6.3 Introduction to `rclone`

Rclone ("*rsync for cloud storage*") is a command-line program to sync files and directories to and from different cloud storage providers.

From https://rclone.org/:

> Rclone is a command-line program to manage files on cloud storage. It is a feature-rich alternative to cloud vendors' web storage interfaces. Over 70 cloud storage products support rclone including S3 object stores, business & consumer file storage services, as well as standard transfer protocols.

Official instructions for using rclone with Cubbit DS3 are available at https://docs.cubbit.io/integrations/rclone

4.6.4 Installing `rclone`

The page at https://rclone.org/install/ provide detailed instructions for installing `rclone` under many Operating Systems and HW architectures.

The following sections detail the results performing the installation on some relevant hosts.

Install rclone on Ubuntu 22.04.3 LTS

(Tested on host HW2228)

Check if rclone is already installed:

```
gmacario@hw2228:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.3 LTS
Release:        22.04
Codename:       jammy
gmacario@hw2228:~$ rclone
Command 'rclone' not found, but can be installed with:
sudo snap install rclone  # version 1.64.0, or
sudo apt  install rclone  # version 1.53.3-4ubuntu1.22.04.2
See 'snap info rclone' for additional versions.
gmacario@hw2228:~$
```

A more recent version of `rclone` (1.64.0 as of 2023-11-05) is available via snap, therefore let us install `rclone` with the following command:

```
sudo snap install rclone
```

Verify that the installation was performed correctly:

```
gmacario@hw2228:~$ rclone version
rclone v1.64.0
- os/version: ubuntu 22.04 (64 bit)
- os/kernel: 6.2.0-36-generic (x86_64)
- os/type: linux
- os/arch: amd64
- go/version: go1.21.1
- go/linking: static
- go/tags: snap
gmacario@hw2228:~$
```

Installing rclone on Windows 10 Pro

(Tested on host ALPHA)

Download the installation file for the Windows OS from https://rclone.org/downloads/ (as of 2023-11-05 this is `rclone-v1.64.2-windows-amd64.zip`), then uncompress the `.zip` archive and copy the `rclone.exe` file into a directory listed in the `PATH` environment variable.

You may verify that the installation was performed successfully by typing `rclone --version` from the command-line prompt.

Example (from Git bash)

```
gmaca@alpha MINGW64 ~
$ which rclone
/c/Users/gmaca/bin/rclone

gmaca@alpha MINGW64 ~
$ rclone --version
rclone v1.64.2
- os/version: Microsoft Windows 10 Pro 22H2 (64 bit)
- os/kernel: 10.0.19045.3636 (x86_64)
- os/type: windows
- os/arch: amd64
- go/version: go1.21.3
- go/linking: static
- go/tags: cmount

gmaca@alpha MINGW64 ~
$
```

4.6.5 Configuring rclone for Cubbit DS3

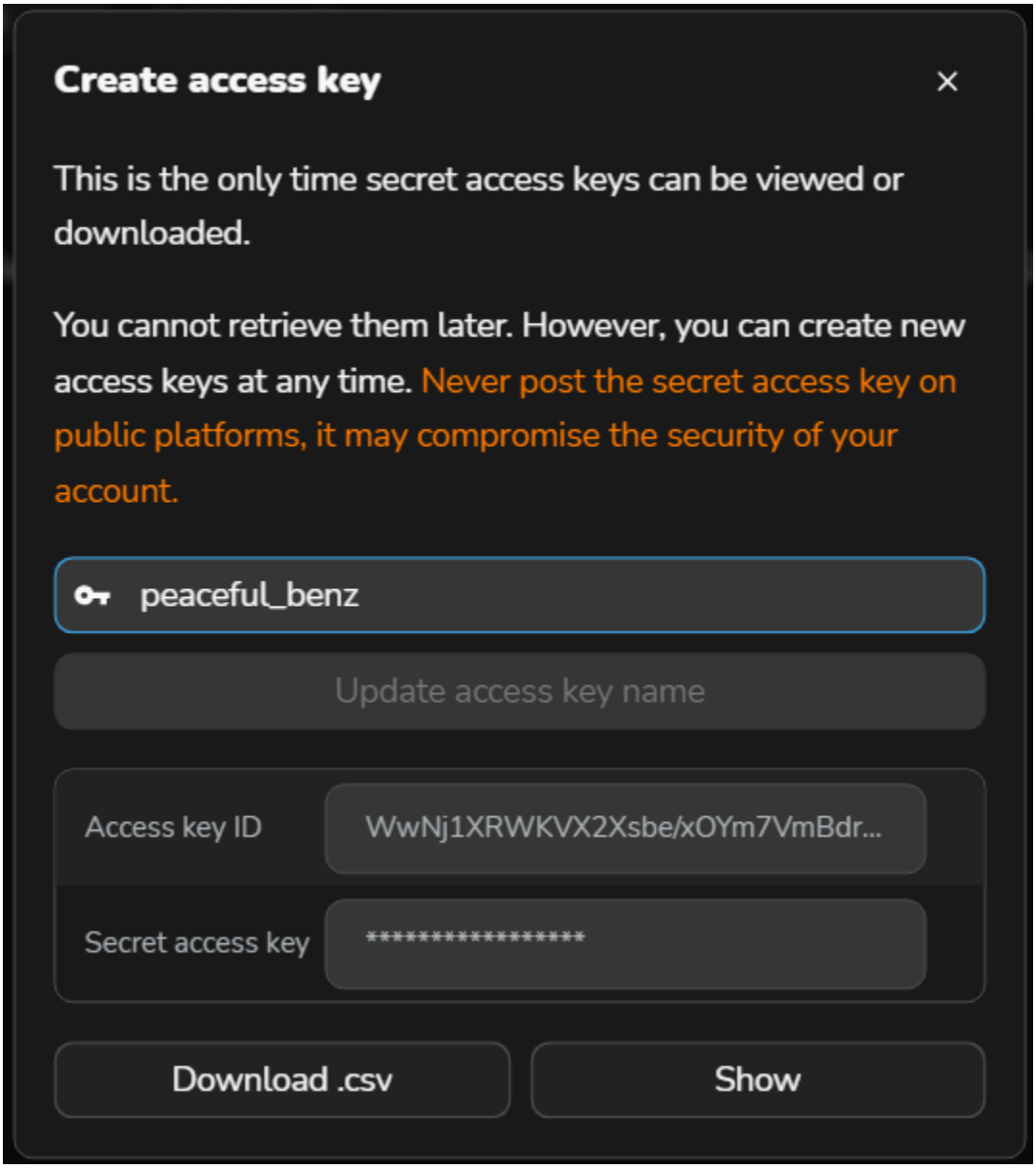
After the `rclone` tool has been installed it must be configured in order to access the data stored on Cubbit DS3.

NOTE: A similar procedure can be used to configure rclone for other object storage backends, for instance Amazon Simple Storage Service (Amazon S3) or Azure Blob Storage.

Create Access Key ID and Secret for Cubbit DS3

Login to Cubbit DS3 Object Storage from https://www.cubbit.io/it/log-in

From your Workspace click "Settings > API keys", then click "+ Generate new client API key"



Accept the proposed key name, or change it if you prefer, then click "Download .csv". This file contains the Access key ID and Secret access key which will be used to configure a S3 client application such as S3 Browser.

Run `rclone config`

Prerequisite: `snap` installed as detailed in the previous chapter.

Run the `rclone config` command and type the following options (you must replace `**********` with the values from the `.csv` file)

```
n                       # e/n/d/r/c/s/q> (New remote)
cubbit                  # name> (Enter name for new remote.)
5                       # Storage> (Amazon S3 Compliant Storage Provider)
29                      # provider> (Any other S3 compatible provider)
1                       # env_auth> (Enter AWS credentials in the next step)
```

(continues on next page)

(continued from previous page)

```
************          # access_key_id>
************          # secret_access_key>
eu-west-1             # region>
https://s3.cubbit.eu  # endpoint> (Required when using an S3 clone.)
                      # location_constraint> (Press Enter to leave empty)
1                     # acl> (Owner gets FULL_CONTROL.No one else has access rights␣
↪(default).(private))
n                     # Edit advanced config?
y                     # y/e/d> (Yes this is OK)
q                     # e/n/d/r/c/s/q> (Quit config)
```

Once the `rclone config` command is complete the following command dumps the config file as JSON:

```
rclone config dump
```

4.6.6 Some useful `rclone` commands

List files in a bucket

```
rclone ls cubbit:${BUCKET_NAME}
```

Example result:

```
gmacario@hw2228:~$ rclone ls cubbit:baroloteam
  4985990 IMG_20230826_124002_1.jpg
  4421972 IMG_20230826_124009_1.jpg
    64546 newplot.png
gmacario@hw2228:~$
```

An error is returned if the bucket does not exist or we have no access:

```
gmacario@hw2228:~$ rclone ls cubbit:newbucket
2023/11/05 11:05:33 Failed to ls: LambdaRuntimeError: Forbidden
        status code: 403, request id: , host id:
gmacario@hw2228:~$
```

NOTE: The `rclone ls` command has a number of useful options which may be discovered with the `rclone ls --help` command:

```
gmacario@hw2228:~$ rclone ls --help

Lists the objects in the source path to standard output in a human
readable format with size and path. Recurses by default.

Eg

    $ rclone ls swift:bucket
        60295 bevajer5jef
        90613 canole
        94467 diwogej7
        37600 fubuwic
```

(continues on next page)

(continued from previous page)

```
Any of the filtering options can be applied to this command.

There are several related list commands

  * `ls` to list size and path of objects only
  * `lsl` to list modification time, size and path of objects only
  * `lsd` to list directories only
  * `lsf` to list objects and directories in easy to parse format
  * `lsjson` to list objects and directories in JSON format

`ls`,`lsl`,`lsd` are designed to be human-readable.
`lsf` is designed to be human and machine-readable.
`lsjson` is designed to be machine-readable.

Note that `ls` and `lsl` recurse by default - use `--max-depth 1` to stop the recursion.

The other list commands `lsd`,`lsf`,`lsjson` do not recurse by default - use `-R` to␣
↪make them recurse.

Listing a nonexistent directory will produce an error except for
remotes which can't have empty directories (e.g. s3, swift, or gcs -
the bucket-based remotes).

Usage:
  rclone ls remote:path [flags]

Flags:
  -h, --help   help for ls


# Filter Flags

Flags for filtering directory listings.

      --delete-excluded                     Delete files on dest excluded from sync
      --exclude stringArray                 Exclude files matching pattern
      --exclude-from stringArray            Read file exclude patterns from file (use -␣
↪to read from stdin)
      --exclude-if-present stringArray      Exclude directories if filename is present
      --files-from stringArray              Read list of source-file names from file␣
↪(use - to read from stdin)
      --files-from-raw stringArray          Read list of source-file names from file␣
↪without any processing of lines (use - to read from stdin)
  -f, --filter stringArray                  Add a file filtering rule
      --filter-from stringArray             Read file filtering patterns from a file␣
↪(use - to read from stdin)
      --ignore-case                         Ignore case in filters (case insensitive)
      --include stringArray                 Include files matching pattern
      --include-from stringArray            Read file include patterns from file (use -␣
↪to read from stdin)
```

(continues on next page)

(continued from previous page)

```
      --max-age Duration                    Only transfer files younger than this in s␣
↪or suffix ms|s|m|h|d|w|M|y (default off)
      --max-depth int                       If set limits the recursion depth to this␣
↪(default -1)
      --max-size SizeSuffix                 Only transfer files smaller than this in KiB␣
↪or suffix B|K|M|G|T|P (default off)
      --metadata-exclude stringArray        Exclude metadatas matching pattern
      --metadata-exclude-from stringArray   Read metadata exclude patterns from file␣
↪(use - to read from stdin)
      --metadata-filter stringArray         Add a metadata filtering rule
      --metadata-filter-from stringArray    Read metadata filtering patterns from a file␣
↪(use - to read from stdin)
      --metadata-include stringArray        Include metadatas matching pattern
      --metadata-include-from stringArray   Read metadata include patterns from file␣
↪(use - to read from stdin)
      --min-age Duration                    Only transfer files older than this in s or␣
↪suffix ms|s|m|h|d|w|M|y (default off)
      --min-size SizeSuffix                 Only transfer files bigger than this in KiB␣
↪or suffix B|K|M|G|T|P (default off)

# Listing Flags

Flags for listing directories.

      --default-time Time   Time to show if modtime is unknown for files and directories␣
↪(default 2000-01-01T00:00:00Z)
      --fast-list           Use recursive list if available; uses more memory but fewer␣
↪transactions


Additional help topics:

Use "rclone [command] --help" for more information about a command.
Use "rclone help flags" for to see the global flags.
Use "rclone help backends" for a list of supported services.
gmacario@hw2228:~$
```

Sync a local folder to a bucket on Cubbit DS3

```
rclone sync -P ./backup-folder cubbit:${BUCKET_NAME}
```

Note that `BUCKET_NAME` must exist otherwise the command will fail (it may be created from https://console.cubbit.eu/)

On the other hand, if you sync files to a subfolder of a bucket, the subfolder will be created automatically if it does not exist.

Example 1

```
gmacario@gmpowerhorse:~ $ rclone sync -P ~/Downloads cubbit:bk-gmpowerhorse/test02
Transferred:      146.829M / 146.829 MBytes, 100%, 7.582 MBytes/s, ETA 0s
Errors:                 0
Checks:                 0 / 0, -
Transferred:           17 / 17, 100%
Elapsed time:       19.3s
gmacario@gmpowerhorse:~ $
```

Example 2

```
cd ~/Dropbox/Cubbit_MIRROR/CUBBIT_SHARED/SHARED_WITH_BAROLOTEAM
rclone sync -P . cubbit:baroloteam-from-dropbox
```

Result:

```
gmaca@alpha MINGW64 ~/Dropbox/Cubbit_MIRROR/CUBBIT_SHARED/SHARED_WITH_BAROLOTEAM
$ rclone sync -P . cubbit:baroloteam-from-dropbox
Transferred:        4.945 GiB / 4.945 GiB, 100%, 117.173 KiB/s, ETA 0s
Transferred:          724 / 724, 100%
Elapsed time:      33m8.1s

gmaca@alpha MINGW64 ~/Dropbox/Cubbit_MIRROR/CUBBIT_SHARED/SHARED_WITH_BAROLOTEAM
$
```

Example 3

```
cd ~/Dropbox/Cubbit_MIRROR
rclone sync -P . cubbit:mirror-from-dropbox
```

Result:

```
gmaca@alpha MINGW64 ~/Dropbox/Cubbit_MIRROR
$ rclone sync -P . cubbit:mirror-from-dropbox
...
2023/11/09 22:21:17 ERROR : Win10-images/Win10_21H1_Italian_x64.iso: Failed to copy:␣
↪multi-thread copy: failed to open source: The file cannot be accessed by the system.
2023/11/09 22:21:18 ERROR : ARCHIVE/2023-07-15-google-takeout/Tutti i messaggi compresi␣
↪Spam e Cestino-005.mbox: Failed to copy: multi-thread copy: failed to open source:
 The file cannot be accessed by the system.
Transferred:        3.285 GiB / 211.470 GiB, 2%, 241.274 KiB/s, ETA 1w3d11h
Errors:                 4 (retrying may help)
Checks:              3292 / 5125, 64%
Transferred:          295 / 2853, 10%
Elapsed time:    2h2m18.9s
Checking:
 * CUBBIT_SHARED/SHARED_W...30830_17_02_10_Pro.jpg: checking
 * CUBBIT_SHARED/SHARED_W...30830_17_02_11_Pro.jpg: checking
```

(continues on next page)

(continued from previous page)

```
 * CUBBIT_SHARED/SHARED_W...30830_17_02_12_Pro.jpg: checking
 * CUBBIT_SHARED/SHARED_W...30830_17_20_12_Pro.jpg: checking
 * CUBBIT_SHARED/SHARED_W...30830_17_20_13_Pro.jpg: checking
 * CUBBIT_SHARED/SHARED_W...30830_17_20_16_Pro.jpg: checking
 * CUBBIT_SHARED/SHARED_W...30830_17_20_22_Pro.jpg: checking
 * CUBBIT_SHARED/SHARED_W...30830_17_20_23_Pro.jpg: checking
Transferring:
 * ARCHIVE/2023-07-15-goo...230715T062408Z-003.zip:  0% /1.984Gi, 9.408Ki/s, 60h48m14
 * ARCHIVE/2023-07-15-goo...230715T062408Z-004.zip:  1% /1.611Gi, 46.785Ki/s, 9h54m17
 * warez/2019-09-04-solid...dWorks 2018.part01.rar:  3% /1Gi, 45.942Ki/s, 6h5m31s
 * warez/2019-09-04-solid...dWorks 2018.part02.rar:  1% /1Gi, 87.172Ki/s, 3h16m33s


gmaca@alpha MINGW64 ~/Dropbox/Cubbit_MIRROR
$
```

TODO

Backup files from gmpowerhorse (Ubuntu 20.04.6 LTS)

Prerequisites:

- Bucket already created from https://console.cubbit.eu/
  - Bucket name: `bk-gmpowerhorse`
  - Bucket versioning: Versioning disabled Object Lock: Object Lock disable
  - Ownership Control: Object writer
- Cubbit DS3 API key saved in a `.csv` file

Install rclone using apt

```
sudo apt install rclone
```

Check installed version

```
gmacario@gmpowerhorse:~ $ rclone --version
rclone v1.50.2
- os/arch: linux/amd64
- go version: go1.13.8
gmacario@gmpowerhorse:~ $
```

Type the following command to make sure you can access bucket `bk-gmpowerhorse`:

```
rclone ls cubbit:bk-gmpowerhorse
```

Result:

```
gmacario@gmpowerhorse:~ $ rclone ls cubbit:bk-gmpowerhorse
gmacario@gmpowerhorse:~ $
```

Now use rclone to synchronize contents of folder `~/Downloads` to bucket `bk-gmpowerhorse` on Cubbit DS3:

```
rclone sync -P ~/Downloads cubbit:bk-gmpowerhorse
```

Result:

```
gmacario@gmpowerhorse:~ $ rclone sync -P ~/Downloads cubbit:bk-gmpowerhorse
Transferred:      146.829M / 146.829 MBytes, 100%, 7.824 MBytes/s, ETA 0s
Errors:                 0
Checks:                 0 / 0, -
Transferred:           17 / 17, 100%
Elapsed time:       18.7s
gmacario@gmpowerhorse:~ $
```

Now check from `gmacario@hw2228` that all the files have been transferred:

```
gmacario@hw2228:~$ rclone ls cubbit:bk-gmpowerhorse
gmacario@hw2228:~$ rclone ls cubbit:bk-gmpowerhorse
 80609819 CLI_Linux_Debian_5.5.2.zip
 53237226 FingKit_CLI_Linux_Debian.zip
  1572864 bios-gmpowerhorse/BIOS_CD/7F5_0146.iso
  1048576 bios-gmpowerhorse/DOS_Flash/7F5_0146.bin
    27660 bios-gmpowerhorse/DOS_Flash/ASSIGNPW.EXE
     2841 bios-gmpowerhorse/DOS_Flash/DOSFM.txt
    54441 bios-gmpowerhorse/DOS_Flash/FLASHBIN.EXE
     3102 bios-gmpowerhorse/DOS_Flash/Flashbin.txt
     1003 bios-gmpowerhorse/DOS_Flash/README.TXT
     3388 bios-gmpowerhorse/DOS_Flash/flsh.cpu
     2957 bios-gmpowerhorse/README
    18746 bios-gmpowerhorse/hp-lxbios-1.5-1.i386.rpm
    13894 bios-gmpowerhorse/hp-lxbios-mod-1.5-1_2.6.9.67.ELsmp.src.rpm
    48836 bios-gmpowerhorse/lxbios_readme.pdf
      411 iottly-device-agent.service
 16079382 iottlyagent_1.6.4_linux_AMD64.tar.gz
  1235710 sp59252.tgz
gmacario@hw2228:~$
```

Command:

```
cd
rclone sync -P ~/Downloads cubbit:bk-gmpowerhorse
```

Result>

```
gmacario@gmpowerhorse:~ $ rclone sync -P . cubbit:bk-gmpowerhorse/home_gmacario
...
2023-11-09 23:34:43 ERROR : S3 bucket bk-gmpowerhorse path home_gmacario: not deleting␣
↪files as there were IO errors
2023-11-09 23:34:43 ERROR : S3 bucket bk-gmpowerhorse path home_gmacario: not deleting␣
↪directories as there were IO errors
2023-11-09 23:34:43 ERROR : Attempt 3/3 failed with 17 errors and: failed to open source␣
↪object: open /home/gmacario/github/gmacario/hassio-gmpowerhorse/config/.storage/core.
↪uuid: permission denied
Transferred:      414.161M / 414.161 MBytes, 100%, 81.090 kBytes/s, ETA 0s
Errors:                17 (retrying may help)
```

(continues on next page)

(continued from previous page)

```
Checks:             99165 / 99165, 100%
Transferred:        13571 / 13571, 100%
Elapsed time:   1h27m9.9s
2023/11/09 23:34:43 Failed to sync with 17 errors: last error was: failed to open source␣
↪object: open /home/gmacario/github/gmacario/hassio-gmpowerhorse/config/.storage/core.
↪uuid: permission denied
gmacario@gmpowerhorse:~ $
```

TODO

4.7 HOWTO Install DepthAI on a Raspberry Pi 4B for the ARNEIS project

4.7.1 Introduction

The following document explains how to install Luxonis DepthAI software on a Raspberry Pi for the ARNEIS project.

4.7.2 Prerequisites

- One Raspberry Pi prepared as detailed in "HOWTO Prepare a Raspberry Pi 4B for the ARNEIS project".
  - Tested on rpi4gm35 (Raspberry Pi 4B 4GB)
- One MicroSD card of at least 4GB.
  - **IMPORTANT**: The card should be blank, or at least should not contain any important data since it will be completely erased.
  - Tested with a SanDisk Ultra 256 GB MicroSDXC
- One desktop PC or laptop for formatting the SD card and controlling the RPi
  - OS: A recent version of Windows or Linux or macOS
  - The PC should have a MicroSDHC card reader. Alternatively, an additional USB MicroSD card reader is required
- Fast internet connection

4.7.3 Step-by-step instructions

Prepare the Raspberry Pi

Follow the instructions at https://arneis.readthedocs.io/en/latest/howto/howto-prepare-rpi4b-for-arneis.html.

Install DepthAI

Reference: https://docs.luxonis.com/en/latest/pages/tutorials/first_steps

Logged in as `pi@rpi4gm35`, type the following command to install DepthAI software:

```
sudo apt update
sudo apt install -y python3-pip python3-venv

mkdir -p ~/github/luxonis
cd ~/github/luxonis
git clone https://github.com/luxonis/depthai.git
```

Now create a Python virtualenv:

```
cd ~/github/luxonis/depthai
python3 -m venv virtualenv
source virtualenv/bin/activate
pip install -U pip
```

Install requirements:

```
python3 install_requirements.py
```

Install some missing binary packages (for some unknown reasons neither pip nor virtualenv automatically install them)

```
sudo apt-get -y install libatlas-base-dev python3-h5py
```

Add a new udev rule for the script to be able to access the OAK-D-Lite device correctly.

```
echo 'SUBSYSTEM=="usb", ATTRS{idVendor}=="03e7", MODE="0666"' \
    | sudo tee /etc/udev/rules.d/80-movidius.rules
sudo udevadm control --reload-rules \
    && sudo udevadm trigger
```

Check Linux kernel messages

```
sudo dmesg -w
```

Connect the OAK-D-Lite to one USB 3.0 port of the Raspberry Pi using a USB 3.0 cable (USB-A to USB-C).

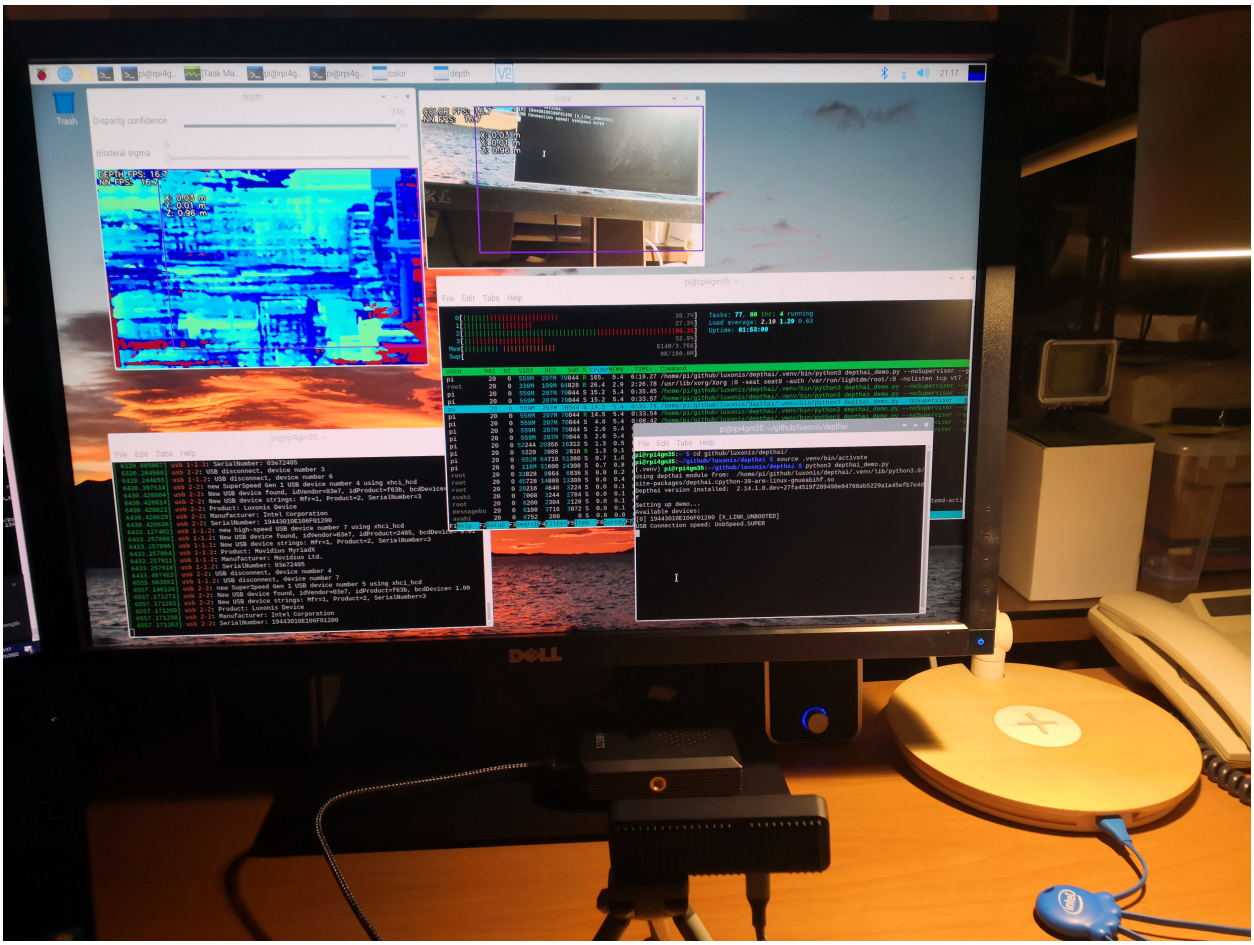
As soon as the OAK-D-Lite gets recognize the following messages should be displayed on the kernel log:

```
[ 5253.298901] usb 1-1.2: new high-speed USB device number 4 using xhci_hcd
[ 5253.429951] usb 1-1.2: New USB device found, idVendor=03e7, idProduct=2485,␣
↪bcdDevice= 0.01
[ 5253.429971] usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 5253.429987] usb 1-1.2: Product: Movidius MyriadX
[ 5253.430002] usb 1-1.2: Manufacturer: Movidius Ltd.
[ 5253.430017] usb 1-1.2: SerialNumber: 03e72485
```

Run the demo script

```
python3 depthai_demo.py
```

Result: The `depthai_demo.py` program is executed correctly on the Raspberry Pi.



Note 1: First execution of depthai_demo.py

The first time that `depthai_demo.py` is launched, it may take a few minutes before the camera windows are be displayed.

Note 2: Launching depthai_demo.py from a remote shell

The `depthai_demo.py` script should be invoked from a terminal on the main Raspberry Pi display.

If the command is invoked from a remote SSH shell, make sure that the `DISPLAY` environment variable is properly set before launching the demo script.

Example:

```
(.venv) pi@rpird102:~/github/luxonis/depthai $ export DISPLAY=:0.0
(.venv) pi@rpird102:~/github/luxonis/depthai $ python3 depthai_demo.py
Using depthai module from:  /home/pi/github/luxonis/depthai/.venv/lib/python3.7/site-
↪packages/depthai.cpython-37m-arm-linux-gnueabihf.so
Depthai version installed:  2.14.1.0.dev+27fa4519f289498e84768ab5229a1a45efb7e4df
Setting up demo...
Available devices:
```

(continues on next page)

(continued from previous page)

```
[0] 19443010E106F01200 [X_LINK_UNBOOTED]
USB Connection speed: UsbSpeed.SUPER
```

4.8 HOWTO Install DepthAI Demo v3.0.9 on Windows 10

4.8.1 Introduction

The following document explains how to prepare and configure a Windows laptop to be used with the ARNEIS project.

4.8.2 References

- https://docs.luxonis.com/en/latest/pages/tutorials/first_steps/#use-windows-installer

4.8.3 Prerequisites

- One OAK-D-Lite
- One desktop PC or laptop for controlling the OAK-D-Lite
  - OS: A recent version of Windows or Linux or macOS
  - Tested on alpha (HW: Dell Precision M4600, OS: Windows 10 version 21H2)
- Fast internet connection

4.8.4 Step-by-step instructions

Install DepthAI

Download `DepthAI-setup-v3.0.9.exe` from https://github.com/luxonis/depthai/releases

Double click `DepthAI-setup-v3.0.9.exe` to launch the install wizard

> **Windows protected your PC**
>
> Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.
>
> More info
>
> Don't run

Click "More info".

> **Windows protected your PC**
>
> Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.
>
> - App: DepthAI-setup-v3.0.9.exe
> - Autore: Luxonis Holding Corporation
>
> Run anyway | Don't run

Click "Run anyway".

Select Destination Location

Where should DepthAI be installed?

Setup will install DeptyAI into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

`C:\Users\gmaca\AppData\Local\Programs\DepthAI`

At least 364,5 MB of free disk space is required.

Next | Cancel

Click "Next".

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing DepthAI, then click Next.

Additional shortcuts:

- [x] Create a desktop shortcut

Back | Next | Cancel

Accept proposed values, then click "Next".

Ready to Install

Setup is now ready to begin installing DepthAI on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Users\gmaca\AppData\Local\Programs\DepthAI

Additional tasks:
    Additional shortcuts:
        Create a desktop shortcut
```

Back | Install | Cancel

Click "Install".

Installing

Please wait while Setup installs DepthAI on your computer.

...

Completing the DepthAI Setup Wizard

Setup has finished installing DepthAI on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

- [x] Launch DepthAI

Click "Finish".

Installing Depthai Requirements

If no OAK-D-Lite is connected a dialog box like this one should be displayed:

An error occurred

```
File "C:\Users\gmaca\AppData\Local\ProgramsDepthAI\depthai\depthai_demo.py",␣
↪line 625, in run
    self.instance.run_all(self.conf)
    ...
    No DepthAI device found!
```

OK

Click "OK" to terminate the program.

Connect the OAK-D-Lite

Now connect your OAK-D-Lite to your laptop using a proper USB 3.0 USB-A to USB-C connector, then double click on the "DepthAI" icon on your desktop.

Result: the DepthAI demo main window is displayed with the augmented video captured by the OAK-D-Lite.

4.9 HOWTO Train a NN for Object Detection and create a deployable model for the OAK-D-Lite

4.9.1 Introduction

To be able to train a model to recognize our objects, I have used the Easy_Object_Detection_With_Custom_Data_Demo_Training.ipynb Jupyter notebook published on https://github.com/luxonis/depthai-ml-training.

The training process has been performed inside a Virtual Machine running Ubuntu version 21.10 while the conversion of the trained model through OpenVINO™ has been made inside another VM running Ubuntu version 20.04 LTS. I would suggest to do the whole process on the latter.

4.9.2 Environment deploying

Since not all the used Python libraries are added to the notebook commands, I have changed my notebook with those. I have tried to compile each addition to add in this document as bash script (I prefer to use anaconda):

```
echo -e "\n\nDownloading Anaconda installer. This may take sometime."
curl -O https://repo.anaconda.com/archive/Anaconda3-2021.11-Linux-x86_64.sh
echo -e "\n\n Download finished. Starting installation!"
bash Anaconda3-2021.11-Linux-x86_64.sh -b -p /home/vagrant/anaconda3
export PATH="~/anaconda3/bin:$PATH"
sudo chown -R $USER:$USER ~/anaconda3
conda init bash
echo -e "\n\n Creating environment and installing major libraries\n"
conda create --name trainenv python=3.6
echo -e "\n\n Activating trainenv environment\n"
conda activate trainenv
pip install numpy==1.17.5
pip install tensorflow==1.15.0
pip install jupyter ipykernel scipy
```

(continues on next page)

(continued from previous page)

```
pip install tf_slim Cython contextlib2 pillow lxml matplotlib pycocotools gdown
pip install google-api-python-client
sudo apt-get update && sudo apt-get install -y -qq protobuf-compiler python-pil python-
↪lxml python-tk
sudo apt-get install -y pciutils cpio
echo -e "\n\nInstalling VSCODE\n"
sudo snap install --classic code
echo -e "\n\nFINISHED\n"
```

You should have your images stored somewhere and for each you should have an associated *.xml* file.

Then follow the code put inside the notebook and you are good to go!

4.9.3 Running the notebook

I've usually used the notebook by just changing the path displayed or adding some minor fix. Create a Python folder (maybe on Desktop), if you want you can create a *depthai-ml-training/colab-notebook/* folder there but it's not required.

I've made just a little change to the **content** folder location. As I changed the path used in the rows of code, I've made sure that the *content folder* was placed inside *Python* so you should have something like:

```
Python
└──depthait-ml-training
│
└──content
```

This is not required to make it works (I believe) but as I wanted to know where every file was downloaded, I kept everything together so it's more easy to manage or delete if needed.

The notebook downloads library, folders with photos, programs, … For several Gb, so be advised that you may need a flat internet! **Don't do it on mobile network**.

You may encounter some trouble running this code

```
apt-get update && apt-get install -y -qq protobuf-compiler python-pil python-lxml python-
↪tk
!pip install -q Cython contextlib2 pillow lxml matplotlib
!pip install -q pycocotools
%cd /content/models/research
!protoc object_detection/protos/*.proto --python_out=.
import os
os.environ['PYTHONPATH'] += ':/content/models/research/:/content/models/research/slim/'
```

Make sure the apt installation succeed and that the pip library install do as well.

The correct result of that cell should look like this with a green check under:

```
/content/models/research
object_detection/protos/input_reader.proto: warning: Import object_detection/protos/
↪image_resizer.proto but not used.
```

The following cell generate the **label_map.pbtxt** as well as **test, train records**. The first file will be required to run the model with our code. It contains the names of the "**ITEMS**" on which the NN will be trained for.

You may encounter some problems running

```
import re
iou_threshold = 0.50
num_classes = get_num_classes(label_map_pbtxt_fname)
with open(pipeline_fname) as f:
    s = f.read()
with open(pipeline_fname, 'w') as f:

    # fine_tune_checkpoint
    s = re.sub('fine_tune_checkpoint: ".*?"',
               'fine_tune_checkpoint: "{}"'.format(fine_tune_checkpoint), s)

    # tfrecord files train and test.
    s = re.sub(
        '(input_path: ".*?)(train.record)(.*?")', 'input_path: "{}"'.format(train_record_
↪fname), s)
    s = re.sub(
        '(input_path: ".*?)(val.record)(.*?")', 'input_path: "{}"'.format(test_record_
↪fname), s)

    # label_map_path
    s = re.sub(
        'label_map_path: ".*?"', 'label_map_path: "{}"'.format(label_map_pbtxt_fname), s)

    # Set training batch_size.
    s = re.sub('batch_size: [0-9]+',
               'batch_size: {}'.format(batch_size), s)

    # Set training steps, num_steps
    s = re.sub('num_steps: [0-9]+',
               'num_steps: {}'.format(num_steps), s)

    # Set number of classes num_classes.
    s = re.sub('num_classes: [0-9]+',
               'num_classes: {}'.format(num_classes), s)
    # Set number of classes num_classes.
    s = re.sub('iou_threshold: [0-9].[0-9]+',
               'iou_threshold: {}'.format(iou_threshold), s)

    f.write(s)
```

The error says something like *'fine_tune_checkpoint: ".?"'** is not a variable, so instead of processing that with regular expression, it search for a variable with that name. Which I don't know how to fix yet, aside from changing VM or PC…

I didn't use the part of code with Tensorboard. I didn't care to try it.

I've added the `pip install flask-ngrok-` after that because it was missing and it's a required library.

This is the important code for training the NN:

```
model_dir = 'training/'
# Optionally remove content in output model directory for a fresh start.
# !rm -rf {model_dir}
# os.makedirs(model_dir, exist_ok=True)
```

(continues on next page)

(continued from previous page)

```
!python /content/models/research/object_detection/model_main.py \
    --pipeline_config_path={pipeline_fname} \
    --model_dir={model_dir} \
    --alsologtostderr \
    --num_train_steps={num_steps} \
    --num_eval_steps={num_eval_steps}
```

I've usually modified this to put the *training* folder inside the content (content/training). The original notebook puts this folder instead inside of /content/models/research/training. It is **IMPORTANT** that this code doesn't compute with low minutes of execution. On my ubuntuvm with 1500 steps it required like 24 hours with the test and validation images provided. As before if it doesn't run properly, I don't know how to fix yet, aside from changing VM or PC…

If it works properly you should have inside your training folder something like:

```
checkpoint
eval_=
events.out.tfevents.1643438685.mio-VirtualBox
export
graph.pbtxt
model.ckpt-1446.data-00000-of-00001
model.ckpt-1446.index
model.ckpt-1446.meta
model.ckpt-1461.data-00000-of-00001
model.ckpt-1461.index
model.ckpt-1461.meta
model.ckpt-1476.data-00000-of-00001
model.ckpt-1476.index
model.ckpt-1476.meta
model.ckpt-1491.data-00000-of-00001
model.ckpt-1491.index
model.ckpt-1491.meta
model.ckpt-1500.data-00000-of-00001
model.ckpt-1500.index
model.ckpt-1500.meta
```

Those are 5 of the latest model trained and autotested. Since it has 5 checkpoints, those are the latest 5 models and the number near it should have at least 1 number equals to the number of steps you have defined at the start, in my case 1500.

After this part there is the code that handles the generation of the **frozen_inference_graph.pb** which is a frozen graph that cannot be trained anymore, it defines the graphdef and is actually a serialized graph and can be loaded. The saved model is a model generated by tf.saved_model.builder and is has to be imported into a session, this file contains the full graph with all training weights (just like the frozen graph) but here can be trained upon, and this one is not serialized and needs to be loaded. (link for more infos https://stackoverflow.com/questions/52934795/what-is-difference-frozen-inference-graph-pb-and-saved-model-pb)

In the notebook after the generation of the frozen graph there is a testing with some images, put inside the validation folder.

Now we finally get to the part where we convert the trained NN model to an *object* used by our OAK-D-LITE camera.

As stated in the notebook I've used this version of OpenVINO

```
url = "https://registrationcenter-download.intel.com/akdlm/irc_nas/17662/l_openvino_
↪toolkit_p_2021.3.394.tgz"
```

(continues on next page)

(continued from previous page)

```
!wget {url}

## Get the name of the tgz
parsed = urlparse(url)
openvino_tgz = os.path.basename(parsed.path)
openvino_folder = os.path.splitext(openvino_tgz)[0]

## Extract & install openvino
!tar xf {openvino_tgz}
%cd {openvino_folder}
!./install_openvino_dependencies.sh && \
    sed -i 's/decline/accept/g' silent.cfg && \
    ./install.sh --silent silent.cfg
```

I've had some issue with the installation command which I overcome by copying the following line and removing the spaces and running it inside the shell directly

```
!./install_openvino_dependencies.sh && sed -i 's/decline/accept/g' silent.cfg && ./
↪install.sh --silent silent.cfg
```

As well as running

```
!source /opt/intel/openvino/bin/setupvars.sh && \
     /opt/intel/openvino/deployment_tools/demo/demo_squeezenet_download_convert_run.sh
```

Which in some cases even required to be **sudo**ers.

If you get errors running the following part

```
%cd /opt/intel/openvino_2021/deployment_tools/model_optimizer/extensions/front/tf/

#openvino fixes: edit
# Read in the file, make sure the .json corresponds to the model!!!
with open('ssd_v2_support.json', 'r') as file :
  filedata = file.read()
```

Run a **chmod 777** command on that file:

```
sudo chmod 777 ~/opt/intel/openvino_2021/deployment_tools/model_optimizer/front/tf/ssd_
↪v2_support.json
```

The next part is what creates the model to be deployable (it's not the final product yet..)

```
#CONVERT TF MODEL to OPEN VINO IRv10. saved in IR_V10_fruits_mnssdv2_6k directory or
#choose own name for --output_dir "choose name"
%cd "/content/models/research/fine_tuned_model/"
!source /opt/intel/openvino_2021/bin/setupvars.sh && \
    python /opt/intel/openvino_2021/deployment_tools/model_optimizer/mo.py \
    --input_model frozen_inference_graph.pb \
    --tensorflow_use_custom_operations_config /opt/intel/openvino_2021/deployment_tools/
↪model_optimizer/extensions/front/tf/ssd_v2_support.json \
    --tensorflow_object_detection_api_pipeline_config pipeline.config \
```

(continues on next page)

(continued from previous page)

```
    --reverse_input_channels \
    --output_dir ./pascal_animals \
    --data_type FP16
```

I've had some issue with it but I fixed them by adding the FULL path for each line. As before I put the **pascal_animals** inside the content too: /content/pascal_animals/.

If you have managed to get to here, you can just download those files and use the http://blobconverter.luxonis.com/ to convert them through blob to be able to run your NN as a blob file on your camera. Just remember to use shaves like 5 or 6.

4.10 HOWTO Control a LEGO Powered Up Hub for the ARNEIS Project

4.10.1 Introduction

This document illustrates the available options to control a LEGO® Powered Up Hub (LEGO Item No bb0961c01).

The LEGO Powered Up Hub is an embedded device based on a small microcontroller which can control up to four LEGO Powered Up devices, such as motors, lights and sensors. In addition to a button and a multicolor programmable LED, the Hub is provided with a Bluetooth Low Energy interface which allows the Hub to act as a BLE Peripheral device and to communicate with a BLE Central device such as a mobile phone, a laptop or a Raspberry Pi.

The LEGO Group has released a few official apps for the iOS™ and Android™ operating systems. Those apps can be downloaded and installed from their respective app stores (App Store for iOS devices, Google Play for Android devices) and provide a seamless user experience for controlling official LEGO sets and MOCs.

Additionally, a few open source projects such as Pybricks have been developed to simplify programming the LEGO Powered Up Hubs.

Each of the following chapters will analyze in more details the possible options which may be used to program and control the Hubs.

4.10.2 Using the "LEGO® Powered Up" App

This app may be used to control official LEGO sets based on Powered Up hubs using a mobile device such as a phone or a tablet.

Tutorials

LEGO Powered Up basic tutorials: https://www.youtube.com/watch?v=MIpcyS4xzsw

Installing the app

Install the "LEGO® POWERED UP" app from

- Google Play Store: https://play.google.com/store/apps/details?id=com.lego.common.poweredup
- Apple App Store

Predefined controller modes

After launching the app, you can choose an official set to play with.

Create your own program

Beyond using the app for controlling the official LEGO sets, the "LEGO® Powered Up" app has also a "Create" mode which can be used to develop a custom HMI to control original constructions and MOCs.

Create mode

After selecting the "Create" mode, you must provide a name for your project, then select the aspect you want to configure:

- Controller: Place buttons and widgets on the controller
- Coding: Place the elements to control the flow of your program

After tapping "Controller" you will be asked to choose a theme, then you will be able to place buttons and other widgets of your virtual controller.

In the "Coding" view you may stack the elements to create the correct execution flow.

4.10.3 Using the "LEGO® Technic® Control+" App

This app provides an out-of-the box experience for official LEGO sets, most notably the LEGO Set 42100 (Leibherr R 9800 Excavator), which include LEGO Powered Up Bluetooth Hubs.

NOTE: This app is very intuitive and user-friendly, although its goal is only to act as a remote control for official LEGO designs. For much higher flexibility we recommend other options, such as the "LEGO® Powered Up" App described in the previous chapter.

Install the "LEGO® TECHNIC® CONTROL+" app from

- Google Play Store: https://play.google.com/store/apps/details?id=com.lego.technic.controlplus
- Apple App Store: `https://apps.apple.com/us/app/lego-technic-control/id1465808291`

Launch the app and select set "Liebherr R 9800".

Connect 2x XL motor and 1x L motor to the Hub.

Follow the instructions displayed by the app to make sure that all the motors are wired to the correct port of the Hub.

The app will also check the version of the firmware on the Hub, and propose the update the firmware if needed.

Repeat the steps for the second hub in the set.

After all the checks are successful, the app is ready and can be used to control the excavator and have hours of entertainment with the LEGO set.

4.10.4 Using Pybricks

Pybricks is Python coding for smart LEGO® hubs. Run MicroPython scripts directly on the hub, and get full control of your motors and sensors.

Pybricks is used by students, teachers, and hobbyists all around the world. It's free, open source, and supported by a community of robotics experts.

Install the Pybricks firmware

Instructions at https://pybricks.com/install/technic-boost-city/

Before you begin, turn the hub off. The update works best with fresh batteries. If you use the City Hub, you must unplug all motors and sensors. Follow these steps:

1. Go to Pybricks Code.
2. Press and hold the hub button. Wait for a blinking pink light.
3. While you hold it, click the *firmware update* button.
4. Select the *LEGO Bootloader* and click *Pair*.
5. Wait until the light turns off, and then blinks red/green/blue.
6. Release the button and wait for the installation to finish.

Load some Python program

From the Pybricks Code web interface, create or open a MicroPython source file and upload it to the LEGO Technic BT Hub. You may find some examples under the `/code` folder of the ARNEIS repository on GitHub.

4.10.5 Controlling from a Raspberry Pi

Install undera/pylgbst on the Raspberry Pi

Logged in as pi@rpi4gm35

```
mkdir -p ~/github/undera
cd ~/github/undera
git clone https://github.com/undera/pylgbst
```

Create a Python virtual environment

```
cd ~/github/undera/pylgbst
python3 -m venv .venv
source .venv/bin/activate
```

TODO: Install prerequisites:

```
sudo apt -y install libbluetooth-dev
sudo apt -y install libboost-python-dev
sudo apt -y install libboost-thread-dev
sudo apt -y install libglib2.0-dev
# pip install -U boost
```

(continues on next page)

(continued from previous page)

```
# pip install -U gattlib
pip install -U bluepy
pip install -U pylgbst
```

Launch pylgbst demo:

```
cd ~/github/undera/pylgbst
source .venv/bin/activate
cd examples
python3 demo.py
```

Result:

```
(.venv) pi@rpird102:~/github/undera/pylgbst/examples $ python3 demo.py
42      INFO    root    Trying get_connection_bluepy
74      INFO    comms-bluepy    Discovering devices...
88      INFO    root    Trying get_connection_bluegiga
90      INFO    root    Trying get_connection_gatt
92      INFO    root    Trying get_connection_bleak
148     INFO    root    Trying get_connection_gattool
150     INFO    root    Trying get_connection_gattlib
165     INFO    comms-gattlib   Discovering devices using hci0...
Traceback (most recent call last):
  File "demo.py", line 259, in <module>
    hub = MoveHub(**parameters)
  File "/home/pi/github/undera/pylgbst/.venv/lib/python3.7/site-packages/pylgbst/hub.py",
↪ line 231, in __init__
    connection = get_connection_auto(hub_name=self.DEFAULT_NAME)
  File "/home/pi/github/undera/pylgbst/.venv/lib/python3.7/site-packages/pylgbst/__init__
↪.py", line 75, in get_connection_auto
    raise Exception("Failed to autodetect connection, make sure you have installed␣
↪prerequisites")
Exception: Failed to autodetect connection, make sure you have installed prerequisites
Exception ignored in: <function Hub.__del__ at 0xb64d15d0>
Traceback (most recent call last):
  File "/home/pi/github/undera/pylgbst/.venv/lib/python3.7/site-packages/pylgbst/hub.py",
↪ line 76, in __del__
    if self.connection and self.connection.is_alive():
AttributeError: 'MoveHub' object has no attribute 'connection'
(.venv) pi@rpird102:~/github/undera/pylgbst/examples $
```

Troubleshooting

```
(.venv) pi@rpird102:~/github/undera/pylgbst/examples $ hciconfig
hci0:   Type: Primary  Bus: UART
        BD Address: E4:5F:01:35:8F:98  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING
        RX bytes:1611 acl:0 sco:0 events:103 errors:0
        TX bytes:3575 acl:0 sco:0 commands:103 errors:0

(.venv) pi@rpird102:~/github/undera/pylgbst/examples $
```

4.10.6 Interfacing an input sensor

This chapter discusses several alternatives we have in case we need to interface one or more input sensors to the Edge Controller of ARNEIS. The alternatives presented here are the following:

- Direct GPIO input pin of the Raspberry Pi
- Input sensors connected to the LEGO Powered Up Hub
- Input sensors connected to the SBrick Plus
- etc.

Direct GPIO input pin of the Raspberry Pi

This is probably the most flexible option in terms of sensor interfacing.

In the internet there is abundance of tutorials, videos and blog posts which explain how to interface an input sensor to the GPIO pins of the Raspberry Pi and write a program to check the sensor state. Google is your friend.

Some quick links:

- #369 Definitive Guide to Attaching Sensors to the Raspberry Pi (Tutorial) - YouTube video by Andreas Speiss, 2021-01-24

On the other hand, this option has the disadvantage that the feedback loop sensor/actuator will be slower since a longer path (Sensor –> Raspberry Pi –> BLE –> Technic Hub –> Actuator) should be followed.

Input sensors connected to the LEGO® Powered Up Hub

At the moment only few types of input sensors with Powered Up interface are available:

- LEGO Powered Up Color & Distance Sensor - Code 88007
- LEGO WeDo Tilt Sensor - Code 63522
- Infrared Sensor - Code TODO

Those sensors are directly usable with LEGO® Technic® BT Hub using different languages, including Pybricks:

- https://docs.pybricks.com/en/stable/pupdevices/colordistancesensor.html
- https://docs.pybricks.com/en/stable/pupdevices/tiltsensor.html
- https://docs.pybricks.com/en/stable/pupdevices/infraredsensor.html

As an added bonus, the ColorDistanceSensor can send infrared signals to control Power Functions infrared receivers. You can use this technique to control medium, large, extra large, and train motors. You can find more details at https://docs.pybricks.com/en/stable/pupdevices/pfmotor.html.

Provided that the available sensors are good enough for the use case, this option is probably the one which guarantees the quickest feedback loop (Sensor –> Technic Hub –> Actuator)

Input sensors connected to the SBrick Plus

Another possible option is to connect and interface LEGO® Power Functions sensors.

The LEGO® Power Functions family provides a richer set of input sensors with respect to the relatively newer Powered Up family:

- TODO

To interface Power Functions components including input sensors and control them from either a mobile app or a Python scripts running on a Raspberry Pi the SBrick Plus brick can be used.

Additionally, we may use the RXTX mode of the Powered Up Color Sensor which allows to communicate with a Power Functions infrared receiver and thus control motors of the Power Functions family. Unfortunately this solution only works for actuators, not for sensors.

You may find more details at the following links:

- Using Power Functions with the Powered Up app - LEGO® Customer Service
- LEGO Powered Up app with Power Functions and remote support! Too good to be true? - RacingBrick, 2020-03-21

4.11 HOWTO Install Raspberry Pi OS 64-bit for the ARNEIS project

4.11.1 Introduction

This HOWTO documents how to install Raspberry Pi OS 64-bit on a Raspberry Pi 3B+ or higher to be used as an Edge Controller of the ARNEIS project.

4.11.2 Reference documents

- No PC Needed: How to Install Raspberry Pi OS Over the internet - Tom's Hardware, 2022-02-12

4.11.3 Prerequisites

- One Raspberry Pi.
  - Tested on rpi4gm35 (Raspberry Pi 4B 4GB)
- One MicroSD card of at least 4GB.
  - **IMPORTANT**: The card should be blank, or at least should not contain any important data since it will be completely erased.
  - Tested with a SanDisk Ultra 256 GB MicroSDXC
- One desktop PC or laptop for formatting the SD card and controlling the RPi
  - OS: A recent version of Windows or Linux or macOS
  - The PC should have a MicroSDHC card reader. Alternatively, an additional USB MicroSD card reader is required
- Fast internet connection

4.11.4 Install base Operating System

The installation of Raspbian Pi OS 64-bit is performed through the following tests:

1. Update the Bootloader of the Raspberry Pi
2. Prepare a MicroSD card with Raspberry Pi OS
3. TODO (See below)

Each of thosee steps above is described in more details in the corresponding chapter down below.

Install Raspberry Pi Imager

Launch a browser on your laptop and open https://www.raspberrypi.com/software/

Download Raspberry Pi Imager and install it on your laptop.

Make sure you are running the latest version (1.7.2 at the time of this writing).

Update the Bootloader of the Raspberry Pi

Launch Raspberry Pi imager.

Insert a MicroSD card into one slot of your laptop. Alternatively, insert the MicroSD into the USB card reader, then plug the USB card reader into one empty USB port of your laptop.

Choose Operating System: Misc utility images > Beta Test Bootloader > SD Card Boot

- Boot from SD Card if available, otherwise boot from USB
- Released: 2022-02-28
- Online - 0.0 GB download

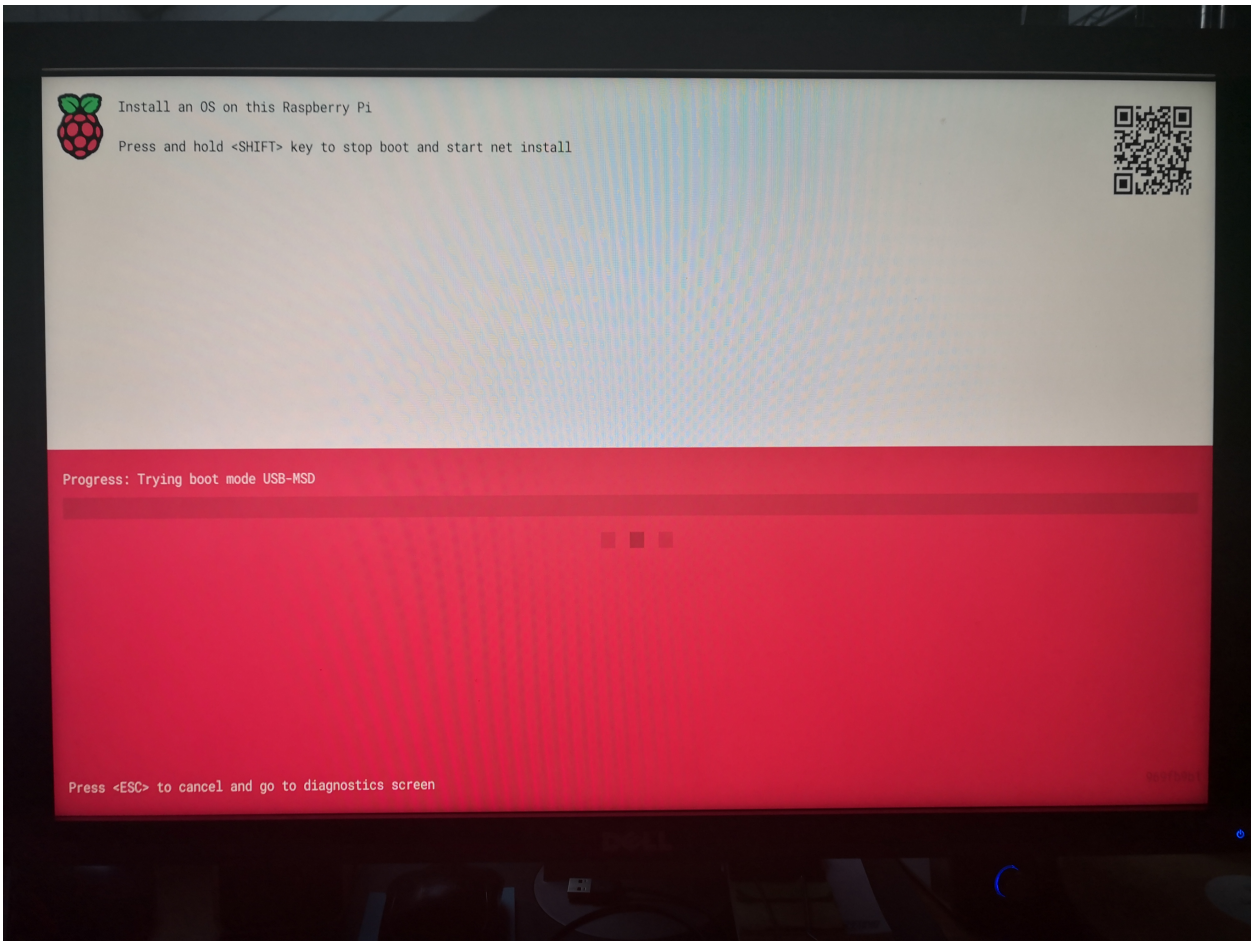
Choose Storage: (select MicroSDHC)

Click "Write" to write the MicroSD card with the selected image.

- Insert the MicroSD card into your Raspberry Pi.
- Connect a display using a MicroHDMI-to-HDMI cable
- Connect a USB keyboard to the first USB 2.0 port of the RPi4
- Connect a USB mouse to the second USB 2.0 port of the RPi4
- Connect a 5Vdc, 3A power supply to the USB-C port of the RPi4

Wait until the Bootloader is updated, then remove power from the Raspberry Pi.

Remove the MicroSDHC and power up the Raspberry Pi. The updated bootloader should be displayed.



Prepare a MicroSD card with Raspberry Pi OS

Make sure the Ethernet cable is connected.

Press and hold the SHIFT key to stop boot and start net install. The Raspberry Pi imager will be executed on the Raspberry Pi.

NOTE: If you are using an older version of Raspberry Pi which does not support the updated bootloader, you may always prepare the MicroSD card running Raspberry Pi imager on the laptop and following similar instructions to what described in this section.

Insert a MicroSD card into the Raspberry Pi.

Choose Language and Keyboard.

Choose Operating System: Raspberry Pi OS (other) > Raspberry Pi OS (64-bit)

- A port of Debian Bullseye with the Raspberry Pi Desktop (Compatible with Raspberry Pi 3/4/400)
- Released: 2022-04-04
- Online - 0.7 GB download

Choose Storage: Internal SD card reader - 31.3 GB

Click on the "cog" icon to open the Advanced options dialog box

Image customization options: for this session only

- [x] Set hostname: `rpird102`.local
- [x] Enable SSH
  - Use password authentication: Yes
  - Allow public-key authentication only
- [x] Set username and password

    - Username: `pi`
    - Password: `xxxx`
  - [x] Configure wireless LAN
    - SSID: `xxxx`
    - Hidden SSID: No
    - Password: `yyyy`
    - Wireless LAN country: IT
  - [x] Set locale settings
    - Time zone: Europe/Rome
    - Keyboard layout: it

  Persistent settings

  - [ ] Play sound when finished
  - [x] Eject media when finished
  - [x] Enable telemetry

then click "SAVE".

Click "WRITE" to download the image and write it to the MicroSDHC.

> **Warning**
>
> All existing data on 'Internal SD card reader' will be erased. Are you sure you want to continue?
>
> NO | SI

Click "YES".

…

After a few minutes the MicroSD should be ready for the first boot.

4.11.5 First boot of the RPi with the new MicroSD card

- Insert the MicroSD card into your Raspberry Pi.
- Connect a display using a MicroHDMI-to-HDMI cable
- Connect a USB keyboard to the first USB 2.0 port of the RPi4
- Connect a USB mouse to the second USB 2.0 port of the RPi4
- Connect a 5Vdc, 3A power supply to the USB-C port of the RPi4

Turn on the power supply and wait for Raspberry Pi OS to boot.

> **Welcome to Raspberry Pi**
>
> Welcome to the Raspberry Pi Desktop!
>
> Before you start using it, there are a few things to set up.
>
> Press 'Next' to get started.
>
> Cancel | Next

Click "Next".

Set Country

Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.

- Country: Italy
- Language: Italian
- Timezone: Rome
- [x] Use English language
- [ ] Use US keyboard

Press 'Next' when you have made your selection.

Back | Next

Fill in the form as shown above, then click "Next"

…

Change Password

The default 'pi' user account currently has the password 'raspberry'. It is strongly recommended that you change this to a different password that only you know

- Enter new password: xxxx
- Confirm new password: xxxx
- [x] Hide characters

Press 'Next' to activate your new password.

Back | Next

Fill in the form as instructed, then click "Next"

Set Up Screen

You should be able to set the taskbar along the top of the screen. Tick the box if some or all of it does not fit on the screen.

[ ] The taskbar does not fit one the screen

The change will take effect when the Pi is restarted.

Press 'Next' to save your setting.

Back | Next

Verify and update if needed, then click "Next".

Select WiFi Network

Select your WiFi network from the list.

…

Press 'Next' to connect, or 'Skip' to continue without connecting.

Back | Skip | Next

Select the desired WiFi network, then click "Next". If the select network is protected you will be requested the password

Enter WiFi Network

Enter the password for the WiFi network 'xxxx'.

- Password: yyyy

[x] Hide characters

Press 'Next' to connect, or 'Skip' to continue without connecting.

Back | Skip | Next

Click "Next".

Update Software

The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Next' …

Back | Skip | Next

Click "Next".

Download updates - please wait.

When the update is complete the following popup should be displayed

System is up to date

OK

Click "OK" to continue.

Setup Complete

Your Raspberry Pi is now set up and ready to go.

To run applications, click the raspberry icon in the top left corner of the screen to open the menu.

Press 'Restart' to restart your Pi so the new settings will take effect.

Back | Later | Restart

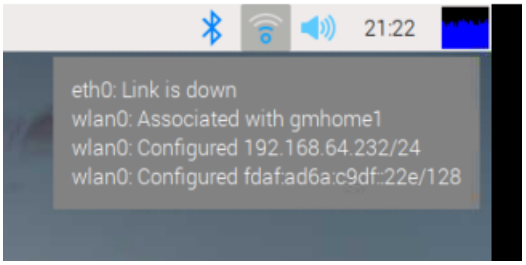
Click "Restart".

Verify that the RPi reboots correctly.

Trash
SiComputer

Display assigned IP addresses

To know the IP addresses assigned to the Raspberry Pi just move the mouse over the network icon at the top right of the desktop



Configure hostname, SSH and VNC

Open a terminal and type the following command

```
sudo raspi-config
```

- Select "1" (System Options), then "S4" (Hostname)
- Enter hostname: `rpi4gm35` (will replace default hostname `raspberrypi`)
- Select "3" (Interface Options), then "I2" (SSH)
- Select "Yes" to enable SSH
- Select "3" (Interface Options), then "I3" (VNC)
- Select "Yes" to enable VNC
- Select "Finish" to exit `raspi-config`. Reboot if requested

Verify that the RPi is accessible from the laptop via SSH and VNC (you may need to scan the local Wi-Fi network to get the IPv4 address assigned by the router)

HINT: To scan the network and identify the open services you can use one of those tools

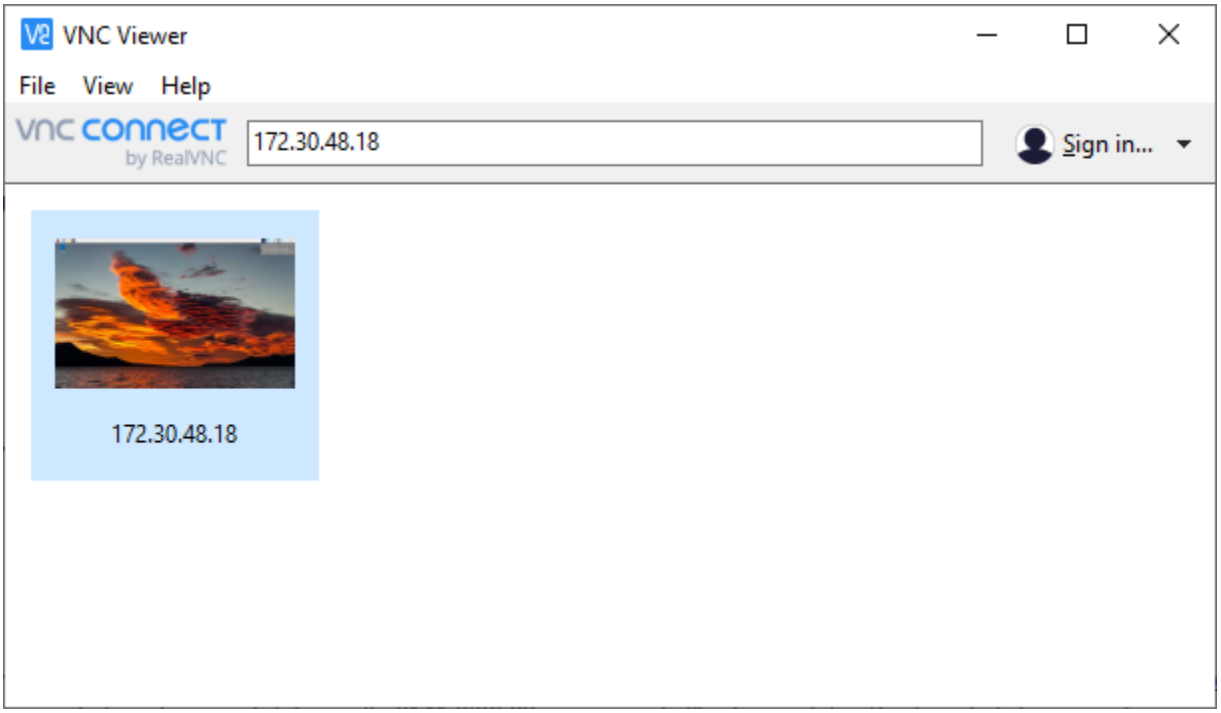
- The Fing app on a mobile phone or on your laptop.
- nmap

It this works disconnect the USB keyboard, mouse and display

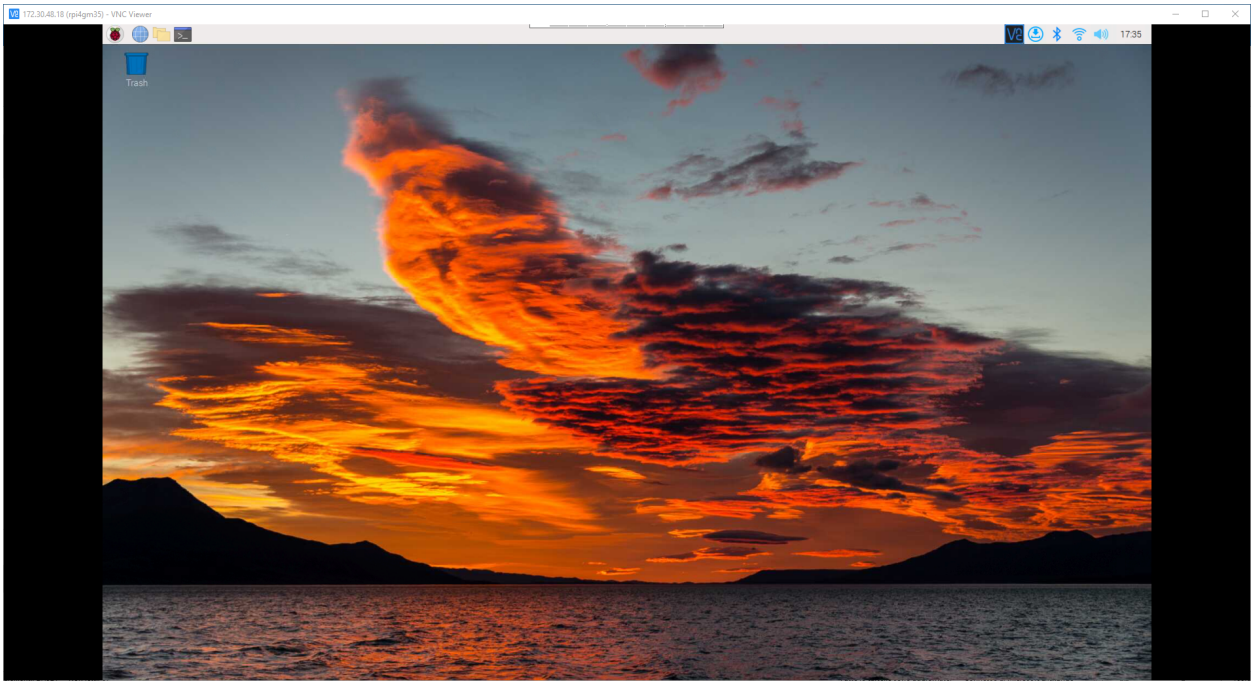
Reboot your RPi4 and verify that the device is still accessible from SSH:

```
gmacario@alpha:~$ ssh pi@rpird102.local
pi@rpird102.local's password:
Linux rpird102 5.15.32-v8+ #1538 SMP PREEMPT Thu Mar 31 19:40:39 BST 2022 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May  1 21:28:04 2022
pi@rpird102:~ $
```

Do the same using a VNC client (in my case I used the free to use VNC® Viewer)



Double click on the selected profile to connect to the remote desktop of the Raspberry Pi:



(Recommended) Create a public DNS entry

If you have administrative rights to a DNS zone you may choose to access your Raspberry Pi using a symbolic name rather than an IP address.

If so, access your DNS administrative page (in my case, https://register.it/) and create an A record to map the name to the IP address assigned to your Raspberry Pi.

In my case

```
A rpi4gm35 172.30.48.18
```

Wait until the DNS zone is propagated, then verify that the device can be accessed by another host (in our case, our laptop) using the assigned name rather than its IP address:

```
gpmacario@HW2457 MINGW64 ~
$ ping rpi4gm35.gmacario.it

Esecuzione di Ping rpi4gm35.gmacario.it [172.30.48.18] con 32 byte di dati:
Risposta da 172.30.48.18: byte=32 durata=8ms TTL=64
Risposta da 172.30.48.18: byte=32 durata=7ms TTL=64
Risposta da 172.30.48.18: byte=32 durata=6ms TTL=64
Risposta da 172.30.48.18: byte=32 durata=6ms TTL=64

Statistiche Ping per 172.30.48.18:
    Pacchetti: Trasmessi = 4, Ricevuti = 4,
    Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 6ms, Massimo =  8ms, Medio =  6ms

gpmacario@HW2457 MINGW64 ~
$
```

Configure public SSH keypair

Logged in as pi@rpird102, create a public/private SSH keypair:

```
ssh-keygen
```

Type the following commands to be able to login to your Raspberry Pi through your public SSH key - for instance:

```
cat <<END >>~/.ssh/authorized_keys
ssh-rsa AAAAB3Nza....W1cG35r8= gpmacario@HW2457
END
```

Test

```
gpmacario@HW2457 MINGW64 ~
$ ssh pi@rpi4gm35.gmacario.it
Linux rpi4gm35 5.10.92-v7l+ #1514 SMP Mon Jan 17 17:38:03 GMT 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

(continues on next page)

(continued from previous page)

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan 20 09:12:23 2022
pi@rpi4gm35:~ $
```

Install Virtual Keyboard

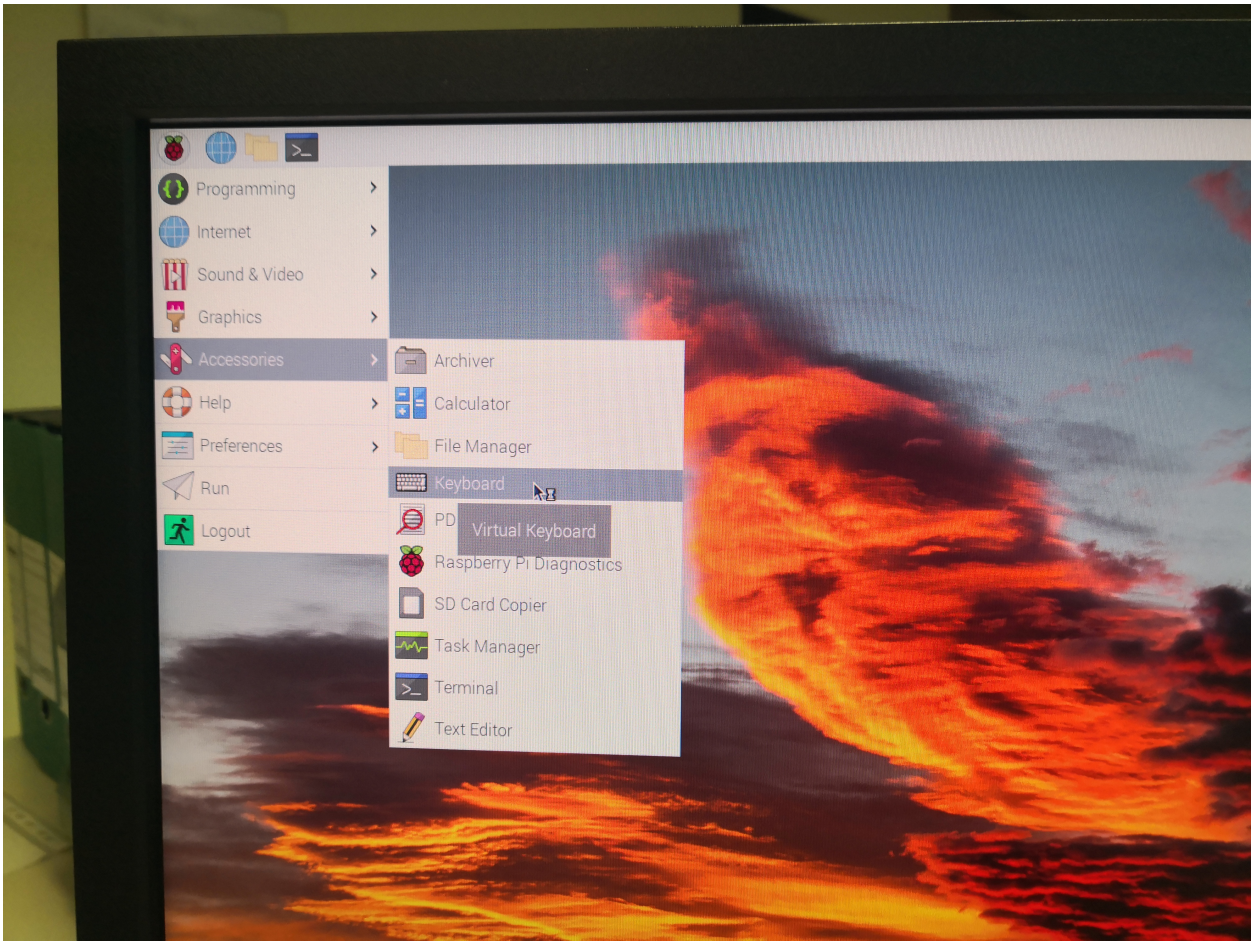
Reference: https://pimylifeup.com/raspberry-pi-on-screen-keyboard/

Logged in as pi@rpi4gm35, type the following commands to setup the On-Screen Keyboard:

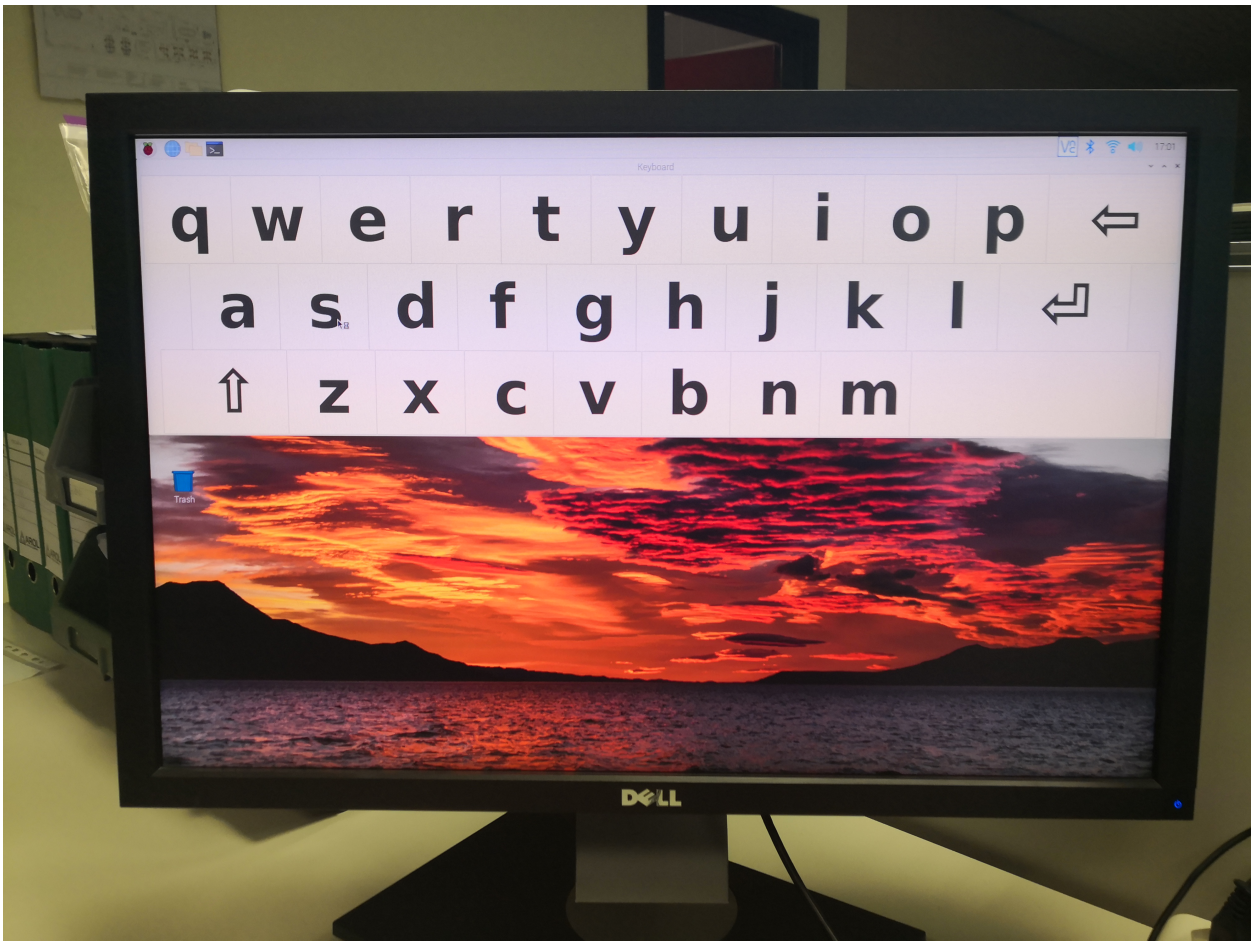
```
sudo apt update
sudo apt upgrade
sudo apt install -y matchbox-keyboard
```

Test: On the Raspberry Pi OS dashboard run

Accessories > Keyboard



Result:



Install byobu

```
sudo apt -y install byobu
```

Install git and tig

```
sudo apt update
sudo apt -y install git tig
```

Install git-aware-prompt

Reference: https://github.com/jimeh/git-aware-prompt

Clone git-aware-prompt sources from GitHub

```
mkdir ~/.bash
cd ~/.bash
git clone https://github.com/jimeh/git-aware-prompt.git
```

and customize the default shell prompt

```
cat <<END >>~/.bashrc

# Configure git-aware-prompt
export GITAWAREPROMPT=~/.bash/git-aware-prompt
source "\${GITAWAREPROMPT}/main.sh"
export PS1="\${debian_chroot:+(\$debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\
↪033[01;34m\]\w\[\033[00m\] \[$txtcyn\]\$git_branch\[$txtred\]\$git_dirty\[$txtrst\]\$ "
END
```

Logout and login for applying the changes.

Now when inside a directory versioned with git your prompt should show the branch where you are in, as in the following example

```
pi@rpi4gm35:~/.bash/git-aware-prompt (master)$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
pi@rpi4gm35:~/.bash/git-aware-prompt (master)$
```

Notice that the `(master)` branch is part the prompt.

Clone ARNEIS sources from GitHub

Logged in as pi@rpi4gm35

```
mkdir -p ~/github/B-AROL-O
cd ~/github/B-AROL-O
git clone https://github.com/B-AROL-O/ARNEIS
```

4.11.6 Make the Raspberry Pi an agent node of a k3s cluster

Make sure the OS is up-to-date

```
gmacario@HW2457:~$ ssh pi@rpi3pmv38.local
Linux rpi3pmv38 5.10.103-v8+ #1530 SMP PREEMPT Tue Mar 8 13:06:35 GMT 2022 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar 26 08:48:24 2022 from 192.168.64.105
pi@rpi3pmv38:~ $ sudo apt update && sudo apt -y dist-upgrade && sudo apt -y autoremove --
↪purge
Hit:1 http://archive.raspberrypi.org/debian bullseye InRelease
Hit:2 http://deb.debian.org/debian bullseye InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Hit:4 http://security.debian.org/debian-security bullseye-security InRelease
```

(continues on next page)

(continued from previous page)

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@rpi3pmv38:~ $
```

Attach Agent Node

```
pi@rpi3pmv38:~ $ sudo -i
root@rpi3pmv38:~# export K3S_URL=https://xxxx.example.com:6443
root@rpi3pmv38:~# export K3S_
↪TOKEN=K1015exxxx
```

Modify Linux cmdline

Display the current contents of file /boot/cmdline.txt:

```
root@rpi3pmv38:~# cat /boot/cmdline.txt
console=serial0,115200 console=tty1 root=PARTUUID=ff8e3abf-02 rootfstype=ext4 fsck.
↪repair=yes rootwait quiet splash plymouth.ignore-serial-consoles
root@rpi3pmv38:~#
```

Edit the boot configuration file according to the suggestions in the previous error message

```
sudo vi /boot/cmdline.txt
```

Finally, verify that the command-line has changed accordingly:

```
root@rpi3pmv38:~# cat /boot/cmdline.txt
console=serial0,115200 console=tty1 root=PARTUUID=ff8e3abf-02 rootfstype=ext4 fsck.
↪repair=yes rootwait quiet splash plymouth.ignore-serial-consoles cgroup_memory=1␣
↪cgroup_enable=memory
root@rpi3pmv38:~#
```

Reboot to ensure that the Linux kernel will use the updated command-line.

After the Raspberry Pi has rebooted you may inspect the Linux command-line with the following command:

```
pi@rpi3pmv38:~ $ cat /proc/cmdline
coherent_pool=1M 8250.nr_uarts=0 snd_bcm2835.enable_compat_alsa=0 snd_bcm2835.enable_
↪hdmi=1 video=HDMI-A-1:1920x1080M@60 vc_mem.mem_base=0x3ec00000 vc_mem.mem_
↪size=0x40000000  console=ttyS0,115200 console=tty1 root=PARTUUID=ff8e3abf-02␣
↪rootfstype=ext4 fsck.repair=yes rootwait quiet splash plymouth.ignore-serial-consoles␣
↪cgroup_memory=1 cgroup_enable=memory
pi@rpi3pmv38:~ $
```

Retry attaching Agent Node

```
pi@rpi3pmv38:~ $ sudo -i
root@rpi3pmv38:~# export K3S_URL=https://xxxx.example.com:6443
root@rpi3pmv38:~# export K3S_
↪TOKEN=K1015exxxx
```

(continued from previous page)

```
[INFO]  systemd: Enabling k3s-agent unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s-agent.service → /etc/
↪systemd/system/k3s-agent.service.
[INFO]  No change detected so skipping service start
root@rpi3pmv38:~#
```

Verify that the Agent Node is up-and-running

Verify the cluster configuration using `kubectl`

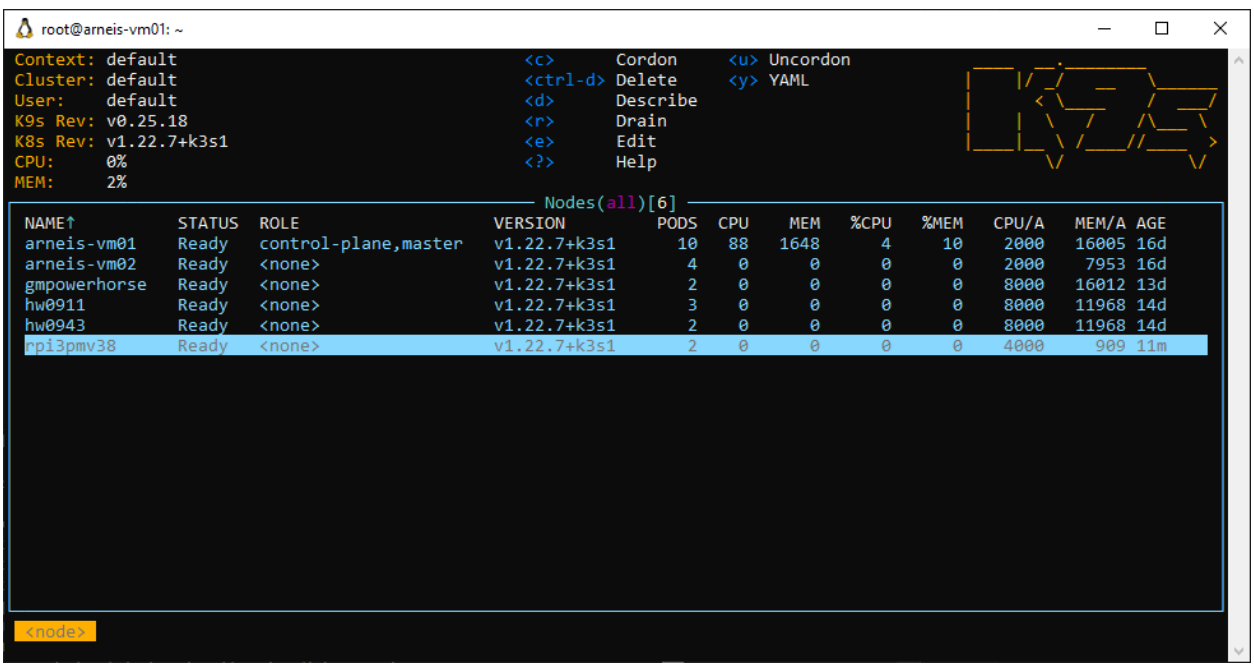
```
kubectl get nodes
```

then read the online help to understand how to display the properties of the new node.

If you have k9s - the Kubernetes CLI installed, the procedure is even more intuitive:

```
k9s
```

As soon as the k9s main window is displayed, type "`:node`" then press **Enter** to view the list of nodes which are part of the cluster.



If everything works as expected, after a few seconds your new node should show up in the Nodes view.

Use the Up and Down arrows to select that node (in our case, `rpi3pmv38`), then type "`d`" to describe the node:

```
gmacario@alpha: ~
Context: default                                  <c>       Copy
Cluster: default                                  <n>       Next Match
User:    default                                  <shift-n> Prev Match
K9s Rev: v0.25.18                                 <r>       Toggle Auto-Refresh
K8s Rev: v1.22.7+k3s1                             <f>       Toggle FullScreen
CPU:     0%
MEM:     2%
                                                 Describe(rpi3pmv38)
Name:               rpi3pmv38
Roles:              <none>
Labels:             beta.kubernetes.io/arch=arm64
                    beta.kubernetes.io/instance-type=k3s
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=arm64
                    kubernetes.io/hostname=rpi3pmv38
                    kubernetes.io/os=linux
                    node.kubernetes.io/instance-type=k3s
Annotations:        flannel.alpha.coreos.com/backend-data: {"VNI":1,"VtepMAC":"22:36:45:c7:c5:6c"}
                    flannel.alpha.coreos.com/backend-type: vxlan
                    flannel.alpha.coreos.com/kube-subnet-manager: true
                    flannel.alpha.coreos.com/public-ip: 192.168.64.232
                    k3s.io/hostname: rpi3pmv38
                    k3s.io/internal-ip: 192.168.64.232
                    k3s.io/node-args: ["agent"]
<node>  <describe>
```

Make sure that label `beta.kubernetes.io/arch` is set to `arm64`.

4.12 HOWTO Configure a Static IP Address on Linux

4.12.1 Introduction

This document explains how to configure a static IP address on the Ethernet interface of a Linux host.

With recent versions of Linux distributions, the network configuration is performed with NetworkManager.

Ubuntu configures networking through a higher abstraction tool called Netplan which internally invokes NetworkManager as a renderer.

4.12.2 Reference documents

- https://www.baeldung.com/linux/set-static-ip-address
- Netplan documentation
- NetworkManager - Debian WIki
- Ubuntu Forum: Need advice: Redoing networking on an old server?

4.12.3 Configuring Netplan

From https://netplan.io:

> Netplan is a utility for easily configuring networking on a linux system. You simply create a YAML description of the required network interfaces and what each should be configured to do. From this description Netplan will generate all the necessary configuration for your chosen renderer tool.

TODO

4.12.4 Configuring NetworkManager

NOTE: Even though NetworkManager was designed for the Gnome desktop environment, it works "just-fine" also in server editions of those OSs. Checked on the following

Vendor	OS Name	network-manager	version
Canonical	Ubuntu 22.04	network-manager	1.36.6-0ubuntu2
Raspberry Pi	Raspbian GNU/Linux 10 (buster)	dhcpd5	1:8.1.2-1+rpt1
Raspberry Pi	Debian GNU/Linux 11 (bullseye)	network-manager	1.30.6-1+deb11u1
Raspberry Pi	Debian GNU/Linux 12 (bookworm)	network-manager	1.42.4-1+rpt1

Logged in to the host which will act as `raspinstall01` execute the following commands:

```
sudo cp /etc/network/interfaces /etc/network/interfaces.ORIG
sudo vi /etc/network/interfaces/eth0-static
```

and write the following contents to the config file:

```
# file:/etc/network/interfaces.d/eth0-static

# Configure static IP address for eth0
auto eth0
iface eth0 inet static
    address 172.20.0.1
    netmask 255.255.0.0

# EOF
```

Reboot `raspinstall01` to allow the changes to take effect.

Login again and verify that `eth0` was assigned the desired IP address:

```
gmacario@raspinstall01:~ $ ip address show eth0
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default␣
↪qlen 1000
    link/ether e4:5f:01:dd:4c:0e brd ff:ff:ff:ff:ff:ff
    inet 172.20.0.1/24 brd 172.20.0.255 scope global eth0
       valid_lft forever preferred_lft forever
gmacario@raspinstall01:~ $
```

4.13 HOWTO Prepare a Raspberry Pi 4B for the ARNEIS project

4.13.1 Introduction

The following document explains how to prepare and configure a Raspberry Pi for the ARNEIS project.

4.13.2 Prerequisites

- One Raspberry Pi.
  - Tested on rpi4gm35 (Raspberry Pi 4B 4GB)
- One MicroSD card of at least 4GB.
  - **IMPORTANT**: The card should be blank, or at least should not contain any important data since it will be completely erased.
  - Tested with a SanDisk Ultra 256 GB MicroSDXC
- One desktop PC or laptop for formatting the SD card and controlling the RPi
  - OS: A recent version of Windows or Linux or macOS
  - The PC should have a MicroSDHC card reader. Alternatively, an additional USB MicroSD card reader is required
- Fast internet connection

4.13.3 Step-by-step instructions

Prepare the MicroSD card with Raspberry Pi OS

Launch a browser on your laptop and open https://www.raspberrypi.com/software/

Download Raspberry Pi Imager and install it on your laptop.

Insert a MicroSD card into one slot of your laptop. Alternatively, insert the MicroSD into the USB card reader, then plug the USB card reader into one empty USB port of your laptop.

Launch the Raspberry Pi Imager and select the following options

- Sistema operativo: Raspberry Pi OS (32-bit)
- Scheda SD: MXT-USB Storage Device USB Device - 255.9 GB

then click "SCRIVI".

> **Attenzione**
>
> Tutti i dati esistenti in 'MXT-USB Storage Device USB Device' verranno eliminati. Sei sicuro di voler continuare?
>
> NO | SI

Click "SI".

...

> **Scrittura completata senza errori**
>
> Scrittura di **Raspberry Pi OS (32-bit)** in **MXT-USB Storage Device USB Device** completata.
>
> Ora puoi rimuovere la scheda SD dal lettore
>
> CONTINUA

Remove the MicroSD from your laptop

First boot of the RPi with the new MicroSD card

(2022-01-12 08:35 CET)

- Insert the MicroSD card into your Raspberry Pi.
- Connect a display using a MicroHDMI-to-HDMI cable
- Connect a USB keyboard to the first USB 2.0 port of the RPi4
- Connect a USB mouse to the second USB 2.0 port of the RPi4
- Connect a 5Vdc, 3A power supply to the USB-C port of the RPi4

Turn on the power supply and wait for Raspberry Pi OS to boot.

> **Welcome to Raspberry Pi**
>
> Welcome to the Raspberry Pi Desktop!
>
> Before you start using it, there are a few things to set up.
>
> Press 'Next' to get started.
>
> Cancel | Next

Click "Next".

> **Set Country**
>
> Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.
>
> - Country: Italy
> - Language: Italian
> - Timezone: Rome
> - [x] Use English language
> - [ ] Use US keyboard
>
> Press 'Next' when you have made your selection.
>
> Back | Next

Fill in the form as shown above, then click "Next"

…

> **Change Password**
>
> The default 'pi' user account currently has the password 'raspberry'. It is strongly recommended that you change this to a different password that only you know
>
> - Enter new password: xxxx
> - Confirm new password: xxx

- [x] Hide characters

Press 'Next' to activate your new password.

Back | Next

Fill in the form as instructed, then click "Next"

Set Up Screen

You should be able to set the taskbar along the top of the screen. Tick the box if some or all of it does not fit on the screen.

[ ] The taskbar does not fit one the screen

The change will take effect when the Pi is restarted.

Press 'Next' to save your setting.

Back | Next

Verify and update if needed, then click "Next".

Select WiFi Network

Select your WiFi network from the list.

...

Press 'Next' to connect, or 'Skip' to continue without connecting.

Back | Skip | Next

Select the desired WiFi network, then click "Next". If the select network is protected you will be requested the password

Enter WiFi Network

Enter the password for the WiFi network 'xxxx'.

- Password: yyyy

[x] Hide characters

Press 'Next' to connect, or 'Skip' to continue without connecting.

Back | Skip | Next

Click "Next".

Update Software

The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Next' ...

Back | Skip | Next

Click "Next".

Download updates - please wait.

When the update is complete the following popup should be displayed

System is up to date

OK

Click "OK" to continue.

Setup Complete

Your Raspberry Pi is now set up and ready to go.

To run applications, click the raspberry icon in the top left corner of the screen to open the menu.

Press 'Restart' to restart your Pi so the new settings will take effect.

Back | Later | Restart

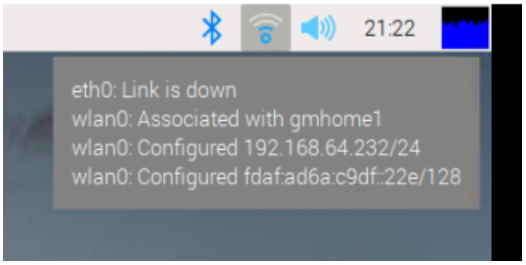
Click "Restart".

Verify that the RPi reboots correctly.

Trash
SiComputer

Display assigned IP addresses

To know the IP addresses assigned to the Raspberry Pi just move the mouse over the network icon at the top right of the desktop



Configure hostname, SSH and VNC

Open a terminal and type the following command

```
sudo raspi-config
```

- Select "1" (System Options), then "S4" (Hostname)
- Enter hostname: `rpi4gm35` (will replace default hostname `raspberrypi`)
- Select "3" (Interface Options), then "I2" (SSH)
- Select "Yes" to enable SSH
- Select "3" (Interface Options), then "I3" (VNC)
- Select "Yes" to enable VNC
- Select "Finish" to exit `raspi-config`. Reboot if requested

Verify that the RPi is accessible from the laptop via SSH and VNC (you may need to scan the local Wi-Fi network to get the IPv4 address assigned by the router)

HINT: To scan the network and identify the open services you can use one of those tools

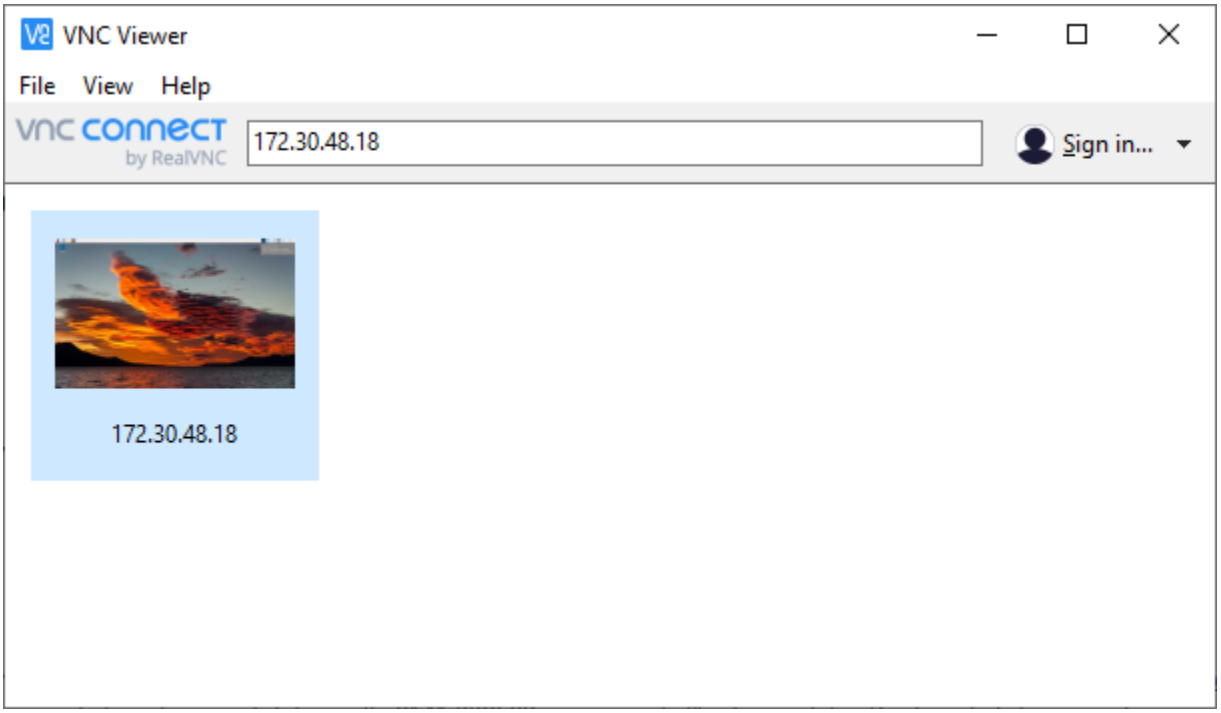
- The Fing app on a mobile phone or on your laptop.
- nmap

It this works disconnect the USB keyboard, mouse and display

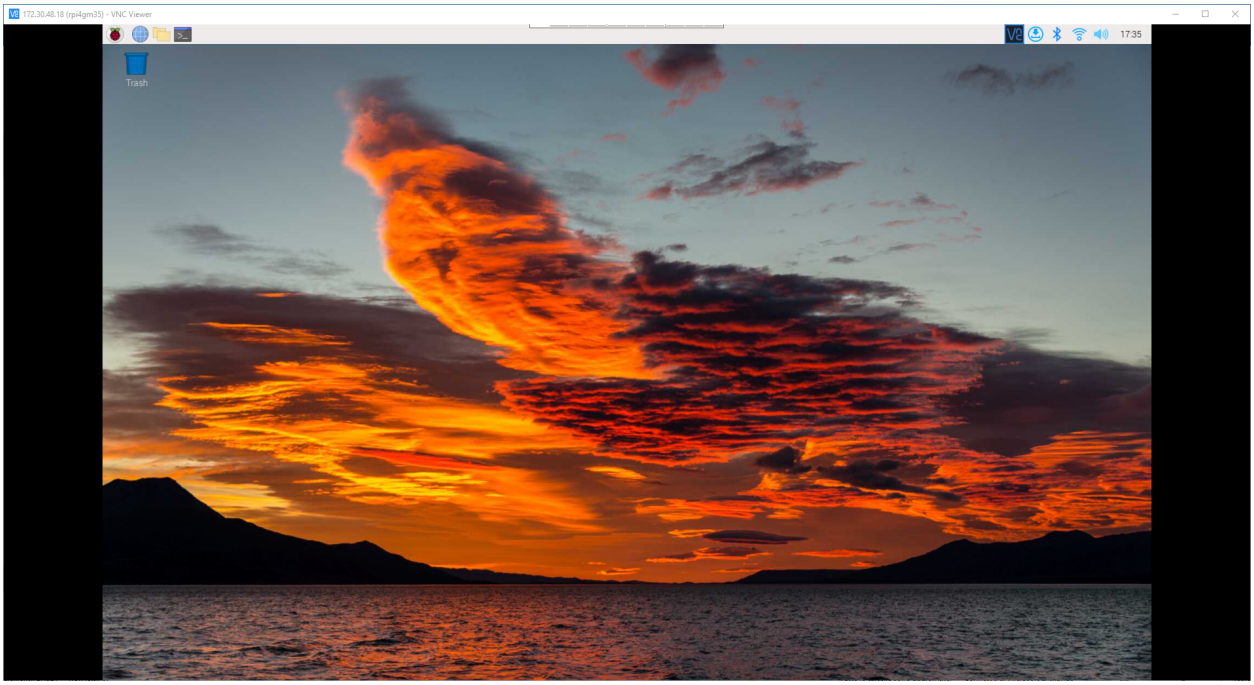
Reboot your RPi4 and verify that the device is still accessible from SSH:

```
gpmacario@HW2457 MINGW64 ~
$ ssh pi@172.30.48.18
pi@172.30.48.18's password:
Linux rpi4gm35 5.10.63-v7l+ #1488 SMP Thu Nov 18 16:15:28 GMT 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jan 12 16:36:38 2022
pi@rpi4gm35:~ $
```

Do the same using a VNC client (in my case I used the free to use VNC® Viewer)



Double click on the selected profile to connect to the remote desktop of the Raspberry Pi:



(Recommended) Create a public DNS entry

If you have administrative rights to a DNS zone you may choose to access your Raspberry Pi using a symbolic name rather than an IP address.

If so, access your DNS administrative page (in my case, https://register.it/) and create an A record to map the name to the IP address assigned to your Raspberry Pi.

In my case

```
A rpi4gm35 172.30.48.18
```

Wait until the DNS zone is propagated, then verify that the device can be accessed by another host (in our case, our laptop) using the assigned name rather than its IP address:

```
gpmacario@HW2457 MINGW64 ~
$ ping rpi4gm35.gmacario.it

Esecuzione di Ping rpi4gm35.gmacario.it [172.30.48.18] con 32 byte di dati:
Risposta da 172.30.48.18: byte=32 durata=8ms TTL=64
Risposta da 172.30.48.18: byte=32 durata=7ms TTL=64
Risposta da 172.30.48.18: byte=32 durata=6ms TTL=64
Risposta da 172.30.48.18: byte=32 durata=6ms TTL=64

Statistiche Ping per 172.30.48.18:
    Pacchetti: Trasmessi = 4, Ricevuti = 4,
    Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 6ms, Massimo =  8ms, Medio =  6ms

gpmacario@HW2457 MINGW64 ~
$
```

Configure public SSH keypair

Logged in as pi@rpi4gm35, create a public/private SSH keypair:

```
ssh-keygen
```

Type the following commands to be able to login to your Raspberry Pi through your public SSH key - for instance:

```
cat <<END >>~/.ssh/authorized_keys
ssh-rsa AAAAB3Nza....W1cG35r8= gpmacario@HW2457
END
```

Test

```
gpmacario@HW2457 MINGW64 ~
$ ssh pi@rpi4gm35.gmacario.it
Linux rpi4gm35 5.10.92-v7l+ #1514 SMP Mon Jan 17 17:38:03 GMT 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

(continues on next page)

(continued from previous page)

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan 20 09:12:23 2022
pi@rpi4gm35:~ $
```

Install Virtual Keyboard

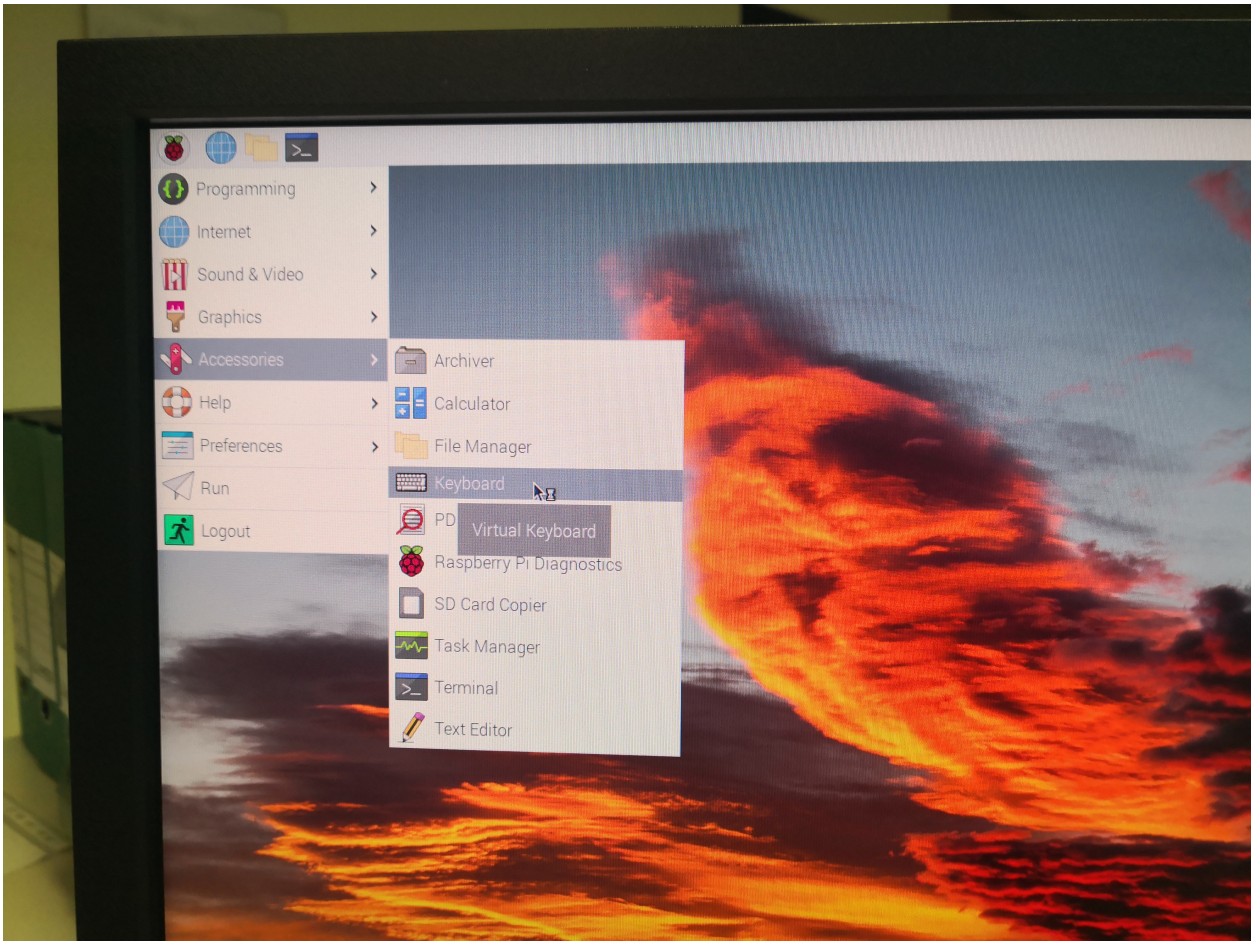
Reference: https://pimylifeup.com/raspberry-pi-on-screen-keyboard/

Logged in as pi@rpi4gm35, type the following commands to setup the On-Screen Keyboard:

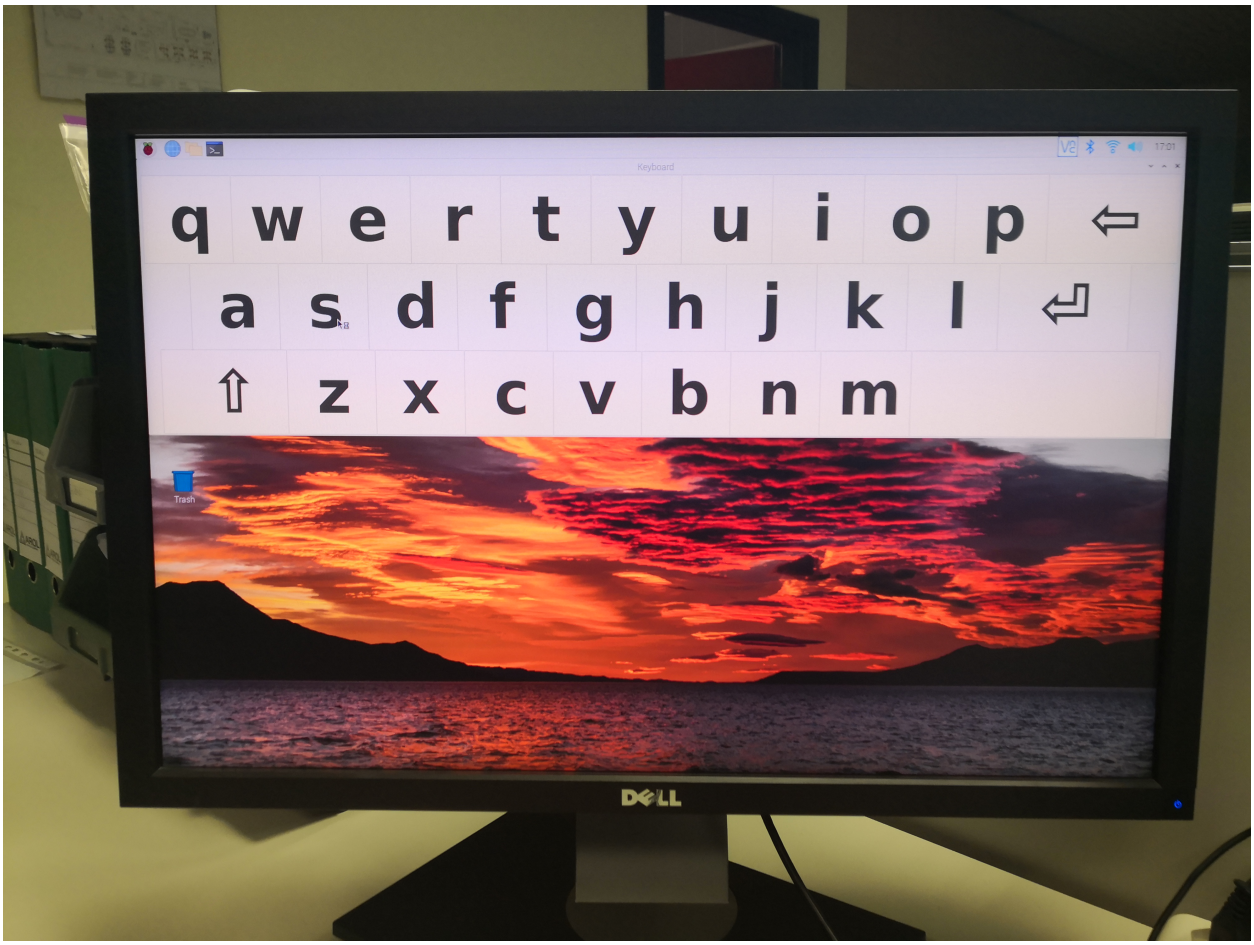
```
sudo apt update
sudo apt upgrade
sudo apt install -y matchbox-keyboard
```

Test: On the Raspberry Pi OS dashboard run

Accessories > Keyboard



Result:



Install byobu

```
sudo apt -y install byobu
```

Install git and tig

```
sudo apt update
sudo apt -y install git tig
```

Install git-aware-prompt

Reference: https://github.com/jimeh/git-aware-prompt

Clone git-aware-prompt sources from GitHub

```
mkdir ~/.bash
cd ~/.bash
git clone https://github.com/jimeh/git-aware-prompt.git
```

then append the following lines to `~/.bashrc` to customize the default shell prompt:

```
# Configure git-aware-prompt
export GITAWAREPROMPT=~/.bash/git-aware-prompt
source "\${GITAWAREPROMPT}/main.sh"
export PS1="\${debian_chroot:+(\$debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\
↪033[01;34m\]\w\[\033[00m\] \[$txtcyn\]\$git_branch\[$txtred\]\$git_dirty\[$txtrst\]\$ "
```

Logout and login for applying the changes.

Now when inside a directory versioned with git your prompt should show the branch where you are in, as in the following example

```
pi@rpi4gm35:~/.bash/git-aware-prompt (master)$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
pi@rpi4gm35:~/.bash/git-aware-prompt (master)$
```

Notice that the `(master)` branch is part the prompt.

Clone ARNEIS sources from GitHub

Logged in as pi@rpi4gm35

```
mkdir -p ~/github/B-AROL-O
cd ~/github/B-AROL-O
git clone https://github.com/B-AROL-O/ARNEIS.git
```

4.14 HOWTO Install MicroPython on an Arduino Nano 33 BLE

4.14.1 Introduction

This document explains how to install MicroPython on an Arduino Nano 33 BLE embedded board, then run a sample MicroPython program.

The Nano 33 BLE is Arduino's 3.3V compatible board in the smallest available form factor: 45x18mm!

The Arduino Nano 33 BLE is a completely new board on a well-known form factor. It comes with an embedded 9 axis inertial sensor what makes this board ideal for wearable devices, but also for a large range of scientific experiments in the need of short-distance wireless communication.

4.14.2 Reference Documents

- MicroPython Home Page
- Arduino Nano 33 BLE product page on the Arduino Store
- MicroPython with Arduino Boards
- Install MicroPython on your Nano BLE
- Getting started with OpenMV and Nano 33 BLE
- Nano 33 BLE Python® API guide (a collection of useful scripts)]

- https://github.com/micropython/micropython
- MicroPython 101 - A course for learning MicroPython with the Arduino Nano ESP32

4.14.3 Step-by-step instructions

Reference: https://docs.arduino.cc/tutorials/nano-33-ble-sense/micropython-installation#arduino-nano-33-ble

Install Arduino IDE

Download latest release of the Arduino IDE (2.2.1 as of 2023-09-19) from https://www.arduino.cc/en/software.

```
# Download Arduino IDE
mkdir -p ~/Downloads && cd ~/Downloads
curl -fsSL -O https://downloads.arduino.cc/arduino-ide/arduino-ide_2.2.1_Linux_64bit.
↪AppImage
chmod a+x arduino-ide_2.2.1_Linux_64bit.AppImage

# Run Arduino IDE
~/Downloads/arduino-ide_2.2.1_Linux_64bit.AppImage
```

Follow the on-screen instructions to install the Arduino IDE on your PC.

Run a sample sketch on the Arduino Nano 33 BLE

Arduino: File > Examples > 01.Basics > Blink

Arduino: Select Board: Arduino Nano 33 BLE on `/dev/ttyACM0`

> The "Arduino Mbed OS Nano Boards [v4.0.6]" core has to be installed for the currently selected "Arduino Nano 33 BLE" board. Do you want to install it now?

Click **YES**.

Arduino IDE: Sketch > Upload

Wait until the sketch has been successfully uploaded to the Arduino Nano 33 BLE. The LED on the board should blink every second.

Try modifying the sketch changing the value in the `delay(...)` statements, then upload the new sketch and make sure the blink frequency changes accordingly.

Installing the Core

Arduino IDE: Tools > Board > Board Manager

- Make sure that the latest version of "Arduino Mbed OS Nano Boards" (4.0.6 as of 2023-09-20) is installed.

Update Bootloader

Arduino IDE: File > Examples > Nano33BLE_System > Nano33_updateBLandSoftDevice

and open the sketch.

Arduino IDE: Sketch > Upload

Arduino IDE: Tools > Serial Monitor

```
Flashing bootloader...
Flash Address = E0000
Flashed 11%
Flash Address = E1000
Flashed 22%
Flash Address = E2000
Flashed 34%
Flash Address = E3000
Flashed 45%
Flash Address = E4000
Flashed 56%
Flash Address = E5000
Flashed 68%
Flash Address = E6000
Flashed 79%
Flash Address = E7000
Flashed 91%
Flash Address = E8000
Flashed 100%

Bootloader 32bit CRC is: CB51658F
Writing in UICR memory the address of the new bootloader...

Bootloader update successfully completed!

Would you like to install the Soft Device (required for OpenMV)? Y/N
Storing SoftDevice length info at 0xFF000...
Flashing SoftDevice...
Flash Address = A0000
Flashed 2%
Flash Address = A1000
Flashed 5%
...
Flash Address = C3000
Flashed 97%
Flash Address = C4000
Flashed 100%


SoftDevice update complete! The board is restarting...
```

Now close the serial monitor, we will need to use the serial port for other things very soon.

Downloading firmware

Now you will need to find the specific firmware that you need to flash to your board. You can find the available firmware on the MicroPython page.

Download the .bin file that corresponds to the board you have.

Latest stable Nano BLE Sense Firmware (as of 2023-09-20):

- (stable) 20230426-v1.20.0.bin

Result:

```
gmacario@hw2228:~ $ ls -la Downloads/ARDUINO*
-rw-rw-r-- 1 gmacario gmacario 274216 set 20 13:40 Downloads/ARDUINO_NANO_33_BLE_SENSE-
↪20230426-v1.20.0.bin
gmacario@hw2228:~ $ sha256sum Downloads/ARDUINO*
fbe9740fdaf7d341387424c46dd43cbe64abf9a8daaea453c268dab740938a36  Downloads/ARDUINO_NANO_
↪33_BLE_SENSE-20230426-v1.20.0.bin
gmacario@hw2228:~ $
```

Board Installation

Reference: https://docs.arduino.cc/micropython/basics/board-installation

~~Download Arduino MicroPython Installer here~~ ==> NO, THIS ONLY APPLIES TO Nano ESP32!

Finding BOSSAC

Extract `bossac` from the installation package within Arduino IDE plugins.

In our example we have installed Arduino IDE on hw2228 which is running Ubuntu 22.04 desktop:

```
mkdir -p ~/bin && cd ~/bin
cd ~/.arduino15/staging/packages
ls -la bossac*
tar tvzf bossac*
cd ~
tar xvzf ~/.arduino15/staging/packages/bossac*.tar.gz
ls -la ~/bin/bossac
```

Now add the installation directory to `PATH` and print `bossac` help:

```
PATH=~/bin:$PATH

bossac --help
```

Result:

```
gmacario@hw2228:~ $ PATH=~/bin:$PATH
gmacario@hw2228:~ $ bossac --help
Usage: bossac [OPTION...] [FILE]
Basic Open Source SAM-BA Application (BOSSA) Version 1.9.1-17-g89f3556
Flash programmer for Atmel SAM devices.
Copyright (c) 2011-2018 ShumaTech (http://www.shumatech.com)
```

(continues on next page)

(continued from previous page)

```
Examples:
  bossac -e -w -v -b image.bin   # Erase flash, write flash with image.bin,
                                 # verify the write, and set boot from flash
  bossac -r0x10000 image.bin     # Read 64KB from flash and store in image.bin

Options:
  -e, --erase           erase the entire flash starting at the offset
  -w, --write           write FILE to the flash; accelerated when
                        combined with erase option
  -r, --read[=SIZE]     read SIZE from flash and store in FILE;
                        read entire flash if SIZE not specified
  -v, --verify          verify FILE matches flash contents
  -o, --offset=OFFSET   start erase/write/read/verify operation at flash OFFSET;
                        OFFSET must be aligned to a flash page boundary
  -p, --port=PORT       use serial PORT to communicate to device;
                        default behavior is to use first serial port
  -b, --boot[=BOOL]     boot from ROM if BOOL is 0;
                        boot from FLASH if BOOL is 1 [default];
                        option is ignored on unsupported devices
  -c, --bod[=BOOL]      no brownout detection if BOOL is 0;
                        brownout detection is on if BOOL is 1 [default]
  -t, --bor[=BOOL]      no brownout reset if BOOL is 0;
                        brownout reset is on if BOOL is 1 [default]
  -l, --lock[=REGION]   lock the flash REGION as a comma-separated list;
                        lock all if not given [default]
  -u, --unlock[=REGION] unlock the flash REGION as a comma-separated list;
                        unlock all if not given [default]
  -s, --security        set the flash security flag
  -i, --info            display device information
  -d, --debug           print debug messages
  -U, --usb-port[=BOOL] force serial port detection to USB if BOOL is 1 [default]
                        or to RS-232 if BOOL is 0
  -R, --reset           reset CPU (if supported)
  -a, --arduino-erase   erase and reset via Arduino 1200 baud hack
  -h, --help            display this help text
  -V, --version         display version info

Report bugs to <bugs@shumatech.com>
gmacario@hw2228:~ $
```

Flash the firmware using the `bossac` command as shown

```
bossac -e -w --offset=0x16000 --port={port} -i -d -U -R {firmware}
```

In our example we have

- port: `/dev/ttyACM0`
- firmware: `~/Downloads/ARDUINO_NANO_33_BLE_SENSE-20230426-v1.20.0.bin`

(2023-09-21 08:51 CEST)

Result:

```
macario@hw2228:~ $ bossac -e -w --offset=0x16000 --port=/dev/ttyACM0 -i -d -U -R ~/
↪Downloads/ARDUINO_NANO_33_BLE_SENSE-20230426-v1.20.0.bin
Set binary mode
version()=Arduino Bootloader (SAM-BA extended) 2.0 [Arduino:IKXYZ]
Connected at 921600 baud
identifyChip()=nRF52840-QIAA
write(addr=0,size=0x34)
writeWord(addr=0x30,value=0x400)
writeWord(addr=0x20,value=0)
version()=Arduino Bootloader (SAM-BA extended) 2.0 [Arduino:IKXYZ]
Device       : nRF52840-QIAA
Version      : Arduino Bootloader (SAM-BA extended) 2.0 [Arduino:IKXYZ]
Address      : 0x0
Pages        : 256
Page Size    : 4096 bytes
Total Size   : 1024KB
Planes       : 1
Lock Regions : 0
Locked       : none
Security     : false
Erase flash
chipErase(addr=0x16000)

Done in 0.001 seconds
Write 274216 bytes to flash (67 pages)
[                              ] 0% (0/67 pages)write(addr=0x34,size=0x1000)
writeBuffer(scr_addr=0x34, dst_addr=0x16000, size=0x1000)
write(addr=0x34,size=0x1000)
writeBuffer(scr_addr=0x34, dst_addr=0x17000, size=0x1000)
write(addr=0x34,size=0x1000)
writeBuffer(scr_addr=0x34, dst_addr=0x18000, size=0x1000)
[=                             ] 4% (3/67 pages)write(addr=0x34,size=0x1000)
writeBuffer(scr_addr=0x34, dst_addr=0x19000, size=0x1000)
...
[============================ ] 98% (66/67 pages)write(addr=0x34,size=0x1000)
writeBuffer(scr_addr=0x34, dst_addr=0x58000, size=0x1000)
[=============================] 100% (67/67 pages)
Done in 10.430 seconds
reset()
gmacario@hw2228:~ $
```

Run MicroPython

Launch a terminal emulator and connect to the serial port

```
MicroPython v1.20.0 on 2023-04-26; Arduino Nano 33 BLE Sense with NRF52840
Type "help()" for more information.
>>>
```

Display help

```
>>> help()
Welcome to MicroPython!

For online help please visit http://micropython.org/help/.

Quick overview of commands for the board:
  board.LED(n)    -- create an LED object for LED n (n=1,2,3,4)

If compiled with SD=<softdevice> the additional commands are
available:
  ble.enable()    -- enable bluetooth stack
  ble.disable()   -- disable bluetooth stack
  ble.enabled()   -- check whether bluetooth stack is enabled
  ble.address()   -- return device address as text string

Control commands:
  CTRL-A        -- on a blank line, enter raw REPL mode
  CTRL-B        -- on a blank line, enter normal REPL mode
  CTRL-D        -- on a blank line, do a soft reset of the board
  CTRL-E        -- on a blank line, enter paste mode

For further help on a specific object, type help(obj)
>>>
```

NOTE: MicroPython is a subset of Python

```
MicroPython v1.20.0 on 2023-04-26; Arduino Nano 33 BLE Sense with NRF52840
Type "help()" for more information.
>>> foo = "Giovanni"
>>> foo
'Giovanni'
>>> foo[1:-1]
'iovann'
>>> foo[::2]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NotImplementedError: only slices with step=1 (aka None) are supported
>>>
```

4.15 HOWTO Getting started with Foundries.io

4.15.1 Introduction

This document explains how to use Foundries.io FoundriesFactory cloud service.

FoundriesFactory is a cloud service to build, test, deploy, and maintain secure, updatable IoT and Edge products. It is used to customize open source software projects including u-Boot, OP-TEE, OE/Yocto Project, the Linux microPlatform™ and Docker®.

4.15.2 Reference documents

- https://foundries.io/
- https://docs.foundries.io/
- https://foundries.io/insights/news/foundriesio-arduino-secure-embedded-solution/

4.15.3 Preparation

Login to https://app.foundries.io/, or first create an account if you have not one.

Create a Factory

Browse https://app.foundries.io/factories

> **No Factories**
>
> Create your first Factory, and start your next product.
>
> New Factory… | Learn more

Click on "New Factory…"

> **Create Factory**
>
> **Choose a name for your factory**
>
> Fields marked with * are required
>
> - **Platform** *: Default (RaspberryPi 64-bit)
> - **Factory name** *: (empty)
> - 2 to 26 lowercase alphanumeric characters, must start with an alphanumeric character, can contain also - and _
>
> Cancel | Prev | Next

Fill in the required information

- Platform: `Default (RaspberryPi 4 64-bit)`
- Factory name: `test-fio-raspi4`

then click "Next".

Select a subscription plan for your factory

Pay Monthly | Pay Yearly

Free / $0 / 30-day trial

- No credit card required
- For personal use
- 3 builds a day
- 10 managed devices
- community support

Commercial / $5,000 / product / month

- For commercial use
- Unlimited builds
- Unlimited managed devices
- Priority support

Prices do not include taxes: we might have to collect taxes based on your billing country.

Cancel | Prev | Next

Select "Free", then click "Next"

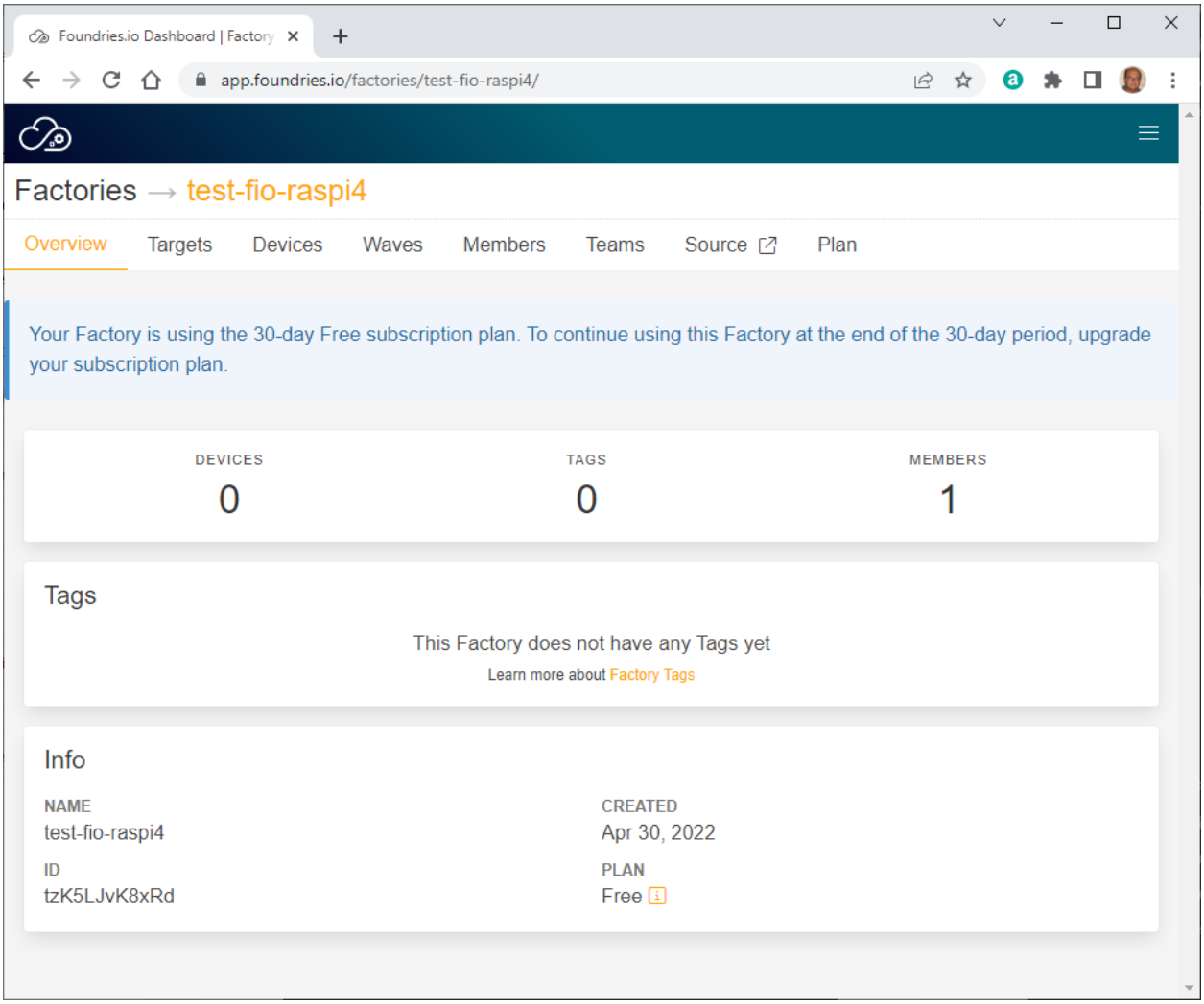
Almost there, review & create your factory

Fields marked with * are required

- Factory name: test-fio-raspi4
- Factory platform: Default (RaspberryPi 4 64-bit)
- Selected plan: Free Factory
- Plan fee: $ 0
- Taxes: $ 0
- Total before taxes: $ 0
- Total due on **Apr 30, 2022**: $ 0
- [ ] I agree to the FoundriesFactory Subscription terms *

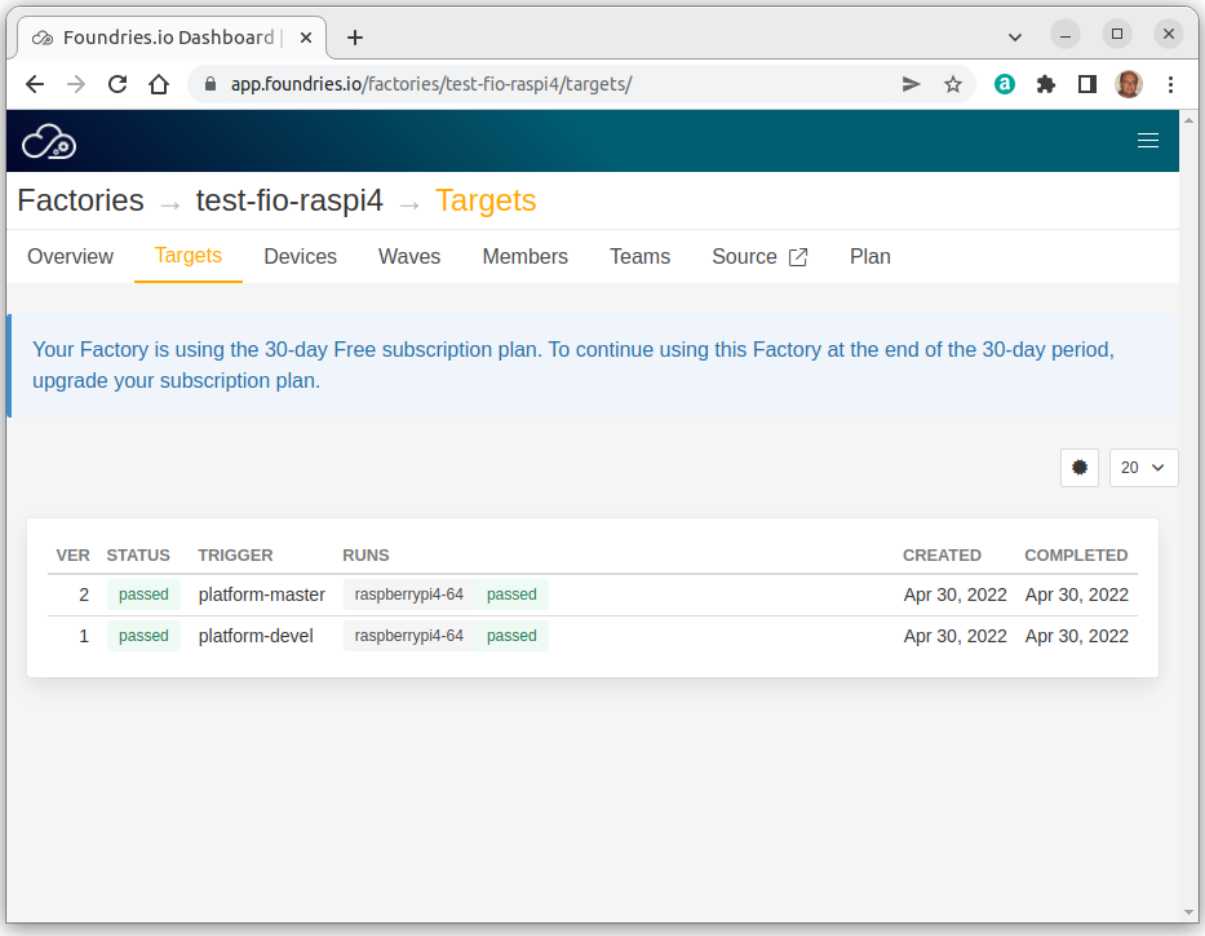
Cancel | Prev | Create Factory

Review the displayed information, check "I agree", then click "Create Factory".

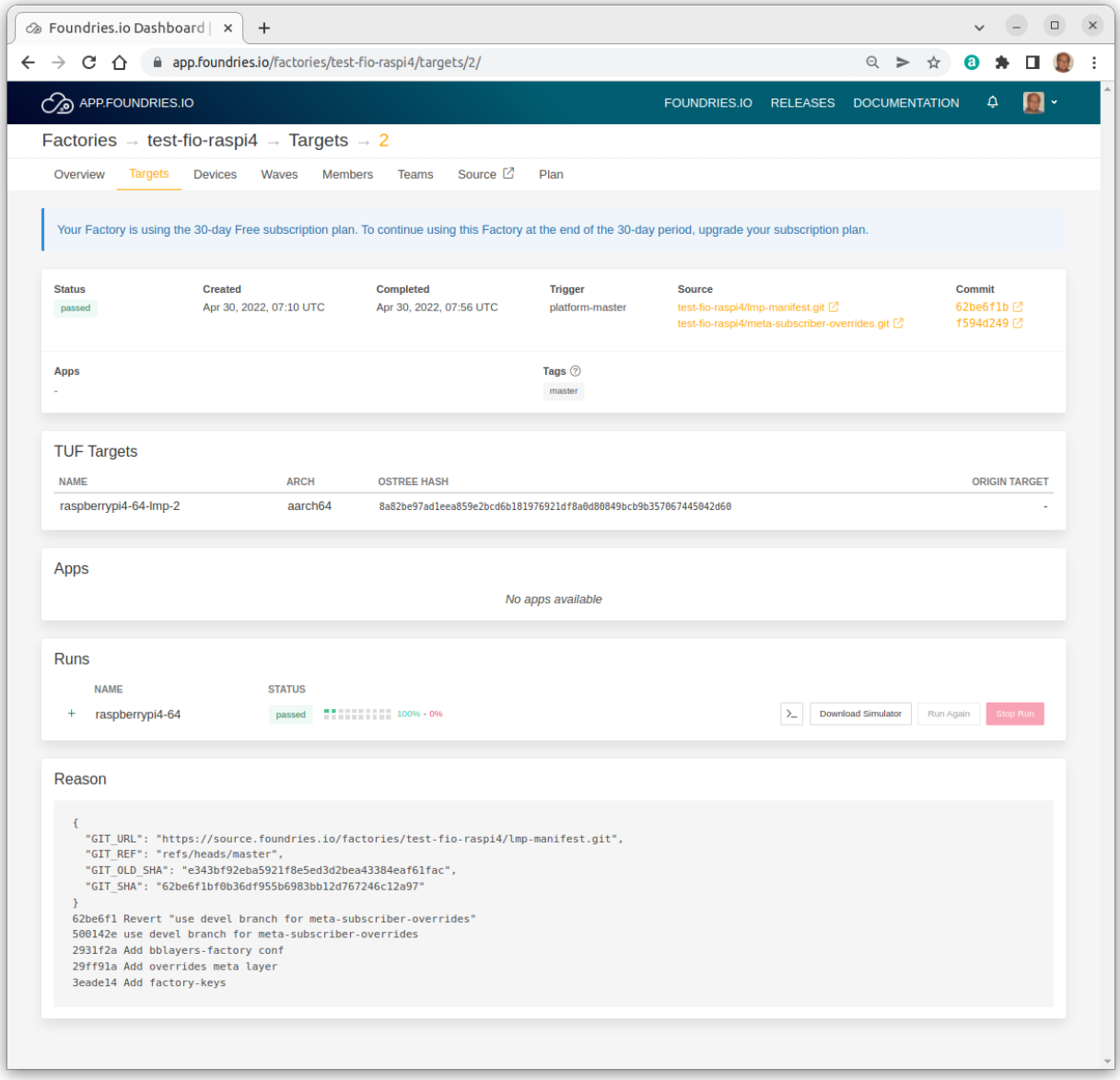


Inspect the built system images

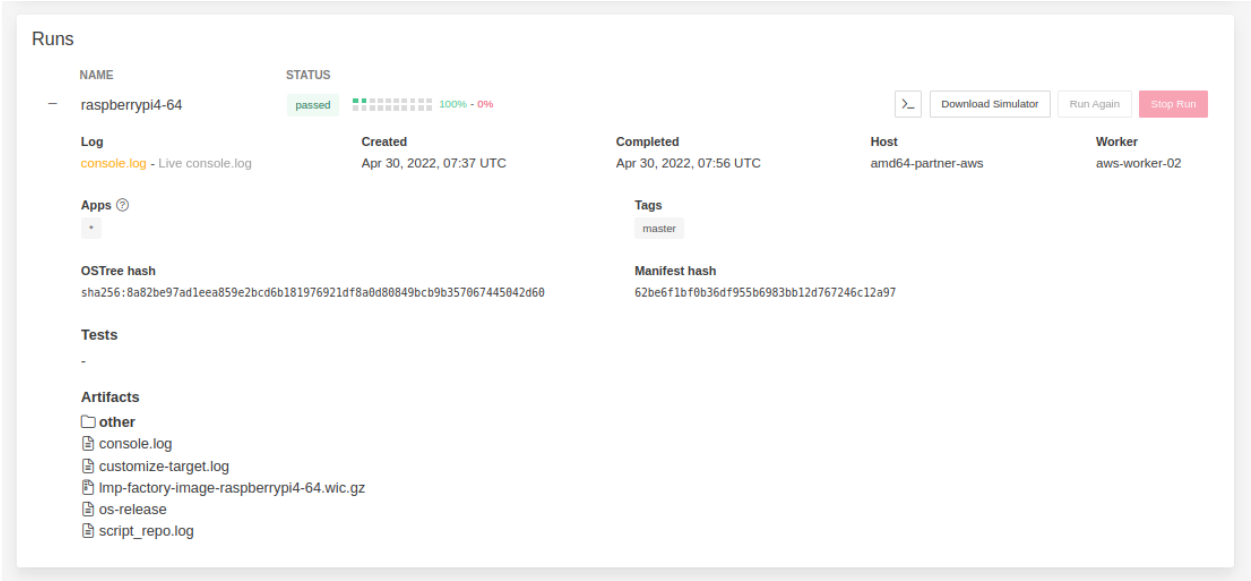
If you click on tab "Targets" you will find the list of completed builds:



As an example, click on version “2” to display the details of this specific target build:



If you expand the “Runs” panel you have access to the build artifacts:



Click on a specific artifact to display its contents.

Install the `fioctl` tool

Install the `fioctl` tool following the instructions at https://docs.foundries.io/latest/getting-started/install-fioctl/index.html - for instance, on Linux

```
export FIOCTL_VERSION=v0.25
sudo curl -o /usr/local/bin/fioctl -LO https://github.com/foundriesio/fioctl/releases/
↪download/$FIOCTL_VERSION/fioctl-linux-amd64
sudo chmod +x /usr/local/bin/fioctl
```

Verify the correct installation of `fioctl`

```
gmacario@hw2228:~ $ fioctl
Manage Foundries Factories

Usage:
  fioctl [command]

Available Commands:
  completion       Generate completion script
  config           Manage configuration common to all devices in a factory
  configure-docker Configure a hub.foundries.io Docker credential helper
  devices          Manage devices registered to a factory
  event-queues     Manage event queues configured for a Factory
  help             Help about any command
  keys             Manage keys in use by your factory fleet
  login            Access Foundries.io services with your client credentials
  logout           Remove Foundries.io client credentials from system
  secrets          Manage secret crendentials configured in a factory
  status           Get dashboard view of a factory and its devices
  targets          Manage factory's TUF targets
  teams            List teams belonging to a FoundriesFactory
```

(continues on next page)

(continued from previous page)

```
  users           List users with access to a FoundriesFactory
  version         Show version information of this tool.
  waves           Manage factory's waves

Flags:
  -c, --config string   config file (default is $HOME/.config/fioctl.yaml)
  -h, --help            help for fioctl
  -v, --verbose         Print verbose logging

Use "fioctl [command] --help" for more information about a command.
gmacario@hw2228:~ $
```

4.15.4 Deploy your first device using FoundriesFactory

Follow the instructions at https://docs.foundries.io/latest/getting-started/flash-device/index.html

In our example we will be creating a bootable microSD Card for the Raspberry Pi 4.

Download LmP system image

Login to https://app.foundries.io/, then select a Factory which you have access to - in our example, `test-fio-raspi4`.

(Optional) Take note of the Source URLs and their related Commits - in our example:

Source	Commit
test-fio-raspi4/lmp-manifest.git	500142e1
test-fio-raspi4/meta-subscriber-overrides.git	f594d249

Navigate to the "Targets" section of your Factory, then click the latest Target with the `platform-devel` Trigger. Expand the run in the "Runs" section with corresponds with the name of the board, then click "Download Simulator" (TBV).

In our example, this operation will download file `lmp-factory-image-raspberrypi4-64.wic.gz` (218 MB).

Flash LmP system image

Write the image to a blank microSD Card using a tool such as balenaEtcher.

Boot device and connect to the network

1. Connect an Ethernet cable to the board
2. Insert the microSD into the Raspberry Pi and
3. Power up the board

You board will connect to the network via Ethernet and will be ready to connect within a minute or two of booting.

```
Linux-microPlatform 3.4.3-1-86-4-g500142e raspberrypi4-64 tty1

raspberrypi4-64 login:
```

Default username/password: `fio/fio`

Verify assigned IP address

```
fio@raspberrypi4-64:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen␣
↪1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen␣
↪1000
    link/ether e4:5f:01:35:8f:96 brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.174/24 brd 192.168.64.255 scope global dynamic noprefixroute eth0
       valid_lft 43123sec preferred_lft 43123sec
    inet6 fdaf:ad6a:c9df::17a/128 scope global noprefixroute
       valid_lft forever preferred_lft forever
    inet6 fdaf:ad6a:c9df:0:d17:705f:18ab:1617/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
    inet6 fe80::eab4:9416:5a41:8b86/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
4: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group␣
↪default qlen 1000
    link/ether ce:fd:ee:3d:6e:47 brd ff:ff:ff:ff:ff:ff permaddr dc:a6:32:4e:db:b2
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group␣
↪default
    link/ether 02:42:4f:27:c0:b7 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
fio@raspberrypi4-64:~$
```

Register your device

Reference: https://docs.foundries.io/latest/getting-started/register-device/index.html

From a console on the device run this command to register the device to your factory:

```
sudo lmp-device-register -n <device-name>
```

Example:

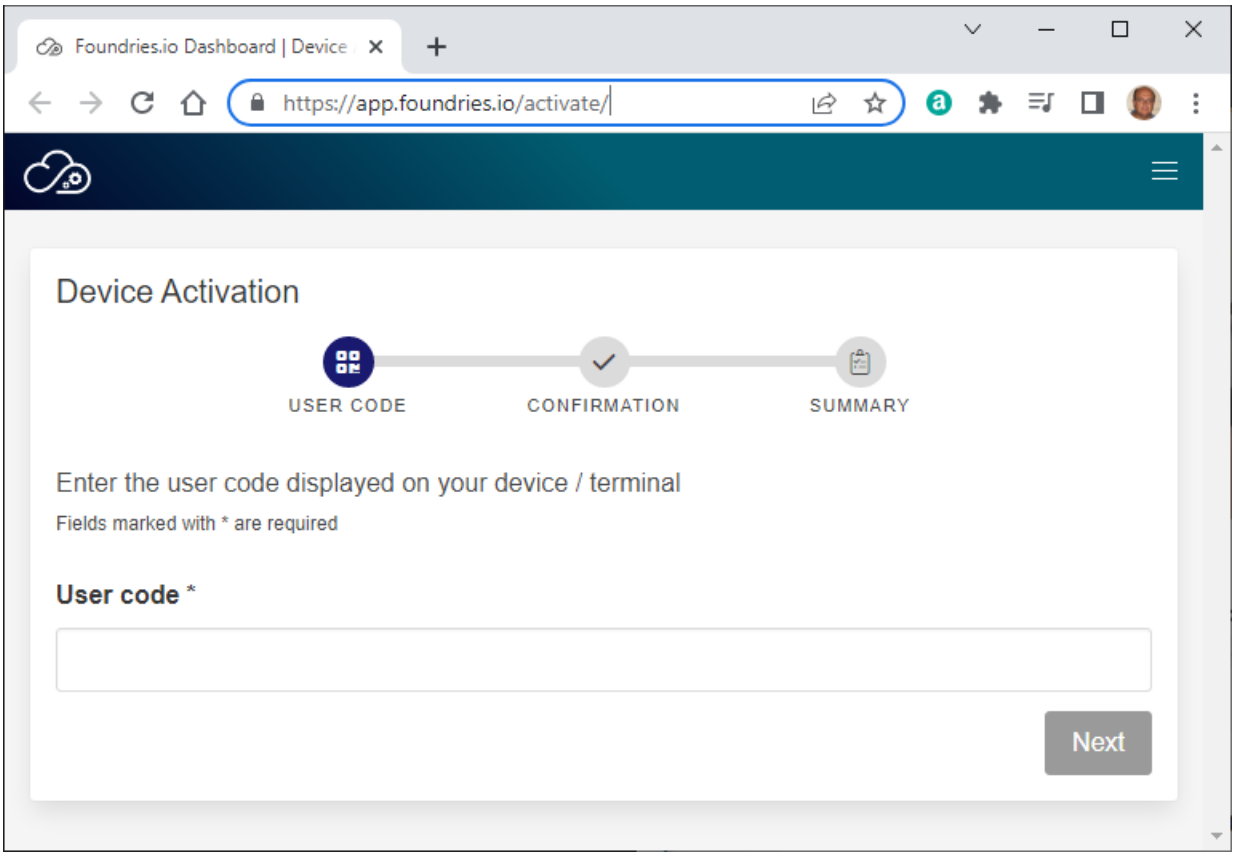
```
fio@raspberrypi4-64:~$ sudo lmp-device-register -n rpird102
Password:
Token is not specified, getting an oauth token from Foundries' auth endpoint...

--------------------------------------------------------------------------------
Visit the link below in your browser to authorize this new device. This link
will expire in 15 minutes.
  Device Name: 1f658bf4-cfb1-4546-88cf-01577b67da1b
  User code: yDeY-Aevh
```

(continues on next page)

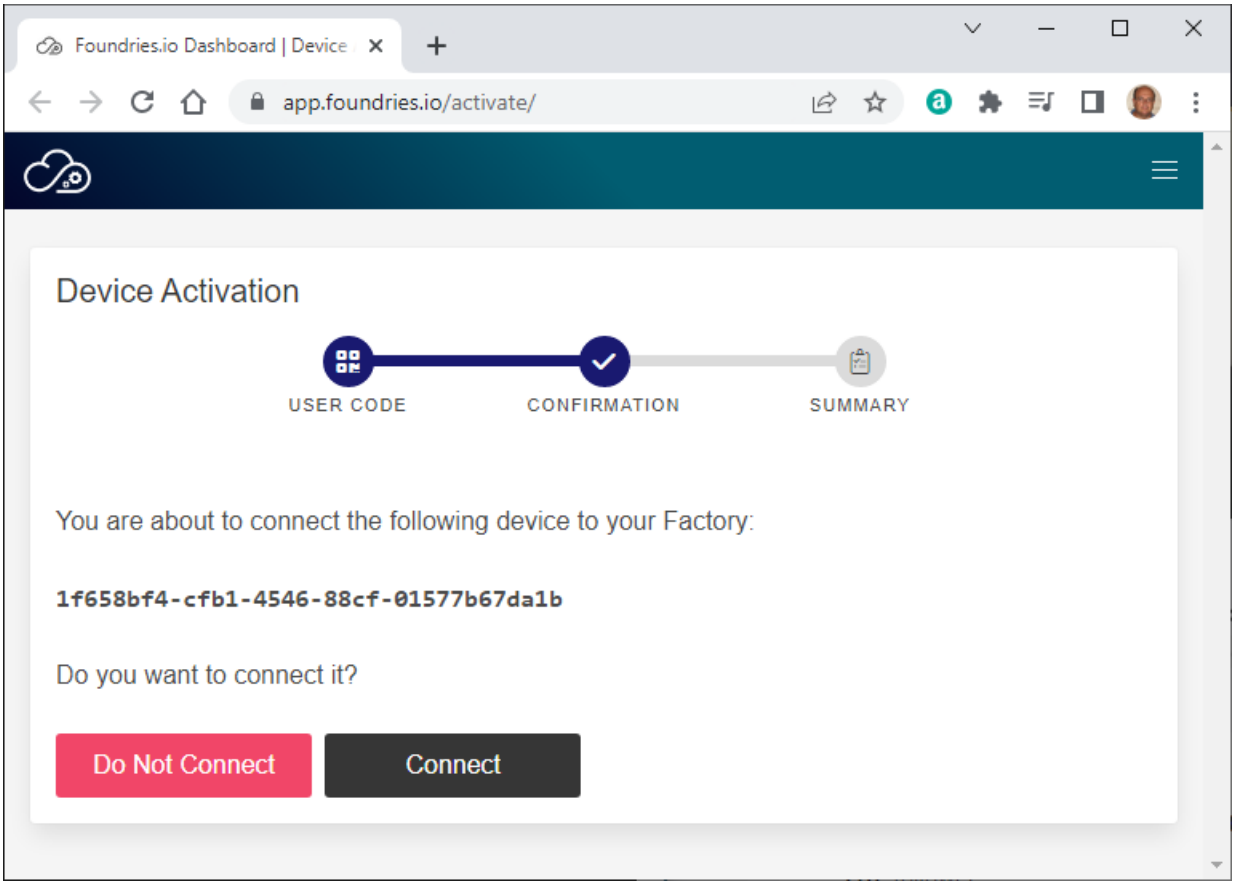
(continued from previous page)

```
  Browser URL: https://app.foundries.io/activate/

Waiting for authorization /
```

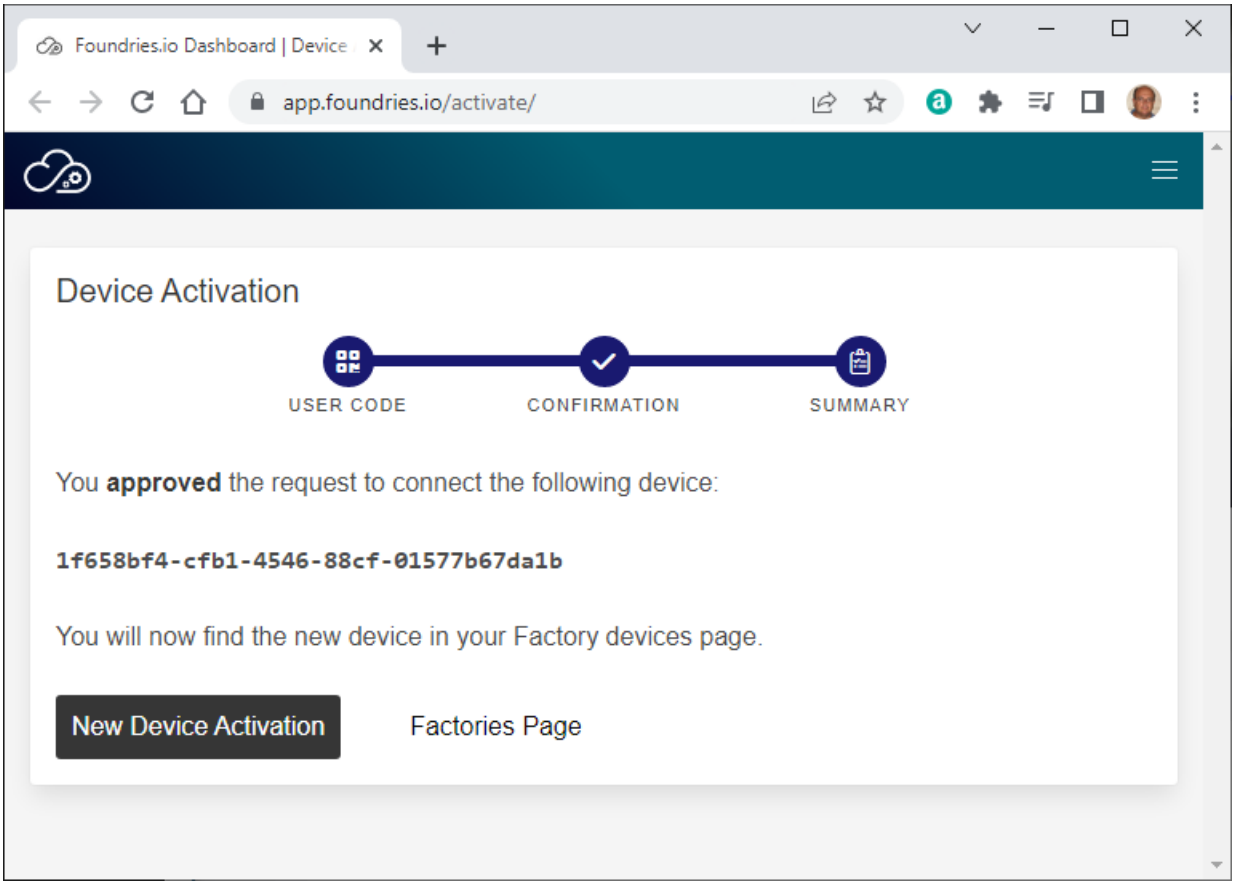
From your browser visit the displayed URL https://app.foundries.io/activate/:



Type the specified User code, then click "Next":



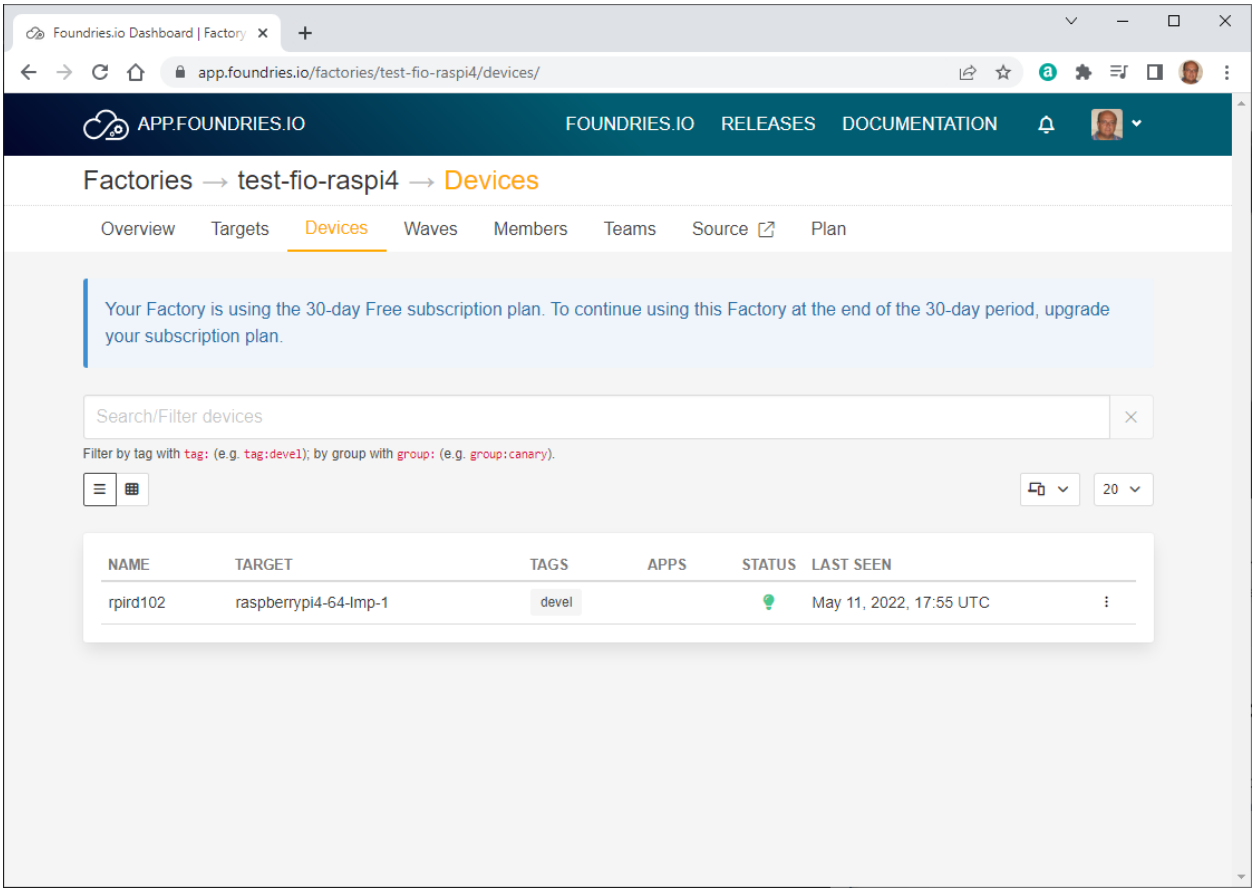
Verify the displayed UUID, then click “Connect” if OK.



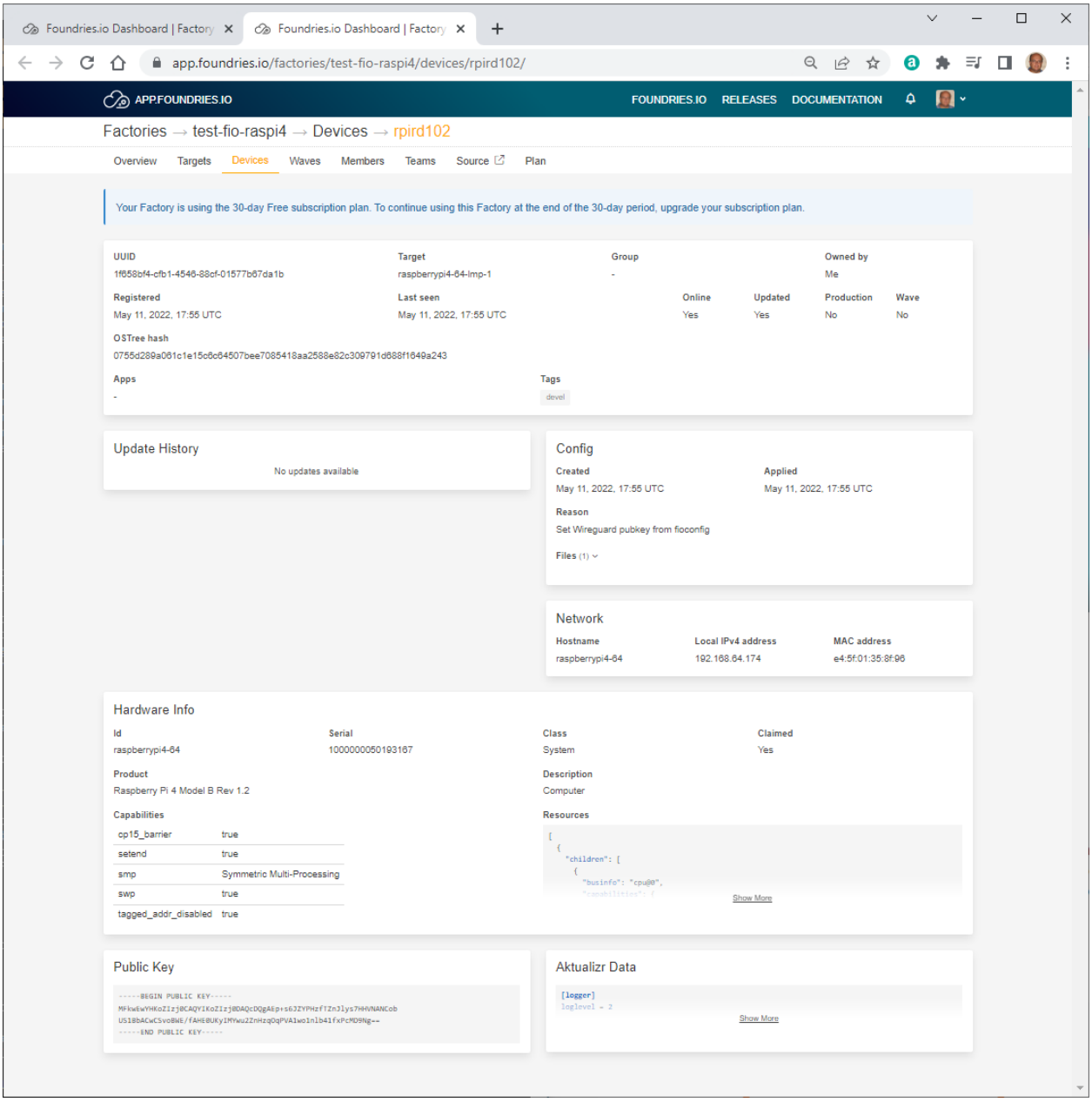
Some more messages will be displayed on the console:

```
fio@raspberrypi4-64:~$ sudo lmp-device-register -n rpird102
Password:
Token is not specified, getting an oauth token from Foundries' auth endpoint...

------------------------------------------------------------------------------
Visit the link below in your browser to authorize this new device. This link
will expire in 15 minutes.
  Device Name: 1f658bf4-cfb1-4546-88cf-01577b67da1b
  User code: yDeY-Aevh
  Browser URL: https://app.foundries.io/activate/

Registering device, rpird102, to factory test-fio-raspi4.
Device UUID: 1f658bf4-cfb1-4546-88cf-01577b67da1b
Device is now registered.
Starting aktualizr-lite daemon
fio@raspberrypi4-64:~$
```

Verify on the “Devices” tab of the Factories page that the new device is now in the list of registered devices:



Click on the device to get more details:



NOTE: This device was registered on Gianpaolo RPi4, TBV whether the image will also work on the actual `rpird102`.

4.15.5 Clone the factory Source Repositories

Create a Source Code Access Token

Login to https://app.foundries.io/ then click "Settings" > "API Tokens" > "New Token"

Generate API Token

1. INFO

Fill in the required information:

- Description: `git gmacario@hw2288`

- Expiration date: 2022-06-30

then click "Next".

Generate API Token

1. SCOPES

Fill in the required information:

- Scopes: Use for source code access (`source:create`, `source:read-update`)
- Factory: `test-fio-raspi4`

then click "Generate".

Generate API Token

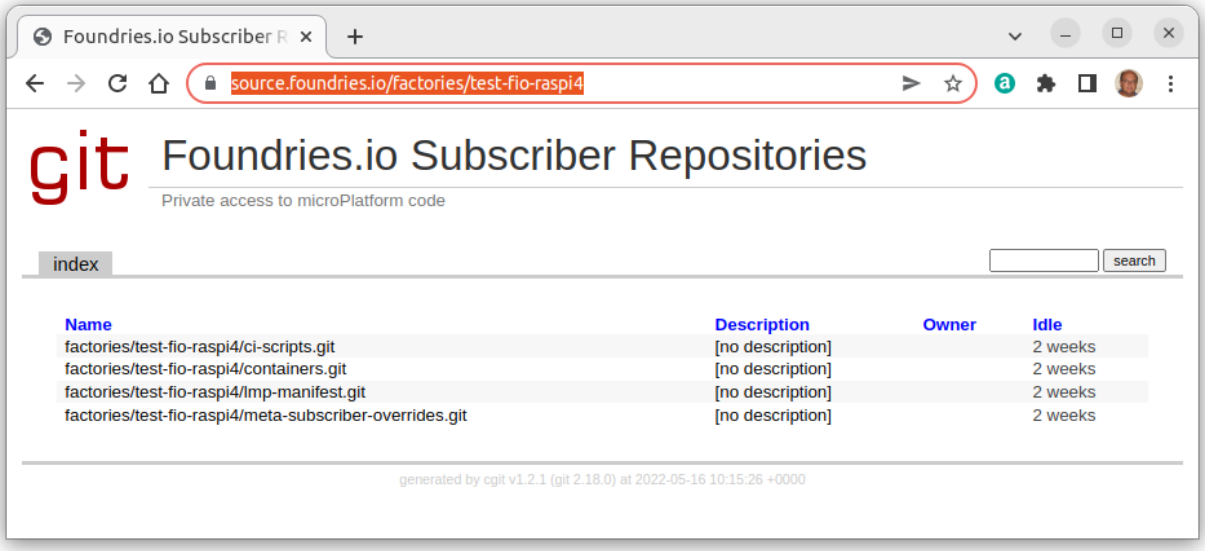
Your new API token has been created. Make sure you save the token value as you won't be able to access it again.

You may use the generated API token to authenticate and access private git source repositories, as shown in the next subsection.

Browse the factory Source Repositories from the web interface

Logged in to https://app.foundries.io rowse the Foundries.io Subscriber Repositories. In our example

https://source.foundries.io/factories/test-fio-raspi4



Configure Git

```
git config --global credential.helper store

mkdir -p ~/source.foundries.io/factories/<factory>
cd ~/source.foundries.io/factories/<factory>
```

Clone the Container Source Repository

Clone the `containers.git` repository:

```
git clone -b devel https://source.foundries.io/factories/<factory>/containers.git
```

In our example

```
gmacario@hw2228:~ $ git clone -b devel https://source.foundries.io/factories/test-fio-
↪raspi4/containers.git
Cloning into 'containers'...
Username for 'https://source.foundries.io': gmacario@gmail.com
Password for 'https://gmacario@gmail.com@source.foundries.io':
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 8 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (8/8), 1.21 KiB | 620.00 KiB/s, done.
gmacario@hw2228:~ $
```

Note: When requested by git, provide your Foundries.io as username (in our example, `gmacario@gmail.com`), then provide the generated API Token as password.

Using a similar command you may also clone the other source repositories of the factory:

```
git clone https://source.foundries.io/factories/<factory>/ci-scripts.git

git clone https://source.foundries.io/factories/<factory>/lmp-manifest.git

git clone https://source.foundries.io/factories/<factory>/meta-subscriber-overrides.git
```

4.15.6 Deploy your first Application

Reference: https://docs.foundries.io/latest/tutorials/deploying-first-app/deploying-first-app.html

Configure your device

To improve your experience during this tutorial, you will configure both `aktualizr-lite` and `fioconfig` to check every minute.

```
sudo mkdir -p /etc/sota/conf.d/
sudo bash -c 'printf "[uptane]\npolling_sec = 60" > /etc/sota/conf.d/z-01-polling.toml'

sudo bash -c 'printf "DAEMON_INTERVAL=60" > /etc/default/fioconfig'

sudo systemctl restart aktualizr-lite
sudo systemctl restart fioconfig
```

To watch the `aktualizr-lite` logs and see the updates, leave a device terminal running the command:

```
sudo journalctl --follow --unit aktualizr-lite
```

Make changes to the `devel` branch of `containers.git`

Make some changes to the `devel` branch of `containers.git` – for instance, you may rename the `shellhttpd.disabled` to `shellhttpd` to ensure that the Docker image gets built and the container run in the target.

```
cd containers
git checkout devel
git mv shellhttpd.disabled shellhttpd
git commit -sm "Enable container: shellhttpd"
git push
```

Any change on the git repository will cause the target image to be rebuilt

Debugging your device

Logged in on the development host, check the list of subscribed devices

```
fioctl login
fioctl devices list
```

Result:

```
gmacario@hw2228:~ $ fioctl devices list
NAME      TARGET                 STATUS  APPS  UP-TO-DATE  IS-PROD
----      ------                 ------  ----  ----------  -------
rpird102  raspberrypi4-64-lmp-1  OK            true        false
gmacario@hw2228:~ $
```

Read information about device `rpird102`

```
gmacario@hw2228:~ $ fioctl device show rpird102
UUID:       1f658bf4-cfb1-4546-88cf-01577b67da1b
Owner:      625ff36d02d292c0e482ecbf
Factory:    test-fio-raspi4
Production: false
Up to date: true
Target:     raspberrypi4-64-lmp-1 /␣
↪sha256(0755d289a061c1e15c6c64507bee7085418aa2588e82c309791d688f1649a243)
Ostree Hash:   0755d289a061c1e15c6c64507bee7085418aa2588e82c309791d688f1649a243
Created:    2022-05-11T17:55:47+00:00
Last Seen:  2022-05-16T09:41:00+00:00
Tag:        devel
Network Info:
   Hostname:   raspberrypi4-64
   IP:      192.168.69.128
   MAC:     e4:5f:01:35:8f:96
Hardware Info: (hidden, use --hwinfo)
Aktualizr config: (hidden, use --aktoml)
Active Config:
   Created At:    2022-05-11T17:55:48+00:00
   Applied At:    2022-05-11T17:55:48+00:00
   Change Reason: Set Wireguard pubkey from fioconfig
   Files:
      wireguard-client
       | enabled=0
       |
       | pubkey=Po/
       bjvs8w8bGgBLdPsh1owUO9r18rfLf/a+981Hjp2g=

-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEp+s6JZYPHzfTZnJlys7HHVNANCob
US1BbACwCSvoBWE/fAHE0UKyIMYwu2ZnHzqOqPVA1wo1nlb41fxPcMD9Ng==
-----END PUBLIC KEY-----
gmacario@hw2228:~ $
```

Take note of the value of field `Tag:` (in our case, `devel`), then double check on https://app.foundries.io/factories/test-fio-raspi4/targets/ which is the the latest Target with that tag.

Testing the Container

The changes introduced in the source repository of the factory will trigger a new target image build and - depending on the device configuration - after a few minutes the target will load the new image.

As part of the changed a new Docker container will be run on the target; we can verify this by logging into the target device and issuing the `docker ps` command:

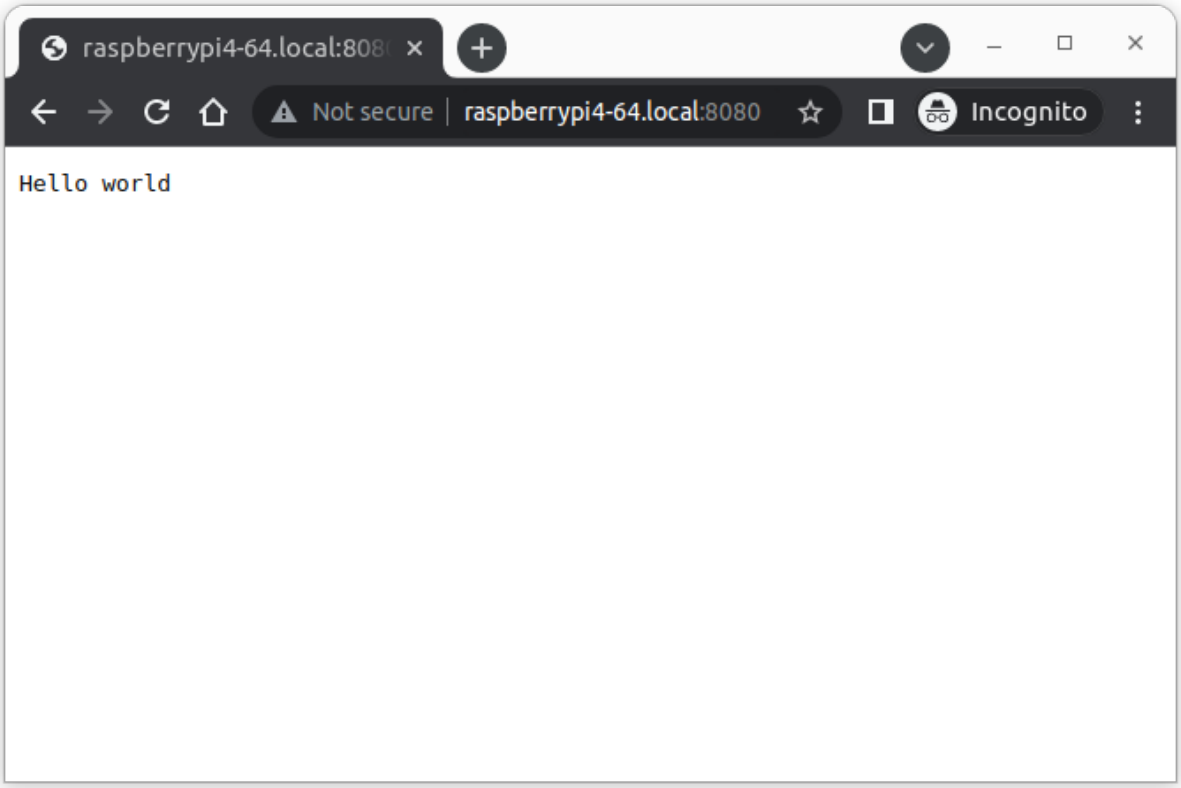
```
gmacario@hw2228:~$ ssh fio@raspberrypi4-64.local
fio@raspberrypi4-64.local's password:
Last login: Mon May 16 09:33:03 2022 from 192.168.69.67
fio@raspberrypi4-64:~$ docker ps
CONTAINER ID   IMAGE                                          COMMAND                  ␣
↪CREATED        STATUS        PORTS                                          NAMES
fe5f4f8787ee   hub.foundries.io/test-fio-raspi4/shellhttpd   "/usr/local/bin/http..."  ␣
↪14 hours ago   Up 14 hours   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   shellhttpd_
↪httpd_1
```

(continues on next page)

(continued from previous page)

```
fio@raspberrypi4-64:~$
```

Also, the `shellhttpd` container will deploy a simple HTTP server which can be verified for instance through a web browser:



Enabling Specific Applications

TODO

4.16 HOWTO Testing Arduino Portenta X8

4.16.1 Press coverage

- https://www.electronicsweekly.com/news/sponsored-content-report-highlights-demand-industry-led-iot-security-guidelines-202

4.16.2 Documentation

- Portenta X8 Getting Started (Beta)
  - Sources: https://github.com/arduino/docs-content/commits/main/content/hardware/04.pro/boards/portenta-x8/tutorials/out-of-the-box/content.md
  - docs-content/pull/159: Portenta X8: Getting Started update [PC-889]
- foundries.io
  - https://foundries.io/company/partners/arduino/
- docs.foundries.io
  - Supported Boards

4.16.3 First experiences with Portenta X8

- Connect the patch antenna to the Portenta X8 uFL connector
- Connect the Portenta X8 to a laptop using a USB-A to USB-C cable
- Wait until the upper LED turns green

Browse http://192.168.7.1/

> Welcome to the **Arduino Portenta X8**
>
> Setup your board to get started
>
> - Configure Wi-Fi
> - Register Factory name
>
> GO TO DOCUMENTATION
>
> BOARD INFO

Click on "BOARD INFO"

> **BOARD INFO**
>
> - Hostname: `PORTENTA-X8-A13B209DAB6FAD9`
> - Wi-Fi: `AROLIOT`
> - Ethernet: `USBRNDIS`
> - Factory: ->
>
> LAUNCH SHELL
>
> GO TO ARDUINO CLOUD

Expand "AROLIOT"

- Hostname: `portenta-x8-a13b209dab6fad9`
- IPv4: `172.30.48.28/24`
- MAC: `74:7A:90:CE:F3:94`

Expand "USBRNDIS"

- Hostname: `portenta-x8-a13b209dab6fad9`
- IPv4: `192.168.7.1/24`

- MAC: `32:2E:E3:05:3D:31`

Click on "LAUNCH SHELL"

```
To list installed python packages
pip3 list

To add a package
apk add <packagename>

To explore list of available packages
https://pkgs.alpinelinux.org/packages

Arduino Portenta-X8 documentation under
https://docs.arduino.cc/hardware/portenta-x8


portenta-x8:~#
```

Clone ARNEIS sources from GitHub

Logged in as `root@portenta-x8` from the shell accessible from the web interface

```
apk add git
git clone https://github.com/B-AROL-O/ARNEIS
```

Build ARNEIS documentation

Build ARNEIS documentation from sources in `docs/`:

```
cd ~/ARNEIS/docs
pip install -r requirements.txt
apk add make
make html
```

Result: TODO

4.16.4 Inspect the host OS on the Portenta X8

From a laptop connected on the same network (172.30.48.0/24)

```
gpmacario@HW2457 MINGW64 ~
$ ssh fio@172.30.48.28
fio@172.30.48.28's password:
fio@portenta-x8-a13b209dab6fad9:~$
```

Inspect the host OS

Logged in as `fio@portenta-x8-xxxx`, type the following commands to inspect the host OS

```
uname -a
cat /etc/os-release
cat /proc/cpuinfo
free -h
```

Result:

```
fio@portenta-x8-a13b209dab6fad9:~$ uname -a
Linux portenta-x8-a13b209dab6fad9 5.4.134-lmp-standard #1 SMP PREEMPT Tue Jul 20␣
↪21:09:28 UTC 2021 aarch64 aarch64 aarch64 GNU/Linux
fio@portenta-x8-a13b209dab6fad9:~$ cat /etc/os-release
ID=lmp-xwayland
NAME="Linux-microPlatform XWayland"
VERSION="3.3.2-391-83-21-g3ad61e0"
VERSION_ID=3.3.2-391-83-21-g3ad61e0
PRETTY_NAME="Linux-microPlatform XWayland 3.3.2-391-83-21-g3ad61e0"
HOME_URL="https://foundries.io/"
SUPPORT_URL="https://support.foundries.io/"
LMP_MACHINE="portenta-x8"
LMP_FACTORY="arduino"
LMP_FACTORY_TAG="devel"
fio@portenta-x8-a13b209dab6fad9:~$ cat /proc/cpuinfo
processor       : 0
BogoMIPS        : 16.00
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 1
BogoMIPS        : 16.00
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 2
BogoMIPS        : 16.00
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 3
```

(continues on next page)

(continued from previous page)

```
BogoMIPS        : 16.00
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

fio@portenta-x8-a13b209dab6fad9:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:           1.9Gi       308Mi       1.3Gi        26Mi       343Mi       1.6Gi
Swap:          1.9Gi          0B       1.9Gi
fio@portenta-x8-a13b209dab6fad9:~$
```

Inspect disk utilization

Logged in as `fio@portenta-x8-xxxx`, type the following commands:

```
df -h
```

Result:

```
fio@portenta-x8-a13b209dab6fad9:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           976M   26M  950M   3% /run
devtmpfs        651M     0  651M   0% /dev
/dev/mmcblk2p2   14G  2.1G   11G  16% /sysroot
tmpfs           976M     0  976M   0% /dev/shm
tmpfs           4.0M     0  4.0M   0% /sys/fs/cgroup
tmpfs           976M     0  976M   0% /tmp
tmpfs           976M   12K  976M   1% /var/volatile
/dev/mmcblk2p1   84M   28K   83M   1% /var/rootdirs/mnt/boot
tmpfs           196M     0  196M   0% /run/user/63
overlay          14G  2.1G   11G  16% /var/lib/docker/overlay2/
↪71bfdab5583038b5068957a0ac6808c843636759bd1e8e23610686a294748ec5/merged
overlay          14G  2.1G   11G  16% /var/lib/docker/overlay2/
↪ebf99a4712b01bd0d3e6a27f6c4f7b1dda3c1018f4abe2598b083240d29b9875/merged
fio@portenta-x8-a13b209dab6fad9:~$
```

Inspect the network interfaces

Logged in as `fio@portenta-x8-xxxx`, type the following command:

```
ip addr
```

Result:

```
fio@portenta-x8-a13b209dab6fad9:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen␣
↪1000
```

(continues on next page)

(continued from previous page)

```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
3: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default␣
↪qlen 1000
    link/ether 6e:4f:24:e2:aa:4b brd ff:ff:ff:ff:ff:ff
4: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group␣
↪default qlen 1000
    link/ether 74:7a:90:ce:f3:94 brd ff:ff:ff:ff:ff:ff
    inet 172.30.48.28/24 brd 172.30.48.255 scope global dynamic noprefixroute wlan0
       valid_lft 598757sec preferred_lft 598757sec
    inet6 fe80::d422:768c:a623:311d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
5: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
6: can1: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
7: usb0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group␣
↪default qlen 1000
    link/ether 46:f2:7f:29:29:18 brd ff:ff:ff:ff:ff:ff
    inet 192.168.7.1/24 brd 192.168.7.255 scope global noprefixroute usb0
       valid_lft forever preferred_lft forever
    inet6 fe80::67f4:afe1:2c7d:abe8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
8: usb1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group␣
↪default qlen 1000
    link/ether 32:f2:98:46:1b:c2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.8.1/24 brd 192.168.8.255 scope global noprefixroute usb1
       valid_lft forever preferred_lft forever
    inet6 fe80::f0db:a7c1:d850:58d8/64 scope link tentative noprefixroute
       valid_lft forever preferred_lft forever
9: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group␣
↪default
    link/ether 02:42:b0:7b:86:28 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
10: br-9a0e17781cd2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP␣
↪group default
    link/ether 02:42:59:99:dc:7b brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-9a0e17781cd2
       valid_lft forever preferred_lft forever
    inet6 fe80::42:59ff:fe99:dc7b/64 scope link
       valid_lft forever preferred_lft forever
12: veth7267f54@if11: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-
↪9a0e17781cd2 state UP group default
    link/ether 9a:a2:8f:91:28:58 brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::98a2:8fff:fe91:2858/64 scope link
       valid_lft forever preferred_lft forever
```

(continues on next page)

(continued from previous page)

```
14: veth68241db@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-
↪9a0e17781cd2 state UP group default
    link/ether 5a:a7:6c:72:19:e6 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::58a7:6cff:fe72:19e6/64 scope link
       valid_lft forever preferred_lft forever
fio@portenta-x8-a13b209dab6fad9:~$
```

Inspect Docker configuration

Logged in as `fio@portenta-x8-xxxx`, type the following commands:

```
docker version
docker info
docker images
docker ps -a
```

Result:

```
fio@portenta-x8-a13b209dab6fad9:~$ docker version
Client:
 Version:           20.10.7
 API version:       1.41
 Go version:        go1.16.7
 Git commit:        e9b8231d6a
 Built:             Tue Feb 15 18:02:50 2022
 OS/Arch:           linux/arm64
 Context:           default
 Experimental:      true

Server:
 Engine:
  Version:          20.10.7
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.16.7
  Git commit:       013d6655bb0f4c86bcd9d48372ef67afd0ded65e
  Built:            Tue Feb 15 18:01:38 2022
  OS/Arch:          linux/arm64
  Experimental:     false
 containerd:
  Version:          v1.5.4-12-g1c13c54ca.m
  GitCommit:        1c13c54cae4f53510a7a45ae3e4af49030a76193.m
 runc:
  Version:          1.0.0-rc95+dev
  GitCommit:        bfcbc947d5d11327f2680047e2e6e94f4ee93d2a-dirty
 docker-init:
  Version:          0.19.0
  GitCommit:        b9f42a0-dirty
fio@portenta-x8-a13b209dab6fad9:~$ docker info
Client:
 Context:    default
 Debug Mode: false
```

(continues on next page)

(continued from previous page)

```
Server:
 Containers: 2
  Running: 2
  Paused: 0
  Stopped: 0
 Images: 2
 Server Version: 20.10.7
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: journald
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc io.containerd.runc.v2 io.containerd.runtime.v1.linux
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 1c13c54cae4f53510a7a45ae3e4af49030a76193.m
 runc version: bfcbc947d5d11327f2680047e2e6e94f4ee93d2a-dirty
 init version: b9f42a0-dirty (expected: de40ad007797e)
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 5.4.134-lmp-standard
 Operating System: Linux-microPlatform XWayland 3.3.2-391-83-21-g3ad61e0
 OSType: linux
 Architecture: aarch64
 CPUs: 4
 Total Memory: 1.905GiB
 Name: portenta-x8-a13b209dab6fad9
 ID: 6YE7:I5FE:OINP:4YR2:ZMSQ:CC24:GZVF:QRDX:PB3V:KDQO:XMDL:OQFJ
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false

fio@portenta-x8-a13b209dab6fad9:~$ docker images
REPOSITORY                                      TAG       IMAGE ID       CREATED      ␣
↪SIZE
hub.foundries.io/arduino/python-devel-arduino   <none>    60de54fcb301   2 weeks ago  ␣
↪268MB
```

(continues on next page)

(continued from previous page)

```
hub.foundries.io/arduino/go-webapp-arduino      <none>    495e2e950a62   2 weeks ago  ␣
↪127MB
fio@portenta-x8-a13b209dab6fad9:~$ docker ps -a
CONTAINER ID   IMAGE                                          COMMAND            ␣
↪CREATED         STATUS      PORTS                                   NAMES
49c117163405   hub.foundries.io/arduino/go-webapp-arduino      "/entrypoint.sh"   13␣
↪months ago   Up 5 days   0.0.0.0:80->1323/tcp, :::80->1323/tcp   x8-setup
782933c5d575   hub.foundries.io/arduino/python-devel-arduino   "/entrypoint.sh"   13␣
↪months ago   Up 5 days                                           x8-devel
fio@portenta-x8-a13b209dab6fad9:~$
```

Verify Bluetooth subsystem of the Portenta X8

Logged in as `fio@portenta-x8-xxxx`, type the following commands:

```
hciconfig
```

Result:

```
fio@portenta-x8-a13b209dab6fad9:~$ hciconfig
hci0:   Type: Primary  Bus: UART
        BD Address: 43:43:A1:12:1F:AC  ACL MTU: 1021:8  SCO MTU: 64:1
        DOWN
        RX bytes:2045 acl:0 sco:0 events:190 errors:0
        TX bytes:37360 acl:0 sco:0 commands:190 errors:0

fio@portenta-x8-a13b209dab6fad9:~$
```

Bring up the Bluetooth interface, then scan nearby BLE devices

```
sudo hciconfig hci0 up
hcitool dev
sudo hcitool lescan
```

Result:

```
fio@portenta-x8-a13b209dab6fad9:~$ sudo hciconfig hci0 up
fio@portenta-x8-a13b209dab6fad9:~$ hcitool dev
Devices:
        hci0    43:43:A1:12:1F:AC
fio@portenta-x8-a13b209dab6fad9:~$ sudo hcitool lescan
LE Scan ...
2E:75:1E:87:EB:CF (unknown)
7C:64:56:88:EF:A7 (unknown)
2F:ED:4A:2E:A5:B5 (unknown)
72:D4:7E:59:0D:F8 (unknown)
65:4E:A0:F6:BE:B4 (unknown)
65:4E:A0:F6:BE:B4 (unknown)
1F:F2:44:20:1A:28 (unknown)
2C:9B:97:13:4A:9B (unknown)
28:C4:3F:11:AB:CE (unknown)
02:93:45:6B:9E:AB (unknown)
```

(continues on next page)

(continued from previous page)

```
CD:1B:AC:76:93:BF (unknown)
6E:E5:2C:AB:EB:DD (unknown)
6E:E5:2C:AB:EB:DD (unknown)
1B:9E:4C:3A:9B:B1 (unknown)
55:37:EF:1F:3B:86 (unknown)
7C:81:F0:23:19:98 (unknown)
07:DD:3B:78:21:46 (unknown)
62:27:88:AC:C0:F0 (unknown)
62:A4:EB:0B:D6:AE (unknown)
5C:1B:DB:65:60:57 (unknown)
7C:49:DF:24:CE:3B (unknown)
37:9F:05:BE:F8:E9 (unknown)
F5:83:BC:3A:7B:93 (unknown)
62:A4:EB:0B:D6:AE (unknown)
62:27:88:AC:C0:F0 (unknown)
44:E8:BB:5E:76:AB (unknown)
D9:04:87:92:BE:06 (unknown)
D9:04:87:92:BE:06 BT510
44:E8:BB:5E:76:AB (unknown)
E3:A2:5D:E5:24:9C (unknown)
^Cfio@portenta-x8-a13b209dab6fad9:~$
```

4.17 HOWTO Create a VM on Azure for the ARNEIS project

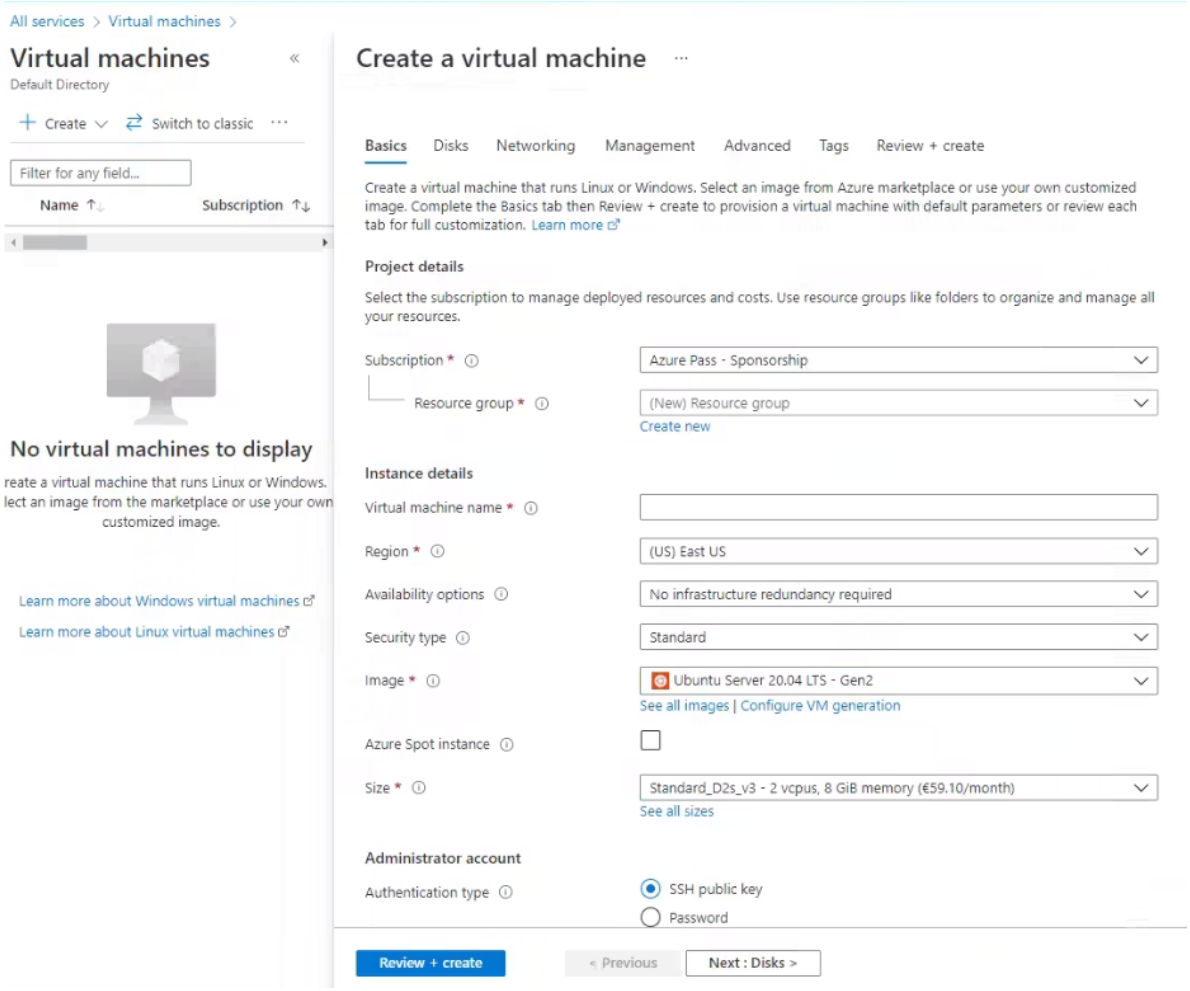
4.17.1 Introduction

This document details how to create a Virtual Machine on Microsoft Azure cloud that will be used for the ARNEIS project.

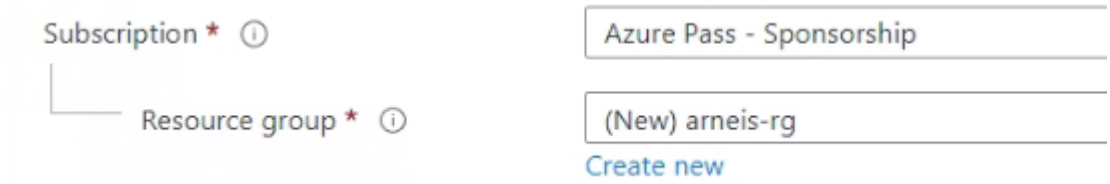
4.17.2 Step-by-step instructions

Create a VM using the web interface

- Login to http://portal.azure.com using your Azure credentials
- Click on **Virtual Machines**
- Click on **+ Create**
- You will have 2 options: **Virtual Machine** or **Start with a preset configuration**
- We started with *Virtual Machine*



- Configure options in section **Project details**
  - Leave **Subscription** to `Azure Pass - Sponsorship` unless you want to charge expenses to someone else
  - Set **Resource group** to `arneis-rg` (short for "ARNEIS Resource Group")

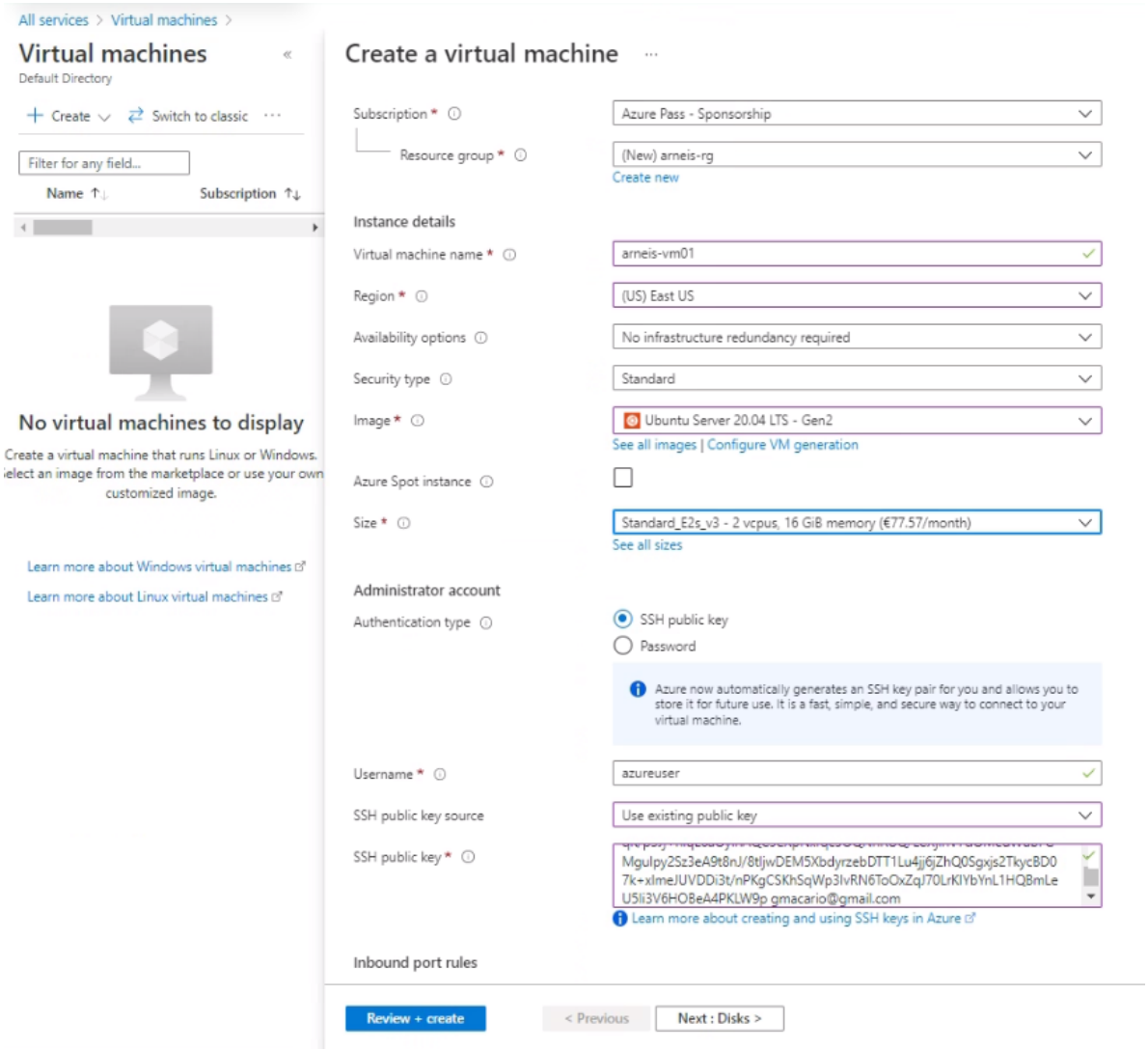


- Configure options in section **Instance details**
  - Choose the **Virtual machine name** as `arneis-vm01`
  - Select **Region** (you may leave default "(US) East US" or choose a region which is geographically closer to your clients)
  - In **Availability options** choose "No infrastructure redundancy required"
  - Leave **Security type** to "Standard"
  - In field **Image** select "Ubuntu Server 20.04 LTS - Gen2"

  - In field **Size** choose "Standard_E2s_v3 - 2 vcpus, 16 GiB memory (EUR 77.57/month)" unless you have different requirements

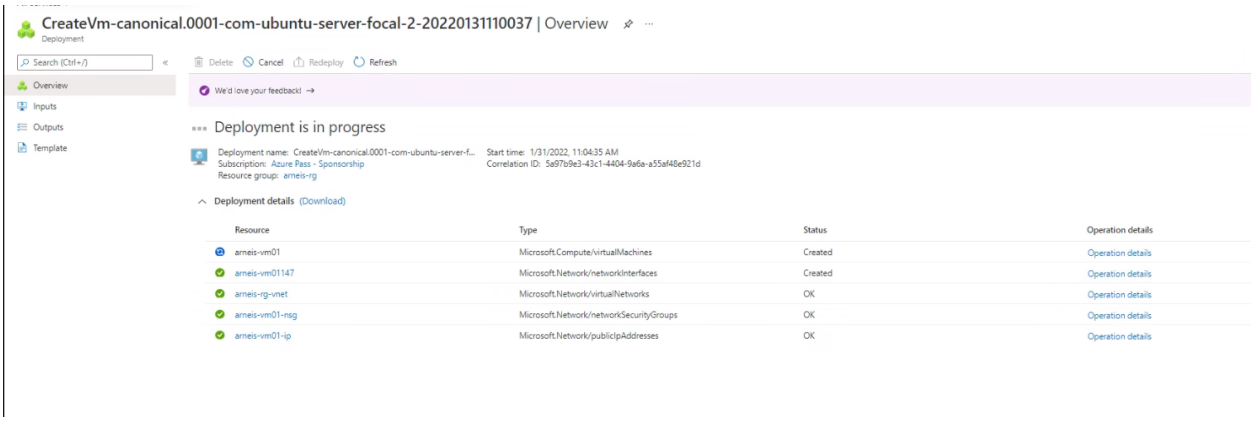
- Add your public SSH key if you have one, otherwise let Azure automatically generate one for you

  Result:



- Review and accept the default values in the following pages: **Disks**, **Networking**, **Management** and **Advanced**
- Click **Create** to start creating the VM

Now you just have to wait for the deployment to complete:



Test the VM

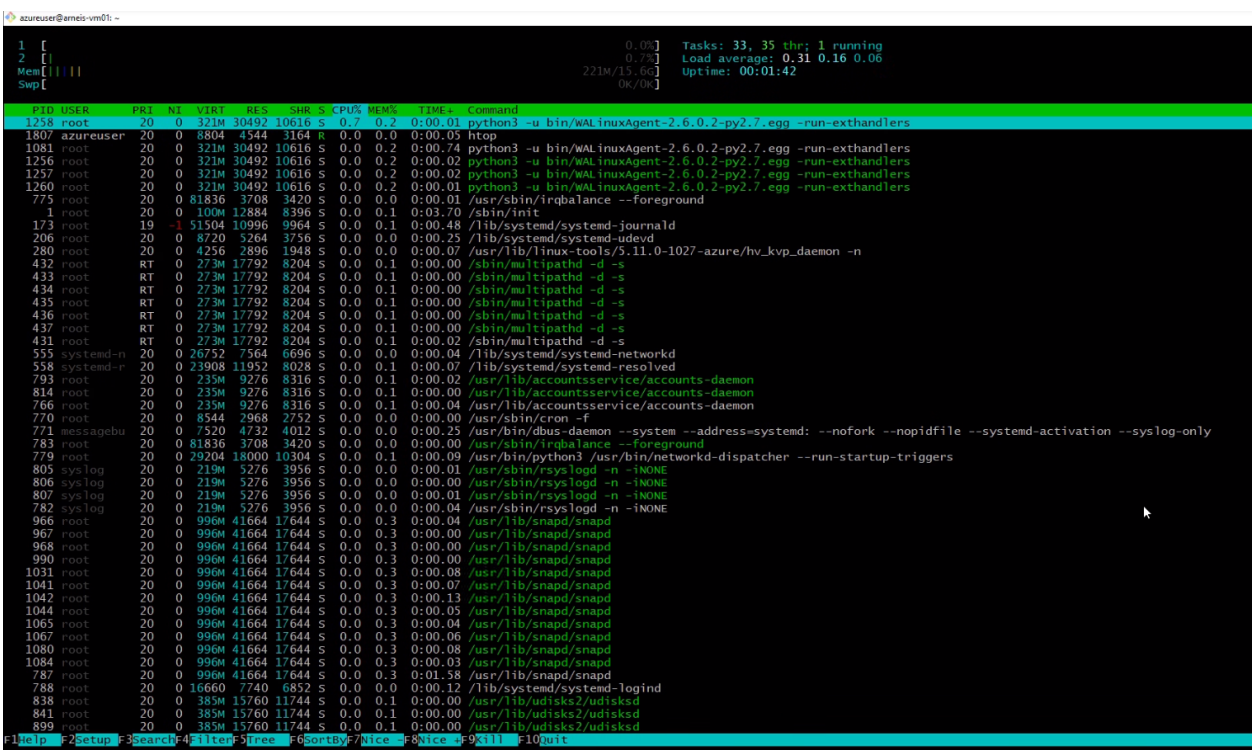
After the machine is deployed and started, the IP address assigned to the VM will be displayed on the VM overview page in the Azure portal.

With this information you should be able to remotely log in to the VM via SSH:

```
ssh -i <private-key-path> azureuser@<vm-public-ip-address>
```

Then you may run some commands on the shell, for instance:

```
htop
```



(Recommended) Create a public DNS entry

If you have administrative rights to a DNS zone you may choose to access your VM using a symbolic name rather than an IP address.

If so, access your DNS administrative page (in my case, https://register.it/) and create an A record to map the name to the IP address assigned to your VM.

In my case

```
A arneis-vm01 20.124.132.35
```

Wait until the DNS zone is propagated, then verify that the device can be accessed by another host (in our case, our laptop) using the assigned name rather than its IP address:

```
gmaca@alpha MINGW64 ~
$ ssh -i ~/.ssh/gmacario-gmail-com azureuser@arneis-vm01.gmacario.it
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1027-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Jan 31 17:52:17 UTC 2022

  System load:  0.0               Processes:             155
  Usage of /:   9.8% of 28.90GB   Users logged in:       1
  Memory usage: 7%                IPv4 address for cni0: 10.42.0.1
  Swap usage:   0%                IPv4 address for eth0: 10.0.0.4

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

0 updates can be applied immediately.


*** System restart required ***
Last login: Mon Jan 31 17:51:47 2022 from 93.43.242.87
azureuser@arneis-vm01:~$
```

Configure firewall

Logged into https://portal.azure.com/, select Virtual Machine "arneis-vm01", then in menu "Settings" click **Networking**.

In tab "Inbound port rules", click **Add inbound port rule**, then specify the following:

- **Source**: Any
- **Source port ranges**: *
- **Destination**: Any
- **Service**: HTTP

- **Destination port ranges**: 80
- **Protocol**: TCP
- **Action**: Allow
- **Priority**: (leave default)

then click **Add**.

Repeat for the following ports:

- 443/TCP (HTTPS)
- 6443/TCP (Kubernetes API Server)

You should obtain the following

Inbound port rules | Outbound port rules | Application security groups | Load balancing

Network security group arneis-vm01-nsg (attached to network interface: arneis-vm01147)
Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
310	Port_80	80	TCP	Any	Any	Allow
320	Port_443	443	TCP	Any	Any	Allow
330	Port_6443	6443	Any	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

4.18 HOWTO Install a K3s cluster for the ARNEIS project

4.18.1 Introduction

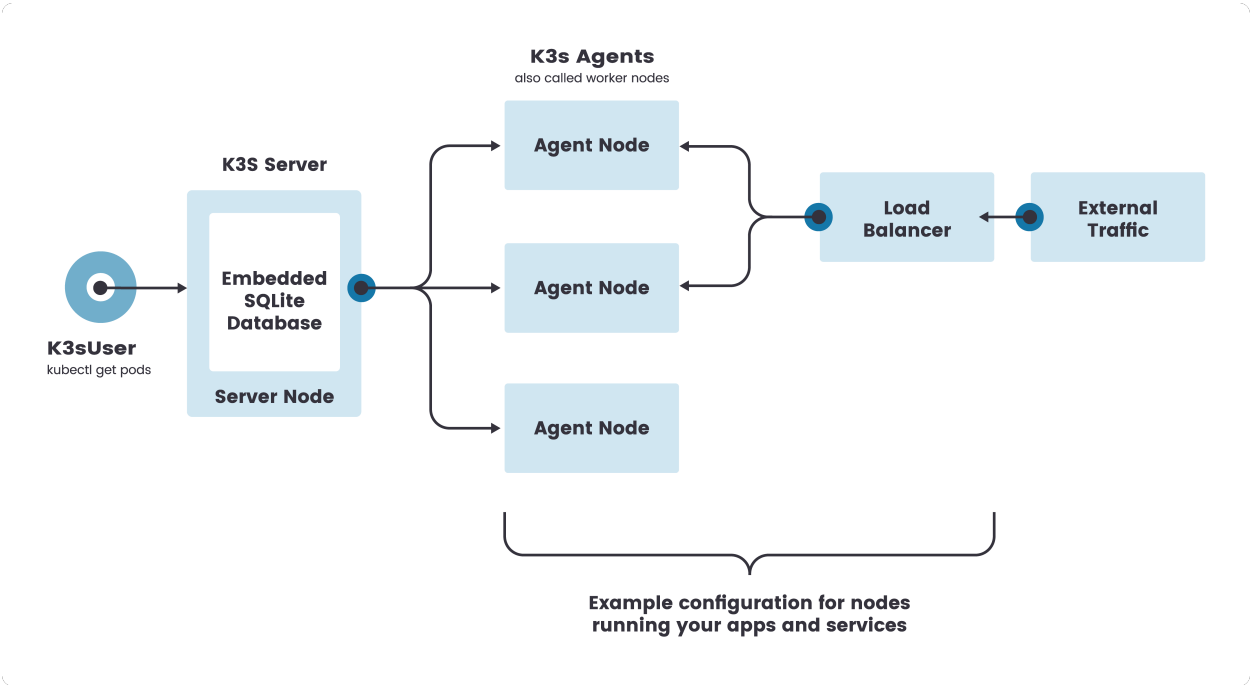
This document explains how to install a K3s cluster to be used for the ARNEIS project.

K3s (or "Lightweight Kubernetes") is a simplified installation of the Kubernetes distribution built for IoT and Edge computing.

K3s is an Open Source project started and maintained by Rancher.com.

Architecture

The following diagram shows a possible deployment of the K3s architecture:



(Image credits: https://rancher.com/docs/k3s/latest/en/architecture/)

The main host will act as both a K3s Server (previously called "master" in Kubernetes literature) and Agent (worker) Node. This is the smallest possible deployment of a K3s cluster. Additionally, other machines - either physical or virtual - may be added to the topology to act as Agent Nodes, thus adding redundancy and increasing the computation and storage capacity of the cluster.

References

- Rancher docs
- K3S Installation - from rancher.com/docs

4.18.2 Prerequisites

Host acting as K3s Server

- Administrative login to a host (either physical or virtual) with requirements detailed at https://rancher.com/docs/k3s/latest/en/installation/installation-requirements/ - more specifically:
  - CPU: min 2 cores
  - RAM: min 16 GiB
  - Disk: min 8 GiB SSD
  - OS: Ubuntu server 20.04 LTS
  - Fast internet connection (for updating OS and installing software)
  - Firewall configured to accept incoming ports:
    - 22/tcp (SSH)
    - 80/tcp (HTTP)

    * 443/tcp (HTTPS)
    * 6443/tcp,udp (Kubernetes API server)
  - Tested on `arneis-vm01` (Virtual Machine on Azure Cloud - See *documentation*)

Host(s) acting as Agent Node(s)

- Administrative login to a host (either physical or virtual) with the requirements detailed at https://rancher.com/docs/k3s/latest/en/installation/installation-requirements/
  - Tested on `arneis-vm02` (Virtual Machine on Azure Cloud - See *documentation*)
  - Also tested on `rpi4gm35` (Raspberry Pi 4B - See *documentation*)

4.18.3 Deploy the first node of the cluster

A minimal K3s cluster is made of one host (either physical or virtual) which acts both as Server and Agent Node.

In our example, we will deploy the main node of the K3s cluster on `root@arneis-vm01` (Ubuntu 20.04.4 LTS):

```
root@arneis-vm01:~# cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04.4 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.4 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
root@arneis-vm01:~#
```

Install K3s on the main node

Logged in as `root@arneis-vm01`, install k3s

```
curl -sfL https://get.k3s.io | \
    INSTALL_K3S_EXEC="--node-external-ip $(curl ifconfig.co)" sh -
```

Result:

```
root@arneis-vm01:~# curl -sfL https://get.k3s.io | \
>     INSTALL_K3S_EXEC="--node-external-ip $(curl ifconfig.co)" sh -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    14  100    14    0     0    145      0 --:--:-- --:--:-- --:--:--   145
[INFO]  Finding release for channel stable
[INFO]  Using v1.22.7+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/
↪sha256sum-amd64.txt
```

(continues on next page)

(continued from previous page)

```
[INFO]  Skipping binary downloaded, installed k3s matches hash
[INFO]  Skipping installation of SELinux RPM
[INFO]  Skipping /usr/local/bin/kubectl symlink to k3s, already exists
[INFO]  Skipping /usr/local/bin/crictl symlink to k3s, already exists
[INFO]  Skipping /usr/local/bin/ctr symlink to k3s, already exists
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s.service
[INFO]  systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/
↪system/k3s.service.
[INFO]  systemd: Starting k3s
root@arneis-vm01:~#
```

NOTE: The `--node-external-ip addr` option is required for some hosts such as Azure VMs. This option will make sure that the K3s API Server will advertise its public IP address.

The page at https://rancher.com/docs/k3s/latest/en/installation/install-options/server-config/ provides more detail about the `--node-external-ip value` option.

Verify the installation of the main node

The installation of the cluster might take a few minutes.

Check the nodes in the cluster

Logged in as `root@<server-node>`, type the following command to list all the nodes which have joined the cluster:

```
kubectl get nodes
```

Since the cluster has just been created, only one node should be displayed as shown below:

```
root@arneis-vm01:~# kubectl get nodes
NAME          STATUS   ROLES                  AGE   VERSION
arneis-vm01   Ready    control-plane,master   48s   v1.22.7+k3s1
root@arneis-vm01:~#
```

If the result is the same, proceed with the next check.

Check the K3s core services

Logged in as `root@<server-node>`, type the following command to verify that all the K3S core services are up and running:

```
kubectl get all --all-namespaces
```

Result

```
root@arneis-vm01:~# kubectl get all --all-namespaces
NAMESPACE     NAME                                          READY   STATUS      RESTARTS␣
↪  AGE
kube-system   pod/local-path-provisioner-84bb864455-zlqqm   1/1     Running     0       ␣
↪  4h29m
kube-system   pod/helm-install-traefik-crd--1-kldlb         0/1     Completed   0       ␣
↪  4h29m
kube-system   pod/helm-install-traefik--1-ndkrh             0/1     Completed   2       ␣
↪  4h29m
kube-system   pod/svclb-traefik-b9tvv                       2/2     Running     0       ␣
↪  4h29m
default       pod/busybox-sleep                             1/1     Running     0       ␣
↪  4h22m
kube-system   pod/metrics-server-ff9dbcb6c-2wh9z            1/1     Running     0       ␣
↪  4h29m
kube-system   pod/coredns-96cc4f57d-dsv74                   1/1     Running     0       ␣
↪  4h29m
kube-system   pod/traefik-56c4b88c4b-hxn24                  1/1     Running     0       ␣
↪  4h29m
kube-system   pod/svclb-traefik-zf9lj                       2/2     Running     0       ␣
↪  61s

NAMESPACE     NAME                      TYPE           CLUSTER-IP      EXTERNAL-IP    ␣
↪PORT(S)                      AGE
default       service/kubernetes        ClusterIP      10.43.0.1       <none>         ␣
↪443/TCP                      4h30m
kube-system   service/kube-dns          ClusterIP      10.43.0.10      <none>          53/
↪UDP,53/TCP,9153/TCP       4h30m
kube-system   service/metrics-server    ClusterIP      10.43.137.232   <none>         ␣
↪443/TCP                      4h30m
kube-system   service/traefik           LoadBalancer   10.43.115.89    20.124.132.35  ␣
↪80:30032/TCP,443:32150/TCP   4h29m

NAMESPACE     NAME                          DESIRED   CURRENT   READY   UP-TO-DATE  ␣
↪AVAILABLE   NODE SELECTOR   AGE
kube-system   daemonset.apps/svclb-traefik   2         2         2       2            2 ␣
↪          <none>          4h29m

NAMESPACE     NAME                                     READY   UP-TO-DATE   AVAILABLE  ␣
↪AGE
kube-system   deployment.apps/local-path-provisioner   1/1     1            1          ␣
↪4h30m
kube-system   deployment.apps/coredns                  1/1     1            1          ␣
↪4h30m
kube-system   deployment.apps/metrics-server           1/1     1            1          ␣
↪4h30m
kube-system   deployment.apps/traefik                  1/1     1            1          ␣
↪4h29m

NAMESPACE     NAME                                              DESIRED   CURRENT  ␣
↪READY   AGE
kube-system   replicaset.apps/local-path-provisioner-84bb864455   1         1         1 ␣
↪     4h29m
```

(continues on next page)

(continued from previous page)

```
kube-system   replicaset.apps/coredns-96cc4f57d                    1         1         1 ␣
↪     4h29m
kube-system   replicaset.apps/metrics-server-ff9dbcb6c             1         1         1 ␣
↪     4h29m
kube-system   replicaset.apps/traefik-56c4b88c4b                   1         1         1 ␣
↪     4h29m

NAMESPACE     NAME                                  COMPLETIONS   DURATION   AGE
kube-system   job.batch/helm-install-traefik-crd    1/1           24s        4h30m
kube-system   job.batch/helm-install-traefik        1/1           41s        4h30m
root@arneis-vm01:~#
```

Make sure that all the pods in `NAMESPACE=kube-system` have `STATUS=Running`, with the exception of the pods whose name begins with `helm-install-`. Those are one-time pods used for installing other Kubernetes resources; in this case, make sure they have `STATUS=Completed`.

Please also verify that LoadBalancer `service/traefik` has a publicly accessible `EXTERNAL-IP` address, and not a private (non-routable) IP Address. Refer to the explanation about `--node-external-ip addr` option earlier in this document.

Check the TLS certificates installed on the K3s server

Logged in as `root@<k3s-server>`, type the following command:

```
ls -la /var/lib/rancher/k3s/server/tls/
```

Result:

```
root@arneis-vm01:~# ls -la /var/lib/rancher/k3s/server/tls/
total 128
drwx------ 4 root root 4096 Mar  9 10:49 .
drwx------ 7 root root 4096 Mar  9 15:16 ..
-rw-r--r-- 1 root root 1173 Mar  9 10:49 client-admin.crt
-rw------- 1 root root  227 Mar  9 10:49 client-admin.key
-rw-r--r-- 1 root root 1178 Mar  9 10:49 client-auth-proxy.crt
-rw------- 1 root root  227 Mar  9 10:49 client-auth-proxy.key
-rw-r--r-- 1 root root  570 Mar  9 10:49 client-ca.crt
-rw------- 1 root root  227 Mar  9 10:49 client-ca.key
-rw-r--r-- 1 root root 1165 Mar  9 10:49 client-controller.crt
-rw------- 1 root root  227 Mar  9 10:49 client-controller.key
-rw-r--r-- 1 root root 1161 Mar  9 10:49 client-k3s-cloud-controller.crt
-rw------- 1 root root  227 Mar  9 10:49 client-k3s-cloud-controller.key
-rw-r--r-- 1 root root 1153 Mar  9 10:49 client-k3s-controller.crt
-rw------- 1 root root  227 Mar  9 10:49 client-k3s-controller.key
-rw-r--r-- 1 root root 1144 Mar  9 10:49 client-kube-apiserver.crt
-rw------- 1 root root  227 Mar  9 10:49 client-kube-apiserver.key
-rw-r--r-- 1 root root 1144 Mar  9 10:49 client-kube-proxy.crt
-rw------- 1 root root  227 Mar  9 10:49 client-kube-proxy.key
-rw------- 1 root root  227 Mar  9 10:49 client-kubelet.key
-rw-r--r-- 1 root root 1153 Mar  9 10:49 client-scheduler.crt
-rw------- 1 root root  227 Mar  9 10:49 client-scheduler.key
-rw-r--r-- 1 root root 3213 Mar  9 13:57 dynamic-cert.json
```

(continues on next page)

(continued from previous page)

```
drwxr-xr-x 2 root root 4096 Mar  9 10:49 etcd
-rw-r--r-- 1 root root  591 Mar  9 10:49 request-header-ca.crt
-rw------- 1 root root  227 Mar  9 10:49 request-header-ca.key
-rw-r--r-- 1 root root  570 Mar  9 10:49 server-ca.crt
-rw------- 1 root root  227 Mar  9 10:49 server-ca.key
-rw------- 1 root root 1679 Mar  9 10:49 service.key
-rw-r--r-- 1 root root 1356 Mar  9 15:16 serving-kube-apiserver.crt
-rw------- 1 root root  227 Mar  9 15:16 serving-kube-apiserver.key
-rw------- 1 root root  227 Mar  9 10:49 serving-kubelet.key
drwx------ 2 root root 4096 Mar  9 10:49 temporary-certs
root@arneis-vm01:~#
```

Run a sample Pod on the cluster

The commands shown in this section have the purpose to verify that the cluster is ready to execute a simple workload.

Logged in as `root@arneis-vm01`, create a file `test.yaml` with the following contents:

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: busybox-sleep
spec:
  containers:
  - name: busybox
    image: busybox
    args:
    - sleep
    - "1000000"
```

then run `kubectl apply -f test.yaml`:

```
kubectl apply -f test.yaml
```

Result:

```
root@arneis-vm01:~# kubectl apply -f test.yaml
pod/busybox-sleep created
root@arneis-vm01:~#
```

Now check that the new Pod is up and running

```
kubectl get pods
```

Result:

```
root@arneis-vm01:~# kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
busybox-sleep   1/1     Running   0          59s
root@arneis-vm01:~#
```

4.18.4 Add a new agent node to the cluster

After we have verified that K3s server is up and running we are ready to add new Agent Node(s) to the cluster.

In our example, we will add host `arneis-vm02` (Ubuntu 20.04.4 LTS) as a K3s Agent Node:

```
root@arneis-vm02:~# cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04.4 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.4 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
root@arneis-vm02:~#
```

Verify accessibility of the K3s API server

Logged in as `root@<agent-node>` try to access URL `https://<main-node>:6443/` to make sure that the network connectivity to the cluster API server can properly be established and there are no blocking firewalls in between. You can verify the connectivity using several methods, such as:

1. Using curl
2. Using a web browser

Example 1: Using curl

Verify HTTPS connectivity to the K3s API server (by default, on port `443`) with the following command:

```
curl https://<k3s-api-server-host>:<k3s-api-server-port>/
```

Example:

```
root@arneis-vm02:~# curl https://arneis-vm01.gmacario.it:6443/
curl: (60) SSL certificate problem: unable to get local issuer certificate
More details here: https://curl.haxx.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
root@arneis-vm02:~#
```

NOTE: The error is expected since the K3s API server uses custom certificates available under `/var/lib/rancher/k3s/server/tls` which are used to establish trust between clients (agents) and server.

As a workaround, try adding the `-k` option to `curl` (you could also add the `-v` option to increase verbosity to `curl`):

```
root@arneis-vm02:~# curl -v -k https://arneis-vm01.gmacario.it:6443/
*   Trying 20.124.132.35:6443...
* TCP_NODELAY set
* Connected to arneis-vm01.gmacario.it (20.124.132.35) port 6443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/certs/ca-certificates.crt
  CApath: /etc/ssl/certs
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Request CERT (13):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Certificate (11):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384
* ALPN, server did not agree to a protocol
* Server certificate:
*  subject: O=k3s; CN=k3s
*  start date: Mar  9 10:49:18 2022 GMT
*  expire date: Mar  9 13:57:56 2023 GMT
*  issuer: CN=k3s-server-ca@1646822958
*  SSL certificate verify result: unable to get local issuer certificate (20),␣
↪continuing anyway.
> GET / HTTP/1.1
> Host: arneis-vm01.gmacario.it:6443
> User-Agent: curl/7.68.0
> Accept: */*
>
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* Mark bundle as not supporting multiuse
< HTTP/1.1 401 Unauthorized
< Audit-Id: 49df68d0-efe8-4e1b-91ca-65070259bc52
< Cache-Control: no-cache, private
< Content-Type: application/json
< Date: Wed, 09 Mar 2022 15:48:42 GMT
< Content-Length: 165
<
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {

  },
  "status": "Failure",
  "message": "Unauthorized",
  "reason": "Unauthorized",
  "code": 401
* Connection #0 to host arneis-vm01.gmacario.it left intact
```

(continues on next page)

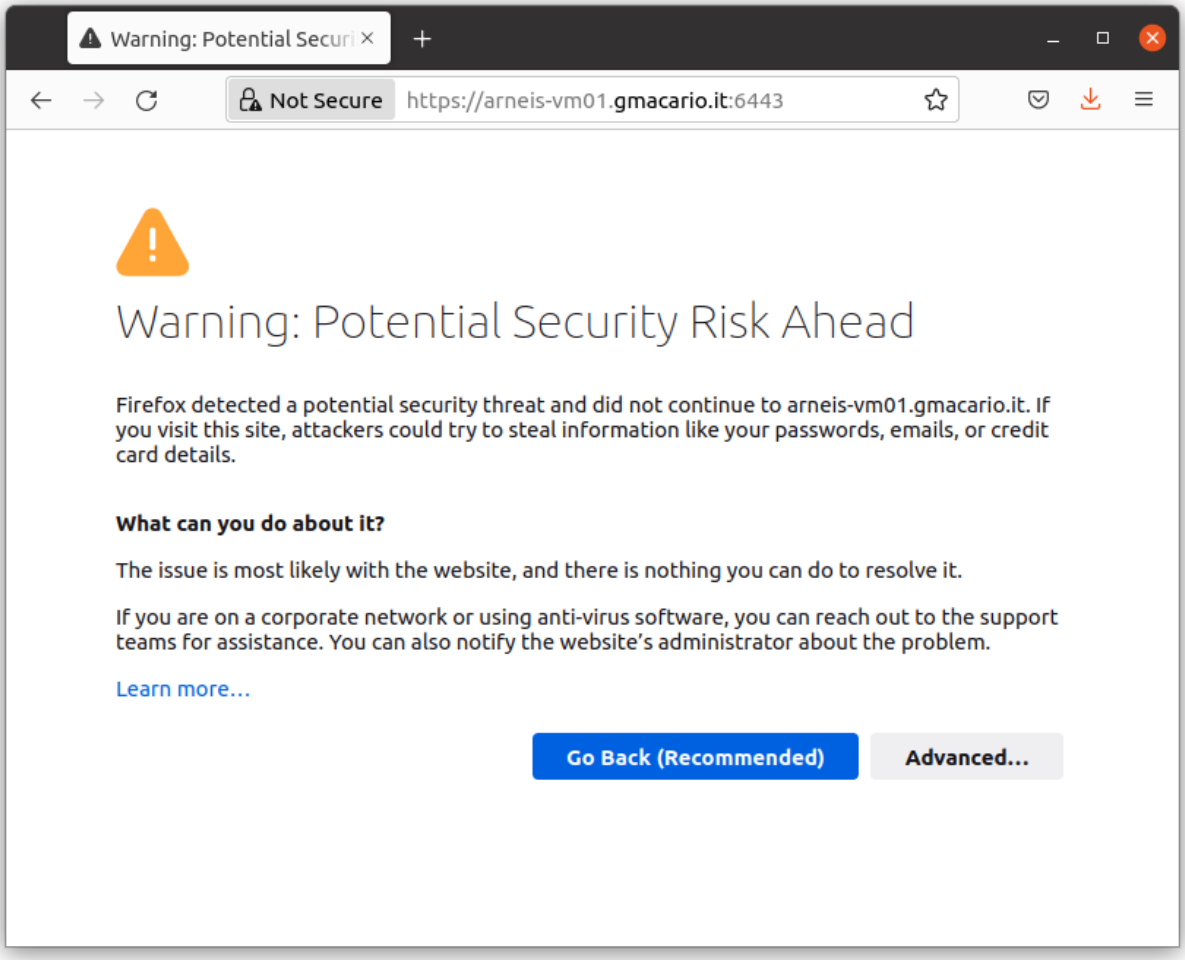
(continued from previous page)

```
}root@arneis-vm02:~#
```

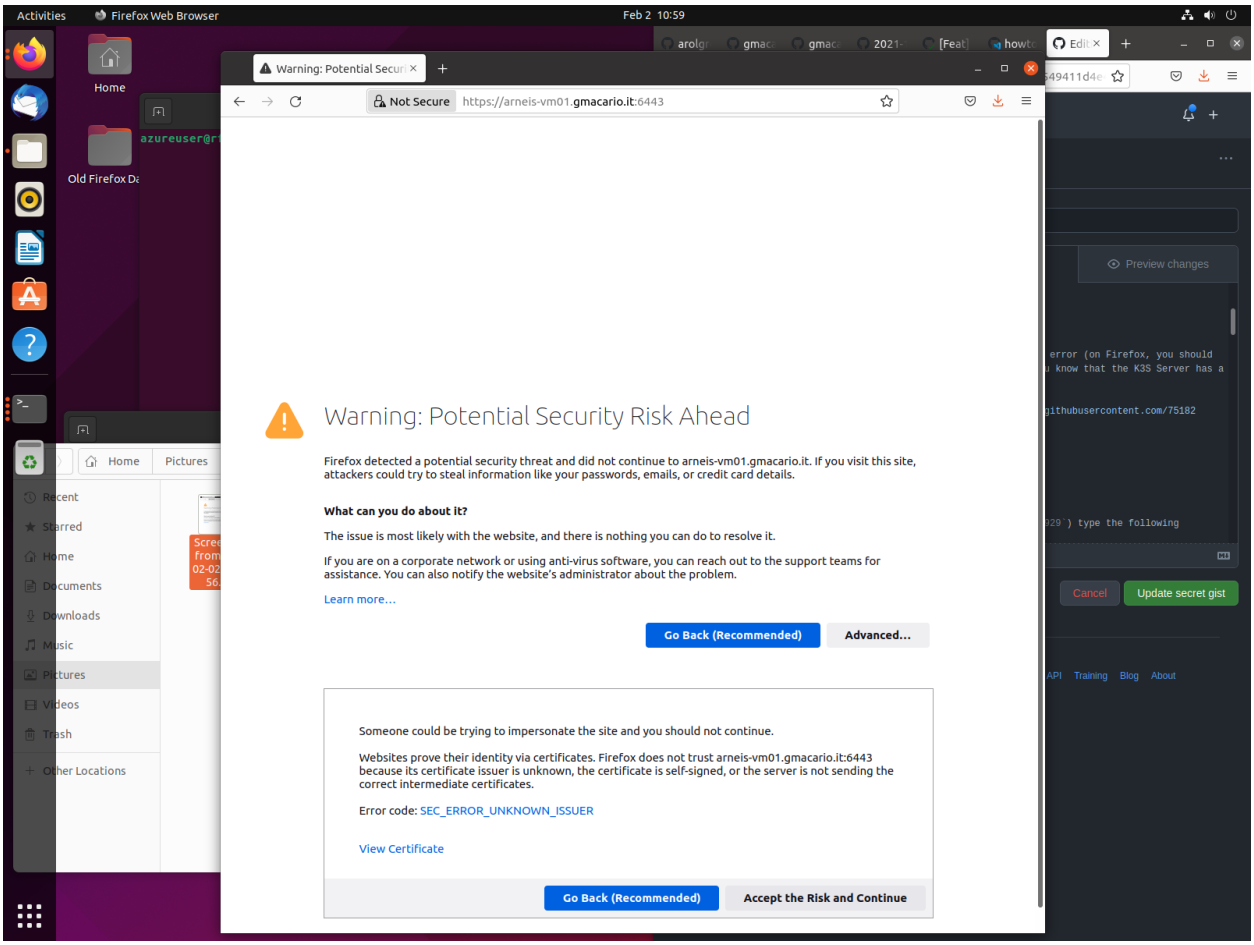
Again, HTTP error code 401 is expected since the K3s API server requires authentication.

Example 2: Using a web browser

If you access the K3S API URL from a browser, the following error (on Firefox, you should have a similar error message on other browsers) will let you know that the K3S Server has a self-signed TLS certificate:

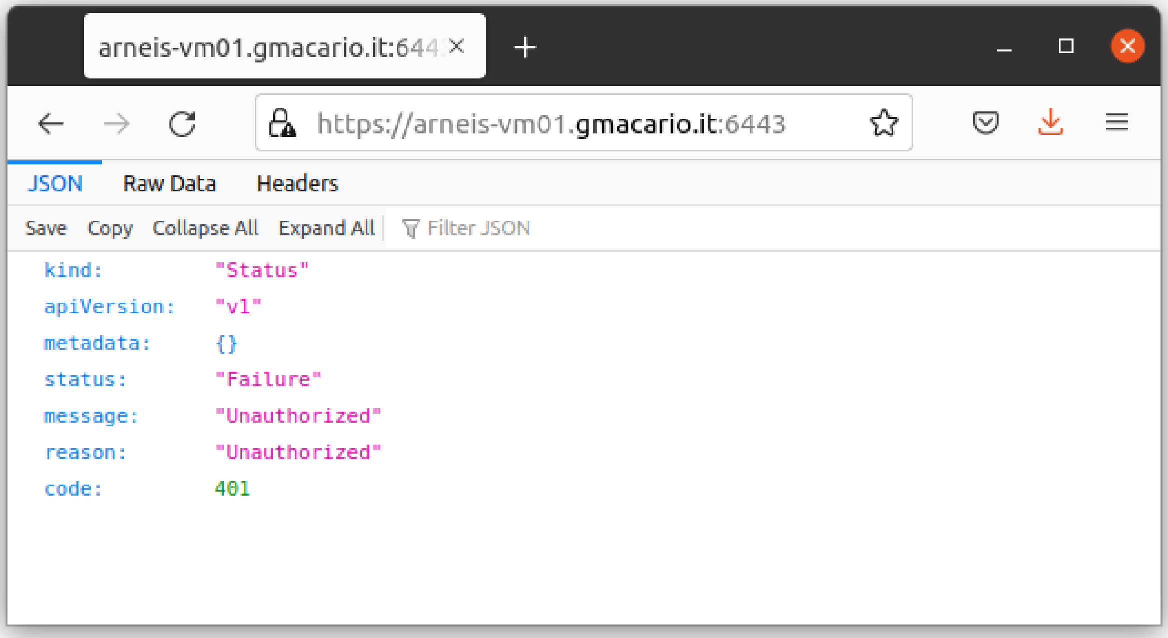


If so, click "Advanced…"



then click “Accept the Risk and Continue”.

Once you passed the self-signed certificate warning, you should receive a 401 (Unauthorized) error, but this is correct since we did not provide the node-token - see below.



Obtain the cluster node-token

The node-token is saved in a file under the folder `/var/lib/rancher/k3s/server` of the k3s server.

Logged in as `root@arneis-vm01`, display the k3s node-token with the following command:

```
cat /var/lib/rancher/k3s/server/node-token
```

Result (for security reasons the node-token has partially been anonymized)

```
root@arneis-vm01:~# cat /var/lib/rancher/k3s/server/node-token
K1015exxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxf06408::server:f22587xxxxxxxxxxxxxxxxxxxxx8672c3
root@arneis-vm01:~#
```

Attach the Agent Node

Logged in as `root@<agent-node>` (in our example, `root@arneis-vm02`) type the following commands to install the required software on the node and connect it to the k3s Server

Make sure you replace the placeholders

- `<myserver>` –> `arneis-vm01.gmacario.it`
- `<mynodetoken>` –> (result of the command in the previous section)

```
export K3S_URL=https://<myserver>:6443
export K3S_TOKEN=<mynodetoken>
curl -sfL https://get.k3s.io | sh -
```

Result:

```
root@arneis-vm02:~# export K3S_URL=https://arneis-vm01.gmacario.it:6443
root@arneis-vm02:~# export K3S_
↪TOKEN=K1015exxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxf06408::server:f22587xxxxxxxxxxxxxx
root@arneis-vm02:~# curl -sfL https://get.k3s.io | sh -
[INFO]  Finding release for channel stable
[INFO]  Using v1.22.7+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/
↪sha256sum-amd64.txt
[INFO]  Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/
↪k3s
[INFO]  Verifying binary download
[INFO]  Installing k3s to /usr/local/bin/k3s
[INFO]  Skipping installation of SELinux RPM
[INFO]  Creating /usr/local/bin/kubectl symlink to k3s
[INFO]  Creating /usr/local/bin/crictl symlink to k3s
[INFO]  Creating /usr/local/bin/ctr symlink to k3s
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-agent-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s-agent.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s-agent.service
[INFO]  systemd: Enabling k3s-agent unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s-agent.service → /etc/
↪systemd/system/k3s-agent.service.
[INFO]  systemd: Starting k3s-agent
root@arneis-vm02:~#
```

Check that the Agent Node has joined the cluster

Logged in as `root@<server-node>`, type the following command to list all the nodes of the cluster:

```
kubectl get nodes
```

```
root@arneis-vm01:~# kubectl get nodes
NAME          STATUS   ROLES                  AGE    VERSION
arneis-vm02   Ready    <none>                 34m    v1.22.7+k3s1
arneis-vm01   Ready    control-plane,master   5h3m   v1.22.7+k3s1
root@arneis-vm01:~#
```

Verify that the new Agent Node is listed. If not, refer to the section “Troubleshooting the agent install script” below.

4.18.5 Troubleshooting installation issue

This section lists a few methods to troubleshoot possible installation problems.

Run the agent install script in verbose mode

Let's retry adding the `-x` option to `sh`:

```
root@arneis-vm02:~# curl -sfL https://get.k3s.io | sh -x -
+ set -e
+ set -o noglob
+ GITHUB_URL=https://github.com/k3s-io/k3s/releases
+ STORAGE_URL=https://storage.googleapis.com/k3s-ci-builds
+ DOWNLOADER=
+ escape
+ sed -e s/\([][!#$%&()*;<=>?\_`{|}]\)/\\\1/g;
+ printf %s
+ quote
+ eval set --
+ set --
+ verify_system
+ [ -x /sbin/openrc-run ]
+ [ -x /bin/systemctl ]
+ HAS_SYSTEMD=true
+ return
+ setup_env
+ [ -z https://arneis-vm01.gmacario.it:6443 ]
+ [ -z␣
↪K1015exxxx
```

(continued from previous page)

```
+ SYSTEMD_DIR=/etc/systemd/system
+ SERVICE_K3S=k3s-agent.service
+ UNINSTALL_K3S_SH=/usr/local/bin/k3s-agent-uninstall.sh
+ KILLALL_K3S_SH=/usr/local/bin/k3s-killall.sh
+ [ true = true ]
+ FILE_K3S_SERVICE=/etc/systemd/system/k3s-agent.service
+ FILE_K3S_ENV=/etc/systemd/system/k3s-agent.service.env
+ get_installed_hashes
+ sha256sum /usr/local/bin/k3s /etc/systemd/system/k3s-agent.service /etc/systemd/system/
↪k3s-agent.service.env
+ PRE_INSTALL_HASHES=da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38  /
↪usr/local/bin/k3s
592ab27950afde56cde608c071201c498c70d1f031a18d41f4c40d48ffb91ccf  /etc/systemd/system/
↪k3s-agent.service
4ed99b900d46582af831a1858036cf2baf1205db18d6ac5391d35a75db4db0d8  /etc/systemd/system/
↪k3s-agent.service.env
+ [  = true ]
+ INSTALL_K3S_CHANNEL_URL=https://update.k3s.io/v1-release/channels
+ INSTALL_K3S_CHANNEL=stable
+ download_and_verify
+ can_skip_download
+ [  != true ]
+ return 1
+ setup_verify_arch
+ [ -z  ]
+ uname -m
+ ARCH=x86_64
+ ARCH=amd64
+ SUFFIX=
+ verify_downloader curl
+ command -v curl
+ [ -x /usr/bin/curl ]
+ DOWNLOADER=curl
+ return 0
+ setup_tmp
+ mktemp -d -t k3s-install.XXXXXXXXXX
+ TMP_DIR=/tmp/k3s-install.pGUb5wqYIH
+ TMP_HASH=/tmp/k3s-install.pGUb5wqYIH/k3s.hash
+ TMP_BIN=/tmp/k3s-install.pGUb5wqYIH/k3s.bin
+ trap cleanup INT EXIT
+ get_release_version
+ [ -n  ]
+ [ -n  ]
+ info Finding release for channel stable
+ echo [INFO]  Finding release for channel stable
[INFO]  Finding release for channel stable
+ version_url=https://update.k3s.io/v1-release/channels/stable
+ curl -w %{url_effective}+ sed -e s|.*/||
 -L -s -S https://update.k3s.io/v1-release/channels/stable -o /dev/null
+ VERSION_K3S=v1.22.7+k3s1
+ info Using v1.22.7+k3s1 as release
+ echo [INFO]  Using v1.22.7+k3s1 as release
```

(continues on next page)

(continued from previous page)

```
[INFO]  Using v1.22.7+k3s1 as release
+ download_hash
+ [ -n  ]
+ HASH_URL=https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/sha256sum-amd64.
↪txt
+ info Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/
↪sha256sum-amd64.txt
+ echo [INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.22.
↪7+k3s1/sha256sum-amd64.txt
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/
↪sha256sum-amd64.txt
+ download /tmp/k3s-install.pGUb5wqYIH/k3s.hash https://github.com/k3s-io/k3s/releases/
↪download/v1.22.7+k3s1/sha256sum-amd64.txt
+ [ 2 -eq 2 ]
+ curl -o /tmp/k3s-install.pGUb5wqYIH/k3s.hash -sfL https://github.com/k3s-io/k3s/
↪releases/download/v1.22.7+k3s1/sha256sum-amd64.txt
+ [ 0 -eq 0 ]
+ grep  k3s$ /tmp/k3s-install.pGUb5wqYIH/k3s.hash
+ HASH_EXPECTED=da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38  k3s
+ HASH_EXPECTED=da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38
+ installed_hash_matches
+ [ -x /usr/local/bin/k3s ]
+ sha256sum /usr/local/bin/k3s
+ HASH_INSTALLED=da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38  /usr/
↪local/bin/k3s
+ HASH_INSTALLED=da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38
+ [ da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38 =␣
↪da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38 ]
+ return
+ info Skipping binary downloaded, installed k3s matches hash
+ echo [INFO]  Skipping binary downloaded, installed k3s matches hash
[INFO]  Skipping binary downloaded, installed k3s matches hash
+ return
+ setup_selinux
+ rpm_channel=stable
+ rpm_site=rpm.rancher.io
+ [ stable = testing ]
+ [ -r /etc/os-release ]
+ . /etc/os-release
+ NAME=Ubuntu
+ VERSION=20.04.4 LTS (Focal Fossa)
+ ID=ubuntu
+ ID_LIKE=debian
+ PRETTY_NAME=Ubuntu 20.04.4 LTS
+ VERSION_ID=20.04
+ HOME_URL=https://www.ubuntu.com/
+ SUPPORT_URL=https://help.ubuntu.com/
+ BUG_REPORT_URL=https://bugs.launchpad.net/ubuntu/
+ PRIVACY_POLICY_URL=https://www.ubuntu.com/legal/terms-and-policies/privacy-policy
+ VERSION_CODENAME=focal
+ UBUNTU_CODENAME=focal
+ [ debian = suse ]
```

(continues on next page)

(continued from previous page)

```
+ [ 20 = 7 ]
+ rpm_target=el8
+ rpm_site_infix=centos/8
+ package_installer=yum
+ [ yum = yum ]
+ [ -x /usr/bin/dnf ]
+ policy_hint=please install:
    yum install -y container-selinux
    yum install -y https://rpm.rancher.io/k3s/stable/common/centos/8/noarch/k3s-selinux-
↪0.4-1.el8.noarch.rpm

+ [  = true ]
+ can_skip_download
+ [  != true ]
+ return 1
+ [ ! -d /usr/share/selinux ]
+ info Skipping installation of SELinux RPM
+ echo [INFO]  Skipping installation of SELinux RPM
[INFO]  Skipping installation of SELinux RPM
+ policy_error=fatal
+ [  = true ]
+ [ debian = coreos ]
+ [  = coreos ]
+ chcon -u system_u -r object_r -t container_runtime_exec_t /usr/local/bin/k3s
+ grep ^\s*SELINUX=enforcing /etc/selinux/config
+ create_symlinks
+ [  = true ]
+ [  = skip ]
+ [ ! -e /usr/local/bin/kubectl ]
+ [  = force ]
+ info Skipping /usr/local/bin/kubectl symlink to k3s, already exists
+ echo [INFO]  Skipping /usr/local/bin/kubectl symlink to k3s, already exists
[INFO]  Skipping /usr/local/bin/kubectl symlink to k3s, already exists
+ [ ! -e /usr/local/bin/crictl ]
+ [  = force ]
+ info Skipping /usr/local/bin/crictl symlink to k3s, already exists
+ echo [INFO]  Skipping /usr/local/bin/crictl symlink to k3s, already exists
[INFO]  Skipping /usr/local/bin/crictl symlink to k3s, already exists
+ [ ! -e /usr/local/bin/ctr ]
+ [  = force ]
+ info Skipping /usr/local/bin/ctr symlink to k3s, already exists
+ echo [INFO]  Skipping /usr/local/bin/ctr symlink to k3s, already exists
[INFO]  Skipping /usr/local/bin/ctr symlink to k3s, already exists
+ create_killall
+ [  = true ]
+ info Creating killall script /usr/local/bin/k3s-killall.sh
+ echo [INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
+ tee /usr/local/bin/k3s-killall.sh
+ chmod 755 /usr/local/bin/k3s-killall.sh
+ chown root:root /usr/local/bin/k3s-killall.sh
+ create_uninstall
```

(continues on next page)

(continued from previous page)

```
+ [  = true ]
+ info Creating uninstall script /usr/local/bin/k3s-agent-uninstall.sh
+ echo [INFO]  Creating uninstall script /usr/local/bin/k3s-agent-uninstall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-agent-uninstall.sh
+ tee /usr/local/bin/k3s-agent-uninstall.sh
+ chmod 755 /usr/local/bin/k3s-agent-uninstall.sh
+ chown root:root /usr/local/bin/k3s-agent-uninstall.sh
+ systemd_disable
+ systemctl disable k3s-agent
+ rm -f /etc/systemd/system/k3s-agent.service
+ rm -f /etc/systemd/system/k3s-agent.service.env
+ create_env_file
+ info env: Creating environment file /etc/systemd/system/k3s-agent.service.env
+ echo [INFO]  env: Creating environment file /etc/systemd/system/k3s-agent.service.env
[INFO]  env: Creating environment file /etc/systemd/system/k3s-agent.service.env
+ touch /etc/systemd/system/k3s-agent.service.env
+ chmod 0600 /etc/systemd/system/k3s-agent.service.env
+ tee /etc/systemd/system/k3s-agent.service.env
+ grep -E ^(K3S|CONTAINERD)_
+ read x v
+ sh -c export
+ echo HOME='/root'
+ read x v
+ echo K3S_TOKEN=
↪'K1015exxxx
```

(continued from previous page)

```
+ read x v
+ echo MAIL='/var/mail/root'
+ read x v
+ echo PATH='/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin'
+ read x v
+ echo PWD='/root'
+ read x v
+ echo SGX_AESM_ADDR='1'
+ read x v
+ echo SHELL='/bin/bash'
+ read x v
+ echo SHLVL='0'
+ read x v
+ echo SUDO_COMMAND='/bin/bash'
+ read x v
+ echo SUDO_GID='1000'
+ read x v
+ echo SUDO_UID='1000'
+ read x v
+ echo SUDO_USER='azureuser'
+ read x v
+ echo TERM='xterm'
+ read x v
+ echo USER='root'
+ read x v
+ echo XDG_DATA_DIRS='/usr/local/share:/usr/share:/var/lib/snapd/desktop'
+ read x v
+ echo _='/usr/bin/sh'
+ read x v
+ sh -c export
+ tee -a /etc/systemd/system/k3s-agent.service.env
+ grep -Ei ^(NO|HTTP|HTTPS)_PROXY
+ read x v
+ echo HOME='/root'
+ read x v
+ echo K3S_TOKEN=
↪'K1015exxxx
```

(continued from previous page)

```
+ read x v
+ echo MAIL='/var/mail/root'
+ read x v
+ echo PATH='/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin'
+ read x v
+ echo PWD='/root'
+ read x v
+ echo SGX_AESM_ADDR='1'
+ read x v
+ echo SHELL='/bin/bash'
+ read x v
+ echo SHLVL='0'
+ read x v
+ echo SUDO_COMMAND='/bin/bash'
+ read x v
+ echo SUDO_GID='1000'
+ read x v
+ echo SUDO_UID='1000'
+ read x v
+ echo SUDO_USER='azureuser'
+ read x v
+ echo TERM='xterm'
+ read x v
+ echo USER='root'
+ read x v
+ echo XDG_DATA_DIRS='/usr/local/share:/usr/share:/var/lib/snapd/desktop'
+ read x v
+ echo _='/usr/bin/sh'
+ read x v
+ create_service_file
+ [ true = true ]
+ create_systemd_service_file
+ info systemd: Creating service file /etc/systemd/system/k3s-agent.service
+ echo [INFO]  systemd: Creating service file /etc/systemd/system/k3s-agent.service
[INFO]  systemd: Creating service file /etc/systemd/system/k3s-agent.service
+ tee /etc/systemd/system/k3s-agent.service
+ [  = true ]
+ return 0
+ service_enable_and_start
+ [ -f /proc/cgroups ]
+ grep memory /proc/cgroups
+ read -r n n n enabled
+ echo 1
+ read -r n n n enabled
+ [ 1 -eq 0 ]
+ [  = true ]
+ [ true = true ]
+ systemd_enable
+ info systemd: Enabling k3s-agent unit
+ echo [INFO]  systemd: Enabling k3s-agent unit
[INFO]  systemd: Enabling k3s-agent unit
+ systemctl enable /etc/systemd/system/k3s-agent.service
```

(continues on next page)

(continued from previous page)

```
Created symlink /etc/systemd/system/multi-user.target.wants/k3s-agent.service → /etc/
↪systemd/system/k3s-agent.service.
+ systemctl daemon-reload
+ [  = true ]
+ [  = true ]
+ get_installed_hashes
+ sha256sum /usr/local/bin/k3s /etc/systemd/system/k3s-agent.service /etc/systemd/system/
↪k3s-agent.service.env
+ POST_INSTALL_HASHES=da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38  /
↪usr/local/bin/k3s
592ab27950afde56cde608c071201c498c70d1f031a18d41f4c40d48ffb91ccf  /etc/systemd/system/
↪k3s-agent.service
4ed99b900d46582af831a1858036cf2baf1205db18d6ac5391d35a75db4db0d8  /etc/systemd/system/
↪k3s-agent.service.env
+ [ da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38  /usr/local/bin/k3s
592ab27950afde56cde608c071201c498c70d1f031a18d41f4c40d48ffb91ccf  /etc/systemd/system/
↪k3s-agent.service
4ed99b900d46582af831a1858036cf2baf1205db18d6ac5391d35a75db4db0d8  /etc/systemd/system/
↪k3s-agent.service.env =␣
↪da1a566c6b3d470102ce431afdb921275ebe663659219562ac4d9854e5bbdf38  /usr/local/bin/k3s
592ab27950afde56cde608c071201c498c70d1f031a18d41f4c40d48ffb91ccf  /etc/systemd/system/
↪k3s-agent.service
4ed99b900d46582af831a1858036cf2baf1205db18d6ac5391d35a75db4db0d8  /etc/systemd/system/
↪k3s-agent.service.env ]
+ [  != true ]
+ info No change detected so skipping service start
+ echo [INFO]  No change detected so skipping service start
[INFO]  No change detected so skipping service start
+ return
+ cleanup
+ code=0
+ set +e
+ trap - EXIT
+ rm -rf /tmp/k3s-install.pGUb5wqYIH
+ exit 0
root@arneis-vm02:~#
```

Check processes on the server node

Logged in as `root@<server-node>` (in our case, `root@arneis-vm01`), make sure there is one `k3s` process running:

Check whether service `k3s.service` is running correctly

```
systemctl status k3s.service
```

Result:

```
root@arneis-vm01:~# systemctl status k3s.service
 k3s.service - Lightweight Kubernetes
     Loaded: loaded (/etc/systemd/system/k3s.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-03-09 10:49:20 UTC; 2h 32min ago
       Docs: https://k3s.io
```

(continues on next page)

(continued from previous page)

```
    Process: 1279 ExecStartPre=/bin/sh -xc ! /usr/bin/systemctl is-enabled --quiet nm-
↪cloud-setup.service (code=exited, status=0/SUCCESS)
    Process: 1281 ExecStartPre=/sbin/modprobe br_netfilter (code=exited, status=0/
↪SUCCESS)
    Process: 1283 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
   Main PID: 1288 (k3s-server)
      Tasks: 129
     Memory: 1.5G
     CGroup: /system.slice/k3s.service
             ├─1288 /usr/local/bin/k3s server
             ├─1344 containerd
             ├─1927 /var/lib/rancher/k3s/data/
↪31ff0fd447a47323a7c863dbb0a3cd452e12b45f1ec67dc55efa575503c2c3ac/bin/containerd-shim-
↪runc-v2 -namespace k8s.io -id 2d3eb89ad0348830e1695cd39df2>
             ├─1956 /pause
             ├─2118 /var/lib/rancher/k3s/data/
↪31ff0fd447a47323a7c863dbb0a3cd452e12b45f1ec67dc55efa575503c2c3ac/bin/containerd-shim-
↪runc-v2 -namespace k8s.io -id d2d99e90d65b431b8b29ce1837f1>
             ├─2140 /pause
             ├─2163 /var/lib/rancher/k3s/data/
↪31ff0fd447a47323a7c863dbb0a3cd452e12b45f1ec67dc55efa575503c2c3ac/bin/containerd-shim-
↪runc-v2 -namespace k8s.io -id 6a589841e9e0dab8eb072ec10cd7>
             ├─2185 /pause
             ├─2310 local-path-provisioner start --config /etc/config/config.json
             ├─2332 /metrics-server --cert-dir=/tmp --secure-port=4443 --kubelet-
↪preferred-address-types=InternalIP,ExternalIP,Hostname --kubelet-use-node-status-port -
↪-metric-resolution=15s
             ├─2386 /coredns -conf /etc/coredns/Corefile
             ├─3375 /var/lib/rancher/k3s/data/
↪31ff0fd447a47323a7c863dbb0a3cd452e12b45f1ec67dc55efa575503c2c3ac/bin/containerd-shim-
↪runc-v2 -namespace k8s.io -id 80865e63f2f0a03c7ccc756c2357>
             ├─3402 /pause
             ├─3411 /var/lib/rancher/k3s/data/
↪31ff0fd447a47323a7c863dbb0a3cd452e12b45f1ec67dc55efa575503c2c3ac/bin/containerd-shim-
↪runc-v2 -namespace k8s.io -id 4d23dfe4c6359855af86763fae78>
             ├─3439 /pause
             ├─3613 /bin/sh /usr/bin/entry
             ├─3656 /bin/sh /usr/bin/entry
             ├─3749 traefik traefik --global.checknewversion --global.sendanonymoususage␣
↪--entrypoints.metrics.address=:9100/tcp --entrypoints.traefik.address=:9000/tcp --
↪entrypoints.web.ad>
             ├─4088 /var/lib/rancher/k3s/data/
↪31ff0fd447a47323a7c863dbb0a3cd452e12b45f1ec67dc55efa575503c2c3ac/bin/containerd-shim-
↪runc-v2 -namespace k8s.io -id 1abb65ceb22a7d7f06870f851056>
             ├─4110 /pause
             └─4152 sleep 1000000

Mar 09 10:50:27 arneis-vm01 k3s[1288]: E0309 10:50:27.285999    1288 remote_runtime.
↪go:334] "ContainerStatus from runtime service failed" err="rpc error: code = NotFound␣
↪desc = an error occ>
Mar 09 10:50:27 arneis-vm01 k3s[1288]: I0309 10:50:27.286030    1288 kuberuntime_gc.
↪go:361] "Error getting ContainerStatus for containerID" containerID=
↪"8038952f3c91bcd4ae626d9886bd95dd14ce>
```

(continues on next page)

(continued from previous page)

```
Mar 09 10:56:50 arneis-vm01 k3s[1288]: I0309 10:56:50.155034    1288 topology_manager.
↪go:200] "Topology Admit Handler"
Mar 09 10:56:50 arneis-vm01 k3s[1288]: I0309 10:56:50.200279    1288 reconciler.go:225]
↪"operationExecutor.VerifyControllerAttachedVolume started for volume \"kube-api-access-
↪kq6v2\" (Unique>
Mar 09 11:00:30 arneis-vm01 k3s[1288]: time="2022-03-09T11:00:30Z" level=info msg=
↪"certificate CN=k3s,O=k3s signed by CN=k3s-server-ca@1646822958: notBefore=2022-03-09␣
↪10:49:18 +0000 UTC no>
Mar 09 11:00:30 arneis-vm01 k3s[1288]: time="2022-03-09T11:00:30Z" level=info msg=
↪"Updating TLS secret for k3s-serving (count: 10): map[listener.cattle.io/cn-10.0.0.
↪4:10.0.0.4 listener.catt>
Mar 09 11:00:30 arneis-vm01 k3s[1288]: time="2022-03-09T11:00:30Z" level=info msg=
↪"Active TLS secret k3s-serving (ver=855) (count 10): map[listener.cattle.io/cn-10.0.0.
↪4:10.0.0.4 listener.c>
Mar 09 11:00:30 arneis-vm01 k3s[1288]: time="2022-03-09T11:00:30Z" level=info msg=
↪"Updating TLS secret for k3s-serving (count: 10): map[listener.cattle.io/cn-10.0.0.
↪4:10.0.0.4 listener.catt>
Mar 09 12:47:28 arneis-vm01 k3s[1288]: time="2022-03-09T12:47:28Z" level=info msg=
↪"certificate CN=arneis-vm02 signed by CN=k3s-server-ca@1646822958: notBefore=2022-03-
↪09 10:49:18 +0000 UTC >
Mar 09 12:47:28 arneis-vm01 k3s[1288]: time="2022-03-09T12:47:28Z" level=info msg=
↪"certificate CN=system:node:arneis-vm02,O=system:nodes signed by CN=k3s-client-
↪ca@1646822958: notBefore=202>
root@arneis-vm01:~#
```

Check processes on the agent node

Logged in as `root@<agent-node>` (in our case, `root@arneis-vm02`), make sure there is one `k3s` process running:

```
ps ax | grep k3s
```

```
root@arneis-vm02:~# ps ax | grep k3s
   3013 ?        Ssl    0:03 /usr/local/bin/k3s agent
   3444 pts/0    S+     0:00 grep --color=auto k3s
root@arneis-vm02:~#
```

The service is up and running, but there is no `arneis-vm02` node listed as a result of `kubectl get nodes`. Let's check whether service `k3s-agent.service` is running correctly

```
systemctl status k3s-agent.service
```

Result:

```
root@arneis-vm02:~# systemctl status k3s-agent.service
 k3s-agent.service - Lightweight Kubernetes
     Loaded: loaded (/etc/systemd/system/k3s-agent.service; enabled; vendor preset:␣
↪enabled)
     Active: active (running) since Wed 2022-03-09 12:47:25 UTC; 20min ago
       Docs: https://k3s.io
   Main PID: 3013 (k3s-agent)
      Tasks: 18
```

(continues on next page)

(continued from previous page)

```
     Memory: 253.1M
     CGroup: /system.slice/k3s-agent.service
             ├─3013 /usr/local/bin/k3s agent
             └─3032 containerd

Mar 09 13:01:03 arneis-vm02 k3s[3013]: time="2022-03-09T13:01:03Z" level=info msg=
↪"Connecting to proxy" url="wss://10.0.0.4:6443/v1-k3s/connect"
Mar 09 13:03:14 arneis-vm02 k3s[3013]: time="2022-03-09T13:03:14Z" level=error msg=
↪"Failed to connect to proxy" error="dial tcp 10.0.0.4:6443: connect: connection timed␣
↪out"
Mar 09 13:03:14 arneis-vm02 k3s[3013]: time="2022-03-09T13:03:14Z" level=error msg=
↪"Remotedialer proxy error" error="dial tcp 10.0.0.4:6443: connect: connection timed out
↪"
Mar 09 13:03:19 arneis-vm02 k3s[3013]: time="2022-03-09T13:03:19Z" level=info msg=
↪"Connecting to proxy" url="wss://10.0.0.4:6443/v1-k3s/connect"
Mar 09 13:05:29 arneis-vm02 k3s[3013]: time="2022-03-09T13:05:29Z" level=error msg=
↪"Failed to connect to proxy" error="dial tcp 10.0.0.4:6443: connect: connection timed␣
↪out"
Mar 09 13:05:29 arneis-vm02 k3s[3013]: time="2022-03-09T13:05:29Z" level=error msg=
↪"Remotedialer proxy error" error="dial tcp 10.0.0.4:6443: connect: connection timed out
↪"
Mar 09 13:05:34 arneis-vm02 k3s[3013]: time="2022-03-09T13:05:34Z" level=info msg=
↪"Connecting to proxy" url="wss://10.0.0.4:6443/v1-k3s/connect"
Mar 09 13:07:44 arneis-vm02 k3s[3013]: time="2022-03-09T13:07:44Z" level=error msg=
↪"Failed to connect to proxy" error="dial tcp 10.0.0.4:6443: connect: connection timed␣
↪out"
Mar 09 13:07:44 arneis-vm02 k3s[3013]: time="2022-03-09T13:07:44Z" level=error msg=
↪"Remotedialer proxy error" error="dial tcp 10.0.0.4:6443: connect: connection timed out
↪"
Mar 09 13:07:49 arneis-vm02 k3s[3013]: time="2022-03-09T13:07:49Z" level=info msg=
↪"Connecting to proxy" url="wss://10.0.0.4:6443/v1-k3s/connect"
root@arneis-vm02:~#
```

NOTE: In our example the agent is trying to connect to proxy via URL `wss://10.0.0.4:6443/v1-k3s/connect` but this is probably not correct. Let's investigate it further.

Check network configuration on the server node

Logged in as `root@<server-node>` (in our case, `root@arneis-vm01`), check all the assigned IPv4 addresses, as well as the routing table:

```
ip addr | grep -w inet
ip route
```

Result:

```
root@arneis-vm01:~# ip addr | grep -w inet
    inet 127.0.0.1/8 scope host lo
    inet 10.0.0.4/24 brd 10.0.0.255 scope global eth0
    inet 10.42.0.0/32 scope global flannel.1
    inet 10.42.0.1/24 brd 10.42.0.255 scope global cni0
root@arneis-vm01:~# ip route
```

(continues on next page)

(continued from previous page)

```
default via 10.0.0.1 dev eth0 proto dhcp src 10.0.0.4 metric 100
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.4
10.42.0.0/24 dev cni0 proto kernel scope link src 10.42.0.1
168.63.129.16 via 10.0.0.1 dev eth0 proto dhcp src 10.0.0.4 metric 100
169.254.169.254 via 10.0.0.1 dev eth0 proto dhcp src 10.0.0.4 metric 100
root@arneis-vm01:~#
```

NOTE: IPv4 address `10.0.0.4` is from an internal (private) network, while the public IP address of `arneis-vm01` is the following:

```
root@arneis-vm01:~# curl ifconfig.co
20.124.132.35
root@arneis-vm01:~# host arneis-vm01.gmacario.it
arneis-vm01.gmacario.it has address 20.124.132.35
root@arneis-vm01:~#
```

Check network configuration on the agent node

Logged in as `root@<agent-node>` (in our case, `root@arneis-vm02`), check all the assigned IPv4 addresses, as well as the routing table:

```
ip addr | grep -w inet
ip route
```

Result:

```
root@arneis-vm02:~# ip addr | grep -w inet
    inet 127.0.0.1/8 scope host lo
    inet 10.2.0.4/24 brd 10.2.0.255 scope global eth0
root@arneis-vm02:~# ip route
default via 10.2.0.1 dev eth0 proto dhcp src 10.2.0.4 metric 100
10.2.0.0/24 dev eth0 proto kernel scope link src 10.2.0.4
168.63.129.16 via 10.2.0.1 dev eth0 proto dhcp src 10.2.0.4 metric 100
169.254.169.254 via 10.2.0.1 dev eth0 proto dhcp src 10.2.0.4 metric 100
root@arneis-vm02:~#
```

Test with server IP address rather than FQDN

```
export K3S_URL=https://20.124.132.35:6433
export K3S_TOKEN=K100xxxxxxx
curl -sfL https://get.k3s.io | sh -
```

Result:

```
root@hw0929:~# curl -sfL https://get.k3s.io | sh -
[INFO]  Finding release for channel stable
[INFO]  Using v1.22.6+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.22.6+k3s1/
↪sha256sum-amd64.txt
root@hw0929:~#
```

Check the version of the installed OS:

```
root@hw0929:~# cat /etc/os-release
PRETTY_NAME="Ubuntu 21.10"
NAME="Ubuntu"
VERSION_ID="21.10"
VERSION="21.10 (Impish Indri)"
VERSION_CODENAME=impish
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=impish
root@hw0929:~#
```

4.18.6 Controlling the cluster using k9s

Quoting https://k9scli.io/

> K9s is a terminal based UI to interact with your Kubernetes clusters. The aim of this project is to make it easier to navigate, observe and manage your deployed applications in the wild. K9s continually watches Kubernetes for changes and offers subsequent commands to interact with your observed resources.

Compared to the official `kubectl` command-line tool, `k9s` is a much easier way for monitoring and controlling your Kubernetes cluster.

Install k9s

The page at https://k9scli.io/topics/install/ provides instructions for installing k9s on the most popular Operating Systems.

For instance, you can install k9s on the main node of your cluster by executing the following command when logged in as `root@arneis-vm01`:

```
curl -sS https://webinstall.dev/k9s | bash
```

then follow the instructions that will be shown on the terminal. For instance, you may be asked to append the `$HOME/.local/bin` directory to your `PATH` environment variable.

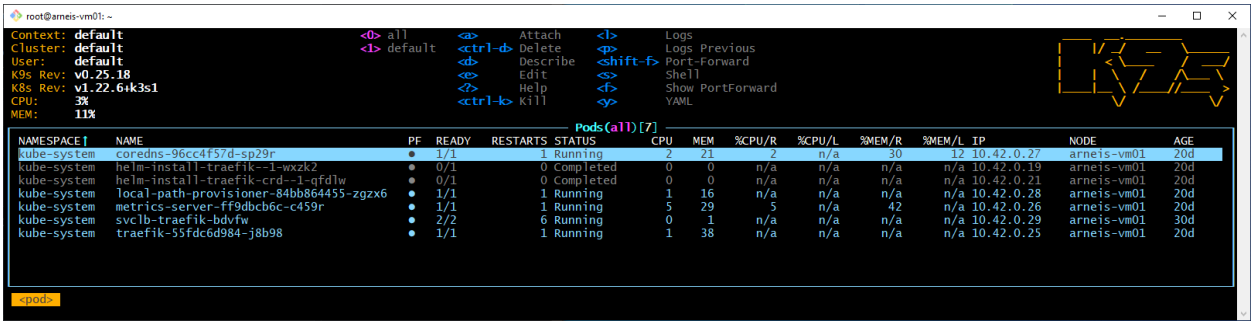
For further details on `webi`, please check https://webinstall.dev/ or its source repository on GitHub: https://github.com/webinstall/webi-installers

Using k9s

Logged in as `root@arneis-vm` you can launch k9s by providing the `--kubeconfig` option to specify the `k3s.yaml` file:

```
k9s --kubeconfig /etc/rancher/k3s/k3s.yaml
```

Result:



For further details about what you can do using the k9s command-line interface, the https://k9scli.io/topics/commands/ page provides a comprehensive list of the available CLI arguments and key bindings.

CHAPTER

FIVE

LEGO®

Here you find our collection of activities related to the development and construction of the ARNEIS LEGO MOC.

- **The original set we use:**
  - *LEGO Set 42100 (Liebherr R 9800 Excavator)*
  - *Unboxing LEGO® Set 42100*
- **The ARNEIS MOC:**
  - *LEGO® MOC studies*
  - *OAK-D-Lite LEGO® mechanical adapter*
  - *LEGO® MOCs Work in Progress*
  - *LEGO® ARNEIS Project*

5.1 LEGO Set 42100 (Liebherr R 9800 Excavator)

(Image courtesy of https://www.lego.com/en-it/product/liebherr-r-9800-excavator-42100)

- Catalog of LEGO® Set 42100-1 - BrickLink.com
- *Unboxing LEGO® Set 42100*

5.2 Unboxing LEGO® Set 42100

Reference: Catalog of LEGO® Set 42100-1 - BrickLink.com

5.2.1 Box 6273597

- Description: White carton box

Set of Bags #1

Bag 1A

- Code: 151S0 (238S1)
- QRCode: 6172092
- Size: cm 26w x 24h

Picture	Qty	Code	Description	Color
	18	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	Dark Bluish Gray
	8	32278	Technic, Liftarm Thick 1 x 15	Light Bluish Gray
	14	32316	Technic, Liftarm Thick 1 x 5	Dark Bluish Gray
	3	32498	Technic, Gear 36 Tooth Double Bevel	Black
	11	41239	Technic, Liftarm Thick 1 x 13	Dark Bluish Gray
	3	46834	Technic, Clutch Connector Female / Outside	White
	6	48989	Technic, Pin Connector Perpendicular 3L with 4 Pins	Light Bluish Gray
	2	55615	Technic, Pin Connector Perpendicular 3 x 3 Bent with 4 Pins	Light Bluish Gray
	6	60484	Technic, Liftarm, Modified T-Shape Thick 3 x 3	Dark Bluish Gray

Transparent Baglet 1A1

- Size: cm 11w x 8h

Picture	Qty	Code	Description	Color
	74	2780	Technic, Pin with Friction Ridges	Black
	36	43093	Technic, Axle 1L with Pin with Friction Ridges	Blue
	4	62462	Technic, Pin Connector Round 2L with Slot (Pin Joiner Round)	White

Transparent Baglet 1A2

- Size: cm 11w x 8h

Picture	Qty	Code	Description	Color
	74	2780	Technic, Pin with Friction Ridges	Black
	36	43093	Technic, Axle 1L with Pin with Friction Ridges	Blue
	4	62462	Technic, Pin Connector Round 2L with Slot (Pin Joiner Round)	White

Bag 1B

- Code: 251S0 (337S1)
- QRCode: 6172092
- Size: cm 26w x 24h

Picture	Qty	Code	Description	Color
	4	24121	Technic, Gear Rack 11 x 11 Curved	Black
	4	32009	Technic, Liftarm, Modified Bent Thick 1 x 11.5 Double	Dark Bluish Gray
	4	32498	Technic, Gear 36 Tooth Double Bevel	Black
	10	32524	Technic, Liftarm Thick 1 x 7	Light Bluish Gray
	8	32526	Technic, Liftarm, Modified Bent Thick L-Shape 3 x 5	Dark Bluish Gray
	2	32793	Technic, Pin Connector Block, Liftarm 1 x 3 x 3	Yellow
	4	3648	Technic, Gear 24 Tooth (2nd Version - 1 Axle Hole)	Dark Bluish Gray
	20	40490	Technic, Liftarm Thick 1 x 9	Dark Bluish Gray
	10	64179	Technic, Liftarm, Modified Frame Thick 5 x 7 Open Center	Light Bluish Gray

Bag 1C

- Code: 351S0 (138S1)
- QRCode: 6172090
- Size: cm 23w x 21h

Picture	Qty	Code	Description	Color
	5	15458	Technic, Panel Plate 3 x 11 x 1	Dark Bluish Gray
	1	18938	Technic Turntable Large Type 3 Top, 60 Tooth	Black
	1	18939	Technic Turntable Large Type 3 Base	Light Bluish Gray
	7	32269	Technic, Gear 20 Tooth Double Bevel	Tan
	2	32523	Technic, Liftarm Thick 1 x 3	Yellow
	12	32525	Technic, Liftarm Thick 1 x 11	Black
	2	3666	Plate 1 x 6	Red
	10	44294	Technic, Axle 7L	Light Bluish Gray
	4	42529	Technic Tread Sprocket Wheel Extra Large	Dark Bluish Gray
	2	46372	Technic, Gear 28 Tooth Double Bevel	Light Bluish Gray
	3	46835	Technic, Clutch Connector Male / Inside	Dark Bluish Gray
	4	55013	Technic, Axle 8L with Stop	Dark Bluish Gray

Transparent Baglet 1C1

- Size: cm 11w x 10h

Picture	Qty	Code	Description	Color
	41	11214	Technic, Axle 1L with Pin 2L with Friction Ridges	Red
	2	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Orange
	11	32062	Axle 2L Notched	Red
	10	32270	Technic, Gear 12 Tooth Double Beve	Black
	19	3713	Technic Bush	Red
	3	43093	Technic, Axle 1L with Pin with Friction Ridges	Blue
	6	60483	Technic, Liftarm Thick 1 x 2 - Axle Hole	White
	14	6536	Technic, Axle and Pin Connector Perpendicular	Dark Bluish Gray

Bag 1D

- Code: Unnamed
- QRCode: 6201713
- Size: cm 17w x 17h

Picture	Qty	Code	Description	Color
	104	57518	Technic, Link Tread Wide with 2 Pin Holes	Dark Bluish Gray

Bag 1E

- Code: Unnamed
- QRCode: 6230060
- Size: cm 14w x 13h

Picture	Qty	Code	Description	Color
	2	26287	Technic, Axle Connector 3L	Light Bluish Gray
	7	32073	Technic, Axle 5L	Yellow
	1	32209	Technic, Axle 5.5L with Stop	Dark Bluish Gray
	8	3705	Technic, Axle 4L	Black
	3	3706	Axle 6L	Red
	16	4185	Technic Wedge Belt Wheel (Pulley)	Dark Bluish Gray
	22	42003	Technic, Axle and Pin Connector Perpendicular 3L with 2 Pin Holes	Light Bluish Gray
	2	6636	Tile 1 x 6	Red
	4	87083	Technic, Axle 4L with Stop	Dark Bluish Gray
	28	87407	Technic, Gear 20 Tooth Bevel with Pin Hole	Light Bluish Gray
	2	94925	Technic, Gear 16 Tooth (Second Version - Reinforced)	Light Bluish Gray

Bag 1F

- Code: Unnamed
- QRCode: 6230059
- Size: cm 11w x 11h

Picture	Qty	Code	Description	Color
	2	10928	Technic, Gear 8 Tooth with Dual Face	Dark Bluish Gray
	8	15100	Technic, Pin with Friction Ridges and Pin Hole	Yellow
	4	18651	Technic, Axle 2L with Pin with Friction Ridges	Black
	7	24316	Technic, Axle 3L with Stop	Reddish Brown
	12	32039	Technic, Axle Connector with Axle Hole	Dark Bluish Gray
	16	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Light Bluish Gray
	29	32184	Technic, Axle and Pin Connector Perpendicular 3L with Center Pin Hole	Dark Bluish Gray
	16	32523	Liftarm Thick 1 x 3	Dark Bluish Gray
	19	4274	Technic, Pin 1/2 without Friction Ridges	Light Bluish Gray
	3	4519	Technic, Axle 3L	Light Bluish Gray
	2	49283	Technic, Axle and Wire Connector	Yellow

Bag 1G

- Code: Unnamed
- QRCode: 6230060
- Size: cm 14w x 11h

Picture	Qty	Code	Description	Color
	4	2780	Technic, Pin with Friction Ridges	Black
	2	32523	Liftarm Thick 1 x 3	Red
	17	4265c	Technic Bush 1/2 Smooth	Light Bluish Gray
	2	49283	Technic, Axle and Wire Connector	Red
	2	49283	Technic, Axle and Wire Connector	Green
	2	63864	Tile 1 x 3	Dark Bluish Gray
	86	6558	Technic, Pin 3L with Friction Ridges	Blue
	3	99008	Technic, Axle 4L with Center Stop	Tan

Set of Bags #2

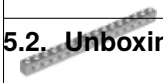
Bag 2A

- Code: ?
- QRCode: 6171966
- Size: cm 20w x 20h

Picture	Qty	Code	Description	Color
	9	32316	Technic, Liftarm Thick 1 x 5	White
	3	3460	Plate 1 x 8	Dark Bluish Gray
	8	4185	Technic Wedge Belt Wheel (Pulley)	Dark Bluish Gray
	17	64179	Technic, Liftarm, Modified Frame Thick 5 x 7 Open Center	Light Bluish Gray

Bag 2B

- Code: 251S0 (238S1)
- QRCode: 6172092
- Size: cm 26w x 23h

Picture	Qty	Code	Description	Color
	2	11478	Technic, Liftarm Thin 1 x 5 - Axle Holes	Light Bluish Gray
	1	15458	Technic, Panel Plate 3 x 11 x 1	Dark Bluish Gray
	3	15462	Technic, Axle 5L with Stop	Reddish Brown
	2	18944	Technic, Panel Curved 3 x 13	White
	1	18945	Technic, Panel Plate 5 x 11 x 1 Tapered	White
	1	3035	Plate 4 x 8	White
	1	3036	Plate 6 x 8	Dark Bluish Gray
	9	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	Dark Bluish Gray
	6	32184	Technic, Axle and Pin Connector Perpendicular 3L with Center Pin Hole	White
	4	32524	Technic, Liftarm Thick 1 x 7	White
	5	32525	Technic, Liftarm Thick 1 x 11	White
	2	32526	Technic, Liftarm, Modified Bent Thick L-Shape 3 x 5	Dark Bluish Gray
	2	3701	Technic, Brick 1 x 4 with Holes	Dark Bluish Gray
	2	3703	Technic, Brick 1 x 16 with Holes	Light Bluish Gray

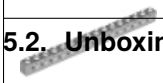
Transparent Baglet 2B1

- Size: cm 11w x 10h

Picture	Qty	Code	Description	Color
	121	2780	Technic, Pin with Friction Ridges	Black

Bag 2C

- Code: 251S0 (238S1)
- QRCode: 6172092
- Size: cm 26w x 23h

Picture	Qty	Code	Description	Color
	2	11478	Technic, Liftarm Thin 1 x 5 - Axle Holes	Light Bluish Gray
	1	15458	Technic, Panel Plate 3 x 11 x 1	Dark Bluish Gray
	3	15462	Technic, Axle 5L with Stop	Reddish Brown
	2	18944	Technic, Panel Curved 3 x 13	White
	1	18945	Technic, Panel Plate 5 x 11 x 1 Tapered	White
	1	3035	Plate 4 x 8	White
	1	3036	Plate 6 x 8	Dark Bluish Gray
	9	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	Dark Bluish Gray
	6	32184	Technic, Axle and Pin Connector Perpendicular 3L with Center Pin Hole	White
	4	32524	Technic, Liftarm Thick 1 x 7	White
	5	32525	Technic, Liftarm Thick 1 x 11	White
	2	32526	Technic, Liftarm, Modified Bent Thick L-Shape 3 x 5	Dark Bluish Gray
	2	3701	Technic, Brick 1 x 4 with Holes	Dark Bluish Gray
	2	3703	Technic, Brick 1 x 16 with Holes	Light Bluish Gray

Transparent Baglet 2C1

- Size: cm 11w x 10h

Picture	Qty	Code	Description	Color
	121	2780	Technic, Pin with Friction Ridges	Black

Bag 2D

- Code: 351S0
- QRCode: 6172092
- Size: cm 26w x 23h

Picture	Qty	Code	Description	Color
	7	15458	Technic, Panel Plate 3 x 11 x 1	White
	2	32269	Technic, Gear 20 Tooth Double Bevel	Tan
	13	32278	Technic, Liftarm Thick 1 x 15	Light Bluish Gray
	3	41239	Technic, Liftarm Thick 1 x 13	White
	2	4162	Tile 1 x 8	White
	5	41239	Technic, Liftarm Thick 1 x 13	Dark Bluish Gray
	1	44294	Technic, Axle 7L	Light Bluish Gray
	4	6629	Technic, Liftarm, Modified Bent Thick 1 x 9 (6 - 4)	White

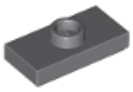
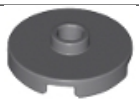
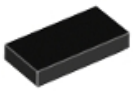
Transparent Baglet 2D1

- Size: cm 11w x 10h

Picture	Qty	Code	Description	Color
	3	18654	Technic, Liftarm Thick 1 x 1 (Spacer)	White
	6	30374	Bar 4L (Lightsaber Blade / Wand)	Light Bluish Gray
	21	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Orange
	7	32062	Technic, Axle 2L Notched	Red
	22	4274	Technic, Pin 1/2 without Friction Ridges	Light Bluish Gray
	9	4519	Technic, Axle 3L	Light Bluish Gray
	9	54200	Slope 30 1 x 1 x 2/3	Dark Bluish Gray
	2	6553	Technic, Axle 2L with Reverser Handle Axle Connector	Dark Bluish Gray
	7	87082	Technic, Pin 3L with Friction Ridges and Center Pin Hole	Light Bluish Gray
	9	98138	Tile, Round 1 x 1	Light Bluish Gray

Transparent Baglet 2D2

- Size: cm 11w x 10h

Picture	Qty	Code	Description	Color
	15	15100	Technic, Pin with Friction Ridges and Pin Hole	Black
	6	15573	Plate, Modified 1 x 2 with 1 Stud with Groove and Bottom Stud Holder (Jumper)	Dark Bluish Gray
	4	18674	Tile, Round 2 x 2 with Open Stud	Dark Bluish Gray
	1	3069b	Tile 1 x 2 with Groove	Black
	4	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Light Bluish Gray
	1	32523	Technic, Liftarm Thick 1 x 3	Red
	4	3713	Technic Bush	Red
	15	4073	Plate, Round 1 x 1	Flat Silver
	51	43093	Technic, Axle 1L with Pin with Friction Ridges	Blue
	2	49283	Technic, Axle and Wire Connector	Red
	3	6536	Technic, Axle and Pin Connector Perpendicular	Dark Bluish Gray

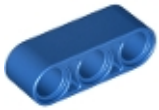
Bag 2E

- Code: ?
- QRCode: 6171966
- Size: cm 20w x 18h

Picture	Qty	Code	Description	Color
	15	26287	Technic, Axle Connector 3L	Light Bluish Gray
	2	2877	Brick, Modified 1 x 2 with Grille / Fluted Profile	Dark Bluish Gray
	8	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	White
	11	32523	Technic, Liftarm Thick 1 x 3	White
	5	32525	Technic, Liftarm Thick 1 x 11	Black
	19	32526	Technic, Liftarm, Modified Bent Thick L-Shape 3 x 5	White
	8	3705	Technic, Axle 4L	Black
	3	40490	Technic, Liftarm Thick 1 x 9	White
	15	42003	Technic, Axle and Pin Connector Perpendicular 3L with 2 Pin Holes	Light Bluish Gray
	1	48989	Technic, Pin Connector Perpendicular 3L with 4 Pins	Light Bluish Gray
	2	61904	Technic, Axle and Pin Connector Block 4 x 3 x 2 1/2 (Linear Actuator Holder)	Dark Bluish Gray
	3	87079	Tile 2 x 4	White
	4	94925	Technic, Gear 16 Tooth (Second Version - Reinforced)	Light Bluish Gray

Bag 2F

- Code: ?
- QRCode: 6230069
- Size: cm 14w x 12h

Picture	Qty	Code	Description	Color
	17	11214	Technic, Axle 1L with Pin 2L with Friction Ridges	Red
	2	18651	Technic, Axle 2L with Pin with Friction Ridges	Black
	4	32013	Technic, Axle and Pin Connector Angled #1	White
	2	32063	Technic, Liftarm Thin 1 x 6	White
	2	32073	Technic, Axle 5L	Yellow
	1	32270	Technic, Gear 12 Tooth Double Bevel	Black
	2	32291	Technic, Axle and Pin Connector Perpendicular Double	Black
	1	32523	Technic, Liftarm Thick 1 x 3	Green
	1	32523	Technic, Liftarm Thick 1 x 3	Yellow
	1	32523	Technic, Liftarm Thick 1 x 3	Blue
	1	3706	Technic, Axle 6L	Red
	4	4595	Brick, Modified 1 x 2 x 2/3 with Studs on Sides	Light Bluish Gray
	7	60483	Technic, Liftarm Thick 1 x 2 - Axle Hole	White
	16	62462	Technic, Pin Connector Round 2L with Slot (Pin Joiner Round)	White

Bag 2G

- Code: ?
- QRCode: 6230066
- Size: cm 11w x 10h

Picture	Qty	Code	Description	Color
	111	6558	Technic, Pin 3L with Friction Ridges	Blue

Bag 2H

- Code: ?
- QRCode: 6230066
- Size: cm 11w x 10h

Picture	Qty	Code	Description	Color
	2	14719	Tile 2 x 2 Corner	White
	3	20482	Tile, Round 1 x 1 with Bar and Pin Holder	Red
	3	20482	Tile, Round 1 x 1 with Bar and Pin Holder	Dark Azure
	8	2412b	Tile, Modified 1 x 2 Grille with Bottom Groove / Lip	Dark Bluish Gray
	64	2412b	Tile, Modified 1 x 2 Grille with Bottom Groove / Lip	White
	1	2780	Technic, Pin with Friction Ridges	Black
	2	3023	Plate 1 x 2	Black
	4	34103	Plate, Modified 1 x 3 with 2 Studs (Double Jumper)	Light Bluish Gray
	4	41677	Technic, Liftarm Thin 1 x 2 - Axle Holes	Dark Bluish Gray
	4	4733	Brick, Modified 1 x 1 with Studs on 4 Sides	Dark Bluish Gray
	4	6589	Technic, Gear 12 Tooth Bevel	Tan

Shrink wrap bag

- Size: A4

Picture	Qty	Code	Description	Color
?	1	?	Manual for Set 42100 (1/2)	-
?	1	?	Manual for Set 42100 (2/2)	-
	1	42100stk01	Sticker Sheet for Set 42100 - Sheet 1 (64870/6277261)	-
	1	42100stk02	Sticker Sheet for Set 42100 - Sheet 2 (64955/6277276)	-

5.2.2 Box 6278443

- Description: Black carton box

Picture	Qty	Code	Description	Color
	4	bb0959c01	Electric, Motor Powered Up, L	Dark Bluish Gray
	3	bb0960c01	Electric, Motor Powered Up, XL	Dark Bluish Gray
	2	bb0961c01	Electric Battery Box Powered Up Bluetooth Hub with Dark Bluish Gray Bottom	Light Bluish Gray

5.2.3 Set of Bags #3

Bag 3A

- Code: 751S0 (638S1)
- QRCode: 6172092
- Size: cm 26w x 25h

Picture	Qty	Code	Description	Color
	3	15458	Technic, Panel Plate 3 x 11 x 1	White
	6	15458	Technic, Panel Plate 3 x 11 x 1	Dark Bluish Gray
	2	32278	Technic, Liftarm Thick 1 x 15	White
	2	32278	Technic, Liftarm Thick 1 x 15	Light Bluish Gray
	1	3708	Technic, Axle 12L	Black
	7	41239	Technic, Liftarm Thick 1 x 13	White
	6	41239	Technic, Liftarm Thick 1 x 13	Dark Bluish Gray
	2	61904	Technic, Axle and Pin Connector Block 4 x 3 x 2 1/2 (Linear Actuator Holder)	Dark Bluish Gray
	1	64179	Technic, Liftarm, Modified Frame Thick 5 x 7 Open Center	Light Bluish Gray
	6	64782	Technic, Panel Plate 5 x 11 x 1	White
	2	92693c01	Technic Linear Actuator Mini with Dark Bluish Gray Head and Orange Axle	Light Bluish Gray

Bag 3B

- Code: 551S0 (538S1)
- QRCode: 6172090
- Size: cm 23w x 23h

Picture	Qty	Code	Description	Color
	5	14720	Technic, Liftarm, Modified H-Shape Thick 3 x 5 Perpendicular	Light Bluish Gray
	4	18945	Technic, Panel Plate 5 x 11 x 1 Tapered	White
	9	32524	Technic, Liftarm Thick 1 x 7	White
	3	32525	Technic, Liftarm Thick 1 x 11	Black
	19	32526	Technic, Liftarm, Modified Bent Thick L-Shape 3 x 5	White
	8	40490	Technic, Liftarm Thick 1 x 9	Dark Bluish Gray
	10	40490	Technic, Liftarm Thick 1 x 9	White
	4	40918c01	Technic Linear Actuator 12 x 2 x 2 (Contracted)	Light Bluish Gray
	1	64178	Technic, Liftarm, Modified Frame Thick 5 x 11 Open Center	Light Bluish Gray
	2	6629	Technic, Liftarm, Modified Bent Thick 1 x 9 (6 - 4)	White
	3	87408	Technic, Pin Connector Toggle Joint Smooth Double with Axle and Pin Holes	Black

Bag 3C

- Code: Unnamed
- QRCode: 6171968
- Size: cm 20w x 20h

Picture	Qty	Code	Description	Color
	1	11946	Technic, Panel Fairing #21 Very Small Smooth, Side B	White
	9	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	Dark Bluish Gray
	10	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	White
	2	32269	Technic, Gear 20 Tooth Double Bevel	Tan
	3	32316	Technic, Liftarm Thick 1 x 5	Dark Bluish Gray
	15	32525	Technic, Liftarm Thick 1 x 11	White
	3	60485	Technic, Axle 9L	Yellow
	4	99009	Technic Turntable Small Bottom	Light Bluish Gray
	4	99010	Technic Turntable Small Top	Black

Transparent Baglet 3C1

- Size: cm 9w x 9h

Picture	Qty	Code	Description	Color
	19	11214	Technic, Axle 1L with Pin 2L with Friction Ridges	Red
	8	15100	Technic, Pin with Friction Ridges and Pin Hole	Yellow
	4	18654	Technic, Liftarm Thick 1 x 1 (Spacer)	White
	3	2780	Technic, Pin with Friction Ridges	Black
	3	32002	Technic, Pin 3/4	Tan
	6	32062	Technic, Axle 2L Notched	Red
	2	4265c	Technic Bush 1/2 Smooth	Light Bluish Gray
	7	32556	Technic, Pin 3L without Friction Ridges	Tan
	2	3673	Technic, Pin without Friction Ridges	Light Bluish Gray
	2	3749	Technic, Axle 1L with Pin without Friction Ridges	Tan
	8	41677	Technic, Liftarm Thin 1 x 2 - Axle Holes	Dark Bluish Gray
	5	43093	Technic, Axle 1L with Pin with Friction Ridges	Blue
	10	4519	Technic, Axle 3L	Light Bluish Gray
	1	49283	Technic, Axle and Wire Connector	Blue

Bag 3D

- Code: Unnamed
- QRCode: 6171968
- Size: cm 20w x 19h

Picture	Qty	Code	Description	Color
	4	11478	Technic, Liftarm Thin 1 x 5 - Axle Holes	Light Bluish Gray
	2	32013	Technic, Axle and Pin Connector Angled #1	White
	11	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Light Bluish Gray
	4	32056	Technic, Liftarm, Modified Bent Thin L-Shape 3 x 3	Dark Bluish Gray
	3	32073	Technic, Axle 5L	Yellow
	7	32209	Technic, Axle 5.5L with Stop	Dark Bluish Gray
	5	32270	Technic, Gear 12 Tooth Double Bevel	Black
	4	32523	Technic, Liftarm Thick 1 x 3	White
	22	42003	Technic, Axle and Pin Connector Perpendicular 3L with 2 Pin Holes	Light Bluish Gray
	12	6538c	Technic, Axle Connector 2L (Smooth with x Hole + Orientation)	Dark Bluish Gray
	3	62520c01	Technic, Universal Joint 3L	Light Bluish Gray
	11	6589	Technic, Gear 12 Tooth Bevel	Tan
	2	87083	Technic, Axle 4L with Stop	Dark Bluish Gray

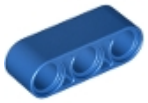
Transparent Baglet 3D1

- Size: cm 9w x 9h

Picture	Qty	Code	Description	Color
	18	18651	Technic, Axle 2L with Pin with Friction Ridges	Black
	3	18654	Technic, Liftarm Thick 1 x 1 (Spacer)	Dark Bluish Gray
	2	27940	Technic, Axle and Pin Connector Hub with 2 Axles	Dark Bluish Gray
	5	32523	Technic, Liftarm Thick 1 x 3	Dark Bluish Gray
	20	60483	Technic, Liftarm Thick 1 x 2 - Axle Hole	White

Bag 3E

- Code: 152S0 (638S1)
- QRCode: 6172092
- Size: cm 26w x 21h

Picture	Qty	Code	Description	Color
	1	11947	Technic, Panel Fairing #22 Very Small Smooth, Side A	White
	5	18946	Technic, Gear 16 Tooth with Clutch on Both Sides	Red
	1	26287	Technic, Axle Connector 3L	Light Bluish Gray
	4	32063	Technic, Liftarm Thin 1 x 6	White
	17	32184	Technic, Axle and Pin Connector Perpendicular 3L with Center Pin Hole	White
	2	32316	Technic, Liftarm Thick 1 x 5	White
	1	32523	Technic, Liftarm Thick 1 x 3	Yellow
	1	32523	Technic, Liftarm Thick 1 x 3	Blue
	2	3705	Technic, Axle 4L	Black
	4	3706	Technic, Axle 6L	Red
	2	3707	Technic, Axle 8L	Black
	1	3737	Technic, Axle 10L	Red
	2	41678	Technic, Axle and Pin Connector Perpendicular Double Split	Dark Bluish Gray
	6	44294	Technic, Axle 7L	Light Bluish Gray
	6	48989	Technic, Pin Connector Perpendicular 3L with 4 Pins	Light Bluish

Transparent Baglet 3E1

- Size: cm 11w x 9h

Picture	Qty	Code	Description	Color
	67	2780	Technic, Pin with Friction Ridges	Black
	4	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Orange
	7	3713	Technic Bush	Red
	21	43093	Technic, Axle 1L with Pin with Friction Ridges	Blue
	5	6536	Technic, Axle and Pin Connector Perpendicular	Dark Bluish Gray

Transparent Baglet 3E2

- Size: cm 11w x 9h

Picture	Qty	Code	Description	Color
	67	2780	Technic, Pin with Friction Ridges	Black
	4	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Orange
	7	3713	Technic Bush	Red
	21	43093	Technic, Axle 1L with Pin with Friction Ridges	Blue
	5	6536	Technic, Axle and Pin Connector Perpendicular	Dark Bluish Gray

Bag 3F

- Code: Unnamed
- QRCode: 6231597
- Size: cm 11w x 11h

Picture	Qty	Code	Description	Color
	2	4274	Technic, Pin 1/2 without Friction Ridges	Light Bluish Gray
	1	49283	Technic, Axle and Wire Connector	Yellow
	122	6558	Technic, Pin 3L with Friction Ridges	Blue

5.2.4 Set of Bags #4

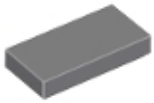
Bag 4A

- Code: 252S0 (438S1)
- QRCode: 6172092
- Size: cm 26w x 23h

Picture	Qty	Code	Description	Color
	2	14720	Technic, Liftarm, Modified H-Shape Thick 3 x 5 Perpendicular	Light Bluish Gray
	3	15458	Technic, Panel Plate 3 x 11 x 1	White
	2	15458	Technic, Panel Plate 3 x 11 x 1	Dark Bluish Gray
	6	3001	Brick 2 x 4	Dark Bluish Gray
	12	3002	Brick 2 x 3	Dark Bluish Gray
	2	3036	Plate 6 x 8	Dark Bluish Gray
	1	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	White
	2	32192	Technic, Axle and Pin Connector Angled #4 - 135 degrees	Dark Bluish Gray
	3	32278	Technic, Liftarm Thick 1 x 15	White
	2	32316	Technic, Liftarm Thick 1 x 5	Dark Bluish Gray
	1	32524	Technic, Liftarm Thick 1 x 7	White
	2	32525	Technic, Liftarm Thick 1 x 11	Black
	1	32526	Technic, Liftarm, Modified Bent Thick L-Shape 3 x 5	White

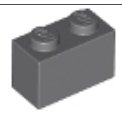
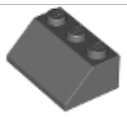
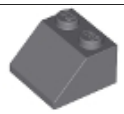
Transparent Baglet 4A1

- Size: cm 11w x 9h

Picture	Qty	Code	Description	Color
	10	18651	Technic, Axle 2L with Pin with Friction Ridges	Black
	12	2412b	Tile, Modified 1 x 2 Grille with Bottom Groove / Lip	White
	4	3069b	Tile 1 x 2 with Groove	Dark Bluish Gray
	3	3070b	Tile 1 x 1 with Groove	Dark Bluish Gray
	1	32062	Technic, Axle 2L Notched	Red
	8	4265c	Technic Bush 1/2 Smooth	Light Bluish Gray
	2	3713	Technic Bush	Red
	9	4274	Technic, Pin 1/2 without Friction Ridges	Light Bluish Gray
	21	43093	Technic, Axle 1L with Pin with Friction Ridges	Blue
	2	62462	Technic, Pin Connector Round 2L with Slot (Pin Joiner Round)	Flat Silver
	7	6558	Technic, Pin 3L with Friction Ridges	Blue
	8	98138	Tile, Round 1 x 1	Light Bluish Gray

Bag 4B

- Code: Unnamed
- QRCode: 6171970
- Size: cm 20w x 20h

Picture	Qty	Code	Description	Color
	8	25214	Brick, Round 1 x 1 d. 90 Degree Elbow - No Stud - Type 2 - Axle Hole	Pearl Dark Gray
	4	26287	Technic, Axle Connector 3L	Light Bluish Gray
	4	3004	Brick 1 x 2	Dark Bluish Gray
	4	3038	Slope 45 2 x 3	Dark Bluish Gray
	6	3039	Slope 45 2 x 2	Dark Bluish Gray
	4	32073	Technic, Axle 5L	Yellow
	2	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	Dark Bluish Gray
	2	32523	Technic, Liftarm Thick 1 x 3	White
	4	3460	Plate 1 x 8	Dark Bluish Gray
	5	3460	Plate 1 x 8	White
	4	3666	Plate 1 x 6	Dark Bluish Gray
	4	3710	Plate 1 x 4	White
	8	3894	Technic, Brick 1 x 6 with Holes	Dark Bluish Gray

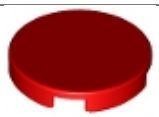
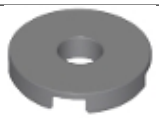
Transparent Baglet 4B1

- Size: cm 9w x 8h

Picture	Qty	Code	Description	Color
	46	2780	Technic, Pin with Friction Ridges	Black

Bag 4C

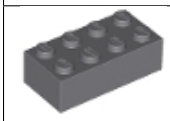
- Code: Unnamed
- QRCode: 6231618
- Size: cm 14w x 12h

Picture	Qty	Code	Description	Color
	6	11214	Technic, Axle 1L with Pin 2L with Friction Ridges	Red
	8	14769	Tile, Round 2 x 2 with Bottom Stud Holder	Red
	7	15100	Technic, Pin with Friction Ridges and Pin Hole	Black
	2	15535	Tile, Round 2 x 2 with Hole	Dark Bluish Gray
	6	2431	Tile 1 x 4	Dark Bluish Gray
	8	26603	Tile 2 x 3	Dark Bluish Gray
	4	3021	Plate 2 x 3	Dark Bluish Gray
	4	3045	Slope 45 2 x 2 Double Convex Corner	Dark Bluish Gray
	1	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Light Bluish Gray
	2	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Orange
	1	32523	Technic, Liftarm Thick 1 x 3	Dark Bluish Gray
	4	34103	Plate, Modified 1 x 3 with 2 Studs (Double Jumper)	Light Bluish Gray
	1	3705	Technic, Axle 4L	Black

5.2.5 Set of Bags #5

Bag 5A

- Code: 551S0 (634S1)
- QRCode: 6172090
- Size: cm 23w x 23h

Picture	Qty	Code	Description	Color
	1	15458	Technic, Panel Plate 3 x 11 x 1	Dark Bluish Gray
	2	15458	Technic, Panel Plate 3 x 11 x 1	White
	2	23948	Technic, Axle 11L	Light Bluish Gray
	3	24119	Technic, Panel Curved 7 x 3 with 2 Pin Holes through Panel Surface	White
	3	3001	Brick 2 x 4	Dark Bluish Gray
	1	3002	Brick 2 x 3	Dark Bluish Gray
	1	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	White
	1	32316	Technic, Liftarm Thick 1 x 5	White
	4	32524	Technic, Liftarm Thick 1 x 7	White
	3	32525	Technic, Liftarm Thick 1 x 11	White
	2	40490	Technic, Liftarm Thick 1 x 9	White
	7	40490	Technic, Liftarm Thick 1 x 9	Dark Bluish Gray
	2	41239	Technic, Liftarm Thick 1 x 13	White
	6	41239	Technic, Liftarm Thick 1 x 13	Dark Bluish

Transparent Baglet 5A1

- Size: cm 9w x 9h

Picture	Qty	Code	Description	Color
	15	18654	Technic, Liftarm Thick 1 x 1 (Spacer)	Dark Bluish Gray
	19	2780	Technic, Pin with Friction Ridges	Black
	17	32062	Technic, Axle 2L Notched	Red
	16	4519	Technic, Axle 3L	Light Bluish Gray
	16	6558	Technic, Pin 3L with Friction Ridges	Blue

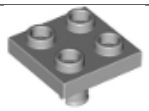
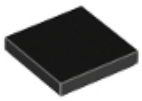
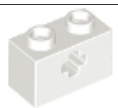
Transparent Baglet 5A2

- Size: cm 9w x 9h

Picture	Qty	Code	Description	Color
	15	18654	Technic, Liftarm Thick 1 x 1 (Spacer)	Dark Bluish Gray
	19	2780	Technic, Pin with Friction Ridges	Black
	17	32062	Technic, Axle 2L Notched	Red
	16	4519	Technic, Axle 3L	Light Bluish Gray
	16	6558	Technic, Pin 3L with Friction Ridges	Blue

Bag 5B

- Code: Unnamed
- QRCode: 6171972
- Size: cm 20w x 20h

Picture	Qty	Code	Description	Color
	1	2431	Tile 1 x 4	Dark Bluish Gray
	2	2476	Plate, Modified 2 x 2 with Pin on Bottom	Light Bluish Gray
	1	26287	Technic, Axle Connector 3L	Light Bluish Gray
	2	27940	Technic, Axle and Pin Connector Hub with 2 Axles	Dark Bluish Gray
	1	2817	Plate, Modified 2 x 2 with Pin Holes	Black
	1	3068b	Tile 2 x 2 with Groove	Black
	2	32014	Technic, Axle and Pin Connector Angled #6 - 90 degrees	Dark Bluish Gray
	2	32056	Technic, Liftarm, Modified Bent Thin L-Shape 3 x 3	Dark Bluish Gray
	6	32063	Technic, Liftarm Thin 1 x 6	White
	3	32064	Technic, Brick 1 x 2 with Axle Hole	White
	1	32073	Technic, Axle 5L	Yellow
	1	32184	Technic, Axle and Pin Connector Perpendicular 3L with Center Pin Hole	Dark Bluish Gray
	2	32184	Technic, Axle and Pin Connector Perpendicular 3L with Center Pin Hole	White
	6	32192	Technic, Axle and Pin Connector Angled #4 - 135 degrees	Dark Bluish

Transparent Baglet 5B1

- Size: cm 9w x 9h

Picture	Qty	Code	Description	Color
	12	15100	Technic, Pin with Friction Ridges and Pin Hole	Black
	1	18654	Technic, Liftarm Thick 1 x 1 (Spacer)	White
	1	24316	Technic, Axle 3L with Stop	Reddish Brown
	2	3023	Plate 1 x 2	Dark Bluish Gray
	2	32054	Technic, Pin 3L with Friction Ridges and Stop Bush	Light Bluish Gray
	4	32062	Technic, Axle 2L Notched	Red
	33	4265c	Technic Bush 1/2 Smooth	Light Bluish Gray
	2	32556	Technic, Pin 3L without Friction Ridges	Tan
	1	3713	Technic Bush	Red
	35	43093	Technic, Axle 1L with Pin with Friction Ridges	Blue
	1	4349	Minifigure, Utensil Loudhailer / Megaphone / SW Blaster	Black
	3	48729b	Bar 1L with Clip Mechanical Claw - Cut Edges and Hole on Side	Black
	2	6536	Technic, Axle and Pin Connector Perpendicular	White
	6	6558	Technic, Pin 3L with Friction Ridges	Blue

Bag 5C

- Code: Unnamed
- QRCode: 6171972
- Size: cm 10w x 19h

Picture	Qty	Code	Description	Color
	8	10197	Technic, Axle and Pin Connector Hub with 2 Perpendicular Axles	Dark Bluish Gray
	10	32065	Technic, Liftarm Thin 1 x 7	White
	2	32065	Technic, Liftarm Thin 1 x 7	Black
	4	32140	Technic, Liftarm, Modified Bent Thick L-Shape 2 x 4	Dark Bluish Gray
	3	32316	Technic, Liftarm Thick 1 x 5	Dark Bluish Gray
	2	32316	Technic, Liftarm Thick 1 x 5	Light Bluish Gray
	8	32449	Technic, Liftarm Thin 1 x 4 - Axle Holes	White
	6	32523	Technic, Liftarm Thick 1 x 3	Light Bluish Gray
	5	32524	Technic, Liftarm Thick 1 x 7	Dark Bluish Gray
	6	3707	Technic, Axle 8L	Black
	10	42003	Technic, Axle and Pin Connector Perpendicular 3L with 2 Pin Holes	Light Bluish Gray
	4	44294	Technic, Axle 7L	Light Bluish Gray
	12	60483	Technic, Liftarm Thick 1 x 2 - Axle Hole	White

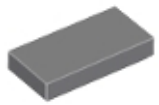
Transparent Baglet 5C1

- Size: cm 9w x 9h

Picture	Qty	Code	Description	Color
	15	18654	Technic, Liftarm Thick 1 x 1 (Spacer)	Dark Bluish Gray
	19	2780	Technic, Pin with Friction Ridges	Black
	17	32062	Technic, Axle 2L Notched	Red
	16	4519	Technic, Axle 3L	Light Bluish Gray
	16	6558	Technic, Pin 3L with Friction Ridges	Blue

Bag 5D

- Code: Unnamed
- QRCode: 6231628
- Size: cm 11w x 10h

Picture	Qty	Code	Description	Color
	22	11214	Technic, Axle 1L with Pin 2L with Friction Ridges	Red
	4	18651	Technic, Axle 2L with Pin with Friction Ridges	Black
	4	22961	Technic, Axle and Pin Connector Hub with 1 Axle	Light Bluish Gray
	4	2412b	Tile, Modified 1 x 2 Grille with Bottom Groove / Lip	White
	5	3069b	Tile 1 x 2 with Groove	Dark Bluish Gray
	5	3069b	Tile 1 x 2 with Groove	Trans-Clear
	6	44	Technic, Axle and Pin Connector Toggle Joint Smooth	Black
	5	34103	Plate, Modified 1 x 3 with 2 Studs (Double Jumper)	Light Bluish Gray
	14	41677	Technic, Liftarm Thin 1 x 2 - Axle Holes	Dark Bluish Gray
	6	4073	Plate, Round 1 x 1	Yellow
	1	62462	Technic, Pin Connector Round 2L with Slot (Pin Joiner Round)	White
	6	87082	Technic, Pin 3L with Friction Ridges and Center Pin Hole	Light Bluish Gray
	7	98138	Tile, Round 1 x 1	Trans-Clear

Bag 5E

- Code: Unnamed
- QRCode: 6231628
- Size: cm 11w x 11h

Picture	Qty	Code	Description	Color
	22	4274	Technic, Pin 1/2 without Friction Ridges	Light Bluish Gray
	83	6538c	Technic, Axle Connector 2L (Smooth with x Hole + Orientation)	Dark Bluish Gray
	2	62462	Technic, Pin Connector Round 2L with Slot (Pin Joiner Round)	Black
	5	6536	Technic, Axle and Pin Connector Perpendicular	Dark Bluish Gray

Bag 5F

- Code: Unnamed
- QRCode: 6231628
- Size: cm 11w x 11h

Picture	Qty	Code	Description	Color
	57	32039	Technic, Axle Connector with Axle Hole	Dark Bluish Gray
	42	99021	Pneumatic Hose Connector with Axle Connector	Dark Bluish Gray

Transparent Bag 5G

- Size: cm 9w x 27h

Picture	Qty	Code	Description	Color
TODO				
	8	5102c13	Technic Pneumatic Tube 13L	Black
	8	5102c17	Technic Pneumatic Tube 17L	Black
	8	5102c21	Technic Pneumatic Tube 21L	Black

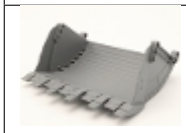
5.2.6 Set of Bags #6

Bag 6A

- Code: Unnamed
- QRCode: 6201721
- Size: cm 17w x 14h

Picture	Qty	Code	Description	Color
	100	3941	Brick, Round 2 x 2 with Axle Hole	Dark Bluish Gray

5.2.7 Other loose parts present in the box

Picture	Qty	Code	Description	Color
	1	46891	Technic Digger Bucket 10 x 19	Dark Bluish Gray

5.3 LEGO® MOC studies

In this folder are stored various studies as LEGO® MOC (alias for *My Own Creation*).

Resources

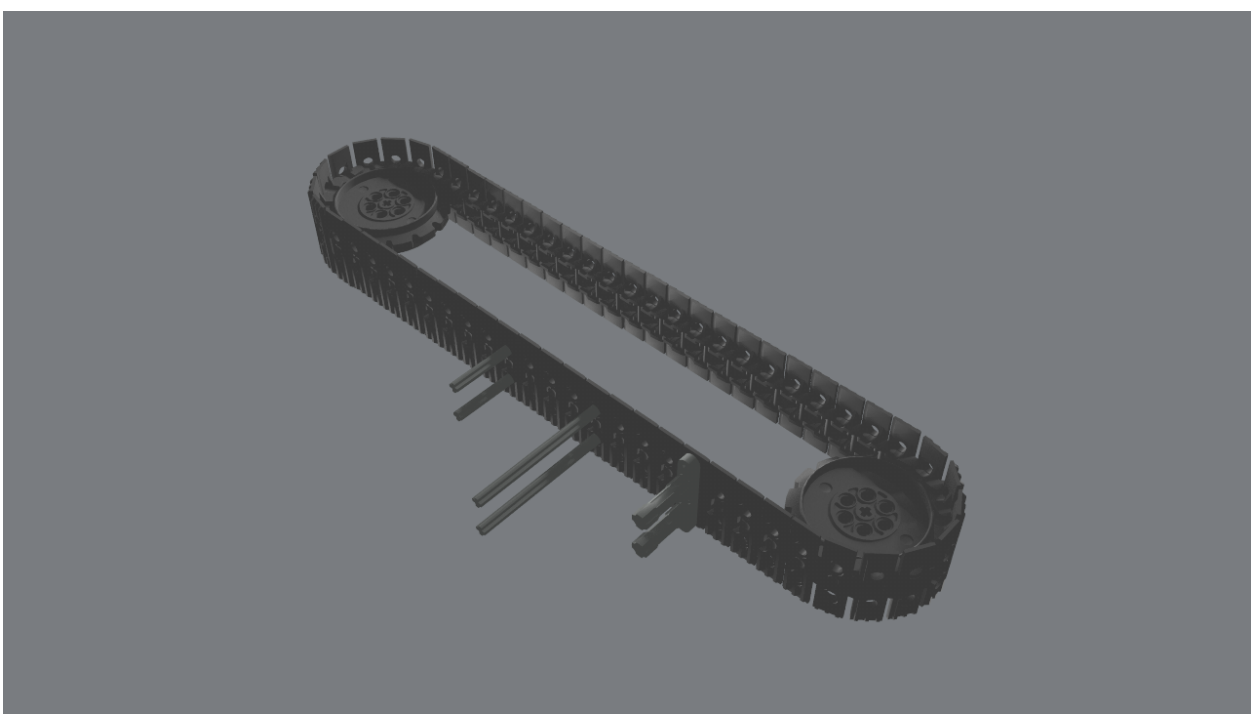
For the realization of the LEGO® project, various software applications and public resources are used :

Resource	Description	Used For
Bricklink Studio	Full featured CAD application	Project design, BOM creation, rendering, instruction manual editing, export to `ldr`
Bricklink PartDesigner	Single LEGO® Part Editor	Creation of parts still not available in *Bricklink Studio*. Correction of parts not working whern exported to `ldr`
LeoCAD	Open Source CAD application	Check correctness of `ldr` files
LDraw	Centralized resources for LEGO® CADs	Reference for getting last *parts* used in *Bricklink PartDesigner*

5.3.1 First Conveyor (v.001)

This is the first idea of a conveyor used to move bottles. The main idea is to use a chain where some kind of *pushers* are used to move bottles over a flat surface.

The chain is not used as a base surface for the bottles for the difficulty of maintaining the chain as an flat orizontal base during the movement.



Files

`Conveyor_001.ldr` : the LEGO project exported in LDraw file format. `Conveyor_001.png` : a rendered version image

5.3.2 Bottle Conveyor (2022-01-22)

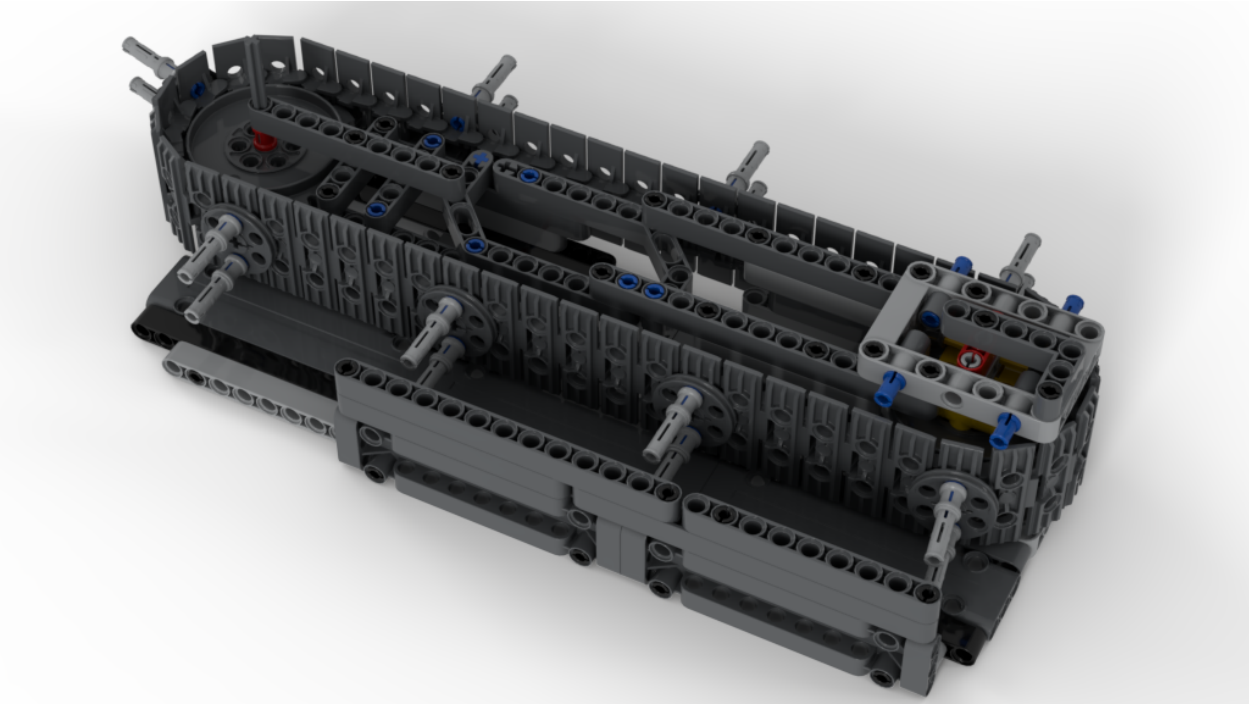
This is the evolution of the **First Conveyor (v.001)** idea.

This project represent the complete structure needed to sustain the chain.

The parts used in this design are all from the LEGO® 42100 Liebherr R9800 set.

Main features of this design are:

- smooth bottle sliding surface
- adjustable chain tensioner
- easily reconfigurable bottle pushers



Files

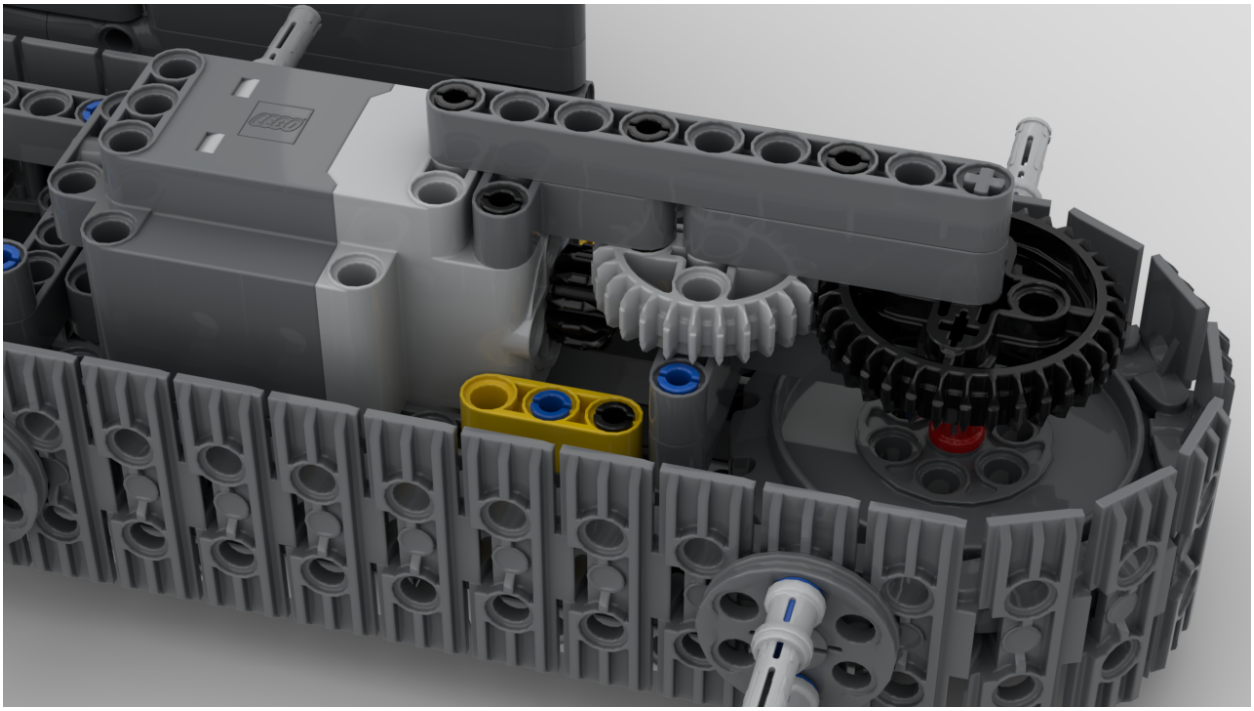
`bottle-conveyor-20220122.io` : the LEGO project in *Bricklink Studio* CAD. `bottle-conveyor-20220122.ldr` : the LEGO project exported in LDraw file format. `bottle-conveyor-20220122.png` : a rendered version image `bottle-conveyor-20220122.pdf` : building instruction manual

5.3.3 Motorized Bottle Conveyor (2022-02-06)

Second evolution of the conveyor idea.

Now a motor and relative gearing is applied to the driving sprocket wheel.

The parts used in this design still are only from the LEGO® 42100 Liebherr R9800 set.



Files

`bottle-conveyor-20220206.io` : the LEGO project in *Bricklink Studio* CAD. `bottle-conveyor-20220206.ldr` : the LEGO project exported in LDraw file format. NOTE: some parts are missing as not ufficially published. `bottle-conveyor-20220206.png` : a rendered version image `bottle-conveyor-20220206.pdf` : building instruction manual

5.4 OAK-D-Lite LEGO® mechanical adapter

Main goal of this project is to construct a mechanical support for the OAK-D-Lite camera, by using only LEGO® Technic® parts.

5.4.1 Main features

- Light and simple frame
- Leaves the dissipation surface uncovered
- Stable fastening
- Quick and easy way to dis/assemble camera from/to the adapter
- Easy access to the USB port

5.4.2 Project Files

- `oak-d-lite-support.io`: the LEGO project in *Bricklink Studio* CAD
- `oak-d-lite-support.ldr`: the LEGO project exported in LDraw file format
- `oak-d-lite-support.png`: a rendered version image
- `oak-d-lite-support.pdf`: building instruction manual

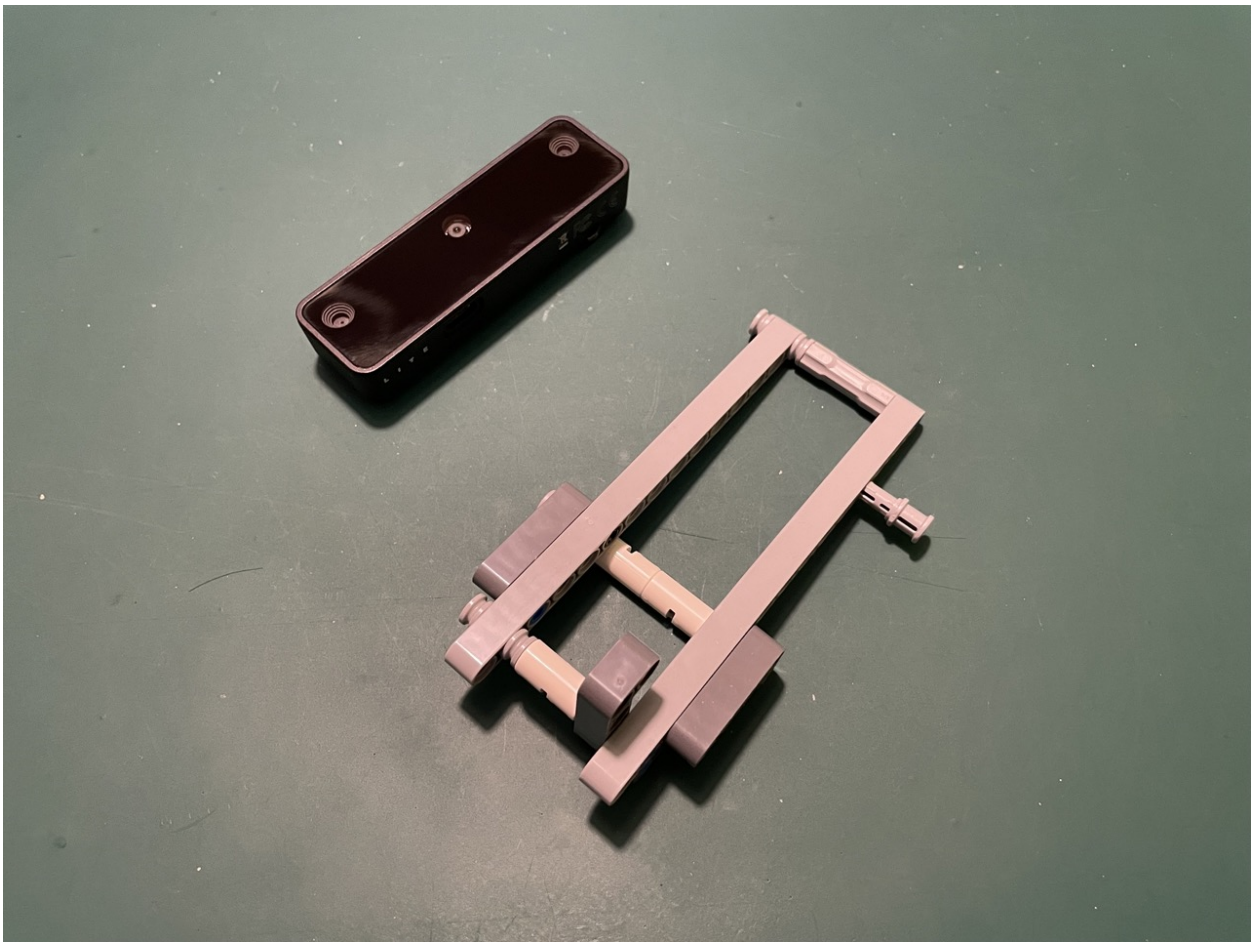
NOTE: For the realization of the LEGO® project, various software applications and public resources are used. See here for details.

5.4.3 License

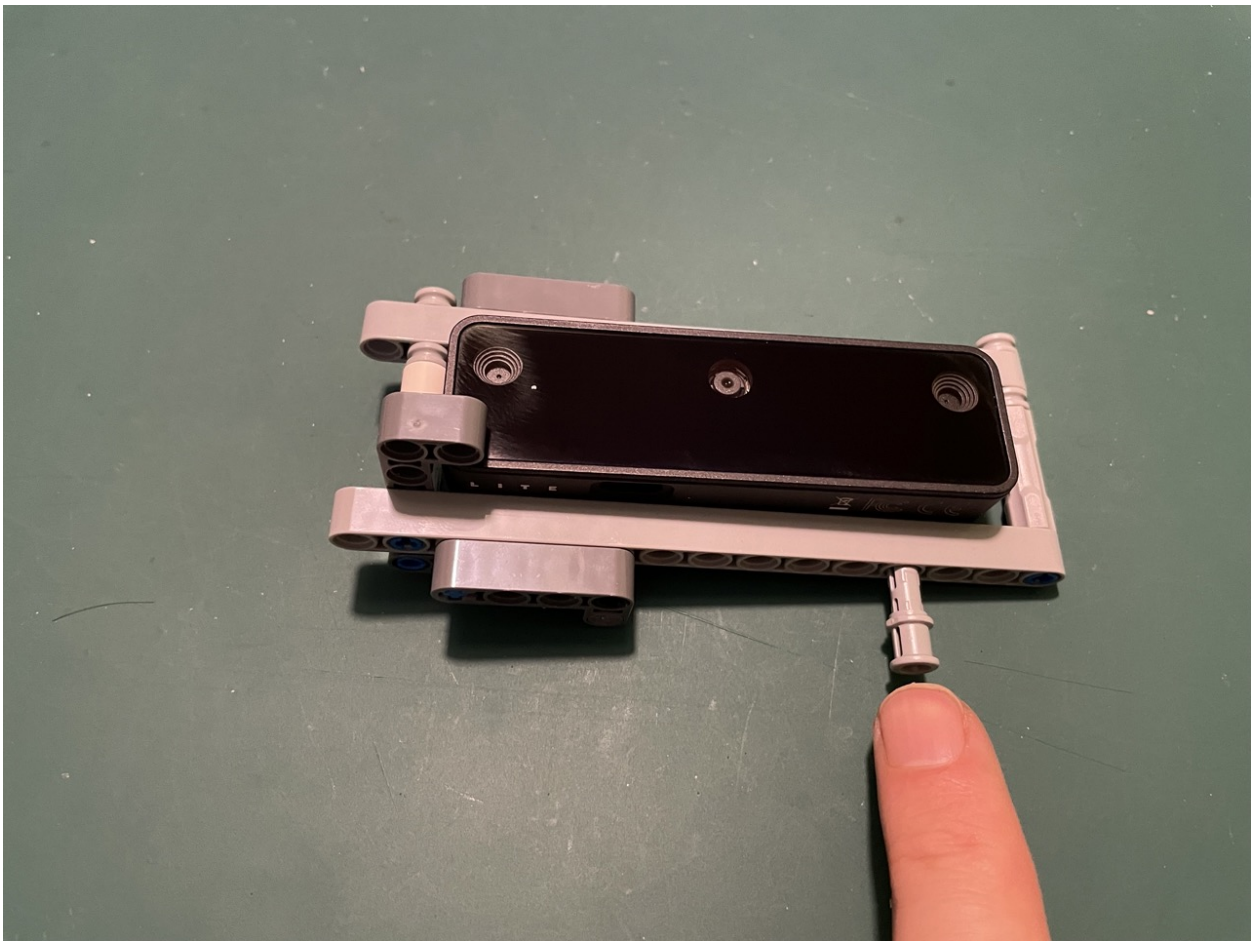
All file, photos, as well as all the ARNEIS project, is released under the MIT License.

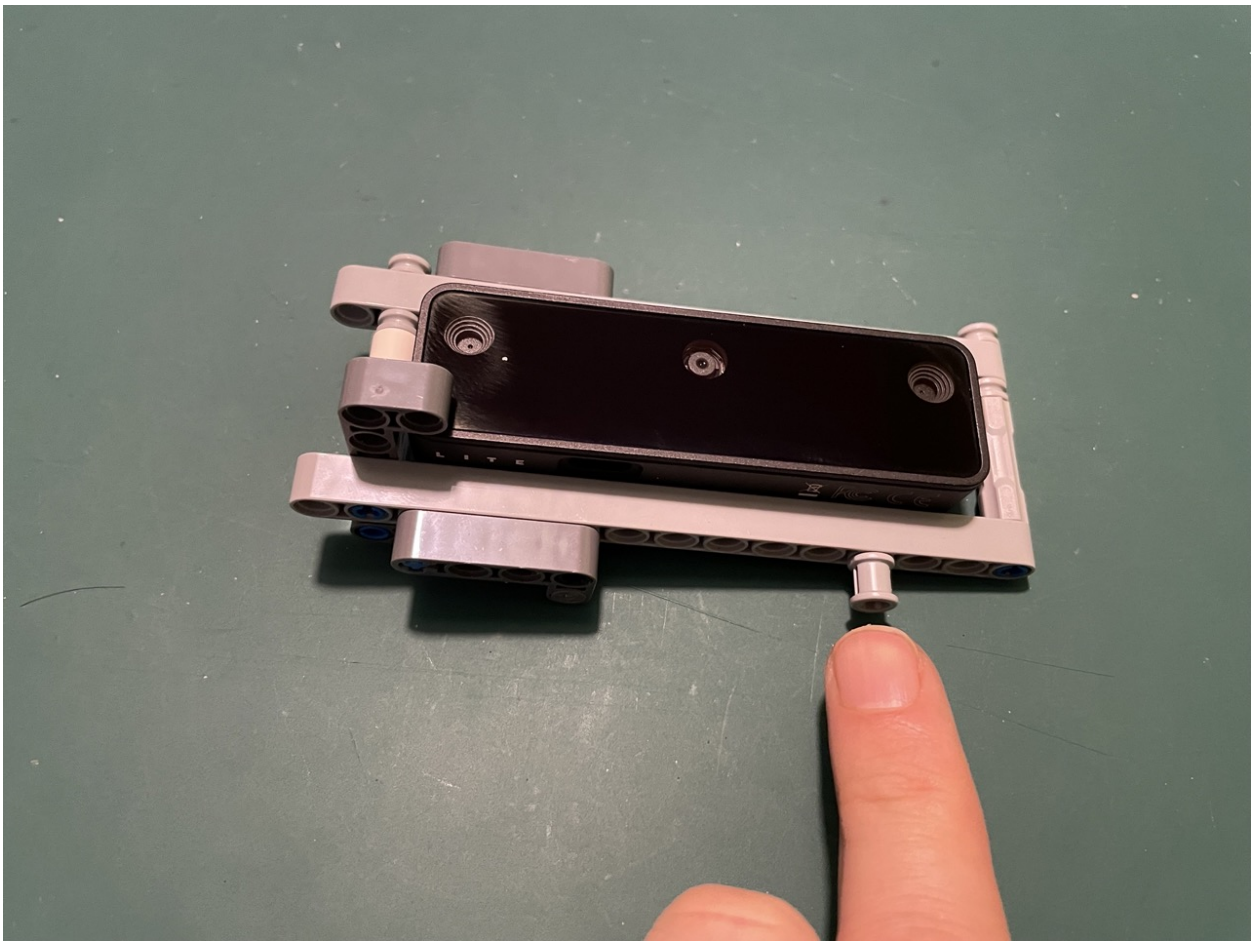
5.4.4 Photo gallery

Here is the adapter and the OAK-D-Lite ready to be assembled

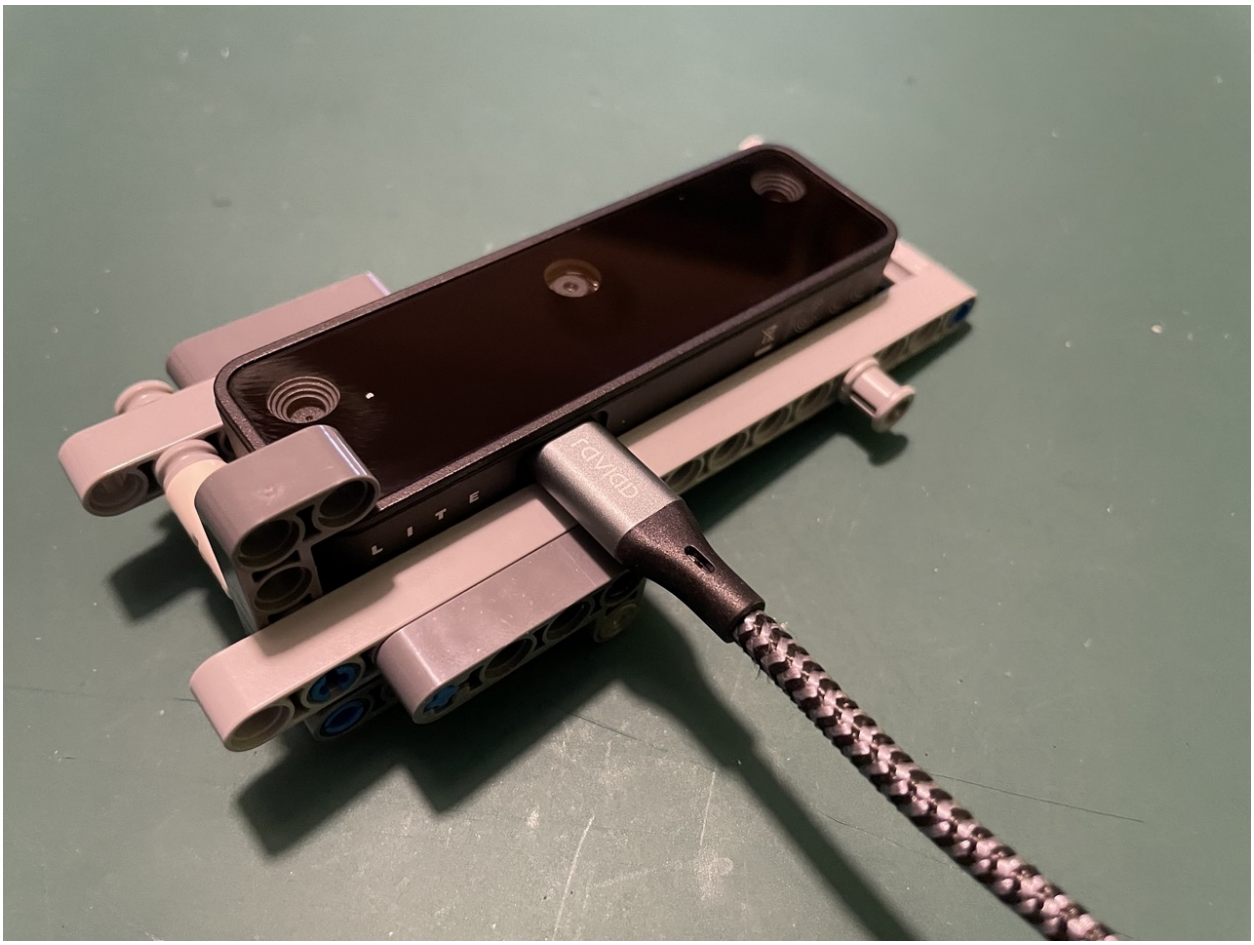


Just push this pin to lock the camera into the adapter

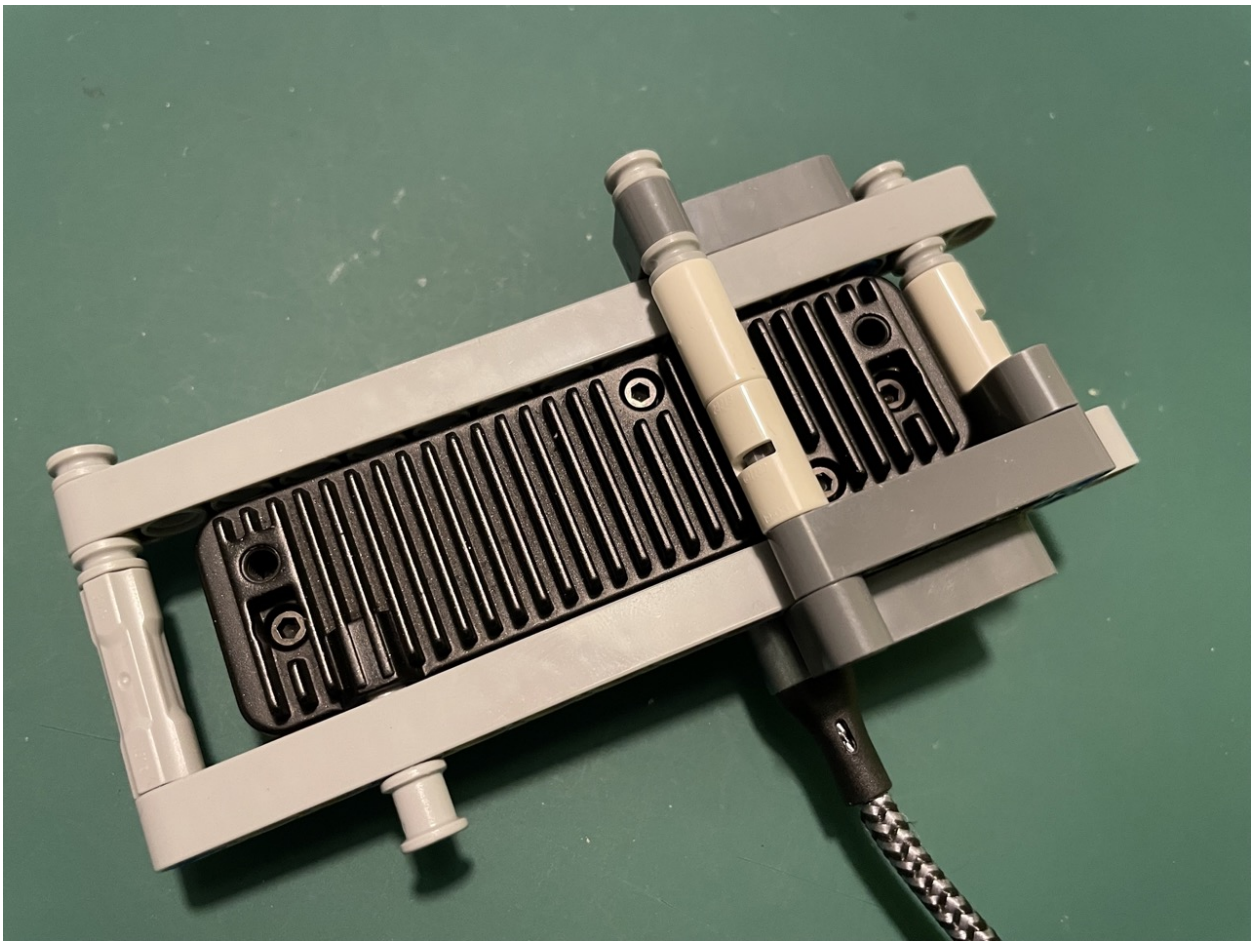




Here is the camera with the USB cable inserted



As you can see all the dissipation surface is free to cool



5.5 LEGO® MOCs Work in Progress

Here are some physical assembly based on the *MOC studies*.

5.5.1 First conveyor idea

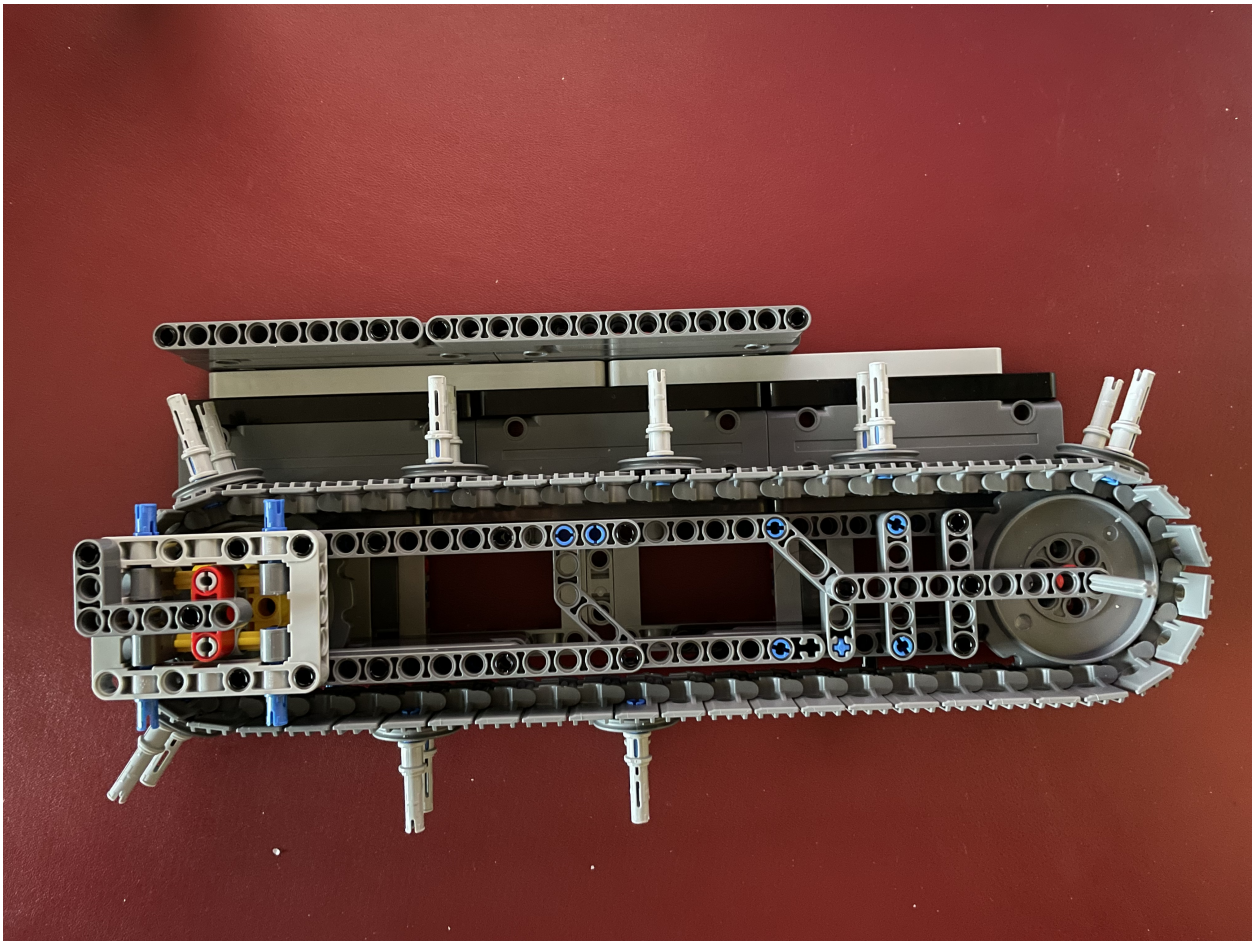
This shows the basic idea used for the movement of miniature bottles.

The bottles slide on a smooth surface and are pushed by small pushers mounted on a track chain mounted transversely.

To adjust the correct chain tension, one of the two sprockets is mounted on a support slide that can be suitably adjusted before being fixed in the best position suitable for correct tensioning.

Here is the conveyor still with no motor

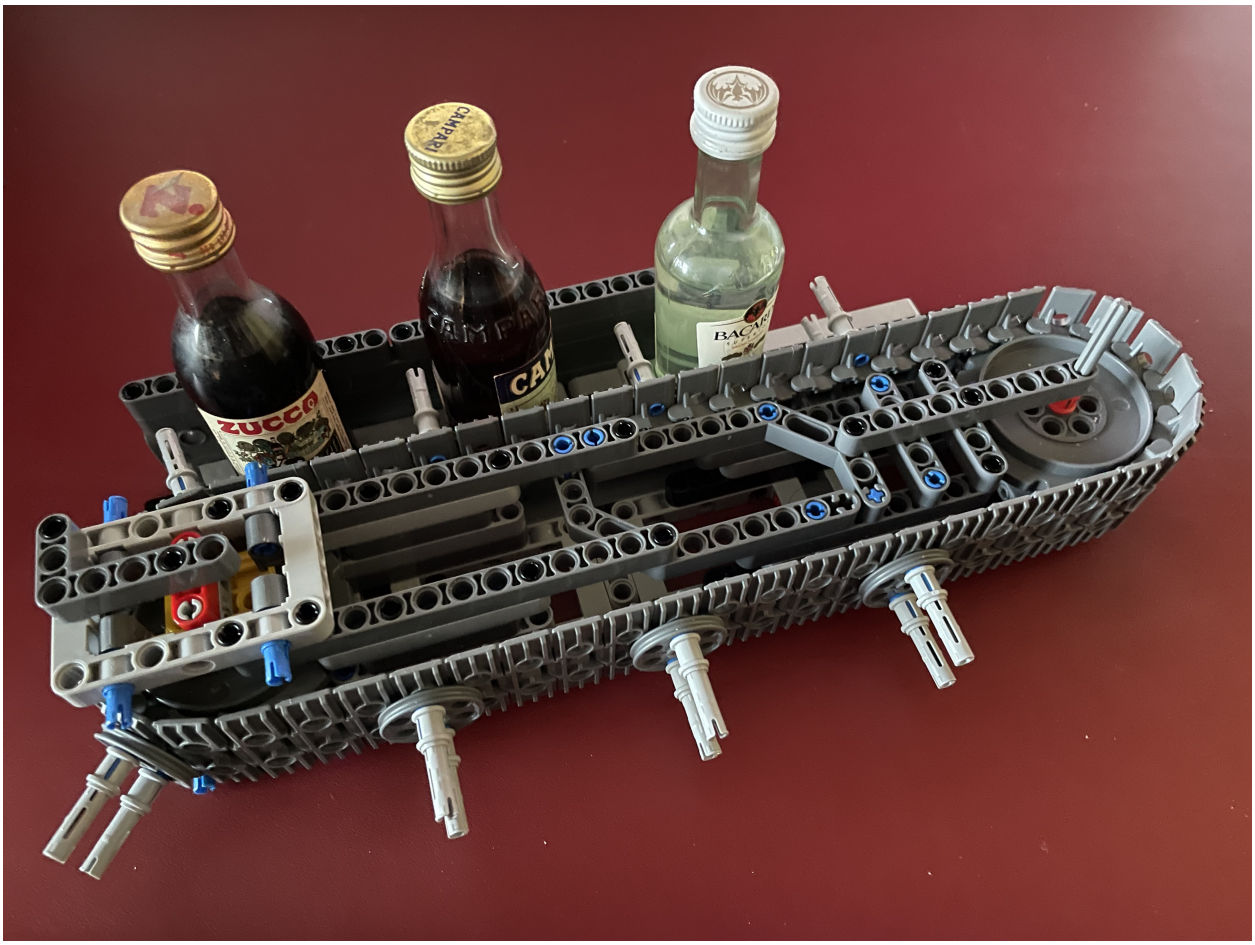
On left side of the image is visible the support slide of the sprocket (yellow and red parts). The right sprocket will be used to pull the chain by a gear train driven by a motor.



And with some mignon bottles

A small part of the route has been prepared for the passage of the bottles.

The bottles lean against a small contrasting wall which creates a path for their guided movement.



And some manual movement

Here are some manual operations to verify the overall idea of the movement of the bottles in the layout.

5.5.2 Motorizing the conveyor and taking some bottles snapshots

Next step in conveyor development is the motorization of the movement.

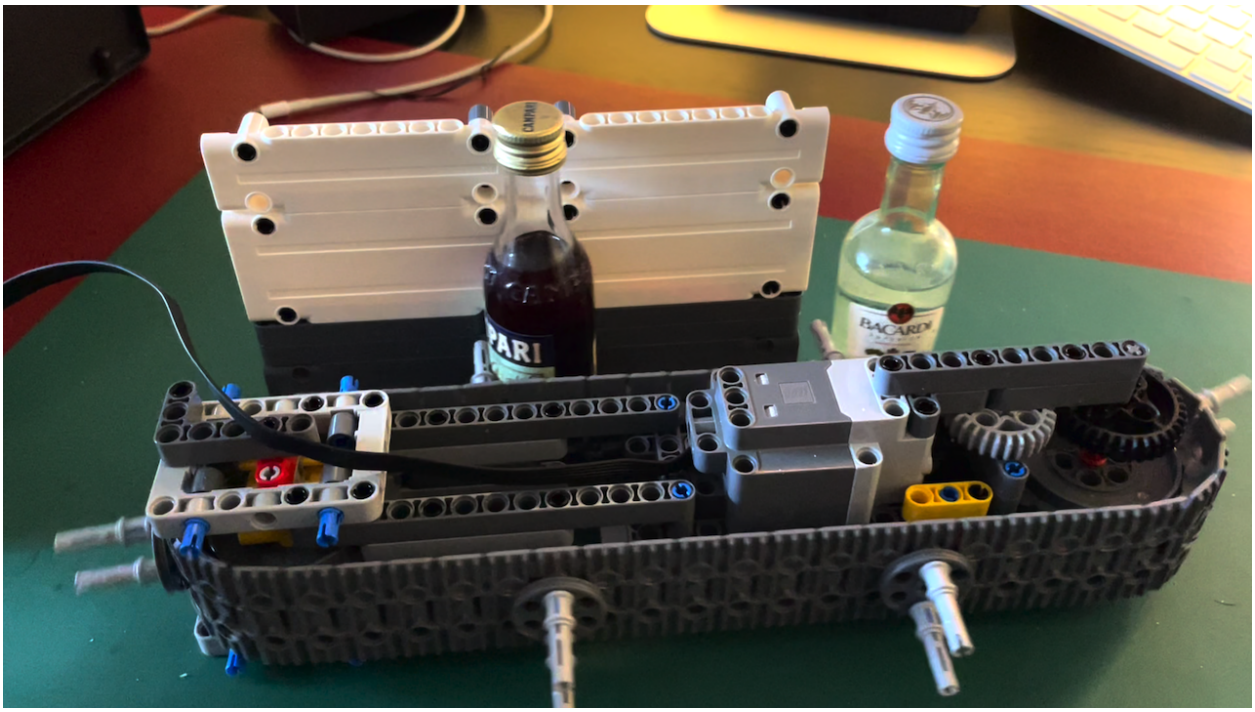
The main improvements from the previous version of the project are the following:

- added a gear train and a motor to drive the conveyor
- introduced a white background in the area used to take bottles snapshots
- added more space between the bottles

The motor and the white backplane

The motor is integrated inside the main structure of the conveyor for a reduced space impact on the project.

The white background may facilitate the contrast in the picture of the bottle's neck.

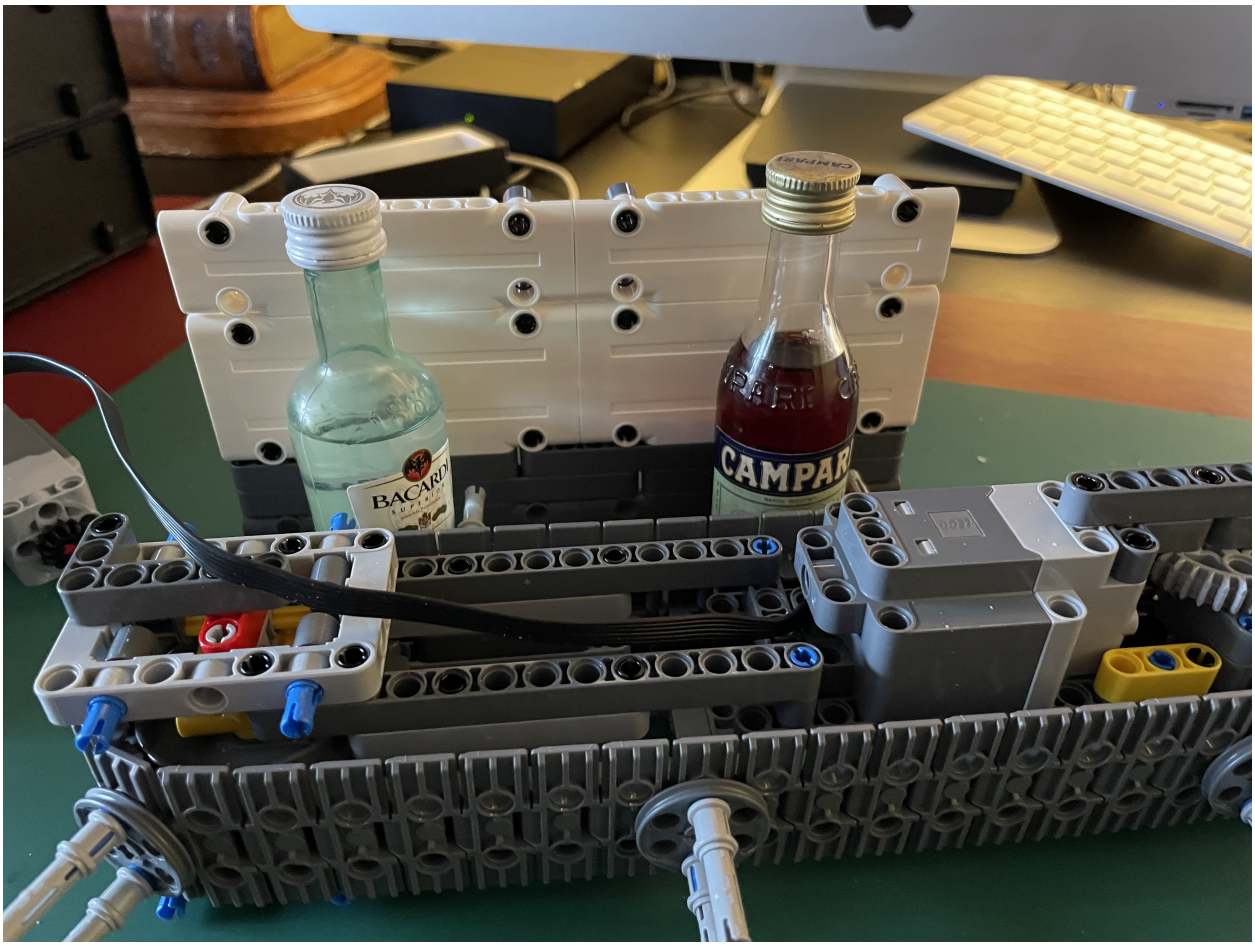


Conveyor movement in all its glory

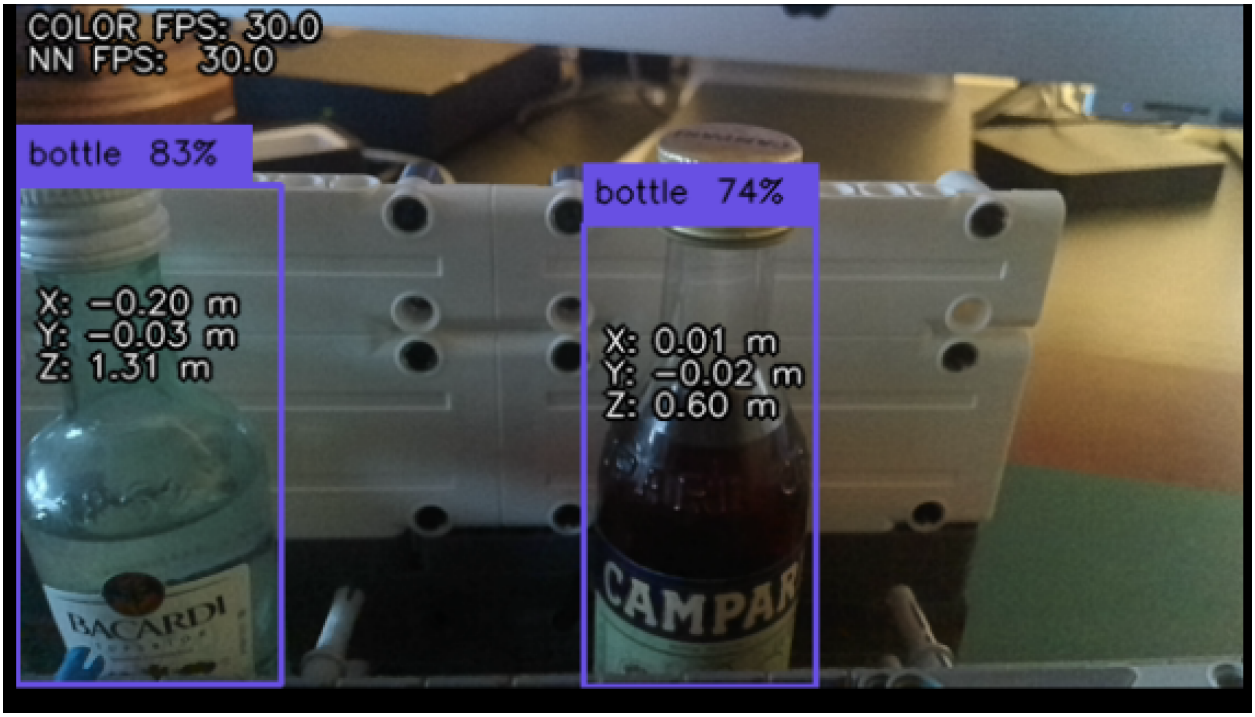
The motor used is an Electric, Motor Powered Up, XL driven by a Electric Battery Box Powered Up Bluetooth Hub (out of the image)

Bottles in pose and the resulting image

Two mignon bottles in pose for a good snapshot…



… and the resulting picture from a OAK-D-Lite camera



5.6 LEGO® ARNEIS Project

Here is the LEGO® MOC (alias for *My Own Creation*) used for the ARNEIS project.

Resources

Various software applications and public resources have been used for the realization of this LEGO® project:

Resource	Description	Used For
Bricklink Studio	Full featured CAD application	Project design, BOM creation, rendering, instruction manual editing, export to `ldr`
Bricklink PartDesigner	Single LEGO® Part Editor	Creation of parts still not available in *Bricklink Studio*. Correction of parts not working whern exported to `ldr`
LeoCAD	Open Source CAD application	Check correctness of `ldr` files
LDraw	Centralized resources for LEGO® CADs	Reference for getting last *parts* used in *Bricklink PartDesigner*

5.6.1 ARNEIS Conveyor

This conveyor is designed to move "mignon" (miniature) bottles. The main idea is to use a chain where some kind of *pushers* are used to move bottles over a flat surface.

The chain is not used as a base surface for the bottles due to the difficulty of ensuring that it will stay horizontal during the movement.

A special support is designed to hold the OAK-D-Lite AI Camera used in this project. It is possible to change the tilt of the camera by using the dedicated linear extender arm.

The white wall acts as a background for taking better snapshots of the bottles during their trip on the conveyor.

Driving the conveyor

In order to verify the correct operation of the conveyor a small MicroPython program has been developed.

Navigate to the Pybricks home page and follow the instructions to activate the IDE for your browser and your operating system.

In this project two motors are used:

- 1x Technic Motor Powered Up XL (Item no. bb0960c01): used to drive the conveyor
- 1x Technic Motor Power UP L (Item no. bb0959c01): used as a *Speed Dial*

With the **XL** motor connected to the **Port B** of the Technic Hub and the **L** motor connected to the **Port A**, use the following source code to program the Hub:

```
from pybricks.pupdevices import Motor
from pybricks.parameters import Port, Stop
from pybricks.tools import wait

# We'll use two motors. One is a dial
# to set the speed of the other motor.
motor = Motor(Port.B)
dial = Motor(Port.A)

# Say hello :)
print("Hello, Pybricks!")

# First, we'll move the dial to zero.
dial.run_target(500, 0, Stop.COAST)

while True:
    # Set the speed based on dial angle
    speed = dial.angle()*3
    if abs(speed) < 50:
        speed = 0

    # Run motor at desired speed
    motor.run(speed)

    angle = dial.angle()
    print("angle=" + str(angle) + "    ", end='\r')
```

(continues on next page)

(continued from previous page)

```
# Wait briefly, then repeat
wait(10)
```

You may adjust the speed and direction of the conveyor by rotating the dial (clockwise or counterclockwise).

Project resources

Files

`arneis-conveyor-oak-d-lite.io` : the LEGO project in *Bricklink Studio* CAD. `arneis-conveyor-oak-d-lite.ldr` : the LEGO project exported in LDraw file format. `arneis-conveyor-oak-d-lite.png` : a 3D rendered version image. `arneis-conveyor-20220312.gif` : a 3D rendered video of rotating MOC. `arneis-conveyor-20220312.pdf` : building instruction manual. `arneis-conveyor-oak-d-lite-build.mp4` : a 3D rendered video of building MOC.

CHAPTER

SIX

INDICES AND TABLES

- genindex
- search