

Hands On: Containerizing Python and Deploying to IoT and Edge Devices



PyCon Iran 2022



David Tischler
Developer Advocate
t: @balena_io
e: david@balena.io



Agenda

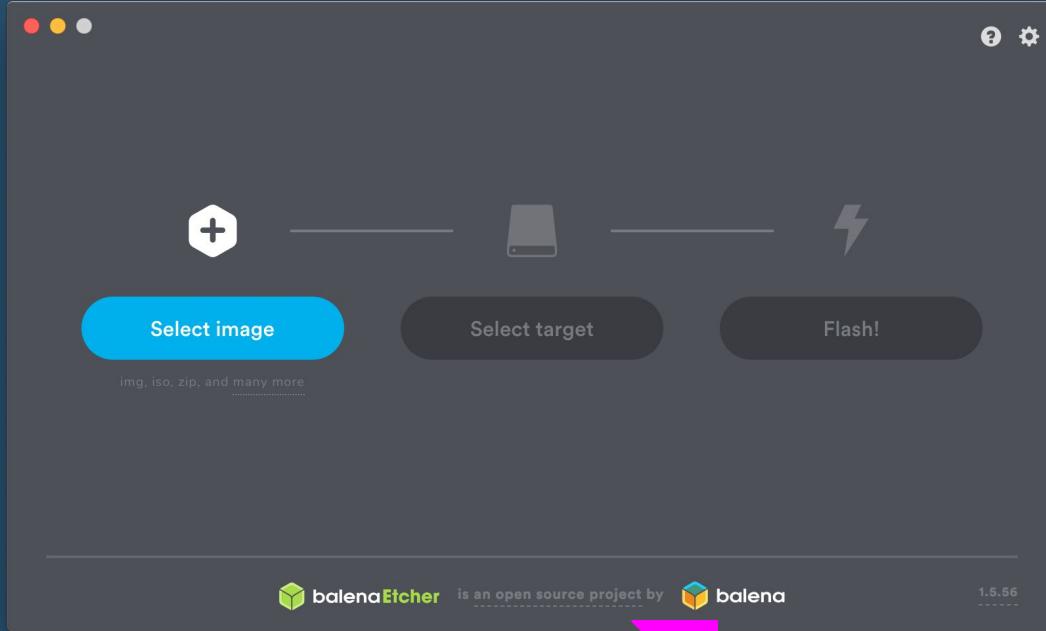
- Introduction and Background
- Building a (very) simple Python application on a Raspberry Pi
- What is a Container?
- Setting up a Balena Device
- Creating a Container
- Load the Container onto a Balena Device
- Make Changes, Redeploy
- Upgrade Hardware, Redeploy
- Push a Container Halfway Around the World



Who is Balena?



Etcher, anyone?



balena

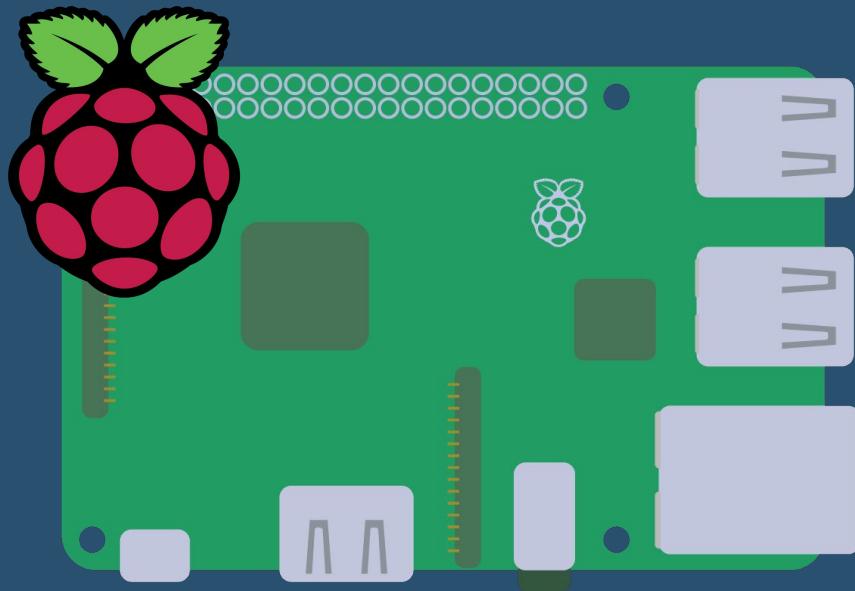
“Platform built to develop, deploy, and manage fleets of connected devices at scale.”

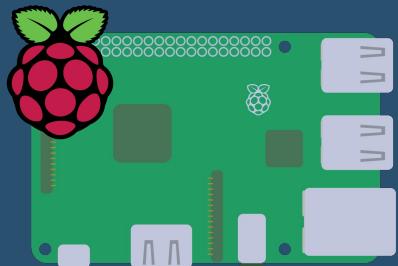
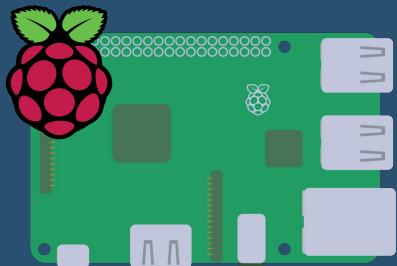
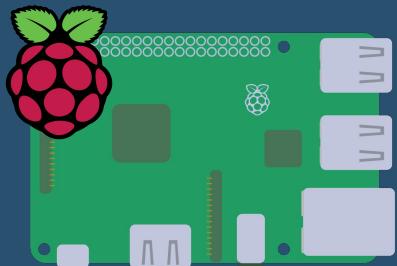
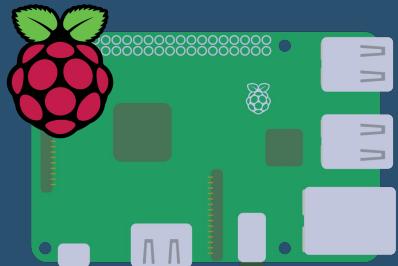
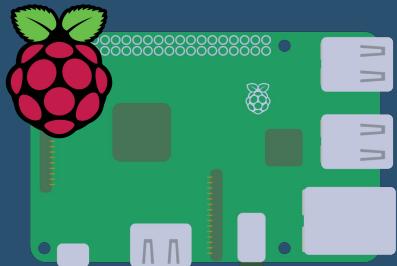
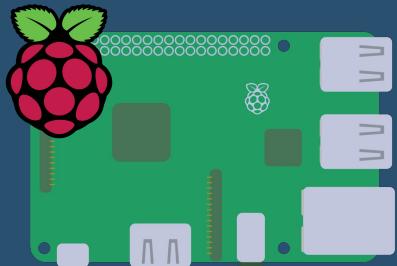
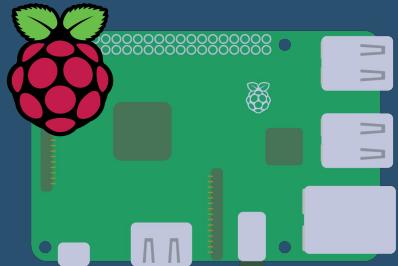
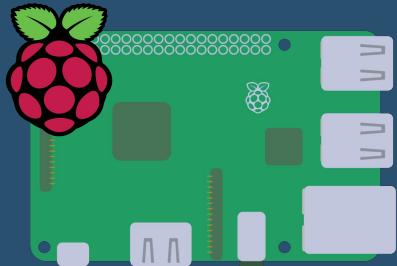
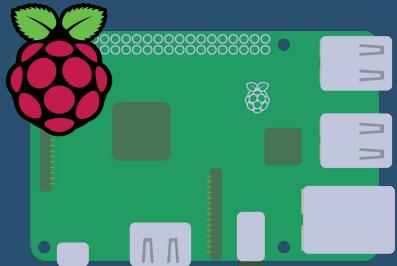
- balenaCloud
 - Web-based control of IoT devices, broken down by Applications, Devices, Workloads, and more
 - Terminal access to device, logs, container build info, and more.
- balenaOS
 - Based on Yocto Linux
- balenaEngine
 - Container runtime

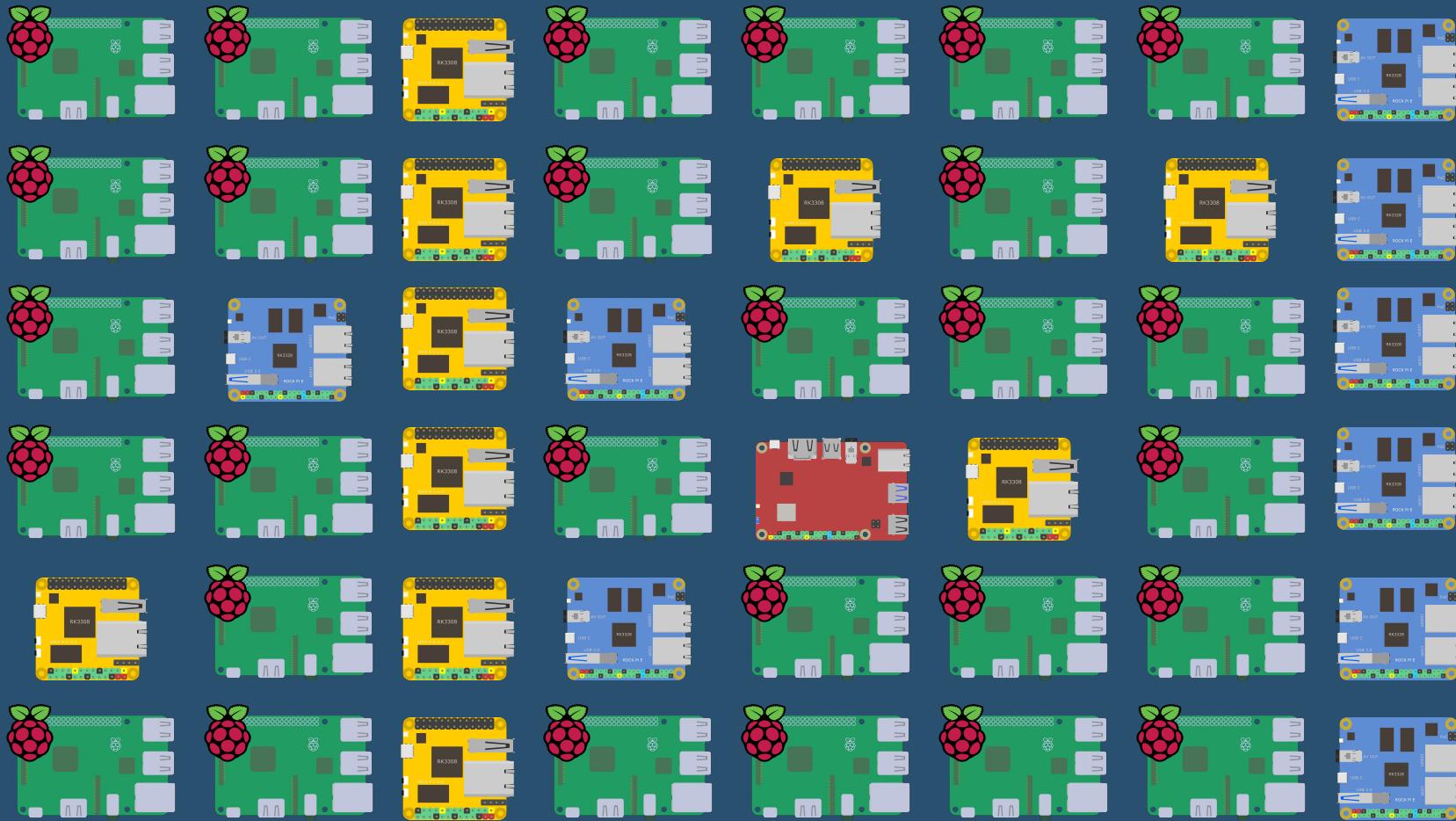


What do we mean by IoT
“Fleet Management”?











Architecture



balenaOS

Built on Yocto Linux, Optimized for
Embedded Devices and IoT Usage

- Minimal by design (increased security)
- Less RAM / resource requirements
- Redundant RootFS partitions
- More tolerant of power loss
- Minimized SD Card writes



balenaEngine

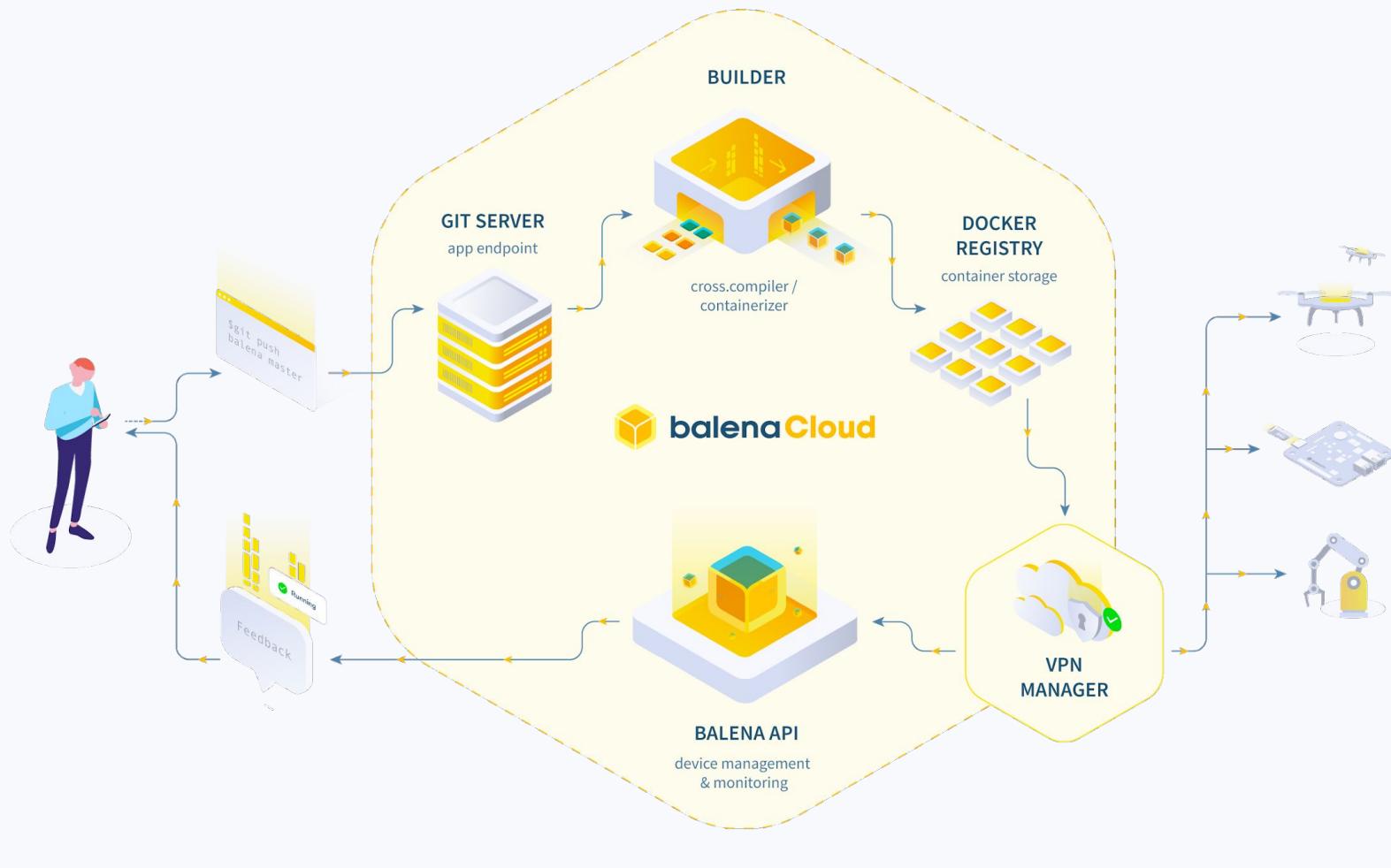
Container engine built for Embedded
and IoT use-cases, based on the Moby
Project from Docker

- Minimal by design
- Less RAM / resource requirements
- Delta pulls to save bandwidth
- Fault tolerance on container pulls



balenaEngine





Let's Begin



A Simple IoT Device



+



<https://pypi.org/project/RPi.bme280/>

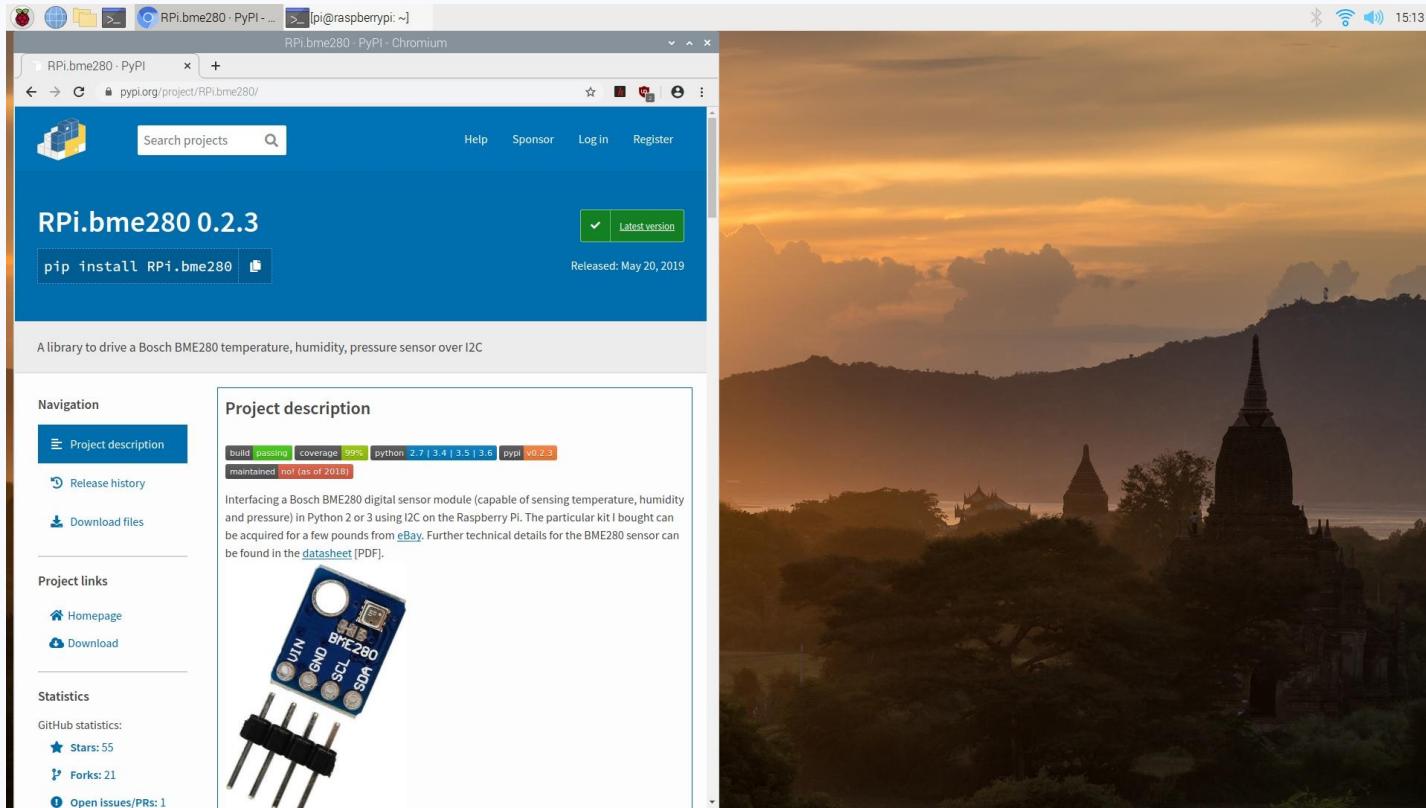
The screenshot shows the PyPI project page for `RPi.bme280`. The title bar indicates the page is for `RPi.bme280 · PyPI`. The main header features the project name `RPi.bme280 0.2.3` in white text on a dark blue background. Below the title is a button with the command `pip install RPi.bme280`. To the right, there is a green button labeled "Latest version" with a checkmark icon. A release date of "Released: May 20, 2019" is also displayed. The main content area contains a brief description: "A library to drive a Bosch BME280 temperature, humidity, pressure sensor over I2C". On the left, a sidebar titled "Navigation" lists "Project description" (which is currently selected), "Release history", and "Download files". Below this is a section titled "Project links" with "Homepage" and "Download" options. The "Statistics" section shows GitHub metrics: 54 stars, 21 forks, and 1 open issue/PR. The central "Project description" section includes a badge for "build passing", coverage at 99%, and compatibility with Python 2.7 | 3.4 | 3.5 | 3.6, along with a note about PyPI version v0.2.3 and maintenance status (not maintained as of 2018). It also describes the BME280 sensor and its use with the Raspberry Pi. An image of the BME280 sensor module is shown, along with a caption "GPIO pins out".



Starting with Raspbian



<https://pypi.org/project/RPi.bme280/>



A library to drive a Bosch BME280 temperature, humidity, pressure sensor over I2C

RPi.bme280 0.2.3

pip install RPi.bme280

Released: May 20, 2019

Project description

build passing coverage 99% python 2.7 | 3.4 | 3.5 | 3.6 | pypi v0.2.3
maintained n/a (as of 2018)

Interfacing a Bosch BME280 digital sensor module (capable of sensing temperature, humidity and pressure) in Python 2 or 3 using I2C on the Raspberry Pi. The particular kit I bought can be acquired for a few pounds from [eBay](#). Further technical details for the BME280 sensor can be found in the [datasheet](#) [PDF].



Install the easy way...

The image shows a Raspberry Pi desktop interface with two open windows.

The left window is a Chromium browser displaying the PyPI project page for RPi.bme280. It features a "Cheeseshop install" section with the command `$ sudo pip install RPi.bme280`. Below this is a "Software Driver - Example Usage" section containing Python code:

```
import smbus2
import bme280

port = 1
address = 0x76
bus = smbus2.SMBus(port)

calibration_params = bme280.load_calibration_params(bus, address)

# the sample method will take a single reading and return a
# compensated_reading object
data = bme280.sample(bus, address, calibration_params)

# the compensated_reading class has the following attributes
print(data.id)
print(data.timestamp)
print(data.temperature)
print(data.pressure)
print(data.humidity)

# there is a handy string representation too
print(data)
```

The right window is a terminal window titled "pi@raspberrypi: ~". It shows the output of the pip installation command:

```
pi@raspberrypi: ~ pip install RPi.bme280
Collecting RPi.bme280
  Downloading https://files.pythonhosted.org/packages/8a/55/2c738564ce0478052f15d6331759c0c/a4f4d518e8ed4468976deacf42de/RPi.bme280-0.2.3-py2.py3-none-any.whl
Collecting smbus2 (from RPi.bme280)
  Downloading https://www.piwheels.org/simple/smbus2/smbus2-0.3.0-py2.py3-none-any.whl
Installing collected packages: smbus2, RPi.bme280
Successfully installed RPi.bme280-0.2.3 smbus2-0.3.0
pi@raspberrypi: ~
```

A small green cube icon is located in the bottom right corner of the desktop.

Sample code is provided

The image shows a dual-pane view on a Raspberry Pi desktop. On the left, a Chromium browser window displays the PyPI page for RPi.bme280. It includes a terminal-like input field with the command \$ sudo pip install RPi.bme280. Below this, a "Software Driver - Example Usage" section shows sample Python code. On the right, a terminal window titled "pi@raspberrypi: ~" shows the same Python code running. The terminal has a sunset background and includes a menu bar with File, Edit, Tabs, Help, and a status bar showing the date and time.

\$ sudo pip install RPi.bme280

Software Driver - Example Usage

Once installed, confirm the I2C address (see prerequisites, it will most likely be 0x76 or 0x77) and port.

Then in a python script or REPL:

```
import smbus
import bme280

port = 1
address = 0x76
bus = smbus.SMBus(port)

calibration_params = bme280.load_calibration_params(bus, address)

# the sample method will take a single reading and return a
# compensated_reading object
data = bme280.sample(bus, address, calibration_params)

# the compensated_reading class has the following attributes
print(data.id)
print(data.timestamp)
print(data.temperature)
print(data.pressure)
print(data.humidity)

# there is a handy string representation too
print(data)
```

This then should print something like:

```
ee50df9c-3aa3-4772-8767-73b6bb74f30f
2016-11-18 17:33:28.937863
20.563
980.91
48.41
compensated_reading(id=ee50df9c-3aa3-4772-8767-73b6bb74f30f,
```



It works!

The image shows a Raspberry Pi desktop environment with a terminal window and a Chromium browser window.

Chromium Browser: The title bar says "RPI.bme280 - PyPi - pi@raspberrypi: ~". The page content is from pypi.org/project/RPi.bme280/. It displays the following text:

Once installed, confirm the I2C address (see prerequisites, it will most likely be 0x76 or 0x77) and port.

Then in a python script or REPL:

```
import smbus
import bme280

port = 1
address = 0x76
bus = smbus.SMBus(port)

calibration_params = bme280.load_calibration_params(bus, address)

# the sample method will take a single reading and return a
# compensated_reading object
data = bme280.sample(bus, address, calibration_params)

# the compensated_reading class has the following attributes
print(data.id)
print(data.timestamp)
print(data.temperature)
print(data.pressure)
print(data.humidity)

# there is a handy string representation too
print(data)
```

This then should print something like:

```
ee50df9c-3aa3-4772-8767-73b6bb74f30f
2016-11-18 17:33:28.937863
20.563
980.91
48.41
compensated_reading(id=ee50df9c-3aa3-4772-8767-73b6bb74f30f,
    timestamp=2016-11-18 17:33:28.937863, temp=20.563 °C,
    pressure=980.91 hPa, humidity=48.41 % RH)
```

Terminal Window: The title bar says "pi@raspberrypi: ~". The terminal shows the following output:

```
pi@raspberrypi:~$ pip install RPi.bme280
Collecting RPi.bme280
  Downloading https://files.pythonhosted.org/packages/8a/55/2c738564ce0478052f15d6331759c0c/a4f4d5151e8e4d468976deacf42de/RPi.bme280-0.2.3-py3-none-any.whl
Collecting smbus2 (from RPi.bme280)
  Downloading https://www.piwheels.org/simple/smbus2/smbus2-0.3.0-py2.py3-none-any.whl
Installing collected packages: smbus2, RPi.bme280
Successfully installed RPi.bme280-0.2.3 smbus2-0.3.0
pi@raspberrypi:~$ i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -
10: -
20: -
30: -
40: -
50: -
60: -
70: -- -- -- -- 76 --
pi@raspberrypi:~$ nano 280.py
pi@raspberrypi:~$ python 280.py
3cc941b3-6061-40c8-93ca-3bb602708af4
2020-09-30 15:12:37.440184
29.8143812565
956.639394945
34.2555253018
compensated_reading(id=3cc941b3-6061-40c8-93ca-3bb602708af4, timestamp=2020-09-30 15:12:37.440184, temp=29.824 °C, pressure=956.64 hPa, humidity=34.26 % RH)
pi@raspberrypi:~$
```



Now, how do I build 500 of
these?



Let's try it again, but with
balena



balena - Home

https://www.balena.io

What is balena? balenaCloud More products Resources Pricing Customers About Login Sign up

Build your IoT project with balena

The infrastructure you need to develop, deploy, and manage fleets of connected devices at scale.

Learn more

balenaCloud

Your first 10 devices are always free and full-featured.

Email

Get started

Balena is for fleet owners



balenaCloud | Signup

https://dashboard.balena-cloud.com/signup

Your first 10 devices are always free
Upgrade anytime. Hassle free.

 Sign up with Github  Sign up with Google

or sign up with

Email*

Password*

Send me the balena monthly newsletter
We send one newsletter a month, and do not share your e-mail with third parties

Sign up

Already have an account? [Log in](#)

By clicking "Sign up" I agree to balena's [Terms of Service](#)
As part of delivering this service, we collect certain account data from you. Please review our [Privacy Policy](#) for more information.

Need help?



The screenshot shows the balenaCloud dashboard interface. At the top, there are standard browser controls (red, yellow, green, close, back, forward) and a URL bar displaying "dashboard.balena-cloud.com". To the right of the URL bar are download, upload, and refresh icons. The main header includes the balenaCloud logo, navigation links for "Getting Started", "Docs", "Forums", "Status", and a user profile for "David T". A dropdown menu is open next to the user profile.

The left sidebar contains a navigation menu with the following items:

- Organizations
- PyCon (selected)
- Summary (highlighted in blue)
- Fleets
- Members
- Teams
- Billing

The main content area features a large central button labeled "Create fleet" with a gear icon. Below this button, a dashed arrow points from the text "You don't have any items yet." to the "Create fleet" button. The text "How about adding one?" is displayed below the arrow. In the bottom right corner of the main content area, there is a small blue button with a question mark icon labeled "Need help ?".

At the bottom left of the sidebar, the text "v9.2.0" is visible. The bottom right corner of the entire window has a small blue footer element containing a yellow and green geometric logo.



A screenshot of the balenaCloud dashboard interface. The URL in the browser bar is `dashboard.balena-cloud.com`. The main navigation bar includes links for "Getting Started", "Docs", "Forums", "Status", and a user profile for "David T". On the left, a sidebar menu lists "Organizations", "PyCon", "Fleets", and the selected "PyConiran" project. The "Devices" section is currently active. The main content area displays the "PyConiran" project summary, which includes:

- Architecture:** armv7hf
- Slug:** pycon/pyconiran
- Created:** Jan 31st 2022, 3:08 pm
- Status:** Starter

The dashboard features three main cards:

- Devices:** 0. A message says "You don't have any devices yet. How about adding one?" with a blue "+ Add device" button.
- Releases:** 0. A message says "You don't have any releases yet. How about creating one?" with a blue "Create release" button.

At the bottom right, there's a "Need help?" button with a question mark icon. The footer shows the version "v9.2.0".



The screenshot shows the 'Add new device' page on the balenaCloud dashboard. At the top, there are dropdown menus for 'Select device type' (set to 'Raspberry Pi') and 'Select OS type' (set to 'balenaOS'). Below these are dropdowns for 'Select version' (set to 'v2.83.21+rev1 (recommended)') and 'Show outdated versions'. A section for 'Selected edition' shows 'Development' (selected) and 'Recommended for first time users'. A note states: 'Development images should be used when you are developing an application and want to use the fast local mode workflow. This variant should never be used in production.' Another section for 'Production' notes: 'Production images are ready for production deployments, but don't offer easy access for local development.' Under 'Network Connection', 'Ethernet only' is selected. There is also an 'Advanced' button. A warning message in a box states: '⚠ Warning! The Raspberry Pi 3 is not capable of connecting to 5GHz WiFi networks unless you use an external WiFi adapter that supports it. The Raspberry Pi 3 B+ is capable of connecting to both 5GHz and 2.4GHz networks.' At the bottom right is a blue 'Flash' button with a gear icon. The bottom left corner of the window shows 'v9.2.0'.

Add new device

Select device type

Raspberry Pi 3

Select OS type

balenaOS

Select version

v2.83.21+rev1 (recommended)

Show outdated versions

Selected edition

Development Recommended for first time users

Development images should be used when you are developing an application and want to use the fast local mode workflow. This variant should never be used in production.

Production

Production images are ready for production deployments, but don't offer easy access for local development.

Network Connection

Ethernet only

WiFi + Ethernet

+ Advanced

⚠ Warning! The Raspberry Pi 3 is not capable of connecting to 5GHz WiFi networks unless you use an external WiFi adapter that supports it. The Raspberry Pi 3 B+ is capable of connecting to both 5GHz and 2.4GHz networks.

Flash



Flash SD Card with Etcher



balena.io

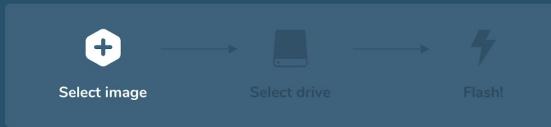
An open source project by  balena | More products ▾

 balenaEtcher

Forums Mailing list Changelog Etcher Pro

Flash. Flawless.

Flash OS images to SD cards & USB drives, safely and easily.



Select image → Select drive → Flash!

[Download for macOS](#)

v1.5.71 [See what's new](#)

 **Validated Flashing**
No more writing images on corrupted cards and wondering why your device

 **Hard Drive Friendly**
Makes drive selection obvious to avoid wiping your entire hard-drive

FEATURES

A better way to burn.





Select image

img, iso, zip, and many more

Select target

Flash!



balenaEtcher

is an open source project by



balena

1.5.56



Power Up!



dashboard.balena-cloud.com

Getting Started Docs Forums Status David T DT

PyConiran

Architecture armv7hf
Slug pycon/pyconiran
Created Jan 31st 2022, 3:08 pm
Starter

Devices 1

Online Config Updating Offline Port prov Inactive

Releases 0

You don't have any releases yet.
How about creating one?

Create release

Add device

| Name | Status | Device type | Last seen | UUID | OS version | OS variant | IP address |
|--------------|----------|----------------|----------------------------|---------|-----------------------|-------------|---------------|
| bold-pancake | ✓ Online | Raspberry Pi 3 | Online (for a few seconds) | bcea9c6 | balenaOS 2.83.21+rev1 | Development | 192.168.0.240 |

v9.2.0

Need help ?

Organizations

PyCon

Fleets

PyConiran

Summary

Devices

Releases

Variables

Configuration

Actions

Settings

Members

Teams



dashboard.balena-cloud.com

Danger! The device is undervolted!

bold-pancake

Reboot | Restart services |

| STATUS | UUID | TYPE |
|--------|---------|----------------|
| Online | bcea9c6 | Raspberry Pi 3 |

| ONLINE FOR | HOST OS VERSION | SUPERVISOR VERSION |
|---------------|--------------------------------------|--------------------|
| a few seconds | balenaOS 2.83.21+rev1 development | 12.10.3 |

| CURRENT RELEASE | TARGET RELEASE |
|-----------------|----------------|
| Factory build | Factory build |

| LOCAL IP ADDRESS | PUBLIC IP ADDRESS | MAC ADDRESS |
|------------------|---|-------------|
| 192.168.0.240 | 2600:8800:b4c8:0:b700:1dfd B8:27:EB:DE:ED:AC B8:27:EB:B8:B8:F9 | |

TAGS (0)
No tags configured yet

NOTES
Add device notes...

CPU ~77% Temperature ~36C

Memory 128 MB/970 MB Storage /dev/mmcblk0p6 69 MB/13.8 GB

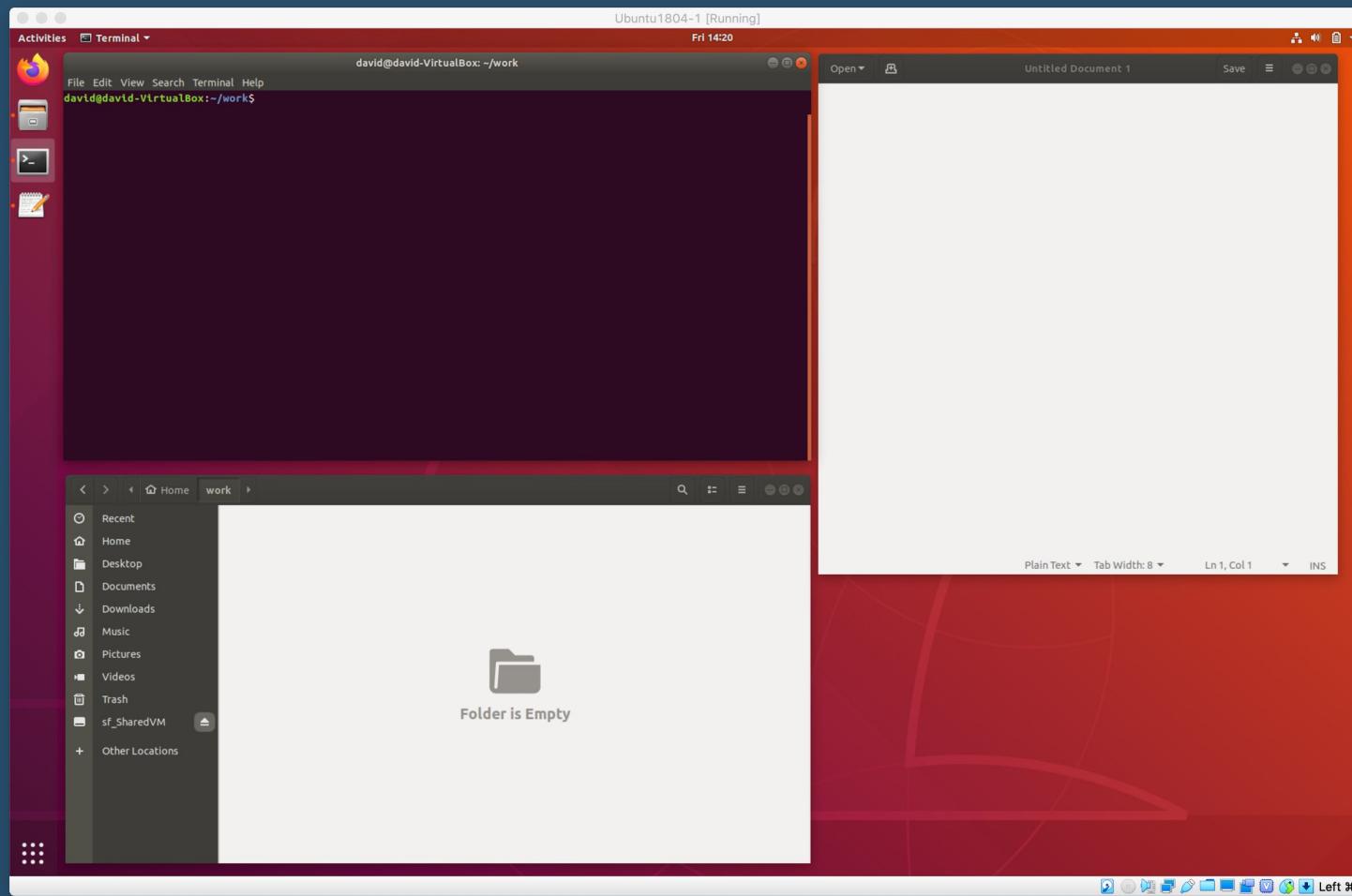
Logs UTC Timestamps
Add filter Search entries... Views
Loading logs...

Terminal Select a target Start terminal session Need help



Great, but where is our
Python?





Ubuntu1804-1 [Running]
Fri 14:24

Activities Text Editor

david@david-VirtualBox: ~/work

File Edit View Search Terminal Help
david@david-VirtualBox:~/work\$

Dockerfile.template

```
FROM balenalib/%{BALENA_MACHINE_NAME%}-python:latest
WORKDIR /usr/src/app
RUN pip install RPL.bme280
COPY . .
CMD [ "python", "./280.py" ]
```

PlainText Tab Width: 8 Ln 9, Col 29 INS

280.py

```
import smbus2
import bme280
port = 1
address = 0x76
bus = smbus2.SMBus(port)

calibration_params = bme280.load_calibration_params(bus, address)
# the sample method will take a single reading and return a
# compensated_reading object
data = bme280.sample(bus, address, calibration_params)

# the compensated_reading class has the following attributes
print(data.id)
print(data.timestamp)
print(data.temperature)
print(data.pressure)
print(data.humidity)

# there is a handy string representation too
print(data)
```

Python Tab Width: 8 Ln 22, Col 12 INS

Home work

Recent Home Desktop Documents Downloads Music Pictures Videos Trash sf_SharedVM Other Locations

280.py Dockerfile.template



Install balena-CLI

Windows, Mac, or Linux

Provides native tooling for:

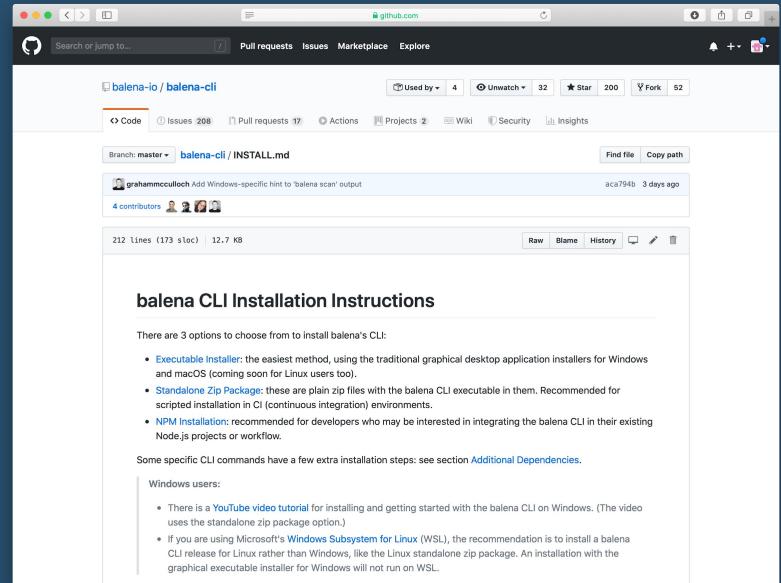
- Device and App Info
- Building, Pushing Containers
- SSH to Devices

Instructions:

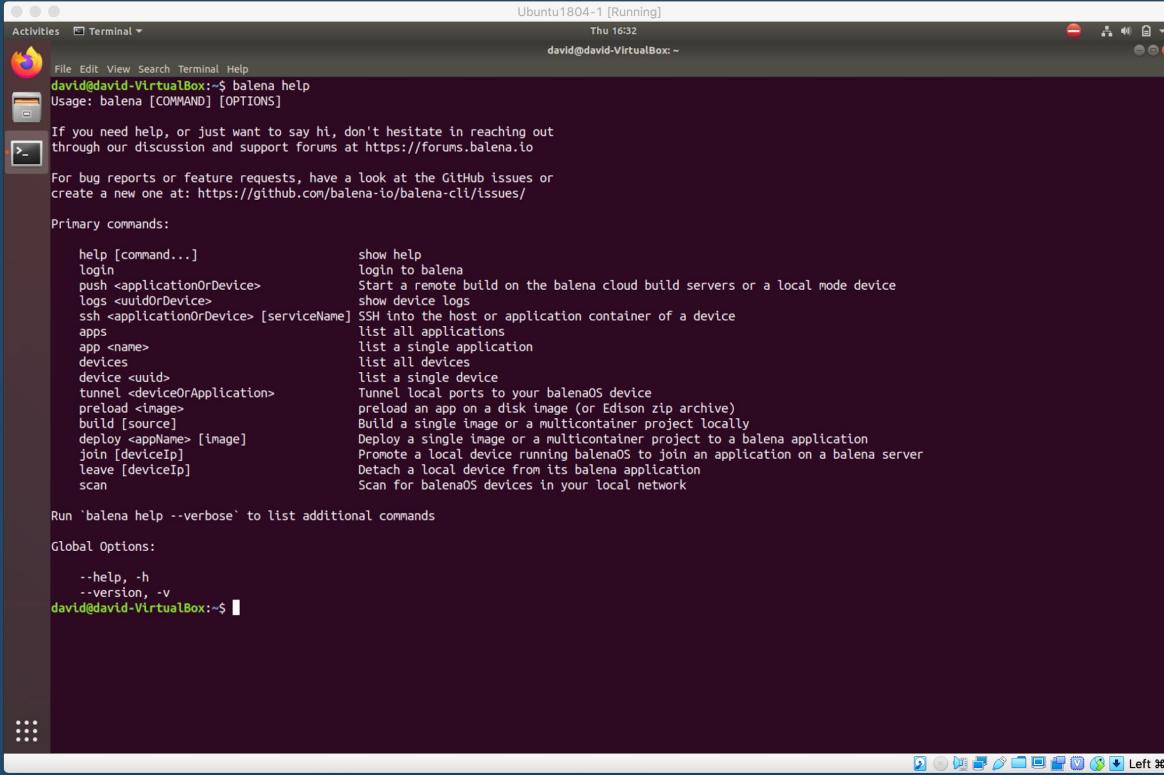
<https://github.com/balena-io/balena-cli/blob/master/INSTALL.md>

Download:

<https://github.com/balena-io/balena-cli/releases/latest>



Install balena-CLI



The screenshot shows a terminal window titled "Ubuntu1804-1 [Running]" with the command "david@david-VirtualBox:~\$ balena help" entered. The output provides usage information and a detailed list of primary commands:

```
File Edit View Search Terminal Help
david@david-VirtualBox:~$ balena help
Usage: balena [COMMAND] [OPTIONS]

If you need help, or just want to say hi, don't hesitate in reaching out
through our discussion and support forums at https://forums.balena.io

For bug reports or feature requests, have a look at the GitHub issues or
create a new one at: https://github.com/balena-io/balena-cli/issues/

Primary commands:

  help [command...]          show help
  login                        login to balena
  push <applicationOrDevice>  Start a remote build on the balena cloud build servers or a local mode device
  logs <uuidOrDevice>          show device logs
  ssh <applicationOrDevice> [serviceName] SSH into the host or application container of a device
  apps                          list all applications
  app <name>                   list a single application
  devices                      list all devices
  device <uuid>                list a single device
  tunnel <deviceOrApplication> Tunnel local ports to your balenaOS device
  preload <image>              preload an app on a disk image (or Edison zip archive)
  build [source]               Build a single image or a multicontainer project locally
  deploy <appName> [image]     Deploy a single image or a multicontainer project to a balena application
  join [deviceIp]              Promote a local device running balenaOS to join an application on a balena server
  leave [deviceIp]             Detach a local device from its balena application
  scan                         Scan for balenaOS devices in your local network

Run 'balena help --verbose' to list additional commands

Global Options:

  --help, -h
  --version, -v
david@david-VirtualBox:~$
```



Push Our Container

The screenshot shows a desktop environment with several windows open:

- Terminal:** A terminal window titled "Ubuntu1804-1 [Running]" showing a command-line session for pushing a Docker image to Balena. The session includes output from balena push PyCon and details about the Docker build process.
- Dockerfile.template:** A code editor window showing a Dockerfile template. It specifies building from a base image, setting the working directory to /usr/src/app, installing RPi.bme280 via pip, and running a Python script (280.py).
- 280.py:** Another code editor window showing Python code for interacting with an I2C device (BME280) using smbus2 and RPi.bme280 modules. The code includes loading calibration parameters, taking a sample, and printing the compensated reading.
- File Browser:** A file browser window showing a directory structure with files named 280.py and Dockerfile.template.



Push Our Container

The screenshot shows a desktop environment with several windows open:

- Terminal Window:** Titled "Ubuntu1804-1 [Running]". It displays the output of a Docker command:

```
[Success] Release successfully created!
[Info] Release: ce7ccf2d18a47d9634627736bd024b86 (id: 1553308)
[Info] Service Image Size Build Time
[Info] main 247.12 MB 38 seconds
[Info]
[Info] Build finished in 1 minute, 1 second
```
- File Browser:** Shows a file tree with "280.py" and "Dockerfile.template" in the "work" directory.
- Dockerfile.template:** A text editor showing the Dockerfile template:

```
FROM balenalib/%{BALENA_MACHINE_NAME%}-python:latest
WORKDIR /usr/src/app
RUN pip install RPi.bme280
COPY . .
CMD [ "python", "./280.py" ]
```
- 280.py:** A text editor showing the Python script:

```
import smbus2
import bme280

port = 1
address = 0x76
bus = smbus2.SMBus(port)

calibration_params = bme280.load_calibration_params(bus, address)

# the sample method will take a single reading and return a
# compensated_reading object
data = bme280.sample(bus, address, calibration_params)

# the compensated_reading class has the following attributes
print(data.datetime)
print(data.timestamp)
print(data.temperature)
print(data.pressure)
print(data.humidity)

# there is a handy string representation too
print(data)
```



Device Updating

The screenshot shows the balenaCloud dashboard for a device named "bold-pancake".

Device Status: The device is currently **Updating** (36% complete). It has been online for 7 minutes.

System Information: Host OS version is **balenaOS 2.83.21+rev1 (development)**, Supervisor version is **12.10.3**. Current release is **Factory build**.

Network: Local IP address is **192.168.0.240**, Public IP address is **2600:8800:b4c0:b700::1dfd**, MAC address is **BB:27:EB:DE:ED:AC**.

Logs: Shows log entries from January 31, 2022, at 22:19:24, indicating configuration changes and network creation.

Terminal: A terminal session is open, showing a progress bar for a command and a "Start terminal session" button.

System Metrics: CPU usage is ~77%, Temperature is ~40C. Memory usage is 128 MB / 970 MB, Storage usage is 69 MB / 13.8 GB.

Services: A table shows a service named "main" in the process of **Downloading** (36%), Release 6179d1e.



Success!

The screenshot shows the balenaCloud dashboard interface for a device named "bold-pancake".

Device Status: The device is online (status: Online, UUID: bcea9c6, Type: Raspberry Pi 3). A warning message indicates "Danger! The device is undervolted!"

Online For: 9 minutes

HOST OS VERSION: balenaOS 2.83.21+rev1 (development)

SUPERVISOR VERSION: 12.10.3

CURRENT RELEASE: Factory build (target release: 6179die)

LOCAL IP ADDRESS: 192.168.0.240

PUBLIC IP ADDRESS: 2600:8800:b4c0:b700:1dfd:88:27:EB:DE:ED:AC

MAC ADDRESS: B8:27:EB:8B:B8:F9

TAGS (0): No tags configured yet

NOTES: Add device notes...

SERVICES: main (Status: Starting, Release: 6179die)

Metrics: CPU usage (~77%), Temperature (~40C), Memory (128 MB/970 MB), Storage (/dev/mmcblk0p6) (324 MB/13.8 GB)

Logs: UTC, Timestamps, Views. Log entries show system activity, including main processes and compensated readings.

Terminal: Select a target, Start terminal session.



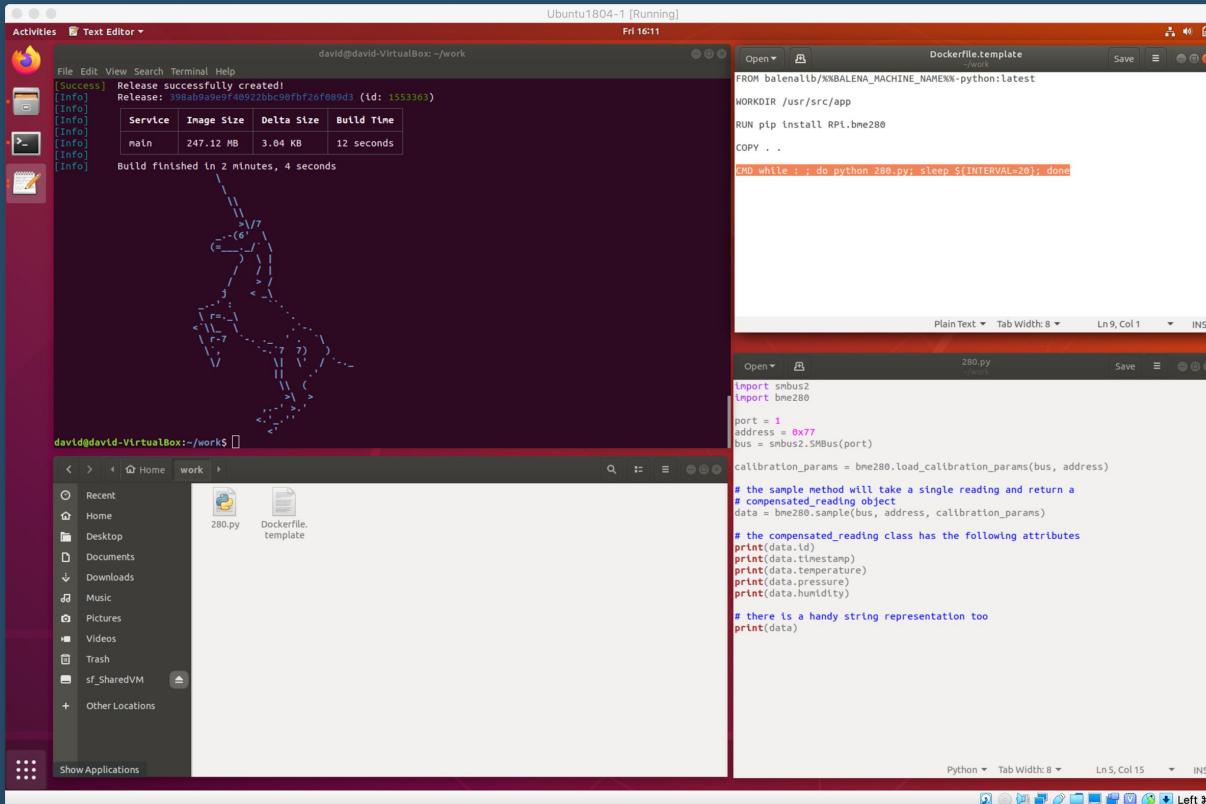
Hmmm, Maybe Not...

The screenshot shows the balenaCloud dashboard interface. On the left, a sidebar lists 'Organizations', 'PyCon', 'Fleets', 'PyConiran', 'Devices', and a selected item 'bold-pancake'. Below these are sections for 'Device Variables', 'Device Configuration', 'Actions', 'Diagnostics', and 'Location'. At the bottom of the sidebar are buttons for 'v9.2.0' and '<>'. The main area is titled 'Logs' and contains a log viewer with a search bar ('Search entries...'), UTC/Timestamps toggle, and a 'Views' dropdown. The log output is as follows:

```
023 °C, pressure=956.99 hPa, humidity=21.74 % rH)
31.01.22 22:28:52 (+0000) Service exited 'main sha256:9958ac42a41dc3c2cie141643b2a4c30b3af1c4e9f5c0109246a8cd23b62d7'
31.01.22 22:29:01 (+0000) Restarting service 'main sha256:9958ac42a41dc3c2cie141643b2a4c30b3af1c4e9f5c0109246a8cd23b62d7'
31.01.22 22:29:02 (+0000) main 3475df96-58cf-49ff-afe7-1411324cae2c
31.01.22 22:29:02 (+0000) main 2022-01-31 22:29:02.019248+00:00
31.01.22 22:29:02 (+0000) main 25.043444162240483
31.01.22 22:29:02 (+0000) main 956.9957810963503
31.01.22 22:29:02 (+0000) main 21.80351514669247
31.01.22 22:29:02 (+0000) main compensated_reading(id=3475df96-58cf-49ff-afe7-1411324cae2c, timestamp=2022-01-31 22:29:02.019248UTC, temp=25.
043 °C, pressure=957.00 hPa, humidity=21.80 % rH)
31.01.22 22:29:03 (+0000) Service exited 'main sha256:9958ac42a41dc3c2cie141643b2a4c30b3af1c4e9f5c0109246a8cd23b62d7'
31.01.22 22:29:03 (+0000) Restarting service 'main sha256:9958ac42a41dc3c2cie141643b2a4c30b3af1c4e9f5c0109246a8cd23b62d7'
31.01.22 22:29:19 (+0000) main bc280040-b482-4787-9ee3-0a8f3728b245
31.01.22 22:29:19 (+0000) main 2022-01-31 22:29:19.469483+00:00
31.01.22 22:29:19 (+0000) main 25.04854641354177
31.01.22 22:29:19 (+0000) main 956.9496653047436
31.01.22 22:29:19 (+0000) main 21.750382558764827
31.01.22 22:29:19 (+0000) main compensated_reading(id=bc280040-b482-4787-9ee3-0a8f3728b245, timestamp=2022-01-31 22:29:19.469483UTC, temp=25.
049 °C, pressure=956.95 hPa, humidity=21.75 % rH)
31.01.22 22:29:21 (+0000) Service exited 'main sha256:9958ac42a41dc3c2cie141643b2a4c30b3af1c4e9f5c0109246a8cd23b62d7'
31.01.22 22:29:21 (+0000) Restarting service 'main sha256:9958ac42a41dc3c2cie141643b2a4c30b3af1c4e9f5c0109246a8cd23b62d7'
31.01.22 22:29:49 (+0000) main c1370702-ce86-4478-b83e-f63c201f42c2
31.01.22 22:29:49 (+0000) main 2022-01-31 22:29:49.927821+00:00
31.01.22 22:29:49 (+0000) main 25.043444162240483
31.01.22 22:29:49 (+0000) main 956.9417492660898
31.01.22 22:29:49 (+0000) main 21.750603739838482
31.01.22 22:29:49 (+0000) main compensated_reading(id=c1370702-ce86-4478-b83e-f63c201f42c2, timestamp=2022-01-31 22:29:49.927821UTC, temp=25.
043 °C, pressure=956.94 hPa, humidity=21.75 % rH)
31.01.22 22:29:51 (+0000) Service exited 'main sha256:9958ac42a41dc3c2cie141643b2a4c30b3af1c4e9f5c0109246a8cd23b62d7'
```



Let's Push a Hotfix



Device Update

The screenshot shows the balenaCloud dashboard interface for managing a device named "bold-pancake".

Device Summary:

- Status:** Updating (36%)
- UUID:** bcea9c6
- Type:** Raspberry Pi 3
- Online For:** 7 minutes
- HOST OS VERSION:** balenaOS 2.83.21+rev1 (development)
- SUPERVISOR VERSION:** 12.10.3
- CURRENT RELEASE:** Factory build
- LOCAL IP ADDRESS:** 192.168.0.248
- PUBLIC IP ADDRESS:** 2600:8800:b4c0:b700::1dfd
- MAC ADDRESS:** B8:27:EB:DE:ED:AC
B8:27:EB:B8:B8:F9
- TAGS (0):** No tags configured yet
- NOTES:** Add device notes...

Services:

| Service | Status | Release |
|---------|-----------------|---------|
| main | Downloading 36% | 6179d1e |

Metrics:

- CPU: ~77% Usage, ~40C Temperature
- Memory: 128 MB / 970 MB
- Storage: 69 MB / 13.8 GB

Logs:

```
31.01.22 22:19:24 (+0000) Applied configuration change {"SUPERVISOR_VPN_CONTROL": "true"}  
31.01.22 22:19:25 (+0000) Creating network 'default'  
31.01.22 22:25:31 (+0000) Creating volume 'resin-data'  
31.01.22 22:25:31 (+0000) Downloading image 'registry2.balenacloud.com/v2/6f78c86e382b5bb8ab4bda4a6a92eff0sha256:59d628b325bf04dbf26ca3f04621cd35f2110ebe713ae531ed22b232c6828ce'
```

Terminal:

Select a target ▾

Start terminal session

Need help ?



Much Better

The screenshot shows the balenaCloud dashboard interface. On the left, a sidebar menu includes 'Organizations', 'PyCon', 'Fleets', 'PyConiran', 'Devices', and a selected item 'bold-pancake'. Below these are sections for 'Summary', 'Device Variables', 'Device Configuration', 'Actions', 'Diagnostics', and 'Location'. At the bottom of the sidebar, the version 'v9.2.0' is displayed. The main content area is titled 'Logs' and contains a list of log entries. A search bar labeled 'Search entries...' is at the top of the log list. To the right of the search bar are buttons for 'UTC' and 'Timestamps', and icons for download, delete, and refresh. A 'Views' dropdown menu is also present. The log entries themselves are timestamped and show various system and application logs, such as service starts and compensated reading logs. The last few entries are highlighted in yellow. A 'Need help?' button is located in the bottom right corner of the log area.

```
31.01.22 22:39:43 (+0000) Started service 'main sha256:d2ca61ed51e513252be8f8b942540691bc378a92768d82075c74e57221478b20'
31.01.22 22:39:43 (+0000) [main] c81bf06e-6ed7-4056-957e-5a06829e68b3
31.01.22 22:39:43 (+0000) [main] 2022-01-31 22:39:43.557881+00:00
31.01.22 22:39:43 (+0000) [main] 25.16079601580277
31.01.22 22:39:43 (+0000) [main] 956.7725532945769
31.01.22 22:39:43 (+0000) [main] 21.75727512608605
31.01.22 22:39:43 (+0000) [main] compensated_reading(id=c81bf06e-6ed7-4056-957e-5a06829e68b3, timestamp=2022-01-31 22:39:43.557881UTC, temp=25.161 °C, pressure=956.77 hPa, humidity=21.76 % RH)
31.01.22 22:39:43 (+0000) [main] c81bf06e-6ed7-4056-957e-5a06829e68b3
31.01.22 22:39:43 (+0000) [main] 2022-01-31 22:39:43.557881+00:00
31.01.22 22:39:43 (+0000) [main] 25.16079601580277
31.01.22 22:39:43 (+0000) [main] 956.7725532945769
31.01.22 22:39:43 (+0000) [main] 21.75727512608605
31.01.22 22:39:43 (+0000) [main] compensated_reading(id=c81bf06e-6ed7-4056-957e-5a06829e68b3, timestamp=2022-01-31 22:39:43.557881UTC, temp=25.161 °C, pressure=956.77 hPa, humidity=21.76 % RH)
31.01.22 22:40:04 (+0000) [main] c58cb1df-1c82-4afb-8ebf-9e5ba281f92b
31.01.22 22:40:04 (+0000) [main] 2022-01-31 22:40:04.195286+00:00
31.01.22 22:40:04 (+0000) [main] 25.186307308688992
31.01.22 22:40:04 (+0000) [main] 956.7310567792695
31.01.22 22:40:04 (+0000) [main] 21.53856617693546
31.01.22 22:40:04 (+0000) [main] compensated_reading(id=c58cb1df-1c82-4afb-8ebf-9e5ba281f92b, timestamp=2022-01-31 22:40:04.195286UTC, temp=25.186 °C, pressure=956.73 hPa, humidity=21.54 % RH)
31.01.22 22:40:24 (+0000) [main] ea60511d-f329-4093-bf11-dd0d45cd7c6b
31.01.22 22:40:24 (+0000) [main] 2022-01-31 22:40:24.912817+00:00
31.01.22 22:40:24 (+0000) [main] 25.196511827880748
31.01.22 22:40:24 (+0000) [main] 956.7198609866718
31.01.22 22:40:24 (+0000) [main] 21.667508299384263
31.01.22 22:40:24 (+0000) [main] compensated_reading(id=ea60511d-f329-4093-bf11-dd0d45cd7c6b, timestamp=2022-01-31 22:40:24.912817UTC, temp=25.197 °C, pressure=956.72 hPa, humidity=21.67 % RH)
```



Let's Make Some Upgrades



1 BME280 is out, BME680 is in

- Let's increase the capabilities of our IoT Node, by swapping out sensors from the BME280 to a BME680:



2

Let's Upgrade the Software Also

Balena builds lots of sample projects to get started with, such as:

“Build a remote controlled digital display with Screenly OSE and Raspberry Pi”

“Turn your old speakers or Hi-Fi into Bluetooth, Airplay and Spotify receivers.”

“Build a TTN LoRaWAN Gateway with balenaFin and balenaCloud”

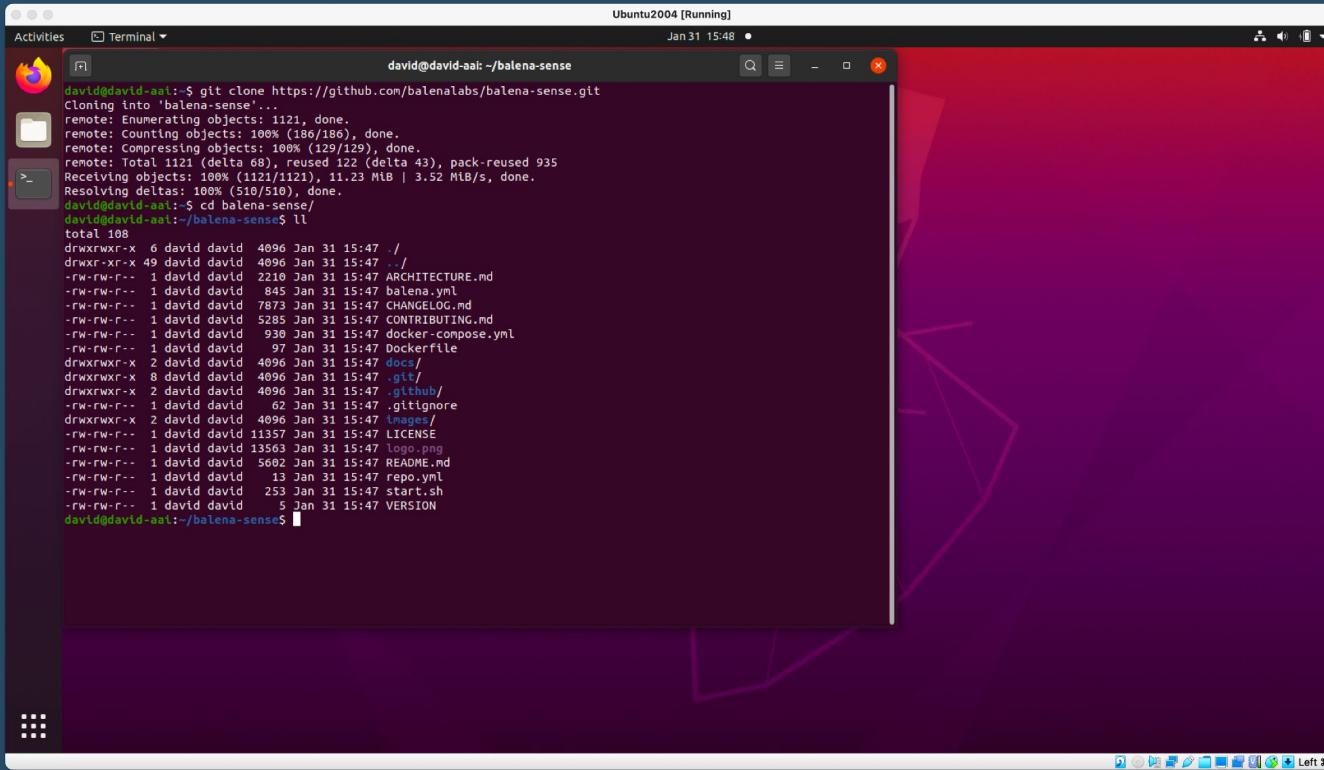
For now, we’re talking about weather data, so let’s go with **balenaSense**: “Build an air quality monitor with InfluxDB, Grafana, and Docker on a Raspberry Pi”

- <https://github.com/balenalabs/balena-sense>



2

Clone balenaSense



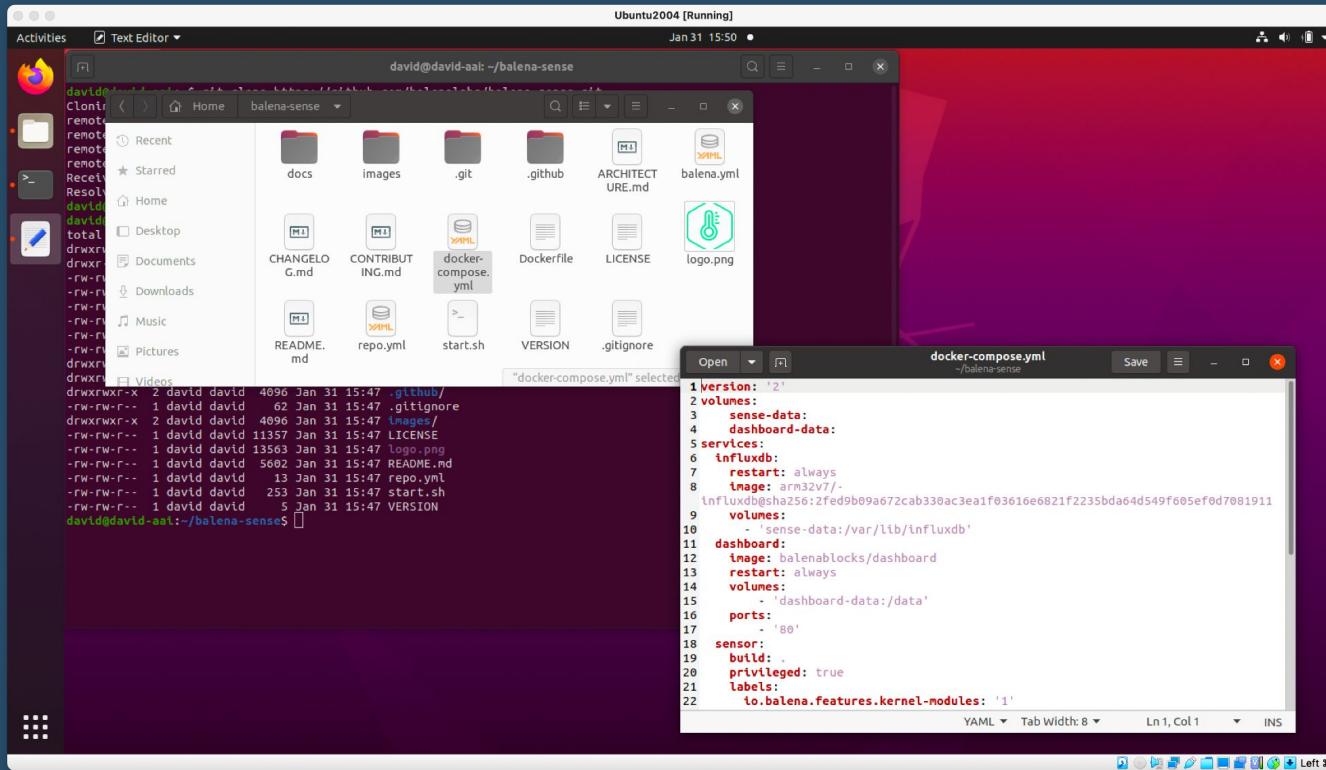
The screenshot shows a terminal window on an Ubuntu 20.04 LTS desktop environment. The terminal title is "Ubuntu2004 [Running]" and the date and time are "Jan 31 15:48". The user is running a command to clone the "balenaSense" repository from GitHub. The terminal output is as follows:

```
david@david-aai:~$ git clone https://github.com/balenalabs/balena-sense.git
Cloning into 'balena-sense'...
remote: Enumerating objects: 1121, done.
remote: Counting objects: 100% (186/186), done.
remote: Compressing objects: 100% (129/129), done.
remote: Total 1121 (delta 68), reused 122 (delta 43), pack-reused 935
Receiving objects: 100% (1121/1121), 11.23 MB | 3.52 MB/s, done.
Resolving deltas: 100% (510/510), done.
david@david-aai:~$ cd balena-sense/
david@david-aai:~/balena-sense$ ll
total 108
drwxrwxr-x  6 david david 4096 Jan 31 15:47 .
drwxr-xr-x 49 david david 4096 Jan 31 15:47 ..
-rw-rw-r--  1 david david 2710 Jan 31 15:47 ARCHITECTURE.md
-rw-rw-r--  1 david david  845 Jan 31 15:47 balena.yml
-rw-rw-r--  1 david david 7873 Jan 31 15:47 CHANGELOG.md
-rw-rw-r--  1 david david 5285 Jan 31 15:47 CONTRIBUTING.md
-rw-rw-r--  1 david david  930 Jan 31 15:47 docker-compose.yml
-rw-rw-r--  1 david david   97 Jan 31 15:47 Dockerfile
drwxrwxr-x  2 david david 4096 Jan 31 15:47 docs/
drwxrwxr-x  8 david david 4096 Jan 31 15:47 .git/
drwxrwxr-x  2 david david 4096 Jan 31 15:47 .github/
-rw-rw-r--  1 david david   62 Jan 31 15:47 .gitignore
drwxrwxr-x  2 david david 4096 Jan 31 15:47 images/
-rw-rw-r--  1 david david 11357 Jan 31 15:47 LICENSE
-rw-rw-r--  1 david david 13563 Jan 31 15:47 logo.png
-rw-rw-r--  1 david david  5002 Jan 31 15:47 README.md
-rw-rw-r--  1 david david   13 Jan 31 15:47 repo.yml
-rw-rw-r--  1 david david  253 Jan 31 15:47 start.sh
-rw-rw-r--  1 david david     5 Jan 31 15:47 VERSION
david@david-aai:~/balena-sense$
```



2

Clone balenaSense



3

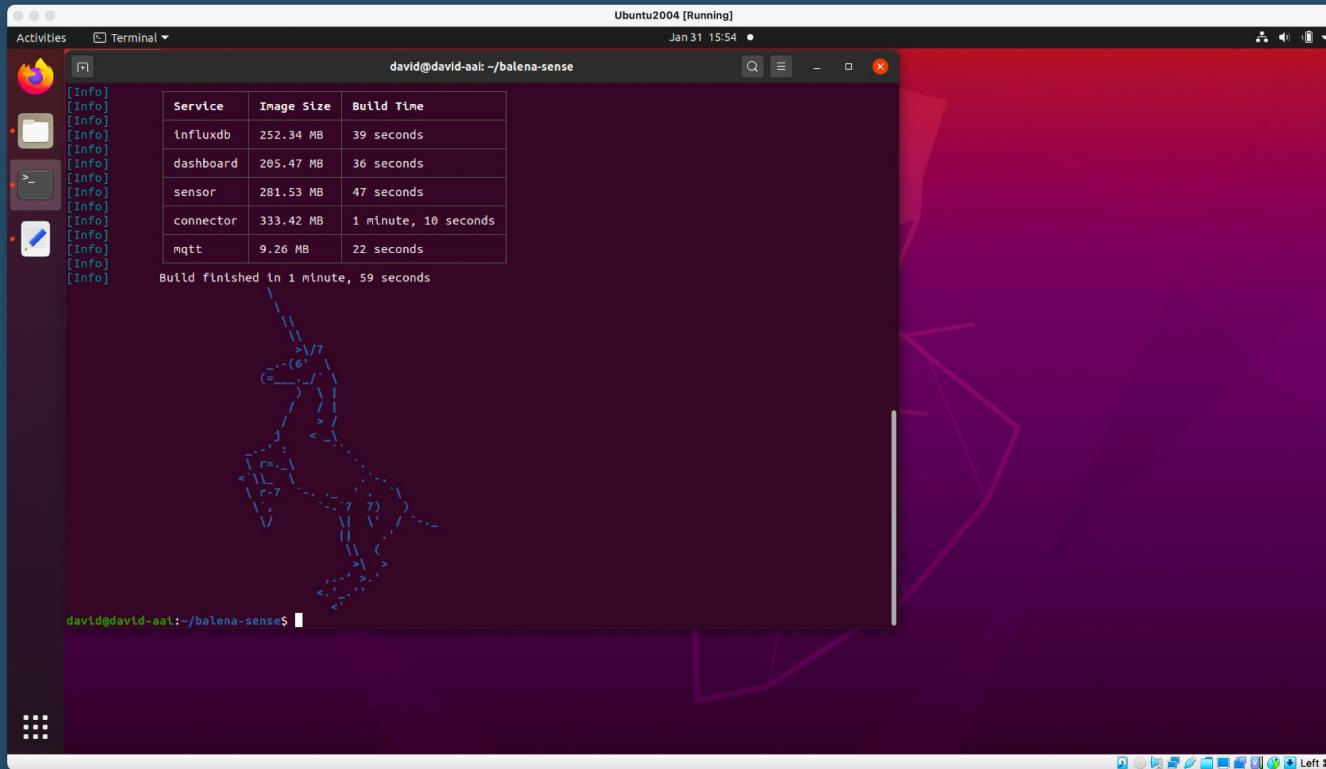
Push Our New Containers

```
Ubuntu2004 [Running]
Activities Terminal david@david-aai: ~/balena-sense Jan 31 15:52 •
david@david-aai:~/balena-sense$ balena push PyConIran
[info] Starting build for PyConIran, user david_tischler1
[info] Dashboard link: https://dashboard.balena-cloud.com/apps/1904879/devices
[sensor] Step 1/4 : FROM balenablocks/sensor
[sensor] ---- e1f96d181e09
[sensor] Step 2/4 : WORKDIR /usr/src/app
[sensor] ---- Running in 32fd43a0f8ee
[sensor] Removing intermediate container 32fd43a0f8ee
[sensor] ---- 19ef66e40bd9
[sensor] Step 3/4 : COPY start.sh start.sh
[sensor] ---- 856b5383ad86
[sensor] Step 4/4 : CMD ["bash", "start.sh"]
[sensor] ---- Running in 0191fe56967a
[sensor] Removing intermediate container 0191fe56967a
[sensor] ---- 46b16b125196
[sensor] Successfully built 46b16b125196
[influxdb] [=====] 100%
[connector] [=====] 51%
[connector] [=====>] 42%
[mqtt] [=====] 100%
```



3

Another Unicorn



3

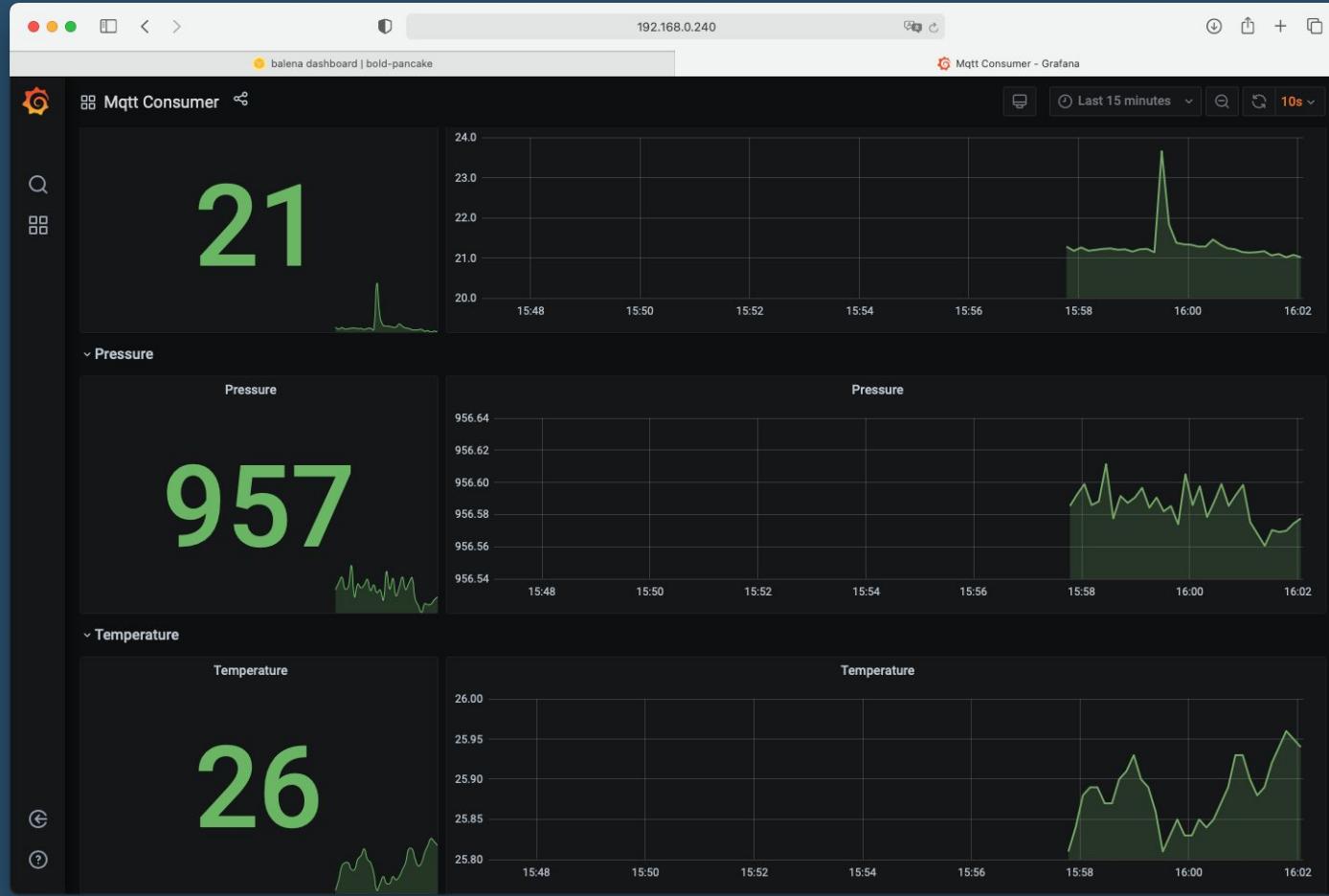
Push Our New Containers

The screenshot shows the balenaCloud dashboard interface. On the left, a sidebar lists organization, fleet, and device management sections, along with a summary section containing Device Variables, Device Configuration, Actions, Diagnostics, and Location.

The main content area displays the following information:

- Device Status:** The device is currently **Updating** (bcea9c6) with a progress bar at 55%. It has been **ONLINE FOR** 36 minutes. The **HOST OS VERSION** is **balenaOS 2.83.21+rev1** and the **SUPERVISOR VERSION** is **12.10.3**. The **CURRENT RELEASE** is **9216e6d** and the **TARGET RELEASE** is **500074e**. Local IP address is **192.168.0.240**, Public IP address is **2600:8800:b4c0:b780::1dfd**, and MAC address is **B8:27:EB:DE:ED:AC**.
- Logs:** A log viewer showing entries from the device's terminal. One entry indicates a service exit: `31.01.22 22:53:26 (+0000) Service exited 'main sha256:d2ca61ed51e513252be8f8b942540691bc378a92768d82075c74e57221478b20`. Another entry shows a download: `31.01.22 22:53:27 (+0000) Killed service 'main sha256:d2ca61ed51e513252be8f8b942540691bc378a92768d82075c74e57221478b20`.
- Terminal:** A terminal session window titled "Select a target" with a "Start terminal session" button.
- Services:** A table listing services and their status. Services include connector, main, mqtt, and sensor. Their current status is Downloading (45%, 63%), Downloaded (9216e6d, 500074e), and Downloading (63%).





Scaling and a Global Fleet



dashboard.balena-cloud.com

Getting Started Docs Forums Status David T DT

Add device Tags Actions

Map Hybrid

Organizations PyCon Fleets PyConFran

Summary Devices Releases Variables Configuration Actions Settings Members Teams

v9.3.6

Need help ?

Google Keyboard shortcuts Map data ©2022 Google, INEGI | Terms of Use

The screenshot shows the balenaCloud interface with a sidebar on the left containing navigation links such as 'Organizations', 'PyCon', 'Fleets', 'PyConFran', 'Summary' (selected), 'Devices', 'Releases', 'Variables', 'Configuration', 'Actions', 'Settings', 'Members', and 'Teams'. A bottom status bar indicates 'v9.3.6'. The main area features a world map with three yellow location markers. One marker is positioned in the United States (Arizona), another in Europe (Spain), and a third in Asia (Pakistan). The map is labeled with various countries and regions, including Canada, Greenland, Norway, Sweden, Iceland, United Kingdom, France, Spain, Portugal, Morocco, Algeria, Libya, Egypt, Mauritania, Mali, Niger, Chad, Sudan, South Sudan, Ethiopia, Somalia, Kenya, DRC, Tanzania, Angola, Zambia, Mozambique, Madagascar, India, Nepal, Turkmenistan, Kyrgyzstan, Kazakhstan, Uzbekistan, Iran, Iraq, Syria, Turkey, Greece, Romania, Austria, Germany, Poland, Denmark, Norway, Sweden, Finland, and Russia. The map also shows major bodies of water like the North Atlantic Ocean, Arctic Ocean, and Indian Ocean.



Dashboard URL: dashboard.balena-cloud.com

The screenshot shows the balenaCloud dashboard interface for the "PyConiran" organization. The left sidebar lists various sections: Organizations, PyCon, Fleets, PyConiran (selected), Summary, Devices (selected), Releases, Variables, Configuration, Actions, Settings, Members, and Teams. The bottom of the sidebar shows the version v9.3.6.

The main content area displays the following information:

- PyConiran**: Shows a Raspberry Pi icon with a red notification badge containing the number 3. Details include:
 - Architecture: armv7hf
 - Slug: pycon/pyconiran
 - Created: Jan 31st 2022, 3:08 pm
 - Status: Starter
- Devices**: Shows 3 devices online. Status breakdown: Online (3), Config (0), Updating (0), Offline (0), Post prov (0), Inactive (0). A green progress bar indicates device status. Buttons include "Create release" and "Actions".
- Releases**: Shows 5 releases. A dropdown menu is set to "track latest". A button "Create release" is present.

A table lists the three devices currently online:

| Name | Status | Device type | Last seen | UUID | OS version | OS variant | IP ad |
|---------------|----------|---------------------------------|-----------------------|---------|-----------------------|-------------|-------------|
| Barcelona-Pi3 | ✓ Online | Raspberry Pi 3 | Online (for 3 hours) | c644e1c | balenaOS 2.83.21+rev1 | Development | 192.168.1.4 |
| New-Delhi | ✓ Online | Raspberry Pi 4 (using 64bit OS) | Online (for 3 hours) | 4346201 | balenaOS 2.88.5+rev1 | Development | 192.168.1.2 |
| Phoenix-Pi3A | ✓ Online | Raspberry Pi 3 | Online (for a minute) | edd76b9 | balenaOS 2.83.21+rev1 | Development | 192.168.0.2 |

A "Need help?" button is located in the bottom right corner of the main content area.



Let's think about that workflow for a moment...

- We built a container, and loaded it onto a device.
 - Containers can be updated. Changed. Added to. Improved.
 - We can push containers onto a device, over the air, no matter where it is located.
 - The Container(s) are your workload / application.
- We deployed to three devices. But, we could have deployed to 1,000 devices. Or 10,000.
 - Or 100,000.
- We can manage and interact with the device from the web dashboard.



Let's think about that workflow for a moment...

- We can prototype on cheap / available hardware, but go to production with enterprise-grade hardware.
 - The container will run the same.
 - The devices show up in the dashboard.
 - You can push containers to entire Fleets at a time.



So, Scaling is Easy!



Things to try

Similar to balenaSense, we create awesome open-source projects that you can make with readily available hardware.

- balenaSound
- balenaDash (kiosk)
- OpenDataCam (Jetson Nano)
- LoRa Gateway (Things Network)
- HomeAssistant

.... and more



Additional Resources

balena Blog: <https://www.balena.io/blog/>

balena Forums: <https://forums.balena.io/>

balena Labs GitHub: <https://github.com/balenalabs>



Questions / Q&A

David Tischler
Developer Advocate
t: @balena_io
e: david@balena.io





balena