Yocto at Balena

Building ~70 Unique OS’es for IoT Devices
David Tischler
Developer Advocate
t: @balena_io
e: david@balena.io
Who is Balena?
Etcher, anyone?
You could also use `dd`
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
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-os/balena-cli/issues/

Primary commands:

- help [command...]
- login
- push <applicationOrDevice>
- loginDevice
- ssh <applicationOrDevice> [serviceName] SSH into the host or application container of a device
- apps
- list all applications
- list all devices
- deploy <appName> [image]
- build <sources>
- tunnel <deviceOrApplication>
- preload <image>
- join [device]
- leave [device]
- scan

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

Global Options:

- -h, --help
- -v, --version

balena help
<table>
<thead>
<tr>
<th>Status</th>
<th>Name</th>
<th>Last Seen</th>
<th>UUID</th>
<th>OS Version</th>
<th>OS Variant</th>
<th>Supervisor Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bold-feather</td>
<td>Online (for a few seconds)</td>
<td>70faef65</td>
<td>balenaOS 2.46.1-rev3</td>
<td></td>
<td>10.6.27</td>
</tr>
</tbody>
</table>

1 - 1 of 1
balenaOS

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

- The Yocto project enables us to build an OS, add a few custom bits, and create a business model around it.
- balenaOS is open source, just like Yocto.
<table>
<thead>
<tr>
<th>Device</th>
<th>Version</th>
<th>Architecture</th>
<th>Production</th>
<th>Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeagleBone Green Wireless</td>
<td>BalenaOS 2.51.1+</td>
<td>armv7hf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTI Astro TX2 G+</td>
<td>BalenaOS 2.56.0+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTI Orbbity TX2</td>
<td>BalenaOS 2.51.1+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTI Photon Nano</td>
<td>BalenaOS 2.69.1+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTI Photon Xavier NX</td>
<td>BalenaOS 2.67.3+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTI Spacely TX2</td>
<td>BalenaOS 2.67.3+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compulab iOT-gate-imx8</td>
<td>BalenaOS 2.71.5+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compulab MX8M</td>
<td>BalenaOS 2.51.1+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coral Dev Board</td>
<td>BalenaOS 2.67.3+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digi ConnectCore BX SBC Pro</td>
<td>BalenaOS 2.68.1+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Etcher Pro</td>
<td>BalenaOS 2.72.0+</td>
<td>aarch64</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
69 Devices in Total
How is this accomplished?
inherit kernel-resin
inherit kernel-dev-tree
PACKAGES += "$(PM)-fixup-scr"
SRC_URI_remove = "file://N683-ARM-dts-nanopi-air-Add-WiFi-eMMC.patch"
FILESEXTRAPATHS_prepend := "$(THISDIR)/files"
SRC_URI_append_nanopi-air = "\ 
file://nanopi-air/ARM-Linux-mainline-Add-back-eMMC-support-for-Nanopi-Neo.patch 
file://nanopi-air/board-nanopi-air-k3-camera-wifi-blueooth-otg.patch 
" 
SRC_URI_append = " \ 
files/general-add-configs-overlay.patch 
files/general-add-overlay-compilation-support.patch 
files/general-sunxi-overlays.patch 
" 
BALENA_CONFIGS_append = " aap_power"
BALENA_CONFIGS_DEPS[aap_power] = "\"
```bash
#TYPE: Machine
#NAME: genericx86-64-ext
#DESCRIPTION: Machine configuration for an extra configured genericx86-64 device
MACHINEOVERRIDE = "$genericx86-64:\$MACHINE"
include conf\machine\genericx86-64.conf?
```
Repeat 68 more times
Adding a new board
https://github.com/balena-os

- Does your board have a Yocto build already?
- Check for existing “similar” boards
  - SoC Manufacturer
  - Product Line
- If so, clone the appropriate repo and begin porting in your board
- If not, then you will need to go the longer way (but not terrible), which is starting from scratch as documented here:
  - https://www.balena.io/os/docs/custom-build/
Building your Own

Bake your own Image

In order to build your very own version of balenaOS for one of our supported boards, you will first need to make sure you have a working Yocto environment setup. Then pick the device type you want to build, in this example we will use the Raspberry Pi 3. So first we need to grab the `balena-raspberry` and initialise all its submodules.

```bash
git clone https://github.com/balena-os/balena-raspberry
cd balena-raspberry
git submodule update --init --recursive
```

We can then use the helpful `BARYS` tool to setup and start our build. To see all the functionality `BARYS` provides run `./balena-yocto-scripts/build/barys -h` from within the repo.

Now to actually build a development version of balenaOS for the Raspberry Pi 3, we can run the following:

```bash
./balena-yocto-scripts/build/barys -m raspberry3
```

Now sit tight and maybe go and make several cups of tea, this is going to take a little while.

Supporting your Own Board

Pre-requisites: a Yocto Board Support Package (BSP) layer for your particular board. It should be compatible to the Yocto releases balenaOS supports.

Repositories used to build balenaOS host Operating System (OS) are typically named `balena-{board-family}`. For example, consider `balena-raspberry` which is used for building the OS for Raspberry Pi, or `balena-intel` repository which can be used to build a balena image for the Intel NUC boards.
Yocto Project Call-to-Action

- [https://lists.yoctoproject.org/g/yocto/topic/open_source_maintainers_an/82722441](https://lists.yoctoproject.org/g/yocto/topic/open_source_maintainers_an/82722441)
  - There are a few things companies can do to help:
    - a) Publicly acknowledge you use the project.
    - b) Embrace employee's Open Source contributions, code and otherwise
    - c) Consider Yocto Project membership
    - d) Support employees in spending some time on open source projects
    - e) Transition roles

- [https://www.yoctoproject.org/join/](https://www.yoctoproject.org/join/)
Additional Resources

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

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

balena-os GitHub:  https://github.com/balena-os
IoT Happy Hour
Weekly IoT hangout & freeform chat
Fridays | 4 PM UTC
Questions / Q&A
David Tischler
Developer Advocate
t: @balena_io
e: david@balena.io