OpenBalena

Open-source IoT Fleet Management

Outline

- Why?
- What?
- How?
- And?

Why?

- Manage 100s/1000s of IoT devices
- Automate updates
- Speed up provisioning
- Remote access
- Focus on software/services rather than fleet management
- Faster to market

What?

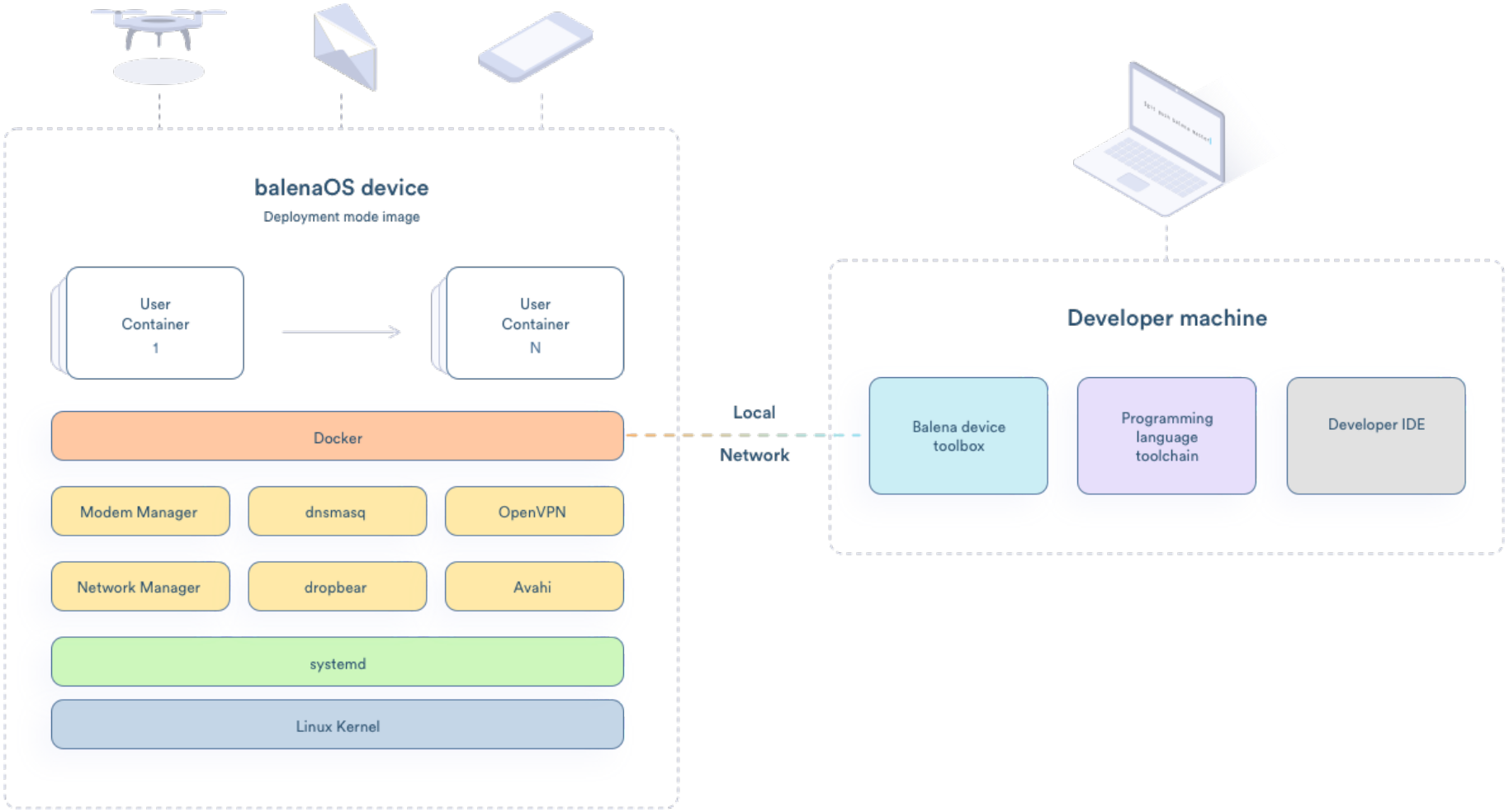
- 

  - https://www.balena.io/open/
  - Only CLI compared to BalenaCloud
  - AGPLv3 licensed

How?

- Uses Docker
- Light-weight images using BalenaOS
- Images based on Yocto-Linux
- Supports multiple devices
  - RPis
  - BeagleBone/Board
  - NVidia Jetsons
  - Intel Edison/NUC
  - MS Surface
  - Generic aarch64, x86_64
  - ...

How?



Source: https://www.balena.io/os/

And?

- Sort of...
- Get started example
  - https://www.balena.io/open/docs/getting-started/
  - Requires x86_64 (aarch64 didn't work)
  - Ubuntu 18.04 (Arch Linux didn't work)
  - Slightly oudated docs: app/apps -> fleet/fleets
  - Requires DNS for endpoints:

    api, registry, vpn, s3, tunnel

Local setup

- RPi4 (192.168.2.150)
  - running bind9
    - openbalena.local
    - Slingshot DNS servers as forwarders
  - dhcpcd.conf
    - static IP address/router/DNS
- Router
  - uses RPi4 as DNS server

Local setup

- VirtualBox with Ubuntu 18.04
  - bridged network interface
  - runs OpenBalena server
  - /etc/hosts with local endpoints

    {api|registry|vpn|s3|tunnel}.openbalena.local

Create fleet

- Copy *ca.crt* onto machine with balena CLI
- point *NODE_EXTRA_CA_CERTS* env variable to *ca.crt*
- point CLI to correct domain:
  - create ~/.balenarc.yml
  - insert domain:

    balenaUrl: 'openbalena.local'
- log into balena
- create fleet

    balena fleet create <fleetname>

Add device

- Spare RPi2
- Download image archive for RPi2 and unzip it
- Use balena CLI to provision image
  - balena os configure <img> --fleet <fleetname>
- Copy image onto SD card using dd
- Re-mount SD card
  - open resin-boot partition
  - open config.json file
  - insert custom DNS servers and save file:

    "dnsServers": "192.168.2.150 8.8.8.8",

Add device

- Insert SD card in device
- Boot device
- ...but doesn't register with server API???
- Potential reason (from journalctl):
  - RPi has no HW clock, set to Sept 20
  - TLS/SSL fails to due *certificate not yet valid*
  - Few seconds later NTP corrects time

What works?

- Balena CLI *scan* finds (local) device
- Balena CLI can *ssh* into device
- Re-joining device via Balena CLI *join* command supposedly succeeds, but device still doesn't show up
- Tried NVidia Jetson images but fail to provision image ("Unsupported filesystem.")

What next?

- Using proper public DNS names
- Using cloud instance for server
- Maybe trialing paid-for service (which uses the open-source components under the hood) to further evaluate