OpenPrinting®

Distribution Methods Roadmap
Snap, OCI containerization

Till Kamppeter – OpenPrinting

6 May 2024

What is Snap?

- **Sandboxed packaging**
- **OS-distribution-independent**
  - **You package and test once**, put your **Snap** into the **Snap Store**, and users of **any distro** (Ubuntu, Debian, SUSE, Red Hat, Windows, ...) can use it.
  - **All libraries and other dependencies** come with your Snap
- Your app runs in a **security shell** isolated from the host system
  - Communication to outside only via **well-defined interfaces**
  - **Snap Store has control**, has to explicitly permit "dangerous" interfaces
  - This way we can **trust third-party apps**
  - We are not dependent any more on distro maintainers for secure packages
- **User experience as with smartphone apps**

What is Snap?

- **Don't fear the daemons, we snap them, too!**
  - Snap is universal, not only desktop apps but also daemons, system utilities, sub-systems, drivers, operating system cores, kernels, … can get snapped
  - => **All-Snap operating system**, like **Ubuntu Core Desktop**
- **Packaging moves from distros to upstream**
  - 10+ distros, each packaging XXX, inventing the wheel 10+ times
  - So let upstream, XXX.org, snap it, distros take the Snap
  - Distro version released, app updates continue from upstream
- **Immutable distros, Immutable sub-systems, Immutable apps**
  - Ubuntu Core: **Immutable core**, all-Snap distro, desktop under development
  - Snaps are **immutable apps** (or **immutable sub-systems**, like the CUPS Snap)

What is Snap?

- Compressed and **GPG-signed read-only squashfs images**
- Includes **metadata** in a ***.yaml** file
- Installed Snap has a **writable file system** area inside its confinement
- Come in **5 types** (app, os core, gadget, kernel, desktop session)
- Support **transactional (atomic) updates** and **rollback**
- Can handle **binary diffs** for smaller download on upgrades
- **Available on multiple distros** and supported by default on all Ubuntu installs since Ubuntu 14.04 (**10 years!!**)

What is Snap?

- **Read-only** file system image (squashfs)
- **GPG signed**
- **Confinement via**:
  - **AppArmor** (File system access rules)
  - **seccomp** (System call restrictions)
  - **Namespaces** (Separate resource spaces: PIDs, users, network, ...)
- **`snapd`** and **`snap-confine`** wrap around all executables in a snap, to ensure only the allowed writable dirs can be accessed

What is Snap?

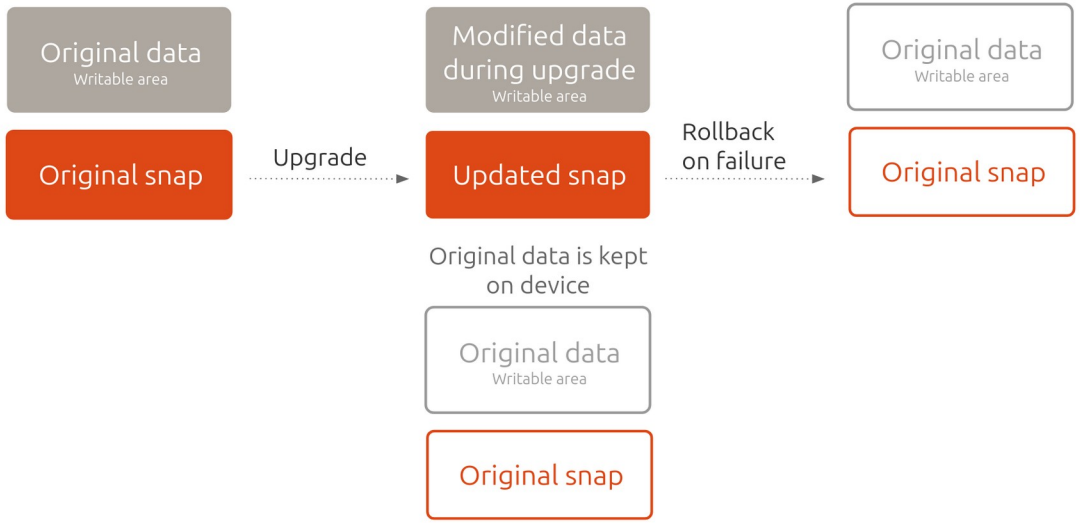
- **“root-safe”**
  - Applications can **run as root** but can not break out of the package confinement, **no need for specific user or group setup** to maintain security.
  - Example: **Daemon Snaps**
- **Storage-efficient**
  - Image stays compressed after install
  - **Core Snaps** and **content provider Snaps** hold common libraries and data files

What is Snap?

- Snapped applications are **completely encapsulated** (AppArmor, seccomp, namespaces)
- By default, they cannot communicate with the host system or with other Snaps
- Communication is possible via **well-defined interfaces**: "network", "cups", "dbus", ...
- A "**plug**" has to be connected with a "**slot**" of the system or of another Snap in order to communicate
  - **“Safe” interfaces**
    - Ex.: “cups” which allows listing available printers and printing
    - **are auto-connected** when installing from Snap Store
  - **“Dangerous” interfaces**
    - Ex.: “cups-control” which allows creating/removing printers, delete all jobs ...
    - **need manual connection** or **permission** from Snap Store team for auto-connection

What is Snap?

- Transactional (atomic) updates
- Current version and its writable area saved, for rollback
- Automatic rollback and reboot after kernel panic or boot failure



The CUPS Snap

- **Complete immutable printing stack in one Snap**
- **Current upstream releases** of all components
  - CUPS
  - cups-filters
  - Ghostscript
  - QPDF
  - cups-browsed
- Provides interface **slots**: “cups”, “cups-control”
- **Plugs** interfaces: “network”, “network-bind”, “network-manager-observe”, “avahi-control”, “raw-usb”
- **System user/group** “snap_daemon” instead of “lp”

The CUPS Snap

- “**cups-control**” interface: **Full admin** access to CUPS
  - **Snap mediation**: cupsd allows admin access from a Snap only it plugs “cups-control”
  - Considered “**dangerous**”, needs permission for auto-connect
  - For **printer setup tools**
- “**cups**” interface: **Printing-only** access to CUPS
  - **Requires Snap mediation** to work, therefore **we force use of CUPS Snap,** using Snap’s domain socket
  - Considered “**safe**”, so it gets auto-connected
  - For **applications which print**

The CUPS Snap

- **“cups” interface forces use of CUPS Snap, if classic CUPS is used**
  - Auto-installs CUPS Snap as content-provider Snap
  - Runs CUPS Snap as proxy, relaying to classic CUPS
- **The “cups” interface, for Snaps of applications which print is complete so far, but**
  - Still uses a content provider (“default-provider”) workaround to auto-install the CUPS Snap
  - Snapd team wants that user gets asked whether they want to install the CUPS Snap on first print attempt
    => Needs further design work on snapd
    => But “cups” can be used though, workaround documented

The CUPS Snap

- **Snap Automation GitHub action**
  - http://github.com/ubuntu/desktop-snaps
  - **Snap Update Automation**: New upstream release of any component of the Snap => Snap gets updated
  - **Snap versioning automation**: Version number like of classic packages: 2.4.7-4, 2.4.7-5, 2.4.8-1, ... Auto-bumped on commits
  - Thanks **Rudra Pratap Singh** for your great contributions here to make update automation work with OpenPrinting and to make versioning automation available for everyone!

The CUPS Snap as a (classic) distro's CUPS

- **What is needed:**
  - **DONE: Security concept** on the snapd side completed
  - **DONE: All drivers** available on Debian retro-fitted into **Printer Applications** (only Braille embossers missing)
  - **DONE: Look-up service for Printer Applications** on OpenPrinting web site:
    - No follow-up on hardware-look-up feature request for Snap Store
    - Could support also other platforms, like Docker

The CUPS Snap as a (classic) distro's CUPS

- **What is needed:**
  - **Desktop Integration:**
    - **Attempt to introduce CUPS Snap in Ubuntu 23.10 failed**
      - **CUPS is in distro core**
        → not only used in standard Ubuntu but also in all flavors
        → All desktops need to work with New Architecture
    - **Printer setup tools:** GNOME Control Center, KDE Settings, system-config-printer
    - **Print dialogs:** GTK (**DONE**), Qt, LibreOffice, Mozilla (Firefox, Thunderbird), Chromium

Printer Application Snaps

- Way of **distribution-independent packaging for printer/scanner drivers**
- **Plugs** interfaces: “avahi-control”, “home”, “network”, “network-bind”, “raw-usb”, “hardware-observe”
- **Kept up-to-date** with Snap Update an Versioning Automation
- **ipp-usb** uses shell script working as “**UDEV observation daemon**” to launch ipp-usb when printer appears
  - Snap does not support UDEV rules
  - Script is based on “**`udevadm`**” command line tool, especially “**`udevadm monitor`**”

Printing and scanning in all-Snap distro Ubuntu Core Desktop

- **Printing**
  - **CUPS Snap + ipp-usb Snap + CPDB CUPS backend Snap**
  - **Driverless (IPP)** printing
  - **Printer Applications Snaps** for drivers
  - **Applications**
    - Plug **“cups”** interface
    - Use **Common Print Dialog Backends (CPDB)**
    - Use **xdg-desktop-portal** (not all desktops/toolkits)
  - **Printer setup tools**
    - Plug **“cups-control”** interface

Printing and scanning in all-Snap distro Ubuntu Core Desktop

- **Scanning**
  - **Driverless (eSCL or WSD)** scanning
  - **Scanner Applications** for drivers
  - **Applications**
    - Snapped with only **sane-airscan** SANE backend (driverless support)

What are OCI Containers

- Software running in a **container/sandbox**
  - System software
  - Server/cloud applications
  - Restricted access to other containers and host system
- Most well-known platform/tool is **Docker**
  - Has the container image “store” **Docker Hub**
- **ROCKs/rockcraft**
  - Easy container image build similar to snapcraft
  - Ubuntu is base distro, as with Snap

Why OCI Containers?

- **Immutable desktop distributons**
  - Most immutable desktop distros do not support Snap
  - Many of them allow adding system software as OCI containers
  - Desktop apps are added as Flatpaks
- **Server/cloud**
  - OCI containers are a standard format here

OpenPrinting OCI Container Images

- **Tons of CUPS Images in Docker Hub**
  - All from **third-parties**, none of them from OpenPrinting
  - Can one **trust** these people?
  - Images can be **highly specialized**, only for a very restricted use case
- **=> We need general-purpose, “official” images from OpenPrinting**
  - **GSoC project by Rudra Pratap Singh** to create OCI images for OpenPrinting