



Desktop Linux

As easy as a smartphone – **Just in a Snap!**

An introduction into the universal packaging format

Till Kamppeter - Opportunity Open Source 3.0 IIT Kanpur 2025



What the hell are Snaps?
And why should I use them?



What the hell are Snaps?

- **App developers** provide apps as **source code**
 - Only **tech-savvy users** can use it directly
 - Or the devs need **goodwill of distro maintainers** to get their app packaged
 - Or they **package their app themselves**, for 10+ distros and have to test on 10+ distros

That is a nightmare! Isn't it?



What the hell are Snaps?

- You have a **smartphone**? There it is much easier: **Google Play Store**, **App Store**
- And remember that Canonical developed a **smartphone OS** (Ubuntu Touch)?
- They have **learned** from it!
⇒ And now we have ...

Snap!

By the way, **Snap** is already more than 10 years old!



What the hell are Snaps?

Sandboxed packaging

- **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
 - **OS-distribution-independent**
 - Package and test once, put into Snap Store, and users of any distro can use it.
 - All libraries and other dependencies come with your Snap
- => User experience as with smartphone apps**



What the hell are Snaps?

- **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
- **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
 - Snaps are **immutable apps**



Snap packages



Snap Package Properties

- 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 **different 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!!**)



Snap Package Security

- **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, ...)
- **snapped** and **snap-confine** wrap around all executables in a snap, to ensure only the allowed resources can be accessed



Snap Package Security

- “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 when mounted
- **Core Snaps** and **content provider Snaps** hold common libraries and data files



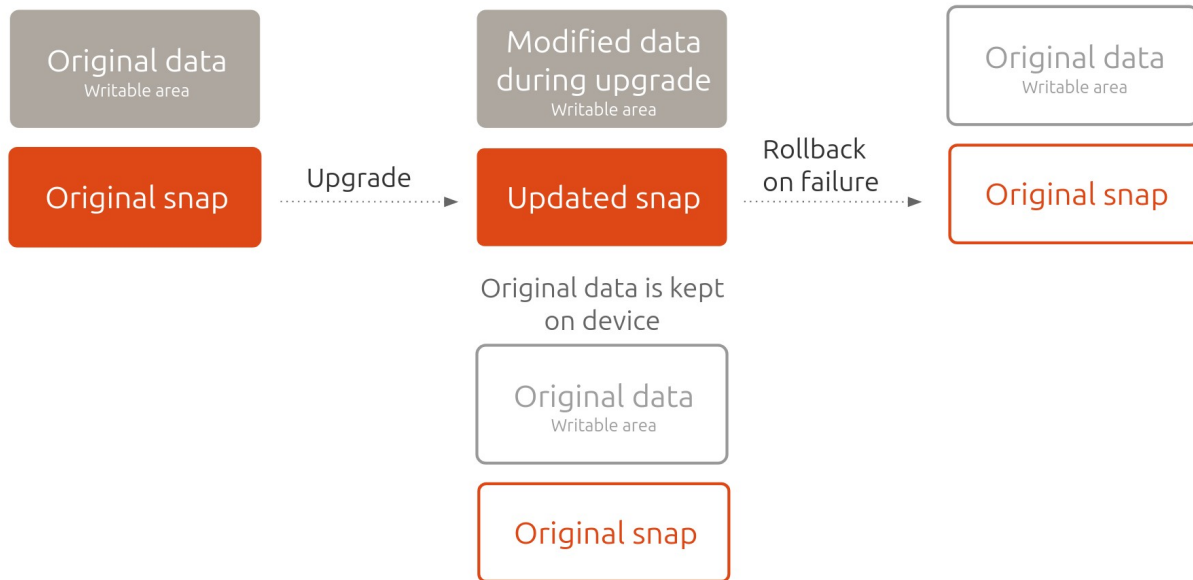
Interfaces: Safe vs. Dangerous

- Snapped app **completely encapsulated** (AppArmor, seccomp, namespaces)
- Cannot communicate with host system or with other Snaps
- Communication is only possible via **well-defined interfaces**: "network", "cups", "dbus", ...
- **"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



Updating Snaps

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



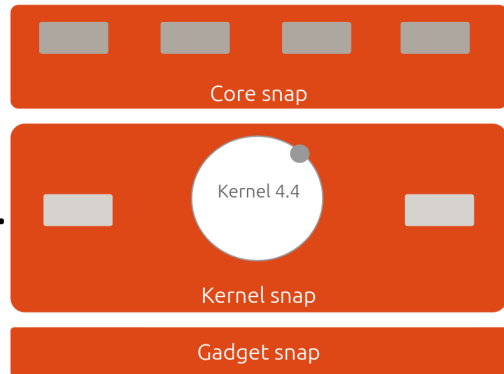


Ubuntu Core – All-Snap OS



Ubuntu Core Operating System

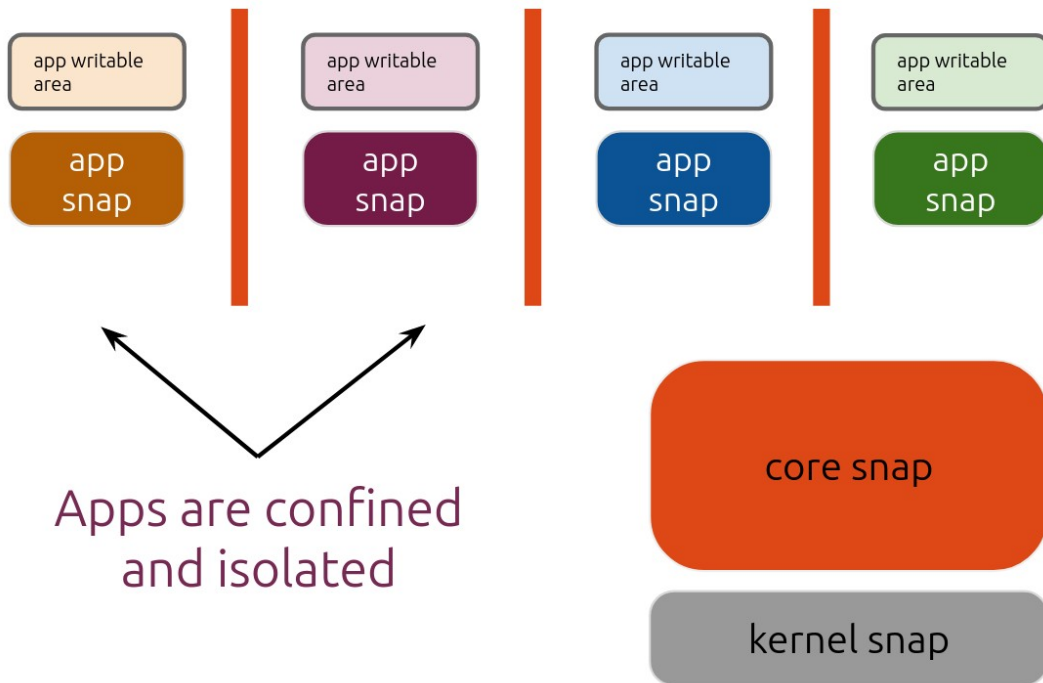
- Originally created for IoT ...
- The all-Snap Ubuntu Core OS consists of
 - **Gadget Snap**
 - Bootloader, partitioning, hardware specifics .
 - **Kernel Snap**
 - **Core Snap**
 - Minimum base operating system
 - core, core18, core20, ..., core24, ... based on Ubuntu LTS
- Comes in one image but Snaps separately updateable
- Applications added as Snaps





No interdependencies between Snaps

- Every Snap can be **independently** updated (and rolled back)





Ubuntu Core Desktop



Ubuntu Core Desktop – Building Blocks

- **Easy to maintain** for end users, like a **smartphone**
- **Boot Base** = Core Snap
- **Additional Bases:** Extra Core Snaps needed for Apps using other coreXX base Snap
- **Ubuntu Desktop Session Snap:** Wayland, Desktop environment (GNOME, later KDE and others)
- All building blocks **independently updateable** and **exchangeable**





Ubuntu Core Desktop

- **Principally as Ubuntu Core**, but image comes with
 - **Desktop Session Snap**
 - **Common Applications**
- Everything **easily** exchangeable: Other desktop, gaming kernel, ...
- **Development work done in LXD containers**, with GUI frontend **Workshops**



Ubuntu Core Desktop – TODO

- **Still to be done for a first release**
 - Gaming: **Nvidia driver** support
 - Productivity: **Printer setup tools** for all-IPP and Printer Application support
 - Productivity: **Scanner Applications**
 - **Development**: IDE support, GUI DEBs, classic Snaps
 - **TPM full disk encryption**
 - **Remote management** via Canonical Landscape
 - **Active Directory** login
 - **Distro infrastructure**: ISOs, testing, stable release tracks, documentation



Ubuntu Core Desktop

- **Advantages**

- **Stability:** Read-only system files, atomic updates, no dependency conflicts
- **Security:** Secure boot, read-only system files, encapsulation
- **Composability:** Defined modules which do not affect each other
- **Manageability:** Defined modules, atomic updates, single package format
- **Privacy:** Encapsulated apps with well-defined permissions



Thank you! Questions?



The Making of ...



snapcraft – Let's go snapping ...

- **snapcraft** creates Snaps, orchestrating disparate components and building systems into one cohesive **distributable package**
- It can **re-use DEB packages** from Ubuntu (of the Ubuntu LTS release the Core Snap used is based on).
- It's **extensible** and **new plugins** to leverage different technologies are being developed all the time. A few examples of its plugins are Java, Python, Catkin (ROS), Go, CMake, qmake, make, autotools, etc.



snapcraft – Let's go snapping ...

- **Single snapcraft.yaml file** that describes everything
- Defines apps, build process, build dependencies, runtime dependencies, interfaces
- Fully supported and integrated in **Launchpad**
- GitHub build service provided via <https://build.snapcraft.io/>
- **Detailed documentation** and tutorials at <https://snapcraft.io/>



ubuntu-image – Assemble your all-Snap OS

- The **magic tool** putting everything together
- Using a signed “assertion” file to define which Snaps end up inside the image
- Reads gadget .yaml to create **partitioning**
- Can build full disk images (i.e. SD card) or multi-partition images (i.e. to dd single img files to specific eMMC partitions on a pre-partitioned flash device)
- Available as a Snap! (`snap install ubuntu-image ...`)
- **Detailed documentation** at: <https://docs.ubuntu.com/core/en/guides/build-device/image-building>



Want to know more?



More info/links

- **Snap Store** and **home page** of Snap:
<https://snapcraft.io>
- **Discuss** your questions in the forums:
<https://forum.snapcraft.io/>
- **Documentation**:
<https://snapcraft.io/docs>
- Want to **learn snapping**? – Workshops!
<https://forum.snapcraft.io/t/40263>
- Let the **important people** talk about Snap:
<https://www.youtube.com/watch?v=ido6kGmSHWI>



More info/links

- Learn about **immutable OS** distributions:
<https://ubuntu.com/blog/ubuntu-core-an-immutable-linux-desktop>
- **Ubuntu Core Desktop – Introduction**
<https://discourse.ubuntu.com/t/ubuntu-core-desktop-deep-dive/>
- **Ubuntu Core Desktop – GitHub**
<https://github.com/canonical/ubuntu-core-desktop/>
- **Ubuntu Core Desktop – Installation HOWTO**
<https://www.omgubuntu.co.uk/2023/06/try-ubuntu-snap-desktop>
- **Ubuntu Core Desktop – Talk on Ubuntu Summit 2023**
<https://www.youtube.com/watch?v=ahWrhnjjYDk>



More info/links

- Ubuntu blogs from Oliver Smith about **optimizing performance of Snaps**:
 - <https://ubuntu.com/blog/how-are-we-improving-firefox-snap-performance-part-1>
 - <https://ubuntu.com/blog/how-are-we-improving-firefox-snap-performance-part-2>
 - <https://ubuntu.com/blog/improving-firefox-snap-performance-part-3>
 - <https://ubuntu.com/blog/firefox-snap-updates-and-upgrades>
- Want to watch some **snappy videos**? Here we go:
 - <https://www.youtube.com/watch?v=TfB6QwR2GYg>
 - <https://www.youtube.com/watch?v=ido6kGmSHWI>
 - <https://www.youtube.com/watch?v=m5QKJH9tDjQ>