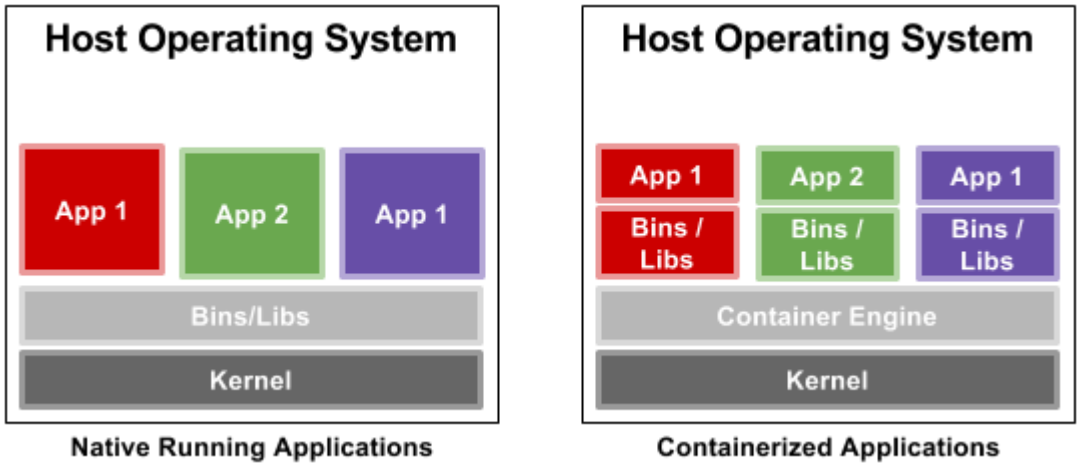
Snappy:- A Distro Agnostic, Container System, for Linux.

Comparing Containers and Virtual Machines

Difference between containers and virtual machines is that both have a specific purpose and place with very little overlap, and one doesn't obsolete the other.

Virtual machines are used when you want to host an entire operating system or ecosystem or maybe to run applications incompatible with the underlying environment. **Virtual machines are very heavy**.

Containers are a lightweight environment that you spin up to host one to a few isolated applications at bare-metal performance. The purpose of a container is to launch a limited set of applications or services and have them run within a self-contained sandboxed environment.



A Comparison of Applications Running in a Traditional Environment to Containers

Linux Process ID Namespaces (Kernel 2.6.24 released 2008), made it possible to have multiple “nested” process trees. Each process tree can have an entirely isolated set of processes. This can ensure that processes belonging to one process tree cannot inspect or kill - in fact cannot even know of the existence of - processes in other sibling or parent process trees.

This isolation prevents processes running within a given container from monitoring, or affecting processes running in another container. These containerized services are light and do not influence or disturb the host machine. This is one of the reasons data centers have chosen to adopt the technology.

A **P**rocess **I**dentification **N**umber (**PID)** is automatically assigned to each process when it is created. A process is nothing but running instance of a program and each process has a unique **PID** on a Unix-like system.

Linux Containers:-.

There are currently 4 Linux Container Systems:- **Docker, AppIimages, Flatpak & Snappy.**

Each one has their pros & cons. We will be considering Snappy.

What is Snappy?

Canonical describes a Snap as a universal Linux package which can work on most distro's.
Snappy is a software deployment and package management system originally designed and built by Canonical for the Ubuntu phone operating system (December 2014). The packages ar called **snaps** and the deamon for using them **snapd.**

- **snap** is both the command line interface and the application package format
- **snapd** is the daemon (background service) that manages and maintains your snaps
- **snapcraft** (October 2015) is the command and the framework used to build your own snaps
- **Snap Store** provides a place to upload your snaps, and for users to browse and install

A **snap** package for the Ubuntu Core system contains all its dependencies. This means a you can always be assured that there are no regressions triggered by changes to dependencies.

Although snaps were developed by Canonical they are supported on numerous, but not all linux distributions (https://docs.snapcraft.io/installing-snapd/6735)

Snaps are containerized applications holding all of a program's dependencies. They're designed to work securely within the Linux platform.

Snaps are basically an application compiled together with its dependencies and libraries – providing a sandbox environment in which the application can run.

Snaps are and fast and easy to install, they receive the latest updates, automatically updated daily and are isolated from the OS and other apps.

Snaps automatic update and roll-back features give flexibility for developers and a more seamless experience for users.

Snaps suit desktop, server and cloud environments.

Snaps are isolated from the OS via various security mechanisms, yet can still function as if it were installed by the standard means (exchanging data with the host OS and other installed applications).

Snaps are currently supported on over 40 Linux distros including Arch Linux, Debian, Fedora, Linux Mint, Manjaro, openSUSE, Solus, and Ubuntu. Instead of worrying about DEB, or RPM etc, you can simply a use a Snap package.

Do You already have snap installed?

```
echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:......./snap/bin
```

How to Install Snap if it's not installed.

https://docs.snapcraft.io/installing-snapd/6735
https://itsfoss.com/install-snap-linux/

Where are Snap Apps Stored on the Internet?

On the net https://snapcraft.io/store

The Ubuntu app store is used as the default back-end, but other stores can also be enabled.

Where are Snap Apps & Data Stored on Your Machine?

The **/snap** directory by default is where the files and folders appear on your system.

It has the following structure:

```
/snap/bin                    - Symlinks to snap applications
/snap/<snapname>/<revision> - Mountpoint for snap content
/snap/<snapname>/current     - Symlink to current revision, if enabled
```

A snap will take up more space than a normally installed app because it includes all the dependencies.

Install "Hello World" with Snap

```
$ sudo snap install hello
hello (stable) 2.10 from 'canonical' installed.
```

Snaps are updated automatically in the background every day.

Running a Snap.

Installed snaps are available under **`/snap/bin`** (**`echo $PATH)`**. This makes it accessible by just calling the command:

```
$ hello
```

unless you also have a copy of '**`hello`**' also installed in `/usr/bin/hello`. In that case you would have to call:

```
$ /snap/bin/hello
```

Refresh & Listing installed snaps

`Snaps` are updated automatically in the background every day but,

`'snap refresh'` Will manually update to the latest versions of all your installed snaps any time.

`'snap list'` Wll show a list of snaps installed on your system:

```
$ snap list
Name     Version        Rev   Tracking  Publisher   Notes
core     16-2.35.1      5419  beta      canonical✓  core
spotify  1.0.88.353     19    stable    spotify✓    -
vlc      3.0.4          555   stable    videolan✓   -
```

Revert if something goes wrong.

You can always revert back to the previous working state!

```
$ sudo snap revert hello
hello reverted to 2.10

$ hello
Hello, world!
```

With this rollback system built-in, you can confidently update knowing that you will always have a way to go back to the previous working state

Removing a snap.

`snap remove <snap name>` removes a snap in a snap.

In our **`'hello'`** case just do:

```
$ sudo snap remove hello
hello removed
```

Nice and clean, nothing is left-over! Application code, its run time dependencies and associated user data are all cleaned up. If your snap declared a service, they will as well be shut down and removed.

The Snap Manual.

`'man snap'` will give you the Snap Manual

Snap Help

The snap command lets you install, configure, refresh and remove snaps.

For more information about a command, run '**`snap help <command>`**'.

Note:- For security reasons the command `‘snap man <command>’` does not currently work. This may be fixed in the future.

Examples of snap help are:-

`‘snap help’` Note: A green tick (given the default color and unicode support) after a publisher’s name indicates that the publisher has been verified.

`'snap help --all'` provides a short summary of all the snap commands.

`‘snap help find’` find app. packages to install

`‘snap find hello-world’`

`‘snap find --narrow hello-world’`

Channels.

Channels are an important snap concept for developers and experimenters. They define which release of a snap is installed and tracked for updates. We are only looking briefly at ***risk-levels*** here.

A channel consists of three parts: tracks, risk-levels and branches.

- ***Tracks*** enable snap developers to publish multiple supported releases of their application under the same snap name.
- ***Risk-levels*** There are four risk-levels: ***stable***, ***candidate***, ***beta*** and ***edge*** that represent a progressive trade-off between stability and new features.
- ***Branches*** are optional and hold temporary releases intended to help with bug-fixing.

The complete channel name can be structured as three distinct parts separated by slashes:

`<track>/<risk>/<branch>`

By default Snaps are normally installed using the stable risk-level.

```
$ sudo snap install vlc
```

Use the `--channel` option to select a different risk-level. The following command will install the latest beta snap of VLC:

```
$ sudo snap install --channel=beta vlc
If the beta snap isn’t available, the next most stable snap will be installed.
```

Further information is available at *https://docs.snapcraft.io/channels/551* .

References:

All Snap Commands

https://www.mankier.com/8/snap#Commands

Install & Learn

https://itsfoss.com/use-snap-packages-ubuntu-16-04/

https://docs.snapcraft.io/getting-started/3876

https://codeburst.io/how-to-install-and-use-snap-on-ubuntu-18-04-9fcb6e3b34f9

https://itsfoss.com/install-snap-linux/

https://www.linux.com/learn/intro-to-linux/2018/5/get-started-snap-packages-linux

https://tutorials.ubuntu.com/tutorial/basic-snap-usage#0

Channels

https://docs.snapcraft.io/channels/551

Where to Find the snapcraft.io store.

https://snapcraft.io/store
https://forum.snapcraft.io/t/the-system-snap-directory/2817

Other Reading:- Infrastructure for container projects.

https://linuxcontainers.org/
https://www.linuxjournal.com/content/everything-you-need-know-about-linux-containers-part-i-linux-control-groups-and-process
https://www.linuxjournal.com/content/everything-you-need-know-about-linux-containers-part-ii-working-linux-containers-lxc