

**IN SPACE, NO ONE CAN HEAR YOU SNAP ...**



**DEV TREK THE NEXT GENERATION**

Igor Ljubuncic | Developer Advocate | Canonical | Stardate 99068.39



# Igor Ljubuncic

- Developer Advocate
- Linux
- Books
- Petrolhead





# What are we going to do?

- Talk about snaps, talk about you and me
- Snap, huh, yeah, what is it good for?
- Architecture, syntax
- Publishing to the Snap Store
- ?????
- Profit!



# “What be snaps?”

-- Niccolo Machiavelli





# What be snaps?

- Snaps are self-contained software packages
- Work on all major Linux distributions (40+) without modification
- Simple to create and publish (with reason)
- Safe automatic updates

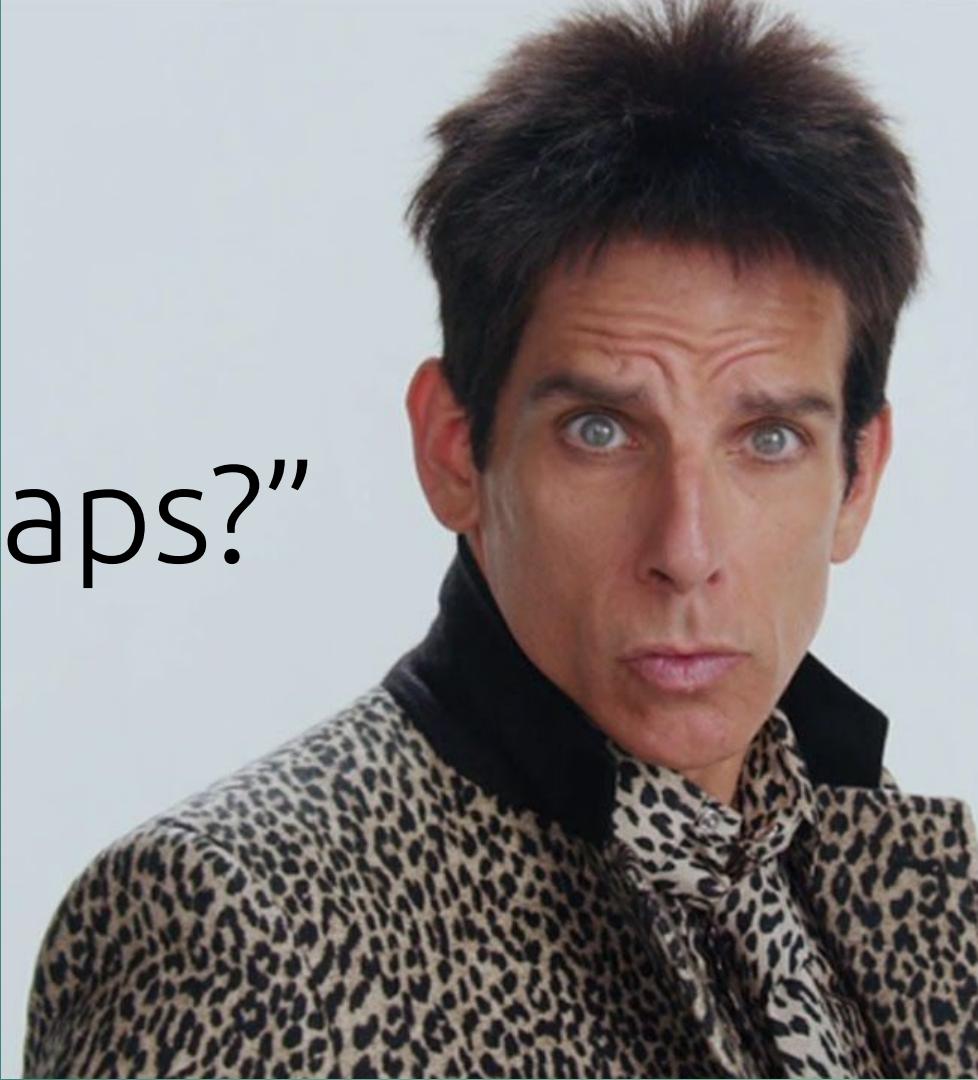


# Traditional methods of software delivery

- Knowledge of Linux internals
- Complex packaging code
- Not transactional
- Unbounded
- Risky

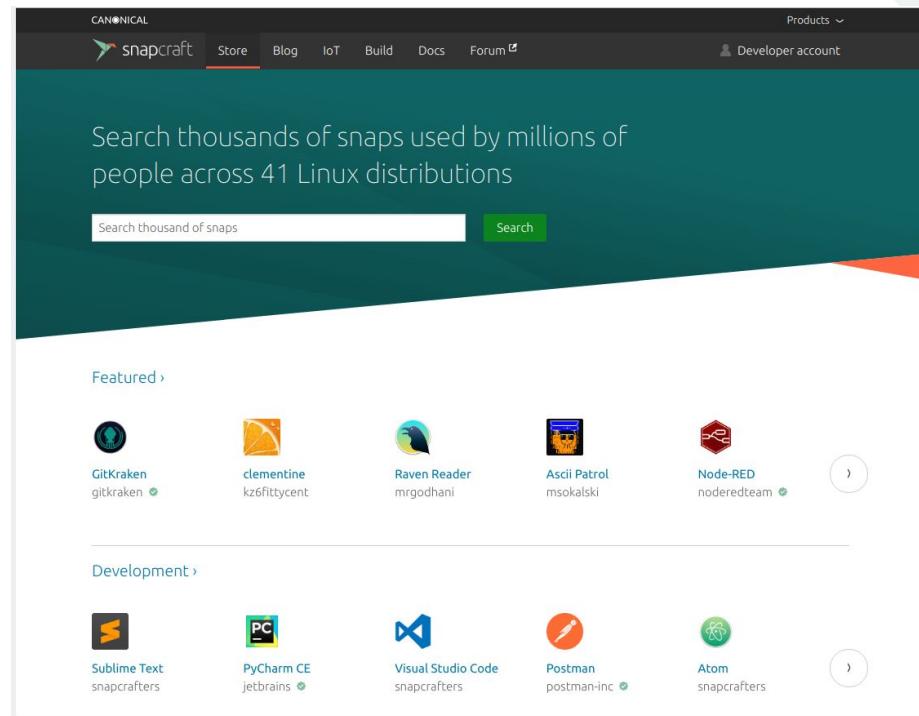
“But why snaps?”

-- Zoolander



# Out of the box discovery by millions

- All Ubuntu versions since 14.04
- 40+ Linux distributions
- All software welcome,  
regardless of license or cost



The screenshot shows the Canonical Snapcraft Store website. The header includes the Canonical logo, a search bar, and links for Store, Blog, IoT, Build, Docs, and Forum. A "Developer account" button is also present. The main content area features a search bar with the placeholder "Search thousand of snaps" and a "Search" button. Below the search bar, a teal banner states: "Search thousands of snaps used by millions of people across 41 Linux distributions". The page then displays two sections: "Featured" and "Development". The "Featured" section shows icons for GitKraken, clementine, Raven Reader, Ascii Patrol, and Node-RED. The "Development" section shows icons for Sublime Text, PyCharm CE, Visual Studio Code, Postman, and Atom. Each software entry includes the name, developer, and a small green circular icon with a checkmark.

Search thousand of snaps

Search

Featured ›

GitKraken  
gitkraken

clementine  
kz6fiftycent

Raven Reader  
mrgodhani

Ascii Patrol  
msokalski

Node-RED  
noderedteam

Development ›

Sublime Text  
snapcrafters

PyCharm CE  
jetbrains

Visual Studio Code  
snapcrafters

Postman  
postman-inc

Atom  
snapcrafters

# Click & Install

- One snap to rule them all ...
- Consistent behaviour
- Fewer moving parts



The screenshot shows the Canonical snapcraft store interface. At the top, there is a navigation bar with links for Store, Blog, IoT, Build, Docs, and Forum. On the right side of the bar, there is a 'Developer account' link. Below the navigation bar, the page title is 'Live For Speed' by 'Taqi Raza (mmtrt) · Games'. To the right of the title, there is a dropdown menu for 'latest/stable S3-6T' and a green 'Install' button. The main content area features a large image of a racing car with the 'Live for Speed' logo. Below the image, there is a menu with options: Training, Multiplayer, Single player, Hot lapping, Replays, and Options. At the bottom of the image, there is a button labeled 'Unlock Live for Speed'. The footer of the page contains the text 'Live for Speed 0.67 © 2002-2018 Scowen Roberts, Eric Bailey, Victor van Vlaardingen'. To the right of the main content, there is a sidebar titled 'Single Player' with options for Skill level (Novice, PRO), Race (Add AI driver, Start grid 1), and Save (Local, Network). At the bottom right of the sidebar, there is a 'back' button. Below the main content, there is a section titled 'Live for Speed (LFS) is a online racing simulator.' followed by a detailed description of the game's features and a note about using a steering wheel. To the right of this section, there is a 'Details for Live For Speed' section with a 'License' link (Proprietary) and a 'Last updated' link (12 March 2019). At the bottom right, there is a 'Share this snap' link.

Live for Speed (LFS) is a online racing simulator.

Live For Speed is a serious racing simulator. No arcade modes, no steering aids - YOU have to do the driving.

The forces affecting the car's motion are simulated individually at each wheel. We recommend the use of a steering wheel, because although you can use the keyboard and/or mouse, a wheel and pedals provide a lot more accuracy. The

Details for Live For Speed

License Proprietary

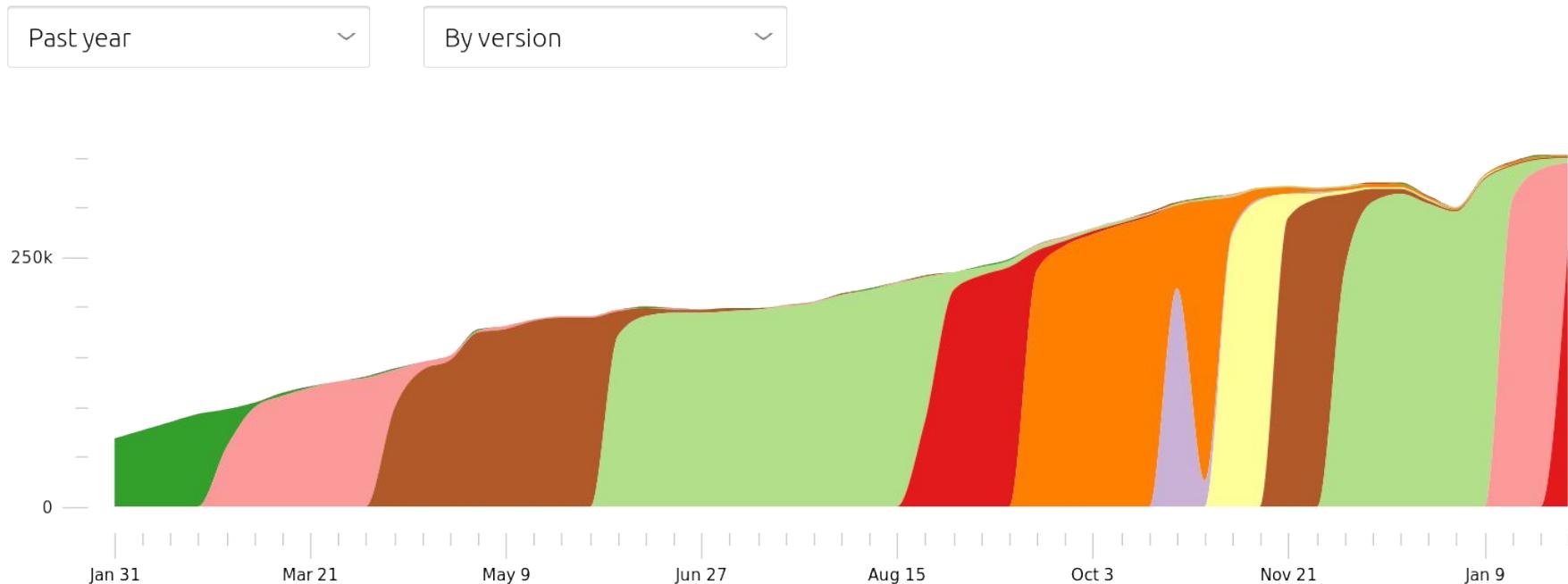
Last updated 12 March 2019

Share this snap



# Analytics

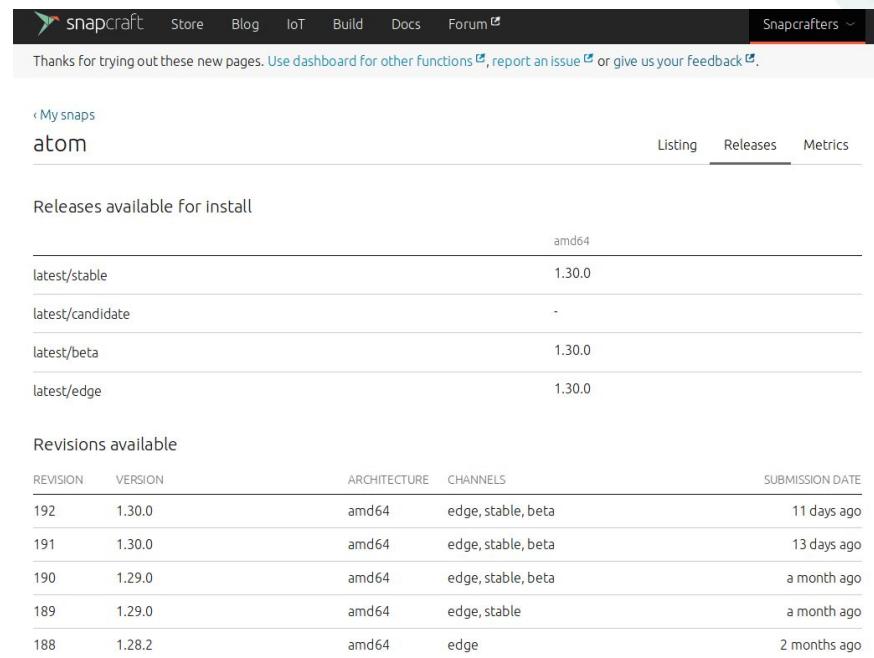
- See the growth across geos & measure impact of each release





# Full control of update lifecycle

- Updates published instantly
- Cryptographically signed
- Risk-based channels
- OS frozen or EOL? Not a problem!
- Updates, 4 times/day



The screenshot shows the Snapcraft interface for the 'atom' snap. The top navigation bar includes links for Store, Blog, IoT, Build, Docs, Forum, and Snapcrafters. A message at the top says, "Thanks for trying out these new pages. Use [dashboard](#) for other functions, [report an issue](#) or give us your feedback." The main content area shows the 'Releases available for install' section, which lists four releases: latest/stable (1.30.0, amd64), latest/candidate (-), latest/beta (1.30.0, amd64), and latest/edge (1.30.0, amd64). Below this is the 'Revisions available' section, which lists five revisions: 192 (1.30.0, amd64, edge, stable, beta, 11 days ago), 191 (1.30.0, amd64, edge, stable, beta, 13 days ago), 190 (1.29.0, amd64, edge, stable, beta, a month ago), 189 (1.29.0, amd64, edge, stable, a month ago), and 188 (1.28.2, amd64, edge, 2 months ago).

REVISION	VERSION	ARCHITECTURE	CHANNELS	SUBMISSION DATE
192	1.30.0	amd64	edge, stable, beta	11 days ago
191	1.30.0	amd64	edge, stable, beta	13 days ago
190	1.29.0	amd64	edge, stable, beta	a month ago
189	1.29.0	amd64	edge, stable	a month ago
188	1.28.2	amd64	edge	2 months ago

**Snap a day (or rather four times a day) keeps the doctor away!**



# “How does it work?”

-- Friedrich Nietzsche





# Snap major components

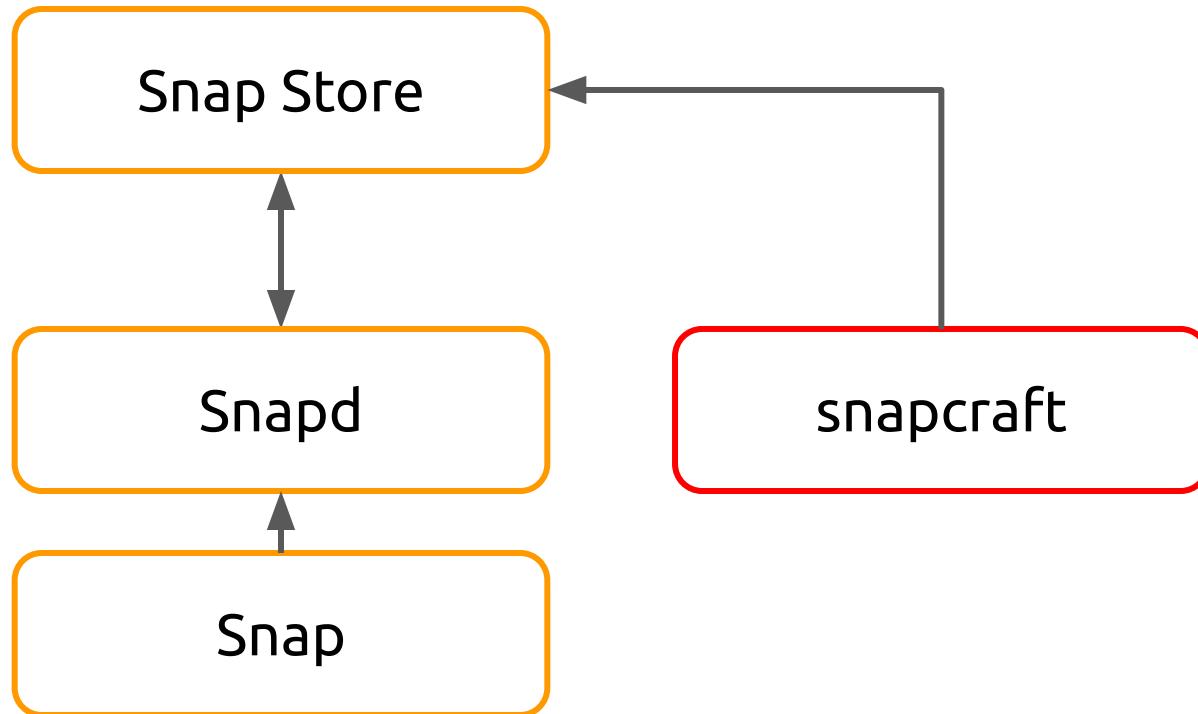
- Snapd - background service to install and run snaps
- Snap - userspace component of the snapd service, e,g:

```
snap install foo
```

- Snap Store - central, online repository of snap applications
- Snapcraft - command line tool to build and publish snaps



# The world of snaps





# Snapcraft capabilities

- Command-line tool to build and publish snaps
- Snaps as final artifact
- Snap = single compressed SquashFS filesystem
- Application code + declarative metadata
- Format extension .snap



# Where does it all start?

- Build configuration in `snapcraft.yaml`
- YAML syntax
- Similar to RPM spec file ...



# Snapcraft.yaml (just an example)

```
name: wethr
version: "1.4.0"
summary: Command line weather tool.
description:
  Get current weather.
base: core18
apps:
  wethr:
    command: wethr
    plugs:
    - network
parts:
  wethr:
    plugin: nodejs
    source-tag: "v1.4.0"
    source: https://github.com/twobucks/wethr.git
```



# Metadata

Mandatory fields: name, version, summary, description

name: wethr

version: "1.4.0"

summary: Command line weather tool.

description:

Get current weather.



# Metadata

	<b>Definition</b>	<b>Type</b>	<b>Length</b>	<b>Note</b>
<b>Name</b>	Name	String	1-40	Unique
<b>Version</b>	App version	String	1-32	No semantic meaning
<b>Summary</b>	Brief	String	1-79	
<b>Description</b>	Full	String	High	



# Confinement level

- Isolation & security
- Snaps cannot change other snaps
- Snaps cannot change system
- Different confinement levels
- Interfaces



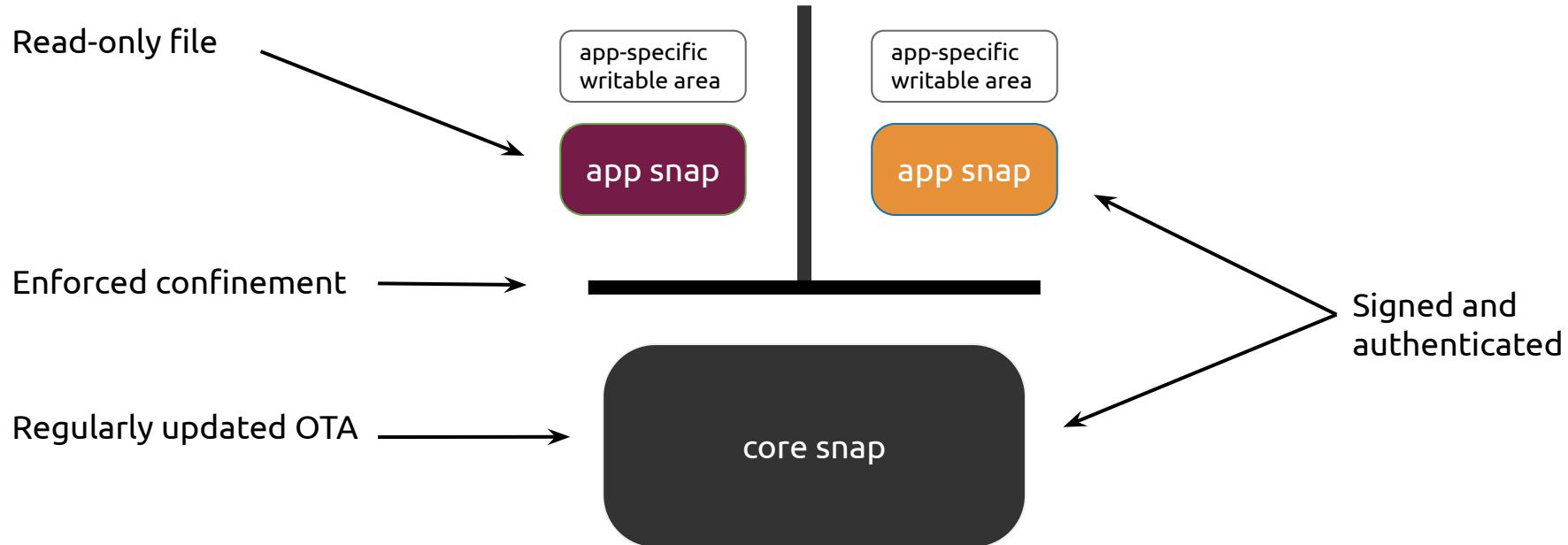
# Confinement level

	<b>Strict</b>	<b>Devmode</b>	<b>Classic</b>
<b>Access to network</b>	N	Y	System
<b>Access to home dir</b>	N	Y	System
<b>Access to audio</b>	N	Y	System
<b>Access to webcam</b>	N	Y	System
<b>Access to display</b>	N	Y	System
<b>Used for</b>	Preferred	Troubleshooting	Stopgap measure
<b>Other</b>	Interfaces override		Requires review

Wethr example - confinement: strict



# Application isolation





# Base

- In general, snaps cannot see the root FS on end user systems
- Applications need some location to act as the root FS
- Base - special kind of snap /w minimal set of libraries
- Mounted as root FS for applications
- The core18/20 base is recommended

base: core18



# Build definition

```
apps:  
  wethr:  
    command: wethr  
    plugs:  
    - network  
  
parts:  
  wethr:  
    plugin: nodejs  
    source-tag: "v1.4.0"  
    source: https://github.com/twobucks/wethr.git
```



# Build definition - apps

apps:

  wethr:

    command: wethr

    plugs:

      - network

- apps: defines the application(s) in the snap
- wethr: defines a block for the wethr application
- command: defines the path to executable
- plugs: access resources not available under strict confinement



# Build definition - parts

```
parts:  
  wethr:  
    plugin: nodejs  
    source-tag: "v1.4.0"  
    source: https://github.com/twobucks/wethr.git
```

- parts: sources needed to assemble the app
- plugin: language specific tools
- source-tag: tag for source repositories under version control
- source: URL or path to download for the build



# Snapcraft commands

build	Builds artifacts based on the snapcraft.yaml.
clean	Remove content - cleans downloads, builds or...
init	Initializes a snapcraft project.
push	Pushes a snap to the online snap store.
register	Registers a snap with the online snap store.
snap	Create a snaps.

And more ...



# Application build process

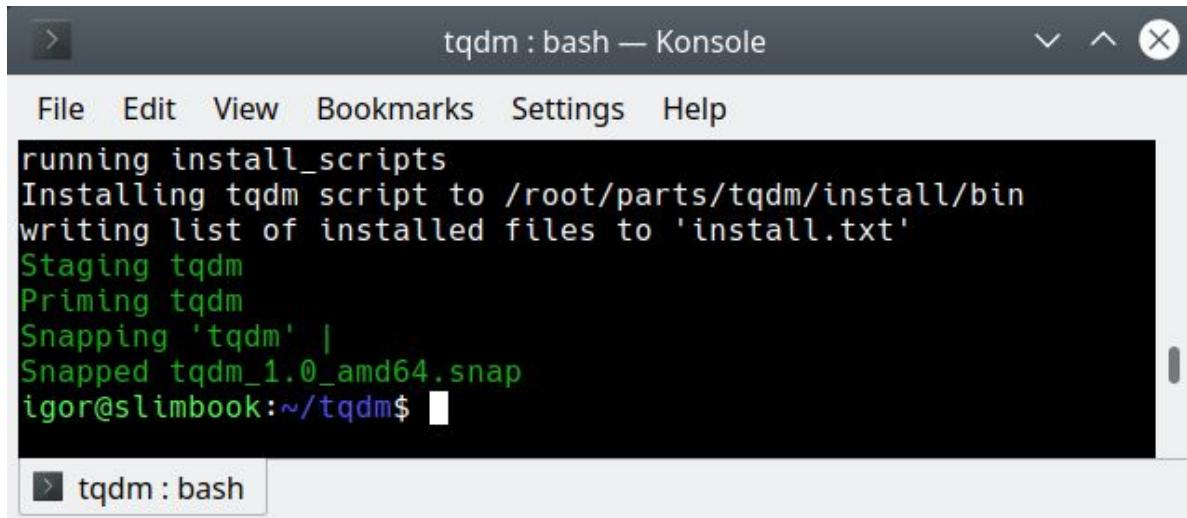
- Clean environment
- No library dependencies or conflicts

snapcraft



# Snap created

- There could be errors - later in the series
- Successful build - <name>.snap



```
running install_scripts
Installing tqdm script to /root/part
parts/tqdm/install/bin
writing list of installed files to 'install.txt'
Staging tqdm
Priming tqdm
Snapping 'tqdm' |
Snapped tqdm_1.0_amd64.snap
igor@slimbook:~/tqdm$
```



# Snap file format

- Unpack snap or mount as loopback device:

```
unsquashfs <file>.snap
```

```
mount <file>.snap <mount point> -t squashfs -o loop
```

```
drwxr-xr-x 2 igor igor 4096 Dec  5 12:48 bin/
-rwxr-xr-x 1 igor igor    61 Dec  5 12:48 command-tqdm.wrapper*
drwxr-xr-x 3 igor igor 4096 Dec  5 12:48 etc/
drwxr-xr-x 4 igor igor 4096 Dec  5 12:48 lib/
drwxr-xr-x 3 igor igor 4096 Dec  5 12:48 meta/
drwxr-xr-x 3 igor igor 4096 Dec  5 12:48 snap/
drwxr-xr-x 6 igor igor 4096 Apr 16 2018 usr/
```

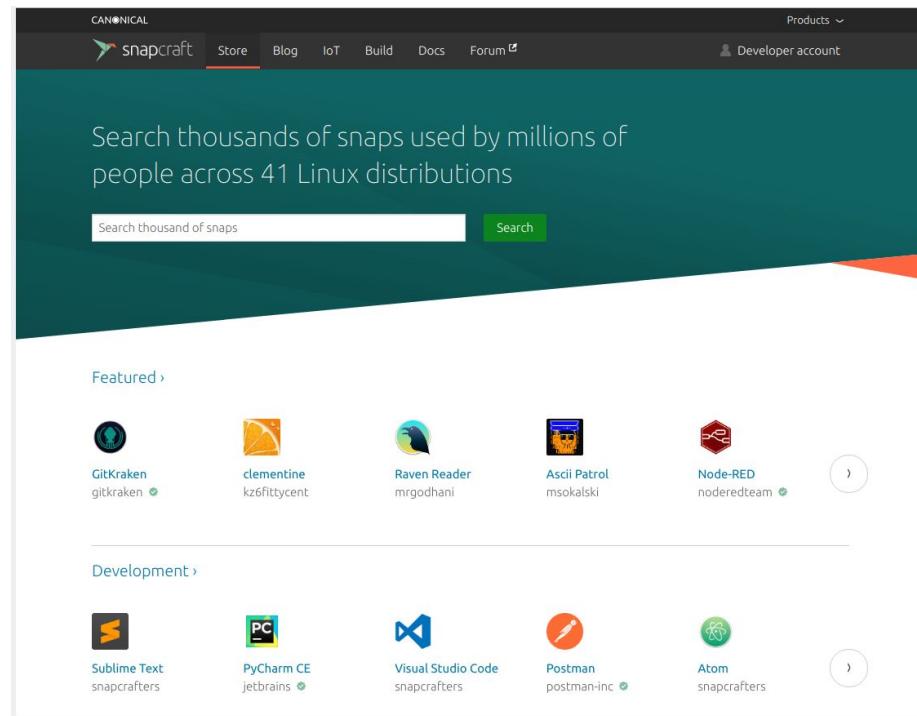


# Snap file format

- Concept similar to LD\_LIBRARY\_PATH
- \$SNAP\* environment variables
  - \$SNAP (install path, RO)
  - \$SNAP\_DATA (path in /var, RW)
  - \$SNAP\_USER\_DATA (path in /home, RW)
  - And others

# Publishing a snap

- Create your dev account
- Register your app's name
- Release your app





# Publishing a snap

```
snapcraft login
```

```
snapcraft push --release=<channel> <file>.snap
```



# Store channels

<track>/<risk>/<branch>

- Track - trade-off between stability and new features
- Risk - multiple supported releases of the same application
- Branch - optional for temporary releases and bug-fixing

--channel=latest/edge



# Tracks

- Default = latest
- Minor updates, e.g 2.0.1, 2.0.2
- Major updates, e.g. 2.1, 2.2
- Long-Term Support, e.g. 3.2, 4.1



# Risk

- Most important aspect of channels
- Levels: stable, candidate, beta, and edge

--stable, --channel=stable, --channel=latest/stable

- Users can switch between channels



# Upload to the store

- Do not use stable right away
- Staged deployment

```
sudo snapcraft push --release=beta <file>.snap
```

- Automated checks + manual review
- Compelling page



# “Questions?”

-- Audience (you)





# Thank you!

igor.ljubuncic@canonical.com

