

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

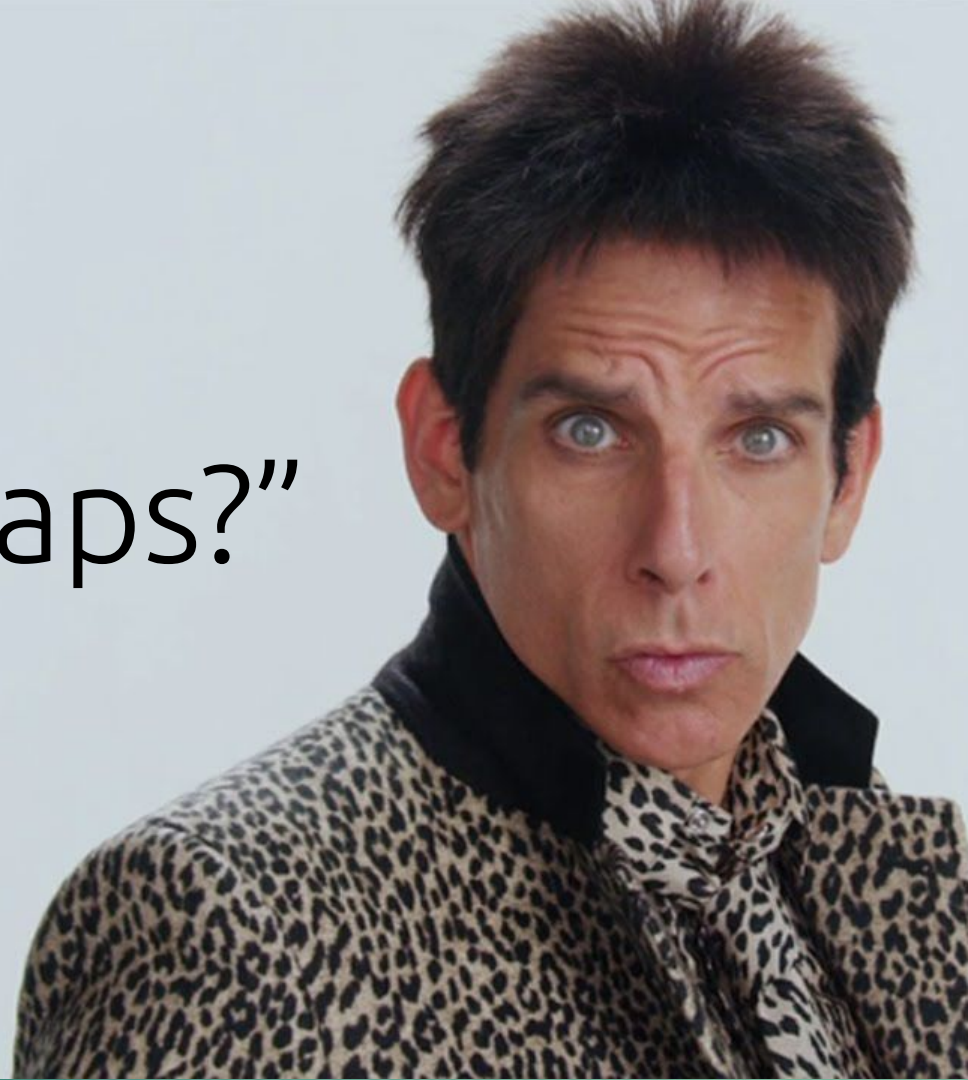
- Knowledge of Linux internals
- Complex packaging code
- Not transactional
- Unbounded
- Risky



“But why snaps?”

-- Zoolander

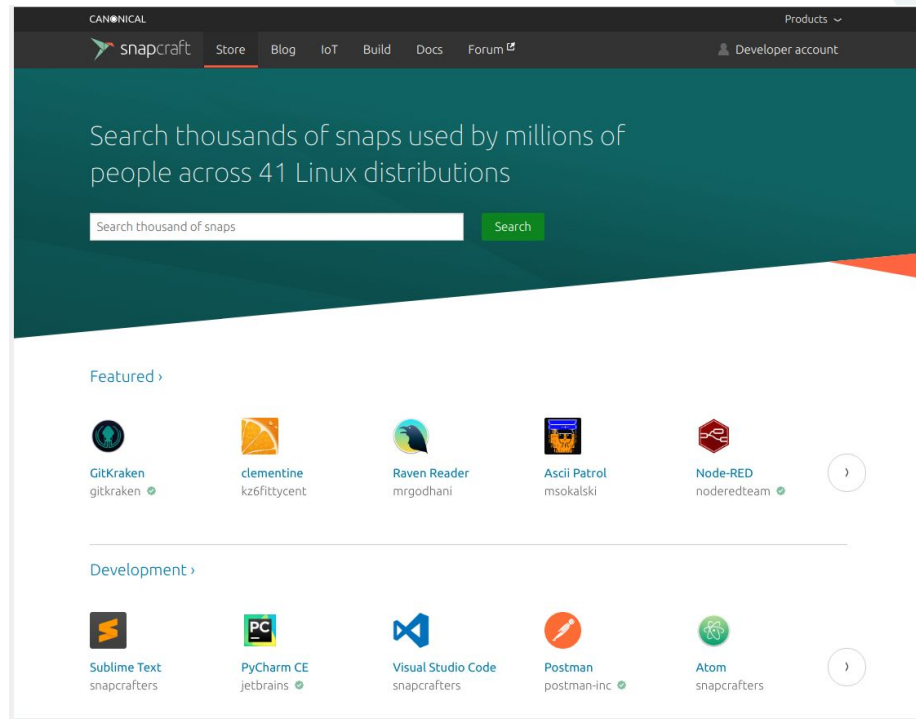
@ Copyright Zoolander 2 by Panorama Films, Red Hour Films and
Scott Rudin Prod; used for illustration purposes only





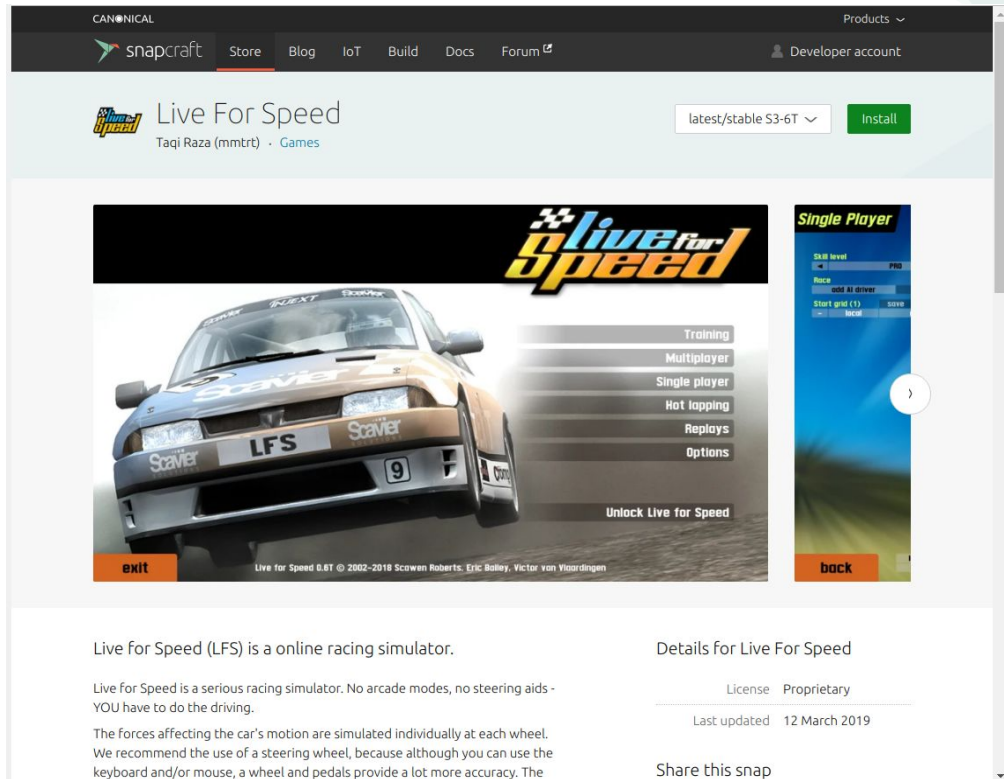
Out of the box discovery by millions

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



Click & Install

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




The screenshot shows the Snapcraft store page for the 'Live For Speed' snap. The header includes the Canonical logo, the Snapcraft logo, and navigation links for Store, Blog, IoT, Build, Docs, and Forum. A 'Products' dropdown menu is visible in the top right. Below the header, the snap name 'Live For Speed' is displayed, along with the developer 'Taqi Raza (mmrt)' and a 'Games' category tag. A version selector shows 'latest/stable S3-6T' and an 'Install' button. The main content area features a large image of the game's title screen, which includes a car and a menu with options like Training, Multiplayer, Single player, Hot lapping, Replays, and Options. To the right of the main image is a 'Single Player' preview window showing game settings. Below the main image, there is a description of the game as an online racing simulator, followed by details about its license (Proprietary) and the last update date (12 March 2019). A 'Share this snap' button is located at the bottom right.

CANONICAL

Products ▾


snappy Store Blog IoT Build Docs Forum

Developer account

 Live For Speed

Taqi Raza (mmrt) · Games

latest/stable S3-6T ▾ Install



Single Player

Skill level: PRO

Race: 1000 km AI driver

Start grid (1): 5310

Local

back

Live for Speed 6.6T © 2002-2018 Scawen Roberts, Eric Bailey, Victor van Vaaingen

Live for Speed (LFS) is an 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

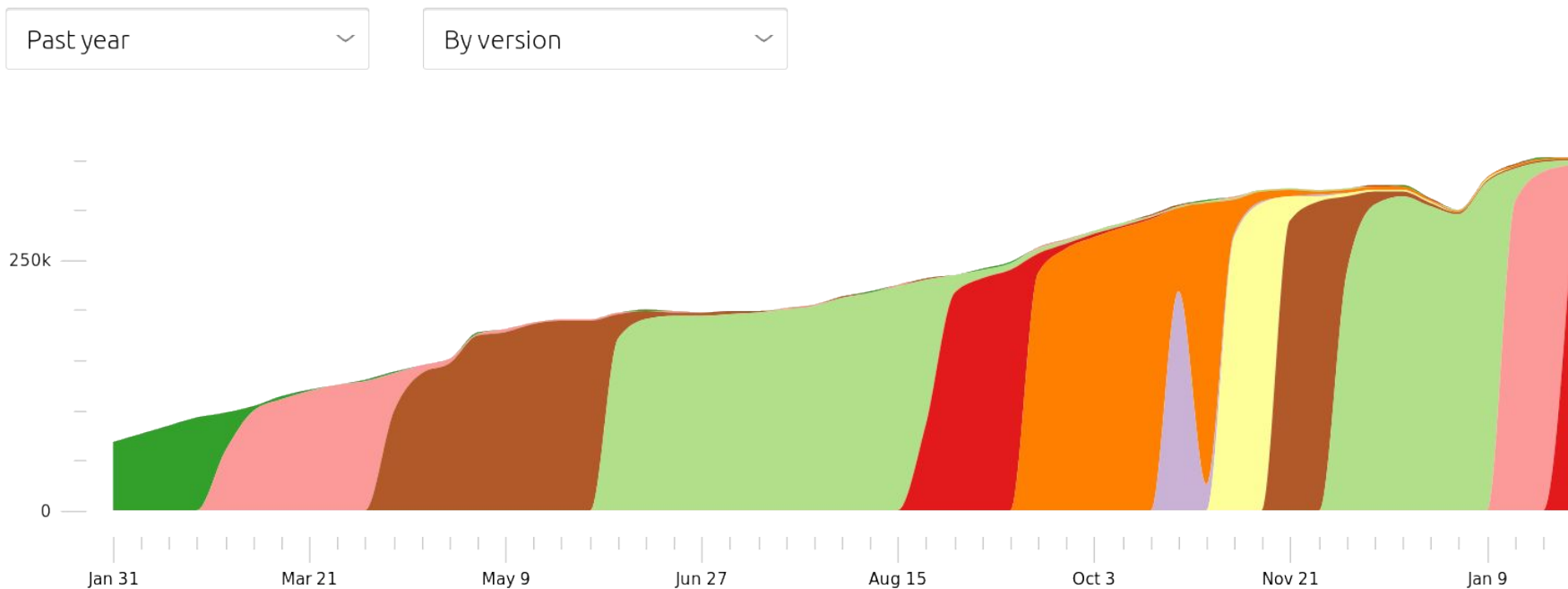
Last updated: 12 March 2019

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

snapcraft Store Blog IoT Build Docs Forum [Snapcrafters](#)

Thanks for trying out these new pages. [Use dashboard for other functions](#), [report an issue](#) or [give us your feedback](#).

My snaps

atom Listing Releases Metrics

Releases available for install

	amd64
latest/stable	1.30.0
latest/candidate	-
latest/beta	1.30.0
latest/edge	1.30.0

Revisions available

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





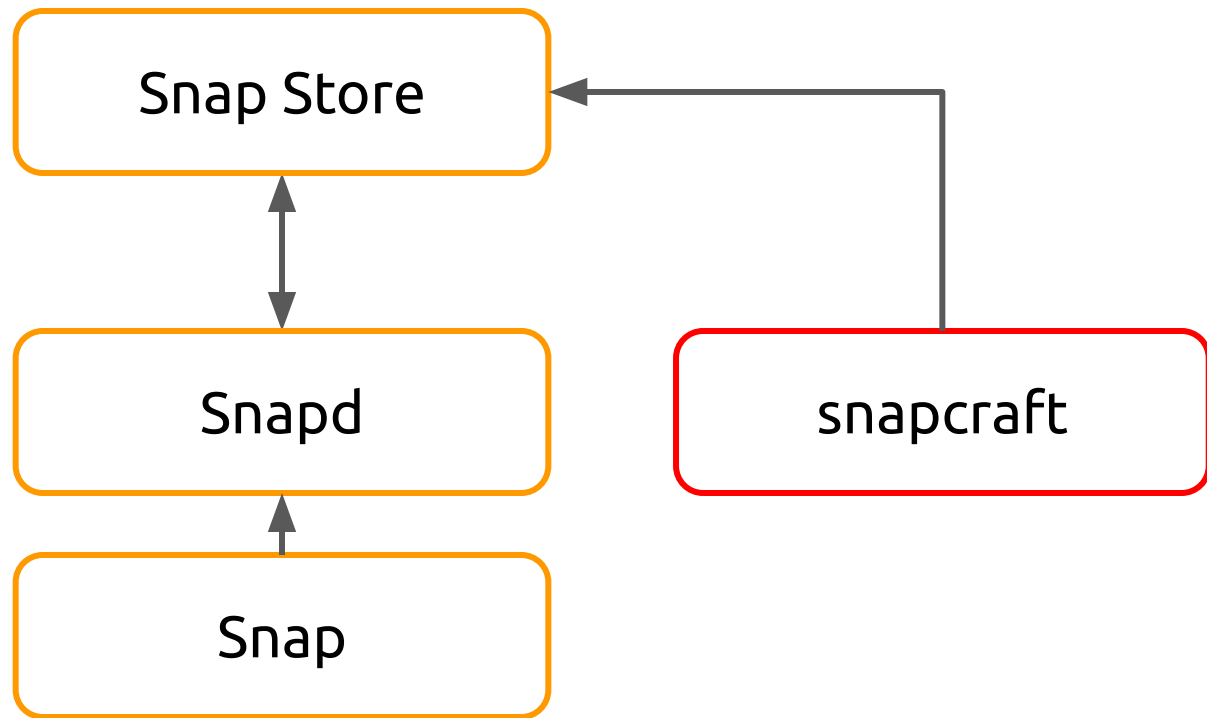
Snap major components

- 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

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



Confinement level

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

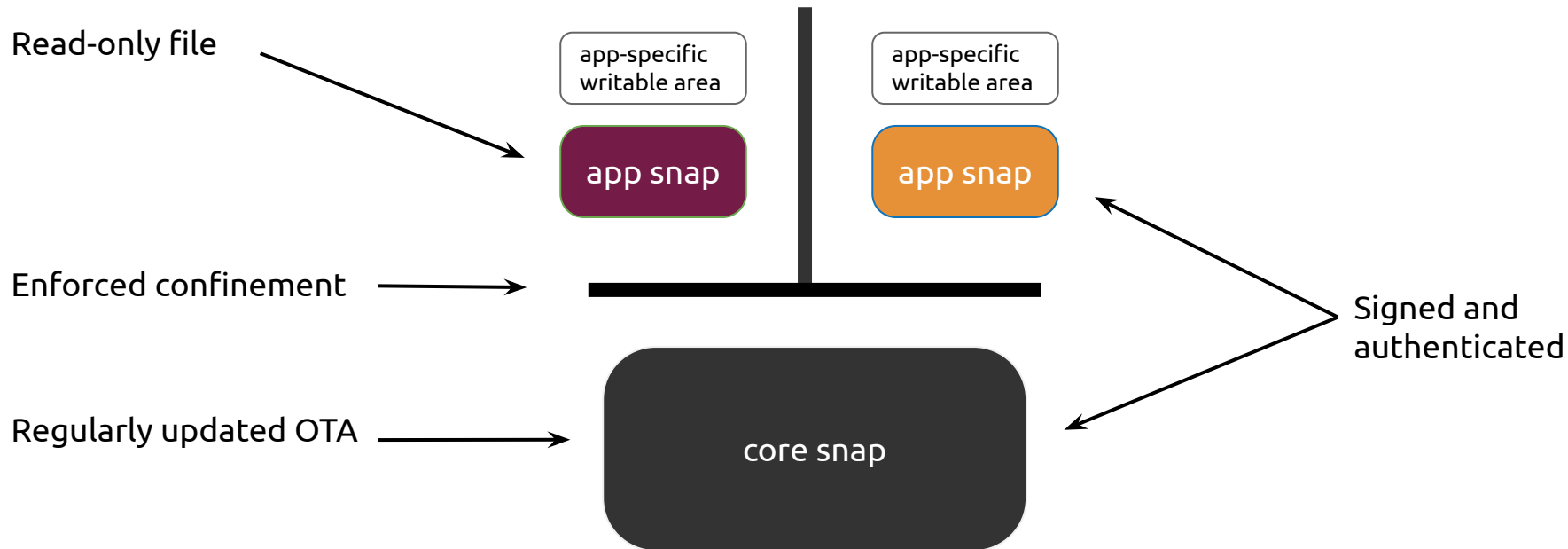


Confinement level

	Strict	Devmode	Classic
Access to network	N	Y	System
Access to home dir	N	Y	System
Access to audio	N	Y	System
Access to webcam	N	Y	System
Access to display	N	Y	System
Used for	Preferred	Troubleshooting	Stopgap measure
Other	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

 plugins:

 - 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

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

```
tqdm : bash — Konsole
File Edit View Bookmarks Settings Help
running install_scripts
Installing tqdm script to /root/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$
```

tqdm : bash



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/
```

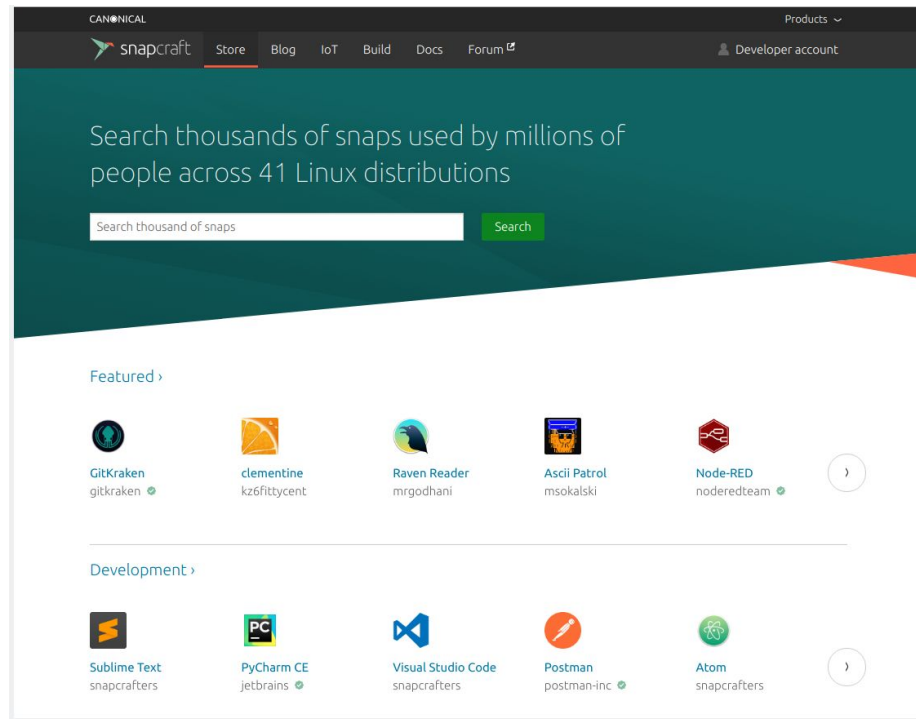


Snap file format

- 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

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

- Automated checks + manual review
- Compelling page



“Questions?”

-- Audience (you)





Thank you!

igor.ljubuncic@canonical.com

