CANONICAL

Optimising IoT bandwidth with delta updates

March 2019

Introduction

With more devices connecting to the internet as part of the ever-evolving and growing IoT industry, the demands on bandwidth increase in parallel. For any organisation deploying devices in the field, consideration needs to be given to the amount and frequency of data that will be transmitted to and from the device back to the cloud for analysis. In situations where those devices could start to grow into the thousands and beyond, the cost of transferring such data has the possibility to escalate quickly. In addition, the increased volume of data being transferred could reduce the IoT solution's effectiveness by creating a high latency environment.

The effect of potential bandwidth and latency problems will vary between use cases depending on data transfer frequency and volume. Due to IoT solutions being so diverse in their nature, in many scenarios data transfer times may not be time critical and data payloads so small that high latency isn't a significant concern. However, all organisations should consider an optimum mechanism in which to transfer data in the most resourceful and cost-effective way possible and to suit their specific requirements.

The data in question isn't just that which drives new insight and value to businesses but also that which is needed to keep remote device operating systems up to date, secure, and running the latest applications. And for that purpose, deploying software where only the delta change is updated rather than the whole application is a much more attractive option. Snaps, the universal Linux application packaging format, features exactly this attribute. As snaps run on over 40 Linux flavours and with Linux such a popular choice for embedded devices, snaps provide the necessary requirements to deploy various types of software in IoT environments.

This whitepaper will explore snap deltas in more detail including use cases of how over-the-air (OTA) updates can be delivered efficiently to IoT devices automatically, reducing bandwidth requirements while effectively deploying software updates.

Snap updates on IoT devices

Snaps, running on an Ubuntu Core, Ubuntu Classic or other Linux-based systems, provide several advantages for IoT and connected devices. Key among these is the ability for snaps to be updated in a safe and transactional manner. This makes the prospect of automated updates more attractive, because most failure scenarios are handled transparently by the snap system, freeing the publishers from the technical aspects of the update and allowing them to focus on releasing up-to-date versions of their software.

With snaps, publishers can release a new software version and have it be deployed across their installed base of devices without user intervention in a matter of hours.

Here we can see the typical update cycles for several common and popular snaps. For each graphic, the vertical axis shows the percentage of devices that have each version of the corresponding snap. The horizontal axis shows dates for the past 3 months. Each different version is denoted by a differently-colored block.

These graphics show that, once a snap is released, devices begin updating to it almost immediately. After a week, 90% or more of all devices in the field will have received the update.

Figure 1 - Release cadence and update adoption of the Core snap

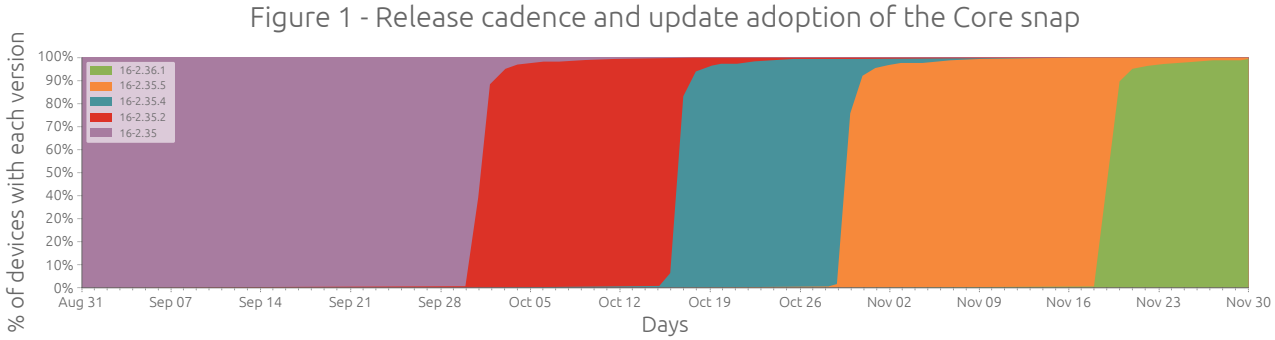


Figure 1 shows the release cadence and update adoption for the Core snap.The Core snap is a special kind of snap that provides a minimal set of libraries common to most applications and is required on all snap-based devices. This snap has new releases about once a month, although they can be more frequent if an urgent fix or update is required. Here, we can see that once a new version is released, about 90% of devices will receive the update within a week.

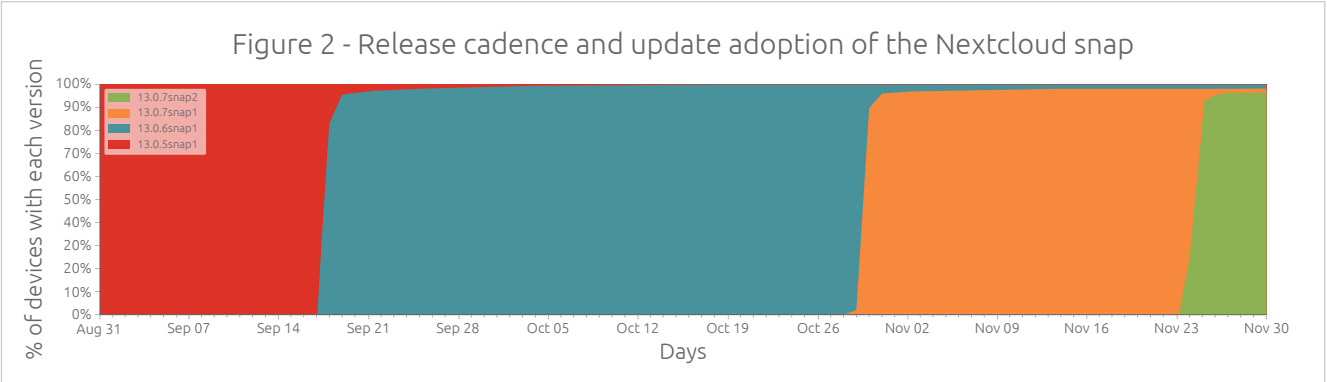


Figure 2 - Release cadence and update adoption of the Nextcloud snap

Figure 2 shows data for the popular Nextcloud self-hosted cloud server software has a one-month release cadence where we can observe similar update behaviour, with new releases being updated on over 90% of devices after 1 week.

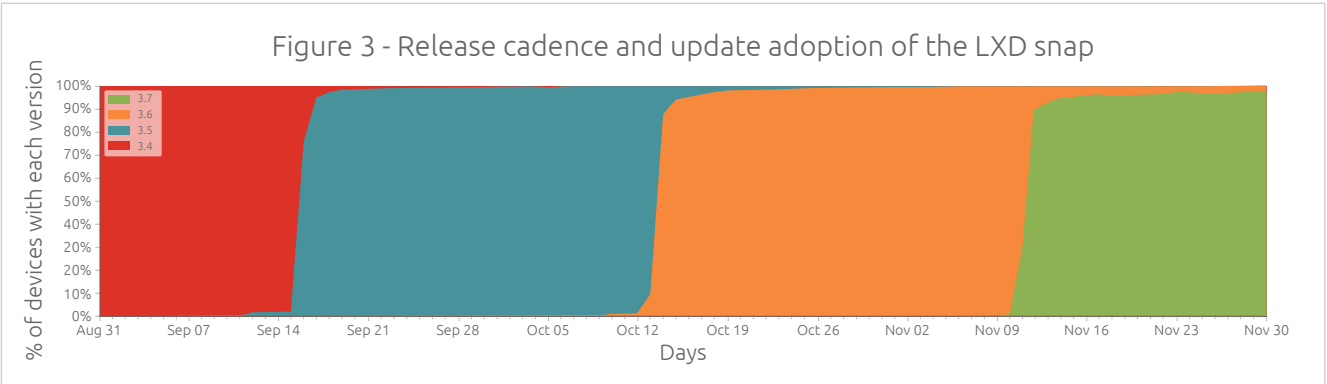


Figure 3 - Release cadence and update adoption of the LXD snap

Figure 3 shows data for Canonical's LXD software for managing and creating Linux containers. LXD is distributed as a snap and releases stable versions every 2 to 4 weeks, continuing the trend of 90% of clients being updated within one week of release.

For device manufacturers and operators, this means being able to react with security updates to entire fleets of devices, or deliver new functionality in a very homogeneous fashion to all users of a family of devices.

However, pushing software updates which can range from a few to a hundred megabytes or more, could potentially incur costs in time and bandwidth which organisations may be conscious about. IoT fleets consisting of thousands of devices or deployed in bandwidth-constrained environments, such as cellular uplinks, are specific cases where bandwidth usage becomes an important issue.

In order to reduce the amount of data needing to be transferred during update operations, the snap system supports a form of data compression known as snap deltas.

What are snap deltas?

Snap deltas constitute a form of differential data compression, where only the data that changed between the snap currently on the device (the source) and the snap the device intends to update to (the target) is transmitted.

Snap deltas are only applicable if a device is updating an already-installed snap, not for new installs. Fortunately, the most common case with connected devices is updating some of the snaps that were pre-installed when the device was provisioned in the factory, rather than new installs.

Snaps are downloaded from the Snap Store, a repository which enables publishing and discovery of software packaged in the snap format. When the publisher of a snap (in the case of devices, this is commonly the device manufacturer) releases a new revision (target) of their snap, the Snap Store servers look at the most likely source revisions that devices in the field may have, and automatically generates snap deltas between those revisions. That enables devices with a matching source revision to apply the delta and obtain a snap identical to the desired target, which is stored locally as if it had been downloaded directly and functions in all respects just like the full target snap. It can be used for all snap operations, including upgrading and rolling back.

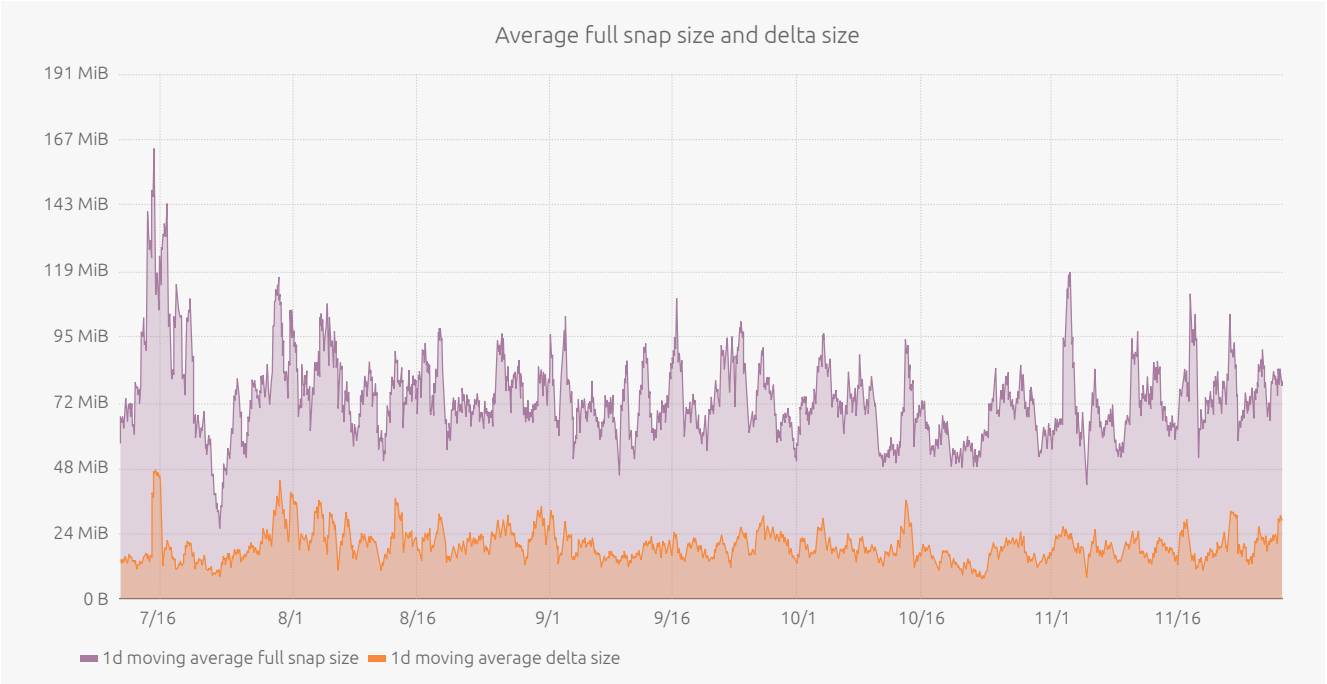
It's important to note that this happens without any change in the development workflow: snap publishers don't have to do anything special to benefit from snap deltas - the Snap Store automatically generates the deltas and provides them to clients. This also means that snap deltas are available regardless of how a developer builds and publishes their snap: snaps built locally using Snapcraft (the command-line tool developers can use to package their software as snaps and publish it to the Snap Store), snaps built using the build.snapcraft.io service, and snaps built using any other mechanism or tool such as electron-builder or CI service such as Travis, still result in the proper snap delta files being made available to clients.

The algorithm to apply the delta on the device to the source snap in order to obtain the target snap does incur a slight processing delay, but that is often eclipsed by the reduced download time and bandwidth savings. The processing delay is even more negligible in the very common case of unattended updates, where even on a low-power device, the delta application delay is in the order of seconds.

The following table shows some typical snaps of varying sizes, the ratio of delta vs. full download size, and the time spent applying the delta, both on a configuration that can be found on IoT devices, and on a typical PC-class system.

Snap name	Full download size	Delta download size	Delta size compared to full download (full download = 100%)	Time to apply delta, Broadcom BCM2837 Quad Core ARM Cortex-A53 USB storage	Time to apply delta, Intel Core i7-4500U CPU @ 1.80GHz SSD storage
alsa-utils	6.5MiB	62.8KiB	0.94%	.45s	0.045s
nextcloud	197MiB	112MiB	56.85%	24.0s	1.16s
core	87.9MiB	27.55MiB	31.34%	10.75s	0.42s
pc-kernel	154.77MiB	54.20MiB	35.01%	18.9s	.788s

The gains from delta compression are variable and depend on a multitude of factors. Historically, however, the delta download size is around 30% of the full download. The below graph shows average sizes for full snaps and deltas, computed over a 1-day window, during a 4.5-month period. This illustrates the ratio between a full snap and the corresponding delta.



Upload and delta generation

It's worth mentioning that, while the snap delta technology is most beneficial for downloads performed from clients, it can also be used to reduce the time and bandwidth consumed by a developer when publishing a snap. While working on packaging new software, developers frequently upload (push, in snap terminology) new versions of their snap package to the Snap Store. In this scenario, the Snapcraft packaging tool will generate a delta file containing only the changes from the most recent revision to the current one, and upload that to the Snap Store. The Store then applies the delta to result in the same snap that would have been uploaded, verifying the contents with a digital signature. As with download deltas, this happens transparently with no changes in the development workflow.

Snap deltas in use

Dell Edge Gateways

Dell's IoT technology is focused on intelligent gateways that can operate reliably in extreme temperatures and help connect endpoints even in the most challenging industrial or enterprise environments.

The Dell Edge Gateway 5000 and 3000 series devices are designed to aggregate, secure and relay data from diverse sensors. A typical Dell Edge Gateway is shipped with custom gadget and kernel snaps, as well as several snaps which provide additional system functionality such as advanced networking, Bluetooth support, and cellular connectivity.

Provided the device has internet connectivity, and given the typical update cadence for the mentioned snaps, the bandwidth consumption of one of the gateways to update snaps, looks like this on a monthly basis (per individual device and per fleet of 10,000 devices, with snaps as shipped on 3000-series devices):

Snap name	Updates/ month avg	Full snap size avg (MB)	Delta size avg (MB)	Bw per 10000 devices/ month, no deltas, GB	Bw per 10000 devices/ month, with deltas, GB
alsa-utils *	0.05	6.47	0.24	3.31	0.12
bluez	0.21	3.56	1.28	7.29	2.61
caracalla	0.37	1.02	0.02	3.64	0.08
caracalla-kernel	1.26	129.23	57.23	1,585.79	702.30
core	1.78	82.01	29.31	1,425.65	509.45
locationd	0.16	20.52	6.09	31.47	9.35
modem-manager	0.42	7.91	1.28	32.36	5.23
network-manager	0.26	7.80	2.05	19.94	5.25
tpm2	0.10	2.08	1.04	2.13	1.06
udisks2	0.16	3.19	1.27	4.89	1.95
uefi-fw-tools	0.21	19.14	3.95	as advanced networking, Bluetooth support, and cellular connectivity. 39.14	8.07
wifi-ap	0.26	28.91	10.67	73.91	27.27
wpa-supplicant	0.16	3.14	2.97	4.82	4.55
			Total	**3,234.36**	**1,277.30**
			Reduction %	**60.51%**	

Snap delta usage reduces bandwidth consumption by a total of 60.51% for 3000-series devices, as detailed above.

Fingbox

Fingbox is a network monitoring device that sits in a network and can monitor and manage it by communicating with an app on the user's mobile phone. Fingbox is powered by Ubuntu Core and uses snaps for the preinstalled software, as well as to allow the user or manufacturer to augment the device's capabilities by delivering additional software via OTA updates.

Because Fingbox typically sits on a home or corporate network, it is important to both keep it up to date for security reasons, and do so as unobtrusively as possible with regards to network resource consumption.

A Fingbox ships with the snaps shown in the table below. The figures are in MB, for a single Fingbox.

Snap name	Updates/ month avg	Full snap size avg (MB)	Delta size avg (MB)	Bw per month, no deltas, MB	Bw per month, with deltas, MB
carl	0.18	1.12	0.00	0.198	0.0004
core	1.83	69.34	20.55	126.630	37.526
dnsfwd	1.71	0.12	0.01	0.201	0.024
domotz-kevin	1.40	10.23	0.70	14.352	0.988
domotz-platform	0.75	47.89	7.42	35.979	5.576
domotz-platform-kernel	0.05	54.64	54.64	0.000	0.000
fingbox-agent	3.48	4.87	2.05	16.921	7.127
			Total	**194.281**	**51.241**
			Reduction %	**73.63%**	

Notes:

- Domotz-platform-kernel only has one version which is shipped on devices; thus, its bandwidth requirements are zero since it will not download any updates.

Using snap deltas, the bandwidth usage of a Fingbox sitting on a typical home network and using snap deltas can be reduced by 73.63%. The current generation of Fingbox does not have deltas enabled, but next generation Fingbox devices will ship with deltas and existing devices in the field can be enabled transparently via an OTA update at their discretion.

Rigado Cascade

Rigado's Cascade 'Edge-as-a-Service' offering includes both hardware (the Cascade 500 IoT Gateway) and a platform (Edge Connect, Edge Direct and Edge Protect) which together provide IoT teams with secure edge computing, a containerised application platform and a wide variety of wireless connectivity options – all in a simple monthly subscription.

With Cascade, IoT developers are able to get started faster and focus on their unique applications, rather than wasting time on the 'heavy lifting' required to create and support their own edge infrastructure.

Using the Edge platform tools, IoT developers can publish their software (packaged as snaps) to devices integrating Rigado's IoT technology. The developer simply uploads the snap to the Edge platform, from which it is distributed to all the developers' devices in the field, automatically.

A typical Rigado Cascade device ships with a set of snaps, which provide basic OS and snap functionality, as well as software to connect to the Edge platform for updates, reporting and security.

Additional snaps provided by the device's manufacturer can also be installed in the Cascade device. Assuming a typical update cadence for Rigado's base set of snaps and a hypothetical device-specific snap with the characteristics noted below, the monthly bandwidth consumption for Cascade devices can be seen in the following table. Because Rigado's platform can be used to enable many categories of devices, it is conceivable for an organisation to have a significant fleet of Cascade-powered devices. An organisation with a 10,000-device fleet might expect to see the aggregated bandwidth savings shown on the next page:

Snap name	Updates/ month avg	Full snap size avg (MB)	Delta size avg (MB)	Bw per 10000 devices/ month, no deltas, GB	Bw per 10000 devices/ month, with deltas, GB
bluez	0.21	3.36	1.40	6.87	2.85
cascade	1.60	0.52	0.22	8.14	3.36
cascade-configuration	0.89	0.08	0.03	0.68	0.63
cascade-kernel	1.74	35.55	17.34	605.49	295.28
core	1.83	69.33	20.55	1,240.68	367.74
modem-manager	0.42	5.77	1.23	23.61	5.05
network-manager	0.26	5.90	1.89	15.07	4.83
pivot-agent	0.43	2.45	2.45	10.32	10.32
rigado-deviceops	4.33	3.25	2.10	137.58	88.78
rigado-edge-connect	1.48	3.58	2.13	51.88	30.80
rigado-fw-loader	0.11	0.004	0.00	0.000	0.000
rigado-network-config	1.14	8.45	4.23	94.09	47.13
wifi-ap	0.26	25.86	9.48	66.12	24.25
fictional-snap	1.12	175.48	51.56	1,921.05	564.46
			Total	**4,181.58**	**1,445.47**
			Reduction %	**65.43%**	

Using snap deltas, the bandwidth usage is reduced by 65.43%.

Notes:

- The hypothetical fictional-snap is based on the size and delta performance of nextcloud, a popular personal cloud file storage solution, but with an update cadence matching the average of other snaps on a Cascade device.

- Rigado-fw-loader only has one version which is shipped on devices; thus, its bandwidth requirements are zero since it will not download any updates.

- Pivot-agent is an example of snaps whose delta doesn't provide any savings over the full download (it's actually larger, so the deltas were discarded). In this case, the Snap Store delivers a full download, which explains why the bandwidth column with and without deltas contains the same value for this snap.

Snap deltas in desktop environments

Of course, the bandwidth-saving benefits of snap deltas aren't limited to IoT devices. Any Linux system using snaps will see reductions in bandwidth thanks to this feature. For example, Ubuntu 18.10 ships with a few preinstalled snaps and the user can also choose from a large number of desktop applications which can be installed easily and updated automatically.

Soon after an Ubuntu 18.10 system is started, the system will try to update the snaps it was shipped with. A full download of these updates would consume 293 MB of bandwidth. However, thanks to the use of deltas, only 69.5 MB needs to be downloaded to update to the latest version of these snaps.

This can add up significantly in the case of organisations with large numbers of Ubuntu-based systems, and crucially also benefits individual users and enthusiasts who might choose Ubuntu as their desktop operating system. Some of these users access the internet through bandwidth-constrained connections and the use of snap deltas make Ubuntu a much better network citizen.

Further, Ubuntu Desktop systems are likely to have traditionally-large desktop applications installed as snaps. For illustrative purposes, the following table shows a list of featured snaps along with the ones shipped with the system.

Snap name	Updates/ month avg	Full snap size avg (MB)	Delta size avg (MB)	Bw per month, no deltas, MB	Bw per month, with deltas, MB
0ad	0.58	897.46	185.32	0.510	0.105
brave	2.11	179.81	93.01	0.370	0.191
bw (bitwarden)	1.01	14.75	4.73	0.015	0.005
core *	1.81	82.08	31.28	0.145	0.055
darktable	0.37	92.96	32.28	0.034	0.012
firefox	2.89	182.48	85.09	0.516	0.240
gimp	1.21	155.18	32.32	0.183	0.038
gitkraken	2.91	161.75	35.08	0.460	0.100
gnome-3-26-1604*	1.32	133.33	25.19	0.172	0.033
gnome-calculator*	1.57	2.61	1.06	0.004	0.002
gnome-characters *	1.72	15.92	0.94	0.027	0.002
gnome-logs *	1.50	18.18	4.95	0.027	0.007
Ggnome-system-monitor*	1.66	5.52	0.92	0.009	0.001

[1] *The additional snaps were taken from the Snap Store's list of snaps featured at the time of this writing.*

gtk-common-themes *	1.60	36.24	7.18	0.057	0.011
jgalaxian	0.57	87.93	21.19	0.049	0.012
notepad-plus-plus	2.95	34.70	1.98	0.100	0.006
ruby	4.29	32.24	14.56	0.135	0.061
sftpclient	8.85	207.05	64.90	1.789	0.561
skype	5.81	139.70	42.26	0.793	0.240
slack	0.54	129.42	56.22	0.068	0.030
taskbook	0.68	23.81	12.36	0.016	0.008
tmnationsforever	0.97	125.68	8.31	0.119	0.008
vlc	0.57	200.06	6.18	0.111	0.003
vscode	1.92	110.50	27.95	0.207	0.052
wifi-ap	0.26	25.86	9.48	66.12	24.25
fictional-snap	1.12	175.48	51.56	1,921.05	564.46
			Total	**5.394**	**1.682**
			Reduction %	**68.82%**	

The snaps that are preinstalled on Ubuntu 18.10 are marked with an asterisk. The table shows data for a single system.

A system with all these snaps installed, under the shown update cadence, would save 68.82% of monthly bandwidth using snap deltas for updates.

Snap deltas in cloud environments

So far, the focus has been on the use of snap deltas in IoT and desktop environments. For cloud deployments of snap supported operating systems such as Ubuntu, bandwidth is usually much higher, making bandwidth-saving mechanisms such as snap deltas a less critical concern. Public clouds such as Amazon Web Services provide a cloud-local cache of snap packages, so any updates are served within the cloud environment itself, and Canonical's Enterprise Proxy solution can be installed on a private cloud to provide similar local caching benefits. However, snap delta technology is transparent, and even in environments where its benefits are less relevant, the bandwidth savings are still automatic.

Conclusion

Illustrated by real life use cases from Dell, Fing and Rigado, the use of snap deltas can deliver significant savings in bandwidth when deployed across thousands of IoT devices. When these devices reach the millions, the impact on an organisation becomes even more substantial. Selecting an embedded operating system compatible with snaps, such as Ubuntu Core, therefore becomes an important choice for any device manufacturer when looking at the wider picture of ongoing maintenance, update capability and costs of their fleet. Regardless of their infrastructure elsewhere, deploying software in a manner which reduces data to the edge will provide long term benefits both in terms of update speed and total cost of ownership.

Further reading:

What is snapcraft?

Snap updates are getting smaller, here's why

Build.snapcraft.io gets your code ready to distribute in minutes

Next steps:

Explore IoT snaps or build your own - snapcraft.io

Learn more about Ubuntu Core and contact us

Download Ubuntu Core

ubuntu
Delivered by Canonical