What is an “AppImage”? How do I install it?

Ask Question

186

44

I just read Meet Etcher, A Stylish Open-Source USB Image Writer Tool. It talks about downloading an AppImage.

> Yes, Linux; the Linux packages is distributed as an .appimage for 32-bit and 64-bit distributions, and should run across all major Linux distributions without

> .deb (or .rpm) installer.

What are AppImages? How do they differ from snaps?

packaging appimage

edited May 19 '16 at 22:32

muru

1

asked May 19 '16 at 4:11

DK Bose

13.1k 12 39 83

7 I think the `appimage` tag could become useful, so I just created it. An in the case we won't need or want to keep it, there would be always the option to burninate it again anyway. – Byte Commander May 19 '16 at 11:14

5 Answers

235

Basic Information

Regarding installation

I am quoting the appImage project page here:

> AppImages can be

> need for root rights.

Making it executable

You can make the appImage executable as follows:

```
chmod a+x exampleNam
```

Executing it

You can execute an appImage as follows:

```
./exampleName.AppIma
```

Additional Information

About appImage

You can find some general informations about appImage here.

I am quoting the appImage project page here:

> The key idea of the AppImage format is one app = one file. Every AppImage contains an app and all the files the app needs to run. In other words, each AppImage has no dependencies

> operating system(s).

Wikipedia adds

> AppImage (and the predecessors klik and portablelinuxapps) do not install software in the traditional sense (i.e., it do not put files all over the place in the system).
>
> It use one file per application. Each one is self-contained: it includes all libraries the application depends on and that are not part of the base system. In this regard, it is similar to "application virtualization". One can use a AppImage file even if they are not a superuser, or they are using a live CD. AppImage files are often simpler than compiling and installing an application, as no installation actually took place. The AppImage file is a compressed image which is temporarily

> having to extract the program or modify the underlying system.

The `README.md` of the AppImageKit-project offers a lot additional informations like *Use cases*, *the problem space* and *objectives*.

Use Cases

- As a user, I want to go to an upstream download page, download an application from the original author, and run it on my Linux desktop system just like I would do with a Windows or Mac application.
- As a tester, I want to be able to get the latest bleeding-edge version of an application from a continuous build server and test it on my system, without needing to compile and without having to worry that I might mess up my system.
- As an application author or ISV, I want to provide packages for

and OS X, without the need to get it 'into' a distribution and without having to build for gazillions of different distributions.

Objectives

1. Be Simple.

   > AppImage is intended to be a very simple format that is easy to understand, create, and manage.

2. Maintain binary compatibility.

   > AppImage is a format for binary software distribution. Software packaged as AppImage is intended to be as binary-compatible as possible with as many systems as possible. The need for (re-)compilation of software should be greatly reduced.

> An AppImage should run on all base operating systems (distributions) that it was created for (and later versions). For example, you could target Ubuntu 9.10, openSUSE 11.2, and Fedora 13 (and later versions) at the same time, without having to create and maintain separate packages for each target system.

4. Remove the need for installation.

> AppImages contain the app in a format that allows it to run directly from the archive, without having to be installed first. This is comparable to a Live CD. Before Live CDs, operating systems had

> they could be used.

5. Keep apps compressed all the time.

> Since the application remains packaged all the time, it is never uncompressed on the hard disk. The computer uncompresses the application on-the-fly while accessing it. Since decompression is faster than reading from hard disk on most systems, this has a speed advantage in addition to saving space. Also, the time needed for installation is entirely removed.

6. Allow to put apps anywhere.

> AppImages are "relocatable", thus allowing the user to

> (including CD-ROMs, DVDs, removable disks, USB sticks).

7. Make applications read-only.

> Since AppImages are read-only by design, the user can be reasonably sure that an app does not modify itself during operation.

8. Do not require recompilation.

> AppImages must be possible to create from already-existing binaries, without the need for recompilation. This greatly speeds up the AppImage creation process, since no compiler has to be involved. This also allows third parties to

> AppImages. (Nevertheless, it can be beneficial for upstream application developers to build from source specifically for the purpose of generating an AppImage.)

9. Keep base operating system untouched.

> Since AppImages are intended to run on plain systems that have not been specially prepared by an administrator, AppImages may not require any unusual preparation of the base operating system. Hence, they cannot rely on special kernel patches, kernel modules, or any applications that do not come with the targeted distributions

10. Do not require root.

> Since AppImages are intended to be run by end users, they should not reqiure an administrative account (root) to be installed or used. They may, however, be installed by an administrator (e.g., in multi-user scenarios) if so desired.

edited May 19 '16 at 22:39

muru
1

answered May 19 '16 at 6:07

dufte
7,912 5 27 39

1 Maybe the actual base Q/A should be above the explanation? – UniversallyUniqueID May 19 '16 at 14:13

4 How does an appimage persist settings between launches? – Dan Dascalescu Jan 14 '17 at 3:04

1 Could you elaborate a bit on "read-only" feature? Can I save changes inside the AppImage? Or it

1 @DanDascalescu: Your question deserves to be elevated to a full question on AskUbuntu, not just a comment. Will you ask it? – Kurt Pfeifle Feb 24 '18 at 15:44

1 @KurtPfeifle: since you didn't want to just go ahead and ask it yourself... :) – Dan Dascalescu Feb 26 '18 at 9:01

21

The basic idea might look similar between the two systems, but there are some design differences between snaps and Appimages.

Some "big" ones that come to my mind are:

1. Security, in terms of confinement. Snap packages run in a sandbox and they are not allowed to escape from it and reach other parts of the system that they should **not** touch. This is a stronger security layer that runs parallel to the permissions system.

it at the beginning (and also later on), but if you think about it in terms of system administration, this is the thing that an administrator wants for their users.

2. Security. Installing software taken from around the net, is as safe as going around licking poles in the streets. Sometimes nothing happens, sometimes you get some very big health issues. Snap packages have their proper repositories, that are controlled by Canonical, like the usual standard Ubuntu repositories. You can go on and install `.deb` files from around, but that will be your choice, and not a design issue.
3. Installation. AppImages are meant to be the equivalent of the "portable Windows executables". All the libraries are self-contained and any user can just download and

packages are proper packages, and they need to be installed (as `root` , or with `sudo`) via the apposite package manager (`snap install tic-tac-toe` throws an error: it needs `sudo` !)

4. Removal. To remove a snap package, you need to use the package manager `snap remove ...` with the right permissions to do so. Appimages, on the other hand, they are just "there". So any user does not want that Appimage? He/she just removes the file and it is gone.

While I strongly suggest to be cautious when using Appimages, I personally use some of them myself.

I find them particularly useful on my work system, where I do not have `root` access (only the admin has that) but I need the latest version of a particular software that, fortunately, the developer has

I am a bit afraid that some malign code is indeed contained in them, so I checked as much as possible the identity of the publisher. I am not 100% sure that this software is benign, but I have done all I could.

edited May 19 '16 at 22:42

muru
1

answered May 19 '16 at 7:59

dadexix86
5,418 25 95

2 Basically, everyone's trying to implement OSX app packaging on linux, but nothing quite gets there. – OrangeDog May 19 '16 at 10:25

1 I have no idea about how that works :) For sure snap and Appimages do not work well with software with many dependencies shared between many other softwares. They are perfectly fine with already self-contained software or with few dependencies, but the risk is to have many copies of the same library that do the same task, like in Windows. This should be solved in snap packages (a software that dependes on snap

library version might use it, like with the usual system). – dadexix86 May 19 '16 at 10:29

OSX (and iOS) apps work like AppImage (just download single file and run) but with the sandboxed security of snap. – OrangeDog May 19 '16 at 10:30

OSX is trying to get rid of these and only allow installs from their store. – LtWorf Sep 26 '16 at 10:54

AppImage should indeed contain option for Sand Boxing. Hopefully we'll see that in the future. – Royi May 28 '18 at 20:14

16

While snap focuses on Ubuntu only, AppImage is cross-distribution and also runs on Fedora, debian, openSUSE, CentOS etc.

AppImage needs no runtime or infrastructure support from the Linux distribution and therefore runs next to everywhere. It enables application authors to ship their software directly to Linux users like the do for Windows and OS X; without Canonical or anyone else "in between" the

If an application is provided in AppImage format, then an user can go to the original author's website to download it, e.g., MuseScore from https://musescore.org/en/download. Make the AppImage executable (either using your file manager or `chmod a+x ./yourAppImage`), then you can run the application simply by double-clicking.

edited May 19 '16 at 12:18

answered May 19 '16 at 6:24

probono
526 3 6

So if I want to install Leafpad, do I click on "Set me up"? What will happen then? And how can I uninstall an AppImage? I'm looking at bintray.com/probono/AppImages but didn't find the answer. If you don't mind, you could edit your answer to include the information as well as anything else you think users may like to know. – DK Bose May 19 '16 at 6:59

2 Not sure if that would be a solution for you aswell - but leafpad is available

'only' 0.8.17 - so even older. – dufte May 19 '16 at 7:30

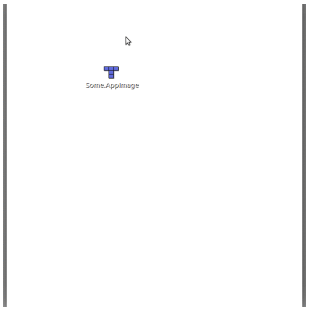
1 True, but for someone just wanting to test the waters, it's, like the site says, a "Hello, World" type of thing. – DK Bose May 19 '16 at 11:57

Excellent. After this you can simply double click. Thats what I needed! – Dawoodjee Jun 26 '18 at 19:29

13

Before you can run an AppImage, you need to make it executable. This is a Linux security feature. There are two main ways to make an AppImage executable:

1. Using the GUI



or

2. On the command line

```
chmod a+x Some.Appima
```

Now double-click your file to run it OR Right-click > Run

2,569 15 21

answered Oct 10 '16 at 2:01

probono

526 3 6

5

AppImages require FUSE to run. Filesystem in Userspace (FUSE) is a system that lets non-root users mount filesystems.

Install FUSE

Many distributions have a working FUSE setup out-of-the-box. However if it is not working for you, you may need to install and configure FUSE manually.

For example, on Ubuntu:

```
sudo apt-get install
sudo modprobe fuse
sudo usermod -a -G fu
```

For example, on openSUSE:

```
sudo zypper install f
```

Fallback

If you don't want to install FUSE, you can either mount or extract the AppImage.

To mount the AppImage and run the application, simply run

If that does not work, you might have an experimental type 2 AppImage. These require you to pass `-o offset=...` to the `mount` command. Run the AppImage with `--appimage-offset` to find out the correct number for the offset.

edited Apr 21 '18 at 5:35

George Udosen
20.1k 9 43 67

answered Oct 13 '16 at 15:06

insign
166 1 6

After writing the 3rd line (`sudo usermod -a -G fuse`) followed by my user name, I keep getting the `usermod: group 'fuse' does not exist` ... Any idea on why? – Martec Nov 10 '18 at 19:09