Desktop

Cloud and Server

Web and Design

We reduced our Docker images by 60% with -no-install-recommends

Internet of Things



Francisco Jiménez Cabrera on 15 November 2019

Share on: 😝 🗶 in

People and culture

This article is more than 4 years old.

Tags: Design , Development , docker , web

Here at Canonical, we use Dockerfiles on a daily basis for all our web projects. Something that caught our attention recently was the amount of space that we were using for each Docker image, and we realized that we were installing more dependencies than we needed.

In this article, I'll explain how we improved our image build time and reduced the image size by using the flag --no-install-recommends in our Dockerfiles.

Using APT

As you may know, Advanced Package Tool, or APT, is the interface to handle the installation and removal of software on Debian based distributions. It simplifies the process of managing software by automating the retrieval, configuration and installation of software packages.

Every package has different types of dependencies:

- Required packages • Recommended packages
- Suggested packages

The required packages are mandatory since they are necessary for the correct operation of the package. Still, the recommended and suggested packages are not essential, and they are there to offer some extra functionality that we might not need to use. By default APT will install required and recommended packages.

Disabling recommended packages

To avoid the installation of recommended packages, we included the flag --noinstall-recommends when using APT in our Dockerfile.

RUN apt-get update && apt-get install --no-install-recommends --yes python3

By doing this, we achieve a decrease of around 60% in our Docker images size.

jp.ubuntu.com	feature	b5896a1364ee	2 weeks ago	185MB
jp.ubuntu.com	master	f57b679d1394	2 weeks ago	550MB
cn.ubuntu.com	feature	7f00d3306fcc	2 weeks ago	270MB
cn.ubuntu.com	master	4845c1aa4663	2 weeks ago	638MB

This obviously will vary according to the dependencies you are using. In our case, we did it for all our Python websites which reduced the size of all our Docker containers significantly. Also, the build time sped up about 15%.

```
Successfully built 60908df3b515
 uccessfully tagged cn.ubuntu.com:previous
 ocker build --build-arg COMMIT_ID=test -t cn.ubuntu.com:previous . 0.21s user 0.13s system 1% cpu <mark>26.713</mark> total
Successfully built a36d6965c88f
 uccessfully tagged cn.ubuntu.com:new
      build --build-arg COMMIT_ID=test -t cn.ubuntu.com:new . 0.26s user 0.06s system 1% cpu 23.795 total
```

I recommend doing this whenever you run apt install in your Dockerfiles, and I hope you find it useful if you are trying to reduce the size of your containers.

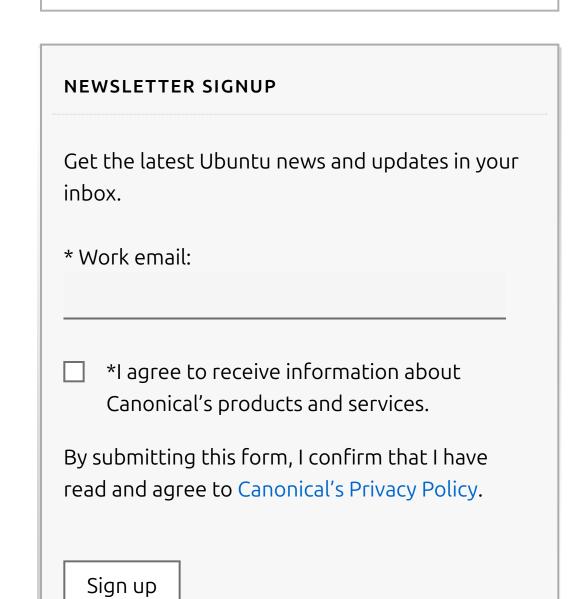
It is important to keep in mind that doing this could result in some missing libraries in your projects which you may have to add back explicitly, but this will ultimately give you more control in the dependencies in your project.

Happy dockering!

What's the risk of unsolved vulnerabilities in Docker images?

Recent surveys found that many popular containers had known vulnerabilities. Container images provenance is critical for a secure software supply chain in production. Benefit from Canonical's security expertise with the LTS Docker images portfolio, a curated set of application images, free of vulnerabilities, with a 24/7 commitment.

Integrate with hardened LTS images >



Related posts

© 2024 Canonical Ltd.

Introducing a VSCode extension for Vanilla CSS Framework

The Vanilla CSS Framework is a utility class-based and customizable SASS library that is the go-to when it comes to styling websites and dashboards across the... Visual Testing: GitHub Actions Migration & Test Optimisation

What is Visual Testing? Visual testing analyses the visual appearance of a user interface. Snapshots of pages are taken to create a "baseline", or the current... Let's talk open design

Why aren't there more design contributions in open source? Help us find out!

OpenStack	What is OpenStack Features Managed Consulting Install Support		
Ceph	What is Ceph Managed Consulting Docs Install		
Kubernetes	What is Kubernetes Charmed Kubernetes Managed Install Docs Resources		
Managed Services	OpenStack Kubernetes Ceph Apps Observability Firefighting		
AI/ML	MLOps Kubeflow MLflow Consulting Data Science		
Robotics	What is ROS ROS ESM Community Docs		
IoT	App store Embedded Linux Management		
Ubuntu Core	Features Success stories Services Docs		
Ubuntu Desktop	Organisations Developers Flavours WSL		
Ubuntu Server	Hyperscale Docs		
Cloud	What is cloud computing What is private cloud What is hybrid cloud What is multi-cloud Public cloud		
Security	ESM Livepatch Certifications & Hardening CVEs Notices Docker Images		
Landscape	Features Managed Pricing Install Docs Log in to Landscape		
Containers	What are containers Chiselled Ubuntu Chiselled and .NET		
Downloads	Desktop Server Core Cloud		
Support	Your subscriptions Account users Pricing Discourse		
Observability	What is observability Managed		
Pricing	Consulting Desktops Devices		
Solutions	Al Data Infrastructure Secure open source		
Sectors	Automotive Industrial Government Telco Finance		
Contact us	About us Community Careers Blog Resources Press centre		

Ubuntu and Canonical are registered trademarks of Canonical Ltd.

Legal information Data privacy Manage your tracker settings Report a bug on this site