

Tarea 10: Instalación de Docker en **Linux**



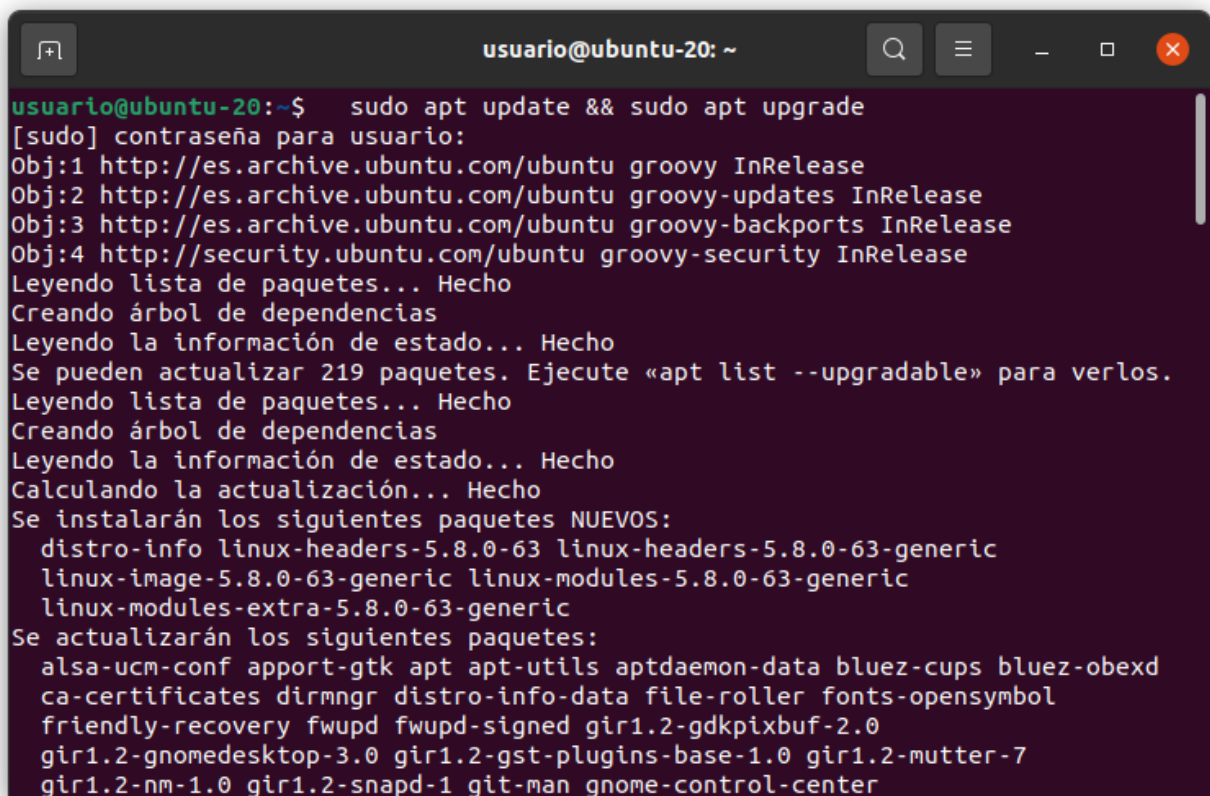
docker

Índice

1. Actualización de repositorios
2. Instalación de Docker en Linux
3. Trabajo con imágenes de Docker
4. Administración de contenedores de Docker

1. Actualización de repositorios

Primero vamos a actualizar el repositorio y el sistema operativo para que la instalación de apache sea mas segura.



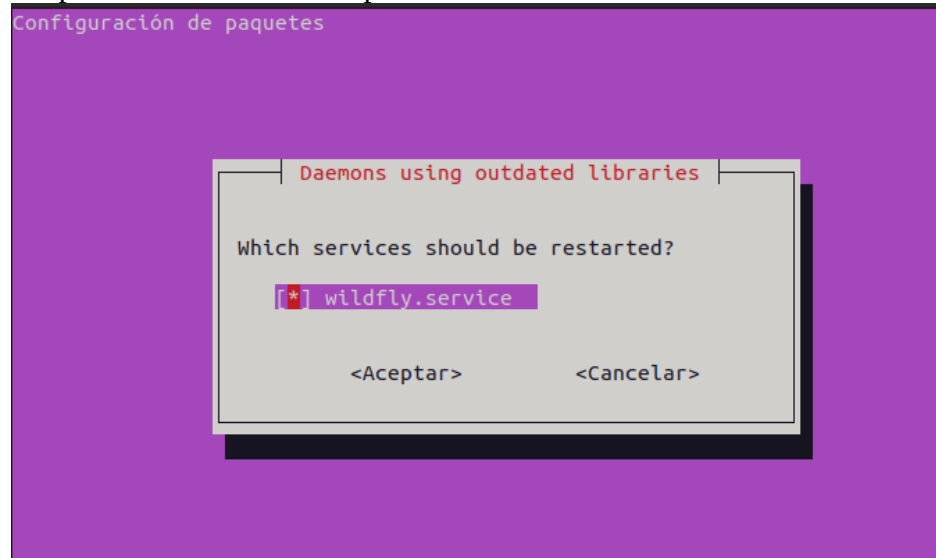
```
usuario@ubuntu-20: ~  
usuario@ubuntu-20:~$ sudo apt update && sudo apt upgrade  
[sudo] contraseña para usuario:  
Obj:1 http://es.archive.ubuntu.com/ubuntu groovy InRelease  
Obj:2 http://es.archive.ubuntu.com/ubuntu groovy-updates InRelease  
Obj:3 http://es.archive.ubuntu.com/ubuntu groovy-backports InRelease  
Obj:4 http://security.ubuntu.com/ubuntu groovy-security InRelease  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se pueden actualizar 219 paquetes. Ejecute «apt list --upgradable» para verlos.  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Calculando la actualización... Hecho  
Se instalarán los siguientes paquetes NUEVOS:  
  distro-info linux-headers-5.8.0-63 linux-headers-5.8.0-63-generic  
  linux-image-5.8.0-63-generic linux-modules-5.8.0-63-generic  
  linux-modules-extra-5.8.0-63-generic  
Se actualizarán los siguientes paquetes:  
  alsa-ucm-conf apport-gtk apt apt-utils aptdaemon-data bluez-cups bluez-obexd  
  ca-certificates dirmngr distro-info-data file-roller fonts-opensymbol  
  friendly-recovery fwupd fwupd-signed gir1.2-gdkpixbuf-2.0  
  gir1.2-gnomedesktop-3.0 gir1.2-gst-plugins-base-1.0 gir1.2-mutter-7  
  gir1.2-nm-1.0 gir1.2-snapd-1 git-man gnome-control-center
```

2. Instalación de Docker en Linux

Ahora vamos a instalar algunos paquetes previos a la instalación de Docker

```
-----
usuario@ubuntu-20:~$ sudo apt install apt-transport-https ca-certificates curl
software-properties-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
ca-certificates ya está en su versión más reciente (20210119~20.10.1).
curl ya está en su versión más reciente (7.68.0-1ubuntu4.3).
software-properties-common ya está en su versión más reciente (0.99.3.1).
fijado software-properties-common como instalado manualmente.
apt-transport-https ya está en su versión más reciente (2.1.10ubuntu0.3).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Aceptamos las condiciones presentadas



Luego, añadimos la clave de GPG para el repositorio oficial de Docker en el sistema:

```
usuario@ubuntu-20:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | s
udo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (s
ee apt-key(8)).
OK
```

Agregamos el repositorio de Docker a las fuentes de APT:

```
usuario@ubuntu-20:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
Repositorio: «deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable»
Descripción:
Archive for codename: focal components: stable
Más información: https://download.docker.com/linux/ubuntu
Añadiendo repositorio.
Oprima [INTRO] para continuar o Ctrl+c para cancelar.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-groovy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-groovy.list
```

A continuación, actualizamos el paquete de base de datos con los paquetes de Docker del repositorio recién agregado:

```
usuario@ubuntu-20:~$ sudo apt update
Obj:1 http://ppa.launchpad.net/linuxuprising/java/ubuntu groovy InRelease
Obj:2 http://packages.microsoft.com/repos/code stable InRelease
Obj:3 http://security.ubuntu.com/ubuntu groovy-security InRelease
Obj:5 http://es.archive.ubuntu.com/ubuntu groovy InRelease
Obj:6 https://download.docker.com/linux/ubuntu focal InRelease
Obj:7 http://es.archive.ubuntu.com/ubuntu groovy-updates InRelease
Obj:8 http://es.archive.ubuntu.com/ubuntu groovy-backports InRelease
Obj:4 https://packages.gitlab.com/gitlab/gitlab-ce/ubuntu groovy InRelease
Obj:9 https://packages.gitlab.com/gitlab/gitlab-ee/ubuntu groovy InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
```

Nos aseguramos de estar a punto de realizar la instalación desde el repositorio de Docker en lugar del repositorio predeterminado de Ubuntu:

```
usuario@ubuntu-20:~$ apt-cache policy docker-ce
docker-ce:
  Instalados: (ninguno)
  Candidato: 5:20.10.10~3-0~ubuntu-focal
  Tabla de versión:
    5:20.10.10~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.9~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.8~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.7~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.6~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.5~3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.4~3-0~ubuntu-focal 500
```

Observamos que docker-ce no está instalado, pero la opción más viable para la instalación es del repositorio de Docker para Linux (focal).

Por último, instalamos Docker:

```
usuario@ubuntu-20:~$ sudo apt install docker-ce
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin
  libslirp0 pigz slirp4netns
Paquetes sugeridos:
  aufs-tools cgroupfs-mount | cgroup-lite
Se instalarán los siguientes paquetes NUEVOS:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-scan-plugin libslirp0 pigz slirp4netns
```

Con esto, Docker quedará instalado, el demonio se iniciará y el proceso se habilitará para ejecutarse en el inicio. Comprobamos que funciona:

```
usuario@ubuntu-20:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Tue 2021-11-02 18:18:40 CET; 50s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 6406 (dockerd)
       Tasks: 8
      Memory: 29.5M
      CGroup: /system.slice/docker.service
              └─6406 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont
```

3. Trabajo con imágenes con Docker

Para verificar si podemos acceder a imágenes y descargarlas de Docker Hub escribimos el comando siguiente

```
usuario@ubuntu-20:~$ sudo docker run hello-world
```

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

4. Administración de contenedores de Docker

Después de usar Docker durante un tiempo, tendrá muchos contenedores activos (en ejecución) e inactivos en su computadora. Para ver los activos, utilizamos lo siguiente:

```
usuario@ubuntu-20:~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

A través de este tutorial, inicié dos contenedores: uno de la imagen hello-world y otro de la imagen ubuntu. Ambos contenedores ya no están en ejecución, pero aún existen en el sistema.

Para ver todos los contenedores, activos e inactivos, ejecutamos docker ps con el conmutador -a:

```
usuario@ubuntu-20:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
74afd31b60c3	hello-world	"/hello"	10 minutes ago	Exited (0) 10 minutes ago
	exciting_heisenberg			

Para ver el último contenedor que creamos, lo pasamos al conmutador -l:

```
usuario@ubuntu-20:~$ sudo docker ps -l
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
74afd31b60c3	hello-world	"/hello"	12 minutes ago	Exited (0) 11 minutes ago
	exciting_heisenberg			

Listar las imágenes de Docker de nuevo mostrará la nueva imagen, así como la anterior de la que se derivó:

```
usuario@ubuntu-20:~$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	feb5d9fea6a5	5 weeks ago	13.3kB