## 1. Instalación

**En Ubuntu 16.04**

Comprobarversión ubuntu:

lsb\_release–a

Eliminar instalaciones anteriores de Docker

sudo apt-get remove docker docker-engine docker.io

Actualización de la base de datos del sistema:

sudo apt-getupdate

Agregamos la clave GPG para el repositorio oficial de Docker al sistema:

sudo apt-key adv --keyserver hkp://p80.pool.sks-eyservers.net:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D

Agregar el repositorio Docker a fuentes APT:

sudo apt-add-repository 'deb https://apt.dockerproject.org/repo ubuntu-xenial main'

Actualizamos la base de datos de paquetes, con los paquetes Docker desde el repositorio recién agregado:

sudo apt-getupdate

Aseguramos instalar desde el repositorio de Docker en lugar del repositorio predeterminado de Ubuntu 16.04:

apt-cache policy docker-engine

Instalación de Docker:

sudo apt-get install -y docker-engine

Docker está instalado, el daemon iniciado, y el proceso habilitado para iniciar en el arranque. Comprobación de que se está ejecutando:

sudo systemctl status docker

**En Ubuntu 18.04**

Eliminar instalaciones anteriores de Docker:

sudo apt-get remove docker docker-engine docker.io

Actualización de la base de datos del sistema:

sudo apt-getupdate

Actualización de paquetes:

sudo apt-getupgrade

Instalar dependencias Docker:

apt-get install -yapt-transport-httpsca-certificatescurl gnupg-agent \

software-properties-common

Añadir clave GPG oficial de Docker:

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-keyadd–

Añadir el repositorio de Docker:

sudo add-apt-repository"deb [arch=amd64] https://download.docker.com/linux/ubuntu \ bionicstable"

Actualización de repositorios e instalación de Docker:

sudo apt-getupdate

sudo apt-getinstall –y docker-ce docker-ce-cli containerd.io docker-compose

Comprobar versión:

docker - -version

Ejecutar docker sin sudo:

Se añadeel nombre de usuario al grupo docker:

sudousermod –aGdocker$(whoami) obien*:*

sudousermod -aG docker $USERNAMEo *$*USER

Después de esto debemos cerrar y abrir sesión.

## 2. Gestión de imágenes y contenedores

Existe un repositorio oficial de Docker llamado Docker Hub donde cualquiera puede subir o descargar imágenes. Ofrece una interfaz web que nos permite buscar fácilmente imágenes. También se puede usar en modo comando como se ve a continuación.

Para comprobar si puede acceder y descargar imágenes de Docker Hub:

sudo docker run hello-world

(debe aparecer ”Hello from Docker”)

### Búsqueda deimágenes

Por ejemplo, si queremos buscar una imagen de Ubuntu:

docker searchubuntu

En la columna “**OFICIAL”** el mensaje“[**OK]”** indica una imagen creada y apoyada por la empresa.

### Descarga de imágenes en nuestro equipo

Por ejemplo, para descargar una imagen de ubuntu:

dockerpullubuntu

### Ver imágenes disponibles en nuestro equipo

dockerimages

Aparecerán las imágenes descargadas y las creadas ej.ubuntu y hello-world

**Ejecutar un nuevo contenedor docker**

docker run –itubuntu

las opciones –i y –t hacen que Docker asigne un pseudo-TTY conectado a la entrada estándar del contenedor, creando una shell interactiva en el contenedor. A partir de ahora mandaremos comandos al contenedor desde “dentro” del mismo. Por ejemplo, podemos listar los directorios con el comando ls. Podemos también instalar algunos paquetes:

apt-get update

apt-get installpaqueteElegir uno o varios:nmap,vim…

exit

**Crear una imagen**

Creamos una nueva imagen a partir del contenedor en el que hemos instalado nuevos paquetes. Para ello buscamos el ID del contenedor a partir del cual queremos crear la imagen:

docker ps –a

Y creamos la imagen:

dockercommit -m "inf-add" -a "nombre\_autor" id-contenedornomb\_nueva\_imagen

**Eliminar imagen**

dockerrmi imagen -f

**Eliminar contenedor**

dockerrm id-contenedor

**Ver contenedores**

|  |  |
| --- | --- |
| Ver ultimo contendor creado | *sudo docker ps -l* |
| Parar contenedor ejecutándose/activo | *sudo docker stop ID-contenedor* |

**Ruta de contenedores**

/var/lib/docker

**Docker Hub**

Logearse en “Docker Hub”

sudo dockerlogin -u docker-registry-nomb-usuario

Subir imágenes a “Docker Hub”

sudo dockerpushdocker-registry-nomb-usuario/docker-image-name

**Subcomando de Docker**

|  |  |
| --- | --- |
| docker [option] [command] [arguments] | |
| attach | Attach to a running container |
| build | Build an image from a Dockerfile |
| commit | Create a new image from a container's changes |
| cp | Copy files/folders between a container and the local filesystem |
| create | Create a new container |
| diff | Inspect changes on a container's filesystem |
| events | Get real time events from the server |
| exec | Run a command in a running container |
| export | Export a container's filesystem as a tar archive |
| history | Show the history of an image |
| images | Listimages |
| import | Import the contents from a tarball to create a filesystem image |
| info | Display system-wideinformation |
| inspect | Return low-level information on a container or image |
| kill | Kill a running container |
| load | Load an image from a tar archive or STDIN |
| login | Log in to a Docker registry |
| logout | Log out from a Docker registry |
| logs | Fetch the logs of a container |
| network | Manage Docker networks |
| pause | Pause all processes within a container |
| port | List port mappings or a specific mapping for the CONTAINER |
| ps | Listcontainers |
| pull | Pull an image or a repository from a registry |
| push | Push an image or a repository to a registry |
| rename | Rename a container |
| restart | Restart a container |
| rm | Remove one or more containers |
| rmi | Remove one or more images |
| run | Run a command in a new container |
| save | Save one or more images to a tar archive |
| search | Search the Docker Hub for images |
| start | Start one or more stopped containers |
| stats | Display a live stream of container(s) resource usage statistics |
| stop | Stop a running container |
| tag | Tag an image into a repository |
| top | Display the running processes of a container |
| unpause | Unpause all processes within a container |
| update | Update configuration of one or more containers |
| version | Show the Docker version information |
| volume | Manage Docker volumes |
| wait | Block until a container stops, then print its exit code |