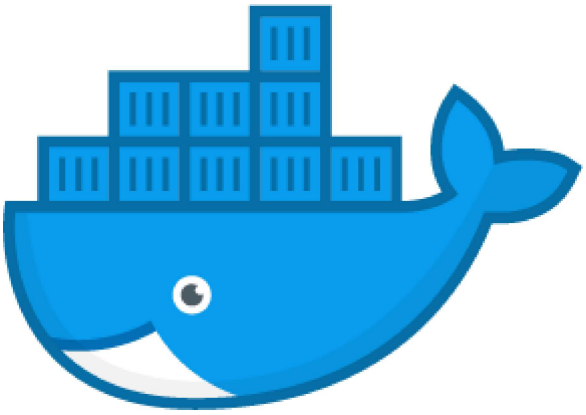


Comprender el concepto de capas de imágenes con Dockerfile

Docker - Principiantes | Intermedio | Avanzado

Ver en GitHub Únete a Slack Hoja de trucos de Docker

Hoja de trucos de composición de Docker

Comprender el concepto de capas de imágenes con Dockerfile

El contenedor Docker es una instancia ejecutable de una imagen, que en realidad se crea escribiendo una capa legible/escribible encima de algunas capas de solo lectura.

La imagen principal utilizada para crear otra imagen a partir de un Dockerfile es de solo lectura. Cuando ejecutamos instrucciones en esta imagen principal, se siguen sumando nuevas capas. Estas capas se crean cuando ejecutamos el comando de compilación docker.

Las instrucciones EJECUTAR, COPIAR, AÑADIR contribuyen principalmente a la adición de capas en una compilación de Docker.

Cada capa es de solo lectura, excepto la última, que se agrega a la imagen para generar un contenedor ejecutable. Esta última capa se denomina “capa contenedora”. Todos los cambios realizados en el contenedor, como la creación de nuevos archivos, la instalación de aplicaciones, etc., se realizan en esta capa delgada.

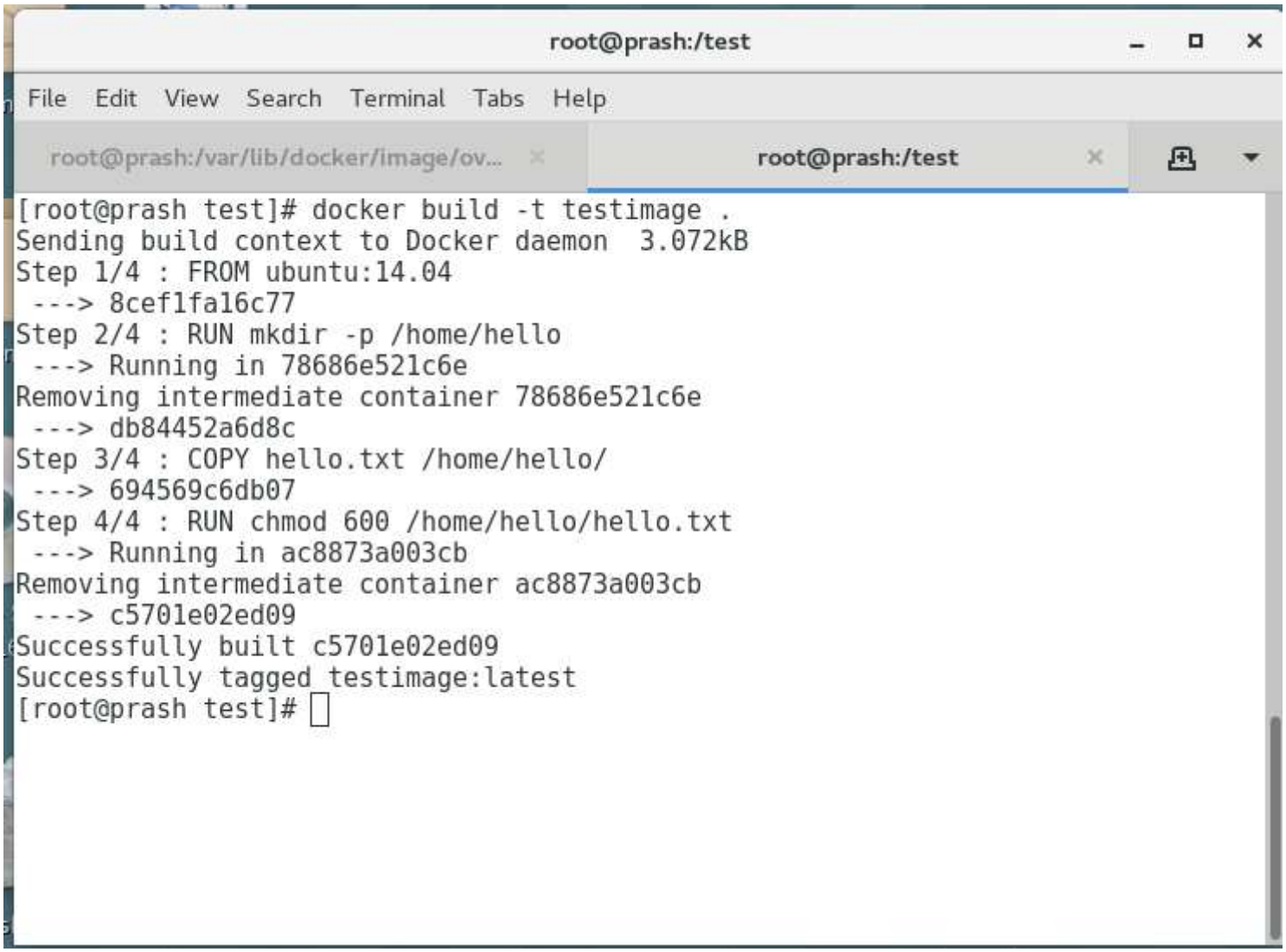
Entendamos esta estratificación usando un ejemplo:

Considere el Dockerfile que se proporciona a continuación:

```
FROM ubuntu:latest
RUN mkdir -p /hello/hello
COPY hello.txt /hello/hello
RUN chmod 600 /hello/hello/hello.txt
```

Id. de capa

Cada instrucción del Dockerfile genera una capa. Cada una de esta capa tiene una identificación única generada aleatoriamente. Este ID se puede ver en el momento de la compilación. Vea la imagen a continuación:



Para ver todas estas capas una vez que se crea una imagen a partir de un Dockerfile, podemos usar el comando de historial de Docker.

```
[root@prash test]# docker history testimage
IMAGE               CREATED             CREATED BY                                      SIZE
MENT
c5701e02ed09        15 minutes ago      /bin/sh -c chmod 600 /home/hello/hello.txt      6B
694569c6db07        15 minutes ago      /bin/sh -c #(nop) COPY file:91d8cf0a303cd4d8…   6B
db84452a6d8c        15 minutes ago      /bin/sh -c mkdir -p /home/hello                 0B
8cef1fa16c77        13 months ago       /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>           13 months ago       /bin/sh -c mkdir -p /run/systemd && echo 'do…   7B
<missing>           13 months ago       /bin/sh -c sed -i 's/^#\s*\(deb.*universe\)$…   2.76kB
<missing>           13 months ago       /bin/sh -c rm -rf /var/lib/apt/lists/*          0B
<missing>           13 months ago       /bin/sh -c set -xe   && echo '#!/bin/sh' > /…   195kB
<missing>           13 months ago       /bin/sh -c #(nop) ADD file:62b6d11e1f009825b…   223MB
[root@prash test]#
```

Para ver más información sobre la imagen de Docker y las capas, use el `docker inspect` comando como tal:

```
# docker inspect testimage:latest

[
    {
        "Id": "sha256:c5701e02ed095ae7cabaef9fcef009d1f272206ff707deca13a680e024db7f02",
        "RepoTags": [
            "testimage:latest"
        ],
        "RepoDigests": [],
        "Parent": "sha256:694569c6db07ecef432cee1a9a4a6d45f2fd1f6be16814bf59e101bed966e612",
        "Comment": "",
        "Created": "2019-06-03T23:47:01.026463541Z",
        "Container": "ac8873a003cb9ed972b4675f8d27181b99112e7530a5803ff89780e3ecc18b1c",
        "ContainerConfig": {
            "Hostname": "",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/sh",
```

```
            "-c",
            "chmod 600 /home/hello/hello.txt"
        ],
        "ArgsEscaped": true,
        "Image": "sha256:694569c6db07ecef432cee1a9a4a6d45f2fd1f6be16814bf59e101bed966e61
        "Volumes": null,
        "WorkingDir": "",
        "Entrypoint": null,
        "OnBuild": null,
        "Labels": null
    },
    "DockerVersion": "18.03.1-ce",
    "Author": "",
    "Config": {
        "Hostname": "",
        "Domainname": "",
        "User": "",
        "AttachStdin": false,
        "AttachStdout": false,
        "AttachStderr": false,
        "Tty": false,
        "OpenStdin": false,
        "StdinOnce": false,
        "Env": [
            "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
        ],
        "Cmd": [
            "/bin/bash"
        ],
        "ArgsEscaped": true,
        "Image": "sha256:694569c6db07ecef432cee1a9a4a6d45f2fd1f6be16814bf59e101bed966e61
        "Volumes": null,
        "WorkingDir": "",
        "Entrypoint": null,
        "OnBuild": null,
        "Labels": null
    },
    "Architecture": "amd64",
    "Os": "linux",
    "Size": 222876395,
    "VirtualSize": 222876395,
    "GraphDriver": {
        "Data": {
            "LowerDir": "/var/lib/docker/overlay2/86a76eac21ae67f6d78e59076107a121e6dfb9
            "MergedDir": "/var/lib/docker/overlay2/31c68adcd824f155d23de4197b3d0b8776b07
            "UpperDir": "/var/lib/docker/overlay2/31c68adcd824f155d23de4197b3d0b8776b079
```

```
                "WorkDir": "/var/lib/docker/overlay2/31c68adcd824f155d23de4197b3d0b8776b079c
            },
            "Name": "overlay2"
        },
        "RootFS": {
            "Type": "layers",
            "Layers": [
                "sha256:05b0f7f2a81723fd647744a7340477ef9619f5ddeba3f2ca039dac3dd143cd59",
                "sha256:0c3819952093832ffd8865bf72bc17f2f5475795cffe97e2b4c4ff36e638c244",
                "sha256:14fa4a9494bf9e61f83a1bb96cd9e963ab0cbbdaf8ed91ff5eec5196c5bf7b12",
                "sha256:b33859b66bfd3ad176ccf3be8dbd52d6b9823de8cc26688f2efeb76a0ea24a78",
                "sha256:4622c8e1bdc0716e185fa3b338fa415dfdad3724336315de0bebd173a6ceaf05",
                "sha256:6427efc3a0d7bae1fe315b844703580b2095073dcdf54a6ed9c7b1c0d982d9b0",
                "sha256:59cd898074ac7765bacd76a11724b8d666ed8e9c14e7806dfb20a486102f6f1e",
                "sha256:ad24f18512fddb8794612f7ec5955d06dcee93641d02932d809f0640263b8e79"
            ]
        },
        "Metadata": {
            "LastTagTime": "2019-06-04T05:17:01.430558997+05:30"
        }
    }
]
```


¿Quieres visualizar capas de Docker Image?

```
docker run --rm -it -v /var/run/docker.sock:/var/run/docker.sock  wagoodman/dive testimage
```

```
[● Layers]──────────────────────────────────────────────────────────────────────── [Cur
Cmp   Size  Command                                                               Perm
     63 MB  FROM e388568efdf7281                                                  drwx
    988 kB  [ -z "$(apt-get indextargets)" ]                                      -rwx
    745 B   set -xe   && echo '#!/bin/sh' > /usr/sbin/policy-rc.d  && echo 'exit 101' > -rwx
      7 B   mkdir -p /run/systemd && echo 'docker' > /run/systemd/container       -rwx
      0 B   mkdir -p /hello/hello                                                 -rwx
     37 B   #(nop) COPY file:666735678ded52c6f9e0693ca27b4dc3d466e3d79c585a58c3b9a91357 -rwx
     37 B   chmod 600 /hello/hello/hello.txt                                     -rwx
                                                                                 -rwx
[Layer Details]─────────────────────────────────────────────────────────────────── -rwx
                                                                                 -rwx
Tags:   (unavailable)                                                            -rwx
```

```
Id:     e388568efdf72814bd6439a80d822ce06b631689a82292a2b96382d020d63a4c
Digest: sha256:43c67172d1d182ca5460fc962f8f053f33028e0a3a1d423e05d91b532429e73d
Command:
#(nop) ADD file:08e718ed0796013f5957a1be7da3bef6225f3d82d8be0a86a7114e5caad50cbc in /

[Image Details]──────────────────────────────────────────────────────────────────

Total Image size: 64 MB
Potential wasted space: 308 B
Image efficiency score: 99 %

Count   Total Space  Path
    2         234 B  /var/lib/dpkg/diversions
    2          74 B  /hello/hello/hello.txt
```

Contribuyente

Prashansa Kulshrestha
Ajeet S Raina

Practice Docker Tutorial

free Ubuntu VM

Tweets by @collabnix

Collabnix
@collabnix

The Rise of Low-Code/No-Code Application Platforms collabnix.com/the-rise-of-lo…

Empotrar Ver en Twitter

dockerlabs is maintained by collabnix.

This page was generated by GitHub Pages.