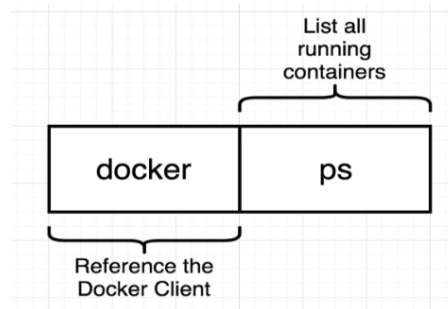


Docker Client – Escuchando Contenedores en Ejecución

Vamos a echar otro vistazo a un comando muy utilizado, el comando *docker ps*:



Este comando mostrará todos los diferentes contenedores en ejecución que se encuentran actualmente en su ordenador:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
jorge@MacBook-Pro-de-Jorge ~ %
```

Cuando ejecutamos este comando, veremos los headers de una tabla. En este momento no tenemos contenedores ejecutándose, por lo que no tendremos ningún registro de salida en esta tabla.

Es importante mencionar que solo hemos creado contenedores en base a imágenes que se ejecutan muy rápidamente y luego se cierran de forma automática. Por ejemplo, cuando ejecutamos el comando *docker run echo hi there*, el contenedor se inicia y luego nos regresa a la línea de comando de la terminal casi inmediatamente:

```
jorge@MacBook-Pro-de-Jorge ~ % docker run busybox echo hi there
hi there
jorge@MacBook-Pro-de-Jorge ~ %
```

Así que, si queremos que el comando *docker ps* nos de información relevante, debemos tener algún contenedor que esté ejecutándose durante algún tiempo. Para conseguir que un contenedor funcione un poco más de tiempo, vamos a cambiar el comando *echo* por *ping*:

```
jorge@MacBook-Pro-de-Jorge ~ % docker run busybox ping google.com
```

Este comando hará ping al servidor de Google y medirá la cantidad de latencia. Este comando seguirá ejecutándose durante bastante tiempo:

```
jorge@MacBook-Pro-de-Jorge ~ % docker run busybox ping google.com
PING google.com (142.251.34.206): 56 data bytes
64 bytes from 142.251.34.206: seq=0 ttl=37 time=21.731 ms
64 bytes from 142.251.34.206: seq=1 ttl=37 time=19.045 ms
64 bytes from 142.251.34.206: seq=2 ttl=37 time=19.260 ms
64 bytes from 142.251.34.206: seq=3 ttl=37 time=16.889 ms
64 bytes from 142.251.34.206: seq=4 ttl=37 time=18.977 ms
64 bytes from 142.251.34.206: seq=5 ttl=37 time=19.812 ms
```

Ahora podemos ejecutar `docker ps` en una segunda ventana de la terminal y deberíamos ver este contenedor que se está ejecutando:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c24ee07d079c   busybox   "ping google.com"       39 seconds ago Up 38 seconds          serene_herschel
jorge@MacBook-Pro-de-Jorge ~ %
```

Podemos ver que se imprimen los datos del contenedor que ejecutamos, y de entre sus datos, podemos ver que tiene el Id del contenedor, el cual podemos usar para muchas otras operaciones:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c24ee07d079c   busybox   "ping google.com"       39 seconds ago Up 38 seconds          serene_herschel
jorge@MacBook-Pro-de-Jorge ~ %
```

Podemos ver también el nombre de la imagen que se utilizó para crear el contenedor:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c24ee07d079c   busybox   "ping google.com"       39 seconds ago Up 38 seconds          serene_herschel
jorge@MacBook-Pro-de-Jorge ~ %
```

Podemos ver hace cuánto tiempo se creó el contenedor:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c24ee07d079c   busybox   "ping google.com"       39 seconds ago Up 38 seconds          serene_herschel
jorge@MacBook-Pro-de-Jorge ~ %
```

Obtenemos el estatus, que nos indica que el contenedor ha estado activo por 24 segundos:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c24ee07d079c   busybox   "ping google.com"       39 seconds ago Up 38 seconds          serene_herschel
jorge@MacBook-Pro-de-Jorge ~ %
```

Tenemos una lista de puertos usados, pero en el caso de nuestro contenedor `busybox` está vacío, ya que la imagen usada no expone ningún puerto:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c24ee07d079c   busybox   "ping google.com"       39 seconds ago Up 38 seconds          serene_herschel
jorge@MacBook-Pro-de-Jorge ~ %
```

Y al final podremos ver un nombre generado de forma aleatoria para identificar fácilmente este contenedor:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
c24ee07d079c   busybox   "ping google.com"       39 seconds ago Up 38 seconds          serene_herschel
jorge@MacBook-Pro-de-Jorge ~ %
```

Podemos detener el contenedor que ejecuta el proceso de ping.

Como es de esperar, si ejecutamos de nuevo el comando *docker ps*, ya no veremos ese contenedor ejecutándose:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
jorge@MacBook-Pro-de-Jorge ~ %
```

Así que el comando *docker ps* muestra específicamente los contenedores en ejecución.

Podemos modificar un poco el comando *docker ps* para que muestre todos los contenedores que se han creado. Para ello, ejecutamos el comando *docker ps --all*:

```
jorge@MacBook-Pro-de-Jorge ~ % docker ps --all
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
4f0c849576e3   busybox   "sh"                    About a minute ago   Exited (0) About a minute ago   brave_feynman
83024b03649c   hello-world "/hello"              About a minute ago   Exited (0) About a minute ago   inspiring_kilby
c24ee07d079c   busybox   "ping google.com"      24 hours ago         Exited (0) 24 hours ago         serene_herschel
4adef375dd18   busybox   "ping google.com"      25 hours ago         Exited (0) 25 hours ago         laughing_payne
9378a30ab43c   busybox   "echo hi there"        31 hours ago         Exited (0) 31 hours ago         brave_cohen
184d71b17162   hello-world "echo hi there"        2 days ago           Created                                jolly_villani
157458e4cb7f   hello-world "ls"                   2 days ago           Created                                sleepy_rosalind
2d7d3ad0e02b   busybox   "ls"                   3 days ago           Exited (0) 3 days ago         objective_tesla
d1b0c4409fdc   busybox   "echo hi there"        3 days ago           Exited (0) 3 days ago         gallant_lamport
6d31eabb3264   busybox   "echo hi there"        3 days ago           Exited (0) 3 days ago         exciting_euclid
0435ad479913   hello-world "/hello"               4 days ago           Exited (0) 4 days ago         festive_kilby
15f292ca42d5   hello-world "/hello"               13 days ago          Exited (0) 13 days ago         blissful_bohr
jorge@MacBook-Pro-de-Jorge ~ %
```

Cuando ejecutamos este comando, vemos una lista de todos los contenedores que hemos creado. No importa si esos contenedores ya se dejaron de ejecutar, o si se ejecutaron en nuestro nombre o de forma natural.

En la práctica, el comando *docker ps* se usa con mucha frecuencia. Uno de los usos más comunes de este comando es obtener el Id de un contenedor en ejecución, para después usar ese Id con otros comandos.