

Este diagrama de flujo ilustra el ciclo de vida de un contenedor Docker, mostrando los estados posibles y las acciones que los gestionan:

- CREATED** (Estado inicial, círculo amarillo): Se crea ejecutando `>_docker create` o `>_docker run`. Se transita a **PAUSE** o **START** ejecutando `>_docker start`.
- PAUSE** (Estado pausado, círculo naranja): Se crea ejecutando `>_docker pause` desde **START** o **STOPPED**. Se transita a **START** ejecutando `>_docker unpause`.
- START** (Estado de inicio, rectángulo azul): Se transita a **PAUSE** ejecutando `>_docker pause` o directamente a **RUNNING** ejecutando `>_docker start`.
- STOPPED** (Estado detenido, círculo rojo): Se crea ejecutando `>_docker stop` desde **START** o **RUNNING**, o `>_docker kill` desde **RUNNING**. Se transita a **START** ejecutando `>_docker start` o a **DELETED** ejecutando `>_docker rm`.
- DELETED** (Estado eliminado, círculo negro): Estado final tras ejecutar `>_docker rm` desde **STOPPED**.
- RUNNING** (Estado en ejecución, círculo verde): Estado activo. Se transita a **PAUSE** ejecutando `>_docker pause`, a **STOPPED** ejecutando `>_docker stop` o `>_docker kill`, o a **DELETED** ejecutando `>_docker rm`.
- Transiciones de emergencia desde RUNNING:**
 - die** (rectángulo azul): Proceso de contenedor salido. Transita a **STOPPED**.
 - OOM** (rectángulo azul): Muerto por falta de memoria. Transita a **STOPPED**.
 - kill** (rectángulo azul): Acción manual de eliminación. Transita a **STOPPED**.
- Transiciones de mantenimiento:**
 - restart** (rectángulo azul): Acción manual de reinicio. Transita de **STOPPED** a **START**.
 - die** (rectángulo azul): Acción manual de parada. Transita de **STOPPED** a **START**.

Si queremos correrlo, podemos hacer “docker start” o “docker run” que creará un contenedor y lo arrancará todo en una línea.

luego tenemos “docker stop” y “docker kill”. Estos dos detienen el contenedor, la única diferencia es la ejecución interna, aparte de que “kill”, fuerza la detención.

Por ultimo, el contenedor, esté parado o creado, tenemos la opción de borrarlo con “`docker rm`”.

Pregunta 2: Estados de un contenedor

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
27efe2681f45	nginx	"/docker-entrypoint..."	3 seconds ago	Exited (0) Less than a second ago		container5
795102a93e3d	nginx	"/docker-entrypoint..."	3 seconds ago	Up Less than a second	0.0.0.0:8085->80/tcp	container4
c46462af3851	nginx	"/docker-entrypoint..."	4 seconds ago	Up 3 seconds (Paused)	0.0.0.0:8084->80/tcp	container3
37243301005f	nginx	"/docker-entrypoint..."	5 seconds ago	Up 4 seconds	0.0.0.0:8083->80/tcp	container2
99d0286b99bc	nginx	"/docker-entrypoint..."	5 seconds ago	Created		container1

Script(en GIT también)

```
# Levantar 5 contenedores
docker create --name container1 -p 8082:80 nginx
docker run -d --name container2 -p 8083:80 nginx
docker run -d --name container3 -p 8084:80 nginx
docker run -d --name container4 -p 8085:80 nginx
docker run -d --name container5 -p 8086:80 nginx

#Ponerlos en diferentes estados
docker pause container3
docker restart container4
docker stop container5

# Mostrar los contenedores corriendo
docker ps -a
```

El proceso de levantarlo es:

1. ir al directorio del script
2. hacer “bash” para entrar en ubuntu
3. hacer “sh script.sh” para ejecutar el script
4. hacer captura del “docker ps -a”

Pregunta 3: Composicion y configuracion

<https://github.com/unaialberdi22/ExamenSantiEval2>