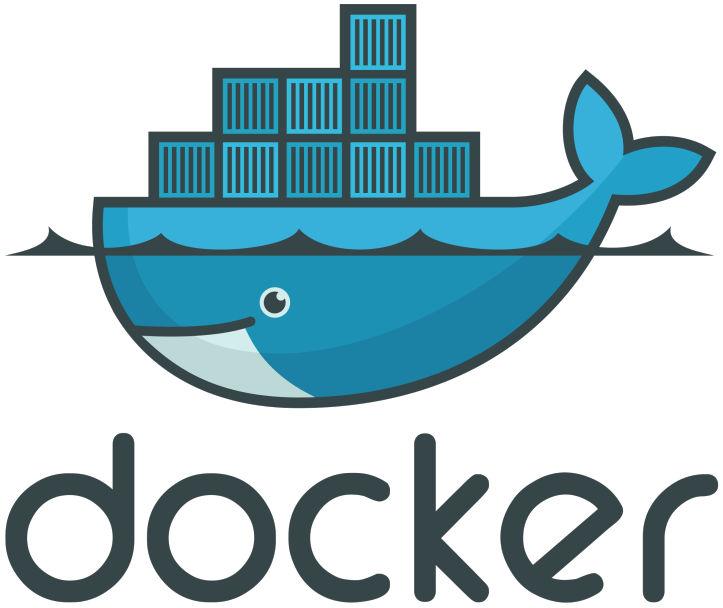
Docker 101 – Módulo 2

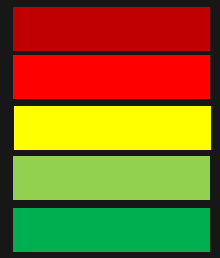
GBM Tech Academy

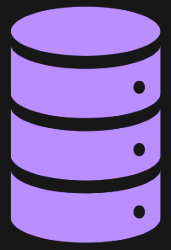
Agenda

- Tecnología de contenedores
- Creación y ejecución de contenedores
- Manipulación de contenedores
- Reinicio y parada de contenedores



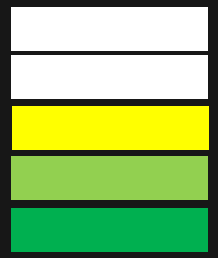
Virtual Machines vs Containers

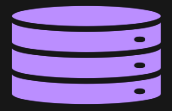

Utilization


Size


Boot Time

MINUTES


Utilization


Size


Boot Time

SECONDS

Virtual Machine

B2B Integrator

Libs

Dep

OS

Virtual Machine

File Gateway

Libs

Dep

OS

Hypervisor

Operating System

Hardware Infrastructure

Container

B2B Integrator

Libs

Dep

Container

File Gateway

Libs

Dep

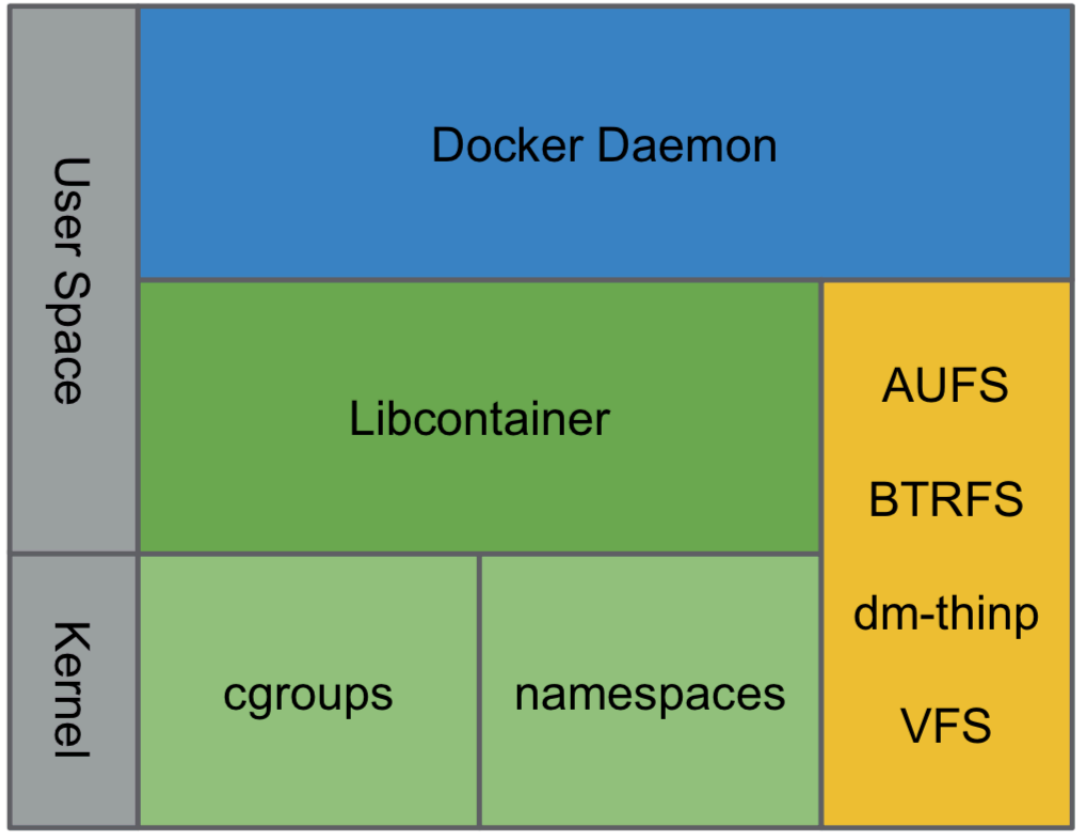
Docker

Operating System

Hardware Infrastructure

Linux y Contenedores

- Namespaces
  - Resource Isolation
- Cgroups
  - Resource Allocation
- Union File Systems
  - Layered File Systems
- Container Format
  - libcontainer
- Security
  - AppArmor, Seccomp, Capabilties



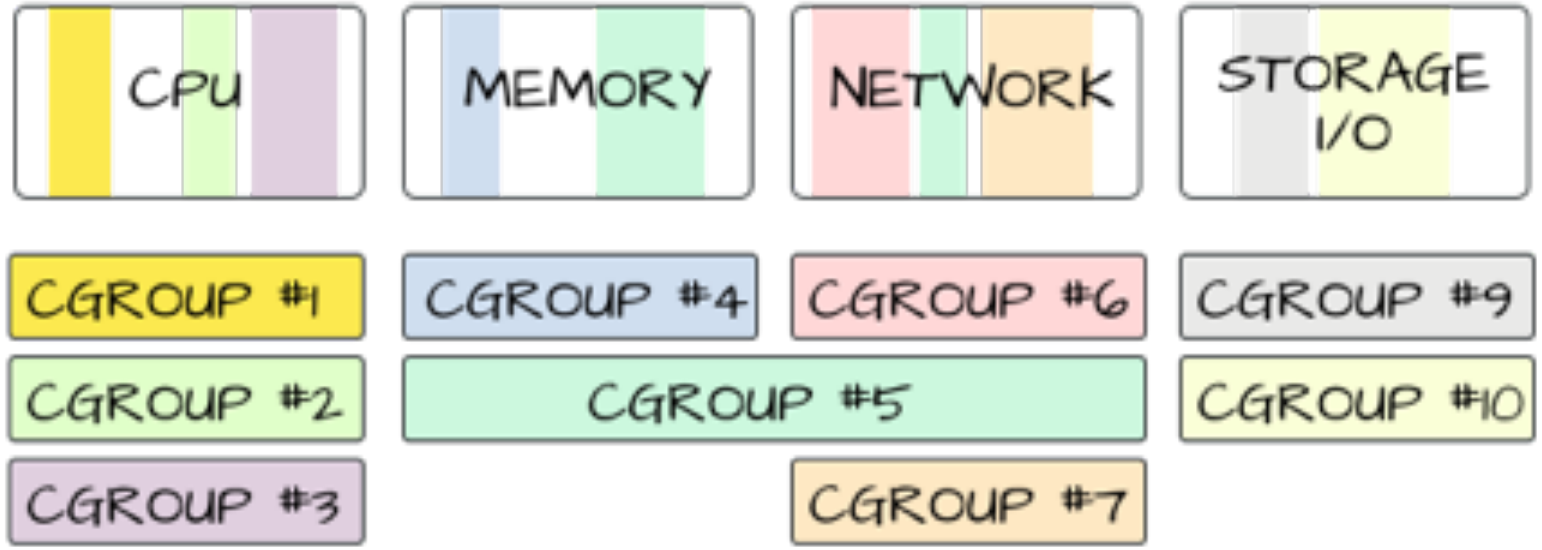
Namespaces

- Process trees.
- Mounts.
- Network.
- User accounts.
- Hostnames.
- Inter-process communication.

```
pid_t pid = clone(..., flags, ...)

    CLONE_NEWUTS     hostname, domainname
    CLONE_NEWIPC     IPC objects
    CLONE_NEWPID     Process IDs
    CLONE_NEWNET     Network configuration
    CLONE_NEWNS      File system mounts
    CLONE_NEWUSER    User and Group IDs

setns(int fd, int nstype)
    CLONE_NEWIPC
    CLONE_NEWNET
    CLONE_NEWUTS

Also: unshare(flags)
```

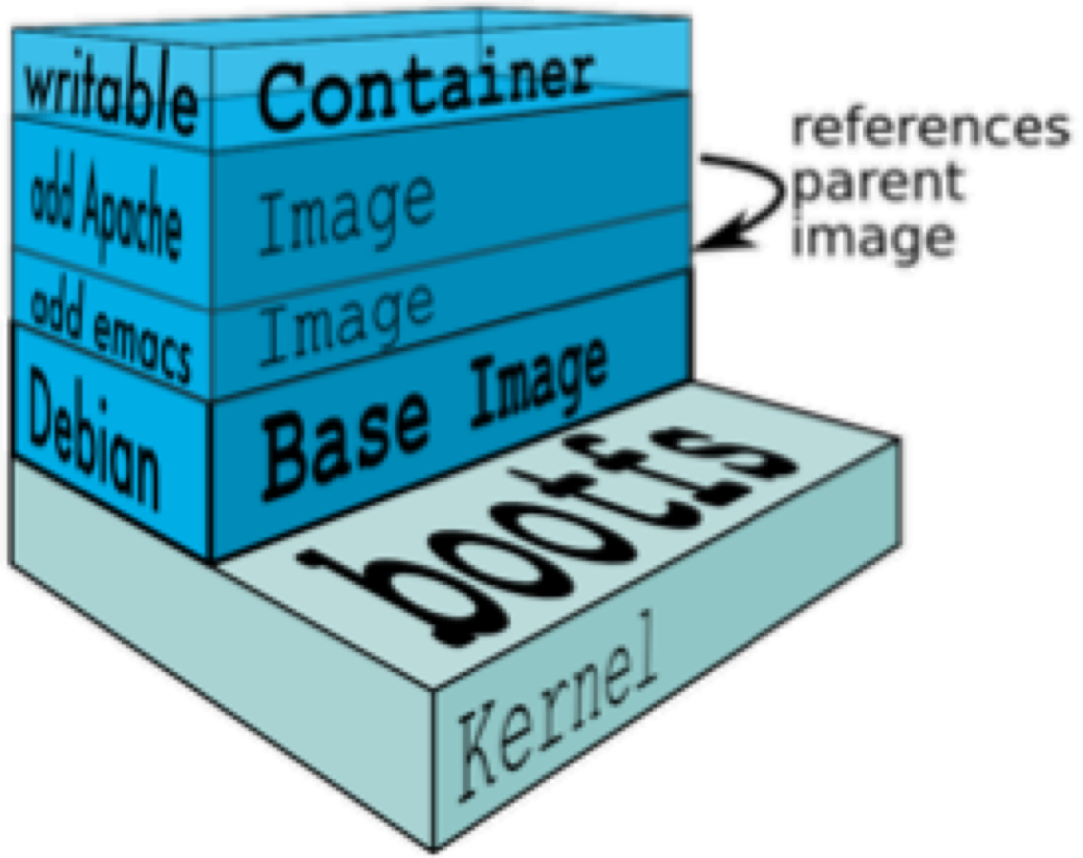
Cgroups: resource isolation, allocation and accounting

- cpu
- memory
- block i/o
- devices
- network
- numa
- freezer

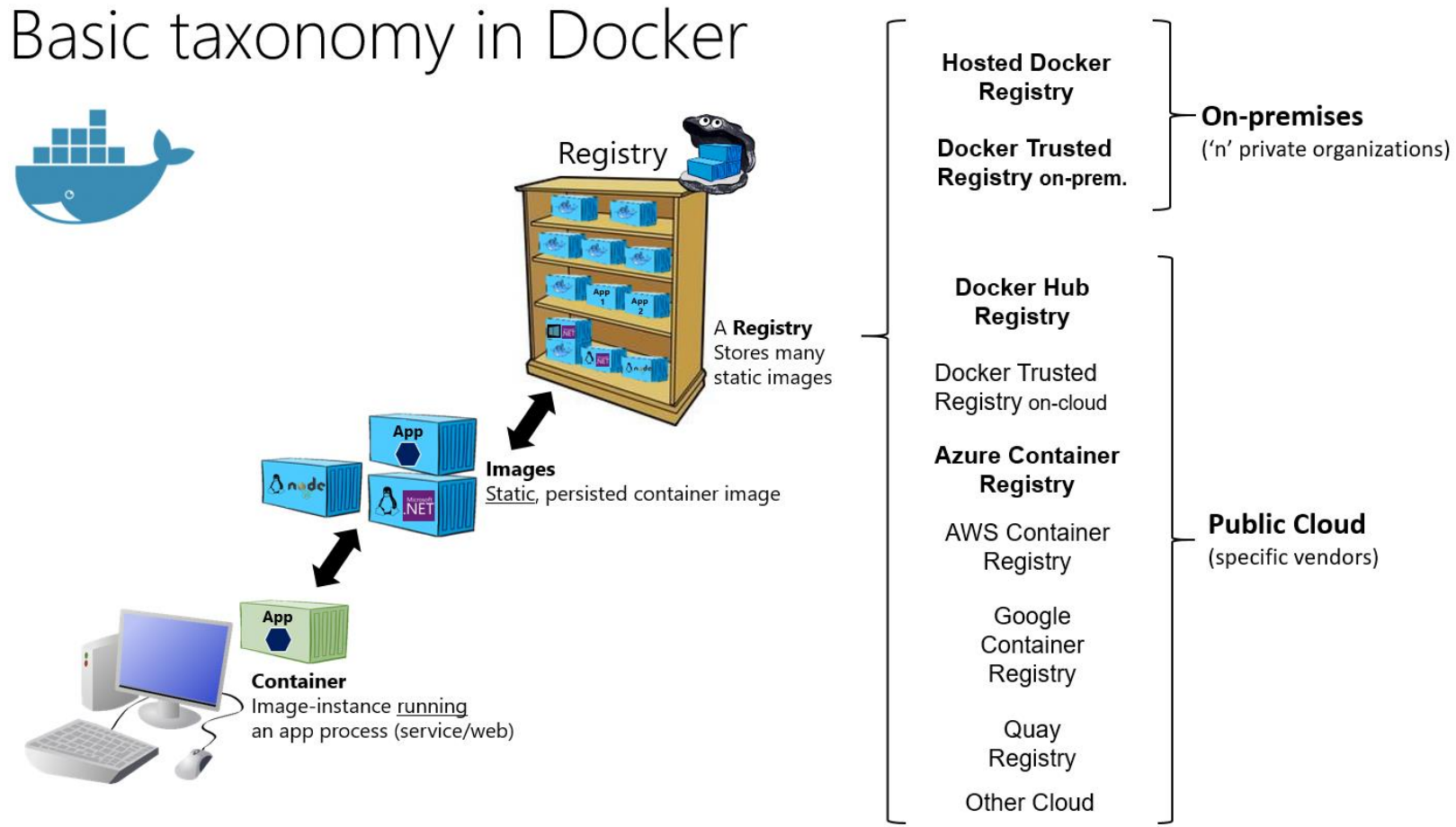


Layered File System

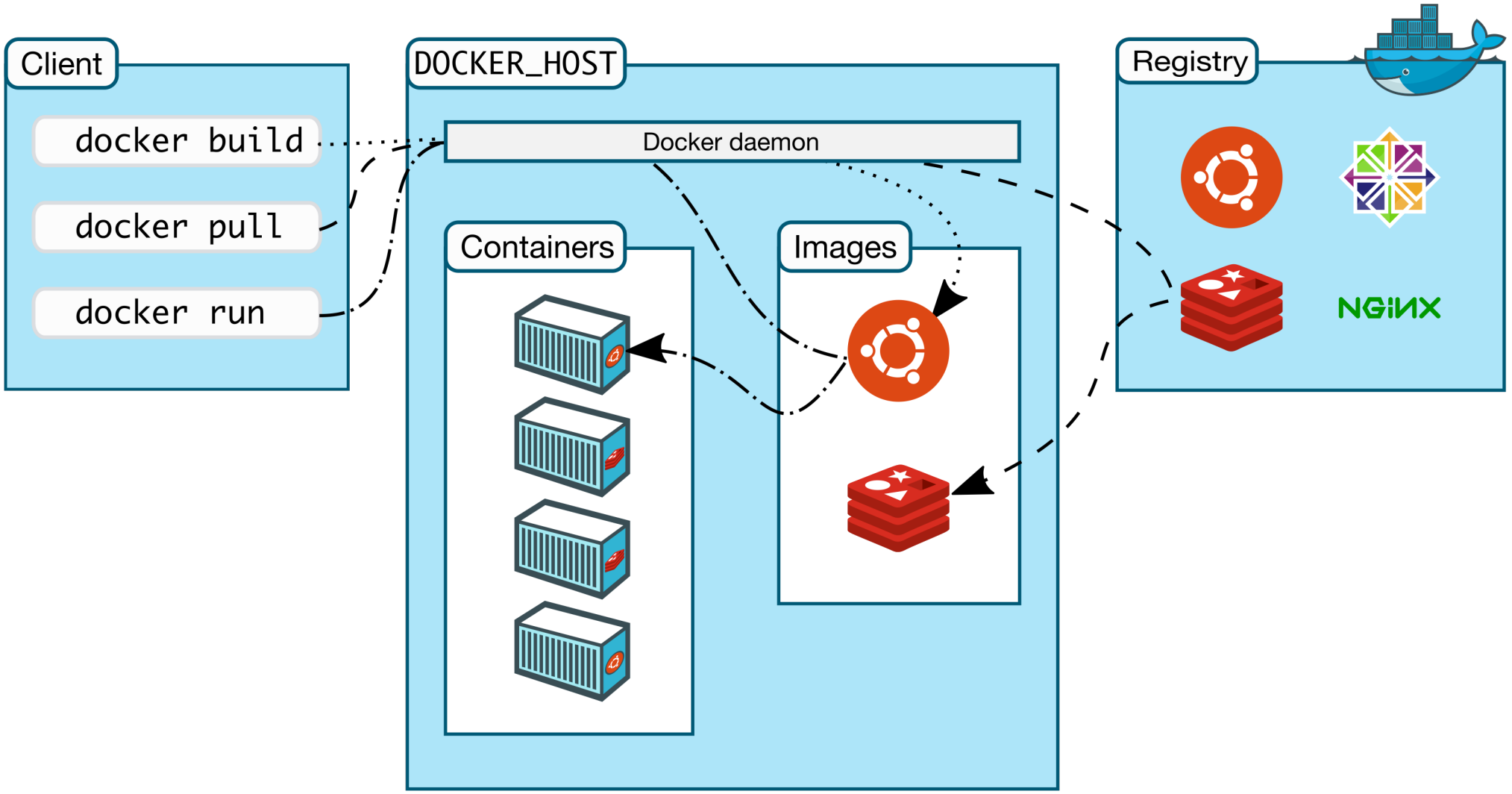
- **File-system Isolation**
  Building a rootfs dir and chroot into it. With mount namespace, use pivot-root.
- **Features**
  Layering, CoW, Caching, Diffing
- **Solutions**
  AUFS, btrfs, vfs, and devicemapper



Docker Container Lifecycle



Docker Container Lifecycle



Comandos básicos

Tarea	Comando base(*)
Información de Docker	docker versión docker info
Ayuda	docker –h docker command –h
Stats de Docker	docker stats

* Se muestran las opciones y parámetros mas comunes, documentación completa se encuentra en el sitio de docker

Creación y ejecución de contenedores

Tarea	Comando base(*)
Iniciar nuevo contenedor	**docker** run [-d] [-p host:cont] [-v host:cont] IMAGE[:tag] [alt. command]
Ejecutar comando en contenedor	**docker** exec [-it] NAME COMMAND
Vincular un contenedor	**docker** attach NAME
Ver los logs de un contenedor	**docker** logs NAME [–f] [–tail n]

* Se muestran las opciones y parámetros mas comunes, documentación completa se encuentra en el sitio de docker

Manipulación de contenedores

Tarea	Comando base(*)
Lista de contenedores en ejecución	**docker** ps [-a] [-q] | **docker** container ps [-a] [-q]
Inspeccionar un contendor	**docker** inspect NAME | **docker** container inspect NAME
Eliminar un contenedor	**docker** rm NAME | **docker** container rm NAME
Eliminar contenedores detenidos	**docker** container prune

* Se muestran las opciones y parámetros mas comunes, documentación completa se encuentra en el sitio de docker

Inicio y parada de contenedores

Tarea	Comando base(*)
Lista de contenedores en ejecución	**docker** ps [-a] [-q] | **docker** container ps [-a] [-q]
Arrancar un contenedor	**docker** start NAME | **docker** container start NAME
Detener un contenedor	**docker** stop NAME | **docker** container stop NAME
“Matar” un contenedor	**docker** kill NAME | **docker** container kill NAME
Inspeccionar un contendor	**docker** inspect NAME | **docker** container inspect NAME
Eliminar un contenedor	**docker** rm NAME | **docker** container rm NAME

* Se muestran las opciones y parámetros mas comunes, documentación completa se encuentra en el sitio de docker