Mondragon Unibertsitatea

Dockers: Docker Engine

Docker Engine

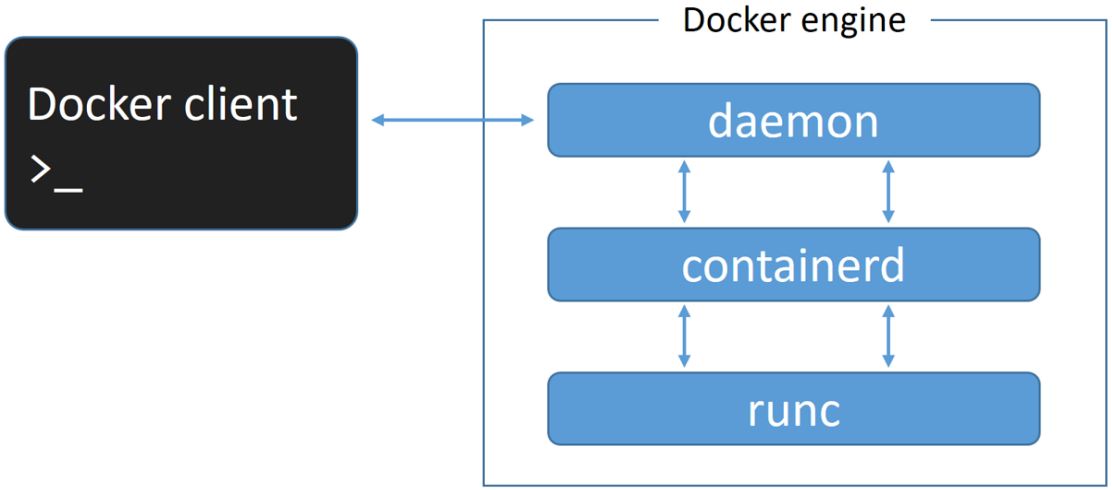
Mondragon Unibertsitatea
Goi Eskola Politeknikoa

- Puedes usar Docker sin entender nada de lo que trataremos en este capítulo.
- Este será un capítulo teórico sin ejercicios prácticos.
- Dividimos el capítulo en tres secciones:
  - **Idea general**
  - **Información detallada**
  - **Los comandos**

Docker Engine: Idea general

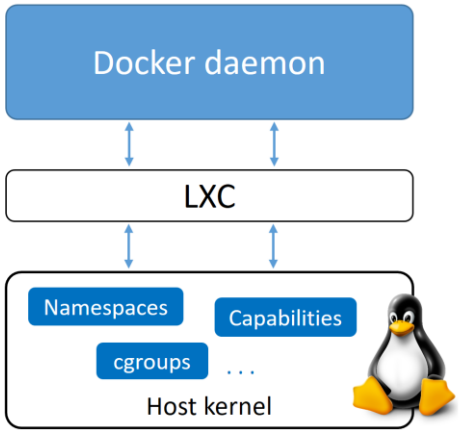
Mondragon Unibertsitatea
Goi Eskola Politeknikoa

- El Docker Engine tiene un diseño modular con muchos componentes intercambiables. En la medida de lo posible, se basan en normas abiertas definidas por la Open Container Initiative (OCI).
- Docker Engine es como un motor de coche - ambos son modulares y creados mediante la conexión de muchas pequeñas piezas especializadas:
  - Un motor de coche está hecho de muchas partes especializadas que trabajan en conjunto para hacer un coche de conducción - colectores de admisión, cuerpo del acelerador, cilindros, bujías de encendido, colectores de escape, etc.
  - El Docker Engine está hecho de muchas herramientas especializadas que trabajan juntas para crear y ejecutar contenedores - APIs, driver de ejecución, runtime, shims, etc.



Docker Engine en detalle

- Cuando se lanzó por primera vez Docker, el Docker Engine tenía dos componentes principales:
  - El demonio Docker
  - LXC (Linux Containers : https://linuxcontainers.org/)
- El demonio Docker era un binario monolítico. Contenía todo el código para el cliente Docker, la API Docker, el tiempo de ejecución del contenedor, la creación de imágenes y más.
- LXC proporcionó al demonio acceso a los bloques de construcción fundamentales de los contenedores que existían en el núcleo de Linux. Cosas como *namespaces (*espacios de nombres) y *grupos de control* (cgroups).



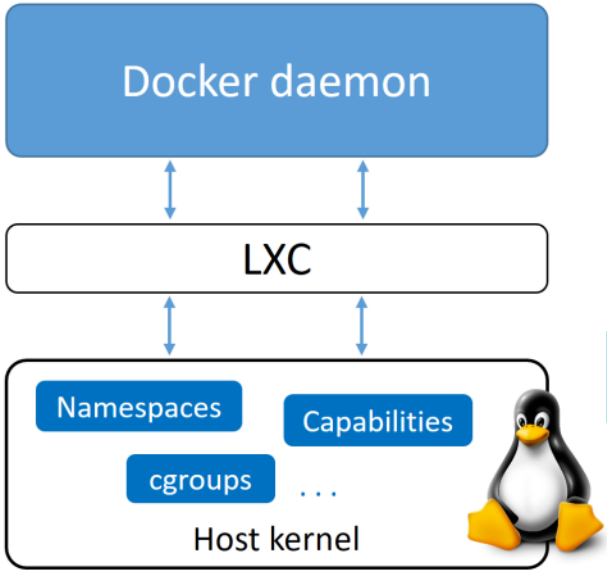
Docker Engine en detalle

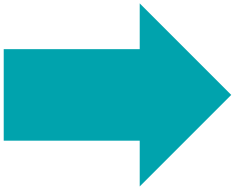
Mondragon Unibertsitatea
Goi Eskola Politeknikoa

- LXC: La dependencia de LXC fue un problema desde el principio.
  - LXC es específico de Linux. Esto era un problema para un proyecto que pretendía ser multiplataforma.
  - El hecho de depender de una herramienta externa para algo tan esencial para el proyecto era un riesgo enorme que podía obstaculizar el desarrollo.
- Docker. Inc. desarrolló una herramienta propia que se llama *libcontainer* para reemplazar LXC.
  - *Libcontainer:* herramienta agnóstica a la plataforma que proporcionaba a Docker acceso a los bloques de construcción de contenedores fundamentales que existen dentro del kernel.
- **demonio Docker monolítico:**
  - Es difícil innovar.
  - Va más lento.
  - No era lo que el ecosistema (o Docker, Inc.) quería
- **toda la *ejecución del contenedor* y el código de *runtime* del contenedor se han eliminado completamente del demonio y se han refactorizado en pequeñas herramientas especializadas.**

Docker Engine en detalle

Mondragon Unibertsitatea
Goi Eskola Politeknikoa





Docker commands (CLI)

API and other features

Container supervisor
`start|stop|pause..`

Enables daemonless containers

Container runtime
(interface to kernel primitives)

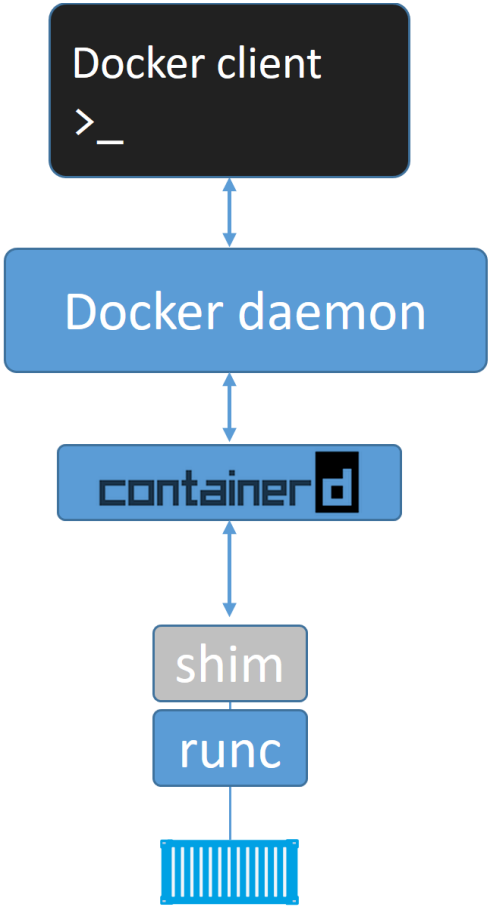
Running containers

Docker Engine: The Deep Dive

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

La forma más común de iniciar contenedores es utilizando el Docker CLI.

```
$ docker container run --name ctr1 -it alpine:latest sh
```



Issue `**docker container run**` command to **Docker API**.

Receive instruction at API endpoint.
Instruct **containerd** to start new container.

Give **runc** OCI bundle (image).
Instruct **runc** to create container.

Create container

Docker Engine en detalle

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Beneficios de este modelo

- Tener toda la lógica y el código para iniciar y gestionar contenedores fuera/eliminado del demonio significa que todo lo relativo a la parte del tiempo de ejecución del contenedor está desacoplado del demonio Docker: **"daemonless containers"**
  - hace posible realizar tareas de mantenimiento y actualizaciones en el demonio Docker sin afectar a los contenedores que están en ejecución.

- En el modelo anterior, toda la lógica de tiempo de ejecución estaba implementado en el demonio, arrancar y detener el demonio mataba todos los contenedores que estaban en ejecución en el host.

Mondragon Unibertsitatea

Goi Eskola Politeknikoa

Eskerrik asko

Muchas gracias

Thank you