



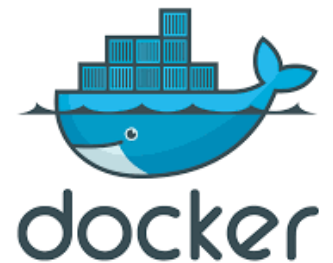
UNIVERSIDAD
DE GRANADA



Programación Web

DESPLIEGUE DE APLICACIONES WEB CON CONTENEDORES: DOCKER

Curso 2018-2019



SPLIEGUE DE LICACIONES WEB CON CONTENEDORES: DOCKER

Contenedores

- En la virtualización con hipervisores, una o más máquinas se ejecutan virtualmente sobre el hardware a través de una capa de intermediación
- Los contenedores se ejecutan en espacio de usuario, sobre el kernel del S.O.
- Se pueden ejecutar muchos espacios de usuario aislados en un mismo equipo

- Los contenedores son menos flexibles, pues sólo pueden ejecutar el mismo S.O. o similar al instalado en la máquina anfitriona
- Son populares en entornos de servicios *multi-tenant*.

Docker

- Motor que facilita el despliegue de aplicaciones en contenedores
- Añade un motor de despliegue de aplicaciones sobre un entorno de ejecución virtualizada de contenedores
- Diseñado para poder desplegar aplicaciones de forma eficiente desde el entorno de desarrollo (¿portátil?) a producción

Propiedades

- Fácil y rápido: modelo “copy-on-write”
- Separación lógica de tareas:
 - Desarrolladores: aplicaciones corriendo en contenedores
 - Administradores: gestión de contenedores
 - El entorno de desarrollo y producción es el **mismo**
- Ciclo de desarrollo rápido y eficiente
- Arquitectura orientada a servicios

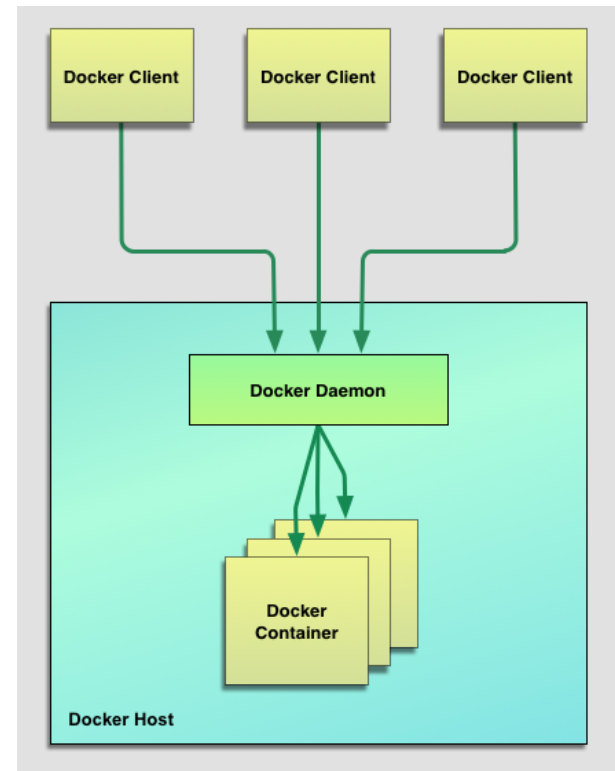
Componentes

- Servidor y clientes
- Imágenes
- Registros
- Contenedores

Cliente-servidor

- Servidor: demonio
- Cliente: programa

`docker`



Imágenes

- “Código fuente” para los contenedores.
- Se construyen mediante la unión de sistemas de archivos
- Se apilan capas correspondientes a distintos sistemas de archivos

Registros

- Repositorios de imágenes
- Hay públicos y privados

Contenedores

- Ubicación para el despliegue de aplicaciones y servicios
- Se “lanzan” y pueden tener uno o más procesos en ejecución
- Las imágenes → construyen contenedores
- Docker → ejecuta contenedores
- Son contenedores a imagen de los de mercancías, pero de software

Usos habituales de contenedores

- Desarrollo local y eficiente de flujos de trabajo
- Ejecución de servicios independientes y aislados
- Entornos aislados para prueba y validación
- Construir una PaaS
- Entornos de desarrollo ligeros y aislados
- Servicios SaaS

Componentes técnicos

- Docker se ejecuta en arquitecturas x64 con un kernel de Linux moderno (mejor 3.8+)
 - Libcontainer
 - Linux kernel namespaces
 - Aislantes para:
 - Sistema de archivos
 - Procesos
 - Red
 - Recursos: CPU y memoria
 - Copy-onwrite
 - Registro en bitácora

Instalación de docker

- Servidores con Ubuntu
- RHEL o derivados (CentOS)
- OS X con Boot2Docker
- Windows con Boot2Docker

Interfaces de usuario

- Shipyard
- DockerUI
- maDocker
- Programa `docker`

Lista de mandatos de docker

- <http://docs.docker.com/reference/commandline/cli>

Tutoriales de uso de docker

- <https://docs.docker.com/get-started/>
- https://github.com/DiCITS/MasterCienciaDatos2019/blob/master/Docker/starting_docker.md#starting-with-docker-containers--primeros-pasos-con-contenedores-en-docker-

Algunos mandatos

- Listado de imágenes disponibles

```
docker images
```

Obtener nuevas imágenes

```
docker pull nginx
```

Lanzar un contenedor a partir de una imagen

```
docker run -d -p
```

```
<puerto:host>:<puerto:contenedor> --name
```

```
<nombre_cont> <contenedor>
```

```
docker run -d -p 10080:80 --name mynginx  
nginx
```

Gestión de contenedores

- Ver contenedores en ejecución
`docker ps`
- Detener un contenedor en ejecución
`docker stop <nombre>`
- Reiniciar un contenedor
`docker restart <nombre>`
- Eliminar un contenedor en ejecución
`docker kill <nombre>`
- Bitácora de un contenedor
`Docker logs <nombre>`

Editar el contenido de un contenedor

- Editar el contenido de un contenedor

```
docker exec -i -t nginx /bin/bash
vim /usr/share/nginx/html/index.html
apt-get install vim
```

Creación de imágenes propias

<https://docs.docker.com/get-started/part2/>

- Se necesita crear un fichero de configuración, llamado `dockerfile`
- En un directorio vacío, crear un fichero de texto “`dockerfile`”
- Crear la imagen:
`docker build --tag"etiqueta"`