

Introducción a Docker

1. Docker como alternativa ligera a la virtualización	13
1.1 El enfoque por virtualización	14
1.2 Docker para optimizar el uso de los recursos	15
1.3 Posicionamiento de Docker respecto a la virtualización	18
2. Principio de los contenedores	21
2.1 Principio de los contenedores industriales	22
2.2 Docker y el enfoque normalizado	23
3. Los aspectos principales de Docker	24
3.1 Espacio de nombres	24
3.2 Cgroups	26
3.3 LXC	26
3.4 Libcontainer	27
3.5 Sistema de archivos en capas	29
3.5.1 Principio de aislamiento de los archivos	29
3.5.2 Enfoque por virtualización	29
3.5.3 Herramienta de los sistemas de archivos en capas	30
3.5.4 Gestión de las modificaciones de archivos	32

3.5.5 Última capa en modo escritura	34
3.5.6 Tecnologías utilizadas	35
4. Aspectos adicionales de Docker	36
5. El ecosistema Docker	37
6. Arquitecturas de servicios	38
6.1 Histórico de las arquitecturas de servicios	38
6.1.1 Aspectos principales	38
6.1.2 Enfoque EAI	39
6.1.3 Enfoque SOA	39
6.1.4 Microservicios	40
6.1.5 Relacionado con la construcción de los SI	40
6.2 Arquitectura de micro-servicios	41
6.2.1 Aspectos principales	41
6.2.2 Ventajas	43
6.2.3 Inconvenientes	46
6.3 Qué aporta Docker	47
6.4 Hilo conductor	48

Primeros pasos

1. Instalación de Docker	49
1.1 Utilizar máquinas pre-configuradas	50
1.2 Instalación de Docker en Linux	62
1.2.1 Requisitos previos de sistema	62
1.2.2 Instalación por administrador de paquetes	62
1.2.3 Instalación por script	67
1.3 Instalación de Docker en Windows	68
1.3.1 Una paradoja resuelta	68
1.3.2 Instalación en Windows 10	70
1.3.3 Instalación en Windows Server 2016	74
1.3.4 Máquina virtual	75
2. Hello World, Docker	76
2.1 Puesta en marcha de un contenedor sencillo	76
2.2 ¿ Qué ha sucedido ?	77
2.2.1 Recuperación de la imagen	77
2.2.2 Anatomía de la imagen obtenida	78
2.2.3 Ejecución del proceso	82
2.2.4 Parada del contenedor	84
3. Utilizar imágenes Docker preexistentes	86

3.1 El Docker Store	86
3.1.1 El principio	86
3.1.2 Búsqueda y cualificación de imágenes	88
3.1.3 Ejemplo de búsqueda	90
3.1.4 Caso de las imágenes de la comunidad	92
3.1.5 Complementos a las imágenes oficiales	94
3.1.6 Enlace con el registro Docker Hub	96
3.1.7 Búsqueda por la línea de comandos	97
3.1.8 Precauciones con una imagen no oficial	100
3.2 Gestión de la cuenta Docker Hub y almacenes privados	106
3.2.1 Creación de una cuenta	107
3.2.2 Características de la cuenta	108
3.2.3 Automated build y cuenta GitHub	110
3.2.4 Conexión a la cuenta en línea de comandos	117
3.2.5 Webhook en evento de push en Docker Hub	118
3.2.6 Desconexión de las cuentas Docker Hub y GitHub	120
4. Un segundo contenedor	122
4.1 Recuperación de la imagen	122
4.2 Explicación de los tags	123
4.3 Primera ejecución	125

4.4 Ejecución en modo interactivo	128
4.5 Persistencia de las modificaciones en forma de imagen	130
4.6 Primeros pasos con el cliente Docker	133
4.6.1 Limpieza en los contenedores	133
4.6.2 Limpieza en las imágenes	134
4.6.3 La gran limpieza	134
4.6.4 Eliminación automática a la salida	135
4.6.5 Asignación de un nombre de contenedor	136
4.6.6 Modificación del punto de entrada por defecto	137
4.6.7 Enviar variables de entorno	138
4.6.8 Modificación de la hostname	140
4.7 Manipulación de los contenedores	141
4.7.1 Ejecución en modo bloqueante	141
4.7.2 Ejecución en segundo plano	143
4.7.3 Gestión correcta del ciclo de vida de los contenedores	148
4.7.4 Exposición de archivos	151
4.7.5 Supervisión de los contenedores	153
5. Regreso a los primeros pasos	155

Creación de sus propias imágenes

1. Creación manual de una nueva imagen	157
1.1 Instalación de un software en un contenedor	157
1.2 Persistencia de la imagen para un uso futuro	160
1.3 Utilización de la imagen creada	161
1.4 Conexión desde la máquina host	163
1.5 Seguimiento de las operaciones	165
2. Utilización de un Dockerfile	166
2.1 Interés de los archivos Dockerfile	166
2.2 Utilización de un archivo Dockerfile	168
2.3 Resultados de la utilización de un Dockerfile completo	170
2.4 Anatomía de un archivo Dockerfile	171
2.4.1 FROM	172
2.4.2 RUN	172
2.4.3 ENV	173
2.4.4 VOLUME	175
2.4.5 COPY	179
2.4.6 ENTRYPOINT	180
2.4.7 EXPOSE	182
2.4.8 CMD	182
2.5 Nuestro primer Dockerfile	182

2.5.1 Creación y prueba del script	183
2.5.2 Creación de Dockerfile	183
2.5.3 Generación de la imagen	185
2.5.4 Ejecución del contenedor	187
2.5.5 Detener y volver a lanzar el contenedor	188
2.5.6 Gestión de los argumentos	189
2.5.7 Reconstrucción de la imagen y cacheado	191
2.6 Comandos adicionales	192
2.6.1 Gestión de los archivos	194
2.6.2 Noción de contexto	194
2.6.3 Resultados de la asignación del proceso de arrancar	195
2.6.4 Observación sobre el formato línea de comandos o ejecución	196
2.6.5 Comandos variados	199
3. Compartición y reutilización sencilla de imágenes	201
3.1 Enviar a su cuenta Docker Hub	205
3.2 Exportación e importación en forma de archivos	210
4. Buenas prácticas	211
4.1 Principio de la caché de las imágenes	211
4.2 Principio de la caché durante la compilación	216

4.2.1	Vuelta a las imágenes intermedias	216
4.2.2	Anatomía de una compilación de imagen	220
4.2.3	Análisis de una modificación de la Dockerfile	222
4.2.4	Gestión correcta de las etiquetas	225
4.2.5	Invalidación de la caché por modificación de la imagen básica	229
4.2.6	Invalidación de la caché, modificando el contexto	234
4.3	Consecuencias de la escritura de Dockerfile	236
4.3.1	El problema en las operaciones no idempotentes	236
4.3.2	Solución del problema de la caché	239
4.3.3	Efectos beneficiosos del número y tamaño de las imágenes	243
4.3.4	Programación de los comandos en el Dockerfile	245
4.4	Consecuencias de la elección de las imágenes básicas	246
4.4.1	La imagen básica correcta	246
4.4.2	Su propia imagen básica	248
4.5	Arborescencia recomendada	251
4.5.1	Ventajas de una arborescencia tipo	251
4.5.2	Integración de los archivos	252
4.5.3	Limitación del contexto	254
4.6	La pregunta del proceso único	255
4.6.1	Principio general	255
4.6.2	Excepción al principio general con Supervisor	255

4.6.3 Crítica	257
4.6.4 Enfoque intermedio	258
5. Para ir más lejos	260
Instalación de un registro privado	
1. Primeros pasos para un registro privado	261
1.1 Observación inicial importante	262
1.2 Advertencia sobre el antiguo registro	262
1.3 Imagen Docker en local	263
1.4 Apuntar a un registro dado	265
1.5 Registro en una red pública	268
1.5.1 Escenario y preparación de las máquinas	269
1.5.2 Puesta en marcha del registro	270
1.5.3 Depositar la imagen desde otra máquina	273
1.5.4 Utilización de la imagen desde una tercera máquina	276
1.5.5 Eliminación de la imagen en la máquina origen	277
2. Un registro más profesional	279
2.1 Gestión de la persistencia	279
2.1.1 Gestión local por volumen	279

2.1.2 Observación sur SELinux	279
2.1.3 Un almacenamiento más seguro	281
2.2 Securización del registro	282
2.2.1 Preparación de las claves para el canal encriptado	284
2.2.2 Implantación del front de Nginx	284
2.2.3 Acceso por el cliente Docker	287
2.2.4 Herramientas adicionales de diagnóstico	293
2.2.5 Generación del archivo de los usuarios autorizados	295
2.2.6 Añadir la autenticación en Nginx	297
2.2.7 Corrección en los headers	298
2.2.8 Gestión de la conexión	300
2.3 Un verdadero registro en producción	301
3. Utilización de un servicio de registro en el cloud	302
3.1 Principio	304
3.2 Oferta de pago de Docker Hub	306
3.3 Azure Container Registry	309
3.3.1 Preparación	309
3.3.2 Creación del registro	310
3.3.3 Configuración	314

4. Enfoques adicionales	316
4.1 El API del registro	316
4.2 Implementación de un espejo	317
4.3 Implementación de una caché para los paquetes	321
4.3.1 Posible solución	321
4.3.2 Implementación de una caché de paquetes	323
Implementación de una arquitectura de software	
1. Presentación de la aplicación ejemplo	327
1.1 Arquitectura	327
1.2 Instalación	329
1.3 Utilización	332
1.4 Herramienta	334
1.5 Aspectos principales del ejercicio	335
1.5.1 Una palabra sobre las arquitecturas de micro-servicios	335
1.5.2 Relación con la programación SOLID	336
2. Creación de la arquitectura de ejemplo	337
2.1 Principios de construcción	337
2.2 Detalles del servicio optimizer	338

2.2.1 Funcionamiento	338
2.2.2 Integración en Docker	343
2.2.3 Pruebas	346
2.2.4 Observaciones	348
2.3 Detalles del servicio de calculador	349
2.3.1 Funcionalmente	349
2.3.2 Integración en Docker	351
2.3.3 Pruebas	353
2.4 Implementación de enlaces entre contenedores	356
2.5 Detalles del servicio reporting	361
2.5.1 Funcionamiento	361
2.5.2 Dockerización	363
2.5.3 Pruebas	364
2.6 Detalles del servicio notifier	366
2.6.1 Funcionamiento	366
2.6.2 Dockerización	369
2.6.3 Pruebas	370
2.7 Detalles del servicio de persistencia	374
2.8 Implementación eventual de una imagen básica	375
2.9 Detalles del servicio portal	378
2.9.1 Funcionamiento	378

2.9.2 Dockerización	381
2.9.3 Pruebas	382
2.10 Estado alcanzado	383
3. Desarrollar automáticamente con Docker Compose	384
3.1 Principio de Docker Compose	384
3.2 Escritura del archivo docker-compose.yml	385
3.3 Implementación	388
3.3.1 Preparación	388
3.3.2 Ejecución de los contenedores	389
3.3.3 Gestión de los contenedores	391
3.4 Paralelización de las operaciones	392
3.5 Límites	394
4. Explotación de una infraestructura Docker	395
4.1 La red en Docker	395
4.1.1 Modo de funcionamiento estándar (bridge)	395
4.1.2 Modos de funcionamiento alternativos	398
4.1.3 Soporte de los enlaces entre contenedores	401
4.1.4 Otras opciones	403
4.1.5 Límites de la capa de red existente	403
4.2 Los volúmenes Docker	403

4.2.1 El problema de la persistencia	404
4.2.2 Los volúmenes como solución sencilla	404
4.2.3 Enlace directo sobre un directorio local	405
4.2.4 Compartir volúmenes	407
4.2.5 Gestión de los volúmenes huérfanos	408
4.2.6 Sofisticación del enfoque	409
4.2.7 Aplicación para la gestión de los logs	410
4.2.8 Ir más lejos con los volúmenes	411
	412

Despliegue en un cluster

1. Descripción global del enfoque

1.1 Objetivo	413
1.2 Situación actual	413
1.3 Desarrollo del ejemplo	414
1.4 Advertencia	415
	415

2. Montaje de un cluster Swarm

2.1 Funcionamiento resumido de Swarm	416
2.2 Descripción de las máquinas utilizadas	416
2.3 Inicialización del cluster	417
	421

2.4 Adjuntar el resto de máquinas	421
2.5 Control remoto	422
3. Despliegue en el cluster Swarm	423
3.1 Aspectos principales de los servicios	423
3.2 Envío de las imágenes al registro	425
3.3 Ejecución de un stack	427
3.4 Escalado	431
4. Alguna información adicional sobre Swarm	433
4.1 ¿ Qué sucede en su interior ?	433
4.1.1 Comandos de diagnóstico	434
4.1.2 Gestión de la red overlay y de los puertos	437
4.2 Para ir más allá	438
4.2.1 Detener la plataforma	438
4.2.2 Conexión a un registro privado	439
4.2.3 Gestión de las restricciones	440
4.2.4 Despliegue incremental (rolling update)	440

Ir aún más allá con Docker

1. Docker en su fábrica de software

	445
1.1 Docker a todos los niveles	445
1.2 Producir imágenes como salida de build	447
1.2.1 Posicionamiento	447
1.2.2 Herramienta	447
1.2.3 Consejos de implementación	448
1.3 Utilizar contenedores para la fábrica de software	449
1.3.1 Posicionamiento	449
1.3.2 Herramienta	449
1.3.3 Consejos de implementación	450
1.4 Utilizar los contenedores para las pruebas	451
1.4.1 Posicionamiento	451
1.4.2 Herramienta	451
1.4.3 Consejos de implementación	452
1.4.4 Variante asociada	452
1.5 Vuelta al registro	454
2. Antes de empezar en producción con Docker	455
2.1 Seguridad	455
2.2 Restricción sobre los recursos	456
3. Docker y Windows	457

4. Conclusión	457
----------------------	------------

índice	459
---------------	------------