

INCEPTION

Conceptos Fundamentales de Docker

Guía de Conceptos para el Proyecto

42 School - System Administration

ÍNDICE

1. Introducción a Docker
2. Contenedores vs Máquinas Virtuales
3. Imágenes Docker
4. Docker Compose
5. Redes en Docker
6. Volúmenes y Persistencia
7. NGINX
8. PHP-FPM
9. FastCGI
10. MariaDB
11. WordPress
12. SSL/TLS
13. WP-CLI
14. Procesos y PID 1
15. Proxy Reverso
16. Conceptos de Seguridad

1. INTRODUCCIÓN A DOCKER

¿Qué es Docker?

Docker es una plataforma de código abierto que permite automatizar el despliegue de aplicaciones dentro de contenedores de software. Un contenedor empaqueta una aplicación junto con todas sus dependencias (bibliotecas, archivos de configuración, etc.) en una unidad estándar que puede ejecutarse de manera consistente en cualquier entorno.

¿Por qué usar Docker?

Portabilidad: Los contenedores funcionan igual en desarrollo, testing y producción.

Aislamiento: Cada contenedor está aislado de otros contenedores y del sistema host.

Eficiencia: Los contenedores comparten el kernel del sistema operativo, siendo más ligeros que las máquinas virtuales.

Reproducibilidad: El mismo contenedor produce los mismos resultados en cualquier lugar.

2. CONTENEDORES VS MÁQUINAS VIRTUALES

Máquina Virtual (VM)

Una máquina virtual es un sistema operativo completo que se ejecuta sobre un hipervisor. Cada VM incluye una copia completa del sistema operativo, las aplicaciones, las bibliotecas necesarias y varios gigabytes de espacio en disco. Esto las hace pesadas y lentas de iniciar.

Contenedor Docker

Un contenedor comparte el kernel del sistema operativo host y aísla solo la aplicación y sus dependencias. No necesita un sistema operativo completo, lo que lo hace mucho más ligero (típicamente megabytes en lugar de gigabytes) y rápido de iniciar (segundos en lugar de minutos).

Diferencias Clave

Tamaño: VMs (varios GB) vs Contenedores (varios MB)

Velocidad: VMs (minutos para arrancar) vs Contenedores (segundos)

Recursos: VMs (reservan recursos) vs Contenedores (comparten recursos dinámicamente)

Aislamiento: VMs (aislamiento completo de hardware) vs Contenedores (aislamiento a nivel de proceso)

3. IMÁGENES DOCKER

¿Qué es una Imagen Docker?

Una imagen Docker es una plantilla de solo lectura que contiene un conjunto de instrucciones para crear un contenedor. Es como una 'foto' o 'instantánea' de un sistema con todo lo necesario para ejecutar una aplicación: el sistema operativo base, el software, las bibliotecas, las dependencias y el código de la aplicación.

Dockerfile

Un Dockerfile es un archivo de texto que contiene las instrucciones necesarias para construir una imagen Docker. Define qué sistema operativo base usar, qué software instalar, qué archivos copiar y qué comando ejecutar cuando se inicie el contenedor.

Capas (Layers)

Las imágenes Docker están compuestas por capas. Cada instrucción en el Dockerfile crea una nueva capa. Las capas son de solo lectura y se apilan unas sobre otras. Docker reutiliza capas comunes entre diferentes imágenes, ahorrando espacio en disco.

Imágenes Base

En el proyecto Inception, todas las imágenes se construyen desde Debian Bookworm como imagen base. Debian es una distribución Linux estable y bien documentada. Bookworm es el nombre en código de Debian 12, la versión estable actual.

4. DOCKER COMPOSE

¿Qué es Docker Compose?

Docker Compose es una herramienta para definir y ejecutar aplicaciones Docker multi-contenedor. Utiliza un archivo YAML para configurar todos los servicios de la aplicación, sus redes y volúmenes. Con un solo comando se pueden crear e iniciar todos los servicios definidos.

docker-compose.yml

Es el archivo de configuración donde se define la infraestructura completa. En él se especifica: qué contenedores crear, desde qué imágenes, qué puertos exponer, qué volúmenes montar, qué redes usar y qué variables de entorno configurar.

Servicios

Un servicio es la definición de un contenedor en Docker Compose. Cada servicio tiene un nombre (como nginx, wordpress, mariadb) y una configuración asociada. Los servicios pueden comunicarse entre sí usando sus nombres como hostnames gracias al DNS interno de Docker.

Orquestación

Docker Compose orquesta el arranque de los contenedores en el orden correcto, respetando las dependencias. Si WordPress depende de MariaDB, Compose se asegura de que MariaDB esté listo antes de iniciar WordPress.

5. REDES EN DOCKER

Docker Network

Una red Docker permite que los contenedores se comuniquen entre sí. Docker proporciona aislamiento de red por defecto, de modo que los contenedores en diferentes redes no pueden comunicarse directamente.

Red Bridge

Es el tipo de red más común en Docker. Crea una red privada interna en el host donde los contenedores conectados pueden comunicarse entre sí. Es como tener un switch virtual al que todos los contenedores se conectan. En Inception, todos los servicios están en una red bridge llamada 'inception'.

DNS Interno

Docker proporciona un servidor DNS interno automático. Cada contenedor puede comunicarse con otros usando el nombre del servicio como hostname. Por ejemplo, WordPress puede conectarse a MariaDB simplemente usando 'mariadb:3306' como dirección del servidor de base de datos.

Aislamiento

Los contenedores en la misma red pueden comunicarse, pero están aislados del host y de otras redes. Solo los puertos explícitamente expuestos (como el 443 de NGINX) son accesibles desde fuera de la red Docker.

6. VOLÚMENES Y PERSISTENCIA

¿Qué es un Volumen?

Un volumen Docker es un mecanismo para persistir datos generados y utilizados por los contenedores. Los contenedores son efímeros por naturaleza: cuando se eliminan, todos sus datos se pierden. Los volúmenes permiten que los datos sobrevivan al ciclo de vida del contenedor.

Bind Mount

Un bind mount vincula un directorio del sistema host a un directorio dentro del contenedor. En Inception, los datos de WordPress y MariaDB se almacenan en `/home/usuario/data/` del host y se montan dentro de los contenedores. Esto permite acceder a los datos directamente desde el host y garantiza su persistencia.

Persistencia de Datos

Gracias a los volúmenes, los datos importantes persisten incluso si los contenedores se detienen, reinician o eliminan. En el proyecto, el blog de WordPress y la base de datos MariaDB se mantienen seguros en el host, permitiendo reconstruir los contenedores sin perder ningún dato.

7. NGINX

¿Qué es NGINX?

NGINX es un servidor web de alto rendimiento y proxy reverso. Es conocido por su eficiencia, estabilidad y bajo consumo de recursos. Puede manejar miles de conexiones simultáneas gracias a su arquitectura basada en eventos.

Servidor Web

Como servidor web, NGINX puede servir archivos estáticos directamente: HTML, CSS, JavaScript, imágenes, etc. Es muy eficiente en esta tarea porque utiliza técnicas avanzadas de transferencia de archivos a nivel de kernel del sistema operativo.

Arquitectura de Eventos

A diferencia de otros servidores que crean un proceso o hilo por conexión, NGINX usa un modelo asíncrono basado en eventos. Tiene un proceso maestro y múltiples procesos worker que pueden manejar miles de conexiones cada uno sin bloqueo.

TLS/SSL

NGINX maneja el cifrado TLS (Transport Layer Security), la versión moderna de SSL. En el proyecto Inception, NGINX es el punto de entrada que acepta conexiones HTTPS en el puerto 443, usando TLS 1.2 o TLS 1.3 para cifrar toda la comunicación.

8. PHP-FPM

¿Qué es PHP-FPM?

PHP-FPM (FastCGI Process Manager) es una implementación alternativa de PHP FastCGI. Es un gestor de procesos PHP optimizado para sitios web de alto tráfico. Separa el procesamiento de PHP del servidor web, permitiendo una arquitectura más escalable y eficiente.

Gestión de Procesos

PHP-FPM mantiene un conjunto de procesos PHP workers siempre listos para procesar peticiones. Puede gestionar dinámicamente estos procesos: crear más cuando hay mucha carga y eliminar algunos cuando la demanda disminuye.

Comunicación por Socket

PHP-FPM puede comunicarse a través de sockets Unix o puertos TCP. En el proyecto Inception, WordPress usa PHP-FPM escuchando en el puerto 9000, y Adminer en el puerto 9001. NGINX se conecta a estos puertos para enviarles las peticiones PHP que necesitan procesamiento.

Ventajas sobre mod_php

A diferencia de mod_php que se ejecuta dentro del servidor web, PHP-FPM es un proceso separado. Esto permite: mejor aislamiento, posibilidad de ejecutar diferentes versiones de PHP, reiniciar PHP sin reiniciar el servidor web, y mejor uso de recursos en servidores con múltiples sitios.

9. FASTCGI

¿Qué es FastCGI?

FastCGI es un protocolo para interconectar programas externos con un servidor web. Es una evolución de CGI (Common Gateway Interface) diseñada para mejorar el rendimiento manteniendo procesos persistentes en lugar de crear un nuevo proceso para cada petición.

CGI vs FastCGI

CGI tradicional crea un nuevo proceso por cada petición, lo que consume muchos recursos. FastCGI mantiene procesos persistentes que pueden manejar múltiples peticiones. Esto reduce drásticamente el overhead de crear y destruir procesos.

Comunicación NGINX-PHP

Cuando NGINX recibe una petición para un archivo PHP, no puede ejecutarlo directamente. En su lugar, usa el protocolo FastCGI para comunicarse con PHP-FPM. NGINX envía la petición a través de FastCGI, PHP-FPM procesa el script PHP y devuelve el resultado a NGINX, que finalmente lo envía al cliente.

Parámetros FastCGI

FastCGI transmite información sobre la petición a través de parámetros. Estos incluyen: qué script ejecutar (`SCRIPT_FILENAME`), información del cliente (`REMOTE_ADDR`), datos de la petición HTTP (`REQUEST_METHOD`, `QUERY_STRING`), y más. Estos parámetros permiten que el script PHP tenga todo el contexto necesario para generar la respuesta correcta.

10. MARIADB

¿Qué es MariaDB?

MariaDB es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto. Es un fork de MySQL creado por los desarrolladores originales de MySQL tras su adquisición por Oracle. Es completamente compatible con MySQL en términos de comandos, APIs y protocolos.

Base de Datos Relacional

Una base de datos relacional organiza la información en tablas con filas y columnas. Las tablas pueden relacionarse entre sí mediante claves. MariaDB usa SQL (Structured Query Language) para consultar y manipular estos datos. Es perfecta para WordPress, que necesita almacenar posts, usuarios, comentarios, configuraciones, etc.

ACID

MariaDB garantiza las propiedades ACID: Atomicidad (las transacciones son todo o nada), Consistencia (los datos siempre cumplen las reglas), Aislamiento (las transacciones no interfieren entre sí) y Durabilidad (los datos confirmados persisten incluso ante fallos). Esto asegura la integridad de los datos del blog.

Puerto 3306

MariaDB escucha por defecto en el puerto 3306. En el proyecto Inception, este puerto NO está expuesto al exterior, solo es accesible desde la red interna Docker. WordPress se conecta a 'mariadb:3306' para acceder a la base de datos de forma segura.

11. WORDPRESS

¿Qué es WordPress?

WordPress es un sistema de gestión de contenidos (CMS - Content Management System) de código abierto escrito en PHP. Originalmente diseñado para blogs, ahora potencia más del 40% de todos los sitios web en Internet. Permite crear y gestionar sitios web sin necesidad de programar.

CMS (Content Management System)

Un CMS es un software que permite crear, editar, organizar y publicar contenido digital. WordPress proporciona una interfaz web amigable donde los usuarios pueden escribir posts, subir imágenes, gestionar comentarios, cambiar el diseño del sitio y mucho más, sin tocar código.

Arquitectura WordPress

WordPress está escrito en PHP y requiere una base de datos MySQL o MariaDB. El código PHP de WordPress consulta la base de datos para obtener posts, usuarios, configuraciones, etc., y genera páginas HTML dinámicas. Los archivos de WordPress incluyen: código PHP (core), temas (apariencia visual), plugins (funcionalidades extra) y uploads (imágenes y archivos).

Usuarios y Roles

WordPress tiene un sistema de roles: Administrador (control total), Editor (gestiona contenido), Autor (publica sus propios posts), Colaborador (escribe pero no publica) y Suscriptor (solo lee). En el proyecto Inception se crean dos usuarios: un administrador y un usuario regular.

12. SSL/TLS

¿Qué es SSL/TLS?

SSL (Secure Sockets Layer) y TLS (Transport Layer Security) son protocolos criptográficos que proporcionan comunicaciones seguras en una red. TLS es la versión moderna y más segura de SSL. Permiten cifrar la comunicación entre el navegador del usuario y el servidor web, protegiendo la privacidad y la integridad de los datos.

Cifrado

El cifrado transforma los datos en un formato ilegible durante la transmisión. Solo el destinatario con la clave correcta puede descifrarlos. Esto protege información sensible como contraseñas, datos personales o información de pago de ser interceptada por atacantes.

Certificados

Un certificado SSL/TLS es un archivo que contiene una clave pública y información sobre la identidad del servidor. En producción, estos certificados son emitidos por Autoridades de Certificación (CA) confiables. En el proyecto Inception, se usan certificados autofirmados generados con OpenSSL, válidos para desarrollo pero que generan advertencias en navegadores.

TLS 1.2 y 1.3

TLS 1.2 (2008) y TLS 1.3 (2018) son las versiones actuales del protocolo. TLS 1.0 y 1.1 están obsoletos y tienen vulnerabilidades conocidas. El proyecto Inception configura NGINX para aceptar SOLO TLS 1.2 y 1.3, rechazando versiones antiguas inseguras. TLS 1.3 es más rápido y seguro, eliminando algoritmos de cifrado débiles.

HTTPS

HTTPS (HTTP Secure) es HTTP sobre TLS. Cuando un sitio usa HTTPS, toda la comunicación está cifrada. El candado en la barra de direcciones del navegador indica una conexión HTTPS segura. El puerto estándar de HTTPS es el 443, mientras que HTTP usa el puerto 80.

13. WP-CLI

¿Qué es WP-CLI?

WP-CLI (WordPress Command Line Interface) es una herramienta de línea de comandos para gestionar WordPress. Permite realizar prácticamente cualquier acción que se puede hacer desde el panel de administración web, pero de forma automatizada mediante comandos.

Automatización

WP-CLI es ideal para scripts de instalación y configuración automatizada. En lugar de seguir el asistente de instalación web de WordPress manualmente, se puede usar WP-CLI para descargar WordPress, crear la configuración, instalar la base de datos y crear usuarios automáticamente mediante comandos.

Uso en Inception

En el proyecto Inception, WP-CLI se usa en el script de inicialización del contenedor WordPress. Permite descargar el core de WordPress, configurar la conexión a la base de datos, instalar WordPress con título y administrador, crear el segundo usuario, y todo esto sin intervención manual. Esto hace que el contenedor sea reproducible y automatizado.

14. PROCESOS Y PID 1

¿Qué es un Proceso?

Un proceso es un programa en ejecución. Cada proceso tiene un identificador único llamado PID (Process ID). En un sistema Linux, todos los procesos forman un árbol jerárquico, donde cada proceso tiene un proceso padre (excepto el proceso raíz).

PID 1 en Linux

En un sistema Linux normal, el PID 1 es el proceso init (o systemd en distribuciones modernas). Es el primer proceso que arranca y el ancestro de todos los demás procesos. Si el PID 1 muere, el sistema se apaga.

PID 1 en Docker

En un contenedor Docker, el proceso especificado en CMD o ENTRYPOINT se convierte en el PID 1. Docker monitorea este proceso: si el PID 1 termina, Docker considera que el contenedor ha terminado y lo detiene. Por lo tanto, es crucial que el proceso principal del contenedor se ejecute en primer plano (foreground) y no termine prematuramente.

Daemon vs Foreground

Un daemon es un proceso que se ejecuta en segundo plano (background). En servidores tradicionales, servicios como NGINX se ejecutan como daemons para liberar el terminal. Sin embargo, en Docker, si un servicio se daemoniza, el comando que lo lanzó termina inmediatamente. Docker ve que el PID 1 terminó y mata el contenedor. Por eso, en Docker los servicios deben ejecutarse en foreground (primer plano) usando opciones especiales.

Ejemplos en Inception

NGINX usa 'daemon off' para forzar ejecución en foreground. PHP-FPM usa el flag '-F' (foreground). MariaDB usa 'mariadb' que por defecto corre en foreground. Estas configuraciones aseguran que el servicio sea el PID 1 y el contenedor permanezca activo mientras el servicio está en ejecución.

15. PROXY REVERSO

¿Qué es un Proxy?

Un proxy es un intermediario entre clientes y servidores. Existen dos tipos principales: proxy directo (forward proxy) que representa a los clientes, y proxy reverso (reverse proxy) que representa a los servidores.

Proxy Reverso

Un proxy reverso se sitúa delante de uno o más servidores backend y recibe todas las peticiones de los clientes. Los clientes creen estar comunicándose directamente con el servidor, pero en realidad se comunican con el proxy, que decide a qué servidor backend reenviar cada petición.

NGINX como Proxy Reverso

En el proyecto Inception, NGINX actúa como proxy reverso. Es el único punto de entrada accesible desde Internet (puerto 443). Recibe todas las peticiones HTTPS y, según la ruta solicitada, reenvía la petición al servicio apropiado: WordPress para rutas generales, Adminer para /adminer/, y static-site para /portfolio.

Ventajas del Proxy Reverso

Seguridad: Los servidores backend no están expuestos directamente a Internet.

Balanceo de carga: Puede distribuir peticiones entre múltiples servidores backend.

SSL/TLS Termination: El proxy maneja el cifrado, liberando a los backends de esta tarea.

Caché: Puede cachear respuestas para mejorar el rendimiento.

Compresión: Puede comprimir respuestas antes de enviarlas al cliente.

16. CONCEPTOS DE SEGURIDAD

Principio de Mínimo Privilegio

Este principio establece que un proceso o usuario debe tener únicamente los permisos mínimos necesarios para realizar su función. En Docker, los servicios se ejecutan con el usuario `www-data` (no `root`) siempre que sea posible. Si un atacante compromete el servicio, tiene acceso limitado al sistema.

Variables de Entorno

Las variables de entorno permiten configurar aplicaciones sin hardcodear valores sensibles en el código fuente. En Inception, las contraseñas y configuraciones se almacenan en un archivo `.env` que NO se sube al repositorio Git. Docker inyecta estas variables en los contenedores en tiempo de ejecución.

Aislamiento de Red

Los contenedores en Docker están aislados por defecto. En Inception, MariaDB NO expone su puerto 3306 al exterior, solo es accesible dentro de la red Docker interna. WordPress puede conectarse a MariaDB, pero Internet no puede acceder directamente a la base de datos.

Gestión de Secretos

Los secretos (contraseñas, claves API, certificados) nunca deben almacenarse en el código fuente o en imágenes Docker. El subject del proyecto enfatiza el uso del archivo `.env` para credenciales y sugiere Docker secrets para información confidencial. Esto previene la exposición accidental de credenciales en repositorios públicos.

Actualizaciones y Parches

Mantener el software actualizado es crucial para la seguridad. Usar imágenes base oficiales como Debian y actualizar los paquetes durante la construcción de la imagen ayuda a proteger contra vulnerabilidades conocidas. En producción, es importante reconstruir imágenes periódicamente para incorporar parches de seguridad.

GLOSARIO DE TÉRMINOS

Término	Definición
API	Application Programming Interface - Interfaz para que programas se comuniquen
Backend	Parte del sistema que funciona en el servidor, no visible para el usuario
Bridge	Tipo de red Docker que conecta contenedores en una red privada
CMS	Content Management System - Sistema de gestión de contenidos
Container	Unidad estándar de software que empaqueta código y dependencias
Daemon	Proceso que se ejecuta en segundo plano
DNS	Domain Name System - Sistema que traduce nombres a direcciones IP
Dockerfile	Archivo con instrucciones para construir una imagen Docker
FastCGI	Protocolo para comunicación entre servidor web y aplicaciones
Fork	Copia de un proyecto de software que se desarrolla independientemente
Frontend	Parte del sistema visible e interactiva para el usuario
HTTPS	HTTP Secure - HTTP cifrado con TLS
Image	Plantilla de solo lectura para crear contenedores Docker
Kernel	Núcleo del sistema operativo que gestiona hardware y procesos
Layer	Capa de una imagen Docker, resultado de una instrucción del Dockerfile
LEMP	Linux, NGINX, MySQL/MariaDB, PHP - Stack de tecnologías web
Mount	Vincular un sistema de archivos o directorio a un punto en el árbol de directorios
PHP	PHP: Hypertext Preprocessor - Lenguaje para desarrollo web
PID	Process ID - Identificador único de un proceso
Port	Número que identifica un punto de comunicación en una red
RDBMS	Relational Database Management System - Sistema de BD relacional
SQL	Structured Query Language - Lenguaje para gestionar bases de datos
TLS	Transport Layer Security - Protocolo para comunicación segura
Volume	Mecanismo para persistir datos en Docker
Worker	Proceso que realiza tareas o maneja peticiones
YAML	Yet Another Markup Language - Formato para archivos de configuración

REFERENCIAS Y RECURSOS

Documentación Oficial

Docker Documentation: docs.docker.com

NGINX Documentation: nginx.org/en/docs

MariaDB Knowledge Base: mariadb.com/kb/en

WordPress Codex: codex.wordpress.org

Herramientas

Docker Compose: docs.docker.com/compose

WP-CLI: wp-cli.org

OpenSSL: openssl.org

Recursos de Aprendizaje

Docker Get Started: docs.docker.com/get-started

NGINX Beginner's Guide: nginx.org/en/docs/beginners_guide.html

PHP-FPM Documentation: php.net/manual/en/install.fpm.php