



# PROYECTO DESARROLLO WEB

Ismael Monjas Llorente  
Curso 24/25

## Índice

Planteamiento de una arquitectura de red .....	2
Selección de tecnologías y contenedores .....	2
Configuración de Red .....	2
Instalación y configuración de servidores .....	3
Configuración .....	3
Despliegue.....	5
Comprobaciones .....	5
Documentación del despliegue y control de versiones.....	7
Documentación .....	7
Control de versiones.....	7
Glosario de terminos .....	8

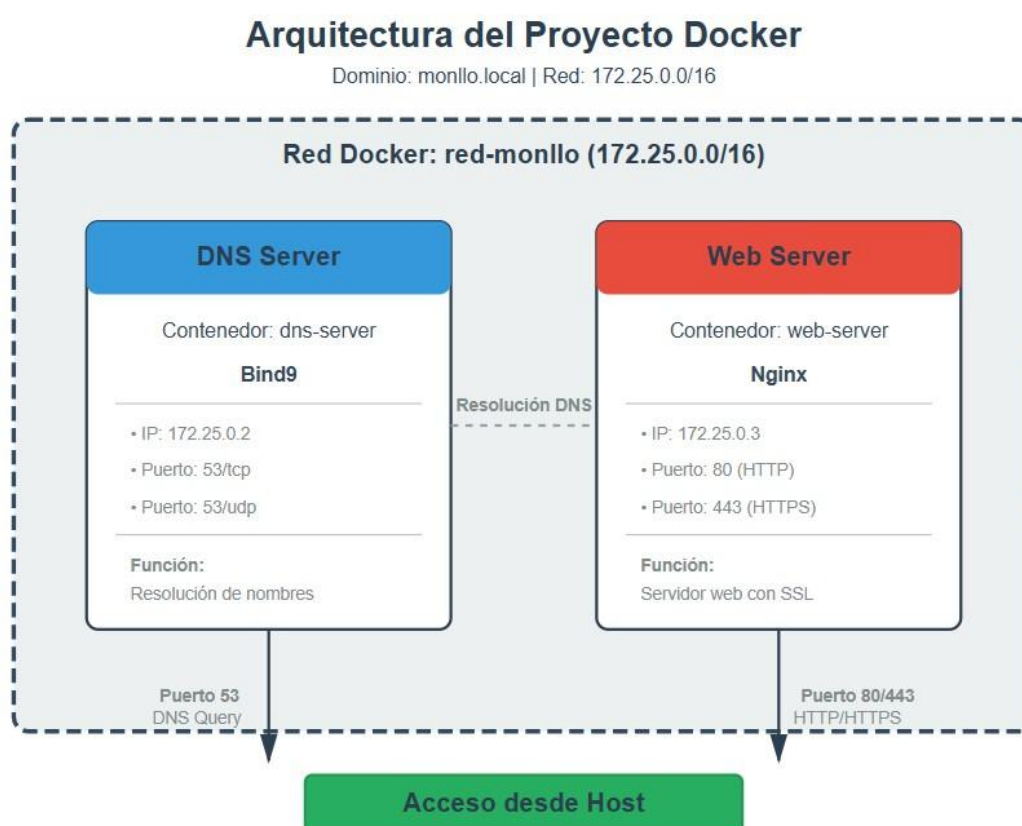
# Planteamiento de una arquitectura de red

## Selección de tecnologías y contenedores

En este proyecto he usado dentro de un Docker Compose, que orquesta varios contenedores, un servidor DNS Bind9, que es un servidor estable y muy utilizado, un servidor web Nginx, que tiene un alto rendimiento y un soporte nativo para SSL/TLS, aportando seguridad a este último con SSL mediante certificados auto firmados generados con OpenSSL.

## Configuración de Red

Se ha creado una red bridge dentro del Docker, que proporciona aislamiento y permite la comunicación interna entre contenedores y da IPs estáticas que facilitan la configuración del DNS. Tiene el siguiente esquema:

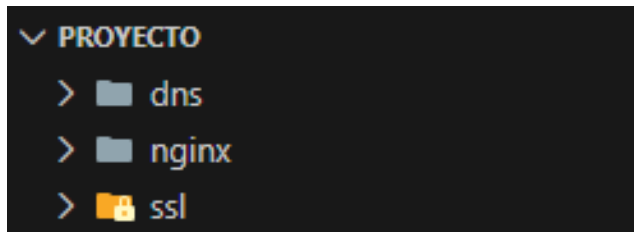


# Instalación y configuración de servidores

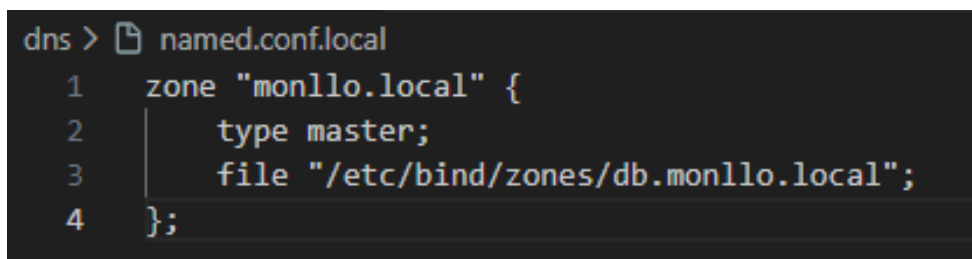
En este apartado se mostrarán los pasos seguidos para la configuración del Docker-compose y los archivos necesarios para los servidores

## Configuración

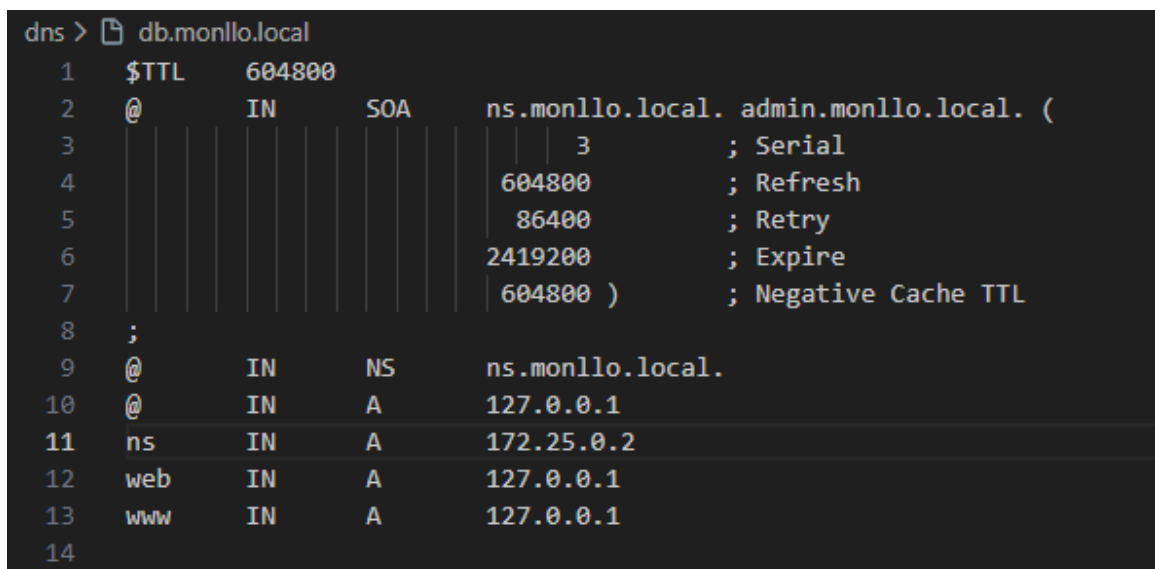
Crear la estructura de carpetas.



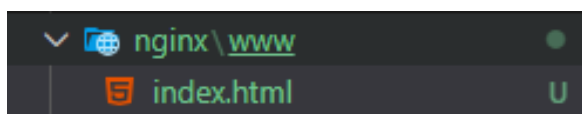
Crear la configuración del DNS (dns/named.conf.local).



Crear la zona DNS (dns/db.monllo.local).



Creemos el HTML.



Crear la configuración de nginx (nginx/nginx.conf).

```
nginx > N nginx.conf
1  server {
2      listen 80;
3      listen 443 ssl;
4      server_name web.monllo.local www.monllo.local monllo.local;
5      ssl_certificate /etc/nginx/ssl/cert.pem;
6      ssl_certificate_key /etc/nginx/ssl/key.pem;
7      ssl_protocols TLSv1.2 TLSv1.3;
8      ssl_ciphers HIGH:!aNULL:!MD5;
9      root /usr/share/nginx/html;
10     index index.html;
11     location / {
12         try_files $uri $uri/ =404;
13     }
14     # Redirigir HTTP a HTTPS (opcional)
15     if ($scheme = http) {
16         return 301 https://$server_name$request_uri;
17     }
18 }
```

Creamos el Docker compose.

```
docker-compose.yml
1  version: '3.8'
2  services:
3      dns:
4          image: ubuntu/bind9
5          container_name: dns-server
6          ports:
7              - "53:53/tcp"
8              - "53:53/udp"
9          volumes:
10             - ./dns/named.conf.local:/etc/bind/named.conf.local
11             - ./dns/db.monllo.local:/etc/bind/zones/db.monllo.local
12          networks:
13             red-monllo:
14                 ipv4_address: 172.25.0.2
15             restart: unless-stopped
16      web:
17          image: nginx:latest
18          container_name: web-server
19          ports:
20              - "80:80"
21              - "443:443"
22          volumes:
23              - ./nginx/www:/usr/share/nginx/html
24              - ./nginx/nginx.conf:/etc/nginx/conf.d/default.conf
25              - ./ssl:/etc/nginx/ssl
26          networks:
27             red-monllo:
28                 ipv4_address: 172.25.0.3
29          depends_on:
30              - dns
31          restart: unless-stopped
32
33  networks:
34      red-monllo:
35          driver: bridge
36          ipam:
37              config:
38                  - subnet: 172.25.0.0/16
```

## Crear certificados SSL.

[illegible][illegible]

## Despliegue

Hacer docker-compose up.

```
PS C:\Users\Ismael\OneDrive - Educacyl\GS DAW\2\DespliegueAW\PROYECTO\PROYECTO> docker-compose -p "proyecto-daw" up -d
time="2025-11-13T21:16:19+01:00" level=warning msg="C:\\Users\\Ismael\\OneDrive - Educacyl\\GS DAW\\2\\DespliegueAW\\PROYECTO\\PROYECTO\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 3/3
  ✓ Network proyecto-daw_red-monllo Created 0.1s
  ✓ Container dns-server Started 1.0s
  ✓ Container web-server Started 1.2s
ps C:\Users\Ismael\OneDrive - Educacyl\GS DAW\2\DespliegueAW\PROYECTO\PROYECTO>
```

## Comprobaciones

Comprobar que los Docker corren bien.

```
PS C:\Users\Ismael\OneDrive - Educacy\GS DAW\2\DespliegueAW\PROYECTO\PROYECTO> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
0fe44bfdd4183	nginx:latest	"/docker-entrypoint. ....	About a minute ago	Up About a minute	0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp
64b6732ec424	ubuntu/bind9	"docker-entrypoint.sh"	About a minute ago	Up About a minute	0.0.0.0:53->53/tcp, 0.0.0.0:53->53/udp, [::]:53->53/tcp, [::]:53->53/udp

Comprobar el dns.

```
PS C:\Users\Ismael\OneDrive - Educacy1\GS DAW\2\DespliegueAW\PROYECTO\PROYECTO> nslookup www.monllo.local
Servidor: localhost
Address: 127.0.0.1

Nombre: www.monllo.local
Address: 127.0.0.1
```

Comprobar en navegador.



Comprobar red interna.

```
PS C:\Users\Ismael\OneDrive - Educacyl\GS DAW\2\DespliegueAW\PROYECTO\PROYECTO> docker network inspect proyecto_red-monllo
[
  {
    "Name": "proyecto_red-monllo",
    "Id": "39bd493dc66a9c1eb0da329a168ffadfee6a273fa89c9efaf1a2d84555de6b76",
    "Created": "2025-11-13T21:28:01.608886971Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.25.0.0/16",
          "Gateway": "172.25.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "1bab7a9f445d214b2c4ed8dc191c41974c97fe391198aeb26caf37ee4eead9a0": {
        "Name": "web-server",
        "EndpointID": "be5bd0343ce398412d504640f48feaf4f4e5310c9e2b4f3e7c3630f565939f33",
        "MacAddress": "e6:e7:7f:a1:02:76",
        "IPv4Address": "172.25.0.3/16",
        "IPv6Address": ""
      },
      "8768ad46ddcdb5ab4b3aaa3078c744e73703eda9b8c31bb801567130ccfd2709": {
        "Name": "dns-server",
        "EndpointID": "6a7fa669beb513d54b40e70a1c9604c158fa79c0f25897838bf2605247f1581c",
        "MacAddress": "ea:06:c6:06:16:b6",
        "IPv4Address": "172.25.0.2/16",
        "IPv6Address": ""
      }
    },
    "Options": {
      "com.docker.network.enable_ipv4": "true",
      "com.docker.network.enable_ipv6": "false"
    },
    "Labels": {
      "com.docker.compose.config-hash": "56f7add902c2e7e62a90771f2a407b3dae8d884cc93254e0978d5e022e4a4828",
      "com.docker.compose.network": "red-monllo",
      "com.docker.compose.project": "proyecto",
      "com.docker.compose.version": "2.40.3"
    }
  }
]
```

# Documentación del despliegue y control de versiones

## Documentación

La documentación técnica se podrá encontrar entre los archivos adjuntados del trabajo, en el documento "DOCUMENTACION.txt".

## Control de versiones

En este proyecto ha sido versionado mediante git y subido a GitHub. En el siguiente enlace se podrá acceder al repositorio donde se aloja y se podrá comprobar todos los commits realizados durante el desarrollo del mismo.

<https://github.com/ismaelmonllo-edu/proyecto-dockermonllo>



## Glosario de términos

**Bind9 (Berkeley Internet Name Domain):** Servidor DNS de código abierto más utilizado en Internet. Permite resolver nombres de dominio a direcciones IP y es compatible con los estándares RFC de DNS.

**Bridge (Red Bridge):** Tipo de red virtual en Docker que conecta contenedores en una red privada aislada, permitiendo la comunicación entre ellos mientras están separados de la red del host.

**Certificado Autofirmado:** Certificado SSL/TLS que no ha sido emitido por una Autoridad Certificadora (CA) reconocida, sino generado localmente. Apropiado para entornos de desarrollo y testing.

**Contenedor:** Unidad estándar de software que empaqueta código y todas sus dependencias para que la aplicación se ejecute de manera rápida y confiable en diferentes entornos informáticos.

**Docker:** Plataforma de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando portabilidad y eficiencia.

**Docker Compose:** Herramienta para definir y ejecutar aplicaciones Docker multi-contenedor mediante un archivo YAML de configuración, simplificando la orquestación de servicios.

**DNS (Domain Name System):** Sistema de nomenclatura jerárquico que traduce nombres de dominio legibles por humanos (como monllo.local) en direcciones IP que las computadoras pueden entender.

**Dominio .local:** Sufijo de dominio utilizado para redes locales privadas que no están expuestas a Internet, evitando conflictos con dominios públicos.

**HTTPS (HyperText Transfer Protocol Secure):** Versión segura del protocolo HTTP que utiliza cifrado SSL/TLS para proteger las comunicaciones entre el navegador y el servidor web.

**IP Estática:** Dirección IP que se asigna manualmente a un dispositivo y permanece constante, a diferencia de las IPs dinámicas que cambian con cada conexión.

**Nginx:** Servidor web de alto rendimiento y código abierto que también funciona como proxy inverso, balanceador de carga y caché HTTP.

**OpenSSL:** Biblioteca de software de código abierto que implementa los protocolos SSL y TLS, utilizada para generar certificados y claves criptográficas.

**Orquestación:** Proceso de automatizar la configuración, coordinación y gestión de sistemas informáticos y servicios, especialmente contenedores.

**Puerto:** Punto de conexión lógico en un sistema operativo que permite que los servicios de red escuchen o envíen información. Por ejemplo, puerto 80 para HTTP y 443 para HTTPS.

**Red Privada:** Red de computadoras que utiliza rangos de direcciones IP privadas (como 172.25.0.0/16) no enrutables en Internet, proporcionando aislamiento y seguridad.

**SSL/TLS (Secure Sockets Layer / Transport Layer Security):** Protocolos criptográficos que proporcionan comunicaciones seguras a través de redes informáticas, cifrando los datos transmitidos.

**Subnet (Subred):** División lógica de una red IP. La notación /16 indica el tamaño de la red (en este caso, 65,534 direcciones disponibles).

**Volumen (Docker):** Mecanismo de persistencia de datos en Docker que permite almacenar información fuera del ciclo de vida del contenedor, facilitando el acceso y la modificación de archivos de configuración.

**YAML (YAML Ain't Markup Language):** Formato de serialización de datos legible por humanos, comúnmente utilizado para archivos de configuración como docker-compose.yml.

**Zona DNS:** Porción del espacio de nombres DNS administrada por una organización o entidad específica, conteniendo los registros DNS para ese dominio.