



MANUAL DE IMPLANTACIÓN

APLICACIÓN WEB CON BACKEND Y
FRONTEND UTILIZANDO AWS-EBS

Elisa Córdoba

2º ASIR

INDICE

1. Entorno local.....	3
Requisitos necesarios.....	3
Arquitectura final.....	3
Árbol del proyecto.....	4
Software instalado.....	4
1.1 Pasos de implantación.....	4
1.1.1 Base de datos MySQL.....	4
1.1.2 Implantación del Backend.....	4
1.1.3 Implantación del Frontend	
Fichero Docker-compose.yml.....	5
Resumen.....	5
2. Entorno AWS-EBS.....	6
Requisitos necesarios.....	6
Arquitectura final.....	6
Software instalado.....	7
2.1 Base de Datos (RDS).....	7
2.2 Backend.....	8
2.3 Frontend.....	9

En este manual de implantación se mostrarán todos los requisitos y pasos necesarios para implementar correctamente la aplicación de Apartamentos4v sobre reservas de habitaciones.

Las características de la arquitectura son:

- Entorno EC2 o RDS para MySQL
- Entorno EBS para el Backend (Web PHP con tecnología Symfony, que enganchará a la base de datos)
- Entorno EBS (Web estática realizada con tecnología Angular js)

Inicialmente, se realiza de modo local y posteriormente en AWS-EBS.

1.- Entorno local

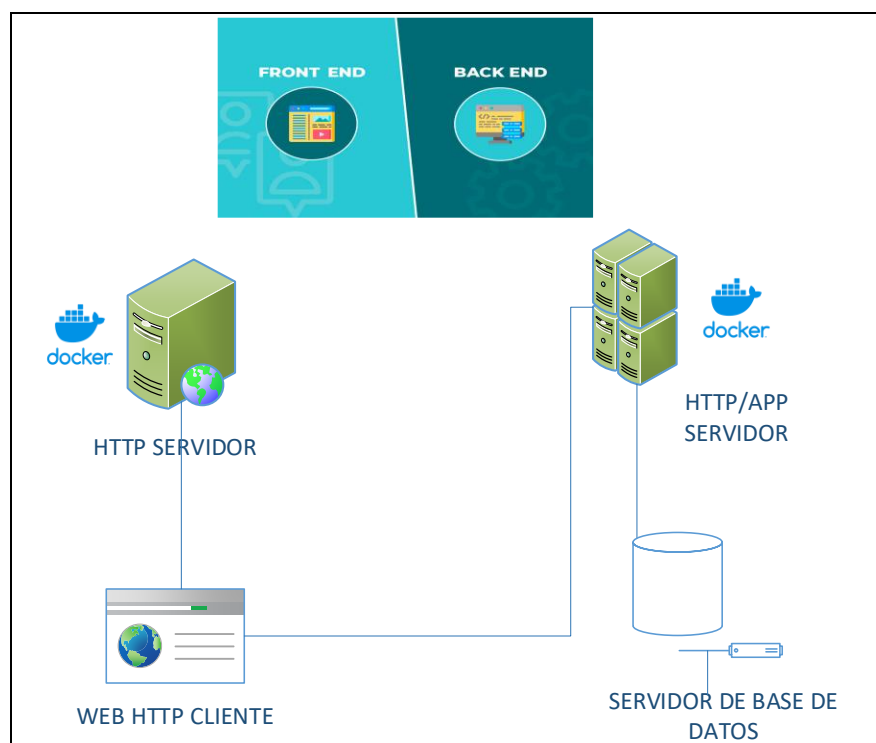
En este apartado se desarrolla la implantación local de la aplicación.

Requisitos necesarios

- XAMPP
- Visual Studio Code con wsl 2
- Docker Compose

Arquitectura final

La figura muestra la arquitectura de la aplicación



Los componentes son:

- Backend: Contenedor que contiene el software basado en Symfony se conecta a la base de datos MySQL
- Frontend: Contiene la página Web que se muestra a los clientes, los datos son obtenidos a través del backend mediante una solicitud web interna
- Base de datos: MySQL, usada por el Backend

Árbol del proyecto:

```
/project-root  
  /backend (Contenedor para el backend con Apache y PHP)  
  /frontend (Contenedor para el frontend con AngularJS)  
  /src  
  /dist (Generado después de la construcción)  
  /package.json  
  Dockerfile.backend  
  Dockerfile.frontend  
  docker-compose.yml (Docker Compose)
```

Software Instalado

- Apache2.8
- PHP 8.3 + Extensiones
- Docker Compose

1.1 Pasos de Implantación

1.1.1. Base de datos MySQL

1. Se ejecuta el programa XAMPP
2. Se pulsa sobre Start para iniciar el módulo Apache.
3. Se pulsa sobre Start para iniciar la base de datos MySQL
4. Se pulsa sobre Config para cambiar el puerto de MySQL.
5. Se pulsa sobre Admin para acceder a la página de phpAdmin.
6. Se generan la base de datos y las tablas necesarias.

1.1.2. Implantación del Backend

- El Dockerfile configura el backend con PHP 8.3, Apache, y Symfony.
- El archivo `docker-compose.yml` configura tanto el backend como la base de datos MySQL y establece las conexiones entre los servicios.
- Para acceder a la aplicación Symfony a través de <http://localhost:8080>.

1.1.3. Implantación del Frontend

- La imagen `node:18` construye la aplicación AngularJS.
- Se copian los archivos `package.json` y `package-lock.json` para instalar las dependencias de AngularJS.
- Se ejecuta `npm run build --prod` para generar los archivos estáticos optimizados de la aplicación AngularJS en la carpeta `dist/`.
- Se activa el módulo `rewrite` de Apache para que las URLs limpias funcionen correctamente.
- Se copian los archivos generados por AngularJS (en `dist/`) al directorio adecuado de Apache (`/var/www/html/`), donde Apache servirá los archivos estáticos.

Fichero `Docker-compose.yml`

1. Backend:
 - Servicio backend: El `Dockerfile.backend` construye el contenedor para Symfony. Exponemos el puerto 8080 del contenedor en el puerto 80 de la máquina local para acceder a Symfony.
 - Redes y entorno: Se configuran las variables de entorno necesarias para que Symfony se conecte a la base de datos MySQL.
 2. Frontend:
 - Servicio frontend: El `Dockerfile.frontend` construye el contenedor para el frontend con AngularJS. El puerto 80 en el contenedor, hace que el frontend esté disponible en `http://localhost`.
 - Redes: El frontend y el backend comparten la misma red (`symfony_network`) para que puedan comunicarse si es necesario.
 3. Base de datos (MySQL):
 - Servicio db: Usamos la imagen oficial de `mysql:8.0` para crear el contenedor de MySQL. Se exponen los puertos necesarios para conectarse a la base de datos desde otros servicios.
 4. Redes: Ambos servicios (`frontend` y `backend`) están conectados a la misma red de Docker (`symfony_network`) para que puedan comunicarse.
- `docker-compose up --build`: levantará los contenedores para el backend (Symfony), el frontend (AngularJS) y la base de datos (MySQL).

Resumen:

Frontend (AngularJS): Se construye y sirve con Apache en un contenedor Docker.
Backend (Symfony): Se ejecuta con PHP 8.3 y Apache en otro contenedor Docker.
Ambos servicios están configurados para usar una base de datos MySQL.
Docker Compose se utiliza para gestionar ambos servicios y la base de datos en un solo archivo.

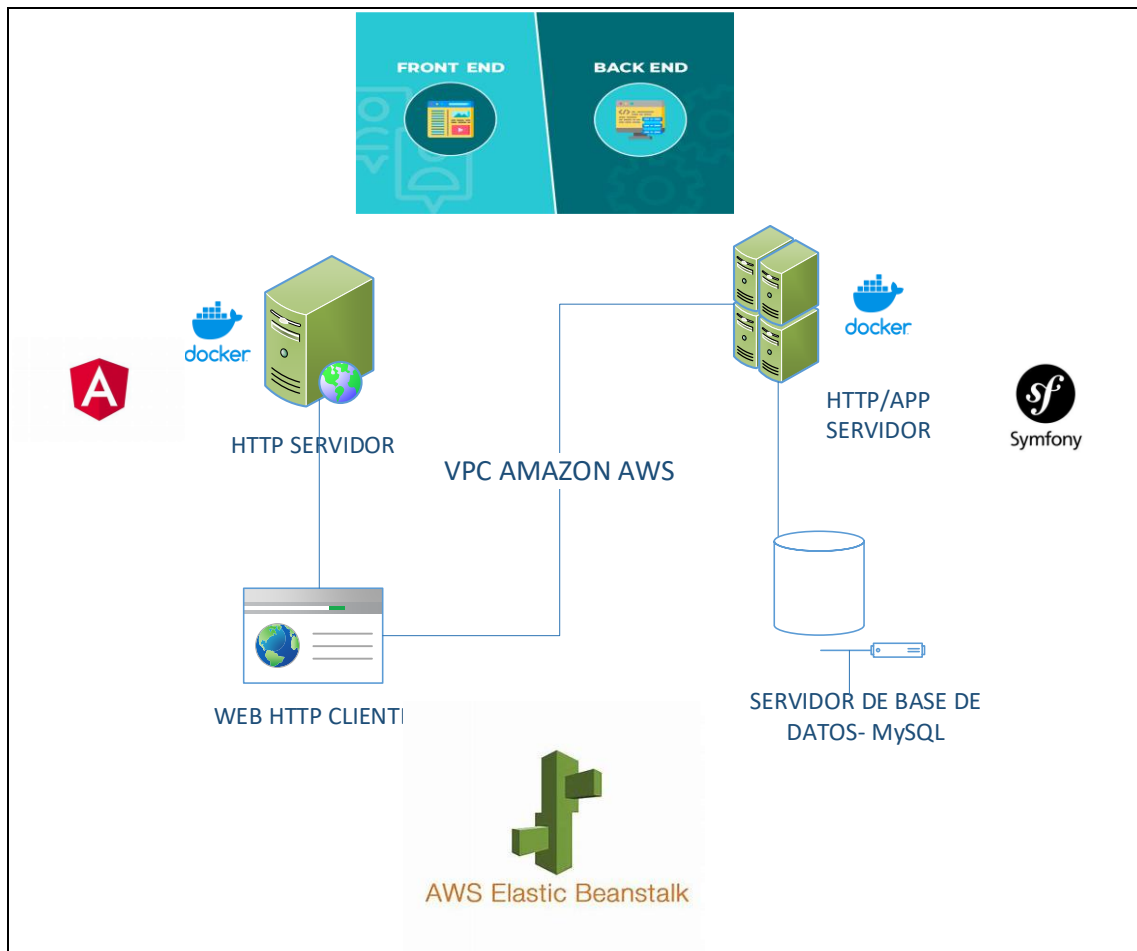
2. Entorno AWS-EBS

En este apartado se muestra la aplicación en aws-EBS

Requisitos necesarios

1. Una cuenta de Amazon AWS permita el acceso a los servicios de AWS (RDS, Elastic BeanStalk y EC2)
2. Gestor de bases de datos (HeidiSQL o parecido)

Arquitectura Final



Servicio RDS: Instancia de Base de datos MySQL, usada por el Backend

Backend: Contenedor que contiene el software basado en Symfony, esta aislado del exterior mediante una red virtual privada, VPC, y se conecta a la base de datos MySQL, generada mediante el servicio RDS.

Frontend: Contenedor que contiene la página Web que se muestra en la parte cliente de la aplicación, los datos son obtenidos a través del Backend.

VPC Amazon AWS: Red interna generada para la comunicación segura en la aplicación.

Software Instalado

El software que va instalado en los contenedores es el siguiente:

- Apache2
- PHP 8.3 + Extensiones
- Docker Compose

2.1. Base de datos (Servicio RDS)

Acciones que hay que realizar para desplegar en el servicio RDS la instancia de MySQL.

Paso 1: Se accede al servicio RDS desde la consola de AWS

Paso 2: Se pulsa en Crear base de datos

Paso 3: Se selecciona el modo Creación sencilla.

Paso 4: Elegimos MySQL, con la capa gratuita.

Paso 5: En el identificador, ponemos el nombre de la base de datos y usuario maestro.

Paso 6: En las credenciales, se elige Autoadministrado y se asigna la contraseña de administrador.

Paso 7: Miramos como queda la configuración de la base de datos generada.

Paso 8: Pulsamos en Crear base de datos.

Paso 9: Se modifica el acceso a la base de datos generada, haciéndolo público.

Paso 10: Pulsamos sobre el nombre de la base de datos, y vamos a Conectividad y Seguridad.

Paso 11: Ponemos las reglas de seguridad pulsando en Crear grupo de seguridad, pulsando sobre el ID de grupo de seguridad.

Paso 12: Se pulsa la regla que permite el acceso a todos los puertos e IPs.

Paso 13: Para probar el entorno generado, se utiliza HeidiSQL y se elige la opción MySQL on RDS.

Paso 14: Se copia el punto de enlace de Conectividad y Seguridad en el apartado de Nombre de host/IP.

Paso 15: Si se produce un error de conexión, es porque no se han guardado las reglas establecidas.

Paso 16: Guardadas las reglas, se consigue una conexión, pero con una base de datos vacía.

Paso 17: Mediante HeidiSQL, se crea la base de datos y las tablas utilizando SQL.

Archivo → BBDD_MySQL.pdf

2.2. Backend

Se edita en el Dockerfile la línea: env.

```
DATABASE_URL=mysql://username:password@dbHost:3306/databaseName?serverVersion=10.6.18-MySQL&charset=utf8mb4
```

Se reemplazan los siguientes valores:

- username: Usuario creado en RDS
- password: Contraseña establecida en RDS
- dhHost: Direccion DNS asignada en AWS (automático)
- databaseName: Nombre de la base de datos a usar

Acciones que hay que realizar para desplegar en el servicio Elastic Beanstalk el Backend.

Paso 1: Se accede al servicio Elastic Beanstalk desde la consola de AWS

Paso 2: Se pulsa en la parte izquierda en Aplicaciones y luego en “Crear Aplicacion”

Paso 3: Le damos un nombre a la aplicación y le damos a crea

Paso 4: Una vez creada, aparece en la página de Aplicaciones.

Paso 5: A continuación, se muestra la página del entorno y se pulsa para crear nuevo entorno.

Paso 6: En la configuración se elige Entorno de servidor Web, el nombre de la aplicación sale por defecto.

Paso 7: Ponemos un nombre de dominio, y verificamos si está disponible.

Paso 8: En el tipo de plataforma elegimos Docker, estando administrada.

Paso 9: Se elige nombre de dominio, que tiene que estar disponible

Paso 10: Cargamos el código de la aplicación con una etiqueta de versión de la carpeta Backend sin poner el fichero docker-compose.yml.

Paso 11: Se seleccionan unos roles determinados.

Paso 12; Se elige una VPC y una subred de las disponibles.

Paso 13: Ponemos el informe del estado en básico.

Paso 14: Las actualizaciones administradas por la plataforma deben estar desactivadas.

Paso 15: Finalmente, pulsamos Enviar.

Archivo→ Backend.pdf

2.3. Frontend

Se edita en el Dockerfile la línea:

```
RUN sed -i 's/<backend_domain>/backend_domain_here/g' /var/www/html/main-ZKT6JCA3.js.
```

Reemplazamos el valor backend_domain_here por el nombre del dominio generado por Elastic Beanstalk en el Backend.

Acciones que hay que realizar para desplegar en el servicio Elastic Beanstalk el Backend.

Pasos para desplegar en el servicio Elastic Beanstalk el Frontend.

Paso 1: Se accede al servicio Elastic Beanstalk desde la consola de AWS

Paso 2: Se pulsa en la parte izquierda en Aplicaciones y luego en “Crear Aplicación”

Paso 3: Le damos un nombre a la aplicación y le damos a crear

Paso 4: Una vez creada, aparece en la página de Aplicaciones.

Paso 5: A continuación, se muestra la página del entorno y se pulsa para crear nuevo entorno.

Paso 6: En la configuración se elige Entorno de servidor Web, el nombre de la aplicación sale por defecto.

Paso 7: Ponemos un nombre de dominio, y verificamos si está disponible.

Paso 8: En el tipo de plataforma elegimos Docker, estando administrada.

Paso 9: Se elige nombre de dominio, que tiene que estar disponible

Paso 10. Cargamos el código de la aplicación con una etiqueta de versión de la carpeta frontend. No se pone el fichero docker-compose.yml

Paso 11: Se seleccionan unos roles determinados

Paso 12: Elegimos la misma VPC y subred que se ha elegido en el Backend

Paso 13: Activamos la dirección IP pública para acceder a la página Web.

Paso 14: Ponemos el informe del estado en básico.

Paso 15: Las actualizaciones administradas por la plataforma deben estar desactivadas.

Paso 16: Finalmente, pulsamos Enviar.

Archivo→ Frontend.pdf