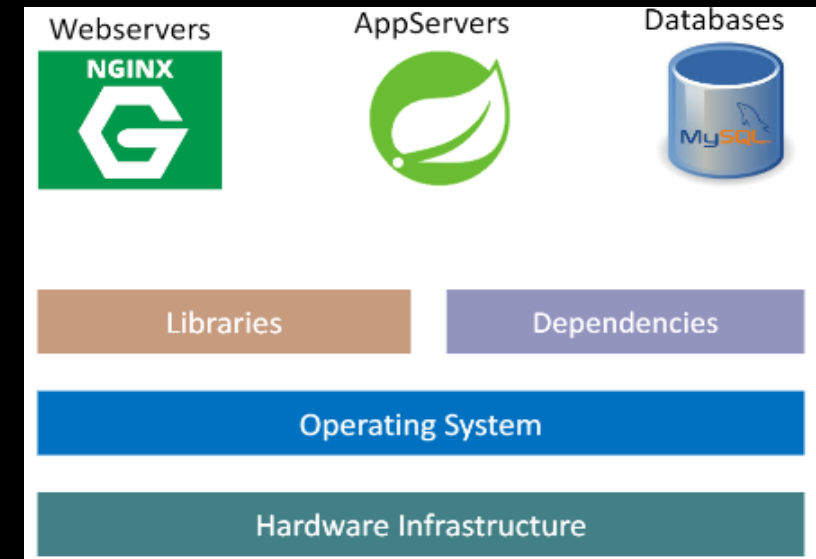


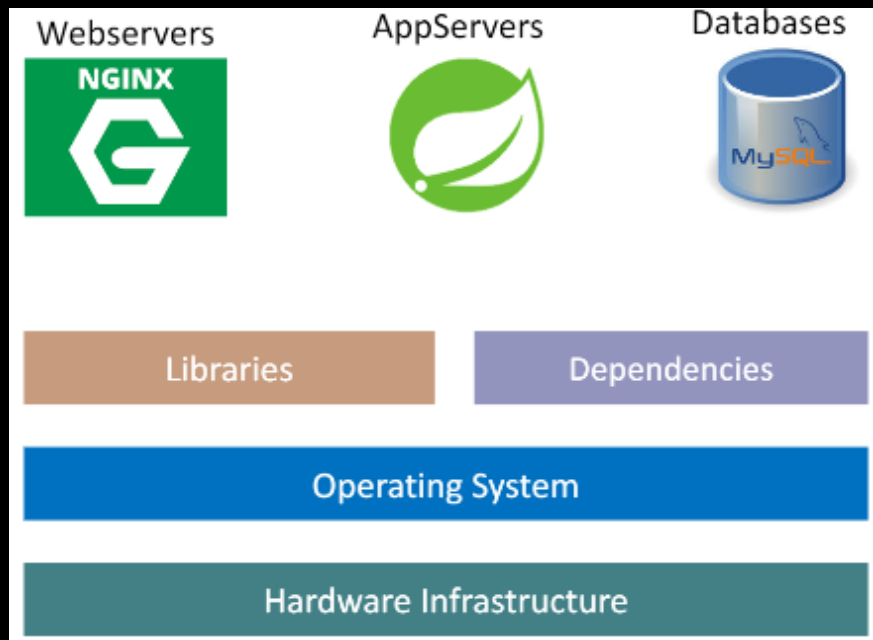
# Problemas con la infra tradicional?

- Instalar y Configurar
  - Consume mucho tiempo.
  - Se tiene que repetir el proceso en cada ambiente.
- Compatibilidad y Dependencia
  - El sistema es más propenso a errores por librerías y dependencias.
- Inconsistencia entre ambientes
  - Es complicado mantener todos los ambientes iguales
- Necesitas más soporte técnico
  - Soporte tanto de software como hardware.



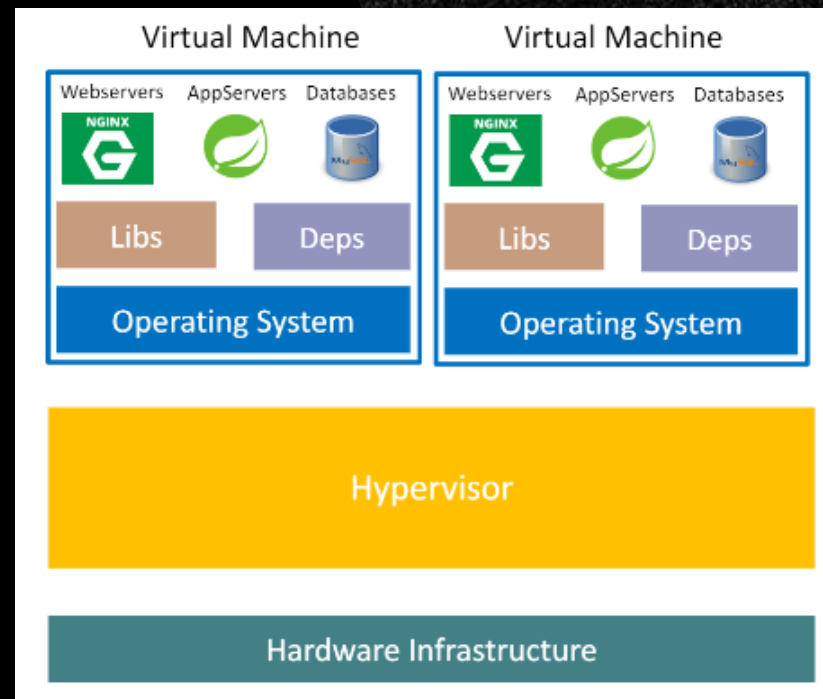
## Máquinas Físicas

- Configurar todo el hardware
- Configurar el SO y todo lo necesario para el proyecto



## Máquinas Virtuales

- Nos olvidamos del hardware
- Solo configuramos el SO y todo lo necesario para el proyecto



# Que es Docker?

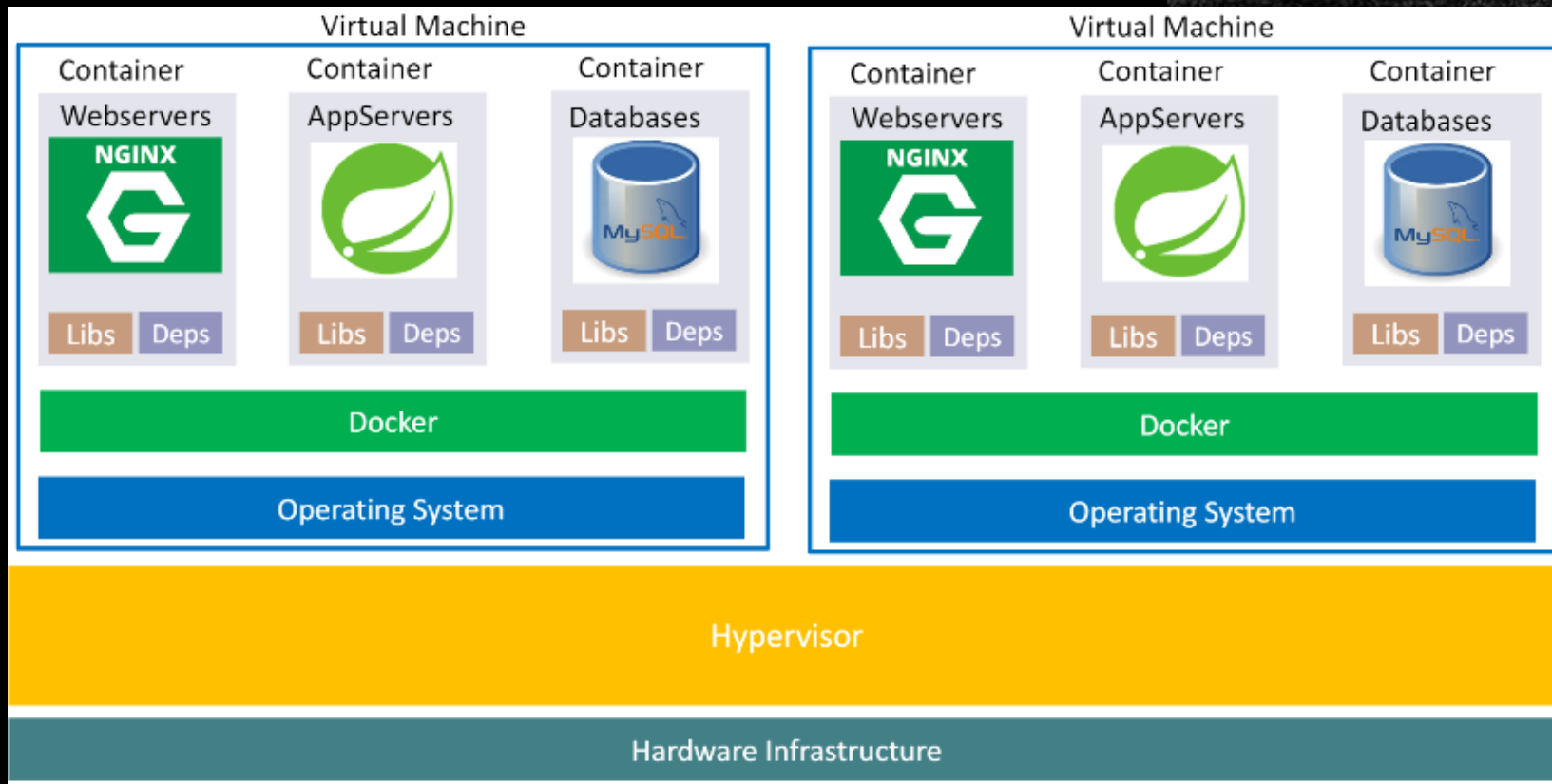
- Es una plataforma que permite empaquetar un aplicación en un contenedor.
- Un contenedor es un entorno totalmente aislado, no tiene conocimiento de tu sistema operativo o tus archivos.
- Dentro del contenedor se puede agregar todo lo necesario para ejecutar una aplicación como: SO, Bibliotecas, Herramientas y el código fuente.

# ***Cual es la diferencia entre Docker y una VM?***

Las máquinas virtuales (VM) virtualizan el hardware del servidor, mientras que los contenedores virtualizan el sistema operativo de un sistema. Docker es un sistema operativo (o runtime) para contenedores.

# Máquinas virtuales con Docker

- Nos olvidamos del hardware y SO
- Solo configuramos todo lo necesario para el proyecto





# ***Terminología***

- Imagen de docker
  - Es una plantilla con instrucciones para crear un container.
  - Una imagen puede estar basada en otra imagen facilitando la reutilización, por ejemplo: si tengo una app en ASPNET simplemente puedo usar una imagen de ASPNET creada por el equipo de Microsoft con todas las dependencias necesarias y yo solo tendría que configurar mi app.
- Contenedor de docker
  - Es una instancia de mi app creada a partir de una imagen.
  - Se puede crear, iniciar, parar mover o eliminar
  - También podemos ingresar al contenedor

# ***Ventajas de usar Docker***

- **Flexible:** Aplicaciones muy complejas pueden ser dockerizadas.
- **Portable:** Se puede construir en local y desplegar en la nube fácilmente.
- **Bajo acoplamiento:** Un container es un entorno totalmente aislado, no le interesa el sistema operativo en donde este hosteado.
- **Escalable:** Los contenedores se pueden incrementar y distribuir fácilmente.
- **Seguro:** Los contenedores aplican restricciones y aislamientos a los procesos sin que se requiera ninguna configuración por parte del usuario.