



# Introducción al Despliegue Web

2 DAW

Profesora: Consuelo Moya

# INTRODUCCIÓN

- Para desplegar una aplicación web vamos a necesitar una **infraestructura** que nos permita alojarla y hacerla accesible a los usuarios a través de Internet.





# OPCIONES COMUNES DE DESPLIEGUE

- 1.1 1. Hosting Compartido
- 1.2 2. VPS (*Virtual Private Server*)
- 1.3 3. Servidor Dedicado
- 1.4 4. Cloud Computing
  - 1.4.1 4.1 IaaS (*Infrastructure as a Service*)
  - 1.4.2 4.2 PaaS (*Platform as a Service*)
  - 1.4.3 4.4 FaaS (*Function as a Service*) o Serverless
  - 1.4.4 4.5 CaaS (*Containers as a Service*)
  - 1.4.5 4.6 KaaS (*Kubernetes as a Service*)
- 1.5 5. Hosting estático con CDNs (*Content Delivery Network*)

# HOSTING COMPARTIDO



- En un hosting compartido, las aplicaciones web comparten los recursos del servidor (CPU, RAM, espacio de disco, etc.) con otros sitios web que están alojados en el mismo servidor físico.
- **Ventajas:**
  - Es la opción más económica.
  - Fácil de gestionar a través de paneles de control como [Plesk](#) o [cPanel](#).
  - Es ideal para principiantes o proyectos pequeños.
- **Desventajas:**
  - Pueden existir problemas de rendimiento, debido a que se están compartiendo los recursos del servidor con otros sitios web.
  - Tienes poco control sobre la configuración del servidor y lo que puedes instalar en él.
- **Ejemplos de proveedores:**
  - [Hostinger](#)
  - [Arsys](#)
  - [IONOS](#)
  - [Cdmon](#)
  - [Hetzner](#)

# VPS ( VIRTUAL PRIVATE SERVER )

- En este caso, el servidor físico se divide en varios servidores virtuales o *Virtual Private Server* (VPS), donde cada uno tiene asignados sus propios recursos.
- Se trata de una opción intermedia entre el hosting compartido y un servidor dedicado.
- **Ventajas:**
  - Tienes más control sobre la configuración del servidor y lo que puedes instalar en él.
  - Puede ofrecer un mejor rendimiento que el hosting compartido.
- **Desventajas:**
  - El coste es mayor que el del hosting compartido.
  - Necesitas tener algunos conocimientos técnicos básicos sobre cómo gestionar un servidor.
- **Ejemplos de proveedores:**
  - [DigitalOcean](#)
  - [OVH](#)
  - Algunos proveedores de hosting compartido también ofrecen planes de VPS.

# SERVIDOR DEDICADO

- En un servidor dedicado, tienes un servidor físico completo donde sólo se alojará tu aplicación web.
- **Ventajas:**
  - Tendremos mejor rendimiento ya que no compartimos los recursos con otros sitios web.
  - Ideal para aplicaciones grandes, con mucho tráfico.
- **Desventajas:**
  - Es una opción más cara que el hosting compartido y un VPS.
  - Necesitas conocimientos técnicos avanzados para administrar el servidor (seguridad, actualizaciones, etc.).
- **Ejemplos de proveedores:**
  - [Hetzner](#)
  - Algunos proveedores de VPS también ofrecen planes de servidores dedicados.

# Cloud Computing

- Los servicios de Cloud Computing que podemos utilizar para desplegar aplicaciones web son los siguientes:
- **IaaS** (*Infrastructure as a Service*).
- **PaaS** (*Platform as a Service*).
- **FaaS** (*Function as a Service*) o Serverless.
- **CaaS** (*Containers as a Service*).
- **KaaS** (*Kubernetes as a Service*).

# IaaS (*Infrastructure as a Service*)

- Proporciona recursos de infraestructura como servidores virtuales, redes y almacenamiento. En este servicio el usuario se encarga de gestionar todo lo demás, desde la configuración del sistema operativo hasta el software de la aplicación que quiera desplegar.
- Esta opción es más flexible que PaaS, pero también más compleja de gestionar.
- **Ventajas:**
  - Ofrece más flexibilidad y control al usuario para configurar el entorno según las necesidades específicas de su aplicación.
  - Facilita la escalabilidad ya que puedes aumentar o reducir la infraestructura según la demanda.
- **Desventajas:**
  - Requiere conocimientos técnicos avanzados.
  - El coste depende de los recursos que se usen. Puede ser más caro que otras opciones.
- **Ejemplos de proveedores:**
  - [Amazon Web Services \(AWS\)](#)
  - [Microsoft Azure](#)
  - [Google Cloud Platform \(GCP\)](#)
  - [Oracle Cloud Infrastructure \(OCI\)](#)



# PaaS (*Platform as a Service*)

- Estas plataformas gestionan la infraestructura y se enfocan en simplificar el despliegue de las aplicaciones. El usuario sólo se tiene que preocupar de la aplicación y la plataforma se encarga del resto (servidores, escalabilidad, mantenimiento).
- **Ventajas:**
  - Es fácil de usar. Ideal para desarrolladores que no quieren complicarse con la administración de servidores.
  - Permite el autoescalado en función de la demanda.
- **Desventajas:**
  - Tienes menos control ya que no tienes acceso al servidor donde se ejecuta la aplicación.
  - El coste depende de los recursos que se usen. Puede ser más caro que otras opciones.
- **Ejemplos de proveedores:**
  - [Vercel](#)
  - [Heroku](#)
  - [AWS Elastic Beanstalk](#)
  - [Google App Engine](#)
  - [Netlify](#)
  - [Platform.sh](#)
  - [Firebase](#)

# FaaS (*Function as a Service*) o Serverless

- Este tipo de servicio no se utiliza para desplegar aplicaciones web completas, sino que se utiliza para **desplegar funciones** que responden a eventos o peticiones HTTP. El usuario no gestiona ningún servidor, es el proveedor el encargado de ejecutar las funciones en su infraestructura.
- **Ventajas:**
  - Sólo pagas por el tiempo que tu código está activo.
  - Ideal para aplicaciones que tienen cargas de trabajo intermitentes.
  - No necesitas gestionar servidores.
- **Desventajas:**
  - Tienes menos control sobre el entorno de ejecución.
  - Sólo es útil para tareas pequeñas, no para aplicaciones completas.
- **Ejemplos de proveedores:**
  - [AWS Lambda](#)
  - [Cloud Run Functions](#)
  - [Azure Functions](#)

# CaaS (Containers as a Service)

- Las tecnologías de contenedores, como [Docker](#), permiten empaquetar aplicaciones y sus dependencias para garantizar que se pueda ejecutar de la misma manera en cualquier entorno.
- **Ventajas:**
  - Facilitan la portabilidad de las aplicaciones.
  - Es ideal para arquitecturas basadas en microservicios.
- **Desventajas:**
  - Requiere conocimientos técnicos avanzados.
- **Ejemplos de proveedores:**
  - [AWS Elastic Container Services \(ECS\)](#)
  - [Azure Container Apps](#)

# KaaS (*Kubernetes as a Service*)

- Kubernetes es una plataforma para orquestar y gestionar contenedores a gran escala.
- **Ventajas:**
  - Kubernetes facilita la gestión de aplicaciones distribuidas y su escalado.
- **Desventajas:**
  - Requiere conocimientos técnicos avanzados.
- **Ejemplos de proveedores:**
  - [AWS Elastic Kubernetes Service \(EKS\)](#)
  - [Google Kubernetes Engine \(GKE\)](#)
  - [Azure Kubernetes Service \(AKS\)](#)

# Hosting estático con CDNs (*Content Delivery Network*)

- Para desplegar aplicaciones web estáticas (HTML, CSS y JavaScript) podemos utilizar una CDN (*Content Delivery Network*) para distribuir el contenido en servidores que están repartidos por todo el mundo.
- **Ventajas:**
  - Muy eficiente para sitios web estáticos.
  - Existen proveedores gratuitos y muy económicos.
  - Mejora la velocidad de carga de los sitios web estáticos.
- **Desventajas:**
  - Está limitado para contenido de sitios web estáticos.
- **Ejemplos de proveedores:**
  - [GitHub Pages](#)
  - [Cloudflare Pages](#)
  - [Surge.sh](#)