

## **::: Notas de Localhost a Producción :::**

**::: By Freddy Alcarazo :::**

**<https://github.com/alcarazolabs>**

(19 de julio del 2021)

Proveedores de dominio y hosting:

- <https://www.namecheap.com/>
- <https://es.siteground.com/> mejor soporte
- Firebase (sube los precios cuando suben las conexiones)
- Google domains
- <https://www.ionos.com/>
- <https://www.cloudflare.com/> (fácil admin de dns, ssl flexible, droplets)
- Goodaddy no, por que cobra por ssl y subasta dominios.

Otros:

- ✓ Se recomienda encriptación SSL 2048.
- ✓ <https://improvmx.com/> (Sirve para redireccionar correos a Gmail usando los dns del dominio).. en Gmail se debe de configurar para responder a nombre de un dominio x.
- ✓ Las desventajas de VPS no tienen CPanel.
- ✓ Hostgator no sirve para apps de alto trafico pero es barato.
- ✓ <https://www.digitalocean.com/> buen proveedor de VPS, ofrecer cluster para cloud.
- ✓ No dejar puertos abiertos en hosting (habilitar firewall)
- ✓ <https://www.ovh.com/> vps
- ✓ Heroku ideal para node.js
- ✓ <https://aws.amazon.com/es/rds/>

Servidores CLOUD:

- ✓ Se necesita configurar mucho kubernetes, dockers.

## Basic Components

Domain + SSL (DNS Management)

Hosting / VPS / Cloud

Repo (?)

FTP / SFTP / SSH

CI / CD

- No comprar hostings y dominios para clientes a nombre propio. Debe de quedar consiente de que el cliente será el dueño.
- En el administrador de dominio en un registro tipo A apuntar la ip de hosting.
- Evitar poner BD's en docker (se puede) pero no se recomienda.
- VPS y CLOUD soportan SSH
- La mayoría de hostings ofrecen SFTP, el cual utiliza ssh por debajo.

### Sistema de integración continua con git deploy:

<https://es.wikipedia.org/wiki/CI/CD>

- Usar **git deploy php** para subir archivos via ftp al servidor (sirve para subir archivos modificados y no todos cuando se modifica algo, evita que se corrumpan los archivos mientras se suben). Requiere que php se ejecute en la máquina, cuando escribimos php en la consola funcione.

- github actions
- Google built
- Gitlab

<https://github.com/alcarazolabs/git-deploy-php>

### Fabric Python:

<https://code.tutsplus.com/es/tutorials/getting-started-with-the-fabric-python-library--cms-30555>

- Linux alpine versión de Linux muy pequeña.
- yalm vs json.. yalm es mejor que json en docker
- Protocolo tcp **zeromq** (protocolo muy veloz)

<https://www.papertrail.com/> **servicio cloud** (Papertrail permite ver logs de manera centralizada) también existe Google cloud loggin pero las alarmas son difíciles de configurar.

El kuberniquis de Google es muy fácil de usar

### **Implementar herramientas de login para no estar a siegas es mortal!.**

Sincronizar dos clusters independientes.¿como sincronizarlos? Kuberniquis es orquestador de contenedores..ahora a ese orquestado pero tengo dos independientes, se le puede poner uno encima un Orquestador de orquestadores “rancher” se le configura las api key de aws y azure para que se conecte. Desde rancher se pueden administrar varios clusters de kuberniques de multiples nubes. El seconectara a todas las nubes y orquesta a todas ellas.

- Minikube sirve para probar kubernetes en local  
<https://kubernetes.io/es/docs/tasks/tools/install-minikube/>

Se usa kubernetes para de alta escalabilidad.

- Terraform permite configurar maquinas a puro código.  
[https://es.wikipedia.org/wiki/Terraform\\_\(software\)](https://es.wikipedia.org/wiki/Terraform_(software))

### **Optimización de velocidad:**

<https://en.wikipedia.org/wiki/Webpack>

### **Seguridad**

- Cerrar puertos.
- Redis viene sin clave de igual manera mongodb
- Existen firewalls para wordpress.
- Usar capchas honeybots para formularios.

**Minificar recursos utilizando HTTP2. Esto sirve para que en una sola petición se carguen todos los archivos html, css y js. Lo habitual es hacer una petición al servidor para cargar cada archivo.**

**Usar http2 (cloudflare regala http2).**

**El http2 se incorpora en los headers del servidor. Meter el push del archivo css, y el push del js en las peticiones del servidor en los headers y esto devolverá todo en una sola petición.**

**Utilizar formato de image WebP <https://es.wikipedia.org/wiki/WebP> e lugar de jpg**

### **Medir la velocidad de sitios:**

- Gmetrix: <https://gtmetrix.com/> (Es bueno) Da un vistazo general del sitio y monitoriar los sitios
- Page speed insight <https://developers.google.com/speed/pagespeed/insights/?hl=es> (Indica que esta fallando tanto en mobile como en desktops) Hacer 2 a 3 análisis.

**-Estrategias para mejorar el CEO**

**- Hacer backups (copias de seguridad)**

## **Pre/Post Production**

Speed Optimization

Analytics

Security

Logging

Backups

**Gracias a Juliam Lamprea y Perfil TIC.**

**Charla:** <https://www.youtube.com/watch?v=RkrbM6PCbl0>