

K8s + Proxmox: El Clúster "On-Premise" Perfecto

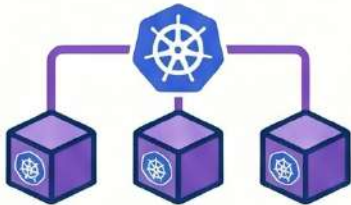
Infraestructura escalable y automatizada para microservicios.

1. Base de Virtualización con Proxmox



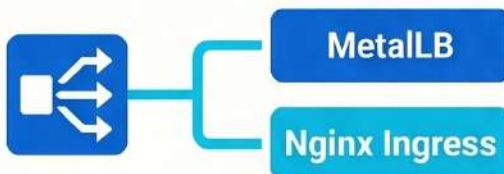
Hypervisor robusto que gestiona recursos físicos mediante **máquinas virtuales optimizadas**.

2. Orquestación con Kubernetes v1.30



Clúster con **nodos redundantes** para garantizar que las aplicaciones nunca se detengan.

3. Balanceo de Red y Tráfico

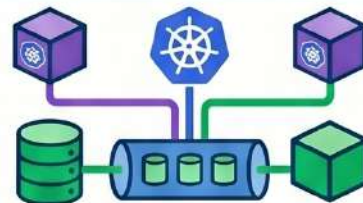


Uso de MetalLB para IPs locales y Nginx Ingress para rutas por dominio.

4. Almacenamiento Persistente via NFS



Datos desacoplados que sobreviven a fallos de contenedores mediante volúmenes compartidos.

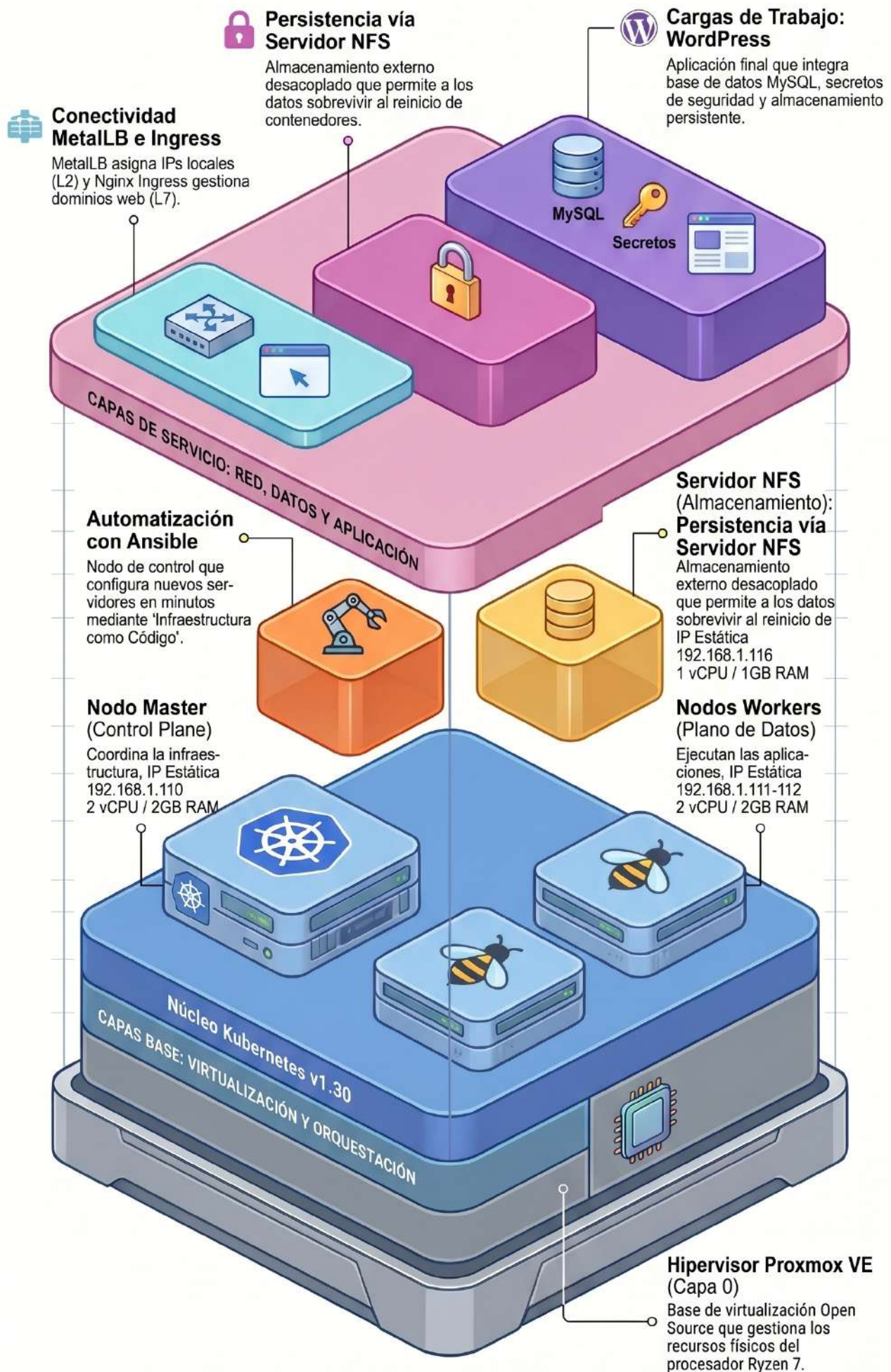


5. Automatización Total con Ansible



Escalado horizontal: configura nuevos nodos en minutos sin intervención manual.

Arquitectura del Clúster: Kubernetes sobre Proxmox VE



Ruta de Despliegue: Clúster Kubernetes de Alta Disponibilidad sobre Proxmox

Fase 1: Cimientos del Hipervisor (Proxmox)



Configuración Crítica de BIOS
Activar modo SVM (AMD) o Intel VT-x para virtualización por hardware

Red de Gestión Estática
IP estática obligatoria (ej. 192.168.1.100) para acceso seguro

Optimización de Repositorios
Desactivar 'enterprise', habilitar 'no-subscription' para actualizaciones

Fase 2: La Plantilla Maestra (Ubuntu Server)



Creación de la "Gold Template"
Ubuntu 24.04 LTS con OEMU Agent y OpenSSH, listo para clonar nodos

Ajuste de Red con Netplan
Definir rutas explícitas (routes), evitando 'gateway4' en Ubuntu 24.04

Fase 3: Despliegue del Núcleo Kubernetes

Preparación del Kernel de Linux



Tráfico L2 vs. Tráfico L7
Asigna IPs locales reales (rango .200-.250) mediante ARP



Runtime y CNI



Configuración de Nodos
1 Master (Control Plane),
2 Workers. Mínimo 2 vCPUs,
2 GB RAM cada uno

Fase 4: Servicios de Red y Almacenamiento



MetalLB
Asigna IPs locales reales (rango .200-.250) mediante ARP



Nginx Ingress
Gestiona tráfico web basado en nombres de dominio



Persistencia vía NFS
Servidor externo con 'NFS Subdir External Provisioner' para datos persistentes

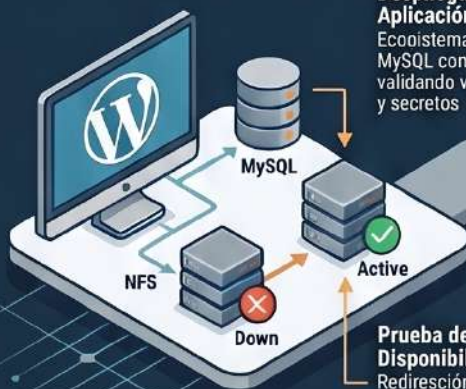
Fase 5: Automatización y Escalado (Ansible)

Infraestructura como Código (IaC)
Automatiza configuración y parches con Playbooks (ej. Worker 03)

Seguridad con Ansible Vault
Tokens y contraseñas cifradas para seguridad profesional



Fase 6: Validación Final (WordPress)



Despliegue de Aplicación Stateful
Ecosistema WordPress + MySQL conectado a NFS, validando volúmenes y secretos

Prueba de Alta Disponibilidad (HA)
Redirección automática de tráfico al Worker 02 tras fallo del Worker 01, sin interrupción

Inventario de Nodos: Resumen de topología lógica

ID	Nombre del Nodo	Función	IP Estática
110	k8s-master	Control Plano	192.168.1.110
111	k8s-worker-01	Worker	192.168.1.111
112	k8s-worker-02	Worker	192.168.1.112
115	ansible-server	Automatización	192.168.1.115
116	nfs-server	Almacenamiento	192.168.1.116