Universidad Latina de Costa Rica

BISOF-18 Sistemas Operativos II

Eduardo Domínguez

Análisis de Caso #1

Plataforma Web Institucional Escalable con Contenedores

Introducción

Este documento presenta un análisis de la propuesta para implementar una plataforma web institucional para la Universidad Latina de Costa Rica. El objetivo es ofrecer un servicio con alta disponibilidad, seguridad, rendimiento y escalabilidad, mediante el uso de contenedores Docker, balanceadores de carga y herramientas de automatización. La solución se plantea como un entorno robusto que integra prácticas modernas de sistemas operativos, redes y arquitectura distribuida.

Desarrollo

Diseño de la Plataforma

La plataforma se construye sobre contenedores, que aíslan aplicaciones como Java, NodeJS, Drupal y WordPress. Estos contenedores son gestionados y expuestos mediante un balanceador de carga, el cual distribuye las peticiones de manera uniforme, evitando sobrecarga en un único servidor y asegurando la continuidad del servicio.

Balanceo de Carga y Alta Disponibilidad

HAProxy cumple un papel esencial en el balanceo de carga, implementando algoritmos como Round Robin y Least Connections. En caso de que un contenedor o servicio falle, el tráfico se redirige automáticamente hacia los contenedores disponibles. Esto garantiza la alta disponibilidad del sistema y permite escalar horizontalmente mediante la creación de nuevas réplicas.

Seguridad

La seguridad se fortalece con el uso de HTTPS y certificados SSL, ya sean de Let’s Encrypt o auto-firmados. Cada usuario cuenta con credenciales exclusivas para acceder mediante SSH, limitadas a su propio directorio, lo que reduce la exposición de información sensible. También se aplican permisos estrictos y se recomienda el uso de herramientas como Fail2ban y reglas de firewall para mitigar accesos no autorizados.

Escalabilidad

La plataforma permite escalar horizontalmente gracias a un sistema de archivos compartido (NFS) accesible desde todas las máquinas virtuales. Aunque NFS puede introducir cierta latencia, esta se mitiga con cachés locales como Varnish o servicios externos como Cloudflare. Para la automatización de la creación y configuración de servidores se proponen herramientas como Terraform y Ansible.

Responsabilidades

El proveedor es responsable de mantener la infraestructura activa y actualizada, instalar componentes como Apache, PHP y MariaDB, así como aplicar parches de seguridad. Los usuarios deben gestionar sus aplicaciones, archivos y bases de datos. Cada usuario es responsable de la seguridad de sus propios proyectos dentro de la plataforma.

Riesgos y Mitigación

- Propagación de malware: mitigación con escáneres de seguridad y protección en el balanceador. - Ataques DDoS: mitigación con firewalls avanzados y uso de CDN. - Desincronización de software: mitigación con Ansible para homogeneizar entornos. - Dependencia de NFS: mitigación adoptando soluciones de almacenamiento distribuidas como Ceph o GlusterFS.

Comparación con Entornos Reales

En plataformas reales como Drupal y WordPress, además de esta infraestructura, se aplican plugins de optimización, cachés, optimización de imágenes y copias de seguridad automáticas. Estas prácticas aseguran un mejor desempeño bajo alta demanda y reducen los riesgos de pérdida de datos.

Conclusiones

La propuesta para la Universidad Latina consiste en una plataforma PaaS moderna y escalable que ofrece seguridad, alta disponibilidad y flexibilidad. El uso de contenedores Docker, balanceadores de carga como HAProxy, almacenamiento compartido NFS y herramientas de automatización como Terraform y Ansible, permiten establecer un entorno confiable. Con una adecuada distribución de responsabilidades entre proveedor y usuario, junto con mecanismos de seguridad y redundancia, se asegura la continuidad del servicio y la protección de los recursos institucionales.