



Práctica 4.1. Preguntas sobre el tema

DESPLIEGUE DE APLICACIONES WEB

Alejandro Leo Carretero
2º DAW-A

Actividad 4.1. Preguntas sobre Administración de Servidores Web	3
¿Qué es el protocolo HTTPS y por qué es importante?	3
¿Cuál es la diferencia entre escalabilidad vertical y horizontal?	3
Explica en detalle las diferencias entre los entornos de desarrollo, pruebas y producción en el despliegue de aplicaciones web.	3
Analiza las ventajas y desventajas del uso de contenedores como Docker frente a máquinas virtuales para el despliegue de aplicaciones web.	4
Describe cómo se implementa un sistema de integración y despliegue continuo (CI/CD) y su impacto en el desarrollo ágil.	4
Detalla los pasos para implementar un certificado SSL/TLS en un servidor web.	5
Explora cómo la escalabilidad horizontal y vertical impacta el rendimiento de aplicaciones web con ejemplos prácticos.	5
Describe el proceso de configuración de un sistema de monitoreo para aplicaciones web utilizando herramientas como Prometheus o Grafana.	6
Explica cómo los servicios en la nube, como AWS o Azure, simplifican el despliegue de aplicaciones web.	6
Analiza las consideraciones de seguridad más importantes al desplegar aplicaciones web en la nube.	6
Explica la importancia de las pruebas automatizadas en el ciclo de vida del despliegue de aplicaciones web.	7

Actividad 4.1. Preguntas sobre Administración de Servidores Web

¿Qué es el protocolo HTTPS y por qué es importante?

HTTPS es una versión segura de HTTP que utiliza cifrado a través de SSL o TLS para proteger la comunicación entre el cliente y el servidor.

Importancia:

- Cifrado de datos: Protege información sensible como contraseñas y datos personales.
 - Autenticación: Garantiza que el servidor con el que te comunicas es el legítimo.
 - Integridad: Impide que los datos transferidos sean alterados o interceptados.
 - Mejora de confianza: Los navegadores destacan sitios con HTTPS como seguros, aumentando la credibilidad de los mismos.
-

¿Cuál es la diferencia entre escalabilidad vertical y horizontal?

- Escalabilidad vertical: Incrementa la capacidad de un servidor existente añadiendo más recursos como CPU, RAM o almacenamiento. Es simple de implementar pero tiene un límite físico y suele ser más costosa.
 - Escalabilidad horizontal: Consiste en añadir más servidores para distribuir la carga de trabajo. Es más eficiente a largo plazo, pero requiere sistemas distribuidos para gestionar los servidores adicionales.
-

Explica en detalle las diferencias entre los entornos de desarrollo, pruebas y producción en el despliegue de aplicaciones web.

- Desarrollo: Es el entorno donde los desarrolladores trabajan directamente con el código. Utiliza datos ficticios y permite cambios frecuentes.
- Pruebas: Este entorno se utiliza para verificar que la aplicación funciona como se espera antes de ser desplegada. Aquí se ejecutan pruebas automatizadas y manuales con una configuración similar a la producción, pero sin datos reales.

- Producción: Es el entorno donde la aplicación está disponible para los usuarios finales. Usa infraestructura optimizada y datos reales, y se espera estabilidad total.
-

Analiza las ventajas y desventajas del uso de contenedores como Docker frente a máquinas virtuales para el despliegue de aplicaciones web.

- Ventajas de Docker:
 - Son ligeros, ya que comparten el sistema operativo anfitrión.
 - Se inician más rápido y facilitan la portabilidad entre diferentes entornos.
 - Facilitan el escalado y la orquestación con herramientas como Kubernetes.
 - Desventajas de Docker:
 - Proporcionan menor aislamiento que las máquinas virtuales.
 - Dependen de las capacidades del sistema operativo subyacente.
 - Ventajas de máquinas virtuales:
 - Aislamiento total al incluir un sistema operativo completo.
 - Soportan múltiples sistemas operativos en una misma máquina física.
 - Desventajas de máquinas virtuales:
 - Son más pesadas y consumen más recursos.
 - Tardan más tiempo en iniciar.
-

Describe cómo se implementa un sistema de integración y despliegue continuo (CI/CD) y su impacto en el desarrollo ágil.

1. Configurar un sistema de control de versiones como Git.
2. Definir un pipeline de CI/CD con herramientas como Jenkins, GitLab CI o GitHub Actions.
3. Automatizar la ejecución de pruebas en cada cambio del código.
4. Configurar procesos de construcción y despliegue en entornos de pruebas o producción.
5. Monitorear los despliegues para identificar errores rápidamente.

Impacto en el desarrollo ágil:

- Acelera la entrega de nuevas funcionalidades.
 - Mejora la calidad del software mediante pruebas automatizadas.
 - Permite a los equipos colaborar de forma continua sin interrupciones en los despliegues.
-

Detalla los pasos para implementar un certificado SSL/TLS en un servidor web.

1. Generar una solicitud de firma de certificado (CSR) en el servidor.
 2. Enviar la solicitud a una autoridad certificadora para obtener el certificado.
 3. Descargar e instalar el certificado en el servidor.
 4. Configurar el servidor para usar HTTPS en el puerto 443.
 5. Probar la configuración utilizando herramientas como SSL Labs para asegurar que el certificado está correctamente instalado.
-

Explora cómo la escalabilidad horizontal y vertical impacta el rendimiento de aplicaciones web con ejemplos prácticos.

- En la escalabilidad vertical, aumentar los recursos de un servidor mejora su rendimiento. Ejemplo: Incrementar la RAM de un servidor de 8 GB a 16 GB puede permitir manejar más usuarios concurrentes.
- En la escalabilidad horizontal, añadir más servidores permite dividir la carga de trabajo. Ejemplo: Implementar tres servidores detrás de un balanceador de carga puede distribuir 100 mil solicitudes entre ellos y reducir el tiempo de respuesta.

Impacto:

La escalabilidad horizontal es ideal para manejar grandes volúmenes de tráfico distribuyendo la carga, mientras que la vertical es más útil para soluciones rápidas en sistemas con menor complejidad.

Describe el proceso de configuración de un sistema de monitoreo para aplicaciones web utilizando herramientas como Prometheus o Grafana.

1. Instalar Prometheus en un servidor para recopilar métricas.
 2. Configurar los servicios para exponer métricas en un formato compatible con Prometheus.
 3. Especificar los endpoints de las métricas en el archivo de configuración de Prometheus.
 4. Instalar Grafana y configurarlo para conectarse con Prometheus como fuente de datos.
 5. Crear dashboards personalizados en Grafana para visualizar métricas como el uso de CPU, tiempo de respuesta y disponibilidad.
-

Explica cómo los servicios en la nube, como AWS o Azure, simplifican el despliegue de aplicaciones web.

- Ofrecen soluciones administradas para despliegue como Elastic Beanstalk o App Service, reduciendo la complejidad técnica.
 - Permiten escalado automático para manejar incrementos de tráfico sin intervención manual.
 - Proveen herramientas integradas para redes, bases de datos y almacenamiento.
 - Ofrecen alta disponibilidad y recuperación ante desastres.
 - Facilitan el pago por uso, reduciendo los costos iniciales.
-

Analiza las consideraciones de seguridad más importantes al desplegar aplicaciones web en la nube.

- Utilizar políticas de acceso basadas en roles para limitar permisos.
- Asegurar las claves de acceso y credenciales con servicios como Secrets Manager.
- Habilitar el cifrado de datos en tránsito y en reposo.
- Implementar sistemas de detección de amenazas y firewalls.
- Configurar auditorías periódicas para identificar vulnerabilidades.

- Mantener el software actualizado para protegerlo contra exploits conocidos.
-

Explica la importancia de las pruebas automatizadas en el ciclo de vida del despliegue de aplicaciones web.

- Ayudan a detectar errores rápidamente antes de que el código llegue a producción.
- Permiten validar cambios en el código de forma continua sin intervención manual.
- Mejoran la calidad y confiabilidad del software al identificar problemas en escenarios complejos.
- Aceleran el tiempo de desarrollo y despliegue, facilitando un flujo ágil.
- Reducen los riesgos asociados con cambios frecuentes en la aplicación.