TEMA 1

1. Definición de Centro de Datos (TIA-942)

- Centro de Datos: Instalación que alberga infraestructura de TI y telecomunicaciones críticas, garantizando disponibilidad, continuidad, seguridad física y lógica, y condiciones adecuadas para el funcionamiento del equipo.

2. Clasificación de Centros de Datos

- Single-Tenant (Empresariales)
  - Propiedad Privada: Corporaciones, instituciones o entidades gubernamentales.
  - Uso: Soporte a transacciones y procesamiento de datos internos.
  - Administración: Gestionados por personal interno.
- Multi-Tenant (Compartidos)
  - Propiedad/Administración: Compañías de telecomunicaciones o proveedores de servicios.
  - Uso: Ofrecen servicios de TI a terceros.
  - Administración: Múltiples clientes comparten la infraestructura manteniendo entornos separados.

3. Objetivos Generales de un Centro de Datos

- Facilidad de Administración: Simplificar la gestión de recursos y operaciones.
- Flexibilidad y Escalabilidad: Adaptarse a las necesidades cambiantes y crecer según demanda.
- Eficiencia de Red: Optimizar el rendimiento y la conectividad de la red.

4. Elementos Fundamentales

- Infraestructura de Red/Comunicaciones: Routers, switches, cableado.
- Infraestructura de Procesamiento: CPUs, servidores, GPUs.
- Infraestructura de Almacenamiento: SAN, NAS, SDS.
- Virtualización:
  - SDN (Software Defined Networks): Separa el plano de control del de datos, permitiendo redes programables y flexibles.
  - NFV (Network Functions Virtualization): Virtualiza funciones de red (firewalls, balanceadores) como máquinas virtuales.
  - Máquinas Virtuales vs Contenedores: VMs tienen sistemas operativos independientes; contenedores comparten el núcleo del SO y son más ligeros.

5. Normativas y Estándares

- ANSI/TIA-942: Estándar para diseño físico y telecomunicaciones en centros de datos, enfocado en fiabilidad, escalabilidad y disponibilidad.
- CENELEC EN 50173-5: Especificaciones técnicas de cableado para interoperabilidad y rendimiento.
- ISO/IEC 24764: Configuraciones de cableado para alta demanda de ancho de banda y confiabilidad.

6. Redundancia y Niveles de Redundancia

- N: Número básico de elementos necesarios.
- N+1: Un componente adicional para soportar fallos o mantenimiento.
- 2N: Duplicado completo de cada elemento crítico.
- 2(N+1): Redundancia doble con replicación adicional.

7. Tiers de Disponibilidad (ANSI/TIA-942)

- Tier I: Básico, sin redundancia (99.671% disponibilidad).
- Tier II: Redundancia parcial (N+1) (99.741%).
- Tier III: Redundancia completa (N+1) en todos los componentes críticos (99.982%).
- Tier IV: Tolerante a fallos con redundancia 2N o 2N+1 (99.995%).

8. Infraestructura Física

- Racks y Gabinetes:
  - Dimensiones: Altura en unidades de rack (1U = 1.75”), ancho estándar de 19” o 23”, profundidad variable.
  - Organización: Montaje ordenado para equipos informáticos y de comunicación.
- Suelos Técnicos Elevables:
  - Usos: Canalización de cables y circulación de aire acondicionado.
- Climatización:
  - CRAC (Computer Room Air Conditioners): Sistemas especializados para mantener condiciones ambientales estables.
  - HVAC: Sistemas generales de climatización, combinados con la estructura del centro.
- Distribución Hot Aisle / Cold Aisle:
  - Hot Aisle: Pasillos traseros con aire caliente.
  - Cold Aisle: Pasillos frontales con aire frío.
  - Contención: Separación física para mejorar eficiencia de enfriamiento.

9. Alimentación Eléctrica

- SAI (Sistemas de Alimentación Ininterrumpida): Baterías que respaldan equipos durante cortes breves.
- Generadores: Proveen energía en caso de fallos prolongados.
- PDU (Unidad de Distribución de Energía): Distribuyen la energía a los equipos, pueden ser pasivas, medibles, monitoreadas, conmutadas o de transferencia automática.
- PUE (Power Usage Effectiveness): Métrica de eficiencia energética (Energía Total / Energía TI).

10. Seguridad Física

- Controles de Acceso: Autenticación mediante tarjetas o biometría.
- Vigilancia y Monitoreo: Cámaras, sensores de temperatura y humedad.
- Protección contra Incendios: Sistemas de detección y extinción.
- Continuidad del Negocio: Planes de recuperación ante desastres y redundancia de equipos.

11. Gestión de Cableado y Etiquetado

- Tipos de Cableado:
  - Backbone: Fibra óptica multimodo o monomodo.
  - Horizontal: Par trenzado categoría 5e, 6, 6A u 8A.
- Organización:

    - ToR (Top-of-Rack): Switches en la parte superior de cada rack.
    - EoR (End-of-Row): Switches al final de filas de racks.
- Etiquetado:
    - Componentes: Cables, paneles de parcheo, racks.
    - Requisitos: Legible, consistente y con identificadores únicos.

12. Infraestructura de Procesamiento y Almacenamiento

- Servidores:
    - Independientes: Más espacio, menos eficiencia.
    - En Rack y Blade: Más compactos y eficientes.
- Almacenamiento:
    - SAN (Storage Area Network): Acceso a nivel de bloque, red dedicada.
    - NAS (Network Attached Storage): Acceso a nivel de archivo, compartido en red.
    - SDS (Software Defined Storage): Gestión de almacenamiento separada del hardware.

TEMA 2

1. Definición de Sistemas de Almacenamiento

- Sistemas de Almacenamiento: Conjuntos de dispositivos y tecnologías que permiten guardar y gestionar grandes volúmenes de datos, asegurando disponibilidad rápida y segura para usuarios y aplicaciones.

2. Tipos de Sistemas de Almacenamiento

- SAN (Storage Area Network):
    - Descripción: Red dedicada para la gestión centralizada de almacenamiento.
    - Características: Alta escalabilidad, velocidad de acceso a datos, acceso a nivel de bloque.
    - Uso Ideal: Aplicaciones que requieren acceso rápido y eficiente a grandes volúmenes de datos.
- NAS (Network Attached Storage):
    - Descripción: Dispositivo de almacenamiento conectado a una red común (LAN).
    - Características: Acceso a nivel de archivo, fácil distribución y gestión de archivos.
    - Uso Ideal: Compartición de archivos en entornos colaborativos y accesibles por múltiples usuarios.

3. Virtualización del Almacenamiento

- Logical Volume Manager (LVM):
    - Descripción: Herramienta que abstrae y gestiona el espacio de almacenamiento de manera flexible.
    - Componentes Clave:
        - Volumen Físico (PV): Discos o particiones físicas.
        - Grupo de Volúmenes (VG): Conjunto de PVs.
        - Volumen Lógico (LV): Unidades de almacenamiento lógicas dentro de VGs.
    - Ventajas:
        - Flexibilidad: Crear, modificar y eliminar volúmenes sin afectar el sistema operativo.
        - Escalabilidad: Expandir almacenamiento sin interrupciones.
        - Gestión Eficiente: Agrupar y administrar almacenamiento de manera óptima.
    - Funcionamiento: Abstracción en capas que permite gestionar el almacenamiento independientemente del hardware físico.

4. RAID (Redundant Array of Independent Disks)

- Definición: Técnica que combina múltiples discos en una sola unidad lógica para mejorar rendimiento, capacidad y/o tolerancia a fallos.
- Características:
    - Data Striping: Distribuye bloques de datos entre discos para mejorar el rendimiento.
    - Paridad: Bloques de paridad para recuperación de datos en caso de fallo.
    - Mirroring: Réplica de datos en múltiples discos para redundancia.
    - Spare Disks: Discos de reserva que reemplazan automáticamente a los fallidos.

5. Niveles de RAID Estándar

- RAID 0 (Striping):
    - Descripción: Distribuye datos entre múltiples discos.
    - Ventajas: Alto rendimiento en lectura/escritura.
    - Desventajas: No ofrece redundancia; pérdida de datos si falla cualquier disco.
    - Uso Ideal: Aplicaciones que requieren alto rendimiento sin necesidad de redundancia.
- RAID 1 (Mirroring):
    - Descripción: Duplica datos en dos o más discos.
    - Ventajas: Alta redundancia y tolerancia a fallos; buena velocidad de lectura.
    - Desventajas: Duplicación de costos de almacenamiento; baja velocidad de escritura.
    - Uso Ideal: Datos críticos que requieren alta disponibilidad.
- RAID 5 (Striping con Paridad):
    - Descripción: Distribuye datos y paridad de forma cíclica entre los discos.
    - Ventajas: Redundancia con paridad; buen rendimiento de lectura.
    - Desventajas: Escrituras más lentas debido al cálculo de paridad; tolerancia limitada a un fallo de disco.
    - Uso Ideal: Sistemas que requieren equilibrio entre rendimiento, capacidad y redundancia.
- RAID 6 (Striping con Doble Paridad):
    - Descripción: Similar a RAID 5 pero con dos bloques de paridad.
    - Ventajas: Tolerancia a dos fallos de discos.
    - Desventajas: Mayor sobrecarga en el cálculo de paridad; escrituras más lentas.
    - Uso Ideal: Sistemas que requieren alta disponibilidad y tolerancia a múltiples fallos de discos.

6. RAID Anidado (Nested RAID)

- RAID 0+1 (Stripe + Mirror):
    - Descripción: Combina RAID 0 (striping) y RAID 1 (mirroring).
    - Ventajas: Alto rendimiento y redundancia.

    - Desventajas: Degradación del RAID 0 puede afectar la redundancia completa; costo elevado.
    - Uso Ideal: Entornos que requieren tanto alto rendimiento como redundancia.
- RAID 1+0 (Mirror + Stripe):
    - Descripción: Combina RAID 1 (mirroring) y RAID 0 (striping).
    - Ventajas: Mejor tolerancia a fallos que RAID 0+1; no degrada completamente si falla un RAID 1.
    - Desventajas: Costo elevado; complejidad en la configuración.
    - Uso Ideal: Entornos críticos que requieren alta disponibilidad y rendimiento.
- RAID 5+0 y RAID 10+0:
    - Descripción: Combinaciones más complejas que integran múltiples niveles de RAID para optimizar rendimiento y redundancia.
    - Ventajas: Alta tolerancia a fallos y mejor rendimiento.
    - Desventajas: Mayor complejidad y costos; reconstrucción lenta en caso de fallos.

7. Comparativa General de Niveles de RAID

Nivel RAID	Ventajas	Desventajas	Uso Típico
RAID 0	Alto rendimiento, máxima capacidad	Sin redundancia, pérdida de datos	Aplicaciones de alto rendimiento
RAID 1	Alta redundancia, buena lectura	Duplicación de costos, baja escritura	Datos críticos
RAID 5	Buen equilibrio rendimiento/redundancia	Escrituras lentas, tolerancia limitada	Servidores de archivos
RAID 6	Tolerancia a dos fallos de discos	Mayor sobrecarga, escrituras lentas	Sistemas críticos de alta disponibilidad
RAID 10	Alto rendimiento y redundancia	Costoso, requiere al menos 4 discos	Entornos de misión crítica

8. Combinación de Tecnologías: LVM y RAID

- Integración LVM con RAID:
    - Descripción: Combinar LVM y RAID permite gestionar el almacenamiento de manera flexible mientras se asegura redundancia y rendimiento.
    - Ventajas:
        - Flexibilidad: Gestionar volúmenes lógicos sobre arreglos RAID.
        - Redundancia y Rendimiento: Utilizar RAID para mejorar seguridad y rendimiento, gestionado por LVM para adaptabilidad.
    - Escenarios Útiles:
        - Centros de Datos Grandes: Donde se requiere alta disponibilidad y gestión eficiente del almacenamiento.
        - Entornos Dinámicos: Donde las necesidades de almacenamiento cambian frecuentemente y se necesita escalar sin interrupciones.

9. Motivaciones para Usar LVM y RAID

- LVM:
    - Problema a Solucionar: Limitaciones en la gestión de particiones tradicionales, dificultad para redimensionar y flexibilidad limitada.
    - Solución: Abstracción y gestión dinámica del espacio de almacenamiento.
- RAID:
    - Problema a Solucionar: Necesidad de mejorar rendimiento, capacidad y/o asegurar la disponibilidad de datos frente a fallos de discos.
    - Solución: Combinar múltiples discos para lograr objetivos específicos de rendimiento y redundancia.

10. Ventajas y Desventajas Clave

- RAID:
    - Ventajas:
        - Mejor rendimiento y/o redundancia.
        - Flexibilidad para agregar o quitar discos.
    - Desventajas:
        - Costos adicionales.
        - Mayor complejidad en la gestión.
        - Dependencia de la configuración y tipo de RAID utilizado.
- LVM:
    - Ventajas:
        - Alta flexibilidad y escalabilidad.
        - Simplificación de la gestión del almacenamiento.
    - Desventajas:
        - Requiere conocimientos adicionales para su configuración y mantenimiento.
        - Potencial sobrecarga en el sistema si no se gestiona adecuadamente.

11. Combinación de Tecnologías: LVM y RAID

- Integración de LVM y RAID:
    - ¿Se puede hacer? Sí, se pueden combinar LVM (Logical Volume Manager) y RAID para gestionar el almacenamiento de manera flexible mientras se asegura redundancia y rendimiento.
    - ¿Tiene sentido? Sí, ya que permite aprovechar las ventajas de ambos: la flexibilidad y escalabilidad de LVM junto con la redundancia y mejora de rendimiento de RAID.
    - Escenarios Útiles:
        - Centros de Datos Grandes: Donde se requiere alta disponibilidad y gestión eficiente del almacenamiento.
        - Entornos Dinámicos: Donde las necesidades de almacenamiento cambian frecuentemente y se necesita escalar sin interrupciones.
- Ejemplo de Configuración:
    - LVM sobre RAID 5:
        - RAID 5: Proporciona striping con paridad, mejorando rendimiento y tolerancia a fallos.
        - LVM: Gestiona volúmenes lógicos sobre el arreglo RAID, permitiendo flexibilidad en la asignación de espacio.

12. Redes de Almacenamiento (Storage Networks)

12.1. Tipos de Conexiones

- SATA III:
  - Velocidad: 6 Gbps.
  - Tecnología Base: ATA.
  - Ámbito: Local.
  - Uso: Conexión directa a servidores o estaciones de trabajo.
- SAS-3:
  - Velocidad: 12 Gbps.
  - Tecnología Base: SCSI.
  - Ámbito: Local.
  - Uso: Conexión a dispositivos de almacenamiento de alto rendimiento.
- Fibre Channel:
  - Velocidad: 16-64 Gbps.
  - Tecnología Base: SCSI.
  - Ámbito: Red.
  - Uso: Conexiones de alta velocidad en SANs.

12.2. Modelos de Almacenamiento Secundario

- Direct Attached Storage (DAS):
  - Características:
    - Acceso a nivel de bloques.
    - Conexión directa a servidores.
    - Ideal para almacenamiento rápido y dedicado.
  - Conexiones:
    - Cliente-Servidor: SATA-III, SAS.
    - Servidor-Almacenamiento: SATA-III, SAS.
- Network Attached Storage (NAS):
  - Características:
    - Acceso a nivel de ficheros/directorios.
    - Almacenamiento conectado a la red local.
    - Ideal para compartir almacenamiento centralizado.
  - Conexiones:
    - Cliente-NAS: Ethernet, Wi-Fi.
    - NAS-Almacenamiento: SATA-III, SAS.
  - Protocolos: SMB/CIFS (Windows), NFS (Linux/Unix), AFP (MacOS).
- Storage Area Network (SAN):
  - Características:
    - Acceso a nivel de bloques.
    - Red dedicada para almacenamiento.
    - Ideal para centros de datos y aplicaciones críticas.
  - Conexiones:
    - Cliente-Intermediario: iSCSI, FCoE, Fibre Channel, InfiniBand.
    - Intermediario-Almacenamiento: Fibre Channel, iSCSI, FCoE, InfiniBand.
  - Protocolos: FCP, FCoE, iSCSI, AoE, NVMe-oF.

12.3. Protocolos de Redes de Almacenamiento

- Fibre Channel Protocol (FCP):
  - Descripción: Encapsula SCSI sobre fibra óptica.
  - Topologías: Punto a punto (FC-P2P), Anillo Arbitrado (FC-AL), Conmutada (FC-SW).
- Fibre Channel over Ethernet (FCoE):
  - Descripción: Encapsula paquetes FC en tramas Ethernet.
  - Ventajas: Combina las ventajas de Fibre Channel y Ethernet.
- iSCSI (Internet Small Computer System Interface):
  - Descripción: Encapsula SCSI sobre TCP/IP.
  - Ventajas: Permite enrutamiento y flexibilidad en infraestructuras de red.
- Ata over Ethernet (AoE):
  - Descripción: Encapsula ATA sobre Ethernet.
  - Características: Protocolo a nivel de enlace, no soporta enrutamiento.
- NVM Express over Fabrics (NVMe-oF):
  - Descripción: Encapsula tráfico NVMe sobre distintos protocolos de red.
  - Ventajas: Acceso directo y en paralelo al almacenamiento, alta velocidad.

12.4. Estructura en 3 Capas de una Red de Almacenamiento

- Capa de Almacenamiento (Storage Layer):
  - Componentes: Dispositivos de almacenamiento (SSD, HDD), controladoras RAID, discos de reserva.
  - Funciones: Residen los datos y gestionan el acceso físico al almacenamiento.
- Capa de Infraestructura:
  - Componentes: Switches, routers, cableado específico para la red de almacenamiento.
  - Funciones: Facilita la comunicación entre los hosts y los dispositivos de almacenamiento.
- Capa de Hosts:
  - Componentes: Servidores, initiators (dispositivos que acceden al almacenamiento).
  - Funciones: Gestionan el acceso a los datos mediante software adicional como multipath y drivers.

13. Copias de Seguridad (Backups)

13.1. Conceptos Básicos

- Definición: Réplica exacta de la información resguardada, almacenada en formato estándar y con trazabilidad.
- Necesidad:
  - Recuperación ante Catástrofes: Parte esencial del Disaster Recovery Plan.
  - Continuidad del Negocio: Asegura la disponibilidad de datos críticos.
  - Cumplimiento Legal: Responde a regulaciones como la LOPD.

13.2. Componentes de las Copias de Seguridad

- Datos a Resguardar: Depende de la importancia de la información.
- Métodos de Copia:
  - Copias de Seguridad Completa: Copia todos los datos en un momento específico.
  - Copias de Seguridad Diferenciales: Copian solo los cambios desde la última copia completa.
  - Copias de Seguridad Incrementales: Copian solo los cambios desde la última copia, ya sea completa o incremental.
- Procedimientos:
  - Planificación de Copias: Definir qué, cuándo y cómo realizar las copias.
  - Recuperación: Procedimientos para restaurar los datos desde las copias.

13.3. Estrategias de Backup

- Copia de Seguridad Completa:
  - Ventajas: Simplifica la restauración.
  - Desventajas: Requiere mucho espacio y tiempo.
- Copia de Seguridad Diferencial:
  - Ventajas: Menos espacio que una copia completa.
  - Desventajas: Requiere restaurar la copia completa y la diferencial.
- Copia de Seguridad Incremental:
  - Ventajas: Menor tiempo y espacio requerido.
  - Desventajas: Requiere todas las copias incrementales para la restauración.

13.4. Dispositivos de Backup

- Discos Duros (HDD):
  - Uso: Backups on-line u off-line.
  - Ventajas: Alta capacidad, acceso rápido.
  - Desventajas: Fallos mecánicos a largo plazo.
- Unidades de Estado Sólido (SSD):
  - Uso: Backups de acceso rápido.
  - Ventajas: Alta velocidad, resistente a fallos mecánicos.
  - Desventajas: Coste más elevado que HDD.
- Cintas Magnéticas (Tape Drives):
  - Uso: Backups a largo plazo.
  - Ventajas: Gran capacidad, bajo coste por GB.
  - Desventajas: Acceso secuencial, lentitud en recuperación.
- NAS (Network Attached Storage):
  - Uso: Backups centralizados (on-line/near-line).
  - Ventajas: Acceso en red, redundancia RAID.
  - Desventajas: Vulnerable a fallos locales.
- SAN (Storage Area Network):
  - Uso: Backups empresariales masivos.
  - Ventajas: Alta capacidad y rendimiento.
  - Desventajas: Alto coste y complejidad.
- Cloud Storage:
  - Uso: Backups off-site.
  - Ventajas: Escalable, accesible desde cualquier lugar.
  - Desventajas: Dependencia de la conexión a Internet.

13.5. Parámetros de Backup

- Tiempo de Creación: Debe estar dentro de la ventana de backup.
- Tiempo de Almacenamiento: Determina cuánto tiempo se conserva la copia antes de moverla o eliminarla.
- Ubicación de la Copia:
  - On-line: Acceso directo e inmediato (SAN, NAS).
  - Near-line: Acceso automático cuando es necesario.
  - Off-line: Requiere intervención manual (discos almacenados).
  - Off-site: Copias en ubicaciones separadas (nubes).

13.6. Estrategias y Métodos

- Multipath: Acceso a discos por diferentes caminos para redundancia y rendimiento.
- Métodos de Almacenamiento:
  - Full Backup: Copia completa.
  - Differential Backup: Copias los cambios desde la última copia completa.
  - Incremental Backup: Copias los cambios desde la última copia (completa o incremental).

13.7. Ventajas y Desventajas de las Estrategias de Backup

- Copia Completa:
  - Ventajas: Fácil restauración.
  - Desventajas: Alto uso de almacenamiento y tiempo.
- Copia Diferencial:
  - Ventajas: Menor espacio que completa.

    - Desventajas: Necesita una copia completa y la diferencial para restaurar.
- Copia Incremental:
    - Ventajas: Menor espacio y tiempo.
    - Desventajas: Complejidad en la restauración; requiere todas las copias incrementales.

TEMA 3

1. Definición y Tipos de Clústeres

- Clúster (Clúster de Computadores):
    - Definición: Agrupación de computadores junto con las infraestructuras de comunicación y almacenamiento asociadas, destinados a trabajar de forma conjunta para ofrecer un servicio específico o realizar operaciones de cómputo particulares.
    - Objetivo: Mejorar la disponibilidad, rendimiento y capacidad de procesamiento mediante la cooperación entre nodos.
- Tipos de Clústeres:
    - Clústeres de Balanceo de Carga (Load Balancing Clusters):
        - Descripción: Los nodos comparten la carga de trabajo y el tráfico de red.
        - Punto Clave: Repartición equitativa del trabajo para maximizar el número de peticiones atendidas.
        - Objetivo: Atender el máximo número de peticiones del servicio.
    - Clústeres de Alta Disponibilidad (Failover Clusters):
        - Descripción: Los nodos garantizan la disponibilidad del servicio incluso en caso de fallos.
        - Punto Clave: Tolerancia y recuperación ante fallos (failover).
        - Objetivo: Garantizar la prestación y consistencia del servicio sin interrupciones.
    - Clústeres de Alto Rendimiento (High Performance Clusters):
        - Descripción: Nodos que trabajan conjuntamente en tareas de cálculo intensivo.
        - Punto Clave: Procesamiento distribuido.
        - Objetivo: Maximizar el rendimiento y la capacidad de cálculo para tareas como renderizado de gráficos, análisis de datos, etc.

2. Conceptos Previos: Source NAT (SNAT) vs Proxy

- Source NAT (SNAT):
    - Funcionamiento:
        - Traducción de las IPs de origen en la capa de red (IP).
        - No altera ni inspecciona el contenido de los paquetes.
    - Características:
        - Opera en la capa de red.
        - Modifica únicamente la dirección IP de origen de los paquetes.
        - No proporciona funcionalidades adicionales como filtrado o inspección de contenido.
- Proxy (DProxy):
    - Funcionamiento:
        - Actúa como intermediario en la capa de aplicación.
        - Altera e inspecciona los paquetes según el origen y destino.
    - Características:
        - Opera en la capa de aplicación.
        - Proporciona funcionalidades como filtrado de paquetes, monitorización de red, etc.
        - Mejora la seguridad y control del tráfico de red.

3. Balanceo de Carga

- Definición:
    - Conjunto de equipos que comparten la carga de trabajo y el tráfico de red para distribuir las solicitudes entre múltiples nodos backend.
- Componentes:
    - Back-end: Equipos (nodos) que ofrecen los servicios (e.g., servidores web, bases de datos).
    - Front-end: Equipos (balanceadores) que distribuyen las solicitudes entre los nodos backend.
- Escalabilidad:
    - Vertical (Scale Up): Mejorar los nodos existentes añadiendo más recursos (CPU, RAM).
    - Horizontal (Scale Out): Agregar más servidores/nodos para mejorar las prestaciones.
- Limitaciones:
    - No ofrece alta disponibilidad por sí solo:
        - Punto único de fallo.
        - No gestiona redundancia ni fallos en servidores backend.
        - Solución: Implementar redundancia en el balanceador y la infraestructura.

4. Implementaciones de Balanceo de Carga

- Balanceo de Carga por DNS:
    - Funcionamiento:
        - El servidor DNS distribuye el tráfico entre múltiples servidores proporcionando diferentes direcciones IP para un mismo dominio.
        - Método Round Robin: Las resoluciones DNS alternan las IPs de los servidores.
    - Ventajas:
        - Bajo coste.
        - Escalable mediante la actualización de registros DNS.
    - Desventajas:
        - Balanceo no equitativo debido a cachés DNS en clientes.
        - No detecta fallos ni sobrecargas en los nodos.

        - Punto de fallo si el balanceador DNS falla.
- Balanceo de Carga a Nivel de Aplicación (Proxy Inverso):
    - Funcionamiento:
        - El balanceador toma decisiones de redirección basadas en el contenido de las cabeceras HTTP/HTTPS.
    - Ventajas:
        - Mejora el rendimiento distribuyendo solicitudes.
        - Escalabilidad dinámica.
        - Alta disponibilidad mediante redirección a nodos alternativos en caso de fallos.
    - Desventajas:
        - Punto único de fallo; requiere redundancia.
    - Ejemplos de Implementaciones:
        - HAProxy: https://www.haproxy.org/
        - Varnish Cache: https://varnish-cache.org/
        - NginX: https://www.nginx.com/
        - Apache HTTP Server (mod_proxy, mod_proxy_balancer): https://httpd.apache.org/
- Balanceo de Carga a Nivel IP:
    - Linux Virtual Server (LVS):
        - Descripción: Marco de balanceo de carga de código abierto integrado en el kernel de Linux.
        - Componentes:
            - IPVS (IP Virtual Server): Toma decisiones de balanceo a nivel IP.
            - KTCPVS (Kernel TCP Virtual Server): Maneja el balanceo a nivel de transporte.
        - Métodos de Implementación:
            - Virtual Server via NAT: Usa NAT para redirigir el tráfico.
            - Virtual Server via IP-Tunneling: Encapsula las solicitudes en túneles IP.
            - Virtual Server via Direct Routing: Balancea a nivel Ethernet sin encapsulación.
        - Ventajas:
            - Alta disponibilidad.
            - Soporta múltiples protocolos TCP/UDP.
        - Desventajas:
            - Complejidad en la configuración.

5. Estrategias de Balanceo de Carga

- Round-Robin:
    - Descripción: Reparte las cargas entre servidores de forma alterna.
    - Supuestos:
        - Todos los servidores tienen la misma capacidad.
        - Las cargas de trabajo son similares.
    - Ventaja: Equitativo y sencillo de implementar.
- Weighted Round-Robin:
    - Descripción: Asigna un peso a cada servidor basado en su capacidad.
    - Funcionamiento: Reparte las cargas proporcionalmente al peso asignado.
    - Ventaja: Mejor distribución según capacidades reales de los servidores.
- Least Connection:
    - Descripción: Reparte las cargas a los servidores con menos conexiones activas.
    - Ventaja: Equilibrio dinámico basado en la carga actual de los servidores.
- Weighted Least Connection:
    - Descripción: Combina el menor número de conexiones con un peso asignado.
    - Funcionamiento: Asigna solicitudes al servidor con la mejor relación entre conexiones abiertas y peso.
- Asignación Estática:
    - Descripción: Asigna de manera fija servidores a direcciones IP de origen.
    - Ventaja: Predecible y simple.
    - Desventaja: Menor flexibilidad y balanceo dinámico.

6. Persistencia de Conexiones

- Necesidad:
    - Algunas aplicaciones requieren mantener información sobre conexiones/peticiones previas (sesiones).
    - HTTP/HTTPS son protocolos sin estado, lo que puede llevar a que las solicitudes sucesivas sean dirigidas a diferentes servidores.
- Métodos de Manejo de Persistencia:
    - Session Replication:
        - Descripción: Replica la información de las sesiones en todos los nodos del clúster.
        - Ventajas: Sencillez de implementación.
        - Desventajas: Alto consumo de recursos computacionales y de almacenamiento.
    - Sticky Cookies:
        - Descripción: Utiliza cookies con IDs únicos de sesión para identificar y redirigir las solicitudes del cliente al mismo servidor backend.
        - Funcionamiento:
            1. El balanceador asigna una cookie única en la primera respuesta (Set-Cookie).
            2. Las solicitudes subsecuentes incluyen la cookie.
            3. El balanceador usa la cookie para identificar el servidor correspondiente.
        - Ventajas: Mantiene la sesión del cliente en un único servidor.
        - Desventajas: Menor flexibilidad si el servidor backend falla; requiere manejo adecuado de cookies.

7. Implementaciones Específicas

- Linux Virtual Server (LVS):
  - Virtual Server via NAT:
    - Funcionamiento:
      1. Cliente envía solicitud al balanceador.
      2. Balanceador usa NAT para redirigir la solicitud al servidor backend.
      3. Servidor responde al balanceador.
      4. Balanceador retransmite la respuesta al cliente.
  - Virtual Server via IP-Tunneling:
    - Funcionamiento:
      1. Cliente envía solicitud al balanceador.
      2. Balanceador encapsula la solicitud en túneles IP hacia el servidor backend.
      3. Servidor desencapsula y procesa la solicitud.
      4. Servidor responde directamente al cliente.
  - Virtual Server via Direct-Routing:
    - Funcionamiento:
      1. Cliente envía solicitud al balanceador.
      2. Balanceador redirige la solicitud al servidor backend a nivel Ethernet.
      3. Servidor responde directamente al cliente, evitando el balanceador en la respuesta.

1. Alta Disponibilidad (High Availability - HA)

Definición:

- Alta Disponibilidad: Conjunto de nodos que garantizan que un servicio permanezca operativo incluso ante fallos imprevistos (hardware/software) o paradas previstas (mantenimiento).

Características Principales:

- Comunicación entre nodos: Los nodos del clúster deben comunicarse constantemente para monitorear el estado.
- Nodos redundantes: Disponibilidad de nodos de respaldo para asumir funciones en caso de fallo.
- Detección de fallos: Identificación rápida de fallos en hardware o software.
- Recuperación automática: Reinicio de servicios en nodos alternativos sin intervención manual.
- Integridad de datos: Asegurar que los datos permanecen consistentes y no se corrompen durante fallos.

Uso:

- Servicios críticos que no pueden permitirse tiempo de inactividad, como:
  - Bases de datos críticas.
  - Aplicaciones web de comercio electrónico.
  - Sistemas de ficheros compartidos.
  - Servidores de correo.

Requisitos:

- Fiabilidad:
  - Capacidad de funcionar de manera consistente.
  - Medida por MTBF (Mean Time Between Failures).
- Disponibilidad:
  - Porcentaje de tiempo que el sistema está operativo.
  - Técnicas para mantener la disponibilidad:
    - Redundancia.
    - Failover automático.
    - Balanceo de carga.
- Facilidad de mantenimiento:
  - Gestionar actualizaciones sin afectar la disponibilidad.
  - Medida por MTTR (Mean Time To Repair).

Métricas Clave:

- Uptime: Tiempo total en funcionamiento.
- Downtime: Tiempo total fuera de servicio.
- MTTF (Mean Time To Failure): Tiempo promedio hasta un fallo.
- MTTR (Mean Time To Repair): Tiempo promedio para reparar un fallo.
- MTBF (Mean Time Between Failures): Tiempo promedio entre fallos.

Principios Básicos:

- Redundancia Hardware: Replicación de componentes físicos clave para evitar puntos únicos de fallo.
- Redundancia Software: Replicación de aplicaciones o servicios en diferentes nodos.
- Redundancia de Datos: Replicación de datos en múltiples ubicaciones o dispositivos.

2. Tipos de Clústeres

Clústeres de Balanceo de Carga (Load Balancing Clusters):

- Descripción: Distribuyen la carga de trabajo y tráfico de red entre múltiples nodos.
- Objetivo: Maximizar la cantidad de peticiones atendidas.
- Implementaciones:
  - Balanceo por DNS: Distribuye tráfico mediante DNS Round Robin.
  - Balanceo a Nivel de Aplicación (Proxy Inverso): Distribuye tráfico basado en contenido HTTP/HTTPS.
  - Balanceo a Nivel IP: Utiliza herramientas como Linux Virtual Server (LVS).

Clústeres de Alta Disponibilidad (Failover Clusters):

- Descripción: Garantizan que un servicio permanezca disponible incluso si uno o más nodos fallan.
- Objetivo: Mantener la prestación y consistencia del servicio sin interrupciones.
- Tipos:

- Activo ⇔ Pasivo:
  - Activo: Maneja las solicitudes.
  - Pasivo: En espera para asumir funciones si el activo falla.
  - Ventajas: Configuración simple.
  - Desventajas: Uso ineficiente de recursos, costos duplicados.
- Activo ⇔ Activo:
  - Descripción: Todos los nodos manejan solicitudes simultáneamente.
  - Ventajas: Alto rendimiento, mejor tolerancia a fallos.
  - Desventajas: Configuración compleja, mayor latencia.

Clústeres de Alto Rendimiento (High Performance Clusters - HPC):

- Descripción: Nodos que trabajan conjuntamente en tareas de cálculo intensivo.
- Objetivo: Maximizar el rendimiento y la capacidad de cálculo.
- Usos:
  - Simulaciones físicas.
  - Bioinformática.
  - Predicción climática.
  - Inteligencia Artificial.
  - Análisis de Big Data.

3. Sharding

Definición:

- Sharding: Técnica de dividir datos en partes más pequeñas (shards) y distribuirlas en múltiples servidores para mejorar la escalabilidad y rendimiento.

Tipos de Sharding:

- Horizontal:
  - Descripción: Divide los datos por filas o registros.
  - Ejemplo: Distribuir diferentes filas de una base de datos en distintos nodos.
  - Ventajas: Mayor escalabilidad y balanceo de carga.
  - Desventajas: Consultas que abarcan múltiples shards pueden ser lentas.
- Vertical:
  - Descripción: Divide los datos por columnas o funciones.
  - Ejemplo: Distribuir diferentes columnas de una base de datos en distintos nodos.
  - Ventajas: Optimización basada en diferentes funciones.
  - Desventajas: Limitada escalabilidad y mayor complejidad en consultas.

4. Persistencia de Conexiones en Balanceo de Carga

Necesidad:

- Mantener la información de sesiones para aplicaciones que requieren estado, ya que HTTP/HTTPS son protocolos sin estado.

Métodos:

- Session Replication:
  - Descripción: Replica la información de las sesiones en todos los nodos.
  - Ventajas: Sencillo de implementar.
  - Desventajas: Alto consumo de recursos.
- Sticky Cookies:
  - Descripción: Usa cookies con IDs únicos para redirigir solicitudes del cliente al mismo servidor backend.
  - Funcionamiento:
    1. Balanceador asigna una cookie única en la primera respuesta.
    2. Solicitudes posteriores incluyen la cookie.
    3. Balanceador usa la cookie para identificar el servidor correspondiente.
  - Ventajas: Mantiene la sesión del cliente en un único servidor.
  - Desventajas: Menor flexibilidad si el servidor falla; requiere manejo adecuado de cookies.

5. Implementaciones Específicas de Balanceo de Carga

Linux Virtual Server (LVS):

- Descripción: Marco de balanceo de carga de código abierto integrado en el kernel de Linux.
- Componentes:
  - IPVS (IP Virtual Server): Toma decisiones de balanceo a nivel IP.
  - KTCPVS (Kernel TCP Virtual Server): Maneja balanceo a nivel de transporte.
- Métodos de Implementación:
  - Virtual Server via NAT:
    - Usa NAT para redirigir el tráfico.
  - Virtual Server via IP-Tunneling:
    - Encapsula solicitudes en túneles IP hacia los servidores backend.
  - Virtual Server via Direct-Routing:
    - Balancea a nivel Ethernet sin encapsulación.
- Ventajas:
  - Alta disponibilidad.
  - Soporta múltiples protocolos TCP/UDP.
- Desventajas:
  - Configuración compleja.

Ejemplos de Balanceadores de Carga a Nivel de Aplicación:

- HAProxy: haproxy.org
- Varnish Cache: varnish-cache.org

- NginX: nginx.com
- Apache HTTP Server: httpd.apache.org (con módulos mod_proxy, mod_proxy_balancer)

6. Estrategias de Balanceo de Carga

- Round-Robin:
  - Descripción: Reparte las cargas de forma alterna entre servidores.
  - Ventajas: Equitativo y sencillo.
  - Supuestos: Todos los servidores tienen la misma capacidad y carga de trabajo similar.
- Weighted Round-Robin:
  - Descripción: Asigna pesos a los servidores según su capacidad y reparte las cargas proporcionalmente.
  - Ventajas: Mejor distribución basada en capacidades reales.
- Least Connection:
  - Descripción: Reparte las cargas a los servidores con menos conexiones activas.
  - Ventajas: Balance dinámico basado en la carga actual.
- Weighted Least Connection:
  - Descripción: Combina el número de conexiones con el peso asignado a cada servidor.
  - Ventajas: Optimiza la distribución considerando capacidad y carga actual.
- Asignación Estática:
  - Descripción: Asigna servidores de manera fija basándose en las IPs de origen.
  - Ventajas: Predecible y simple.
  - Desventajas: Menor flexibilidad y capacidad de balanceo dinámico.

7. Integridad y Consistencia en Alta Disponibilidad

Problemas en Configuraciones Activo ⇔ Activo:

- Acceso Simultáneo: Múltiples nodos accediendo a los mismos datos puede causar corrupción.

Soluciones:

- Control de Concurrencia: Bloqueos o versiones para gestionar acceso concurrente.
- Consenso Distribuido: Algoritmos como Paxos o Raft para coordinar operaciones.
- Transacciones: Agrupar operaciones para garantizar atomicidad.

Split-Brain:

- Descripción: Situación donde múltiples nodos actúan como activos debido a fallos de comunicación, causando acceso simultáneo a datos compartidos.
- Soluciones:
  - Fencing:
    - Quórum: Asegura que solo la mayoría de nodos accedan a los recursos.
    - STONITH (Shoot The Other Node In The Head): Aísla nodos defectuosos para prevenir acceso indebido.
- Heartbeat:
  - Descripción: Mecanismo de sondeo para monitorear el estado de los nodos.
  - Implementación: Redes dedicadas para heartbeat, evitando interferencias con tráfico normal.

8. Computación de Altas Prestaciones (High Performance Computing - HPC)

Definición:

- Clúster HPC: Conjunto de computadoras interconectadas que trabajan juntas como una sola unidad para resolver problemas de computación intensiva.

Componentes Clave:

- Nodos de Cálculo: Computadoras individuales que ejecutan partes del cálculo.
- Interconexión: Redes de alta velocidad y baja latencia, como InfiniBand.
- Almacenamiento Compartido: Almacenamiento centralizado para manejar grandes volúmenes de datos.

Usos:

- Simulaciones físicas.
- Bioinformática.
- Predicción climática.
- Inteligencia Artificial.
- Análisis de Big Data.

Programación Paralela en HPC:

- Modelos de Programación:
  - Memoria Compartida:
    - Ejemplos: OpenMP, MPICH.
    - Descripción: Todos los procesadores acceden a la misma memoria.
  - Memoria Distribuida:
    - Ejemplos: MPI (Message Passing Interface), MapReduce.
    - Descripción: Cada nodo tiene su propia memoria y se comunican entre sí mediante mensajes.
- MPI (Message Passing Interface):
  - Descripción: Protocolo para el intercambio de mensajes entre nodos.
  - Características: Alto nivel de abstracción, oculta detalles de hardware y software.
- MapReduce:
  - Descripción: Modelo de programación orientado al procesamiento de grandes volúmenes de datos.
  - Fases:
    - Map: Transforma el input en pares clave-valor.
    - Shuffle & Sort: Agrupa y ordena los pares por clave.

        - Reduce: Procesa y combina los valores para producir el resultado final.
    - Ejemplo: Calcular el salario promedio de empleados por departamento.

Hadoop:

- Descripción: Plataforma de software de código abierto para almacenar y procesar grandes conjuntos de datos distribuidos en clústeres.
- Componentes Principales:
    - HDFS (Hadoop Distributed File System):
        - Características: Almacenamiento distribuido con bloques grandes (64MB) y múltiples réplicas.
        - Estructura:
            - NameNode: Almacena metadatos y distribución de bloques.
            - DataNodes: Almacenan los bloques asignados.
    - Otros Componentes:
        - MapReduce: Procesamiento distribuido.
        - Spark: Procesamiento en tiempo real.
        - HBase: Base de datos NoSQL distribuida.
        - Mahout: Machine Learning distribuido.
        - Flume: Transferencia de datos estructurados/no estructurados.

TEMA 4

Clústeres de Computadoras

Objetivo:

- Mejorar rendimiento, disponibilidad y capacidad de procesamiento mediante la agrupación de múltiples computadores.

Tipos de Clústeres:

- Clústeres de Balanceo de Carga (Load Balancing Clusters):
    - Función: Distribuyen la carga de trabajo y tráfico de red entre múltiples nodos.
    - Ejemplos: LinuxHA.
- Clústeres de Alta Disponibilidad (High Availability Clusters):
    - Función: Garantizan que los servicios permanezcan operativos ante fallos.
    - Ejemplos: HAProxy.
- Clústeres de Altas Prestaciones (High Performance Clusters - HPC):
    - Función: Ejecutan tareas de cálculo intensivo de manera distribuida.
    - Usos: Simulaciones físicas, bioinformática, predicción climática, inteligencia artificial, análisis de Big Data.

4. Seguridad en Redes de Centros de Datos

Objetivo:

- Proteger la infraestructura y los datos tanto a nivel físico como lógico.

Componentes Principales:

- Acceso Remoto e Interconexión de Redes:
    - VPNs (Virtual Private Networks): Extienden redes privadas sobre redes públicas de manera segura.
        - Tipos de VPN:
            - Site-to-Site VPN (VPN Punto a Punto):
                - Descripción: Conecta redes locales completas entre sí.
                - Uso: Interconexión de sedes corporativas.
                - Características: Configuración compleja, requiere dispositivos de red configurados.
            - Remote Access VPN (VPN de Acceso Remoto):
                - Descripción: Permite a usuarios individuales conectarse de forma segura a una red corporativa.
                - Uso: Acceso seguro desde ubicaciones remotas.
                - Características: Configuración más sencilla, orientada a usuarios.
        - Tecnologías de VPN:
            - IPSec (Internet Protocol Security):
                - Funciones: Autenticación, integridad y confidencialidad a nivel IP.
                - Modos:
                    - Transporte: Protege la carga útil (datos de capa de transporte y superiores).
                    - Túnel: Protege todo el paquete IP.
                - Protocolos Relacionados: AH (Authentication Header), ESP (Encapsulating Security Payload).
            - WireGuard:
                - Características: Protocolo moderno, rápido y seguro.
                - Ventajas: Simplicidad, alta velocidad, fácil configuración.
            - OpenVPN:
                - Características: Basado en SSL/TLS, altamente configurable.
                - Ventajas: Alta seguridad, amplia compatibilidad.
- Firewalls:
    - Definición: Barreras de seguridad que controlan y filtran el tráfico de red.
    - Tipos:
        - Filtros de Paquetes (Stateless Firewall):
            - Descripción: Inspecciona cada paquete de forma aislada.
            - Ventajas: Sencillo y rápido.
            - Desventajas: No considera el estado de la conexión.
        - Filtros con Estado (Stateful Firewall):
            - Descripción: Mantiene un registro de las conexiones y toma decisiones basadas en el estado.

            - Ventajas: Mayor seguridad y control.
            - Desventajas: Más complejo y consume más recursos.
        - Filtros a Nivel de Aplicación (Proxy):
            - Descripción: Controla el tráfico de aplicaciones específicas.
            - Tipos:
                - Forward Proxy (Proxy Directo):
                    - Función: Intermediario entre usuarios y servidores externos.
                    - Ventajas: Oculta IPs de usuarios, control de acceso.
                    - Desventajas: Mayor complejidad.
                - Reverse Proxy (Proxy Inverso):
                    - Función: Intermediario entre servidores internos y clientes externos.
                    - Ventajas: Balanceo de carga, optimización web, mayor privacidad.
                    - Desventajas: Mayor complejidad y costos.
    - Topologías de Firewalls:
        - Cortafuegos Básicos de Borde:
            - Descripción: Un solo dispositivo conecta la red interna con la externa.
            - Ventajas: Simplicidad.
            - Desventajas: Punto único de fallo.
        - Host Oculto (Screened Host):
            - Descripción: Máquina bastión que ofrece servicios tanto internos como externos.
            - Ventajas: Aislamiento de servicios.
            - Desventajas: Potencialmente vulnerable desde el exterior.
        - Zona Desmilitarizada (DMZ):
            - Descripción: Red aislada para servicios externos.
            - Ventajas: Mayor seguridad, evita comprometer la red interna.
            - Topología con Doble Firewall: DMZ entre dos firewalls para mayor seguridad.
- IDS/IPS (Intrusion Detection/Prevention Systems):
    - IDS (Sistemas de Detección de Intrusiones):
        - Descripción: Monitorean redes o sistemas para detectar accesos no autorizados.
        - Tipos:
            - NIDS (Network IDS): Monitorea tráfico de red en busca de patrones de ataque.
                - Ejemplos: SNORT, Suricata.
            - HIDS (Host IDS): Monitorea actividades y configuraciones en dispositivos individuales.
                - Ejemplos: OSSEC, SAGAN.
    - IPS (Sistemas de Prevención de Intrusiones):
        - Descripción: Detectan y bloquean actividades maliciosas en tiempo real.
        - Tipos:
            - NIPS (Network IPS): Monitorea y previene ataques en el tráfico de red.
            - HIPS (Host IPS): Monitorea y previene actividades sospechosas en hosts individuales.
            - WIPS (Wireless IPS): Protege redes inalámbricas contra ataques específicos.

5. Sharding

Definición:

- Sharding: Técnica de dividir datos en partes más pequeñas (shards) y distribuirlas en múltiples servidores para mejorar escalabilidad y rendimiento.

Tipos de Sharding:

- Horizontal:
    - Descripción: Divide los datos por filas o registros.
    - Ejemplo: Distribuir diferentes filas de una base de datos en distintos nodos.
    - Ventajas: Mayor escalabilidad, balanceo de carga.
    - Desventajas: Consultas que abarcan múltiples shards pueden ser lentas.
- Vertical:
    - Descripción: Divide los datos por columnas o funciones.
    - Ejemplo: Distribuir diferentes columnas de una base de datos en distintos nodos.
    - Ventajas: Optimización basada en diferentes funciones.
    - Desventajas: Limitada escalabilidad, mayor complejidad en consultas.

Ventajas del Sharding:

- Escalabilidad: Más shards permiten agregar más nodos.
- Carga Distribuida: Repartir la carga reduce la presión sobre cada nodo.
- Seguridad: Distribuir datos puede reducir riesgos.

Desventajas del Sharding:

- Consultas Lentas: Operaciones que abarcan múltiples shards pueden ser más lentas.
- Complejidad: Mayor complejidad en la gestión y mantenimiento de datos distribuidos.

TEMA 5

1. Conceptos Fundamentales

- Abstracción:
    - Definición: Simplifica la complejidad del hardware, presentando una vista simplificada a las aplicaciones.
    - Ventajas: Permite que las aplicaciones funcionen sin conocer detalles específicos del hardware subyacente.
    - Ejemplos:
        - Sistemas de Ficheros: Gestionan cómo se almacenan y acceden los archivos sin exponer detalles físicos.
        - Procesos: Permiten que múltiples programas se ejecuten simultáneamente sin interferir entre sí.

        - Dispositivos de Entrada/Salida: Gestionan la comunicación con periféricos como teclados y discos duros.
- Analogía de Abstracción:
    - Coche: Las ruedas, frenos, motor y puertas son gestionados por el sistema operativo del coche, ocultando su complejidad al conductor.

2. Emulador vs. Simulador vs. Virtualización

- Emulador:
    - Definición: Permite ejecutar programas diseñados para una plataforma diferente.
    - Funcionamiento: Traduce constantemente las instrucciones del sistema invitado a la plataforma base.
    - Ejemplo: Ejecutar videojuegos de consolas en una PC.
- Simulador:
    - Definición: Recrea el comportamiento exacto de un sistema para imitar una realidad.
    - Uso: Formación, pruebas de sistemas, investigación científica.
- Virtualización:
    - Definición: Crea entornos independientes que permiten ejecutar múltiples sistemas operativos o aplicaciones en un mismo hardware.
    - Ventajas: Maximiza la eficiencia y el uso de recursos computacionales.
    - Ejemplo: Ejecutar Windows y Linux en un solo servidor físico mediante máquinas virtuales.

---

3. Hipervisor

- Definición: Software que crea y gestiona máquinas virtuales (VMs) en un equipo físico.
- Funciones Principales:
    - Gestión de VMs: Crear, iniciar, parar y eliminar máquinas virtuales.
    - Aislamiento: Mantiene las VMs separadas entre sí.
    - Asignación de Recursos: Distribuye CPU, memoria, almacenamiento y red entre las VMs.
    - Migración en Caliente: Mueve VMs entre hardware sin interrupción.
- Componentes:
    - Host (Anfitrión): Equipo físico que provee los recursos para las VMs.
    - Guest (Invitado): Sistema operativo y aplicaciones que se ejecutan en una VM, actuando como si tuvieran su propio hardware.

4. Tipos de Virtualización

1. Virtualización Completa:
    - Descripción: Crea múltiples sistemas operativos aislados sobre un único servidor físico.
    - Tipos:
        - Tipo 1 (Bare-Metal):
            - Instalación: Directamente sobre el hardware físico, reemplazando al sistema operativo.
            - Subtipos:
                - Micronúcleo: Funcionalidades básicas en el hipervisor; funciones adicionales en módulos externos.
                - Monolíticos: Hipervisor incluye todas las funcionalidades básicas y adicionales.
            - Ejemplos: VMware ESXi.
        - Tipo 2 (Hosted):
            - Instalación: Sobre un sistema operativo anfitrión.
            - Características:
                - Traducción Binaria: Traduce instrucciones críticas en tiempo real para ejecución segura.
                - Ejecución Directa: Ejecuta instrucciones seguras directamente sin traducción.
            - Ejemplos: VMware Workstation, VirtualBox.
2. Paravirtualización:
    - Descripción: El sistema operativo invitado es consciente de la virtualización y utiliza una API específica para comunicarse con el hipervisor.
    - Ventajas: Mayor rendimiento al evitar la traducción binaria.
    - Desventajas: Requiere modificaciones en el kernel del SO invitado.
3. Virtualización Asistida por Hardware:
    - Descripción: El hardware incluye extensiones que facilitan la ejecución de VMs.
    - Ventajas: Mejora rendimiento y eficiencia al permitir acceso directo a recursos físicos.
    - Tecnologías:
        - Intel VT-x / VT-d
        - AMD-V
4. Virtualización a Nivel de Sistema Operativo (Basada en Contenedores):
    - Descripción: Aísla aplicaciones dentro de un mismo sistema operativo, compartiendo el kernel del host.
    - Componentes Clave:
        - Chroot: Crea un entorno aislado cambiando el directorio raíz de un proceso.
        - Namespaces: Aíslan recursos como procesos, redes y archivos.
        - Cgroups: Limita y prioriza el uso de recursos (CPU, memoria, I/O) para cada contenedor.
    - Ejemplo: Docker.

5. Docker

- Definición: Plataforma de software que automatiza la creación y despliegue de aplicaciones dentro de contenedores livianos.
- Componentes Clave:
    - Imagen:
        - Descripción: Plantilla de solo lectura para crear contenedores.

        - Características: Jerarquía de capas, contiene código, dependencias y librerías necesarias.
        - Creación: A partir de un Dockerfile.
    - Contenedor:
        - Descripción: Instancia en ejecución de una imagen.
        - Características: Aislamiento de procesos y sistema de archivos, comparte el kernel del host.
    - Docker Engine: Gestiona la creación y ejecución de contenedores.
    - Docker Compose:
        - Descripción: Herramienta para definir y gestionar aplicaciones multi-contenedor mediante un archivo YAML.
        - Funciones: Definición de servicios, automatización del despliegue, manejo de redes y volúmenes.
- Virtualización vs. Docker:
    - Virtualización Completa:
        - Abstracción: Virtualiza el hardware del SO.
        - Uso: Varios SO en el mismo hardware.
        - Recursos: Mayor uso de recursos debido a múltiples SO.
    - Docker (Contenedores):
        - Abstracción: Virtualiza el SO para las aplicaciones.
        - Uso: Ejecuta múltiples aplicaciones aisladas en el mismo SO.
        - Recursos: Mejor uso de recursos, ejecución rápida, mayor portabilidad.
- Dockerfile:
    - Definición: Archivo con instrucciones para automatizar la creación de una imagen Docker.
    - Instrucciones Comunes:
        - FROM: Define la imagen base.
        - LABEL: Información sobre el creador.
        - ENV: Define variables de entorno.
        - WORKDIR: Establece el directorio de trabajo.
        - RUN: Ejecuta comandos durante la construcción.
        - ADD/COPY: Agrega o copia archivos al contenedor.
        - ENTRYPOINT: Define el comando por defecto al iniciar el contenedor.
        - VOLUME: Establece volúmenes a montar.
        - EXPOSE: Indica puertos en los que el contenedor escuchará.
        - CMD: Define valores predeterminados para comandos.

TEMA 6

1. Definición de Cloud Computing

- Cloud Computing: Modelo de computación que permite el acceso a recursos de TI bajo demanda a través de Internet, eliminando la necesidad de infraestructura física in situ.

2. Ventajas del Cloud Computing

1. Escalabilidad:
    - Descripción: Aumento o disminución de recursos según las necesidades.
    - Beneficio: Adaptación rápida a cambios en la demanda.
2. Reducción de Costes:
    - Eliminación de inversión en hardware: No se necesitan comprar servidores, redes o sistemas de almacenamiento.
    - Reducción de gastos operativos: Menores costos en mantenimiento, energía y personal técnico.
    - Pago por uso: Solo se paga por los recursos utilizados en cada momento.
3. Facilidad de Mantenimiento:
    - Las tareas de mantenimiento son gestionadas por los operadores de Cloud Computing.
4. Accesibilidad Global 24/7:
    - Descripción: Acceso a datos y aplicaciones desde cualquier dispositivo con conexión a Internet.
    - Beneficio: Facilita la colaboración entre equipos distribuidos geográficamente.

3. Inconvenientes del Cloud Computing

1. Dependencia de la Conectividad:
    - Descripción: Acceder a servicios en la nube requiere una conexión a Internet estable.
    - Impacto: Interrupciones en la conexión afectan la disponibilidad de servicios.
2. Problemas de Seguridad y Privacidad:
    - Descripción: Datos almacenados y controlados por el proveedor de servicios.
    - Riesgo: La seguridad y privacidad dependen exclusivamente del proveedor.
3. Dependencia de los Proveedores:
    - Descripción: Problemas en el proveedor afectan al cliente.
    - Desafío: Migrar entre proveedores puede ser complicado.

4. Modelos de Despliegue del Cloud Computing

1. Nube Pública:
    - Descripción: Infraestructura compartida con otros usuarios.
    - Ventajas:
        - Modelo de pago por uso.
        - Alta escalabilidad y flexibilidad.
        - Amplio catálogo de servicios (almacenamiento, cómputo, bases de datos).
    - Ejemplos: Amazon AWS, Microsoft Azure, Google Cloud Platform.
    - Casos de Uso:
        - Pruebas y desarrollo de aplicaciones.
        - Análisis de datos y almacenamiento de archivos.
2. Nube Privada:

- Descripción: Infraestructura dedicada a una sola organización.
- Ventajas:
  - Mayor control sobre seguridad y cumplimiento normativo.
- Tipos:
  - On-Premise: Gestionada internamente.
  - Gestionada: Gestionada por un proveedor externo.
- Casos de Uso:
  - Empresas con altos requisitos de seguridad (banca).
  - Aplicaciones críticas para el negocio.

3. Nube Híbrida:
   - Descripción: Combinación de nubes públicas y privadas.
   - Ventajas:
     - Flexibilidad para elegir la mejor opción para cada carga de trabajo.
   - Casos de Uso:
     - Retail (Pública para promociones, Privada para datos de clientes).
     - Banca (Pública para aplicaciones, Privada para datos de clientes).

Comparativa de Modelos de Despliegue:

Característica	Nube Pública	Nube Privada	Nube Híbrida
Control	Bajo	Alto	Medio
Coste	Variable	Fijo	Ambos
Escalabilidad	Alta	Media	Alta
Seguridad	Dependiente	Alta	Variable
Flexibilidad	Alta	Media	Alta

5. Modelos de Servicio del Cloud Computing

1. Infraestructura como Servicio (IaaS):
   - Definición: Alquiler bajo demanda de recursos informáticos básicos (servidores, almacenamiento, red).
   - Funcionamiento:
     - Proveedor: Ofrece hardware virtualizado (máquinas virtuales).
     - Usuario: Control total para instalar sistemas operativos y aplicaciones.
   - Ejemplos:
     - Máquinas Virtuales: Amazon EC2, Microsoft Azure Virtual Machines.
     - Almacenamiento: Amazon S3, Google Cloud Storage.
     - Redes: Amazon VPC, Google VPC.
   - Casos de Uso:
     - Control personalizado de la infraestructura.
     - Migración de aplicaciones existentes a la nube.
2. Plataforma como Servicio (PaaS):
   - Definición: Alquiler bajo demanda de entornos de desarrollo y despliegue de aplicaciones.
   - Funcionamiento:
     - Proveedor: Gestiona el entorno de desarrollo, sistemas operativos y middleware.
     - Usuario: Desarrolla y despliega aplicaciones sin gestionar la infraestructura subyacente.
   - Ejemplos:
     - Heroku, AWS Elastic Beanstalk, Microsoft Azure Apps.
   - Componentes Clave:
     - Máquinas Virtuales: Virtualización de recursos físicos.
     - Plataformas de Ejecución: Contenedores y orquestadores.
     - Servicios Gestionados: Bases de datos, cachés, mensajería.
   - Casos de Uso:
     - Desarrollo rápido de aplicaciones.
     - Implementación de entornos colaborativos.
3. Software como Servicio (SaaS):
   - Definición: Distribución de software en la nube donde las aplicaciones se alojan y gestionan por un proveedor.
   - Funcionamiento:
     - Proveedor: Gestiona el software, infraestructura y seguridad.
     - Usuario: Accede y utiliza las aplicaciones a través de Internet.
   - Ejemplos:
     - Microsoft 365, Google Workspace, Dropbox.
   - Características:
     - Arquitectura Multitenant: Una sola instancia sirve a múltiples clientes.
     - Distribución vía Internet: Acceso a través de navegadores sin instalaciones locales.
     - Seguridad y Gestión de Datos: Cifrado, cumplimiento normativo.
   - Casos de Uso:
     - Aplicaciones de productividad.
     - Almacenamiento y compartición de archivos.

6. Seguridad en la Nube

1. Amenazas y Riesgos:
   - Ciberataques:
     - Acceso No Autorizado: Intentos de acceder a datos sensibles.
     - Denegación de Servicio (DDoS): Sobrecargar servidores para inutilizarlos.
     - Ataques a APIs: Vulnerabilidades en las APIs que exponen datos.
   - Riesgos Específicos de la Nube:

        - Multitenancy: Fallos en el aislamiento pueden permitir acceso a datos de otros usuarios.
        - Pérdida o Fuga de Datos: Riesgo de que los datos sean perdidos o expuestos.
        - Dependencia de Terceros: Falta de control sobre servicios y estándares de seguridad del proveedor.
2. Medidas de Seguridad:
    - Cifrado de Datos:
        - En Tránsito y en Reposo: Protege datos durante la transmisión y almacenamiento.
        - Gestión de Claves: Manejo seguro de las claves de cifrado.
    - Gestión de Identidades y Accesos:
        - Autenticación Multifactor (MFA): Requiere múltiples formas de autenticación.
        - Roles y Permisos: Limita el acceso según el rol del usuario.
    - Auditorías y Monitoreo:
        - Auditorías Frecuentes: Revisar configuraciones y accesos.
        - Monitoreo Continuo: Supervisar el uso y acceso a servicios y datos.

7. Gestión de la Nube

1. Modelos de Precios:
    - Pago por Uso (On-demand):
        - Descripción: Pago basado en recursos utilizados.
        - Ventajas: Flexibilidad y escalabilidad inmediata.
        - Desventajas: Costoso a largo plazo.
    - Instancias Reservadas (Reserved Instances):
        - Descripción: Reserva de recursos a largo plazo (1-3 años) con descuentos.
        - Ventajas: Costos reducidos con compromiso de uso constante.
        - Desventajas: Menos flexibilidad.
    - Precios por Capacidad o Tamaño:
        - Descripción: Pago según la capacidad de la instancia (CPU, RAM).
        - Ventajas: Flexibilidad en elección de recursos.
        - Desventajas: Requiere ajuste preciso para evitar sobre/infrautilización.
    - Precios por Volumen o Uso Escalonado:
        - Descripción: Pago inversamente proporcional al uso (más uso, menor costo por unidad).
        - Ventajas: Económico para alto volumen de datos.
        - Desventajas: Costos elevados con bajo consumo.
    - Modelo de Precios Spot:
        - Descripción: Aprovecha instancias sobrantes a precios muy bajos.
        - Ventajas: Costos reducidos hasta un 90%.
        - Desventajas: Disponibilidad no garantizada.
2. Configuración y Automatización:
    - Infraestructura como Código (IaC):
        - Descripción: Gestión de infraestructura mediante archivos de configuración.
        - Beneficios: Replicación, versionado y auditoría de recursos.
    - Gestión de Versiones y Despliegue en IaC:
        - Control de Versiones: Permite revertir configuraciones problemáticas.
        - Automatización del Despliegue: Asegura configuraciones exactas y consistentes.
3. Contingencia y Recuperación ante Desastres:
    - Planes de Recuperación:
        - Incluyen: Respaldo de datos, replicación de sistemas críticos, procedimientos de recuperación.
    - Respaldo y Replicación de Datos:
        - Backups Regulares: Minimiza riesgo de pérdida de información.
        - Replicación en Múltiples Regiones: Asegura disponibilidad y recuperación.
    - Pruebas de Resiliencia:
        - Pruebas de Recuperación: Simulaciones de fallos para evaluar y ajustar planes.
        - Resiliencia a Fallos: Implementación de sistemas redundantes para continuidad.

TEMA 7

1. Introducción a la Automatización

- Definición:
    - Automatización en Centros de Datos: Surgen por la necesidad de gestionar automáticamente (o semi-automáticamente) las infraestructuras y servicios de los centros de datos.
- Objetivos de la Automatización:
    - Reducción de Errores: Minimiza los errores manuales.
    - Escalabilidad: Facilita la administración de un gran número de servidores.
    - Rapidez y Eficiencia: Acelera el despliegue de servicios y actualizaciones.
    - Simplificación de Tareas de Administración: Hace más sencillas las labores de gestión.
- Beneficios Adicionales:
    - Información de Configuración Centralizada: Centraliza la gestión de configuraciones.
    - Tareas Identificadas y Documentadas: Favorece buenas prácticas en la administración de sistemas mediante:
        - Modularidad y Reutilización de Configuraciones.
        - Uso de Sistemas de Control de Versiones.
        - Mejora y Despliegue Continuo: Posibilidad de realizar testing continuo.

2. Conceptos Clave

- Orquestación de Sistemas:
    - Definición: Coordinación de la configuración y despliegue de varios componentes en un sistema.

    - Importancia: Fundamental en entornos que utilizan virtualización, contenerización o cloud computing.
- Aprovisionamiento de Sistemas:
    - Definición: Conjunto de procesos para preparar una infraestructura para prestar el servicio esperado.
    - Tareas Incluidas:
        - Instalación de componentes/servicios necesarios.
        - Configuración y puesta en marcha.
- Infraestructura como Código (IaC):
    - Definición: Estrategia que permite describir la infraestructura mediante código automatizable.
    - Características:
        - Código comprensible tanto por administradores como por herramientas de automatización.
- DevOps:
    - Definición: Integración del desarrollo de software (Dev) y operaciones/administración (Ops) para agilizar despliegues y mejoras continuas.
    - Uso de IaC y Metodologías Ágiles.

3. Aproximaciones en la Automatización

- Enfoque Declarativo:
    - Descripción: Define el estado final deseado de la infraestructura.
    - Características:
        - El lenguaje describe cómo debe ser el estado final.
        - La herramienta gestiona automáticamente las acciones necesarias para alcanzar ese estado.
    - Ventajas:
        - Gestión automática del estado.
        - Menos control sobre el flujo de ejecución.
    - Herramientas Comunes: Terraform, Puppet, Kubernetes.
    - Ejemplo:
        - Qué quiero lograr: Que Apache esté en ejecución y tres instancias de una aplicación en contenedores.
        - Cómo lo quiero lograr: Definir en el código el estado deseado y dejar que la herramienta lo implemente.
- Enfoque Procedimental:
    - Descripción: Define las tareas específicas y el orden en que deben ejecutarse.
    - Características:
        - Control granular sobre el flujo de ejecución.
        - El administrador define los pasos para alcanzar el estado deseado.
    - Ventajas:
        - Control completo sobre la secuencia de tareas.
        - Ideal cuando el orden de ejecución es crucial.
    - Herramientas Comunes: Ansible, Bash scripting.
    - Ejemplo:
        - Qué quiero lograr: Que Apache esté en ejecución y tres instancias de una aplicación en contenedores.
        - Cómo lo quiero lograr: Especificar cada paso (instalar Apache, configurar archivos, iniciar servicios) en el orden correcto.
- Comparativa Declarativo vs Procedimental:

Característica	Declarativo	Procedimental
Propósito	Mantener el sistema en un estado final automáticamente	Ejecutar tareas en una secuencia específica
Herramientas	Terraform, Puppet, Kubernetes	Ansible, Bash scripting
Control	Limitado: la herramienta decide los pasos	Completo: el administrador define el orden
Uso Ideal	Importa el estado final	El flujo de tareas es crucial
Idempotencia	Sí: asegura el estado final siempre	Sí: tareas se ejecutan sólo si es necesario

---

4. Herramientas de Automatización

a. Puppet

- Tipo: Procedural.
- Lenguaje de Configuración: DSL basado en Ruby.
- Arquitectura: Cliente-Servidor con agentes instalados en los equipos gestionados.
- Metodología: Pull (los clientes consultan al servidor periódicamente).
- Características:
    - Utiliza manifiestos que describen recursos y su estado.
    - Permite herencia y modularidad.
    - Idempotencia: Las tareas se ejecutan solo si el estado actual no coincide con la configuración deseada.

b. Ansible

- Tipo: Declarativo.
- Lenguaje de Configuración: YAML.
- Arquitectura: Agentless, utiliza SSH para conectarse a los nodos gestionados.
- Metodología: Push (el servidor inicia las tareas en los nodos).
- Características:
    - Usa playbooks que describen tareas y su estado.
    - Reutilización de tareas mediante roles.
    - Idempotencia: Las tareas se ejecutan solo si el estado actual no coincide con la configuración deseada.
- Componentes Clave:
    - Nodos:
        - Nodo de Control: Ejecuta comandos Ansible.

        - Nodos Gestionados: Reciben y ejecutan las instrucciones.
    - Inventory: Lista de nodos gestionados.
    - Playbooks: Archivos YAML que describen las tareas a realizar.
    - Roles: Colecciones reutilizables de tareas.
    - Módulos: Scripts que ejecutan tareas específicas.
    - Variables y Templates: Para configuraciones dinámicas.

c. Terraform

- Tipo: Declarativo.
- Lenguaje de Configuración: HCL (HashiCorp Configuration Language).
- Arquitectura: Gestiona el estado de la infraestructura en un backend.
- Características:
    - Define y provisiona infraestructura en múltiples proveedores de nube.
    - Modularidad mediante el uso de módulos.
    - Idempotencia: Las tareas se ejecutan solo si el estado actual no coincide con la configuración deseada.