

# GUÍA COMPLETA DE ADMINISTRACIÓN DE SISTEMAS LINUX

## UNIDAD 1: Realización de tareas básicas de administración del sistema

### 1.1 Introducción y filosofía de Linux para administradores de sistemas

#### Historia y filosofía de Linux

Linux es un sistema operativo de código abierto (Open Source) basado en Unix, creado por Linus Torvalds en 1991. La filosofía del software libre permite que cualquier persona pueda ver, modificar y distribuir el código fuente, fomentando la colaboración global y la transparencia.

#### Conceptos clave:

- **Open Source:** El código fuente está disponible públicamente
- **Código Libre:** Libertad de usar, estudiar, modificar y distribuir
- **Distribuciones (Distros):** Versiones de Linux adaptadas para diferentes propósitos

#### Principales familias de distribuciones:

- **Debian-based:** Ubuntu, Linux Mint, Kali Linux
- **RHEL-based:** CentOS, Rocky Linux, AlmaLinux, Fedora
- **SUSE-based:** openSUSE
- **Arch-based:** Manjaro, EndeavourOS

Cada distribución tiene su propósito específico: servidores empresariales, estaciones de trabajo, seguridad informática, sistemas embebidos, etc.

### 1.2 Instalación del servidor Linux

#### Significado de conceptos clave:

**Máquina Virtual (VM):** Entorno virtualizado que simula un computador completo dentro de otro sistema operativo, permitiendo ejecutar Linux sin afectar el sistema principal.

### Modos de red en virtualización:

- **NAT (Network Address Translation):** La VM comparte la IP del host y puede acceder a Internet, pero no es visible desde otras máquinas de la red
- **Bridge (Puente):** La VM obtiene su propia IP en la red local, como si fuera un equipo físico independiente
- **Host-Only:** La VM solo puede comunicarse con el host, sin acceso a Internet ni a la red externa

**Snapshots:** Capturas del estado completo de una VM en un momento específico, permitiendo restaurar el sistema a ese punto exacto si algo falla.

### Particiones esenciales en Linux:

- **/ (root):** Partición principal del sistema, contiene todo el árbol de directorios
- **/boot:** Archivos necesarios para el arranque del sistema (kernel, GRUB)
- **/home:** Datos personales de los usuarios
- **/swap:** Memoria de intercambio (RAM virtual en disco)
- **/var:** Datos variables (logs, bases de datos, correos)
- **/tmp:** Archivos temporales

## 1.3 Comandos Básicos

### Navegación y manipulación de archivos:

- **pwd** (Print Working Directory): Muestra la ruta del directorio actual
- **cd** (Change Directory): Cambia de directorio
- **ls** (List): Lista archivos y directorios
- **touch:** Crea archivos vacíos o actualiza fecha de modificación
- **echo:** Imprime texto en pantalla o redirige a archivos
- **cat:** Muestra contenido de archivos
- **tac:** Muestra contenido de archivos en orden inverso
- **mkdir:** Crea directorios
- **cp:** Copia archivos o directorios
- **mv:** Mueve o renombra archivos
- **rm:** Elimina archivos o directorios
- **rm -rf:** Elimina recursivamente directorios y su contenido (¡usar con precaución!)

### Comandos de información y búsqueda:

- **man:** Manual de comandos (documentación completa)

- **apropos:** Busca comandos por palabra clave
- **which:** Muestra la ubicación de un comando ejecutable
- **history:** Muestra historial de comandos ejecutados
- **locate:** Busca archivos en una base de datos indexada
- **uname:** Información del kernel del sistema
- **lsb\_release -d:** Información de la distribución Linux

#### Comandos avanzados:

- **grep:** Filtra y busca patrones en texto
- **sudo:** Ejecuta comandos con privilegios de superusuario
- **awk:** Procesa y analiza texto estructurado
- **alias:** Crea atajos para comandos (ejemplo: `alias ll='ls -la'`)

## 1.4 Carpetas esenciales del sistema

#### Estructura del sistema de archivos Linux (FHS - Filesystem Hierarchy Standard):

- **/bin:** Binarios esenciales del sistema (comandos básicos como ls, cp, cat)
- **/etc:** Archivos de configuración del sistema y aplicaciones
- **/dev:** Archivos de dispositivos (hardware representado como archivos)
- **/var:** Datos variables (logs, colas de impresión, bases de datos)
- **/tmp:** Archivos temporales (se limpian al reiniciar)
- **/lib:** Librerías compartidas esenciales para el sistema
- **/home:** Directorios personales de usuarios
- **/root:** Directorio personal del usuario root (administrador)
- **/boot:** Archivos del gestor de arranque y kernel
- **/opt:** Software opcional de terceros
- **/usr:** Programas y datos de usuario (aplicaciones, documentación)
- **/proc:** Sistema de archivos virtual con información de procesos
- **/sys:** Sistema de archivos virtual con información del kernel y hardware
- **/mnt y /media:** Puntos de montaje para dispositivos externos

## 1.5 Configurando Redes

#### Conceptos de red en Linux:

**Dirección IP:** Identificador único de un dispositivo en la red (ejemplo: 192.168.1.100)

**Máscara de subred:** Define qué porción de la IP identifica la red y qué porción identifica el host (ejemplo: 255.255.255.0 o /24)

**Gateway:** Puerta de enlace que conecta la red local con otras redes (Internet)

**DNS:** Servidor que traduce nombres de dominio a direcciones IP

#### **Herramientas de configuración de red:**

- **ifconfig:** Comando tradicional para ver y configurar interfaces (obsoleto pero aún usado)
- **ip addr:** Comando moderno para gestión de direcciones IP
- **nmtui:** Interfaz de texto interactiva para NetworkManager (Debian/Ubuntu)
- **nmcli:** Interfaz de línea de comandos para NetworkManager
- **netplan:** Sistema de configuración de red en Ubuntu moderno (archivos YAML)

## **1.6 Trabajando con archivos de texto**

#### **Editores de texto en terminal:**

- **vi/vim:** Editor modal potente y omnipresente en sistemas Unix/Linux
  - Modo comando: navegación y operaciones
  - Modo inserción: edición de texto
  - Modo visual: selección de texto
- **nano:** Editor simple e intuitivo, ideal para principiantes
  - Comandos mostrados en pantalla
  - Ctrl+O para guardar, Ctrl+X para salir

Los editores de texto en terminal son esenciales porque los servidores Linux a menudo no tienen interfaz gráfica.

## **1.7 Gestión de usuarios y grupos**

#### **Conceptos de usuarios y grupos:**

**Usuario:** Cuenta individual que permite acceso al sistema con permisos específicos

**Grupo:** Colección de usuarios que comparten permisos comunes

#### **Operaciones principales:**

- **Crear usuario:** Añadir nuevas cuentas al sistema
- **Crear grupo:** Definir conjuntos de permisos compartidos
- **Añadir usuarios a grupos:** Otorgar permisos grupales a usuarios específicos
- **Grupo sudo/wheel:** Grupo especial que permite ejecutar comandos como root
- **Eliminar usuarios/grupos:** Remover cuentas y permisos del sistema

### Archivos importantes:

- `/etc/passwd`: Base de datos de usuarios
- `/etc/group`: Base de datos de grupos
- `/etc/shadow`: Contraseñas cifradas

## 1.8 Configurando Permisos

### Sistema de permisos Unix:

#### Tres tipos de permisos:

- **Lectura (r/4)**: Ver contenido de archivos o listar directorios
- **Escritura (w/2)**: Modificar archivos o crear/eliminar en directorios
- **Ejecución (x/1)**: Ejecutar archivos o acceder a directorios

#### Tres niveles de permisos:

- **Usuario (u)**: Propietario del archivo
- **Grupo (g)**: Grupo propietario del archivo
- **Otros (o)**: Resto de usuarios del sistema

#### Métodos de modificación:

- **Notación numérica**: `rwX = 7`, `rw- = 6`, `r-X = 5`, `r-- = 4`
- **Notación simbólica**: `u+x` (añadir ejecución al usuario), `g-w` (quitar escritura al grupo)

#### Cambio de propiedad:

- Modificar usuario propietario
- Modificar grupo propietario

## UNIDAD 2: Administración avanzada del sistema

### 2.1 Detección de hardware

#### Lista de hardware en Linux:

El sistema Linux representa el hardware como archivos y proporciona múltiples comandos para inspeccionar los componentes del sistema.

#### Comandos principales:

- **lspci:** Lista dispositivos conectados al bus PCI (tarjetas de red, video, etc.)
- **lsusb:** Lista dispositivos USB conectados
- **lsblk:** Muestra información de dispositivos de bloques (discos, particiones)
- **lshw:** Listado detallado de todo el hardware
- **dmidecode:** Información de BIOS y hardware del sistema
- **hdparm:** Información y parámetros de discos duros
- **sensors:** Temperatura y voltajes de sensores hardware
- **inxi:** Resumen completo de información del sistema
- **hwinfo:** Información exhaustiva de hardware

## 2.2 Gestión de software

### Gestión de paquetes:

Los sistemas Linux utilizan gestores de paquetes para instalar, actualizar y eliminar software.

### Familias principales:

- **Debian/Ubuntu:** apt, apt-get, dpkg
- **RHEL/CentOS:** yum, dnf, rpm
- **SUSE:** zypper
- **Arch:** pacman

### Operaciones fundamentales:

- **Instalar software:** Añadir aplicaciones desde repositorios oficiales o de terceros
- **Eliminar software:** Desinstalar aplicaciones y sus dependencias
- **Update:** Actualizar lista de paquetes disponibles
- **Upgrade:** Actualizar paquetes instalados a versiones más recientes
- **Ver repositorios:** Listar fuentes de software configuradas
- **Añadir repositorio:** Agregar nuevas fuentes de software (PPA, repos externos)

## 2.3 Gestión de procesos

### Conceptos de procesos:

Un proceso es un programa en ejecución con recursos asignados (CPU, memoria, archivos).

### Operaciones con procesos:

- **Visualizar procesos:** Ver todos los procesos activos con ps, top, htop
- **Terminar procesos:** Detener procesos con kill, killall, pkill
- **Cambiar prioridad:** Ajustar el nice value (prioridad de CPU) con nice y renice
- **Procesos en segundo plano:** Ejecutar comandos con & o mover con bg/fg

#### **Estados de procesos:**

- Running (R): En ejecución
- Sleeping (S): En espera
- Stopped (T): Detenido
- Zombie (Z): Terminado pero no liberado

#### **Señales comunes:**

- SIGTERM (15): Terminación ordenada
- SIGKILL (9): Terminación forzada
- SIGHUP (1): Recargar configuración

## **2.4 Programación de tareas**

#### **Automatización de tareas:**

**Cron:** Planificador de tareas periódicas (diarias, semanales, mensuales)

- Utiliza el archivo crontab para definir tareas
- Formato: minuto hora día mes día\_semana comando
- Ideal para backups, limpieza, actualizaciones automáticas

**AT:** Planificador de tareas puntuales (una sola vez en el futuro)

- Ejecuta comandos en un momento específico
- Útil para tareas no recurrentes

## **2.5 Logs del sistema en Linux**

#### **Sistema de registro (logging):**

Linux registra eventos del sistema y aplicaciones en archivos de log ubicados principalmente en /var/log/.

#### **Tipos de logs:**

- **Logs del sistema:** Eventos del kernel y sistema operativo (/var/log/syslog, /var/log/messages)
- **Logs de aplicaciones:** Eventos de servicios específicos (Apache, MySQL, SSH)
- **Logs de eventos:** Autenticación, inicio de sesión, errores (/var/log/auth.log)

#### Herramientas:

- **journalctl:** Consulta logs del systemd journal
- **dmesg:** Mensajes del kernel
- **tail -f:** Monitoreo en tiempo real de logs

## 2.6 Gestión del almacenamiento

#### Sistema de archivos:

Linux soporta múltiples sistemas de archivos: ext4, xfs, btrfs, zfs, ntfs, fat32.

#### Gestión de particiones:

- **Crear particiones:** Dividir discos en secciones lógicas con fdisk, parted, gparted
- **Formatear:** Aplicar sistema de archivos a partición (mkfs)
- **Montar:** Hacer accesible una partición en el árbol de directorios (mount)
- **Desmontar:** Desconectar partición del sistema (umount)
- **Eliminar particiones:** Remover divisiones del disco

**Archivo /etc/fstab:** Define montajes automáticos al arranque

#### Conceptos:

- **UUID:** Identificador único de particiones
- **Punto de montaje:** Directorio donde se accede a la partición
- **LVM:** Logical Volume Manager, gestión flexible de almacenamiento

## 2.7 Administración de servicios

#### Gestión de servicios con systemd:

Systemd es el sistema de inicio y gestor de servicios moderno en Linux.

#### Operaciones principales:

- **Ver servicios en ejecución:** Listar todos los servicios activos



- **Ver puertos usados:** Identificar qué puertos de red utiliza cada servicio
- **Habilitar servicio:** Configurar inicio automático al arranque
- **Iniciar servicio:** Poner en marcha un servicio
- **Ver estado:** Verificar si servicio está activo y su salud
- **Detener servicio:** Apagar un servicio en ejecución
- **Reiniciar servicio:** Detener y volver a iniciar (útil tras cambios de configuración)

**Comandos systemctl:** systemctl status, start, stop, restart, enable, disable

## UNIDAD 3: Configuración avanzada del sistema

### 3.1 GRUB2

**GRUB (GRand Unified Bootloader):**

Gestor de arranque que permite elegir sistema operativo y pasar parámetros al kernel.

**Funciones principales:**

- **Editar menú de GRUB:** Modificar opciones de arranque, timeout, kernel por defecto
- **Recuperar contraseña de root:** Proceso para resetear contraseña de administrador olvidada
  - Interrumpir arranque en GRUB
  - Editar parámetros del kernel
  - Iniciar en modo de emergencia
  - Remontar sistema de archivos con permisos de escritura
  - Cambiar contraseña con passwd
  - Reiniciar sistema

**Archivos importantes:**

- /etc/default/grub: Configuración principal
- /boot/grub/grub.cfg: Archivo generado (no editar directamente)

### 3.2 Shell Script

**Scripting en Bash:**

Los scripts de shell automatizan tareas mediante secuencias de comandos Unix/Linux.

**Elementos fundamentales:**

- **Shebang:** Primera línea que indica el intérprete (#!/bin/bash)
- **Variables:** Almacenan datos (nombre="Juan", edad=25)
- **Condiciones:** Estructuras if/else para tomar decisiones
- **Bucles:** for, while para tareas repetitivas
- **Funciones:** Bloques de código reutilizables

#### Operadores de condición:

- -eq: igual
- -ne: no igual
- -gt: mayor que
- -lt: menor que
- -f: archivo existe
- -d: directorio existe

### 3.3 Configuración de SSH

#### SSH (Secure Shell):

Protocolo para acceso remoto seguro a sistemas Linux con cifrado de comunicaciones.

#### Componentes:

- **Servidor SSH:** Servicio que escucha conexiones (sshd)
- **Cliente SSH:** Herramienta para conectar a servidores remotos
- **Puerto por defecto:** 22 (TCP)

#### Funcionalidades:

- **Conexión remota:** Acceder a terminal de servidor remoto
- **SCP (Secure Copy):** Copiar archivos de forma segura entre sistemas
- **Llaves SSH:** Autenticación automática sin contraseña mediante par de llaves pública/privada

#### Archivos de configuración:

- /etc/ssh/sshd\_config: Configuración del servidor
- ~/.ssh/: Directorio de llaves del usuario

### 3.4 Configuración de Otros Servicios

#### Servicios de red fundamentales:

### **DNS (Domain Name System):**

- Traduce nombres de dominio a direcciones IP
- Servidor: BIND9, dnsmasq, PowerDNS
- Resuelve consultas de nombres en la red

### **DHCP (Dynamic Host Configuration Protocol):**

- Asigna automáticamente direcciones IP a clientes
- Servidor: isc-dhcp-server, dnsmasq
- Configura IP, máscara, gateway y DNS

### **FTP (File Transfer Protocol):**

- Protocolo para transferencia de archivos
- Servidores: vsftpd, ProFTPD
- Alternativas seguras: SFTP, FTPS

## **UNIDAD 4: Servicios de aplicaciones**

### **4.1 Administración de servicios HTTP**

#### **Servidores web:**

##### **Apache2:**

- Servidor web más popular y versátil
- Módulos extensibles
- Configuración basada en archivos .conf
- Virtual Hosts: múltiples sitios en un servidor

##### **NGINX:**

- Servidor web moderno y eficiente
- Excelente rendimiento
- Proxy inverso y balanceador de carga
- Configuración basada en bloques server

#### **Operaciones comunes:**

- Instalación del servicio
- Publicación de páginas web
- Cambio de puerto de escucha (80, 443, 8080)

- Creación de hosts virtuales
- Configuración SSL/TLS

## 4.2 Administración y configuración de WordPress

### WordPress:

- Sistema de gestión de contenidos (CMS)
- Basado en PHP y MySQL/MariaDB
- Requiere servidor web (Apache/NGINX)

### Componentes necesarios:

- Servidor web
- PHP y extensiones
- Base de datos MySQL/MariaDB
- Archivos de WordPress

## 4.3 Administración y configuración de Correos

### Servidor de correo:

### Componentes del sistema de correo:

- **MTA (Mail Transfer Agent):** Envía/recibe correos (Postfix, Sendmail)
- **MDA (Mail Delivery Agent):** Entrega correos a buzones (Dovecot)
- **Webmail:** Interfaz web para correo (Roundcube, Squirrelmail)

### Protocolos:

- SMTP: Envío de correo (puerto 25, 587)
- POP3: Descarga de correo (puerto 110)
- IMAP: Acceso a correo (puerto 143)

## 4.4 Configuración de servidor de impresión

### CUPS (Common Unix Printing System):

Sistema de impresión estándar en Linux que permite compartir impresoras en red.

### Características:

- Interfaz web de administración (puerto 631)

- Soporte para múltiples protocolos
- Drivers para impresoras diversas

**cups-pdf:**

- Impresora virtual que genera archivos PDF
- No requiere hardware físico
- Útil para documentación y pruebas

## 4.5 Configuración de servidor Web Proxy y repositorios

**Apt-Cacher-NG:**

Proxy caché para repositorios de paquetes Linux (APT).

**Beneficios:**

- Reduce ancho de banda
- Acelera actualizaciones
- Centraliza descargas en red local
- Cachea paquetes .deb

## 4.6 Configuración de Rutas Estáticas y Rutas Por Defecto

**Enrutamiento en Linux:**

**Ruta por defecto (Default Gateway):**

- Destino para paquetes sin ruta específica
- Normalmente apunta al router

**Rutas estáticas:**

- Rutas manuales para redes específicas
- Persisten tras reinicio si se configuran correctamente
- Comando: `ip route add`

**Tabla de enrutamiento:**

- Muestra todas las rutas configuradas
- Comando: `ip route show` o `route -n`

# UNIDAD 5: Alta Disponibilidad y Clustering

## 5.1 Conceptos de Cluster/HA

### Alta Disponibilidad (High Availability):

Capacidad de un sistema para permanecer operativo minimizando tiempo de inactividad.

### Conceptos fundamentales:

#### Activo/Activo:

- Múltiples nodos procesando simultáneamente
- Balanceo de carga entre nodos
- Aprovecha todos los recursos

#### Activo/Pasivo:

- Un nodo activo, otros en espera
- Failover cuando el principal falla
- Menor utilización de recursos

#### Prioridad:

- Orden de preferencia entre nodos
- Determina qué nodo toma control

#### Monitorización:

- Vigilancia constante de salud de nodos
- Detección de fallos

#### STONITH (Shoot The Other Node In The Head):

- Mecanismo para apagar nodo problemático
- Previene corrupción de datos

#### Split Brain:

- Ambos nodos creen ser el principal
- Solución: fencing, quorum

#### Fencing:

- Aislamiento de nodo fallido
- Previene acceso a recursos compartidos

**Recuperación:**

- Proceso de restaurar servicio tras fallo

**Failover:**

- Transferencia automática de servicios a nodo de respaldo

**Balanceo de Carga:**

- Distribución de trabajo entre múltiples nodos

## 5.2 Sincronización de archivos en Linux

**Rsync:**

Herramienta para sincronizar archivos y directorios entre sistemas.

**Características:**

- Transferencia incremental (solo cambios)
- Compresión durante transferencia
- Preserva permisos y atributos
- Funciona sobre SSH para seguridad

**Casos de uso:**

- Backups
- Mirrors de servidores
- Distribución de contenido

## 5.3 Soluciones de Cluster en Linux

**Heartbeat:**

- Sistema de monitoreo entre nodos
- Detecta fallos mediante pulsos regulares
- Usado en distribuciones Debian

**Pacemaker/Corosync:**

- Suite completa de clustering
- Corosync: comunicación entre nodos
- Pacemaker: gestión de recursos
- Estándar en RHEL/CentOS

#### **Keepalived:**

- IP virtual flotante (VIP)
- Protocolo VRRP
- Simple y efectivo para HA básica

#### **HAProxy:**

- Balanceador de carga HTTP/TCP
- Alta performance
- Health checks automáticos

## **5.4 Aplicaciones Prácticas de un Cluster**

#### **Escenarios comunes:**

##### **Base de Datos:**

- MySQL/MariaDB en cluster
- PostgreSQL con replicación
- Failover automático

##### **Balanceador de Carga:**

- Distribuir tráfico web
- HAProxy o NGINX
- Múltiples backends

##### **Servidores Web:**

- Apache/NGINX en HA
- Contenido sincronizado
- IP flotante para acceso



# UNIDAD 6: Seguridad en Linux

## 6.1 Comprensión de la seguridad de Linux

### Conceptos de criptografía:

#### Cifrado con GPG (GNU Privacy Guard):

- Cifrado asimétrico de archivos
- Par de llaves pública/privada
- Firma y verificación de documentos

#### Firmas digitales:

- Garantizan autenticidad
- Verifican integridad
- No repudio

#### Certificados digitales:

- Identidad en comunicaciones
- Emitidos por Autoridades Certificadoras (CA)
- Usados en SSL/TLS

#### Funciones Hash:

##### MD5 (Message Digest 5):

- Hash de 128 bits
- Obsoleto por vulnerabilidades
- Aún usado para checksums

##### SHA1 (Secure Hash Algorithm 1):

- Hash de 160 bits
- Vulnerabilidades conocidas
- En desuso

##### SHA256:

- Hash de 256 bits
- Actualmente seguro
- Estándar moderno

### **Usos de hash:**

- Verificación de integridad de archivos
- Almacenamiento seguro de contraseñas
- Firmas digitales

### **OpenSSL:**

- Librería de criptografía
- Generación de llaves y certificados
- Certificados autofirmados para redes locales

### **Detección de amenazas:**

#### **Vulnerabilidades del kernel:**

- Exploits de escalación de privilegios
- Parches de seguridad regulares

#### **Rootkits:**

- Malware que oculta su presencia
- Modifica sistema a nivel profundo

#### **Herramientas de detección:**

- **rkhunter**: Rootkit Hunter
- **chkrootkit**: Check Rootkit

## **6.2 Mejora de la seguridad de Linux**

### **Módulos PAM (Pluggable Authentication Modules):**

Sistema modular de autenticación que permite configurar políticas de seguridad.

#### **Funcionalidades:**

- Bloqueo por intentos fallidos (pam\_tally2, pam\_faillock)
- Autenticación de dos factores (2FA)
- Bloqueo por IP de origen
- Requisitos de contraseñas fuertes
- Límites de sesión

### **SELinux (Security-Enhanced Linux):**

- Sistema de control de acceso obligatorio
- Políticas de seguridad granulares
- Estándar en RHEL/CentOS
- Modos: Enforcing, Permissive, Disabled

#### **AppArmor:**

- Alternativa a SELinux
- Más simple de configurar
- Estándar en Ubuntu/SUSE
- Perfiles por aplicación

## **6.3 Firewall de Linux**

#### **Cortafuegos (Firewall):**

Control de tráfico de red entrante y saliente mediante reglas.

#### **IPtables/Nftables:**

- IPtables: firewall tradicional del kernel Linux
- Nftables: sucesor moderno de iptables
- Control granular de paquetes
- Tablas: filter, nat, mangle

#### **Operaciones comunes:**

- Bloquear puertos (22-SSH, 80-HTTP, 21-FTP)
- Permitir puertos específicos
- Reglas por origen/destino
- NAT y port forwarding

#### **UFW (Uncomplicated Firewall):**

- Frontend simple para iptables
- Sintaxis amigable
- Estándar en Ubuntu/Debian

#### **firewall-cmd:**

- Frontend para firewalld
- Zonas de confianza
- Estándar en RHEL/CentOS

## 6.4 IDS/IPS

### Sistema de Detección/Prevención de Intrusos:

#### IDS (Intrusion Detection System):

- Monitorea tráfico de red
- Detecta patrones maliciosos
- Alerta a administradores

#### IPS (Intrusion Prevention System):

- IDS con capacidad de bloqueo
- Acciones automáticas ante amenazas

#### Snort:

- IDS/IPS de código abierto
- Análisis de protocolo
- Detección basada en firmas
- Reglas personalizables
- Monitoreo de tráfico: ICMP, HTTP, SSH

#### Suricata:

- IDS/IPS moderno
- Multi-threading
- Mayor rendimiento que Snort

## 6.5 Distros Especializadas en soluciones de Firewall

#### pfSense:

- Firewall basado en FreeBSD
- Interfaz web completa
- Routing, VPN, QoS

#### OPNsense:

- Fork de pfSense
- Más moderno y actualizado
- Interfaz mejorada

**PacketFence NAC:**

- Network Access Control
- Control de acceso a red
- Gestión de dispositivos

**OpenWRT:**

- Firmware para routers
- Linux embebido
- Altamente personalizable

## **UNIDAD 7: Sistemas de archivos de red**

### **Network File System (NFS)**

**Introducción a NFS:**

NFS permite compartir directorios y archivos entre sistemas Unix/Linux a través de la red.

**¿Qué es NFS?**

- Protocolo de sistema de archivos distribuido
- Desarrollado por Sun Microsystems
- Transparente para usuarios y aplicaciones

**Cómo funciona:**

- Arquitectura cliente-servidor
- Servidor exporta directorios
- Clientes montan directorios remotos
- Acceso como si fueran locales

**Ventajas:**

- Compartición transparente
- Acceso simultáneo
- Administración centralizada
- Rendimiento aceptable en LAN

**Desventajas:**

- Seguridad limitada (versiones antiguas)
- Dependiente de red estable
- Performance en WAN limitado
- Sin cifrado nativo (v3 y anteriores)

#### **Configuración básica:**

- Archivo /etc/exports: define recursos compartidos
- Permisos y opciones de exportación
- Control de acceso por IP/red
- Montaje manual o automático (fstab)

## **Administración e implementación de SAMBA**

### **SAMBA:**

Suite de software que permite interoperabilidad entre Linux y Windows para compartir archivos e impresoras.

#### **Conceptos y arquitectura:**

- Implementa protocolo SMB/CIFS
- Compatible con redes Windows
- Servidor y cliente en Linux

#### **Componentes:**

- smbd: Servicio de archivos e impresoras
- nmbd: Resolución de nombres NetBIOS
- winbindd: Integración con dominios Windows

#### **Servicios de Compartir Archivos:**

- Linux como servidor de archivos para Windows
- Windows como servidor de archivos para Linux
- Permisos híbridos Unix/Windows
- Configuración en /etc/samba/smb.conf

#### **Samba4 como Controlador de Dominio:**

- Compatible con Active Directory
- Gestión centralizada de usuarios
- Políticas de grupo (GPO)

- Autenticación Kerberos
- Unir clientes Windows al dominio

## **UNIDAD 8: Otras Soluciones de Gestión de Usuarios**

### **Zentyal:**

- Servidor Linux para PyMEs
- Interfaz web unificada
- Controlador de dominio
- Gateway, firewall, servicios de red

### **OpenLDAP:**

- Directorio ligero de acceso abierto
- Base de datos centralizada de usuarios
- Autenticación unificada
- Integración con múltiples servicios

### **FreeIPA:**

- Identity, Policy, Audit
- Suite completa de gestión de identidades
- Kerberos, LDAP, DNS integrados
- Basado en proyectos open source

### **Red Hat Directory Server:**

- Solución empresarial basada en OpenLDAP
- Soporte comercial
- Alta disponibilidad

## **UNIDAD 9: Cloud y Virtualización**

### **Cloud Computing**

#### **Modelos de servicios:**

##### **On-Premise:**

- Infraestructura propia en local
- Control total

- Mayor inversión inicial
- Responsabilidad completa de mantenimiento

#### **Servicios de Colocación:**

- Hardware propio en datacenter externo
- Conectividad y energía gestionada
- Control de hardware, no de infraestructura física

#### **PaaS (Platform as a Service):**

- Plataforma de desarrollo en la nube
- Sin gestión de infraestructura
- Ejemplos: Heroku, Google App Engine

#### **SaaS (Software as a Service):**

- Software listo para usar
- Acceso vía web
- Ejemplos: Gmail, Office 365, Salesforce

#### **IaaS (Infrastructure as a Service):**

- Recursos de cómputo virtualizados
- VMs, almacenamiento, redes
- Ejemplos: AWS EC2, Azure VMs, Google Compute

#### **Servicios de nube principales:**

##### **AWS (Amazon Web Services):**

- Líder del mercado
- Mayor catálogo de servicios
- EC2, S3, RDS, Lambda

##### **Azure (Microsoft):**

- Integración con ecosistema Microsoft
- Fuerte en empresas
- VMs, Storage, SQL Database

##### **GCP (Google Cloud Platform):**

- Fortaleza en big data y ML



- Compute Engine, Cloud Storage, BigQuery

### **Digital Ocean:**

- Simplicidad y precios competitivos
- Droplets (VMs simples)
- Ideal para desarrolladores y proyectos pequeños

## **Introducción a Contenedores**

### **Contenedores vs Máquinas Virtuales:**

#### **Máquinas Virtuales:**

- Sistema operativo completo
- Hypervisor como intermediario
- Mayor consumo de recursos
- Aislamiento fuerte
- Inicio lento (minutos)

#### **Contenedores:**

- Comparten kernel del host
- Motor de contenedores (Docker)
- Ligeros y eficientes
- Aislamiento por namespaces
- Inicio rápido (segundos)

### **Docker:**

#### **Docker Registry:**

- Repositorio de imágenes
- Docker Hub: registro público principal
- Registros privados

#### **Objetos de Docker:**

#### **Imágenes:**

- Plantillas read-only
- Capas inmutables
- Base para contenedores

### **Contenedores:**

- Instancia ejecutable de imagen
- Aislado del host y otros contenedores
- Efímero por naturaleza

### **Volúmenes:**

- Persistencia de datos
- Independientes del ciclo de vida del contenedor
- Compartibles entre contenedores

### **Redes:**

- Comunicación entre contenedores
- Bridge, host, overlay, macvlan
- Aislamiento de red

### **Comandos básicos de Docker:**

- docker pull: Descargar imagen
- docker run: Ejecutar contenedor
- docker stop: Detener contenedor
- docker rm: Eliminar contenedor
- docker ps: Listar contenedores activos
- Mapeo de puertos: -p host:contenedor
- Mapeo de volúmenes: -v host:contenedor

### **Portainer.io:**

- Interfaz gráfica web para Docker
- Gestión visual de contenedores
- Despliegue simplificado
- Templates predefinidos

## **Despliegue Automático de Contenedores**

### **Docker Compose:**

- Herramienta de orquestación multi-contenedor
- Definición en archivo YAML (docker-compose.yml)
- Levanta aplicaciones completas con un comando
- Gestión de dependencias entre servicios

- Redes y volúmenes automáticos

**Casos de uso:**

- Stack LAMP/LEMP
- WordPress con MySQL
- Aplicaciones multi-tier

## UNIDAD 10: Herramientas de Administración

### Webmin

**¿Qué es Webmin?**

Interfaz web para administración de sistemas Linux, permite gestionar el servidor desde navegador.

**Capacidades:****Administración de usuarios:**

- Crear cuentas de usuario
- Eliminar usuarios
- Modificar grupos y permisos
- Gestión visual de home directories

**Gestión de servicios:**

- Iniciar servicios
- Detener servicios
- Reiniciar servicios
- Configurar inicio automático
- Ejemplos: SSH, Apache, MySQL

**Configuración de red:**

- Configurar IP estática
- Gestionar interfaces de red
- DNS y routing
- Firewall

**Administración de software:**

- Actualizar sistema
- Instalar paquetes
- Eliminar software
- Gestión de repositorios

#### **Gestión de archivos:**

- Explorador de archivos web
- Editar archivos de configuración
- Eliminar y mover archivos
- Permisos

#### **Monitorización del sistema:**

- Uso de CPU
- Consumo de memoria
- Almacenamiento disponible
- Procesos activos
- Gráficos en tiempo real

## **UNIDAD 11: Infraestructura como Código**

### **Terraform**

#### **¿Qué es Terraform?**

Herramienta de Infrastructure as Code (IaC) que permite definir y provisionar infraestructura mediante código declarativo.

#### **Uso y utilidad:**

- Automatización de despliegue
- Infraestructura versionable
- Reproducibilidad
- Multi-cloud

#### **Arquitectura de Terraform:**

#### **Componentes principales:**

- Providers: Plugins para interactuar con APIs (AWS, Azure, GCP, Digital Ocean)
- Resources: Componentes de infraestructura (VMs, redes, discos)

- State: Estado actual de la infraestructura
- Modules: Bloques reutilizables de configuración

### **Comandos principales:**

- terraform init: Inicializa proyecto
- terraform plan: Preview de cambios
- terraform apply: Aplica cambios
- terraform destroy: Elimina infraestructura

### **Cloud Provisioning:**

- Compatible con AWS, Azure, GCP, Digital Ocean
- Despliegue multi-cloud
- Gestión unificada

### **Terraform Registry:**

- Repositorio de módulos públicos
- Providers oficiales y community
- Reglas de seguridad
- Best practices

## **Introducción a Ansible**

### **¿Qué es Ansible?**

Herramienta de automatización de configuración y gestión de sistemas mediante SSH.

### **Características:**

- Agentless (sin agentes en nodos)
- Basado en Python
- Sintaxis YAML (legible)
- Idempotente (ejecutar múltiples veces sin cambiar resultado)

### **Componentes:**

#### **Archivo de Inventario:**

- Lista de hosts gestionados
- Grupos de servidores
- Variables por host/grupo

- Formato INI o YAML

### **Comandos Ad-Hoc:**

- Tareas puntuales sin playbooks
- Sintaxis: `ansible [grupo] -m [módulo] -a [argumentos]`
- Casos: reiniciar servicios, actualizar paquetes, copiar archivos

### **Playbooks:**

- Archivos YAML con tareas automatizadas
- Estructura declarativa
- Tasks, handlers, variables
- Reutilizables y versionables

### **Variables:**

- Valores dinámicos en playbooks
- Definidas en inventario, playbook o archivos separados
- Facts: variables automáticas del sistema

### **Templates:**

- Archivos Jinja2 con variables
- Generación dinámica de configuración
- Ejemplos: archivos de configuración personalizados

### **Módulos:**

- Unidades de trabajo de Ansible
- Categorías:
  - Gestión de sistema (service, user, package)
  - Windows (win\_service, win\_feature)
  - Instalación de apps (apt, yum, pip)
  - Cloud (ec2, azure\_rm, gcp)
  - Redes (ios\_config, nxos\_config)

## **UNIDAD 12: Telefonía IP con Asterisk**

### **Fundamentos de Asterisk**

#### **¿Qué es Asterisk?**

PBX (Private Branch Exchange) de código abierto para implementar sistemas de telefonía IP.

### **Protocolos e interfaces:**

#### **SIP (Session Initiation Protocol):**

- Protocolo estándar para VoIP
- Señalización de llamadas
- Más usado actualmente

#### **IAX (Inter-Asterisk eXchange):**

- Protocolo propietario de Asterisk
- Trunk entre servidores Asterisk
- Menor overhead

#### **DAHDI (Digium/Asterisk Hardware Device Interface):**

- Interfaz para hardware telefónico
- Tarjetas PCI de telefonía
- Conexión a líneas analógicas/digitales

#### **ATA (Analog Telephone Adapter):**

- Adaptador para teléfonos analógicos
- Convierte análogo a VoIP

### **Codecs y formatos:**

#### **Codecs de audio:**

- **uLaw (G.711u):** Estándar norteamericano, 64 kbps
- **aLaw (G.711a):** Estándar europeo, 64 kbps
- **G.722:** HD voice, 64 kbps
- **G.729:** Compresión alta, 8 kbps (licencia propietaria)
- **Opus:** Moderno, adaptativo

#### **Distros especializadas en VoIP:**

- **FreePBX:** Interfaz web para Asterisk
- **Elastix:** Suite completa (descontinuado)
- **Issabel:** Fork de Elastix
- **3CX:** Solución comercial

- **VitalPBX:** Alternativa moderna

## Instalación de Asterisk

### Instalación On-Premise:

- Servidor físico o VM local
- Control total del sistema
- Compilación desde fuente o paquetes

### Instalación Cloud:

- VM en proveedor cloud (Digital Ocean, AWS, Azure)
- Accesible desde Internet
- Escalable
- Issabel PBX sobre Rocky Linux 8

## Módulos de Asterisk

### Extensiones:

- Números internos para usuarios
- Configuración de softphones
- Apps: Zoiper, Linphone, MicroSIP
- Registro de dispositivos

### IVR (Interactive Voice Response):

- Menú telefónico automatizado
- Enrutamiento de llamadas por opciones
- Mensajes grabados
- Árboles de decisión

### Ring Group (Grupo de Llamada):

- Múltiples extensiones suenan simultáneamente
- Estrategias: ringall, hunt, random
- Timeout configurable

### Music On Hold:

- Música mientras espera
- Archivos de audio personalizados



- Diferentes categorías

Esta guía proporciona los conceptos y significados fundamentales para cada tema del programa de administración de sistemas Linux. Para dominar cada sección, se requiere práctica hands-on siguiendo los ejemplos prácticos específicos mencionados en los temas.