

GESTIÓN DE RECURSOS EN UNA RED EN SISTEMAS OPERATIVOS LIBRES

CFGs: Desarrollo de
Aplicaciones Multiplataforma
Profesor: Javier Palacios
Curso: 2020-21

CONFIGURACIÓN DE SISTEMAS OPERATIVOS LIBRES

1. TÉCNICAS DE CONEXIÓN REMOTA
2. HERRAMIENTAS DE ANÁLISIS Y ADMINISTRACIÓN
3. SERVICIOS EN RED
 - SERVIDORES DE FICHEROS
 - SERVIDORES DE IMPRESIÓN
 - SERVIDORES WEB
 - SERVIDORES DE CORREO ELECTRÓNICO
 - SERVIDORES DE APLICACIONES

1. TÉCNICAS DE CONEXIÓN REMOTA

OpenSSH es un conjunto de aplicaciones implementación libre basadas en el protocolo SSH y orientado a acceso remoto seguro. Proporciona funcionalidad de shell, pero también encapsula otros protocolos no seguros proporcionando prestaciones avanzadas de seguridad, como el cifrado extremo a extremo.

El cliente `ssh` en Linux suele venir preinstalado, pero para actuar como servidor suele ser necesario instalar el paquete `openssh-server`.

La configuración del servicio se realiza mediante el fichero `/etc/ssh/sshd_config`. El servicio se puede gestionar mediante `service` o `systemctl` en función del sistema.

1. TÉCNICAS DE CONEXIÓN REMOTA

Comandos cliente:

- **ssh** – Es un cliente de shell remoto seguro mediante ssh.
 - Sintaxis:
 - `ssh [opciones] [[usuario@]máquina]]`
 - Algunas opciones:
 - `usuario` Usuario de la máquina a la que se quiere conectar. Si no se especifica se solicitará en modo interactivo.
 - `máquina` Puede ser un hostname, una IP, o una IP y un puerto.
 - `-p <puerto>` Permite indicar un número de puerto diferente del puerto por defecto (22).
- **sftp** – Es un cliente para acceder a ficheros de forma remota mediante el protocolo SFTP (SSH File Transfer Protocol).
 - Sintaxis:
 - `sftp [opciones] [[usuario@]máquina]]`
 - Algunas opciones:
 - `-help` Muestra una lista de comandos para interactuar con el servidor.
 - `-P <puerto>` Permite indicar un número de puerto diferente del puerto por defecto (22).

1. TÉCNICAS DE CONEXIÓN REMOTA

Otra alternativa es utilizar aplicaciones para conectarse a **escritorio remoto** (solo para sistemas con GUI):



- **Remmina:** Es un cliente de escritorio compatible con múltiples protocolos (RDP, VNC, NX, XDMCP, SPICE y SSH).

- Basados en VNC:



- **TigerVNC** es un cliente y servidor VNC Libre orientado al control remoto de equipos.



- **RealVNC** es un software de acceso remoto, proporciona servidor y cliente sobre VNC. Dispone de licencia personal libre y licencia comercial para uso empresarial.

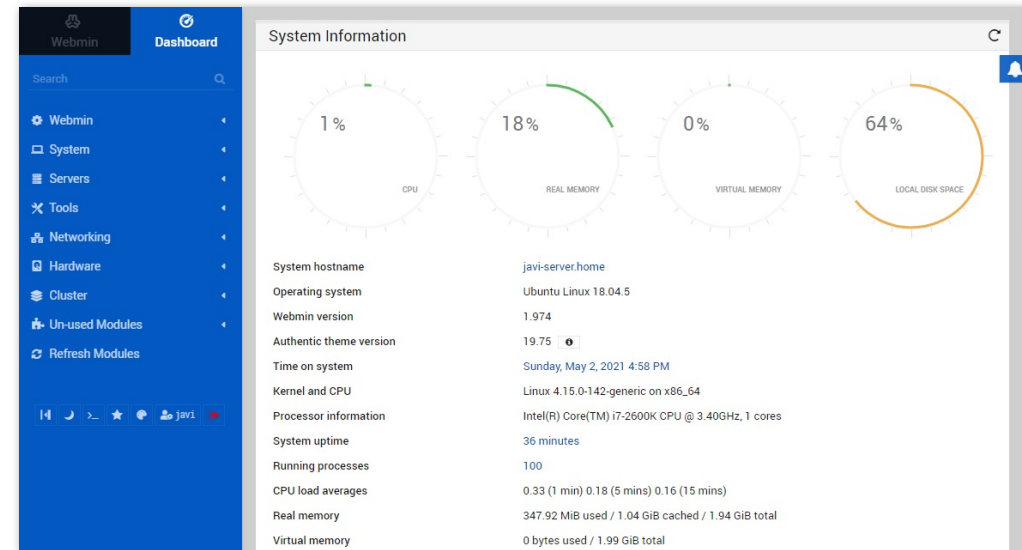


- **TeamViewer** es un software informático multiplataforma para acceso remoto. Ofrece opciones gratuitas (Freeware) limitadas y versión profesional de pago. Proporciona entre otras funciones capacidad de compartir y controlar escritorios, reuniones en línea, videoconferencia y transferencia de archivos.

2. HERRAMIENTAS DE ANÁLISIS Y ADMINISTRACIÓN

A parte de las herramientas integradas en el propio sistema operativo, existen herramientas de terceros que facilitan la administración de los servidores:

- **Webmin:** es una herramienta de software libre (licencia BSD) de configuración de sistemas Unix y GNU/Linux mediante acceso web (puerto 10000). Permite configurar aspectos propios del sistema operativo, así como múltiples servicios, algunos integrados, y otros instalables en forma de módulos, incluyendo servidores web Apache, PHP, MariaDB, DNS, LDAP, DHCP o Samba.



3.1. SERVIDORES DE FICHEROS

Existen diversos métodos de compartición de archivos en red en Linux. La elección del sistema suele depender del uso que se vaya a dar:

- **FPT:** es un protocolo de red para la transferencia de archivos, funciona sobre TCP (puerto 20 y 21). Un servidor FTP proporciona acceso a archivos y carpetas. Existen clientes específicos como Filezilla, pero el propio explorador de archivos de Windows permite acceder a carpetas publicadas en servidores FTP. Un servidor muy utilizado en Linux es **vsftpd**. Se considera un protocolo muy inseguro y desaconsejado (todo el tráfico, incluidas contraseñas, se transmiten sin cifrar), **FTPS** o **SFTP** (FTP sobre SSH) son alternativas más seguras (cifrado extremo a extremo).
- **NFS:** es un protocolo de nivel de aplicación utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Incluido por defecto en mayoría de sistemas operativos tipo UNIX, en Linux se utiliza **nfs-kernel-server** como servidor y **nfs-common** como cliente.
- **SMB/CIFS:** es un protocolo de red que permite compartir recursos (archivos, impresoras...) en un entorno de red de computadoras que usen el sistema operativo Windows.
- **SAMBA:** es una implementación libre del protocolo de archivos compartidos SMB, (renombrado como CIFS) para que sistemas de tipo UNIX puedan actuar como servidores o clientes en redes Windows. Samba también permite a sistemas tipo linux actuar como clientes o servidores del servicio Active Directory de Microsoft.
- Otros servicios orientados a la compartición de archivos en entornos en red son: **UPnP**, **DLNA**.

3.2. SERVIDORES DE IMPRESIÓN

Un sistema Linux puede establecerse como servidor de impresión. Una de las opciones más populares es CUPS:

- **CUPS:** es un sistema de impresión de código abierto basado en estándares desarrollado por Apple Inc. para macOS y otros sistemas tipo UNIX. CUPS utiliza el Protocolo de impresión de Internet (IPP) para admitir la impresión en impresoras locales y de red.
- Proporciona una interfaz web para gestionar el servicio en el puerto 361.



3.3. SERVIDORES WEB

Algunos servidores web habitualmente utilizados en Linux:

- **Apache** (paquete apache2): es un servidor web HTTP de código abierto multiplataforma, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616. Es modular y altamente configurable, suele utilizarse con la pila de software LAMP.
- **Nginx**: es un servidor web/proxy multiplataforma, de alto rendimiento libre y de código abierto. Como proxy puede actuar como proxy inverso ligero y un proxy de correo electrónico (IMAP/POP3).



3.1. SERVIDORES DE CORREO ELECTRÓNICO

Servidores de correo (MTA: mail/message transfer agent) electrónico habitualmente utilizados en Linux:

- **Postfix:** es un servidor de correo de código abierto y libre para el enrutamiento y envío de correo electrónico. Funciona sobre sistemas tipo Unix. Soporta múltiples protocolos de autenticación, Greylisting y antispam con add-on.
- **Sendmail:** es otro MTA (agente de transporte de correo), tiene versión libre y de código abierto y propietaria. Utiliza Simple Mail Transfer Protocol (SMTP), aunque soporta otros protocolos. Soporta múltiples protocolos de autenticación, Greylisting y SpamAssassin.
- **Exim:** otro MTA (agente de transporte de correo) para sistemas tipo Unix desarrollado por la universidad de Cambridge. Soporta múltiples protocolos de autenticación, Greylisting y SpamAssassin.



Servidores de correo con cliente webmail integrado:

- **Mail-in-a-box:** es un es un servidor de correo hosteado (web) de código abierto y libre para sistemas operativos tipo UNIX, soporta IMAP/SMTP, compatible con ActiveSync, multiusuario y multidominio.
- **iRedMail:** Otro servidor de webmail POP3/IMAP/SMTP sobre TLS (HTTPS), libre y de código abierto. Compatible con OpenLDAP, MySQL, MariaDB, PostgreSQL.

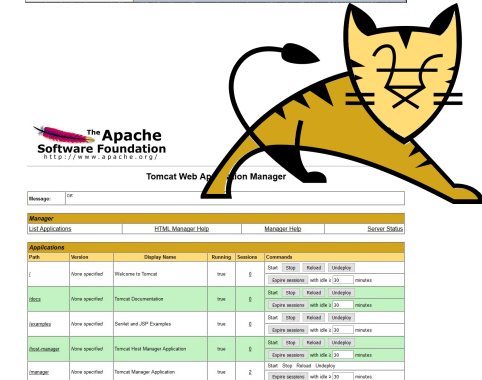
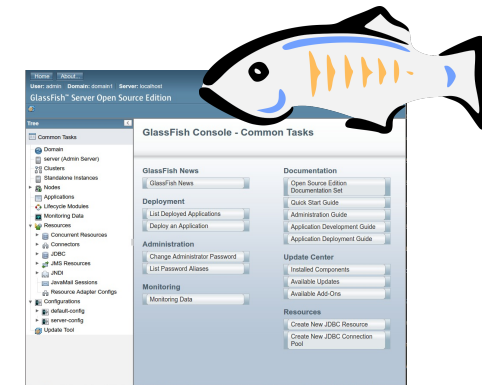


3.1. SERVIDORES DE APLICACIONES

Algunos servidores de aplicaciones habitualmente utilizados en Linux:

- **Servidor de aplicaciones Java:**

- **Glassfish:** es un servidor de aplicaciones propiedad de Oracle Corporation. Permite ejecutar aplicaciones que implementan Java EE. Tiene una versión libre y de código abierto, y otra comercial orientada a empresas.
- **Tomcat:** no es propiamente un servidor de aplicaciones, sino un contenedor de servlets y JSP. Se desarrollo para el proyecto Jakarta EE (una implementación OpenSource de Java EE) de la Apache Software Foundation.

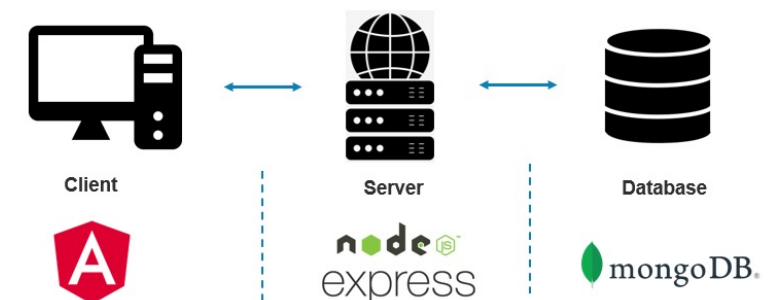
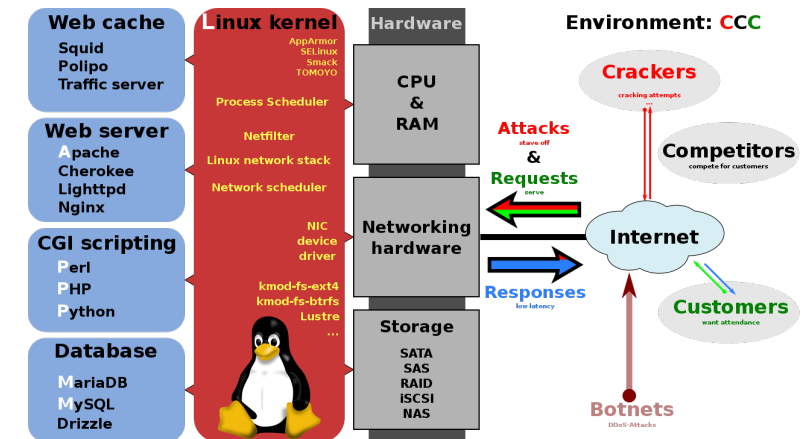


3. SERVICIOS EN RED

3.1. SERVIDORES DE APLICACIONES

Algunos servidores de aplicaciones habitualmente utilizados en Linux:

- Servidores de aplicaciones web:
 - **LAMP**: es un acrónimo usado para describir un servidor de aplicaciones web basado en Linux que incluye al menos la siguientes pila de software:
 - **Apache**, como servidor web, aunque puede ser sustituido por nginx (**LEMP**).
 - **MySQL/MariaDB**, como gestor de bases de datos. También son aceptados PostgreSQL o sistemas no SQL como MongoDB.
 - **PHP**, el lenguaje de programación en entorno servidor. Otras alternativas son Pearl y Python.
 - **MEAN**: es el acrónimo utilizado para describir una pila de software para aplicaciones web basado en javascript (Fullstack):
 - **MongoDB**, como gestor de bases de datos (no SQL).
 - **Express**, como Framework para aplicaciones web de NodeJS.
 - **Angular**, como Framework para front-end. Puede ser sustituido por React (**MERN**)
 - **Node.js**, como interprete Javascript en el lado servidor.



3.1. OTROS SERVICIOS

Existen diversos métodos de compartición de archivos en red en Linux. La elección del sistema suele depender del uso que se vaya a dar:

- **DNS:**

- **BIND** (Berkeley Internet Name Domain) es un servidor de DNS patrocinado por la Internet Systems Consortium. Es uno de los servidores más utilizado en sistemas UNIX y GNU/Linux.

- **DHCP:**

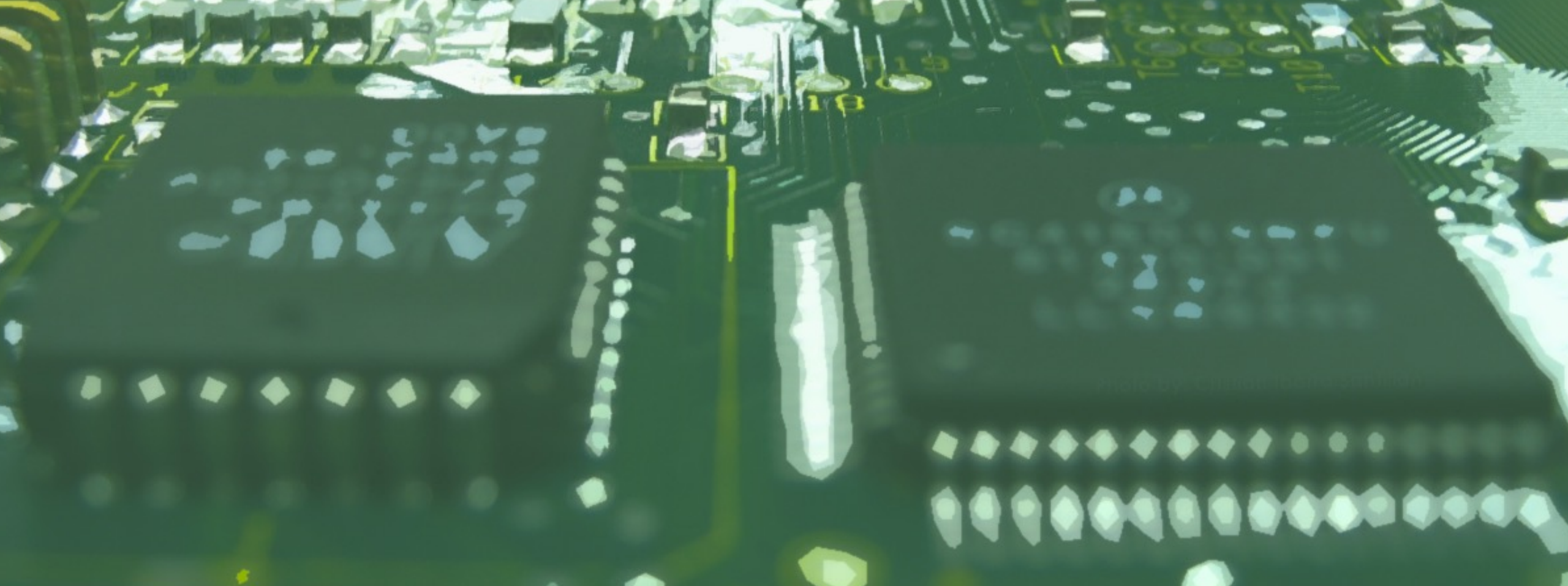
- **isc-dhcp-server:** es una solución libre y de código abierto para implementar servidores DHCP, agentes de retransmisión y clientes. Admite tanto IPv4 como IPv6.

- **Web cache Proxy:**

- **Squid:** es un servidor proxy para web con caché de código abierto y libre. Permite realizar filtrado de tráfico, así como acelerar el acceso a servidores web almacenando en caché peticiones web y DNS recurrentes.

- **Directorio:**

- **OpenLDAP:** es una implementación libre y de código abierto del protocolo Lightweight Directory Access Protocol (LDAP), también proporciona un servidor y clientes para los principales servicios.
- **SAMBA:** es una implementación libre del protocolo de compartición de recursos CIFS (anteriormente conocido como SMB) para sistemas de tipo UNIX. Permite a sistemas UNIX o GNU/Linux actuar como servidores o clientes en redes de Windows. Samba también permite proporcionar o consumir servicios de Active Directory, impresión, directorios compartidos o autenticación.



GRACIAS

CFGS: Desarrollo de
Aplicaciones Multiplataforma
Profesor: Javier Palacios
Curso: 2020-21