

DevOps

roadmap.sh

Aprende un lenguaje de programación

No importa qué lenguaje de programación elijas, pero es importante aprender al menos uno. Podrás utilizar ese lenguaje para escribir scripts de automatización.

- [Guía para elegir un lenguaje para DevOps](#)

Python

Python es un lenguaje multiparadigma. Al ser un lenguaje interpretado, el código se ejecuta en cuanto se escribe y la sintaxis de Python permite escribir código de forma programática funcional, procedimental u orientada a objetos. Python se recomienda a menudo como el primer lenguaje que deben aprender los nuevos programadores, debido a su enfoque en la legibilidad, la coherencia y la facilidad de uso. Esto tiene sus inconvenientes, ya que el lenguaje no es especialmente eficaz en la mayoría de las tareas de producción.

Visite los siguientes recursos para obtener más información:

- [Visite la hoja de ruta dedicada a Python](#)
- [Sitio web de Python](#)
- [Introducción a Python](#)
- [Automatizar las cosas aburridas](#)
- [W3Schools - Tutorial de Python](#)
- [Curso intensivo de Python](#)
- [Codecademy - Aprende Python 3](#)

Go

Go es un lenguaje de programación de código abierto apoyado por Google. Go se puede utilizar para escribir servicios en la nube, herramientas CLI, se utiliza para el desarrollo de API, y mucho más.

Visite los siguientes recursos para obtener más información:

- [Visita Dedicated Go Roadmap](#)
- [Un recorrido por Go - Conceptos básicos de Go](#)
- [Documentación de referencia de Go](#)
- [Aprende Go | Boot.dev](#)
- [Go by Example - programas de ejemplo comentados](#)
- [Aprende Go | Codecademy](#)
- [Tutorial de Go de W3Schools](#)
- [Creación de una API RESTful JSON en Go](#)

Ruby

Ruby es un lenguaje de programación interpretado de alto nivel que mezcla Perl, Smalltalk, Eiffel, Ada y Lisp. Ruby se centra en la simplicidad y la productividad, junto con una sintaxis que se lee y escribe de forma natural. Ruby admite programación procedimental, funcional y orientada a objetos, y está tipado dinámicamente.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Ruby](#)
- [Aprende Ruby en 20 minutos](#)
- [Aprende Ruby | Codecademy](#)

Rust

Rust es un moderno lenguaje de programación de sistemas centrado en la seguridad, la velocidad y la concurrencia. Consigue estos objetivos siendo seguro en memoria sin utilizar la recolección de basura.

Visite los siguientes recursos para obtener más información:

- [The Rust Programming Language - libro en línea](#)
- [Rust by Example - colección de ejemplos ejecutables](#)
- [The Rust Programming Book - Versión en vídeo](#)
- [Comprehensive Rust by Google - Aprende Rust en 4 días](#)
- [Curso Microsoft Learn](#)
- [Guía rápida de Rust](#)
- [Rust Katas - Pequeños ejercicios interactivos de Rust](#)

JavaScript

JavaScript le permite añadir interactividad a sus páginas. Ejemplos comunes que puede haber visto en los sitios web son deslizadores, interacciones de clic, ventanas emergentes y así sucesivamente. Además de utilizarse en el front-end de los navegadores, existe Node.js, que es un entorno de ejecución de JavaScript de código abierto, multiplataforma y back-end que se ejecuta en el motor V8 y ejecuta código JavaScript fuera de un navegador web.

Visite los siguientes recursos para obtener más información:

- [Visita Dedicated JavaScript Roadmap](#)
- [W3Schools - Tutorial de JavaScript](#)
- [Tutorial moderno de JavaScript](#)
- [Codecademy - Aprenda JavaScript](#)
- [Curso intensivo de JavaScript para principiantes](#)
- [Curso intensivo de Node.js](#)
- [Tutorial de Node.js para principiantes](#)
- [W3Schools - Tutorial de Node.js](#)
- [¿Qué es NPM?](#)
- [Documentación oficial](#)

Sistema operativo

Un sistema operativo sirve de puente entre el usuario de un ordenador y su hardware. La función de un sistema operativo es ofrecer un entorno en el que el usuario pueda ejecutar programas de forma cómoda y eficaz. En términos sencillos, podemos decir que un sistema operativo (SO) es una interfaz entre el usuario y el hardware del ordenador. Un SO permite que los programas de software se comuniquen con el hardware de un ordenador, El **kernel** es el nombre de la pieza de software que alberga los elementos fundamentales del **Sistema Operativo**.

Visita los siguientes recursos para saber más:

- [Todo lo que necesitas saber sobre los SO.](#)
- [Aprender Sistemas Operativos](#)
- [Tutoriales sobre Sistemas Operativos.](#)
- [¿Qué son los Sistemas Operativos?](#)
- [¡Sistemas Operativos!](#)

Windows

Windows es un sistema operativo basado en una interfaz gráfica de usuario (GUI) desarrollado por Microsoft. Es un sistema operativo propietario híbrido basado en el núcleo. Según una encuesta, hasta abril de 2022, Windows es el sistema operativo más popular del mundo, con una cuota de mercado del 75%.

Visite los siguientes recursos para obtener más información:

- [Sitio oficial de Windows](#)
- [Documentación de Windows para desarrolladores](#)

GNU/Linux

Ubuntu / Debian

Debian es una distribución de Linux libre y de código abierto desarrollada por el Proyecto Debian, una organización comunitaria de software totalmente voluntaria. Debian es la distribución de Ubuntu.

Ubuntu es una distribución de Linux libre y de código abierto basada en Debian. Ubuntu está disponible en tres versiones: Desktop, Server y Core.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Debian](#)
- [Sitio web de Ubuntu](#)
- [Aprenda gratis a usar Linux](#)
- [Sistema operativo Linux - Curso acelerado para principiantes](#)
- [La línea de comandos de Linux por William Shotts](#)
- [r/linuxupskillchallenge](#)
- [Introducción a Linux - Curso completo para principiantes](#)
- [Fundamentos de Linux](#)

openSUSE Linux

openSUSE es una distribución de Linux gratuita cuyo objetivo es promover el uso de Linux en todo el mundo. openSUSE está disponible en dos versiones: Leap y Tumbleweed.

Visite los siguientes recursos para obtener más información:

- [Sitio web de openSUSE Linux](#)
- [Documentación de openSUSE](#)
- [Guía de inicio no oficial de openSUSE](#)

RHEL / Derivados

Red Hat Enterprise Linux (RHEL) es una popular distribución del sistema operativo Linux diseñada para uso empresarial. Está desarrollada y mantenida por Red Hat, Inc. y está disponible bajo un modelo basado en suscripción.

Existen varias distribuciones de Linux que están basadas en RHEL, o que han sido derivadas de RHEL de alguna manera. Estas distribuciones se conocen como derivados de RHEL. Algunos ejemplos de derivados de RHEL son: AlmaLinux, CentOS, CloudLinux, Oracle Linux y Scientific Linux.

Los derivados de RHEL se utilizan a menudo en entornos empresariales porque ofrecen la estabilidad y fiabilidad de RHEL, pero con la ventaja añadida de ser alternativas gratuitas o de menor coste.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Red Hat Enterprise Linux](#)
- [Documentación de RHEL](#)
- [Guías de introducción a RHEL](#)
- [Qué es Red Hat Enterprise Linux \(RHEL\) - Red Hat](#)
- [Aprenda Linux 101](#)

Unix

FreeBSD

FreeBSD es un sistema operativo libre y de código abierto similar a Unix que incluye muchas características como multitarea preventiva, protección de memoria, memoria virtual y facilidades multiusuario.

Visite los siguientes recursos para obtener más información:

- [Sitio web de FreeBSD](#)
- [Documentación oficial de FreeBSD](#)
- [Manual de FreeBSD](#)
- [Recursos de FreeBSD para novatos](#)

OpenBSD

OpenBSD es un sistema operativo libre y de código abierto similar a Unix, centrado en la portabilidad, la estandarización, la corrección, la seguridad proactiva y la criptografía integrada. El popular software **OpenSSH** está desarrollado a partir de OpenBSD.

Visite los siguientes recursos para obtener más información:

- [Sitio web de OpenBSD](#)
- [Documentación oficial de OpenBSD](#)
- [Manual de OpenBSD](#)
- [Guía de instalación de OpenBSD](#)

NetBSD

NetBSD es un sistema operativo de código abierto de tipo Unix gratuito, rápido, seguro y altamente portable. Está disponible para una amplia gama de plataformas, desde servidores a gran escala y potentes sistemas de escritorio hasta dispositivos portátiles e integrados.

Visite los siguientes recursos para obtener más información:

- [Sitio web de NetBSD](#)
- [Documentación oficial de NetBSD](#)
- [Guía de NetBSD](#)

Vivir en terminal

Un terminal es simplemente una interfaz basada en texto para el ordenador, se utiliza para interactuar con su sistema informático a través de CLI (interfaz de línea de comandos)

Visita los siguientes recursos para saber más:

- [¿Qué es CLI?](#)

Supervisión de procesos

Un proceso es una instancia de un programa informático que se está ejecutando. Cada proceso se identifica por un número único llamado ID de proceso (PID). Un proceso es un programa en ejecución. El sistema operativo rastrea los procesos mediante el uso de identificadores de proceso. Un identificador de proceso (PID) es un número único que identifica a un proceso específico. Un PID se asigna automáticamente a cada proceso cuando se crea en el sistema.

Hay varios comandos de linux que se pueden utilizar para monitorizar procesos. Los más comunes son:

- `ps` - muestra una instantánea de los procesos actuales.
- `top` - muestra los procesos de Linux.
- `htop` - visor interactivo de procesos.
- `atop` - monitor interactivo avanzado para ver la carga en un sistema Linux.
- `lsof` - lista los archivos abiertos.

La utilidad `ps` muestra una línea de encabezado, seguida por líneas que contienen información sobre todos sus procesos que tienen terminales de control.

- [Documentación de ps](#)
- [Cheatsheet ps](#)
- [Curso acelerado de Linux - El comando ps](#)

El programa `top` muestra periódicamente una lista ordenada de los procesos del sistema. La clave de ordenación por defecto es pid, pero pueden usarse otras claves en su lugar. Hay disponibles varias opciones de salida.

- [Documentación de top](#)
- [Cheatsheet top](#)

`htop` es un proceso multiplataforma basado en ncurses. Es similar a `top`, pero le permite desplazarse vertical y horizontalmente, e interactuar utilizando un dispositivo señalador (ratón). Puede observar todos los procesos que se ejecutan en el sistema, junto con sus argumentos de línea de comandos, así como verlos en formato de árbol, seleccionar varios procesos y actuar sobre todos ellos a la vez.

- [Documentación de htop](#)
- [Cheatsheet htop](#)

`lsof` lista en su salida estándar información sobre los ficheros abiertos por los procesos.

- [Cheatsheet lsof](#)
- [Documentación de lsof](#)
- [Curso acelerado de Linux - El comando lsof](#)

Supervisión del rendimiento

Hay muchas herramientas disponibles para monitorizar el rendimiento de tu aplicación. Algunas de las más populares son

- **nmon** - Una herramienta de monitorización del sistema para sistemas Linux y AIX.
- **iostat** - Una herramienta que reporta estadísticas de CPU y estadísticas de entrada/salida para dispositivos, particiones y sistemas de archivos de red.
- **sar** - Un comando de monitorización del sistema utilizado para informar sobre varias cargas del sistema, incluyendo actividad de CPU, memoria/paging, carga de dispositivos, red.
- **vmstat** - Herramienta que informa de las estadísticas de la memoria virtual.

nmon es una herramienta de línea de comandos de monitorización del rendimiento totalmente interactiva para Linux. Es una herramienta de evaluación comparativa que muestra el rendimiento de la cpu, memoria, red, discos, sistema de archivos, nfs, procesos principales, recursos y micropartición de energía.

- [Documentación de nmon](#)
- [Guía del comando nmon](#)

El comando **iostat** en Linux se utiliza para monitorizar las estadísticas de entrada/salida del sistema para dispositivos y particiones. Monitoriza la entrada/salida del sistema observando el tiempo que los dispositivos están activos en relación con sus tasas medias de transferencia. Los informes producidos por iostat se pueden utilizar para cambiar la configuración del sistema para equilibrar la entrada/salida entre los discos físicos.

- [Documentación de iostat](#)
- [Guía de comandos iostat](#)

Abreviatura de **S**ystem **A**ctivity **R**eport (Informe de Actividad del Sistema), **sar** es una herramienta de línea de comandos para Unix y sistemas operativos tipo Unix que muestra un informe de diferente información sobre el uso y la actividad de los recursos en el sistema operativo.

- [Comando SAR en Linux para monitorizar el rendimiento del sistema](#)
- [SAR Página Principal](#)
- [SAR Hombre Página 2](#)
- [Sar tutorial para principiantes](#)

Abreviatura de **V**irtual **m**emory **s**tatistic reporter (Estadística de memoria virtual), **vmstat** es una herramienta de línea de comandos para sistemas operativos Unix y similares a Unix que reporta diversa información sobre el sistema operativo, como el uso de memoria, paginación, procesos, E/S, CPU y disco.

- [Comando vmstat en Linux con Ejemplos](#)
- [Comandos Linux: explorando la memoria virtual con vmstat](#)
- [Página de manual de VMstat](#)
- [Tutorial de vmstat](#)

Herramientas de red

Las herramientas de red se utilizan para solucionar problemas de red. También se utilizan para supervisar el tráfico de red y probar la conectividad de la red. Algunas de las herramientas de red más comunes son

- `traceroute` - Traza la ruta tomada por los paquetes a través de una red IP.
- `ping` - Envía paquetes de solicitud de eco a un host para probar la conexión a Internet.
- `mtr` - Combina la funcionalidad de `traceroute` y `ping` en una única herramienta de diagnóstico.
- `nmap` - Escanea hosts en busca de puertos abiertos.
- `netstat` - Muestra conexiones de red, tablas de enrutamiento, estadísticas de interfaz, conexiones de enmascaramiento y membresías de multidifusión.
- `ufw` y `firewalld` - Herramientas de gestión de cortafuegos.
- `iptables` y `nftables` - Herramientas de gestión de cortafuegos.
- `tcpdump` - Vuelca el tráfico de una red.
- `dig` - Utilidad de búsqueda DNS.
- `scp` - Copia segura.

El comando `traceroute` es un comando de Linux que imprime la ruta que sigue un paquete de red desde su origen (por ejemplo, tu ordenador) hasta el host de destino (por ejemplo, `roadmap.sh`). Es bastante valioso para investigar conexiones de red lentas, ya que puede ayudarnos a detectar el tramo lento del viaje del paquete de red a través de Internet.

- [Cómo ejecutar Traceroute en Linux](#)
- [Comando Traceroute en Linux con ejemplos](#)

El comando `ping` (**P**acket **I**nternet **G**roper) se utiliza para comprobar la conectividad de red entre el host y el servidor/host. Este comando toma como entrada la dirección IP o la URL y envía un paquete de datos a la dirección especificada con el mensaje "PING" y obtiene una respuesta del servidor/host este tiempo se registra lo que se denomina latencia.

- [¿Qué es el comando ping?](#)
- [Comando ping con ejemplos](#)

`mtr` combina la funcionalidad de los programas `traceroute` y `ping` en una única herramienta de diagnóstico de redes.

- [Javatpoint: Comando Linux mtr](#)
- [Comando mtr en Linux](#)
- [Cómo hacer traceroute con el comando mtr en Linux](#)

NMAP significa Network Mapper y es una herramienta de código abierto utilizada para explorar y auditar la seguridad de la red, como comprobar cortafuegos y escanear puertos.

- [NMAP Libro Manual Oficial](#)
- [Qué es Nmap y Cómo Usarlo](#)

Netstat es una utilidad de línea de comandos para mostrar todas las conexiones de red de un sistema. Muestra todas las conexiones de sockets tcp, udp y unix. Aparte de los sockets conectados, también muestra los sockets a la escucha que están esperando conexiones entrantes.

- [Comando netstat en Linux con Ejemplos](#)
- [Tutorial de Netstat](#)
- [Comandos Netstat - Tutorial de Administración de Redes](#)
- [Tutorial de Línea de Comandos Linux para Principiantes - Comando netstat](#)

UFW, o *uncomplicated firewall*, es una utilidad basada en línea de comandos para gestionar reglas de cortafuegos en Arch Linux, Debian y Ubuntu. Su objetivo es hacer que la configuración del cortafuegos sea lo más sencilla posible. Es una interfaz para la herramienta de cortafuegos `iptables`.

- [Documentación de ufw](#)
- [Introducción básica a UFW](#)
- [Fundamentos de UFW](#)

Iptables es una utilidad cortafuegos de línea de comandos que utiliza cadenas de políticas para permitir o bloquear el tráfico que será aplicado por el marco netfilter del kernel linux. El mecanismo de filtrado de paquetes de Iptables está organizado en tres tipos diferentes de estructuras: tablas, cadenas y objetivos.

- [Tutorial de Iptables](#)
- [Guía de principiantes a avanzados de Iptables](#)

`tcpdump` es una herramienta de línea de comandos utilizada para analizar el tráfico de red que pasa por su sistema. Puede utilizarse para capturar y filtrar paquetes y mostrarlos en un formato legible por humanos. La información capturada también puede ser analizada posteriormente.

- [Documentación de tcpdump](#)
- [Introducción básica a Tcpdump](#)
- [50 formas de aislar tráfico con Tcpdump](#)
- [Interpretación de la salida y los datos de Tcpdump](#)

`dig` son las siglas de Domain Information Groper. Se utiliza para recuperar información sobre servidores de nombres DNS. Lo utilizan sobre todo los administradores de red para verificar y solucionar problemas de DNS y para realizar búsquedas de DNS. Sustituye a herramientas más antiguas como `nslookup` y `host`.

- [¿Qué es el comando dig?](#)
- [Más información sobre dig](#)
- [¿Qué es el comando host?](#)
- [¿Qué es el comando nslookup?](#)
- [¿Qué es DNS?](#)

SCP es el acrónimo de Secure Copy Protocol (protocolo de copia segura), una utilidad de línea de comandos que permite al usuario copiar de forma segura archivos y directorios entre dos ubicaciones, normalmente entre sistemas unix o linux. El protocolo garantiza que la transmisión de archivos se realiza de forma cifrada para evitar que cualquier persona con intenciones sospechosas obtenga información confidencial. SCP utiliza el cifrado a través de una conexión SSH (Secure Shell), lo que garantiza que los datos que se transfieren están protegidos de ataques sospechosos.

- [Comando SCP para Linux](#)
- [10 ejemplos de comandos SCP](#)
- [Explicación del comando SCP](#)

Manipulación de texto

Algunos de los comandos más comunes que utilizará en el terminal son los de manipulación de texto. Estos comandos le permiten buscar, reemplazar y manipular texto en archivos y secuencias. Estos son algunos de los comandos más comunes que utilizará:

- `awk` - Un lenguaje de programación diseñado para el procesamiento de texto y típicamente utilizado como una herramienta de extracción de datos e informes.
- `sed` - Un editor de flujos para filtrar y transformar texto.
- `grep` - Utilidad de línea de comandos para buscar en conjuntos de datos de texto plano líneas que coincidan con una expresión regular.
- `sort` - Utilidad de línea de comandos para ordenar líneas de archivos de texto.
- `cut` - Utilidad de línea de comandos para cortar secciones de cada línea de los archivos.
- `uniq` - Utilidad de línea de comandos para informar u omitir líneas repetidas.
- `cat` - Utilidad de línea de comandos para concatenar archivos e imprimirlos en la salida estándar.
- `echo` - Utilidad de línea de comandos para mostrar una línea de texto.
- `fmt` - Utilidad de línea de comandos para el formateo óptimo de texto.
- `tr` - Utilidad de línea de comandos para traducir o eliminar caracteres.
- `nl` - Utilidad de línea de comandos para numerar líneas de archivos.
- `wc` - Utilidad de línea de comandos para imprimir el recuento de líneas nuevas, palabras y bytes de los archivos.

`awk` es un lenguaje de scripting de propósito general utilizado para manipular datos o texto y generar informes en el mundo Linux. Se utiliza sobre todo para escanear y procesar patrones. Busca en uno o más archivos para ver si contienen líneas que coincidan con los patrones especificados y luego realiza las acciones asociadas.

- [¿Qué es AWK? ¿Cómo se utiliza?](#)
- [¿Cómo funciona AWK?](#)
- [Curso acelerado de Linux – awk](#)

El comando `sed` (**S**tream **E**ditor) en UNIX puede realizar muchas funciones en archivos como buscar, encontrar y reemplazar, insertar o borrar. Usando SED puedes editar archivos incluso sin abrirlos en editores como **VI Editor**.

- [¿Qué es SED? con ejemplos](#)
- [Manual detallado](#)
- [Curso acelerado de Linux - El comando sed](#)

El comando `grep` (**g**lobal search for **r**egular **e**xpression and **p**rint out) busca en archivo(s) un patrón particular de caracteres, y muestra todas las líneas que contienen ese patrón. Puede utilizarse con otros comandos como `ps`, lo que lo hace más útil.

- [¿Qué es Grep? con ejemplos](#)
- [Manual detallado](#)
- [Curso acelerado de Linux - El comando grep](#)

El comando `sort` se utiliza para ordenar el contenido de un fichero en un orden determinado. Por defecto, ordena un fichero asumiendo que el contenido está en ASCII. Pero también se puede utilizar para ordenar numéricamente utilizando las opciones adecuadas.

- [Comando Sort con ejemplos](#)
- [Opciones](#)
- [Tutoriales Linux | comando Sort GeeksforGeeks](#)

La utilidad `cut` recorta partes seleccionadas de cada línea (según se especifique en list) de cada archivo y las escribe en la salida estándar.

- [Documentación de cut](#)
- [Cheatsheet cut](#)

La utilidad `uniq` lee el `archivo_de_entrada` especificada comparando líneas adyacentes, y escribe una copia de cada línea de entrada única en el `archivo_de_salida`.

- [Documentación de uniq](#)
- [Cheatsheet uniq](#)

El comando `cat` (concatenar) se utiliza con mucha frecuencia en Linux. Lee datos del archivo y da su contenido como salida. Nos ayuda a crear, ver y concatenar archivos.

- [Comando Cat con ejemplos](#)
- [Opciones](#)
- [Tutoriales Linux | comando cat | GeeksforGeeks](#)

`echo` es un comando integrado en Linux que se utiliza para mostrar líneas de texto/cadena que se pasan como argumento. Se utiliza principalmente en scripts de shell y archivos por lotes para mostrar texto de estado o variables `ENV` en la pantalla o en un archivo.

- [Comando Echo con Ejemplos](#)
- [Curso acelerado de Linux - El comando echo](#)

El comando `fmt` sirve para formatear y optimizar el contenido de los archivos de texto. Será realmente útil cuando se trate de embellecer grandes archivos de texto estableciendo un ancho de columna y espacios uniformes.

- [Comando fmt con Ejemplos](#)

La utilidad `tr` copia la entrada estándar a la salida estándar con sustitución o eliminación de los caracteres seleccionados.

- [Documentación de tr](#)
- [Cheatsheet tr](#)

La utilidad `nl` lee líneas del archivo nombrado o de la entrada estándar si se omite el argumento archivo, aplica una operación de filtro de numeración de líneas configurable y escribe el resultado en la salida estándar.

- [Documentación de nl](#)
- [Cheatsheet nl](#)

La utilidad `wc` muestra el número de líneas, palabras y bytes que contiene cada archivo de entrada, o entrada estándar (si no se especifica ningún archivo) a la salida estándar.

- [Documentación de wc](#)
- [Cheatsheet wc](#)

Scripting

Bash scripting

- [Tutorial de secuencias de comandos Bash](#)

Powershell

- [Documentación de PowerShell](#)

Editores

Los editores son herramientas que permiten crear o editar archivos en el sistema de archivos.

Vim

Vim es un editor de texto altamente configurable construido para hacer muy eficiente la creación y modificación de cualquier tipo de texto. Se incluye como "vi" en la mayoría de los sistemas UNIX y en Apple OS X.

Vim se suministra con `vimtutor`, un tutor diseñado para describir suficientes comandos de Vim como para que puedas utilizarlo fácilmente como editor polivalente.

Visite los siguientes recursos para obtener más información:

- [Vim](#)
- [Archivos de ayuda de Vim](#)
- [Wiki de consejos sobre Vim](#)
- [Aventuras Vim](#)
- [GNU Nano](#)
- [Manual de GNU Nano](#)
- [Documentación de PowerShell](#)
- [Emacs de GNU](#)
- [Documentación de GNU Emacs](#)

Sistemas de control de versiones

Los sistemas de control de versiones y fuentes permiten a los desarrolladores seguir y controlar los cambios en el código a lo largo del tiempo. Estos servicios suelen incluir la posibilidad de realizar revisiones atómicas del código, bifurcarse a partir de puntos concretos y comparar versiones del código. Son útiles para determinar quién, qué, cuándo y por qué se hicieron cambios en el código.

Visita los siguientes recursos para obtener más información:

- [Git](#)
- [¿Qué es el control de versiones?](#)

Git

Git es un sistema de control de versiones distribuido, gratuito y de código abierto, diseñado para gestionar con rapidez y eficacia desde proyectos pequeños a muy grandes.

Visita los siguientes recursos para obtener más información:

- [Curso acelerado de Git y GitHub para principiantes](#)
- [Aprende Git con tutoriales, noticias y consejos - Atlassian](#)
- [Git Cheatsheet \(en inglés\)](#)

Servicios de alojamiento de repositorios

Cuando se trabaja en equipo, a menudo se necesita un lugar remoto donde colocar el código para que otros puedan acceder a él, crear sus propias ramas y crear o revisar pull requests. Estos servicios suelen incluir funciones de seguimiento de incidencias, revisión de código e integración continua. Algunas opciones populares son GitHub, GitLab, BitBucket y AWS CodeCommit.

Visita los siguientes recursos para obtener más información:

- [GitHub](#)
- [GitLab](#)
- [BitBucket](#)
- [Cómo elegir el mejor repositorio de código fuente](#)

GitHub

GitHub es un proveedor de alojamiento en Internet para el desarrollo de software y el control de versiones mediante Git. Ofrece las funcionalidades de control de versiones distribuidas y gestión de código fuente de Git, además de sus propias características.

Visita los siguientes recursos para obtener más información:

- [Sitio web de GitHub](#)
- [Documentación de GitHub](#)
- [Cómo utilizar Git en un equipo de desarrollo profesional](#)
- [¿Qué es GitHub?](#)
- [Git frente a GitHub: ¿Cuál es la diferencia?](#)
- [Git y GitHub para principiantes](#)
- [Git y GitHub - CS50 más allá de 2019](#)

GitLab

GitLab es un proveedor de alojamiento en Internet para el desarrollo de software y el control de versiones mediante Git. Ofrece las funcionalidades de control de versiones distribuidas y gestión de código fuente de Git, además de sus propias características.

Visita los siguientes recursos para obtener más información:

- [Sitio web de GitLab](#)
- [Documentación de GitLab](#)

Bitbucket

Bitbucket es un servicio de alojamiento y repositorio de código fuente basado en Git que es la alternativa de Atlassian a otros productos como GitHub, GitLab, etc.

Bitbucket ofrece opciones de alojamiento a través de Bitbucket Cloud (servidores de Atlassian), Bitbucket Server (local del cliente) o Bitbucket Data Centre (número de servidores en el entorno local o en la nube del cliente).

Visita los siguientes recursos para obtener más información:

- [Sitio web de Bitbucket](#)
- [Breve descripción de Bitbucket](#)
- [Primeros pasos con Bitbucket](#)
- [Usando Git con Bitbucket Cloud](#)
- [Tutorial de Bitbucket | Cómo usar Bitbucket Cloud](#)
- [Tutorial de Bitbucket | Bitbucket para principiantes](#)

¿Qué es y cómo se configura un ___?

Cortafuegos

El cortafuegos es un dispositivo de seguridad de red que supervisa y filtra el tráfico de red entrante y saliente en función de las políticas de seguridad previamente establecidas por una organización. Es una barrera que se sitúa entre una red interna privada e Internet pública. El objetivo principal de un cortafuegos es permitir la entrada de tráfico no peligroso y mantenerlo fuera.

Visite los siguientes recursos para obtener más información:

- [¿Qué es un cortafuegos?](#)
- [Tipos de cortafuegos](#)
- [Introducción al cortafuegos en las redes informáticas](#)
- [¿Qué es un cortafuegos en una red informática?](#)
- [¿Por qué necesitamos Cortafuegos?](#)
- [Cortafuegos y Seguridad en Redes – SimpliLearn](#)

Proxy de reenvío

Forward Proxy, a menudo llamado servidor proxy, es un servidor que se sitúa delante de un grupo de **máquinas cliente**. Cuando esos ordenadores hacen peticiones a sitios y servicios en Internet, el servidor proxy intercepta esas peticiones y luego se comunica con los servidores web en nombre de esos clientes, como un intermediario.

Usos comunes:

- Para bloquear el acceso a determinados contenidos
- Para proteger la identidad del cliente en línea
- Para proporcionar Internet restringido a organizaciones

Visite los siguientes recursos para obtener más información:

- [¿Qué es el proxy directo?](#)
- [Proxy directo frente a proxy inverso](#)

Proxy inverso

Un servidor proxy inverso es un tipo de servidor proxy que normalmente se sitúa detrás del cortafuegos en una red privada y dirige las peticiones de los clientes al servidor backend apropiado. Proporciona un nivel adicional de seguridad al ocultar a los clientes los detalles relacionados con el servidor, como la **dirección IP**. También se conoce como **proxy del lado del servidor**.

Usos comunes:

- Equilibrio de carga
- Aceleración web
- Seguridad y anonimato

Visite los siguientes recursos para obtener más información:

- [¿Qué es el proxy inverso?](#)
- [Documentación de NGINX](#)

Equilibrador de carga

El equilibrador de carga actúa como un **policía de tráfico** que se sienta delante de sus servidores y redirige las peticiones de los clientes a todos los servidores capaces de satisfacerlas de forma que se maximice la velocidad y la utilización de la capacidad y se garantice que ningún servidor esté sobrecargado. Si uno de los servidores se cae, el equilibrador de carga redirige el tráfico al resto de servidores en línea.

Visite los siguientes recursos para obtener más información:

- [¿Qué es el equilibrio de carga?](#)
- [Conceptos y algoritmos de equilibrio de carga](#)

Servidor caché

Un servidor caché es un **servidor o servicio de red** dedicado que actúa como un servidor que guarda localmente páginas Web u otros contenidos de Internet. Al colocar la información solicitada previamente en un almacenamiento temporal, o caché, un servidor de caché acelera el acceso a los datos y reduce la demanda de ancho de banda de una empresa.

Visite los siguientes recursos para obtener más información:

- [¿Qué es el almacenamiento en caché?](#)
- [¿Qué es un servidor de caché?](#)
- [Caché del Sitio vs Caché del Navegador vs Caché del Servidor](#)

Servidores web

Nginx

NGINX es un potente servidor web y utiliza una arquitectura sin hilos y basada en eventos que le permite superar a Apache si se configura correctamente. También puede hacer otras cosas importantes, como equilibrar la carga, almacenar HTTP en caché o utilizarse como proxy inverso.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [NGINX explicado en 100 segundos](#)

Tomcat

Tomcat es una implementación de código abierto de las especificaciones Jakarta Servlet, Jakarta Server Pages, Jakarta Expression Language, Jakarta WebSocket, Jakarta Annotations y Jakarta Authentication. Estas especificaciones forman parte de la plataforma Jakarta EE.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Tomcat](#)
- [Documentación oficial \(Tomcat 10.0\)](#)
- [Apache Tomcat](#)

Apache

Apache es un servidor HTTP gratuito y de código abierto, disponible en muchos sistemas operativos, pero utilizado principalmente en distribuciones Linux. Es una de las opciones más populares para los desarrolladores web, ya que representa más del 30% de todos los sitios web, según estimaciones de W3Techs.

Visite los siguientes recursos para obtener más información:

- [Sitio web del servidor Apache](#)
- [¿Qué es el servidor web Apache?](#)

MS IIS

Internet Information Services (IIS) para Windows® Server es un servidor Web flexible, seguro y manejable para alojar cualquier cosa en la Web.

Visite los siguientes recursos para obtener más información:

- [Sitio Web oficial](#)
- [Aprenda Windows Web Server IIS](#)

Contenedores

Los contenedores son una construcción en la que se utilizan **cgroups**, **namespaces** y **chroot** para encapsular y aislar completamente un proceso. Este proceso encapsulado, denominado imagen de contenedor, comparte el núcleo del host con otros contenedores, lo que permite que los contenedores sean significativamente más pequeños y rápidos que las máquinas virtuales.

Estas imágenes están diseñadas para la portabilidad, lo que permite realizar pruebas locales completas de una imagen estática e implantarlas fácilmente en una plataforma de gestión de contenedores.

Visite los siguientes recursos para obtener más información:

- [¿Qué son los contenedores?](#)
- [¿Qué es un contenedor?](#)
- [¿Qué son los contenedores?](#)
- [Artículos sobre contenedores - The New Stack](#)

Docker

Docker es una plataforma para trabajar con aplicaciones en contenedores. Entre sus características se encuentran un demonio y un cliente para gestionar e interactuar con contenedores, registros para almacenar imágenes y una aplicación de escritorio para empaquetar todas estas características.

Visite los siguientes recursos para obtener más información:

- [Documentación de Docker](#)
- [Tutorial de Docker](#)
- [Docker simplificado en 55 segundos](#)

Vea el roadmap.sh de Docker para más información.

LXC

LXC es un conocido tiempo de ejecución de contenedores de Linux que consiste en herramientas, plantillas y vinculaciones de bibliotecas y lenguajes. Es de muy bajo nivel, muy flexible y cubre casi todas las características de contención soportadas por el kernel upstream.

Visita los siguientes recursos para obtener más información:

- [Sitio web de LXC](#)
- [Documentación de LXC](#)
- [Introducción a los contenedores LXC](#)

Proveedores de nube

Los proveedores de nube proporcionan una capa de API para abstraer la infraestructura y aprovisionarla en función de los límites de seguridad y facturación. La nube se ejecuta en servidores de centros de datos, pero las abstracciones dan hábilmente la apariencia de interactuar con una única "plataforma" o gran aplicación. La capacidad de aprovisionar, configurar y asegurar rápidamente los recursos con los proveedores de la nube ha sido clave tanto para el enorme éxito, como para la complejidad, de los DevOps modernos.

Visite los siguientes recursos para obtener más información:

- [Proveedor de servicios en la nube](#)
- [¿Qué son los proveedores de servicios en nube?](#)

AWS

Amazon Web Services es la plataforma de computación en nube líder del mercado desde 2011, por delante de Azure y Google Cloud. AWS ofrece más de 200 servicios con centros de datos repartidos por todo el mundo.

El servicio AWS es una plataforma en línea que proporciona soluciones de computación en nube escalables y rentables. Se trata de una plataforma en la nube ampliamente adoptada que ofrece varias operaciones bajo demanda como potencia de cálculo, almacenamiento de bases de datos, entrega de contenido, etc.

Visite los siguientes recursos para obtener más información:

- [Sitio web de AWS](#)
- [Documentación de AWS](#)
- [Registrarse en AWS](#)
- [Cómo aprender AWS](#)
- [Guía de aceleración de AWS](#)
- [Aspectos esenciales de Cloud Practitioner](#)
- [Guía de AWS por SimpliLearn](#)
- [Tutorial de AWS para principiantes](#)
- [Curso de AWS para principiantes](#)
- [Curso DevOps en AWS](#)
- [AWS Practitioner](#)

Azure

Microsoft Azure es un servicio de computación en nube operado por Microsoft. Azure ofrece actualmente más de 200 productos y servicios en la nube.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Azure](#)
- [Documentación oficial](#)
- [Guía de inicio de Azure](#)
- [Conozca Azure](#)

Google Cloud

Google Cloud es la oferta de servicios de computación en nube de Google, que proporciona más de 150 productos/servicios entre los que elegir. Consiste en un conjunto de activos físicos, como ordenadores y discos duros, y recursos virtuales, como máquinas virtuales (VM), que se encuentran en los centros de datos de Google. Se ejecuta en la misma infraestructura que Google utiliza internamente para sus productos de usuario final, como la Búsqueda, Gmail, Google Drive y YouTube.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Google Cloud](#)
- [Documentación oficial](#)
- [Guía de introducción a Google Cloud](#)
- [Curso completo de Coursera sobre Google Cloud](#)
- [Google Cloud por Edureka en You-Tube](#)
- [5 consejos para convertirse en arquitecto profesional certificado de Google Cloud](#)

DigitalOcean

DigitalOcean es un servicio de computación en nube que ofrece productos y servicios de computación, almacenamiento, bases de datos gestionadas, contenedores e imágenes y redes.

Visite los siguientes recursos para obtener más información:

- [Sitio web de DigitalOcean](#)
- [Documentación oficial](#)
- [Guía de inicio de DigitalOcean](#)

Alibaba Cloud

Alibaba Cloud es un servicio de computación en nube que ofrece más de 100 productos y servicios con centros de datos en 24 regiones y 74 zonas de disponibilidad en todo el mundo.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Alibaba Cloud](#)
- [Documentación oficial](#)
- [Guía de inicio de Alibaba Cloud](#)

Heroku

Heroku es una plataforma en la nube como servicio subsidiario de Salesforce. Heroku admite oficialmente Node.js, Ruby, Java, PHP, Python, Go, Scala y Clojure, junto con cualquier lenguaje que se ejecute en Linux a través de un paquete de compilación de terceros.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Heroku](#)
- [Documentación oficial](#)

- [Guía de inicio de Heroku](#)

Linode

Linode es un servicio de computación en nube propiedad de Akamai Technologies. Linode se posiciona como una alternativa a AWS, GCP y Azure al ofrecer servicios básicos sin complejidad para la mayoría de las cargas de trabajo.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Linode](#)
- [Documentación oficial](#)
- [Guía de inicio de Linode](#)

Vultr

Vultr es un servicio de computación en nube centrado en infraestructuras, disponible en 25 ubicaciones de todo el mundo. Vultr compute ofrece 100% SSD y vCPUs Intel de alto rendimiento.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Vultr](#)
- [Documentación oficial](#)

Protocolos de red

Como ingeniero DevOps necesitarás entender los fundamentos de los protocolos de red, cómo funcionan y cómo se utilizan en el mundo real. Para empezar, deberías aprender sobre **TCP/IP**, **HTTP**, **HTTPS**, **FTP**, **SSH**, **SMTP**, **DNS**, **DHCP**, **NTP**.

Estos son algunos de los recursos para aprender sobre SSH:

- [Introducción a SSH](#)
- [¿Qué es SSH?](#)
- [SFTP usando SSH](#)
- [Guía completa de OpenSSH](#)

Visite lo siguiente para aprender sobre SSL/TLS:

- [Cloudflare - ¿Qué es SSL?](#)
- [Cloudflare - ¿Qué es TLS?](#)
- [Wikipedia - SSL/TLS](#)
- [SSH vs SSL vs TLS](#)

Más información sobre HTTP y HTTPS:

- [Todo lo que necesita saber sobre HTTP](#)
- [¿Qué es HTTP?](#)
- [Visión general de HTTP](#)
- [HTTP/3 de la A a la Z: Conceptos básicos](#)
- [Curso acelerado y exploración de HTTP](#)
- [¿Qué es HTTPS?](#)
- [Por qué es importante HTTPS](#)
- [Activar HTTPS en tus servidores](#)
- [Cómo funciona HTTPS \(cómic\)](#)

Aquí tienes algunos recursos para aprender sobre DNS:

- [¿Qué es DNS?](#)
- [Cómo funciona DNS \(cómic\)](#)
- [DNS y su funcionamiento](#)
- [Registros DNS](#)

FTP

El Protocolo de Transferencia de Archivos (FTP) es un protocolo de comunicación de capa de aplicación basado en TCP/IP que ayuda a transferir archivos entre sistemas de archivos locales y remotos a través de la red. Para transferir un archivo, se utilizan en paralelo 2 conexiones TCP (conexión de control y conexión de datos).

Visite los siguientes recursos para obtener más información:

- [¿Qué es FTP?](#)
- [FTP vs SFTP vs FTPS](#)

HTTP

HTTP es el protocolo de comunicación de capa de aplicación basado en **TCP/IP** que estandariza la forma en que el cliente y el servidor se comunican entre sí. Define cómo se solicita y transmite el contenido a través de Internet.

Visite los siguientes recursos para obtener más información:

- [Todo lo que necesita saber sobre HTTP](#)
- [¿Qué es HTTP?](#)
- [Visión general de HTTP](#)
- [HTTP/3 de la A a la Z: Conceptos básicos](#)
- [Curso acelerado y exploración de HTTP](#)

SSL/TLS

Secure Sockets Layer (SSL) y **Transport Layer Security (TLS)** son protocolos criptográficos utilizados para proporcionar seguridad en las comunicaciones por Internet. Estos protocolos cifran los datos que se transmiten a través de la red, de modo que cualquiera que intente interceptar los paquetes no podrá interpretar los datos. Una diferencia que es importante conocer es que SSL está ahora obsoleto debido a fallos de seguridad, y la mayoría de los navegadores web modernos ya no lo soportan. Sin embargo, TLS sigue siendo seguro y está ampliamente soportado, por lo que es preferible utilizar TLS.

Visite los siguientes recursos para obtener más información:

- [Cloudflare - ¿Qué es SSL?](#)
- [Cloudflare - ¿Qué es TLS?](#)
- [Wikipedia - SSL/TLS](#)
- [SSH vs SSL vs TLS](#)

DNS

DNS (**D**omain **N**ame **S**ystem) es la guía telefónica de Internet. Los humanos acceden a la información en línea a través de nombres de dominio, como nytimes.com o espn.com. Los navegadores web interactúan a través de direcciones IP (Protocolo de Internet). El DNS traduce los nombres de dominio a direcciones IP para que los navegadores puedan cargar los recursos de Internet.

Visite los siguientes recursos para obtener más información:

- [¿Qué es el DNS?](#)
- [Cómo funciona el DNS \(cómic\)](#)
- [DNS y ¿Cómo funciona?](#)
- [Registros DNS](#)

HTTPS

HTTPS (**H**ypertext **T**ransfer **P**rotocol **S**ecure) es la versión segura de HTTP, que es el protocolo principal utilizado para enviar datos entre un navegador web y un sitio web.

HTTPS = HTTP + SSL/TLS

Visite los siguientes recursos para obtener más información:

- [¿Qué es HTTPS?](#)
- [Por qué es importante HTTPS](#)
- [Activar HTTPS en sus servidores](#)
- [Cómo funciona HTTPS \(cómic\)](#)

SSH

SSH (**S**ecure **S**hell) es un protocolo de comunicación de red que permite a dos ordenadores comunicarse a través de una red insegura. Es una alternativa segura a los protocolos de inicio de sesión no protegidos (como telnet, rlogin) y a los métodos inseguros de transferencia de archivos (como FTP). Se utiliza sobre todo para el inicio de sesión remoto seguro y la transferencia de archivos.

SFTP = FTP + SSH

Visite los siguientes recursos para obtener más información:

- [Introducción a SSH](#)
- [¿Qué es SSH?](#)
- [SFTP usando SSH](#)

Modelo OSI

El modelo de Interconexión de Sistemas Abiertos (OSI) es un modelo **conceptual** compuesto por 7 capas, que se propuso para estandarizar la comunicación entre dispositivos a través de la red. Fue el primer modelo estándar para las comunicaciones en red, adoptado por las principales empresas de informática y telecomunicaciones a principios de los años 80.

Visite los siguientes recursos para obtener más información:

- [¿Qué es el modelo OSI?](#)
- [Modelo OSI](#)
- [¿Qué es el modelo TCP/IP?](#)
- [Modelo OSI vs TCP/IP](#)

Correo electrónico

El correo electrónico es un método de comunicación que utiliza dispositivos electrónicos para enviar mensajes a través de redes informáticas. "Correo electrónico" se refiere tanto al sistema de entrega como a los mensajes individuales que se envían y reciben.

Visite los siguientes recursos para obtener más información:

- [¿Qué es un correo electrónico?](#)

Lista blanca vs Lista gris

La lista blanca es un proceso que consiste en añadir un correo electrónico a una lista de remitentes aprobados, de modo que los correos electrónicos de ese remitente nunca se mueven a la carpeta de spam. Esto indica a un servidor de correo electrónico que mueva los mensajes directamente a la bandeja de entrada.

Las **listas grises** son un método para proteger a los usuarios de correo electrónico contra el spam. Un agente de transferencia de correo (MTA) que utilice listas grises "rechazará temporalmente" cualquier correo electrónico de un remitente que no reconozca. Si el correo es legítimo, el servidor de origen lo intentará de nuevo después de un tiempo, y el correo será aceptado si ha transcurrido el tiempo suficiente.

Visite los siguientes recursos para obtener más información:

- [Introducción básica a las listas blancas](#)
- [Introducción detallada a las listas grises](#)

SMTP

El correo electrónico se está convirtiendo en uno de los servicios más valiosos de Internet en la actualidad. La mayoría de los sistemas de Internet utilizan SMTP como método para transferir correo de un usuario a otro. SMTP es un protocolo push y se utiliza para enviar el correo, mientras que POP (protocolo de oficina de correos) o IMAP (protocolo de acceso a mensajes de Internet) se utilizan para recuperar esos correos en el lado del receptor.

SMTP es un protocolo de capa de aplicación. El cliente que desea enviar el correo abre una conexión TCP con el servidor SMTP y envía el correo a través de la conexión. El servidor SMTP está siempre a la escucha. Tan pronto como escucha una conexión TCP de cualquier cliente, el proceso SMTP inicia una conexión a través del puerto 25. Tras establecer con éxito una conexión TCP, el proceso cliente envía el correo instantáneamente.

Visite los siguientes recursos para obtener más información:

- [¿Qué es el Protocolo Simple de Transferencia de Correo \(SMTP\)?](#)

IMAP

IMAP (puerto 143) o IMAPS (puerto 993) te permiten acceder a tu correo electrónico estés donde estés, desde cualquier dispositivo. Cuando lees un mensaje de correo electrónico utilizando IMAP, en realidad no lo estás descargando o almacenando en tu ordenador, sino que lo estás leyendo desde el servicio de correo electrónico. Como resultado, puedes consultar tu correo electrónico desde distintos dispositivos, en cualquier parte del mundo: tu teléfono, un ordenador, el ordenador de un amigo.

IMAP sólo descarga un mensaje cuando haces clic en él, y los archivos adjuntos no se descargan automáticamente. De este modo, puedes consultar tus mensajes mucho más rápidamente que con POP.

Visita los siguientes recursos para obtener más información:

- [Wikipedia: Protocolo de acceso a mensajes de Internet](#)
- [Qué es IMAP y Cómo Usarlo | Tutorial de Correo Electrónico](#)

POP3

POP3 (puerto 110) o POP3s (puerto 995) son las siglas de The Post Office Protocol. Es un protocolo estándar de Internet que utilizan los clientes locales de software de correo electrónico para recuperar mensajes de un servidor de correo remoto a través de una conexión TCP/IP.

Los servidores de correo electrónico alojados por proveedores de servicios de Internet también utilizan POP3 para recibir y retener los correos electrónicos destinados a sus abonados. Periódicamente, estos suscriptores utilizarán el software cliente de correo electrónico para comprobar su buzón en el servidor remoto y descargar los correos electrónicos dirigidos a ellos.

Una vez que el cliente de correo electrónico ha descargado los correos, normalmente se borran del servidor, aunque algunos clientes de correo electrónico permiten a los usuarios especificar que los correos se copien o guarden en el servidor durante un periodo de tiempo.

Visite los siguientes recursos para obtener más información:

- [¿Qué es POP3?](#)

DMARC

DMARC significa Domain-based Message Authentication, Reporting, and Conformance, es un método de autenticación en el correo electrónico que se construye para proteger el correo electrónico de dominio de direcciones de correo electrónico no válidas o comúnmente conocido como email spoofing, ataques de correo electrónico, phishing, estafas, y otras actividades de amenaza.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial de DMARC](#)

SPF

Sender Policy Framework (SPF) se utiliza para autenticar al remitente de un correo electrónico. Con un registro SPF, los proveedores de servicios de Internet pueden verificar que un servidor de correo está autorizado a enviar correo electrónico para un dominio específico. Un registro SPF es un registro DNS TXT que contiene una lista de las direcciones IP autorizadas a enviar correo electrónico en nombre de su dominio.

Visita los siguientes recursos para obtener más información:

- [¿Qué es un registro SPF de DNS?](#)
- [Descripción general de SPF](#)

Claves de dominio

DomainKeys Identified Mail (DKIM) es un método de autenticación de correo electrónico diseñado para detectar direcciones de remitente falsificadas en el correo electrónico (email spoofing), una técnica utilizada a menudo en el phishing y el spam por correo electrónico.

Visite los siguientes recursos para obtener más información:

- [DomainKeys Identified Mail](#)

Serverless

- [¿Qué es Serverless?](#)

Cloudflare

- [Documentación de Cloudflare](#)

AWS Lambda

- [Introducción a AWS Lambda](#)

Azure Functions

- [Descripción general de Azure Functions](#)

Funciones Gcp

Vercel

Netlify

Aprovisionamiento de infraestructuras

Las herramientas de esta categoría se utilizan para aprovisionar infraestructura en proveedores de nube. Esto incluye DNS, redes, políticas de seguridad, servidores, contenedores y toda una serie de construcciones específicas del proveedor. En esta categoría, se recomienda encarecidamente el uso de herramientas independientes del proveedor de la nube. Estos conocimientos pueden aplicarse a la mayoría de los proveedores de nube, y los lenguajes más específicos de dominio suelen tener menos alcance.

Terraform

Terraform es una herramienta de Infraestructura como Código (IaC) de código abierto extremadamente popular que se puede utilizar con muchas API de proveedores de servicios y nubes diferentes. Terraform se centra en un enfoque inmutable de la infraestructura, con un centro de archivos de estado terraform para el seguimiento del estado de su infraestructura en el mundo real.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Terraform](#)
- [Documentación de Terraform](#)
- [Tutoriales de Terraform](#)
- [Vídeo de introducción a Terraform](#)
- [Sitio web de Terraform CDK](#)
- [¿Qué es el CDKTF?](#)
- [Guía de inicio de CDKTF](#)
- [Ejemplos de CDKTF](#)
- [Cómo escalar tu infraestructura Terraform](#)

Pulumi

Pulumi es una herramienta de Infraestructura como Código de código abierto que se puede escribir en TypeScript, JavaScript, Python, Go, .NET, Java y YAML para modelar la infraestructura de la nube.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Pulumi](#)
- [Documentación oficial](#)
- [Guía de inicio de Pulumi](#)

CloudFormation

CloudFormation es el servicio de AWS que ayuda a definir colecciones de recursos de AWS. CloudFormation le permite modelar, aprovisionar y administrar recursos de AWS y de terceros tratando la infraestructura como código.

Visite los siguientes recursos para obtener más información:

- [Sitio web de AWS CloudFormation](#)
- [Documentación oficial](#)
- [Guía de introducción a AWS CloudFormation](#)

- [Plantillas de muestra de CloudFormation](#)

AWS CDK

AWS Cloud Development Kit (AWS CDK) es un marco de desarrollo de software de código abierto que se utiliza para aprovisionar recursos de infraestructura en la nube de forma segura y repetible a través de AWS CloudFormation. AWS CDK ofrece la flexibilidad de escribir infraestructura como código en lenguajes populares como JavaScript, TypeScript, Python, Java, C# y Go.

Visite los siguientes recursos para obtener más información:

- [Sitio web de AWS CDK](#)
- [Documentación oficial](#)
- [¿Qué es AWS CDK?](#)
- [Guía de introducción a AWS SDK](#)
- [Ejemplos de AWS CDK](#)

Gestión de la configuración

La gestión de la configuración es un proceso de ingeniería de sistemas para establecer la coherencia de los atributos de un producto a lo largo de su vida útil. En el mundo de la tecnología, la gestión de la configuración es un proceso de gestión de TI que realiza un seguimiento de los elementos de configuración individuales de un sistema de TI. Los sistemas de TI se componen de activos de TI que varían en granularidad. Un activo de TI puede representar una pieza de software, un servidor o un grupo de servidores. Lo que sigue se centra en la gestión de la configuración, ya que se aplica directamente a los activos de software de TI y al CI/CD de activos de software.

La gestión de la configuración de software es un proceso de ingeniería de sistemas que rastrea y supervisa los cambios en los metadatos de configuración de un sistema de software. En el desarrollo de software, la gestión de la configuración se utiliza habitualmente junto con el control de versiones y la infraestructura CI/CD. Este artículo se centra en su aplicación y uso modernos en entornos de software ágiles de CI/CD.

Visite los siguientes recursos para obtener más información:

- [¿Qué es la gestión de la configuración?](#)

Ansible

Ansible es una herramienta de gestión de configuración, despliegue de aplicaciones y aprovisionamiento de código abierto que utiliza su propio lenguaje declarativo en YAML. Ansible es agentless, lo que significa que sólo necesita conexiones remotas a través de SSH o Windows Remote Management a través de Powershell para funcionar.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Ansible](#)
- [Documentación oficial](#)
- [Guía de introducción a Ansible](#)
- [Curso completo de Ansible para principiantes](#)

Chef

Surgida en 2009, **Chef** (ahora conocida como Progress Chef) es una de las primeras herramientas de gestión de la configuración en ganar popularidad. Las "recetas" de Chef están escritas en Ruby, en un estilo principalmente **declarativo**.

Chef requiere que se instale un cliente en el servidor que se está gestionando. Este cliente consulta regularmente un servidor Chef para determinar cuál debe ser su configuración. También está disponible Chef-Solo, una versión de Chef que permite el aprovisionamiento de un único nodo ejecutando Chef localmente.

Un principio clave del diseño de recetas de Chef es el concepto de **idempotencia**. Todas las recetas de Chef deben poder ejecutarse varias veces y producir el mismo resultado; esto es especialmente necesario en los casos en los que se utiliza el modelo cliente/servidor mencionado anteriormente. Este patrón de gestión de la configuración es muy influyente para futuras herramientas declarativas como Terraform y Cloud Formation.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Chef](#)
- [Tutorial de Chef](#)

Puppet

Puppet, un motor administrativo automatizado para sus sistemas Linux, Unix y Windows, realiza tareas administrativas (como añadir usuarios, instalar paquetes y actualizar configuraciones de servidores) basándose en una especificación centralizada.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Puppet](#)
- [Documentación oficial](#)
- [Introducción a Puppet](#)

CI/CD

CI/CD es un método para entregar frecuentemente aplicaciones a los clientes mediante la introducción de la automatización en las etapas de desarrollo de aplicaciones. Los principales conceptos atribuidos a CI/CD son la integración continua, la entrega continua y el despliegue continuo. CI/CD es una solución a los problemas que la integración de código nuevo puede causar a los equipos de desarrollo y operaciones.

En concreto, CI/CD introduce la automatización continua y la supervisión permanente a lo largo del ciclo de vida de las aplicaciones, desde las fases de integración y prueba hasta la entrega y el despliegue. En conjunto, estas prácticas conectadas suelen denominarse "canalización de CI/CD" y se apoyan en equipos de desarrollo y operaciones que trabajan juntos de forma ágil con un enfoque DevOps o de ingeniería de fiabilidad del sitio (SRE).

Visite los siguientes recursos para obtener más información:

- [CI vs CD](#)
- [¿Qué es CI/CD?](#)
- [Canalización CI/CD: Una breve introducción](#)
- [DevOps CI/CD explicado en 100 segundos](#)

GitLab CI

GitLab ofrece un servicio de CI/CD que puede utilizarse como oferta SaaS o autogestionarse utilizando tus propios recursos. Puedes utilizar GitLab CI con cualquier repositorio alojado en GitLab, o cualquier repositorio de BitBucket Cloud o GitHub en los niveles GitLab Premium autogestionado, GitLab Premium SaaS y superiores.

Visita los siguientes recursos para obtener más información:

- [Sitio web de GitLab](#)
- [Documentación de GitLab](#)
- [Primeros pasos con GitLab CI](#)
- [Tutoriales de aprendizaje de GitLab](#)
- [Ejemplos de GitLab CI/CD](#)

Acciones de GitHub

Automatiza, personaliza y ejecuta tus flujos de trabajo de desarrollo de software directamente en tu repositorio con GitHub Actions. Puedes descubrir, crear y compartir acciones para realizar cualquier trabajo que desees, incluyendo CI/CD, y combinar acciones en un flujo de trabajo completamente personalizado.

Visita los siguientes recursos para obtener más información:

- [Documentación de GitHub Actions](#)
- [Acciones de GitHub](#)
- [Acciones GitHub - Mejora tu flujo GitHub](#)
- [Automatiza tu flujo de trabajo con GitHub Actions](#)

CircleCI

CircleCI es un servicio de CI/CD que puede integrarse con los repositorios de GitHub, BitBucket y GitLab. El servicio que puede ser utilizado como una oferta SaaS o autogestionado utilizando sus propios recursos.

Visite los siguientes recursos para obtener más información:

- [Sitio web de CircleCI](#)
- [Documentación de CircleCI](#)
- [Tutorial de configuración](#)

Jenkins

Jenkins es un servidor de automatización CI/CD de código abierto. Jenkins se utiliza principalmente para la creación de proyectos, la ejecución de pruebas, el análisis estático de código y los despliegues.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Jenkins](#)
- [Manual oficial de Jenkins](#)
- [Guía de inicio de Jenkins](#)

Travis CI

Travis CI es un servicio CI/CD que se utiliza principalmente para construir y probar proyectos alojados en BitBucket y GitHub. Los proyectos de código abierto pueden utilizar Travis CI de forma gratuita.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Travis CI](#)
- [Documentación de Travis CI](#)
- [Tutorial de Travis CI](#)

Drone

Drone es un servicio CI/CD ofrecido por **Harness**. Cada compilación se ejecuta en un contenedor Docker aislado, y Drone se integra con muchos repositorios populares de gestión de código fuente como GitHub, BitBucket y GitLab.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Drone](#)
- [Documentación oficial](#)
- [Guía de inicio de Drone](#)

TeamCity

TeamCity es un servicio CI/CD proporcionado por JetBrains. TeamCity puede utilizarse como oferta SaaS o autogestionarse utilizando sus propios recursos.

Visite los siguientes recursos para obtener más información:

- [Sitio web de TeamCity](#)
- [Documentación oficial](#)
- [Tutoriales de TeamCity](#)
- [Portal de aprendizaje de TeamCity](#)

Azure DevOps

Azure DevOps ha sido desarrollado por Microsoft como un servicio completo de gestión del ciclo de vida de las aplicaciones y CI/CD. Azure DevOps proporciona servicios para desarrolladores que permiten a los equipos planificar el trabajo, colaborar en el desarrollo de código y crear e implementar aplicaciones.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Azure DevOps](#)
- [Documentación oficial](#)
- [Guía de inicio de Azure DevOps](#)

Gestión de secretos

La gestión de secretos es un aspecto importante de DevOps, ya que implica almacenar y gestionar de forma segura la información confidencial, como contraseñas, claves API y otros secretos, que utilizan las aplicaciones y la infraestructura.

Hay varias formas de gestionar secretos en un entorno de nube:

- **Almacenes secretos:** Un almacén de secretos es una base de datos o servicio especializado diseñado para almacenar y gestionar secretos de forma segura. Algunos ejemplos de almacenes secretos son Hashicorp Vault, AWS Secrets Manager y Google Cloud Secret Manager.
- **Cifrado:** Los secretos se pueden cifrar utilizando diversos algoritmos y protocolos de cifrado, como AES, RSA y PGP. Los secretos cifrados pueden almacenarse en distintas ubicaciones, como un sistema de archivos, una base de datos o un servicio de almacenamiento en la nube.
- **Controles de acceso:** El acceso a los secretos debe restringirse únicamente a aquellos usuarios o sistemas que los necesiten, utilizando técnicas como controles de acceso basados en roles, autenticación multifactor y principios de mínimo privilegio.

La gestión eficaz de los secretos es esencial para mantener la seguridad y la integridad de un entorno DevOps. Es importante revisar y actualizar periódicamente las prácticas de gestión de secretos para garantizar que los secretos se almacenan y gestionan de forma segura.

Visite los siguientes recursos para obtener más información:

- [¿Cómo gestionar secretos en aplicaciones web?](#)
- [Guía de gestión de secretos - enfoques, herramientas de código abierto, productos comerciales, retos y preguntas](#)
- [Arquitecturas de gestión de secretos: Encontrar el equilibrio entre seguridad y complejidad](#)

Vault

Vault es una herramienta para almacenar y gestionar de forma segura secretos, como contraseñas, claves API y otra información sensible. Está desarrollada y mantenida por Hashicorp y está disponible como software de código abierto.

Vault está diseñado para ser altamente escalable y flexible, con una amplia gama de características para la gestión de secretos, incluyendo:

- **Cifrado:** Vault utiliza algoritmos y protocolos de cifrado, como AES y RSA, para almacenar secretos de forma segura.
- **Controles de acceso:** Vault admite controles de acceso basados en funciones y autenticación multifactor para garantizar que sólo los usuarios o sistemas autorizados puedan acceder a los secretos.
- **Rotación de secretos:** Vault admite la rotación automática de secretos, lo que permite rotarlos periódicamente para reducir el riesgo de acceso no autorizado.
- **Auditoría:** Vault ofrece funciones de auditoría, lo que permite a los administradores rastrear y supervisar el acceso a los secretos.

Vault se utiliza comúnmente en entornos DevOps para almacenar y gestionar secretos de forma segura, y a menudo se utiliza junto con otras herramientas, como Kubernetes y Helm, para automatizar el despliegue y la gestión de aplicaciones nativas de la nube.

Visite los siguientes recursos para obtener más información:

- [Vault - Sitio web oficial](#)

SOPS

SOPS (Secrets OPERATIONs) es una herramienta de código abierto para almacenar y gestionar de forma segura secretos, como contraseñas, claves API y otra información sensible. Está desarrollada y mantenida por Mozilla y está disponible como herramienta de línea de comandos y como biblioteca para su integración en otras aplicaciones.

SOPS está diseñado para ser fácil de usar y altamente seguro, con una serie de características para la gestión de secretos, incluyendo:

- Cifrado: SOPS utiliza algoritmos y protocolos de cifrado, como AES y RSA, para almacenar secretos de forma segura.
- Controles de acceso: SOPS admite controles de acceso basados en funciones y autenticación multifactor para garantizar que sólo los usuarios o sistemas autorizados puedan acceder a los secretos.
- Rotación de secretos: SOPS admite la rotación automática de secretos, lo que permite que los secretos se roten regularmente para reducir el riesgo de acceso no autorizado.
- Auditoría: SOPS proporciona capacidades de auditoría, permitiendo a los administradores rastrear y supervisar el acceso a los secretos.

SOPS se utiliza comúnmente en entornos DevOps para almacenar y gestionar secretos de forma segura, y a menudo se utiliza junto con otras herramientas, como Kubernetes y Helm, para automatizar el despliegue y la gestión de aplicaciones nativas de la nube.

Visite los siguientes recursos para obtener más información:

- [Mozilla SOPS - Sitio web oficial](#)

Herramientas específicas para la nube

Existen varias herramientas específicas de la nube para almacenar y administrar secretos de forma segura, como:

- AWS Secrets Manager: AWS Secrets Manager es un servicio proporcionado por Amazon Web Services (AWS) para almacenar y gestionar secretos de forma segura. Ofrece funciones como la rotación automática de secretos y la integración con otros servicios de AWS.
- Google Cloud Secret Manager: Google Cloud Secret Manager es un servicio proporcionado por Google Cloud para almacenar y gestionar secretos de forma segura. Ofrece funciones como la rotación automática de secretos y la integración con otros servicios de Google Cloud.
- Azure Key Vault: Azure Key Vault es un servicio proporcionado por Microsoft Azure para almacenar y gestionar secretos de forma segura. Ofrece funciones como la rotación automática de secretos y la integración con otros servicios de Azure.

Estas herramientas específicas de la nube están diseñadas para utilizarse junto con aplicaciones e infraestructuras basadas en la nube y suelen integrarse con otros servicios en la nube, como plataformas de orquestación de contenedores y canalizaciones de entrega continua.

Visite los siguientes recursos para obtener más información:

- [AWS Secrets Manager - Amazon Web Services](#)
- [Google Cloud Secret Manager - Google Cloud](#)
- [Azure Key Vault - Microsoft Azure](#)

Sealed Secrets

Sealed Secrets es una herramienta para almacenar y gestionar secretos de forma segura en un entorno Kubernetes. Está desarrollada y mantenida por Bitnami y está disponible como software de código abierto.

En un flujo de trabajo Sealed Secrets, los secretos se cifran utilizando una clave pública y se almacenan como secretos sellados en un repositorio Git. A continuación, los secretos sellados pueden desplegarse en un clúster Kubernetes, donde se descifran mediante una clave privada y se ponen a disposición de las aplicaciones y la infraestructura que los necesitan.

Sealed Secrets está diseñado para ser altamente seguro y fácil de usar, con una serie de características para la gestión de secretos, incluyendo:

- Cifrado: Sealed Secrets utiliza algoritmos y protocolos de cifrado, como RSA, para almacenar secretos de forma segura.
- Controles de acceso: Sealed Secrets admite controles de acceso basados en funciones y autenticación multifactor para garantizar que sólo los usuarios o sistemas autorizados puedan acceder a los secretos.
- Rotación de secretos: Sealed Secrets admite la rotación automática de secretos, lo que permite rotarlos periódicamente para reducir el riesgo de acceso no autorizado.
- Auditoría: Sealed Secrets ofrece funciones de auditoría, lo que permite a los administradores rastrear y supervisar el acceso a los secretos.

Sealed Secrets se utiliza habitualmente en entornos Kubernetes para almacenar y gestionar secretos de forma segura, y a menudo se utiliza junto con otras herramientas, como Helm, para automatizar el despliegue y la gestión de aplicaciones nativas de la nube.

Visite los siguientes recursos para obtener más información:

- [Sealed Secrets - Bitnami](#)

Supervisión de infraestructuras

La supervisión se refiere a la práctica de hacer visibles el rendimiento y el estado de la infraestructura. Esta sección contiene herramientas comunes utilizadas para la monitorización.

Se trata de un ámbito en el que abundan los proveedores: tenga cuidado al estudiar materiales exclusivamente de un determinado producto o proyecto, ya que hay muchas opiniones y estrategias contradictorias en uso. No existe una solución única para las aplicaciones más complejas que se conectan a Internet, por lo que conocer los pros y los contras de estas herramientas le será útil para planificar cómo supervisar un sistema para un objetivo determinado.

Visite los siguientes recursos para obtener más información:

- [Mejores prácticas para optimizar la monitorización de la infraestructura dentro de los equipos DevOps](#)
- [Siete pasos para una supervisión eficaz de la infraestructura nativa de la nube](#)

Grafana

Grafana es la plataforma de código abierto para la monitorización y la observabilidad. Le permite consultar, visualizar, alertar y comprender sus métricas sin importar dónde estén almacenadas.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Grafana](#)
- [Documentación oficial de Grafana](#)
- [Comunidad de Grafana](#)
- [Webinars y vídeos de Grafana](#)

Prometheus

Prometheus es una aplicación de software libre utilizada para la monitorización de eventos y alertas. Registra métricas en tiempo real en una base de datos de series temporales creada mediante un modelo pull HTTP, con consultas flexibles y alertas en tiempo real.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Prometheus](#)
- [Documentación oficial](#)
- [Introducción a Prometheus](#)

Datadog

Datadog es una plataforma de supervisión y análisis para aplicaciones a gran escala. Abarca la supervisión de la infraestructura, la supervisión del rendimiento de las aplicaciones, la gestión de registros y la supervisión de la experiencia del usuario. Datadog agrega datos a través de toda su pila con más de 400 integraciones para la solución de problemas, alertas y gráficos.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Datadog](#)

- [Documentación oficial](#)

Zabbix

Zabbix es una solución de monitorización de código abierto de clase empresarial para la monitorización de redes y aplicaciones de millones de métricas.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Zabbix](#)
- [Documentación oficial](#)
- [Hoja de ruta de Zabbix](#)

Monitorización de aplicaciones

La supervisión de aplicaciones se refiere a la práctica de hacer visibles el estado y el rendimiento de una aplicación determinada. Esto puede incluir detalles como stacktraces, registros de errores y la línea de código implicada en un fallo determinado. Cuando se combina con la monitorización de la infraestructura, puede proporcionar una imagen completa de lo que ocurre en el sistema y por qué.

Visite los siguientes recursos para obtener más información:

- [Aplicación de técnicas de supervisión básicas frente a avanzadas](#)
- [Por qué las aplicaciones heredadas también necesitan su supervisión](#)

Jaeger

Jaeger es un sistema de rastreo distribuido de extremo a extremo y de código abierto que nos permite supervisar y solucionar problemas de transacciones en sistemas distribuidos complejos.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Jaeger](#)
- [Documentación oficial](#)

New Relic

New Relic es donde los equipos de desarrollo, operaciones, seguridad y negocios resuelven los problemas de rendimiento del software con datos.

Visite los siguientes recursos para obtener más información:

- [Sitio web de New Relic](#)
- [Documentación oficial](#)
- [Centro de desarrollo de New Relic](#)

Datadog

Datadog es una plataforma de supervisión y análisis para aplicaciones a gran escala. Abarca la supervisión de la infraestructura, la supervisión del rendimiento de las aplicaciones, la gestión de registros y la supervisión de la experiencia del usuario. Datadog agrega datos a través de toda su pila con más de 400 integraciones para la solución de problemas, alertas y gráficos.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Datadog](#)
- [Documentación oficial](#)

OpenTelemetry

OpenTelemetry es una colección de herramientas, APIs y SDKs. Utilícela para instrumentar, generar, recopilar y exportar datos de telemetría (métricas, registros y trazas) que le ayudarán a analizar el rendimiento y el comportamiento de su software.

Visite los siguientes recursos para obtener más información:

- [Sitio web de OpenTelemetry](#)
- [Documentación oficial](#)

Gestión de registros

La gestión de registros es el proceso de gestionar los eventos de registro generados por todas las aplicaciones de software y la infraestructura en la que se ejecutan. Implica la recopilación, agregación, análisis sintáctico, almacenamiento, análisis, búsqueda, archivo y eliminación de registros, con el objetivo último de utilizar los datos para solucionar problemas y obtener información empresarial, al tiempo que se garantiza el cumplimiento y la seguridad de las aplicaciones y la infraestructura.

Visite los siguientes recursos para obtener más información:

- [Introducción a la gestión de registros](#)
- [Gestión de registros: Lo que los equipos DevOps necesitan saber](#)
- [Registro para Kubernetes: Qué registrar y cómo hacerlo](#)

Elastic Stack

Elastic Stack es un grupo de productos de código abierto compuesto por Elasticsearch, Kibana, Beats y Logstash, entre otros, que ayudan a almacenar, buscar, analizar y visualizar datos de diversas fuentes, en distintos formatos y en tiempo real.

- **Elastic Search** - Motor de búsqueda y análisis
- **Logstash/fluentlyd** - Canal de procesamiento de datos
- **Kibana** - Panel de control para visualizar datos

Visite los siguientes recursos para obtener más información:

- [Sitio web de Elastic Stack](#)
- [Documentación oficial](#)
- [Características de Elastic Stack](#)
- [Logstash vs Fluentlyd](#)

Loki

Loki es un sistema de agregación de registros horizontalmente escalable, altamente disponible y multiusuario inspirado en Prometheus. Está diseñado para ser muy rentable y fácil de manejar. No indexa el contenido de los registros, sino un conjunto de etiquetas para cada flujo de registro.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Loki](#)
- [Documentación oficial](#)

Graylog

Graylog es una solución líder de gestión centralizada de registros para capturar, almacenar y permitir el análisis en tiempo real de terabytes de datos de máquinas.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Graylog](#)

- [Documentación oficial](#)
- [Vídeos del producto](#)

Splunk

La plataforma Splunk elimina las barreras entre los datos y la acción, capacitando a los equipos de observabilidad, TI y seguridad para garantizar que sus organizaciones sean seguras, resistentes e innovadoras.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Splunk](#)
- [Documentación oficial](#)
- [Vídeos de Splunk](#)

Papertrail

Papertrail es una solución líder de gestión centralizada de registros para capturar, almacenar y permitir el análisis en tiempo real de terabytes de datos de máquinas.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Papertrail](#)
- [Guías Oficiales](#)
- [Blog Oficial](#)

Organización de contenedores

Los contenedores son una construcción en la que se utilizan **cgroups**, **namespaces** y **chroot** para encapsular y aislar completamente un proceso. Este proceso encapsulado, denominado imagen de contenedor, comparte el núcleo del host con otros contenedores, lo que permite que los contenedores sean significativamente más pequeños y rápidos que las máquinas virtuales.

Estas imágenes están diseñadas para la portabilidad, lo que permite realizar pruebas locales completas de una imagen estática e implantarlas fácilmente en una plataforma de gestión de contenedores.

Visite los siguientes recursos para obtener más información:

- [¿Qué son los contenedores?](#)
- [¿Qué es un contenedor?](#)
- [¿Qué son los contenedores?](#)
- [Artículos sobre contenedores - The New Stack](#)

Kubernetes

Vea el roadmap.sh de Kubernetes para más información.

Docker Swarm

Un Docker Swarm es un grupo de máquinas físicas o virtuales que ejecutan la aplicación Docker y que han sido configuradas para unirse en un clúster. Una vez que un grupo de máquinas se han agrupado en un clúster, puedes seguir ejecutando los comandos Docker a los que estás acostumbrado, pero ahora serán llevados a cabo por las máquinas de tu clúster. Las actividades del clúster están controladas por un gestor de enjambre, y las máquinas que se han unido al clúster se denominan nodos.

Visite los siguientes recursos para obtener más información:

- [Sitio web oficial](#)
- [Documentación de Docker Swarm](#)
- [Tutorial de Docker Swarm para principiantes](#)
- [Tutorial: Gestionar Docker Swarm con Portainer](#)
- [Tutorial: Crear un enjambre Docker con almacenamiento persistente usando GlusterFS](#)

ECS / Fargate

ECS es un servicio de orquestación de contenedores que permite ejecutar contenedores Docker en un clúster de instancias EC2. ECS es una buena opción si desea ejecutar contenedores Docker en instancias EC2 y tener un control total sobre la infraestructura subyacente.

Fargate es un servicio de orquestación de contenedores sin servidor que permite ejecutar contenedores Docker sin tener que gestionar servidores, clústeres ni ninguna otra infraestructura. Fargate es una buena opción si desea ejecutar contenedores Docker sin tener que gestionar servidores o clústeres.

GKE / EKS / AKS

GKE - Motor Google Kubernetes

GKE es un servicio gestionado de Kubernetes que permite implementar, gestionar y escalar aplicaciones en contenedores en Google Cloud.

EKS - Servicio Amazon Elastic Kubernetes

Amazon Elastic Kubernetes Service (Amazon EKS) es un servicio Kubernetes totalmente gestionado de AWS.

AKS - Servicio Azure Kubernetes

Azure Kubernetes Service (AKS) administra su entorno Kubernetes alojado, lo que hace que sea rápido y fácil de implementar y administrar aplicaciones en contenedores sin experiencia en orquestación de contenedores.

Gestión de artefactos

Artifactory

- [Sitio web de Artifactory](#)

Nexus

- [Conceptos básicos de gestión de repositorios](#)
- [Instalación y configuración de Nexus](#)
- [Conceptos básicos de seguridad de repositorios de Nexus](#)
- [Mejores prácticas de Nexus](#)

Cloud Smith

GitOps

GitOps es una metodología para gestionar aplicaciones e infraestructuras nativas de la nube utilizando Git como fuente de verdad. Se basa en la idea de que todos los cambios en el sistema, ya estén relacionados con el código, la configuración o la infraestructura, deben realizarse utilizando Git y que Git debe utilizarse para automatizar el despliegue y la gestión de dichos cambios.

En un flujo de trabajo GitOps, los cambios en el sistema se realizan confirmando los cambios de código o configuración en un repositorio Git. A continuación, estos cambios se despliegan automáticamente en el entorno de producción mediante una canalización de entrega continua. El canal se activa con los cambios en el repositorio Git y es responsable de construir, probar y desplegar los cambios en el entorno de producción.

GitOps está diseñado para ser una forma más eficiente y ágil de gestionar entornos nativos en la nube, ya que permite a los desarrolladores realizar cambios en el sistema utilizando herramientas y procesos conocidos y proporciona un historial claro y auditable de todos los cambios realizados en el sistema. A menudo se utiliza junto con herramientas como Kubernetes y Helm para automatizar el despliegue y la gestión de aplicaciones nativas de la nube.

Visita los siguientes recursos para obtener más información:

- [Guía de GitOps](#)

ArgoCD

Argo CD es una herramienta de entrega continua para Kubernetes que se basa en la metodología GitOps. Se utiliza para automatizar el despliegue y la gestión de aplicaciones nativas de la nube mediante la sincronización continua del estado deseado de la aplicación con el estado real de la aplicación en el entorno de producción.

En un flujo de trabajo de Argo CD, los cambios en la aplicación se realizan mediante la confirmación de cambios en el código o la configuración en un repositorio Git. Argo CD supervisa el repositorio y despliega automáticamente los cambios en el entorno de producción mediante una canalización de entrega continua. El canal se activa con los cambios en el repositorio Git y es responsable de construir, probar y desplegar los cambios en el entorno de producción.

Argo CD está diseñado para ser una forma sencilla y eficiente de gestionar aplicaciones nativas de la nube, ya que permite a los desarrolladores realizar cambios en el sistema utilizando herramientas y procesos familiares y proporciona un historial claro y auditable de todos los cambios realizados en el sistema. A menudo se utiliza junto con herramientas como Helm para automatizar el despliegue y la gestión de aplicaciones nativas de la nube.

Visite los siguientes recursos para obtener más información:

- [Argo CD - Proyecto Argo](#)
- [Tutorial de ArgoCD para principiantes](#)

Flux CD

Flux CD es una herramienta de entrega continua para Kubernetes que se basa en la metodología GitOps. Se utiliza para automatizar el despliegue y la gestión de aplicaciones nativas de la nube

mediante la sincronización continua del estado deseado de la aplicación con el estado real de la aplicación en el entorno de producción.

En un flujo de trabajo de Flux CD, los cambios en la aplicación se realizan mediante la confirmación de cambios en el código o la configuración en un repositorio Git. Flux CD supervisa el repositorio y despliega automáticamente los cambios en el entorno de producción mediante una canalización de entrega continua. El canal se activa con los cambios en el repositorio Git y se encarga de compilar, probar y desplegar los cambios en el entorno de producción.

Flux CD está diseñado para ser una forma sencilla y eficaz de gestionar aplicaciones nativas de la nube, ya que permite a los desarrolladores realizar cambios en el sistema utilizando herramientas y procesos conocidos y proporciona un historial claro y auditable de todos los cambios realizados en el sistema. A menudo se utiliza junto con herramientas como Helm para automatizar el despliegue y la gestión de aplicaciones nativas de la nube.

Visite los siguientes recursos para obtener más información:

- [Documentos de Flux CD](#)

Malla de servicios

Una malla de servicios, como el proyecto de código abierto Istio, es una forma de controlar cómo las distintas partes de una aplicación comparten datos entre sí. A diferencia de otros sistemas para gestionar esta comunicación, una malla de servicios es una capa de infraestructura dedicada integrada directamente en una aplicación. Esta capa de infraestructura visible puede documentar lo bien (o no) que interactúan las distintas partes de una aplicación, de modo que resulta más fácil optimizar la comunicación y evitar el tiempo de inactividad a medida que crece una aplicación.

Visite los siguientes recursos para obtener más información:

- [¿Qué es una malla de servicios?](#)
- [Últimas noticias sobre malla de servicios \(TNS\)](#)

Istio

Istio es una plataforma de malla de servicios de código abierto que proporciona una forma de controlar cómo los microservicios comparten datos entre sí. Incluye API que permiten a Istio integrarse en cualquier plataforma de registro, telemetría o sistema de políticas. Istio está diseñado para funcionar en diversos entornos: en las instalaciones, alojado en la nube, en contenedores Kubernetes, en servicios que se ejecutan en máquinas virtuales, etc.

Visite los siguientes recursos para obtener más información:

- [¿Qué es Istio?](#)

Consul

Consul es una solución de malla de servicios que proporciona un plano de control completo con funciones de descubrimiento, configuración y segmentación de servicios. Cada una de estas funciones se puede utilizar por separado según sea necesario, o se pueden utilizar juntas para crear una malla de servicios completa. Consul requiere un plano de datos y admite un modelo de integración nativo y proxy. Consul se suministra con un proxy sencillo integrado para que todo funcione desde el primer momento, pero también admite integraciones de proxy de terceros, como Envoy.

Visite los siguientes recursos para obtener más información:

- [¿Qué es Consul?](#)

Linkerd

Linkerd es una malla de servicios de código abierto diseñada para ser desplegada en una variedad de programadores de contenedores y marcos como Kubernetes. Se convirtió en la "malla de servicios" original cuando su creador Buoyant acuñó el término por primera vez en 2016. Al igual que Finagle de Twitter, en el que se basó, Linkerd fue escrito por primera vez en Scala y diseñado para ser desplegado en una base por host. Linkerd es uno de los primeros productos en asociarse con el término malla de servicios y es compatible con plataformas como Docker y Kubernetes.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Linkerd](#)
- [Documentación de Linkerd](#)
- [¿Qué es Linkerd?](#)

Envoy

Creado originalmente en Lyft, Envoy es un plano de datos de alto rendimiento diseñado para arquitecturas de malla de servicios. Lyft lo creó en código abierto y lo donó al CNCF, donde ahora es uno de los proyectos de código abierto graduados del CNCF. Envoy es un proceso autónomo diseñado para ejecutarse junto a cualquier servidor de aplicaciones. Todos los Envoys forman una malla de comunicación transparente en la que cada aplicación envía y recibe mensajes hacia y desde localhost y desconoce la topología de la red.

Visite los siguientes recursos para obtener más información:

- [Sitio web de Envoy](#)
- [Documentación de Envoy](#)
- [¿Qué es Envoy?](#)

Patrones de diseño en la nube

- [Patrones de diseño en la nube – Microsoft Azure](#)

Disponibilidad

La disponibilidad es el porcentaje de tiempo durante el cual un sistema es funcional y funciona según lo previsto. La disponibilidad puede verse afectada por errores de hardware o software, problemas de infraestructura, ataques maliciosos y carga del sistema. Muchos proveedores de nube suelen ofrecer a sus usuarios un acuerdo de nivel de servicio (SLA) que especifica los porcentajes exactos de tiempo de actividad/tiempo de inactividad prometidos. En este sentido, la disponibilidad está relacionada con la fiabilidad. Por ejemplo, una empresa puede prometer un 99,99% de tiempo de actividad para sus servicios.

Para lograr altos niveles de tiempo de actividad, es importante eliminar los puntos únicos de fallo, de modo que el fallo de un solo dispositivo no interrumpa todo el servicio. La alta disponibilidad en la nube se consigue a menudo creando clusters. Los clusters son grupos de dispositivos (como servidores) que tienen todos acceso al mismo almacenamiento compartido y funcionan como un único servidor para ofrecer una disponibilidad ininterrumpida. De este modo, si un servidor se cae, los demás pueden asumir la carga hasta que vuelva a estar en línea. Los clusters pueden abarcar desde dos servidores hasta incluso varios edificios de servidores.

Visite los siguientes recursos para obtener más información:

- [Cómo funciona la alta disponibilidad en la nube](#)
- [Técnicas para conseguir alta disponibilidad](#)

Gestión de datos

La gestión de datos es el elemento clave de las aplicaciones en nube e influye en la mayoría de los atributos de calidad. Los datos suelen alojarse en diferentes ubicaciones y en varios servidores por razones de rendimiento, escalabilidad o disponibilidad, lo que puede plantear una serie de retos. Por ejemplo, es necesario mantener la coherencia de los datos y sincronizarlos en diferentes ubicaciones.

Además, los datos deben protegerse en reposo, en tránsito y mediante mecanismos de acceso autorizados para mantener las garantías de seguridad de confidencialidad, integridad y disponibilidad. Consulte el control de protección de datos de Azure Security Benchmark para obtener más información.

Visite los siguientes recursos para obtener más información:

- [Patrones de gestión de datos](#)

Diseño e implementación

Un buen diseño engloba factores como la consistencia y coherencia en el diseño y despliegue de componentes, la mantenibilidad para simplificar la administración y el desarrollo, y la reutilización para que los componentes y subsistemas puedan usarse en otras aplicaciones y en otros escenarios. Las decisiones tomadas durante la fase de diseño e implantación tienen un enorme impacto en la calidad y el coste total de propiedad de las aplicaciones y servicios alojados en la nube.

Visite los siguientes recursos para obtener más información:

- [Patrones de diseño e implementación](#)

Gestión y supervisión

La gestión y supervisión de DevOps implica supervisar todo el proceso de desarrollo, desde la planificación, el desarrollo, la integración y las pruebas, hasta la implantación y las operaciones. Implica una visión completa y en tiempo real del estado de las aplicaciones, los servicios y la infraestructura en el entorno de producción. Funciones como el streaming en tiempo real, la repetición histórica y las visualizaciones son componentes críticos de la monitorización de aplicaciones y servicios.

Visite los siguientes recursos para obtener más información:

- [Guía de introducción a la gestión y supervisión](#)

Este roadmap.sh ha sido
traducido por rortegag.com

Los enlaces del documento
son de roadmap.sh y no están
traducidos