En el ecosistema Linux moderno, donde la automatización y la reproducibilidad son claves, la creación de imágenes de sistema ya no se hace a base de scripts improvisados o pasos manuales difíciles de repetir. Para cubrir esa necesidad existe **distrobuilder**, una herramienta desarrollada dentro del proyecto Linux Containers cuyo objetivo es construir imágenes de distribuciones Linux de forma declarativa, consistente y totalmente reproducible.

Distrobuilder no es un gestor de contenedores ni un runtime. Su única función es **generar imágenes de sistema** listas para ser usadas posteriormente en entornos como LXC o Incus, o incluso como root filesystem genérico. Esto lo convierte en una pieza fundamental en la cadena de construcción de contenedores y máquinas virtuales ligeras.

La idea central de distrobuilder es muy simple: describimos cómo debe ser una imagen Linux en un archivo YAML, y la herramienta se encarga de construirla exactamente de esa forma todas las veces que queramos. Esa descripción incluye la distribución base, la versión, la arquitectura, los paquetes que deben instalarse o eliminarse, archivos que deben añadirse y comandos que deben ejecutarse durante la construcción.

El **archivo de definición YAML** es el corazón de todo el proceso. En él se especifica, por ejemplo, si queremos Debian, Ubuntu u otra distribución, qué repositorios usar, qué paquetes son imprescindibles y cuáles deben desaparecer para mantener la imagen lo más limpia posible. También permite definir acciones más avanzadas como crear usuarios, modificar configuraciones del sistema o ajustar servicios para que la imagen final esté lista para producción.

Durante la construcción, distrobuilder **obtiene un sistema base** utilizando las herramientas propias de cada distribución. En el caso de Debian o Ubuntu suele usar debootstrap, aunque también puede trabajar a partir de rootfs ya existentes. A partir de ese sistema mínimo aplica, en orden, todas las instrucciones definidas en el YAML: instalación de paquetes, copiado de archivos, ejecución de scripts y ajustes finales.

Una de las grandes ventajas de distrobuilder es que **todo el proceso es determinista**: si usamos la misma definición y las mismas fuentes, la imagen generada será siempre igual. Esto es especialmente importante en entornos profesionales donde se necesita auditar qué lleva exactamente una imagen o reconstruirla meses después sin sorpresas.

En cuanto a los formatos de salida, distrobuilder es flexible. Puede generar simplemente un root filesystem plano, útil si queremos empaquetarlo o adaptarlo a otro sistema. También puede producir imágenes específicas para LXC, separando los metadatos del sistema de archivos, o imágenes listas para ser importadas directamente en Incus. En todos los casos, el resultado es una imagen limpia, coherente y preparada para su uso inmediato.

Distrobuilder es, además, la herramienta que utiliza oficialmente el proyecto Linux Containers para generar las imágenes públicas que se distribuyen para LXC e Incus. Esto significa que no estamos ante una utilidad experimental, sino ante el estándar de facto para la creación de imágenes en este ecosistema.

Desde el punto de vista operativo, distrobuilder **está escrito en Go** y se distribuye como binario o código fuente. Dependiendo de la distribución que queramos construir, puede requerir herramientas adicionales del sistema anfitrión, como debootstrap, squashfs-tools o utilidades de compresión. Una vez instalado, su uso es directo: se le indica la definición YAML y el tipo de imagen que queremos generar.

En escenarios reales, distrobuilder resulta especialmente útil cuando necesitamos mantener un catálogo propio de imágenes base, ya sea para un laboratorio, un entorno de desarrollo, un clúster de contenedores o una infraestructura de virtualización ligera. En lugar de descargar imágenes genéricas y adaptarlas después, podemos definir exactamente cómo debe ser nuestra imagen y reconstruirla cuando sea necesario.