

kestra. Product Solutions Plugins Learn Company Pricing 26,683 Contact Sales Get Started

One Platform to Run All Your Workflows

Kestra is the open-source orchestration platform for data, AI, and infrastructure workflows. Event-driven, language-agnostic, and built for enterprise scale.

Declarative Any Language API-First 1200+ Plugins No Lock-in

Self-Hosted Cloud

Get Started with Open Source Book a Demo

Xiaomi Bloomberg TOYOTA FILA Apple J.P.Morgan T Systems

Most Orchestrators Are Built for One Team.
Kestra is Built for the Whole Organization.

Kestra

Kestra es una plataforma de orquestación de flujos de trabajo declarativa y event-driven diseñada para ingenieros de datos, perfiles DevOps y arquitectos de software. Permite automatizar procesos complejos de infraestructura, microservicios y pipelines de datos mediante YAML. Combina un editor visual intuitivo con la potencia del Orchestration-as-Code, facilitando la reacción a eventos en tiempo real y la integración de más de 1.200 conectores nativos para entornos escalables y modernos.

[Visitar Sitio Oficial](#) [Preguntar a ChatGPT](#) [Preguntar a Claude](#) [Preguntar a Grok](#)

Contenido del Dossier

- [Información de la Herramienta](#)
- [Consejos de Implantación](#)
- [Preguntas Frecuentes](#)
- [Contratos y Condiciones](#)

INFORMACIÓN DE LA HERRAMIENTA

Qué y para quién es

Kestra es una plataforma de orquestación de flujos de trabajo (workflows) "event-driven" y declarativa que permite automatizar procesos complejos de datos, infraestructura y microservicios. Su enfoque principal es el "Orchestration-as-Code" utilizando YAML, lo que permite definir pipelines tanto desde una interfaz visual intuitiva como mediante código.

Está dirigida a ingenieros de datos, perfiles DevOps, ingenieros de software y arquitectos de soluciones que buscan una alternativa moderna y escalable a herramientas tradicionales (como Airflow), priorizando la facilidad de despliegue y la capacidad de reaccionar a eventos en tiempo real.

Principal ventaja profesional

La capacidad de unificar el desarrollo visual con la gestión declarativa en YAML. Esto permite que perfiles técnicos gestionen la lógica mediante CI/CD y código, mientras que otros departamentos o responsables pueden visualizar, monitorizar e incluso editar flujos desde la UI sin una curva de aprendizaje elevada, eliminando los silos técnicos.

Para quién no es

No es la herramienta ideal para usuarios que buscan una solución de automatización simple "no-code" de consumo (tipo Zapier o Make) para tareas administrativas básicas. Tampoco es adecuada para equipos que rechacen el uso de YAML o que prefieran orquestadores puramente imperativos donde toda la lógica se escriba exclusivamente en un lenguaje de programación específico (como Python en versiones antiguas de Airflow).

Funcionalidades clave

- **Orquestación Declarativa:** Definición de flujos completos mediante archivos YAML simples y legibles.
- **Editor Embebido:** Interfaz web con autocompletado, validación de sintaxis en tiempo real y visualización gráfica (DAG) inmediata.
- **Event-Driven:** Triggers avanzados para iniciar flujos basados en eventos (llegada de archivos, mensajes en colas, webhooks) o programación temporal (cron).
- **Ecosistema de Plugins:** Más de 1.200 conectores nativos para bases de datos (PostgreSQL, BigQuery), nubes (AWS, GCP, Azure), scripts (Python, Node.js, R) y herramientas (Docker, Terraform).
- **Control de Versiones:** Integración nativa con Git para sincronizar flujos directamente con repositorios.
- **Topología en Vivo:** Visualización del estado de ejecución y dependencias en tiempo real para facilitar el depurado (debugging).

Precios

- **Open Source:** Gratuito bajo licencia Apache 2.0. Incluye todas las funciones principales de orquestación, editor de código y acceso a plugins estándar.
- **Edición Team / Pro:** Suscripción anual (precio bajo consulta). Añade gestión de secretos interna, SSO, plugins empresariales y linaje de datos.
- **Enterprise:** Suscripción anual (precio bajo consulta). Incluye Alta Disponibilidad (HA), RBAC (Control de acceso basado en roles) avanzado, auditoría de logs, multi-tenancy y soporte con SLA.
- **Kestra Cloud:** Modelo gestionado (SaaS) con pago por uso, eliminando la carga de mantenimiento de infraestructura.

Perfil del usuario

- **Empresas Data-Driven:** Que necesitan mover y transformar grandes volúmenes de datos entre diferentes sistemas.
- **Departamentos de IT/DevOps:** Para automatizar el ciclo de vida de la infraestructura y tareas de mantenimiento repetitivas.
- **Equipos de IA/ML:** Para orquestar el entrenamiento de modelos y pipelines de inferencia.
- **Perfiles profesionales:** Data Engineers, Platform Engineers, Backend Developers y Analistas de Datos con conocimientos técnicos.

Nivel técnico requerido

- **Para el uso:** Medio. Se requiere familiaridad con la sintaxis YAML y conceptos de flujos de trabajo (tareas, triggers, reintentos).

- **Para la instalación:** Medio-Alto. Requiere conocimientos en Docker o Kubernetes para el despliegue de la infraestructura (servidor, base de datos y almacenamiento).
- **Competencias necesarias:** Conocimientos básicos de infraestructura cloud, gestión de APIs y, opcionalmente, lenguajes de scripting (Python/Bash) para tareas personalizadas.

Ejemplos de uso profesional

- **ETL/ELT Automatizado:** Extracción de datos de una API, transformación en un contenedor Docker y carga en un Data Warehouse.
- **Informes de Negocio:** Monitorización de una base de datos de ventas para generar y enviar un informe por Slack o Email cada vez que se detecte una anomalía.
- **Mantenimiento de Infraestructura:** Apagado automático de instancias cloud no utilizadas durante el fin de semana para optimizar costes.
- **IA Generativa:** Orquestación de llamadas a modelos LLM (OpenAI, Anthropic) para procesar documentos cargados en un bucket de S3.

Uso y distribución

- **Versión web:** Interfaz de usuario completa accesible vía navegador tras la instalación.
- **Versión escritorio:** No dispone de app nativa; se gestiona vía Web UI.
- **Despliegue:** Disponible mediante imagen Docker, Helm charts para Kubernetes o despliegue directo en AWS/GCP/Azure.
- **CLI:** Dispone de una interfaz de línea de comandos para gestionar flujos y plugins desde la terminal.

Open source

Kestra es de código abierto (Apache License 2.0). El repositorio principal está disponible en GitHub, permitiendo auditoría del código y contribuciones de la comunidad.

Integraciones

- **Facilidad de integración:** Muy alta. Ofrece un ecosistema de plugins que cubren casi cualquier necesidad sin escribir código adicional.
- **API propia:** Dispone de una API REST completa para crear, ejecutar y monitorizar flujos de forma programática.
- **Integraciones nativas:** Más de 1.200, incluyendo Snowflake, Databricks, dbt, Airbyte, Fivetran, Kafka, Elasticsearch y servicios de mensajería como Telegram o Discord.

Notas finales

Información legal, licencias y contratos

La versión Open Source es muy permisiva (Apache 2.0), ideal para uso comercial sin costes iniciales. Las versiones Enterprise requieren un contrato comercial específico que incluye garantías de soporte técnico y cumplimiento normativo (SOC2 en su versión Cloud).

Para más información:

- <https://kestra.io>
- <https://kestra.io/pricing>
- <https://github.com/kestra-io/kestra>
- <https://www.linkedin.com/company/kestra>
- <https://kestra.io/slack>

CONSEJOS DE IMPLANTACIÓN

Aplicación profesional

Kestra se orienta a empresas con arquitecturas de datos modernas (Data-First), departamentos de IT y equipos DevOps que gestionan infraestructuras en la nube. Es especialmente útil en organizaciones que buscan centralizar la automatización de procesos sin depender exclusivamente de perfiles que dominen lenguajes de programación complejos como Python, permitiendo que analistas técnicos colaboren en el diseño de flujos. El presupuesto es altamente escalable: desde el coste cero en infraestructura propia (Open Source) hasta modelos de suscripción para necesidades de cumplimiento y alta disponibilidad en entornos corporativos. Puntos clave: reducción de la deuda técnica mediante configuración declarativa y alta eficiencia en la orquestación basada en eventos en tiempo real.

Madurez digital requerida

- Usuarios y equipo: Perfiles con conocimientos técnicos medios que entiendan la lógica de procesos y la sintaxis YAML. Equipos acostumbrados a trabajar con APIs, contenedores (Docker) y sistemas de control de versiones.
- Empresa y departamentos: Organizaciones que ya operan con infraestructuras cloud o híbridas y que requieren una integración fluida entre equipos de ingeniería de datos, operaciones y desarrollo de software para eliminar silos operativos.

Plan orientativo de implantación

Pasos necesarios y estimaciones

- Tiempos estimados de despliegue: De 1 a 3 semanas para una puesta en marcha funcional en entorno de producción, dependiendo de la complejidad de la infraestructura.
- Evaluación inicial: Definición de fuentes de datos, identificación de eventos trigger y selección de plugins necesarios. Estimación de requisitos de computación (CPU/RAM) según el volumen de ejecuciones previstas.
- Implantación inicial: Configuración del entorno mediante Docker Compose o Helm Charts en Kubernetes. Creación de una Prueba de Concepto (PoC) migrando un flujo de trabajo crítico existente o automatizando una tarea manual recurrente.
- Configuración y personalización: Establecimiento de conexiones con sistemas externos (DBs, Cloud Providers), configuración de la persistencia de datos y aseguramiento de la conectividad de red.
- Formación y adaptación: Talleres prácticos sobre la lógica de tareas y triggers en YAML, gestión del editor web y buenas prácticas en la organización de flujos.
- Seguimiento y feedback: Monitorización de logs en tiempo real, ajuste de políticas de reintentos y optimización de recursos basados en las métricas de ejecución.

Necesidades de formación del equipo

Es fundamental capacitar al equipo en la sintaxis declarativa de Kestra y en el uso de su interfaz visual para depuración. Se recomienda formación específica en la gestión de secretos (Secrets Management) y en la integración con repositorios Git para mantener un ciclo de vida de desarrollo profesional (CI/CD).

Perfiles necesarios

- Perfiles técnicos: Ingeniero de Datos o DevOps para la instalación, configuración del clúster y mantenimiento de la plataforma.
- Personal externo recomendado: Consultores en arquitectura de datos para la optimización de pipelines complejos si la empresa carece de experiencia en orquestación moderna.
- Otros: Un responsable de seguridad (CISO) para validar la gestión de accesos y el cumplimiento normativo en las versiones Enterprise.

Retorno de la inversión

- Tiempos: Se estima una reducción de hasta el 40% en el tiempo de desarrollo de nuevos flujos de trabajo gracias a su enfoque declarativo y la biblioteca de plugins preconstruidos.
- Cómo medirlo y KPIs: Tiempo medio de entrega de nuevos flujos (Time-to-Market), reducción del tiempo de inactividad por errores en pipelines (MTTR), ahorro en costes de computación por orquestación eficiente y número de incidencias manuales resueltas automáticamente.

Otros

Kestra destaca frente a competidores por su arquitectura desacoplada y su capacidad de escalar horizontalmente. A diferencia de otras herramientas que requieren un motor de ejecución pesado para cada flujo,

Kestra optimiza el consumo de recursos al ser capaz de gestionar miles de flujos simultáneos de manera ligera, lo que impacta directamente en la reducción de costes de infraestructura cloud.

PREGUNTAS FRECUENTES

¿Qué es Kestra y en qué se diferencia de otras herramientas de orquestación?

Kestra es una plataforma de orquestación de flujos de trabajo de tipo 'event-driven' y declarativa. A diferencia de herramientas tradicionales como Apache Airflow, que se basan predominantemente en código imperativo (Python), Kestra utiliza archivos YAML para definir pipelines. Esto permite separar la lógica de negocio de la infraestructura, facilitando que tanto perfiles técnicos como menos técnicos puedan colaborar en un mismo entorno visual y de código.

¿Es Kestra una tecnología open source?

Sí, el núcleo de Kestra se distribuye bajo la licencia Apache 2.0, lo que permite su uso, modificación y distribución gratuita incluso para fines comerciales. El código fuente está disponible públicamente en su repositorio de GitHub para auditoría y contribuciones de la comunidad.

¿Cumple con la normativa de seguridad y privacidad empresarial?

Kestra ofrece diferentes niveles de cumplimiento según la edición. Mientras que la versión Open Source depende de la infraestructura del usuario, la versión Enterprise incluye características avanzadas de seguridad como Control de Acceso Basado en Roles (RBAC), auditoría de logs para trazabilidad y gestión de secretos interna. Además, su oferta SaaS (Kestra Cloud) está diseñada para cumplir con estándares internacionales como SOC2.

¿Qué nivel de conocimientos técnicos se requiere para operar la plataforma?

Para la creación de flujos, el nivel requerido es medio; es necesario estar familiarizado con la sintaxis YAML y conceptos lógicos de workflows (tareas, disparadores y reintentos). Para la administración e instalación, se requiere un nivel medio-alto, ya que implica el despliegue de infraestructura mediante Docker, Kubernetes o servicios cloud (AWS, GCP, Azure).

¿Qué costes tiene asociados?

Kestra ofrece un modelo escalable. La edición Open Source es gratuita. Las ediciones 'Team', 'Pro' y 'Enterprise' operan bajo un modelo de suscripción anual con precios bajo consulta, añadiendo funcionalidades de gobernanza y alta disponibilidad. También existe 'Kestra Cloud', una modalidad gestionada con pago por uso que elimina la carga operativa de mantenimiento.

¿Cómo gestiona Kestra la integración con otras herramientas?

La plataforma utiliza un ecosistema de más de 1.200 plugins nativos que permiten conectar con bases de datos (PostgreSQL, BigQuery), plataformas cloud, sistemas de mensajería y herramientas de datos como dbt o Snowflake. Además, cuenta con una API REST completa para integraciones programáticas y una interfaz de línea de comandos (CLI).

¿Cuál es la principal ventaja de utilizar un enfoque 'Orchestration-as-Code' con YAML?

La ventaja profesional clave es la unificación del desarrollo. Permite aplicar buenas prácticas de ingeniería de software, como el control de versiones mediante Git y ciclos de CI/CD, sin sacrificar la monitorización visual. Esto elimina silos entre los equipos que prefieren el código y los que necesitan una supervisión gráfica del estado de los procesos en tiempo real.

¿Es adecuado para automatizaciones simples de usuario final tipo Zapier?

No. Kestra está diseñado para infraestructuras de datos complejas, microservicios y operaciones críticas de IT. No es una solución 'no-code' para tareas administrativas básicas o de consumo general; requiere una comprensión de la arquitectura de sistemas y flujos de datos profesionales.

CONTRATOS Y CONDICIONES

Principales recomendaciones

- **Privacidad desde el diseño:** Al ser una herramienta de orquestación que puede mover datos sensibles entre diferentes sistemas, es imperativo configurar el "Secret Manager" (disponible en versiones Team/Enterprise) para evitar que credenciales o datos personales queden expuestos en los archivos YAML de configuración.
- **Elección de despliegue:** Para empresas con altos requisitos de soberanía de datos, se recomienda la versión **Self-hosted** (Autohospedada) frente a la versión Cloud, ya que permite mantener el control total sobre la ubicación física de los flujos y los datos procesados dentro de la infraestructura de la empresa en España/UE.
- **Minimización en logs:** Configurar las políticas de retención de logs y evitar el volcado de datos PII (Información de Identificación Personal) en las salidas de consola de las tareas, ya que estos quedan almacenados en la base de datos de Kestra.
- **Segregación de funciones:** Utilizar el sistema de **Namespaces** para aislar proyectos y aplicar el principio de menor privilegio, asegurando que un flujo de un departamento no tenga acceso a los datos o secretos de otro.

Ley de Inteligencia Artificial (AI Act)

- **Clasificación de riesgo:** Kestra actúa principalmente como un **infraestructura de gestión/orquestación**. Si se utiliza para entrenar modelos de IA o gestionar pipelines de inferencia para "Sistemas de IA de alto riesgo" (según el Anexo III de la Ley), la empresa usuaria es responsable de garantizar la trazabilidad y la documentación técnica que exige la normativa.
- **Gobernanza de datos:** Las funcionalidades de **Lineage** (linaje de datos) en las versiones Pro/Enterprise son fundamentales para cumplir con los requisitos de transparencia y calidad de los datos de entrenamiento exigidos por el AI Act.

Privacidad y protección de datos

- **Responsabilidades:** Kestra Technologies (con sede en Francia, UE) actúa como **Encargado del Tratamiento** cuando se usa su versión Cloud. En versiones Self-hosted, la empresa española es la única responsable del tratamiento y del control de la seguridad.
- **Ubicación de los datos:** Al ser una empresa francesa, el cumplimiento del RGPD es nativo. Los datos de la plataforma Cloud se alojan en regiones de la UE (principalmente Francia/Bélgica), minimizando riesgos de transferencias internacionales.
- **Transferencia internacional:** Si se conectan plugins de terceros (AWS, Google Cloud, OpenAI), la empresa española debe firmar las Cláusulas Contractuales Tipo (SCC) con dichos proveedores, ya que Kestra solo actúa como el puente técnico.
- **Derechos ARCO:** La plataforma permite la eliminación técnica de ejecuciones y flujos, facilitando el cumplimiento del derecho de supresión. No obstante, la gestión del ejercicio de derechos sobre los datos personales dentro de los flujos de trabajo es responsabilidad exclusiva de la empresa usuaria.

Propiedad intelectual

- **Propiedad de datos:** La herramienta es "Data-Agnostic"; Kestra no reclama ningún derecho sobre los datos que transitan por sus tuberías (pipelines).
- **Propiedad del resultado:** Los archivos YAML, scripts y la lógica de negocio desarrollada por la empresa española son de su propiedad exclusiva.
- **Licenciamiento:** La versión Open Source usa **Apache License 2.0**, que permite el uso comercial, modificación y distribución sin pago de regalías, pero sin garantías ni soporte técnico legal vinculado.

Usos y prohibiciones

- **Usos prohibidos:** El contrato de Cloud prohíbe el uso de la plataforma para actividades ilegales, test de penetración sin previo aviso, o el envío de "datos altamente sensibles" (salud, biometría) sin aprobación previa por escrito en ciertas configuraciones.
- **Usos admitidos:** Automatización de procesos empresariales, ETL de datos, gestión de infraestructura (DevOps) y orquestación de modelos de IA.

Seguridad y certificaciones

- **Seguridad:** Implementa cifrado en reposo y en tránsito. Las versiones superiores incluyen RBAC (control de acceso basado en roles) fino y auditoría de logs (Audit Logs) necesaria para cumplimiento normativo

avanzado.

- **Certificaciones:** La infraestructura de **Kestra Cloud posee certificación SOC2**, lo que garantiza controles rigurosos sobre la seguridad, disponibilidad y privacidad de los procesos.

Otros

- **Legislación aplicable:** Los contratos de las versiones de pago se rigen por la **ley francesa**, lo cual es ventajoso para una empresa española al permanecer dentro del marco jurídico de la Unión Europea y la jurisdicción de los tribunales de París.

Fuentes consultadas:

- [Contrato de tratamiento de datos \(DPA\)](#)
- [Términos y condiciones de Kestra Cloud](#)
- [Política de privacidad](#)
- [Licencia Apache 2.0 en GitHub](#)
- [Detalles de precios y niveles de seguridad](#)

Para más información y herramientas:

Explora look4.tools para descubrir las mejores soluciones tecnológicas del mercado.

[Inicio](#) [Todas las herramientas](#) [Categorías](#)

Este documento ofrece recomendaciones generadas mediante análisis humano y sistemas de IA automatizados. La información tiene carácter meramente informativo y no constituye asesoramiento legal, profesional ni garantía de resultados. Las marcas, logotipos y nombres comerciales pertenecen a sus respectivos propietarios y se utilizan únicamente con fines identificativos.