



# Where Kestra runs

Orchestration stays central, and execution runs next to the systems it reaches, in whatever network zones your policy defines.

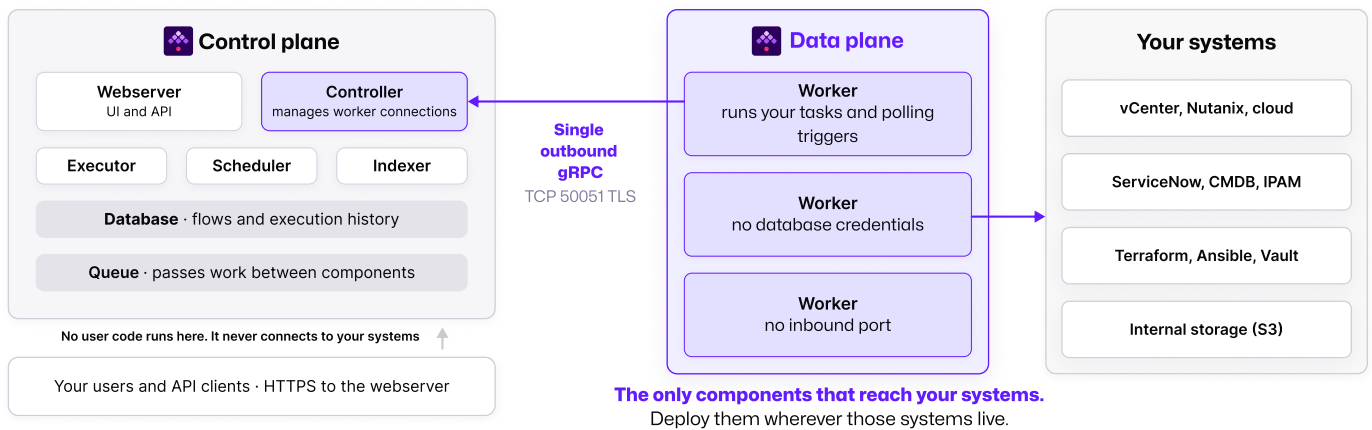
Kestra Enterprise Edition, self-hosted • Updated October 2026



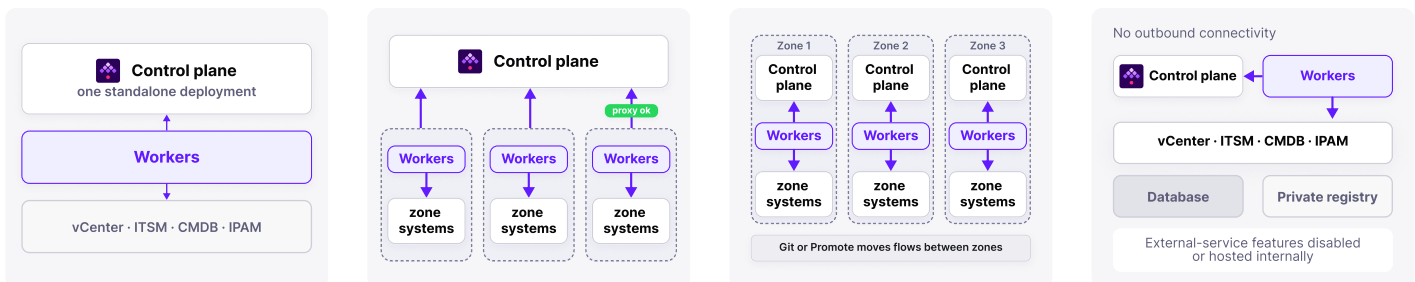
## Architecture

Kestra separates the components that coordinate work from the components that do it. The control plane holds your flows, your execution history and the database connection, and no user code runs there. Workers run every task and every polling trigger, which makes them the components that reach your vCenter, your ITSM, your CMDB, your IPAM and the targets themselves. A worker holds no database credentials and opens a single outbound connection to the control plane, so the network it sits in needs no inbound firewall rule.

## The components, and what each one touches



## Deployment shapes



### Single zone

Control plane and workers on the same network. Every component can run together as one standalone deployment, so there is nothing extra to deploy.

### Multi-zone: central control plane, remote workers

One control plane, with workers placed in each network zone next to the systems they act on. Each worker dials out, so a segmented zone that permits outbound traffic and refuses every inbound connection still gets orchestrated.

### Multi-zone: independent control plane per zone

For policies that prohibit any component from spanning zones. Each zone runs a complete instance of its own, and flows move between zones through Git or through Promote rather than over a live connection.

### Fully air-gapped

Control plane, workers, database and registry all inside the perimeter, with no outbound internet connectivity. Installation is offline, from a private registry.



## What your platform and network teams will ask

Security controls, compliance and data handling are covered in the [Kestra security overview](#).

|                                                                       |                                                                                                                                                                                               |
|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Which component reaches our systems?                                  | Workers, and only workers. The control plane coordinates work and never connects to the systems your flows act on.                                                                            |
| Which direction does the connection run?                              | From the worker to the control plane. Always outbound, always initiated by the worker.                                                                                                        |
| Do we need to open anything inbound in a worker zone?                 | No.                                                                                                                                                                                           |
| What network path is required between a worker and the control plane? | One TCP connection to the control plane's gRPC port, held open for the life of the worker. In a default-deny network that is the one rule to add.                                             |
| Is that traffic encrypted, and is mTLS supported?                     | Yes, with TLS between workers and the control plane, set up as part of the deployment. Mutual TLS is also supported where policy requires each worker to present a certificate.               |
| Can it traverse a proxy or load balancer?                             | Yes. The hop has to allow gRPC, which needs HTTP/2 and a connection that stays open.                                                                                                          |
| Can workers run on Windows?                                           | Yes, as a Windows remote worker in a worker group, which keeps PowerShell and batch work native.                                                                                              |
| What else does a remote worker need access to?                        | Internal storage, and the systems in its own zone. The control plane connection on its own is not enough.                                                                                     |
| Can it run highly available?                                          | Yes. Each component can run as multiple replicas and scales independently. High availability also needs a dedicated queue alongside the database, since PostgreSQL runs as a single instance. |
| Can Kestra run with no internet connectivity?                         | Yes, as a fully air-gapped deployment.                                                                                                                                                        |

### Fully air-gapped

A fully air-gapped deployment keeps every Kestra component and its dependencies inside the perimeter. The control plane, the workers, the database and the container registry all live there, and installation is offline from a private registry. Nothing in the orchestration path needs outbound connectivity.

Features that depend on external services need to be disabled or provided from inside the perimeter as part of an air-gapped deployment.

A worker can also run inside a restricted or isolated network zone and connect outbound to a control plane elsewhere.

## Operating it

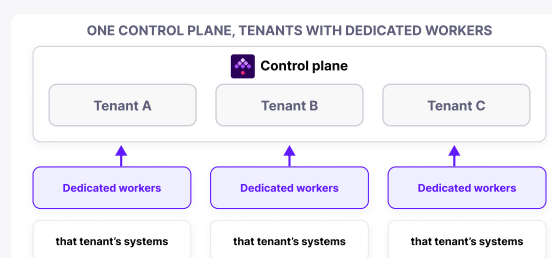
### Install paths

Three production paths.

- **Kubernetes**, with the official Helm chart. Each component, including the controller, scales independently.
- **Docker Compose** on a virtual machine.
- **Standalone JAR** as a service, where no container runtime is permitted. It needs a JVM and an external database.

### Tenants and dedicated workers

One control plane can be divided into tenants. A tenant can get dedicated worker groups, so its execution runs on its own infrastructure and does not compete with other tenants for capacity. Worker groups route a namespace or task to hosts by tag, and nothing is installed on the systems your flows act on.



### Observability

Every component exposes Prometheus metrics and a health endpoint by default, so it reports into the monitoring you already run. Traces, metrics and logs can also be exported over OpenTelemetry once it is enabled, and failures can alert through Kestra's own notification tasks.

