

**is it done?**

**Temporal** is an open-source, cloud-native **workflow orchestration platform** that lets developers define long-running, state-persistent workflows in code while the Temporal service guarantees reliable execution even in the face of service crashes, network partitions, or infrastructure failures.[[1]](#fn1)

Key points:

* **Durable execution**: Every state change is written to an append-only event log so a workflow can resume exactly where it left off after any interruption.[[2]](#fn2)
* **Code-first model**: Workflows and activities are written in familiar languages (Go, Java, TypeScript, Python, PHP, Rust), avoiding external DSLs or JSON-based definitions.[[3]](#fn3)[[2]](#fn2)
* **At-least-once guarantees** with **exact-once semantics** at the workflow level: Temporal automatically retries failed steps with customizable back-off, ensuring business logic runs to completion without duplicating side effects.[[1]](#fn1)
* **Horizontally scalable service**: The Temporal server is stateless; scaling simply adds more worker or service nodes, which makes it suitable for high-throughput cloud deployments.[[4]](#fn4)
* **Strong observability**: Built-in visibility into workflow history, execution status, retries, and errors through Web UI and APIs.[[5]](#fn5)
* **Use cases**: Micro-service sagas, batch job orchestration, payment processing, user provisioning, and disaster-recovery automations like BCP workflows where reliable, multi-step operations must survive failures.[[6]](#fn6)

Because workflows are expressed as regular code and Temporal handles persistence, timers, and retries, developers can focus on business logic while gaining the fault tolerance and consistency normally associated with distributed transaction systems.[[4]](#fn4)

⁂

1. <https://docs.temporal.io/temporal>

1. <https://docs.temporal.io/workflows>

1. <https://docs.temporal.io/workflow-definition>

1. <https://temporal.io/blog/workflow-engine-principles>

1. <https://temporal.io>

1. <https://pipekit.io/blog/temporal-vs-argo-workflows>