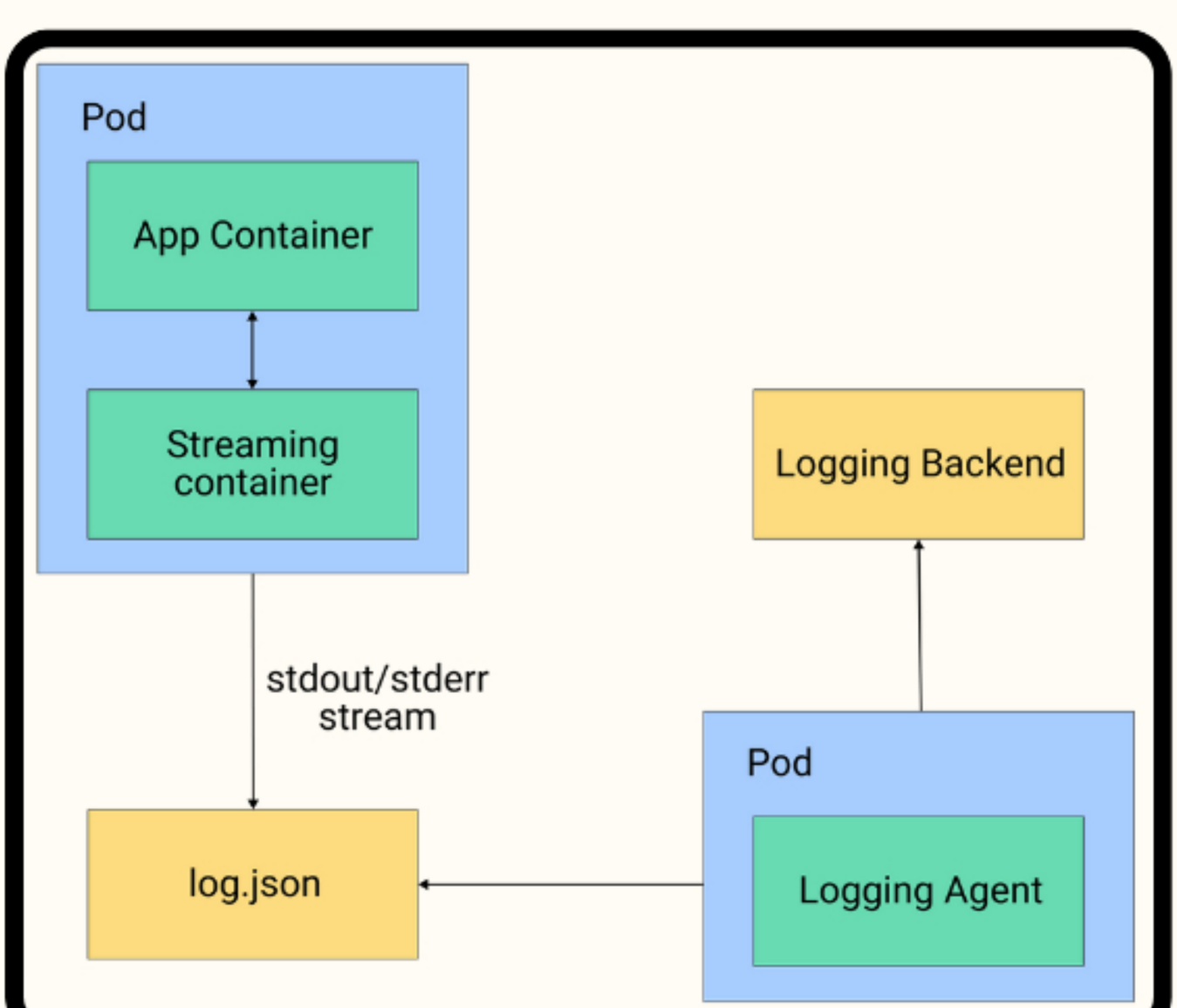
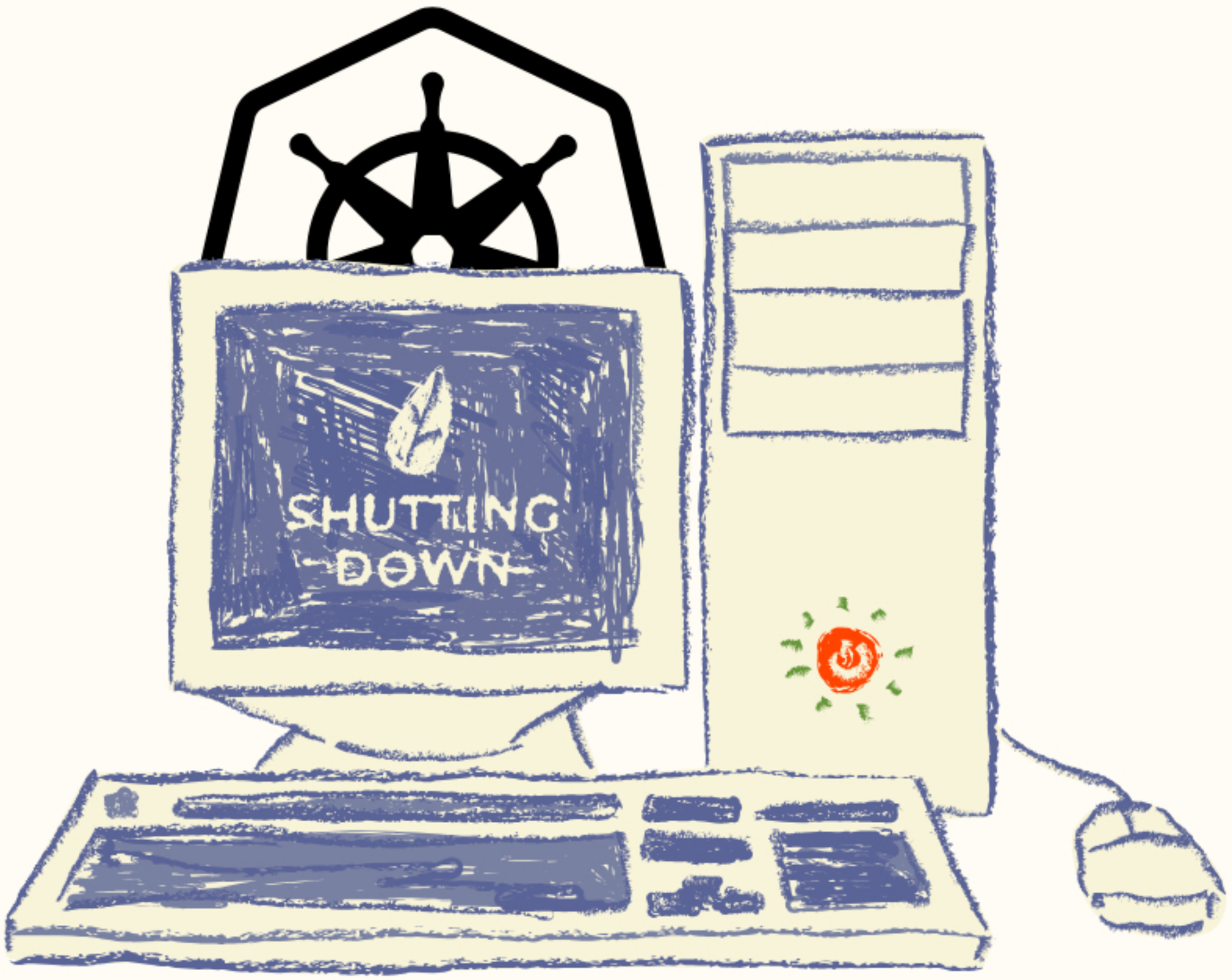
Kubernetes logging Architecture

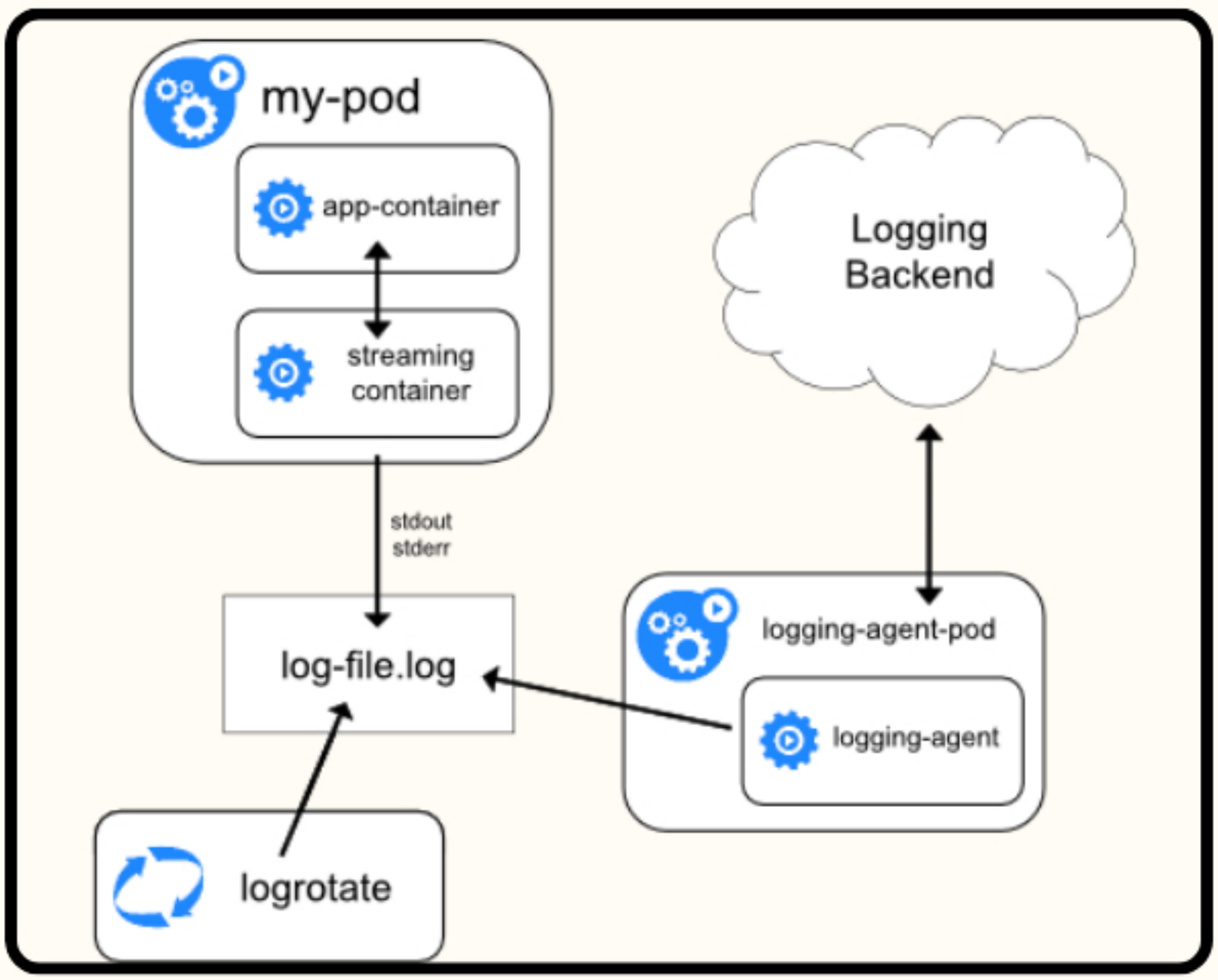
One factor that complicates K8s logging is that most components of K8s are constantly changing, and some are not persistent.

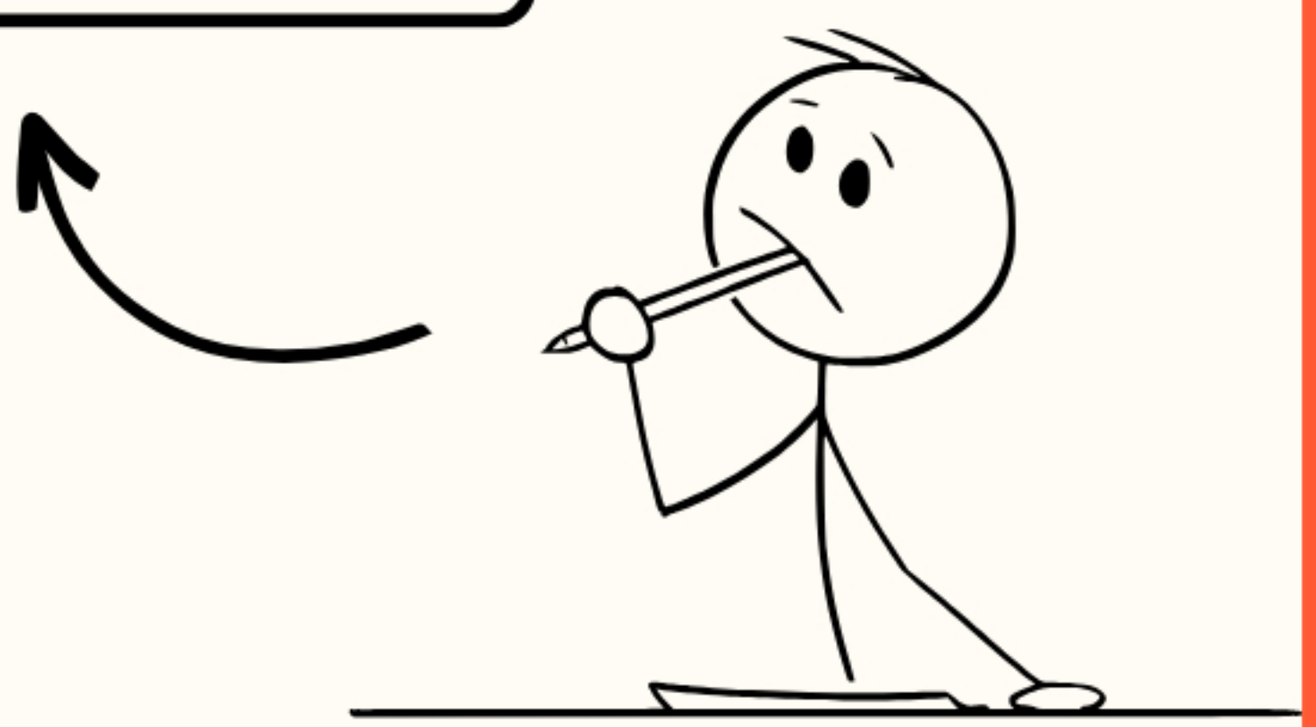


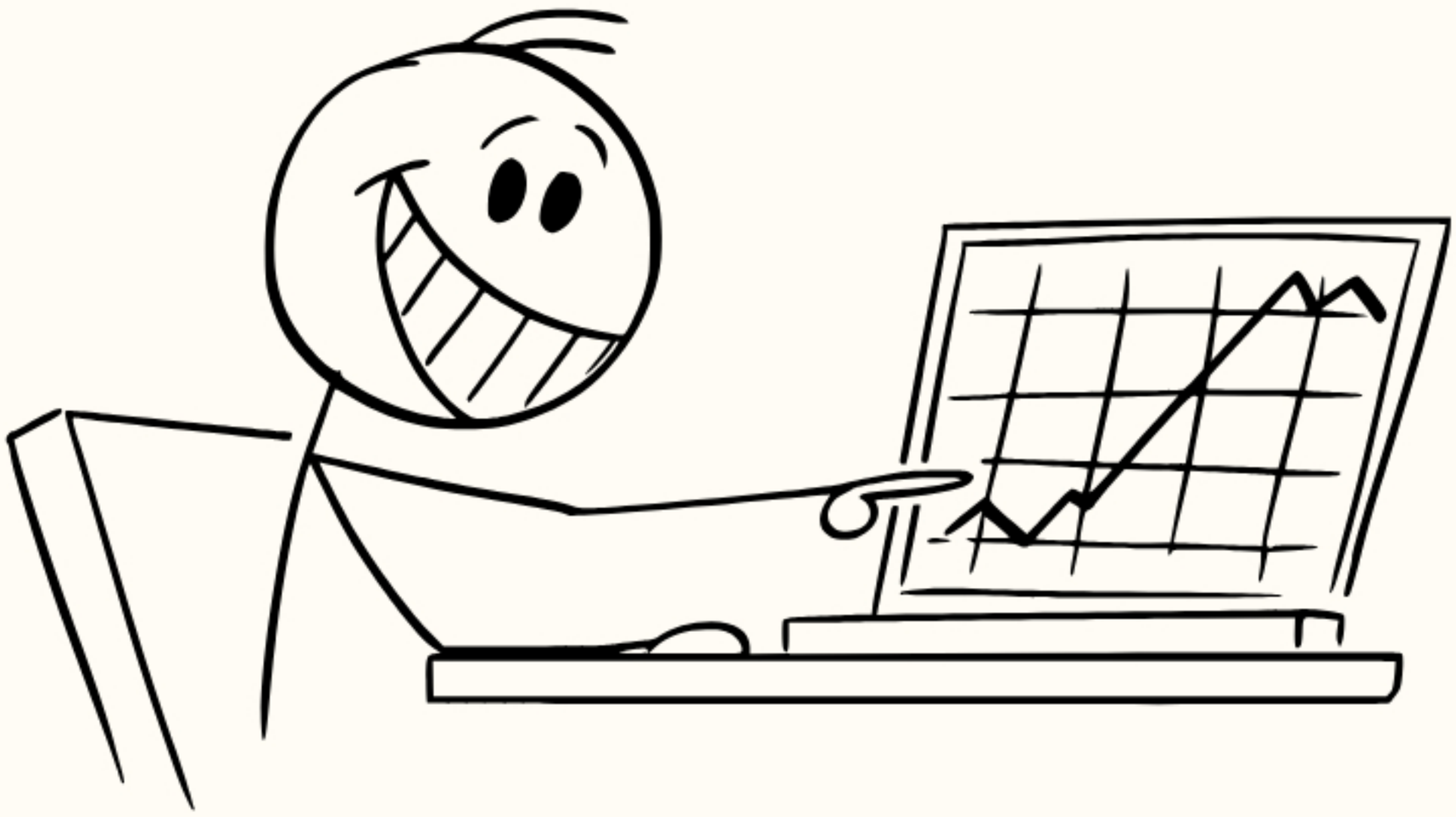
For example, when containers shut down, any data stored inside them, including logs, will go away with them.



To solve this challenge, K8s uses a logging architecture – known as cluster-level logging – that decouples log storage and lifecycles from nodes, Pods, and containers.

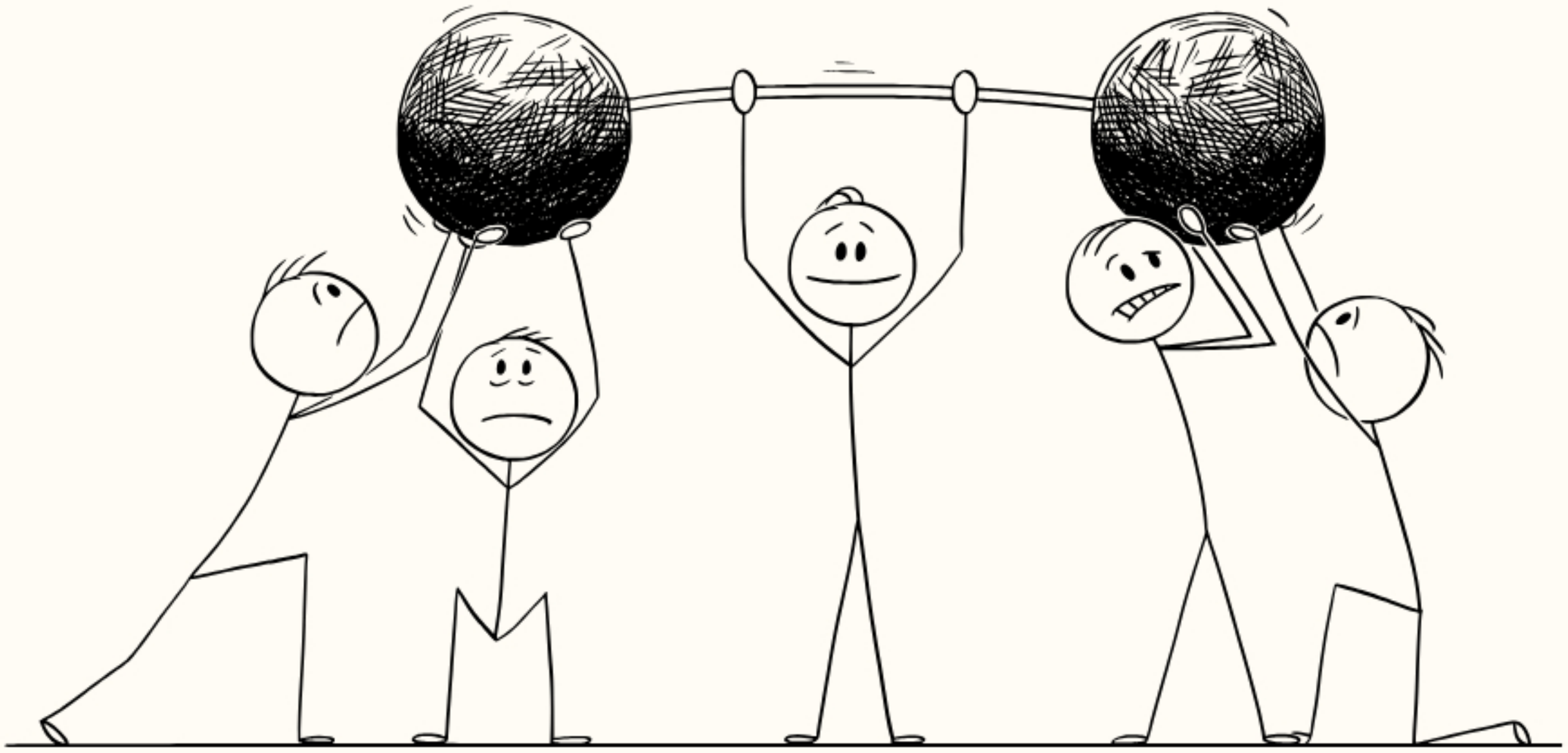






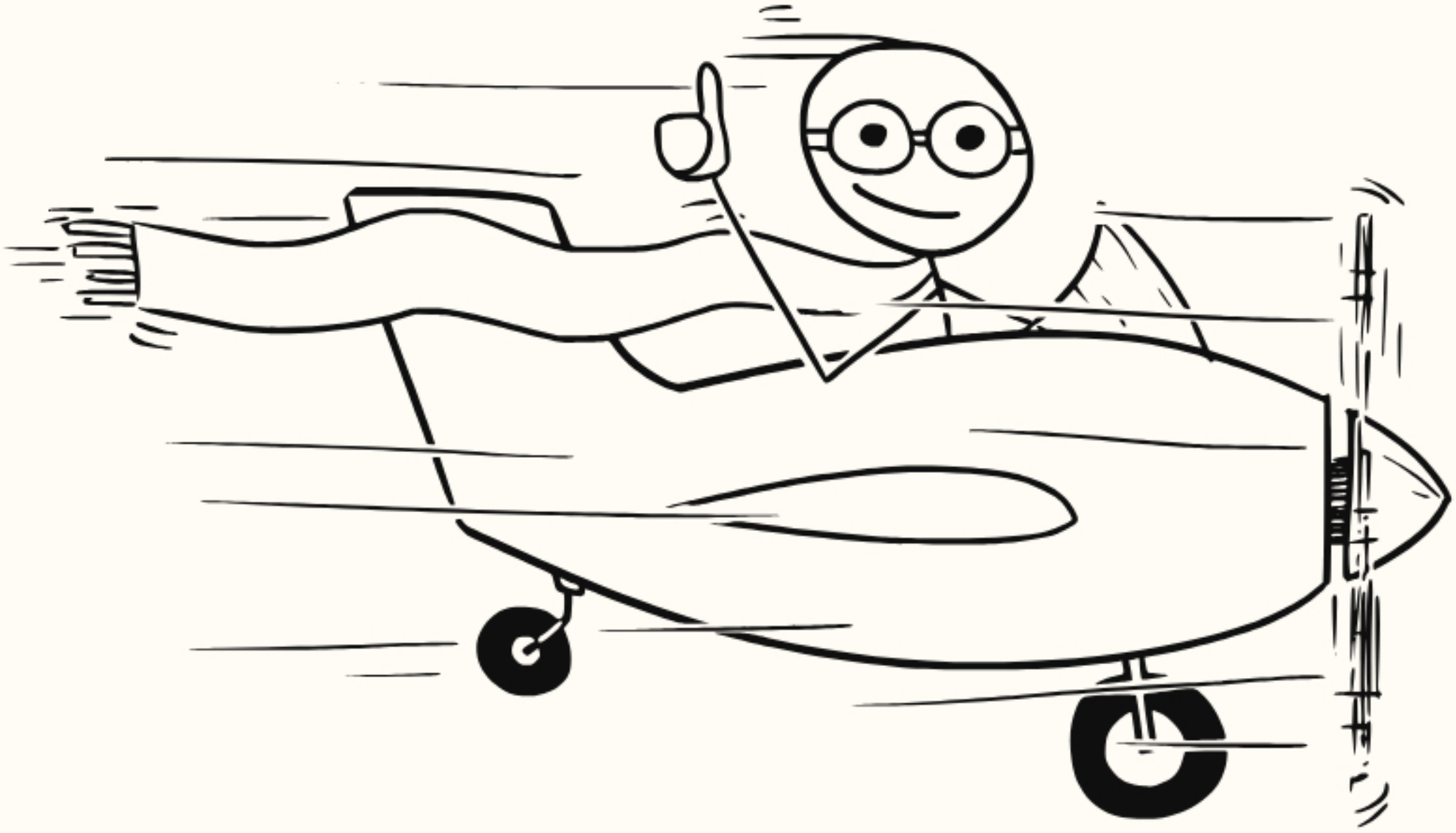
Cluster-level logging provides a separate backend to store, analyze, and query logs from various sources within K8s.

That said, K8s doesn't provide a native storage solution for hosting log data.



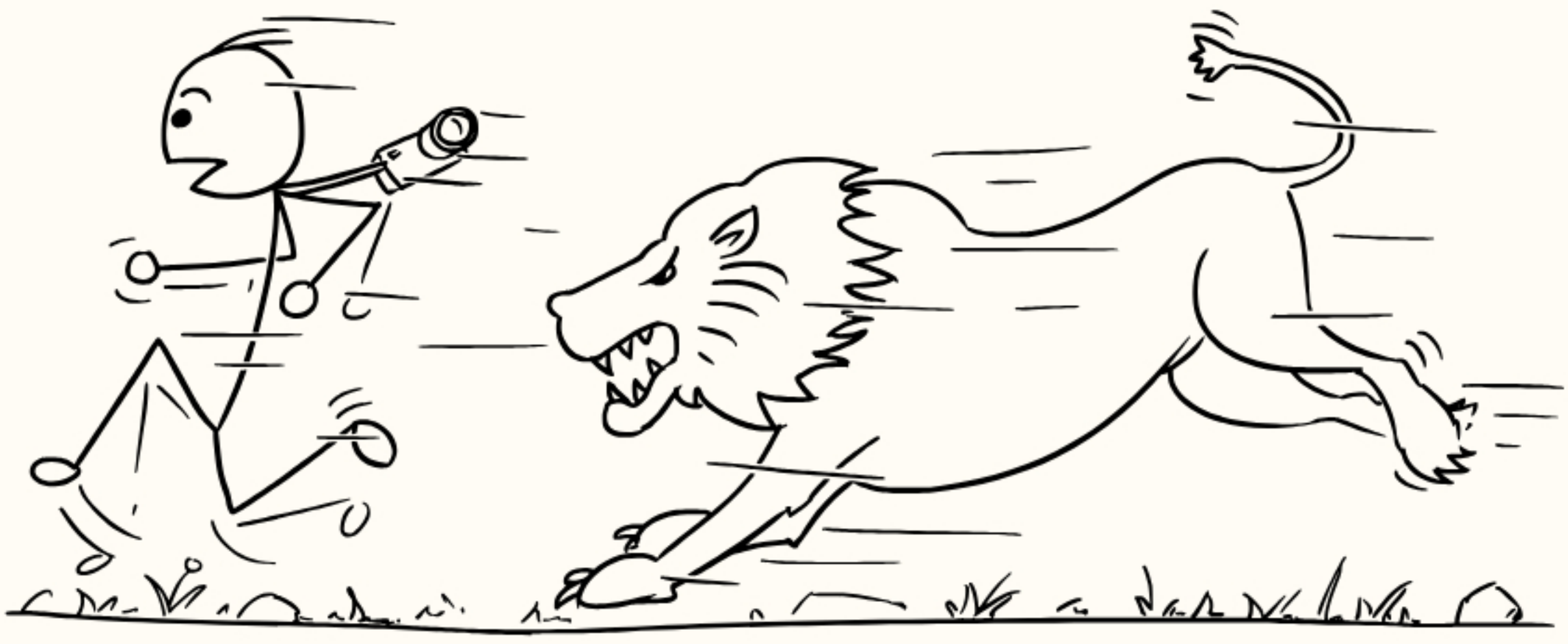
You have to implement that on your own, using a **third-party** logging solution that integrates with K8s.

Kubernetes generates log data by writing logs from control plane components.



Additionally, a container runtime manages and redirects application output to stdout and stderr streams, which can be converted into logs.

Different container runtimes handle these streams in different ways.



Still, they all make it possible to log data about the status of applications as long as those applications can expose data to **stdout** and **stderr**.

When choosing a Kubernetes logging strategy, consider log collection efficiency, scalability, and log rotation or archiving frequency.

Getting these aspects right makes K8s logs a valuable tool for administrators, providing actionable observability across the K8s architecture.