During a crash or failover, the ResourceManager (RM) process remains running but is fenced by ZooKeeper to prevent it from assuming the active role. In standby mode, several subsystems continue to operate:

* **JVM process** – remains active with the same process ID.
* **In-memory state** – application and node caches are preserved.
* **ZooKeeper RM State Store client** – stays connected or quickly reconnects.
* **Web UI (Jetty)** – serves limited standby pages.
* **Metrics/JMX** – monitoring and reporting remain active.
* **Security subsystems** – continue logins and keytab renewals.
* **Log aggregation helpers** – remain operational.

In contrast, a full RM restart terminates the JVM, stops all threads, closes network connections, and discards in-memory data. Upon restart, the RM must reload state from ZooKeeper, restart RPC servers, and reinitialize services, resulting in longer scheduling downtime.

Following the clean start on Aug 11, no communication issues have been observed between ZooKeeper and YARN RM.

In our case, a memory leak may be the root cause—though difficult to pinpoint using Datadog. Identifying it would require tools like jstack, but given the complexity, we will instead focus on monitoring ZooKeeper and tracking YARN heartbeat and connection status.

One drawback of failover without restart is that memory leaks or high heap usage persist, which over time can trigger anOutOfMemoryError (OOM)—potentially the issue we experienced.