**Databricks On-Call Runbook (1-Page)**

**🔹 Quick Navigation**

* **Job Output** → Workflows → Jobs → (Run) → see stacktrace
* **Spark UI** → From job/cluster
  + **Jobs Tab** → which action failed
  + **Stages Tab** → shuffle boundaries, slow stages
  + **Tasks Tab** → skew (long tails), spills, GC time
  + **SQL Tab** → query plan (Exchange = shuffle, BroadcastHashJoin = optimized)
  + **Environment Tab** → active configs
* **Logs** → Compute → Cluster → Driver/Worker logs → stdout/stderr
* **Cluster Events** → Compute → Cluster → Event Log (autoscaling, spot loss, policy violations)

**🔹 Failure → Where to Click → Fix Checklist**

**1) Driver OOM (Out-Of-Memory)**

* **UI path**: Jobs → Output (stacktrace) → Spark UI (last job) → Driver logs (stderr)
* **Checks**: collect() / toPandas() / big display? Driver memory too low?
* **Fixes**:
  + Replace with limit(), take(), or write() to Delta
  + Raise spark.driver.memory, adjust spark.driver.maxResultSize
  + Push aggregations to executors

**2) Executor Skew / Stragglers**

* **UI path**: Spark UI → Stages (slow stage) → Tasks (duration & shuffle read)
* **Checks**: Few tasks ≫ others? Hot key distribution? High GC/spill?
* **Fixes**:
  + Enable **AQE** + spark.sql.adaptive.skewJoin.enabled=true
  + Broadcast small table (/\*+ BROADCAST \*/)
  + Salt hot keys, repartition by join key
  + Pre-aggregate before joins

**3) Shuffle Fetch Failure**

* **UI path**: Spark UI → Failed stage (FetchFailed) → Worker logs (map-side) → Cluster Events
* **Checks**: Executor lost? Disk full? Network timeout? Huge shuffle blocks?
* **Fixes**:
  + Raise retry/timeouts:
  + spark.shuffle.io.maxRetries=10
  + spark.shuffle.io.retryWait=5s
  + spark.network.timeout=600s
  + Increase shuffle partitions / repartition earlier
  + Broadcast join small tables
  + Use AQE to split skewed partitions

**4) Job Stuck in Pending**

* **UI path**: Jobs → Run → Status (Queued vs Waiting for cluster) → Cluster Events
* **Checks**: Concurrency cap? Policy violation? Spot capacity issue? Init script hang? Cloud quota?
* **Fixes**:
  + Switch to allowed node type / region
  + Disable spot for critical runs
  + Use pool / attach to existing cluster
  + Stagger concurrent jobs
  + Simplify/skip init scripts

**🔹 Config Quickies (set in notebook)**

# Adaptive Query Execution (AQE)

spark.conf.set("spark.sql.adaptive.enabled", "true")

spark.conf.set("spark.sql.adaptive.skewJoin.enabled", "true")

# Shuffle & network stability

spark.conf.set("spark.sql.shuffle.partitions", "auto")

spark.conf.set("spark.shuffle.io.maxRetries", "10")

spark.conf.set("spark.shuffle.io.retryWait", "5s")

spark.conf.set("spark.network.timeout", "600s")

**🔹 Senior Engineer Reminders**

* Always start in **Spark UI → Stages tab** → find skew/shuffle bottlenecks.
* If tasks retry repeatedly, check **Worker stderr** for OOM / disk / spot eviction.
* If cluster won’t start, open **Event Log** → capacity, quota, or policy errors.
* Guardrails: never collect() full data → always limit() or write out.

**On-Call Shortcut**

* **Stage stuck?** → Spark UI → **Stages tab** (look for skew, shuffle, spill).
* **App won’t start?** → Cluster → **Event Log** (init, policy, quota, spot).
* **Driver crash?** → Driver logs (stderr) for OOM / GC overload.
* **Python UDF crash?** → Executor stderr for Arrow/memory errors.

| **Failure Type** | **Symptoms (UI / Logs)** | **Where to Click** | **Quick Fixes** |
| --- | --- | --- | --- |
| **Driver OOM (Out-Of-Memory)** | - OutOfMemoryError in **Driver stderr**- Notebook dies after collect() / toPandas()- Spark UI: large stage results | Jobs → Run → OutputSpark UI → Jobs → Last jobCluster → Driver Logs (stderr) | - Replace collect() with limit(), take()- Write to Delta instead of driver fetch- Raise spark.driver.memory & spark.driver.maxResultSize |
| **Executor Skew / Stragglers** | - Stage stuck at 99%- Few tasks ≫ slower than others- High GC time / Shuffle spill in **Tasks tab** | Spark UI → Stages → (Slow stage)Tasks tab (sort by Duration, Shuffle Read) | - Enable **Adaptive Query Execution (AQE)** + skew join handling- Broadcast small table (/\*+ BROADCAST \*/)- Salt hot keys, repartition- Pre-aggregate before joins |
| **Shuffle Fetch Failure** | - FetchFailedException in logs- Stage fails at reduce side- Worker logs: I/O error, lost executor, disk full | Spark UI → Failed Stage → DetailsWorker Logs (map-side)Cluster Event Log | - Raise retries & timeouts:spark.shuffle.io.maxRetries=10spark.shuffle.io.retryWait=5sspark.network.timeout=600s- Increase shuffle partitions- Broadcast joins / AQE |
| **Job Stuck in Pending** | - Run = Queued / Waiting for cluster- Cluster in “Starting” forever- Event Log: capacity, quota, policy errors | Jobs → Run → StatusCluster → Event LogCompute → Instance Pools | - Change node type / region- Disable spot for critical jobs- Use instance pools / existing cluster- Stagger runs- Fix/skip init scripts |
| **Python Worker Crash / Pandas UDF OOM** | - Python worker exited unexpectedly- Arrow memory errors- SIGKILL in stderr | Jobs → Output (error)Cluster → Executor Logs → stderr | - Reduce Arrow batch size:spark.sql.execution.arrow.maxRecordsPerBatch=50000- Use iterator-based Pandas UDFs- Avoid toPandas() on huge DF- Increase executor memory |
| **Init Script / Library Install Failures** | - Cluster stuck in “Initializing”- Logs: Init script failed or Pip install timeout | Cluster → Event Log (startup)Driver Logs (stdout/stderr) | - Validate init script manually (dbfs:/, %sh)- Ensure scripts are **idempotent & fast**- Host libraries in reliable repo (e.g., DBFS, S3, PyPI mirror)- Add logging + timeouts to init scripts |