**When should you INCREASE the number of partitions?**

**Your job is under-utilizing CPU cores**

* If your job isn’t fully using the cluster (many cores are idle), you probably need more partitions to create more parallel tasks.

**Each task is processing too much data**

* If tasks take a very long time (e.g. minutes instead of seconds), partitions are probably too large. Consider splitting them so the workload is better balanced.

**Out-of-memory errors (OOM)**

* If tasks fail with OOM errors or cause heavy GC (garbage collection), it means partitions hold too much data → increase partition count to reduce data per partition.

**Skewed data**

* When some partitions are huge (skewed) while others are small → increase partitions + consider using salting or custom partitioning.

**For shuffle-heavy jobs**

* Wide transformations (e.g. groupByKey, reduceByKey, join) → more partitions = smaller shuffle blocks → less memory pressure during shuffle.

**Large dataset**

* As dataset size grows (say hundreds of GBs or TBs), increase partitions to avoid huge files per partition and spread processing.

**When should you DECREASE the number of partitions?**

**Too many small tasks**

* If job spends more time on task scheduling/management (task overhead) than actual computation (you see many tiny tasks), reduce partitions.

**Shuffle writes many small files**

* After shuffle, if you see thousands of tiny files (too many small output files), reduce partition count.

**Small dataset**

* If dataset is small, no need for hundreds of partitions — fewer partitions will reduce scheduling overhead.

**Final stage optimization**

* Before writing to disk, especially to a small number of output files (e.g. writing one file per partition), reduce the partition count using coalesce or repartition.