1. **Common workflow of spark**

* spark uses master slave architecture
* here one central coordinator and many distributed workers
* central coordinators : driver node
* distributed workers : executors
* standalone cluster manager : default cluster manager
* step 1 : creation of input RDDs; depends on input data
  + Data can be obtained from different data sources.
* Step 2 : post RDD creation
  + the RDD transformation operations like filter() or map() they run for creating new RDDs depending on the business logic.
  + If any intermediate RDDs are required to be reused for later purposes, we can persist those RDDs.
* Step 3 : and one thing
  + if any action-operations like first(), count() are present , spark launch parallel computation

1. **profilers in pyspark**

* Profilers are useful for data review
* Like if we want to ensure whether it is valid or not
* pyspark supports custom profilers ; why means these custom profilers are used to built predictive models
* profile: produces a system profile of some sort.
* stats: returns collected stats of profiling.
* dump: dumps the profiles to a specified path.
* add: This helps to add profile to existing accumulated profile.
* dump(id, path): This dumps a specific RDD id to the path given