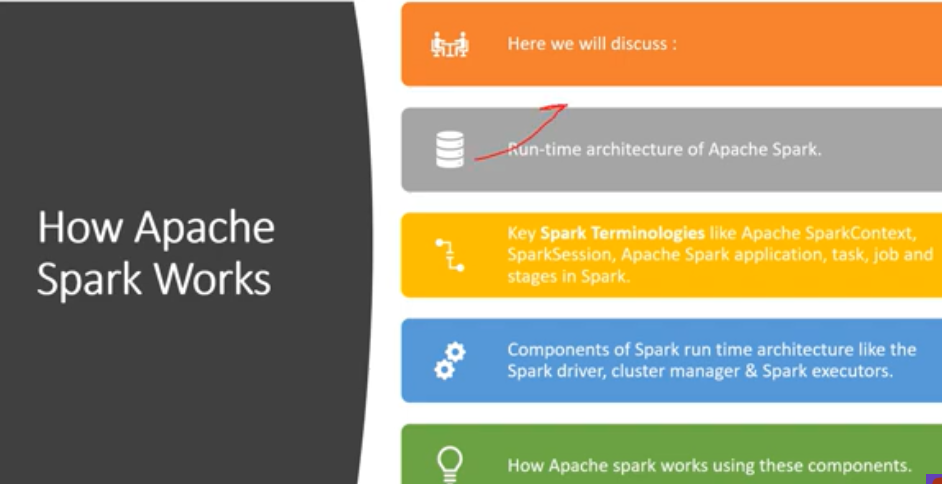
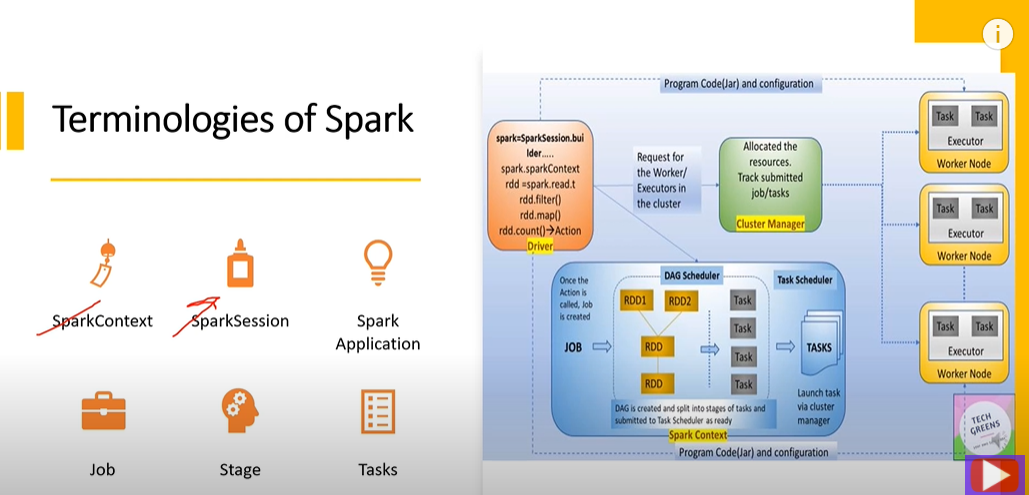
**How sparks works**





Sparkcontext and sparksession- both are the same thing, sparkcontext is available in previous version of spark and sparksession is the enhancement of sparkcontext. Sparksession is the wrapper for the spark context, sparksqlcontext and hivecontext.

As in previous version of spark, the spark context, sparksqlcontext and hivecontext these are the three different entities but now in later version of spark these come into the sparksession object.

So what is sparksession or sparkcontext, it a process which governs the entire execution of spark application. whenever you run the spark application, the first process that runs is spark session or spark context and then it collaborate with the cluster manager and get the resources and do the actual execution of the diff transformation and action.

Sparksession is the encompassing process which overlooks the entire execution of spark application. Spark application is the combination of diff transformation and action, combine together you trying to achieve some output or some volume of data.

JOB- what is job:

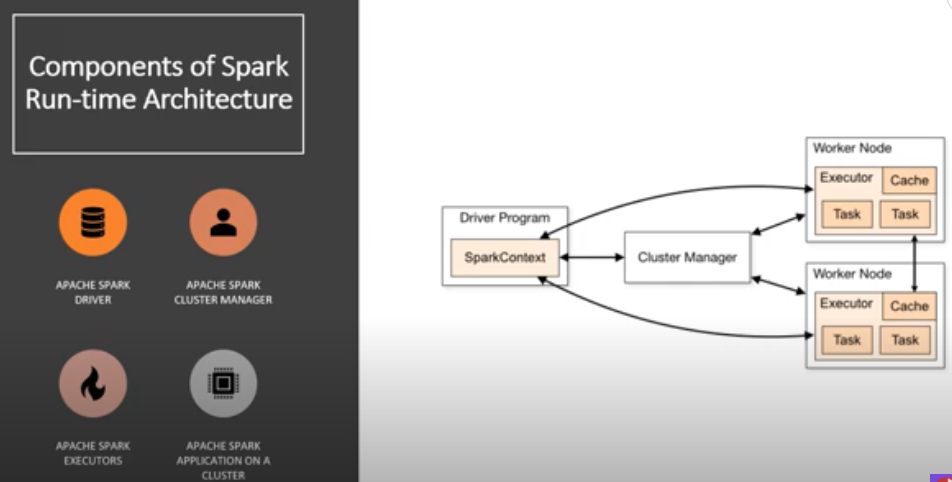
Whenever you perform any action on to your spark application then a job is initiated, for each action there is a job. Job is nothing but a collection of stages and stage is a collection of task.

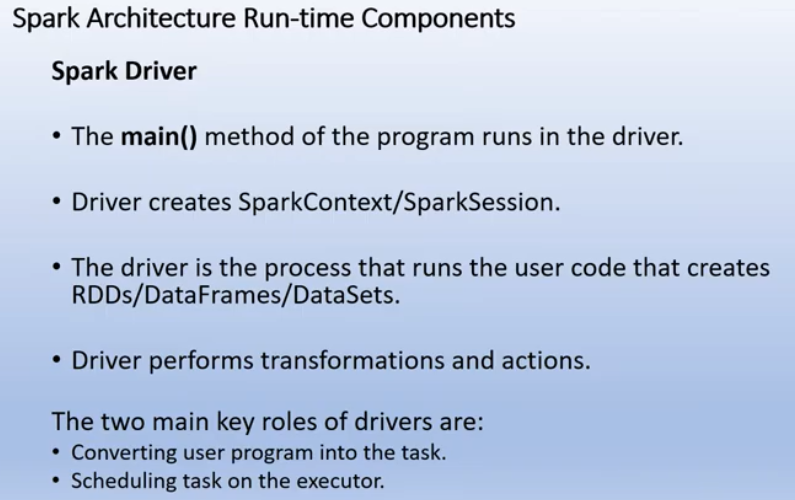
Task is the lowest unit of actual physical execution for a particular transformation on an individual partition of data

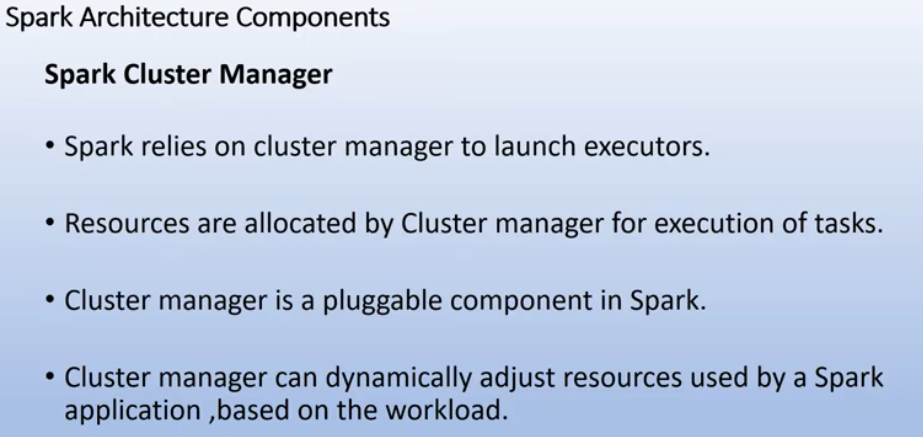
Job- initiated per action in spark

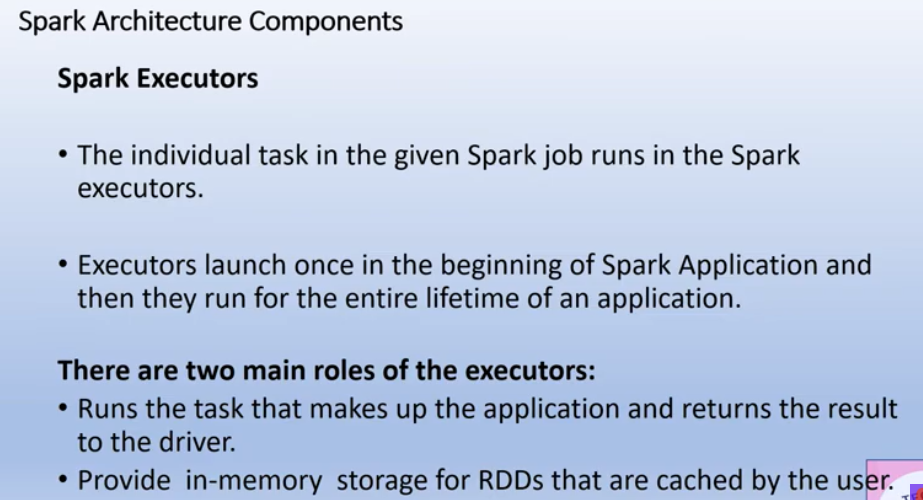
Different stages were created based on the shuffle bounderies and each stage is a collection of task and task is the actual physical unit of execution which is executed on each partition of data

**RUNTIME ARCHITECTURE OF SPARK:**





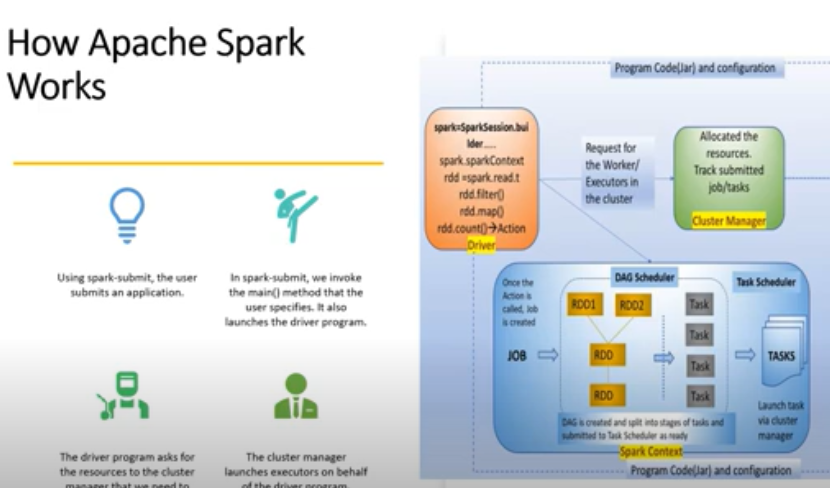




Driver program- it’s the program which initiate the spark context, it is the program where spark context or spark session is created then it govern the entire execution of spark job.

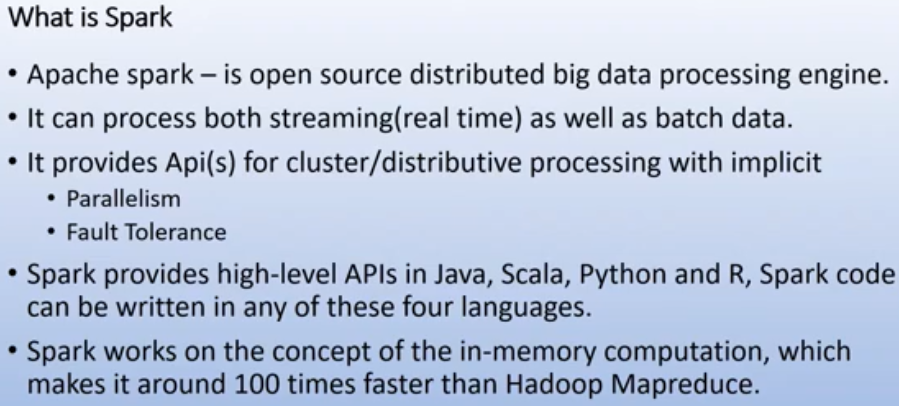
Cluster manager(yarn, mesos, standalone)-this is the element which talks to all the nodes in the cluster to do the book keeping of which executor is running where , what are the resource available in the cluster and also manage the distribution and allocation of those resources for the successful completion of spark application

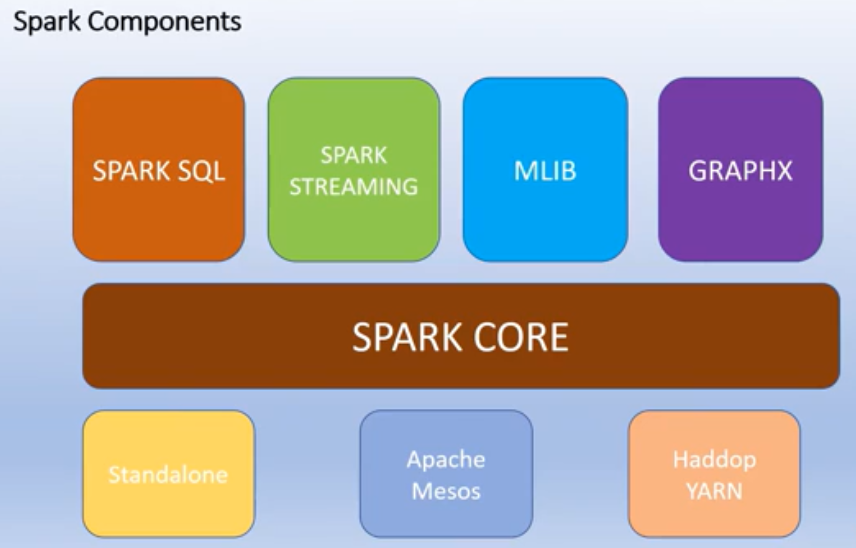
Executors are the worker threads which executes the one or more task

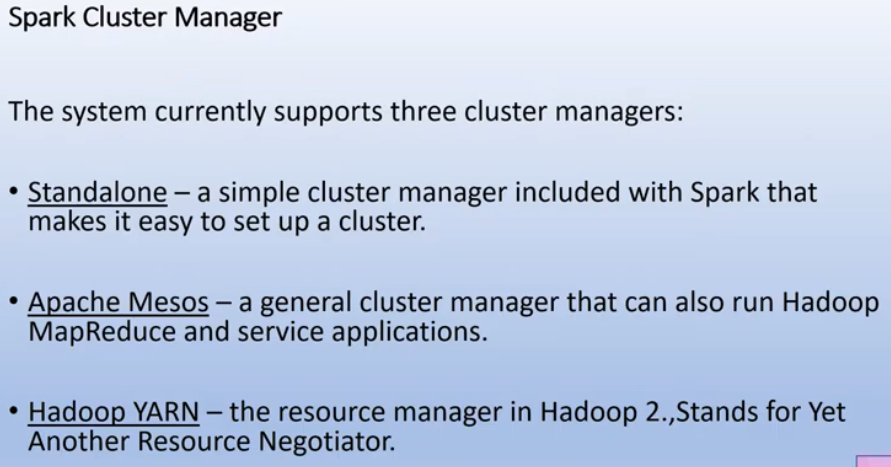


Let see how spark works, suppose we write a spark program using some diff transformations, with joins, shuffle, mapping. So first thing we do is we create a spark session, there would be one session for one spark application which will govern the entire execution of that particular application. As spark is lazily evaluated so every transformation you trying to operate which would go as node in DAG (directed acyclic graph), each transformation is available in the DAG, nothing would be evaluated as everything is lazily evaluated, all execution occur when action is performed.

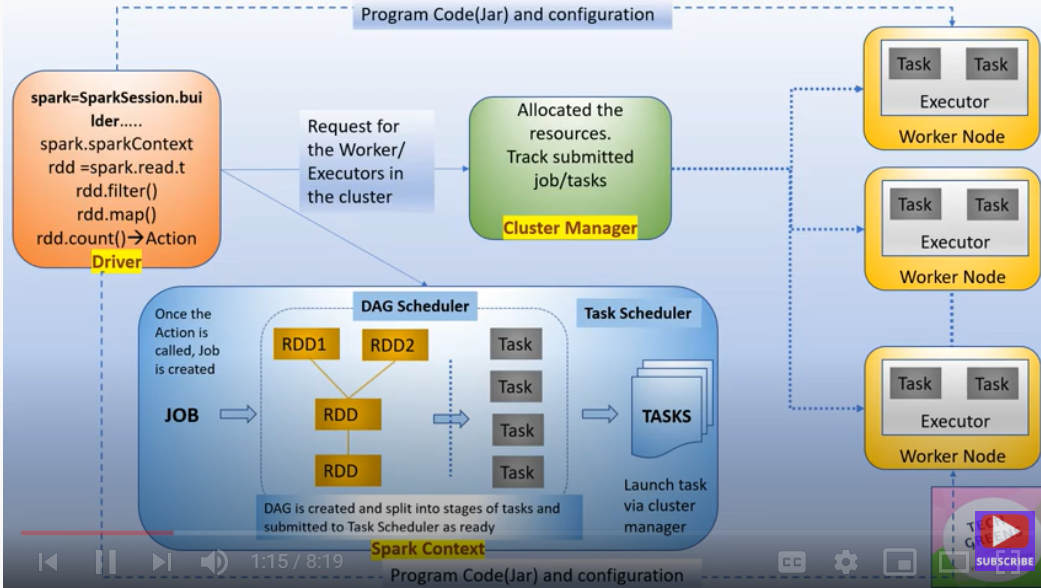
When action called, then spark context is initiated and that spark context is a place where DAG is kept so there is one other component which is DAG scheduler which is graph or lineage of diff transformation is available, based on this diff job is created, there would be one job per action then that job try to apply all these transformation as this is distributed programming the entire dataset is divided into diff partitions and each task or each transformation is individualy apply on one partition. As this job take DAG as a input based on that it will identify that it will run in single stage or separate stage, this is decide on the basis of wide and narrow transformation so each wide transformation result into diff stage and stage will be the collection of diff task and these task would be run by executor , each will need one core to run in parallel and then it will run on one individual partition so on that partition you apply transformation whether its join or filter or gourpby and that partition dataset will reshape into another rdd or dataframe beacause everything is immutable and eventually the driver program through the spark context is monitoring and controlling the execution of this entire process and everything eventually would be return or result goes back to the spark context whether the execution has happen successfully or not, also maintain fault tolerance whether any worker node executor is working or not or loose any connectivity for some reason, the spark context is collecting the heartbeat and it will initiate the new executor if that previous executor doesn’t start in a stipulated amount of time .







Deep dive on how spark works:



Once you do a spark submit, a driver program is launched, this driver program request for resource to the cluster manager and at the same time the main program of the user is called by driver program, based on that the execution logic is processed and parallel spark context is also created, using spark context the diff transformations are processed so till the time the action is not encountered, all the transformation will go to the sparkcontext in the form of DAG that will created the RDD lineage. Once the action is called job is created. As task is launched on the worker node through the cluster manager with the help of the class which is called task scheduler, there conversion of DAG into the task with the help of DAG scheduler.

DAG is created with diff transformation in the program and When action is called, these are split into diff stages and task and submitted to the task scheduler, then these tasks are launched in the diff executors in the worker node through the help of cluster manager

As when you do spark submit, the user program jar and other configuration are also copied on all the available nodes of the cluster so that program will become the local read on all the worker node so that the parallel executors running on the diff worker node do not have any kind of network routing and thus safe the network overhead.

