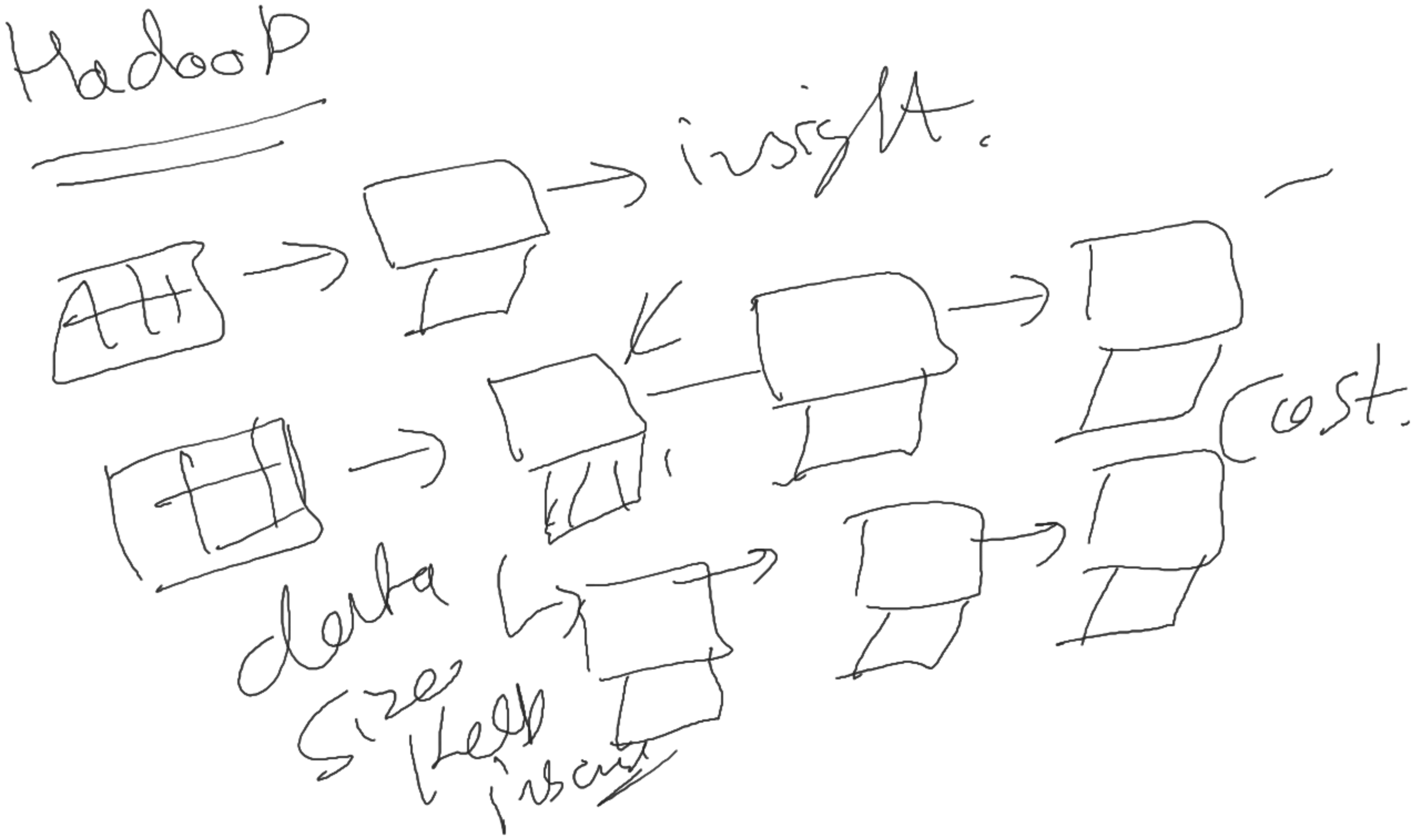
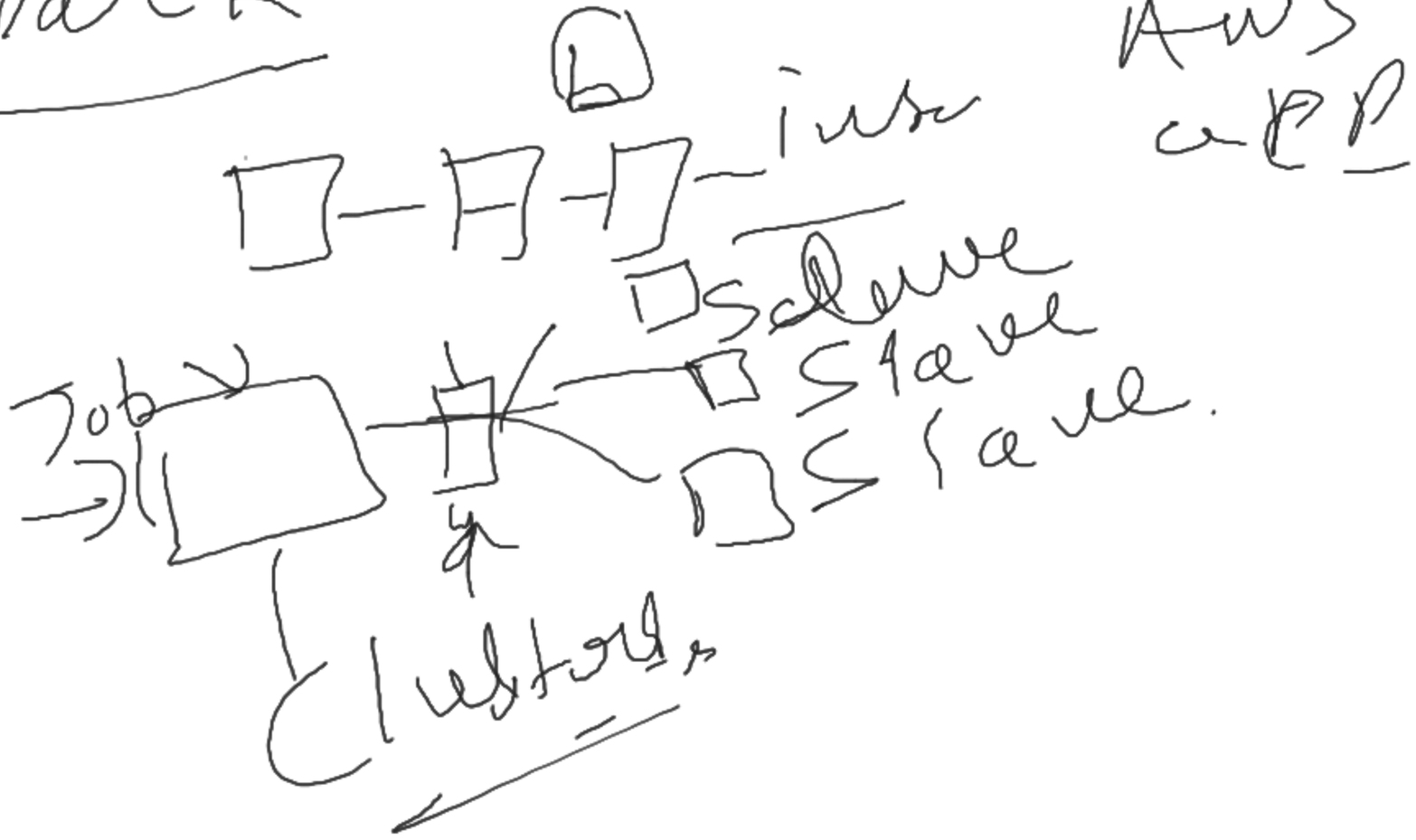

Hadoop
insight.
cost.
data
size
Hdfs
job

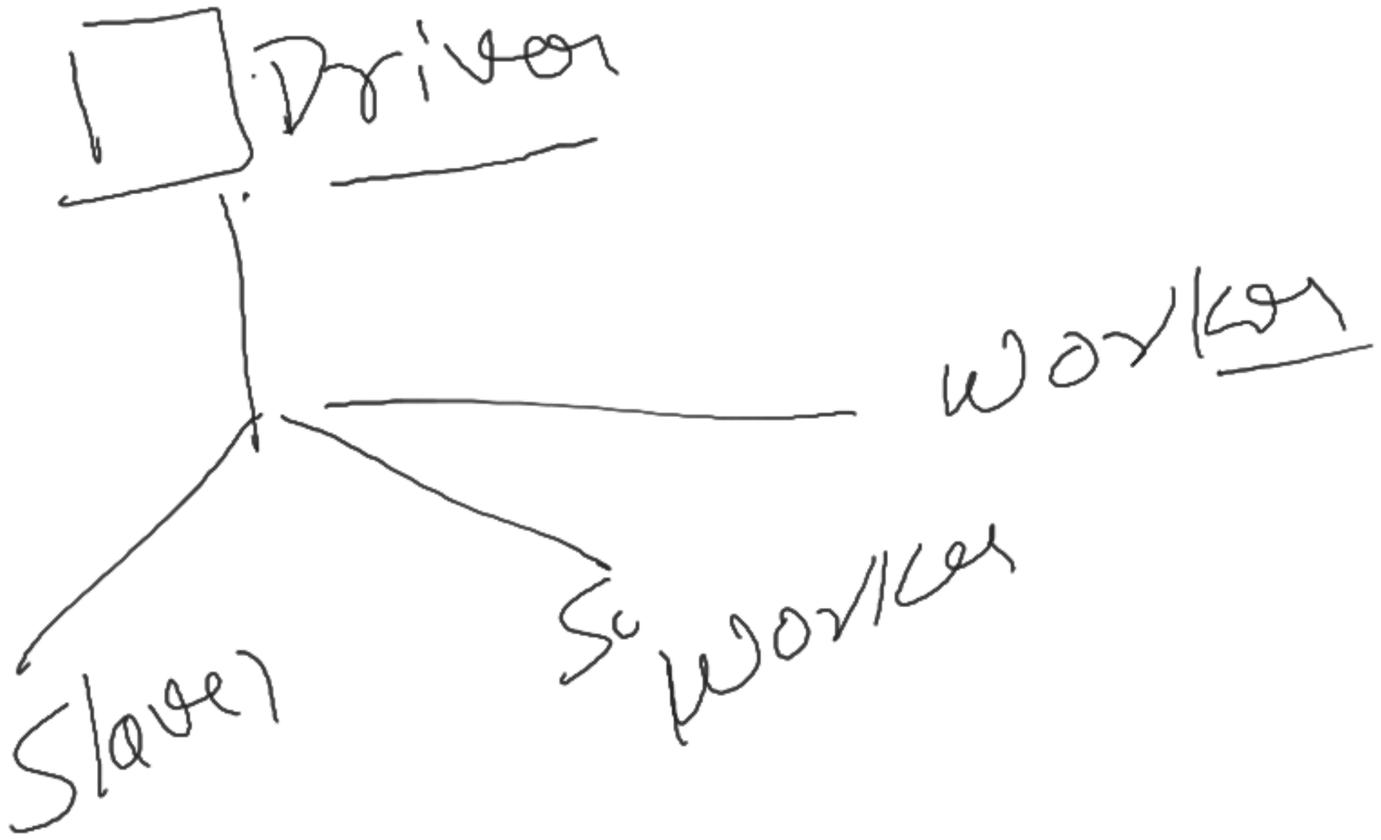
Google
AWS
Azure → insight

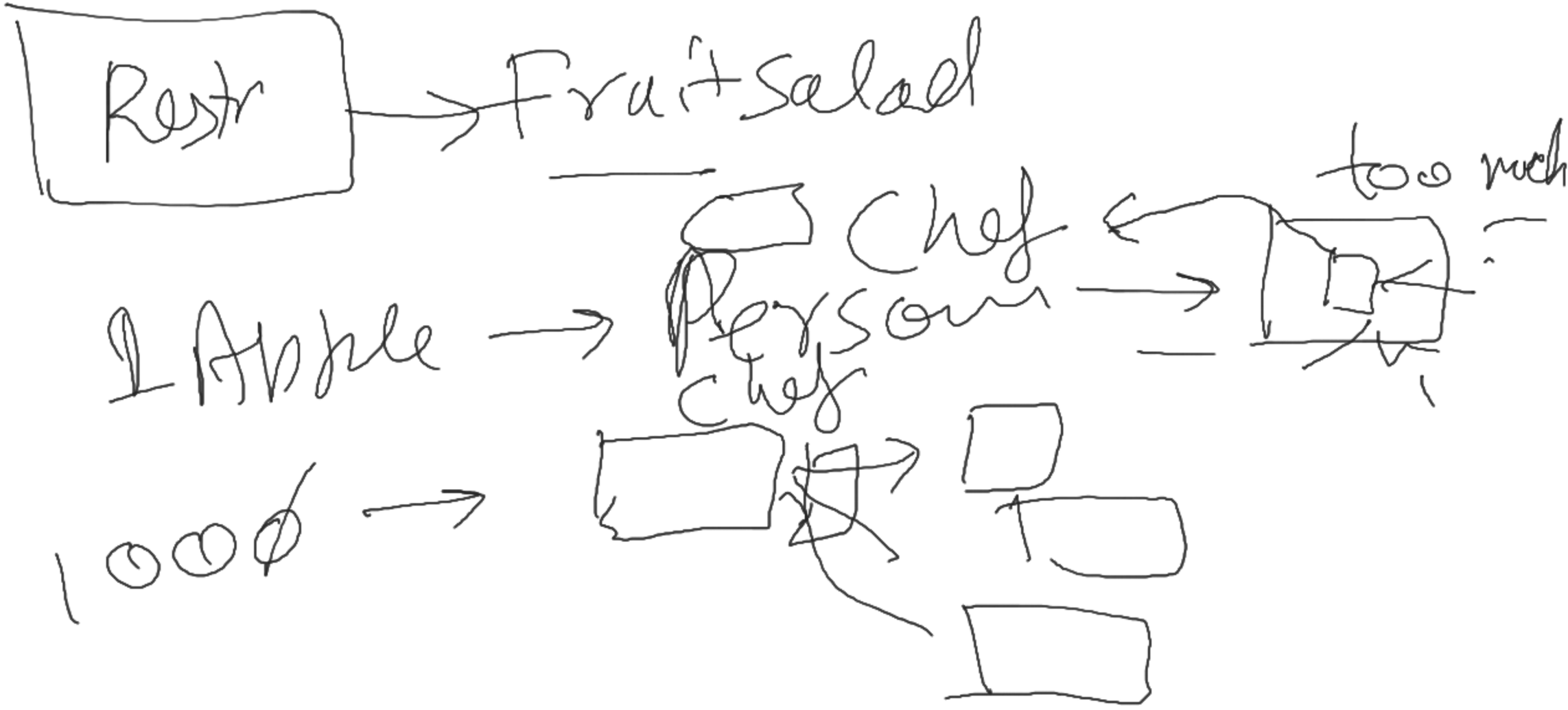
Hard disk

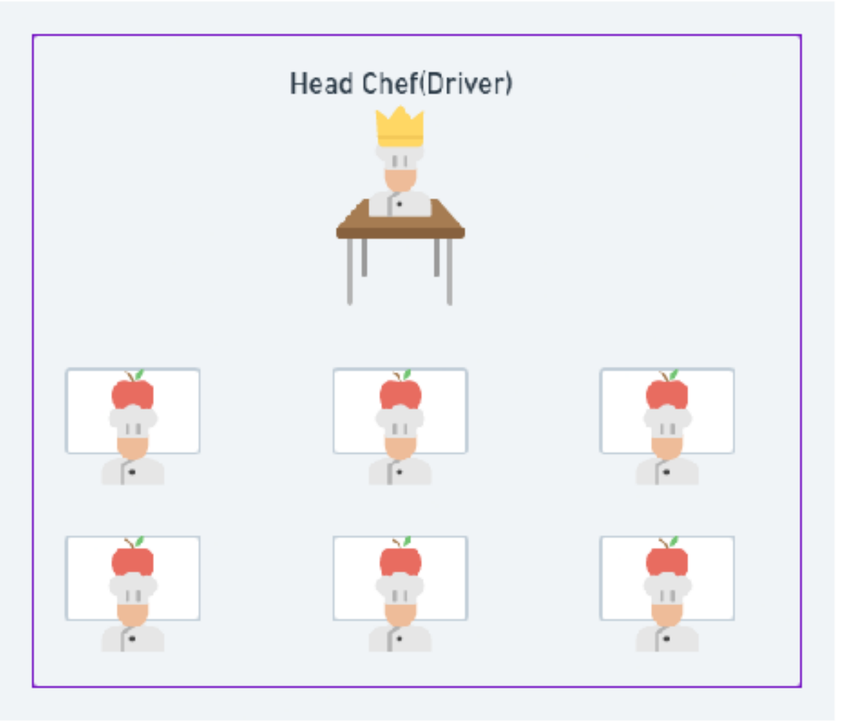
Spark → 100 time
in memory pools

Spark



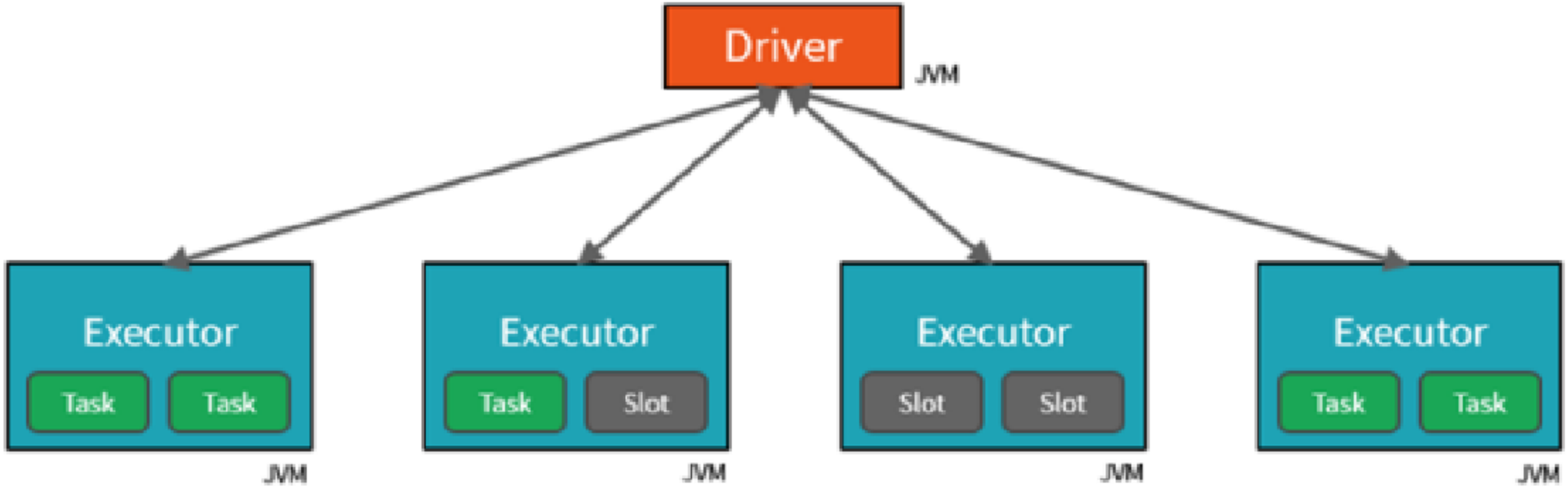

Driver
Workers
Slaves
Sc
Workers


Restr
Fruit salad
too much
Chef
1 Apple
Person
1000
Chef

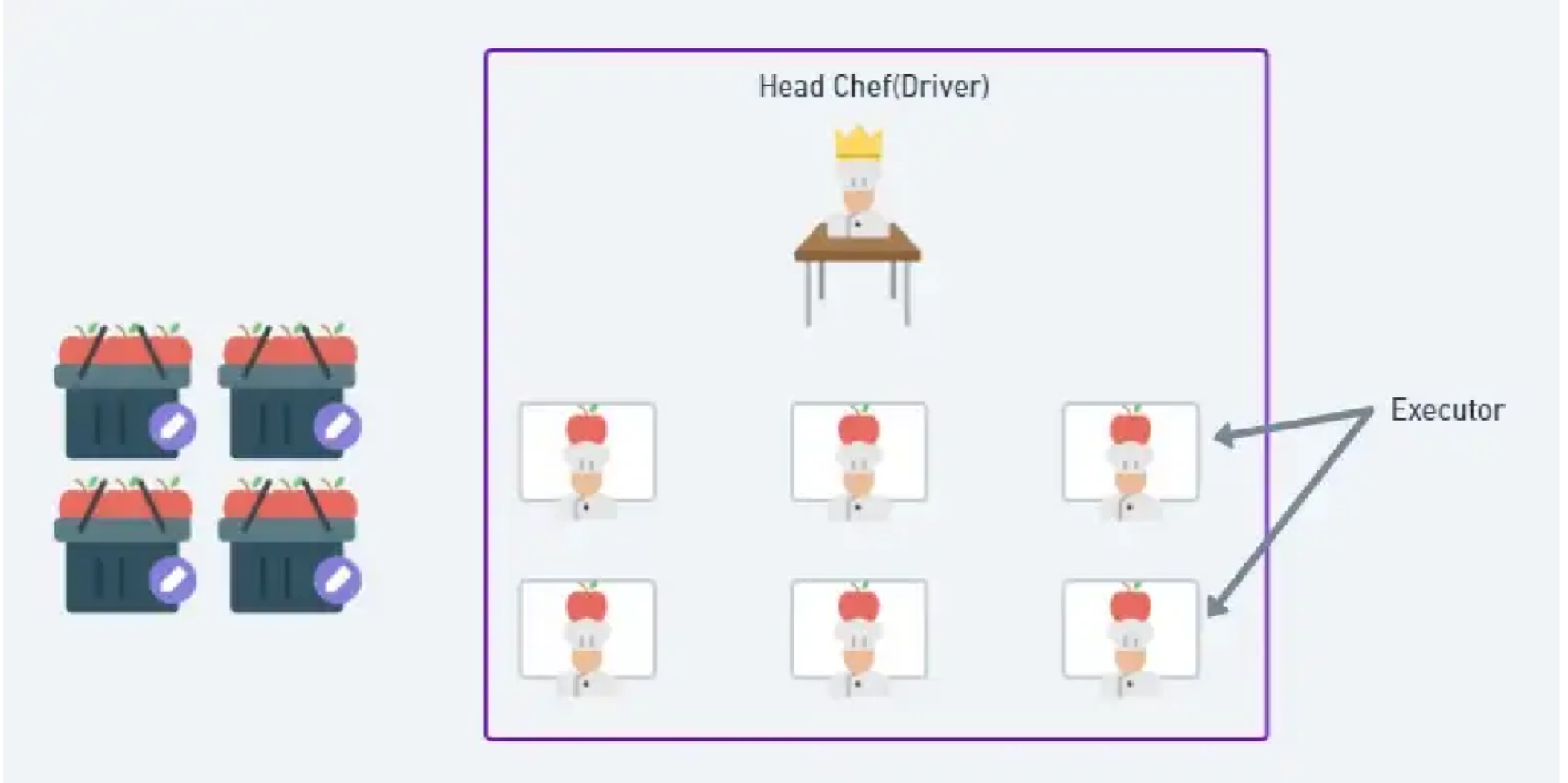


Driver: Consider Driver as a head chef/manager of the restaurant who distributes the work among the other chef.

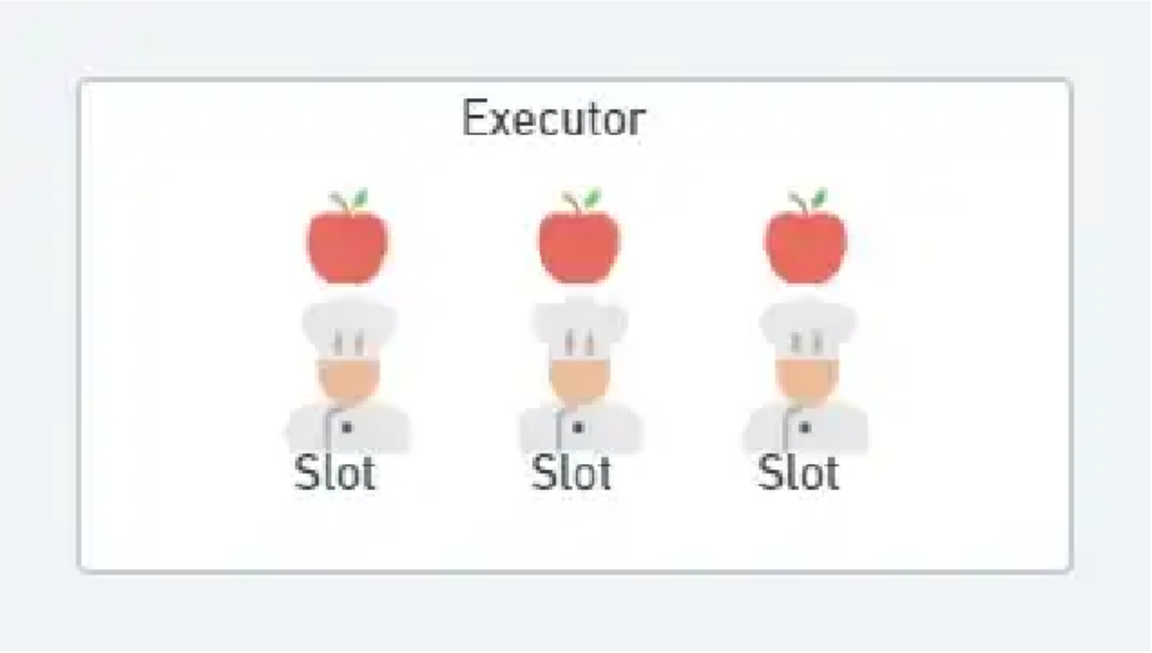
In spark, the Driver is responsible for assigning Tasks to the executors. From a Data engineer's perspective if we want to filter a table consisting of 1000 rows these rows will be divided and distributed across the executors by Driver. Note that the driver will not be allowed to view the data.


Driver
JVM
Executor
Task
Task
JVM
Executor
Task
Slot
JVM
Executor
Slot
Slot
JVM
Executor
Task
Task
JVM

Executors: Consider executors as a workstation/chopping board where the chef performs the task(cutting apples).



In spark, Cores/Slots are the CPU that performs the actual task. The total number of cores present is responsible for executing more tasks parallelly.



In spark, the executors are the Java Virtual Machine where your task is being performed. The executors are responsible for running the individual task and sending the result to the driver.

Slots/Cores: Slots or Cores are referred to the chefs performing the tasks of cutting apples. An executor consists of Cores/Slots. An executor(workstation) can consist of one or more cores/slots.

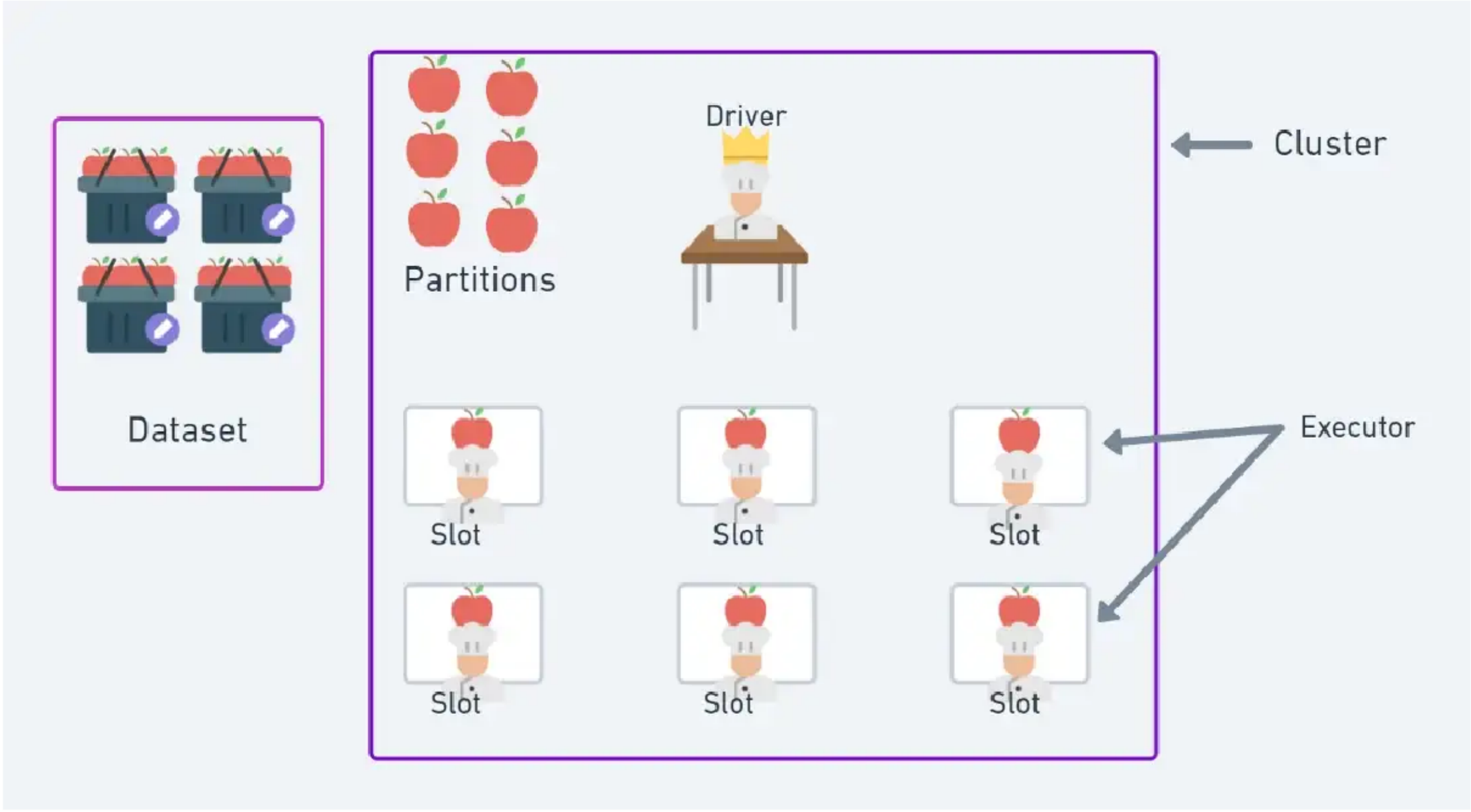
Jobs, Tasks, Dataset, Partitions:

Dataset is a collection of apples in a basket.

Partitioning is the process of splitting the dataset into smaller parts. Partitioning is done by the driver.

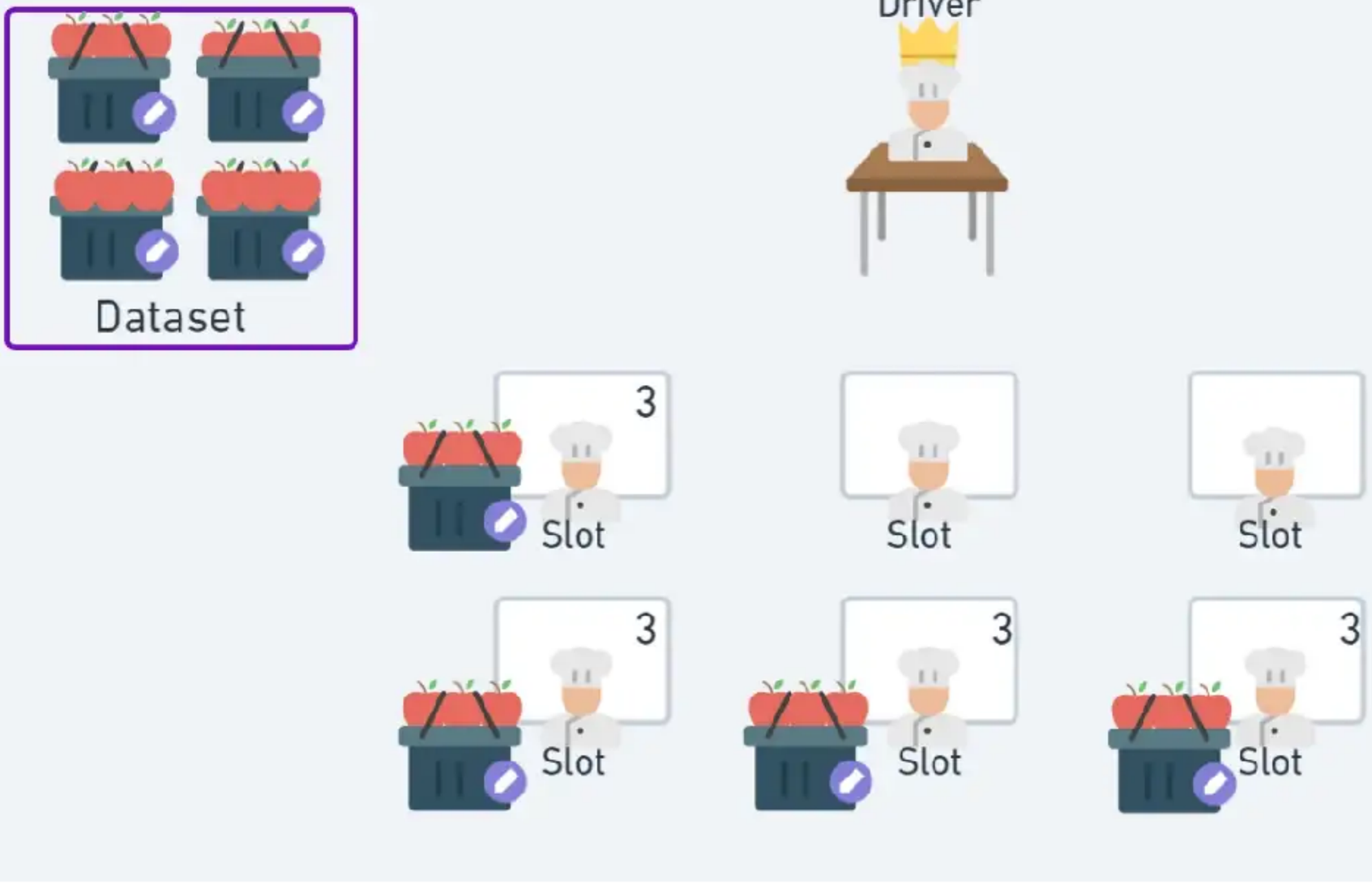
Tasks are usually the total number of partitions that needs to process for example if we have 4 baskets of 3 apples in each then we have 12 apples(Dataset) to be cut. This is divided into 12 partitions by driver hence 12 tasks have to be performed in total. Each task is performed by each core. Hence if we have 3 executors of 2 cores in each executor then we have 6 cores in total therefore there can be only a maximum of 6 tasks that can be performed parallelly.

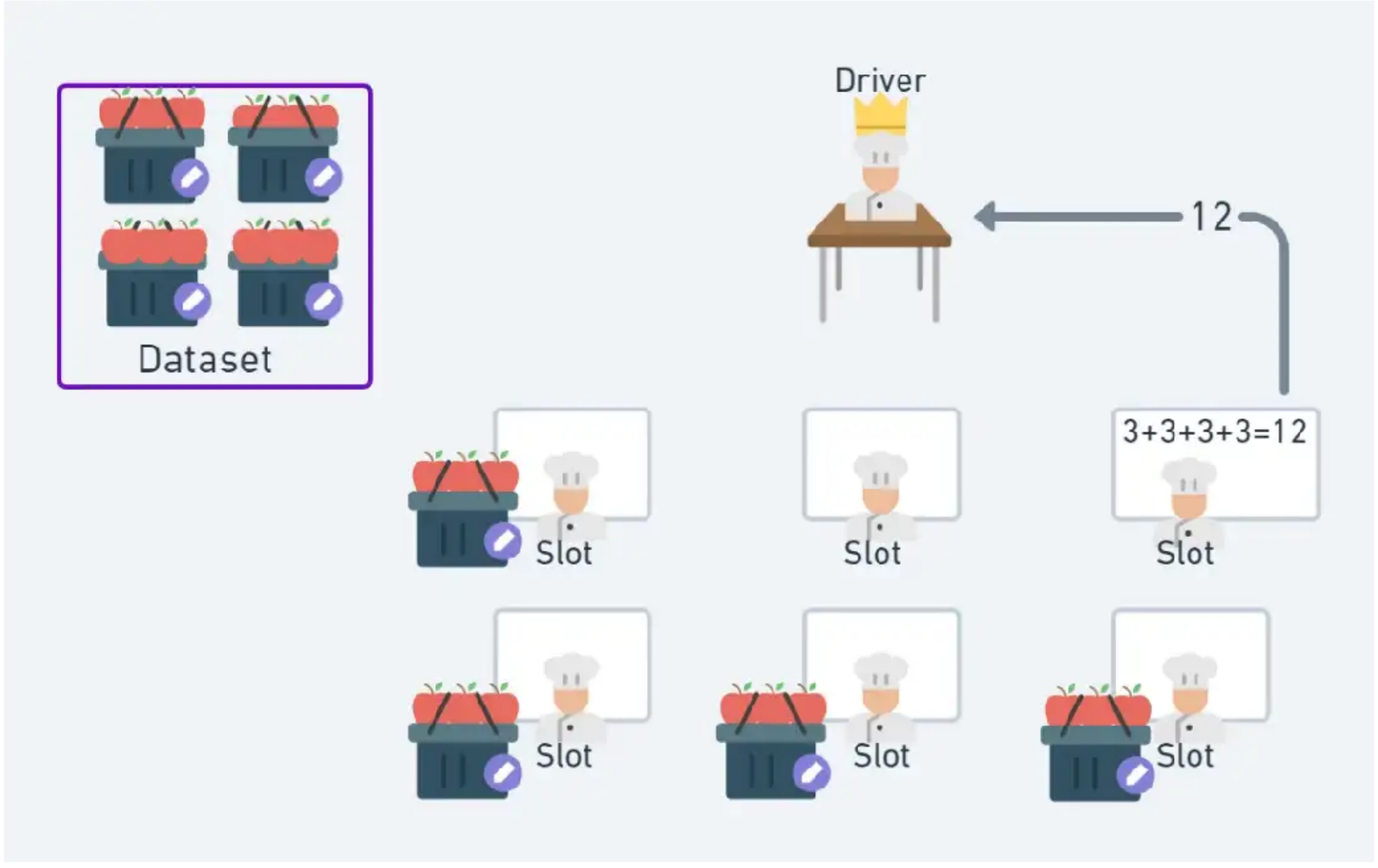
Job is the overall instruction that the driver receives. In this case, we need to cut 4 baskets of 3 apples in each.

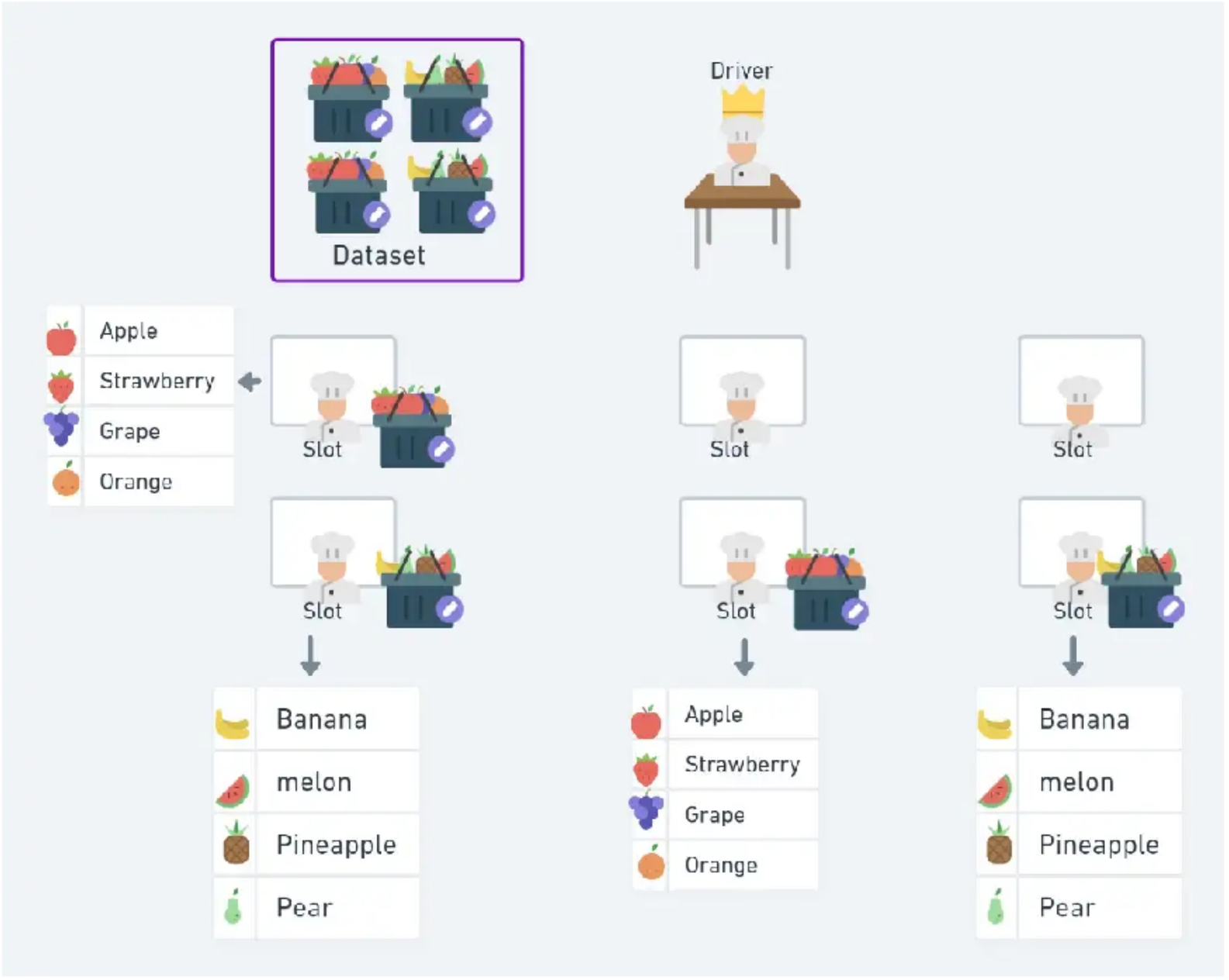

Dataset
Partitions
Driver
Cluster
Slot
Slot
Slot
Slot
Slot
Slot
Executor

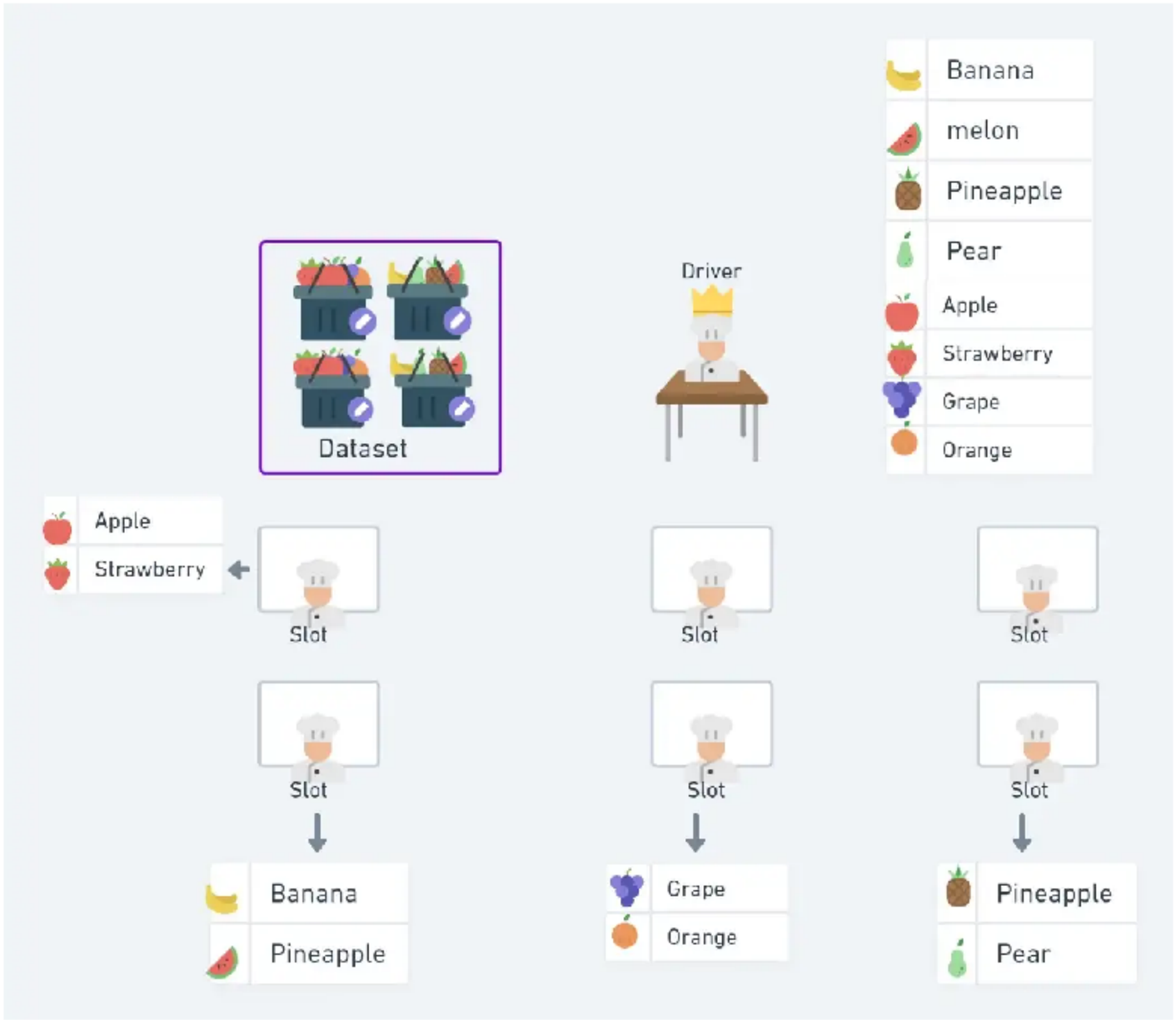
Stages:
Stages in Spark represent groups of tasks that can be executed together to compute the same operation on multiple machines. As we know already that the driver is not allowed to see the dataset then have you wondered how the driver knows the total no of rows in the dataset or apples in our example? Spark tackles this issue by distributing the dataset across executors and asking them for the count. Let's understand the entire process by going through two stages.

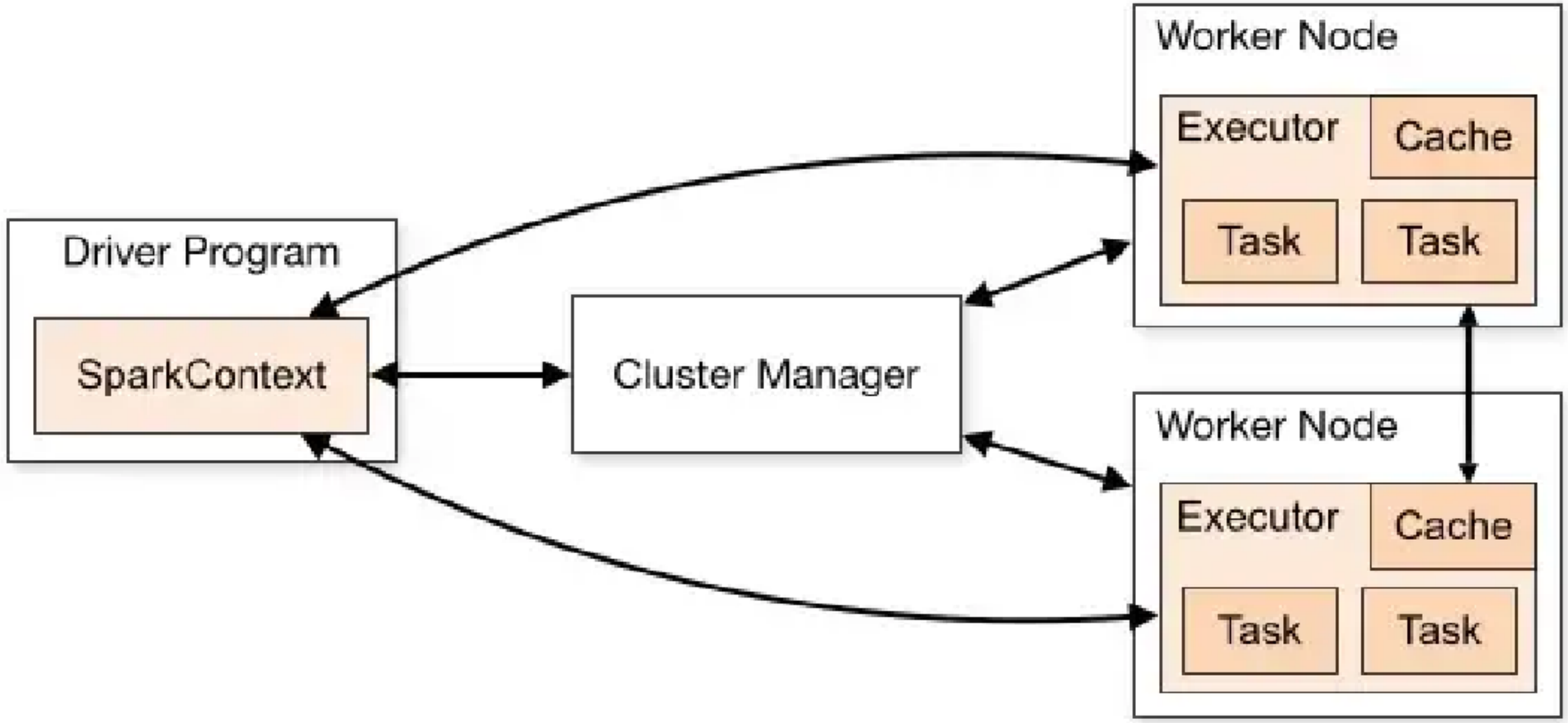
Stage 1: In this stage, the driver divides the dataset and distributes them to a set of executors. The driver now asks the executors to count the number of rows in the dataset. In this stage, the work of the executors is to find the no of apples in each basket. Now the count is stored within each executor on the disk to use in later stages this is also called shuffle data.


Dataset
Driver
3
Slot
Slot
Slot
3
Slot
3
Slot
3
Slot


Dataset
Driver
12
3+3+3+3=12
Slot
Slot
Slot
Slot
Slot
Slot


Dataset
Driver
Apple
Strawberry
Grape
Orange
Slot
Slot
Slot
Slot
Slot
Slot
Banana
melon
Pineapple
Pear
Apple
Strawberry
Grape
Orange
Banana
melon
Pineapple
Pear


Dataset
Driver
Banana
melon
Pineapple
Pear
Apple
Strawberry
Grape
Orange
Apple
Strawberry
Slot
Slot
Slot
Slot
Slot
Slot
Banana
Pineapple
Grape
Orange
Pineapple
Pear


Worker Node
Executor
Cache
Task
Task
Driver Program
SparkContext
Cluster Manager
Worker Node
Executor
Cache
Task
Task

```
orders_df=spark.read.csv('dbfs:/FileStore/tables/Data/orders/part_00000',schema="""order_id INT,order_date DATE,order_customer_id INT,order_status STRING""")
```