Apache Spark

Execution Mode

Prof. Venus R. Patel

The Spark driver

- The driver is the process “in the driver seat” of your Spark Application.
- It is the controller of the execution of a Spark Application and maintains all of the state of the Spark cluster (the state and tasks of the executors).
- It must interface with the cluster manager in order to actually get physical resources and launch executors.
- To conclude, This is just a process on a physical machine that is responsible for maintaining the state of the application running on the cluster.

The Spark executors

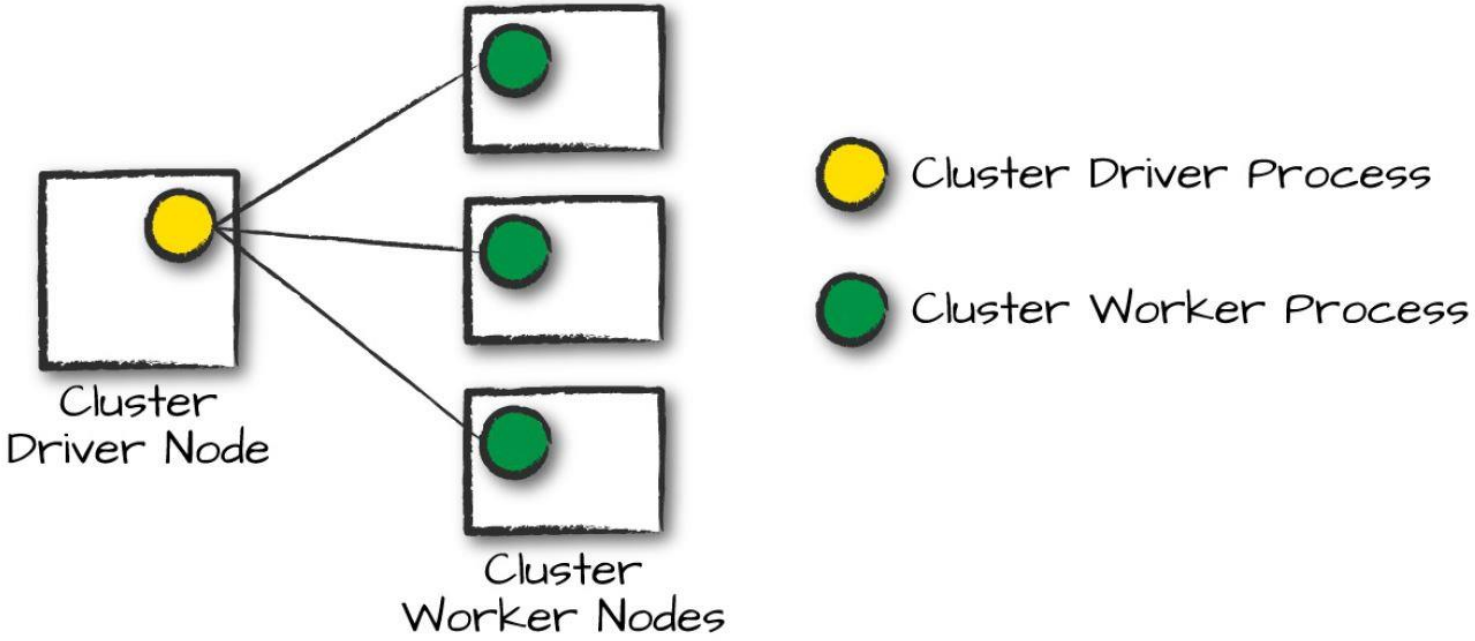
- Spark executors are the processes that perform the tasks assigned by the Spark driver.

- Executors have one core responsibility: take the tasks assigned by the driver, run them, and report back their state (success or failure) and results.

- Each Spark Application has its own separate executor processes.

The Cluster manager

- The Spark Driver and Executors do not exist in a void, and this is where the cluster manager comes in.
- The cluster manager is responsible for maintaining a cluster of machines that will run your Spark Application(s).



Execution Modes

Execution Modes

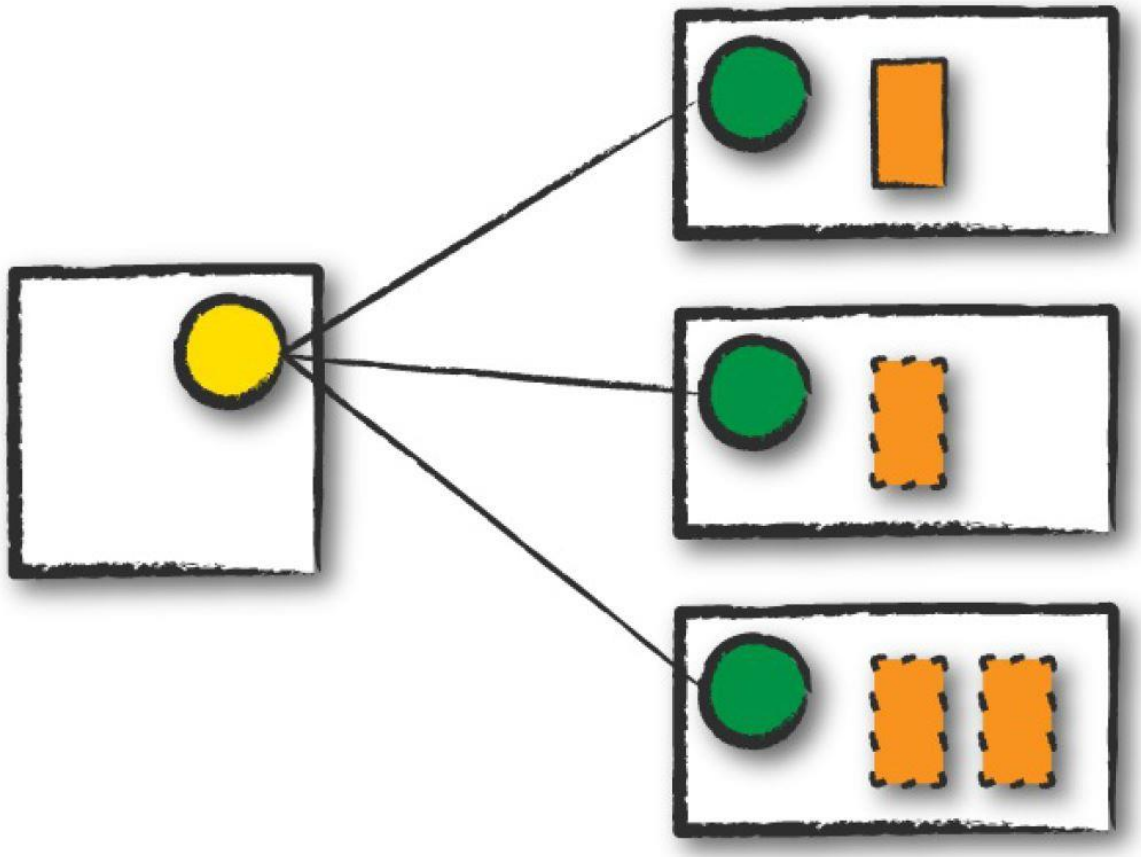
- An execution *mode* gives you the power to determine where the aforementioned resources are physically located when you go to run your application. You have three modes to choose from:

1. Cluster mode
2. Client mode
3. Local mode

1. Cluster mode

- Cluster mode is probably the most common way of running Spark Applications.
- In cluster mode, a user submits a pre-compiled JAR, Python script, or R script to a cluster manager.
- The cluster manager then launches the driver process on a worker node inside the cluster, in addition to the executor processes.
- This means that the cluster manager is responsible for maintaining all Spark Application–related processes.

1. Cluster mode



1. Cluster mode

- Rectangles with solid borders represent Spark *driver process* whereas those with dotted borders represent the *executor processes*.

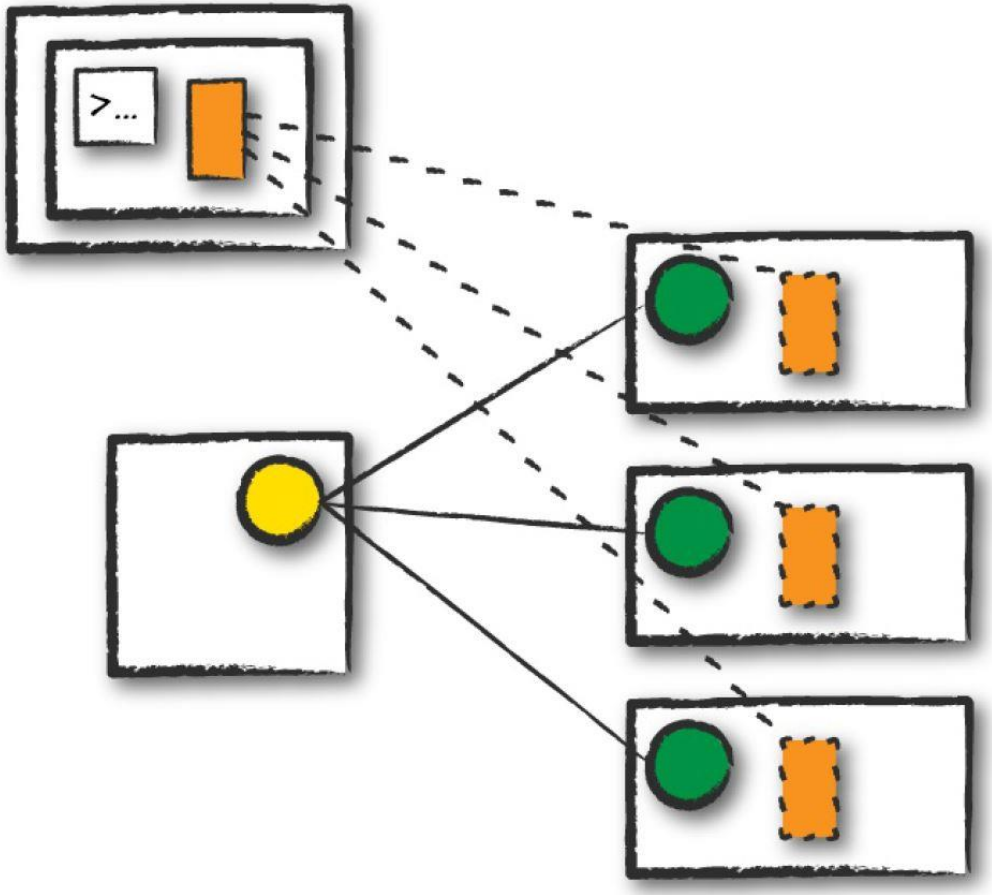
- The cluster manager placed our driver on a worker node and the executors on other worker nodes.

2. Client mode

- Client mode is nearly the same as cluster mode except that the Spark driver remains on the client machine that submitted the application.

- This means that the client machine is responsible for maintaining the Spark driver process, and the cluster manager maintains the executor processes.

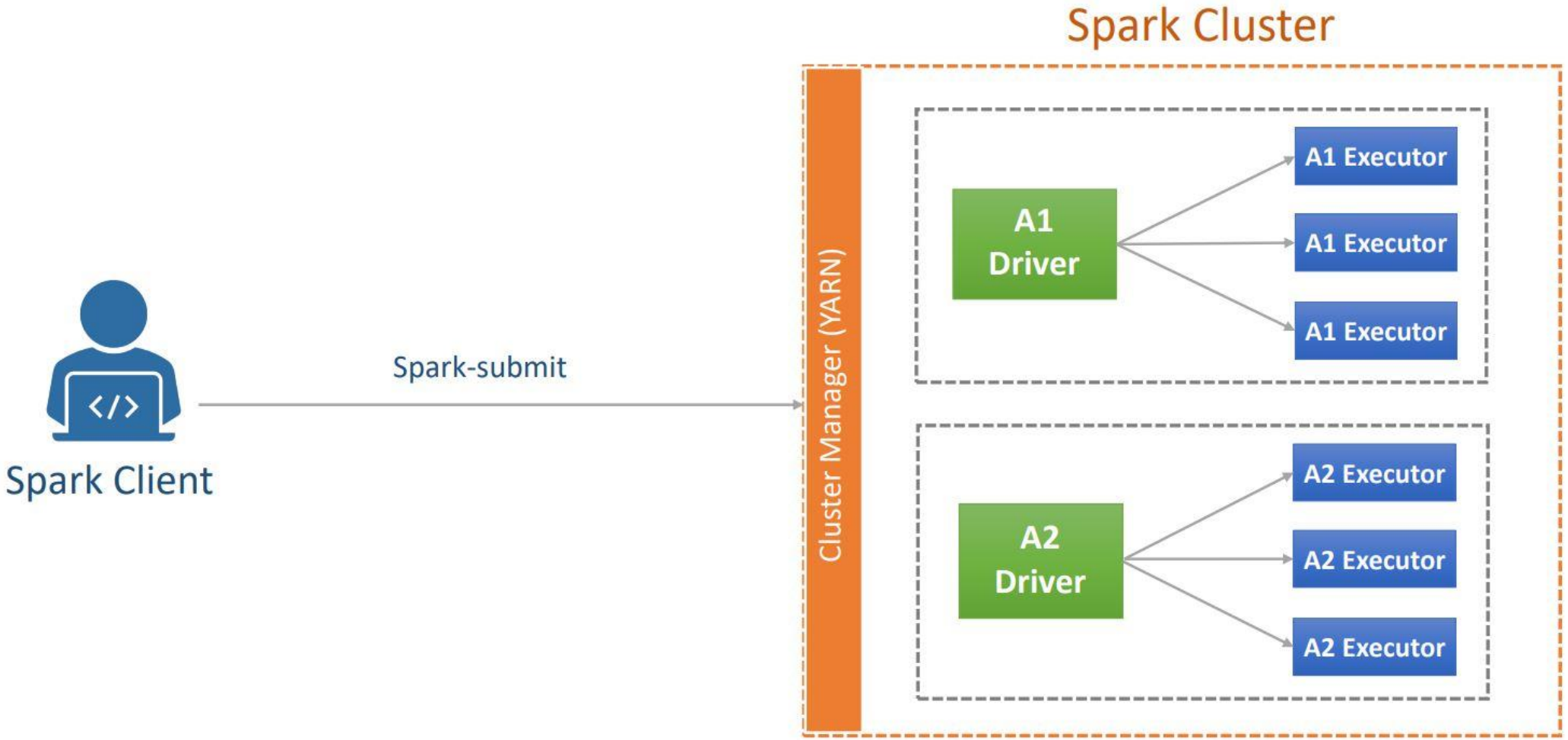
2. Client mode



3. Local mode

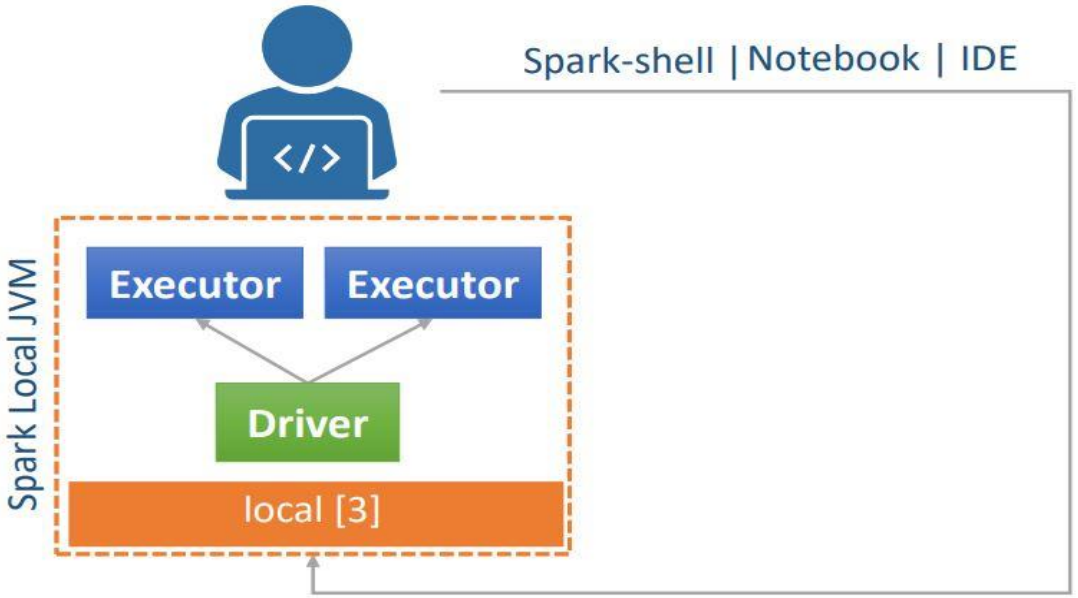
- Local mode is a significant departure from the previous two modes: it runs the entire Spark Application on a single machine.
- It achieves parallelism through threads on that single machine.
- This is a common way to learn Spark, to test your applications, or experiment iteratively with local development.

Summary :

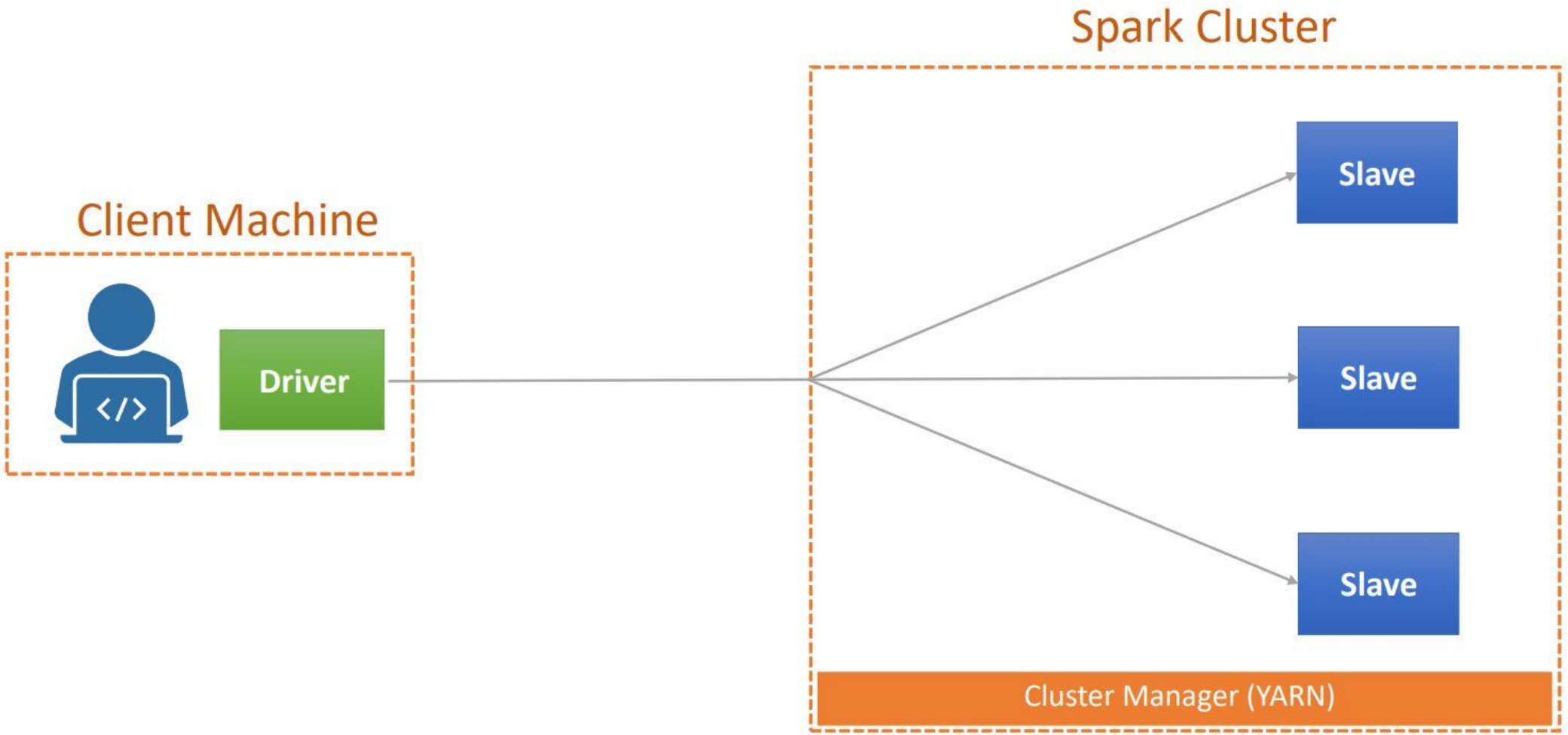

Spark Cluster
Spark Client
Spark-submit
Cluster Manager (YARN)
A1 Driver
A1 Executor
A1 Executor
A1 Executor
A2 Driver
A2 Executor
A2 Executor
A2 Executor

Summary :

1. local[n]
2. YARN
3. Kubernetes
4. Mesos
5. Standalone

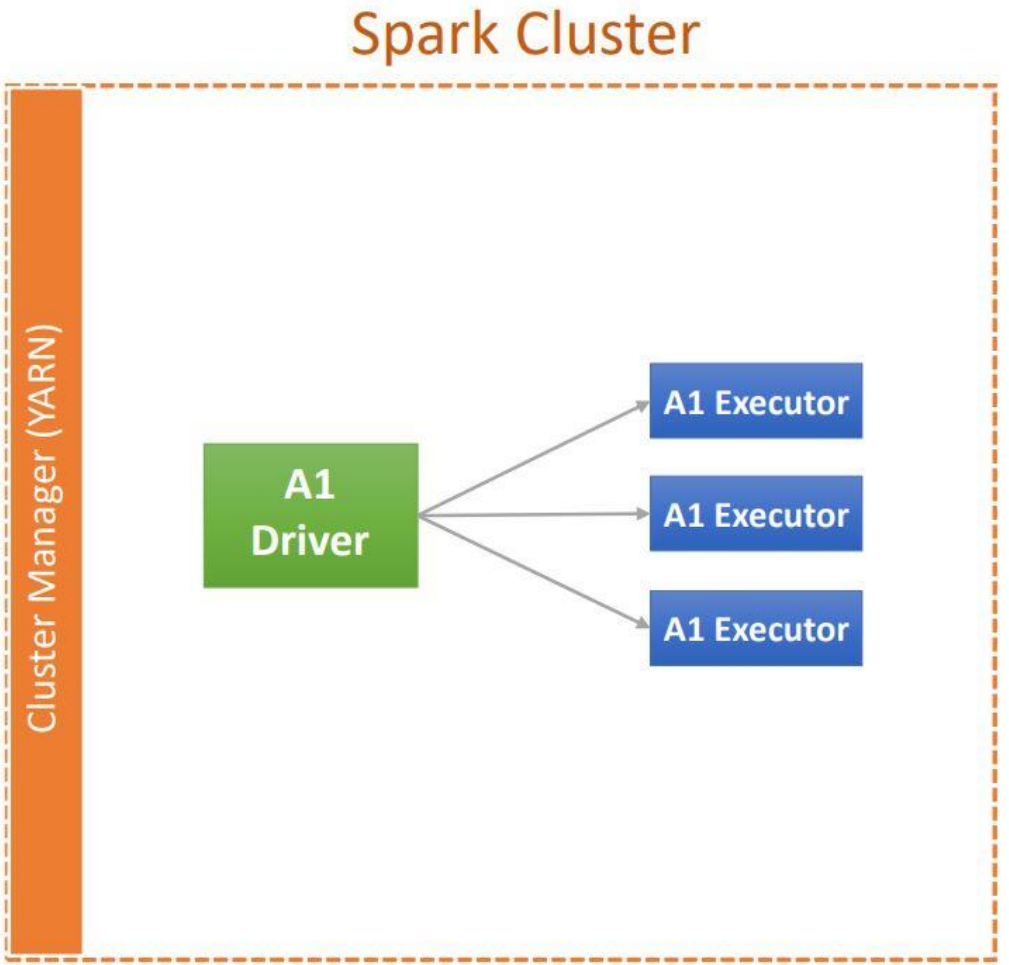


Summary :


Spark Cluster
Client Machine
Driver
Slave
Slave
Slave
Cluster Manager (YARN)

Summary :


Spark Client


Spark Cluster
Cluster Manager (YARN)
A1 Driver
A1 Executor
A1 Executor
A1 Executor