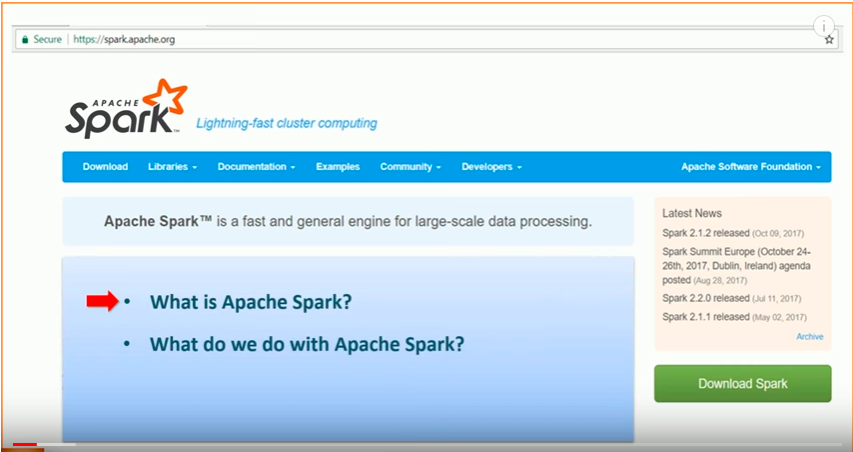
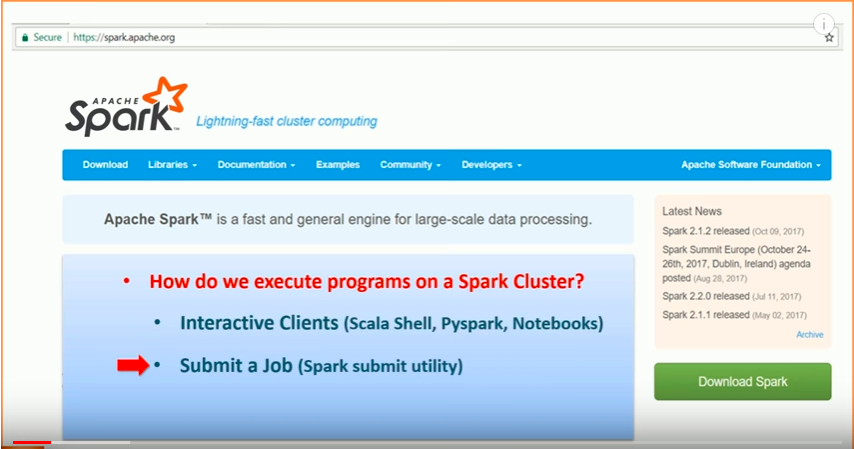
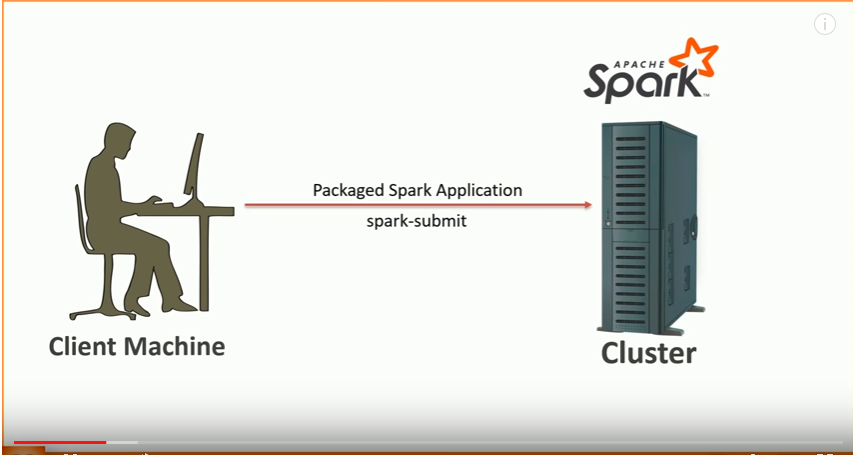
# Spark Architecture Part-1

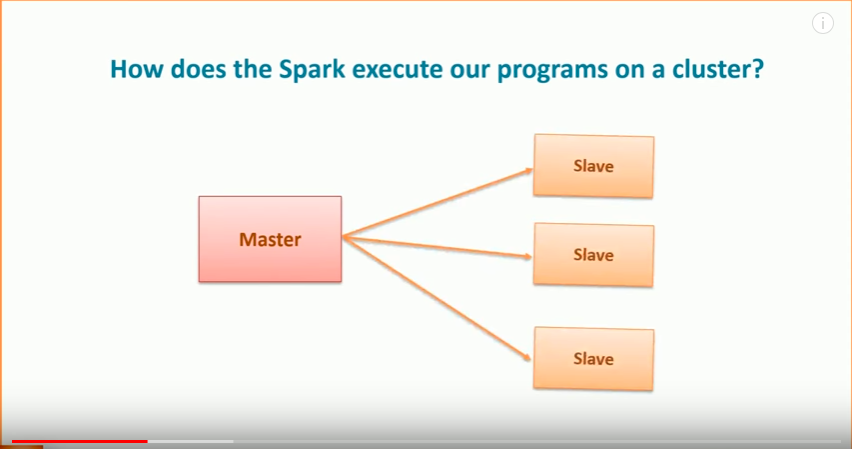


1. What is Apache Spark?
   * + A distributed computing platform.
2. What do we do with Apache Spark?
   * + We create programs and execute them on a Spark Cluster.

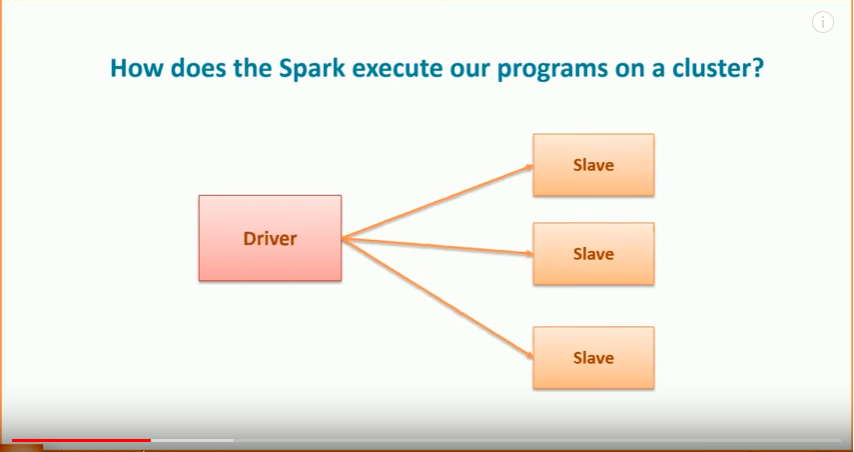


* We use interactive clients for exploration and spark-submit for executing a production application.

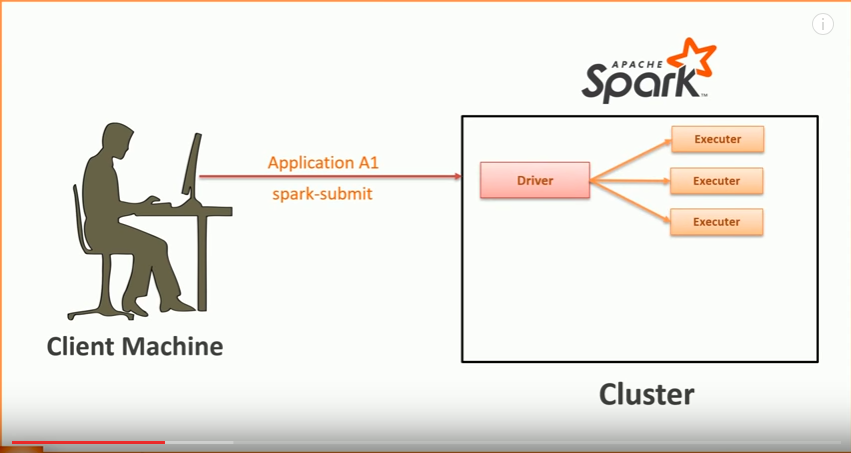




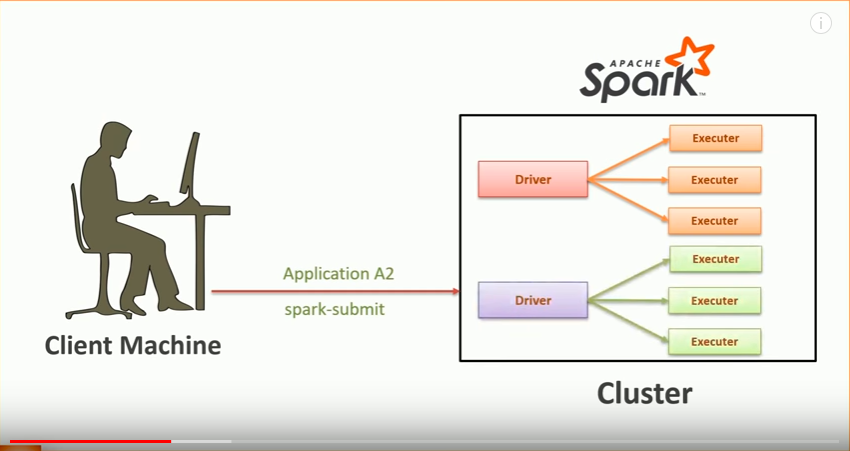
Spark is a distributed processing engine, and it follows the master-slave architecture. So, for every Spark App, it will create one master process and multiple slave processes. In Spark terminology, the master is the driver, and the slaves are the executors.



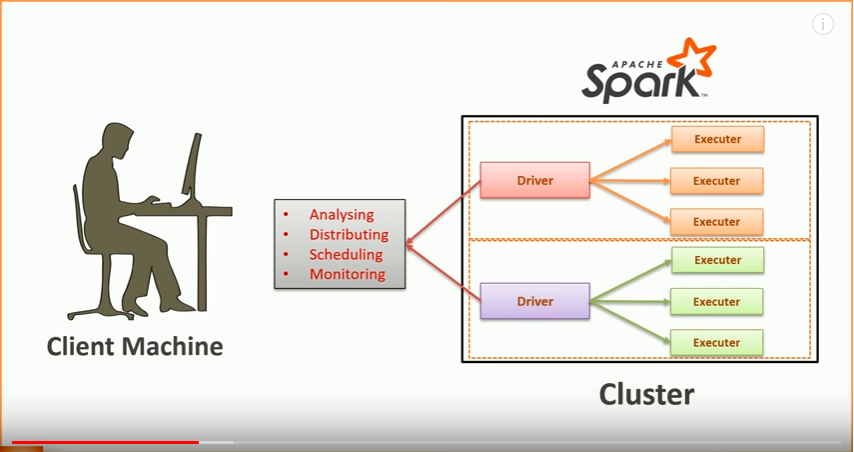
Suppose you are using the Spark Submit utility. You execute an application 'A1' using Spark Submit, and Spark will create one driver process and some executor processes for A1. The entire set of driver and executers is exclusive for the application A1.



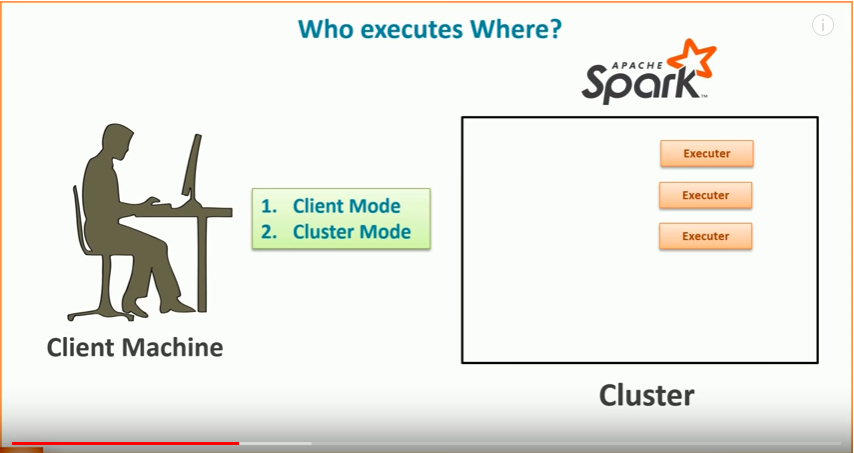
Now, you submit another application A2, and Spark will create one more driver process and some executor process for A2.



So, for every application, Spark creates one driver and a bunch of executors. Since the driver is the master, it is responsible for analyzing, distributing, scheduling and monitoring work across the executors. The driver is also responsible for maintaining all the necessary information during the lifetime of the application.



Now the executors, they are only responsible for executing the code assigned to them by the driver and reporting the status back to the driver.

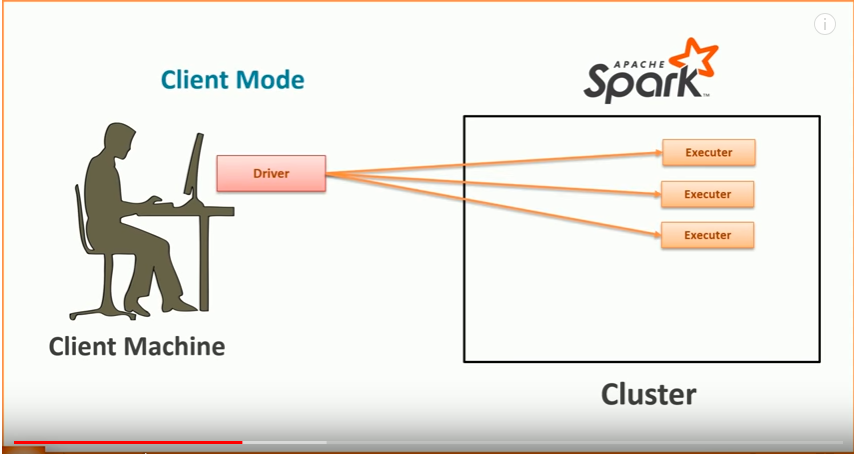


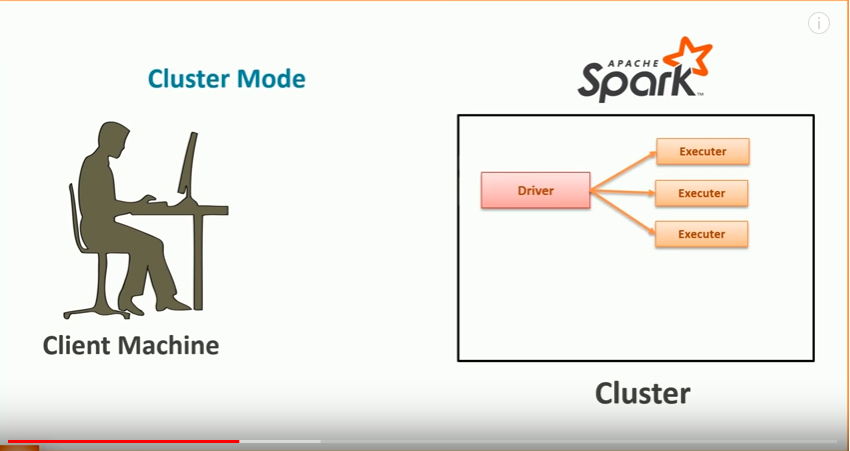
Who executes where?

When you start an application, you have a choice to specify the execution mode, and there are two options.

1. Client Mode - Start the driver on your local machine
2. Cluster Mode - Start the driver on the cluster.

The Client Mode will start the driver on your local machine, and the Cluster Mode will start the driver on the cluster.

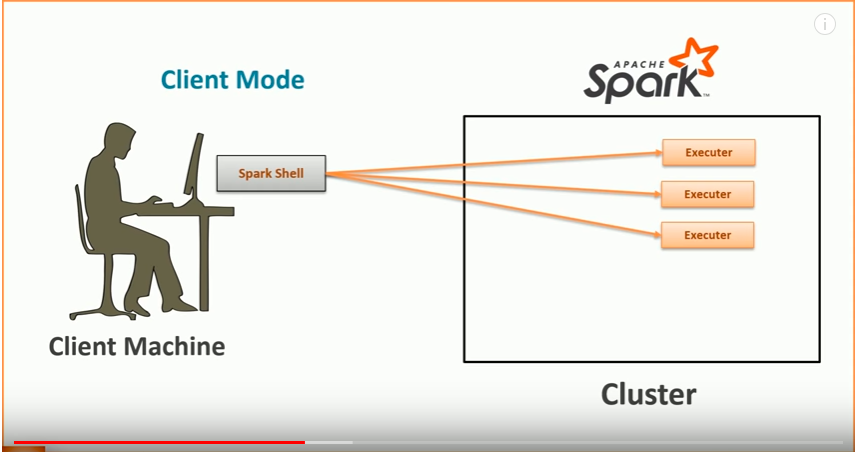


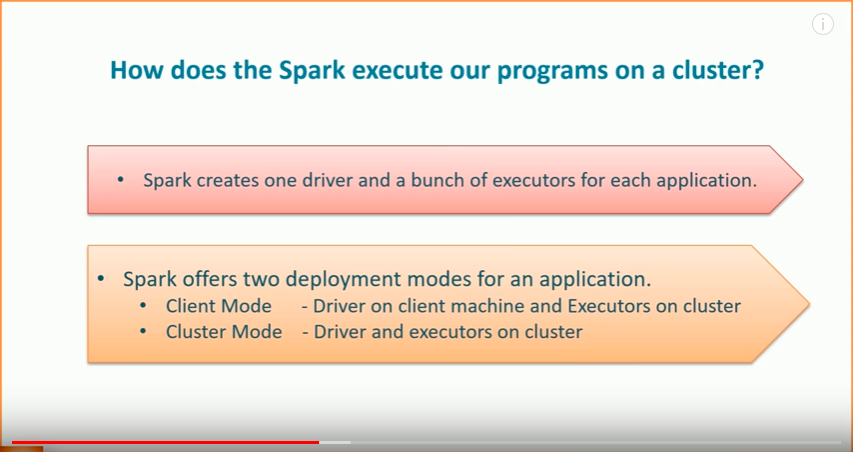


You already know that the driver is responsible for the whole application. If anything goes wrong with the driver, your application state is gone. So, if you start the driver on your local machine, your application is directly dependent on your local computer. You don't want that dependency in a production application. After all, you have a dedicated cluster to run the job. Right?   
Hence, the Cluster mode makes perfect sense for production deployment. Because after spark-submit, you can switch off your local computer and the application executes independently within the cluster.   
On the other side, when you are exploring things or debugging an application, you want the driver to be running locally. If the driver is running locally, you can easily debug it, or at least it can throw back the output on your terminal. Right?   
That's where the client-mode makes more sense over the cluster-mode. And hence, when you start a Spark shell or any other interactive client. You would be using a client mode.   
So, if you are running a Spark shell, your driver is running locally within the same JVM process. You won't find a separate driver process. It's only the Spark shell, and the driver is embedded within the shell.

How does the Spark execute our programs on a cluster?

You learned the answer. Spark will create one driver and a bunch of executors. If you are using an interactive client, your client tool itself is a driver, and you will have some executors on the cluster. If you are using spark-submit in cluster mode, Spark will start your driver and executors on the Cluster.





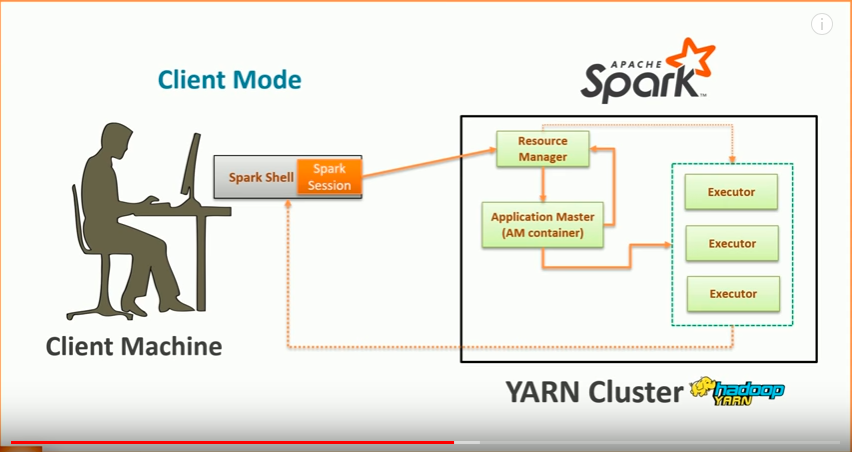
## Who controls the cluster?

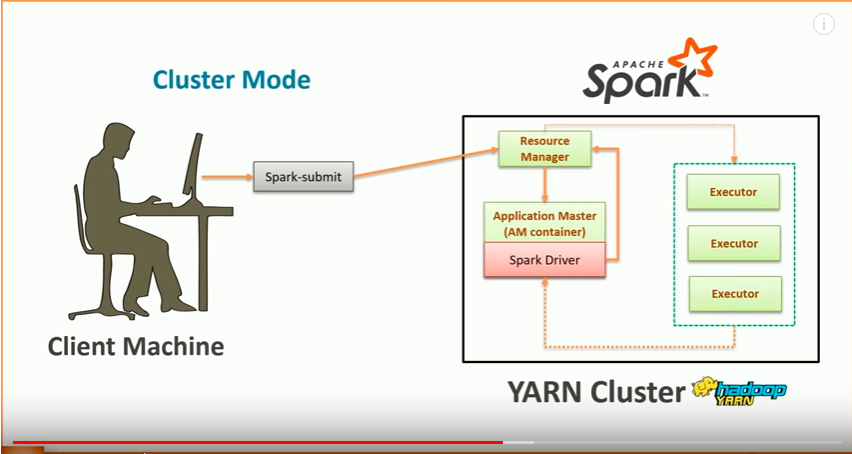
How Spark gets the resources for the driver and the executors?   
That's where we need a cluster manager.   
As on the date, Apache Spark supports four different cluster managers.

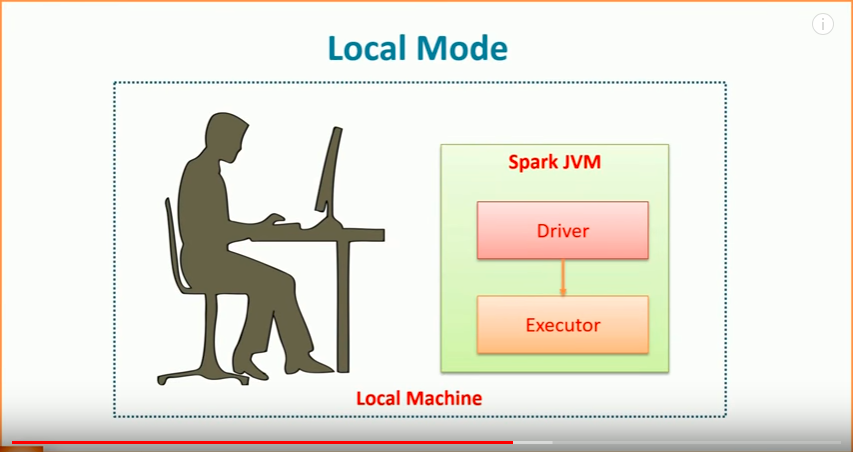
1. Apache YARN
2. Apache Mesos
3. Kubernetes
4. Standalone

YARN is the cluster manager for Hadoop. As of date, YARN is the most widely used cluster manager for Apache Spark.

No matter which cluster manager do we use, primarily, all of them delivers the same purpose.

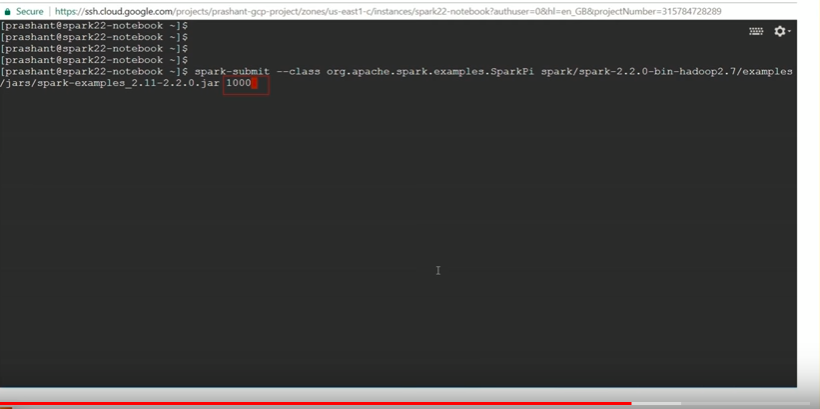


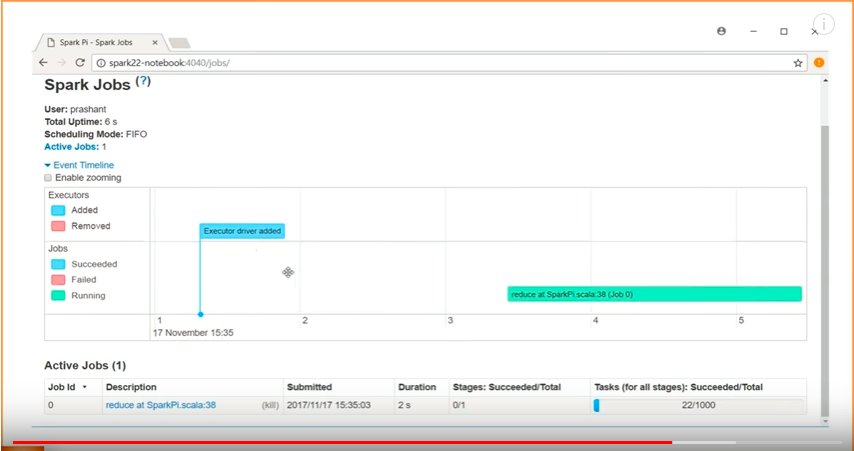




**Demo**

* Execution in Local Mode:

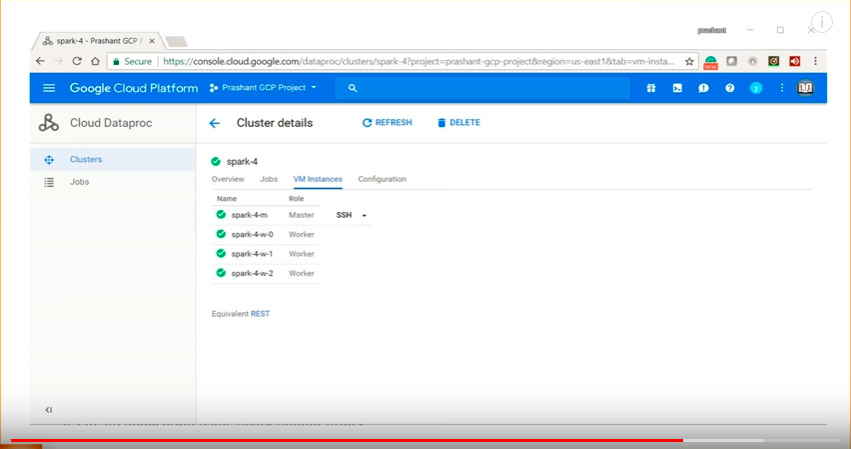




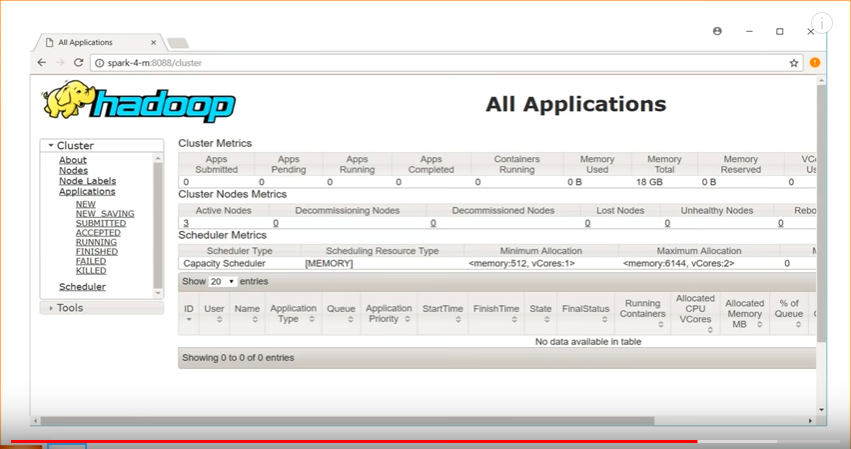
In the Spark UI above, we can see only one executor driver. Because, you can’t expect anything more from a local mode.

Execution in Client and Cluster Mode:

We have setup a four nodes Hadoop and Spark cluster like below. 1 master node and 3 worker nodes. Each worker machine has got 2 CPU cores.

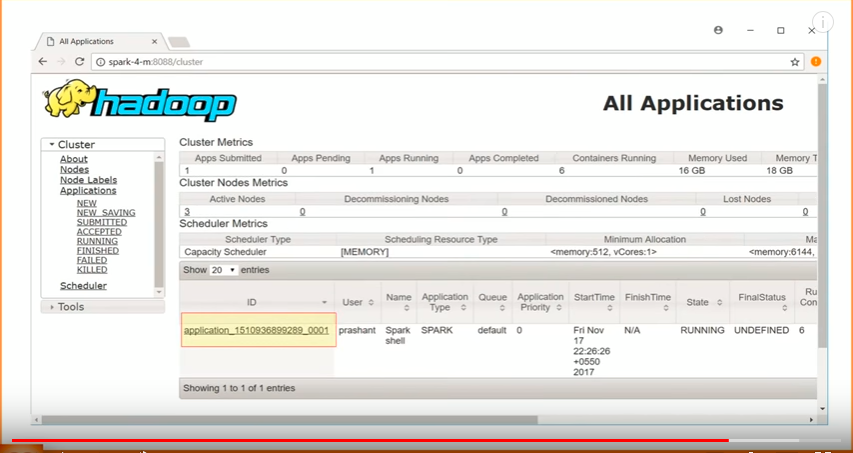


Below is the YARN resource manager UI and there is no application currently running on this cluster.

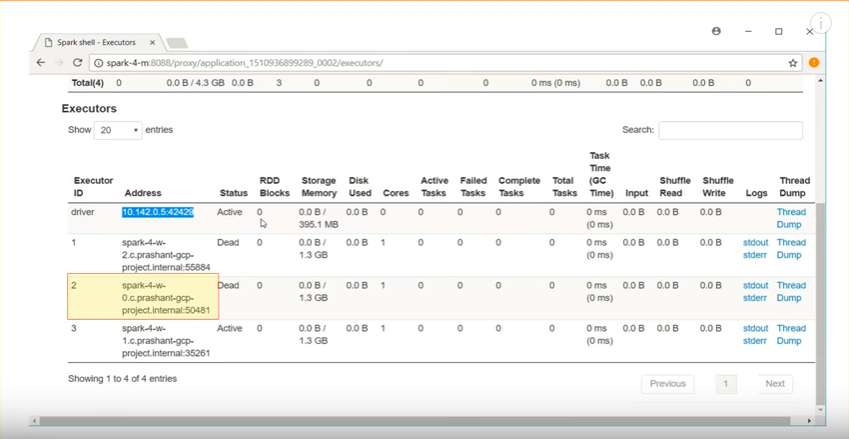


Next we will execute a spark application first in Client mode and then in Cluster mode with the help of 3 executors.

Client Mode YARN Resource Manager UI:

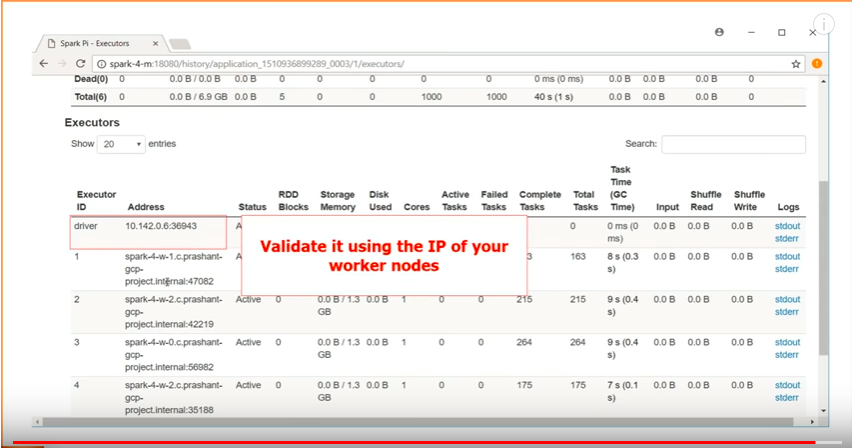


* 1 application is running.
* Go to the tracking UI link at the extreme right and click on the ApplicationMaster link. You will see how many executors are running.



* See in client mode, the driver is running on the local machine which is 10.142.0.5…. and all the 3 executors are running on the cluster.

Cluster Mode YARN Resource Manager UI:



* See in cluster mode, the driver and all the 3 executors are running on the cluster.