**Standalone cluster manager:-**

**Standalone mode** is a simple cluster manager incorporated with Spark. It makes it easy to setup a cluster that Spark itself manages and can run onLinux, Windows, or Mac OSX. Often it is the simplest way to run Spark application in a clustered environment.

**Working process:**

It has masters and number of workers with configured amount of memory and CPU cores. In Spark standalone cluster mode, Spark allocates resources based on the core. By default, an application will grab all the cores in the cluster.

In standalone cluster manager, Zookeeper quorum recovers the master using standby master. Using the file system, we can achieve the manual recovery of the master.

Advantages:-

1. The Spark standalone mode sets the system without any existing cluster management software.

2. We have spark master and spark worker who divides driver and executors for Spark application in Standalone mode.

Disadvantages:

1. Limited in features.
2. Expensive.

**Local :-**

The easiest way to try out Apache Spark from Python on Faculty is in local mode. The entire processing is done on a single server. You thus still benefit from parallelisation across all the cores in your server, but not across several servers.

Advantages:-

Think of local mode as executing a program on your laptop using single JVM. It can be java, scala or python program where you have defined & used spark context object, imported spark libraries and processed data residing in your system.

**Spark Client Mode**

The behaviour of spark job depends on the “driver” component. So here, ”driver” component of spark job will run on the machine from which job is submitted. Hence, this spark mode is basically “client mode”.

When job submitting machine is within or near to “spark infrastructure”. Since there is no high network latency of data movement for final result generation between “spark infrastructure” and “driver”, then, this mode works very fine.

When job submitting machine is very remote to “spark infrastructure”, also have high network latency. Hence, in that case, this spark mode does not work in a good manner.

**Spark Cluster Mode**

Similarly, here “driver” component of spark job will not run on the local machine from which job is submitted. Hence, this spark mode is basically “cluster mode”. In addition, here spark job will launch “driver” component inside the cluster.

When job submitting machine is remote from “spark infrastructure”. Since, within “spark infrastructure”, “driver” component will be running. Thus, it reduces data movement between job submitting machine and “spark infrastructure”. In such case, This mode works totally fine.

While we work with this spark mode, the chance of network disconnection between “driver” and “spark infrastructure” reduces. Since they reside in the same infrastructure. Also, reduces the chance of job failure.