The cluster module in Node.js allows us to create child processes (workers) that share server ports. It is particularly useful for creating scalable network applications. When you fork a Node.js process, it creates a new instance of the V8 engine. This allows us to take advantage of multi-core(4 core, 8 core, etc) systems and improve the performance of your application by distributing the workload across multiple processes.

Here are some key points about the cluster module:

1.Load Balancing: One of the primary purposes of the cluster module is to enable load balancing across multiple CPU cores. Each CPU core can run its own instance of a Node.js process, which helps distribute incoming connections and requests across the available cores.

2.Master-Worker Architecture: The cluster module follows a master-worker architecture. The master process manages the worker processes and distributes incoming connections among them. If a worker process dies, the master process automatically spawns a new worker to replace it.

3.Communication: The master process and worker processes can communicate with each other using inter-process communication (IPC). This allows them to share information such as server ports and other configuration details.

All os excluding windows follows round robin algorithm to assign a CPU

Cluster.isPrimary=Cluster.isMaster

Youtube video- Piyush Garg