

---

## CLIENT

---



Sometimes, if you're not used to dealing with Hadoop, and its communications (Originated in IT world), You may not exactly know some terms like master, client and slave. The "Master" device issues a command and the "Slave" responds, similar to a "Client" requesting information from the "Server".

In simple words, Master is a device that is actively pushing/polling for data from one or multiple devices using a serial communication protocol, while Slave is the counterpart of the Master.

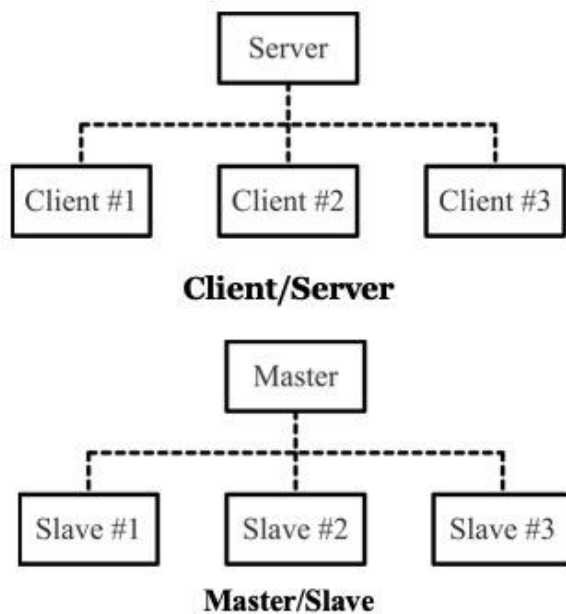
While, client is a device that is actively polling data from one or Multiple Devices.

Or in the simplest way of Understanding-Think like the Client is a Customer, who gives order to the (workers) slave of the Restaurant and the Master is the waiter for that Customer in Restaurant.

The “Master” device issues a command and the “slave” responds, which is similar to a client requesting information from Server.

While the last para is not entirely incorrect, it assumes that Master=client and Slave=Server. This assumption not only ignores the differences in functionality between a Master and Client and also between Slave and Server, but it also neglects the impact of data traffic pattern.

In general following image below demonstrates, A server serves multiple Clients, with Multiple slaves served by only one master.



In Computer Networking, Client/server describes the relation between in which One program (Client) makes a service request from Another program, the server, which fulfils the request.

A client configuration can access the data but not modify data since there will be unpredictable results if multiple clients modify the data in unsynchronized way.

A Master device is in charge of all other devices and can significantly

influence a Slave's data and even its behavior. Once the master/slave relationship is established, the direction of control is always from the master to the slave(s).

Now, these all definitions are quite childish in front of Real Practical life, as the way it looks that yes, it's client and can be easy to manage but it's not.

For a Client establishment we need:-

- Master public IP address
- Hadoop installed Linux operating system

That's it, we are ready for client.

- ✚ At first edit the core file and set the master IP along with port number in Values so that it is linked to the required cluster.
- ✚ Then go to the hdfs file and if you want to make it more secure make more replica's of that and even you can change the block size in next property box.  
What's Replica here??

Replication is the process of making a replica (a copy) of something.

But again question comes, copy for what here. We are just uploading or putting data from clients that's it.

But my cross question is what is the uploaded data in the datanode, gets disconnected or failed any how then.??

Hadoop itself makes 3 copies of the same data as a default to use as recovery file and this term called Replication.

## Contents of directory /

Goto :  go

Name	Type	Size	Replication	Block Size
<a href="#">abhi.txt</a>	file	3.85 KB	3	64 MB
<a href="#">akshay_anil.txt</a>	file	0.04 KB	3	64 MB
<a href="#">mcq.txt</a>	file	404.6 KB	3	64 MB
<a href="#">shreya_shrk.txt</a>	file	13.69 KB	3	64 MB
<a href="#">successfull.txt</a>	file	0.38 KB	3	64 MB

It is necessary for the data to be consistent and more durable.

Okay now what is block size here?? Again another question.

The main logic to overcome BIGDATA problem of VELOCITY and VOLUME can be solved by slicing the data into multiple parts and putting them into different datanodes can save time for uploading the data and also volume by dividing into parts.

So, block size here tells us about the slicing of data into a fixed size of data part. A picture below will show the block sizes of files.

✚ We are ready to put any data in the server as we are successfully established the client. Just by put command we can upload data in the cluster. For conformation, we had 3 datanodes and we can check those reports from clients also. Attached a Picture of conformation of client formation with total report of datanodes.

```
[root@ip-172-31-12-222 ~]# hadoop dfsadmin -report
Configured Capacity: 32174444544 (29.96 GB)
Present Capacity: 26242957312 (24.44 GB)
DFS Remaining: 26241507328 (24.44 GB)
DFS Used: 1449984 (1.38 MB)
DFS Used%: 0.01%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
```

```
-----
Datanodes available: 3 (3 total, 0 dead)
```

```
Name: 54.243.3.85:50010
Decommission Status : Normal
Configured Capacity: 10724814848 (9.99 GB)
DFS Used: 483328 (472 KB)
Non DFS Used: 2078064640 (1.94 GB)
DFS Remaining: 8646266880 (8.05 GB)
DFS Used%: 0%
DFS Remaining%: 80.62%
Last contact: Sat Oct 03 19:34:36 UTC 2020
```

```
Name: 35.173.131.146:50010
Decommission Status : Normal
Configured Capacity: 10724814848 (9.99 GB)
DFS Used: 483328 (472 KB)
Non DFS Used: 1932374016 (1.8 GB)
DFS Remaining: 8791957504 (8.19 GB)
DFS Used%: 0%
DFS Remaining%: 81.98%
Last contact: Sat Oct 03 19:34:36 UTC 2020
```

```
Name: 13.233.111.93:50010
Decommission Status : Normal
Configured Capacity: 10724814848 (9.99 GB)
DFS Used: 483328 (472 KB)
Non DFS Used: 1921048576 (1.79 GB)
DFS Remaining: 8803282944 (8.2 GB)
DFS Used%: 0%
DFS Remaining%: 82.08%
Last contact: Sat Oct 03 19:34:38 UTC 2020
```