

## Cap Theorem

**CAP** Theorem states that in a distributed database system can only comply with 2 of the 3: Consistency, availability and partition tolerance.

### Partition Tolerance

In this condition the system continues to run, despite the number of messages being delayed by the network between nodes. A system that is partition-tolerant can sustain any amount of network failure that doesn't result in a failure of the entire network. Data records are sufficiently replicated across combinations of nodes and networks to keep the system up through intermittent outages. When dealing with modern distributed systems, Partition Tolerance is not an option. It's a necessity. Therefore, we need to adjust between Consistency and Availability.

### Consistency

This condition states that all nodes see the same data at the same time. Simply put, performing a *read* operation will return the value of the most recent *write* operation causing all nodes to return the same data. A system has consistency if a transaction starts with the system in a consistent state, and ends with the system in a consistent state.

### Availability

This condition states that every request gets a response on success/failure. Achieving availability in a distributed system requires that the system remains operational 100% of the time. Every client gets a response, regardless of the state of any individual node in the system. This metric is trivial to measure: either you can submit read/write commands, or you cannot. Hence, the databases are time independent as the nodes need to be available online at all times. high availability isn't feasible when analyzing streaming data at high frequency.