

# CAP Theorem

In this lesson, we will learn about the CAP theorem.

## We'll cover the following

- What Is CAP Theorem?

## What Is CAP Theorem? #

*CAP* stands for **Consistency, Availability, Partition Tolerance**. We've gone through consistency and availability in great detail. *Partition Tolerance* means *Fault Tolerance*. The system is tolerant of failures or partitions. It keeps working even if a few nodes go down.

There are many definitions of the theorem, you'll find online, which state that amongst the three, *Consistency, Availability & the Partition Tolerance*, we have to pick two. I find that a teeny tiny bit of confusing. I will try to give a simpler explanation of the theorem.

*CAP theorem* simply states that in case of a network failure, when a few of the nodes of the system are down, we have to make a choice between *Availability & Consistency*

If we pick *Availability* that means when a few nodes go down, the other nodes are available to the users for making updates. In this situation, the system is inconsistent as the nodes which are down don't get updated with the new data. At the point in time when they come back online, if a user fetches the data from them, they'll return the old values they had when they went down.

If we pick *Consistency*, in that scenario, we have to lock down all the nodes for further writes until the nodes which have gone down come back online. This would ensure the *Strong consistency* of the system as all the nodes will have the same entity values.

Picking between *Availability* and *Consistency* largely depends on our use case and the business requirements. We have been through this in great detail. Also, the limitation of picking one out of the two is due to the design of the distributed systems. We can't have both *Availability* and *Consistency* at the same time.

Nodes, spread around the globe, will take some time to reach a consensus. It's impossible to have zero-latency unless we transit data faster than or at the speed of time.