NAME:SIDDARTH SAKTHI M

REG.NO:230701314

ASSIGNMENT3

The CAP Theorem, also known as Brewer's theorem, is a fundamental principle in distributed systems that describes the trade-offs between three key properties: Consistency, Availability, and Partition Tolerance. The theorem states that in the presence of a network partition, a distributed system can only guarantee two of the three properties at the same time. Let's break down each of these concepts:

1. Consistency

- **Definition**: Every read receives the most recent write or an error. In other words, all nodes in the distributed system see the same data at the same time.
- **Implication**: If a system is consistent, it guarantees that after a write operation is completed, any subsequent read operation will reflect that write. This is similar to the behavior of a single-node database.
- **Example**: If a user updates their account balance to $100, any subsequent read request from any node in the system should return $100.

2. Availability

- **Definition**: Every request (read or write) receives a response, regardless of whether the response contains the most recent data. This means that the system is operational and can respond to requests.
- **Implication**: An available system ensures that every request gets a response, even if that response is not the most up-to-date information. This is crucial for user experience, as users expect systems to be responsive.
- **Example**: If a user queries their account balance, they will receive a response, even if that balance is not the most current due to some recent updates that have not yet propagated through the system.

3. Partition Tolerance

- **Definition**: The system continues to operate despite arbitrary network partitions that prevent some nodes from communicating with others.
- **Implication**: In a distributed system, network failures can occur, leading to partitions where some nodes cannot communicate with others. A partition-tolerant system can still function, ensuring that nodes can continue to operate independently.
- **Example**: If a network failure occurs that isolates a subset of nodes from the rest of the system, the nodes in the isolated partition should still be able to process requests and operate without being completely dependent on the unreachable nodes.

The CAP Trade-off

The CAP theorem states that a distributed system can achieve at most two out of the three properties at any given time:

- **CP (Consistency and Partition Tolerance)**: Systems that prioritize consistency and partition tolerance may sacrifice availability. During network partitions, the system may refuse to process requests to ensure that all nodes remain consistent.
- **AP (Availability and Partition Tolerance)**: Systems that prioritize availability and partition tolerance may sacrifice consistency. In the event of a partition, the system will continue to process requests, but some nodes may return stale or inconsistent data.
- **CA (Consistency and Availability)**: It is impossible to achieve both consistency and availability in the presence of a network partition. Therefore, while a system can be consistent and available, it cannot maintain those properties when partitions occur.

Practical Implications

- **Design Choices**: When designing distributed systems, engineers must make conscious decisions about which properties to prioritize based on the specific requirements of their applications. For example, a banking application may prioritize consistency, while a social media application may prioritize availability.
- **Trade-offs**: Understanding the CAP theorem helps developers recognize the trade-offs they face when building distributed systems, leading to better architectural decisions and expectations about system behavior.