# 2 Phase Commit (2PC)

* + The Two-Phase Commit protocol (2PC) is a widely used pattern to implement **distributed transactions.** We can use this pattern in a Microservices architecture to implement distributed transactions.
  + In a two-phase commit protocol, there is a **coordinator component** that is responsible for controlling the transaction and contains the logic to manage the transaction.
  + The other component is the **participating nodes** (e.g., the Microservices) that run their local transactions.
  + As the name indicates, the two-phase commit protocol runs a distributed transaction in two  
     phases:
    - **Prepare Phase** – The **coordinator asks the participating nodes** whether they are ready to commit the transaction. The participants returned with a yes or no.
    - **Commit Phase** – If all the participating nodes respond affirmatively in phase 1, the **coordinator asks** all of them to commit. If at least one node returns negative, the **coordinator asks** all participants to rollback.
  + The two-phase commit protocol is **slow by design due to the chattiness and dependency on the coordinator**. So, it can lead to **scalability and performance issues in a Microservices-based architecture** involving multiple services.
  + Two-phase commit protocol is **not supported in NoSQL databases**. Therefore, in a microservice architecture where one or more services use NoSQL databases, we can’t apply a two-phase commit.
  + Let’s take an example of Ecommerce where Order & Payment are two micro services which handle by Coordinator microservice to manage Prepare & Commit / Rollback operation.

