1. Text

   Description automatically generated  
   
2. In the earlier section, we created a very simple producer to send 1M short msg to Kafka Server.
3. But kafka is all about scalability.
4. **Agenda**:
   1. We will discuss about scaling up the Producer.
5. Let’s get started.
6. 
7. Apache Kafka was designed with scalability in mind.
8. Scaling a Kafka App is straightforward.
9. If we consider the POS (Point-of-Sale Counters) example, then each POS system can create a KafkaProducer and send the invoices.  
   A picture containing diagram

   Description automatically generated  
   At the Kafka Cluster, it is the Kafka Broker that receives the invoice and acknowledges the successful receipt of the msg.
10. Chart

    Description automatically generated with medium confidence
11. A single Kafka Broker may handle hundreds of msgs or thousands of msgs per second.  
    However, you can increase the Kafka Brokers to receive and acknowledge hundreds of thousands of msgs.
12. On the Producer side, you keep on adding producers to send msgs in parallel to Kafka Cluster.  
    A picture containing chart

    Description automatically generated  
    This arrangement provides Linear Scalability by merely adding more Producers and Brokers.  
    Jatin 🡺 Chart, line chart

    Description automatically generated  
    This approach works perfectly for scaling up your overall streaming bandwidth.
13. However, you also have an opportunity to scale an individual producer using multithreading technique.  
    A picture containing chart

    Description automatically generated  
    Single producer thread is good enough to support the use cases where the data is being produced at a reasonable pace.  
    However, some scenarios may require parallelism at the individual producer level as well.  
    We can handle such requirements using multithreaded Kafka Producer.  
    Multithreaded may not apply to apps that don’t frequently generate new msgs.  
    **For Example**, every POS system may be generating invoices every 2-3 minutes.  
    In that case, single threaded producer more than enough to send invoices to the Kafka Cluster.  
    However, if you have an app that generates or receives data at high speed and wants to send as quickly as possible, then you might want to implement a multithreaded app.