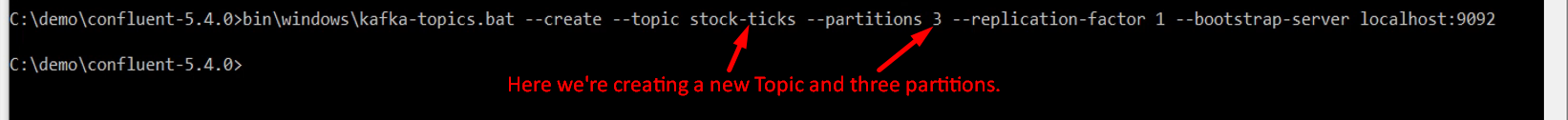
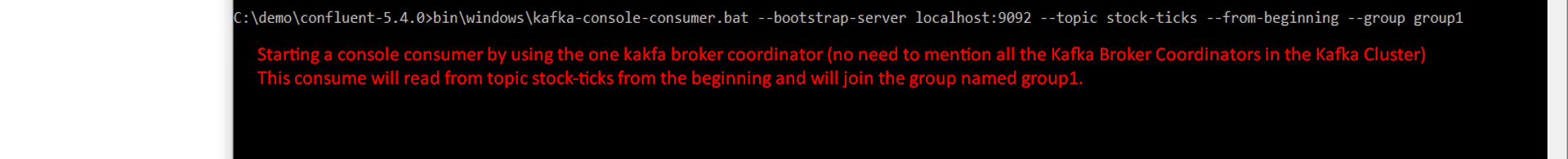
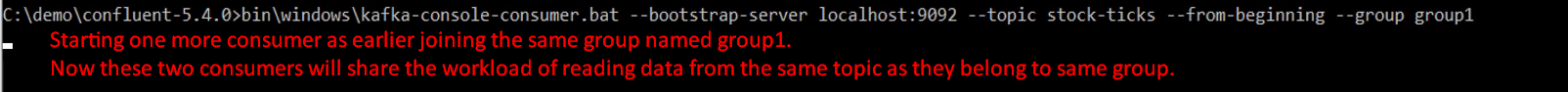
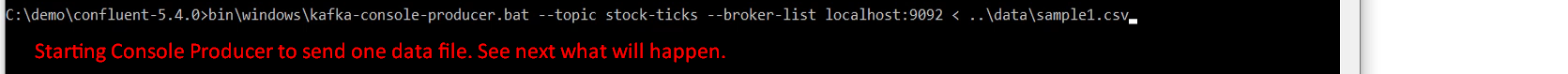
1. Graphical user interface, text, application

   Description automatically generated
2. **Agenda**:
   1. In the earlier lecture, we created 3-Node Kafka Cluster and in this lecture, we’re going to use it and will do the following things.
3. **We will do the followings**:  
   Graphical user interface, diagram

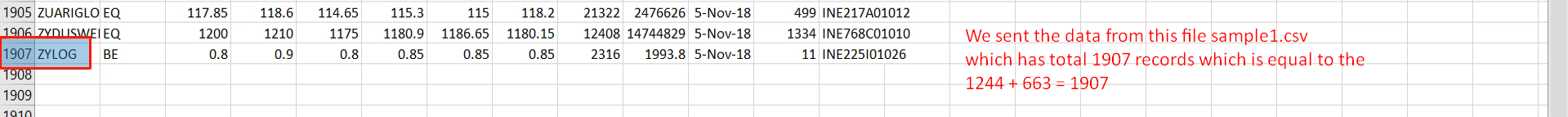
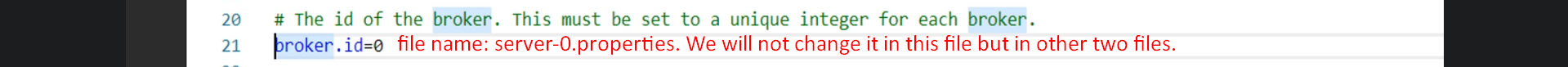
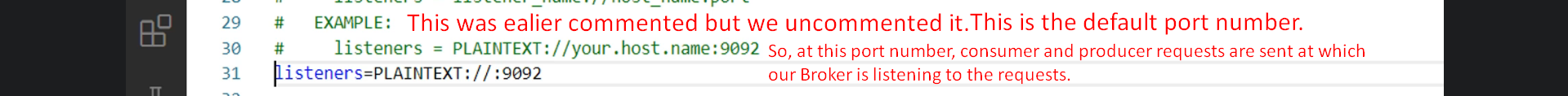
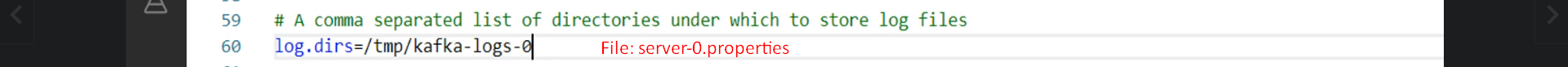
   Description automatically generated
   1. We’re going to create a new Kafka Topic with 3 partitions.
   2. Then we will start 2 consumers in the same group.  
      So, both of these consumers should be reading data from the same topic.  
      However, since they are running in the same group, they should share the workload.  
      We will see this happening.
   3. We will start the Producer to send the file data to the Kafka Cluster.
   4. We will come back to the Consumer and observe the outcomes.



1. Make sure you have 3-Node Kafka Cluster in running state.
2. 
3. 
4. 
5. Graphical user interface, application

   Description automatically generated
6. 
   1. The data will go to the topic.
   2. As the topic is partitioned in 3 parts, so all the data will be distributed among them.
   3. Now we have 3 partitions and 2 consumers.   
      So, one of these two consumers is going to read data from two of 3 partitions.  
      However, all the data will be processed only once.
   4. This is what we will be.  
      A picture containing diagram

      Description automatically generated
7. When we run the Kafka Console Producer, the two consumers start reading and processing the data. See below snapshot.Text

   Description automatically generated
8. But how do we know how all this happened. Let me show you.
9. 
10. Now let’s go the log directory.
    1. Let’s refresh our memory before going further.  
       We had created **server-0.properties** for a Kafka Broker with the following configuration.  
         
         
         
       The last snapshot is important which defines Data Dir location for broker with id=0 which we will use next.
    2. Let’s go further.  
       Graphical user interface, text, application

       Description automatically generated
    3. A picture containing chart

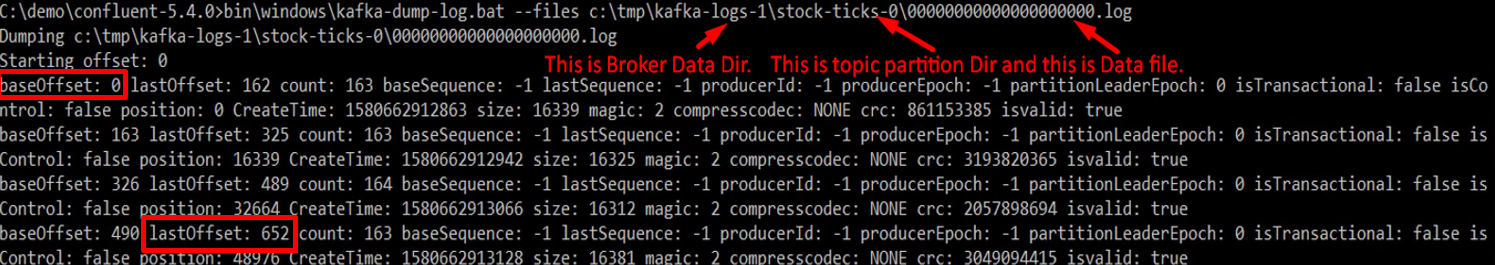
       Description automatically generatedBut what is the suffix 1 at the end of the topic as topic name was “stock-ticks” not “stock-ticks-1”.  
       This suffix 1 is the partition number.  
       We had created three partitions for topic “stock-ticks”, so Kafka Cluster will create three Topic Dirs like this  
       stock-ticks-0, stock-ticks-1, stock-ticks-2 where 0,1,2 are suffix.
    4. Let’s summarize to understand everything clearly.
    5. We create three brokers and each broker has its own Data Dir like “kafka-logs-0” that we defined in the property file which is passed to Kafka Broker when starting it.
    6. The we created a topic “Stock-ticks” with three partitions which will be named like <topic-name>-<partition#> starting from 0.
    7. Graphical user interface

       Description automatically generated
    8. Graphical user interface, application

       Description automatically generated
    9. Graphical user interface, text, application

       Description automatically generated
    10. let’s dump the file.  
        Graphical user interface, text, application

        Description automatically generated
    11. Let’s use the tool **kafka-dump-log.bat**
    12. A picture containing text

        Description automatically generatedYou can dump the data itself too. But we’re skipping.
    13. Let’s dump the other partitions’ log.  
        **Dumping partition data from broker with broker.id=1** This partition has got 653 records. In the above snapshot, you can see.  
         **Dumping partition data from broker with broker.id=2**A picture containing text

        Description automatically generated**Let’s tag these statistics for clarity.**Graphical user interface, application

        Description automatically generated
11. So, what we did in this lecture.
    1. **Kafka cluster stores data in topic partitions.**
    2. **Each partition is managed by a separate broker as a storage dir.**
    3. **The actual data sits in the log file in that storage dir.**
    4. **Consumers can work in consumer group to share the workload when reading from a topic.**
    5. **The purpose of this lecture was to give you insight of how partitions & consumer group are used.**