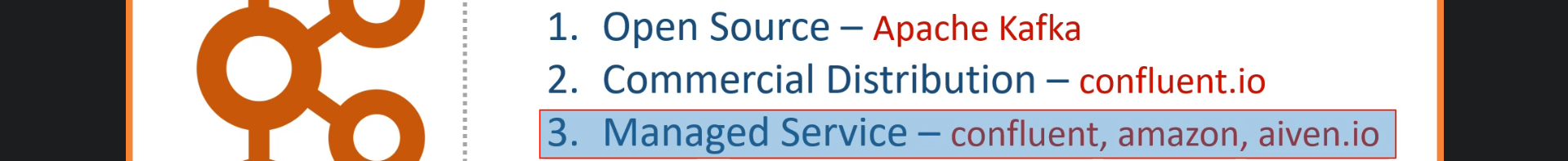
1. A picture containing graphical user interface

   Description automatically generated
2. **Agenda**:
   1. To set up a **single node Kafka** on our local machine.
3. But before that, let me add some contextual theory.
4. **Kafka** comes in **many flavors** and you can classify them into three categories.  
   
   1. **Open Source** – **Apache Kafka**You can download it from the Apache website, install it, use it, and manage it yourself.

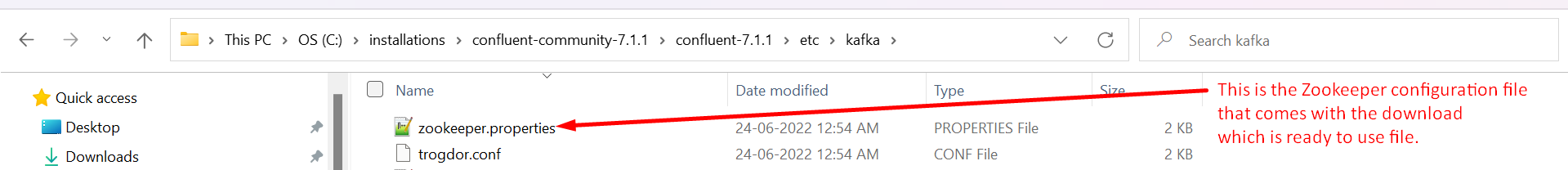
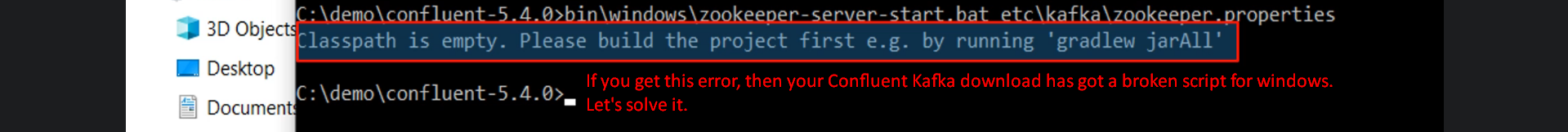
If you’re getting into some **operational issues** or **open bugs**, need to upgrade **Kafka Versions** or facing some infrastructure problems, you’re all alone and expected to develop in-house expertise to handle it.  
The point is straight. Your organization is unlikely to use **open source distribution** unless you have a bunch of **Kafka Admins** and **infrastructure experts** who understand in and out of the **Kafka Cluster**. That is hard to find.

* 1. **Commercial Distribution**: confluent.io
     1. It comes with a bunch of tools and utilities to manage your day-to-day operations and monitor your cluster.
     2. Your commercial vendor is going to deploy a bunch of well-trained and highly skilled support professionals for you.
     3. This option comes with a cost to your organization.
     4. However, you’re most likely to work on these kinds of setups.
     5. **Confluent Kafka** is one such **Commercial Distribution** & this is most popular one.  
        Confluent also offers a **community edition** without any cost & that is what we’re going to use in this course.  
        Why **Confluent Kafka**? Why not **Apache Kafka?**Because you’re going to see most likely **Confluent Kafka** in your **Production Environment.**
  2. **Managed Service – Confluent, Amazon, aiven.io**:
     1. This 3rd option is a fully **managed Kafka Service** in the **cloud**.
     2. In this option, you don’t need to download, install, run, operate, or maintain anything related to the **Kafka Cluster**. Just use the **cluster** for producing and consuming data.  
        All the infrastructure headache is taken care of by the **Managed Service Provider.**Thisoption is the simplest way of using **Kafka** for your projects and comes with different payment plans.  
        **Confluent** is also a **Managed Service Provider**. There are others as well including **Amazon, aiven.io.**

1. However, all these options are to get a **Kafka Cluster.**You still need to develop apps for creating and processing **Data Streams.**
2. So, with that background set, let’s download **Confluent Kafka Community Edition** and set up on our local machine.
3. 
   1. Graphical user interface, funnel chart

      Description automatically generated
   2. Then unzip.
   3. Graphical user interface, text, application, email

      Description automatically generatedYou need **JVM** installed on your system.
   4. **Kafka** comes with a bunch of **command line tools** and you can find them under **bin dir**.
   5. Graphical user interface, text

      Description automatically generated
   6. Well, we’re all set to do the real thing.
4. How to start a **Kafka Cluster**?
   1. Starting a **Kafka Cluster** is a two-step process.
   2. **Step 01**: Start **Zookeeper Server**.  
      The command is 🡺   
        
      This command will also take one mandatory argument 🡺 **Zookeeper Configuration File**.  
      We will talk about configuration file in a later video.   
      But your download (**Kafka Cluster**) already comes a-ready-to-use **Zookeeper configuration file**   
      
   3. 
      1. Fixing the problem:  
         Graphical user interface, application, table, Excel

         Description automatically generated  
         **Copy & paste the following:**   
         **rem Classpath addition for LSB style path**

**if exist %BASE\_DIR%\share\java\kafka\\* (**

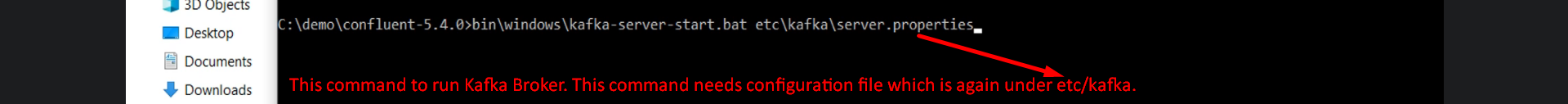
**call :concat %BASE\_DIR%\share\java\kafka\\***

**)**

Graphical user interface, text, application

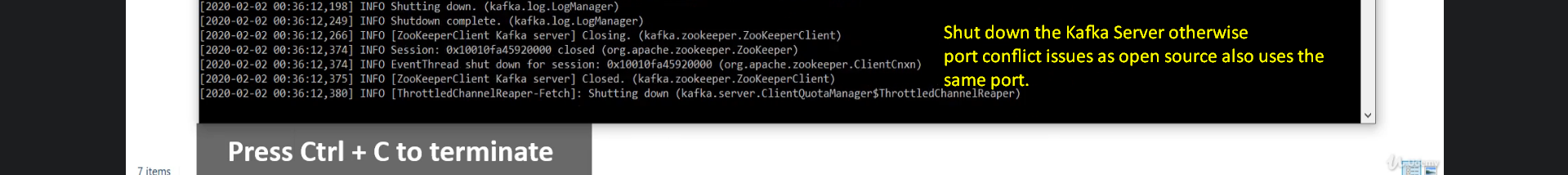
Description automatically generated  
Text

Description automatically generated with medium confidence  
Text

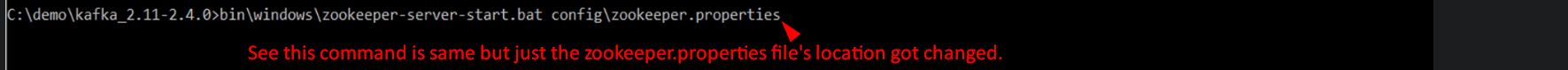
Description automatically generated with low confidence  
**Step 02**:  
Let’s run one **Kafka Broker**.   
Command to run **Kafka Broker**.  


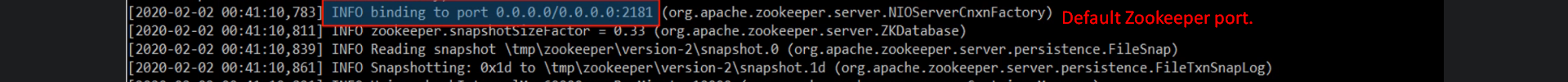
  
Now you’re ready to play with **Single Node Kafka Cluster**.

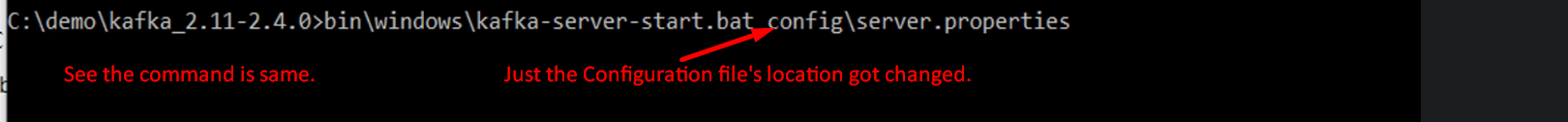
1. **Question: What is Zookeeper?**
2. Text

   Description automatically generated
   1. **Zookeeper** is a kind of DB where **Kafka Brokers** would store a bunch of shared information.
   2. **Zookeeper** is used as a **Shared System** among **Kafka Brokers** to coordinate **Kafka Brokers** for various reasons.
   3. **NOTE:** The **Kafka Community** hasalready that they are going to retire **Zookeeper** in the coming days.
   4. **Zookeeper** must be running even though we have a Single **Kafka Broker**.
3. So, we have seen the **Confluence Community Edition** but if you want to use **Open Source Edition**
   1. **Shutdown the Zookeeper and Kafka Server**.
   2. **Download the Kafka Open Source Binary**.  
      Graphical user interface, text, application

      Description automatically generated
   3. **Un-compress**.
   4. Let’ see the directory structure.   
      You will notice that some directories are missing from the open source version.   
      But nothing to worry. The open source version carries the same thing.   
      It is just the directory structure which is different and Confluent brings some additional capabilities that we understand as we progress through the course.  
      Now, we want to do two things
      1. **Step 01: Start Zookeeper**.



Console Output:   


* + 1. **Step 02**: Let’s run the **Kafka Broker** on a new **Command Windows**.  
       

1. In this lecture, we learnt the following:
   1. How to start a single node Kafka Broker using two types of Kafka Distributions.



* 1. Once, we have **Zookeeper** and **Kafka Broker** running, leave these windows open & you’re ready to use **Kafka Cluster** for all your hands-on activities.
  2. If you’re willing to explore the **Kafka Managed Services**, you can try accessing **Confluent Cloud & Aiven Kafka.**