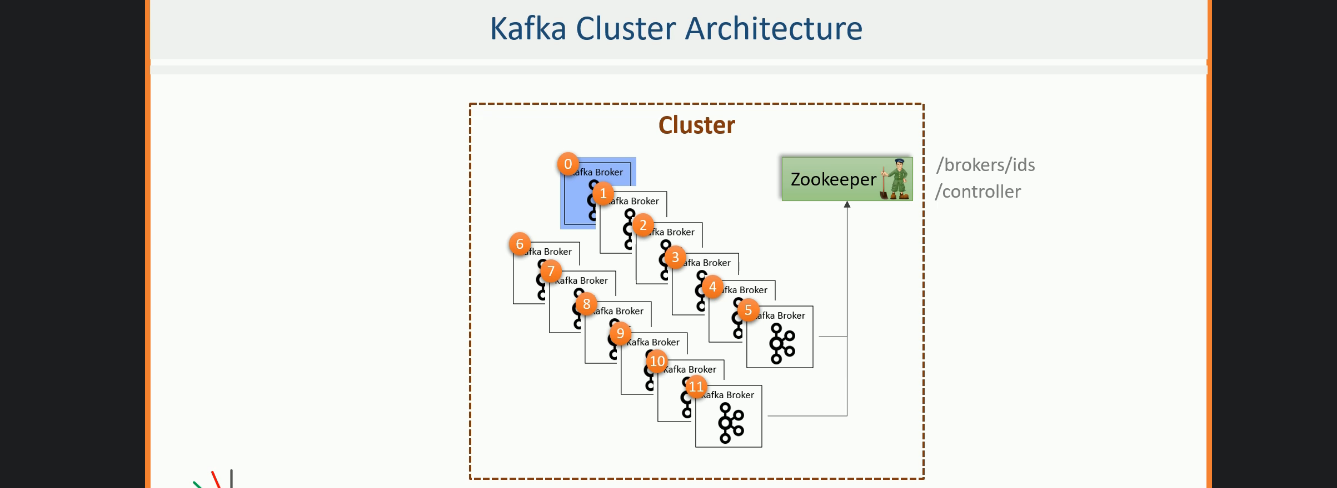
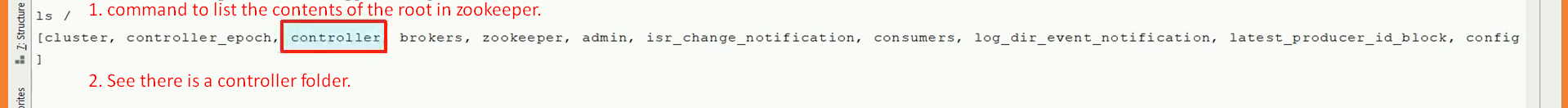
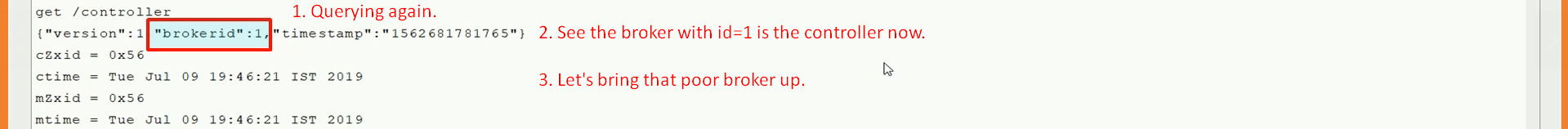
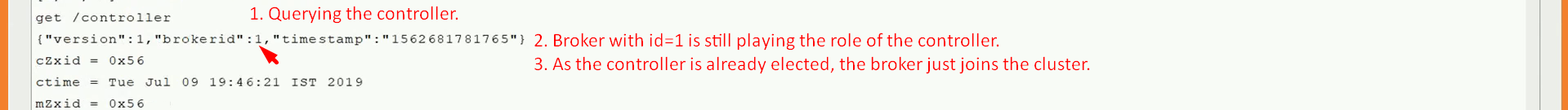
1. Graphical user interface, text, application

   Description automatically generated
2. Kafka is **masterless** cluster means it **doesn’t follow master-slave architecture** and the list of active brokers is maintained under zookeeper.
3. However, we need someone to perform routine **administrative activities** such as monitoring the active list of brokers and reassigning the work when an active broker leaves the cluster.
4. 
5.  All those administrative activities are performed by **a controller in the Kafka Cluster**.
6. The controller is **not a master**.  
   One of the brokers is elected as **controller** with some extra responsibilities.  
   That means that **controller** (Broker) also acts as a regular broker.  
   So, if you have a single node cluster, the same broker will play the role of **controller**.  
   However, at any point of time, there is at least one **controller** and one **broker**.
7. Controller is responsible for monitoring (not maintaining. Maintaining the active list is what we discussed in previous lecture) the list of active brokers in the zookeeper.
8. When the controller notices that a broker left the cluster, it knows it is time to reassign the work to some other broker.
9. The controller selection is straightforward.  
   The 1st broker that starts in the cluster becomes the controller by creating **ephemeral controller** **node** in the zookeeper.  
   Even the other Brokers which are started try to create **ephemeral controller node** but they all get exception 🡪 ‘node already exists’ which mean controller is already elected.  
   In that case, those brokers start watching the controller to disappear.  
   When the controller dies, the **ephemeral controller node** disappears.   
   Again, every broker in the cluster tries to create **ephemeral controller node** in the zookeeper, but only one succeeds and others get **exception** once again.  
   This process shows that there is always a controller in the cluster and there exists only one controller.

Let’s see this in action.  
**Following controller entity will be created only when we start at least one broker.**

1.   
   It means there is a controller node created already.  
   A picture containing text

   Description automatically generated  
     
   A screen with text on it

   Description automatically generated with low confidence  
     
     
     
   **Bringing the down broker up**.  
     
     
     
   
2. **Brief**:
   1. Zookeeper is DB for Kafka Cluster control information.
   2. One of the Brokers in Kafka Cluster is elected to take up the responsibilities of the **Controller** and takes care of the cluster level activities.