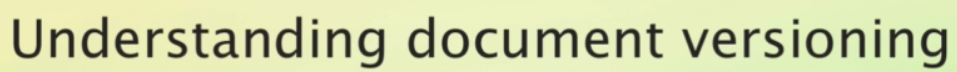
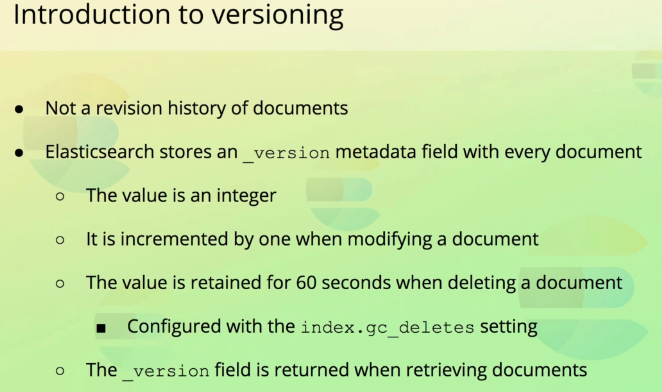
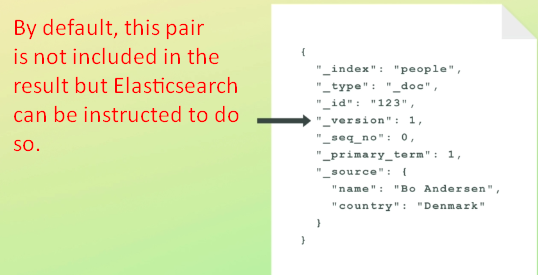
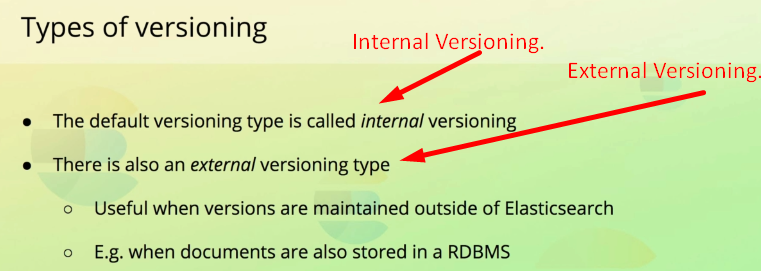
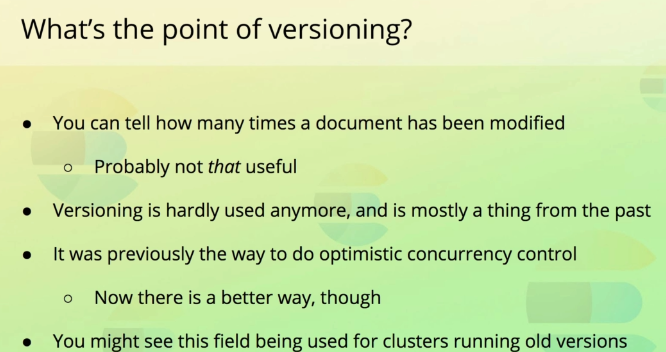
1. 
2. Elasticsearch version the doc we index.
3. It is very simple form of versioning not a **revision history** of documents. Elasticsearch only stores the most recent version of a document.
4. So, versioning doesn’t mean that you can go back in time what a given document looked like in the past.
5. Elasticsearch stores \_version metadata along with the document we index where value starts with one and incremented each time when doc is updated or deleted.
6. 
7. 
8. This is default type of version that is used and called **internal-versioning.**
9. **External Versioning:** This type is versioning is meant for situations where you maintain a documents’ version outside of Elasticsearch, such as within a database.  
   An example could be that you use a relational database as the primary data store, and index data into Elasticsearch to make it searchable.  
   
10. **How to use external versioning?**
    1. To use external versioning, we specify both the version that Elasticsearch should **store** as well as the **version type**.
    2. The version that we specify is constrained to being a natural number. On your screen you can see an example of how that is done with the index API.
11. So what is the pointing of this versioning?
    1. You can tell how many times a document has been modified.  
       **NOTE**: This is not used now or at least it should not be. The reason is that it has previously been the way to optimistic concurrency control.  
       We will get into that in the next lecture. However, with the addition of **primary terms & sequence numbers,** this way of versioning is not best practice anymore. “\_version”is still available for you to do anything but its use is probably quite limited.  
       So, why am I bothering to tell you?  
       Because you may encounter this field in one way or another such as for doing optimistic locking for applications that use a legacy version of Elasticsearch.
    2. 
12. So enough talking: let’s see how we can accomplish optimistic concurrency control with Elasticsearch.