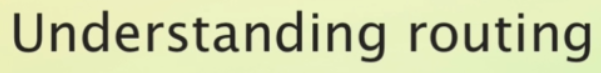
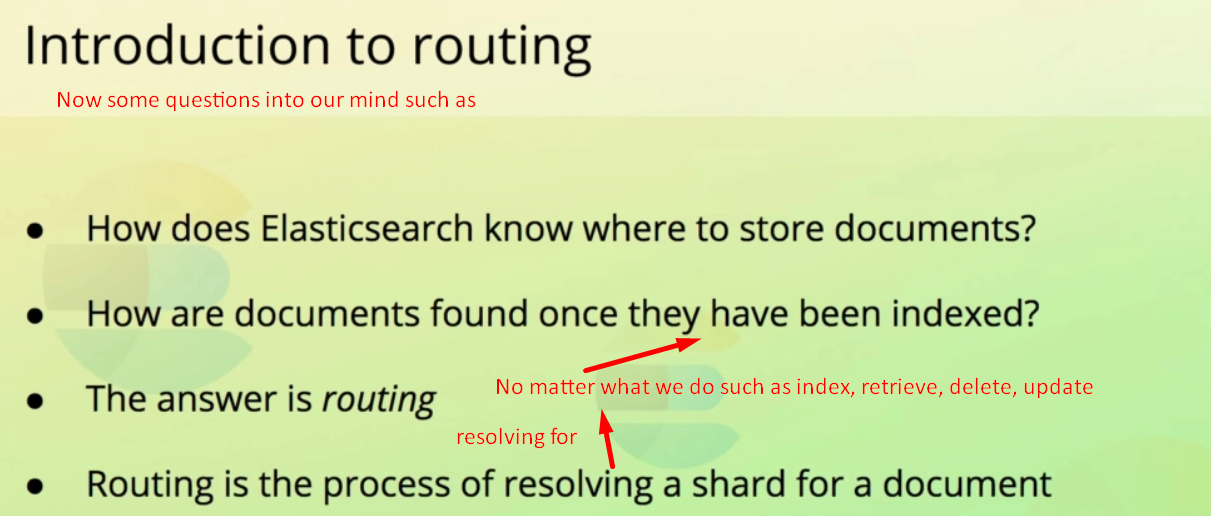
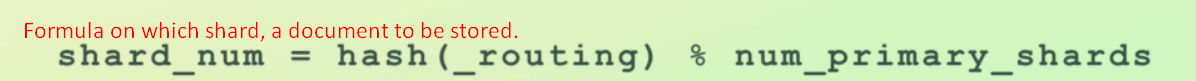
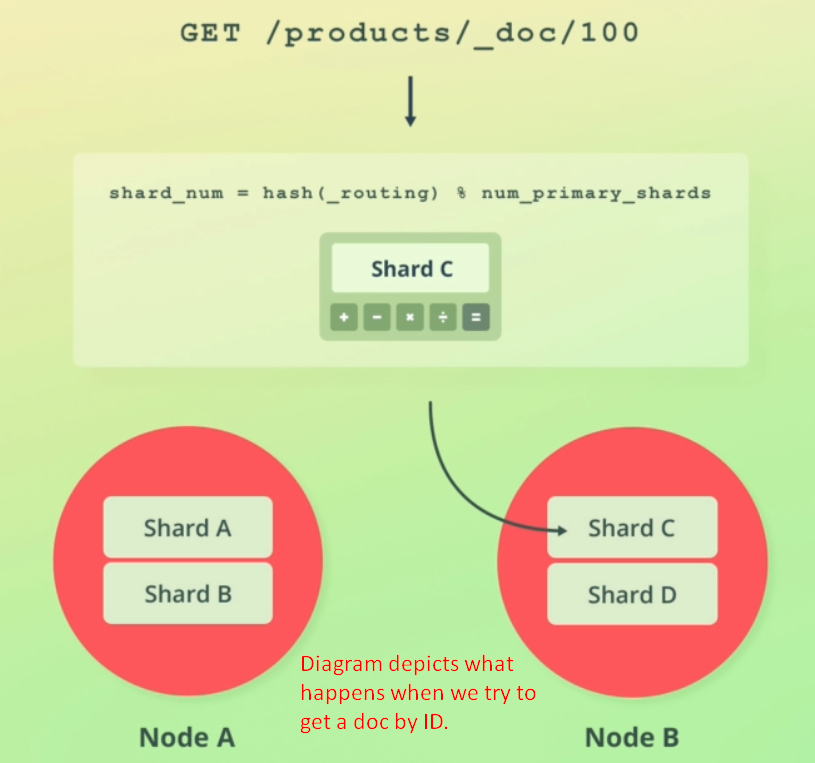
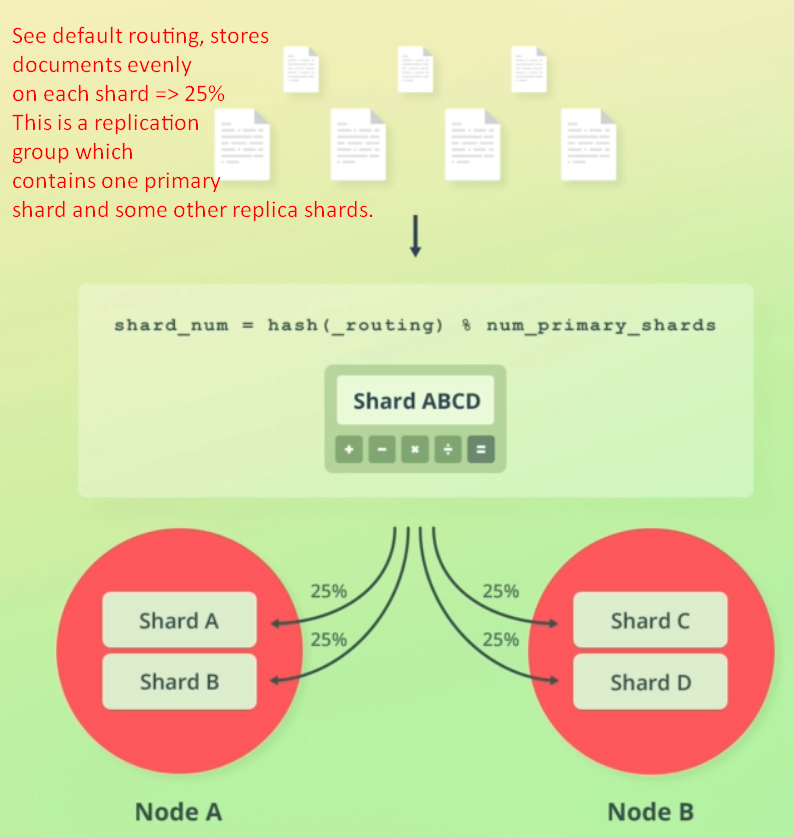
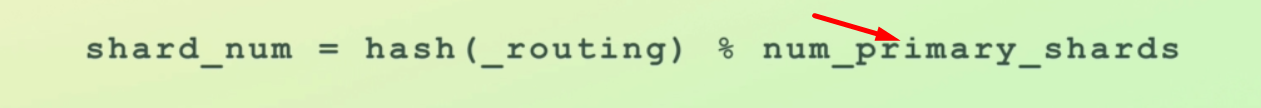
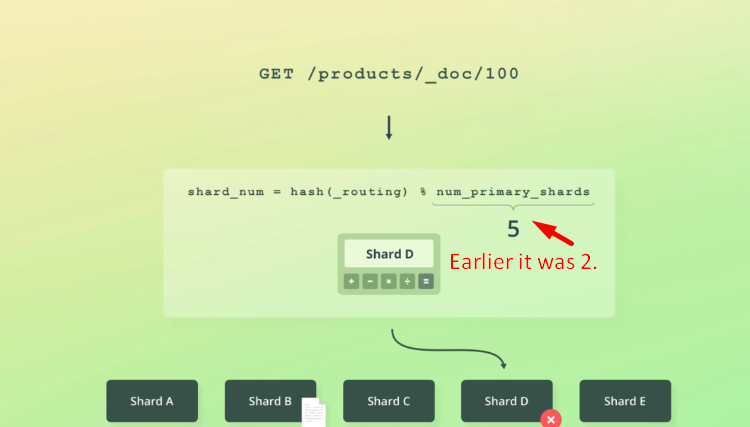
1. 
2. We indexed, updated, deleted and retrieved documents.
3. How does it work under the hood?
4. Particularly, how did Elasticsearch know on which shard to store the documents?
5. 
6. When we index a doc, Elasticsearch uses the following simple formula to calculate on which shard the document should be stored.
7. 
8. We will come back to what \_routing is soon. But by default it’s equals to document ID.  
   This formula applies to figure out a shard for indexing, deleting, updating document.
9. What happened when we retrieve a doc was that the same formula was used based on the ID we specified.   
   Elasticsearch knew that if the doc existed, it had to be on the shard that routing formula yielded when indexing.
10. But searching for documents based on some criteria other than their Ids, works differently as we will see when we get to writing our first search query.  
    
11. Routing is totally transparent to us as users of Elasticsearch. ‘
12. By providing a default routing strategy, Elasticsearch is much easier to use than if we had to deal with this ourselves.  
    Did you notice how I said “default” routing strategy? Because it’s possible to change the default routing.
13. There is benefit from the default routing that it ensures the even distribution of documents over the index’s shards.
14. If we were to change how documents are routed, we would either have to ensure that documents are distributed evenly, or accept that one shard may store significantly more documents than other shard.
15. 
16. 
    1. Elasticsearch stores a bit of metadata together with the documents that we index.
    2. Let’s see some Meta fields such as \_source, \_id, \_routing.
       1. As you saw, when we retrieved a document, the JSON we specified when adding a document, is stored within **‘\_source’** field.  
          You also saw document’s identifier is stored under “**\_id**” field.
       2. Another Meta field is the one named “\_routing”, which you saw in routing formula a moment ago. As default routing uses doc’s ID, we will not see this “\_routing” field as part of the query result. This is because, it is only added if you specify a custom routing value when you index documents.
17. Let’s talk one interesting thing.
18. Do you remember how the number of shards can’t be changed once an index has been created? If we take another look at the routing formula, then we can actually see why. Because the number of shards is used within the formula. If we were to change the no of shards for an index, then the routing formula would yield different results. That is not a problem when indexing new documents, but it would cause big trouble for existing documents.  
    
19. Problem if we change the no of shards.  
    

Suppose that we have an index with two shards, and that we indexed a document. Shard number two was the result of the routing formula so that’s where the document was stored.  
 Let’s imagine for a moment that we were able to increase the number of shards for the index, and that we increased it to five. We index some more documents. All is well. However when we try to retrieve specific documents based on their ID, Elasticsearch is sometimes unable to locate the documents. This is because a document’s ID is then run through the routing formula again, and since one of the formula’s variables has changed, the result may be different. If that is the case, Elasticsearch tries to find a document on the wrong shard, and thus returns a empty response, even though the document does exist. That is not the only reason why additional shards cannot be added to an index, though.

Second problem is that an index’s document would be distributed very unevenly.  
**NOTE**: modifying the number of shards requires creating a new index and re-indexing documents into it. That’s made fairly easy with the **Shrink and Split APIs**.

1. 