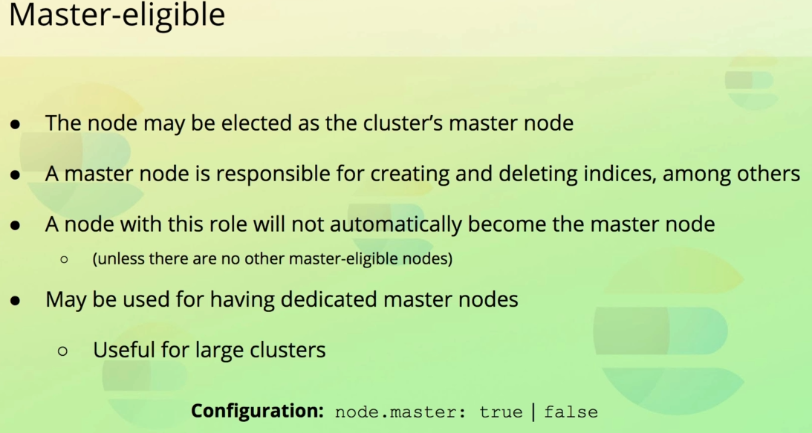
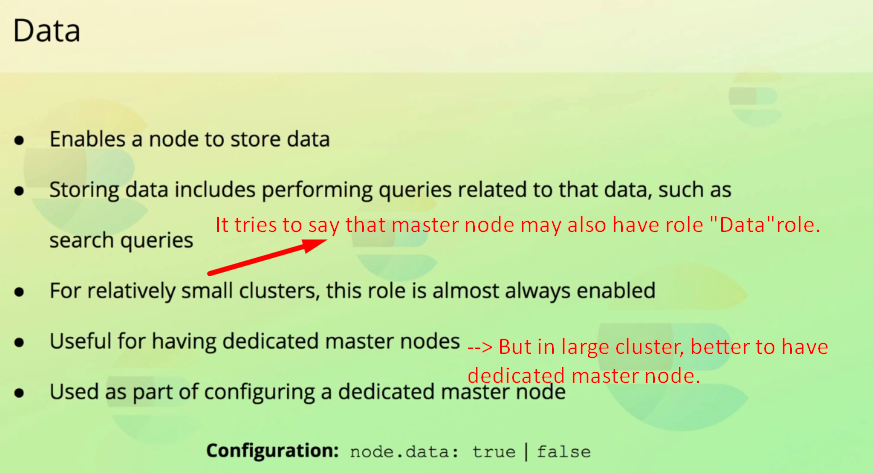
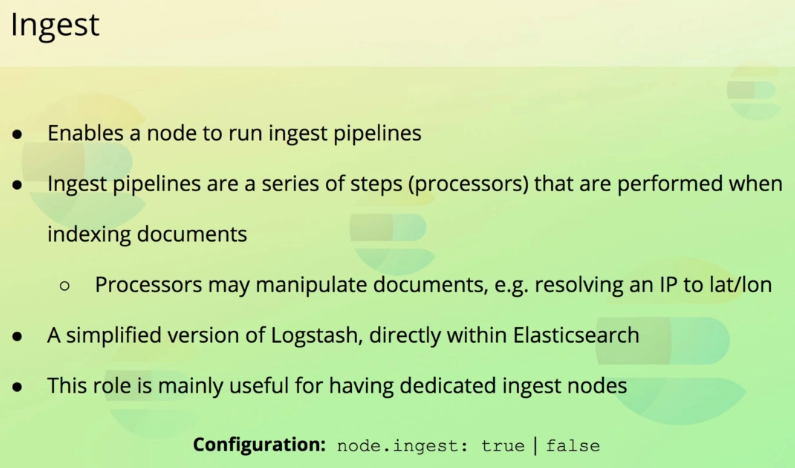
1. 
2. Data is stored in shard and shard on index and index on node and node in cluster.
3. Nodes have one or more roles defining what the node is used for.
4. Let’s take a look at the roles available.
5. **Master Role**:   
   
   1. Master node is responsible to perform cluster-wide actions.
   2. Mainly includes creating and deleting indices, keeping tracks of nodes, allocating shards to nodes.
   3. Cluster’s mater node is elected based on a voting process if there are multiple nodes with role master otherwise the one will be master.
   4. For large cluster, it’s important to have a dedicated master node as it’s crucial for ensuring that the cluster is stable. If the elected master node is too busy doing other things – such as serving search requests then cluster stability may be affected.  
      Searching for data and modifying it is expensive in terms of hardware resources, so if you see high CPU, memory and I/O usage on your master node, then it might be time to add a dedicated master node.

Data Role

1. 
2. 

Ingest

1.   
   
2. This role which enables a node to run “ingest pipelines.”
3. Ingest is a series of steps that should be performed when ingesting a document into Elasticsearch.

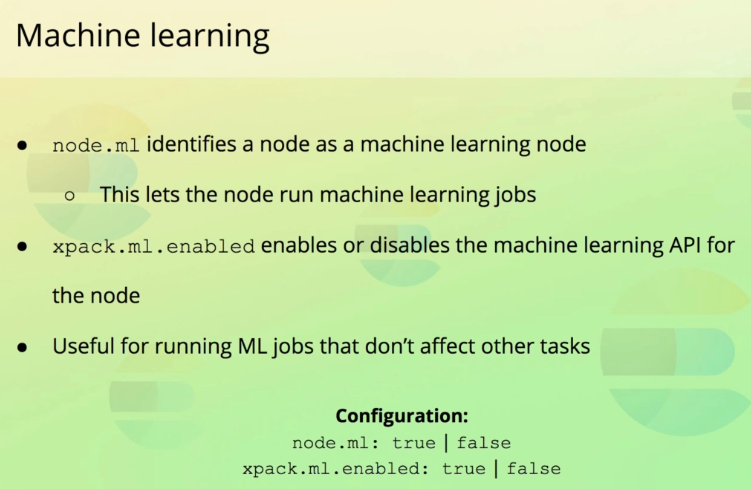
Ingesting refers to adding a document to an index.  
**Steps** are formally referred to as **processors** and they may **manipulate documents** before they add added to an index such as   
adding and removing fields, or changing values.  
**Example**: An example would be to ingest a web server’s access log, where each request is stored as Elasticsearch document.  
We could then transform the visitor’s IP address into geographical data such as latitude, longitude, country etc.

You can think of an Elasticsearch ingest pipeline as simplified Logstalsh pipeline, as it provides a subset of the Logstash functionality.

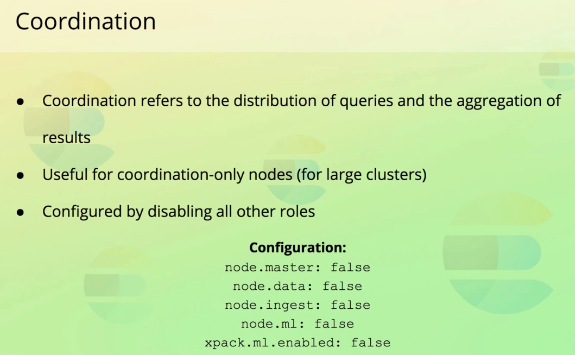
We can build fairly complex ingest pipelines that perform a lot of data transformations but if we want full functionality then Logstash.

Ingest pipelines are useful for relatively simple data transformations.  
If you’re ingesting a high volume of documents, then running them all through an ingest pipeline, may be expensive in terms of hardware resources.  
Some of the Elastic Stack products ship with ingest pipelines.  
For example, if you’use **Filebeat** to ingest Apache Access Logs, each line of the access log will be run through an ingest pipeline before being stored with in Elasticsearch index.  
Better to have a dedicated node to run documents through a pipeline if you’re running a busy website or API.

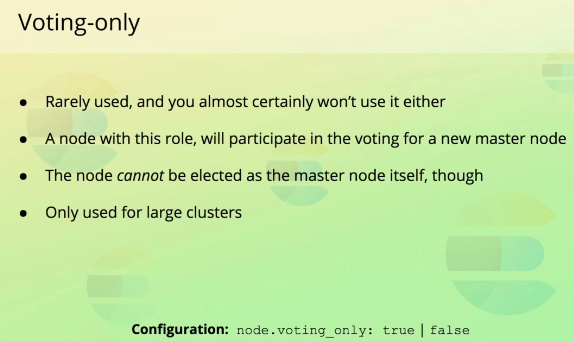
Machine Learning

1.   
   If you intend to make use of machine learning, there are **two settings** related to this feature.
   1. node.ml=true|false 🡸 This enables the node to run machine learning jobs.
   2. Xpack.ml.enabled=true|false🡸Enables or disables the a node’s capability of responding to machine learning API requests.
2. If you make use of machine learning, then these two settings enable you to run dedicated machine learning nodes.  
   That might be useful if you don’t want your background machine learning jobs to slow other things down such as search request.

Coordinate Node

1. 
2. By coordination, I am referring to how Elasticsearch distributes queries internally.
3. We have not talked about how requests are routed and coordinated yet, but we will soon.
4. Such as a node doesn’t search any data on its own; such work is delegated to data nodes.
5. Having a node only serve as a coordination node is actually not possible through a single setting, because there is no role available for doing so.  
   Instead this is accomplished by removing the other roles from a node. All that is left is then the coordination of queries.
6. To have dedicated coordination node is useful for large cluster as it can essentially be used as a load balancer.

Voting-only

1. 
2. How to know which roles a node has?
3. 