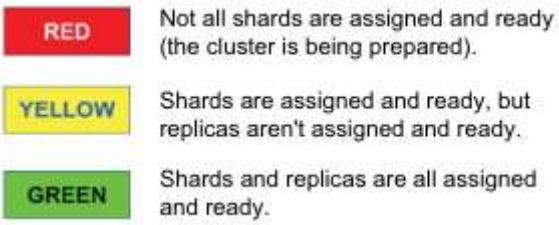
14 - Administration

SCALING THE CLUSTER

- Possível escalar um cluster de um node (em dev) até para um grande número de nodes
- ES é resistente a falhas em nodes
- Cada node roda uma instância do ES
  - Recomendado ser um servidor exclusivo para o ES
- Réplicas só fazem sentidos em nodes diferentes
- Cluster health:
  - 

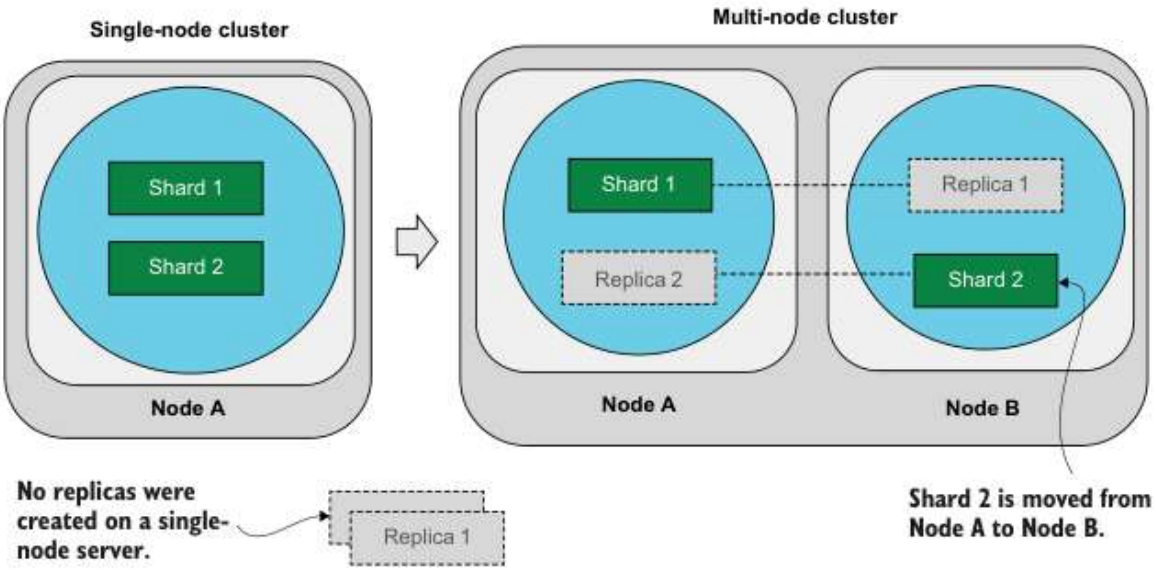
  - Explicação de erros:

Listing 14.2 Asking the cluster for a shard failure explanation

```
GET _cluster/allocation/explain
{
  "index": "chats",
  "shard": 0,
  "primary": false
}
```

```
{
  "index" : "chats",
  "shard" : 0,
  "primary" : false,
  "current_state" : "unassigned",
  "allocate_explanation" : "cannot allocate because
  ➥ allocation is not permitted to any of the nodes",
  "node_allocation_decisions" : [{
    ...
    "deciders" : [{
      "decider" : "same_shard",
       "explanation" : "a copy of this shard is already allocated to
       ➥ this node ..]"}
      ]
...
}
```

- Cada node tem um papel, que pode ser atribuido manualmente
  - master, ingest, data, ml, transform



- Figure 14.5 A newly joined node gets new shards moved from the single-node cluster.
- Shards são realocados automaticamente quando um node é integrado ao cluster
- Aumentar réplicas pode aumentar velocidade de leitura
  - Mas para escrita não, pois é feita nos shards
- Número de réplicas do índice pode ser alterado dinamicamente no _settings
  - Requer mais memória e cpu da máquina, tanto quanto o shard primário.
  - Número de shards não pode ser aumentado dinamicamente, requer index inoperante.

NODE COMMUNICATION

- Comunicação Node-Client é feita por HTTP(S), via API REST, na porta default 9200.
- Comunicação intranode ocorre na porta 9300 por default.
- Mudar esses default pode ser complexo continuamente num cluster grande.

SHARD SIZING

- Shards primários e réplicas demandam mesmos recursos de máquina.
  - E.g. Index com 10 shards de 30GB cada com 2 réplicas cada demandam 900GB de espaço, no mínimo.
  - Importante ainda ter memória adicional para realizar as operações.
- Se tivermos múltiplos índices igual, multiplicamos novamente os requisitos pelo número de índices.

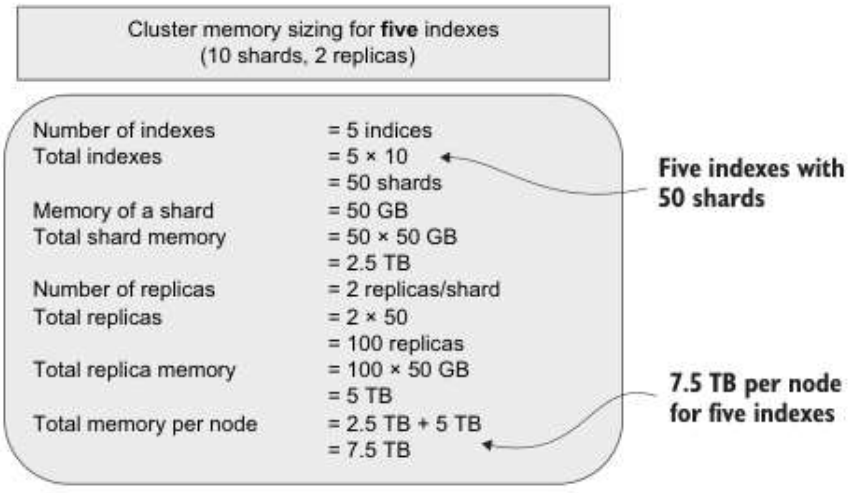


Figure 14.8 Exponential memory usage for five indexes on a node

- Scaling pode ser vertical (máquinas melhores) ou horizontal (mais máquinas)

    - Horizontal costuma ser melhor, mais fácil.

SNAPSHOTS

- Importante ter backups
    - Cloud e/ou local
    - Idealmente automatizado
- Snapshot pode ser de um index, vários ou do cluster todo
- Snapshots são incrementais
    - Além de pela API, Kibana tem uma funcionalidade para fazer snapshot também
    - Snapshots podem ter metadados que definimos na requisição
- Snapshots são úteis para fazer mudanças que não podem ser feitas dinamicamente nos index
    - Criar index novo com settings certo, fazer snapshot do index antigo e restaurá-lo sob o index recém-criado
- Com _slm (snapshot lifecycle management) podemos automatizar snapshots com algumas regras
    - e.g. tempo entre snapshots, tempo de retenção etc
    - Kibana também tem ferramentas para facilitar este SLM
    - SLM pode ser executado manulamente se quisermos, mesmo fora das regras que definimos.
- Nas versões Enterprise 7.12 em diante, snapshots são buscáveis como um index normal.
    - Pode servir como réplica para busca, poupando recursos.

ADVANCED CONFIGURATIONS

- *convention over configuration*
    - Geralmente preferível usar defaults, que funcionam bem sem ajustes
- Existem três arquivos de configuração
    - elasticsearch.yml
        - informações mais básicas e fundamentais
        - e.g. mudar portas, paths etc.
    - log4j2.properties
        - Settings de level de logging do node individual
            - Posso precisar resetar node para config valer
        - Mesma alteração pode ser feita para todo o cluster via API, mas também só se cluster estiver desligado.
            - e.g. alterar level de logs para DEBUG para resolver algum problema
                - sugerível alterar de volta ao INFO após o termino, por exemplo tendo feito o settings como 'transient' e não 'persistent', para que setting não perdure após reset do cluster
    - jvm.options
        - Ajuste de heap memory do node, assunto complexo e detalhado.
        - Sempre Criar outros arquivos auxiliares, nunca editar o arquivo original, pois pode corromper Elasticsearch
        - Nunca deixar -Xms (inicial) e Xmx (máximo) exceder 50% da memória RAM total do node.

CLUSTER MASTERS

- Node master é responsável por todas operações
    - node master na verdade é *master-eligible*, i.e. todo node master pode ser master caso o node de fato master crashe.
- Master é escolhido por eleição de todos os masters na inicialização ou quando o master morre.
    - Há alguns settings de como essa eleição ocorre (e.g. tempo)
- Só master pode alterar o cluster state, e ele que verifica se está em ordem.

    - Comunica aos outros nodes quando há alteração.
- quorum é um subconjunto dos master-eligible nodes que se comunicam com o master node para atualizações.
    - Número de master-eligible nodes mínimo é metade de todos master-eligible nodes + 1
    - Número mínimo absolute recomendado é 3
        - Torna mais difícil que surjam dois masters ao mesmo tempo caso haja problemas de conetividade entre nodes.
- Nodes podem ter múltiplos papéis ao mesmo tempo, mas a custo de desempenho.
    - Master nodes é uma boa ideia ser apenas o master node e nada mais.
    -

Summary

- Elasticsearch horizontally scales when adding new nodes to the cluster. The new nodes join the cluster as long as they are associated with the same cluster name and network settings.
- One way to improve read throughput and, thus, performance is to add replicas to the cluster. The replicas take the read hits and serve data quickly.
- Nodes communicate with each other on a transport port, set to 9300 by default. This can be modified by adjusting the `transport.port` property in the elasticsearch.yml file.
- HTTP clients communicate with Elasticsearch on an http.port (set to 9200 by default) using RESTFul endpoints.
- A node can consist of multiple indexes, and each index can consist of multiple shards. The ideal size of a shard should be not more than 50 GB of memory.
- Shards and replicas occupy space, so our organizational strategy should define the appropriate sizes based on current requirements and future use.
- Although adding replicas improves the client's read/query performance, it comes with a cost. We must be sure to carry out appropriate trials and watch for spikes before assigning a standard size for each shard.
- Elasticsearch allows backing up and restoring data using its snapshot and restore feature. Snapshots let us make backups of the cluster to a repository.
- A repository can be a local filesystem or a cloud-based object store such as AWS S3.
- A snapshot can consist of indexes, data streams (as well as the cluster state, such as persistent or transient), indexing templates, index lifecycle management (ILM) policies, and more.
- A snapshot lifecycle management (SLM) endpoint declared as `_slm` creates a policy that defines the snapshot along with its schedule and other attributes.
- We can initiate restoring data from a snapshot manually by invoking the `_restore` API.
- We can use Kibana's rich interface to define snapshots and policies and restore them. The policies are available under the Stack Management navigation menu.
- Elasticsearch exposes various settings via its elasticsearch.yml, jvm.options, and log4j2.properties files.
- We can tweak many attributes, such as changing the cluster name, moving the log and data path, adding network settings, and so on, by editing the elasticsearch.yml file.
- The config/jvm.options file defines JVM-related data for the node, but we must never edit this file.
- To customize the settings for JVM options, we need to create a new file ending with *.options (like custom_settings.options) and drop it in a folder called jvm.options.d, which should be present in the config folder for archived installations (tar or zip).
- We can set the required heap memory using the `-Xms` and `-Xmx` flags, where

`-Xms` sets the maximum heap and `-Xmx` sets the minimum heap size. As a rule of thumb, never set heap size over 50% of available RAM.

- The master node in a cluster is a critical node that looks after managing and maintaining cluster-related operations and updating the distributed cluster state.
- The master node consults a quorum of nodes to commit the cluster state or elect a new master.
- A quorum is a minimum number of master-eligible nodes carefully selected by the cluster to alleviate node failures when making decisions.
- We should provide a minimum of three master-eligible nodes when forming a cluster, to avoid a split-brain cluster.