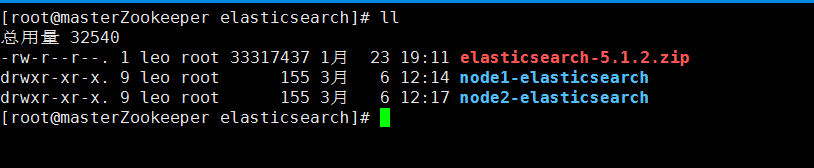
# ElasticSearch

下载elasticsearch

<https://www.elastic.co/downloads/elasticsearch>

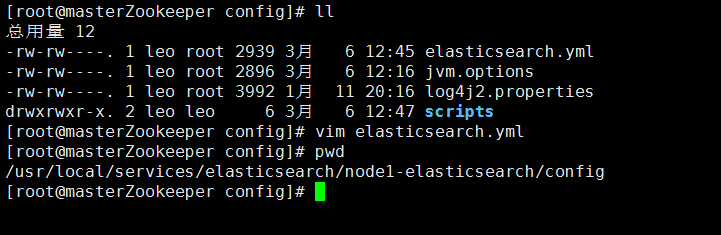
放到安装目录 /usr/local/services/elasticsearch

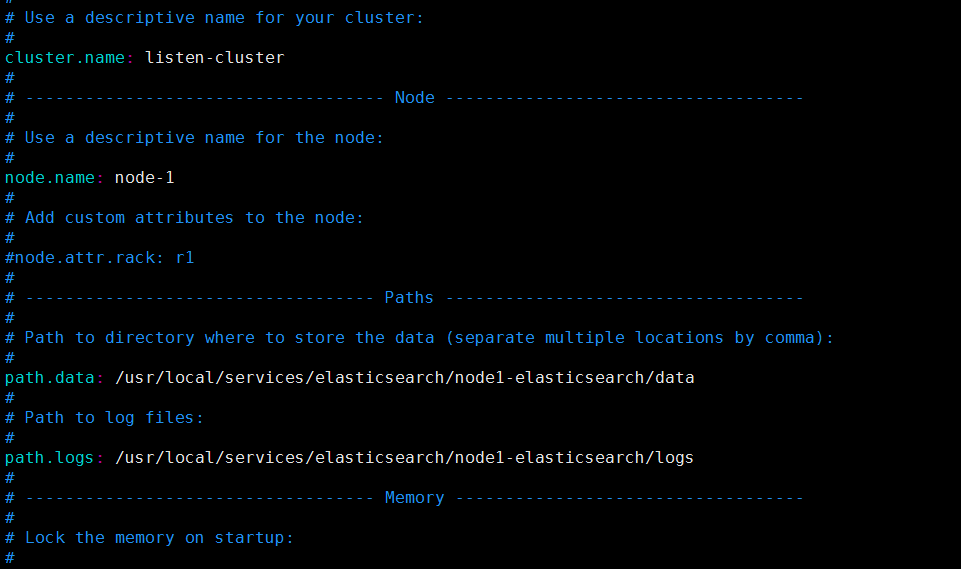
解压 复制出两个节点

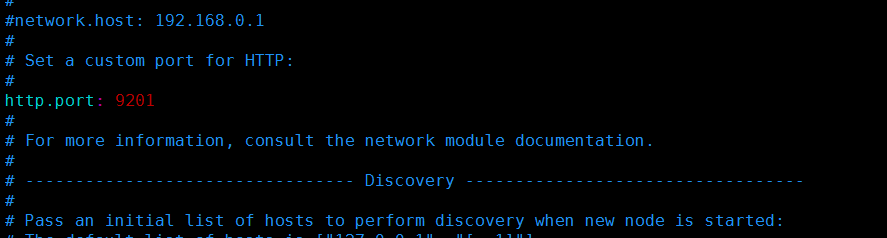


修改配置文件

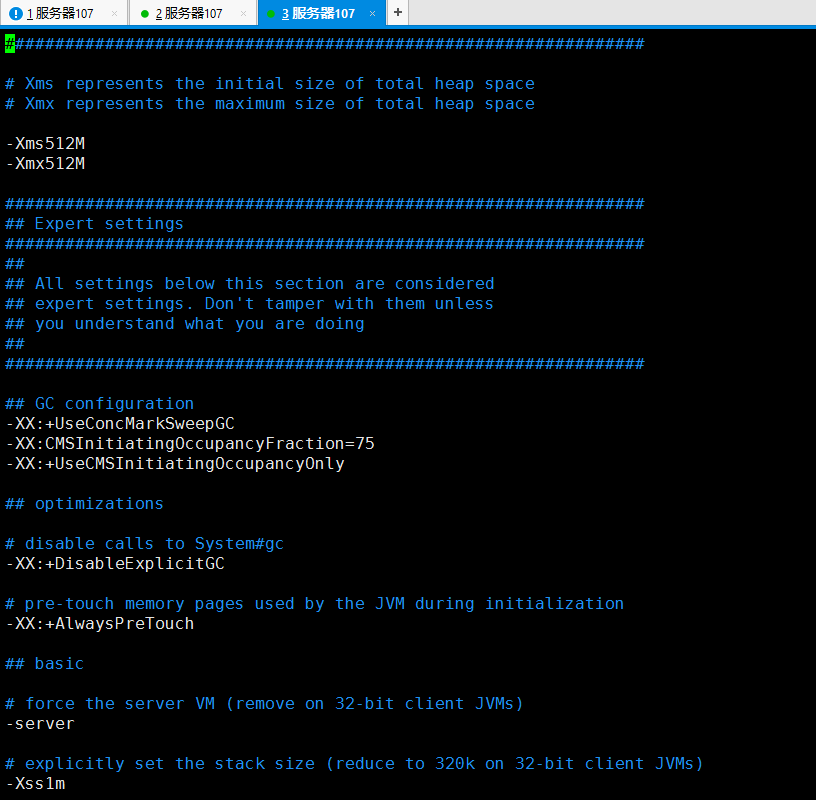
## vim elasticsearch.yml



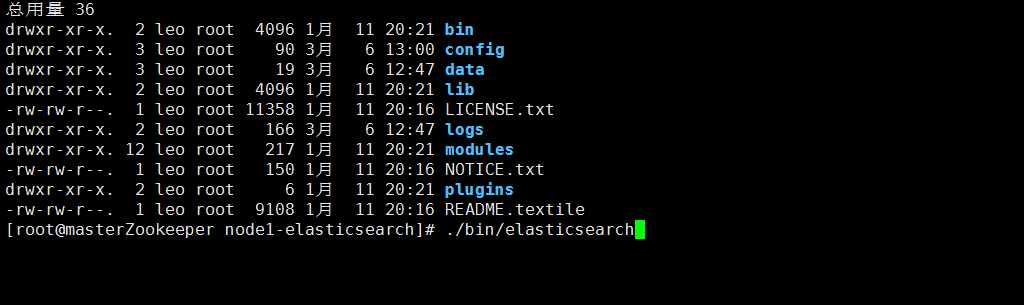




## vim jvm.options

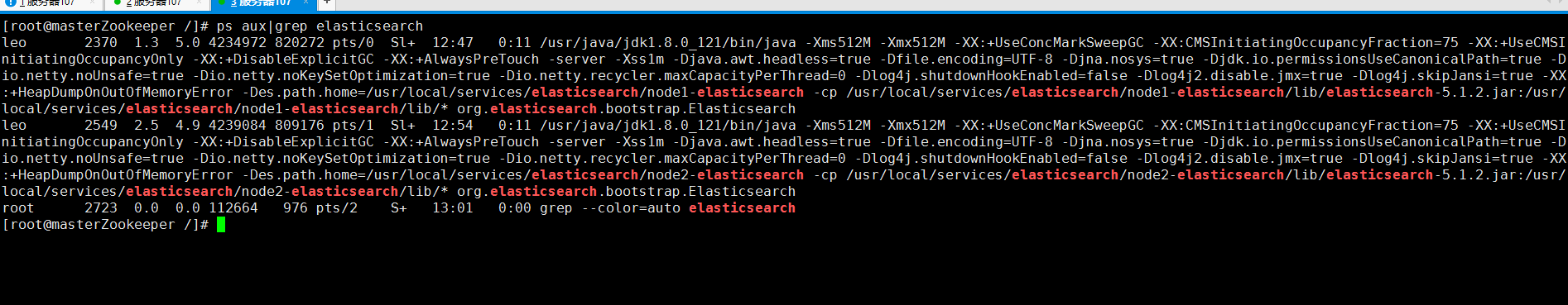


启动elasticsearch



查看进程

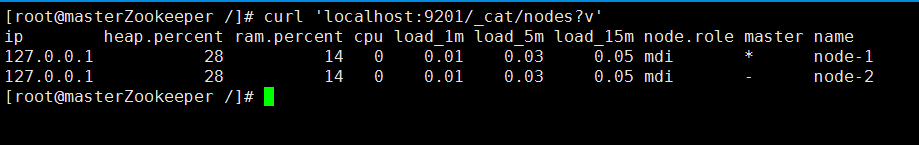
## ps aux|grep elasticsearch



## 简单测试:

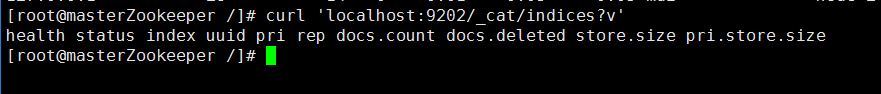
#查找所有节点

curl 'localhost:9201/\_cat/nodes?v'



#查找所有索引库

curl 'localhost:9202/\_cat/indices?v'



#创建新的索引

curl -XPUT 'localhost:9200/customer?pretty'

#查看创建的索引库

curl 'localhost:9201/\_cat/indices?v'

#往索引库中添加一个文档

curl -XPUT 'localhost:9202/customer/external/1?pretty' -d '

　　{

　　 "name": "John Doe"

　　}'

#查看添加的文档

curl -XGET 'localhost:9201/customer/external/1?pretty'

#删除索引库

curl -XDELETE 'localhost:9201/customer?pretty'

#查看是否已经删除

curl 'localhost:9201/\_cat/indices?v'

# 基础概念

    Elasticsearch有几个核心概念。从一开始理解这些概念会对整个学习过程有莫大的帮助。

## 接近实时（NRT）

        Elasticsearch是一个接近实时的搜索平台。这意味着，从索引一个文档直到这个文档能够被搜索到有一个轻微的延迟（通常是1秒）。

## 集群（cluster）

        一个集群就是由一个或多个节点组织在一起，它们共同持有你整个的数据，并一起提供索引和搜索功能。一个集群由一个唯一的名字标识，这个名字默认就是“elasticsearch”。这个名字是重要的，因为一个节点只能通过指定某个集群的名字，来加入这个集群。在产品环境中显式地设定这个名字是一个好习惯，但是使用默认值来进行测试/开发也是不错的。

## 节点（node）

        一个节点是你集群中的一个服务器，作为集群的一部分，它存储你的数据，参与集群的索引和搜索功能。和集群类似，一个节点也是由一个名字来标识的，默认情况下，这个名字是一个随机的漫威漫画角色的名字，这个名字会在启动的时候赋予节点。这个名字对于管理工作来说挺重要的，因为在这个管理过程中，你会去确定网络中的哪些服务器对应于Elasticsearch集群中的哪些节点。  
          
        一个节点可以通过配置集群名称的方式来加入一个指定的集群。默认情况下，每个节点都会被安排加入到一个叫做“elasticsearch”的集群中，这意味着，如果你在你的网络中启动了若干个节点，并假定它们能够相互发现彼此，它们将会自动地形成并加入到一个叫做“elasticsearch”的集群中。  
          
        在一个集群里，只要你想，可以拥有任意多个节点。而且，如果当前你的网络中没有运行任何Elasticsearch节点，这时启动一个节点，会默认创建并加入一个叫做“elasticsearch”的集群。

## 索引（index）

          
        一个索引就是一个拥有几分相似特征的文档的集合。比如说，你可以有一个客户数据的索引，另一个产品目录的索引，还有一个订单数据的索引。一个索引由一个名字来标识（必须全部是小写字母的），并且当我们要对对应于这个索引中的文档进行索引、搜索、更新和删除的时候，都要使用到这个名字。  
          
        在一个集群中，如果你想，可以定义任意多的索引。

## 类型（type）

        在一个索引中，你可以定义一种或多种类型。一个类型是你的索引的一个逻辑上的分类/分区，其语义完全由你来定。通常，会为具有一组共同字段的文档定义一个类型。比如说，我们假设你运营一个博客平台并且将你所有的数据存储到一个索引中。在这个索引中，你可以为用户数据定义一个类型，为博客数据定义另一个类型，当然，也可以为评论数据定义另一个类型。

## 文档（document）

          
        一个文档是一个可被索引的基础信息单元。比如，你可以拥有某一个客户的文档，某一个产品的一个文档，当然，也可以拥有某个订单的一个文档。文档以JSON（Javascript Object Notation）格式来表示，而JSON是一个到处存在的互联网数据交互格式。  
          
        在一个index/type里面，只要你想，你可以存储任意多的文档。注意，尽管一个文档，物理上存在于一个索引之中，文档必须被索引/赋予一个索引的type。

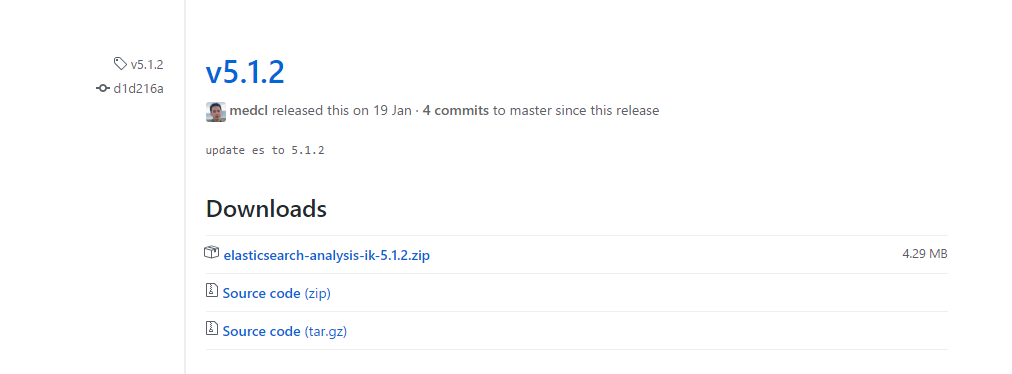
## 分片和复制（shards & replicas）

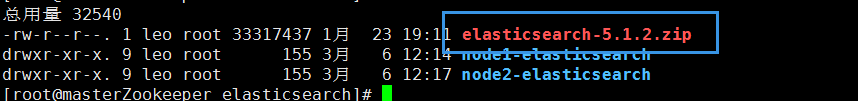
          
        一个索引可以存储超出单个结点硬件限制的大量数据。比如，一个具有10亿文档的索引占据1TB的磁盘空间，而任一节点都没有这样大的磁盘空间；或者单个节点处理搜索请求，响应太慢。  
          
        为了解决这个问题，Elasticsearch提供了将索引划分成多份的能力，这些份就叫做分片。当你创建一个索引的时候，你可以指定你想要的分片的数量。每个分片本身也是一个功能完善并且独立的“索引”，这个“索引”可以被放置到集群中的任何节点上。  
          
        分片之所以重要，主要有两方面的原因：  
          
            - 允许你水平分割/扩展你的内容容量  
            - 允许你在分片（潜在地，位于多个节点上）之上进行分布式的、并行的操作，进而提高性能/吞吐量  
          
        至于一个分片怎样分布，它的文档怎样聚合回搜索请求，是完全由Elasticsearch管理的，对于作为用户的你来说，这些都是透明的。  
          
        在一个网络/云的环境里，失败随时都可能发生，在某个分片/节点不知怎么的就处于离线状态，或者由于任何原因消失了，这种情况下，有一个故障转移机制是非常有用并且是强烈推荐的。为此目的，Elasticsearch允许你创建分片的一份或多份拷贝，这些拷贝叫做复制分片，或者直接叫复制。  
          
        复制之所以重要，有两个主要原因：  
            - 在分片/节点失败的情况下，提供了高可用性。因为这个原因，注意到复制分片从不与原/主要（original/primary）分片置于同一节点上是非常重要的。  
            - 扩展你的搜索量/吞吐量，因为搜索可以在所有的复制上并行运行  
              
        总之，每个索引可以被分成多个分片。一个索引也可以被复制0次（意思是没有复制）或多次。一旦复制了，每个索引就有了主分片（作为复制源的原来的分片）和复制分片（主分片的拷贝）之别。分片和复制的数量可以在索引创建的时候指定。在索引创建之后，你可以在任何时候动态地改变复制的数量，但是你事后不能改变分片的数量。  
          
        默认情况下，Elasticsearch中的每个索引被分片5个主分片和1个复制，这意味着，如果你的集群中至少有两个节点，你的索引将会有5个主分片和另外5个复制分片（1个完全拷贝），这样的话每个索引总共就有10个分片。  
          
        这些问题搞清楚之后，我们就要进入好玩的部分了...

# 分词器ik安装

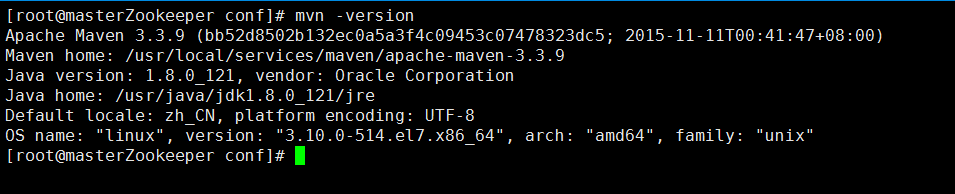
要下载自己所使用es所对应的版本

<https://github.com/medcl/elasticsearch-analysis-ik/releases>



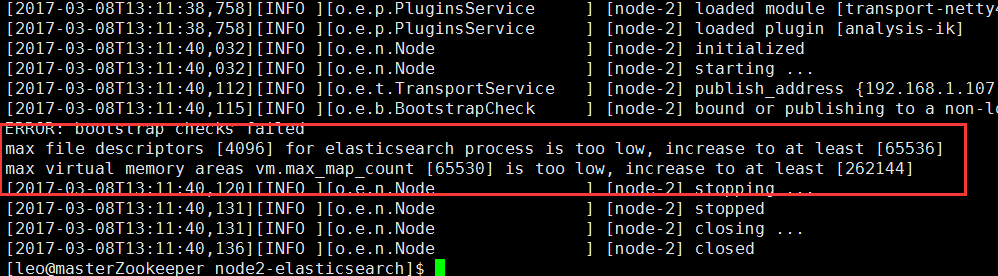


安装好maven



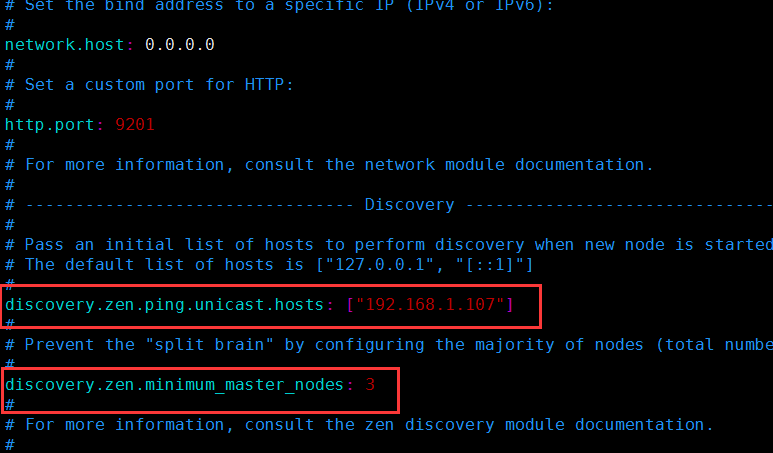
## 遇到问题

max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]

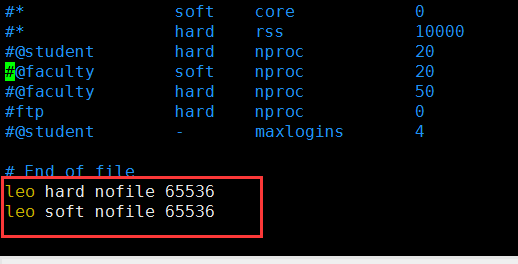


## 解决问题:

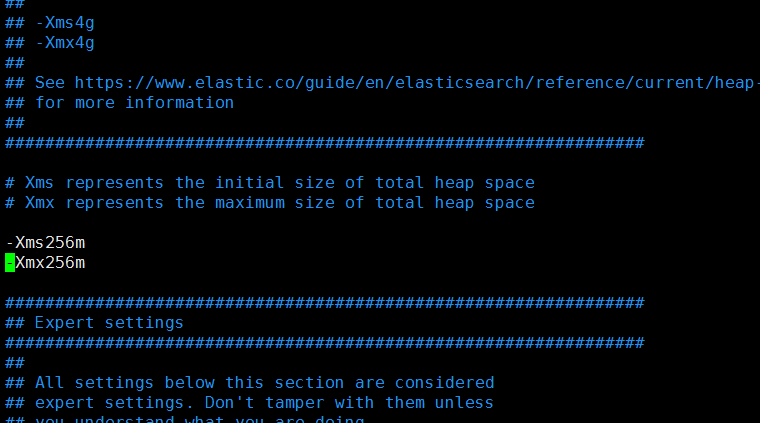
vim elasticsearch.yml



vim /etc/security/limits.conf

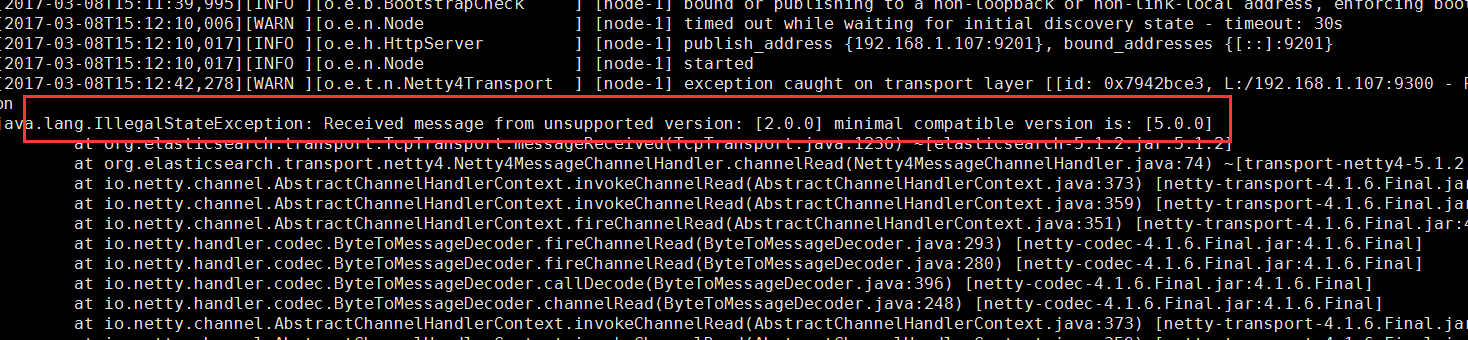


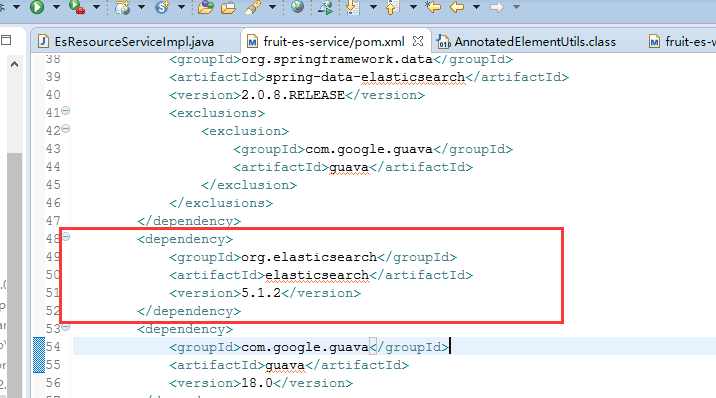
vim jvm.options



Java程序连接问题

版本问题

引入



要写 tcp端口连接

