Elasticsearch 高可用集群部署方案

# 环境规划

* 四台虚拟机：三台部署ES节点，一台部署kibana
* 虚拟机配置：4C8G，系统都是Centos7.8
* 三台ES节点角色为master节点和node节点
* 安装docker
* Elasticsearch 和kibana版本为7.10.2
* 本文档中ip地址需根据实际情况配置

# 下载Elasticsearch 7.10.2 安装包

拉取镜像：

|  |
| --- |
| #拉取elasticsearch镜像  docker pull docker.elastic.co/elasticsearch/elasticsearch:7.10.2 |

# 安装配置elasticsearch集群

## 2.1 环境配置

三台机器都需要需求：

|  |
| --- |
| vim /etc/sysctl.conf  #fs.file-max主要是配置系统最大打开文件描述符数，建议修改为655360或者更高  fs.file-max=655360  #影响Java线程数量，用于限制一个进程可以拥有的VMA(虚拟内存区域)的大小  vm.max\_map\_count = 262144  vim /etc/security/limits.conf  #可打开的文件描述符的最大数(软限制)  \* soft nofile 65536  #可打开的文件描述符的最大数(硬限制)  \* hard nofile 65536  #单个用户可用的最大进程数量(软限制)  \* soft nofile 65536  #单个用户可用的最大进程数量(硬限制)  \* hard nofile 65536  #内存不限制  \* soft memlock unlimited  \* hard memlock unlimited  #centos7特有，修改软限制  vim /etc/security/limits.d/20-nproc.conf  \* soft nproc 40960  #生效  sysctl -p |

## 2.2创建目录

|  |
| --- |
| #三节点，创建数据、配置、日志、插件目录  mkdir -p /usr/local/elasticsearch/data && mkdir -p /usr/local/elasticsearch/config && mkdir -p /usr/local/elasticsearch/logs && mkdir -p /usr/local/elasticsearch/plugins |

## 2.3 node-01配置文件

|  |
| --- |
| # 设置集群名称，集群内所有节点的名称必须一致。  cluster.name: es-cluster  # 设置节点名称，集群内节点名称必须唯一。  node.name: node-01  # 表示该节点会不会作为主节点，true表示会；false表示不会  node.master: true  # 当前节点是否用于存储数据，是：true、否：false  node.data: true  # 索引数据存放的位置  path.data: /usr/share/elasticsearch/data  # 日志文件存放的位置  path.logs: /usr/share/elasticsearch/logs  # 需求锁住物理内存，是：true、否：false  bootstrap.memory\_lock: false  # 监听地址，用于访问该es  network.host: 10.6.16.81  # es对外提供的http端口，默认 9200  http.port: 9200  # TCP的默认监听端口，默认 9300  transport.tcp.port: 9300  # es7.x 之后新增的配置，写入候选主节点的设备地址，在开启服务后可以被选为主节点  discovery.seed\_hosts: ["10.6.16.81:9300", "10.6.16.82:9300","10.6.16.83:9300"]  discovery.zen.fd.ping\_timeout: 1m  discovery.zen.fd.ping\_retries: 5  # es7.x 之后新增的配置，初始化一个新的集群时需要此配置来选举master  cluster.initial\_master\_nodes: ["10.6.16.81:9300", "10.6.16.82:9300","10.6.16.83:9300"]  # 是否支持跨域，是：true，在使用head插件时需要此配置  http.cors.enabled: true  # “\*” 表示支持所有域名  http.cors.allow-origin: "\*" |

## 2.4 node-02配置文件

|  |
| --- |
| # 设置集群名称，集群内所有节点的名称必须一致。  cluster.name: es-cluster  # 设置节点名称，集群内节点名称必须唯一。  node.name: node-02  # 表示该节点会不会作为主节点，true表示会；false表示不会  node.master: true  # 当前节点是否用于存储数据，是：true、否：false  node.data: true  # 索引数据存放的位置  path.data: /usr/share/elasticsearch/data  # 日志文件存放的位置  path.logs: /usr/share/elasticsearch/logs  # 需求锁住物理内存，是：true、否：false  bootstrap.memory\_lock: false  # 监听地址，用于访问该es  network.host: 10.6.16.82  # es对外提供的http端口，默认 9200  http.port: 9200  # TCP的默认监听端口，默认 9300  transport.tcp.port: 9300  # 设置这个参数来保证集群中的节点可以知道其它N个有master资格的节点。默认为1，对于大的集群来说，可以设置大一点的值（2-4）  discovery.zen.minimum\_master\_nodes: 1  # es7.x 之后新增的配置，写入候选主节点的设备地址，在开启服务后可以被选为主节点  discovery.seed\_hosts: ["10.6.16.81:9300", "10.6.16.82:9300","10.6.16.83:9300"]  discovery.zen.fd.ping\_timeout: 1m  discovery.zen.fd.ping\_retries: 5  # es7.x 之后新增的配置，初始化一个新的集群时需要此配置来选举master  cluster.initial\_master\_nodes: ["10.6.16.81:9300", "10.6.16.82:9300","10.6.16.83:9300"]  # 是否支持跨域，是：true，在使用head插件时需要此配置  http.cors.enabled: true  # “\*” 表示支持所有域名  http.cors.allow-origin: "\*" |

## 2.5 node-03配置文件

|  |
| --- |
| # 设置集群名称，集群内所有节点的名称必须一致。  cluster.name: es-cluster  # 设置节点名称，集群内节点名称必须唯一。  node.name: node-03  # 表示该节点会不会作为主节点，true表示会；false表示不会  node.master: true  # 当前节点是否用于存储数据，是：true、否：false  node.data: true  # 索引数据存放的位置  path.data: /usr/share/elasticsearch/data  # 日志文件存放的位置  path.logs: /usr/share/elasticsearch/logs  # 需求锁住物理内存，是：true、否：false  bootstrap.memory\_lock: false  # 监听地址，用于访问该es  network.host: 10.6.16.83  # es对外提供的http端口，默认 9200  http.port: 9200  # TCP的默认监听端口，默认 9300  transport.tcp.port: 9300  # 设置这个参数来保证集群中的节点可以知道其它N个有master资格的节点。默认为1，对于大的集群来说，可以设置大一点的值（2-4）  discovery.zen.minimum\_master\_nodes: 1  # es7.x 之后新增的配置，写入候选主节点的设备地址，在开启服务后可以被选为主节点  discovery.seed\_hosts: ["10.6.16.81:9300", "10.6.16.82:9300","10.6.16.83:9300"]  discovery.zen.fd.ping\_timeout: 1m  discovery.zen.fd.ping\_retries: 5  # es7.x 之后新增的配置，初始化一个新的集群时需要此配置来选举master  cluster.initial\_master\_nodes: ["10.6.16.81:9300", "10.6.16.82:9300","10.6.16.83:9300"]  # 是否支持跨域，是：true，在使用head插件时需要此配置  http.cors.enabled: true  # “\*” 表示支持所有域名  http.cors.allow-origin: "\*" |

## 2.6 启动三台节点elasticsearch容器

|  |
| --- |
| docker run -d --network=host --privileged=true \  -e ES\_JAVA\_OPTS="-Xms1024m -Xmx1024m" \  -e TAKE\_FILE\_OWNERSHIP=true --name es \  -v /usr/local/elasticsearch/config/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml \  -v /usr/local/elasticsearch/data:/usr/share/elasticsearch/data \  -v /usr/local/elasticsearch/logs:/usr/share/elasticsearch/logs \  -v /usr/local/elasticsearch/plugins:/usr/share/elasticsearch/plugins \  docker.elastic.co/elasticsearch/elasticsearch:7.10.2 |

# 安装kibana

## 4.1拉取镜像：

|  |
| --- |
| #拉取kibana镜像  docker pull docker.elastic.co/kibana/kibana:7.10.2 |

## 4.2 安装配置kibana

|  |
| --- |
| #创建目录  mkdir -p /usr/local/kibana/data && mkdir -p /usr/local/kibana/config && mkdir -p /usr/local/elasticsearch/plugins  #修改配置文件  vim /usr/local/kibana/config/kibana.yml  server.name: kibana  server.host: "0"  elasticsearch.hosts: ["http://10.6.16.81:9200","http://10.6.16.82:9200","http://10.6.16.83:9200"]  monitoring.ui.container.elasticsearch.enabled: true  #启动kibana  docker run --name kibana \  -v /usr/local/kibana/config/kibana.yml:/usr/share/kibana/config/kibana.yml \  -v /usr/local/kibana/data:/usr/share/kibana/data \  -v /usr/local/kibana/plugins:/usr/share/kibana/plugins \  -d docker.elastic.co/kibana/kibana:7.10.2 |