ES集群环境安装及head插件使用

1. Docker安装es
2. 使用docker拉取es镜像

docker pull elasticsearch:6.8.3

1. 运行es容器

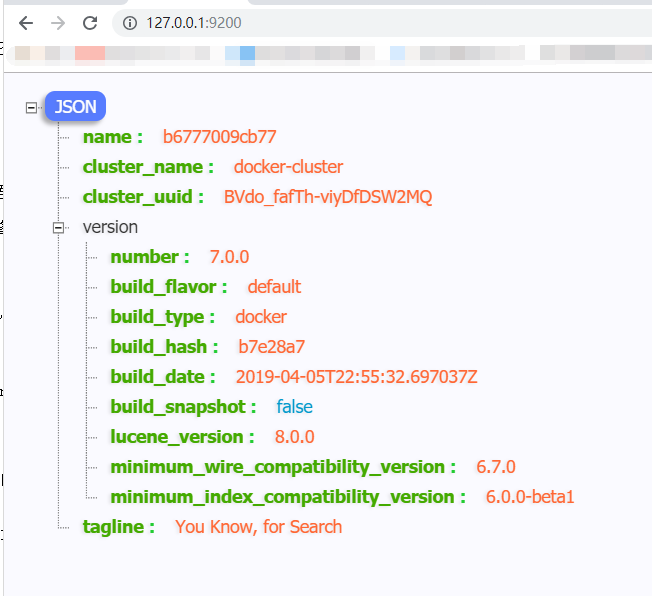
docker run -e ES\_JAVA\_OPTS="-Xms256m -Xmx256m" -d --name elasticsearch  -p 9200:9200 -p 9300:9300  镜像id

-e ES\_JAVA\_OPTS="-Xms256m -Xmx256m" //设置初始堆内存和最大内存 也可以调整虚拟机内存

测试：

在浏览器输入：http://192.168.3.240:9200/ 你的服务器ip 端口号

浏览器返回类似如下信息，证明安装成功



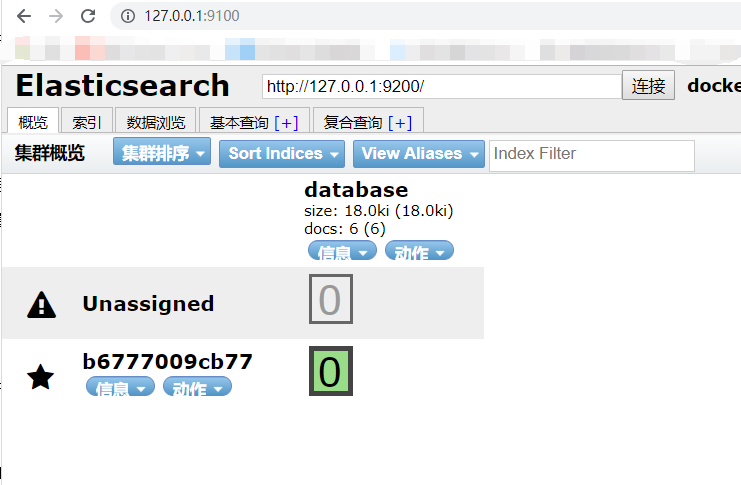
1. Docker安装head插件
2. 拉取镜像

docker pull mobz/elasticsearch-head:5

1. 运行head容器

docker run -d -p 9100:9100 docker.io/mobz/elasticsearch-head:5

1. 在浏览器中打开elasticsearch-head页面，填入ElasticSearch地址



PS:

(1).这里可能出现跨域访问拒绝问题，解决方法：

进入elasticsearch容器内部，修改配置文件elasticsearch.yml

docker ps -a #拿到运行容器elasticsearch 的 id docker exec -it \*\*\*\*\*\*(容器id) /bin/bash

cd ./config

vim elasticsearch.yml

插入以下内容

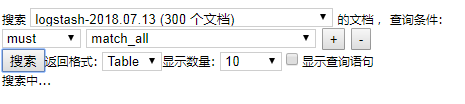
http.cors.enabled: true

http.cors.allow-origin: "\*"

保存退出重启head容器即可

(2).Elasticsearch-head 连接 Elasticsearch-6.0 + 可能会出现无法检索问题

Head 连接Elasticsearch6时其他功能都好使， 但数据浏览模块不能显示数据了，一直显示搜索中，但不返回结果



看一下网络流可知报406 错误

解决办法：

1. cd elasticsearch-head/\_site
3. vi vendor.js

修改如下

1. #1. 6886行
2. contentType: "application/x-www-form-urlencoded
3. 改为
4. contentType: "application/json;charset=UTF-8"
5. #2. 7574行
6. var inspectData = s.contentType === "application/x-www-form-urlencoded" &&
7. 改为
8. var inspectData = s.contentType === "application/json;charset=UTF-8" &&

重新启动elasticsearch-head就可以正常检索了

1. es集群搭建
2. 创建数据文件挂载目录

mkdir –p ES/config

mkdir –p ES/data1

mkdir –p ES/data2

mkdir –p ES/data3

注：如果ELK选的6.X版本的，那么读者需将data1 data2 data3 开启777权限 chmod 777 data1 data2 data3

1. 在ES/config目录下创建配置文件

**es1.yml**

cluster.name: elasticsearch-cluster

node.name: es-node1

network.bind\_host: 0.0.0.0

network.publish\_host: 192.168.9.240

http.port: 9200

transport.tcp.port: 9300

http.cors.enabled: true

http.cors.allow-origin: "\*"

node.master: true

node.data: true

discovery.zen.ping.unicast.hosts:["192.168.9.240:9300","192.168.9.240:9301","192.168.9.240:9302"]

discovery.zen.minimum\_master\_nodes: 2

**es2.yml**

1. cluster.name: elasticsearch-cluster
2. node.name: es-node2
3. network.bind\_host: 0.0.0.0
4. network.publish\_host: 192.168.9.240
5. http.port: 9201
6. transport.tcp.port: 9301
7. http.cors.enabled: true
8. http.cors.allow-origin: "\*"
9. node.master: true
10. node.data: true
11. discovery.zen.ping.unicast.hosts:["192.168.9.240:9300","192.168.9.240:9301","192.168.9.240:9302"]
12. discovery.zen.minimum\_master\_nodes: 2

**es3.yml**

cluster.name: elasticsearch-cluster

node.name: es-node3

network.bind\_host: 0.0.0.0

network.publish\_host: 192.168.9.240

http.port: 9202

transport.tcp.port: 9302

http.cors.enabled: true

http.cors.allow-origin: "\*"

node.master: true

node.data: true

discovery.zen.ping.unicast.hosts:["192.168.9.240:9300","192.168.9.240:9301","192.168.9.240:9302"]

discovery.zen.minimum\_master\_nodes: 2

3.调高JVM线程数限制数量

在centos窗口中，修改配置sysctl.conf

vim /etc/sysctl.conf

加入如下内容：

vm.max\_map\_count=262144

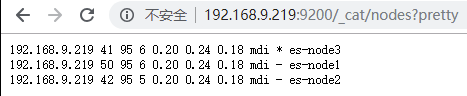
启用配置：

sysctl -p

注：这一步是为了防止启动容器时，报出如下错误：

bootstrap checks failed max virtual memory areas vm.max\_map\_count [65530] likely too low, increase to at least [262144]

4.在浏览器地址栏访问http://192.168.9.219:9200/\_cat/nodes?pretty 查看节点状态



注：节点名称带表示为主节点\*

至此es集群搭建完毕

**附：分布式搜索elasticsearch配置文件详解**

转

配置文件位于%ES\_HOME%/config/elasticsearch.yml文件中，用Editplus打开它，你便可以进行配置。

        所有的配置都可以使用环境变量，例如：

node.rack: ${RACK\_ENV\_VAR}

        表示环境变量中有一个RACK\_ENV\_VAR变量。

        下面列举一下elasticsearch的可配置项：

         1. 集群名称，默认为elasticsearch：

cluster.name: elasticsearch

         2. 节点名称，es启动时会自动创建节点名称，但你也可进行配置：

node.name: "Franz Kafka"

         3. 是否作为主节点，每个节点都可以被配置成为主节点，默认值为true：

node.master: true

         4. 是否存储数据，即存储索引片段，默认值为true：

node.data: true

         master和data同时配置会产生一些奇异的效果：

         1) 当master为false，而data为true时，会对该节点产生严重负荷；

         2) 当master为true，而data为false时，该节点作为一个协调者；

         3) 当master为false，data也为false时，该节点就变成了一个负载均衡器。

         你可以通过连接http://localhost:9200/\_cluster/health或者http://localhost:9200/\_cluster/nodes，或者使用插件http://github.com/lukas-vlcek/bigdesk或http://mobz.github.com/elasticsearch-head来查看集群状态。

         5. 每个节点都可以定义一些与之关联的通用属性，用于后期集群进行碎片分配时的过滤：

node.rack: rack314

         6. 默认情况下，多个节点可以在同一个安装路径启动，如果你想让你的es只启动一个节点，可以进行如下设置：

node.max\_local\_storage\_nodes: 1

         7. 设置一个索引的碎片数量，默认值为5：

index.number\_of\_shards: 5

         8. 设置一个索引可被复制的数量，默认值为1：

index.number\_of\_replicas: 1

         当你想要禁用公布式时，你可以进行如下设置：

index.number\_of\_shards: 1

index.number\_of\_replicas: 0

         这两个属性的设置直接影响集群中索引和搜索操作的执行。假设你有足够的机器来持有碎片和复制品，那么可以按如下规则设置这两个值：

         1) 拥有更多的碎片可以提升索引执行能力，并允许通过机器分发一个大型的索引；

         2) 拥有更多的复制器能够提升搜索执行能力以及集群能力。

         对于一个索引来说，number\_of\_shards只能设置一次，而number\_of\_replicas可以使用索引更新设置API在任何时候被增加或者减少。

         ElasticSearch关注加载均衡、迁移、从节点聚集结果等等。可以尝试多种设计来完成这些功能。

         可以连接http://localhost:9200/A/\_status来检测索引的状态。

         9. 配置文件所在的位置，即elasticsearch.yml和logging.yml所在的位置：

path.conf: /path/to/conf

         10. 分配给当前节点的索引数据所在的位置：

path.data: /path/to/data

         可以可选择的包含一个以上的位置，使得数据在文件级别跨越位置，这样在创建时就有更多的自由路径，如：

path.data: /path/to/data1,/path/to/data2

         11. 临时文件位置：

path.work: /path/to/work

         12. 日志文件所在位置：

path.logs: /path/to/logs

         13. 插件安装位置：

path.plugins: /path/to/plugins

         14. 插件托管位置，若列表中的某一个插件未安装，则节点无法启动：

plugin.mandatory: mapper-attachments,lang-groovy

         15. JVM开始交换时，ElasticSearch表现并不好：你需要保障JVM不进行交换，可以将bootstrap.mlockall设置为true禁止交换：

bootstrap.mlockall: true

         请确保ES\_MIN\_MEM和ES\_MAX\_MEM的值是一样的，并且能够为ElasticSearch分配足够的内在，并为系统操作保留足够的内存。

         16. 默认情况下，ElasticSearch使用0.0.0.0地址，并为http传输开启9200-9300端口，为节点到节点的通信开启9300-9400端口，也可以自行设置IP地址：

network.bind\_host: 192.168.0.1

         17. publish\_host设置其他节点连接此节点的地址，如果不设置的话，则自动获取，publish\_host的地址必须为真实地址：

network.publish\_host: 192.168.0.1

         18. bind\_host和publish\_host可以一起设置：

network.host: 192.168.0.1

         19. 可以定制该节点与其他节点交互的端口：

transport.tcp.port: 9300

         20. 节点间交互时，可以设置是否压缩，转为为不压缩：

transport.tcp.compress: true

         21. 可以为Http传输监听定制端口：

http.port: 9200

         22. 设置内容的最大长度：

http.max\_content\_length: 100mb

         2 3 . 禁止HTTP

http.enabled: false

         2 4 . 网关允许在所有集群重启后持有集群状态，集群状态的变更都会被保存下来，当第一次启用集群时，可以从网关中读取到状态，默认网关类型（也是推荐的）是local：

gateway.type: local

         2 5 . 允许在N个节点启动后恢复过程：

gateway.recover\_after\_nodes: 1

         2 6 . 设置初始化恢复过程的超时时间：

gateway.recover\_after\_time: 5m

         2 7 . 设置该集群中可存在的节点上限：

gateway.expected\_nodes: 2

         2 8 . 设置一个节点的并发数量，有两种情况，一种是在初始复苏过程中：

cluster.routing.allocation.node\_initial\_primaries\_recoveries: 4

         另一种是在添加、删除节点及调整时：

cluster.routing.allocation.node\_concurrent\_recoveries: 2

         2 9 . 设置复苏时的吞吐量，默认情况下是无限的：

indices.recovery.max\_size\_per\_sec: 0

        30 . 设置从对等节点恢复片段时打开的流的数量上限：

indices.recovery.concurrent\_streams: 5

         3 1 . 设置一个集群中主节点的数量，当多于三个节点时，该值可在2-4之间：

discovery.zen.minimum\_master\_nodes: 1

         3 2 . 设置ping其他节点时的超时时间，网络比较慢时可将该值设大：

discovery.zen.ping.timeout: 3s

http://elasticsearch.org/guide/reference/modules/discovery/zen.html上有更多关于discovery的设置。

         3 3 . 禁止当前节点发现多个集群节点，默认值为true：

discovery.zen.ping.multicast.enabled: false

         3 4 . 设置新节点被启动时能够发现的主节点列表（主要用于不同网段机器连接）：

discovery.zen.ping.unicast.hosts: ["host1", "host2:port", "host3[portX-portY]"]

       35.设置是否可以通过正则或者\_all删除或者关闭索引

action.destructive\_requires\_name 默认false 允许 可设置true不允许