## ES单机集群步骤

注：本文以单节点ES运行正常为前提 以三个节点为例

# ****一、安装Head  插件****

****Elasticsearch Head Plugin:**** 对ES进行各种操作，如查询、删除、浏览索引等。

### 1、下载elasticsearch-head并解压

在线下载：wget  https://github.com/mobz/elasticsearch-head/archive/master.zip

或者到github下载：https://github.com/mobz/elasticsearch-head

unzip  elasticsearch-head-master.zip //解压zip文件

mv  elasticsearch-head-master.zip  /home/ntc/code/elasticsearch-head //解压到自定义目录并修改文件夹名为elasticsearch-head

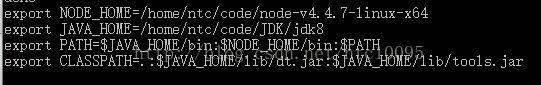
### 2、安装node

由于head插件本质上还是一个nodejs的工程，因此需要安装node，使用npm来安装依赖的包。（npm可以理解为maven）

wget https://npm.taobao.org/mirrors/node/latest-v4.x/node-v4.4.7-linux-x64.tar.gz

tar -zxvf node-v4.4.7-linux-x64.tar.gz

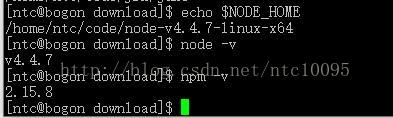
解压完node的安装文件后，需要配置下环境变量,编辑/etc/profile，添加



保存后别忘记立即执行以下

source /etc/profile

这个时候可以测试一下node是否生效：



### 3、安装grunt

grunt是一个很方便的构建工具，可以进行打包压缩、**[测试](http://lib.csdn.net/base/softwaretest" \o "软件测试知识库" \t "https://blog.csdn.net/llwwlql/article/details/_blank)**、执行等等的工作，5.0里的head插件就是通过grunt启动的。因此需要安装一下grunt：

cd  /home/ntc/code/elasticsearch-head

安装nodejs

npm install -g grunt-cli  //执行后会生成node\_modules文件夹

npm install

注：

5.0以上，elasticsearch-head 不能放在elasticsearch的 plugins、modules 目录下，否则elasticsearch启动会报错

### 4、修改head源码

vi /home/ntc/code/elasticsearch-head/Gruntfile.**[js](http://lib.csdn.net/base/javascript" \o "JavaScript知识库" \t "https://blog.csdn.net/llwwlql/article/details/_blank)**

connect: {

server: {

options: {

port: 9100,

hostname: '\*',

base: '.',

keepalive: true

}

}

}

增加hostname属性，设置为\*

修改连接地址：

目录：vi /home/ntc/code/elasticsearch-head/\_site/app.js

修改head的连接地址:

this.base\_uri = this.config.base\_uri || this.prefs.get("app-base\_uri") || "http://localhost:9200";

把localhost修改成你es的服务器地址，如:

this.base\_uri = this.config.base\_uri || this.prefs.get("app-base\_uri") || "http://192.168.40.133:9200";

### 5、运行head 启动nodejs

cd /home/ntc/code/elasticsearch-head //先跳转到head目录下

grunt server //若想在后台运行，结尾追加“&”

# ****二、配置集群****

### 复制单节点

复制单节点之前需要把ES data文件夹下的内容删除 避免后续启动报错

[root@localhost data]# cd elasticsearch-5.5.1/data

[root@localhost data]# rm -rf nodes

复制单节点

[root@localhost data]# cp -R elasticsearch-5.5.1 elasticsearch-5.5.1\_node2

复制单节点

[root@localhost data]# cp -R elasticsearch-5.5.1 elasticsearch-5.5.1\_node3

赋权elasticsearch用户

[root@localhost data]# chown -R elasticsearch elasticsearch-5.5.1\_node2

[root@localhost data]# chown -R elasticsearch elasticsearch-5.5.1\_node3

### 修改配置文件

[root@localhost data]# cd elasticsearch-5.5.1/config

修改节点一配置文件

[root@localhost data]# vim elasticsearch.yml

修改为如下内容：

#node.master 默认值为true 每个节点都不选 默认都可以作为主节点 在主节点死掉情况下可以自动主

#集群名称 节点必须保持一致

cluster.name: my-application

#节点名称 各节点必须不一致

node.name: node-0

#节点数据目录

path.data: /data/elasticsearch-5.5.1/data

#节点日志目录

path.logs: /data/elasticsearch-5.5.1/logs

bootstrap.memory\_lock: false

bootstrap.system\_call\_filter: false

network.host: 0.0.0.0

#节点端口 各节点必须不一致

http.port: 9200

#节点通信端口 各节点必须不一致

transport.tcp.port: 9300

#可被选为主节点的集合 必须写通信端口

discovery.zen.ping.unicast.hosts:["192.168.0.46:9300","192.168.0.46:9301","192.168.0.46:9302"]

#集群最多节点个数

node.max\_local\_storage\_nodes: 3

#允许外网访问

http.cors.enabled: true

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

[root@localhost data]# cd elasticsearch-5.5.1\_node2/config

修改节点二配置文件

[root@localhost data]# vim elasticsearch.yml

修改为如下内容：

#node.master 默认值为true 每个节点都不选 默认都可以作为主节点 在主节点死掉情况下可以自动主

#集群名称 节点必须保持一致

cluster.name: my-application

#节点名称 各节点必须不一致

node.name: node-1

#节点数据目录

path.data: /data/elasticsearch-5.5.1\_node2/data

#节点日志目录

path.logs: /data/elasticsearch-5.5.1\_node2/logs

bootstrap.memory\_lock: false

bootstrap.system\_call\_filter: false

network.host: 0.0.0.0

#节点端口 各节点必须不一致

http.port: 9201

#节点通信端口 各节点必须不一致

transport.tcp.port: 9301

#可被选为主节点的集合 必须写通信端口

discovery.zen.ping.unicast.hosts:["192.168.0.46:9300","192.168.0.46:9301","192.168.0.46:9302"]

#集群最多节点个数

node.max\_local\_storage\_nodes: 3

#允许外网访问

http.cors.enabled: true

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

[root@localhost data]# cd elasticsearch-5.5.1\_node3/config

修改节点三配置文件

[root@localhost data]# vim elasticsearch.yml

修改为如下内容：

#node.master 默认值为true 每个节点都不选 默认都可以作为主节点 在主节点死掉情况下可以自动主

#集群名称 节点必须保持一致

cluster.name: my-application

#节点名称 各节点必须不一致

node.name: node-2

#节点数据目录

path.data: /data/elasticsearch-5.5.1\_node3/data

#节点日志目录

path.logs: /data/elasticsearch-5.5.1\_node3/logs

bootstrap.memory\_lock: false

bootstrap.system\_call\_filter: false

network.host: 0.0.0.0

#节点端口 各节点必须不一致

http.port: 9202

#节点通信端口 各节点必须不一致

transport.tcp.port: 9302

#可被选为主节点的集合 必须写通信端口

discovery.zen.ping.unicast.hosts:["192.168.0.46:9300","192.168.0.46:9301","192.168.0.46:9302"]

#集群最多节点个数

node.max\_local\_storage\_nodes: 3

#允许外网访问

http.cors.enabled: true

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

### 依次启动观察启动信息

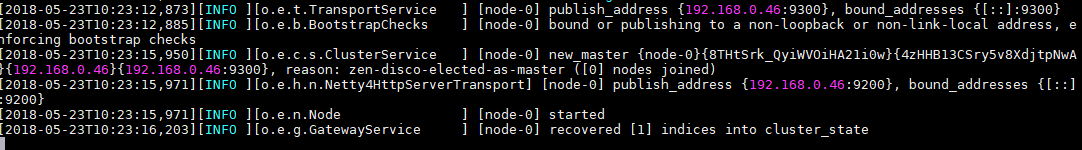
切换用户：

[root@localhost data]# su elasticsearch

[elasticsearch@localhost data]# cd elasticsearch-5.5.1

[elasticsearch@localhost data]# ./bin/elasticsearch

观察启动信息：



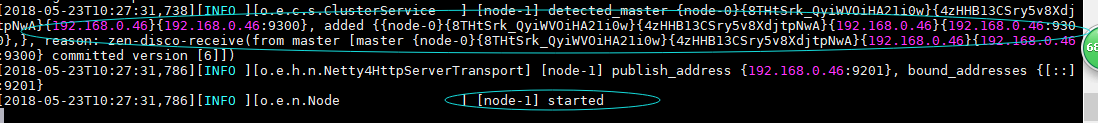
说明node-0启动成功 暂时没有其他节点加入集群

开启新窗口继续启动节点node-1：

[elasticsearch@localhost data]# cd elasticsearch-5.5.1\_node2

[elasticsearch@localhost data]# ./bin/elasticsearch

观察启动信息：



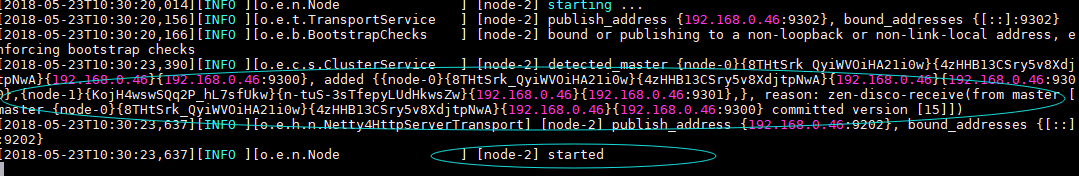
说明node1启动成功 成功加入集群

开启新窗口继续启动节点node-2：

[elasticsearch@localhost data]# cd elasticsearch-5.5.1\_node3

[elasticsearch@localhost data]# ./bin/elasticsearch

观察启动信息：



说明node2启动成功 成功加入集群

### 验证集群是否成功

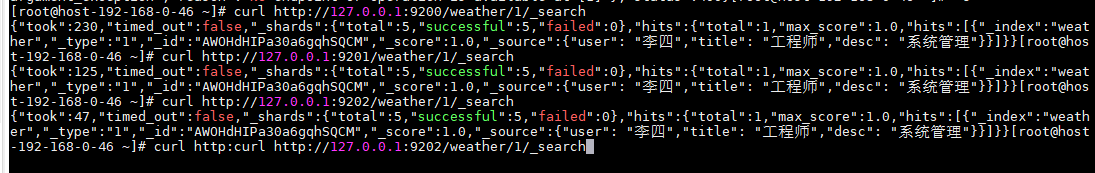
在某个节点中创建索引

[root@localhost data]# curl -X PUT 'localhost:9200/weather'

在索引中添加数据

[root@localhost data]# curl -X POST 'localhost:9200/weather/1' -d '{"user": "李四","title": "工程师","desc": "系统管理"}'

通过另外节点查询数据



三个节点都能查到数据说明集群成功