

Instalación y Configuración

domingo, 16 de noviembre de 2014 12:45

Instalación de Elasticsearch

Instalación de la Máquina Virtual Java (JVM)

ElasticSearch está construido sobre Java y requiere que tengamos configurado Java 7 en nuestra máquina. Por ello, debemos asegurarnos que la máquina virtual de Java está descargada e instalada correctamente en nuestro sistema. Ejecutando la siguiente línea de comando en un terminal podremos saber si lo está y que versión del software tenemos instalada:

```
$ java -version
```

El terminal te debe devolver algo parecido a esto:

```
java version "1.8.0"
Java(TM) SE Runtime Environment (build 1.8.0-b132)
Java HotSpot(TM) 64-Bit Server VM (build 25.0-b70, mixed mode)
```

Si el comando retorna algo similar podemos asegurar que tenemos la máquina virtual bien instalada y podemos movernos al siguiente paso. En caso contrario, debemos proceder a instalar Java en nuestra máquina local antes de instalar ElasticSearch. Para ello, ejecutamos los siguientes pasos en nuestra máquina:

1. Añadimos los repositorios con:

```
$ sudo add-apt-repository ppa:webupd8team/java
```
2. Actualizamos los repositorios con:

```
$ sudo apt-get update
```
3. Instalamos la versión de Java que necesitamos:

```
$ sudo apt-get install oracle-java7-installer
```

Para comprobar que la instalación ha tenido éxito, ejecutamos de nuevo el comando:

```
$ java -version
```

Configuración de la estructura de directorios de trabajo

Para tener un espacio de trabajo homogéneo y ordenado, vamos a crear una estructura de directorios que nos permita tener clasificado el SW que vamos a descargar así como los logs y los ficheros de datos que generemos durante los ejercicios. Para ello, debemos crear la siguiente estructura de directorios en nuestra máquina:

```
/ciff
/ciff/data
/ciff/exercises
/ciff/runtime
/ciff/software
/ciff/tarballs

$ mkdir ./ciff; mkdir ./ciff/data; mkdir ./ciff/exercises; mkdir ./ciff/runtime;
mkdir ./ciff/software; mkdir ./ciff/tarballs
```

Instalación de Elasticsearch

Instalando Elasticsearch desde el fichero comprimido

Las tareas que tenemos que realizar son muy sencillas: descargar Elasticsearch, mover archivo comprimido de Elasticsearch donde queramos realizar la instalación (ciff/software en nuestro caso), descomprimir y crear un enlace simbólico:

```
$ cd ~/ciff/tarballs
$ wget https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.5.1.tar.gz
$ mv ~/ciff/tarballs/elasticsearch-1.5.1.tar.gz ~/ciff/software/
$ cd ~/ciff/software
$ tar xvf elasticsearch-1.5.1.tar.gz
$ ln -s ~/ciff/software/elasticsearch-1.5.1 ~/ciff/runtime/elasticsearch
```

Instalando Elasticsearch desde el fichero .deb

En este otro caso, basta con descargar el paquete .deb de ES y ejecutar el dpkg correspondiente:

```
$ wget https://download.elasticsearch.org/elasticsearch/elasticsearch/elasticsearch-1.5.1.deb
$ sudo dpkg -i elasticsearch-1.5.1.deb
```

Carpetas a tener en cuenta

Si instalamos manualmente

Siempre

- elasticsearch/bin: Scripts necesarios para ejecutar Elasticsearch y para gestión de plugins
- elasticsearch/config: Ficheros de configuración
- elasticsearch/lib: Librerías usadas por Elasticsearch
- elasticsearch/plugins: Almacén de los plugins instalados

Modificables por configuración

elasticsearch/data: Almacén de datos utilizados por Elasticsearch
elasticsearch/logs: Ficheros con información de eventos y errores

Si lo hacemos desde el deb package

Siempre

```
/usr/share/elasticsearch/  
  bin: Scripts necesarios para ejecutar Elasticsearch y para gestión de plugins  
  lib: Librerías usadas por Elasticsearch  
  plugins: Almacén de los plugins instalados  
/etc/elasticsearch  
  Ficheros de configuración  
/var/lib/elasticsearch/data  
  Datos  
/var/log/elasticsearch  
  Logs
```

Modificables por configuración

data: Almacén de datos utilizados por Elasticsearch
logs: Ficheros con información de eventos y errores

Configuración Básica de Elasticsearch

Una vez hayamos instalado el código de Elasticsearch en nuestro sistema, debemos proceder a configurar mínimamente el motor de búsqueda antes de arrancarlo. Para ello, abrimos en modo edición el siguiente fichero y procedemos a agregar/modificar las líneas que se muestran a continuación:

```
$ nano /ciff/runtime/elasticsearch/conf/elasticsearch.yml  
cluster.name: "POC[1-N]"  
node.name: "VRTVW[1-N]"  
path.data: /home/adminuser/ciff/runtime/elasticsearch/data  
path.work: /home/adminuser/ciff/runtime/elasticsearch/tmp  
path.logs: /home/adminuser/ciff/runtime/elasticsearch/log  
network.host: [UseYourIP]  
http.port: 9200
```

También podemos configurar cómo Elasticsearch genera logs en el sistema de ficheros modificando el fichero config/logging.yml.

- Información a mostrar en los logs
- Generación periódica de logs
- Útil para soluciones de backup o monitorización

A tener en cuenta

Límite de ficheros abiertos

ElasticSearch, durante las indexaciones, poseyendo gran cantidad de shards y réplicas, es imprescindible mantener por encima de 32000 el límite de ficheros abiertos. Si queremos conocer cuál es el límite de ficheros abiertos que tenemos actualmente configurado en nuestro entorno haremos uso del comando

```
$ ulimit
```

Si queremos modificar el límite de ficheros abiertos podremos hacerlo actualizando el fichero:

```
/etc/security/limits.conf  
elasticsearch - nofile 65535  
elasticsearch - memlock unlimited
```

Gestión de memoria de ES

ES está desarrollado en Java, por lo tanto, debemos comprender muy bien cómo gestionamos la memoria del sistema de búsqueda si no queremos producir un impacto grave en la máquina donde se está ejecutando el servicio. Los límites en consumo de memoria han de proporcionarse en el fichero de configuración de ES.

Performance de IO

ES hace un uso intensivo del acceso a disco. Es muy importante para entornos de producción hacer uso de diferentes discos para los ficheros de Log y para los ficheros de datos.

Arrancando los servicios de Elasticsearch

Ejecución manual del Servidor

Podemos verificar que Elasticsearch ha sido correctamente configurado arrancando manualmente el motor de búsqueda. Para ello, nos vamos al directorio /ciff/runtime donde hemos deployado el SW. Una vez allí, podemos arrancar Elasticsearch con la siguiente línea de comando:

```
$ ./bin/elasticsearch
```

Tras lo cual, deberíamos ver algo así en nuestro terminal:

```
[2012-03-03 12:36:15,327][INFO ][node] [Ultra-Marine] {0.19.0}[9097]: initializing ...  
[2012-03-03 12:36:15,337][INFO ][plugins] [Ultra-Marine] loaded [], sites []  
...  
[2012-03-03 12:36:20,509][INFO ][http] [Ultra-Marine]  
bound address{inet[/0:0:0:0:0:0:0:0:9200]}, ...  
[2012-03-03 12:36:20,509][INFO ][node] [Ultra-Marine] {0.19.0}[9097]: started  
[2012-03-03 12:36:20,511][INFO ][gateway] [Ultra-Marine] recovered [0] indices into cluster_state
```

Verificando que el servidor está vivo

Ping al Servidor

Para verificar que Elasticsearch está escuchando, bastará con ejecutar un GET sobre la URL <http://localhost:9200> y verificar que recibimos algo como

lo siguiente:

```
$ curl http://localhost:9200/?pretty
{
  "status": 200,
  "name": "POC",
  "version": {
    "number": "1.4.0",
    "lucene_version": "4.10"
  },
  "tagline": "You Know, for Search"
}
```

El Cluster API

Salud del cluster

```
$ curl -XGET 'http://localhost:9200/_cluster/health?pretty=true'
$ curl -XGET 'http://localhost:9200/_cluster/health/[Nodename]?pretty=true'
```

Información de estado del cluster

```
$ curl -XGET 'http://localhost:9200/_cluster/state?pretty=true'
```

Estadísticas sobre los números del cluster

```
$ curl -XGET 'http://localhost:9200/_cluster/stats?human&pretty'
```

Parada del Servidor

Si estamos trabajando en local

Parar el servicio de Elasticsearch es muy sencillo. Bastará con hacer un CTRL+C en el terminal donde lo hayamos arrancado y el servidor se cerrará automáticamente. Elasticsearch ha sido programado para ser robusto y fiable. Por ello, incluso en situaciones donde hagamos una parada forzosa o el sistema operativo caiga, Elasticsearch garantizará que los índices y los datos no queden corruptos.

Si estamos trabajando en remoto

Esta opción será muy útil cuando no tengamos acceso al hilo que ejecuta el servidor o no queramos estar lanzando sentencias kill a mano. Elasticsearch, a través del Cluster API, proporciona un recurso REST

Parada del Cluster entero

También podemos parar el servidor haciendo uso del shutdown que proporciona el Cluster API:

```
$ curl -XPOST 'http://localhost:9200/_shutdown?pretty=true'
```

Parada de un nodo

```
$ curl -XPOST 'http://localhost:9200/_cluster/nodes/_local/_shutdown'
$ curl -XPOST 'http://localhost:9200/_cluster/nodes/nodeId1,nodeId2/_shutdown'
```

Ejecución del servidor como servicio

Instalación manual - Usando un gestor de servicios

Con el formato de ejecución del paso anterior, Elasticsearch se ejecutará en el terminal en modo foreground. Si queremos hacerlo en modo background podemos hacerlo con la siguiente línea de comando:

```
$ bin/elasticsearch -d
```

Aun así, la mejor forma de manejar Elasticsearch es instalándolo como un servicio. En el ejemplo que utilizaremos durante el curso, haremos uso de supervisor como herramienta gestora del servicio. Hay muchos otros service managers en el mundo Linux, podemos utilizar cualquiera.

```
$ apt-get install supervisor
$ service supervisor restart
$ cd /etc/supervisor/conf.d
$ nano elasticsearch.conf
[program:elastic]
command=/ciff/runtime/elasticsearch/bin/elasticsearch
autostart=true
autorestart=false
stderr_logfile=/ciff/data/elasticsearch/log/supervisord-process-elasticsearch-error.log
stdout_logfile=/ciff/data/elasticsearch/log/supervisord-process-elasticsearch-out.log
$ supervisorctl reread
$ supervisorctl update
```

Una vez instalado y configurado supervisor, podemos arrancar o parar el servicio de Elasticsearch con la siguiente línea de comando

```
$ supervisorctl start elastic
$ supervisorctl stop elastic
$ supervisorctl restart elastic
```

Instalación manual - Usando un wrapper

El equipo de desarrollo de ES ha creado un wrapper que permite configurar Linux para gestionar ES como servicio estándar del sistema. El wrapper está disponible en esta [URL](http://github.com/elasticsearch/elasticsearch-servicewrapper/tarball/master) y podemos descargarlo y configurarlo en nuestro sistema ejecutando el siguiente script:

```
$ curl -L http://github.com/elasticsearch/elasticsearch-servicewrapper/tarball/master | tar -xz
```

```
$ sudo mkdir /usr/local/share/elasticsearch
$ sudo mkdir /usr/local/share/elasticsearch/bin
$ sudo mv *servicewrapper*/service /ciff/runtime/elasticsearch/bin/
$ rm -Rf *servicewrapper*
$ sudo /ciff/runtime/elasticsearch/bin/service/elasticsearch install
$ sudo ln -s 'readlink -f /ciff/runtime/elasticsearch/bin/service/elasticsearch'
/usr/local/bin/rcelasticsearch
$ sudo service elasticsearch start
O
$ sudo /etc/init.d/elasticsearch start
```

Ejecución como servicio standard

Si hemos hecho instalación por paquete debian

```
$ sudo /etc/init.d/elasticsearch start
$ sudo /etc/init.d/elasticsearch stop
```

Si queremos que Elasticsearch arranque con la máquina, ejecutamos...

```
$ sudo update-rc.d elasticsearch defaults 95 10
```

¿Y cómo vemos ahora lo que pasa?

ElasticSearch genera ficheros donde podemos ver los logs del servidor según este se va ejecutando. Los logs pueden ser más o menos detallados en función de cómo los configuremos haciendo uso del fichero logging.yml.

Si hemos arrancado haciendo uso de la instalación de deb:

/var/log/elasticsearch

Si lo hemos hecho haciendo uso del proceso manual

elasticsearch/logs