Elasticsearch PHP API: No alive nodes. All the 1 nodes seem to be down.

Introducción

Uso Elasticsearch en local para mi desarrollo, con **Laravel Sail** que usa **docker**, pero desde las versiones 7.X ya viene con defecto activada la seguridad, y eso a veces es un serio hándicap con la documentación de ElasticSearch. Ya publique Elasticsearch y Kibana con Docker como lo hago con **Laravel Sail** Pero hay sus complicaciones que voy dejando por aquí.

Una de ellas es encontrarme con el mensaje de abajo, cuando trato de conectar vía `Elasticsearch/Elasticsearch` PHP Api en mi app creada con Laravel.

```
Elastic\Transport\Exception\NoNodeAvailableException


 No alive nodes. All the 1 nodes seem to be down.
```

Escenario

Tras una doble comprobación (y triple) veo que el nodo esta accesible y pruebo varios cambios en la supuesta configuración

> Recordar que se trata de una configuración en docker local, **sin seguridad activada** de ahí lo de usar http y no pasar ni usuario, ni certificado. Os aviso para que no perdáis el tiempo si el escenario es otro.

```
❯ curl -XGET "http://localhost:9200/"
{
```

```
  "name" : "44edbbb60101",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "6seEH0VRR8mX98dIkzSySg",
  "version" : {
    "number" : "8.5.0",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "c94b4700cda13820dad5aa74fae6db185ca5c304",
    "build_date" : "2022-10-24T16:54:16.433628434Z",
    "build_snapshot" : false,
    "lucene_version" : "9.4.1",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Sin embargo al ejecutar el cliente

```
 Elastic\Transport\Exception\NoNodeAvailableException

  No alive nodes. All the 1 nodes seem to be down.

  at vendor/elastic/transport/src/NodePool/SimpleNodePool.php:77
     73 |             }
     74 |             $dead++;
     75 |         }
     76 |
  ➜  77 |         throw new NoNodeAvailableException(sprintf(
     78 |             'No alive nodes. All the %d nodes seem to be down.',
     79 |             $totNodes
     80 |         ));
     81 |     }
```

La documentación nos dice.

```
$hosts = [
    '192.168.1.1:9200',         // IP + Port
    '192.168.1.2',              // Just IP
```

```
    'mydomain.server.com:9201', // Domain + Port
    'mydomain2.server.com',     // Just Domain
    'https://localhost',        // SSL to localhost
    'https://192.168.1.3:9200'  // SSL to IP + Port
];
```

Bien, en mi configuracion uso siempre el protocolo https, (en producción) y por extensión configure mi cliente usando la misma metodología, `http://localhost`, probando con `http://127.0.0.1`, etc.

Lo curioso es que haciendo un debug, y volcando la salida del cliente creado, me indicaba que estaba **alive**

```
-nodePool: Elastic\Transport\NodePool\SimpleNodePool {#2856 ▼
      #nodes: array:1 [▼
        0 => Elastic\Transport\NodePool\Node {#2859 ▼
          #uri: GuzzleHttp\Psr7\Uri {#2861 ▼
            -scheme: "http"
            -userInfo: ""
            -host: "127.0.0.1"
            -port: 9200
            -path: ""
            -query: ""
            -fragment: ""
            -composedComponents: null
          }
          #alive: true
        }
      ]
```

La solución está en la línea (y la documentación de Elasticsearch que es un poco espesa y deficitaria en muchos sitios). Se trata de no poner el protocolo cuando usemos `http` en lugar de `https`

```
ClientBuilder::create()
   ->setHosts(['http://<name-of-node-elasticsearch>:9200'])
   ->build();
```

Obtener el nombre del container

```
 docker container ls
CONTAINER ID   IMAGE                                                  COMMAND
CREATED       STATUS                 PORTS                                                 NAMES
652c6ecb63b9   sail-8.1/app                                           "start-container"
4 hours ago   Up 4 hours             0.0.0.0:80->80/tcp, 0.0.0.0:5173->5173/tcp, 8000/tcp   sitelight-laravel.test-1
42370b75132d   docker.elastic.co/kibana/kibana:8.5.0                  "/bin/tini -- /usr/l…"
4 hours ago   Up 4 hours             0.0.0.0:5601->5601/tcp                                sitelight-kibana
cf0da3198b67   mysql/mysql-server:8.0                                 "/entrypoint.sh mysq…"
4 hours ago   Up 4 hours (healthy)   0.0.0.0:3306->3306/tcp, 33060-33061/tcp               sitelight-mysql-1
9e89cfc0e2ff   docker.elastic.co/elasticsearch/elasticsearch:8.5.0   "/bin/tini -- /usr/l…"
4 hours ago   Up 4 hours             0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp        sitelight-es01
9e2be014d4a6   redis:alpine                                           "docker-entrypoint.s…"
4 hours ago   Up 4 hours (healthy)   0.0.0.0:6379->6379/tcp                                sitelight-redis-1
```

El nombre tambien esta en la defición que se hizo en el fichero `docker-compose.yml`

```
    elasticsearch:
        image: 'docker.elastic.co/elasticsearch/elasticsearch:8.5.0'
        container_name: sitelight-es01
        environment:
```

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Revision #4
Created 2022-11-10 07:33:56 UTC by Abkrim
Updated 2022-11-10 18:03:43 UTC by Abkrim