Elasticsearch gRPC health check in Go

Specifications

--------------------

Your task is to create a Go microservice that monitors the health of an Elastic Search cluster ([https://www.elastic.co/fr/downloads/elasticsearch](https://www.elastic.co/fr/downloads/elasticsearch" \t "_blank) ). The end goal is to use this microservice as a monitoring tool.

* The microservice should provide gRPC endpoints giving enough information to determine **the health and the state** of the cluster and its indices.
* The code should rely as much as possible on the standard Go library (avoid fancy libraries).
* Bonus: packaging of the full solution using a single helm chart

# Plan

~~Part 1) - Set up VS Code for Go~~

* ~~Read about Go specifications~~

~~Part 2) Create schematics of the project~~

~~Part 3) Create git repo for project~~

Part 4) Create README.md and document each part

~~Part 5) Install + set up Elastic Search cluster~~

* ~~Read about Elastic search cluster (shards, indices etc…)~~

~~Part 6) Create Go class/code to retrieve health status of Elastic Search cluster~~

* ~~Take 2 arguments : (Hostname, Port) of cluster~~
* ~~Create Go unittests for health check~~
* ~~Unittest for argument~~

~~Part 7) Create Go class/code to create gRPC endpoints (vanilla check)~~

* ~~Create Go unittests for gRPC~~
* [~~https://www.grpc.io/docs/languages/go/basics/~~](https://www.grpc.io/docs/languages/go/basics/)
* [~~https://bitbucket.org/blog/writing-a-microservice-in-golang-which-communicates-over-grpc~~](https://bitbucket.org/blog/writing-a-microservice-in-golang-which-communicates-over-grpc)

~~Part 8) Combine gRPC endpoint with Elastic search health check Go program (Part 4 + Part 5)~~

~~Part 9) Wrap the solution in Docker image or Helm chart~~

~~Part 10) Document / How to use~~

Curl Command to get health check : curl -X GET "localhost:9200/\_cat/health?v&pretty"

Curl command to add indice :

curl -X PUT "localhost:9200/my-index-000001?pretty" -H 'Content-Type: application/json' -d'

{

"settings": {

"number\_of\_shards": 3,

"number\_of\_replicas": 2

}

}

'

# Questions

Restrictions on versions of Go, Elastic search cluster, gRPC?

Golang module : Use elasticsearch module or only cURL (http module) ?

Use SSL ?

# Improvements

Periodic call

Alert message (email, slack)

Log real time info, + previous info (1hour, 1day)

docker build -t kevinplltr/elastic-health:0.0.1 .

docker run -p 9000:9000 kevinplltr/elastic-health:0.0.1 host.docker.internal 9200

Docker file : https://hub.docker.com/repository/docker/kevinplltr/elastic-health

# Schematics

ElasticSearch cluster

gRPC endpoint

Health endpoint

Get health from cluster Go code

gRPC Go code