**Pre-Requisites**

We need to set the vm.max\_map\_count kernel parameter:



To set this permanently, add it to /etc/sysctl.conf and reload with sudo sysctl -p

## Docker Compose:

The docker compose file that we will reference:

Docker-compose.yml

The data of our elasticsearch container volumes will reside under /var/lib/docker, if you want them to persist in another location, you can use the driver\_opts setting for the local volume driver.

## Deploy

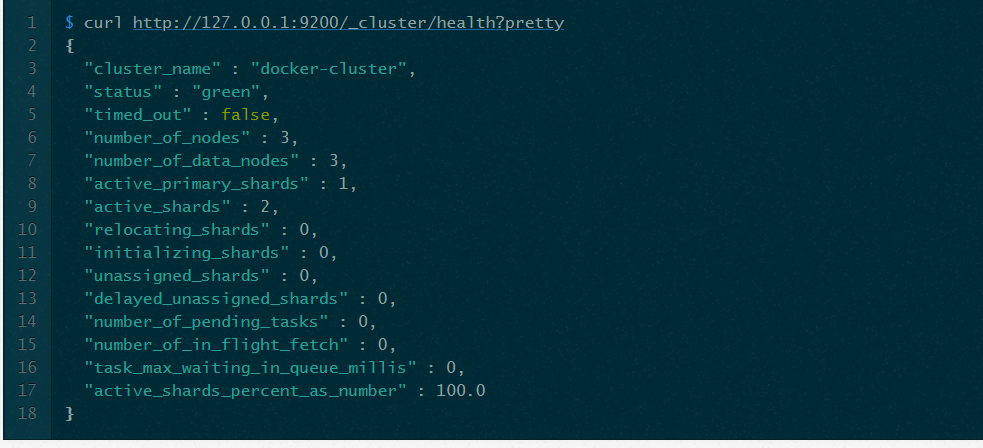
Deploy your elasticsearch cluster with docker compose:



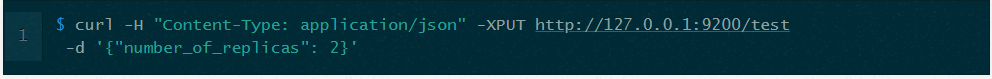
This will run in the foreground, and you should see console output.

## Testing Elasticsearch

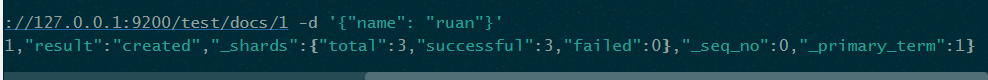
Let’s run a couple of queries, first up, check the cluster health api:



Create a index with replication count of 2:



Ingest a document to elasticsearch:

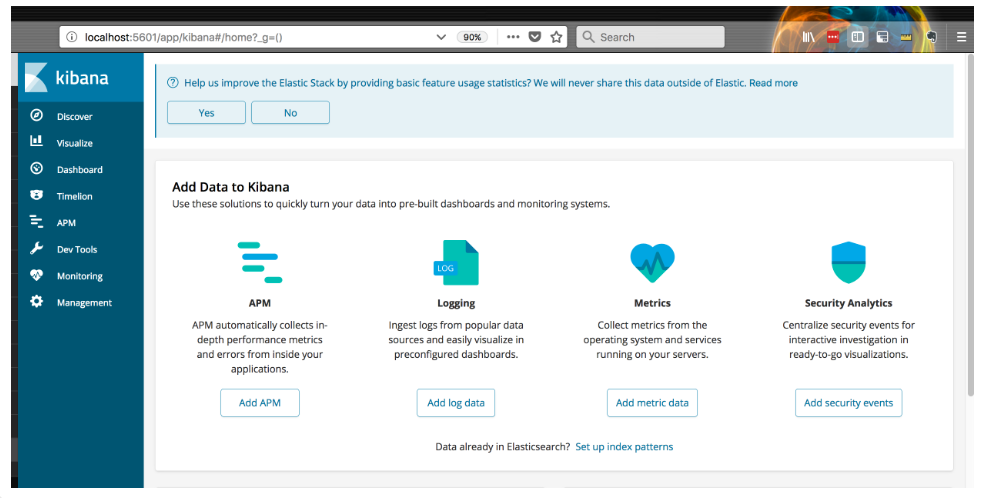


View the indices:

## 

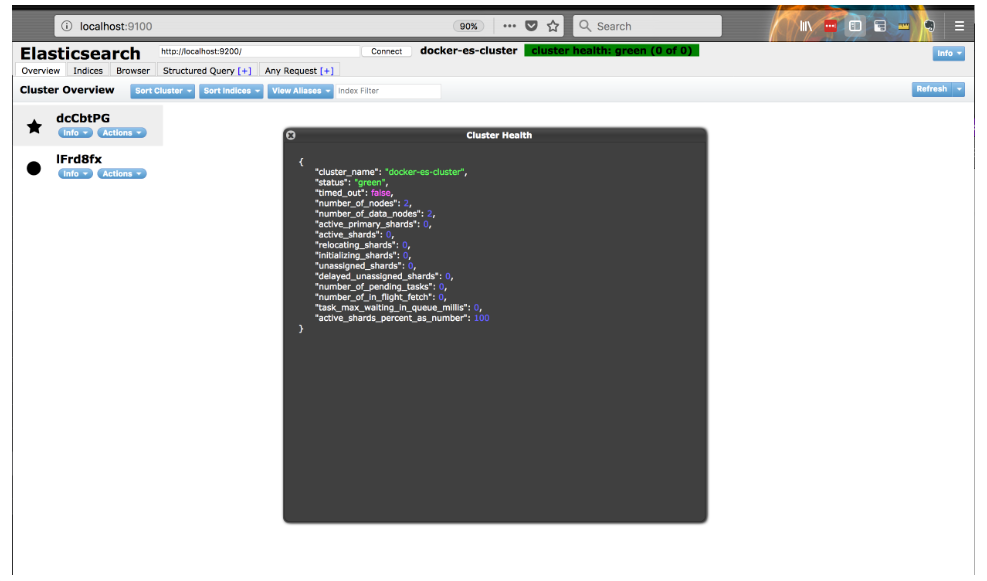
## Kibana

Kibana is also included in the stack and is accessible via <http://localhost:5601/> and you it should look more or less like:



## Elasticsearch Head UI

I always prefer working directly with the RESTFul API, but if you would like to use a UI to interact with Elasticsearch, you can access it via <http://localhost:9100/> and should look like this:



## Deleting the Cluster:

As its running in the foreground, you can just hit ctrl + c and as we persisted data in our compose, when you spin up the cluster again, the data will still be there.