

Q. how to recover deleted index in Elasticsearch?

Refer: <https://www.udemy.com/course/elasticsearch-and-elastic-stack-in-depth-and-hands-on/learn/lecture/7276298>

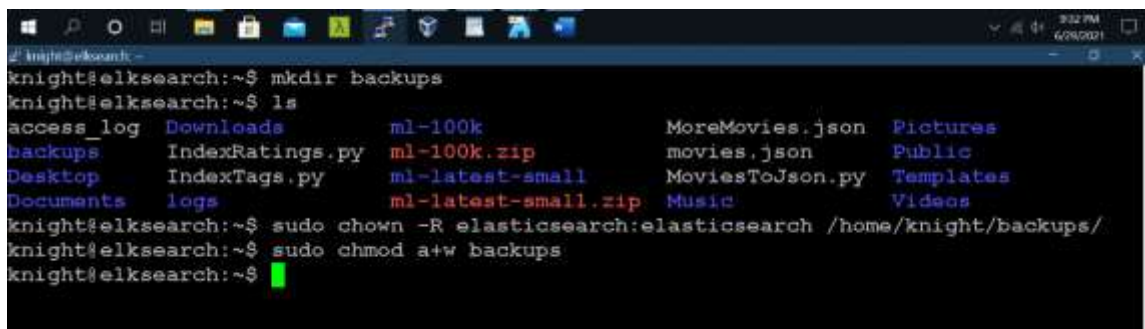
Step 1: Make backup directory on local machine i.e., /home/knight/backups

```
$mkdir backups
```

Step 2: Give permissions to write in that directory.

```
$ sudo chown -R elasticsearch:elasticsearch /home/knight/backups/
```

```
$sudo chmod a+w backups
```

A terminal window screenshot showing the following commands and output:

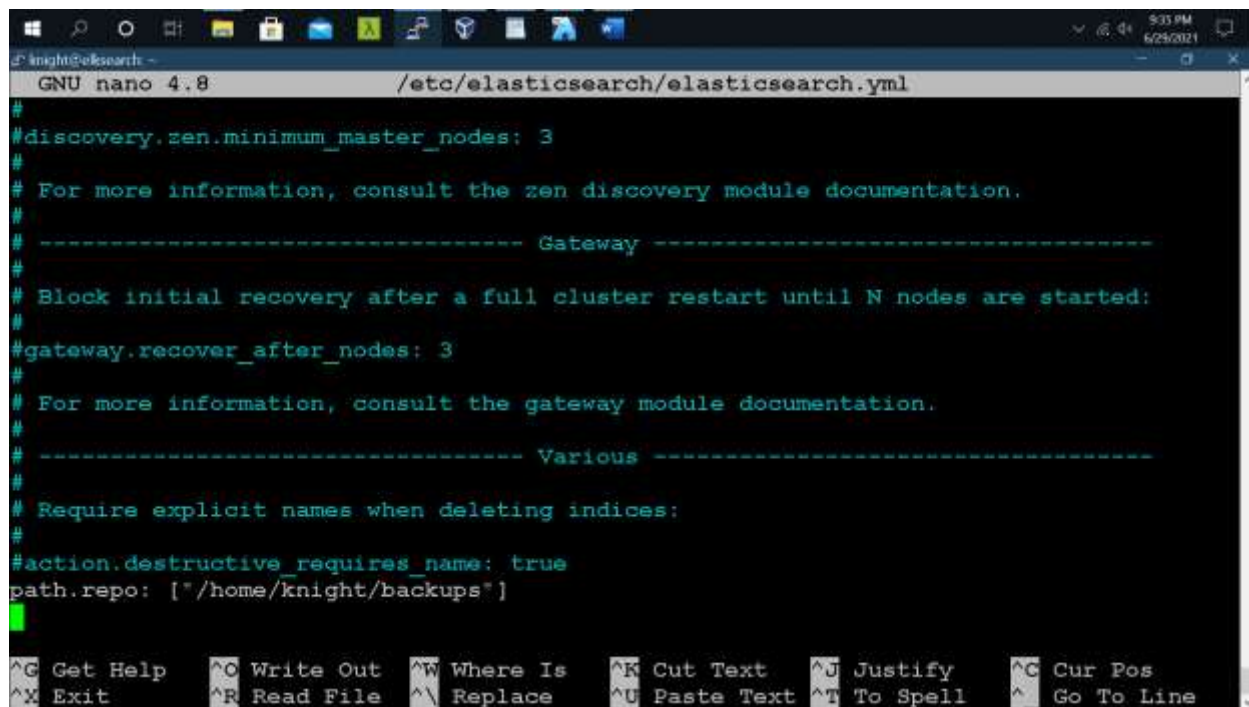
```
knight@elksearch:~$ mkdir backups
knight@elksearch:~$ ls
access_log  Downloads  ml-100k      MoreMovies.json  Pictures
backups     IndexRatings.py  ml-100k.zip  movies.json       Public
Desktop     IndexTags.py    ml-latest-small  MoviesToJson.py   Templates
Documents   logs           ml-latest-small.zip  Music            Videos
knight@elksearch:~$ sudo chown -R elasticsearch:elasticsearch /home/knight/backups/
knight@elksearch:~$ sudo chmod a+w backups
knight@elksearch:~$
```

Step 3: Make entry in the .yaml file of elasticsearch i.e., /etc/elasticsearch/elasticsearch.yaml

```
$ sudo nano /etc/elasticsearch/elasticsearch.yaml
```

Add following line at the end of the yaml file and save,

```
path.repo: ["/home/knight/backups"]
```



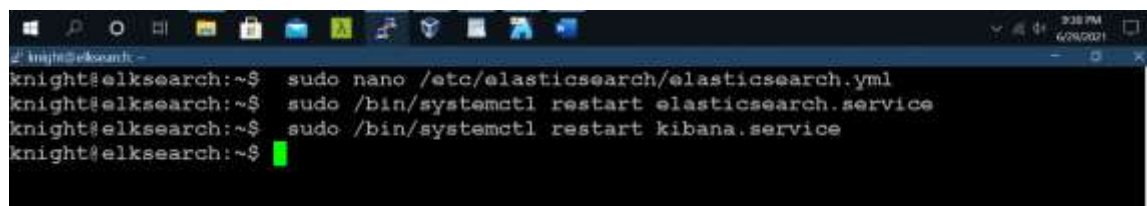
```
GNU nano 4.8 /etc/elasticsearch/elasticsearch.yml
#
#discovery.zen.minimum_master_nodes: 3
#
# For more information, consult the zen discovery module documentation.
#
# ----- Gateway -----
#
# Block initial recovery after a full cluster restart until N nodes are started:
#
#gateway.recover_after_nodes: 3
#
# For more information, consult the gateway module documentation.
#
# ----- Various -----
#
# Require explicit names when deleting indices:
#
#action.destructive_requires_name: true
path.repo: ["/home/knight/backups"]
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

Step 4: Restart the elasticsearch service

```
$ sudo /bin/systemctl restart elasticsearch.service
```

Step 5: restart kibana

```
$ sudo /bin/systemctl restart kibana.service
```



```
knight@elksearch:~$ sudo nano /etc/elasticsearch/elasticsearch.yml
knight@elksearch:~$ sudo /bin/systemctl restart elasticsearch.service
knight@elksearch:~$ sudo /bin/systemctl restart kibana.service
knight@elksearch:~$
```

Step 6: Now go to browser and open kibana – 127.0.0.1:5601 and open DevTool

Step 7: Register a snapshot repository before you can perform snapshot and restore operations.

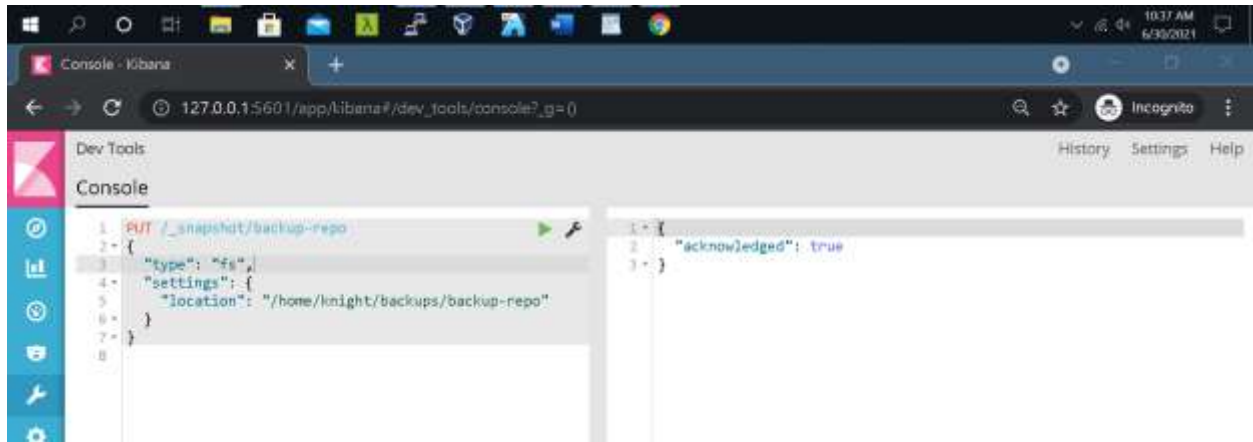
```
>>>PUT /_snapshot/backup-repo
```

```
{
  "type": "fs",
  "settings": {
```

```
"location": "/home/knight/backup/backup-repo"
```

```
}
```

```
}
```



Step 8: View all indices present in ES database

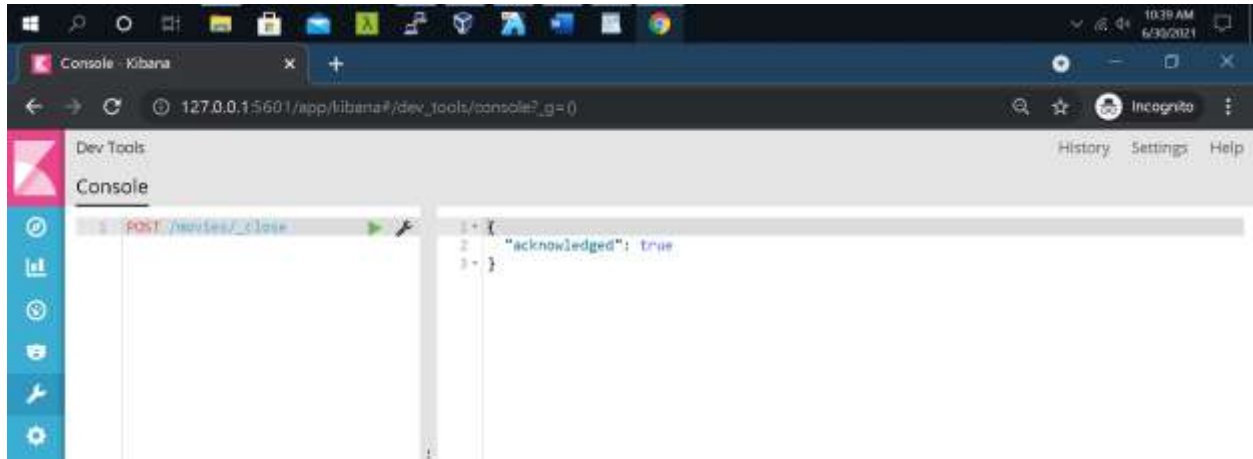
```
>>>GET /_cat/indices?v
```

The screenshot shows the Kibana Dev Tools Console with the output of the GET /_cat/indices?v command. The output is a table with the following columns: index, .count, docs, deleted, store.size, pri.store.size, uuid, pri, rep, and docs. The table lists various indices, including .monitoring-kibana-6, .watches, filebeat-2021.06.18, .watcher-history-6-2021.06.20, .movies, logstash-2017.05.01, logstash-2017.05.03, ratings, .kibana, tags, logstash-2017.05.05, and logstash-2017.04.30.

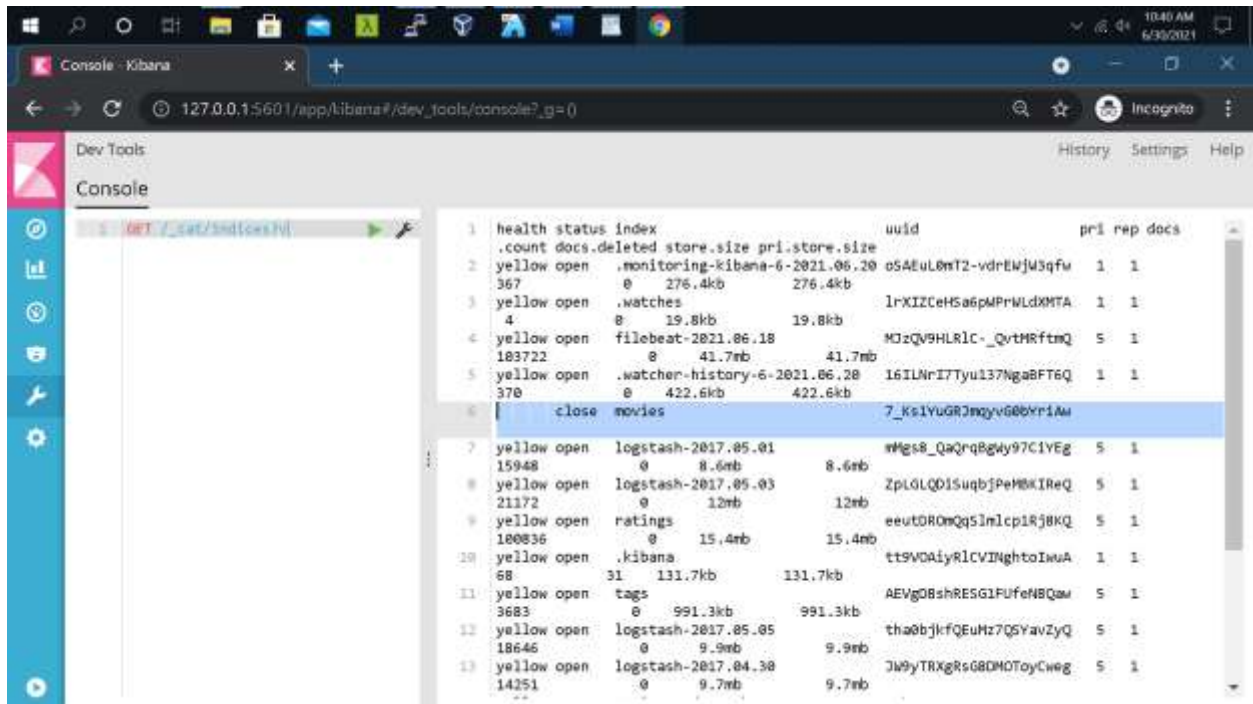
index	.count	docs	deleted	store.size	pri.store.size	uuid	pri	rep	docs
health status index									
1	367	0	276.4kb	276.4kb	05AEuL0MT2-vdrEWjU3qfw	1	1		
2	4	0	19.8kb	19.8kb	1rXIZCeH5a6pMPwLdXMTA	1	1		
3	183722	0	41.7mb	41.7mb	MJzQv9HLRlC-QvtMRftmQ	5	1		
4	370	0	422.6kb	422.6kb	16ILNrI7Tyu137Nga8FT6Q	1	1		
5	9742	0	1.9mb	1.9mb	7_Ks1YuGRJmzyvG0bYr1Aw	5	1		
6	15948	0	8.6mb	8.6mb	mMgs8_QaQrQ8gy97C1YEg	5	1		
7	21172	0	12mb	12mb	ZpLGLQD15uqbJPeMMKIReQ	5	1		
8	180836	0	15.4mb	15.4mb	eeutDR0mQq5Im1cp1Rj8KQ	5	1		
9	68	31	131.7kb	131.7kb	tt9VDAiyRlCVINghtoIkuA	1	1		
10	3683	0	991.3kb	991.3kb	AEVgDBshRESG1FufefN8Qaw	5	1		
11	18646	0	9.9mb	9.9mb	tha0bjk-fQEuMz7Q5YavZyQ	5	1		
12	14251	0	9.7mb	9.7mb	JW9yTRXgRsG8DM0ToYCweg	5	1		

Step 9: Now to test closing and opening of index, type following command

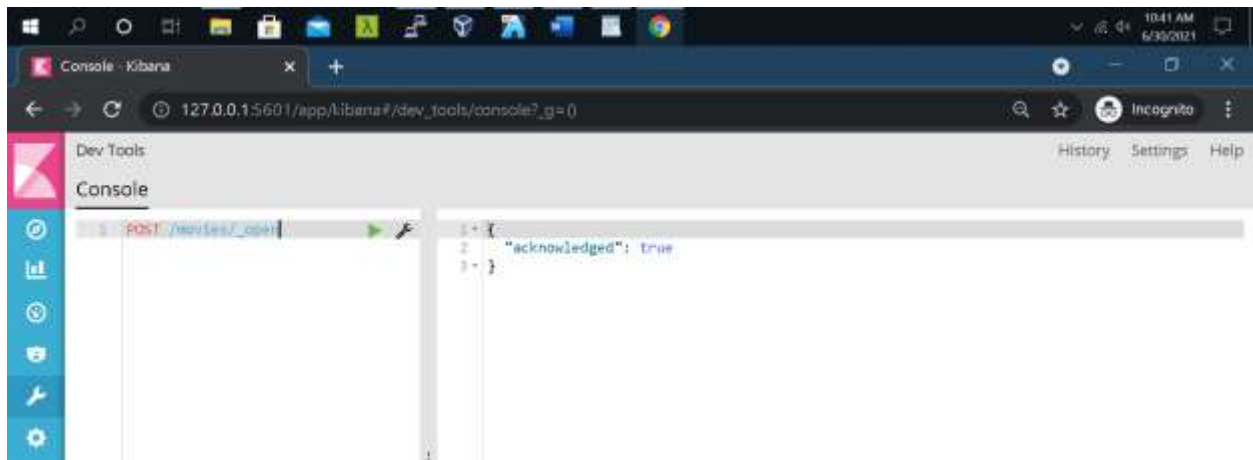
>>>POST /movies/_close



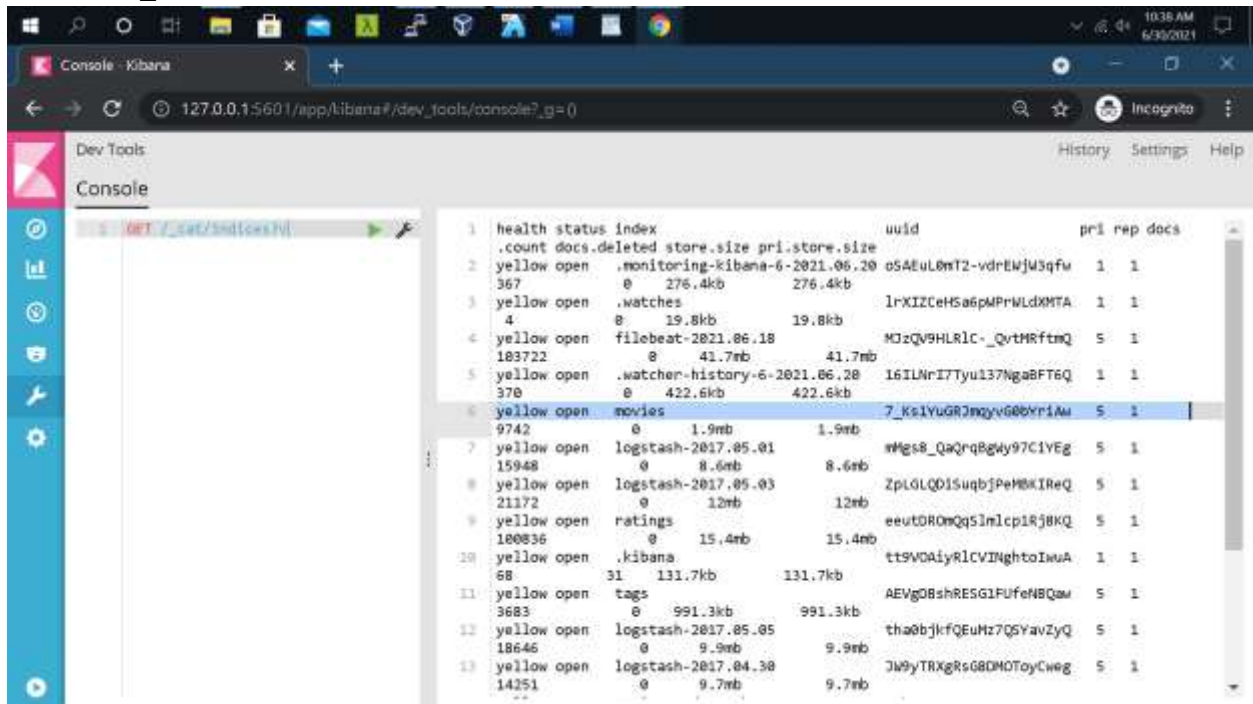
>>>GET /_cat/indices?v



>>>POST /movies/_open



>>>GET /_cat/indices?v



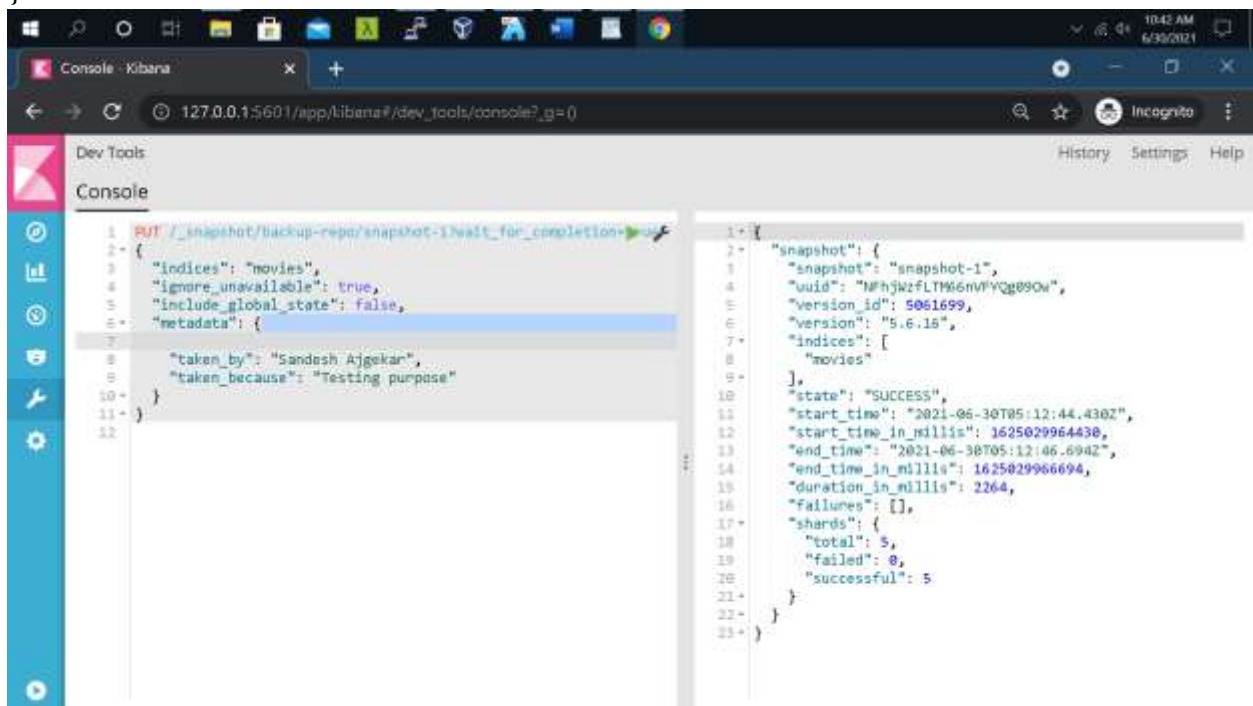
Refer: <https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-close.html#indices-close>

<https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-open-close.html>

Step 10: Now take snapshot of the open index

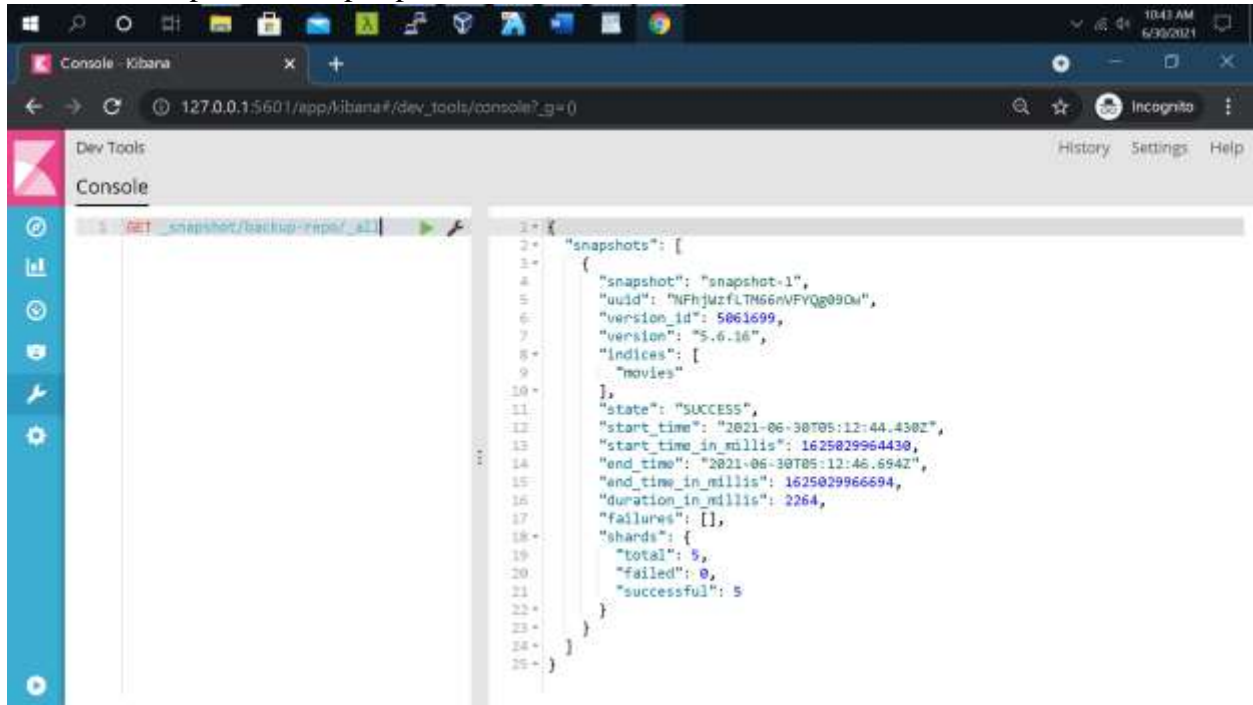
```
>>>PUT /_snapshot/backup-repo/snapshot-1?wait_for_completion=true
```

```
{
  "indices": "movies",
  "ignore_unavailable": true,
  "include_global_state": false,
  "metadata": {
    "taken_by": "Sandesh Ajgekar",
    "taken_because": "Testing purpose"
  }
}
```



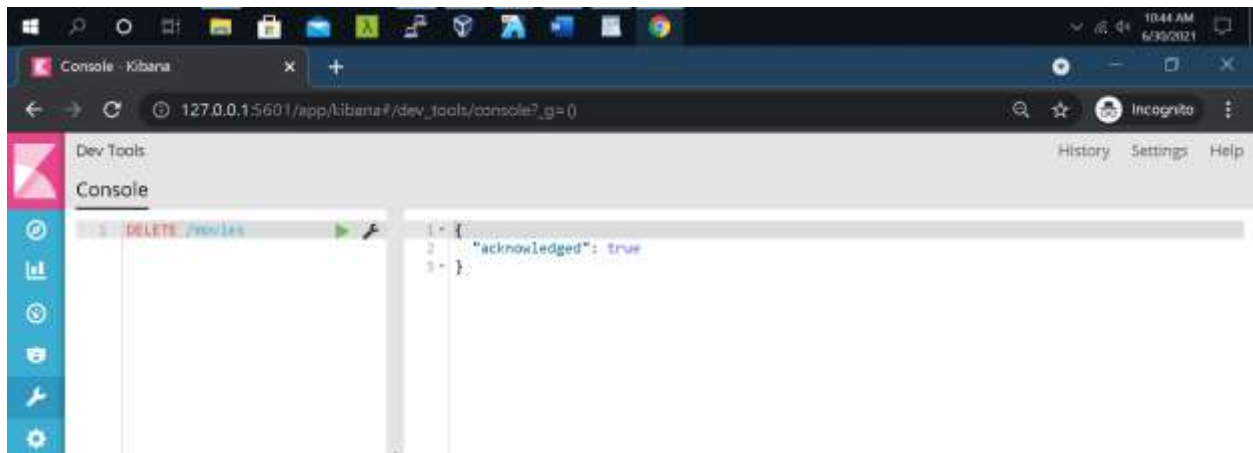
Step 11: To check snapshot, type following command

>>>GET _snapshot/backup-repo/_all

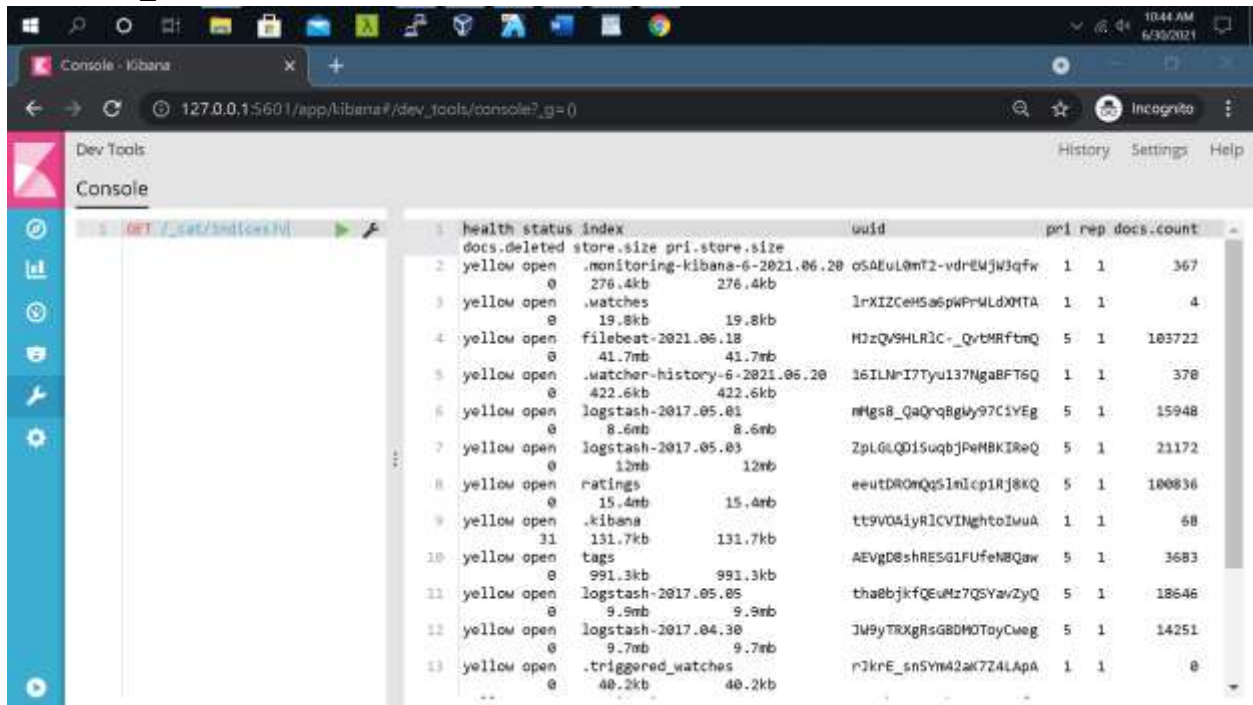


Step 12: now delete the index which snapshot have taken

>>>DELETE /movies



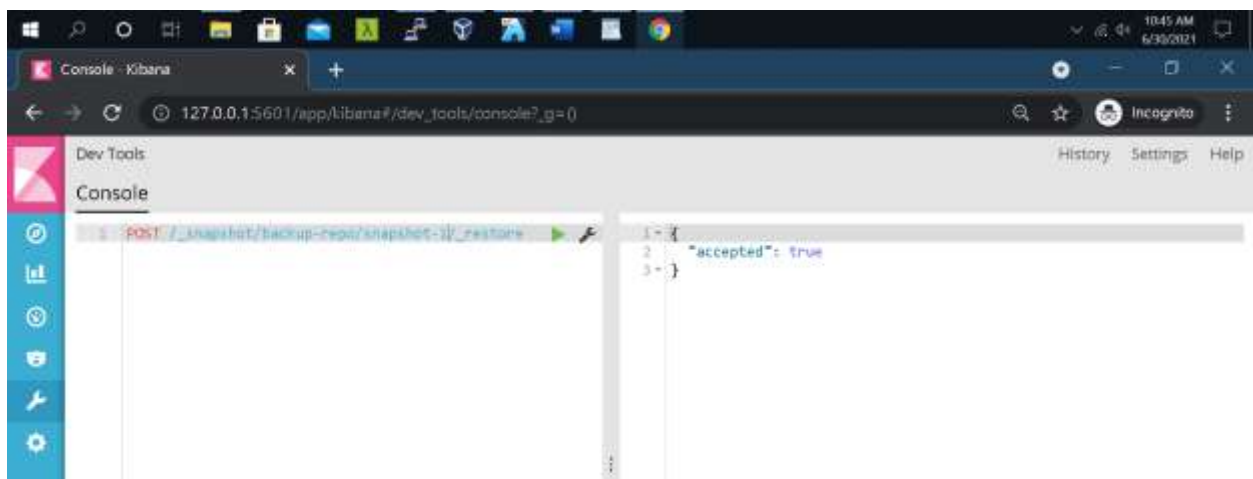
>>>GET /_cat/indices?v



	health	status	index	docs.deleted	store.size	pri	store.size	uuid	pri	rep	docs.count
1	yellow	open	monitoring-kibana-6-2021.06.20	0	276.4kb	0	276.4kb	oSAEuLQmT2-vdrEWJW3qfw	1	1	367
2	yellow	open	.watches	0	19.8kb	0	19.8kb	1rXIZCeH5a6pWPrWLDXMTA	1	1	4
3	yellow	open	filebeat-2021.06.18	0	41.7mb	0	41.7mb	MJzQV9HLR3C-QvbmRftmQ	5	1	103722
4	yellow	open	.watcher-history-6-2021.06.20	0	422.6kb	0	422.6kb	16ILNrI7Tyu137NgaBFT6Q	1	1	370
5	yellow	open	logstash-2017.05.01	0	8.6mb	0	8.6mb	nHgs8_QaQrQ8gWY97C1VEg	5	1	15948
6	yellow	open	logstash-2017.05.03	0	12mb	0	12mb	ZpLGLQD1SuqbJPmBKIReQ	5	1	21172
7	yellow	open	ratings	0	15.4mb	0	15.4mb	eeutDRomQq51n1cp1Rj8KQ	5	1	100836
8	yellow	open	.kibana	31	131.7kb	0	131.7kb	tt9VOAiyR1CVDInghtoImuA	1	1	68
9	yellow	open	tags	0	991.3kb	0	991.3kb	AEvgD8shRESG1FUFehBQaw	5	1	3683
10	yellow	open	logstash-2017.05.05	0	9.9mb	0	9.9mb	thaebjkfQeUHz7Q5YavZyQ	5	1	18646
11	yellow	open	logstash-2017.04.30	0	9.7mb	0	9.7mb	JW9yTRXgRsgBDMOToyCweg	5	1	14251
12	yellow	open	.triggered_watches	0	40.2kb	0	40.2kb	r7krE_sn5YM42ak7Z4LApA	1	1	0

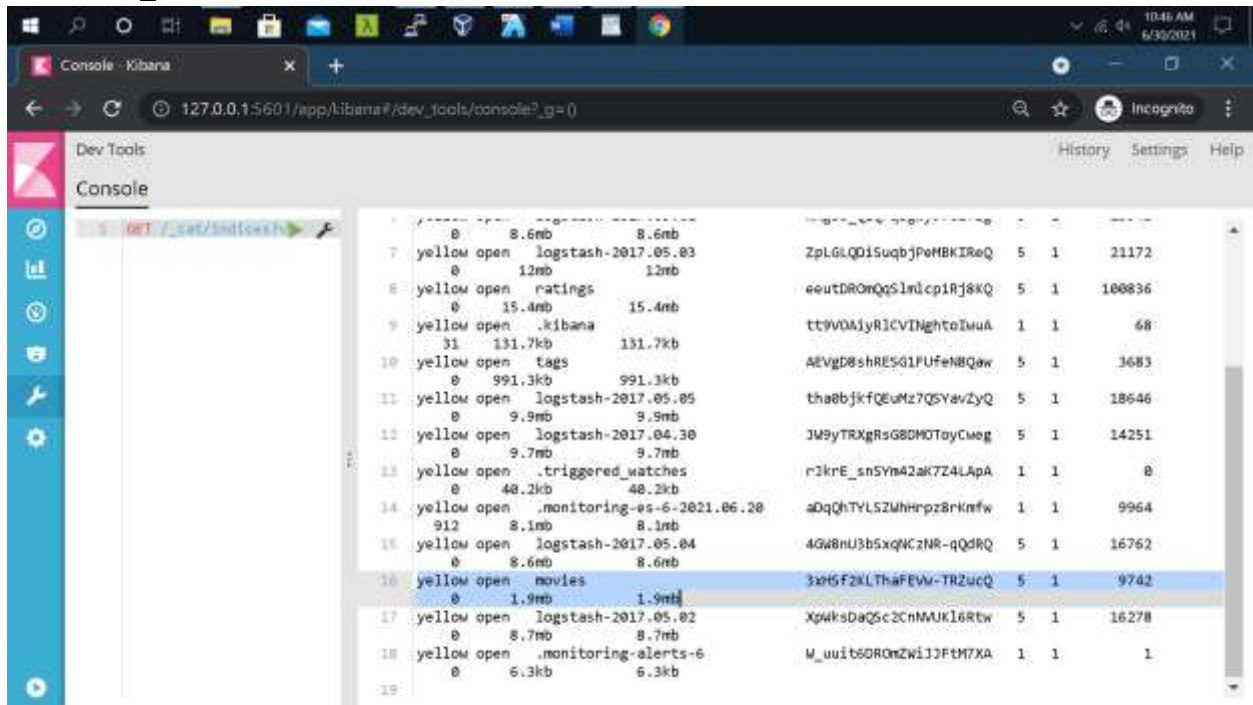
Step 13: now recover deleted index by restoring snapshot, type

>>>POST /_snapshot/backup-repo/snapshot-1/_restore



1	{
2	"accepted": true
3	}

>>>GET /_cat/indices?v



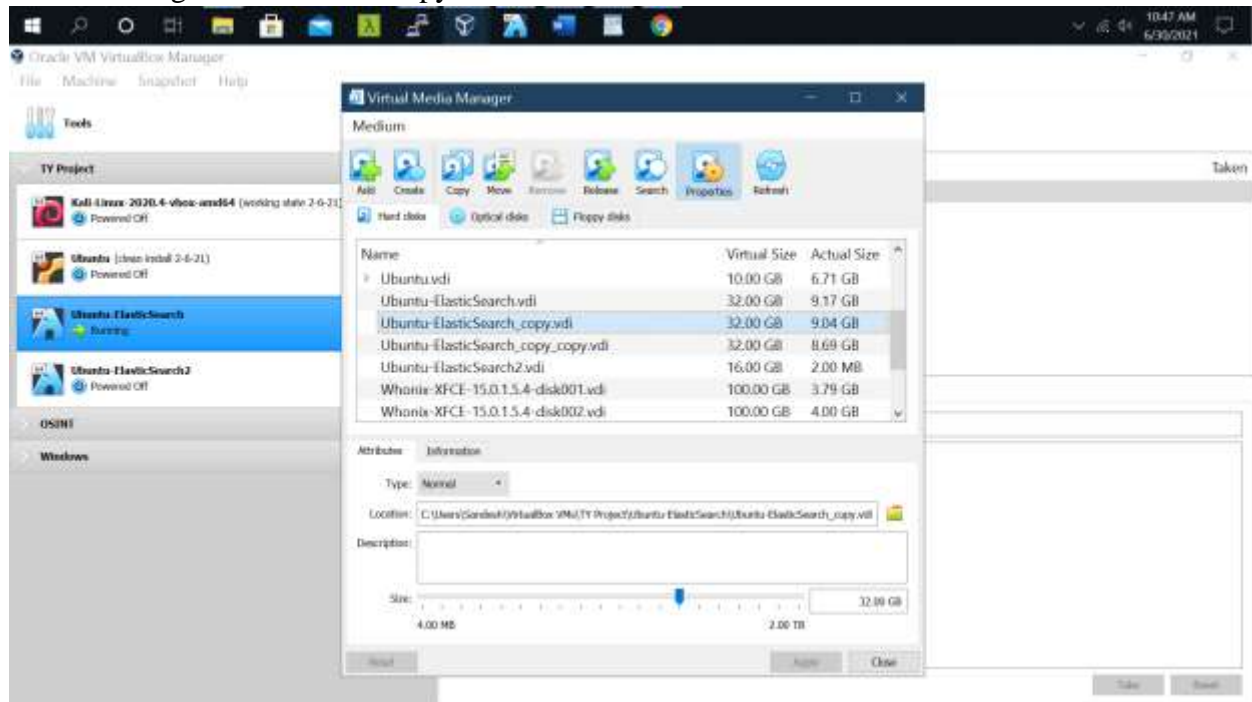
The screenshot shows a web browser window with the Kibana console open. The console displays the command `GET /_cat/indices?v` and its output, which is a table of Elasticsearch indices. The table has columns for index name, status, type, creation time, size in bytes, size in KB, size in MB, and a numeric value. The 'movies' index is highlighted in blue.

7	yellow	open	logstash-2017.05.03	0	8.6mb	8.6mb	21172
8	yellow	open	ratings	0	12mb	12mb	100836
9	yellow	open	.kibana	31	131.7kb	131.7kb	68
10	yellow	open	tags	0	991.3kb	991.3kb	3683
11	yellow	open	logstash-2017.05.05	0	9.9mb	9.9mb	18646
12	yellow	open	logstash-2017.04.30	0	9.7mb	9.7mb	14251
13	yellow	open	.triggered_watches	0	48.2kb	48.2kb	0
14	yellow	open	.monitoring-es-6-2021.06.20	912	8.1mb	8.1mb	9964
15	yellow	open	logstash-2017.05.04	0	8.6mb	8.6mb	16762
16	yellow	open	movies	0	1.9mb	1.9mb	9742
17	yellow	open	logstash-2017.05.02	0	8.7mb	8.7mb	16278
18	yellow	open	.monitoring-alerts-6	0	6.3kb	6.3kb	1

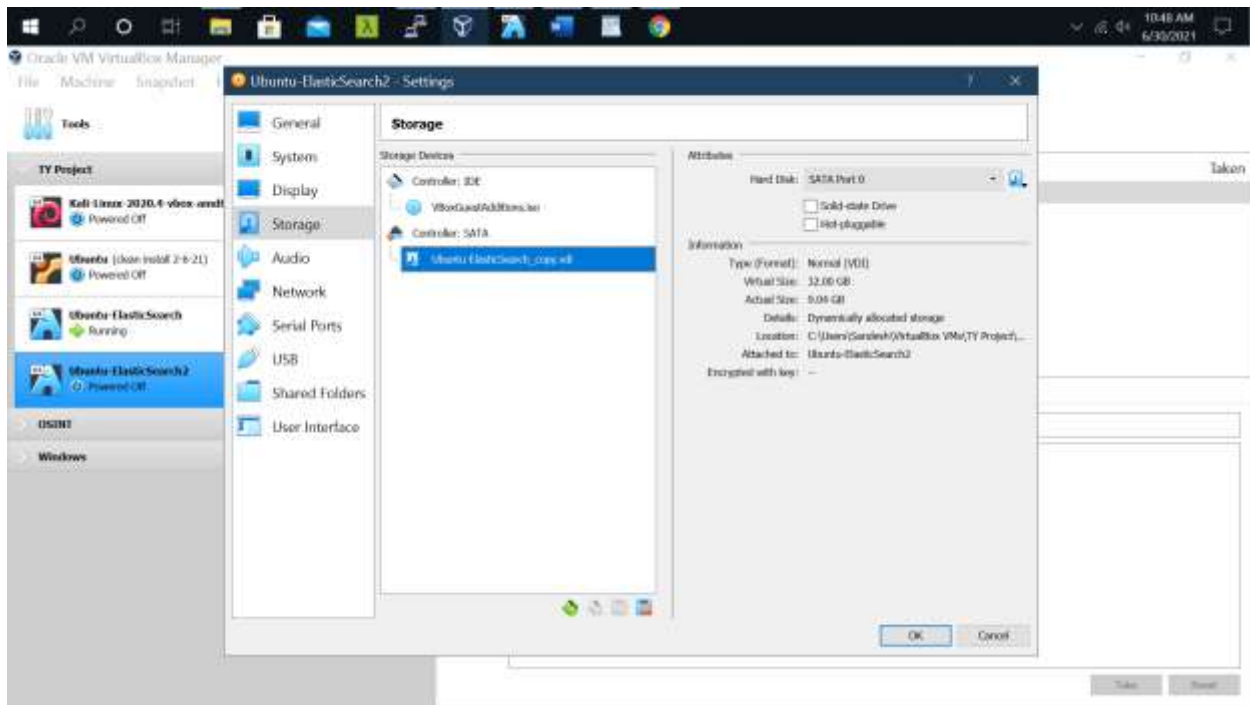
Q. How to Restore from a Snapshot from one cluster node to another cluster node?

Refer: <https://qbox.io/blog/elasticsearch-data-snapshots-restore-tutorial/>

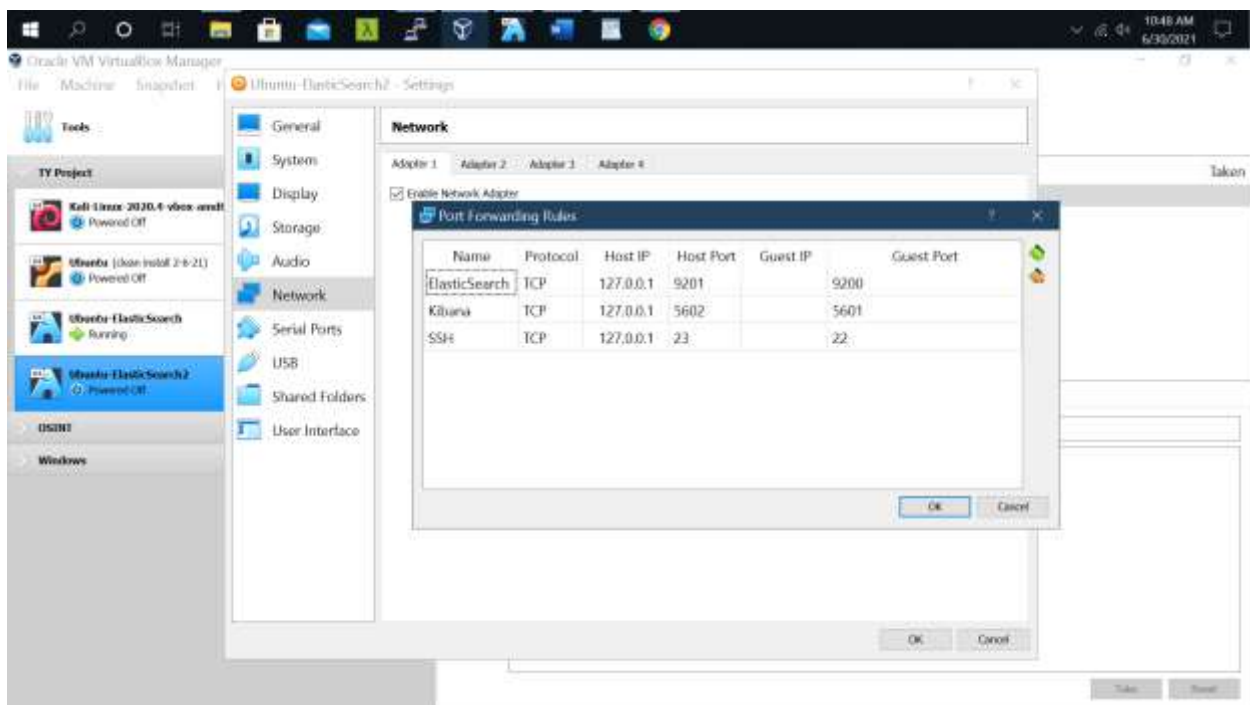
Step 0: Create entire new Elasticsearch virtual machine OR on VirtualBox go to File->Virtual Media Manager-> And make copy of vdi file of 1st Elasticsearch virtual machine.



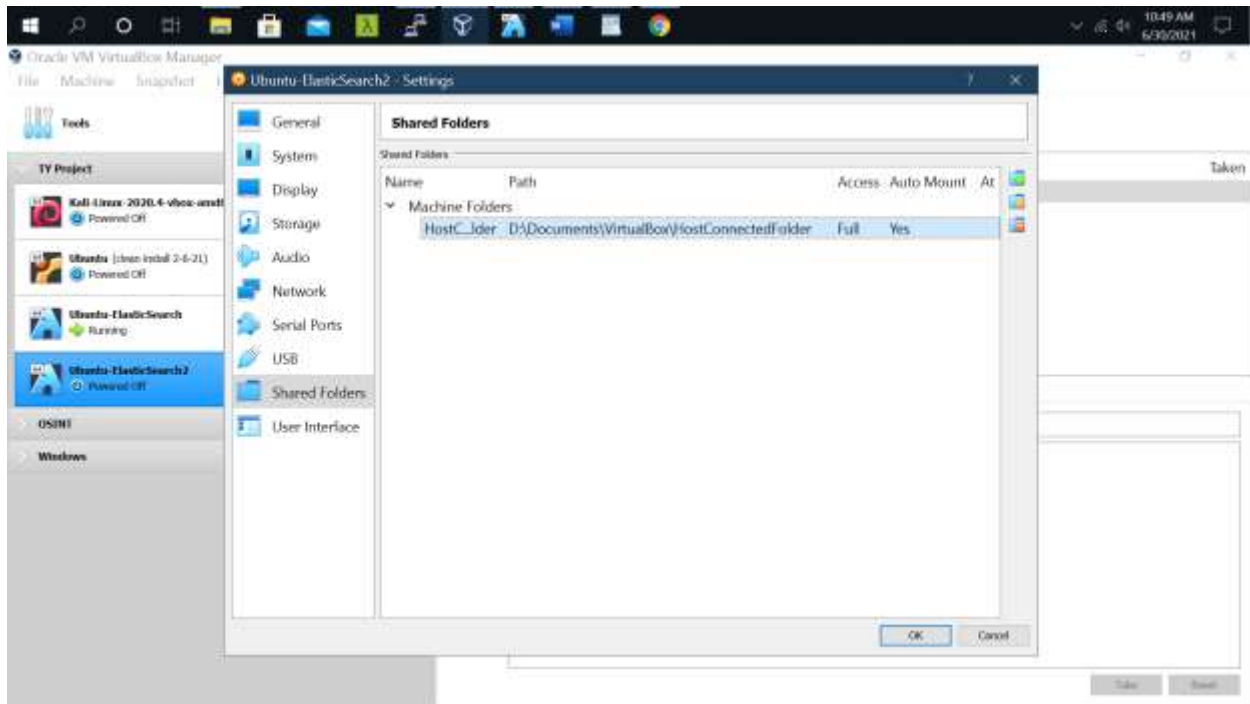
Now create another Virtual machine of name Elasticsearch 2 and go to Settings->Storage->Click on 'Controller:SATA' ->Click on 'Add Hard Disk' -> Select copy of vdi file of 1st Elasticsearch virtual machine ->Remove any other vdi file attached



Go to Network-> Advanced -> port forwarding and make sure you map different ES, Kibana, SSH ports to the local machine then 1st Elasticsearch virtual machine.



Now go to shared folders and connect any host folder to the virtual machine -> ok -> Start the Elasticsearch 2 virtual machine.



Attached same shared folder to 1st Elasticsearch virtual machine same way.

Step 1: Compress backup folder on cluster1 node OR move entire folder.

```
tar -zcvf backup.tar.gz ~/backups/
```

Step 2: Connect two VM's by shared host folder and move zip /entire folder from cluster1 to shared folder to cluster2.

```
knight@elksearch:~$ sudo cp -rp backups /media/sf_HostConnectedFolder/backups
knight@elksearch:~$
```

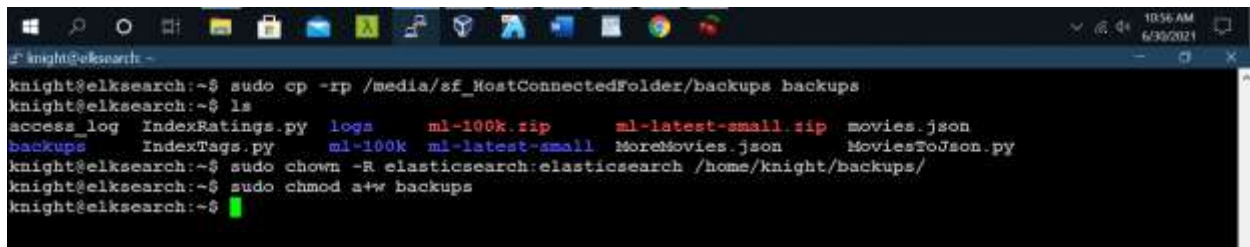
```
knight@elksearch:~$ sudo cp -rp /media/sf_HostConnectedFolder/backups backups
knight@elksearch:~$
```

Step 3: Place unzip folder/ entire folder at exact position on cluster2 as placed at cluster 1.

Step 4: Next, make sure that the Elasticsearch user has needed permissions to access the directory on cluster2 node -

```
$ sudo chown -R elasticsearch:elasticsearch /home/knight/backups/
```

```
$ sudo chmod a+w backups
```

A terminal window showing a series of commands and their outputs. The user runs 'sudo cp -rp /media/sf_MostConnectedFolder/backups backups', then 'ls' to list the contents of the 'backups' directory. The output shows various files including logs, index ratings, index tags, and movie data. Finally, the user runs 'sudo chown -R elasticsearch:elasticsearch /home/knight/backups/' and 'sudo chmod a+w backups' to set permissions. The terminal title is 'knight@elksearch: ~' and the system clock shows 10:56 AM on 6/30/2021.

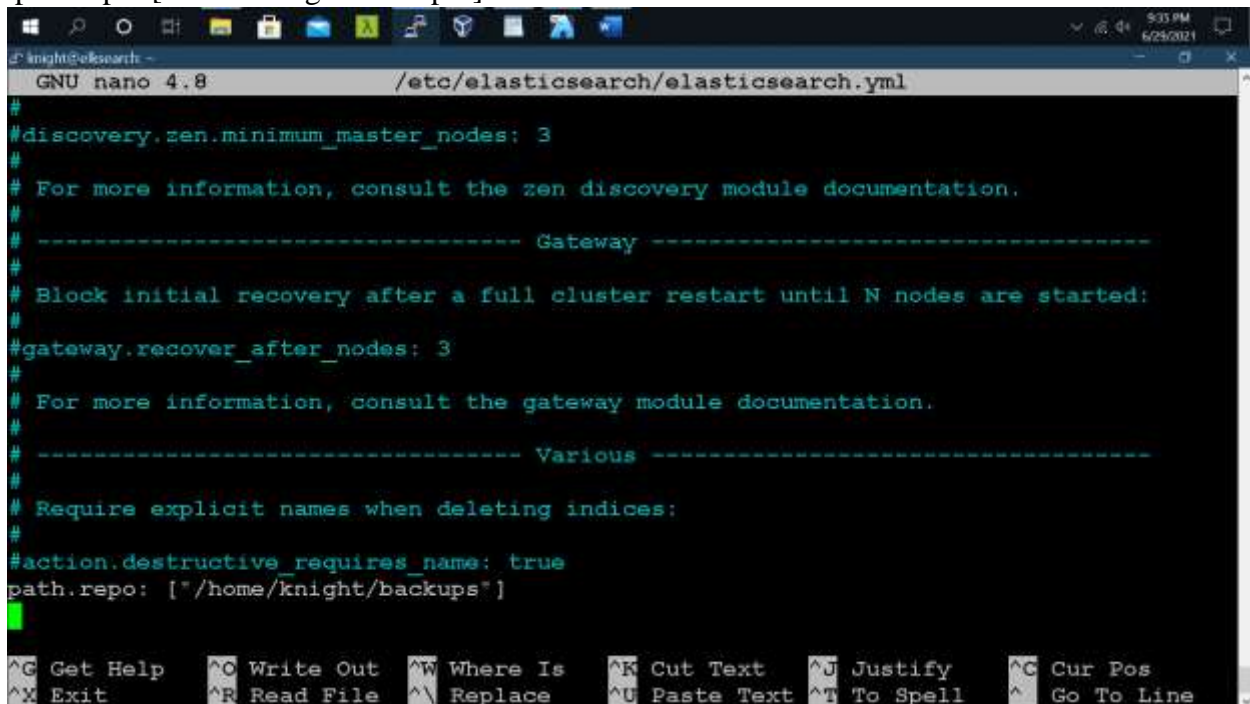
```
knight@elksearch:~$ sudo cp -rp /media/sf_MostConnectedFolder/backups backups
knight@elksearch:~$ ls
access_log  IndexRatings.py  logs          ml-100k.zip      ml-latest-small.zip  movies.json
backups     IndexTags.py     ml-100k       ml-latest-small  MoreMovies.json      MoviesToJson.py
knight@elksearch:~$ sudo chown -R elasticsearch:elasticsearch /home/knight/backups/
knight@elksearch:~$ sudo chmod a+w backups
knight@elksearch:~$
```

Step 5: Make entry in the .yml file of elasticsearch on cluster2 node i.e.,
/etc/elasticsearch/elasticsearch.yml -

```
$ sudo nano /etc/elasticsearch/elasticsearch.yml
```

Add following line at the end of the yml file,

```
path.repo: ["/home/knight/backups"]
```

A terminal window showing the 'nano' text editor editing the file '/etc/elasticsearch/elasticsearch.yml'. The file content includes configuration for discovery, gateway, and various settings. The user has added the line 'path.repo: ["/home/knight/backups"]' at the end of the file. The terminal title is 'GNU nano 4.8 /etc/elasticsearch/elasticsearch.yml'. The system clock shows 9:33 PM on 6/29/2021.

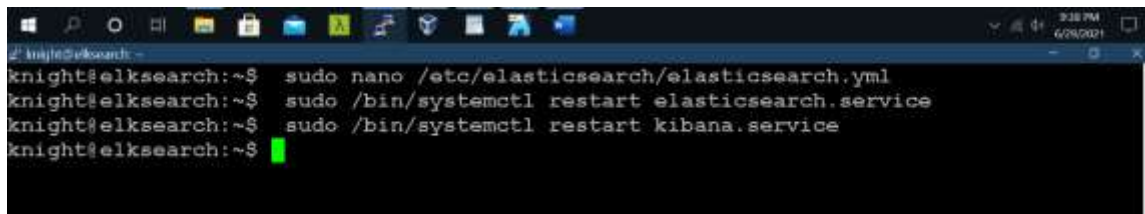
```
GNU nano 4.8 /etc/elasticsearch/elasticsearch.yml
#
#discovery.zen.minimum_master_nodes: 3
#
# For more information, consult the zen discovery module documentation.
#
# ----- Gateway -----
#
# Block initial recovery after a full cluster restart until N nodes are started:
#
#gateway.recover_after_nodes: 3
#
# For more information, consult the gateway module documentation.
#
# ----- Various -----
#
# Require explicit names when deleting indices:
#
#action.destructive_requires_name: true
path.repo: ["/home/knight/backups"]
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

Step 6: Restart the elasticsearch service on cluster2 node -

```
$ sudo /bin/systemctl restart elasticsearch.service
```

Step 7: restart kibana on cluster2 node -

\$ sudo /bin/systemctl restart kibana.service

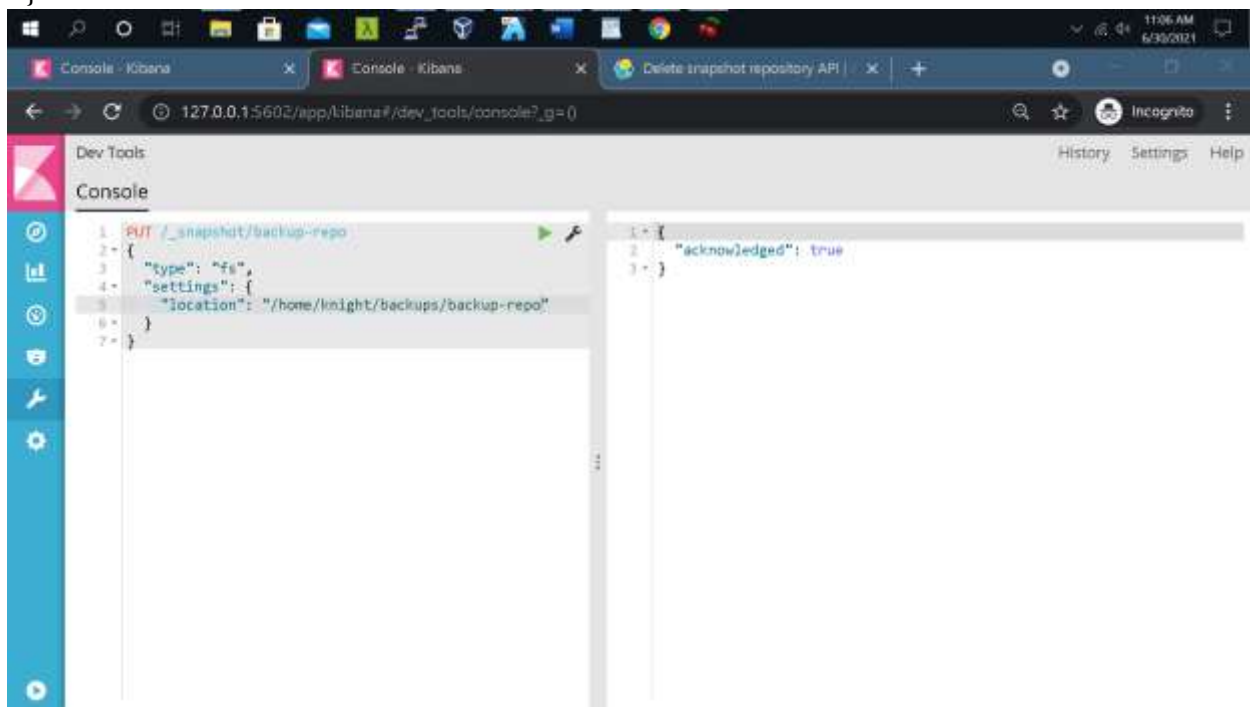


```
knight@elksearch:~$ sudo nano /etc/elasticsearch/elasticsearch.yml
knight@elksearch:~$ sudo /bin/systemctl restart elasticsearch.service
knight@elksearch:~$ sudo /bin/systemctl restart kibana.service
knight@elksearch:~$
```

Step 8: Now go to browser and open kibana – 127.0.0.1:5602 and open DevTool and type -

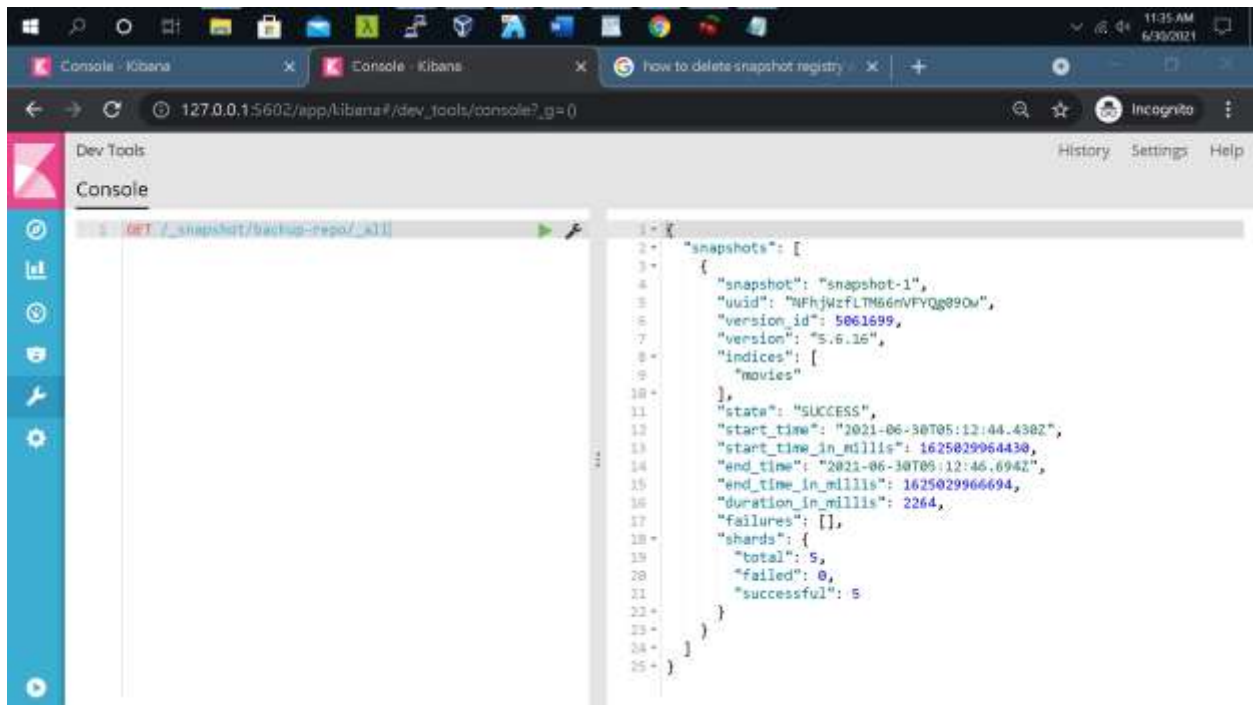
>>> PUT /_snapshot/backup-repo

```
{
  "type": "fs",
  "settings": {
    "location": "/home/knight/backups/backup-repo"
  }
}
```



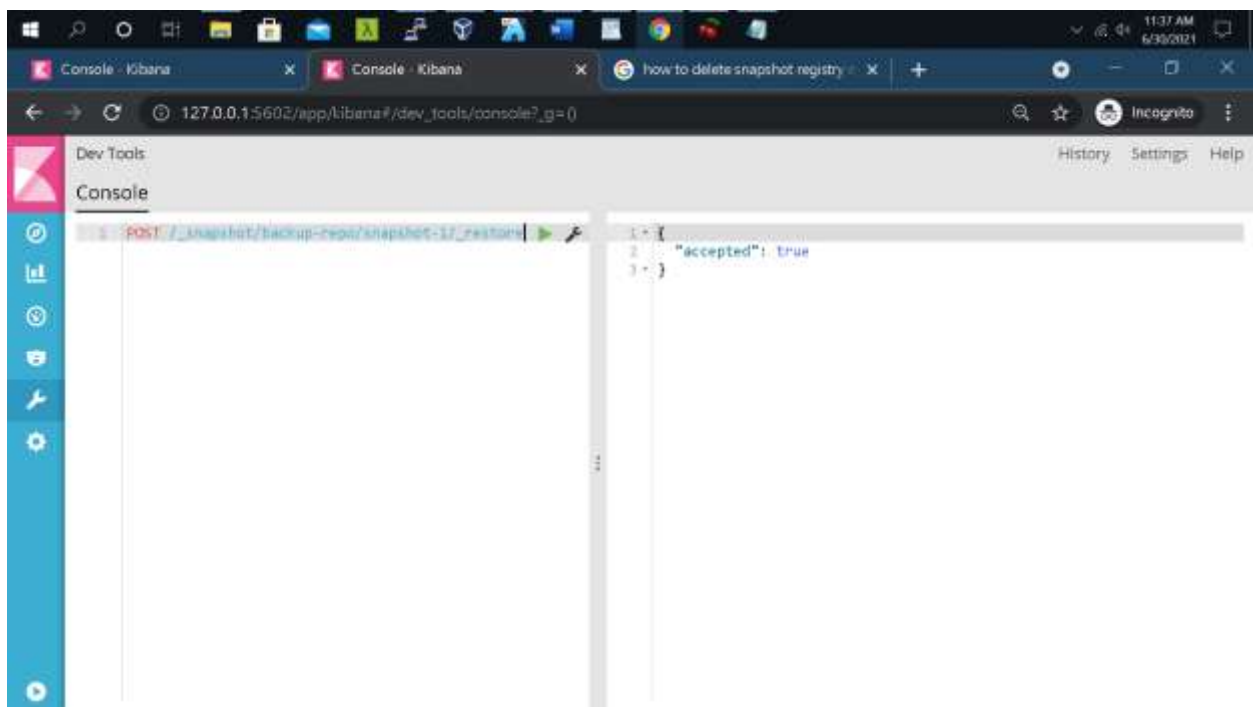
Step 9: Now list the snapshots in the repository, type -

>>>GET _snapshot/backup-repo/_all



Step 10: now recover deleted index by restoring snapshot, type -


`>>>POST /_snapshot/backup-repo/snapshot-1/_restore`



The screenshot shows a web browser window with the Kibana console open. The browser's address bar displays the URL: `127.0.0.1:5602/app/kibana#/dev_tools/console?_g=()`. The console interface includes a sidebar on the left with a 'Dev Tools' section and a 'Console' tab. The main area of the console shows a list of files and their sizes. The file 'movies' is highlighted in blue.

File Name	Size
logstash-2017.05.03	8.6mb
ratings	12mb
.kibana	15.4mb
tags	131.7kb
logstash-2017.05.05	991.3kb
logstash-2017.04.30	9.9mb
.triggered_watches	9.7mb
.monitoring-es-6-2021.05.20	40.2kb
logstash-2017.05.04	912
movies	8.1mb
logstash-2017.05.02	8.6mb
monitoring-alerts-6	1.9mb
	8.7mb
	6.3kb

```
>>>GET /movies/_search?pretty
```



The screenshot shows a web browser window with the Kibana console. The address bar displays the URL `127.0.0.1:5602/app/kibana#/dev_tools/console?_g=()`. The console interface includes a sidebar with Dev Tools and Console tabs. The main area shows a REST client request: `GET /movies/_search?pretty`. The response is a JSON object representing search results for the movie 'Nixon'.

```
{
  "took": 36,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 9742,
    "max_score": 1,
    "hits": [
      {
        "_index": "movies",
        "_type": "movie",
        "_id": "14",
        "score": 1,
        "_source": {
          "id": "14",
          "title": "Nixon",
          "year": 1995,
          "genre": [
            "Drama"
          ]
        }
      }
    ]
  }
}
```