***TAKING BACKUP IN ES***

***Let’s assume we have an elasticsearch index named ‘rmncha-index’. An index in elasticsearch is equivalent to a database in a postgres server.***

***To take backup of an index in elastic search we have to create two repository (physical and logical)***

***Let’s create a physical repository.***

***Fire this command in command line ‘mkdir backup’. This directory will be the root of all backups. i.e parent directory where all backups will be stored.***

***Now we have to tell elasticsearch about this backup directory. For this we have to edit the elasticsearch.yml file located in /config folder of elasticsearch HOME directory.***

***Add the following line of code in the yml file.***

***path.repo : /Users/azar/backup***

***Note: The backup folder that you created, you have specify path in yml file.***

***Save the file and restart elasticsearch server.***

***Elastic search takes backup in snapshots.i.e The backups are incremental in nature.***

***First we need to check whether if any snapshot exists for our elasticsearch index. Fire a GET request in browser localhost:9200/\_snapshot***

***Snapshot can be taken for few or a particular index or of the whole cluster. To take backup index-wise create sub-directory inside the parent repository.***

***Now we will create logical repository.***

***Here we will take snapshot of particular index "rmncha-index"***

***Do a PUT request to elasticsearch server :***

***curl -XPUT 'localhost:9200/\_snapshot/rmnchaindexbackup' -d '{  
    "type": "fs",   
    "settings": {  
        "location": "rmncha-index",  
        "compress": "true"  
    }  
}'***

***After firing this request, our snapshot will be created in server.***

***No need to specify the full path in "location" as we have already specified it in the config file path.repo*.**

***The name in “location” will create a folder with name that is given in location key if it doesn’t exist.***

***To take backup of elasticsearch index.***

***We have to provide a PUT request to server in following format.***

***curl -XPUT 'localhost:9200/\_snapshot/rmnchaindexbackup/18May18' -d '{  
    "indices": "rmncha-index",  
    "ignore\_unavailable": true,  
    "include\_global\_state": false  
}'***

***indices :****field may have multiple index with comma separated values****.***

***ignore\_unavailable :****means don't stop the snapshot process if any shards is not available****.***

***include\_global\_state :****means take snapshot of cluster status also. If the status is yellow than while restore the yellowstatus will be there. If you don't want this than set it to false****.***

***To check if snapshot is completed or not fire this request :***

***curl –XGET ‘ localhost:9200/\_snapshot/rmnchaindexbackup/18May18’***

***Following response will be generated. The “state” key holds the status of backup / snapshot generation. As you can see the state is IN\_PROGRESS. Once status becomes SUCCESS means backup process has been completed and you can find the backup in location folder you have specified.***

***{***

***"snapshots": [***

***{***

***"snapshot": "15.05.2018",***

***"version\_id": 2040499,***

***"version": "2.4.4",***

***"indices": [***

***"rmncha-index"***

***],***

***"state": "IN\_PROGRESS",***

***"start\_time": "2018-05-15T09:54:53.765Z",***

***"start\_time\_in\_millis": 1526378093765,***

***"failures": [],***

***"shards": {***

***"total": 0,***

***"failed": 0,***

***"successful": 0***

***}***

***}***

***]***

***}***

***Restoring Index In Same Cluster***

***To restore index fire a POST request :***

***curl -XPOST localhost:9200/\_snapshot/rmnchaindexbackup/18May18/\_restore***

***Restoring Index In Another Cluster***

***To restore generated snapshot in another cluster, create a zip of backup folder where our backup is stored and paste it in the same location in cluster where we want to restore.***

***After that set path.repo attribute in elasticsearch.yml.***

***Once path is set restart the cluster and create a snapshot repository using following PUT request.***

***curl -XPUT 'localhost:9200/\_snapshot/rmnchaindexbackup' -d '{  
    "type": "fs",   
    "settings": {  
        "location": "rmncha-index",  
        "compress": "true"  
    }  
}'***

***Once our snapshot repository is created and we have already decompressed the zip file and put that in same location in cluster where we want to restore it. We will fire a POST request to restore it.***

***curl -XPOST localhost:9200/\_snapshot/rmnchaindexbackup/18May18/\_restore***