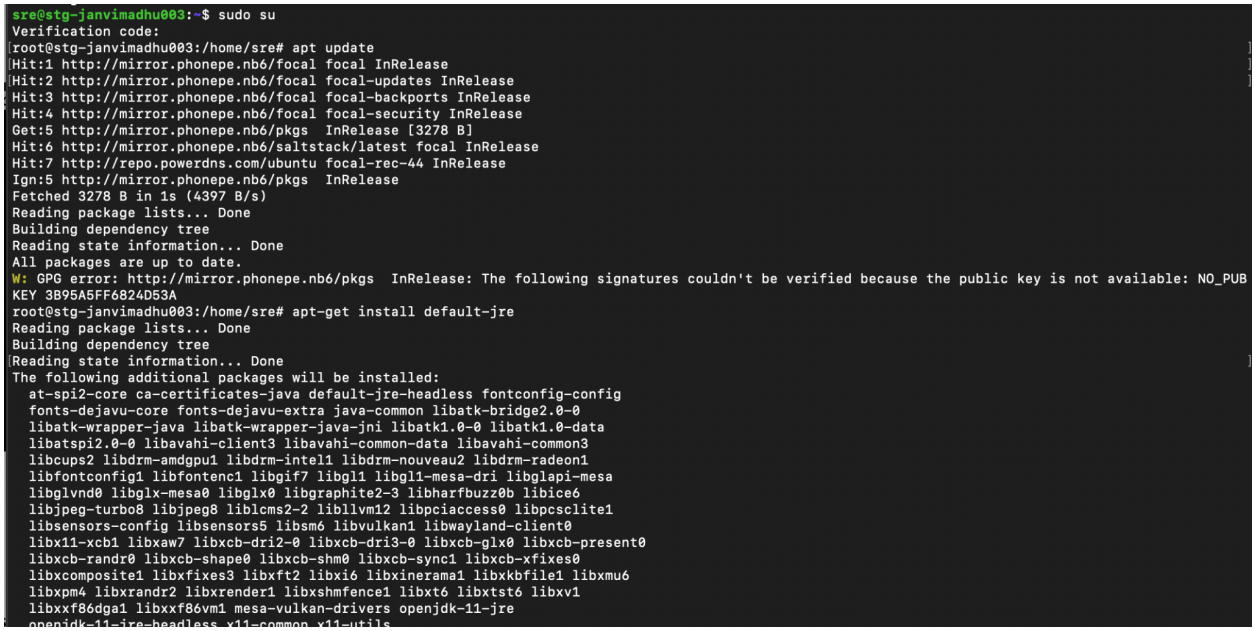
Elastisearch

Create a 3 node elasticsearch cluster with the name *pp_elasticsearch*.

STEPS:-

1. Add the hostnames in all the 3 nodes in **/etc/hosts** path.
2. Install the following dependencies using the below commands (to install elasticsearch.)

- apt update
- apt-get install default-jre



- sudo apt-get install apt-transport-https
- wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
- echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list

- sudo apt-get update

```
root@stg-janvimadhu003:/home/sre# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
OK
root@stg-janvimadhu003:/home/sre# echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list
deb https://artifacts.elastic.co/packages/6.x/apt stable main
root@stg-janvimadhu003:/home/sre# sudo apt-get update
Hit:1 http://mirror.phonepe.nb6/focal focal InRelease
Hit:2 http://mirror.phonepe.nb6/focal focal-updates InRelease
Hit:3 http://mirror.phonepe.nb6/focal focal-backports InRelease
Hit:4 http://mirror.phonepe.nb6/focal focal-security InRelease
Get:5 http://mirror.phonepe.nb6/pkgs  InRelease [3278 B]
Hit:6 http://mirror.phonepe.nb6/saltstack/latest focal InRelease
Hit:7 http://repo.powerdns.com/ubuntu focal-rec-44 InRelease
Get:8 https://artifacts.elastic.co/packages/6.x/apt stable InRelease [7123 B]
Ign:5 http://mirror.phonepe.nb6/pkgs  InRelease
Get:9 https://artifacts.elastic.co/packages/6.x/apt stable/main amd64 Packages [85.4 kB]
Fetched 95.8 kB in 2s (49.2 kB/s)
Reading package lists... Done
W: GPG error: http://mirror.phonepe.nb6/pkgs  InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 3B95A5FF6824D53A
root@stg-janvimadhu003:/home/sre# sudo apt-get install elasticsearch
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 149 MB of archives.
After this operation, 240 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/6.x/apt stable/main amd64 elasticsearch all 6.8.23 [149 MB]
Fetched 149 MB in 2s (74.3 MB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 98705 files and directories currently installed.)
Preparing to unpack .../elasticsearch_6.8.23_all.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (6.8.23) ...
Setting up elasticsearch (6.8.23) ...
Created elasticsearch keystore in /etc/elasticsearch
Processing triggers for systemd (245.4-4ubuntu3.11) ...
```

3. Install elastic search using the below command:

- sudo apt install elasticsearch

4. Do the following configuration on node1 in **/etc/elasticsearch/elasticsearch.yml** using the below command:

```
  GNU nano 4.8                                  /etc/elasticsearch/elasticsearch.yml
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
cluster.name: pp_elasticsearch
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
node.name: stg-janvimadhu002
node.master: true
node.data: true
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ----------------------------------- Paths ------------------------------------
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ----------------------------------- Memory -----------------------------------
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network -----------------------------------
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: 10.57.55.41
#
# Set a custom port for HTTP:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# --------------------------------- Discovery ----------------------------------
#
# Pass an initial list of hosts to perform discovery when new node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
discovery.zen.ping.unicast.hosts: ["10.57.55.41","10.57.55.42","10.57.55.43"]
#
# Prevent the "split brain" by configuring the majority of nodes (total number of master-eligible nodes / 2 + 1):
#
discovery.zen.minimum_master_nodes: 2
#
```

5. Do the same configuration on node2 and node3 with their respective network. host.

6. Restart **elasticsearch.service** to save the configuration on all the 3 nodes.

```
[sre@stg-janvimadhu002:~$ sudo systemctl start elasticsearch.service^C
[sre@stg-janvimadhu002:~$ sudo systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; disabled; v>
     Active: active (running) since Wed 2022-03-23 18:05:11 IST; 3min 37s a>
       Docs: http://www.elastic.co
   Main PID: 200123 (java)
      Tasks: 43 (limit: 4682)
     Memory: 1.2G
     CGroup: /system.slice/elasticsearch.service
             ├─200123 /bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:C>
             └─200220 /usr/share/elasticsearch/modules/x-pack-ml/platform/l>

Mar 23 18:05:11 stg-janvimadhu002 systemd[1]: Started Elasticsearch.
Mar 23 18:05:11 stg-janvimadhu002 elasticsearch[200123]: OpenJDK 64-Bit Ser>
lines 1-13/13 (END)
```

7. Use the below command to check the cluster health (general info on the cluster and gauge its health)

```
root@stg-janvimadhu002:/home/sre# sudo curl http://10.57.55.41:9200
{
  "name" : "stg-janvimadhu002",
  "cluster_name" : "pp_elasticsearch",
  "cluster_uuid" : "GM1x672TQrqw3HJdB0XgVA",
  "version" : {
    "number" : "6.8.23",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "4f67856",
    "build_date" : "2022-01-06T21:30:50.087716Z",
    "build_snapshot" : false,
    "lucene_version" : "7.7.3",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
root@stg-janvimadhu002:/home/sre# sudo curl 10.57.55.41:9200/_cluster/health?pretty
{
  "cluster_name" : "pp_elasticsearch",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 3,
  "number_of_data_nodes" : 3,
  "active_primary_shards" : 0,
  "active_shards" : 0,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 0,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
```

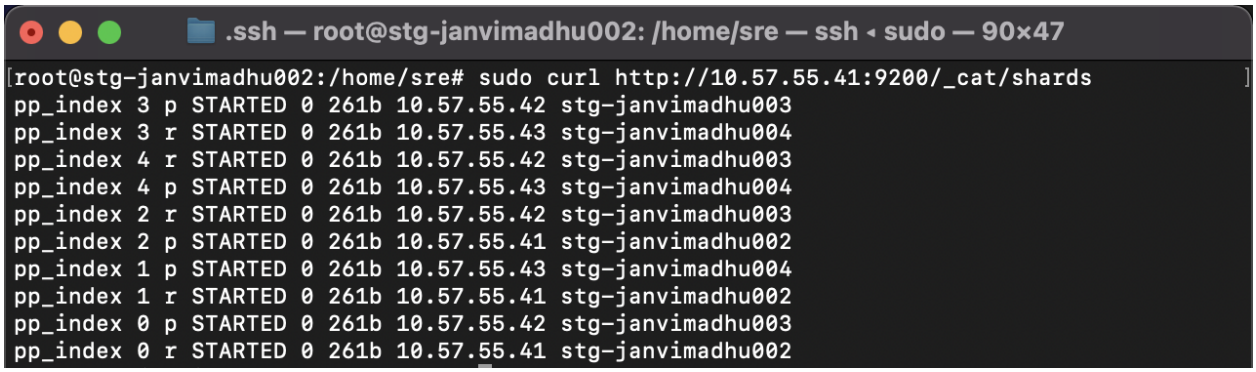
1. Create an index with pp_index and note down the number of primary/replica shards created by default.

STEPS:

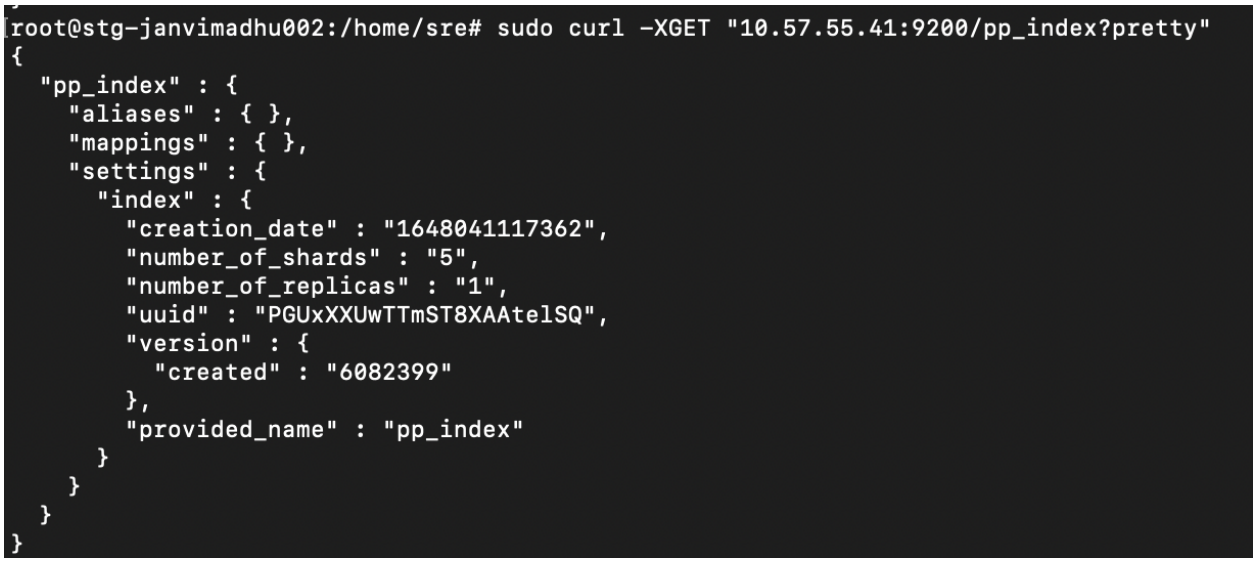
1. To create a index with pp_index we will use the below command:

sudo curl -XPUT “10.57.55.41:9200/pp_index?pretty”

2. To check the number of primary and replica shards created by default use the below command:



3. To check whether or not our index is created or not we can use the command given below:



2. Insert some data into the index by creating 20 to 30 documents.

Using the bulk api we can directly insert multiple data into our index using the below command:

```
root@stg-janvimadhu002:/home/sre# sudo curl -X POST "http://10.57.55.41:9200/pp_index/names/_bulk" -H 'Content-Type: application/json' -d '{"index":{"_id":"4"} }
{"name":"janvi"}
{"index":{"_id": "5"}}
{"name": "janvi"}
> {"index":{"_id":"6"} }
> {"name":"janvi"}
> {"index":{"_id":"7"} }
> {"name":"janvi"}
> {"index":{"_id":"8"} }
> {"name":"janvi"}
> {"index":{"_id":"9"} }
> {"name":"janvi"}
> {"index":{"_id":"10"} }
> {"name":"janvi"}
> {"index":{"_id":"11"} }
> {"name":"janvi"}
> {"index":{"_id":"12"} }
{"name":"janvi"}
{"index":{"_id": "13"}}
{"name": "janvi"}
{"index":{"_id":"14"} }
{"name":"janvi"}
{"index":{"_id":"15"} }
{"name":"janvi"}
{"index":{"_id":"16"} }
{"name":"janvi"}
{"index":{"_id":"17"} }
{"name":"janvi"}
{"index":{"_id":"18"} }
{"name":"janvi"}
{"index":{"_id":"19"} }
{"name":"janvi"}
> {"index":{"_id":"20"} }
> {"name":"janvi"}
> '
{"took":36,"errors":false,"items":[{"index":{"_index":"pp_index","_type":"names","_id":"4","_version":1,"result":"created","_shards":{"total":2,"successful":2,"failed":0},"_seq_no":1,"_primary_term":1
,"status":201}},{"index":{"_index":"pp_index","_type":"names","_id":"5","_version":1,"result":"created","_shards":{"total":2,"successful":2,"failed":0},"_seq_no":0,"_primary_term":1,"status":201}},{"i
ndex":{"_index":"pp_index","_type":"names","_id":"6","_version":1,"result":"created","_shards":{"total":2,"successful":2,"failed":0},"_seq_no":2,"_primary_term":1,"status":201}},{"index":{"_index":"pp
_index","_type":"names","_id":"7","_version":1,"result":"created","_shards":{"total":2,"successful":2,"failed":0},"_seq_no":1,"_primary_term":1,"status":201}},{"index":{"_index":"pp_index","_type":"na
mes","_id":"8","_version":1,"result":"created","_shards":{"total":2,"successful":2,"failed":0},"_seq_no":1,"_primary_term":1,"status":201}},{"index":{"_index":"pp_index","_type":"names","_id":"9","_ve
```

3. Read the data from the indice and write the same to a json file. [file name : read_data.json]

To read the data from an index we can use the below command:

```
root@stg-janvimadhu002:/home/sre# sudo curl -XGET "http://10.57.55.41:9200/pp_index/_search?pretty"
{
  "took" : 18,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 20,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "pp_index",
        "_type" : "names",
        "_id" : "14",
        "_score" : 1.0,
        "_source" : {
          "name" : "janvi"
        }
      },
      {
        "_index" : "pp_index",
        "_type" : "names",
        "_id" : "19",
        "_score" : 1.0,
        "_source" : {
          "name" : "janvi"
        }
      },
      {
        "_index" : "pp_index",
        "_type" : "names",
        "_id" : "5",
        "_score" : 1.0,
        "_source" : {
          "name" : "janvi"
        }
      },
      {
        "_index" : "pp_index",
        "_type" : "names",
        "_id" : "8",
        "_score" : 1.0,
        "_source" : {
          "name" : "janvi"
        }
      },
      {
        "_index" : "pp_index",
        "_type" : "names",
        "_id" : "9",
        "_score" : 1.0,
        "_source" : {
          "name" : "janvi"
        }
      },
      {
        "_index" : "pp_index",
        "_type" : "names",
        "_id" : "10",
        "_score" : 1.0,
        "_source" : {
          "name" : "janvi"
        }
      },
```

To write the same response onto a json file we can use elasticdump command:

```
root@stg-janvimadhu002:/home/sre# sudo elasticdump --input=http://10.57.55.41:9200/pp_index  --output
=read_file.json  --type=data
[Wed, 23 Mar 2022 18:35:10 GMT | starting dump                                                    ]
Wed, 23 Mar 2022 18:35:10 GMT | got 20 objects from source elasticsearch (offset: 0)
Wed, 23 Mar 2022 18:35:10 GMT | sent 20 objects to destination file, wrote 20
[Wed, 23 Mar 2022 18:35:10 GMT | got 0 objects from source elasticsearch (offset: 20)              ]
Wed, 23 Mar 2022 18:35:10 GMT | Total Writes: 20
Wed, 23 Mar 2022 18:35:10 GMT | dump complete
```

```
  GNU nano 4.8                                  read_file.json
{"_index":"pp_index","_type":"names","_id":"14","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"19","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"5","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"8","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"9","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"10","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"12","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"2","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"4","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"6","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"15","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"20","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"1","_score":1,"_source":{"name":"Janvi"}}
{"_index":"pp_index","_type":"names","_id":"7","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"13","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"16","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"18","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"3","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"11","_score":1,"_source":{"name":"janvi"}}
{"_index":"pp_index","_type":"names","_id":"17","_score":1,"_source":{"name":"janvi"}}
```

4. Capture the elasticsearch cluster health along with a timestamp and write it to file. [file name : cluster_health_before_step5.json]

Now lets change our nodes configuration in such a way that stg-janvimadhu002 is master node and stg-janvimadhu003 and stg-prabhjotbedi004 are data nodes.

Now to capture ElasticSearch cluster health along with timestamp onto a file we can use the below command:

```
root@stg-janvimadhu002:/home/sre# sudo curl http://10.57.55.41:9200/_cat/health?format=json > cluster_health_before_step5.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   265  100   265    0     0  33125      0 --:--:-- --:--:-- --:--:-- 33125
```

```
root@stg-janvimadhu002: /home/sre (ssh)
  GNU nano 4.8                                  cluster_health_before_step5.json
[{"epoch":"1650694257","timestamp":"06:10:57","cluster":"pp_elasticsearch","status":"green","node.total":"3","node.data":"3","shards":"10","pri":"5","relo":"0","init":"0","unassign":"0","pending_tasks":"0","max_task_wait_time":"-","active_shards_
```

5. Stop two data nodes from the cluster and ensure you are still able to read the same data from es and write it to a json file. [file name : read_data_after_step5.json]

1. Stop 2 data nodes using systemctl stop elasticsearch.service

```
root@stg-janvimadhu003:/home/sre# sudo systemctl stop elasticsearch.service
root@stg-janvimadhu003:/home/sre# sudo systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; disabled; ven>
     Active: inactive (dead)
       Docs: http://www.elastic.co

Apr 23 10:57:33 stg-janvimadhu003 systemd[1]: Started Elasticsearch.
Apr 23 10:57:33 stg-janvimadhu003 elasticsearch[574383]: OpenJDK 64-Bit Serve>
Apr 23 11:01:17 stg-janvimadhu003 systemd[1]: Stopping Elasticsearch...
Apr 23 11:01:17 stg-janvimadhu003 systemd[1]: elasticsearch.service: Succeede>
Apr 23 11:01:17 stg-janvimadhu003 systemd[1]: Stopped Elasticsearch.
Apr 23 11:01:17 stg-janvimadhu003 systemd[1]: Started Elasticsearch.
Apr 23 11:01:18 stg-janvimadhu003 elasticsearch[574528]: OpenJDK 64-Bit Serve>
Apr 23 11:07:47 stg-janvimadhu003 systemd[1]: Stopping Elasticsearch...
Apr 23 11:07:47 stg-janvimadhu003 systemd[1]: elasticsearch.service: Succeede>
Apr 23 11:07:47 stg-janvimadhu003 systemd[1]: Stopped Elasticsearch.
lines 1-15/15 (END)
```

2. Now when we will try to read data from es it will give a message all shards failed we can write this message in the form of json using the below command.(meaning we cannot read data if our data nodes fail)

```
root@stg-janvimadhu002:/home/sre# sudo curl "http://10.57.55.41:9200/pp_index/_search?pretty" > read_data_after_step5.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2011  100  2011    0     0  14062      0 --:--:-- --:--:-- --:--:-- 14161
```

```
 GNU nano 4.8                                read_data_after_step5.json
{
  "error" : {
    "root_cause" : [ ],
    "type" : "search_phase_execution_exception",
    "reason" : "all shards failed",
    "phase" : "query",
    "grouped" : true,
    "failed_shards" : [ ]
  },
  "status" : 503
}
```

6. Capture the elasticsearch cluster health along with a timestamp and write it to file. [file name : cluster_health_after_step5.json]

```
root@stg-janvimadhu002:/home/sre# sudo curl http://10.57.55.41:9200/_cat/health?format=json > cluster_heal
th_after_step5.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   261  100   261    0     0   9321      0 --:--:-- --:--:-- --:--:--  9666
```

```
root@stg-janvimadhu002: /home/sre (ssh)
 GNU nano 4.8                                cluster_health_after_step5.json
[{"epoch":"1650694792","timestamp":"06:19:52","cluster":"pp_elasticsearch","status":"red","node.total":"1","node.data":"0","shards":"0","pri":"0","relo":"0",">
```

7. Add the nodes back to the cluster.

Use systemctl start elasticsearch.service command on the 2 data nodes to bring them back to the cluster

```
root@stg-janvimadhu003:/home/sre# systemctl start elasticsearch
root@stg-janvimadhu003:/home/sre# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; disabled; vendor preset: enabled)
     Active: active (running) since Sat 2022-04-23 11:50:45 IST; 4s ago
       Docs: http://www.elastic.co
   Main PID: 580366 (java)
      Tasks: 19 (limit: 4682)
     Memory: 1.1G
     CGroup: /system.slice/elasticsearch.service
             ├─580366 /bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -Des.networkaddres
             └─580451 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller
```

8. Increase replica shard count to 3 (for each primary shard you need to have 3 replicas) and capture the location of primary and replica shards

1. To increase the replica shard count by 3 use the below command:

```
root@stg-janvimadhu002:/home/sre# sudo curl -X PUT "10.57.55.41:9200/pp_index/_settings?pretty" -H 'C
ontent-type: application/json' -d'
> {
>  "index" : {
>  "number_of_replicas" : 3
> }
> }
> '
{
  "acknowledged" : true
}
```

```
root@stg-janvimadhu002:/home/sre# sudo curl -X PUT "10.57.55.41:9200/pp_index/_settings?pretty" -H 'C
ontent-type: application/json' -d'
> {
>  "index" : {
>  "number_of_replicas" : 3
> }
> }
> '
{
  "acknowledged" : true
}
```

2. To capture the location of primary and replica shards use the below command (Here I have also made **stg-janvimadhu002** a data node so that maximum number of replica shards can be assigned, unassigned replica shards can get assigned when more data nodes are added to the cluster)

```
5}root@stg-janvimadhu002:/home/sre# sudo curl http://10.57.55.41:9200/_cat/shards
pp_index 3 r STARTED    5 6.7kb 10.57.55.43 stg-janvimadhu004
pp_index 3 p STARTED    5 6.7kb 10.57.55.42 stg-janvimadhu003
pp_index 3 r UNASSIGNED
pp_index 3 r UNASSIGNED
pp_index 1 p STARTED    5 3.5kb 10.57.55.43 stg-janvimadhu004
pp_index 1 r STARTED    5 3.5kb 10.57.55.42 stg-janvimadhu003
pp_index 1 r UNASSIGNED
pp_index 1 r UNASSIGNED
pp_index 4 r STARTED    3 6.7kb 10.57.55.43 stg-janvimadhu004
pp_index 4 p STARTED    3 6.7kb 10.57.55.42 stg-janvimadhu003
pp_index 4 r UNASSIGNED
pp_index 4 r UNASSIGNED
pp_index 2 r STARTED    5 6.7kb 10.57.55.43 stg-janvimadhu004
pp_index 2 p STARTED    5 6.7kb 10.57.55.42 stg-janvimadhu003
pp_index 2 r UNASSIGNED
pp_index 2 r UNASSIGNED
pp_index 0 r STARTED    2 3.5kb 10.57.55.43 stg-janvimadhu004
pp_index 0 p STARTED    2 3.5kb 10.57.55.42 stg-janvimadhu003
pp_index 0 r UNASSIGNED
pp_index 0 r UNASSIGNED
```