

# Snapshot and Restore ElasticSearch on Openshift

## 1. Pré-requisitos:

- Elasticsearch instalado no Openshift de preferência utilizando statefulset;
- Servidor de NFS com acesso liberado para o Openshift
- PV criado e apontando para o NFS, /disponível para o pvc do namespace do Elasticsearch.

## 2. Disponibilizar um NFS

- O pv deve permitir que o access mode esteja em:
  - ReadWriteMany de acordo com a DOC [Persistente Storage - Red Hat Docs](#)
  - Aqui o link para configurar um [NFS Server no RHEL8](#)

## 3. Configurar PV para nfs dentro do Openshift

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteMany
  nfs:
    path: /tmp
    server: 172.17.0.2
```

## 4. Configurar PVC dentro do statefulset YAML do Elasticsearch:

- Configurando uma nova env para de volume dentro do pod

```
- env:
  - name: path.repo
    value: /backup
```

- Criando o volumeMounts apontando para o pv do NFS

```
volumeMounts:
  - mountPath: /usr/share/elasticsearch/data
    name: es-storage-db-hml
  - mountPath: /backup
    name: es-backup-hml-nfs-volume2
```

- Configurando o volume para vincular ao claim:

```
- name: es-backup-hml-nfs-volume2
  persistentVolumeClaim:
    claimName: es-backup-hml-nfs-claim2
```

## 5. Criando o repo path dentro do Elasticsearch via API:

- Opção de chamada por api

```
curl -XPUT -H "content-type:application/json"
'http://localhost:9200/snapshot/backup' -d '{"type": "fs", "settings":
{"location": "my_backup_location"}}'
```

- Opção via [Cerebro do Lmenezes](#):

```
PUT /_snapshot/my_backup
{
  "type": "fs",
  "settings": {
    "location": "my_backup_location"
  }
}
```

## 6. Realizando o snapshot via API:

- Executando em máquina local

```
curl -XPUT -H "content-type:application/json"
'http://localhost:9200/_snapshot/backup' -d '{"type": "fs", "settings":
{"location": "/home/rpinto/es-backup", "compress": true}}'
```

- Executando no Elasticsearch do Openshift:

```
curl -XPUT -H "content-type:application/json"
'http://localhost:9200/_snapshot/backup' -d '{"type": "fs", "settings":
{"location": "/backup", "compress": true}}'
```

## 7. Para listar o backup:

```
curl -XGET 'http://localhost:9200/_snapshot/_all?pretty'
```

- *OBS: É possível visualizar o backup via cérebro, também.*

## 8. Listando os índices:

```
curl -XGET 'http://localhost:9200/_cat/indices'
```

## 9. Restaurando o backup e garantindo que vamos pegar do diretório correto:

```
curl -XPOST 'http://localhost:920/_snapshot/backup/first-snapshot/_restore?wait_for_completion=true'
```

## 10. Listando os índices para garantir que o processo vai ocorrer com sucesso

```
curl -XGET 'http://localhost:9200/_cat/indices'
```

## 11. Configurar Job para executar Script Diariamente:

- **script-snap.sh**

### 11.1 - Criando script pelo vim

```
vim script-snap.sh
```

### 11.2 - Conteúdo do arquivo

```
#!/bin/bash
DATE=$(date +%Y-%m-%d-%H-%M-%S)

status_code=$(curl --write-out %{http_code} --silent --output /dev/null -
XPUT --insecure https://localhost:9200/_snapshot/backup/backup-data-$DATE?
wait_for_completion=true)

if [[ "$status_code" -ne 200 ]] ; then
    exit 1
else
    exit 0
fi
```

### 11.2 - Setando as permissões de execução do script:

```
chmod 755 script-snap.sh
```

## 12. Resumo do Procedimento de Restore

1. Parar Kibana
2. Remover Indices
3. Restaurar Snapshot
4. Iniciar Kibana

## 13. Dicas

- *A melhor forma de testar seria subir um elasticsearch zerado, de preferência sem ter o kibana rodando e sem os índices iniciais criados;*
- *Caso o kibana esteja rodando deve-se parar o kibana e apagar os índices que ele fica tentando recriar de forma automatica, impossibilitando o restore;*
- *É possível criar uma política via interface do kibana para deixar todo o processo funcionando sem a necessidade de uma chamada de um script externo para rodar o backup.*

## 14. Alguns comandos extras:

- Para listar índices e exibir seus tamanhos:

```
curl -XGET 'http://localhost:9200/_cat/indices'
```

- Aguardar até que o comando seja finalizado usando o "wait\_for\_completion=true":

```
curl -XPUT 'http://localhost:9200/_snapshot/esbackup/first-snapshot?wait_for_completion=true'
```

```
curl -XPUT 'http://localhost:9200/_snapshot/backup/primeiro-backup?wait_for_completion=true'
```

- Simulando um deleção de todos os índices:(CUIDADO!)

```
curl -XDELETE 'http://localhost:9200/_all'
```

## 15. Fonte:

- [Snapshot e Restore Elasticsearch Doc](#)
- [Criando volumes persistentes - Red Hat Docs](#)
- [Configurando um NFS Server no RHEL8](#)

- [Cerebro do Lmenezes](#)