**Main Topic Elastic Search**  
 Here we have definition for Logstash , Elastic search and Kibana but mainly we will concentrate about **Elastic search** Only

**Elastic search, Logstash, Kibana (ELK)**

**ELK** is a log management platform that helps DevOps engineers in making better decisions for the company. ELK comprises of Elastic Search, Logstash, and Kibana Open-source software offered by the elastic company, Logstash collects different types of logs and sends them into a pipeline of events.

A diagram of a data processing process

Description automatically generated

A diagram of a logistic system

Description automatically generated

**Logstash**

Logstash is a lightweight, Open source, server-side data processing pipeline that allows you to collect data from various sources, transform it on the fly, and send it to your desired destination. It is most often used as a data pipeline for Elasticsearch, an Open-source analytics and search engine.

Logstash is a part of the ELK stack, it does not mean Logstash is limited to use with elastic search. It can push data not only to Elasticsearch but also to other services like Datadog, MongoDB, and even to message brokers like Amazon SQS, etc.

Initial release date: February 2016

Programming languages: java, Ruby

**Is Logstash Push or Pull**

Logstash acts as an aggregator pulling data from various sources before pushing it down the pipeline, usually into Elasticsearch but into a buffering component in larger production environments.

**Alternatives for Logstash**

* Better Stack, Fluentd, Splunk, Kafka, Beats….etc

A logo for a company

Description automatically generated

Filter

Example grok filter

Grok is filter within Logstash that is Used to parse unstructured data into something structured and queryable. It sits on top of Regular Expression and uses text patterns to match lines in log files.

**Elasticsearch**

Elasticsearch is a search Engine based on Lucene Library. It Provides a distributed, Multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON document.

Elasticsearch is distributed and supports all the data types. Including numerical, textual, structured unstructured, and geospatial data.

Its built with Java, Elasticsearch is a NoSQL database. That means it stores data in an unstructured way and that you cannot use SQL to query it.

Which hides the internal complexities and provides full-text search capabilities with simple REST APIs.

Elasticsearch is Specifically designed to search an index of massive datasets in the order of petabytes.

Elasticsearch offers simple REST-based APIs, a simple HTTP interface, and uses schema-free JSON documents, making it easy to get started and quickly build applications for various use cases.

Initial release date: 8th February 2020

Written in: JAVA

Alternatives to Elasticsearch

* Vespa
* Typesense.
* Manticore Search
* Bleve

**How is Data Stored in Elasticsearch**

Elasticsearch stores data as JSON documents. Each document correlates a set of keys (name of fields or properties) with their corresponding values(strings, numbers, Booleans, dates, arrays of values, geolocations, or other types of data).

**Key Features of Elasticsearch**

* Scalable. It can be scalable across multiple nodes.
* Performance. It is very fast comparatively other search engines.
* Document Oriented
* Schema less.
* Open Source….etc

**Why Elastic search Better than SQL**

Elastic search Manages huge amounts of data: As a comparison to the traditional SQL database management systems that take more than 10 seconds to fetch required search query data. Elasticsearch can do that within a few microseconds.

Elasticsearch works faster when large amounts of data through indexing. The contains ready-made sets of data with which you are operating further ES-filters. So if you search With ES you haven’t to do a direct request to the database, as in MYSQL.

**What is Elasticsearch Index and Document**

An index can be thought of an optimized collection of documents and each document is a collection of fields, which are the key-value pairs that contain your data. By default, Elasticsearch indexes all data in every field and each indexed field has a dedicated, optimized data structure.

**Kibana**

Kibana is an open-source browser-based visualization tool mainly used to analyse large volume of logs in form of line graph, bar graph, pie charges, heat maps, region maps, coordinate maps, gauge, goals, timelion etc.

Kibana written in TypeScript, JavaScript

License Proprietary software (Elastic License 2.0)

**Where is Kibana data stored?**

Kibana stores its objects as documents in the Kibana index in Elasticsearch. The name of the index can be changed via the Kibana. Index configuration setting (starting with Kibana 4.2; prior to that this setting was name Kibana index).

**What is Kibana vs Grafana?**

Grafana’s design for analyzing and visualizing metrics such as system CPU, memory, disk and I/O utilization. The platform does not allow full-text data querying. Kibana, on the other hand, runs on top of Elasticsearch and used primarily for analyzing log messages.

A screenshot of a computer

Description automatically generated

**Topics**

* Installation of Logstash
* Installation of Elasticsearch
* Installation of Kibana
* ELK configuration

**Installation of Logstash**

**Now we are going with Kubernetes Installation, so we need to create Yaml file for Logstash.**

**Now we can Discuss about Elastic search**

**Topics   
 \* How to Register Repository in Elastic search  
 \* How to check master elected node and slaves  
 \* How to check list of Repository’s.**

**\* How to load data to into elastic search  
 \* How to check Number of indexes in elastic search  
 \* How to check data present inside the elastic search indexes**

**\* How to Take backup  
 \* How to check Number of snapshots Present In elastic search  
 \* How to Delete Data present In Elastic search  
 \* How to Restore in Elastic Search  
 \* How to check number of Indexes Present in Elastic search  
  
Here I am using custom image for My elastic search bcz I am registering my elastic search repository with s3 storage so I need to pass my aws access and secret into Elasticsearch so I will pass them using Dockerfile.  
  
 Here is My Docker file   
 A computer screen shot of a program

Description automatically generated  
  
Now you can see above docker file I was mentioned Init-keystore.sh file which container shell script which is used to run aws access and secret key passing script.  
  
Now I will show you my init-keystore .sh file  
 A computer screen shot of colorful text

Description automatically generated  
After Successful Image build I will deploy my elastic search file Here I will provide Elasticsearch yaml file at end.  
kubectl apply -f elasticsearch.yml  
How to Register Repository** In Elasticsearch, a repository Is a feature that allows you to store and manage snapshots of your Elasticsearch cluster’s data. Snapshots are essentially backups of your cluster’s indices and can be used for various purposes, such a disaster recovery, migrating data between clusters, or creating copies of your data for testing and development. **A computer screen with text on it

Description automatically generated  
curl -u elastic:password -X PUT "http://elastictestr-0.service.default.svc.cluster.local:9200/\_snapshot/my\_s3\_repo?pretty" -H 'Content-Type: application/json' -d'**

**{**

**"type": "s3",**

**"settings": {**

**"region": "my-region",**

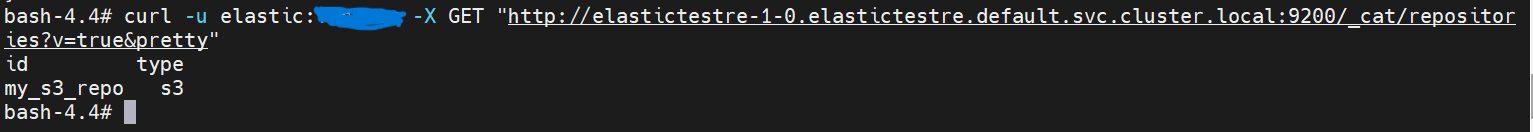
**"bucket": "bucketname",**

**"path\_style\_access": "true",**

**"protocol": "https"**

**}**

**}**

**'  
  
In start curl -u (give your username and password )(default user was elastic,password you can set earlier In init-script I shown you how to set password In that place you can give password it will set password for elatic user)  
 Here I am registering repository in Elasticsearch using Aws S3 storage Whenever I take backup In elastic search it will be stored in Aws S3 Storage  
 You can replace you S3 region, bucket name according to you  
 When it show acknowledged true means successfully created repo  
Now I will check Repository List in my Elasticsearch  
 #curl -u elastic:password-X GET "http:// elatictestre-1-0.service .default.svc.cluster.local:9200/\_cat/repositories?v=true&pretty" **Now you can see we have my\_s3\_repo Repository  
How to check Master Master nodes and slaves in Elasticsearch  
#curl -u elastic:password ‘elastictestre-1-0.service.default.svc.cluster.local:9200/\_cat/master?v’  
#curl -u elastic:password elastictestre-1-0.service.default.svc.cluster.local:9200/\_cat/nodes?v&h=id,ip,port,v,m  
A computer screen shot of a black screen

Description automatically generated  
  
Now We can See the Number Of Indexes Present In Elasticsearch  
In Elastic Search Data will stored In Indexes  
 In Elasticsearch, an Index is a data structure used to store and organize a collections of documents. It is one of the fundamental building blocks of Elasticsearch and is often compared to a database table in a relational database management system (RDMS). However, there are significant differences between Elasticsearch indices and RDBMS tables:  
#curl -u elastic:password -X GET "http://elastictestre-1-0.service.default.svc.cluster.local:9200/\_cat/indices?v&pretty"  
A screen shot of a computer

Description automatically generated  
Now we have only one inex wich geoip which is default index   
Now I will create index and insert data into it  
Now I will show you How to Insert Data into Elasticsearch

Here is the commands to Insert Data into Elastic search In Elastic search Data will be stored in Indexes   
  
#curl -u elastic:password-XPOST "http:// elastictestre-1-0.service.default.svc.cluster.local:9200/my-index/\_doc" -H 'Content-Type: application/json' -d '

{

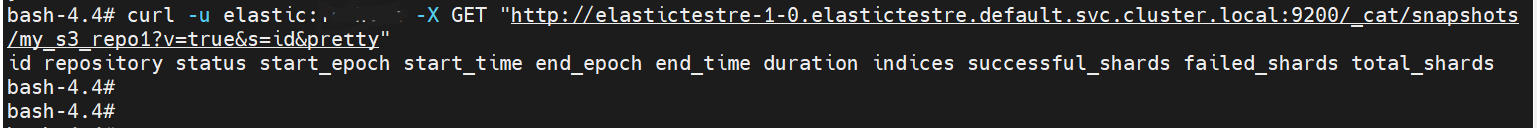
"name": "John Smith67",

"age": 30

}'  
A computer screen with text on it

Description automatically generated  
After Inserting data into Index now we can check number of Indexes present in Elasticsearch  
#curl -u elastic:password -X GET "http://elastictestre-1-0.service.default.svc.cluster.local:9200/\_cat/indices?v&pretty"  
A screen shot of a computer

Description automatically generated

Earlier we have only one index which is geoip but now we have two indexes   
Now we can check number of snapshots present in Elastic search   
 In Elasticsearch, a Snapshot refers to a point-in-time backup of your cluster’s data and indices. Snapshots are essential for data recovery, disaster recovery, data migration, and creating copies of your data for testing and development purposes. They provides a way to capture the state of your Elasticsearch indices and can be restored to recover data in case of data loss or to duplicate data in another cluster.  
#curl -u elastic:password –X GET “https://elastictestre-1-0.service.default.svc.cluster.local:9200/\_cat/snapshots/my\_s3\_repo1?v=true&s=id&pretty”

#curl -u elastic:Techtez -X PUT "http://elastictestre-1-0.service.default.svc.cluster.local:9200/\_snapshot/my\_hostip/snapshot\_3?wait\_for\_completion=true&pretty" -H 'Content-Type: application/json' -d'

{

"indices": "\*,-.\*"

}

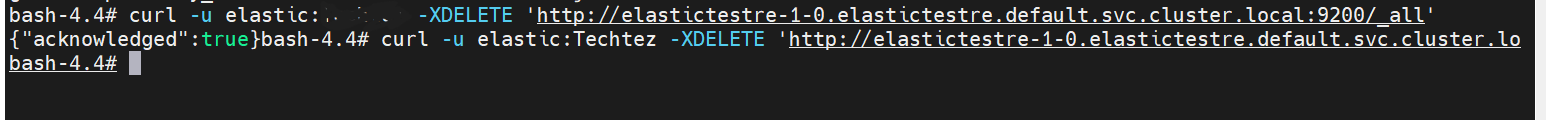
'

Now we don’t have any snapshots here Now I will take backup in Elasticsearch  
How to Take backup In Elasticsearch  
A screen shot of a computer

Description automatically generated

In Backup command we need to provide Name of our repository and snapshot name also here I was given snapshot name as snapshot\_test3  
Now you can see above output successful two, and it will display name of indexes which we have taken backup, In above command I have given \*\* which means I have taken entire cluster backup.  
#curl -u elastic:password –X GET “https://elastictestre-1-0.service.default.svc.cluster.local:9200/\_cat/snapshots/my\_s3\_repo1?v=true&s=id&pretty”  
Now we can check Snapshots count  
A black screen with white text

Description automatically generated

Earlier we have zero snapshots but at present we have one snapshot (snapshot\_test3)  
#curl -u elastic:password -XDELETE ‘https://elastictestre-1-0.service.default.svc.cluster.local:9200/\_all’  
Now We can Delete the data present in Our Elasticsearch and we can do restore  
  
  
You can see above curl command I am deleting all the data present in Elasticsearch.  
  
Now we can check the data in elastic search using curl command   
#curl -u elastic:password -X Get “http://elasticre-1-0.service.default.svc.luster.local:9200/\_cat/indices?v&pretty”  
A screen shot of a computer

Description automatically generated

At present We don’t have data in our Elastic search cluster because I was deleted the data. now I will do restore.  
#curl -u elastic:password -X POST <http://elastictestre-1-0.service.default.svc.cluster.local:9200/_snapshto/my_s3_repo/snapsot_test3/_restore?pretty> -H ‘Content-Type: application/json’ -d  
A computer screen with white text

Description automatically generated

Here you can see It was showing accepted true means Successfully we have done restore, In restore we have to provide snapshot name.  
Now we can check restored Data  
#curl -u elastic:password -X GET “https://elastictestre-1-0.service.default.svc.cluster.local:9200/\_cat/indices?v&pretty”  
A screen shot of a computer

Description automatically generated   
You can see here We restored my\_index8 successfully Now we can check restored data   
  
Now we can check restored data

#Curl -u elastic:password -u XGET <http://elastictestre-1-0.service.default.svc.cluster.local:9200/my_index8/_search?pretty> -H ‘Content-Type: application/json -d’

A black background with a black border

Description automatically generated

Now you can see the data in my\_index8   
john smith67 name   
age 30  
  
In Elastic search Yaml we have (cluster.initial\_master\_nodes) here we have to give the node names which have master eligibility

A black background with white text

Description automatically generated

How can we give Master eligibility for elasticnode (when you set node.master has true it will become master eligible node) and we need to give common cluster name for all the pods (here I am giving elasticsearch)

A black screen with white text

Description automatically generated  
  
Now we need to set discovery.seed\_hosts (Here we need to provide all the pod name which are present in quorum both master and slave names here)  
  
