Creating and Configuring Amazon Elastic search Service Domains :

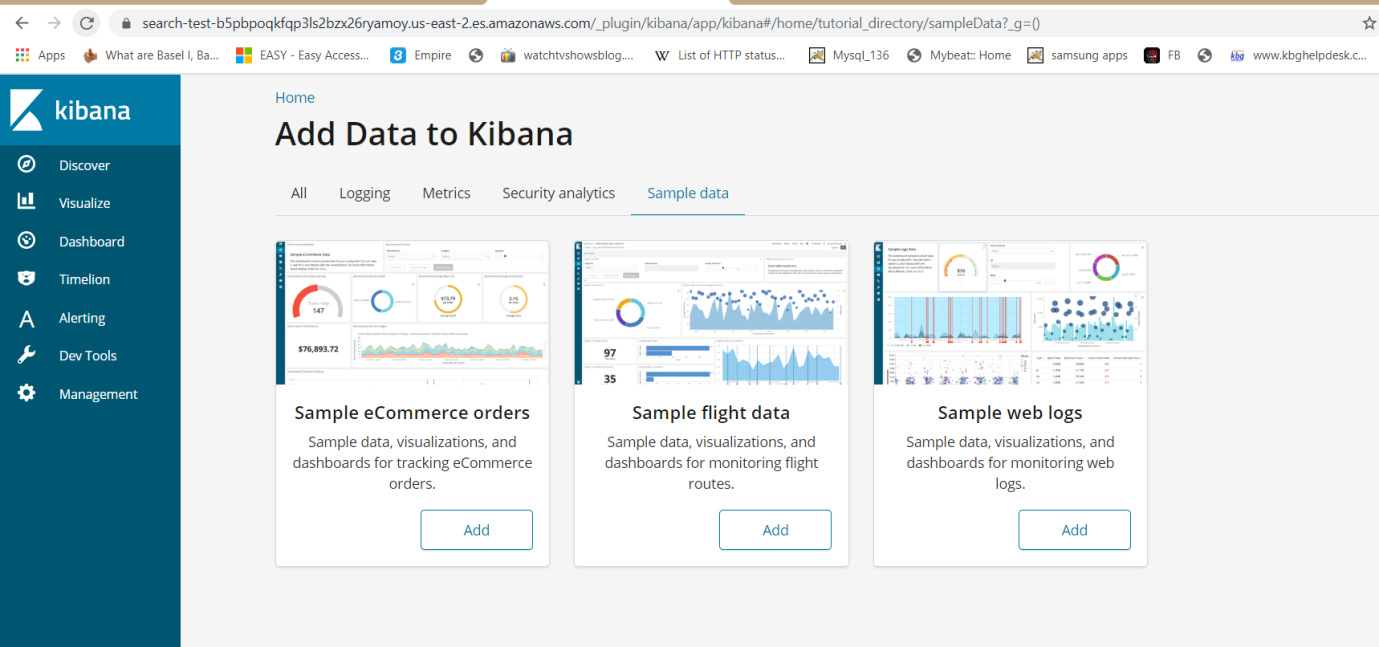
I have Used **Terraform**: v0.11.14 to create Elastic search Service cluster just because it is a managed service and managed by Amazon hence effort to manage the cluster can be avoided. Apart from that Amazon is provide multilayer of security which can be enabled for e.g Amazon Cognito for authentication or we can enable the ldap authentication .

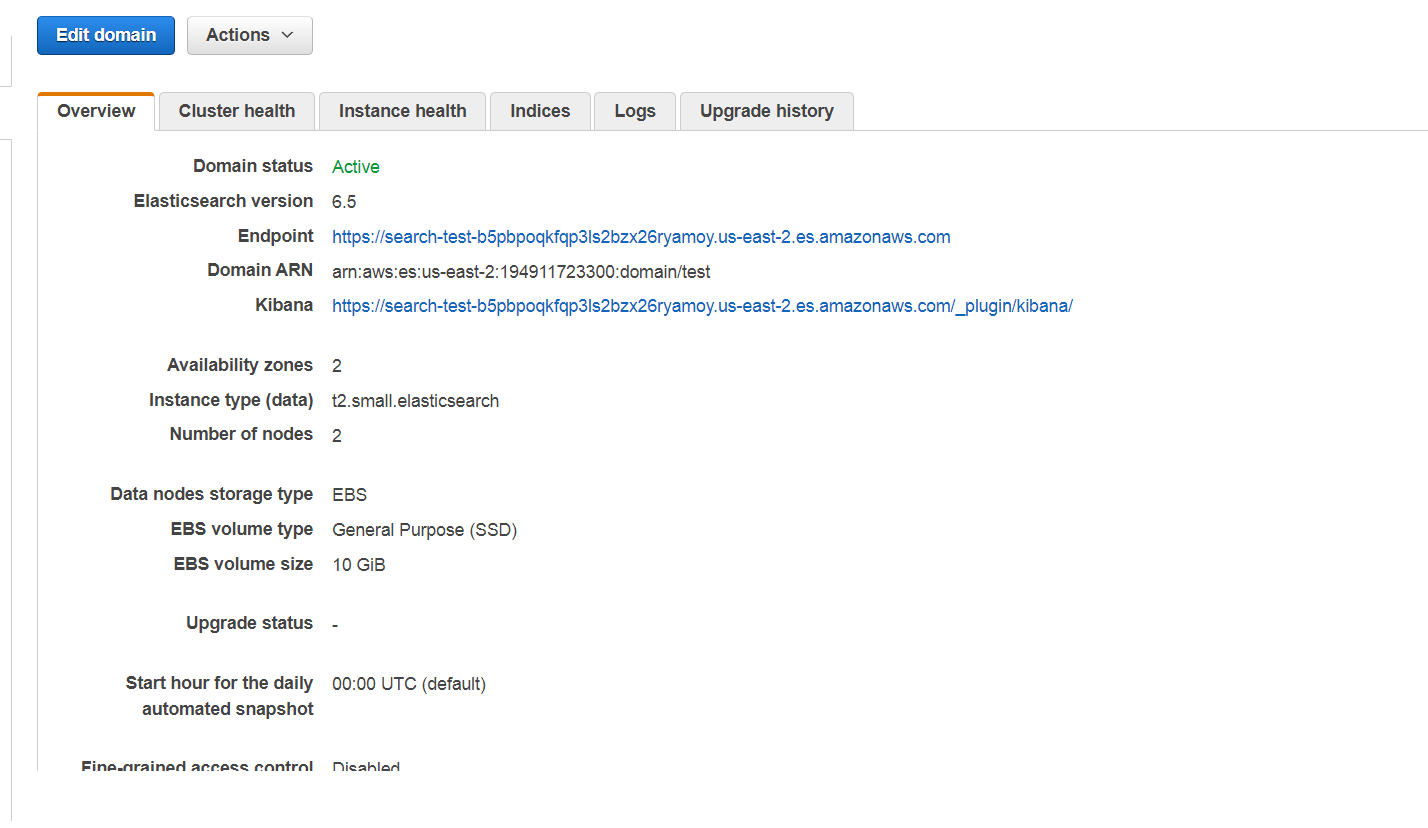
We can use Multiple Encryption supported for Elastic search Service

* Require HTTPS for all traffic to the domain
* Node-to-node encryption
* Enable encryption of data at rest

**Used AWS Elastic search Service:**

I have created cluster manually from aws console and then tried to automate the configuration using terraform resource aws\_elasticsearch\_domain .



For encryption we have used Node-to-node encryption . In Below example you can see Node-to-node encryption is disabled just because I have used t2.small.elastic search node and t2small does not support node to node encryption ,****

Bonus if you extend your code to create a cluster of 3 ElasticSearch nodes Some answers we are looking:

1. What did you choose to automate the provisioning and bootstrapping of the instance? Why?

Instead of provisioning and bootstrapping of the instance I have chosed the AWS managed service Elastic search Service cluster which more easy to configure cluster with in friction of second .Used terraform to automate the service . Code are placed on GIT.

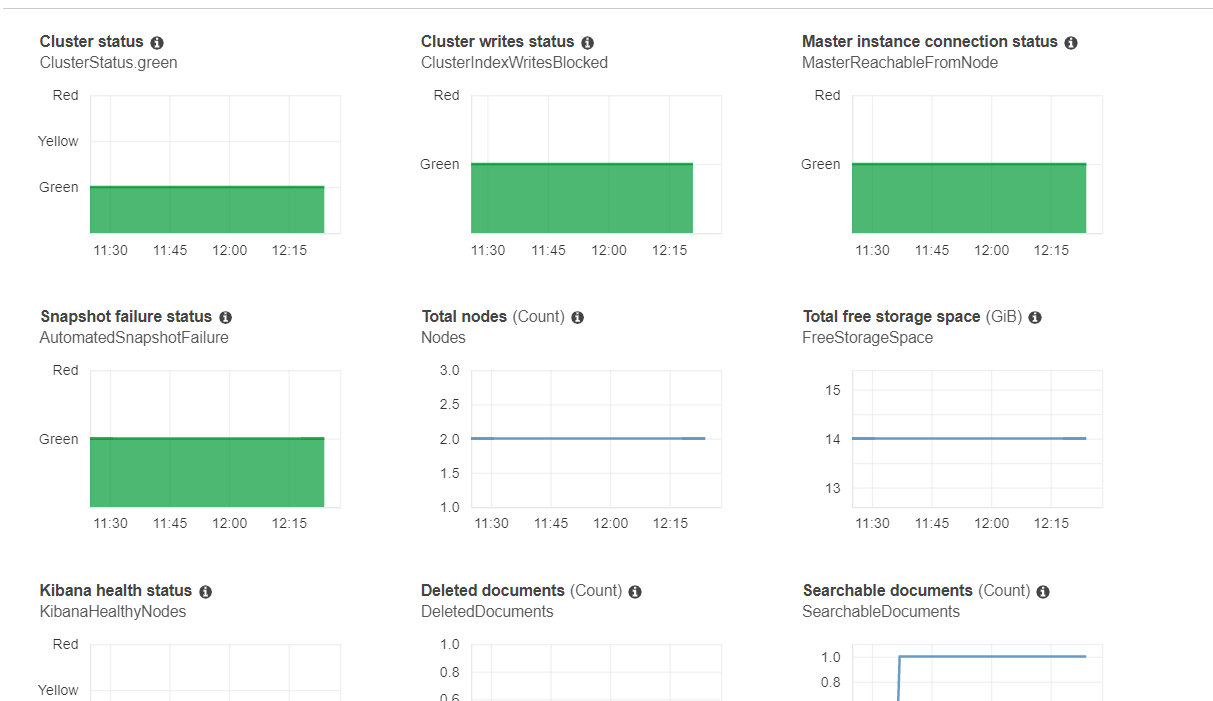
We can opt the option of provisioning the resource using terraform and configuration and deployment using IAC tool like chef and ansible . provisioning of resource is easy using terraform however configuration and deployment is not possible because I am using chef as configuration management tool and don’t have much handy with ansible .thats why I have created the managed service using terraform.

2. How did you choose to secure ElasticSearch? Why? 3. How would you monitor this instance?

We can monitor these cluster using the cluster health apart from that we can use cloud watch.

What metrics would you monitor?

Below are the cluster health we can monitor by default. in case of more metrices are required we have to configure cloud watch metrics.



Could you extend your solution to launch a secure cluster of ElasticSearch nodes?

Yes we can just by increasing the values of Instace\_count in variable.tf

What would need to change to support this use case?

5. Could you extend your solution to replace a running ElasticSearch instance with little or no downtime? How?

Will Opt for a blue green deployment policies .

6. Was it a priority to make your code well structured, extensible, and reusable? 7. What sacrifices did you make due to time?

Yes