**Elasticsearch.yml—>**

**---------------------------------- Network -----------------------------------**

**#**

**# By default Elasticsearch is only accessible on localhost. Set a different**

**# address here to expose this node on the network:**

**#network.host: 192.168.0.1**

**#**

**# By default Elasticsearch listens for HTTP traffic on the first free port it**

**# finds starting at 9200. Set a specific HTTP port here:**

**#**

**#http.port: 9200**

Network.host is the ip where the elasticsearch is available, by default it is available on localhost only, you can change it to another ip eg: 0.0.0.0 if you want to access it from anywhere.

Generally it is not needed to change.

**# --------------------------------------------------------------------------------**

**# Enable security features**

**xpack.security.enabled: true**

**xpack.security.enrollment.enabled: true**

**# Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents**

**xpack.security.http.ssl:**

**enabled: true**

**keystore.path: certs/http.p12**

**# Enable encryption and mutual authentication between cluster nodes**

**xpack.security.transport.ssl:**

**enabled: true**

**verification\_mode: certificate**

**keystore.path: certs/transport.p12**

**truststore.path: certs/transport.p12**

**# Create a new cluster with the current node only**

**# Additional nodes can still join the cluster later**

**cluster.initial\_master\_nodes: ["ip-172-31-2-240"]**

**# Allow HTTP API connections from anywhere**

**# Connections are encrypted and require user authentication**

**http.host: 0.0.0.0**

* Xpack.security.enabled should be kept true to create a fleet server.
* If Xpack.security.enabled is true then http.ssl and transport.ssl should also be kept true.
* These secure the data transfer between the elasticsearch and fleet server, other entities.
* Xpack.security.enrollment.enabled , it enables kibana and other services to enroll in elasticsearch using the enrollment token.
* Http.host Allows HTTP API connections from anywhere, ie, if this is 0.0.0.0 then you curl elasticsearch ip from anywhere.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Kibana.yml→**

server.port: 5601

server.host: 0.0.0.0

The server.host should be 0.0.0.0 to allow kibana to be accessed from anywhere on the internet.

xpack.security.encryptionKey should be set in kibana.yml to generate alarms and other features.

Kibana automatically generates a random key for the use but it changes after kibana restarts hence it is best practice to set the encryption key.

Generate the 32 character using the command :

bin/kibana-encryption-keys generate

<https://www.elastic.co/guide/en/kibana/current/kibana-encryption-keys.html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Heartbeat.yml →**

**heartbeat.monitors:**

**- type: http**

**enabled: true**

**id: my-weather-app**

**name: my-weather-app**

**urls: ["http://localhost:80"]**

**schedule: '@every 10s'**

**- type: http**

**enabled: true**

**id: AppRTC**

**name: AppRTC**

**urls: ["http://localhost:3000"]**

**schedule: '@every 10s'**

We can add multiple monitors in the heartbeat.yml to monitor uptime.

In the above example there are two monitor, with the different url which we want to monitor.

Id is used to query in the elasticsearch and used to uniquely identify this monitor in elasticsearch, and the name is displayed in the human readable form in kibana dashboard.

**setup.kibana:**

**host: "**[**http://172.31.2.240:5601**](http://172.31.2.240:5601)**"**

The host ip is the private ip of the instance on which the kibana is running, Public ip can also be used but it changes every time we restart the host instance on which kibana is running.

**output.elasticsearch:**

**hosts: ["172.31.2.240:9200"]**

**protocol: "https"**

**username: "elastic"**

**password: "6EfwsKZhgdvyxlbhnj6X"**

**ssl.ca\_trusted\_fingerprint:"f8f01d5cd7cedbf44e58073c831362a30f4a8e6c3bb480a140228a5a5139032c"**

* The host ip is also the private ip of the instance on which the elastic is running and is used due to the same reason as the setup.kibana host ip.
* Https is used here because we have enabled the xpack ssl, tls security in the elasticsearch.yml.
* The username and password are of the superuser which is created by elasticsearch at the time of installation, you can also create a new user. These are required to authenticate the heartbeat service to send the data to the elastic.
* The ssl trusted fingerprint is also to authenticate the heartbeat , ssl certificates can also be used but using the fingerprint is more efficient.

* If you do not have the es-ca-trusted-fingerprint , then you can copy it from the fleet server.
* Fleet > Add agent > Fleet Policy > add host > generate token > **Install Commands**
* At the last line of ./elastic-agent install you will find *fleet-server-ca-trusted-fingerprint*, copy the value of this parameter