**About ELK**

Elasticsearch is an open-source, RESTful, distributed search and analytics engine built on Apache Lucene. Since its release in 2010, Elasticsearch has quickly become the most popular search engine and is commonly used for log analytics, full-text search, security intelligence, business analytics, and operational intelligence use cases.

**How does it work?**

You can send data in the form of JSON documents to Elasticsearch using the API or ingestion tools such as [Logstash](https://aws.amazon.com/elasticsearch-service/logstash/) and [Amazon Kinesis Firehose](https://aws.amazon.com/kinesis/data-firehose/). Elasticsearch automatically stores the original document and adds a searchable reference to the document in the cluster’s index. You can then search and retrieve the document using the Elasticsearch API. You can also use [Kibana](https://aws.amazon.com/elasticsearch-service/kibana/), an open-source visualization tool, with Elasticsearch to visualize your data and build interactive dashboards.

We can use ELK stack in different areas like monitoring your application, application performance, and application Infrastructure.

**Application Infrastructure monitoring with ELK**

Elasticsearch requires OpenJDK in order to run the elasticsearch engine.

1. Install the OpenJDK 8 from a [PPA repository](https://launchpad.net/~openjdk-r/+archive/ubuntu/ppa):

sudo add-apt-repository ppa:openjdk-r/ppa

1. Update the system package cache and install:

sudo apt-get update

sudo apt-get install openjdk-8-jdk

**Note:** In case, if you have multiple java installation run the below commands to set Java 8 as the default version

sudo update-alternatives --config java

**Follow the below steps to install elasticsearch on Ubuntu distribution.**

**Installing from the APT repository edit**

1. Download and install the public signing key:

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

1. You may need to install the apt-transport-https package on Debian before proceeding:

sudo apt-get install apt-transport-https

1. Save the repository definition to /etc/apt/sources.list.d/elastic-7.x.list:

echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a

1. Now install elasticsearch with the below command

sudo apt-get update && sudo apt-get install elasticsearch

1. Before starting elasticsearch edit a few settings in elasticsearch.yml file.

Line 17: cluster.name: <Change to custom name>

Line 23: node.name: <Change to custom name>

Line 55: network.host: "0.0.0.0"

Line 59: http.port: 9200

Line 75: discovery.type: single-node

(This line is necessary when you want to run single node cluster)

Save the elasticsearch.yml configuration file and exit.

6. Now start the elasticsearch engine with the below command.

sudo service elasticsearch start

7. Verify elasticsearch is up and running or not

sudo service elasticsearch status

**Note:** In case if you see any issue with starting to verify elasticsearch logs and troubleshoot in the **/var/log/elasricsearch/elasricsearch.log** file.

**Installing Kibana on the same host:**

We can use the same apt repository to install kibana package also

sudo apt-get update && sudo apt-get install kibana

Sudo service kibana start

**Install metricbeat on production servers:**

1. Download and install the public signing key for Linux distributions:

sudo rpm --import https://packages.elastic.co/GPG-KEY-elasticsearch

1. Create a file with a .repo extension (for example, elastic.repo) in your /etc/yum.repos.d/directory and add the following lines:

[elastic-6.x]

name=Elastic repository for 6.x packages

baseurl=https://artifacts.elastic.co/packages/6.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

Now install **metricbeat**

sudo yum install metricbeat

1. To configure the Beat to start automatically during boot, run:

sudo chkconfig --add metricbeat

4. Open the metricbeat.yml file to configure elasticsearch and kibana IP.

1. Cd /etc/metricbeat
2. Vi metricbeat.yml
   1. Line 67: host: "<KIBANA\_IP>:5601"
   2. Line 94: hosts: ["elasticsearch\_IP:9200"]

Save and exit

5. Load Kibana dashboards before starting metricbeat

metricbeat setup --dashboards

6. Start metricbeat service

sudo service metricbeat start

Now verify index is created in elasticsearch with metricbeat name

Command: curl elasticsearch\_IP:9200/\_cat/indices

**Output:**

green open .kibana\_task\_manager rWcHKNvAQkeIeSKXLo5H6Q 1 0 2 0 12.7kb 12.7kb

yellow open **metricbeat-7.0.0** LwftEjePShmFVgqc4Q6Kew 1 1 1332 0 1.7mb 1.7mb

green open .kibana\_1 GKCbe\_bzQOqAFnDuXAlMEQ 1 0 345 123 564.1kb 564.1kb

Above output indicates that remote system metrics is dumping into the elasticsearch engine

Verify system status is available in the kibana dashboard.

Follow the metric beat installation guide for all other prod servers. Once you finished we should see metrics in the kibana dashboard.

Browse this URL to login into the Kibana to see metrics.

*http://<Kibana IP:5601>*

But it is not easy to remember IP and port for every time. So we have created one record set for kibana called **kibana.balluun.com** and pointed to Kibana server.



When we hit the above URL, it will transfer requests to kibana server on port 80. So we have created Nginx as a reverse proxy to redirect traffic to 5601. Below configuration is for redirecting traffic to kibana. Add this in /*etc/nginx/sites-enabled/default*

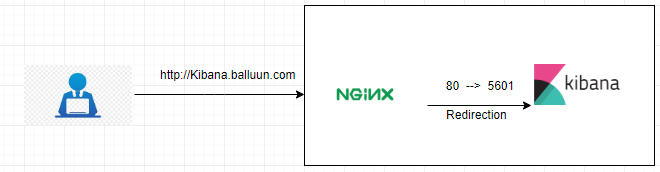
location / {

proxy\_pass http://10.0.3.64:5601/;

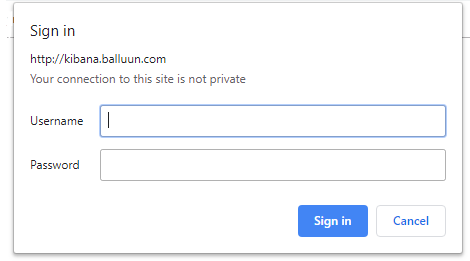
auth\_basic "Private Property";

auth\_basic\_user\_file /etc/nginx/.htpasswd;

}



As we are using open source license we do not have any authentication, here I have configured basic Nginx authentication which will prompt for before redirection.



If credentials are validated successfully it will redirect us to Kibana dashboard

