**KIBANA**

**=========**

The Elastic Stack – is a collection of open-source produced by elastic which allows you to search, analyze and vizualize the logs generated from any source in any format, a practice known as centralized logging. Centralized logging can be very useful to identifying the problem with your servers or applications.

The Elastic stack has four major components:

* **Logstash** – It’s the data processing component in ELK-Stack which sends incoming data to Elastic-Search.
* **ElasticSearch** – Which stores all the collected data.
* **Kibana** – Visualize the collected data in graphically.
* **Beats** – Lightweight, single-purpose data shippers that can send data from hundreds or thousands of machine to either logstash or elastic-search.

**Prerequisites:**

**==========**

* ubuntu 16.04 or more
* RAM : minimum 4GB
* CPU : minimum 2
* Java 8 – which is required by Elastic-search and logstash. Note that java 9 is not supported.
* Nginx installed on your server.

**STEP- 1 : Installing and Configuring ElasticSearch**

**=======================================**

Run the following command to import the GPG key into apt,

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

Next, add the Elastic source list into sources.list.d directory, where apt will look for new sources.

**$ echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list**

Next, update your package lists so APT will read new Elastic source.

**$ sudo apt update**

then install the elasticsearch by using the following command,

**$ sudo apt install elasticsearch**

Once the elasticsearch is finished installing, open elasticsearch main configuration file **“/*etc/elasticsearch/elasticsearch.yml”***

Elasticsearch listens traffic from everywhere on port 9200.

**“/etc/elasticsearch/elasticsearch.yml”**

………..

network.host: 0.0.0.0

………...

save and close the elasticsearch.yml file, Then start the elasticsearch service.

**$ sudo systemctl start elasticsearch**

**$ sudo systemctl enable elasticsearch**

You can test whether your Elasticsearch service is running by sending the HTTP request.

**$ curl -X GET “localhost:9200”**

you will see a response showing some basic information about your local node.

**STEP- 2 : Installing and Configuring the Kibana Dashboard**

**=============================================**

According to the official documentation, you should install kibana only after elasticsearch.

You have already added the ElasticSearch source in the previous step, you can just install the remaining Elastic-stack using apt:

**$ sudo apt-get install kibana**

**$ sudo systemctl enable kibana**

**$ sudo systemctl start kibana**

Because kibana is configured to only localhost, we must set up a reverse proxy to allow external access to it. We will use nginx for this purpose,which should already be installed on your server.

First, use the **openssl** command to create an administrative kibana user which you will use to access the kibana web interface.

The following command will create the administrative Kibana user and password, and store them in the htpasswd.users file.

**$ echo "kibanaadmin:`openssl passwd -apr1`" | sudo tee -a /etc/nginx/htpasswd.users**

Enter username and password at the prompt. Remember or take note of this login, as you will need it to access the kibana web-interface.

Next, we will create an nginx server block file.

**$ sudo nano /etc/nginx/sites-available/default**

Add the following code block into the file,

server {

listen 80;

server\_name localhost 192.168.108.15;

auth\_basic "Restricted Access";

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

location / {

proxy\_pass http://192.168.108.15:5601;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

When you are finished, save and close the file.Next, Enable the new configuration by creating a symbolic link to the sites-enabled directory.

**$ sudo ln -s /etc/nginx/sites-available/default /etc/nginx/sites-enabled/default**

Then check the configuration for syntex errors.

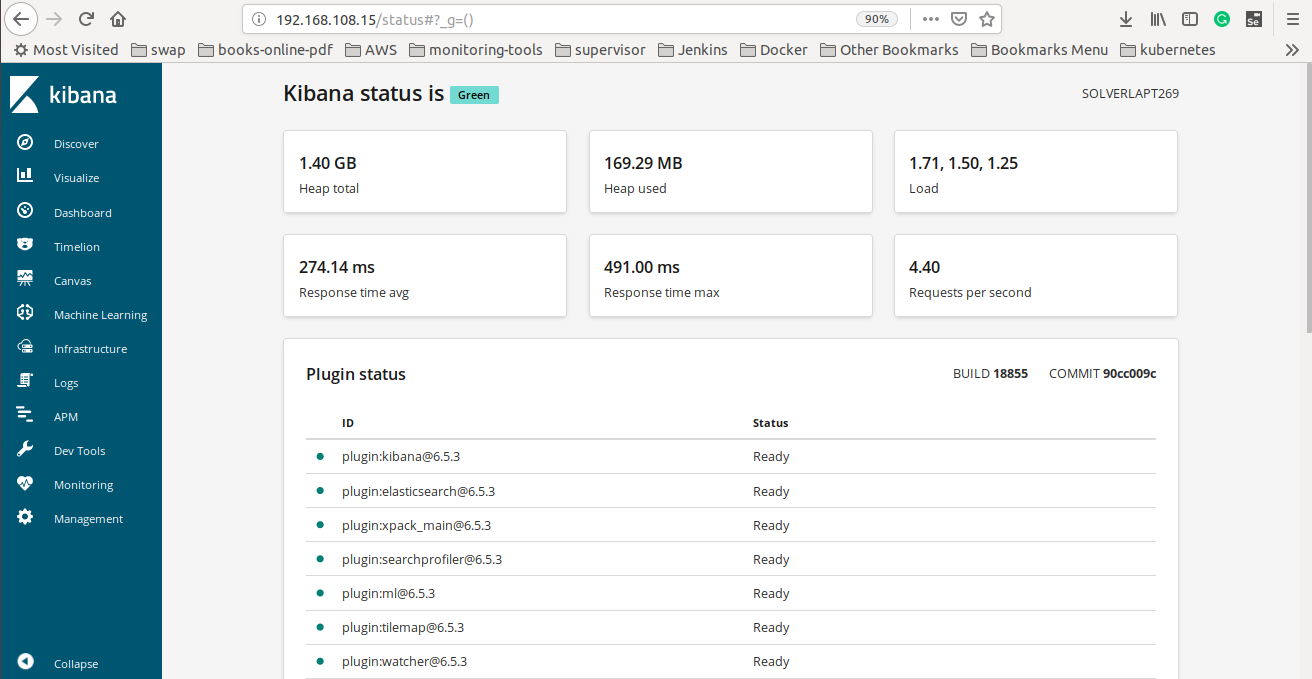
**$ sudo nginx -t**

Once your syntex is ok, Go ahead and restart the nginx service.

**$ sudo systemctl restart nginx**

Kibana is now accessible via your FQDN or public ip address of you elastic-stack server. You can check the kibana server’s status page by navigating to the following address and entering login credentials when prompted.

**$** <http://your-ip-address/status>



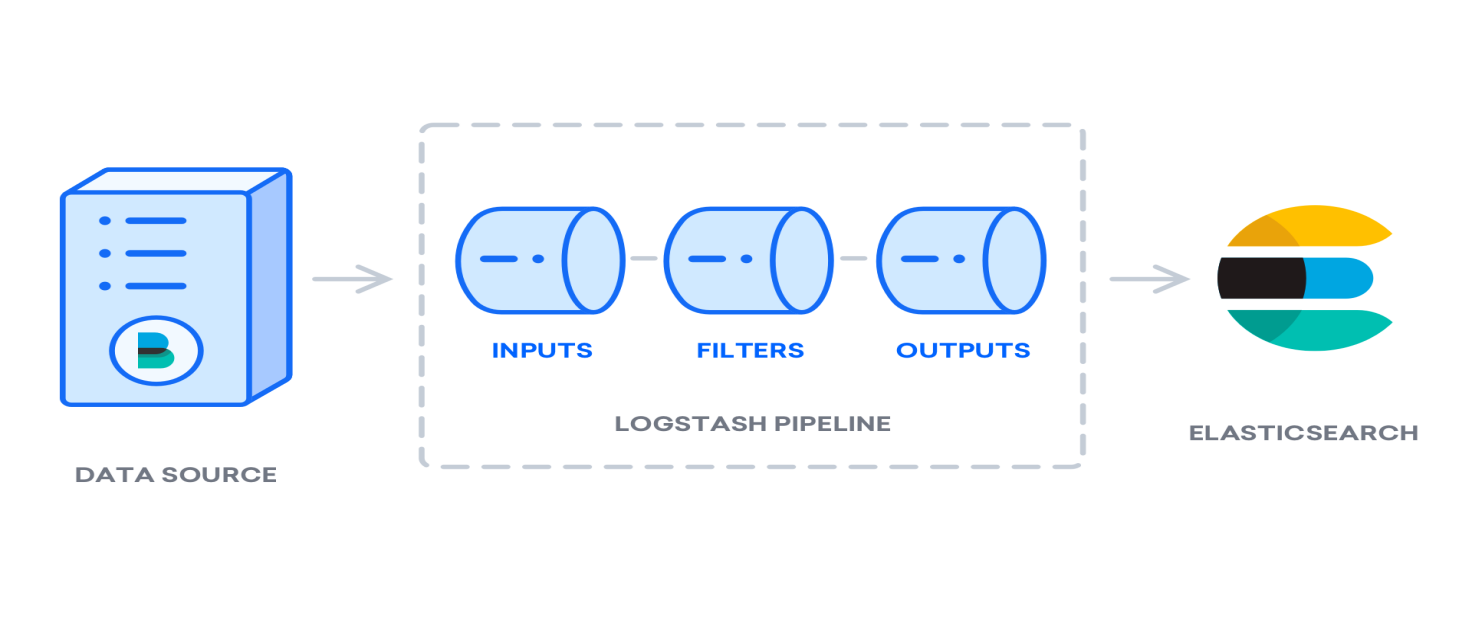
**STEP- 3 : Installing and Configuring Logstash**

**======================================**

Although it’s possible for Beats to send data directly to the Elasticsearch database, we recommend using Logstash to process the data. This will allow you to collect data from different sources, transform it into a common format, and export it to another database.

**$ sudo apt-get install logstash**

After installing Logstash, you can move on to configuring it. Logstash’s configuration files are written in the JSON format and reside in the “/*etc*/logstash/conf.d/” directory. A logstash pipeline has two required elements, **input** and **output ,** and one optional **Filter.** The input plugin data comes from a source, the filter plugin process the data, and the output plugins write the data to a destination.



Create a configuration file called **02-beats-input.conf** where will setup the Filebeat input (**/etc/logstash/conf.d/**).

**/etc/logstash/conf.d/02-beats-input.conf**

input {

beats {

port => 5044

}

}