Install ELK Stack on ubuntu 18.04:

Overview:

Install ELK Stack.

Enable TLS, configure, enable Postgres module.

1. Download and install the public signing key:

[root@node1 ~]# **wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add**

2. Installing from the APT repository:

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

[root@node1 ~]# **sudo apt-get install apt-transport-https**

3. 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 /etc/apt/sources.list.d/elastic-7.x.list

4. You can install the Elasticsearch Debian package with:

[root@node1 ~]# **sudo apt-get update && sudo apt-get install elasticsearch**

5. Download and install the Debian package manually:

The Debian package for Elasticsearch v7.11.2 can be downloaded from the website and installed as follows:

[root@node1 ~]# wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.11.2-amd64.deb

[root@node1 ~]# wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.11.2-$ amd64.deb.sha512

[root@node1 ~]# shasum -a 512 -c elasticsearch-7.11.2-amd64.deb.sha512

[root@node1 ~]# sudo dpkg -i elasticsearch-7.11.2-amd64.deb

6. Elasticsearch can be started and stopped as follows:

[root@node1 ~]# sudo systemctl start elasticsearch.service

7. Now let's install Kibana and Metricbeat:

The Debian package for Kibana v7.11.2 can be downloaded

from the website and installed as follows:

```
[root@node1 ~]#  wget https://artifacts.elastic.co/downloads/kibana/kibana-7.11.2-amd64.deb

[root@node1 ~]#  shasum -a 512 kibana-7.11.2-amd64.deb

[root@node1 ~]#  sudo dpkg -i kibana-7.11.2-amd64.deb
```

8. Run apt-get update, and the repository is ready for use.

For example, you can install Metricbeat by running:

```
[root@node1 ~]#  sudo apt-get update && sudo apt-get install metricbeat
```

9. Configure /etc/hosts file:

```
[root@node1 ~]# vi /etc/hosts
```

Add the below:

```
X.X.X.X node1.elastic.test.com:9200 node1
```

10. Create SSL certificates and enable TLS for Elasticsearch on node1:

Set environment variables (adapt these variables path depending on where and how Elastic was downloaded):

```
[root@node1 ~]# ES_HOME=/usr/share/elasticsearch

[root@node1 ~]# ES_PATH_CONF=/etc/elasticsearchearch
```

11. Create tmp folder

```
[root@node1 ~]# mkdir tmp

[root@node1 ~]# cd tmp/

[root@node1 tmp]# mkdir cert_blog
```

Create instance yaml file

```
[root@node1 cert_blog]# vi ~/tmp/cert_blog/instance.yml
```

add the instance information to yml file

```
instances:

          - name: 'node1'

            dns: [ 'node1.elastic.test.com' ]
```

12. Generate CA and server certificates (once Elasticsearch is installed)

```
[root@node1 tmp]# cd $ES_HOME

[root@node1 elasticsearch]# bin/elasticsearch-certutil cert --keep-ca-key --pem --in ~/tmp/cert_blog/instance.yml --out ~/tmp/cert_blog/certs.zip

bin/elasticsearch-certutil cert --keep-ca-key --ca-cert --in ~/etc/tmp/cert_blog/instance.yml --out ~/etc/tmp/cert_blog/certs.zip
```

13. Unzip certs:

```
[root@node1 elasticsearch]# cd ~/tmp/cert_blog

[root@node1 cert_blog]# unzip certs.zip -d ./certs

# Copy cert to /etc/elasticsearch

[root@node1 ~]# cd $ES_PATH_CONF

[root@node1 elasticsearch]# pwd

/etc/elasticsearch

[root@node1 elasticsearch]# mkdir certs

[root@node1 elasticsearch]# cp ~/tmp/cert_blog/certs/ca/ca* ~/tmp/cert_blog/certs/node1/* certs

[root@node1 elasticsearch]# ll certs

total 12

-rw-r--r--. 1 root elasticsearch 1834 Apr 12 08:47 ca.crt

-rw-r--r--. 1 root elasticsearch 1834 Apr 12 08:47 ca.key

-rw-r--r--. 1 root elasticsearch 1509 Apr 12 08:47 node1.crt

-rw-r--r--. 1 root elasticsearch 1679 Apr 12 08:47 node1.key

[root@node1 elasticsearch]#
```

14. Configure elasticsearch.yml

```
 root@node1 elasticsearch]# vi elasticsearch.yml

## add the following contents

node.name: node1
```

```
network.host: node1.elastic.test.com

xpack.security.enabled: true

xpack.security.http.ssl.enabled: true

xpack.security.transport.ssl.enabled: true

xpack.security.http.ssl.key: certs/node1.key

xpack.security.http.ssl.certificate: certs/node1.crt

xpack.security.http.ssl.certificate_authorities: certs/ca.crt

xpack.security.transport.ssl.key: certs/node1.key

xpack.security.transport.ssl.certificate: certs/node1.crt

xpack.security.transport.ssl.certificate_authorities: certs/ca.crt

discovery.seed_hosts: [ "node1.elastic.test.com" ]

cluster.initial_master_nodes: [ "node1" ]
```

15. Set built-in user password:

```
[root@node1 elasticsearch]# cd $ES_HOME

[root@node1 elasticsearch]# bin/elasticsearch-setup-passwords auto -u "https://node1.elastic.test.com:9200"

Initiating the setup of passwords for reserved users elastic,apm_system,kibana,logstash_system,beats_system,remote_monitoring_user.

The passwords will be randomly generated and printed to the console.

Please confirm that you would like to continue [y/N] y

Changed password for user apm_system

PASSWORD apm_system = <apm_system_password>

Changed password for user kibana

PASSWORD kibana = <kibana_password>

Changed password for user logstash_system

PASSWORD logstash_system = <logstash_system_password>

Changed password for user beats_system

PASSWORD beats_system = <beats_system_password>
```

Changed password for user remote_monitoring_user

PASSWORD remote_monitoring_user = <remote_monitoring_user_password>

Changed password for user elastic

PASSWORD elastic = <elastic_password>

16. Enable TLS for Kibana on node1:

[root@node1 ~]# KIBANA_HOME=/usr/share/kibana

[root@node1 ~]# KIBANA_PATH_CONFIG=/etc/kibana

Copy certs folder from /etc/elasticsearch to /etc/kibana:

[root@node1 kibana]# ls config/certs

total 12

ca.crt my-kibana.crt

my-kibana.key

17. Configure kibana.yml:

[root@node1 kibana]# vi kibana.yml

server.name: "my-kibana"

server.host: "kibana.local"

server.ssl.enabled: true

server.ssl.certificate: /etc/kibana/config/certs/my-kibana.crt

server.ssl.key: /etc/kibana/config/certs/my-kibana.key

elasticsearch.hosts: ["https://node1.elastic.test.com:9200"]

elasticsearch.username: "kibana"

elasticsearch.password: "<kibana_password>"

elasticsearch.ssl.certificateAuthorities: ["/etc/kibana/config/certs/ca.crt"]

Now, start the below in order:

[root@node1 kibana]# systemctl start elasticsearch.server

[root@node1 kibana]# systemctl start kibana.service

[root@node1 kibana]# systemctl start metricbeat

- To enable Postgres module on Metricbeat:

```
[root@node1 kibana]# metricbeat modules enable postgresql
```

Navigate to /etc/metricbeat/modules/postgresql.yml

```
[root@node1 kibana]#  vi /etc/metricbeat/modules/postgresql.yml
```

Add the following:

```
# Module: postgresql
# Docs: https://www.elastic.co/guide/en/beats/metricbeat/7.11/metricbeat-module-postgresql.html
 - module: postgresql
   enabled: true
   metricsets:
     - database
     - bgwriter
     - activity
# it’s best to query database every 60s
          period: 60s
          hosts: ["postgres://x.x.x.x:5432?sslmode=disable","postgres://x.x.x.x:5432?sslmode=disable"]
            username: "postgres"
          password: "InsertYourPostgresPassword"
```

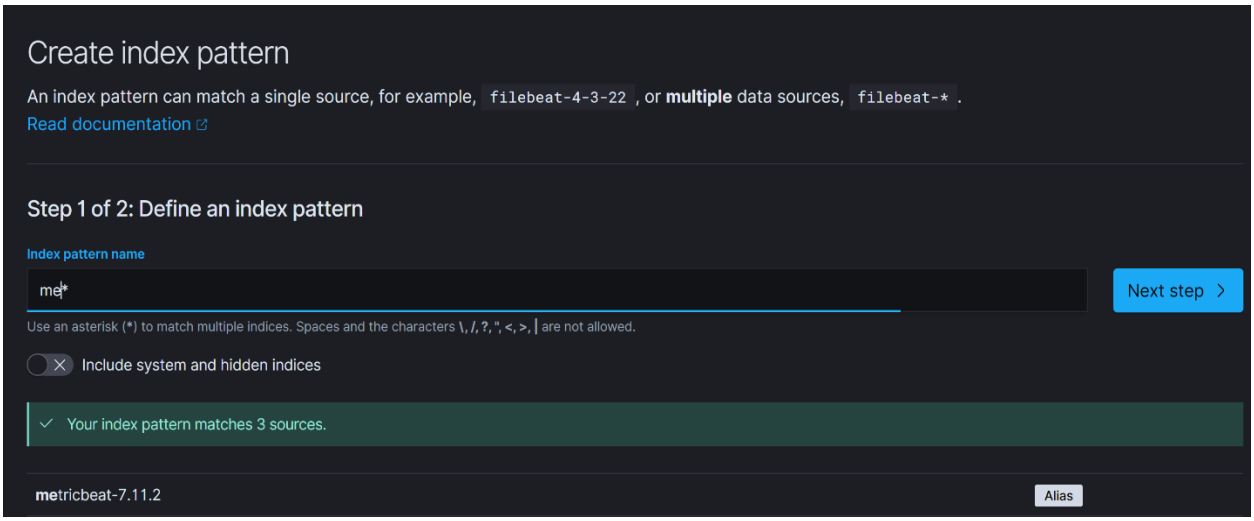
Note that I'm adding two Postgres database.

```
[root@node1 kibana]#  systemctl restart metricbeat
```

To enable default dashboard:

```
[root@node1 kibana]#  metricbeat setup --dashboards
[root@node1 kibana]#   metricbeat setup -e
```

Now navigate to Kibana and create index for Metricbeat:



Access dashboard: