

Create two linux servers:

server1 => install and configure kibana and elasticsearch with basic username and password authentication

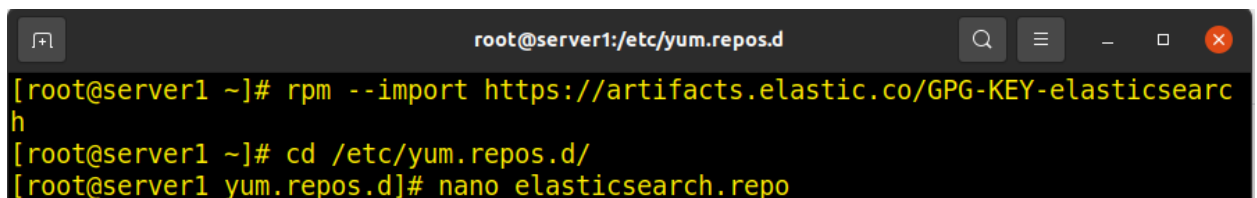
server2 => install and configure metricbeat.

Installing and Configuring ElasticSearch and Kibana on server 1 (Centos7)

a) `# rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch`

b) Create a file called **elasticsearch.repo** in the `/etc/yum.repos.d/` directory as,

```
# cd /etc/yum.repos.d
# nano elasticsearch.repo
```

A terminal window titled 'root@server1:/etc/yum.repos.d' with search, menu, and window control icons. The terminal shows the following commands and their outputs:

```
[root@server1 ~]# rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
h
[root@server1 ~]# cd /etc/yum.repos.d/
[root@server1 yum.repos.d]# nano elasticsearch.repo
```

c) Copy the contents below in the file,

```
[elasticsearch]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=0
autorefresh=1
type=rpm-md
```

```
GNU nano 2.3.1      File: elasticsearch.repo

[elasticsearch]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=0
autorefresh=1
type=rpm-md
```

d) # **sudo yum install --enablerepo=elasticsearch elasticsearch**

```
root@server1:/etc/yum.repos.d

Is this ok [y/d/N]: y
Downloading packages:
elasticsearch-7.15.2-x86_64.rpm | 325 MB 01:17
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Installing : elasticsearch-7.15.2-1.x86_64 1/1
### NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd
sudo systemctl daemon-reload
sudo systemctl enable elasticsearch.service
### You can start elasticsearch service by executing
sudo systemctl start elasticsearch.service
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
Verifying : elasticsearch-7.15.2-1.x86_64 1/1

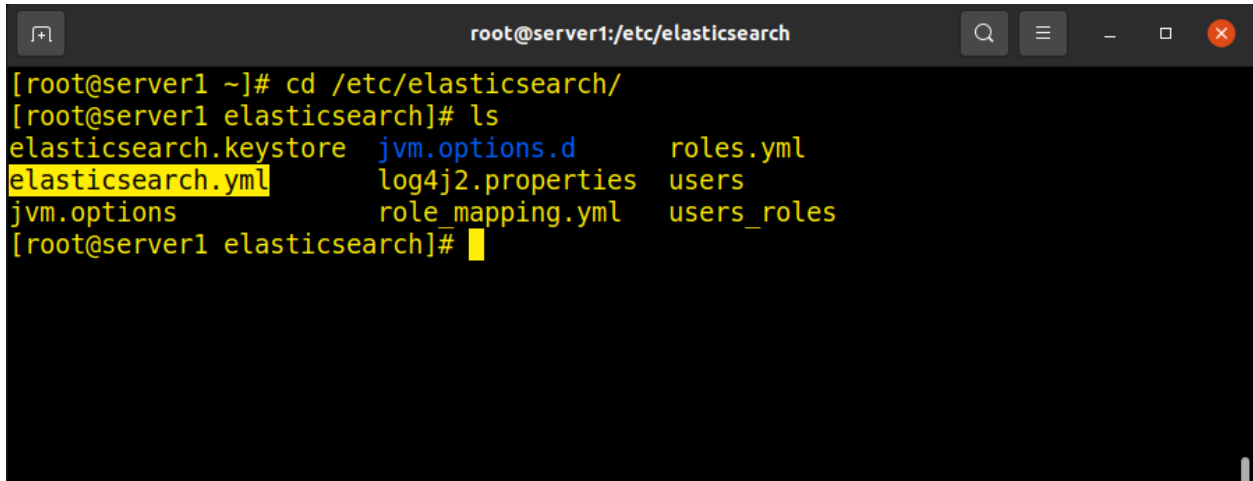
Installed:
  elasticsearch.x86_64 0:7.15.2-1

Complete!
[root@server1 yum.repos.d]#
```

Now the **elasticsearch** is successfully installed.

Now we have to configure the config file of elasticsearch to open it in our browser.

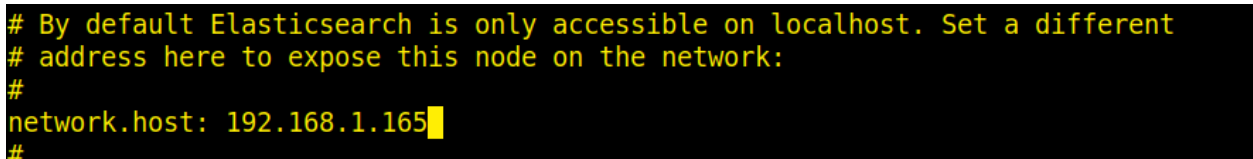
We can find the elasticsearch files in **/etc/elasticsearch** directory as,

A terminal window titled 'root@server1:/etc/elasticsearch' showing the command 'ls' being executed. The output lists several files: 'elasticsearch.keystore', 'jvm.options.d', 'roles.yml', 'elasticsearch.yml' (highlighted in yellow), 'log4j2.properties', 'users', 'jvm.options', 'role_mapping.yml', and 'users_roles'. The prompt is '[root@server1 elasticsearch]#'.

```
root@server1:/etc/elasticsearch
[root@server1 ~]# cd /etc/elasticsearch/
[root@server1 elasticsearch]# ls
elasticsearch.keystore  jvm.options.d      roles.yml
elasticsearch.yml       log4j2.properties  users
jvm.options             role_mapping.yml   users_roles
[root@server1 elasticsearch]#
```

We edit the highlighted file i.e. **elasticsearch.yml** as,

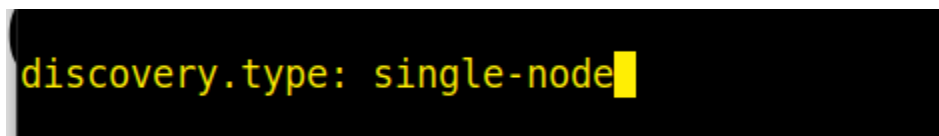
nano elasticsearch.yml

A terminal window showing the configuration of 'network.host' in the 'elasticsearch.yml' file. The text shows a comment about exposing the node on the network, followed by the line 'network.host: 192.168.1.165' which has a yellow cursor at the end.

```
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 192.168.1.165
#
```

Also add the below,

discovery.type: single-node

A terminal window showing the configuration of 'discovery.type' in the 'elasticsearch.yml' file. The text shows the line 'discovery.type: single-node' with a yellow cursor at the end.

```
discovery.type: single-node
```

Now we can access the elasticsearch in our browser.

systemctl enable elasticsearch

systemctl start elasticsearch

We have the ip of our Server1 system i.e. **192.168.1.165** where es is configured so,

```
← → ↻ ⚠ Not secure | 192.168.1.165:9200

{
  "name" : "server1",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "oyqq0YSNR4C8zHDD-cFuXA",
  "version" : {
    "number" : "7.15.2",
    "build_flavor" : "default",
    "build_type" : "rpm",
    "build_hash" : "93d5a7f6192e8a1a12e154a2b81bf6fa7309da0c",
    "build_date" : "2021-11-04T14:04:42.515624022Z",
    "build_snapshot" : false,
    "lucene_version" : "8.9.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

*here, **9200** is the default port for elasticsearch.*

Now, we add the below for the authentication config in our **elasticsearch.yml** file,

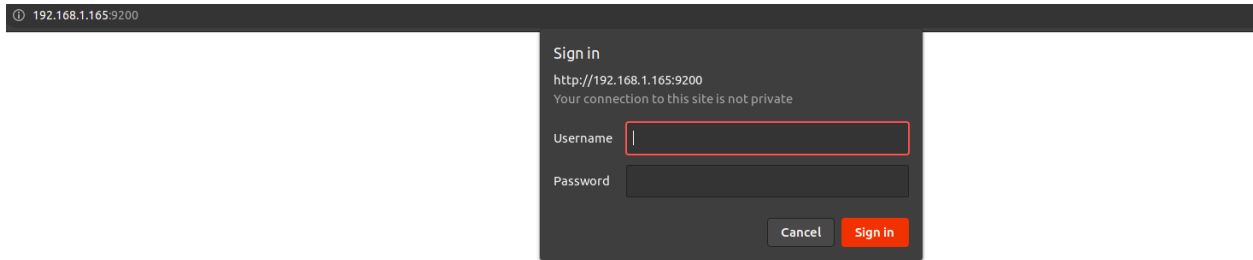
xpack.security.enabled: true

xpack.security.authc.api_key.enabled: true

```
xpack.security.enabled: true
xpack.security.authc.api_key.enabled: true
discovery.type: single-node
```

systemctl restart elasticsearch

Now lets check the browser if authentication security was success or not,



We can see, it's asking for username and password. Thus our auth protection was successful. Now let's add user and password so that we can sign in and access the elasticsearch;

bash usr/share/elasticsearch/bin/elasticsearch-setup-passwords interactive

```
root@server1:/etc/elasticsearch
You will be prompted to enter passwords as the process progresses.
Please confirm that you would like to continue [y/N]y

Enter password for [elastic]:
Reenter password for [elastic]:
Enter password for [apm_system]:
Reenter password for [apm_system]:
Enter password for [kibana_system]:
Reenter password for [kibana_system]:
Enter password for [logstash_system]:
Reenter password for [logstash_system]:
Enter password for [beats_system]:
Reenter password for [beats_system]:
Enter password for [remote_monitoring_user]:
Reenter password for [remote_monitoring_user]:
Changed password for user [apm_system]
Changed password for user [kibana_system]
Changed password for user [kibana]
Changed password for user [logstash_system]
Changed password for user [beats_system]
Changed password for user [remote_monitoring_user]
Changed password for user [elastic]
[root@server1 elasticsearch]#
```

Now we can sign in to our browser using username as **elastic** and our set-up password.

Installing and Configuring Kibana on server 1 (Centos7)

- a) **# rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch**
- b) Create a file called **kibana.repo** in **/etc/yum.repos.d/** directory as,
cd /etc/yum.repos.d
nano kibana.repo

```
[root@server1 elasticsearch]# rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
[root@server1 elasticsearch]# cd /etc/yum.repos.d/
[root@server1 yum.repos.d]# nano kibana.repo
```

- c) Copy the contents below in the file,

```
[kibana-7.x]
name=Kibana repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

```
GNU nano 2.3.1 File: kibana.repo

[kibana-7.x]
name=Kibana repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

d) # sudo yum install kibana

```
root@server1:/etc/yum.repos.d
Transaction Summary
=====
Install 1 Package

Total download size: 277 M
Installed size: 749 M
Is this ok [y/d/N]: y
Downloading packages:
kibana-7.15.2-x86_64.rpm | 277 MB 01:49
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : kibana-7.15.2-1.x86_64 1/1
Creating kibana group... OK
Creating kibana user... OK
Created Kibana keystore in /etc/kibana/kibana.keystore
  Verifying : kibana-7.15.2-1.x86_64 1/1

Installed:
  kibana.x86_64 0:7.15.2-1

Complete!
[root@server1 yum.repos.d]#
```

Now kibana is configured as,

```
# cd /etc/kibana
# nano kibana.yml
```

Add the below in the yml file,

```
server.host: 0.0.0.0
elasticsearch.username: "kibana_system"
elasticsearch.password: "centos"
```

```
server.host: 0.0.0.0
elasticsearch.username: "kibana_system"
elasticsearch.password: "centos"

# Kibana is served by a back end server.
#server.port: 5601
```

We can also add 32 character encryption key with,

xpack.encryptedSavedObjects.encryptionKey: “ at least 32 character value”

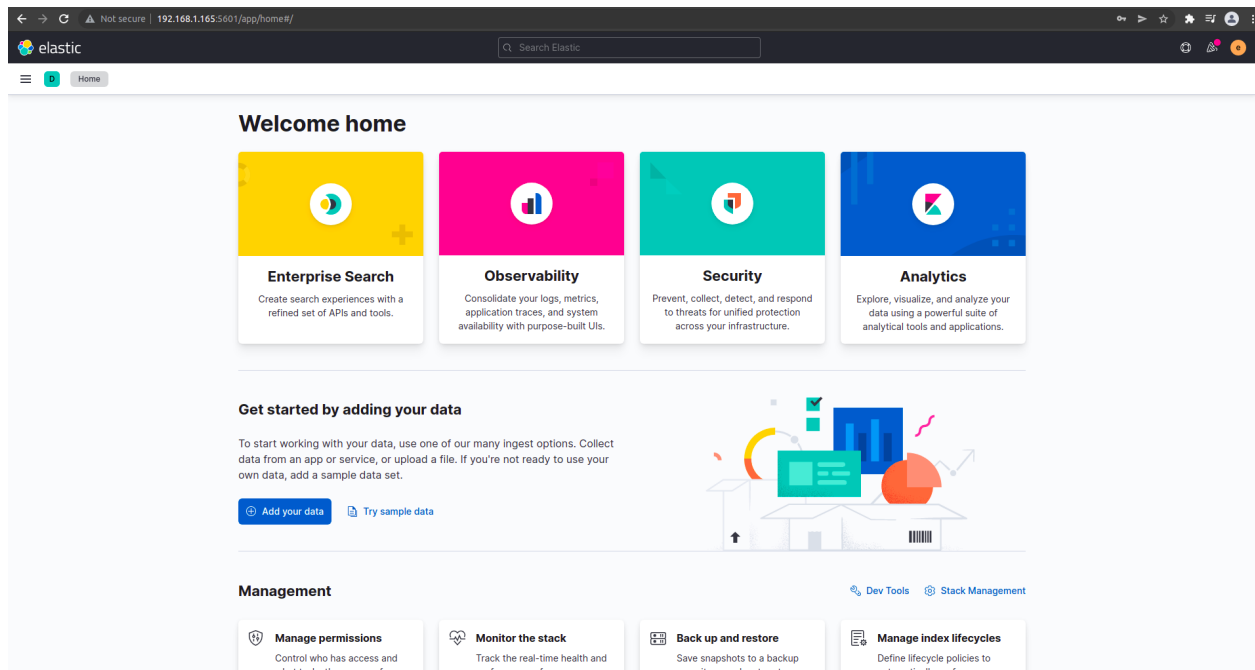
Also change the **elasticsearch.hosts** as below,

```
# The URLs of the Elasticsearch instances to use for all your queries.  
elasticsearch.hosts: ["http://192.168.1.165:9200"]
```

Now we restart the kibana server to see in browser,

\$ systemctl restart kibana

Now we go to **192.168.1.165:5601** and give our required credentials (username and password) to get the dashboard as,



Installing and Configuring Metricbeat on Server 2 (Centos 7)

Metricbeat helps you monitor your servers and the services they host by collecting metrics from the operating system and services.

curl -L -O

https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-7.15.2-x86_64.rpm

sudo rpm -vi metricbeat-7.15.2-x86_64.rpm

```
[root@server2 ~]# curl -L -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-7.15.2-x86_64.rpm
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 40.8M  100 40.8M    0     0 4051k      0  0:00:10  0:00:10 --:--:-- 4652k
[root@server2 ~]# sudo rpm -vi metricbeat-7.15.2-x86_64.rpm
warning: metricbeat-7.15.2-x86_64.rpm: Header V4 RSA/SHA512 Signature, key ID d88e42b4: NOKEY
Preparing packages...
metricbeat-7.15.2-1.x86_64
[root@server2 ~]#
```

Activate and enable the metricbeat,

systemctl enable metricbeat

systemctl start metricbeat

systemctl status metricbeat

```
[root@server2 ~]# systemctl status metricbeat
● metricbeat.service - Metricbeat is a lightweight shipper for metrics.
   Loaded: loaded (/usr/lib/systemd/system/metricbeat.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://www.elastic.co/beats/metricbeat
[root@server2 ~]# systemctl enable metricbeat
Created symlink from /etc/systemd/system/multi-user.target.wants/metricbeat.service to /usr/lib/systemd/system/metricbeat.service.
[root@server2 ~]# systemctl start metricbeat
[root@server2 ~]# systemctl status metricbeat
● metricbeat.service - Metricbeat is a lightweight shipper for metrics.
   Loaded: loaded (/usr/lib/systemd/system/metricbeat.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2021-12-03 04:33:18 EST; 4s ago
     Docs: https://www.elastic.co/beats/metricbeat
   Main PID: 18784 (metricbeat)
   CGroup: /system.slice/metricbeat.service
```

nano to the `/etc/metricbeat/metricbeat.yml` file and modify as below,

```
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["192.168.1.165:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  username: "elastic"
  password: "centos"
```

```
# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify an additional path, the scheme is required: http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  host: "192.168.1.168:5601"
```

192.168.1.165 is the ip where our kibana is configured.

Now in our host machine, allow the incoming requests for **5601** and **9200** ports,

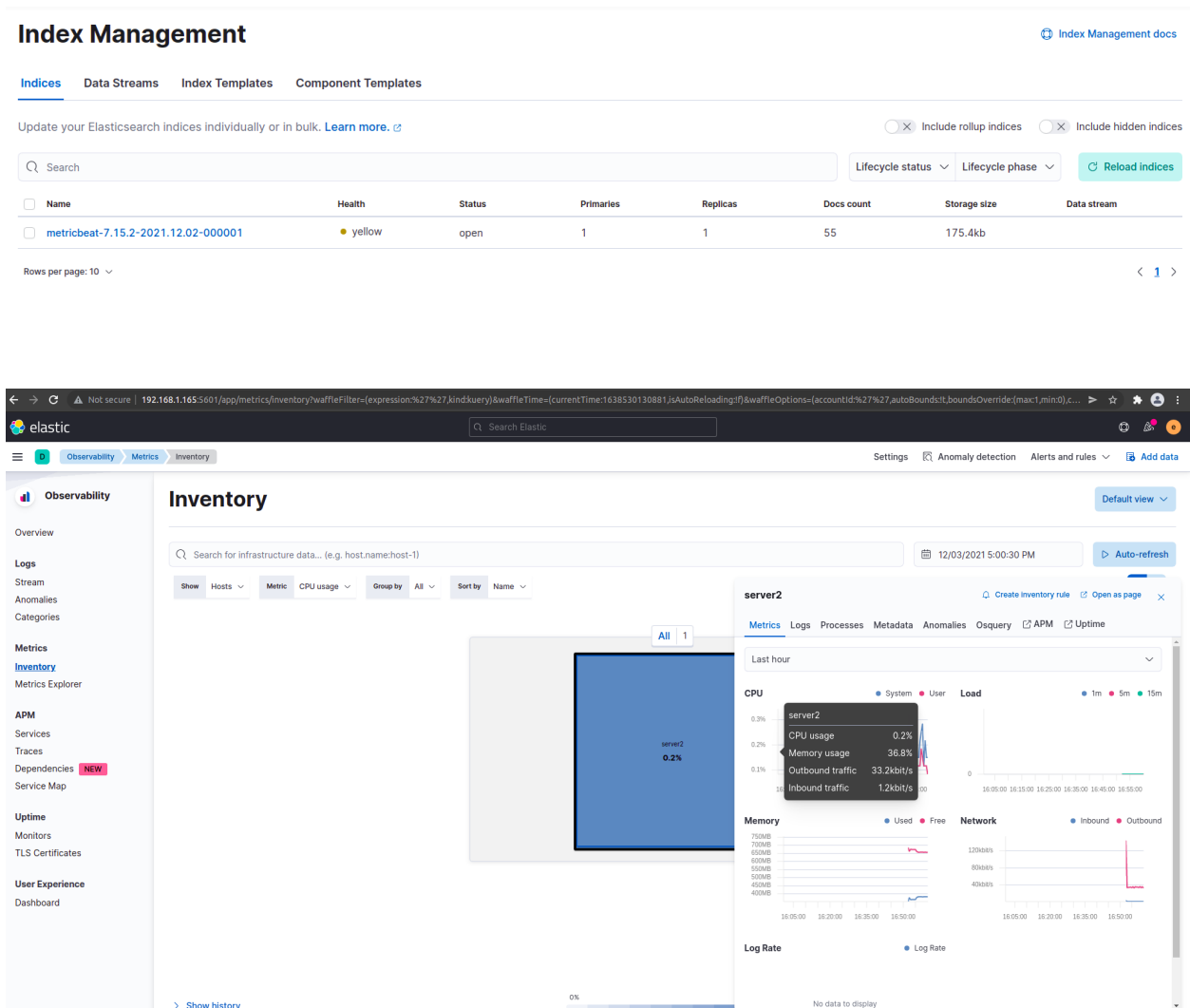
```
lostinservice@lostinservice: ~
root@server2:/etc/metricbeat x lostinservice@lostinservice: ~
lostinservice@lostinservice:~$ sudo ufw allow from 192.168.1.160 to 192.168.1.169
port 5601
Skipping adding existing rule
lostinservice@lostinservice:~$ sudo ufw allow from 192.168.1.160 to 192.168.1.169
port 9200
Rule added
lostinservice@lostinservice:~$ sudo ufw reload
Firewall reloaded
lostinservice@lostinservice:~$
```

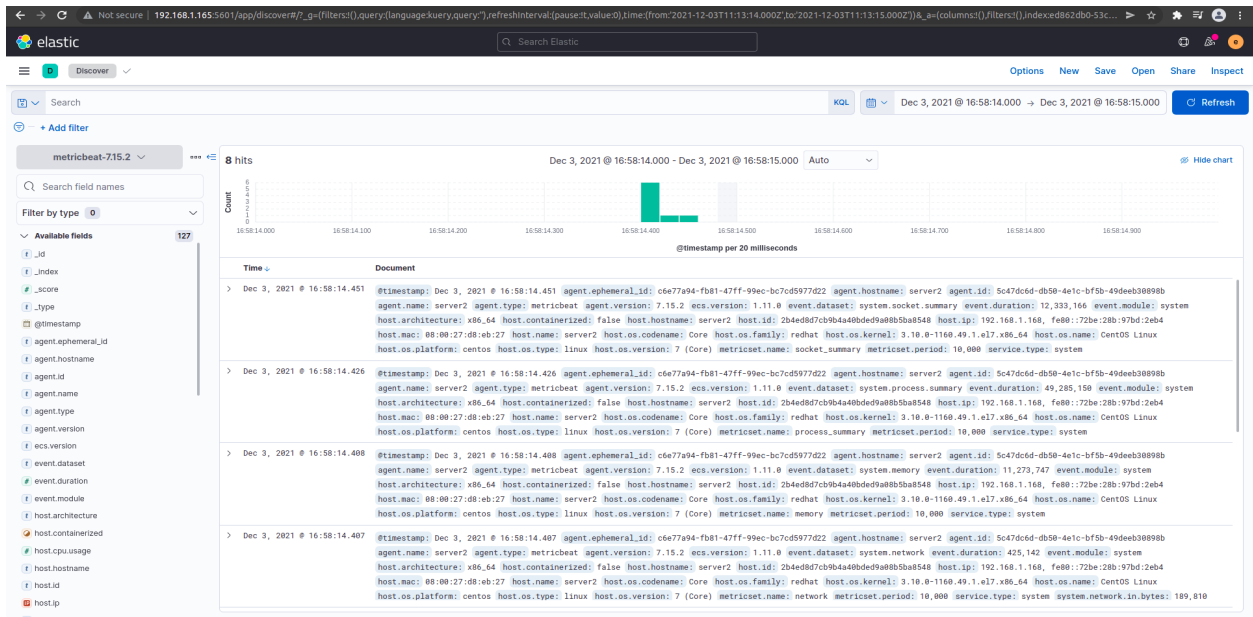
Run the metricbeat,

```
# systemctl restart metricbeat
```

```
# metricbeat -e
```

We can see the metricbeat configured successfully as,





Collect metric from following sources in server1 and send them to elasticsearch. Store them in an index named "server1-metrics".

- Memory usage
- Disk usage
- Load average

We edit the config file to collect the metric as,

```
# cd /etc/metricbeat
# nano metricbeat.yml
```

```
[root@server2 ~]# cd /etc/metricbeat/
[root@server2 metricbeat]# ls
fields.yml  metricbeat.reference.yml  metricbeat.yml  modules.d
[root@server2 metricbeat]# nano metricbeat.yml
```

Add the below in it,

```
GNU nano 2.3.1 File: metricbeat.yml

metricbeat.modules:
- module: system
  metricsets:
    - memory
    - diskio
    - load
  index: "server1-metrics"
  enabled: true
  period: 4s
```

Restart the metricbeat to see the collected metrics as,

systemctl restart metricbeat

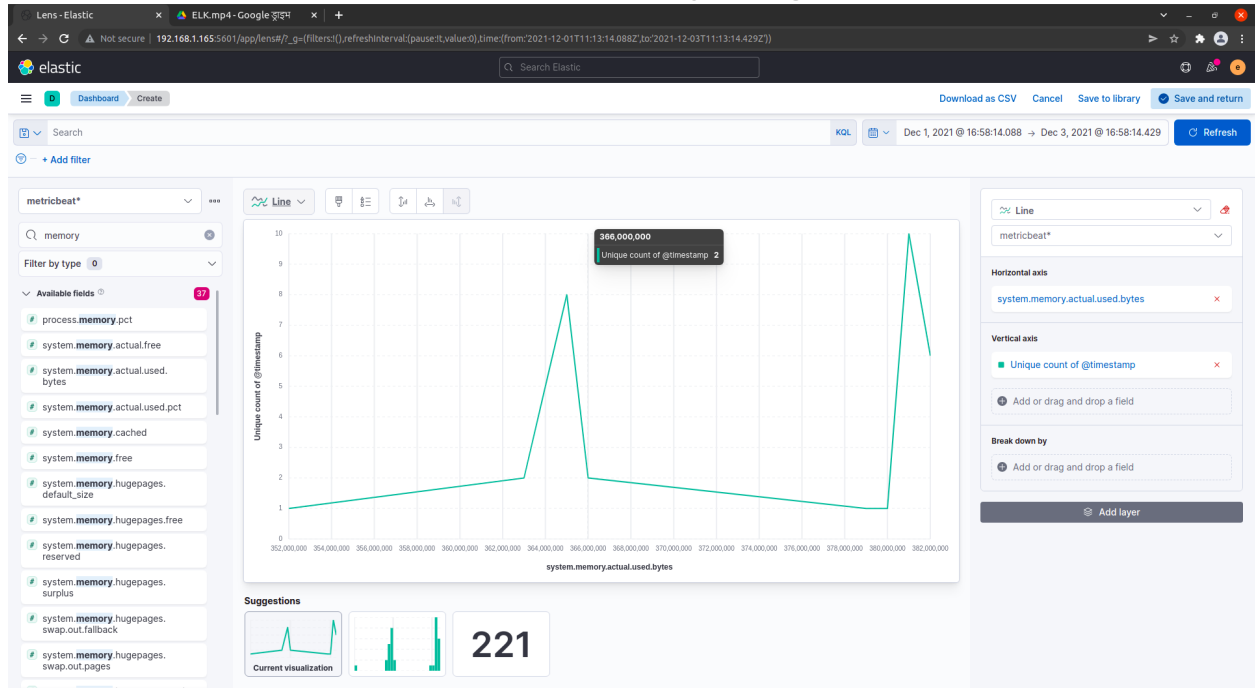
The screenshot shows the Elastic Index Management interface. On the left is a sidebar with navigation links for Management, Data, Alerts and Insights, and Security. The main content area is titled 'Index Management' and shows a table of indices. The 'server1-metrics' index is highlighted. To the right of the table, a detailed view of the 'server1-metrics' index is shown, including its health status (yellow), status (open), primaries (1), docs count (61), storage size (613.9kb), and aliases (none).

Name	Health	Status
<input type="checkbox"/> server1-metrics	yellow	open
<input type="checkbox"/> metricbeat-7.15.2-2021.12.02-000001	yellow	open

server1-metrics	
Health	yellow
Status	open
Primaries	1
Docs Count	61
Storage Size	613.9kb
Aliases	none

Create a dashboard in kibana and generate visual report(line graph) for Memory usage and load average of server1 with relation to time

Visual Report of Line Graph for Memory usage wrt. Timestamp,

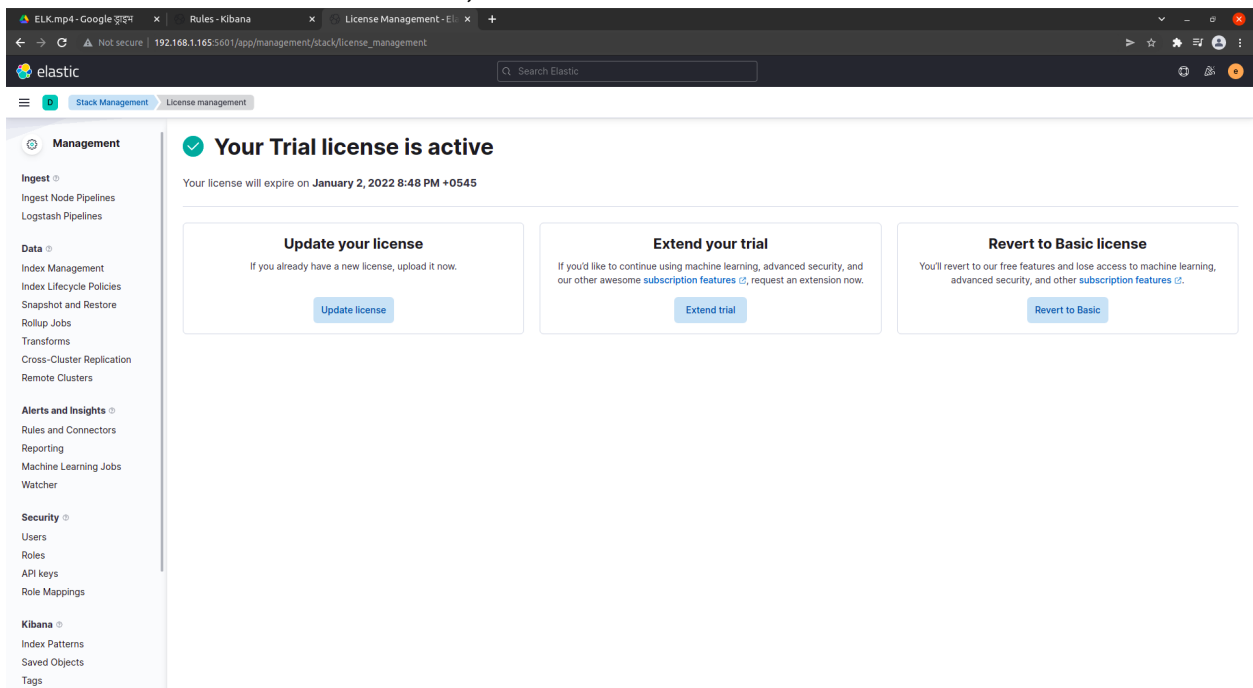


Visual Report of Line Graph for load avg of server 1 wrt. Timestamp,

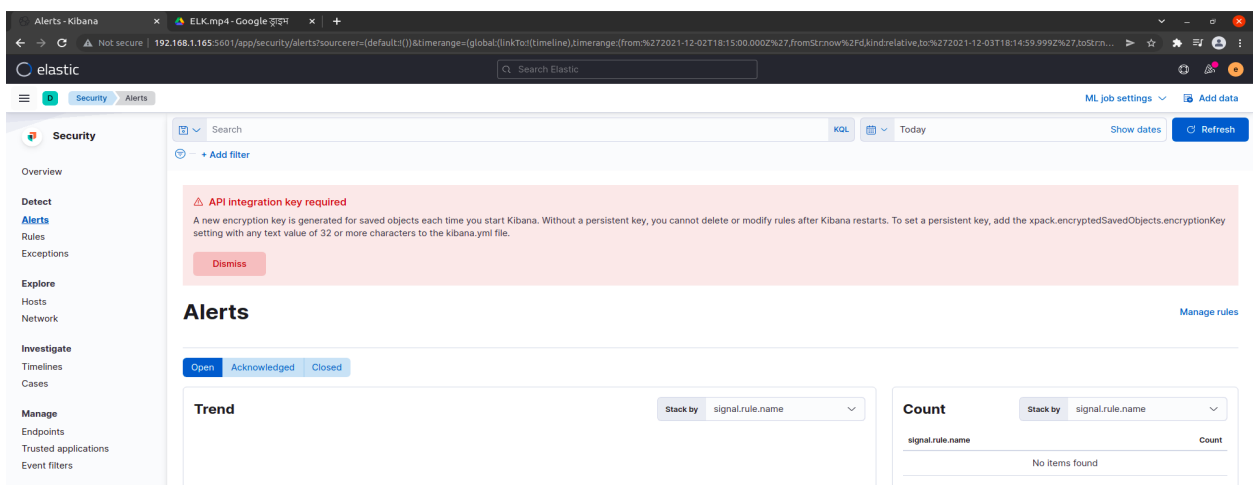


Generate alerts through the kibana system for following thresholds a. when memory usage > 80% for the last 2 minutes send an alert to a slack channel b. When Disk usage > 70% send alerts to a slack channel c. When load average > 1 for last 2 minutes send alert to a slack channel

Activate the trial license first,



For generating alert we need to generate certain rules,



We see the alert home as above. We need to set a persistent key first as suggested above in **kibana.yml** file as,

sudo nano /etc/kibana/kibana.yml

Add below to the file,

xpack.encryptedSavedObjects.encryptionKey:"gxraLwPeOGXtoZsVtJcZCLz31O0221J4"

```
xpack.encryptedSavedObjects.encryptionKey: abrakwpeOGXtomsWtJcZCLz3100221J43c
```

Rules:

For Memory Usage

The screenshot shows the Kibana Rules configuration page for a new rule. The rule is titled "For Memory Usage".

- Use KQL or Lucene to detect issues across indices.** This option is selected (indicated by a green checkmark and the word "Selected").
- Access to ML requires a [Platinum subscription](#).** This option is unavailable (indicated by "Unavailable").
- Aggregate query results to detect when number of matches exceeds threshold.** This option is available (indicated by "Select").
- Event Correlation** (Use Event Query Language (EQL) to match events, generate sequences, and stack data). This option is available (indicated by "Select").
- Indicator Match** (Use indicators from intelligence sources to detect matching events and alerts). This option is available (indicated by "Select").

Index patterns [Reset to default index patterns](#)

metricbeat* x

Enter the pattern of Elasticsearch indices where you would like this rule to run. By default, these will include index patterns defined in Security Solution advanced settings.

Custom query [Import query from saved timeline](#)

system.memory.actual.used.bytes > 0.8 KQL

+ Add filter

Timeline template

None

Select which timeline to use when investigating generated alerts.

Quick query preview

Last hour

Select a timeframe of data to preview query results

[Preview results](#)

[Continue](#)

2 About rule

Name

Alert for memory usage

Description

Alert when memory usage is > 80%

Default severity

Select a severity level for all alerts generated by this rule.

Low

☐ Severity override

Use source event values to override the default severity.

Default risk score

Select a risk score for all alerts generated by this rule.

0 25 50 75 100 21

☐ Risk score override

Use a source event value to override the default risk score.

Tags

Optional

memory ×

Type one or more custom identifying tags for this rule. Press enter after each tag to begin a new one.

> Advanced settings

Continue

3 Schedule rule

Runs every

2 Minutes

Rules run periodically and detect alerts within the specified time frame.

Additional look-back time

Optional

1 Minutes

Adds time to the look-back period to prevent missed alerts.

Continue

Additional look-back time

Ru

Action

On

Select

Ac

Slack connector

Connector name

Untitled

Connector settings

Webhook URL

Remember this value. You must reenter it each time you edit the connector.

Create a Slack Webhook URL [↗](#)

Cancel Save

Ru

Action

On

Select

Ac

Slack connector

Connector name

Memory alert

Connector settings

Webhook URL

Remember this value. You must reenter it each time you edit the connector.

<https://hooks.slack.com/services/T02K9NY18JC/B02PG72LQ3U/zRSXbI>

Create a Slack Webhook URL [↗](#)

Cancel Save

4

Rule actions

Actions frequency

On each rule execution

Select when automated actions should be performed if a rule evaluates as true.

Actions

Memory alert

Slack connector

Add connector

Memory alert

Message

Rule {{context.rule.name}} generated {{state.signals_count}} alerts

Add action

Create rule without activating it

Create & activate rule

For Disk Usage

Index patterns

[Reset to default index patterns](#)

metricbeat*

Enter the pattern of Elasticsearch indices where you would like this rule to run. By default, these will include index patterns defined in Security Solution advanced settings.

Custom query

[Import query from saved timeline](#)

system.fsstat.total_size.used >0.7

KQL

+ Add filter

Timeline template

None

Select which timeline to use when investigating generated alerts.

Quick query preview

Last hour

Preview results

Select a timeframe of data to preview query results

2

About rule

Name

Alert for disk usage

Description

Alert when disk usage is > 0.7

Default severity

Select a severity level for all alerts generated by this rule.

☒ Low

☐ Severity override

Use source event values to override the default severity.

Default risk score

Select a risk score for all alerts generated by this rule.



☐ Risk score override

Use a source event value to override the default risk score.

Tags

Optional

Type one or more custom identifying tags for this rule. Press enter after each tag to begin a new one.

3

Schedule rule

Runs every

02

Minutes



Rules run periodically and detect alerts within the specified time frame.

Additional look-back time

Optional

1

Minutes



Adds time to the look-back period to prevent missed alerts.

Continue

4

Rule actions

Actions frequency

On each rule execution

Select when automated actions should be performed if a rule evaluates as true.

Actions

✓  Disk Usage alert

Slack connector

[Add connector](#)

Disk Usage alert

Message



Rule {{context.rule.name}} generated {{state.signals_count}} alerts

Add action

For Load Average



Custom query

Use KQL or Lucene to detect issues across indices.

✓ Selected



Machine Learning

Access to ML requires a [Platinum subscription](#).

Unavailable



Threshold

Aggregate query results to detect when number of matches exceeds threshold.

Select



Event Correlation

Use Event Query Language (EQL) to match events, generate sequences, and stack data

Select



Indicator Match

Use indicators from intelligence sources to detect matching events and alerts.

Select

Index patterns

[Reset to default index patterns](#)

metricbeat* X

Enter the pattern of Elasticsearch indices where you would like this rule to run. By default, these will include index patterns defined in Security Solution advanced settings.

Custom query

[Import query from saved timeline](#)

system.load.5 > 1

KQL

+ Add filter

Timeline template

None

Select which timeline to use when investigating generated alerts.

Quick query preview

Last hour

Preview results

Select a timeframe of data to preview query results

Continue

2

About rule

Name

System load Alert

Description

Load alert when avg load > 1

Default severity

Select a severity level for all alerts generated by this rule.

☒ Low

☐ Severity override

Use source event values to override the default severity.

Default risk score

Select a risk score for all alerts generated by this rule.

0 25 50 75 100

21

☐ Risk score override

3

Schedule rule

Runs every

02



Minutes



Rules run periodically and detect alerts within the specified time frame.

Additional look-back time

Optional

1

Minutes



Adds time to the look-back period to prevent missed alerts.

Continue

4

Rule actions

Actions frequency

On each rule execution

Select when automated actions should be performed if a rule evaluates as true.

Actions

Load Usage alert

Slack connector

Add connector

Load Usage alert

Message

Rule {{context.rule.name}} generated {{state.signals_count}} alerts

Add action

Create rule without activating it

Create & activate rule

We can see all the rules created as,

Rules

Load Elastic prebuilt rules and timeline templates

Upload value lists

Import rules

Create new rule

Rules

Rule Monitoring

All rules

Updated 10 seconds ago

Search

e.g. rule name

Tags

1

Elastic rules (0)

Custom rules (3)

Showing 3 rules

Selected 0 rules

Bulk actions

Refresh

Refresh settings

Rule	Risk score	Severity	Last run	Last response	Last updated	Version	Tags	Activated
<input type="checkbox"/> Alert for memory usage	21	Low	18 hours ago	succeeded	Dec 4, 2021 @ 00:21:12.581	1	memory	<input checked="" type="checkbox"/>
<input type="checkbox"/> Alert for disk usage	21	Low	18 hours ago	succeeded	Dec 4, 2021 @ 00:34:33.101	1	—	<input checked="" type="checkbox"/>
<input type="checkbox"/> System load Alert	21	Low	18 hours ago	succeeded	Dec 4, 2021 @ 00:37:21.292	1	—	<input checked="" type="checkbox"/>

Rows per page: 20

<

1

>

We can see the alerts generated in slack also as,



incoming-webhook APP 12:38 AM

Rule system load alert generated 12 alerts



incoming-webhook APP 12:44 AM

Rule system load alert generated 13 alerts



incoming-webhook APP 8:14 AM

Rule disk usage alert(>70%) generated 12 alerts

Rule memory alert generated 11 alerts

#