ELK Installation on Centos 7

sudo yum install vim-enhanced -y
===

sudo yum install java-1.8.0-openjdk
(https://www.digitalocean.com/community/tutorials/how-to-install-java-on-centos-and-fedora#install-openjdk-8)
===

Nginx installation
https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-centos-7
===

Refer this link for ELK installation
https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-centos-7

Step 1 — **Installing and Configuring Elasticsearch**

The Elastic Stack components are not available through the package manager by default, but you can install then

yum by adding Elastic's package repository.

All of the Elastic Stack's packages are signed with the Elasticsearch signing key in order to protect your system

package spoofing. Packages which have been authenticated using the key will be considered trusted by your pac

manager. In this step, you will import the Elasticsearch public GPG key and add the Elastic repository in order to

Elasticsearch.

Run the following command to download and install the Elasticsearch public signing key:

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

Next, add the Elastic repository. Use your preferred text editor to create the file elasticsearch.repo in the /etc/yum.repos.d/ directory. Here, we'll use the *vi* text editor:

- **sudo vi /etc/yum.repos.d/elasticsearch.repo**

To provide yum with the information it needs to download and install the components of the Elastic Stack, enter
mode by pressing i and add the following lines to the file.

```
/etc/yum.repos.d/elasticsearch.repo
[elasticsearch-6.x]
name=Elasticsearch 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
type=rpm-md
```

Here you have included the human-readable name of the repo, the baseurl of the repo's data directory, and the gp
required to verify Elastic packages.

When you're finished, press ESC to leave insert mode, then :wq and ENTER to save and exit the file. To learn m
about the text editor vi and its successor *vim*, check out our Installing and Using the Vim Text Editor on a Cloud
tutorial.

With the repo added, you can now install the Elastic Stack. According to the official documentation, you should
Elasticsearch before the other components. Installing in this order ensures that the components each product dep
are correctly in place.

Install Elasticsearch with the following command:

- **sudo yum install elasticsearch**

Once Elasticsearch is finished installing, open its main configuration file, elasticsearch.yml, in your editor:

```
sudo vi /etc/elasticsearch/elasticsearch.yml
```

Note: Elasticsearch’s configuration file is in YAML format, which means that indentation is very important! Be

that you do not add any extra spaces as you edit this file.

Elasticsearch listens for traffic from everywhere on port 9200. You will want to restrict outside access to your Elasticsearch instance to prevent outsiders from reading your data or shutting down your Elasticsearch cluster th the REST API. Find the line that specifies network.host, uncomment it, and replace its value with localhost so it like this:

/etc/elasticsearch/elasticsearch.yml

```
. . .
network.host: localhost
. . .
```

Save and close elasticsearch.yml. Then, start the Elasticsearch service with systemctl:

- **sudo systemctl start elasticsearch**

Next, run the following command to enable Elasticsearch to start up every time your server boots:

- **sudo systemctl enable elasticsearch**

```
thread_pool:
bulk:
queue_size: 200
```

You can test whether your Elasticsearch service is running by sending an HTTP request:

http://127.0.0.1:9200/ or

curl -X GET "localhost:9200"

You will see a response showing some basic information about your local node, similar to this:

Output

```
{
  "name" : "8oSCBFJ",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "1Nf9ZymBQaOWKpMRBfisog",
  "version" : {
    "number" : "6.5.2",
    "build_flavor" : "default",
    "build_type" : "rpm",
    "build_hash" : "9434bed",
    "build_date" : "2018-11-29T23:58:20.891072Z",
    "build_snapshot" : false,
    "lucene_version" : "7.5.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Now that Elasticsearch is up and running, let’s install Kibana, the next component of the Elastic Stack.

Step 2 — Installing and Configuring the Kibana Dashboard

According to the installation order in the official documentation, you should install Kibana as the next componen

Elasticsearch. After setting Kibana up, we will be able to use its interface to search through and visualize the dat

Elasticsearch stores.

Because you already added the Elastic repository in the previous step, you can just install the remaining compon

the Elastic Stack using yum:

```
sudo yum install kibana
```

Then enable and start the Kibana service:

```
sudo systemctl enable kibana
sudo systemctl start kibana
```

-

  Note:
  (instead of localhost use 0.0.0.0 ip ES and kibana
- **Goto /etc/elasticsearch/elasticsearch.yml. Look for value in network.host and change it to 0.0.0.0**

- **This is step if you are using Kibana. Goto /etc/kibana/kibana.yml Look for value in server.host and change it to 0.0.0.0)**

Because Kibana is configured to only listen on localhost, we must set up a reverse proxy to allow external access

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'll use to access the Kibana we

interface. As an example, we will name this account kibanaadmin, but to ensure greater security we recommend

choose a non-standard name for your user that would be difficult to guess.

The following command will create the administrative Kibana user and password, and store them in the htpassw

file. You will configure Nginx to require this username and password and read this file momentarily:

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

-

Enter and confirm a password at the prompt. Remember or take note of this login, as you will need it to access th

Kibana web interface.

=============== only if you want to access via domain name start here =====================

Next, we will create an Nginx server block file. As an example, we will refer to this file as example.com.conf, al

you may find it helpful to give yours a more descriptive name. For instance, if you have a FQDN and DNS recor

up for this server, you could name this file after your FQDN:

sudo vi /etc/nginx/conf.d/example.com.conf

●

Add the following code block into the file, being sure to update example.com and www.example.com to match y

server's FQDN or public IP address. This code configures Nginx to direct your server's HTTP traffic to the Kiba

application, which is listening on localhost:5601. Additionally, it configures Nginx to read the htpasswd.users fil

require basic authentication.

Note that if you followed the prerequisite Nginx tutorial through to the end, you may have already created this fi

populated it with some content. In that case, delete all the existing content in the file before adding the following

example.com.conf'>/etc/nginx/conf.d/example.com.conf

```
server {
    listen 80;

    server_name example.com www.example.com;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.users;

    location / {
        proxy_pass http://localhost: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're finished, save and close the file.

Then check the configuration for syntax errors:

- **sudo nginx -t**

If any errors are reported in your output, go back and double check that the content you placed in your configura

was added correctly. Once you see syntax is ok in the output, go ahead and restart the Nginx service:

- **sudo systemctl restart nginx**

By default, SELinux security policy is set to be enforced. Run the following command to allow Nginx to access

proxied service:

=============== only if you want to access via domain name Ends here =======================

- **sudo setsebool httpd_can_network_connect 1 -P**

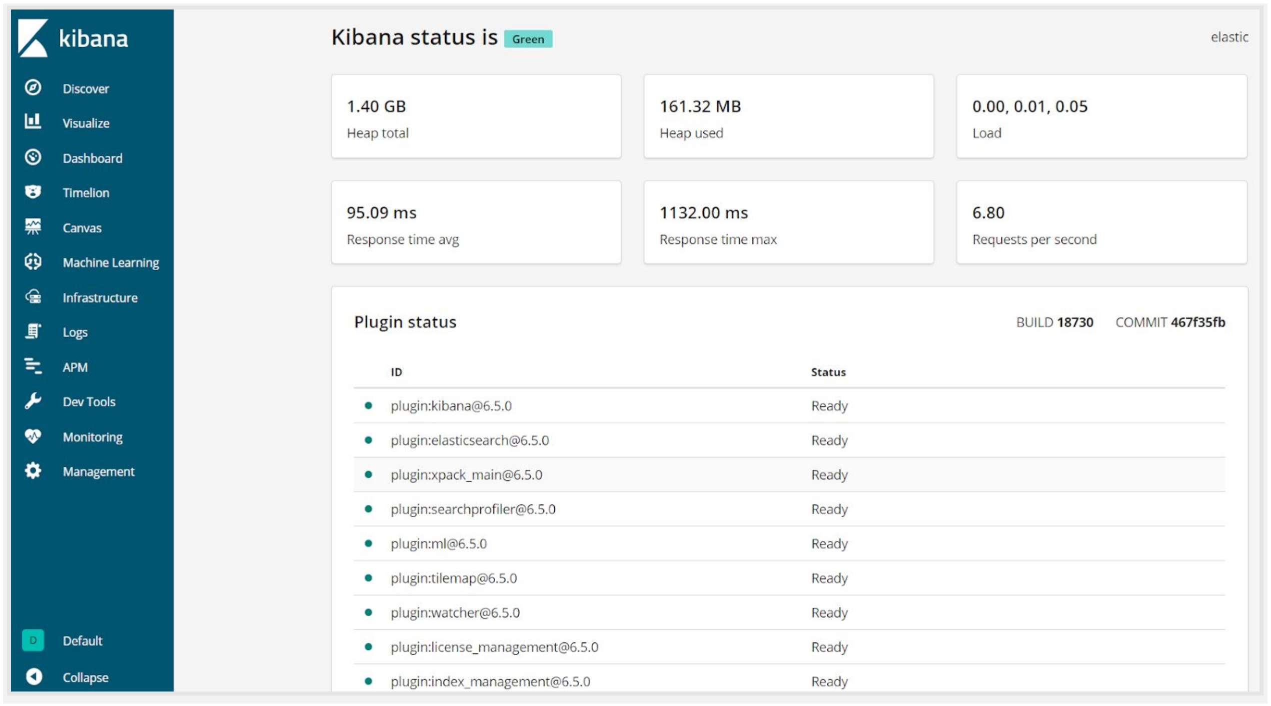
You can learn more about SELinux in the tutorial An Introduction to SELinux on CentOS 7.

Kibana is now accessible via your FQDN or the public IP address of your Elastic Stack server. You can check th

Kibana server's status page by navigating to the following address and entering your login credentials when pro

http://localhost:5601/status

This status page displays information about the server's resource usage and lists the installed plugins.



Kibana status is Green

1.40 GB	Heap total
161.32 MB	Heap used
0.00, 0.01, 0.05	Load
95.09 ms	Response time avg
1132.00 ms	Response time max
6.80	Requests per second

Plugin status — BUILD 18730 COMMIT 467f35fb

ID	Status
plugin:kibana@6.5.0	Ready
plugin:elasticsearch@6.5.0	Ready
plugin:xpack_main@6.5.0	Ready
plugin:searchprofiler@6.5.0	Ready
plugin:ml@6.5.0	Ready
plugin:tilemap@6.5.0	Ready
plugin:watcher@6.5.0	Ready
plugin:license_management@6.5.0	Ready
plugin:index_management@6.5.0	Ready

Note: As mentioned in the Prerequisites section, it is recommended that you enable SSL/TLS on your server. Yo follow this tutorial now to obtain a free SSL certificate for Nginx on CentOS 7. After obtaining your SSL/TLS certificates, you can come back and complete this tutorial.

Now that the Kibana dashboard is configured, let's install the next component: Logstash.

Step 3 — Installing and Configuring Logstash

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

Install Logstash with this command:

- **sudo yum install logstash**

For inserting mongo data into Elastic index using logstash (this plugin didn't work for Us)

==

https://bqstack.com/b/detail/97/Configure-Logstash-to-send-MongoDB-data-into-Elasticsearch

sudo su
cd /usr/share/logstash
bin/logstash-plugin install logstash-input-mongodb

We will be getting the following response:

Validating logstash-input-mongodb

Installing logstash-input-mongodb

Installation successful

It will need single config file for each collection to be inserted into elastic

Create file in /etc/logstash/conf.d/mongodata.conf add following config

==

```
input {
mongodb {
uri => 'mongodb://ntcarfte:bYzgzp9cEH@172.16.16.63:27027/ntcarfte' // mongo connection
placeholder_db_dir => '/opt/logstash-mongodb/'
collection => 'ntc.callpath_utilizationreport_data' // collection name
batch_size => 500
}
}
filter {
mutate {
rename => { "_id" => "mongo_id" }  // _id PK of mongo collection
}
}
output {
elasticsearch {
hosts => "localhost:9200"
index => "indexname"
doc_as_upsert => true
document_id => "%{mongo_id}"
}
}
```

==
Create directory in /opt/logstash-mongodb/ and give 777 permission (delete existed .db file in that incase if you get any issue in while inserting records)

ways to send data to elastic index
- running logstash as service

- Systemctl start logstash or
- Run config file directly to test data

  /usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/***mongodata.conf***

==

Notes
/etc/logstash/logstash.yml to debug log
-**log.level:** debug
==============================

validate all config

```
/usr/share/logstash/bin/logstash  --path.settings /etc/logstash  -f /etc/logstash/conf.d/ -t
```

- open firewall ports 9200,5601

```
firewall-cmd --zone=public --add-port=9200/tcp --permanent

firewall-cmd --zone=public --add-port=5601/tcp --permanent

firewall-cmd --reload

iptables-save | grep 9200
```

List Elastic Index

```
GET
127.0.0.1:9200/_aliases/?pretty=true
```

View Index data

get 127.0.0.1:9200/indexname/doc/_search?pretty

Create Index record

```
curl -X POST '127.0.0.1:9200/ntcarfte/doc/' -H 'Content-Type: application/json' -d'
```

 or

127.0.0.1:9200/indexname/doc/ in postman

```
{
  "name_city_orig" : "LOSANGELES",
  "future_tf_features" : null,
  "date_msg" : "190711",
  "record_id" : "010101",
  "date" : "190712",
  "@timestamp" : "2019-12-24T23:44:53.000Z",
}'
```

Kibana

create stock management->index pattern in Kibana then create chart using this index

============================ HOLD ====================

- dynamic mapping of fields to accommodate input possible values which is skipping by logstsah due to different datatypes

https://stackoverflow.com/questions/63883823/elastic-dynamic-field-mapping/63883910?noredirect=1#comment113218646_63883910

https://blog.ruanbekker.com/blog/2019/04/06/elasticsearch-templates-tutorial/

sudo vi /etc/elasticsearch/elasticsearch.yml

```
indices.memory.index_buffer_size: 60%
```

==

Using **monstache** instead of logstash

https://rwynn.github.io/monstache-site

==

=================================

Install Go Lang

=================================

https://www.digitalocean.com/community/tutorials/how-to-install-go-1-7-on-centos-7

https://medium.com/@nehajirafe/mongodb-to-elasticsearch-sync-using-monstache-cfe1177594b6

go version

Step 2 : Install Monstache

- cd /usr/local/bin
- sudo wget https://github.com/rwynn/monstache/releases/download/v4.15.0/monstache-eee3f26.zip
- sudo unzip monstache-eee3f26.zip
- export PATH=`/usr/local/bin/build/linux-amd64`:$PATH
- `cd /usr/local/bin/build/linux-amd64`
- monstache -v

- ====================================

```
cd /usr/local/bin/build/linux-amd64
```

```
sudo vim mongo-elastic.toml
```

```
==========================================================
mongo-url
="mongodb://admin:bYzgzp9cEH@66.160.227.205:27027/ntcarfte?authSource=
admin"
elasticsearch-url ="http://localhost:9200"
elasticsearch-max-conns = 10
resume = true
resume-name = "default"
#namespace-regex = '[ntcarfte.ntc.cdr]'
#direct-read-namespaces =
["ntcarfte.ntc.callpath_utilizationreport_data"]
direct-read-namespaces =
["ntcarfte.ntc.cdr","ntcarfte.ntc.callpath_utilizationreport_data"]
direct-read-stateful = true
#exit-after-direct-reads = true
disable-change-events = true
gzip = true
fail-fast = false
index-as-update = true
elasticsearch-retry = true
prune-invalid-json = true
index-oplog-time = true
verbose = true

[mongo-dial-settings]
timeout=15
read-timeout=7
write-timeout=7

[mongo-session-settings]
socket-timeout=30
sync-timeout=30

elasticsearch-max-bytes = 800000
elasticsearch-max-conns = 2

[gtm-settings]
```

```
channel-size = 50000

direct-read-concur = 1
direct-read-split-max = -1
direct-read-no-timeout=true

[logs]
info = "/var/log/monstache/info.log"
warn = "/var/log/monstache/warn.log"
error = "/var/log/monstache/error.log"
trace = "/var/log/monstache/trace.log"
stats = "/var/log/monstache/stats.log"

[[mapping]]
namespace = "ntcarfte.ntc.cdr"
index = "cdr_data"
type = "_doc"

[[mapping]]
namespace = "ntcarfte.ntc.callpath_utilizationreport_data"
index = "callpath_data"
type = "_doc"
```

============================

```
# Repeat mapping block it for each collection you want to
replicate
```

==

Test with = > monstache -f mongo-elastic.toml &

For run it as SYSTEMD service

```
Create

==============================
```

```
[Unit]
Description=monstache sync service

[Service]
Type=notify
```

```
ExecStart=/usr/local/bin/build/linux-amd64/monstache -f
/usr/local/bin/build/linux-amd64/mongo-elastic.toml
WatchdogSec=30s
Restart=always

[Install]
WantedBy=multi-user.target
==========================
```

```
systemctl daemon-reload

systemctl enable monstache.service

systemctl start monstache.service

systemctl stop monstache.service

systemctl status monstache.service

#systemctl restart monstache.service
```

```
-==================final config ===================
```

```
#mongo-url="mongodb://admin:bYzgzp9cEH@66.160.227.205:27027/ntcarfte?authSource=admin"
mongo-url="mongodb://admin:wRASdYscrbj5sXSt@66.160.227.11:27018/ntcarfte?authSource=admin"
elasticsearch-url ="http://localhost:9200"
elasticsearch-max-conns = 10
resume = true
resume-name = "default"
enable-oplog = true
namespace-regex = "^ntacrfte.ntc.(cdr|callpath_utilizationreport_data)$"
direct-read-namespaces = ["ntcarfte.ntc.cdr","ntcarfte.ntc.callpath_utilizationreport_data"]
exit-after-direct-reads = true
disable-change-events = false
gzip = true
#fail-fast = false
index-as-update = false
elasticsearch-retry = true
```

```
prune-invalid-json = true
index-oplog-time = true
verbose = true
index-stats =true
stats =true
elasticsearch-max-bytes = 800000

[gtm-settings]
channel-size = 1024
direct-read-concur = 1
direct-read-split-max = -1
#direct-read-no-timeout=true

[mongo-dial-settings]
timeout=300
read-timeout=300
write-timeout=100

[mongo-session-settings]
socket-timeout=600
sync-timeout=61

[logs]
#info = "/var/log/monstache/info.log"
warn = "/var/log/monstache/warn.log"
error = "/var/log/monstache/error.log"
trace = "/var/log/monstache/trace.log"
stats = "/var/log/monstache/stats.log"

[[filter]]
#namespace = "ntcarfte.ntc.callpath_utilizationreport_data"
script = """
module.exports = function(doc,ns) {

        var todayZero = Math.round(new Date().setHours(0,0,0,0) /1000);

        if(ns =='ntcarfte.ntc.cdr'){
             return ( ( parseInt(doc.start_epoch))  > (todayZero - 86400) );
         }

        if(ns =='ntcarfte.ntc.callpath_utilizationreport_data'){
          return (doc.inserted_at > (todayZero - 86400 ) );
        }
   }
```

```
"""
[[mapping]]
namespace = "ntcarfte.ntc.cdr"
index = "cdr_data_4"
type = "_doc"

[[mapping]]
namespace = "ntcarfte.ntc.callpath_utilizationreport_data"
index = "callpath_data_4"
type = "_doc"
```

```
###### milan created this below #######################
export GOBIN="$HOME/projects/bin"
export GOPATH="$HOME/projects/src"
==============================================
```

Incase needed for any issue

```
# source /etc/profile && source ~/.bash_profile
```

Reference

https://github.com/rwynn/monstache-showcase/blob/master/monstache.config.toml

==