ELK Stack Configuration

ELK STACK CONFIGURATION IN A LINUX ENVIRONMENT

VIRAJ ARIYASINGHE

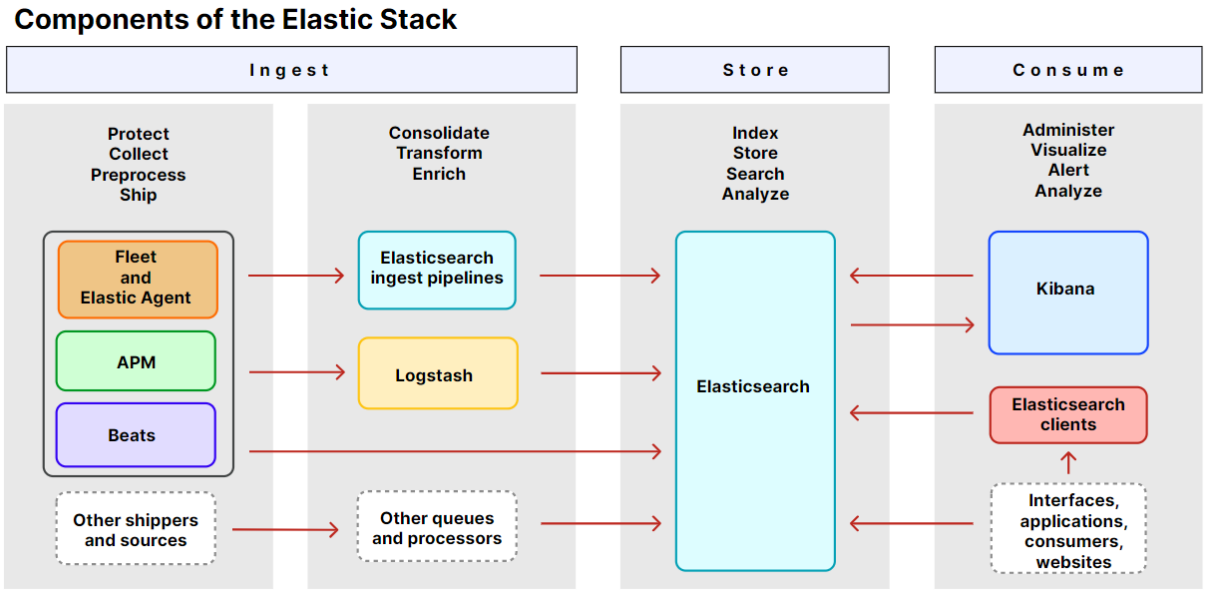
CONTENT

This document will guide you on how to implement an ELK Stack(Elasticsearch, Logstash & Kibana). After the implement ELK stack, we can access all the logs generated by servers via the Kibana dashboard.

1.Installation instructions

1.1.Requirements

- Kibana – This is the Dashboad
- ElasticSearch – This is the software all the logs are stored according to some index and kibana reading the logs from Elasticsearch.
- Logstash - is a lightweight, open-source, server-side data processing pipeline that allows you to collect data from a variety of sources, transform it on the fly, and send it to your desired destination.



1.2.To implement ELK stack, we need two servers

- Master – In this server we are going to install all the major software's (ElasticSerac,Kibana)
- Client – In this server we are going to installing agent software only (Logstash),in our environment this should be Test,QA,Demo,Prod

2.Install Elasticsearch on the master server using Binary file

2.1.Dependencies

I. JDK 1.8.0 or higher (Jdk is already included in the binary installation on elastic search)
II. Minimum system requirements for normal operation of Elasticsearch are 8Gb RAM
III. 4 Gb Should be allocated to java Heap

2.2.Install Procedure using Binary File

I. Download the Binary file of Elastic search using official website for linux x86_64

- wget `https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.11.1-linux-x86_64.tar.gz`

II. Extract the tar.gz to the required location

- tar -xvfz /path/to/download/elasticsearch/elasticsearch-8.8.2-linux-x86_64.tar.gz -C /path/to/install/

III. Create a system user and group (elasticsearch) to run the elasticsearch service

- sudo useradd --system --shell /sbin/nologin elasticsearch

IV. Give the permissions to elastic search install directory, log directory, data directory for elasticsearch user

- sudo chown -R elasticsearch:elasticsearch /path/to/directory

V. Create a service file

- sudo vim /etc/systemd/system/elasticsearch.service

```
[Unit]
Description=Elasticsearch
Documentation=https://www.elastic.co
Wants=network-online.target
After=network-online.target

[Service]
Type=simple
User=elasticsearch
Group=elasticsearch
ExecStart=/path/to/install /bin/elasticsearch -p /var/run/elasticsearch/elasticsearch.pid
WorkingDirectory=/path/to/installdir
LimitMEMLOCK=infinity
LimitNOFILE=65536
TimeoutStopSec=20
Restart=on-failure
RestartSec=5

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

You have to create /var/run/elasticsearch this directory and give the ownership of the directory to the elastic search user to create the .pid file in it

2.3 Edit the elastic search configuration file in the /path/to/installdir/config (elasticsearch.yml)

I. Edit paths for log files and data files

```
#
# ---------------------------------- Paths ------------------------------------
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /data/elk-log-index
#
# Path to log files:
#
path.logs: /logs/elk-logs
#
```

II. Edit the memory lock status to true

```
# ---------------------------------- Memory -----------------------------------
#
# Lock the memory on startup:
#
bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network -----------------------------------
```

III. Set the IP address and the port of the server which runs elastic search

```
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 192.168.1.100
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# --------------------------------- Discovery ----------------------------------
```

The network. Host setting in Elasticsearch defines the host or IP address on which the Elasticsearch node will bind to for communication with other nodes in the cluster. It essentially specifies the network interface on which Elasticsearch will listen for connections.

Port 9200 is used for HTTP communication, and Elasticsearch is configured to listen on all available network interfaces.

Port 9300 is used for inter-node communication (transport protocol), and Elasticsearch is configured to listen on the specific IPv4 address 10.11.22.62.

IV. Disable the x pack security features for ssl protocol (server will be communicated through the http)

```
# ------------------------------------------------------------------------

# Enable security features
xpack.security.enabled: false

xpack.security.enrollment.enabled: false

# Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents
xpack.security.http.ssl:
  enabled: false
  keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: false
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
# Create a new cluster with the current node only
# Additional nodes can still join the cluster later
cluster.initial_master_nodes: ["localhost.localdomain"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
http.host: 0.0.0.0

# Allow other nodes to join the cluster from anywhere
# Connections are encrypted and mutually authenticated
#transport.host: 0.0.0.0

#----------------------- END SECURITY AUTO CONFIGURATION -------------------------
```

The http.host setting specifically applies to the HTTP (RESTful) interface of Elasticsearch. It determines the network interface on which Elasticsearch will listen for incoming HTTP requests.

2.4. Set max file descriptors for elasticsearch

I. Create a elasticsearch.conf file for max file descriptors in /etc/security/limits.d/ directory

- *sudo vim /etc/security/limits.d/elasticsearch.conf*

II. Add the following configuration to the conf file

- *<elasticsearch user> soft nofile 65535*
- *<Elastic user > hard nofile 65535*
- Save and exit

2.5. Set the Max virtual memory Areas configuration

I. Create a file elasticsearch.conf in the /etc/sysctl.d/ directory.

- *sudo vim /etc/sysctl.d/99-elasticsearch.conf*

II. Add the following configuration to the file

- vm.max_map_count=262144
- Save and exit

III. Apply changes to sysctl.conf file

- sudo sysctl -p

2.6 Give port access in firewall

I. sudo firewall-cmd –zone=public –add-port=9200/tcp –permanent
II. Sudo firewall-cmd –reload

2.7 Launch the service

I. Start the elasticsearch service at startup

- Sudo systemctl enable elasticsearch

II. Start the elastic search service

- Sudo systemctl start elasticsearch

You can access the Elastic search API using http:<server-ip> :9200

3. Install Kibana on the master server using Binary file

3.1. Dependencies

I. JDK 1.8.0 or higher (Jdk is already included in the binary installation on elastic search)

3.2.Install Kibana

I. Download the Binary file of Kibana using official website for linux x86_64

- wget https://artifacts.elastic.co/downloads/kibana/kibana-8.8.2-linux-x86_64.tar.gz

II. Extract the tar.gz to the required location

- tar -xvfz /path/to/download/kibana-8.8.2-linux-x86_64.tar.gz -C /path/to/install/

III. Create a system user and group (kibana) to run the kibana service

- sudo useradd --system --shell /sbin/nologin kibana

IV. Give the permissions to kibana install directory, log directory, data directory for kibana user

- sudo chown -R kibana:kibana /path/to/directory

V. Create a service file

- sudo vim /etc/systemd/system/kibana.service

```
[Unit]
Description=Kibana-8.8.1
Documentation=https://www.elastic.co/guide/en/kibana/index.html
Wants=network-online.target
After=network-online.target

[Service]
User=kibana
Group=kibana
Environment=NODE_OPTIONS="--max-old-space-size=4096"
ExecStart=/path/to/install-kibana/bin/kibana

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

3.3.Edit the kibana configuration file in the /path/to/installdir/config (kibana.yml)

I. Set the server port and the host IP address where kibana service located

```
# --==================== System: Kibana Server ====================
# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 9000

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
#server.host: "localhost"
server.host: 10.100.200.10

# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""
```

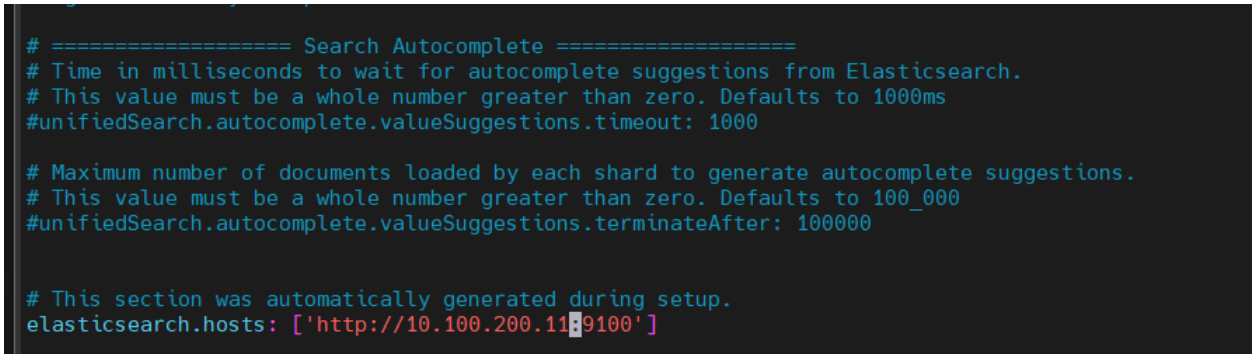
II. Set the authentication to communicate with elastic server

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

# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
elasticsearch.username: "kibana_system"
elasticsearch.password: "******"

# Kibana can also authenticate to Elasticsearch via "service account tokens".
# Service account tokens are Bearer style tokens that replace the traditional username/password based configuration.
# Use this token instead of a username/password.
# elasticsearch.serviceAccountToken: "my_token"

# Time in milliseconds to wait for Elasticsearch to respond to pings. Defaults to the value of
# the elasticsearch.requestTimeout setting.
#elasticsearch.pingTimeout: 1500
```

III. Set the log location where kibana stores the logs

```
# ==================== System: Logging ====================
# Set the value of this setting to off to suppress all logging output, or to debug to log everything. Defaults to 'info'
logging.root.level: info

# Enables you to specify a file where Kibana stores log output.
logging.appenders.default:
  type: file
  fileName: /logs/elk-logs/kibana/kibana.log
  layout:
    type: json

# Logs queries sent to Elasticsearch.
#logging.loggers:
#  - name: elasticsearch.query
#    level: debug

# Logs http responses.
#logging.loggers:
#  - name: http.server.response
#    level: debug

# Logs system usage information.
#logging.loggers:
#  - name: metrics.ops
#    level: debug
```

IV. Set the elastic servers Ip and Port which kibana can communicate



We have to set elasticsearch.host as http not as https

3.4. Give firewall access to port using kibana

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

3.5 Launch Kibana

I. Start kibana service when system bootup

- Sudo systemctl enable kibana

II. Start the kibana service

- Sudo systemctl start kibana

III. You can log into kibana using http://<server-ip>:5601

4.Install Logstash on the Remote server using Binary file

4.1. Dependencies

I. JDK 1.8.0 or higher (Jdk is already included in the binary installation on elastic search)

4.2 Install Logstash

I. Download the Binary file of logstash using official website for linux x86_64

wget https://artifacts.elastic.co/downloads/logstash/logstash-8.11.2-linux-x86_64.tar.gz

II. Extract the tar.gz to the required location

tar -xvfz /path/to/download/ logstash-8.*.*-linux-x86_64.tar.gz -C /path/to/install/

III. Create a system user and group (logstash) to run the logstash service

sudo useradd --system --shell /sbin/nologin logstash

IV. Give the permissions to logstash install directory, log directory, data directory for logstash user (log directory and data directory for logstash is located at the install directory you can customize the location in logstash .yml file)

sudo chown -R logstash:logstash /path/to/directory

V. Create a service file

sudo vim /etc/systemd/system/logstash.service

```
[Unit]
Description=Logstash
Documentation=https://www.elastic.co/guide/en/logstash/index.html
Wants=network-online.target
After=network-online.target

[Service]
#user=logstash
#Group=logstash
Environment=LS_HOME=/path/to/installs/logstash
Environment=LS_SETTINGS_DIR=/path/to/installs/logstash/config
Environment=JAVA_OPTS="-Xmx1g -Xms1g"
```

```
ExecStart=/path/to/installs/logstash /bin/logstash
Restart=always
WorkingDirectory=/path/to/installs/logstash
LimitNOFILE=65536
TimeoutStopSec=30
LimitMEMLOCK=infinity

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

4.3 Configure Logstash.yml

I. Set the log directory for logstash

```
#
# ------------ Debugging Settings --------------
#
# Options for log.level:
#   * fatal
#   * error
#   * warn
#   * info (default)
#   * debug
#   * trace
#
log.level: info
path.logs: /logs/logstash
#
# ------------ Other Settings --------------
```

II. Disable x-pack security configurations(if we use https we have to enable certificate verification settings as well)

```
#
# X-Pack Management
# https://www.elastic.co/guide/en/logstash/current/logstash-centralized-pipeline-management.html
#xpack.management.enabled: false
#xpack.management.pipeline.id: ["main", "apache_logs"]
#xpack.management.elasticsearch.username: logstash_admin_user
#xpack.management.elasticsearch.password: password
#xpack.management.elasticsearch.proxy: ["http://proxy:port"]
#xpack.management.elasticsearch.hosts: ["https://es1:9200", "https://es2:9200"]
# an alternative to hosts + username/password settings is to use cloud_id/cloud_auth
#xpack.management.elasticsearch.cloud_id: management_cluster_id:xxxxxxxxxx
#xpack.management.elasticsearch.cloud_auth: logstash_admin_user:password
# another authentication alternative is to use an Elasticsearch API key
#xpack.management.elasticsearch.api_key: "id:api_key"
#xpack.management.elasticsearch.ssl.ca_trusted_fingerprint: xxxxxxxxxx
#xpack.management.elasticsearch.ssl.certificate_authority: "/path/to/ca.crt"
#xpack.management.elasticsearch.ssl.truststore.path: /path/to/file
#xpack.management.elasticsearch.ssl.truststore.password: password
#xpack.management.elasticsearch.ssl.keystore.path: /path/to/file
#xpack.management.elasticsearch.ssl.keystore.password: password
#xpack.management.elasticsearch.ssl.verification_mode: certificate
#xpack.management.elasticsearch.sniffing: false
#xpack.management.logstash.poll_interval: 5s
```

4.4 Configure pipelines for send logs to the elastic search cluster

- Create a conf.d directory in the config directory in logstash
- Create a conf file for pipeline in the conf.d directory
- Sudo vim /path/to/config/conf.d/<name.conf>

```
input {
  file {
    path => "/logs/logstash/logstash-plain.log"  # Path to Logstash log file
    start_position => "beginning"
    sincedb_path => "/dev/null"
    codec => multiline {
      pattern => "^\[%{TIMESTAMP_ISO8601}"
      negate => true
      what => "previous"
    }
  }
}

filter {
  # Add any additional filters you need for Logstash logs
}

output {
  elasticsearch {
    hosts => ["http://192.168.1.100:9200"]  # Elasticsearch server IP and port
    index => "logstash_logs_web2"
```

```
  }
  stdout {
    codec => rubydebug
  }
}
```

4.5 Configure pipelines.yml file

I. We have to configure pipeline.yml file if we are sending logs of different applications

```
#
# Example of two pipelines:
#
# - pipeline.id: test
#   pipeline.workers: 1
#   pipeline.batch.size: 1
#   config.string: "input { generator {} } filter { sleep { time => 1 } } output { stdout { codec => dots } }"
# - pipeline.id: another_test
#   queue.type: persisted
#   path.config: "/tmp/logstash/*.config"
#
# Available options:
#
```

4.6 Launch Logstash

I. Start the Logstash service at startup
sudo systemctl enable logstash

II. Start the elastic search service
sudo systemctl start logstash

III. You can check the creation of Logstash indexes using the Kibana dash board dev tools using a http request to the elastic search server.

Get /_cat/indices

Install Filebeat on the Remote server using Binary file

5.1. Dependencies

I. JDK 1.8.0 or higher (Jdk is already included in the binary installation on elastic search)

5.2 Install Filebeat

I. Download the Binary file of logstash using official website for linux x86_64

wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.11.1-linux-x86_64.tar.gz

II. Extract the tar.gz to the required location

tar -xvzf /path/to/download/ filebeat-8.*.*-linux-x86_64.tar.gz -C /path/to/install/

III. Create a service file

sudo vim /etc/systemd/system/filebeat.service

```
[Unit]
Description=Filebeat sends log files to Logstash or directly to Elasticsearch.

[Service]
ExecStart=/installs/filebeat-8.11.1-linux-x86_64/filebeat -c installs/filebeat-8.11.1-linux-
      x86_64/filebeat.yml
User=filebeat
Group=filebeat

Restart=always

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

5.3 Configure filebeat.yml

- Edit the input configurations

```
#========================== Filebeat inputs =============================

# List of inputs to fetch data.
filebeat.inputs:
# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# Type of the files. Based on this the way the file is read is decided.
# The different types cannot be mixed in one input
#
# Possible options are:
# * filestream: Reads every line of the log file
# * log: Reads every line of the log file (deprecated)
# * stdin: Reads the standard in
```

```
#------------------------- Filestream input -----------------------------


#- type: filestream

  # Unique ID among all inputs, an ID is required.
# id: my-filestream-id

  # Change to true to enable this input configuration.
  enabled: false

  # Paths that should be crawled and fetched. Glob based paths.
  # To fetch all ".log" files from a specific level of subdirectories
  # /var/log/*/*.log can be used.
  # For each file found under this path, a harvester is started.
  # Make sure not file is defined twice as this can lead to unexpected behaviour.
  paths:
    - /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*
```

```
parsers:
  - multiline:
    type: pattern
    # The regexp Pattern that has to be matched. The example pattern matches all lines starting with [
    pattern: ^\[

    # Defines if the pattern set under the pattern setting should be negated or not. Default is false.
    negate: true

    # Match can be set to "after" or "before". It is used to define if lines should be appended to a pattern
    # that was (not) matched before or after or as long as a pattern is not matched based on negate.
    # Note: After is the equivalent to previous and before is the equivalent to next in Logstash
    match: after

    # The maximum number of lines that are combined into one event.
    # In case there are more than max_lines the additional lines are discarded.
    # Default is 500
    max_lines: 1500

    # After the defined timeout, a multiline event is sent even if no new pattern was found to start a new event
    # Default is 5s.
    #timeout: 5s

    # Do not add new line character when concatenating lines.
    #skip_newline: false

# To aggregate constant number of lines into a single event use the count mode of multiline.
```

```
# To aggregate constant number of lines into a single event use the count mode of multiline.

parsers:
  - multiline:
    type: count

    # The number of lines to aggregate into a single event.
    count_lines: 100

    # The maximum number of lines that are combined into one event.
    # In case there are more than max_lines the additional lines are discarded.
    # Default is 500
    max_lines: 1500

    # After the defined timeout, a multiline event is sent even if no new pattern was found to start a new event
    # Default is 5s.
    #timeout: 5s

    # Do not add new line characters when concatenating lines.
    #skip_newline: false
```

Output configurations

```
#-------------------------- Elasticsearch Output --------------------------

output.elasticsearch:
  hosts: ["10.100.200.11:9100"]
  protocol: "http"
  index: "logs-%{[fields.env]}-%{+yyyy.MM.dd}"
  username: "my_beats"
  password: "****"
```

Index template configuration

```
# overload your Elasticsearch with too many update requests.
#setup.template.overwrite: false

# Elasticsearch template settings
setup.template.settings:

  # A dictionary of settings to place into the settings.index dictionary
  # of the Elasticsearch template. For more details, please check
  # https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping.html
  #index:
    #number_of_shards: 1
    #codec: best_compression

  # A dictionary of settings for the _source field. For more details, please check
  # https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-source-field.html
  #_source:
    #enabled: false


setup.template.enabled: true
setup.template.name: "logs"  # Adjust to match your desired template name
setup.template.pattern: "logs-*"  # Align with your custom index pattern
```

Index life cycle configuration

```
# ======================= Index Lifecycle Management (ILM) ======================

# Configure index lifecycle management (ILM) to manage the backing indices
# of your data streams.

# Enable ILM support. Valid values are true, or false.
setup.ilm.enabled: true

# Set the lifecycle policy name. The default policy name is
# 'beatname'.
#setup.ilm.policy_name: "mypolicy"

# The path to a JSON file that contains a lifecycle policy configuration. Used
# to load your own lifecycle policy.
#setup.ilm.policy_file:

# Disable the check for an existing lifecycle policy. The default is true. If
# you disable this check, set setup.ilm.overwrite: true so the lifecycle policy
# can be installed.
#setup.ilm.check_exists: true
```

Kibana Configurations

```
#============================== Kibana =====================================

setup.kibana:
  host: "http://10.100.200.10:9000"

#============================= Template Setup ================================
```

Logging Configurations

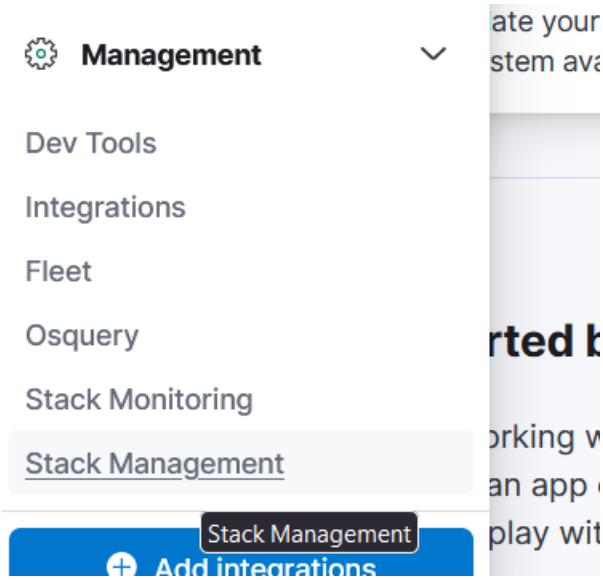
```
# ================================== Logging ===================================

# There are four options for the log output: file, stderr, syslog, eventlog
# The file output is the default.

# Sets log level. The default log level is info.
# Available log levels are: error, warning, info, debug
logging.level: info

# Enable debug output for selected components. To enable all selectors use ["*"]
# Other available selectors are "beat", "publisher", "service"
# Multiple selectors can be chained.
logging.selectors: ["*"]

# Send all logging output to stderr. The default is false.
#logging.to_stderr: false

# Send all logging output to syslog. The default is false.
#logging.to_syslog: false

# Send all logging output to Windows Event Logs. The default is false.
#logging.to_eventlog: false
```

```
#============================== Logging =====================================

logging.level: info
logging.selectors: ["*"]

logging.to_files: true
logging.files:
  path: /logs/filebeat
  rotateeverybytes: 20971520 # = 20MB
  keepfiles: 20

# Optional Keystore
#keystore.path: "${path.config}/beats.keystore"
```

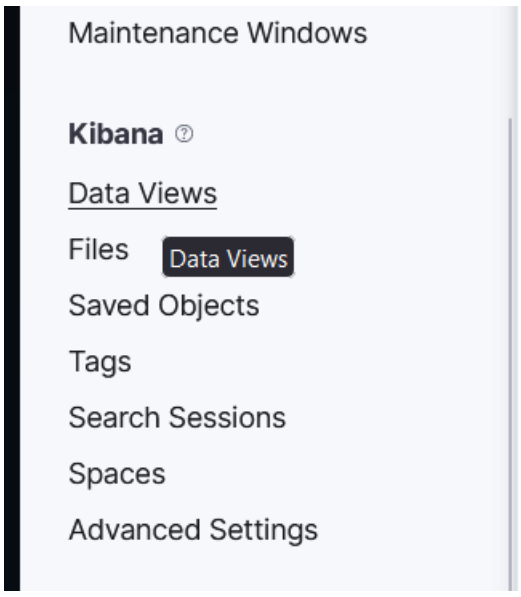
View Elastic search log indexes using Kibana dashboard

I. Goto menu tab in the Kibana

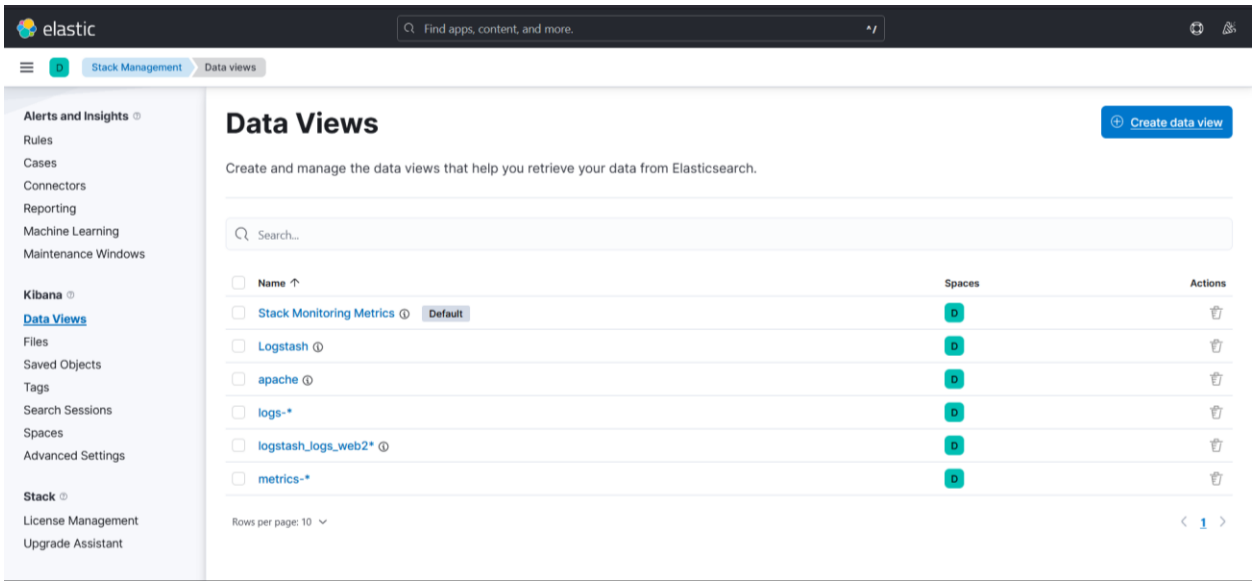
II. In menu Go to stack management in management tab



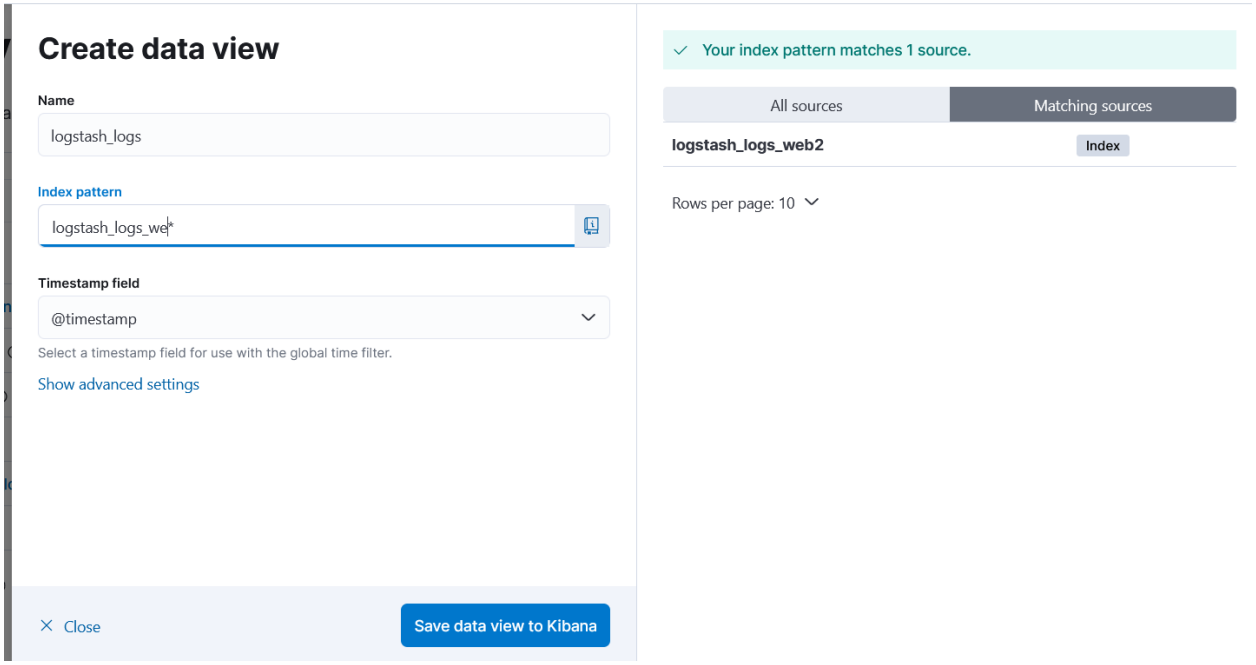
III. Then go to data views tab



IV. Click create new data view

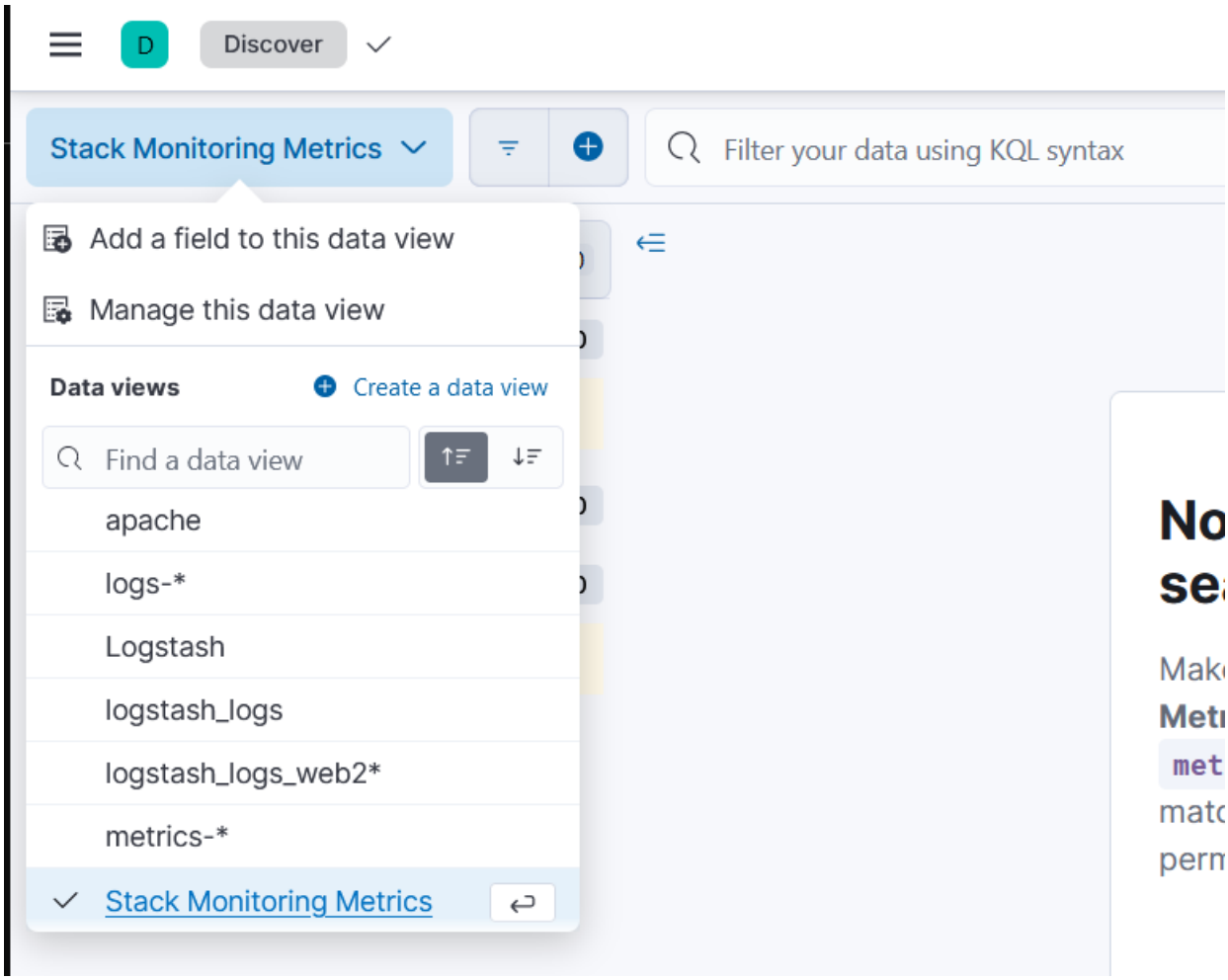


V. Type a name in index pattern according to the index name you enter in the Logstash pipeline. It will automatically select if there is a log index file. Save the file and exit

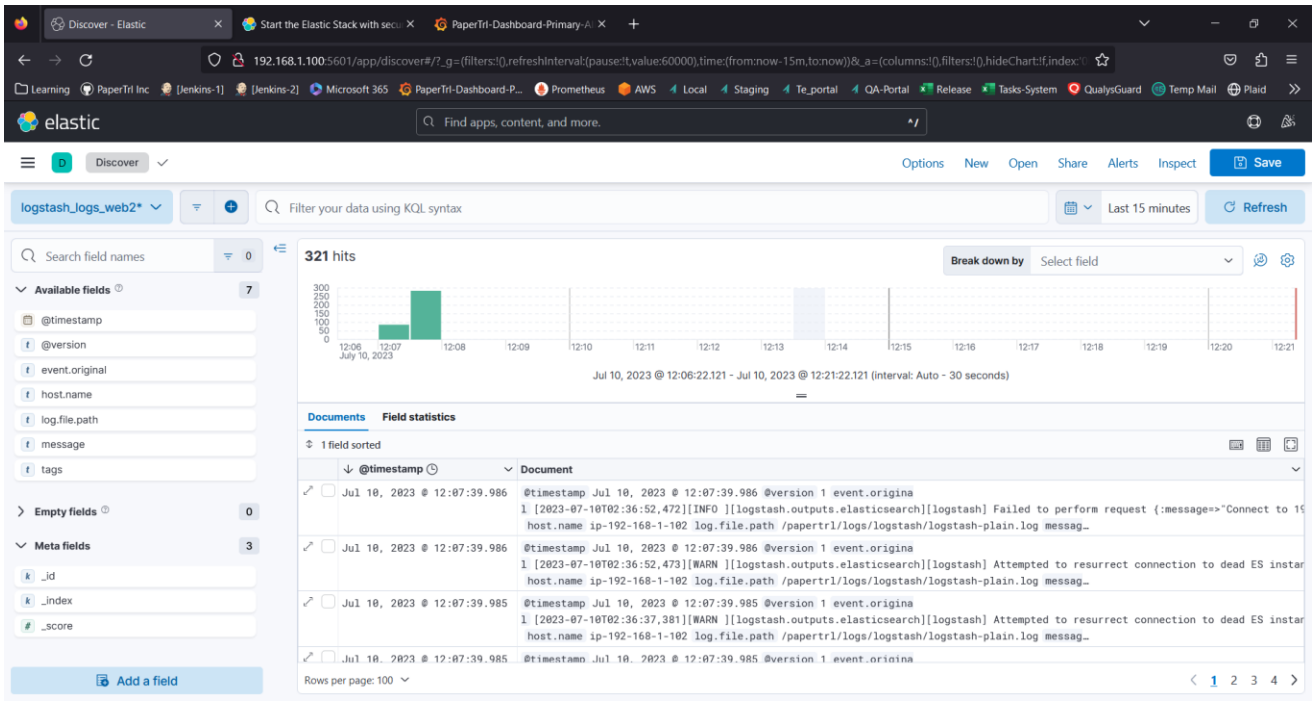


VI. Then go to discover tab in main menu

VII. Then select the index patten from the data views field in stack monitoring metrics



VIII. You can select the message field to view the logs from relevant index



6. Enable User login and Security

I. **Reference :** https://www.elastic.co/guide/en/elasticsearch/reference/8.8/configuring-stack-security.html?blade=kibanasecuritymessage

- systemctl STOP elasticsearch&&kibana

II. Edit the Elasticsearch yml file

- Add:
- xpack.security.enabled: true

- systemctl start elasticsearch
- cd /usr/share/elasticsearch/bin
- ./elasticsearch-setup-passwords auto→ this will generate a passwords pls save it some ware else

III. Edit the Kibana yml as

- elasticsearch.username: "kibana"
- elasticsearch.password: "5KrhmGsSJOpZPvOZoLWQ" generated pswd by elasticsearch
- systemctl start kibana

- log in to the system using UN:elasticsearch and PW:autogenerated Pws

END OF DOCUMENT