Table of Contents

[1 Introduction 2](#_Toc70900869)

[1.1 Preface 2](#_Toc70900870)

[1.2 Audience 2](#_Toc70900871)

[1.3 Related documentation 2](#_Toc70900872)

[2 Jumpcloud 2](#_Toc70900873)

[3 Architecture 3](#_Toc70900874)

[3.1 Overview 3](#_Toc70900875)

[4 Metrics data collection 4](#_Toc70900876)

[4.1 Overview 4](#_Toc70900877)

[4.2 Filebeat 4](#_Toc70900878)

[4.2.1 Suricata module 5](#_Toc70900879)

[4.3 Metricbeat 5](#_Toc70900880)

[4.3.1 System module 5](#_Toc70900881)

[4.4 Auditbeat 5](#_Toc70900882)

[5 Installation and Configuration 5](#_Toc70900883)

[5.1 Docker 5](#_Toc70900884)

[5.1.1 Docker Installation 5](#_Toc70900885)

[5.1.2 Docker-compose Installation 6](#_Toc70900886)

[5.1.3 Docker-compose configuration 7](#_Toc70900887)

[5.1.3.1 Single node Elasticsearch cluster 7](#_Toc70900888)

[5.1.3.1 Three node Elasticsearch cluster 8](#_Toc70900889)

[5.1.4 Docker-compose commands 10](#_Toc70900890)

[5.2 Auditbeat 10](#_Toc70900891)

[5.2.1 Auditbeat installation 10](#_Toc70900892)

[5.2.2 Auditbeat configuration 11](#_Toc70900893)

[5.3 Metricbeat 13](#_Toc70900894)

[5.3.1 Metricbeat installation 13](#_Toc70900895)

[5.3.2 Metricbeat configuration 13](#_Toc70900896)

[5.3 Filebeat 18](#_Toc70900897)

[5.3.1 Filebeat installation 18](#_Toc70900898)

[5.3.2 Filebeat configuration 18](#_Toc70900899)

[5.4 Logstash 22](#_Toc70900900)

[5.4.1 Logstash installation 22](#_Toc70900901)

[5.4.2 Logstash configuration 23](#_Toc70900902)

[5.4.2.1 Pipeline to consume Beats data 23](#_Toc70900903)

[5.4.2.1 Pipeline to consume dcm4chee data 25](#_Toc70900904)

[5.4.3 Running logstash in Docker container 27](#_Toc70900905)

[6 Dashboards 28](#_Toc70900906)

[7 Important Commands and Debugging info 28](#_Toc70900907)

[8 Important References 29](#_Toc70900908)

# 1 Introduction

# 1.1 Preface

This document outlines information about connecting to various servers for collecting data related to various types metrics on servers, alerts generated on Suricata servers along with configuration. Document also provides information visualization of data using beautiful dashboards on Kibana.

# 1.2 Audience

This document is intended for the following audiences:

* Development, administration and operations teams

# 1.3 Related documentation

Elasticsearch documentation: <https://www.elastic.co/>

Logstash documentation: <https://www.elastic.co/logstash>

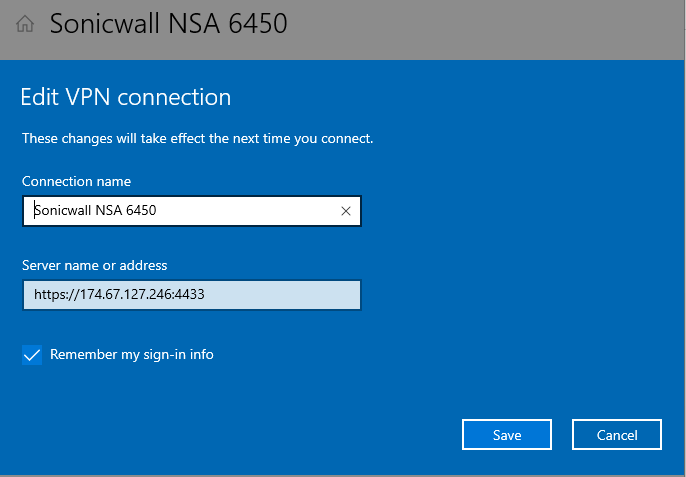
Kibana documentation: <https://www.elastic.co/kibana>

# 2 Jumpcloud

All authorized users must have access to jumpcloud. To connect to jumpclud (which is actually on google cloud) a VPN connection should be established using “Sonicwall Mobile Connect”.

Install windows or required OS version of “Sonicwall Mobile Connect” on the PC/Laptop. Please note that I am going to explain steps below for connecting to VPN using Windows OS version of VPN client. Download “Sonicwall Mobile Connect” client from Microsoft app store and install it.

*Figure 2.1 Sonicwall VPN client configuration*



Please provide above configuration along with username/password to connect to VPN.

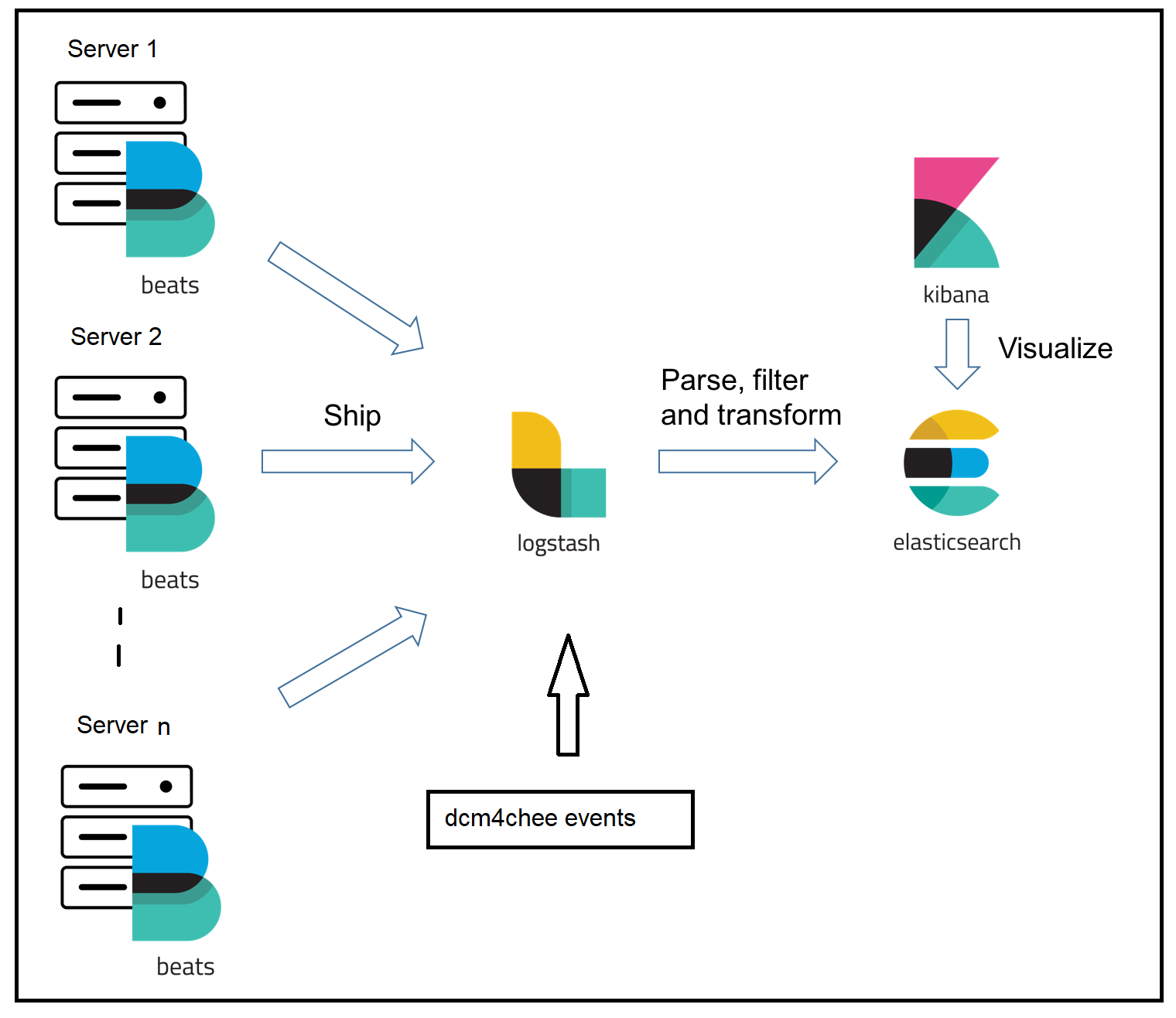
# 3 Architecture

# 3.1 Overview

Architecture is designed to collect System/server statistics or metrics information as well as dcm4chee events from medical servers. System/server statistics or metrics information is collected using Elasticsearch Beats agents.

As shown in figure below, different types of beats agents installed on different servers, send data to UDP 5044 port of VM where Logstash application is installed and running. Logstash pipeline listens on different UDP ports and sends data to Elasticsearch, where it will be stored and indexed. Kibana is used to fetch the data for various indexes and different dashboards are used to visualize the data. For Dcm4chee events, logstash pipeline is configured to collect GELF (Graylog Extended Log Format) events and also to listen on UDP 8514, TCP 8514, 6514 ports for different types of audit events.

*Figure 3.1 Architecture to collect metrics and dcm4chee events*



# 4 Metrics data collection

# 4.1 Overview

System/server statistics or metrics information can be collected using Elasticsearch Beats agents. Beats is a free and open platform for single-purpose data shippers. They can send data from hundreds or thousands of machines and systems to Logstash or Elasticsearch. There are different types of Elasticsearch Beats agents present for collecting different types of statistics.

In this project Metricbeat, Filebeat and Auditbeat are used.

# 4.2 Filebeat

Filebeat is a lightweight shipper for forwarding and centralizing log data. Installed as an agent on servers, Filebeat monitors the log files or locations that you specify, collects log events, and forwards them either to [Elasticsearch](https://www.elastic.co/products/elasticsearch) or [Logstash](https://www.elastic.co/products/logstash) for indexing. It supports many types of modules. In this project, Suricata module of Filebeat is used.

# 4.2.1 Suricata module

This is a module to the Suricata IDS/IPS/NSM log. It parses logs that are in the [Suricata Eve JSON format](https://suricata.readthedocs.io/en/latest/output/eve/eve-json-format.html). When you run the module, it performs a few tasks under the hood:

* Sets the default paths to the log files (but don’t worry, you can override the defaults)
* Makes sure each multiline log event gets sent as a single event
* Uses ingest node to parse and process the log lines, shaping the data into a structure suitable for visualizing in Kibana
* Deploys dashboards for visualizing the log data

# 4.3 Metricbeat

Metricbeat is a lightweight shipper that you can install on your servers to periodically collect metrics from the operating system and from services running on the server. Metricbeat takes the metrics and statistics that it collects and ships them to the output that you specify, such as Elasticsearch or Logstash.

# 4.3.1 System module

The System module allows you to monitor your servers. The default metricsets  are cpu, load, memory, network, process and process\_summary. To disable a default metricset, comment it out in the modules.d/system.yml configuration file. If all metricsets are commented out and the System module is enabled, Metricbeat uses the default metricsets.

# 4.4 Auditbeat

Auditbeat is a lightweight shipper that you can install on your servers to audit the activities of users and processes on your systems. For example, you can use Auditbeat to collect and centralize audit events from the Linux Audit Framework. You can also use Auditbeat to detect changes to critical files, like binaries and configuration files, and identify potential security policy violations.

# 5 Installation and Configuration

This section provides details about installation of various types of beats agents, docker, elasticsearch, logstash, kibana applications and configuration of same.

# 5.1 Docker

# 5.1.1 Docker Installation

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime and in the case of Docker containers - images become containers when they run on [Docker Engine](https://www.docker.com/products/container-runtime). Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Please follow steps mentioned below for installation of Docker.

* sudo yum install -y yum-utils
* sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
* sudo yum makecache
* sudo yum install docker-ce
* sudo systemctl enable --now docker
* systemctl status docker
* sudo usermod -aG docker $USER
* id $USER uid=1000(clutchsolutions) gid=1000(clutchsolutions) groups=1000(clutchsolutions),10(wheel),984(docker)

For more details please refer to link <https://computingforgeeks.com/install-docker-and-docker-compose-on-rhel-8-centos-8/>

# 5.1.2 Docker-compose Installation

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application’s services. Then, with a single command, you create and start all the services from your configuration. To learn more about all the features of Compose, see [the list of features](https://docs.docker.com/compose/#features)

* sudo yum -y install curl
* sudo yum install wget
* curl -s https://api.github.com/repos/docker/compose/releases/latest | grep browser\_download\_url | grep docker-compose-Linux-x86\_64 | cut -d '"' -f 4 | wget -qi -
* chmod +x docker-compose-Linux-x86\_64
* sudo mv docker-compose-Linux-x86\_64 /usr/local/bin/docker-compose
* sudo curl -L https://raw.githubusercontent.com/docker/compose/master/contrib/completion/bash/docker-compose -o /etc/bash\_completion.d/docker-compose
* source /etc/bash\_completion.d/docker-compose

For more details please refer to links

<https://docs.docker.com/compose/>

<https://computingforgeeks.com/how-to-install-latest-docker-compose-on-linux/>

Using Compose is basically a three-step process:

1. Define your app’s environment with a Dockerfile so it can be reproduced anywhere.
2. Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.
3. Run docker-compose up and Compose starts and runs your entire app.

# 5.1.3 Docker-compose configuration

# 5.1.3.1 Single node Elasticsearch cluster

ELasticsearch and Kibana are run in docker container using docker-compose. Currently elasticsearch single node cluster is configured. Please find docker-compose.yml file running elasticsearch and kibana inside docker below.

*CONFIG 5.1.3.1* /home/clutchsolutions/forDocker/*kibana.yml*



*CONFIG 5.1.3.2 docker-compose.yml for single node elasticsearch and kibana*



# 5.1.3.1 Three node Elasticsearch cluster

ELasticsearch and Kibana are run in docker container using docker-compose. To configure three node elasticsearch cluster with kibana, please use below docker-compose.yml file below.

Please use same kibana.yml configuration as single node cluster mentioned above.

*CONFIG 5.1.3.3 docker-compose.yml for three node elasticsearch and kibana*



# 5.1.4 Docker-compose commands

* To start the service

docker-compose up <service-name>

* To start the containers in the background and leaves them running

docker-compose up --detach

* To check the status of services running

docker-compose ps

* To stop the services

docker-compose stop

# 5.2 Auditbeat

# 5.2.1 Auditbeat installation

Please follow steps mentioned below for installation of [auditbeat](https://www.elastic.co/guide/en/beats/auditbeat/current/setup-repositories.html#_yum).

To add the Beats repository for YUM:

1. Download and install the public signing key:

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

1. Create a file with a .repo extension (for example, elastic.repo) in your /etc/yum.repos.d/ directory and add the following lines:

[elastic-7.x]

name=Elastic 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

1. Your repository is ready to use. For example, you can install Auditbeat by running:

sudo yum install auditbeat

1. To configure Auditbeat to start automatically during boot, run:

sudo systemctl enable auditbeat

1. If your system does not use systemd then run:

sudo chkconfig --add auditbeat

# 5.2.2 Auditbeat configuration

Currently Auditbeat is configured on below 13 VM instances.

* elasticsearch-init VM (10.3.0.14)
* logstash-init VM (10.3.0.15)
* 395-stunnelVM (192.168.4.143)
* Suricata VM (192.168.4.155)
* arc1 (10.60.1.150)
* arc2 (10.60.1.160)
* postgres1 (10.60.1.10)
* postgres2 (10.60.1.11)
* ats (10.60.1.190)
* bacula (192.168.139.55)
* river stunnel (192.168.155.143)
* svd stunnel (10.0.1.143)
* stunnel2 (192.168.4.144)

Beats can send data to either elasticsearch or logstash. In this project, all the beats are configured to send data on port 5044 to logstash-init VM (10.3.0.15) where logstash is running. Then logstash application is configured to listen on beats port 5044 and store data in to elasticsearch, which is running on elasticsearch-init VM (10.3.0.14). Auditbeat configuration is stored under /etc/auditbeat/auditbeat.yml file. Please find configuration below.



# 5.3 Metricbeat

# 5.3.1 Metricbeat installation

Please follow steps mentioned below for installation of [metricbeat](https://www.elastic.co/guide/en/beats/metricbeat/master/metricbeat-installation-configuration.html)

To add the Beats repository for YUM:

1. Download and install the public signing key:

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

2. Create a file with a .repo extension (for example, elastic.repo) in your /etc/yum.repos.d/ directory and add the following lines:

1. [elastic-8.x]
2. name=Elastic repository for 8.x packages
3. baseurl=https://artifacts.elastic.co/packages/8.x/yum
4. gpgcheck=1
5. gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
6. enabled=1
7. autorefresh=1

type=rpm-md

* 1. Your repository is ready to use. For example, you can install Metricbeat by running:

sudo yum install metricbeat

* 1. To configure Metricbeat to start automatically during boot, run:

sudo systemctl enable metricbeat

* 1. If your system does not use systemd then run:

sudo chkconfig --add metricbeat

# 5.3.2 Metricbeat configuration

Currently Metricbeat is configured on below 13 VM instances.

* elasticsearch-init VM (10.3.0.14)
* logstash-init VM (10.3.0.15)
* 395-stunnelVM (192.168.4.143)
* Suricata VM (192.168.4.155)
* arc1 (10.60.1.150)
* arc2 (10.60.1.160)
* postgres1 (10.60.1.10) with postgres config
* postgres2 (10.60.1.11) with postgres config
* ats (10.60.1.190)
* bacula (192.168.139.55)
* river stunnel (192.168.155.143)
* svd stunnel (10.0.1.143)
* stunnel2 (192.168.4.144)

Beats can send data to either elasticsearch or logstash. In this project, all the beats are configured to send data on port 5044 to logstash-init VM (10.3.0.15) where logstash is running. Then logstash application is configured to listen on beats port 5044 and store data in to elasticsearch, which is running on elasticsearch-init VM (10.3.0.14).

Metricbeat configuration is stored under /etc/metricbeat/ metricbeat.yml file. Currently all system related metrics are collected, so system.yml is configured for the same and for two server we are configuring postgresql.yml is configure for the same

Please find configuration details below.

/etc/metricbeat/modules.d/System.yml file:



/etc/metricbeat/modules.d/postgresql.yml file:

# Module: postgresql

# Docs: https://www.elastic.co/guide/en/beats/metricbeat/7.10/metricbeat-module-postgresql.html

- module: postgresql

metricsets:

- database

- bgwriter

- activity

period: 10s

hosts: ["postgres://{hostname}:{port}/postgres?sslmode=disable"]

username: {username}

password: {password}

/etc/metricbeat/metricbeat.yml file:



# 5.3 Filebeat

# 5.3.1 Filebeat installation

Please follow steps mentioned below for installation of [filebeat](https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-installation-configuration.html)

To add the Beats repository for YUM:

1. Download and install the public signing key:

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

1. Create a file with a .repo extension (for example, elastic.repo) in your /etc/yum.repos.d/ directory and add the following lines:

[elastic-7.x]

name=Elastic 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

1. Your repository is ready to use. For example, you can install Filebeat by running:

sudo yum install filebeat

1. To configure Filebeat to start automatically during boot, run:

sudo systemctl enable filebeat

If your system does not use systemd then run:

sudo chkconfig --add filebeat

# 5.3.2 Filebeat configuration

Currently Filebeat is configured on below 5 VM instances.

* Suricata VM (192.168.4.155)
* Portal-page1 (10.3.0.4) – Drupal Logs
* Portal-page2 (10.3.0.2) – Drupal Logs
* Portal-page3 (10.240.0.13) – Drupal Logs
* Portal-page4 (10.240.0.10) – Drupal Logs

Beats can send data to either elasticsearch or logstash. In this project, all the beats are configured to send data on port 5044 to logstash-init VM (10.3.0.15) where logstash is running. Then logstash application is configured to listen on beats port 5044 and store data in to elasticsearch, which is running on elasticsearch-init VM (10.3.0.14).

For suricata, filebeat.yml and suricata.yml configuration files are used. Usually configuration files are loaded under /etc folder.

*CONFIG 5.3.2.1 /etc/filebeat/modules.d/suricata.yml*



*CONFIG 5.3.2.2 /etc/filebeat/filebeat.yml*



For Portal Page, filebeat.yml Usually configuration files are loaded under /etc folder.

*CONFIG 5.3.2.1 /etc/filebeat/modules.d/filebeat.yml for Drupal logs*

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: log

# Change to true to enable this input configuration.

enabled: true

# Paths that should be crawledand fetched. Glob based paths.

paths:

- "/home/drupallogs/\*.json.log"

json.keys\_under\_root: false

json.overwrite\_keys: false

json.add\_error\_key: true

json.message\_key: log

#- c:\programdata\elasticsearch\logs\\*

# Exclude lines. A list of regular expressions to match. It drops the lines that are

# matching any regular expression from the list.

#exclude\_lines: ['^DBG']

# Include lines. A list of regular expressions to match. It exports the lines that are

# matching any regular expression from the list.

#include\_lines: ['^ERR', '^WARN']

# Exclude files. A list of regular expressions to match. Filebeat drops the files that

# are matching any regular expression from the list. By default, no files are dropped.

#exclude\_files: ['.gz$']

# Optional additional fields. These fields can be freely picked

# to add additional information to the crawled log files for filtering

fields\_under\_root: true

fields:

tags: ['json']

# level: debug

# review: 1

### Multiline options

# Multiline can be used for log messages spanning multiple lines. This is common

# for Java Stack Traces or C-Line Continuation

# 5.4 Logstash

# 5.4.1 Logstash installation

1. Download and install the public signing key:

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

2. Add the following in your /etc/yum.repos.d/ directory in a file with a .repo suffix, for example logstash.repo

[logstash-7.x]

name=Elastic 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

3. And your repository is ready for use. You can install it with:

sudo yum install logstash

1. Install java if not present

sudo yum install java-11-openjdk-devel

1. Create the logstash.service file.

Edit the file /etc/logstash/startup.options and put the variables for your java binary and java home, something like that:

JAVACMD=/opt/java/jdk1.8.0\_161/bin/java

JAVA\_HOME=/opt/java/jdk1.8.0\_161

Run the script /usr/share/logstash/bin/system-install

1. Start logstash service

sudo systemctl start logstash.service

1. Enable logstash on start up

sudo systemctl enable logstash

# 5.4.2 Logstash configuration

Logstash application is installed and run on “logstash-init” VM (IP 10.3.0.15). Logstash pipelines are configured under /etc/logstash/conf.d/ folder

# 5.4.2.1 Pipeline to consume Beats data

A logstash pipeline is configured to listen for Beats data on 5044 port and then data is stored in to respective beats indices, such as filebeat-7.9.1, metricbeat-7.9.3 and auditbeat-7.10.0. Suricata data is enriched with geo location information based on source and destination IP addresses using MaxMind database.

beats\_input\_elk\_out.conf:



# 5.4.2.1 Pipeline to consume dcm4chee data

This pipeline is configured to receive system and audit log messages from the DICOM Archive [dcm4chee-arc-light](https://github.com/dcm4che/dcm4chee-arc-light/wiki). Details of dcm4chee Docker image are available here <https://github.com/dcm4che-dockerfiles/logstash-dcm4chee>

dcm4chee\_input\_elk\_out.conf pipeline is configured to listen on TCP ports 8514, 6514 and 8514 port of UDP to consume audit log messages and also to consume gelf (Graylog Extended Log Format (GELF) messages and data is enriched and then stored in to “logstash-wildfly-%{+YYYY.MM}" index on Elasticsearch. MM represents month and YYYY represents year. A new index is created every month. To run the pipeline, there are more logstash plugins which needs to be installed. Please find procedure to install the same below.

* Create directory: /usr/share/logstash/data/filter-hashtree
* Download logstash-filter-hashtree and logstash-codec-frame plugins
* /usr/share/logstash/bin/logstash-plugin install --version 6.4.3 logstash-filter-hashtree
* /usr/share/logstash/bin/logstash-plugin install --version 5.0.0 logstash-codec-frame



# 5.4.3 Running logstash in Docker container

Instead of installing and running logstash as a standalone application, it can be run in a docker container. Above beats and dcm4chee pipelines path is provided to docker run command to perform the same along with dependent port and logstash plugins.

Logstash pipeline configuration files are placed under path /home/clutchsolutions/forDocker/logstash/pipeline/ which is provided as mount volume. Please note one important modification that needs to be performed for running logstash pipelines is, elasticsearch output plugin “hosts” field should be configured with IP address instead of hostnames in both the pipelines, as it is not able to resolve IP address from hostname.

elasticsearch {

#hosts => ["http://elasticsearch:9200"]

hosts => ["http://10.3.0.14:9200"]

index => "%{[@metadata][beat]}-%{[@metadata][version]}"

}

* Install docker logstash image:

docker pull docker.elastic.co/logstash/logstash:7.10.0

* To create and run docker image with ports 5044, 8514, 6514 and 12201 :

docker run --rm -p 5044:5044 -p 8514:8514 -p 6514:6514 -p 12201:12201/udp --name logstash -d -v /home/clutchsolutions/forDocker/logstash/pipeline/:/usr/share/logstash/pipeline/ docker.elastic.co/logstash/logstash:7.10.0 sh -c "/usr/share/logstash/bin/logstash-plugin install --version 5.0.0 logstash-codec-frame && /usr/share/logstash/bin/logstash-plugin install --version 6.4.3 logstash-filter-hashtree; /usr/local/bin/docker-entrypoint"

* Docker cmd to know currently running containers:

docker ps

* To enter bash of container:

docker exec -it <container-id> bash

docker exec -it logstash bash

* To get logs of container continuously:

docker logs <container-id> -f

docker logs logstash –f

* To stop the container:

docker stop <container-id or name>

docker stop logstash

* To restart the container:

docker restart <container-id or name>

docker restart logstash

# 6 Dashboards

Kibana is a graphical UI application which is used to create dashboards. Multiple dashboards are created to visualize metrics, events, audit logs and alerts. Kibana by default provides dashboards for events collected by Beats agents. A separate dashboard is created for *dcm4che-Events and Drupal logs*. Kibana supports many number of dashboards and to view a particular dashboard, user must be aware of/remember dashboard title, which is really painful to the customer. A new dashboard (*ASRRAD All Dashboards*) is created with links to all required dashboards.

Kindly find screenshots of all the dashboards along with procedure to reload them from below link

<https://github.com/reddy2020n/asrrad-data-analytics/tree/main/onstak/documentation>

# 7 Important Commands and Debugging info

* To start the containers in the background and leaves them running

docker-compose up –detach

* To check the status of services running

docker-compose ps

* To stop the services

docker-compose stop

* To check containers running

docker ps

* To check logs of a container

docker logs <container-id/name> -f

Get <container-id/name> from docker ps command

* To login into container bash

docker exec –it <container-id/name> bash

* To stop container

docker stop <container-id/name>

* To start container

docker start <container-id/name>

* To restart container

docker restart <container-id/name>

* To shutdown the buggy containers or to remove unused containers and images

docker system prune -af

After the above command just try to build and deploy again,

* To check logs from of any particular beat. For example filebeat

tail –f /var/log/messages | grep filebeat

* To check logstash log

tail –f /var/log/messages | grep logstash

* To start a process for example metricbeat

systemctl start metricbeat

* To stop a process for example metricbeat

systemctl stop metricbeat

* To know status of a process for example metricbeat

systemctl status metricbeat

# 8 Important References

Code base: <https://github.com/reddy2020n/asrrad-data-analytics/>

Beats: <https://www.elastic.co/beats/>

Suricata: <https://github.com/jasonish/docker-suricata>

[DCM4CHEE docker info: https://github.com/dcm4che-dockerfiles/logstash-dcm4chee](https://github.com/dcm4che-dockerfiles/logstash-dcm4chee)

<https://github.com/dcm4che/dcm4chee-arc-light/wiki/Run-secured-archive-services-and-Elastic-Stack-on-a-single-host>