

Elastic Stack 7.9 Deployment Guide

For

Neustar

December, 2020

Version 1.0

Softility, Inc.

607 Herndon Pkwy, Suite 100-C, Herndon, VA 20170

Ph: 703 853 184

Table of Contents

[1 Elastic Stack Servers 6](#_Toc58432988)

[2 Service Accounts 6](#_Toc58432989)

[3 Monitored Datasources and Logs 7](#_Toc58432990)

[4 Prerequisites 7](#_Toc58432991)

[4.1 Command line tools 7](#_Toc58432992)

[4.2 Python Prerequsites 7](#_Toc58432993)

[4.3 Firewall Port access 8](#_Toc58432994)

[4.4 Enable Java 8](#_Toc58432995)

[4.5 Starup Scripts 9](#_Toc58432996)

[4.5.1 Kibana 9](#_Toc58432997)

[4.5.2 Elastic Search 14](#_Toc58432998)

[4.5.3 Logstash 19](#_Toc58432999)

[5 Creating Certificates 21](#_Toc58433000)

[6 Core Components Installation and Configuration 26](#_Toc58433001)

[6.1 ElasticSearch 26](#_Toc58433002)

[6.1.1 Installation Steps 26](#_Toc58433003)

[6.1.2 Extract admin\_dn and node\_dn 26](#_Toc58433004)

[6.1.3 Certificate Path: 27](#_Toc58433005)

[6.2 OpenDistro Security Plugin Installation and configuration: 27](#_Toc58433006)

[6.2.1 Co-ordinate Node Configuration 27](#_Toc58433007)

[6.2.2 Master Node Configuration 29](#_Toc58433008)

[6.2.3 Hot node configuration 31](#_Toc58433009)

[6.2.4 Warm Node configuration 33](#_Toc58433010)

[6.2.5 Update Open Distro config 36](#_Toc58433011)

[6.3 Heap configuration and Initilize elasticsearch nodes 36](#_Toc58433012)

[6.3.1 Heap Configuration 36](#_Toc58433013)

[6.3.2 Initialize Open Distro 37](#_Toc58433014)

[6.3.3 Start Elasticsearch 37](#_Toc58433015)

[6.4 Logstash 39](#_Toc58433016)

[6.4.1 Installation Steps 39](#_Toc58433017)

[6.4.2 Configuration Steps 39](#_Toc58433018)

[6.4.3 Logstash Open Distro Configuration 39](#_Toc58433019)

[6.5 Kibana 40](#_Toc58433020)

[6.5.1 Installation Steps 40](#_Toc58433021)

[6.5.2 Configuration Steps 41](#_Toc58433022)

[6.6 Beats 42](#_Toc58433023)

[6.6.1 FileBeat 42](#_Toc58433024)

[6.6.2 Metric Beat 45](#_Toc58433025)

[6.6.3 Winlog Beat 49](#_Toc58433026)

[7 Stack Monitoring 50](#_Toc58433027)

[8 Logstash GCP Plugins 52](#_Toc58433028)

[8.1 Cloud Storage 52](#_Toc58433029)

[8.2 Pub/Sub 52](#_Toc58433030)

[9 Memcached Installtion and Configuration 53](#_Toc58433031)

[9.1 Prerequisites: 53](#_Toc58433032)

[9.2 Installation: 53](#_Toc58433033)

[9.3 Python3 modules: 54](#_Toc58433034)

[9.4 Running Memcached: 55](#_Toc58433035)

[9.5 Fetching CMDB data from ServiceNow: 56](#_Toc58433036)

[9.6 Creating and updating Memcached database: 56](#_Toc58433037)

[9.7 Updating CMDB data from ServiceNow: 56](#_Toc58433038)

[9.8 Crontab Scripts: 57](#_Toc58433039)

[10 Nginx Setup on Kibana 57](#_Toc58433040)

[11 Kafka and Zookeeper Installation and configuration 63](#_Toc58433041)

[11.1 Hardware Specifications and Prerequsites 63](#_Toc58433042)

[11.1.1 Kafka & zookeeper cluster for each server 63](#_Toc58433043)

[11.1.2 Zookeeper & Kafka Pre-Requisites 63](#_Toc58433044)

[11.1.3 Kafka and Zookeeper Installation 67](#_Toc58433045)

[11.1.4 Zookeeper & Kafka service Creation 73](#_Toc58433046)

[11.1.5 Starting zookeeper & Kafka 75](#_Toc58433047)

[11.1.6 Testing Kafka cluster 75](#_Toc58433048)

[11.1.7 Topics creation 76](#_Toc58433049)

[11.1.8 Monitoring 77](#_Toc58433050)

[12 Elastalert Installation and configuration 81](#_Toc58433051)

[12.1 Elastalert Pre-Requisites 81](#_Toc58433052)

[12.2 System, Network and Other Pre-requisites 82](#_Toc58433053)

[12.3 Download and Install Elastalert and SW Packages 82](#_Toc58433054)

[12.3.1 Installing python3 82](#_Toc58433055)

[12.3.2 Installation of Elastalert and its modules 83](#_Toc58433056)

[12.3.3 Install code base 84](#_Toc58433057)

[12.4 Post Install steps and Configs 85](#_Toc58433058)

[12.4.1 edit config.yaml 85](#_Toc58433059)

[12.4.2 Create Index 86](#_Toc58433060)

[12.5 Installation issues and additional steps 87](#_Toc58433061)

[13 Alerta Installtion and Configuration 89](#_Toc58433062)

[13.1 Prerequisites: 89](#_Toc58433063)

[13.2 MongoDB Installation 90](#_Toc58433064)

[13.2.1 MongoDB Prerequisites 90](#_Toc58433065)

[13.2.2 Download tarball and Extract file 90](#_Toc58433066)

[13.2.3 Create mongodb data and log directories 90](#_Toc58433067)

[13.2.4 MongoDB Configuration 91](#_Toc58433068)

[13.2.5 Service Creation 92](#_Toc58433069)

[13.2.6 Start Mongodb Service 92](#_Toc58433070)

[13.2.7 Recommended ulimit Settings 93](#_Toc58433071)

[13.3 AlertaServer Installtion 94](#_Toc58433072)

[13.3.1 Install Python3 and alerta modules 94](#_Toc58433073)

[13.3.2 Configuration 94](#_Toc58433074)

[13.3.3 Step 4 : export the envirement variables 96](#_Toc58433075)

[13.4 Alerta UI configuration 97](#_Toc58433076)

[13.4.1 install the web console of alerta run: 97](#_Toc58433077)

[13.4.2 Service Creation and Running Alerta 97](#_Toc58433078)

[14 Elasticsearch Deployment using Ansible 98](#_Toc58433079)

[14.1 Steps before running the ansible playbook: 98](#_Toc58433080)

[14.2 Playbook for elastic search cluster: 99](#_Toc58433081)

[14.3 Rolling Restart Ansible playbook 105](#_Toc58433082)

[14.3.1 Playbook for Rolling Restart 106](#_Toc58433083)

[15 Appendix 112](#_Toc58433084)

[15.1 Rolling Restart ES nodes 112](#_Toc58433085)

[15.2 Kiban Dev tool commands 114](#_Toc58433086)

[15.3 Issues and solutions 114](#_Toc58433087)

# Elastic Stack Servers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **IP** | **FQDN** | **OS** | **CPU(s)** | **Ram** |
| Kibana UI Server |  |  |  | 4 | 16 |
| Master-Elastic Server |  |  |  | 4 | 16 |
| Coordinate-Elastic Server |  |  |  | 4 | 16 |
| Hot1-Elastic Server |  |  |  | 8 | 32 |
| Hot2-Elastic Server |  |  |  | 8 | 32 |
| Warm1-Elastic Server |  |  |  | 8 | 32 |
| Warm2-Elastic Server |  |  |  | 8 | 32 |
| Coll Logstash Server |  |  |  | 4 | 16 |
| Coll Logstash Server |  |  |  | 4 | 16 |
| Aggregation Logstash Server |  |  |  | 4 | 16 |
| Aggegation Logstash Server |  |  |  | 4 | 16 |
| Kafka |  |  |  |  |  |
| Elastalert/alerting Engine |  |  |  |  |  |

# Service Accounts

Following are the accounts used by Elastic stack

|  |  |  |
| --- | --- | --- |
| Attribute | Username | Password |
| Elastic server logins | eopsadmin |  |
| Kibana Prod URL login | elastic | elastic |

# Monitored Datasources and Logs

|  |  |  |
| --- | --- | --- |
| **DataSource** | **Filter Condition** | **File Path** |
| Aws VPC | S3 | S3 vpc files |
| Aws CT | S3 | S3 ct files |
| Aws GD | S3 | S3 gd files |
| OnpremSflow | Onprem |  |
| OnpremNetflow | Onprem |  |
| OnpremSyslog | Onprem |  |
| OnpremIpfix | Onprem |  |
| MetricBeat |  |  |
| Packetbeat |  |  |
| Filebeat |  |  |

# Prerequisites

## Command line tools

We need the below command line tools installed on all servers

|  |  |
| --- | --- |
| **Command line Tool** | **Comment** |
| nslookup | To verify DNS name resolution |
| telnet | to validate port connectivity |
| locate | for file lookup |
| git |  |
| Zip, unzip |  |
| wget |  |
|  |  |
|  |  |

## Python Prerequsites

Python prerequsites are required for Memcached and Elastalert and Logstash servers

Python3 supposed to installed along with PIP3 utility

## Firewall Port access

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Application | Ports | Protocol | External Ports | Internal Ports |
| Elasticsearch | 9200, 9300 | TCP | 9200 | 9300 |
| Kibana | 5601 | TCP | 5601 |  |
| Logstash | 9600 | TCP |  | 9600 |
| Nginx | 443 | TCP | 443 |  |
| Memcached | 11211 | TCP |  | 11211 |
| Elastalert,API | 3030, 3333 | TCP | 3030 | 3333 |
| Alerting Engine | 8081, 8000, 443 | TCP | 8000, 443 | 8081 |
| Praeco | 8080, 443 | TCP | 8080, 443 |  |
| Kafka & Zookeeper | 2181, 9092, 9093, 2888, 3888, 9999, 8778, 9000 | TCP | 9000, 9092, 2181, 8778, 9999 | 2181, 9093, 2888, 3888 |
| LS Syslog datasource | 5514, 5515 | TCP |  | 5514, 5515 |
| LS Syslog datasource | 5514, 5515 | UDP |  | 5514, 5515 |
| LS Syslog filebeat | 5525 | TCP |  | 5525 |
| LS winlogbeat | 5535 | TCP |  | 5535 |
| LS flowdata | 2055, 6343 | UDP |  | 2055, 6343 |
| IPFIX | 4739 | TCP |  | 4739 |
| IPFIX | 4739 | UDP |  | 4739 |

## Enable Java

**Java is enabled (Needs to be done in all Elastic Stack servers):**

Perform the below steps on elastic stack all servers:

1. Download java .tar.gz version and place in desired location on the servers /opt/softwares/jdk-8u251-linux-x64.tar.gz
2. Extract to local /opt/eopsadmin/jdk1.8.0\_251 dir.
3. Set JAVA\_HOME variable to this path in user profile and startup scripts as explained in later sections.

vi /home/eopsadmin/.bash.rc

and lines as below:

|  |
| --- |
| export JAVA\_HOME=/opt/eopsadmin/java  export PATH=$PATH:$JAVA\_HOME/bin |

vi /home/eopsadmin/.bash\_profile

Add lines as below:

|  |
| --- |
| export JAVA\_HOME=/opt/eopsadmin/java  export PATH=$PATH:$JAVA\_HOME/bin |

Execute source commands:

source ~/.bashrc

source ~/.bash\_profile

Check Java version using below command:

java –version

which java

echo $JAVA\_HOME

## Starup Scripts

### Kibana

To configure Kibana to start automatically when the system boots up, perform the following steps:

1. Login to the Kibana servers as root user and navigate to /etc/init.d
2. Copy the below content to the “*kibana*” file and place it in the /etc/init.d directory

|  |  |
| --- | --- |
|  | **I) Startup script config and sudo setup**    To configure Kibana to start automatically when the system boots up, perform the following steps:   1. Create a startup script for kibana and copy it to /etc/init.d dir as ‘root’ user vi /etc/init.d/kibana and add the below content   #!/bin/sh  # Init script for kibana  # Maintained by  # Generated by pleaserun.  # Implemented based on LSB Core 3.1:  # \* Sections: 20.2, 20.3  #  ### BEGIN INIT INFO  # Provides: kibana  # Required-Start: $remote\_fs $syslog  # Required-Stop: $remote\_fs $syslog  # Default-Start: 2 3 4 5  # Default-Stop: 0 1 6  # Short-Description:  # Description: Kibana  ### END INIT INFO  name=kibana  program=/opt/eopsadmin/kibana/bin/kibana  args=-c\\\ /opt/eopsadmin/kibana/config/kibana.yml  pidfile="/opt/eopsadmin/kibana/$name.pid"  [ -r /etc/default/$name ] && . /etc/default/$name  [ -r /etc/sysconfig/$name ] && . /etc/sysconfig/$name  export NODE\_OPTIONS  [ -z "$nice" ] && nice=0  trace() {  logger -t "/etc/init.d/kibana" "$@"  }  emit() {  trace "$@"  echo "$@"  }  start() {  # Ensure the log directory is setup correctly.  [ ! -d "/opt/eopsadmin/kibana/log/" ] && mkdir "/opt/eopsadmin/kibana/log/"  chown "$user":"$group" "/opt/eopsadmin/kibana/log/"  chmod 755 "/opt/eopsadmin/kibana/log/"  # Setup any environmental stuff beforehand  # Run the program!  chroot --userspec "$user":"$group" "$chroot" sh -c "  cd \"$chdir\"  exec \"$program\" $args  " >> /opt/eopsadmin/kibana/log/kibana.stdout 2>> /opt/eopsadmin/kibana/kibana.stderr &  # Generate the pidfile from here. If we instead made the forked process  # generate it there will be a race condition between the pidfile writing  # and a process possibly asking for status.  echo $! > $pidfile  emit "$name started"  return 0  }  stop() {  # Try a few times to kill TERM the program  if status ; then  pid=$(cat "$pidfile")  trace "Killing $name (pid $pid) with SIGTERM"  kill -TERM $pid  # Wait for it to exit.  for i in 1 2 3 4 5 ; do  trace "Waiting $name (pid $pid) to die..."  status || break  sleep 1  done  if status ; then  if [ "$KILL\_ON\_STOP\_TIMEOUT" -eq 1 ] ; then  trace "Timeout reached. Killing $name (pid $pid) with SIGKILL. This may result in data loss."  kill -KILL $pid  emit "$name killed with SIGKILL."  else  emit "$name stop failed; still running."  fi  else  emit "$name stopped."  fi  fi  }  status() {  if [ -f "$pidfile" ] ; then  pid=$(cat "$pidfile")  if ps -p $pid > /dev/null 2> /dev/null ; then  # process by this pid is running.  # It may not be our pid, but that's what you get with just pidfiles.  # TODO(sissel): Check if this process seems to be the same as the one we  # expect. It'd be nice to use flock here, but flock uses fork, not exec,  # so it makes it quite awkward to use in this case.  return 0  else  return 2 # program is dead but pid file exists  fi  else  return 3 # program is not running  fi  }  force\_stop() {  if status ; then  stop  status && kill -KILL $(cat "$pidfile")  fi  }  case "$1" in  force-start|start|stop|force-stop|restart)  trace "Attempting '$1' on kibana"  ;;  esac  case "$1" in  force-start)  PRESTART=no  exec "$0" start  ;;  start)  status  code=$?  if [ $code -eq 0 ]; then  emit "$name is already running"  exit $code  else  start  exit $?  fi  ;;  stop) stop ;;  force-stop) force\_stop ;;  status)  status  code=$?  if [ $code -eq 0 ] ; then  emit "$name is running"  else  emit "$name is not running"  fi  exit $code  ;;  restart)  stop && start  ;;  \*)  echo "Usage: $SCRIPTNAME {start|force-start|stop|force-start|force-stop|status|restart}" >&2  exit 3  ;;  esac  exit $?   1. Create kibana config file or copy with the content below to /etc/sysconfig dir as ‘root’ user,   with ‘644’ as file permissions.  user="eopsadmin"  group="eopsadmin"  chroot="/"  chdir="/"  nice=""  Environment="JAVA\_HOME=/opt/eopsadmin/java"  # If this is set to 1, then when `stop` is called, if the process has  # not exited within a reasonable time, SIGKILL will be sent next.  # The default behavior is to simply log a message "program stop failed; still running"  KILL\_ON\_STOP\_TIMEOUT=0   1. Configure the startup script as a service, using the below   As root user run below two:  ‘systemctl enable kibana’  systemctl daemon-reload   1. Validate that softlinks have been created on Run levels directory   #> ll /etc/rc0.d/K50kibana  lrwxrwxrwx. 1 root root 16 Jun 4 20:31 /etc/rc0.d/K50kibana -> ../init.d/kibana     1. Enable sudo permissions for ‘eopsadmin’ user to start and stop kibana.service as below and verify Kibana service is stopped and started by following commands:   sudo systemctl start kibana.service sudo systemctl stop kibana.service  **III) Configure system settings**     1. From root shell prompt edit /etc/security/limits.conf and set below values for 'eopsadmin' user.   eopsadmin - nofile 65535     1. From root shell prompt edit /etc/sysctl.conf and set below values to increase virtual memory.     vm.max\_map\_count=262144  vm.swappiness=1 |

### Elastic Search

1. Login to the servers as root user and navigate to /etc/init.d
2. Create a new file named “elasticsearch” under the above directory and copy the below content

|  |  |
| --- | --- |
|  | 1. **Startup script config and sudo setup:**   Create a startup script for elasticsearch and copy it to /etc/init.d  #!/bin/bash  #  # elasticsearch <summary>  #  # chkconfig: 2345 80 20  # description: Starts and stops a single elasticsearch instance on this system  #  ### BEGIN INIT INFO  # Provides: Elasticsearch  # Required-Start: $network $named  # Required-Stop: $network $named  # Default-Start: 2 3 4 5  # Default-Stop: 0 1 6  # Short-Description: This service manages the elasticsearch daemon  # Description: Elasticsearch is a very scalable, schema-free and high-performance search solution supporting multi-tenancy and near realtime search.  ### END INIT INFO  #  # init.d / servicectl compatibility (openSUSE)  #  if [ -f /etc/rc.status ]; then  . /etc/rc.status  rc\_reset  fi  #  # Source function library.  #  if [ -f /etc/rc.d/init.d/functions ]; then  . /etc/rc.d/init.d/functions  fi  # Sets the default values for elasticsearch variables used in this script  ES\_HOME="/opt/eopsadmin/elasticsearch"  MAX\_OPEN\_FILES=65535  MAX\_MAP\_COUNT=262144  ES\_PATH\_CONF="/opt/eopsadmin/elasticsearch/config"  PID\_DIR="/opt/eopsadmin/elasticsearch/var"  # Source the default env file  ES\_ENV\_FILE="/etc/sysconfig/elasticsearch"  if [ -f "$ES\_ENV\_FILE" ]; then  . "$ES\_ENV\_FILE"  fi  # ES\_USER and ES\_GROUP settings were removed  if [ ! -z "$ES\_USER" ] || [ ! -z "$ES\_GROUP" ]; then  echo "ES\_USER and ES\_GROUP settings are no longer supported. To run as a custom user/group use the archive distribution of Elasticsearch."  exit 1  fi  exec="$ES\_HOME/bin/elasticsearch"  prog="elasticsearch"  pidfile="$PID\_DIR/${prog}.pid"  export ES\_JAVA\_OPTS  export JAVA\_HOME  export ES\_PATH\_CONF  export ES\_STARTUP\_SLEEP\_TIME  lockfile=/var/lock/subsys/$prog  if [ ! -x "$exec" ]; then  echo "The elasticsearch startup script does not exists or it is not executable, tried: $exec"  exit 1  fi  checkJava() {  if [ -x "$JAVA\_HOME/bin/java" ]; then  JAVA="$JAVA\_HOME/bin/java"  else  JAVA=`which java`  fi  if [ ! -x "$JAVA" ]; then  echo "Could not find any executable java binary. Please install java in your PATH or set JAVA\_HOME"  exit 1  fi  }  start() {  checkJava  [ -x $exec ] || exit 5  if [ -n "$MAX\_OPEN\_FILES" ]; then  ulimit -n $MAX\_OPEN\_FILES  fi  if [ -n "$MAX\_LOCKED\_MEMORY" ]; then  ulimit -l $MAX\_LOCKED\_MEMORY  fi  if [ -n "$MAX\_MAP\_COUNT" -a -f /proc/sys/vm/max\_map\_count ] && [ "$MAX\_MAP\_COUNT" -gt $(cat /proc/sys/vm/max\_map\_count) ]; then  sysctl -q -w vm.max\_map\_count=$MAX\_MAP\_COUNT  fi  # Ensure that the PID\_DIR exists (it is cleaned at OS startup time)  if [ -n "$PID\_DIR" ] && [ ! -e "$PID\_DIR" ]; then  mkdir -p "$PID\_DIR" && chown eopsadmin:eopsadmin "$PID\_DIR"  fi  if [ -n "$pidfile" ] && [ ! -e "$pidfile" ]; then  touch "$pidfile" && chown eopsadmin:eopsadmin "$pidfile"  fi  cd $ES\_HOME  echo -n $"Starting $prog: "  # if not running, start it up here, usually something like "daemon $exec"  daemon --user eopsadmin --pidfile $pidfile $exec -p $pidfile -d  retval=$?  echo  [ $retval -eq 0 ] && touch $lockfile  return $retval  }  stop() {  echo -n $"Stopping $prog: "  # stop it here, often "killproc $prog"  killproc -p $pidfile -d 86400 $prog  retval=$?  echo  [ $retval -eq 0 ] && rm -f $lockfile  return $retval  }  restart() {  stop  start  }  reload() {  restart  }  force\_reload() {  restart  }  rh\_status() {  # run checks to determine if the service is running or use generic status  status -p $pidfile $prog  }  rh\_status\_q() {  rh\_status >/dev/null 2>&1  }  case "$1" in  start)  rh\_status\_q && exit 0  $1  ;;  stop)  rh\_status\_q || exit 0  $1  ;;  restart)  $1  ;;  reload)  rh\_status\_q || exit 7  $1  ;;  force-reload)  force\_reload  ;;  status)  rh\_status  ;;  condrestart|try-restart)  rh\_status\_q || exit 0  restart  ;;  \*)  echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload}"  exit 2  esac  exit $?   1. Create an elasticsearch file and copy it to /etc/sysconfig   ################################  # Elasticsearch  ################################  # Elasticsearch home directory  ES\_HOME=/opt/eopsadmin/elasticsearch  # Elasticsearch Java path  #JAVA\_HOME=/opt/eopsadmin/java  JAVA\_HOME=/opt/eopsadmin/java  # Elasticsearch configuration directory  ES\_PATH\_CONF=/opt/eopsadmin/elasticsearch/config  # Elasticsearch PID directory  #PID\_DIR=/var/run/elasticsearch  # Additional Java OPTS  #ES\_JAVA\_OPTS=  # Configure restart on package upgrade (true, every other setting will lead to not restarting)  #RESTART\_ON\_UPGRADE=true  ################################  # Elasticsearch service  ################################  # SysV init.d  #  # The number of seconds to wait before checking if Elasticsearch started successfully as a daemon process  ES\_STARTUP\_SLEEP\_TIME=5  ################################  # System properties  ################################  # Specifies the maximum file descriptor number that can be opened by this process  # When using Systemd, this setting is ignored and the LimitNOFILE defined in  # /usr/lib/systemd/system/elasticsearch.service takes precedence  MAX\_OPEN\_FILES=65535  # The maximum number of bytes of memory that may be locked into RAM  # Set to "unlimited" if you use the 'bootstrap.memory\_lock: true' option  # in elasticsearch.yml.  # When using systemd, LimitMEMLOCK must be set in a unit file such as  # /etc/systemd/system/elasticsearch.service.d/override.conf.  #MAX\_LOCKED\_MEMORY=unlimited  # Maximum number of VMA (Virtual Memory Areas) a process can own  # When using Systemd, this setting is ignored and the 'vm.max\_map\_count'  # property is set at boot time in /usr/lib/sysctl.d/elasticsearch.conf  MAX\_MAP\_COUNT=262144   1. Configure the startup script as a service, using the below   systemctl enable elasticsearch.service  systemctl daemon-reload   1. Validate that softlinks have been created on Run levels directory   Enable sudo permissions for 'eopsadmin' user to start and stop elasticsearch.service  Verify elasticsearch can be started and stopped as follows:  sudo systemctl start elasticsearch.service  sudo systemctl stop elasticsearch.service   1. **Configure system settings:**   From root shell prompt edit /etc/security/limits.conf and set below values for 'eopsadmin' user.  eopsadmin - nofile 65535  From root shell prompt edit /etc/sysctl.conf and set below values to increase virtual memory.  vm.max\_map\_count = 262144  vm.swappiness=1 |

Also, we need to disable swap across all the elastic servers. Please open the /etc/fstab file and comment out any lines that contain the word swap

Ex: Comment out the below line:

/dev/mapper/rhel-swap swap swap defaults 0 0

### Logstash

1. Run this command as 'root' **'sudo /opt/eopsadmin/logstash/bin/system-install'**
2. The above will create the startup service under /etc/systemd/system/

Edit the file logstash.service and verify the code is shown as below

|  |  |
| --- | --- |
|  | **I) Startup script config and sudo setup:**  Navigate to /etc/systemd/system  Create the file logstash.service and include the below code,  [Unit]  Description=logstash  [Service]  Type=simple  User=eopsadmin  Group=eopsadmin  # Load env vars from /etc/default/ and /etc/sysconfig/ if they exist.  # Prefixing the path with '-' makes it try to load, but if the file doesn't  # exist, it continues onward.  EnvironmentFile=-/etc/default/logstash  EnvironmentFile=-/etc/sysconfig/logstash  Environment="JAVA\_HOME=/opt/eopsadmin/java"  #ExecStart=/opt/eopsadmin/logstash/bin/logstash "--path.settings" "/opt/eopsadmin/logstash/config"  ExecStart=/opt/eopsadmin/logstash/bin/logstash "--path.settings" "/opt/eopsadmin/logstash/config" "--path.logs" "/opt/eopsadmin/logstash/logs"  Restart=always  WorkingDirectory=/  Nice=19  LimitNOFILE=65536  StandardOutput=null  [Install]  WantedBy=multi-user.target  Configure logstash as a service,  sudo /bin/systemctl daemon-reload  sudo /bin/systemctl enable logstash.service  sudo systemctl start logstash.service  sudo systemctl stop logstash.service  **II) Configure system settings:**  From root shell prompt edit /etc/security/limits.conf and set below values for 'eopsadmin' user   eopsadmin - nofile 65536    From root shell prompt edit /etc/sysctl.conf and set below values to increase virtual memory.    vm.max\_map\_count = 262144  vm.swappiness=1 |

# Creating Certificates

**Admin certificates:** required for cluster initialization and updating runtime changes

**Node certificates:** required for SSL configuration

**Node Certificate generation:**

1. Run the below commands to generate the .csr and .key files

[eopsadmin@ip-10-75-240-52 eopsadmin]$ openssl req -out laas.neustar.biz.csr -new -newkey rsa:2048 -nodes -keyout laas.neustar.biz.key

Generating a 2048 bit RSA private key

.....+++

...........+++

writing new private key to 'laas.neustar.biz.key'

-----

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [XX]:US

State or Province Name (full name) []:VA

Locality Name (eg, city) [Default City]:Sterling

Organization Name (eg, company) [Default Company Ltd]:Neustar

Organizational Unit Name (eg, section) []:NETENG

Common Name (eg, your name or your server's hostname) []:\*.laas.neustar.biz

Email Address []:neteng@team.neustar

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:el@st!c

An optional company name []:

Copied the two files to different directory:

[eopsadmin@ip-10-75-240-52 neustar-certs]$ pwd

/opt/eopsadmin/certificate/neustar-certs

[eopsadmin@ip-10-75-240-52 neustar-certs]$ ls -lrt

-rw-rw-r--. 1 eopsadmin eopsadmin 1708 Aug 14 17:15 laas.neustar.biz.key

-rw-rw-r--. 1 eopsadmin eopsadmin 1094 Aug 14 17:15 laas.neustar.biz.csr

1. Have the above .csr and .key file singed by Neustar team to obtain the .crt file along with CA certificates.

**Note:** We received the signed crt file content and key file in mail and we obtained/derived the root, intermediate, digi from the signed file using chrome.

We have the below neustar signed certificates

laas.neustar.biz.key

nsdev-cert-signed.pem

nsdev-digi-cert.crt

nsdev-intermediate-cert.crt

* convert crt files to pem using below command

openssl x509 -in <file\_name.crt> -out <file\_name.pem> -outform PEM

* Create a certificate chain

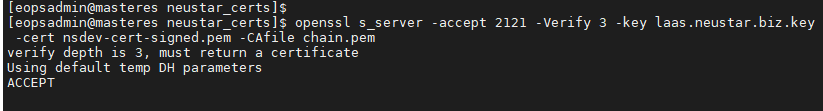
cat nsdev-digi-cert.crt nsdev-intermediate-cert.crt nsdev-cert-signed.pem > nsdev\_chain.pem

we will be using chain.per as root certificate

laas.neustar.biz.key as key and nsdev-cert-signed.pem as node certificate

Verifiy certificates with beow command

openssl s\_server -accept 2121 -Verify 3 -key laas.neustar.biz.key -cert nsdev-cert-signed.pem -CAfile nsdev\_chain.pem



**Admin Certificates:**

Follow similar procedure as node certificate generation for admin certificates aswell. But, **CN should be different for node and admin certificates**.

[eopsadmin@stopvprvlogstash01 eops-secadmin]$ openssl genrsa -out eops-secadmin.key 2048

Generating RSA private key, 2048 bit long modulus

...+++

.............+++

e is 65537 (0x10001)

[eopsadmin@stopvprvlogstash01 eops-secadmin]$ ls -lrt

total 4

rw------. 1 eopsadmin eopsadmin 1679 Nov 5 19:04 eops-secadmin.key

[eopsadmin@stopvprvlogstash01 eops-secadmin]$ openssl req -new -sha256 -key eops-secadmin.key -out eopsadmin-secadmin.csr

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [XX]:US

State or Province Name (full name) []:Virginia

Locality Name (eg, city) [Default City]:Sterling

Organization Name (eg, company) [Default Company Ltd]:Neustar Inc

Organizational Unit Name (eg, section) []:

Common Name (eg, your name or your server's hostname) []:eodsecadmin

Email Address []:infraeng@neustar.biz

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:el@st!c

An optional company name []:

[eopsadmin@stopvprvlogstash01 eops-secadmin]$

[eopsadmin@stopvprvlogstash01 eops-secadmin]$ ls -lrt

total 8

rw------. 1 eopsadmin eopsadmin 1679 Nov 5 19:04 eops-secadmin.key

rw------. 1 eopsadmin eopsadmin 1078 Nov 5 19:08 eopsadmin-secadmin.csr

Have the above .csr and .key file singed by Neustar team to obtain the .crt file along with CA certificates.

Extract intermediate and Digi certificates from signed crt

We have the below neustar signed certificates

eopsadmin-secadmin-1.crt

eopsadmin-secadmin.key

eopsecadmin\_digi.crt

eopsecadmin\_intermediate.crt

* convert crt files to pem using below command

openssl x509 -in <file\_name.crt> -out <file\_name.pem> -outform PEM

* Create a certificate chain

cat eopsecadmin\_digi.pem eopsecadmin\_intermediate.pem eopsadmin-secadmin-1.pem > eopsecadmin\_chain.pem

Convert node header RSA key to Private key

openssl pkcs8 -topk8 -nocrypt -in eopsadmin-secadmin.key > eopsadmin-secadmin-key.pem

**[ Note:** key header is supposed to be -----BEGIN PRIVATE KEY----- **]**

Verify certificates with below command

[eopsadmin@elk-data admin]$

openssl s\_server -accept 2121 -Verify 3 -key eopsadmin-secadmin.key -cert eopsadmin-secadmin-1.pem -CAfile eopsecadmin\_chain.pem

verify depth is 3, must return a certificate

Using default temp DH parameters

ACCEPT

Copy the generated certificates to all the nodes and we will be using node certificates while configuring and admin certificates will be used to initialize cluster.

# Core Components Installation and Configuration

## ElasticSearch

Download the software package and copy to desired location on the server

curl -L -O <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.9.0-linux-x86_64.tar.gz>

### Installation Steps

* Create a new directory “**eopsadmin**” under */opt/* and unzip the software package to this directory.

Move the Elastic Search software package installation dir and untar the software package as shown below:

untar the package elasticsearch-7.9.0-linux-x86\_64.tar.gz as shown below,

[eopsadmin@ip-10-75-240-52 ]$ cd /opt/softwares

[eopsadmin@ip-10-75-240-52 softwares]$ cp elasticsearch-7.9.0-linux-x86\_64.tar.gz /opt/eopsadmin/

[eopsadmin@ip-10-75-240-52 eopsadmin]$ tar -zxvf elasticsearch-7.9.0-linux-x86\_64.tar.gz

Update the permissions for /opt and renameelasticsearch-7.9.0 directory

[eopsadmin@ip-10-75-240-52 eopsadmin]$ mv elasticsearch-7.9.0 elasticsearch

### Extract admin\_dn and node\_dn

Check admin\_dn and node\_dn from the obtained certificates using below command

admin\_dn and node\_dn should have different common name.

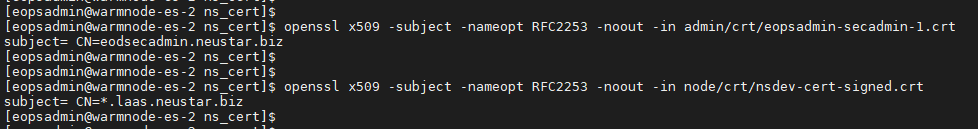
[eopsadmin@warmnode-es-2 ns\_cert]$

[eopsadmin@warmnode-es-2 ns\_cert]$ openssl x509 -subject -nameopt RFC2253 -noout -in admin/crt/eopsadmin-secadmin-1.crt

**subject= CN=eodsecadmin.neustar.biz**

[eopsadmin@warmnode-es-2 ns\_cert]$ openssl x509 -subject -nameopt RFC2253 -noout -in node/crt/nsdev-cert-signed.crt

**subject= CN=\*.laas.neustar.biz**



### Certificate Path:

All the certificates should be under config directory. copy all the admin and node certificates to below path across all the nodes

**/opt/eopsadmin/elasticsearch/config/ns\_cert**

## OpenDistro Security Plugin Installation and configuration:

Elasticsearch and Kibana version 7.9.0

Install Open Distro security plugin in all Elasticsearch nodes and Kibana as well.

**For Elasticsearch**

Navigate to /opt/eopsadmin/elasticsearch

bin/elasticsearch-plugin install <https://d3g5vo6xdbdb9a.cloudfront.net/downloads/elasticsearch-plugins/opendistro-security/opendistro_security-1.11.1.0.zip>

**For Kibana**

Navigate to /opt/eopsadmin/kibana

bin/kibana-plugin install<https://d3g5vo6xdbdb9a.cloudfront.net/downloads/kibana-plugins/opendistro-security/opendistro_security_kibana_plugin-1.11.1.0.zip>

### Co-ordinate Node Configuration

Add the below configurations to elasticsearch.yml in coordinating node

transport.tcp.port: 9300

xpack.ml.enabled: false

cluster.name: “neustar-eopsadmin-logging”

node.name: eopsadmin-coordnode-1

node.master: false

node.data: false

node.ingest: false

network.host: awopuse1elasticcoordnode01.laas.neustar.biz

http.port: 9200

discovery.seed\_hosts: ["awopuse1elasticmasternode01.laas.neustar.biz", "awopuse1elasticdatahotnode01.laas.neustar.biz", "awopuse1elasticdatahotnode02.laas.neustar.biz", "awopuse1elasticdatawarmcoldnode01.laas.neustar.biz", "awopuse1elasticdatawarmcoldnode02.laas.neustar.biz"]

path.data: /opt/eopsadmin/elasticsearch/data

path.logs: /opt/eopsadmin/elasticsearch/logs

cluster.initial\_master\_nodes: ["eopsadmin-masternode-1"]

bootstrap.system\_call\_filter: false

node.max\_local\_storage\_nodes: 10

xpack.security.enabled: false

xpack.monitoring.enabled: true

opendistro\_security.ssl.transport.pemcert\_filepath: certificate/ns\_cert/node/nsdev-cert-signed.pem

opendistro\_security.ssl.transport.pemkey\_filepath: certificate/ns\_cert/node//laas.neustar.biz.key

opendistro\_security.ssl.transport.pemtrustedcas\_filepath: certificate/ns\_cert/node//nsdev\_chain.pem

opendistro\_security.ssl.transport.enforce\_hostname\_verification: false

opendistro\_security.ssl.http.enabled: true

opendistro\_security.ssl.http.pemcert\_filepath: certificate/ns\_cert/node/nsdev-cert-signed.pem

opendistro\_security.ssl.http.pemkey\_filepath: certificate/ns\_cert/node/laas.neustar.biz.key

opendistro\_security.ssl.http.pemtrustedcas\_filepath: certificate/ns\_cert/node/nsdev\_chain.pem

opendistro\_security.allow\_default\_init\_securityindex: true

opendistro\_security.authcz.admin\_dn:

- 'CN=eodsecadmin.neustar.biz'

opendistro\_security.nodes\_dn:

- 'CN=\*.laas.neustar.biz'

opendistro\_security.audit.type: internal\_elasticsearch

opendistro\_security.enable\_snapshot\_restore\_privilege: true

opendistro\_security.check\_snapshot\_restore\_write\_privileges: true

opendistro\_security.restapi.roles\_enabled: ["all\_access", "security\_rest\_api\_access"]

### Master Node Configuration

Add the below configuration to elasticsearch.yml file in master node

transport.tcp.port: 9300

xpack.ml.enabled: false

cluster.name: “neustar-eopsadmin-logging”

node.name: eopsadmin-masternode-1

node.master: true

node.voting\_only: false

node.data: false

node.ingest: false

network.host: awopuse1elasticmasternode01.laas.neustar.biz

http.port: 9200

node.transform: false

node.remote\_cluster\_client: false

discovery.seed\_hosts: ["awopuse1elasticmasternode01.laas.neustar.biz:9300", "awopuse1elasticcoordnode01.laas.neustar.biz:9300", "awopuse1elasticdatahotnode01.laas.neustar.biz:9300", "awopuse1elasticdatahotnode02.laas.neustar.biz:9300", "awopuse1elasticdatawarmcoldnode01.laas.neustar.biz:9300", "awopuse1elasticdatawarmcoldnode02.laas.neustar.biz:9300"]

path.data: /opt/eopsadmin/elasticsearch/data

path.logs: /opt/eopsadmin/elasticsearch/logs

cluster.initial\_master\_nodes: ["eopsadmin-masternode-1"]

bootstrap.system\_call\_filter: false

node.max\_local\_storage\_nodes: 10

xpack.monitoring.enabled: true

xpack.security.enabled: false

opendistro\_security.ssl.transport.pemcert\_filepath: certificate/ns\_cert/node/nsdev-cert-signed.pem

opendistro\_security.ssl.transport.pemkey\_filepath: certificate/ns\_cert/node/laas.neustar.biz.key

opendistro\_security.ssl.transport.pemtrustedcas\_filepath: certificate/ns\_cert/node/nsdev\_chain.pem

opendistro\_security.ssl.transport.enforce\_hostname\_verification: false

opendistro\_security.ssl.http.enabled: true

opendistro\_security.ssl.http.pemcert\_filepath: certificate/ns\_cert/node/nsdev-cert-signed.pem

opendistro\_security.ssl.http.pemkey\_filepath: certificate/ns\_cert/node/laas.neustar.biz.key

opendistro\_security.ssl.http.pemtrustedcas\_filepath: certificate/ns\_cert/node/nsdev\_chain.pem

opendistro\_security.allow\_default\_init\_securityindex: true

opendistro\_security.authcz.admin\_dn:

- 'CN=eodsecadmin.neustar.biz'

opendistro\_security.nodes\_dn:

- 'CN=\*.laas.neustar.biz'

opendistro\_security.audit.type: internal\_elasticsearch

opendistro\_security.enable\_snapshot\_restore\_privilege: true

opendistro\_security.check\_snapshot\_restore\_write\_privileges: true

opendistro\_security.restapi.roles\_enabled: ["all\_access", "security\_rest\_api\_access"]

### Hot node configuration

Add the below configuration to elasticsearch.yml file

transport.tcp.port: 9300

xpack.ml.enabled: false

cluster.name: “neustar-eopsadmin-logging”

node.name: eopsadmin-datanode-1

node.master: false

node.data: true

node.ingest: true

network.host: awopuse1elasticdatahotnode01.laas.neustar.biz

http.port: 9200

discovery.seed\_hosts: ["awopuse1elasticmasternode01.laas.neustar.biz", "awopuse1elasticcoordnode01.laas.neustar.biz", "awopuse1elasticdatahotnode01.laas.neustar.biz", "awopuse1elasticdatahotnode02.laas.neustar.biz", "awopuse1elasticdatawarmcoldnode01.laas.neustar.biz", "awopuse1elasticdatawarmcoldnode02.laas.neustar.biz"]

path.data: /opt/eopsadmin/elasticsearch/data

path.logs: /opt/eopsadmin/elasticsearch/logs

node.attr.box\_type: hot

cluster.initial\_master\_nodes: ["eopsadmin-masternode-1"]

bootstrap.system\_call\_filter: false

node.max\_local\_storage\_nodes: 10

xpack.security.enabled: false

xpack.monitoring.enabled: true

opendistro\_security.ssl.transport.pemcert\_filepath: certificate/ns\_cert/node/nsdev-cert-signed.pem

opendistro\_security.ssl.transport.pemkey\_filepath: certificate/ns\_cert/node//laas.neustar.biz.key

opendistro\_security.ssl.transport.pemtrustedcas\_filepath: certificate/ns\_cert/node//nsdev\_chain.pem

opendistro\_security.ssl.transport.enforce\_hostname\_verification: false

opendistro\_security.ssl.http.enabled: true

opendistro\_security.ssl.http.pemcert\_filepath: certificate/ns\_cert/node/nsdev-cert-signed.pem

opendistro\_security.ssl.http.pemkey\_filepath: certificate/ns\_cert/node/laas.neustar.biz.key

opendistro\_security.ssl.http.pemtrustedcas\_filepath: certificate/ns\_cert/node/nsdev\_chain.pem

opendistro\_security.allow\_default\_init\_securityindex: true

opendistro\_security.authcz.admin\_dn:

- 'CN=eodsecadmin.neustar.biz'

opendistro\_security.nodes\_dn:

- 'CN=\*.laas.neustar.biz'

opendistro\_security.audit.type: internal\_elasticsearch

opendistro\_security.enable\_snapshot\_restore\_privilege: true

opendistro\_security.check\_snapshot\_restore\_write\_privileges: true

opendistro\_security.restapi.roles\_enabled: ["all\_access", "security\_rest\_api\_access"]

### Warm Node configuration

Add the below configuration to elasticsearch.yml file

transport.tcp.port: 9300

xpack.ml.enabled: false

cluster.name: “neustar-eopsadmin-logging”

node.name: eopsadmin-datanode-3

node.master: false

node.data: true

node.ingest: true

network.host: awopuse1elasticdatawarmcoldnode01.laas.neustar.biz

http.port: 9200

discovery.seed\_hosts: ["awopuse1elasticmasternode01.laas.neustar.biz", "awopuse1elasticcoordnode01.laas.neustar.biz", "awopuse1elasticdatahotnode01.laas.neustar.biz", "awopuse1elasticdatahotnode02.laas.neustar.biz", "awopuse1elasticdatawarmcoldnode01.laas.neustar.biz", "awopuse1elasticdatawarmcoldnode02.laas.neustar.biz"]

path.data: /opt/eopsadmin/elasticsearch/data

path.logs: /opt/eopsadmin/elasticsearch/logs

node.attr.box\_type: warm

cluster.initial\_master\_nodes: ["eopsadmin-masternode-1"]

bootstrap.system\_call\_filter: false

node.max\_local\_storage\_nodes: 10

xpack.security.enabled: false

xpack.monitoring.enabled: true

opendistro\_security.ssl.transport.pemcert\_filepath: certificate/ns\_cert/node/nsdev-cert-signed.pem

opendistro\_security.ssl.transport.pemkey\_filepath: certificate/ns\_cert/node//laas.neustar.biz.key

opendistro\_security.ssl.transport.pemtrustedcas\_filepath: certificate/ns\_cert/node//nsdev\_chain.pem

opendistro\_security.ssl.transport.enforce\_hostname\_verification: false

opendistro\_security.ssl.http.enabled: true

opendistro\_security.ssl.http.pemcert\_filepath: certificate/ns\_cert/node/nsdev-cert-signed.pem

opendistro\_security.ssl.http.pemkey\_filepath: certificate/ns\_cert/node/laas.neustar.biz.key

opendistro\_security.ssl.http.pemtrustedcas\_filepath: certificate/ns\_cert/node/nsdev\_chain.pem

opendistro\_security.allow\_default\_init\_securityindex: true

opendistro\_security.authcz.admin\_dn:

- 'CN=eodsecadmin.neustar.biz'

opendistro\_security.nodes\_dn:

- 'CN=\*.laas.neustar.biz'

opendistro\_security.audit.type: internal\_elasticsearch

opendistro\_security.enable\_snapshot\_restore\_privilege: true

opendistro\_security.check\_snapshot\_restore\_write\_privileges: true

opendistro\_security.restapi.roles\_enabled: ["all\_access", "security\_rest\_api\_access"]

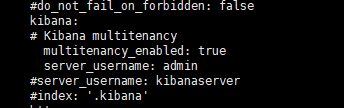
### Update Open Distro config

Update the below config changes in all the elastic nodes

Navigate to /opt/eopsadmin/elasticsearch/plugins/opendistro\_security/securityconfig

edit config.yml

uncomment kibana multitenancy and update username to admin



## Heap configuration and Initilize elasticsearch nodes

### Heap Configuration

Set Xmx and Xms to no more than 50% of your physical RAM. Elasticsearch requires memory for purposes other than the JVM heap and it is important to leave space for this. For instance, Elasticsearch uses off-heap buffers for efficient network communication, relies on the operating system’s filesystem cache for efficient access to files, and the JVM itself requires some memory too. It is normal to observe the Elasticsearch process using more memory than the limit configured with the Xmx setting.

Ideally set Xmx and Xms to no more than the threshold for zero-based compressed oops; the exact threshold varies but 26 GB is safe on most systems, but can be as large as 30 GB on some systems.

[eopsadmin@ip-10-75-240-52 centos]$ pwd

/opt/eopsadmin/elasticsearch/config

[eopsadmin@ip-10-75-240-52 centos]$ vi jvm.options

**-Xms7g**

**-Xmx7g**

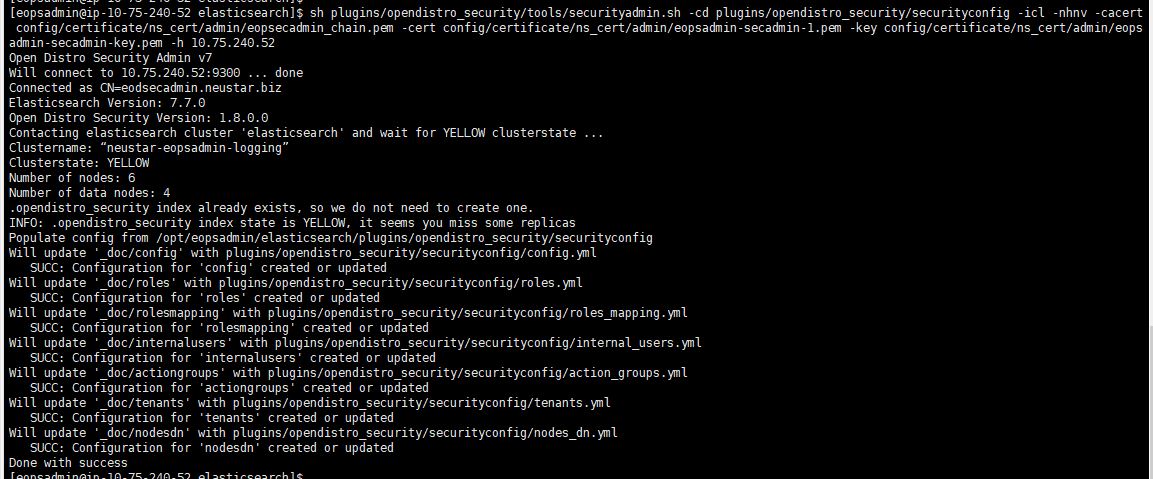
1. Configure the above as per your system resources available on the server. Repeat the steps the above across all master eligible nodes (elastic node servers)

### Initialize Open Distro

Run the securityadmin.sh script **in master node only** to update the cluster configuration and to Initialize the Open Distro

It will show you the cluster status and nodes. Once it shows success status check cluster health.

sh plugins/opendistro\_security/tools/securityadmin.sh -cd plugins/opendistro\_security/securityconfig -icl -nhnv -cacert config/certificate/ns\_cert/admin/eopsecadmin\_chain.pem -cert config/certificate/ns\_cert/admin/eopsadmin-secadmin-1.pem -key config/certificate/ns\_cert/admin/eopsadmin-secadmin-key.pem -h 10.75.240.52



### Start Elasticsearch

Start the elasticsearch using the startup script

**sudo systemctl start elasticsearch.service**

1. Start the elasticsearch on all the servers under the cluster using the startup utility.
2. Ensuring cluster is started successfully:

From any of the servers, using web browser access the url of any of the elastic nodes and you should the output as below.

<http://10.75.240.52:9200/_cluster/health?pretty>

[eopsadmin@ip-10-75-240-52 logs]$ curl -XGET "http://10.75.240.52:9200/\_cluster/health?pretty"

{

"cluster\_name" : "“neustar-eopsadmin-logging”",

"status" : "green",

"timed\_out" : false,

"number\_of\_nodes" : 6,

"number\_of\_data\_nodes" : 4,

"active\_primary\_shards" : 1,

"active\_shards" : 2,

"relocating\_shards" : 0,

"initializing\_shards" : 0,

"unassigned\_shards" : 0,

"delayed\_unassigned\_shards" : 0,

"number\_of\_pending\_tasks" : 0,

"number\_of\_in\_flight\_fetch" : 0,

"task\_max\_waiting\_in\_queue\_millis" : 0,

"active\_shards\_percent\_as\_number" : 100.0

}

**curl -XGET "http://10.75.240.52:9200/\_cat/nodes?v"**

|  |
| --- |
| ip heap.percent ram.percent cpu load\_1m load\_5m load\_15m node.role master name  10.75.242.55 14 15 0 0.00 0.01 0.05 dimrt - eopsadmin-datanode-2  10.75.240.57 33 15 0 0.00 0.01 0.05 dimrt - eopsadmin-datanode-1  10.75.240.58 12 15 0 0.00 0.01 0.05 dimrt - eopsadmin-datanode-3  10.75.242.56 12 15 0 0.00 0.01 0.05 dimrt - eopsadmin-datanode-4  10.75.242.51 28 29 0 0.01 0.03 0.05 r - eopsadmin-coordnode-1  10.75.240.52 20 24 0 0.00 0.01 0.05 mr \* eopsadmin-masternode-1 |

If you are seeing different response than above, look for any error message or in the log file on the server /opt/eopsadmin/elasticsearch/logs/“neustar-eopsadmin-logging”.log

## Logstash

Download the software package and copy to desired location on the server

curl -L -O <https://artifacts.elastic.co/downloads/logstash/logstash-7.9.0.tar.gz>

### Installation Steps

sudo su eopsadmin

cd /opt/softwares

cp logstash-7.9.0.tar.gz /opt/eopsadmin/

cd /opt/eopsadmin/

tar -zxvf logstash-7.9.0.tar.gz

mv logstash-7.9.0/ logstash

The installation and configuration files are available under the directory /opt/eopsadmin/logstash/config/

### Configuration Steps

1. cd /opt/eopsadmin/logstash/config/
2. Create a config file as per your requirement and start the logstash service as shown below

### Logstash Open Distro Configuration

1. We need to enable the Opendistro configuration on the Logstash server and to connect to elasticsearch cluster using secured connection.
2. Open the logstash.yml file and add the below lines:

[eopsadmin@ip-10-75-240-53 conf.d]$ cat /opt/eopsadmin/logstash/config/logstash.yml | grep -v "#" | grep -v '^$'

pipeline.ordered: auto

monitoring.enabled: false

monitoring.cluster\_uuid: vCmvklZbQ6WRwqJ9nVErfQ

1. Open the consumer logstash.conf output section file and add the below lines for each output section.

output {

elasticsearch {

hosts => ["awopuse1elasticdatahotnode01.laas.neustar.biz:9200", "awopuse1elasticdatahotnode02.laas.neustar.biz:9200"]

index => "secops-awsgd-lstash-"

user => "elastic"

password => "elastic"

cacert => '/opt/eopsadmin/logstash/config/certificate/nsdev\_chain.pem

ssl => true

ssl\_certificate\_verification => true

}

stdout { codec => rubydebug }

}

1. Repeat the above steps across all the Consumer Logstash servers.
2. Start Logstash on all the servers using startup script and ensure that it is able to connect to elasticsearch cluster.
3. Check the log file for any errors. If data is successful injected into elasticsearch, then the logstash is configured with x-pack feature.

## Kibana

Download the software package and copy to desired location on the server

### Installation Steps

curl -L -O [https://artifacts.elastic.co/downloads/kibana/kibana-7.9.0-linux-x86\_64.tar.gz](https://artifacts.elastic.co/downloads/kibana/kibana-7.7.0-linux-x86_64.tar.gz)

cp kibana-7.9.0-linux-x86\_64.tar.gz /opt/eopsadmin/

cd /opt/eopsadmin/

tar -zxvf kibana-7.9.0-linux-x86\_64.tar.gz

mv kibana-7.9.0-linux-x86\_64/ kibana

sudo chown -R eopsadmin:eopsadmin kibana/

### Configuration Steps

**Update the kibana.yml configuration file as shown and set the values listed below:**

[eopsadmin@ip-10-75-240-51 config]$ cd /opt/eopsadmin/kibana/config

[eopsadmin@ip-10-75-240-51 config]$ vi kibana.yml

Add the following configurations to kibana.yml file

server.port: 5601

server.host: "awopuse1kibana01.laas.neustar.biz"

elasticsearch.hosts: ["https://awopuse1elasticcoordnode01.laas.neustar.biz:9200"]

elasticsearch.username: "admin"

elasticsearch.password: "admin"

server.ssl.enabled: true

server.ssl.certificate: /opt/eopsadmin/kibana/config/certificate/ns\_cert/node/nsdev-cert-signed.pem

server.ssl.key: /opt/eopsadmin/kibana/config/certificate/ns\_cert/node/laas.neustar.biz.key

elasticsearch.ssl.certificateAuthorities: [ "/opt/eopsadmin/kibana/config/certificate/ns\_cert/node/nsdev\_chain.pem" ]

elasticsearch.ssl.verificationMode: certificate

xpack.security.enabled: false

opendistro\_security.multitenancy.enabled: true

opendistro\_security.multitenancy.tenants.enable\_global: false

opendistro\_security.multitenancy.tenants.enable\_private: false

elasticsearch.requestHeadersWhitelist: ["securitytenant","Authorization"]

opendistro\_security.multitenancy.enable\_filter: false

Run the kibana instance using the kibana startup utility.

sudo systemctl start kibana.service

Ensure that the kibana is started successfully and able to connect to elasticsearch cluster.

The Kibana dashboards can be accessed using the secured url

<https://awopuse1kibana01.laas.neustar.biz:5601/>

## Beats

### FileBeat

1. Download the S/W package as shown below and untar

curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.9.1-linux-x86\_64.tar.gz

tar xzvf filebeat-7.9.1-linux-x86\_64.tar.gz

**Installation steps**

cd /opt/elasticsearch/filebeat/

Rename the directory **filebeat-<version>** to **filebeat**.

cd /opt/elasticsearch/filebeat/

Ensure the directory permissions are owned by ‘root’ and has write and execute permissions as needed. Below is the directory structure and file permissions look like after extracted.

ls -lrt /opt/elasticsearch/filebeat/

|  |
| --- |
| -rw-------. 1 root root 706219 Jul 20 09:39 fields.yml  drwx------. 3 root root 15 Jul 20 09:39 kibana  -rw-------. 1 root root 100619 Jul 20 09:39 filebeat.reference.yml  -rwx------. 1 root root 89226256 Jul 20 09:39 filebeat  -rw-------. 1 root root 13675 Jul 20 09:39 LICENSE.txt  drwx------. 44 root root 4096 Jul 20 09:39 module  drwx------. 2 root root 4096 Jul 20 09:39 modules.d  -rw-------. 1 root root 802 Jul 20 039 README.md  -rw-------. 1 root root 429708 Jul 20 09:39 NOTICE.txt  -rw-----. 1 root root 8448 Jul 29 12:28 filebeat.yml  drwx------. 2 root root 130 Jul 29 12:30 logs  drwxr-x---. 3 root root 60 Jul 29 12:30 data |

**Configure Filebeat:**

Below are the various sections of configuration values to be set within filebeat.yml file to ensure the settings are read by filebeat process.

1. Path settings:

|  |
| --- |
| path.home: /opt/elasticsearch/filebeat  path.config: /opt/elasticsearch/filebeat  path.data: /opt/elasticsearch/filebeat/data  path.logs: /opt/elasticsearch/filebeat/logs |

1. Input settings:

|  |
| --- |
| #============ Filebeat inputs ====================  - type: log  enabled: true  paths:  - /var/log/anaconda/\*  - /var/log/audit/\*  - /var/log/chef/\*  - /var/log/sshd/\*  - /var/log/tuned/\*  - /var/log/iptables.log  - /var/log/cron  - /var/log/kern.log  - /var/log/maillog  - /var/log/messages  - /var/log/secure  - /var/log/user.log  - /var/log/yum.log  exclude\_files: ['\.gz$']  scan\_frequency: 1s  ignore\_older: 24h  close\_inactive: 2h  filebeat.config.modules:  path: ${path.config}/modules.d/\*.yml  reload.enabled: false  setup.template.settings:  index.number\_of\_shards: 1  setup.kibana: |

1. Output settings:

|  |
| --- |
| # Boolean flag to enable or disable the output module.  enabled: true  # The Logstash hosts  #hosts: ["<logstash-destination>:<port>"]  output.logstash:  hosts: ["10.31.40.176:5525 "]  #hosts: ["10.31.45.34:5525","10.31.45.41:5525","10.56.2.59:5525","10.56.2.75:5525"]  loadbalance: true |

1. Drop/tag add:

|  |
| --- |
| processors:  - drop\_fields:  fields: ["host.name"]  ignore\_missing: false  - add\_tags:  tags: Linux\_filebeat\_agent |

**Start and stop the filebeat:**

1. Run the below command to start the filebeat.

**cd /opt/elasticsearch/filebeat**

**./filebeat -c filebeat.yml -e**

1. Filebeat can also be run as service, below are the steps required to run filebeat as a service

As root user, create the file **/etc/systemd/system/filebeat.service** with the following content

|  |
| --- |
| [Unit]  Description=filebeat  After=network.target  [Service]  ExecStart=/opt/elasticsearch/filebeat/filebeat -c /opt/elasticsearch/filebeat/filebeat.yml  KillMode=process  User=axway  WorkingDirectory=/  Restart=always  RestartSec=30s  [Install]  WantedBy=multi-user.target |

1. Next, execute the following commands:

|  |
| --- |
| chmod 644 /etc/systemd/system/filebeat.service  systemctl daemon-reload  systemctl enable filebeat.service |

1. To Start filebeat as a service run the below command

systemctl start filebeat.start

1. To stop the filebeat as a service run the below command

systemctl stop filebeat.start

1. Check logs for any errors
2. **Below is the filebeat.yml file**

**Note:** The important changes to the above file are the input logfile paths, multiline patterns, output logstash servers.

### Metric Beat

* 1. Download the S/W package as shown below,

curl -L -O <https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-7.7.0-linux-x86_64.tar.gz>

**Installation steps**

[eopsadmin@ip-10-75-240-52 ]$ cd /opt/softwares

[eopsadmin@ip-10-75-240-52 softwares]$ tar -zxf metricbeat-7.9.0-linux-x86\_64.tar.gz --directory /opt/eopsadmin/

[eopsadmin@ip-10-75-240-52 softwares]$ cd /opt/eopsadmin/

[eopsadmin@ip-10-75-240-52 eopsadmin]$ mv metricbeat-7.9.0-linux-x86\_64/ metricbeat/

[eopsadmin@ip-10-75-240-52 ]$ cd /opt/eopsadmin/metricbeat

[eopsadmin@ip-10-75-240-52 metricbeat]$ vi metricbeat.yml

[eopsadmin@ip-10-75-240-52 metricbeat]$ cat metricbeat.yml | grep -v "#" | grep -v '^$'

|  |
| --- |
| **vi /opt/eopsadmin/metricbeat/metricbeat.yml**  metricbeat.config.modules:  path: ${path.config}/modules.d/\*.yml  reload.enabled: false  setup.template.settings:  index.number\_of\_shards: 1  index.codec: best\_compression  setup.template:  name: "metricbeat"  pattern: "metricbeat-\*"  setup.kibana:  output.elasticsearch:  hosts: ["https://awopuse1elasticdatahotnode01.laas.neustar.biz:9200", "https://awopuse1elasticdatahotnode02.laas.neustar.biz:9200"]  index: "metricbeat-"  username: "elastic"  password: "elastic"  ssl.certificate\_authorities: "/opt/eopsadmin/kibana/config/certificate/nsdev-cert-signed.crt"  ssl.certificate: "/opt/eopsadmin/kibana/config/certificate/nsdev-cert-signed.crt"  ssl.key: "/opt/eopsadmin/kibana/config/certificate/laas.neustar.biz.key"  processors:  - add\_host\_metadata: ~  - add\_cloud\_metadata: ~  - add\_docker\_metadata: ~  - add\_kubernetes\_metadata: ~ |

**Enable Modules required:**

1. **Elasticsearch Module:** [eopsadmin@ip-10-75-240-52 metricbeat]$ ./metricbeat modules enable elasticsearch

Enabled elasticsearch

[eopsadmin@ip-10-75-240-52 metricbeat]$ vi modules.d/elasticsearch.yml

[eopsadmin@ip-10-75-240-52 metricbeat]$

|  |
| --- |
| **vi /opt/eopsadmin/metricbeat/modules.d/elasticsearch.yml**  - module: elasticsearch  metricsets:  - node  - node\_stats  period: 10s  hosts: ["https://10.75.240.52:9200"]  username: "elastic"  password: "elastic"  ssl.certificate\_authorities: '/opt/eopsadmin/elasticsearch/config/certificate/nsdev-cert-signed.crt'  ssl\_certificate\_verification: true |

1. **kibana Module:**

|  |
| --- |
| **vi /opt/eopsadmin/metricbeat/modules.d/kibana.yml**  - module: kibana  metricsets:  - status  period: 10s  hosts: ["https://awopuse1kibana01.laas.neustar.biz:5601"]  basepath: ""  enabled: true  username: "elastic"  password: "elastic"  ssl.certificate\_authorities: '/opt/eopsadmin/kibana/config/certificate/nsdev-cert-signed.crt'  ssl\_certificate\_verification: true |

1. **Logstash Module:**

**/opt/eopsadmin/metricbeat/modules.d/logstash-xpack.yml**

* **Logstash Module (For Metricbeat version 7.7.0)**

|  |
| --- |
| - module: logstash  metricsets:  - node  - node\_stats  period: 10s  hosts: ["10.31.45.57:9600"]  xpack.enabled: true |

* **Logstash Module (For Metricbeat version 7.9.0)**

|  |
| --- |
| - module: logstash  xpack.enabled: true  period: 10s  hosts: ["10.31.45.57:9600"] |

1. **Kafka Module:**

|  |
| --- |
| **vi /opt/eopsadmin/metricbeat/modules.d/kafka.yml**  - module: kafka  metricsets:  period: 10s  hosts: ["10.75.240.55:9093"]  topics: []  - module: kafka  metricsets:  - broker  period: 10s  hosts: ["0.0.0.0:8778"]  - module: kafka  metricsets:  - consumer  period: 10s  hosts: ["0.0.0.0:8778"]  - module: kafka  metricsets:  - producer  period: 10s  hosts: ["0.0.0.0:8778"] |

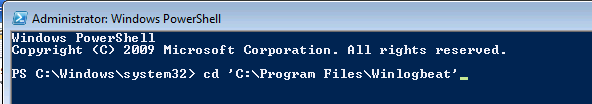
1. **Jolokia Module:**

|  |
| --- |
| **vi /opt/eopsadmin/metricbeat/modules.d/jolokia.yml**  - module: jolokia  metricsets: ["jmx"]  period: 10s  hosts: ["0.0.0.0:8778/jolokia/"]  namespace: "metrics"  jmx.mappings:  - mbean: 'java.lang:type=Runtime'  attributes:  - attr: Uptime  field: uptime  - mbean: 'java.lang:type=Memory'  attributes:  - attr: HeapMemoryUsage  field: memory.heap\_usage  - attr: NonHeapMemoryUsage  field: memory.non\_heap\_usage  - mbean: 'java.lang:type=GarbageCollector,name=ConcurrentMarkSweep'  attributes:  - attr: CollectionTime  field: gc.cms\_collection\_time  - attr: CollectionCount  field: gc.cms\_collection\_count  jmx.application:  jmx.instance: |

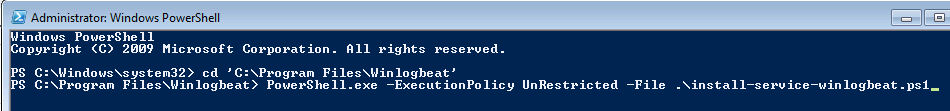
### Winlog Beat

1. Login into the Windows Server
2. Download the S/W package from the below link <https://www.elastic.co/downloads/beats/winlogbeat>
3. Extract the contents into C:\Program Files.
4. Rename the winlogbeat-<version> directory to Winlogbeat.
5. Open a PowerShell prompt as an Administrator.
6. From the PowerShell prompt, run the following commands to install the service.

cd 'C:\Program Files\Winlogbeat'



1. PowerShell.exe -ExecutionPolicy UnRestricted -File .\install-service-winlogbeat.ps1

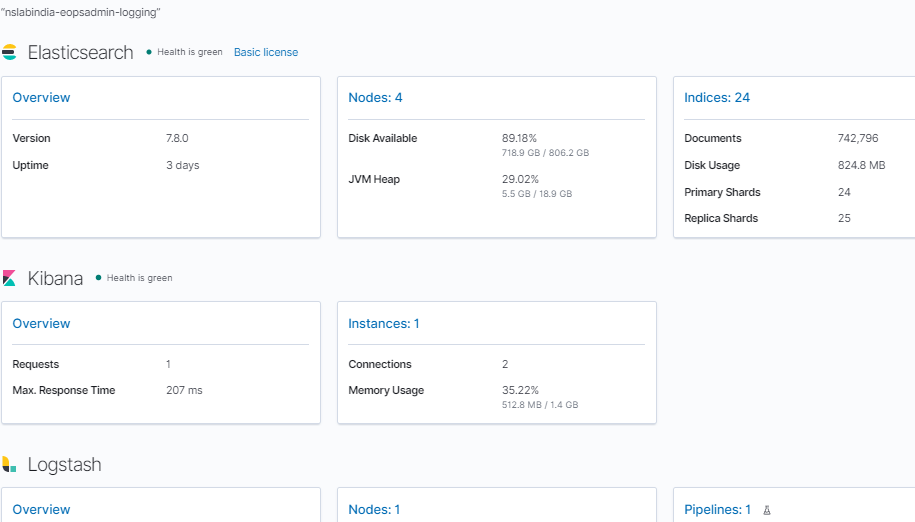


1. Now open winlogbeat.yml file with a text editor, which is located at C:\Program Files\Winlogbeat. And configure the following options.
2. Now run the following command in PowerShell to start and stop the winlogbeat.  
   Start-Service winlogbeat.  
   Stop-Service winlogbeat.
3. Check the log files for any errors.
4. **Please find the wlb yml file**

# Stack Monitoring

use the xpack.monitoring.enabled setting in elasticsearch.yml file

Elasticsearch nodes and Kibana will be shown on X-Pack monitoring directly and metric beat is supposed to be configured to monitor Logstash.



**Metric Beat configurations to monitor Logstash:**

Edit metricbeat.yml file as below and enable logstash-xpack.yml in modules.d

[eopsadmin@elk-data metricbeat]$ cat metricbeat.yml | grep -v "#"

metricbeat.config.modules:

path: ${path.config}/modules.d/\*.yml

reload.enabled: false

setup.template.settings:

index.number\_of\_shards: 1

index.codec: best\_compression

setup.template:

name: "metricbeat"

pattern: "metricbeat-\*"

setup.kibana:

host: "https://sgpl-kibana.laas.neustar.biz:5601"

ssl.enabled: true

username: "admin"

password: "admin"

ssl.certificate\_authorities: ["/opt/eopsadmin/7\_8\_elk/elasticsearch-7.8.0/config/neustar\_certs/nsdev-cert-signed.pem"]

ssl.certificate: "/opt/eopsadmin/7\_8\_elk/elasticsearch-7.8.0/config/neustar\_certs/nsdev-cert-signed.pem"

ssl.key: "/opt/eopsadmin/7\_8\_elk/elasticsearch-7.8.0/config/neustar\_certs/laas.neustar.biz.key"

output.elasticsearch:

hosts: ["https://hotnode-es.laas.neustar.biz:9200"]

index: "metricbeat-"

username: "admin"

password: "admin"

ssl.certificate\_authorities: ["/opt/eopsadmin/7\_8\_elk/elasticsearch-7.8.0/config/neustar\_certs/nsdev-cert-signed.pem"]

ssl.certificate: "/opt/eopsadmin/7\_8\_elk/elasticsearch-7.8.0/config/neustar\_certs/nsdev-cert-signed.pem"

ssl.key: "/opt/eopsadmin/7\_8\_elk/elasticsearch-7.8.0/config/neustar\_certs/laas.neustar.biz.key"

logging.level: debug

logging.to\_files: true

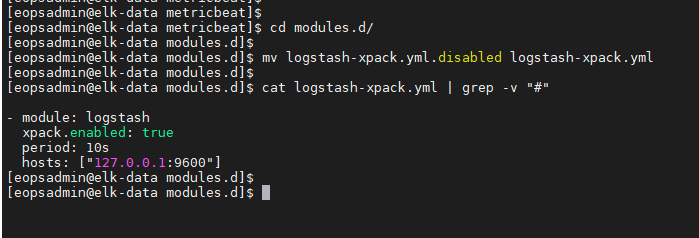
logging.files:

path: /opt/eopsadmin/7\_8\_elk/metricbeat/logs

name: metricbeat

keepfiles: 7

permissions: 0644



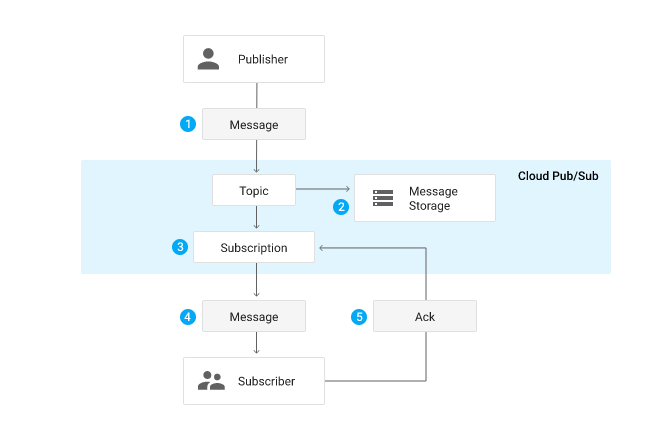
# Logstash GCP Plugins

## Cloud Storage

## Pub/Sub

Pub/Sub is a fully-managed real-time messaging service that allows you to send and receive messages between independent applications

**Message flow**



* A publisher application creates a topic in the Pub/Sub service and sends messages to the topic. A message contains a payload and optional attributes that describe the payload content.
* The service ensures that published messages are retained on behalf of subscriptions. A published message is retained for a subscription until it is acknowledged by any subscriber consuming messages from that subscription.
* Pub/Sub forwards messages from a topic to all of its subscriptions, individually.
* A subscriber receives messages either by Pub/Sub pushing them to the subscriber's chosen endpoint, or by the subscriber pulling them from the service.
* The subscriber sends an acknowledgement to the Pub/Sub service for each received message.
* The service removes acknowledged messages from the subscription's message queue.

**Throughput** based on the Regions

Regions are divided into 2 types:

Large: europe-west1, us-central1, us-east1

Small: All other regions

|  |  |  |
| --- | --- | --- |
| Throughput | Region | Mb/s |
| Producer /Publisher | Large | 200 |
| Producer /Publisher | Small | 50 |
| Pull /subscriber | Large | 400 |
| Pull /subscriber | Small | 100 |
| Push/subscriber | Large | 20 |
| Push /subscriber | Small | 5 |

**Connections:**

Number of open Streaming Pull connections per region

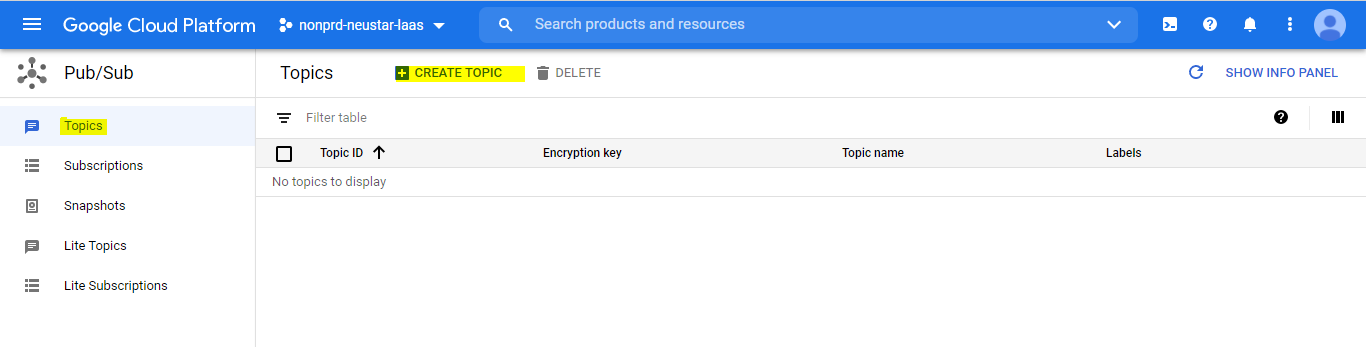
* 24,000 open connections at a time in large regions
* 6,000 open connections at a time in small regions

**Pricing:**

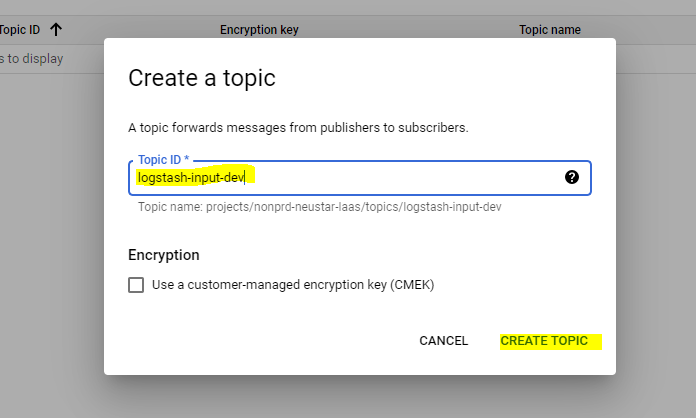
Message ingestion and delivery are priced per volume of data transmitted in a calendar month. The first 10 gigabytes of usage are free. After that, the price for ingestion or delivery of messages is $40 per TiB.

### Create Topic

1. Login to GCP console and choose **Pub/Sub** from the left hand navigation menu. Click on **Topics** and then click on **CREATE TOPIC**.

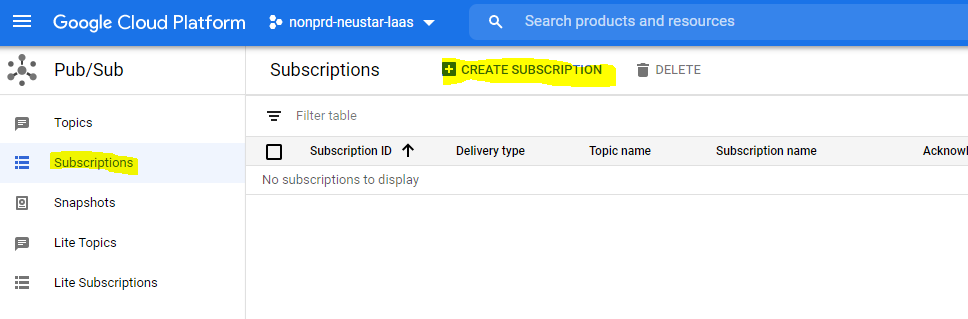


1. Enter Topic Name. Topic Name “**logstash-input-dev”** and click on **CREATE TOPIC.**

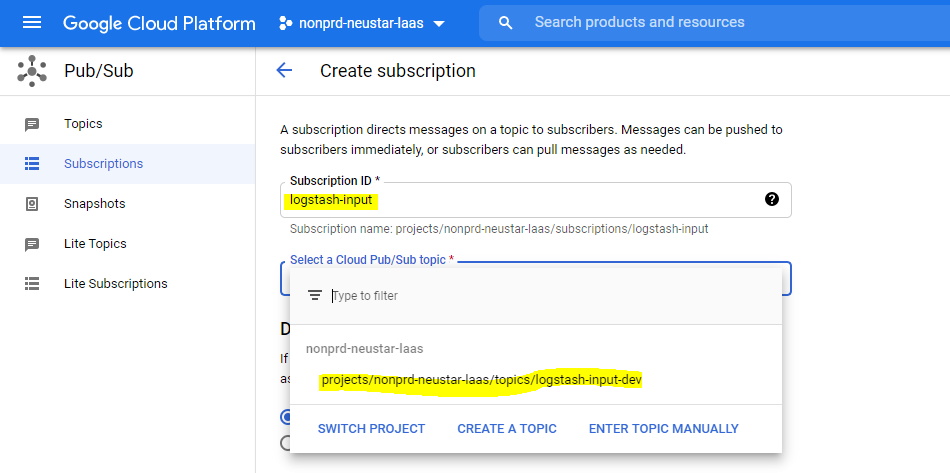
****

### Create Subscription

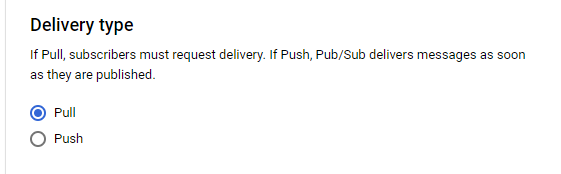
1. After that go to **Subscriptions** from left side navigation. Click on Subscriptions and then click on **CREATE SUBSCRIPTION.**

****

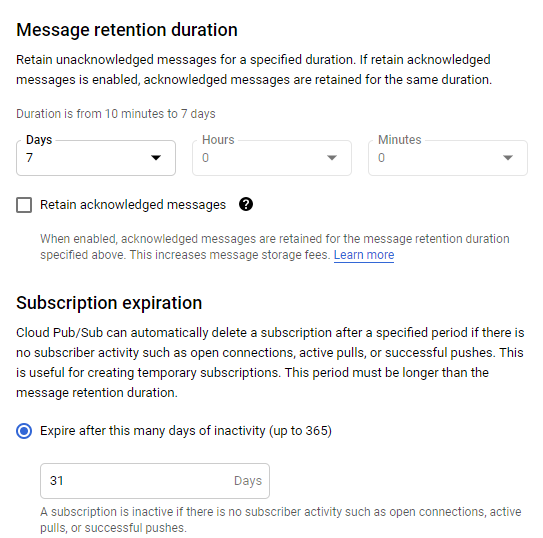
1. Enter Subscription Name and Select previously created Topic(**logstash-input-dev**).

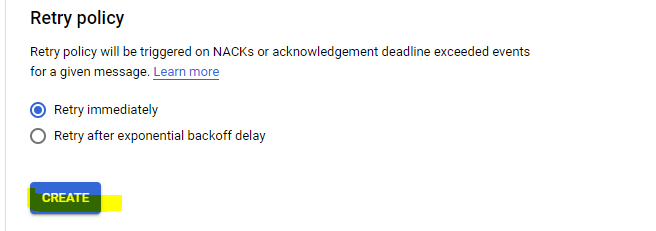


1. Make sure that Delivery type is “Pull”.



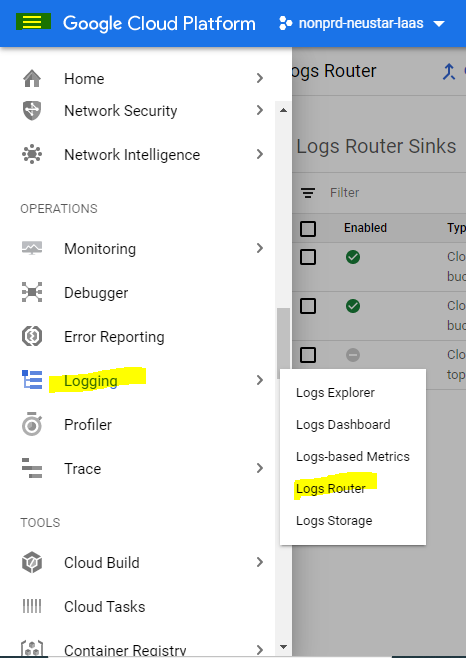
1. Change Message retention duration, Subscription Expiration and retry policy as per our project requirement and click on create.

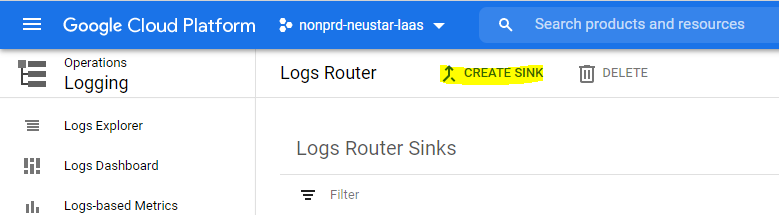


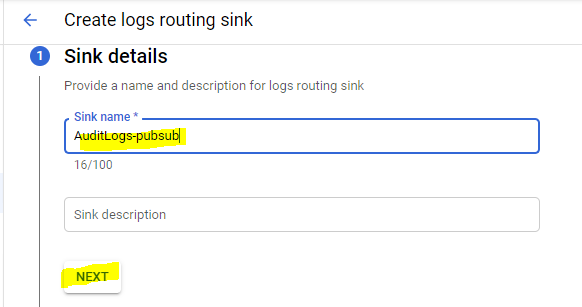


**Route Stack Driver Logs to Cloud Pub/Sub**

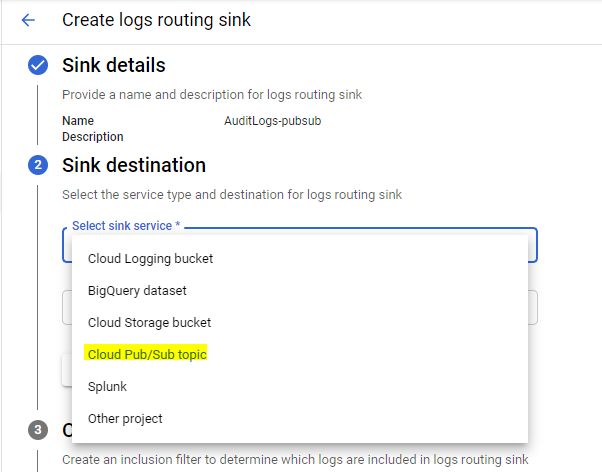
Menu Bar > Logging > Logs Router

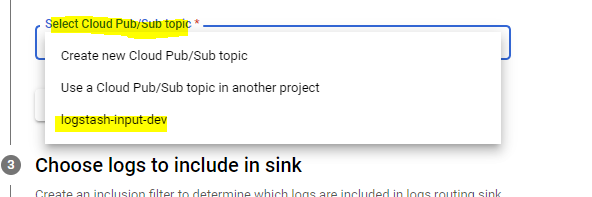


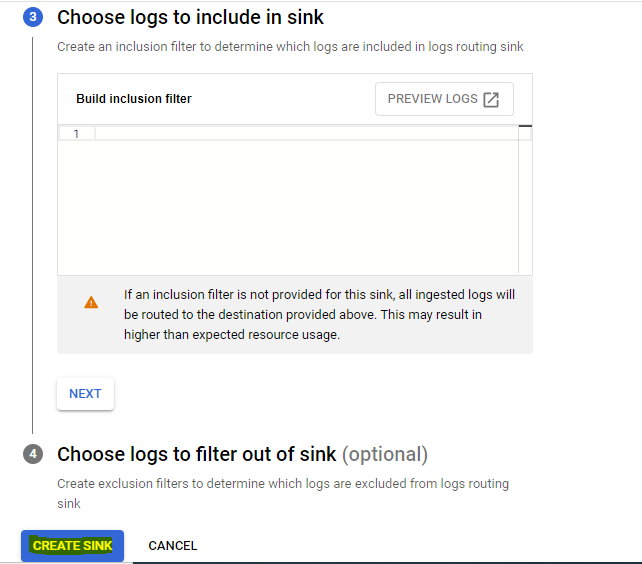
Then create sink, creating sink is send data to destination cloud pub/sub topic.

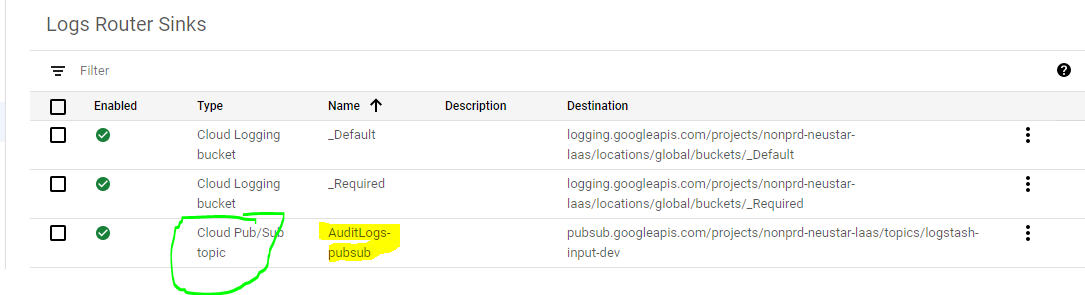


Select Sink service as Cloud Pub/Sub Topic and create sink









Logstash input plugin configuration for Cloud Pub/Sub

|  |
| --- |
| input {  google\_pubsub {  # GCP project id (name)  project\_id => "nonprd-neustar-laas"  # The topic name below is currently hard-coded in the plugin.  # must first create this topic by hand and ensure you are exporting  # logging to this pubsub topic.  topic => "logstash-input-dev"  # The subscription name is customizeable. The plugin will attempt to  # create the subscription (but use the hard-coded topic name above).  subscription => "logstash-input"  #max\_messages => "10  #json\_key\_file => "/home/erjohnso/pkey.json"  #create\_subscription => false  }  }  output { elasticsearch {  hosts => ["https://10.128.0.9:9200"]  ssl => true  ssl\_certificate\_verification => false  user => "admin"  password => "admin"  index => "pubsub-auditlogs-"  }  stdout { codec => rubydebug } } |

Log stash additional parameters for tuning:

message\_count\_threshold => 1000

delay\_threshold\_secs => 10

request\_byte\_threshold => 5000000

**delay\_threshold\_secsedit**

Send the batch once this delay has passed, from the time the first message is queued. Must be greater than 0.

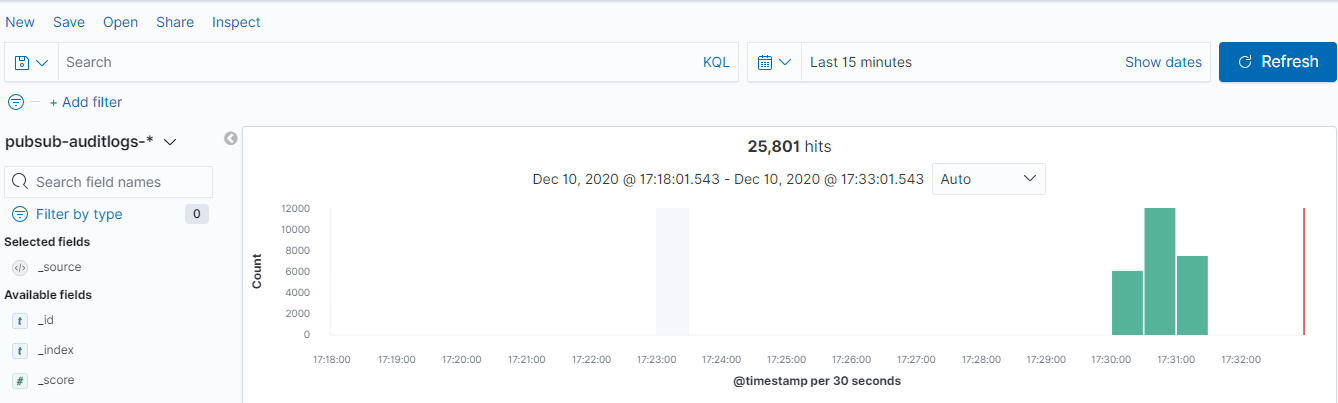
**message\_count\_thresholdedit**

Once this many messages are queued, send all the messages in a single call, even if the delay threshold hasn’t elapsed yet. Must be < 1000. A value of 0 will cause messages to instantly be sent but will reduce total throughput due to overhead.

**request\_byte\_thresholdedit**

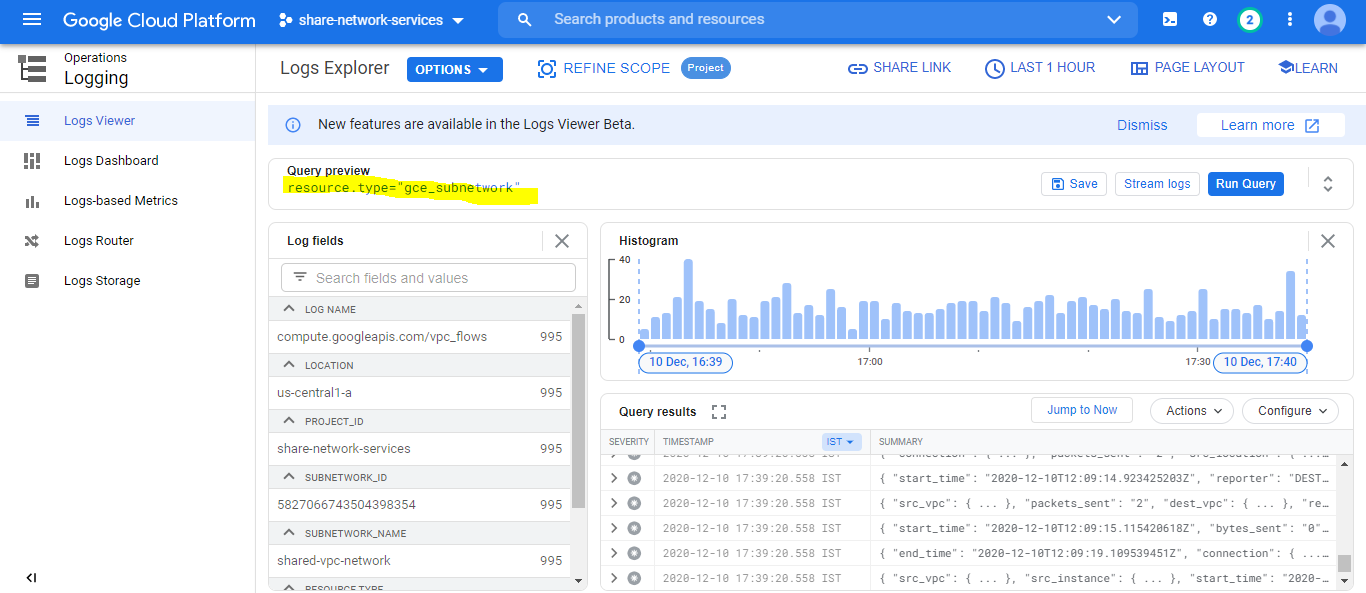
Once the number of bytes in the batched request reaches this threshold, send all of the messages in a single call, even if neither the delay or message count thresholds have been exceeded yet. This includes full message payload size, including any attributes set.

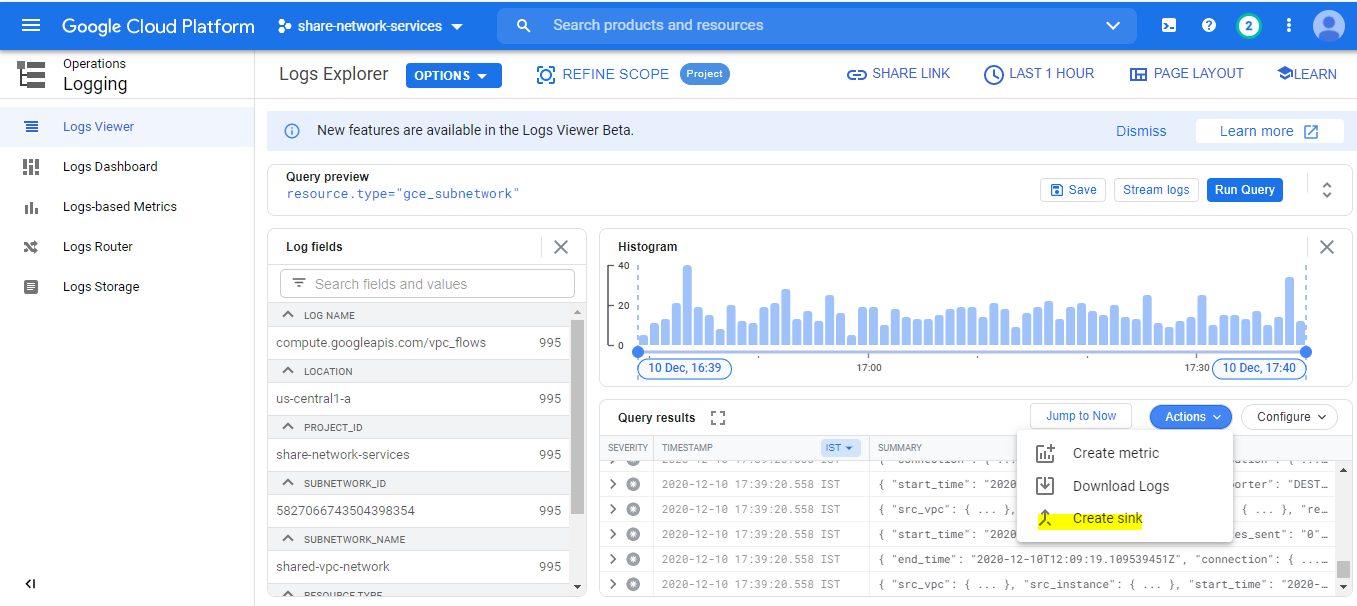






To send specific metric to pub/sub need to use the log viewer option





# Memcached Installtion and Configuration

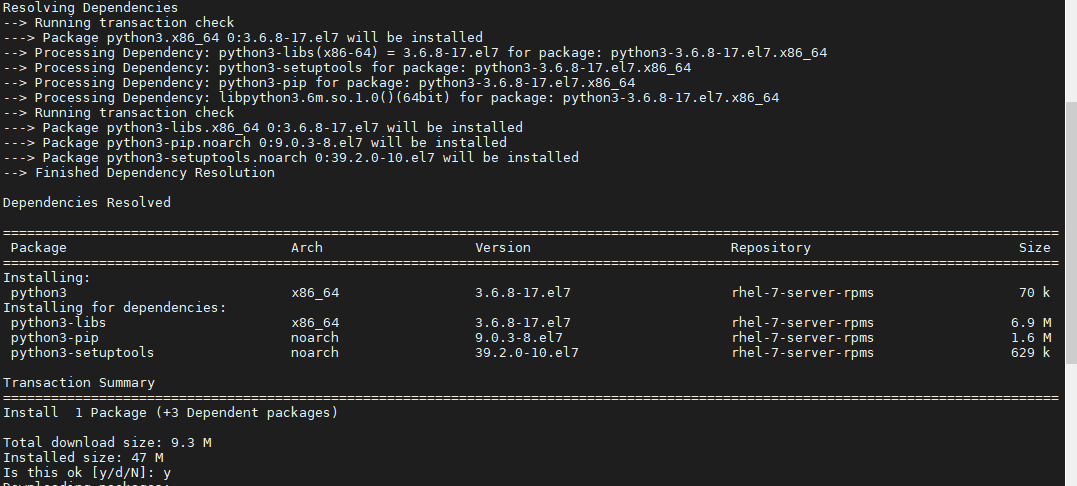
## Prerequisites:

Memcached Requires python3 environment.

Memcached server will be using port 11211 check inbound and outbound permissions.

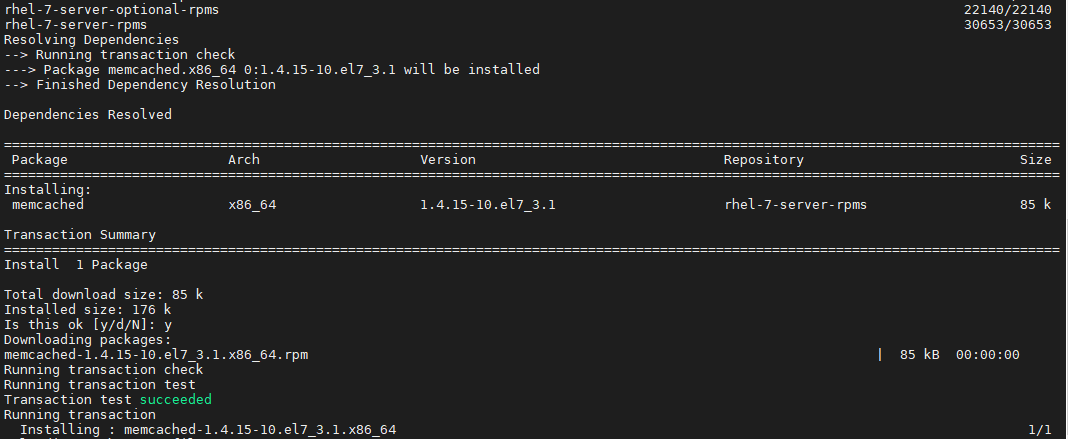
**Python3 Installation:**

[root@stopvprvlogstash01 ~]# yum install python3



## Installation:

[root@stopvprvlogstash01 ~]# yum install memcached

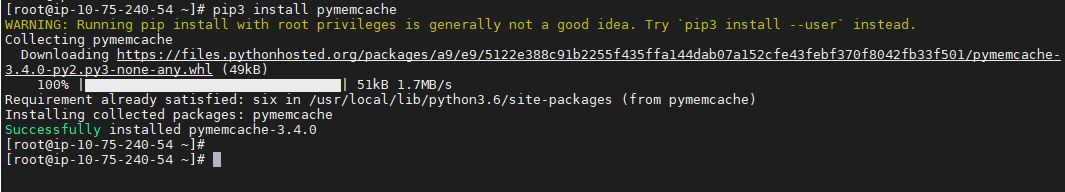


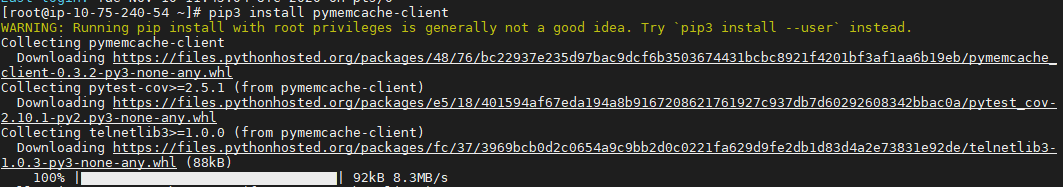
## Python3 modules:

Install below pyhton3 modules on Memcached Server

[root@stopvprvlogstash01 ~]# pip3 install pymemcache

[root@stopvprvlogstash01 ~]# pip3 install pymemcache-client

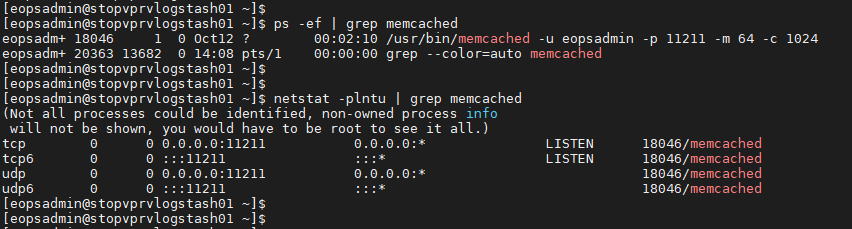




## Running Memcached:

Memcached can be run with below command and configure startup script.

[eopsadmin@stopvprvlogstash01 ~]$ /usr/bin/memcached -u eopsadmin -p 11211 -m 64 -c 1024



**Check Memcached Server Status:**

[eopsadmin@stopvprvlogstash01 ~]$ python3

Python 3.6.8 (default, Sep 26 2019, 11:57:09)

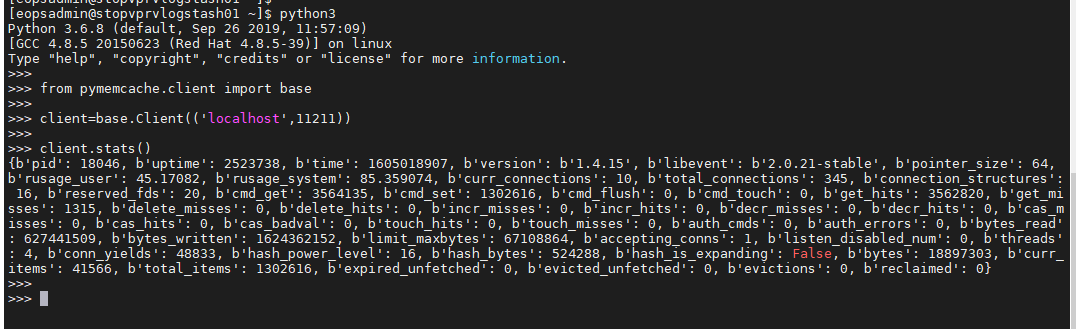
[GCC 4.8.5 20150623 (Red Hat 4.8.5-39)] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> from pymemcache.client import base

>>> client=base.Client(('localhost',11211))

>>> client.stats()



## Fetching CMDB data from ServiceNow:

Execute the below script to fetch CMDB data from ServiceNow to json file

/usr/bin/python /opt/eopsadmin/cmdb\_Inventory/bin/cmdb\_Inventory.py

Use the created json file to create Memcached database.

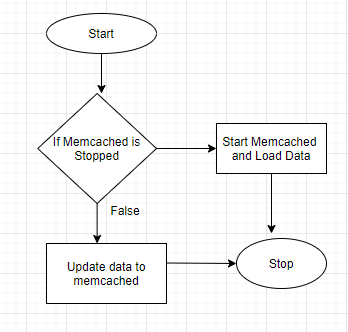
## Creating and updating Memcached database:

Execute below script to load data to Memcached

Below script Is scheduled as cronjob and will be running once in every day to update records.

/usr/bin/python3 /opt/eopsadmin/memcache/bin/memcache.py

**Script Flow Chart:**



## Updating CMDB data from ServiceNow:

Below Python script will fetch the updated records in ServiceNow and write it to json file and memcache.py script will update records.

/usr/bin/python /opt/eopsadmin/cmdb/bin/cmdb\_updated\_Inventory.py

## Crontab Scripts:

0 23 \* \* \* /usr/bin/python /opt/eopsadmin/cmdb/bin/cmdb\_updated\_Inventory.py

5 23 \* \* \* /usr/bin/python3 /opt/eopsadmin/memcache/bin/memcache.py

@reboot /usr/bin/python3 /opt/eopsadmin/memcache/bin/memcache.py

cmdb\_updated\_Inventory.py 🡪 to fetch updated CMDB records.

memcache.py 🡪 to start and update records in Memcached.

# Nginx Setup on Kibana

1. [root@ip-10-75-240-51 centos]# hostname -i

fe80::10b3:aff:fe36:9ac5%ens5 10.75.240.51

1. [root@ip-10-75-240-51 centos]# hostname

ip-10-75-240-51.ec2.internal

1. [root@ip-10-75-240-51 centos]# cd /etc/yum.repos.d/
2. [root@ip-10-75-240-51 yum.repos.d]# ls

redhat.repo

1. [root@ip-10-75-240-51 yum.repos.d]# yum install yum-utils

|  |
| --- |
| Loaded plugins: fastestmirror, package\_upload, product-id, search-disabled-repos, subscription-manager  1 local certificate has been deleted.  Determining fastest mirrors  InfraAuto\_CentOS\_Linux\_CentOS\_7 | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_CentOS\_7\_Updates | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_CentOS\_7\_Updates\_Errata | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_Centos\_7\_Extras | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_Chef\_EL7 | 1.8 kB 00:00:00  InfraAuto\_CentOS\_Linux\_Epel\_7 | 2.4 kB 00:00:00  InfraAuto\_CentOS\_Linux\_Katello\_EL7 | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_epel-subscription-manager-7 | 2.1 kB 00:00:00  Package matching yum-utils-1.1.31-52.el7.noarch already installed. Checking for update.  Nothing to do |

1. [root@ip-10-75-240-51 yum.repos.d]# vi /etc/yum.repos.d/nginx.repo
2. [root@ip-10-75-240-51 yum.repos.d]# cat /etc/yum.repos.d/nginx.repo

|  |
| --- |
| [nginx-stable]  name=nginx stable repo  baseurl=http://nginx.org/packages/centos/$releasever/$basearch/  gpgcheck=1  enabled=1  gpgkey=https://nginx.org/keys/nginx\_signing.key  module\_hotfixes=true  [nginx-mainline]  name=nginx mainline repo  baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/  gpgcheck=1  enabled=0  gpgkey=https://nginx.org/keys/nginx\_signing.key  module\_hotfixes=true |

1. [root@ip-10-75-240-51 yum.repos.d]# yum-config-manager --enable nginx-mainline

|  |
| --- |
| Loaded plugins: fastestmirror, product-id, subscription-manager  ==================================================================================== repo: nginx-mainline  [nginx-mainline]  async = True  bandwidth = 0  base\_persistdir = /var/lib/yum/repos/x86\_64/7  baseurl = http://nginx.org/packages/mainline/centos/7/x86\_64/  cache = 0  cachedir = /var/cache/yum/x86\_64/7/nginx-mainline  check\_config\_file\_age = True  compare\_providers\_priority = 80  cost = 1000  deltarpm\_metadata\_percentage = 100  deltarpm\_percentage =  enabled = 1  enablegroups = True  exclude =  failovermethod = priority  ftp\_disable\_epsv = False  gpgcadir = /var/lib/yum/repos/x86\_64/7/nginx-mainline/gpgcadir  gpgcakey =  gpgcheck = True  gpgdir = /var/lib/yum/repos/x86\_64/7/nginx-mainline/gpgdir  gpgkey = https://nginx.org/keys/nginx\_signing.key  hdrdir = /var/cache/yum/x86\_64/7/nginx-mainline/headers  http\_caching = all  includepkgs =  ip\_resolve =  keepalive = True  keepcache = False  mddownloadpolicy = sqlite  mdpolicy = group:small  mediaid =  metadata\_expire = 21600  metadata\_expire\_filter = read-only:present  metalink =  minrate = 0  mirrorlist =  mirrorlist\_expire = 86400  name = nginx mainline repo  old\_base\_cache\_dir =  password =  persistdir = /var/lib/yum/repos/x86\_64/7/nginx-mainline  pkgdir = /var/cache/yum/x86\_64/7/nginx-mainline/packages  proxy = False  proxy\_dict =  proxy\_password =  proxy\_username =  repo\_gpgcheck = False  retries = 10  skip\_if\_unavailable = False  ssl\_check\_cert\_permissions = True  sslcacert =  sslclientcert =  sslclientkey =  sslverify = True  throttle = 0  timeout = 30.0  ui\_id = nginx-mainline/7/x86\_64  ui\_repoid\_vars = releasever,  basearch  username = |

1. [root@ip-10-75-240-51 yum.repos.d]# yum install nginx

|  |
| --- |
| Loaded plugins: fastestmirror, package\_upload, product-id, search-disabled-repos, subscription-manager  Loading mirror speeds from cached hostfile  InfraAuto\_CentOS\_Linux\_CentOS\_7 | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_CentOS\_7\_Updates | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_CentOS\_7\_Updates\_Errata | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_Centos\_7\_Extras | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_Chef\_EL7 | 1.8 kB 00:00:00  InfraAuto\_CentOS\_Linux\_Epel\_7 | 2.4 kB 00:00:00  InfraAuto\_CentOS\_Linux\_Katello\_EL7 | 2.1 kB 00:00:00  InfraAuto\_CentOS\_Linux\_epel-subscription-manager-7 | 2.1 kB 00:00:00  nginx-mainline | 2.9 kB 00:00:00  nginx-stable | 2.9 kB 00:00:00  (1/2): nginx-stable/7/x86\_64/primary\_db | 55 kB 00:00:00  (2/2): nginx-mainline/7/x86\_64/primary\_db | 179 kB 00:00:00  Resolving Dependencies  --> Running transaction check  ---> Package nginx.x86\_64 1:1.19.2-1.el7.ngx will be installed  --> Finished Dependency Resolution  Dependencies Resolved  ==========================================================================  Package Arch Version Repository Size  =====================================================================================  Installing:  nginx x86\_64 1:1.19.2-1.el7.ngx nginx-mainline 781 k  Transaction Summary  ========================================================================================  Install 1 Package  Total download size: 781 k  Installed size: 2.7 M  Is this ok [y/d/N]: y  Downloading packages:  warning: /var/cache/yum/x86\_64/7/nginx-mainline/packages/nginx-1.19.2-1.el7.ngx.x86\_64.rpm: Header V4 RSA/SHA1 Signature, key ID 7bd9bf62: NOKEY ] 0.0 B/s | 213 kB --:--:-- ETA  Public key for nginx-1.19.2-1.el7.ngx.x86\_64.rpm is not installed  nginx-1.19.2-1.el7.ngx.x86\_64.rpm | 781 kB 00:00:03  Retrieving key from https://nginx.org/keys/nginx\_signing.key  Importing GPG key 0x7BD9BF62:  Userid : "nginx signing key <signing-key@nginx.com>"  Fingerprint: 573b fd6b 3d8f bc64 1079 a6ab abf5 bd82 7bd9 bf62  From : https://nginx.org/keys/nginx\_signing.key  Is this ok [y/N]: y  Running transaction check  Running transaction test  Transaction test succeeded  Running transaction  Installing : 1:nginx-1.19.2-1.el7.ngx.x86\_64 1/1  ----------------------------------------------------------------------  Thanks for using nginx!  Please find the official documentation for nginx here:  \* http://nginx.org/en/docs/  Please subscribe to nginx-announce mailing list to get  the most important news about nginx:  \* http://nginx.org/en/support.html  Commercial subscriptions for nginx are available on:  \* http://nginx.com/products/  ----------------------------------------------------------------------  Uploading Package Profile  Loaded plugins: fastestmirror, product-id, subscription-manager  Verifying : 1:nginx-1.19.2-1.el7.ngx.x86\_64 1/1  Installed:  nginx.x86\_64 1:1.19.2-1.el7.ngx  **Complete!** |

1. [root@ip-10-75-240-51 yum.repos.d]# cd /etc/nginx/conf.d
2. [root@ip-10-75-240-51 conf.d]# cp default.conf default\_conf\_bkp
3. [root@ip-10-75-240-51 conf.d]# ls

default.conf default\_conf\_bkp

1. #### Use Browser to Encoded username password using <https://www.base64encode.org/>

**User info:**

**User (**laas\_autoproxy**) and Password (elastic)**

1. **###copy the NSDev signed cert and Key file to below mentioned conf file locations##**
2. root@ip-10-75-240-51 conf.d]# vi default.conf
3. [root@ip-10-75-240-51 conf.d]# cat default.conf

|  |
| --- |
| server {  listen 10.75.240.51:443 ssl;  listen [::]:443 ssl;  server\_name 10.75.240.51;  ssl\_certificate /etc/pki/tls/certs/nsdev-cert-signed.crt;  ssl\_certificate\_key /etc/pki/tls/private/laas.neustar.biz.key;  access\_log /var/log/nginx/nginx.access.log;  error\_log /var/log/nginx/nginx.error.log;  location / {  #proxy\_pass https://awopuse1kibana01.laas.neustar.biz:5601/;  proxy\_pass https://10.75.240.51:5601/;  proxy\_set\_header Authorization "Basic bGFhc19hdXRvcHJveHk6ZWxhc3RpYw==";  }  } |

1. [root@ip-10-75-240-51 conf.d]# semanage port -a -t http\_port\_t -p tcp 5601

##To verify if the PORT id defined or not execute again and we see beow message:

1. [root@ip-10-75-240-51 conf.d]# semanage port -a -t http\_port\_t -p tcp 5601

ValueError: Port tcp/5601 already defined

1. [root@ip-10-75-240-51 conf.d]# sudo systemctl enable nginx

Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.

1. [root@ip-10-75-240-51 conf.d]# systemctl status nginx

|  |
| --- |
| ● nginx.service - nginx - high performance web server  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)  Active: inactive (dead)  Docs: http://nginx.org/en/docs/ |

1. [root@ip-10-75-240-51 conf.d]# netstat -anp | grep nginx

## we see nill output, as we did not started the nginx service

1. [root@ip-10-75-240-51 conf.d]# systemctl start nginx.service
2. [root@ip-10-75-240-51 conf.d]# systemctl status nginx.service

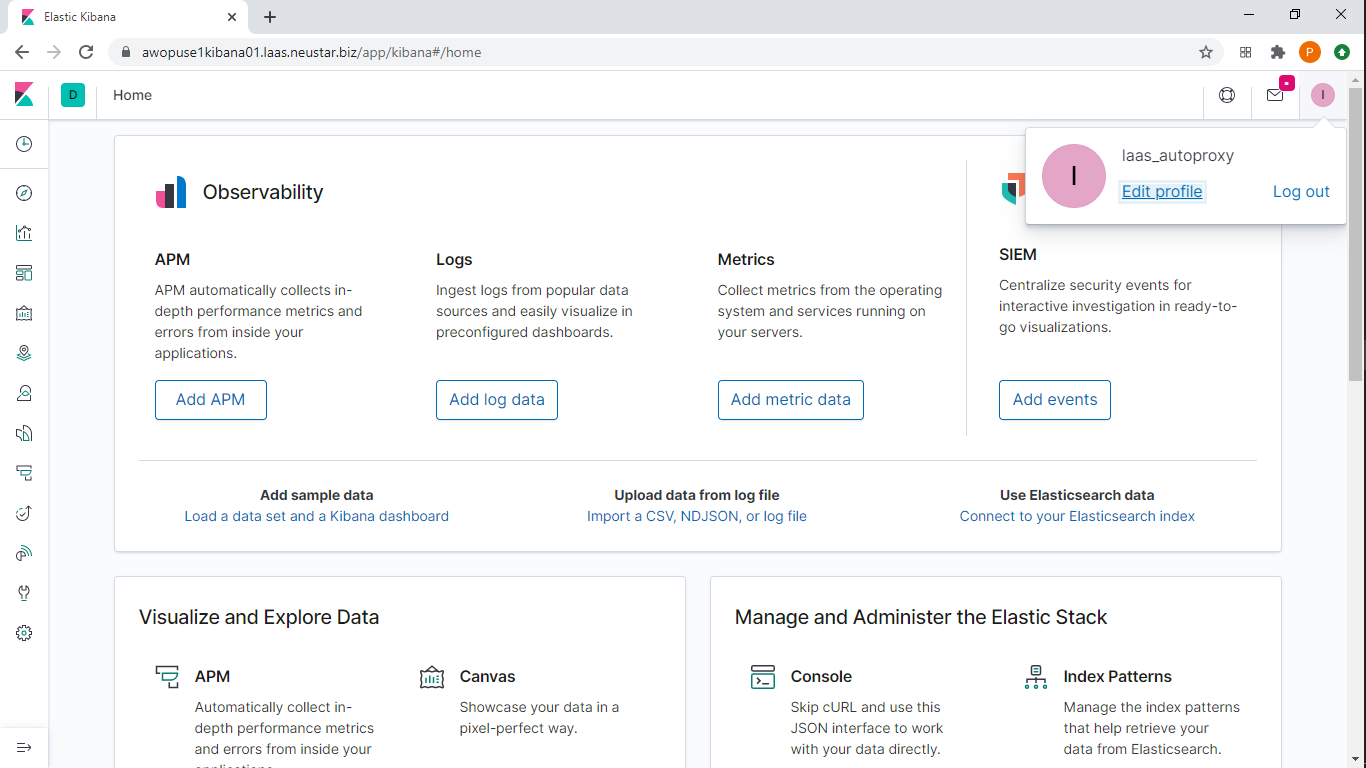
|  |
| --- |
| ● nginx.service - nginx - high performance web server  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)  Active: active (running) since Thu 2020-09-24 16:38:49 UTC; 1s ago  Docs: http://nginx.org/en/docs/  Process: 31271 ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf (code=exited, status=0/SUCCESS)  Main PID: 31272 (nginx)  CGroup: /system.slice/nginx.service  ├─31272 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf  └─31273 nginx: worker process  Sep 24 16:38:49 ip-10-75-240-51.ec2.internal systemd[1]: Starting nginx - high performance web server...  Sep 24 16:38:49 ip-10-75-240-51.ec2.internal systemd[1]: Started nginx - high performance web server. |

1. [root@ip-10-75-240-51 conf.d]# netstat -anp | grep nginx

|  |
| --- |
| tcp 0 0 10.75.240.51:443 0.0.0.0:\* LISTEN 31272/nginx: master  tcp6 0 0 :::443 :::\* LISTEN 31272/nginx: master  unix 3 [ ] STREAM CONNECTED 6740373 31272/nginx: master  unix 3 [ ] STREAM CONNECTED 6740374 31272/nginx: master |

1. Login to kibana UI and createthe user and giv permission to user “laas\_autoproxy”
2. Try login to nginx server (ip/fqdn) and we can connect to kibana-ui

<https://10.75.240.51/> (or) FQDN <https://awopuse1kibana01.laas.neustar.biz/>



# Kafka and Zookeeper Installation and configuration

## Hardware Specifications and Prerequsites

### Kafka & zookeeper cluster for each server

|  |  |
| --- | --- |
| Hardware profile | Specifications |
| Cores | 4 cores |
| physical memory | 16 of RAM |
| Disk | 500 gb |
| Network | 1- Gige Nic |
| File system | XFS |

### Zookeeper & Kafka Pre-Requisites

|  |  |
| --- | --- |
| Requirement’s | Details |
| User needs to be a sudo user  As “eopsadmin” | As the Installed components would be configured as service, we need sudo access to Start|Stop|Restart the service. |
| Java Installation | Kafka requires Java, so Java 1.8 needs to be installed along with servers build |
| Linux distribution | RHEL 7.x/Cent OS 7.x |
| Firewalls to be opened for the mentioned ports | 2181,9092,9093,2888,3888,9999,8778,9000 |

**Note:** All the below steps should be performed in all mentioned servers

**Disable RAM Swap - can set to 0 on certain Linux distro**

[root@ip-10-75-240-55 ~]# sysctl vm.swappiness=0

vm.swappiness = 0

**Add file limits configs - allow to open 100,000 file descriptors**

[root@ip-10-75-240-55 ~]# vi /etc/security/limits.conf

[root@ip-10-75-240-55 ~]# cat /etc/security/limits.conf

# Generated by Chef for node ip-10-77-1-197.nexgen.neustar.biz

# Local modifications will be overwritten!

\* hard core 0

\* soft core 0

\* hard rss 10000

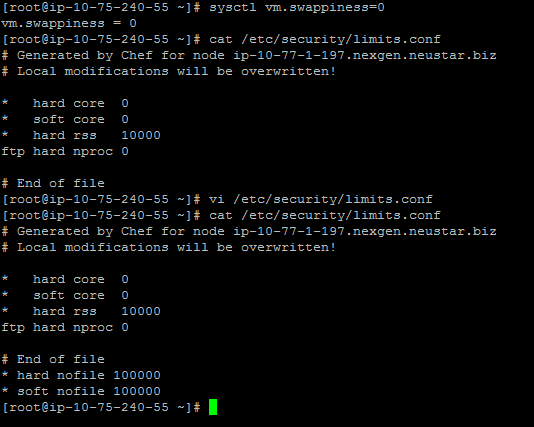
ftp hard nproc 0

# End of file

\* hard nofile 100000

\* soft nofile 100000

[root@ip-10-75-240-55 ~]#



**Add hosts entries (mocking DNS) - put relevant IPs here**

[root@ip-10-75-240-55 ~]# cat /etc/hosts

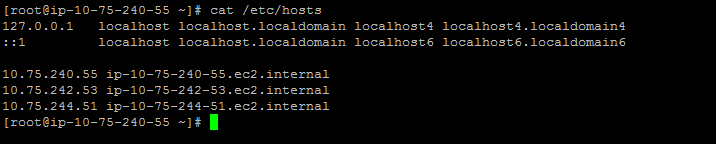
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

10.75.240.55 ip-10-75-240-55.ec2.internal

10.75.242.53 ip-10-75-242-53.ec2.internal

10.75.244.51 ip-10-75-244-51.ec2.internal



[root@ip-10-75-240-55 ~]#

**Creating a directory to place all Kafka related details in one place.**

[root@ip-10-75-240-55 ~]# mkdir /opt/eopsadmin

[root@ip-10-75-240-55 ~]# cd /opt/

[root@ip-10-75-240-55 opt]# ll

total 0

drwxr-xr-x. 2 root root 6 May 27 17:10 eopsadmin

[root@ip-10-75-240-55 opt]# chown eopsadmin:eopsadmin eopsadmin

[root@ip-10-75-240-55 opt]# ll

total 0

drwxr-xr-x. 2 eopsadmin eopsadmin 6 May 27 17:10 eopsadmin

[root@ip-10-75-240-55 opt]#

**Updated the Java path and Kafka path in bashrc file**

[eopsadmin@ip-10-75-240-55 bin]$ cd

[eopsadmin@ip-10-75-240-55 ~]$ vi .bashrc

[eopsadmin@ip-10-75-240-55 ~]$ cat .bashrc

# .bashrc

# Source global definitions

if [ -f /etc/bashrc ]; then

. /etc/bashrc

fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:

# export SYSTEMD\_PAGER=

# User specific aliases and functions

**export JAVA\_HOME=/opt/eopsadmin/jdk1.8.0\_251**

**export PATH=/opt/eopsadmin/kafka/bin:$PATH:$JAVA\_HOME/bin**

[eopsadmin@ip-10-75-240-55 ~]$ source ~/.bashrc

[eopsadmin@ip-10-75-240-55 ~]$ java -version

java version "1.8.0\_251"

Java(TM) SE Runtime Environment (build 1.8.0\_251-b08)

Java HotSpot(TM) 64-Bit Server VM (build 25.251-b08, mixed mode)

[eopsadmin@ip-10-75-240-55 ~]$ echo $JAVA\_HOME

/opt/eopsadmin/jdk1.8.0\_251

[eopsadmin@ip-10-75-240-55 ~]$

### Kafka and Zookeeper Installation

**Download and Install Kafka and Zookeeper SW Packages**

All the updates version of Kafka & zookeeper [http://apachemirror.wuchna.com/Kafka/](http://apachemirror.wuchna.com/kafka/)

wget http://apachemirror.wuchna.com/Kafka/2.5.0/Kafka\_2.12-2.5.0.tgz

**Note**: Both Kafka and zookeeper are bundled in one single tgz file.

**Post installs steps and Configurations**

[root@ip-10-75-240-55 eopsadmin]# wget http://apachemirror.wuchna.com/Kafka/2.5.0/Kafka\_2.12-2.5.0.tgz

--2020-05-27 17:20:59-- http://apachemirror.wuchna.com/Kafka/2.5.0/Kafka\_2.12-2.5.0.tgz

Resolving apachemirror.wuchna.com (apachemirror.wuchna.com)... 159.65.154.237

Connecting to apachemirror.wuchna.com (apachemirror.wuchna.com)|159.65.154.237|:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: 61604633 (59M) [application/x-gzip]

Saving to: ‘Kafka\_2.12-2.5.0.tgz’

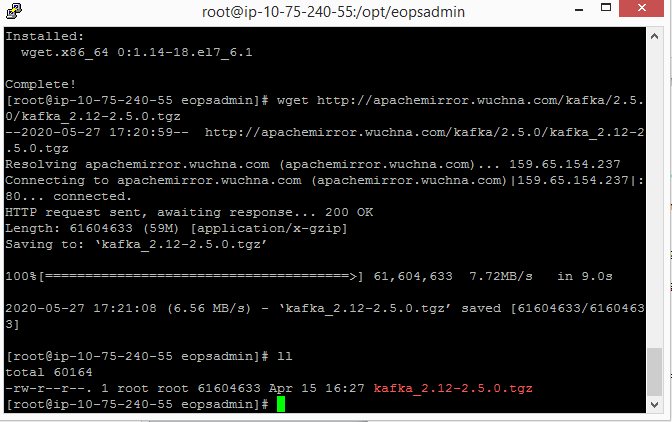
100%[======================================>] 61,604,633 7.72MB/s in 9.0s

2020-05-27 17:21:08 (6.56 MB/s) - ‘Kafka\_2.12-2.5.0.tgz’ saved [61604633/61604633]

[root@ip-10-75-240-55 eopsadmin]# ll

total 60164

-rw-r--r--. 1 root root 61604633 Apr 15 16:27 Kafka\_2.12-2.5.0.tgz



**Change ownership of the downloaded file**

[root@ip-10-75-240-55 eopsadmin]# chown eopsadmin:eopsadmin Kafka\_2.12-2.5.0.tgz

[root@ip-10-75-240-55 eopsadmin]# sudo su eopsadmin

[eopsadmin@ip-10-75-240-55 eopsadmin]$ pwd

/opt/eopsadmin

[eopsadmin@ip-10-75-240-55 eopsadmin]$

-rw-r--r--. 1 eopsadmin eopsadmin 61604633 Apr 15 16:27 Kafka\_2.12-2.5.0.tgz

**UnTar the downloaded Kafka file**

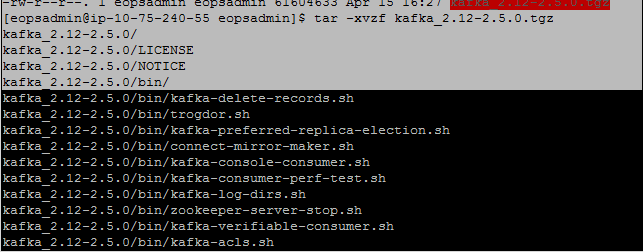
[eopsadmin@ip-10-75-240-55 eopsadmin]$ tar -xvzf Kafka\_2.12-2.5.0.tgz

Kafka\_2.12-2.5.0/

Kafka\_2.12-2.5.0/LICENSE

Kafka\_2.12-2.5.0/NOTICE

Kafka\_2.12-2.5.0/bin/



[eopsadmin@ip-10-75-240-55 eopsadmin]$ ll

total 60164

drwxr-xr-x. 6 eopsadmin eopsadmin 89 Apr 8 01:16 Kafka\_2.12-2.5.0

-rw-r--r--. 1 eopsadmin eopsadmin 61604633 Apr 15 16:27 Kafka\_2.12-2.5.0.tgz

**Rename the directory**

[eopsadmin@ip-10-75-240-55 eopsadmin]$ mv Kafka\_2.12-2.5.0 Kafka

[eopsadmin@ip-10-75-240-55 eopsadmin]$ ll

total 60164

drwxr-xr-x. 6 eopsadmin eopsadmin 89 Apr 8 01:16 Kafka

-rw-r--r--. 1 eopsadmin eopsadmin 61604633 Apr 15 16:27 Kafka\_2.12-2.5.0.tgz

**Below is the list of directories for Kafka**

[eopsadmin@ip-10-75-240-55 eopsadmin]$ cd Kafka

[eopsadmin@ip-10-75-240-55 Kafka]$ ll

total 56

drwxr-xr-x. 3 eopsadmin eopsadmin 4096 Apr 8 01:16 bin

drwxr-xr-x. 2 eopsadmin eopsadmin 4096 Apr 8 01:16 config

drwxr-xr-x. 2 eopsadmin eopsadmin 8192 May 28 16:14 libs

-rw-r--r--. 1 eopsadmin eopsadmin 32216 Apr 8 01:13 LICENSE

-rw-r--r--. 1 eopsadmin eopsadmin 337 Apr 8 01:13 NOTICE

drwxr-xr-x. 2 eopsadmin eopsadmin 44 Apr 8 01:16 site-docs

**Now create the Data directory for Kafka and zookeeper service**

[eopsadmin@ip-10-75-240-55 Kafka]$ pwd

/opt/eopsadmin/Kafka

[eopsadmin@ip-10-75-240-55 Kafka]$ mkdir data-Kafka

[eopsadmin@ip-10-75-240-55 Kafka]$ mkdir data-zoo

**Add above created directory and zookeeper server in highlighted place**

[eopsadmin@ip-10-75-240-55 config]$ cat zookeeper.properties

# the location to store the in-memory database snapshots and, unless specified otherwise, the transaction log of updates to the database.

dataDir=/opt/eopsadmin/Kafka/data-zoo

# the port at which the clients will connect

clientPort=2181

# disable the per-ip limit on the number of connections since this is a non-production config

maxClientCnxns=0

# the basic time unit in milliseconds used by ZooKeeper. It is used to do heartbeats and the minimum session timeout will be twice the tickTime.

tickTime=2000

# The number of ticks that the initial synchronization phase can take

initLimit=10

# The number of ticks that can pass between

# sending a request and getting an acknowledgement

syncLimit=5

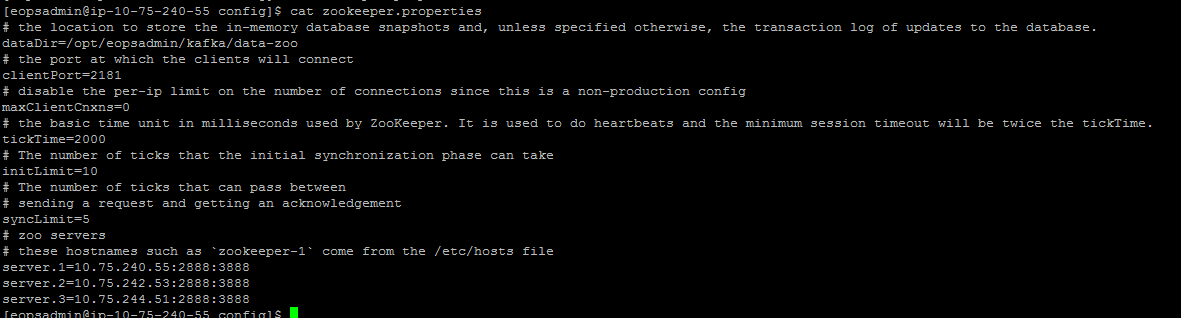
# zoo servers

# these hostnames such as `zookeeper-1` come from the /etc/hosts file

server.1=10.75.240.55:2888:3888

server.2=10.75.242.53:2888:3888

server.3=10.75.244.51:2888:3888



**Add list of Kafka bootstrap server in Producer and consumer properties in all 3 Kafka nodes.**

[eopsadmin@ip-10-75-240-55 config]$ pwd

/opt/eopsadmin/Kafka/config

[eopsadmin@ip-10-75-240-55 config]$ vi producer.properties

[eopsadmin@ip-10-75-240-55 config]$ cat producer.properties | grep -i 9092

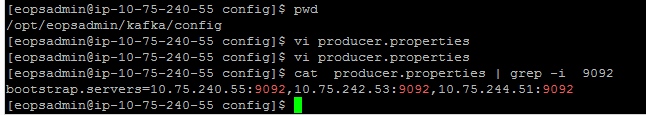
bootstrap.servers=10.75.240.55:9092,10.75.242.53:9092,10.75.244.51:9092

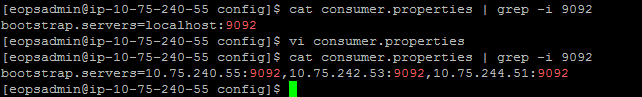
[eopsadmin@ip-10-75-240-55 config]$

[eopsadmin@ip-10-75-240-55 config]$ cat consumer.properties | grep -i 9092

bootstrap.servers=10.75.240.55:9092,10.75.242.53:9092,10.75.244.51:9092

[eopsadmin@ip-10-75-240-55 config]$





**Add list of properties in server properties in all 3 Kafka nodes.**

**Note:** the broker id should unique in all kafka nodes

eopsadmin@ip-10-75-240-55 config]$ cat server.properties | grep -i broker

# The id of the broker. This must be set to a unique integer for each broker.

**broker.id=0**

[eopsadmin@ip-10-75-240-55 config]$ cat server.properties | grep -i 9092

# listeners = PLAINTEXT://your.host.name:9092

listeners=PLAINTEXT://10.75.240.55:9092

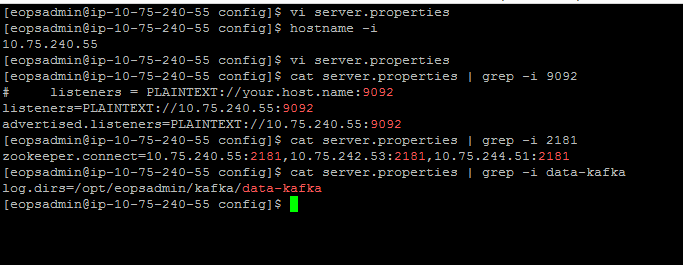
advertised.listeners=PLAINTEXT://10.75.240.55:9092

[eopsadmin@ip-10-75-240-55 config]$ cat server.properties | grep -i 2181

zookeeper.connect=10.75.240.55:2181,10.75.242.53:2181,10.75.244.51:2181

[eopsadmin@ip-10-75-240-55 config]$ cat server.properties | grep -i data-Kafka

log.dirs=/opt/eopsadmin/Kafka/data-kafka



**In zookeeper data directory, create file & add number as shown below in respective zookeeper servers.**

[eopsadmin@ip-10-75-240-55 Kafka]$ cd data-zoo/

[eopsadmin@ip-10-75-240-55 data-zoo]$ vi myid

[eopsadmin@ip-10-75-240-55 data-zoo]$ cat myid

1

[eopsadmin@ip-10-75-242-53 Kafka]$ cd data-zoo/

[eopsadmin@ip-10-75-242-53 data-zoo]$ vi myid

[eopsadmin@ip-10-75-242-52 data-zoo]$ cat myid

2

[eopsadmin@ip-10-75-244-51 Kafka]$ cd data-zoo/

[eopsadmin@ip-10-75-244-51 data-zoo]$ vi myid

[eopsadmin@ip-10-75-244-51 data-zoo]$ cat myid

3

### Zookeeper & Kafka service Creation

**Now create a service for Zookeeper in all zookeeper nodes**

[root@ip-10-75-240-55 centos]# cat /etc/systemd/system/zookeeper.service

[unit]

Requires=network.target remote-fs.target

After=network.target remote-fs.target

[Service]

Type=simple

User=eopsadmin

ExecStart=/opt/eopsadmin/Kafka/bin/zookeeper-server-start.sh /opt/eopsadmin/Kafka/config/zookeeper.properties

ExecStop=/opt/eopsadmin/Kafka/bin/zookeeper-server-stop.sh

Restart=on-abnormal

[Install]

WantedBy=multi-user.target

**Now create a service for Kafka in all Kafka servers**

[root@ip-10-75-240-55 centos]# cat /etc/systemd/system/Kafka.service

[unit]

Requires=zookeeper.service

After=zookeeper.service

[Service]

Type=simple

#User=root

User=eopsadmin

#Environment="KAFKA\_OPTS=-javaagent:/opt/Kafka\_2.13-2.4.0/jolokia-agent.jar=host=\*"

ExecStart=/bin/sh -c '/opt/eopsadmin/Kafka/bin/Kafka-server-start.sh /opt/eopsadmin/Kafka/config/server.properties > /opt/eopsadmin/Kafka/Kafka.log 2>&1'

ExecStop=/opt/eopsadmin/Kafka/bin/Kafka-server-stop.sh

Restart=on-abnormal

[Install]

WantedBy=multi-user.target

### Starting zookeeper & Kafka

**Use below commands to start first zookeeper & then Kafka service in all servers.**

sudo systemctl daemon-reload

sudo systemctl enable zookeeper.service

sudo systemctl start zookeeper.service

sudo systemctl statsu zookeeper.service

sudo systemctl enable kafka.service

sudo systemctl start kafka.service

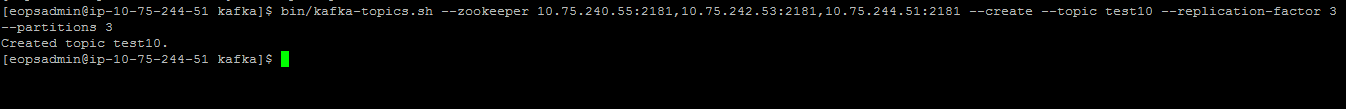
sudo systemctl status kafka.service

### Testing Kafka cluster

**Use below commands to test the kafka cluster.**

[eopsadmin@ip-10-75-244-51 Kafka]$ **bin/Kafka-topics.sh --zookeeper 10.75.240.55:2181,10.75.242.53:2181,10.75.244.51:2181 --create --topic test10 --replication-factor 3 --partitions 3**

Created topic test10



[**eopsadmin@ip-10-75-244-51 Kafka]$ bin/Kafka-console-producer.sh --broker-list 1 0.75.240.55:9092,10.75.242.53:9092,10.75.244.51:9092 --topic test10**

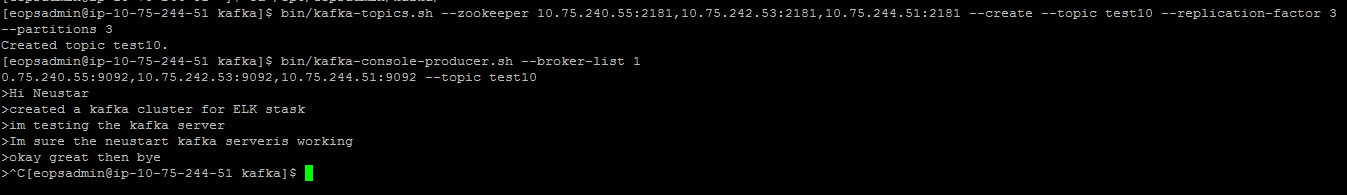
>Hi Neustar

>created a Kafka cluster for ELK stask

>im testing the Kafka server

>Im sure the neustart Kafka serveris working

>okay great then bye



[eopsadmin@ip-10-75-240-55 Kafka]$ **bin/Kafka-console-consumer.sh --bootstrap-server**

10.75.240.55:9092,10.75.242.53:9092,10.75.244.51:9092 –topic test10 –from-beginning

created a Kafka cluster for ELK stask

Hi Neustar

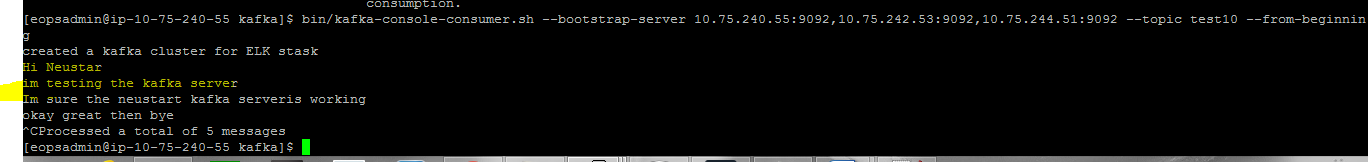
im testing the Kafka server

Im sure the neustart Kafka serveris working

Okay great then bye

okay great then bye

^CProcessed a total of 5 messages



### Topics creation

Execute the below command to create topics in kafka

[eopsadmin@ip-10-75-240-55 Kafka]$ bin/Kafka-topics.sh --zookeeper 10.75.240.55:2181,10.75.242.53:2181,10.75.244.51:2181 --create --topic **eops.neteng.unixsyslog.ls.logging.v01** --replication-factor 3 --partitions 6

Created topic eops.neteng.unixsyslog.ls.logging.v01.

### Monitoring

#### Kafka Manager:

Using Kafka manager UI, we can manage Kafka cluster form the UI rather than CLI

**Create a directory**mkdir **Kafka-manager**

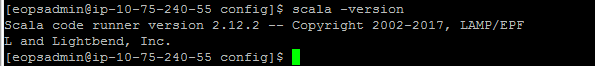
#### Installing scala

wget https://downloads.lightbend.com/scala/2.12.2/scala-2.12.2.rpm

sudo yum localinstall -y scala-2.12.2.rpm

### check scala install ###

scala -version



## Install SBT from RPM ###

curl https://bintray.com/sbt/rpm/rpm | sudo tee /etc/yum.repos.d/bintray-sbt-rpm.repo

sudo yum install sbt

Once java, scala, and sbt are installed we are ready to copy the kafka-manager repo from GitHub using git or you can download the .zip.

git clone https://github.com/yahoo/kafka-manager.git

or

wget https://github.com/yahoo/kafka-manager/archive/1.3.3.22.zip

cd ./kafka-manager

sbt clean dist # this part is going to take some time as it has to download all the dependencies

mv ./target/universal/kafka-manager-1.3.3.8.zip ~/opt/eopsadmin/

unzip kafka-manager-1.3.3.8.zip

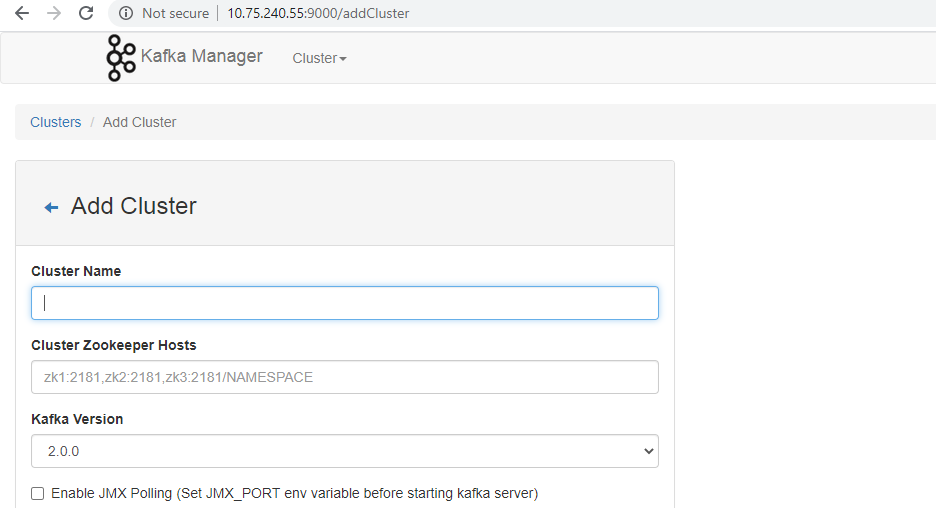
mv kafka-manager-1.3.3.8.zip to kafka-manager

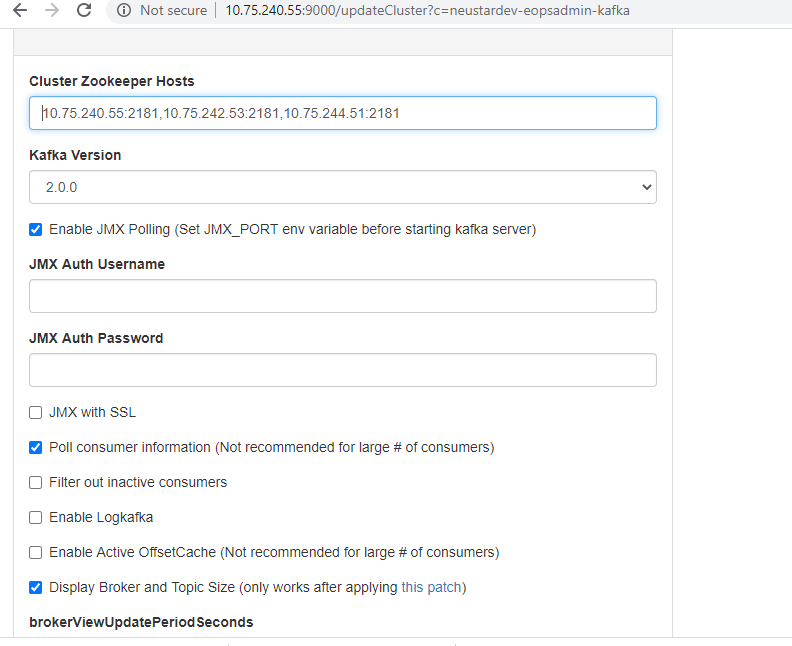
### Start Kafka-Manager ###

nohup ./bin/kafka-manager -Dkafka-manager.zkhosts="10.75.240.55:2181,10.75.242.53:2181,10.75.244.51:2181" &

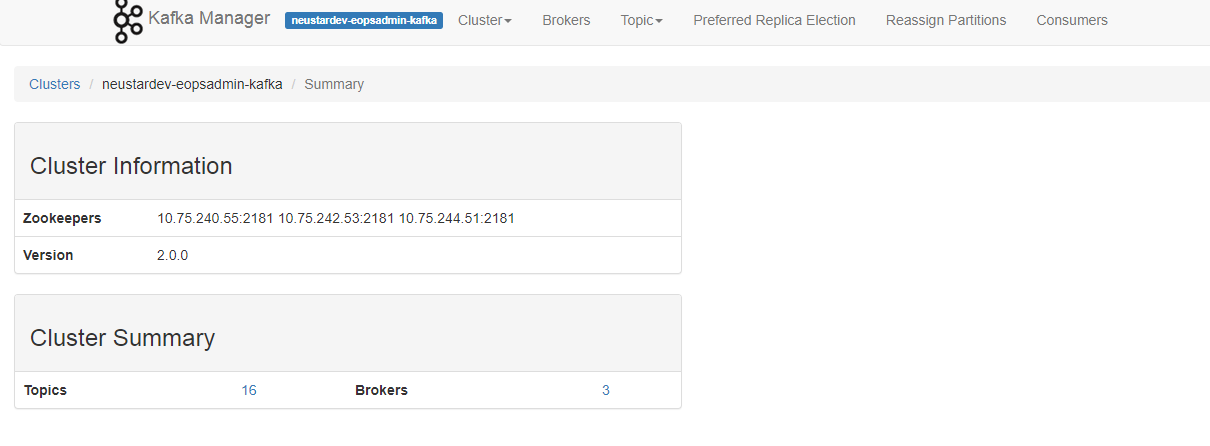
After running the kafka manger using below url

[http:// 10.75.240.55:9000/](http://192.168.1.241:9000/)





Add the above information and save



#### Metrix:

**JMX:** Need to add the JMX\_port number to the kafka startup script

ExecStart=/bin/sh -c 'env JMX\_PORT=9999 /opt/eopsadmin/kafka/bin/kafka-server-start.sh /opt/eopsadmin/kafka/config/server.properties > /opt/eopsadmin/kafka/kafka.log 2>&1'



**Jolokia:**

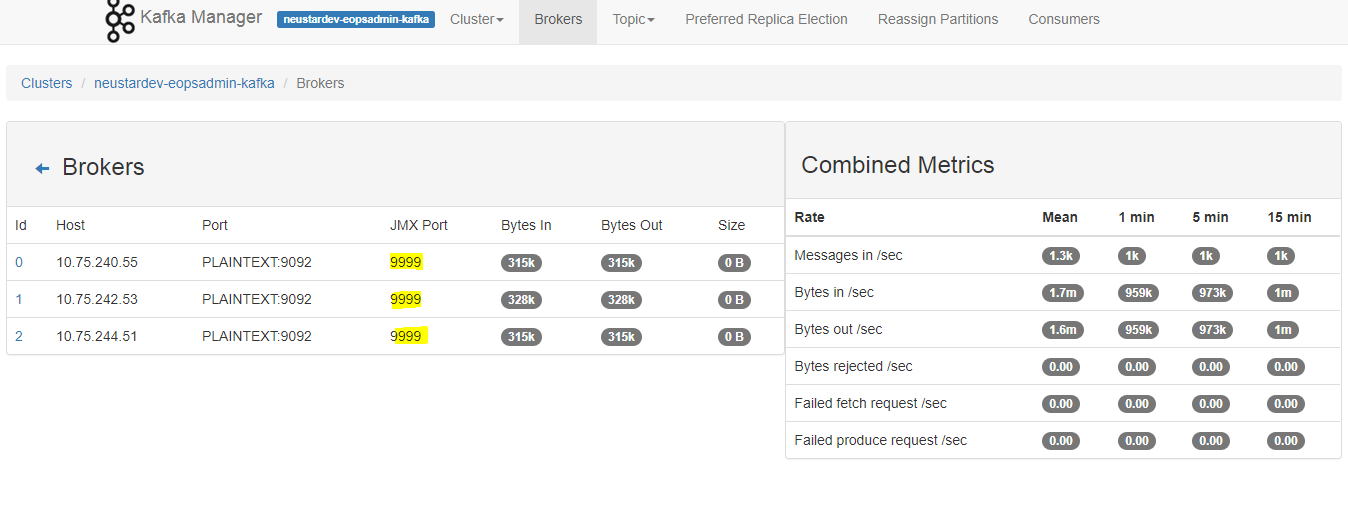
Need to install jolokia agent and place agent at Kafka accessible place.

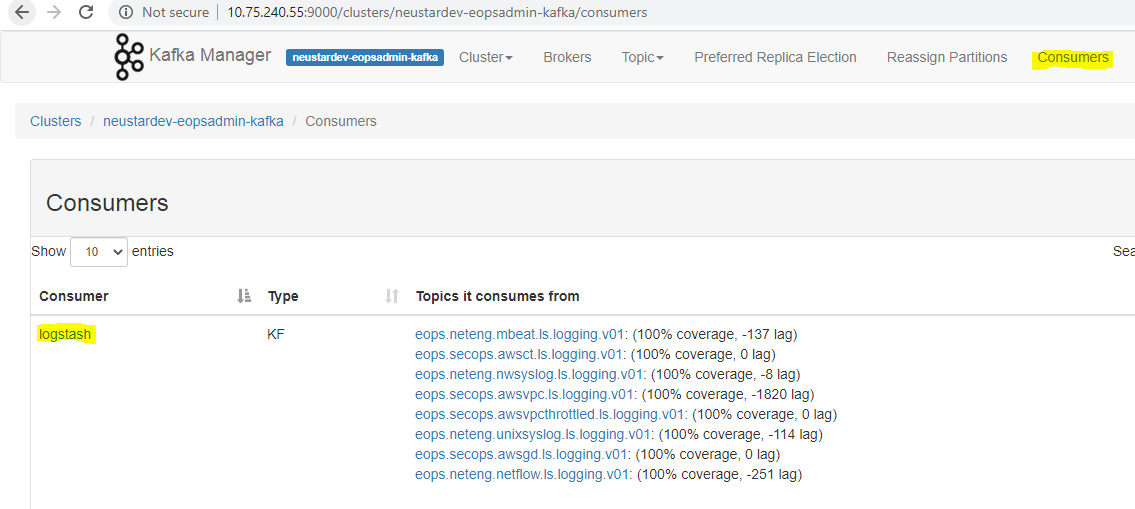
wget [http://search.maven.org/remotecontent?filepath=org/jolokia/jolokia-jvm/1.6.0/jolokia-jvm-1.6.0-agent.jar -O jolokia-agent.jar](http://search.maven.org/remotecontent?filepath=org/jolokia/jolokia-jvm/1.6.0/jolokia-jvm-1.6.0-agent.jar%20-O%20jolokia-agent.jar)

Need add the ENV at the kafka startup-script

Environment="KAFKA\_OPTS=-javaagent:/opt/eopsadmin/jolokia-agent.jar=host=\*"

After adding the jmx and jolokia restart the kafka service.





# Elastalert Installation and configuration

## Elastalert Pre-Requisites

|  |  |
| --- | --- |
| User can be a non sudo user ( I have installed using eopsadmin) | Ensure before **installing** a service, should use sudo before command |
| Python Installation | Elastalert requires Python, so Python 3.6 needs to be installed along with servers build |
| Linux distribution | RHEL 8.x () /Cent OS 7.x |
| Firewalls to be opened for the mentioned ports | 8080, 8000, 9200, 9990 |

## System, Network and Other Pre-requisites

**git (sudo yum install git)  
 httpd (sudo yum install httpd)**

## Download and Install Elastalert and SW Packages

Recheck the OS version in your machine.   
 [eopsadmin@ip-10-75-240-54 ~]$ cat /etc/redhat-release

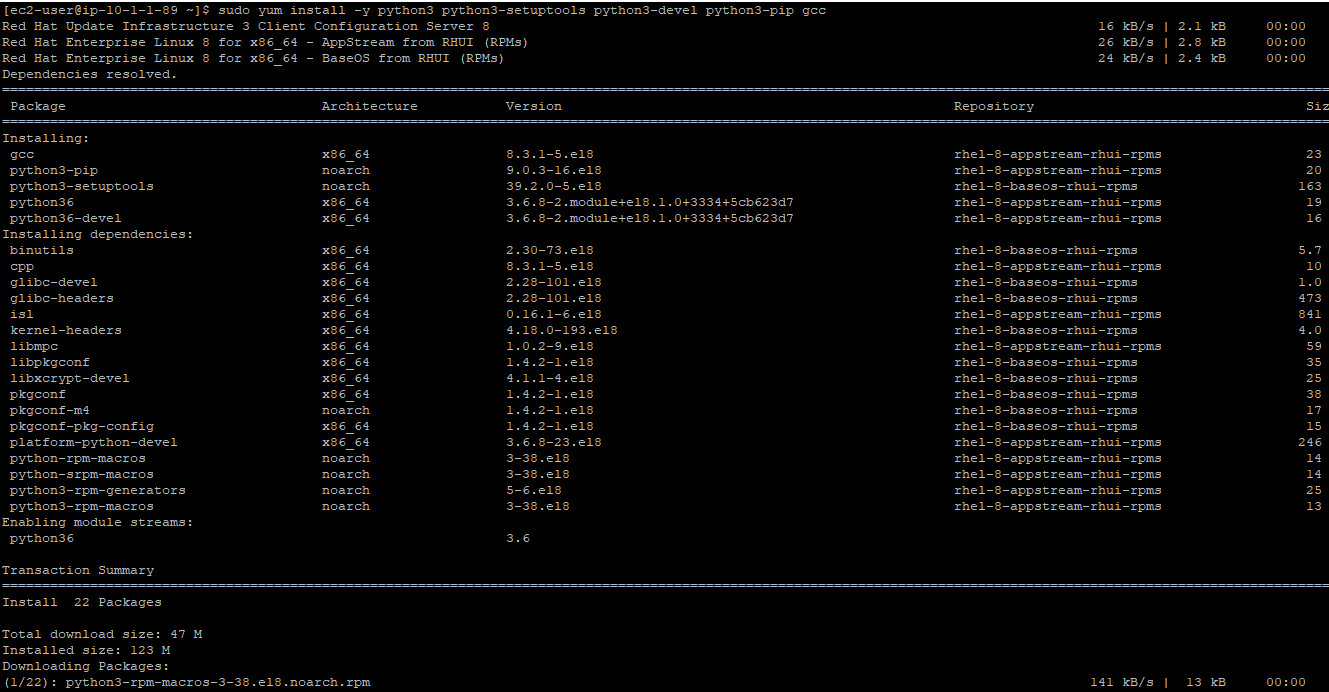
CentOS Linux release 7.8.2003 (Core)

We do not have any version of python in our system so far!  
 By executing python command you can know that as below.

[eopsadmin@ip-10-75-240-54 ~]$ python3  
 -bash: python3: command not found

### Installing python3

[eopsadmin@ip-10-75-240-54 ~]$ sudo yum install -y python3 python3-setuptools python3-devel python3-pip gcc



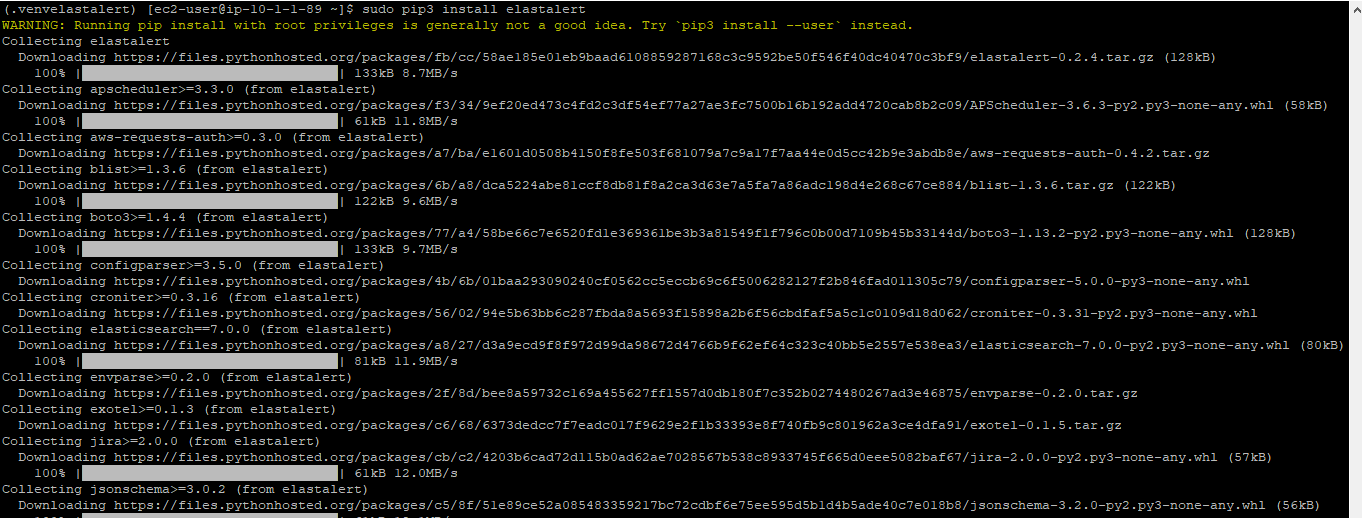
**Run python3 command, if you are able to see shell as below it means python installation is successful**

[eopsadmin@ip-10-75-240-54 ~]$ python3

Enter **exit()** and come out of python shell

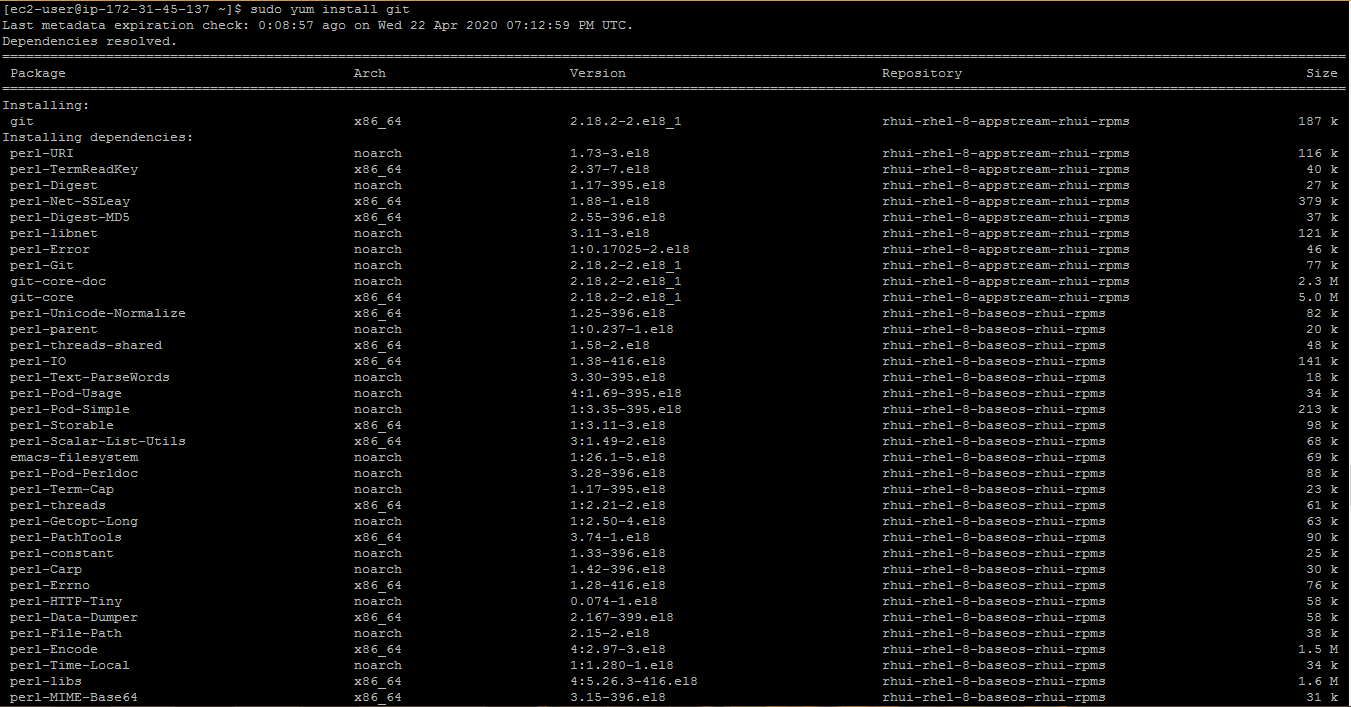
### Installation of Elastalert and its modules

[eopsadmin@ip-10-75-240-54 eopsadmin ]$ sudo pip3 install elastalert



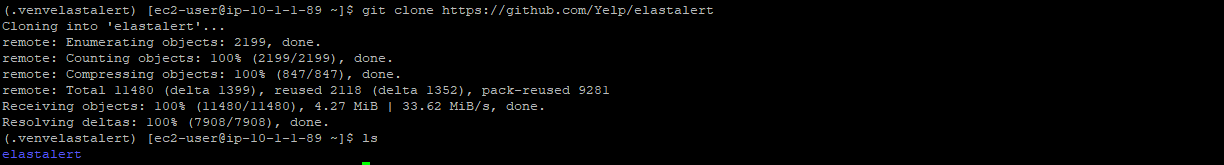
If git is already available in your system, you can ignore this step.

[eopsadmin@ip-10-75-240-54 eopsadmin]$ sudo yum install git



### Install code base

[eopsadmin@ip-10-75-240-54 eopsadmin]$ git clone <https://github.com/Yelp/elastalert>



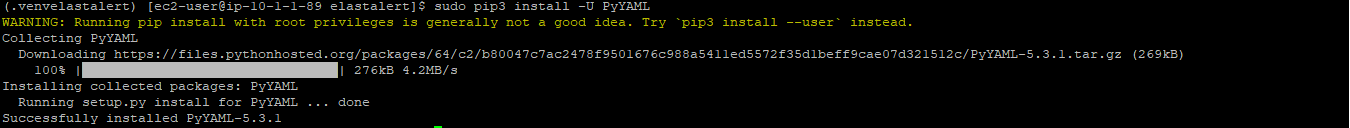
Swtich to elastalert directory as below

[eopsadmin@ip-10-75-240-54 eopsadmin ]$ cd elastalert/

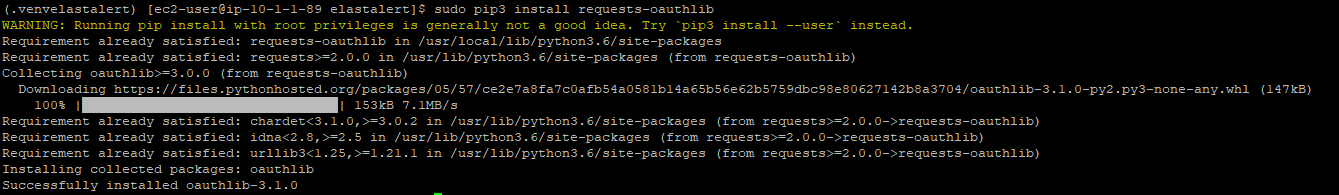
Now run, sudo pip3 install "setuptools>=11.3"

(.elastalert) [eopsadmin@ip-10-75-240-54 elastalert]$ sudo pip3 install "setuptools>=11.3"

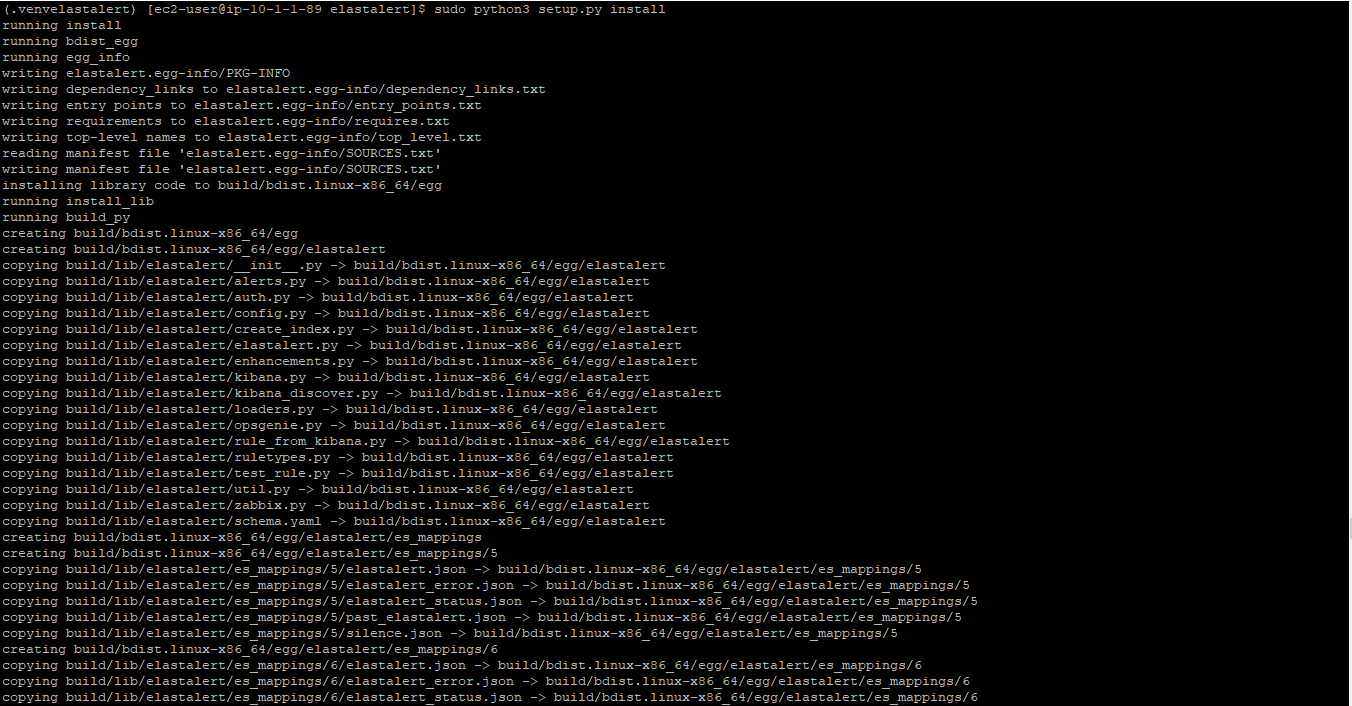
[eopsadmin@ip-10-75-240-54 elastalert]$ sudo pip3 install -U PyYAML



[eopsadmin@ip-10-75-240-54 elastalert]$ sudo pip3 install requests-oauthlib



[eopsadmin@ip-10-75-240-54 elastalert]$ sudo python3 setup.py install



## Post Install steps and Configs

### edit config.yaml

[eopsadmin@ip-10-75-240-54 elastalert]$ cp config.yaml.example config.yaml

Here we are creating **config.yaml** from **config.yaml.example** All elastic search related configs are defined here. [All host, port etc information is taken from **config.yaml** ]

[eopsadmin@ip-10-75-240-54 elastalert]$ sudo vi config.yaml   
  
search for **es\_host,** **es\_port, es\_username, es\_password** in **config.yaml** andspecify **elasticsearch** information accordingly**.**

Add below fields in config.yaml file

rules\_folder: rules

run\_every:

minutes: 1

buffer\_time:

minutes: 15

es\_host: 10.75.242.51

es\_port: 9200

es\_username: elastic

es\_password: elastic

writeback\_index: eopsadmin.elastalert\_status

writeback\_alias: eopsadmin.elastalert\_alerts

alert\_time\_limit:

days: 2

### Create Index

[eopsadmin@ip-10-75-240-54 elastalert]$ elastalert-create-index

This will create **elastalert\_status, elastalert\_status\_error, elastalert\_status\_silence,**

**elastalert\_status\_status, elastalert\_status\_past** indices in elasticsearch

**Issues/Troubleshooting:**

If you get any **elastisearch** related errors **Run,**[eopsadmin@ip-10-75-240-54 elastalert]$ **pip3 install -r requirements.txt**

1. **Rules configuration and running elastalert**

Configure rules file as below

es\_host: 10.75.242.51

es\_port: 9200

es\_username: "elastic"

es\_password: "elastic"

################################################

name: high\_cpu\_percentage050

type: any

index: metricbeat-\*

query\_key:

- beat.name

buffer\_time:

minutes: 5

filter:

- range:

system.process.cpu.total.pct:

from: 0.9

to: 10.0

alert:

- "post"

alertsource: "Elasticsearch"

http\_post\_url: "https://10.75.242.51:9200/eopsadmin.elastalert\_es\_endpoint/"

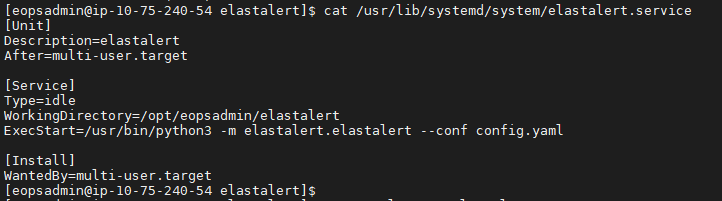
http\_post\_all\_values:

event.summary : "CPU utilization exceeds threshold"

http\_post\_headers:

authorization: Basic ZWxhc3RpYzplbGFzdGlj

Creating elastalert service



Start elastalert 🡪 systemctl start elastalert

## Installation issues and additional steps

Additional steps required for installation of Elastalert in Dev Environment: (If the issue is pertaining to gcc follow below steps)

**Installation of gcc:**

**Install glibc rpms from below links using wget:**

http://mirror.centos.org/centos/7/os/x86\_64/Packages/glibc-headers-2.17-307.el7.1.x86\_64.rpm

http://mirror.centos.org/centos/7/os/x86\_64/Packages/compat-glibc-2.12-4.el7.centos.x86\_64.rpm

http://mirror.centos.org/centos/7/os/x86\_64/Packages/compat-glibc-headers-2.12-4.el7.centos.x86\_64.rpm

<http://ftp.riken.jp/Linux/cern/centos/7/updates/x86_64/Packages/Packages/glibc-devel-2.17-307.el7.1.x86_64.rpm>

* On doing **wget** on above **links** we will below files as **rpm**

glibc-headers-2.17-307.el7.1.x86\_64.rpm   
compat-glibc-2.12-4.el7.centos.x86\_64.rpm   
compat-glibc-headers-2.12-4.el7.centos.x86\_64.rpm  
glibc-devel-2.17-307.el7.1.x86\_64.rpm

* Now run, below command i.e with all rpm’s together as below

rpm -ivh compat-glibc-2.12-4.el7.centos.x86\_64.rpm glibc-devel-2.17-307.el7.1.x86\_64.rpm compat-glibc-headers-2.12-4.el7.centos.x86\_64.rpm glibc-headers-2.17-307.el7.1.x86\_64.rpm --nodeps

**Install gcc rpms from below links using wget:**

http://mirror.centos.org/centos/7/os/x86\_64/Packages/gcc-gfortran-4.8.5-39.el7.x86\_64.rpm

http://mirror.centos.org/centos/7/os/x86\_64/Packages/gcc-c++-4.8.5-39.el7.x86\_64.rpm

<http://mirror.centos.org/centos/7/os/x86_64/Packages/gcc-4.8.5-39.el7.x86_64.rpm>

* On doing **wget** on above **links** we will below files as **rpm**

gcc-gfortran-4.8.5-39.el7.x86\_64  
gcc-c++-4.8.5-39.el7.x86\_64   
gcc-4.8.5-39.el7.x86\_64

* Now run, below command i.e with all rpm’s together as below

rpm -ivh gcc-4.8.5-39.el7.x86\_64.rpm gcc-gfortran-4.8.5-39.el7.x86\_64.rpm gcc-c++-4.8.5-39.el7.x86\_64.rpm --nodeps

**Sample response as output for above steps:**

**[root@ip-10-75-240-54 eopsadmin]# rpm -ivh gcc-4.8.5-39.el7.x86\_64.rpm gcc-gfortran-4.8.5-39.el7.x86\_64.rpm gcc-c++-4.8.5-39.el7.x86\_64.rpm –nodeps  
Preparing... ################################# [100%]  
Updating / installing...  
 1:gcc-4.8.5-39.el7 ################################# [ 33%]  
 2:gcc-gfortran-4.8.5-39.el7 ################################# [ 67%]  
 3:gcc-c++-4.8.5-39.el7 ################################# [100%]**

* Run, **yum install cpp-4.8.5-39.el7.x86\_64**

**If gcc issue is still not resolved and facing errors/issues w.r.t header files run below steps**

* Created a folder with name **linux** under **/usr/include/**
* Run, **cp \*.h linux** ( Execute this command by being on /usr/include/, Here we are copying all headers files that are under /usr/include/ to /usr/include/linux )
* Now check for gcc version:

[root@ip-10-75-240-54 eopsadmin]# **gcc –version**  
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39)  
Copyright (C) 2015 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

# Alerta Installtion and Configuration

## **Prerequisites**:

1. Ptyhon3.x
2. Mongodb3.x

## MongoDB Installation

### MongoDB Prerequisites

Use the below command to install the dependencies required for the MongoDB Community .tgz tarball:

sudo yum install libcurl openssl

mongodb requires mongod user

useradd mongod

Set Selinux status to permissive

sudo vi /etc/selinux/config

Set selinux to permissive

SELINUX=permissive

To set selinux to permissive 🡪 sudo setenforce 0

To check status of selinux 🡪 sestatus

### Download tarball and Extract file

[eopsadmin@ip-10-75-240-54 mongodb]$ wget <https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-4.2.8.tgz>

[eopsadmin@ip-10-75-240-54 mongodb]$ tar -zxvf mongodb-linux-x86\_64-rhel70-4.2.8.tgz

### Create mongodb data and log directories

[root@ip-10-75-240-54 mongodb]# pwd

/opt/eopsadmin/mongodb

[root@ip-10-75-240-54 mongodb]# mkdir mongo

[root@ip-10-75-240-54 mongodb]# chown mongod:mongod mongo/

[root@ip-10-75-240-54 mongodb]#

[root@ip-10-75-240-54 mongodb]# mkdir logs

[root@ip-10-75-240-54 mongodb]# chown mongod:mongod logs/

[root@ip-10-75-240-54 mongodb]#

### MongoDB Configuration

[eopsadmin@ip-10-75-240-54 mongodb]$ mkdir conf

[eopsadmin@ip-10-75-240-54 conf]$ vi mongod.conf

# mongod.conf

# for documentation of all options, see:

# http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.

systemLog:

destination: file

logAppend: true

path: /opt/eopsadmin/mongodb/logs/mongod.log

# Where and how to store data.

storage:

dbPath: /opt/eopsadmin/mongodb/mongo

journal:

enabled: true

# engine:

# mmapv1:

# wiredTiger:

# how the process runs

processManagement:

pidFilePath: /var/run/mongodb/mongod.pid # location of pidfile

# network interfaces

net:

port: 27017

bindIp: 127.0.0.1 # Listen to local interface only, comment to listen on all interfaces.

#security:

#operationProfiling:

#replication:

#sharding:

## Enterprise-Only Options

#auditLog:

#snmp:

### Service Creation

[root@ip-10-75-240-54 /]# vi /usr/lib/systemd/system/mongod.service

[Unit]

Description=mongod

After=multi-user.target

[Service]

Type=idle

WorkingDirectory=/opt/eopsadmin/mongodb

ExecStart=/opt/eopsadmin/mongodb/bin/mongod -f /opt/eopsadmin/mongodb/conf/mongod.conf

[Install]

WantedBy=multi-user.target

### Start Mongodb Service

[root@ip-10-75-240-54 /]# systemctl start mongod

[root@ip-10-75-240-54 /]# systemctl status mongod

● mongod.service - mongod

Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)

Active: active (running) since Tue 2020-07-21 17:35:40 UTC; 3s ago

Main PID: 11376 (mongod)

CGroup: /system.slice/mongod.service

└─11376 /opt/eopsadmin/mongodb/bin/mongod -f /opt/eopsadmin/mongodb/conf/mongod.conf

Jul 21 17:35:40 ip-10-75-240-54.ec2.internal systemd[1]: Started mongod.

### Recommended ulimit Settings

Every deployment may have unique requirements and settings; however, the following thresholds and settings are particularly important for mongod and mongos deployments:

* -f (file size): unlimited
* -t (cpu time): unlimited
* -v (virtual memory): unlimited
* -l (locked-in-memory size): unlimited
* -n (open files): 64000
* -m (memory size): unlimited
* -u (processes/threads): 64000

[eopsadmin@ip-10-75-240-54 mongodb]$ ulimit -a

core file size (blocks, -c) 0

data seg size (kbytes, -d) unlimited

scheduling priority (-e) 0

file size (blocks, -f) unlimited

pending signals (-i) 61841

max locked memory (kbytes, -l) 64

max memory size (kbytes, -m) 10000

open files (-n) 65535

pipe size (512 bytes, -p) 8

POSIX message queues (bytes, -q) 819200

real-time priority (-r) 0

stack size (kbytes, -s) 8192

cpu time (seconds, -t) unlimited

max user processes (-u) unlimited

virtual memory (kbytes, -v) unlimited

file locks (-x) unlimited

## AlertaServer Installtion

### Install Python3 and alerta modules

yum install python3

pip3 install alerta-server==7.4.5 alerta==7.4.4 # stable release

### Configuration

Create a config file at /etc/alertad.conf

And add the below lines

#

# \*\*\*\*\* ALERTA SERVER DEFAULT SETTINGS -- DO NOT MODIFY THIS FILE \*\*\*\*\*

#

# To override these settings use /etc/alertad.conf or the contents of the

# configuration file set by the environment variable ALERTA\_SVR\_CONF\_FILE.

#

# Further information on settings can be found at http://docs.alerta.io

DEBUG = True

SECRET\_KEY = '\*\*\*\*\*'

BASE\_URL = 'http://10.75.240.54:8080'

USE\_PROXYFIX = False

LOG\_HANDLERS = ['console', 'file']

LOG\_FILE = '/opt/eopsadmin/alerta/logs/alertad.log'

LOG\_MAX\_BYTES = 510241024 # 5 MB

LOG\_BACKUP\_COUNT = 2

LOG\_FORMAT = 'verbose'

ALARM\_MODEL='ALERTA'

DEFAULT\_FIELD = 'text'

DEFAULT\_PAGE\_SIZE = 1000

HISTORY\_LIMIT = 100

HISTORY\_ON\_VALUE\_CHANGE = False # do not log if only value changes

DATABASE\_ENGINE = 'mongo'

MONGO\_URI = 'mongodb://10.75.240.54:27017/monitoring'

MONGO\_DATABASE = None # can be used to override default database, above

AUTH\_REQUIRED = False

ADMIN\_USERS = []

CUSTOMER\_VIEWS = False

OAUTH2\_CLIENT\_ID = None # Google or GitHub OAuth2 client ID and secret

OAUTH2\_CLIENT\_SECRET = None

ALLOWED\_EMAIL\_DOMAINS = ['\*']

GITHUB\_URL = None

ALLOWED\_GITHUB\_ORGS = ['\*']

GITLAB\_URL = None

ALLOWED\_GITLAB\_GROUPS = ['\*']

KEYCLOAK\_URL = None

KEYCLOAK\_REALM = None

ALLOWED\_KEYCLOAK\_ROLES = ['\*']

TOKEN\_EXPIRE\_DAYS = 14

API\_KEY\_EXPIRE\_DAYS = 365 # 1 year

CORS\_ORIGINS = [

'http://10.75.240.54',

'http://10.75.240.54:8080',

'http://10.75.240.54:8000',

'http://localhost:8080',

r'https?://\w\*\.?local\.alerta\.io:?\d\*/?.' # => http(s)://.local.alerta.io:<port>

]

SEVERITY\_MAP = {

'critical': 1,

'warning': 4,

'indeterminate': 5,

'ok': 5,

'unknown': 9

}

DEFAULT\_NORMAL\_SEVERITY = 'ok' # 'normal', 'ok', 'cleared'

DEFAULT\_PREVIOUS\_SEVERITY = 'indeterminate'

COLOR\_MAP = {

'severity': {

'critical': 'red',

'warning': '#1E90FF',

'indeterminate': 'lightblue',

'ok': '#00CC00',

'unknown': 'silver'

},

'text': 'black',

'highlight': 'skyblue '

}

DATE\_FORMAT\_SHORT\_TIME = 'HH:mm'

DATE\_FORMAT\_MEDIUM\_DATE = 'EEE d MMM HH:mm'

DATE\_FORMAT\_LONG\_DATE = 'd/M/yyyy h:mm:ss.sss a'

DEFAULT\_AUDIO\_FILE = '/audio/Bike Horn.mp3'

COLUMNS = ['severity', 'status', 'lastReceiveTime', 'duplicateCount',

'customer', 'environment', 'service', 'resource', 'event', 'value', 'text']

SORT\_LIST\_BY = 'lastReceiveTime'

ACTIONS = ['createIssue', 'updateIssue']

GOOGLE\_TRACKING\_ID = 'UA-44644195-5'

AUTO\_REFRESH\_INTERVAL = 30000 # 30s

PLUGINS = ['reject', 'blackout', 'slack']

PLUGINS\_RAISE\_ON\_ERROR = False # keep processing other plugins if exception

ORIGIN\_BLACKLIST = ['foo/bar$', '.\*/qux'] # reject all foo alerts from bar, and everything from qux

ALERTA\_DEFAULT\_PROFILE='Development'

ALLOWED\_ENVIRONMENTS = ['Development']

BLACKOUT\_DURATION = 86400 # 2 hours

NOTIFICATION\_BLACKOUT = True

BLACKOUT\_ACCEPT = ['normal', 'ok', 'cleared']

### Step 4 : export the envirement variables

export ALERTA\_SVR\_CONF\_FILE=~/.alertad.conf

echo "BLACKOUT\_DURATION = 86400" >$ALERTA\_SVR\_CONF\_FILE

export ALERTA\_DEFAULT\_PROFILE=production

## Alerta UI configuration

### install the web console of alerta run:

Step1. wget https://github.com/alerta/alerta-webui/releases/download/v7.4.5/alerta-webui.tar.gz

Step2. tar -zxvf alerta-web.tgz

Step3. cp dist/\* /var/www/html

Step4. vi config.json # Add the below line

{"endpoint": "http://<IP Address>:8080"}

Step 5. Service httpd restart

### Service Creation and Running Alerta

vi /etc/system/system/alertad.service

[Unit]

Description=Alerta Server

[Service]

Type=idle

User=eopsadmin

Group=eopsadmin

WorkingDirectory=/opt/eopsadmin/alerta/bin

Environment="MONGO\_URI=mongodb://10.75.240.54:27017/monitoring"

ExecStart=/opt/eopsadmin/alerta/bin/alertad run --port 8080 --host 10.75.240.54

Restart=always

[Install]

WantedBy=multi-user.target

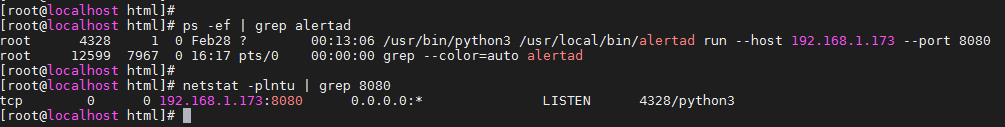
**Start alertad service**

Sudo Systemctl start alertad

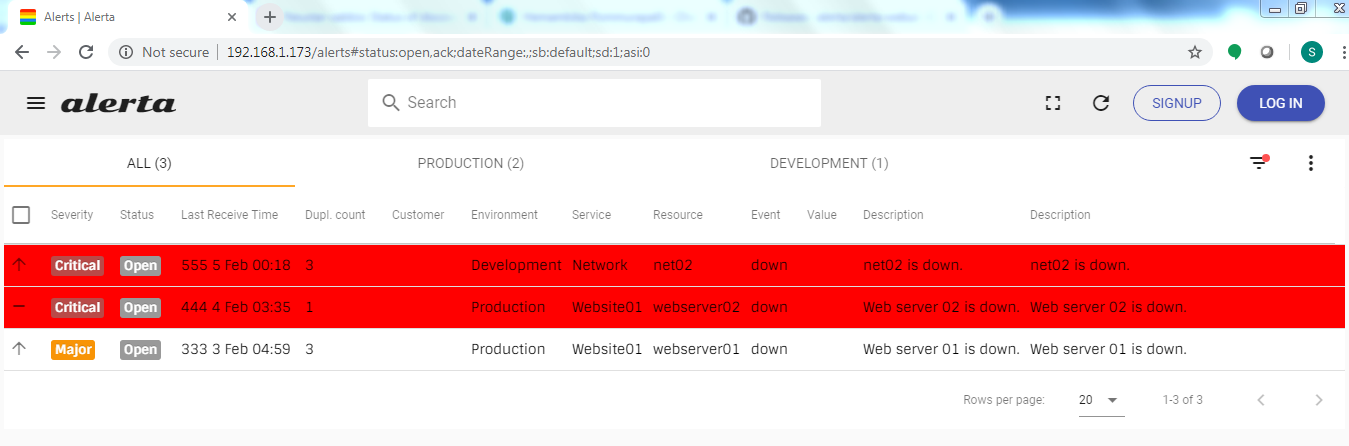
**Running from command line**

alertad run --host <IP Address > --port 8080 &

You can see alertad running on port 8080



**view the Api window in browser with http://<IP Address>**



# Elasticsearch Deployment using Ansible

## Steps before running the ansible playbook:

Step 1: Download the java and elasticsearch tar files and place them under ansible directory , In this playbook we have created a directory named playbooks and kept the tar files.

Step 2: We need to keep our elasticsearch yml file with different configurations related to master node , hot node , warm node ,coordinating node . If cold node presents that configuration also need to keep them.

Step 3: We should not keep the network host and node name in the yml file that we will be preparing for deploying in the nodes, that is managed by playbook.

Step 4: The hosts file should be prepared as different groups with master group , hot node group and warm node group respectively.

Step 5: We should be having the service file that are to be copied to the nodes to run elasticsearch as a service.

The below playbook installs a 5 node elasticsearch cluster with one master and four data nodes .

**Hosts file:**  
Hosts are to be given in the below format.

|  |
| --- |
| [master]  192.168.1.194 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no'  [hot]  192.168.1.195 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no'  192.168.1.125 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no'  [warm]  192.168.1.196 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no'  192.168.1.197 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no' |

## Playbook for elastic search cluster:

**Command** to run the playbook:

ansible-playbook playbooks/cluster.yml --ask-become-pass

|  |
| --- |
| ---  - hosts: all  remote\_user: eopsadmin  become: true  tasks: ##Here we are giving the pre-requesites for the elasticsearch  - name: Editing file for increasing virtual memory on remote hosts /etc/sysctl file  lineinfile:  dest: /etc/sysctl.conf  state: present  line: vm.max\_map\_count = 262144  register: vrt  - debug: var=vrt  - name: Setting maximum number of threads  lineinfile:  dest: /etc/security/limits.d/20-nproc.conf  state: present  regexp: '.\*nproc.\*'  line: 'root soft nproc unlimited'  register: scr  - debug: var=scr  - name: Setting max file descriptors  blockinfile:  path: /etc/security/limits.conf  block: |  root soft nofile 65536  root hard nofile 65536  eopsadmin soft nofile 65536  eopsadmin hard nofile 65536  register: lmt  - debug: var=lmt  - name: Set ulimits  shell: ulimit -n 65535  register: ul  - debug: var=ul  ##Creating a directory eopsadmin , if already exist the step will be skipped  - name: Ansible check directory.  stat:  path: /opt/eopsadmin  register: my\_folder    - name: "echo if directory already existed"  debug:  msg: "the eopsadmin directory is already existed"  when: my\_folder.stat.exists    - name: "Ansible Create directory if not exists"  file:  path: /opt/eopsadmin  state: directory  become: yes #we are copying the java tar file to every node , the file should be on the ansible node  - name: copy java tar file to remote host  copy:  src: /etc/ansible/playbooks/jdk-8u181-linux-x64.tar.gz  dest: /opt/eopsadmin  register: jv  - debug: var=jv  #we are copying the elasticsearch tar file to every node , this file should be present on the ansible node  - name: copy elasticsearch tar file to remote host  copy:  src: /etc/ansible/playbooks/elasticsearch-7.9.1-linux-x86\_64.tar.gz  dest: /opt/eopsadmin  register: els  - debug: var=els  ##We are installing the java with eopsadmin user  - name: Install Java  unarchive:  src: /opt/eopsadmin/jdk-8u181-linux-x64.tar.gz  dest: /opt/eopsadmin  owner: eopsadmin  group: eopsadmin  mode: 0755  remote\_src: yes  register: jav  ignore\_errors: true  - debug: var=jav  #Setting the java env variables in root and eopsadmin user.  - name: Rename java directory  shell: mv /opt/eopsadmin/jdk1.8.0\_181 /opt/eopsadmin/java  - name: Set Java Path for root  blockinfile:  path: /root/.bash\_profile  block: |  export JAVA\_HOME=/opt/eopsadmin/java  export PATH=$PATH:$JAVA\_HOME/bin  register: jpath  - debug: var=jpath  - name: Save Path in root  shell: source /root/.bash\_profile  register: rjava  - debug: var=rjava  - name: Set Java Path for eopsadmin user  blockinfile:  path: /home/eopsadmin/.bash\_profile  block: |  export JAVA\_HOME=/opt/eopsadmin/java  export PATH=$PATH:$JAVA\_HOME/bin  register: jepath  - debug: var=jepath  - name: Save Path in eopsadmin user  shell: source /home/eopsadmin/.bash\_profile  register: ejava  - debug: var=ejava  ##Installation of the elasticsearch with eopsadmin user  - name: Install elasticsearch  unarchive:  src: /opt/eopsadmin/elasticsearch-7.9.1-linux-x86\_64.tar.gz  dest: /opt/eopsadmin  owner: eopsadmin  group: eopsadmin  mode: 0755  remote\_src: yes  register: els  ignore\_errors: true  - debug: var=els  - name: Rename /opt/eopsadmin/elasticsearch folder  shell: mv /opt/eopsadmin/elasticsearch-7.9.1 /opt/eopsadmin/elasticsearch  register: res  - debug: var=res  - name: taking backup elasticsearch yml  shell: mv /opt/eopsadmin/elasticsearch/config/elasticsearch.yml /opt/eopsadmin/elasticsearch/config/elasticsearch\_backup\_yml  #Copying the certificates directory to the all elasticsearch nodes and placing in config directory  - name: Copy certificate to config directory  copy:  src: /etc/ansible/playbooks/certificate  dest: /opt/eopsadmin/elasticsearch/config  #copying the pre-defined master elastic node related config file to the master nodes which are given in hosts file  - name: Copy elasticsearch yml to host  copy:  src: /etc/ansible/playbooks/esyml/master\_es.yml  dest: /opt/eopsadmin/elasticsearch/config  when: "'master' in group\_names"  - name: Rename elasticsearch yml file  shell: mv /opt/eopsadmin/elasticsearch/config/master\_es.yml /opt/eopsadmin/elasticsearch/config/elasticsearch.yml  when: "'master' in group\_names"  - name: changing permission of yml file  shell: chown -R eopsadmin:eopsadmin /opt/eopsadmin/elasticsearch/config/elasticsearch.yml  when: "'master' in group\_names"  #We are not giving the network host and node name in the pre-defined config that is to be taken care of ansible .  - name: get the hostname  command: hostname  when: "'master' in group\_names"  register: ht  - debug: var=ht  - name: Configuring elasticsearch.yml file  blockinfile:  path: /opt/eopsadmin/elasticsearch/config/elasticsearch.yml  block: |  node.name: "{{ht.stdout}}"  network.host: "{{ht.stdout}}"  http.port: 9200  ignore\_errors: true  when: "'master' in group\_names"  register: jn  tags: conf  - debug: var=jn    #copying the pre-defined master elastic node related config file to the hot nodes which are given in hosts file  - name: Copy elasticsearch yml to host  copy:  src: /etc/ansible/playbooks/esyml/hot\_es.yml  dest: /opt/eopsadmin/elasticsearch/config  when: "'hot' in group\_names"  - name: Rename elasticsearch yml file  shell: mv /opt/eopsadmin/elasticsearch/config/hot\_es.yml /opt/eopsadmin/elasticsearch/config/elasticsearch.yml  when: "'hot' in group\_names"  - name: changing permission of yml file  shell: chown -R eopsadmin:eopsadmin /opt/eopsadmin/elasticsearch/config/elasticsearch.yml  when: "'hot' in group\_names"  #We are not giving the network host and node name in the pre-defined config that is to be taken care of ansible  - name: get the hostname  command: hostname  when: "'hot' in group\_names"  register: ht  - debug: var=ht.stdout  - name: Configuring elasticsearch.yml file  blockinfile:  path: /opt/eopsadmin/elasticsearch/config/elasticsearch.yml  block: |  node.name: "{{ht.stdout}}"  network.host: "{{ht.stdout}}"  http.port: 9200  ignore\_errors: true  when: "'hot' in group\_names"  register: jn  tags: conf  - debug: var=jn  #copying the pre-defined master elastic node related config file to the warm nodes which are given in hosts file  - name: Copy elasticsearch yml to host  copy:  src: /etc/ansible/playbooks/esyml/warm\_es.yml  dest: /opt/eopsadmin/elasticsearch/config  when: "'warm' in group\_names"  - name: Rename elasticsearch yml file  shell: mv /opt/eopsadmin/elasticsearch/config/warm\_es.yml /opt/eopsadmin/elasticsearch/config/elasticsearch.yml  when: "'warm' in group\_names"  - name: changing permission of yml file  shell: chown -R eopsadmin:eopsadmin /opt/eopsadmin/elasticsearch/config/elasticsearch.yml  when: "'warm' in group\_names"  #We are not giving the network host and node name in the pre-defined config that is to be taken care of ansible  - name: get the hostname  command: hostname  when: "'warm' in group\_names"  register: ht  - debug: var=ht.stdout  - name: Configuring elasticsearch.yml file  blockinfile:  path: /opt/eopsadmin/elasticsearch/config/elasticsearch.yml  block: |  node.name: "{{ht.stdout}}"  network.host: "{{ht.stdout}}"  http.port: 9200  ignore\_errors: true  when: "'warm' in group\_names"  register: jn  tags: conf  - debug: var=jn.stdout    - name: Replace xms value  replace:  path: /opt/eopsadmin/elasticsearch/config/jvm.options  regexp: '## -Xms4g'  replace: '-Xms5g'  register: xs  - debug: var=xs  - name: Replace xmx value  replace:  path: /opt/eopsadmin/elasticsearch/config/jvm.options  regexp: '## -Xmx4g'  replace: '-Xmx5g'  register: xx  - debug: var=xx  #We are copying the elasticseach service files to configure it as a service.  - name: copy elasticsearch file to init.d  copy:  src: /etc/ansible/playbooks/servicefiles/elasticsearch\_init  dest: /etc/init.d/elasticsearch  mode: 0777  register: esini  - debug: var=esini  - name: copy elasticsearch file to sysconfig  copy:  src: /etc/ansible/playbooks/servicefiles/elasticsearch\_sys  dest: /etc/sysconfig/elasticsearch  mode: 0777  register: esys  - debug: var=esys  - name: Enable elasticsearch  shell: sudo systemctl enable elasticsearch  register: ees  - debug: var=ees  - name: Start elasticsearch  shell: sudo systemctl start elasticsearch.service  register: ses  - debug: var=ses    ... |
|  |

## Rolling Restart Ansible playbook

The Rolling Restart of the cluster, here the playbook should be updated when the number of nodes are increased , here it is a 5node cluster’s rolling restart playbook.  
Steps that are followed by the playbook

Step1: First the number of nodes in the cluster are to be check and registered to a variable.

Step2: The shard allocation on the node is turned off for the nodes and stopping the elasticsearch.

Step3: Performing any changes for the nodes and again starting the elasticsearch service.

Step4: The playbook waits for the node to be rejoined to the cluster , it will be retrying until the node joins the cluster.  
Step5: The shard allocation is turned on again and it will wait for the cluster status to be Green and checks the other node.

Hosts file for the rolling restart is to be given in below format.

|  |
| --- |
| [master]  192.168.1.194 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no'  [hot1]  192.168.1.195 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no'  [hot2]  192.168.1.125 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no'  [warm1]  192.168.1.196 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no'  [warm2]  192.168.1.197 ansible\_connection=ssh ansible\_user=eopsadmin ansible\_ssh\_pass=access ansible\_ssh\_extra\_args='-o StrictHostKeyChecking=no' |

### Playbook for Rolling Restart

Command to run the playbook: In the etc/ansible directory we need to run the comand

ansible-playbook playbooks/rolling\_restart.yml --ask-become-pass

|  |
| --- |
| ---  - hosts: all  remote\_user: eopsadmin  become: true  tasks:  - name: Check number of nodes in the cluster  shell: "curl -u elastic:access 'https://192.168.1.196:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: nodes\_count  - name: Turn off shard reallocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"none\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'warm1' in group\_names"    - name: Stop the elasticsearch service  service:  name: "elasticsearch"  state: stopped  when: "'warm1' in group\_names"    - name: Start the elasticsearch service  service:  name: "elasticsearch"  state: started  when: "'warm1' in group\_names"    - name: Wait for the node to rejoin the cluster  shell: "curl -u elastic:access 'https://192.168.1.196:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: new\_nodes  until: new\_nodes.stdout == nodes\_count.stdout  retries: 200  delay: 3  when: "'warm1' in group\_names"    - name: Turn on shard allocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"all\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'warm1' in group\_names"    - name: Wait until cluster status is green  shell: "curl -u elastic:access -XGET https://192.168.1.196:9200/\_cluster/health --insecure"  register: response  until: response.stdout.find('"status":"green"') != -1  retries: 100  delay: 10  when: "'warm1' in group\_names"    - name: Check number of nodes in the cluster  shell: "curl -u elastic:access 'https://192.168.1.197:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: nodes\_count  - name: Turn off shard reallocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"none\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'warm2' in group\_names"    - name: Stop the elasticsearch service  service:  name: "elasticsearch"  state: stopped  when: "'warm2' in group\_names"    - name: Start the elasticsearch service  service:  name: "elasticsearch"  state: started  when: "'warm2' in group\_names"    - name: Wait for the node to rejoin the cluster  shell: "curl -u elastic:access 'https://192.168.1.197:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: new\_nodes  until: new\_nodes.stdout == nodes\_count.stdout  retries: 200  delay: 3  when: "'warm2' in group\_names"    - name: Turn on shard allocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"all\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'warm2' in group\_names"    - name: Wait until cluster status is green  shell: "curl -u elastic:access -g -XGET https://192.168.1.197:9200/\_cluster/health --insecure"  register: response  until: response.stdout.find('"status":"green"') != -1  retries: 100  delay: 10  when: "'warm2' in group\_names"  - name: Check number of nodes in the cluster  shell: "curl -u elastic:access 'https://192.168.1.195:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: nodes\_count  - name: Turn off shard reallocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"none\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'hot1' in group\_names"    - name: Stop the elasticsearch service  service:  name: "elasticsearch"  state: stopped  when: "'hot1' in group\_names"    - name: Start the elasticsearch service  service:  name: "elasticsearch"  state: started  when: "'hot1' in group\_names"    - name: Wait for the node to rejoin the cluster  shell: "curl -u elastic:access 'https://192.168.1.195:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: new\_nodes  until: new\_nodes.stdout == nodes\_count.stdout  retries: 200  delay: 3  when: "'hot1' in group\_names"    - name: Turn on shard allocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"all\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'hot1' in group\_names"    - name: Wait until cluster status is green  shell: "curl -u elastic:access -g -XGET https://192.168.1.195:9200/\_cluster/health --insecure"  register: response  until: response.stdout.find('"status":"green"') != -1  retries: 100  delay: 10  when: "'hot1' in group\_names"  - name: Check number of nodes in the cluster  shell: "curl -u elastic:access 'https://192.168.1.125:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: nodes\_count  - name: Turn off shard reallocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"none\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'hot2' in group\_names"    - name: Stop the elasticsearch service  service:  name: "elasticsearch"  state: stopped  when: "'hot2' in group\_names"    - name: Start the elasticsearch service  service:  name: "elasticsearch"  state: started  when: "'hot2' in group\_names"    - name: Wait for the node to rejoin the cluster  shell: "curl -u elastic:access 'https://192.168.1.125:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: new\_nodes  until: new\_nodes.stdout == nodes\_count.stdout  retries: 200  delay: 3  when: "'hot2' in group\_names"    - name: Turn on shard allocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"none\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'hot2' in group\_names"    - name: Wait until cluster status is green  shell: "curl -u elastic:access -g -XGET https://192.168.1.125:9200/\_cluster/health --insecure"  register: response  until: response.stdout.find('"status":"green"') != -1  retries: 100  delay: 10  when: "'hot2' in group\_names"  - name: Check number of nodes in the cluster  shell: "curl -u elastic:access 'https://192.168.1.194:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: nodes\_count  - name: Turn off shard reallocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"none\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'master' in group\_names"    - name: Stop the elasticsearch service  service:  name: "elasticsearch"  state: stopped  when: "'master' in group\_names"    - name: Start the elasticsearch service  service:  name: "elasticsearch"  state: started  when: "'master' in group\_names"    - name: Wait for the node to rejoin the cluster  shell: "curl -u elastic:access 'https://192.168.1.194:9200/\_cat/nodes?h=name' --insecure | wc -l"  register: new\_nodes  until: new\_nodes.stdout == nodes\_count.stdout  retries: 200  delay: 3  when: "'master' in group\_names"    - name: Turn on shard allocation  shell: "curl --header \"Content-Type: application/json\" -u elastic:access -g -XPUT https://192.168.1.196:9200/\_cluster/settings -d '{\"transient\" :{\"cluster.routing.allocation.enable\" : \"all\" }}' --insecure"  register: response  failed\_when: response.stdout.find('"acknowledged":true') == -1  when: "'master' in group\_names"    - name: Wait until cluster status is green  shell: "curl -u elastic:access -g -XGET https://192.168.1.194:9200/\_cluster/health --insecure"  register: response  until: response.stdout.find('"status":"green"') != -1  retries: 100  delay: 10  when: "'master' in group\_names"  ... |

# Appendix

## YML’s and conf files

**Note:** All the ES, Kibana, Logstash(producer & consumer), metricbeat, packet beat, winlogbeat yml files and conf files were attached in above sections.

## SLM Retention Schedule

Cluster setting to let the Retention/Deletion of expired snapshots for every 3 Hours



## Templates, ILM, Alias of all Data Sources along with SLM

Templates for each index created, see below:

**Note:** We configured the “wait\_for\_snapshot” policy as part of ILM policy in the delete phase













## Rolling Restart ES nodes

In the case of [full-cluster restart](https://www.elastic.co/guide/en/elasticsearch/reference/current/restart-cluster.html#restart-cluster-full), you shut down and restart all the nodes in the cluster while in the case of [rolling restart](https://www.elastic.co/guide/en/elasticsearch/reference/current/restart-cluster.html#restart-cluster-rolling), you shut down only one node at a time, so the service remains uninterrupted.

**Rolling restart Steps:**

1. Nodes those are not master-eligible.

Command: GET /\_nodes/\_all,master:false

1. Nodes that is master-eligible.

Command: GET /\_nodes/master:true.

1. **Disable shard allocation.**

When you shut down a node, the allocation process waits for index.unassigned.node\_left.delayed\_timeout (by default, one minute) before starting to replicate the shards on that node to other nodes in the cluster, which can involve a lot of I/O. Since the node is shortly going to be restarted, this I/O is unnecessary. You can avoid racing the clock by [disabling allocation](https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-cluster.html#cluster-routing-allocation-enable) of replicas before shutting down the node:

|  |
| --- |
| PUT \_cluster/settings  {  "persistent": {  "cluster.routing.allocation.enable": "primaries"  }  } |

1. **Stop indexing and perform a synced flush.**

Performing a [synced-flush](https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-synced-flush-api.html) speeds up shard recovery.

Command: POST \_flush/synced

When you perform a synced flush, check the response to make sure there are no failures. Synced flush operations that fail due to pending indexing operations are listed in the response body, although the request itself still returns a 200 OK status. If there are failures, reissue the request.

**Note** that synced flush is deprecated and will be removed in 8.0. A flush has the same effect as a synced flush on Elasticsearch 7.6 or later.

1. **Temporarily stop the tasks associated with active machine learning jobs and datafeeds.** (Optional)

POST \_ml/set\_upgrade\_mode?enabled=true

1. **Shut down a single node in case of rolling restart.**

Command: sudo systemctl stop elasticsearch.service

1. **Perform any needed changes.**
2. **Restart the node you changed.**

Command: sudo systemctl start elasticsearch.service

1. **Wait for the node join the cluster**

GET \_cat/nodes

1. **Re-enable shard allocation.**

Once the node has joined the cluster, remove the cluster.routing.allocation.enable

setting to enable shard allocation and start using the node:

|  |
| --- |
| PUT \_cluster/settings  {  "persistent": {  "cluster.routing.allocation.enable": null  }  } |

1. **Repeat in case of rolling restart.**

When the node has recovered and the cluster is stable, repeat these steps for each node that needs to be changed.

1. **Restart machine learning jobs.** (Optional)

If you temporarily halted the tasks associated with your machine learning jobs, use the [set upgrade mode API](https://www.elastic.co/guide/en/elasticsearch/reference/current/ml-set-upgrade-mode.html) to return them to active states:

POST \_ml/set\_upgrade\_mode?enabled=false

## Kiban Dev tool commands

Below are few useful Kibana Devtool commands:

GET \_cat/shards?h=index,shard,prirep,state,unassigned.reason

GET \_cluster/health?pretty

GET \_cluster/allocation/explain?pretty

GET neteng-flowdata-lstash-000987/\_ilm/explain

GET \_cat/shards?v

GET \_cat/shards/neteng-flowdata-lstash-000987

## Issues and solutions

1. **Cluster Yellow due to unassigned\_shards (or) initializing\_shard**

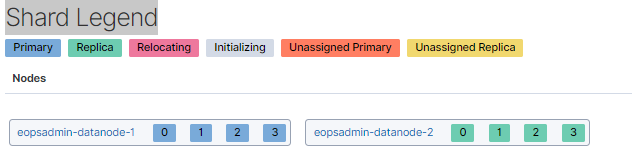
When we check:<https://awopuse1elasticmasternode01.laas.neustar.biz:9200/_cluster/health?pretty>

**Reason:** Checked logs and identified that the replica shards are failing to get updated in the respective nodes due to the “low disk watermark [85%] exceeded” on data nodes

**Log:** view /opt/eopsadmin/elasticsearch/logs/“neustar-eopsadmin-logging”.log

|  |
| --- |
| [ERROR][o.e.x.m.c.c.ClusterStatsCollector] [eopsadmin-masternode-1] collector [cluster\_stats] timed out when collecting data  low disk watermark [85%] exceeded on [0Hdm-bdSTwSjDi2LSzAM-A][eopsadmin-datanode-4][/opt/eopsadmin/elasticsearch/data/nodes/0] free: 221.2gb[14.7%], replicas will not be assigned to this node  low disk watermark [85%] exceeded on [COD9M6QCQGu7lqat1Kb6uQ][eopsadmin-datanode-3][/opt/eopsadmin/elasticsearch/data/nodes/0] free: 223.1gb[14.8%], replicas will not be assigned to this node  [WARN ][o.e.c.r.a.AllocationService] [eopsadmin-masternode-1] failing shard [failed shard, shard [secops-awsvpc-lstash-000303][1], node[M8kM-6DrThO6YVMN8GBiaw], [R], recovery\_source[peer recovery], s[INITIALIZING], a[id=7lOeVJDdTLOvm88tbQuTsQ], unassigned\_info[[reason=ALLOCATION\_FAILED], at[2020-10-12T19:24:10.575Z], failed\_attempts[1], delayed=false, details[failed shard on node [M8kM-6DrThO6YVMN8GBiaw]: failed to perform indices:data/write/bulk[s] on replica [secops-awsvpc-lstash-000303][1], node[M8kM-6DrThO6YVMN8GBiaw], [R], s[STARTED], a[id=8oQnRYFLRNCp6lxoU7l5bA], failure RemoteTransportException[[eopsadmin-datanode-1][10.75.240.57:9300][indices:data/write/bulk[s][r]]]; nested: CircuitBreakingException[[parent] Data too large, data for [<transport\_request>] would be [7102756910/6.6gb], which is larger than the limit of [7074139340/6.5gb], real usage: [7102734632/6.6gb], new bytes reserved: [22278/21.7kb], usages [request=10605760/10.1mb, fielddata=65038715/62mb, in\_flight\_requests=24515592/23.3mb, accounting=70208842/66.9mb]]; ], allocation\_status[no\_attempt]], message [failed to perform indices:data/write/bulk[s] on replica [secops-awsvpc-lstash-000303][1], node[M8kM-6DrThO6YVMN8GBiaw], [R], recovery\_source[peer recovery], s[INITIALIZING], a[id=7lOeVJDdTLOvm88tbQuTsQ], unassigned\_info[[reason=ALLOCATION\_FAILED], at[2020-10-12T19:24:10.575Z], failed\_attempts[1], delayed=false, details[failed shard on node [M8kM-6DrThO6YVMN8GBiaw]: failed to perform indices:data/write/bulk[s] on replica [secops-awsvpc-lstash-000303][1], node[M8kM-6DrThO6YVMN8GBiaw], [R], s[STARTED], a[id=8oQnRYFLRNCp6lxoU7l5bA], failure RemoteTransportException[[eopsadmin-datanode-1][10.75.240.57:9300][indices:data/write/bulk[s][r]]]; nested: CircuitBreakingException[[parent] Data too large, data for [<transport\_request>] would be [7102756910/6.6gb], which is larger than the limit of [7074139340/6.5gb], real usage: [7102734632/6.6gb], new bytes reserved: [22278/21.7kb], usages [request=10605760/10.1mb, fielddata=65038715/62mb, in\_flight\_requests=24515592/23.3mb, accounting=70208842/66.9mb]]; ], allocation\_status[no\_attempt]]], failure [RemoteTransportException[[eopsadmin-datanode-1][10.75.240.57:9300][indices:data/write/bulk[s][r]]]; nested: CircuitBreakingException[[parent] Data too large, data for [<transport\_request>] would be [7095497226/6.6gb], which is larger than the limit of [7074139340/6.5gb], real usage: [7095446472/6.6gb], new bytes reserved: [50754/49.5kb], usages [request=10622200/10.1mb, fielddata=65043964/62mb, in\_flight\_requests=2341104/2.2mb, accounting=68697698/65.5mb]]; ], markAsStale [true]]  org.elasticsearch.transport.RemoteTransportException: [eopsadmin-datanode-1][10.75.240.57:9300][indices:data/write/bulk[s][r]]  Caused by: org.elasticsearch.common.breaker.CircuitBreakingException: [parent] Data too large, data for [<transport\_request>] would be [7095497226/6.6gb], which is larger than the limit of [7074139340/6.5gb], real usage: [7095446472/6.6gb], new bytes reserved: [50754/49.5kb], usages [request=10622200/10.1mb, fielddata=65043964/62mb, in\_flight\_requests=2341104/2.2mb, accounting=68697698/65.5mb] |

Kiban-ui > StackMonitopring > Nodes > Indices and click on the yellow status index > Overview > Shard Legend we can onserver respective shard STATUS



**Solution:** Now we need to try to free the disk on respective data nodes ( by decreasing the number of days of index storage) and then try to re-allocate the replica accordingly with respective shard

<https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-reroute.html>

1. We implemented below : ( and it worked of rone idex replica issue)

|  |
| --- |
| POST /\_cluster/reroute?retry\_failed=true  {  "commands": [  {  "allocate\_replica": {  "index": "neteng-flowdata-lstash-000986", "shard": 1,  "node": "eopsadmin-datanode-2"  }  }  ]  } |

Above will let the shard move to Initilizating status and if allgood it wilbe fixed else it fall back to UNASSIGNED status

1. If above won’t work then we need to move the primary shard to other node (ex. Node2 from Node1) and then try to replicate the shard (to Node 1)

|  |
| --- |
| POST /\_cluster/reroute  {  "commands": [  {  "move": {  "index": "secops-awsvpc-lstash-000303", "shard": 3,  "from\_node": "eopsadmin-datanode-2", "to\_node": "eopsadmin-datanode-1"  }  }  ]  }  POST /\_cluster/reroute?retry\_failed=true  {  "commands": [  {  "allocate\_replica": {  "index": "secops-awsvpc-lstash-000303",  "shard": 3,  "node": "eopsadmin-datanode-2"  }  }  ]  } |

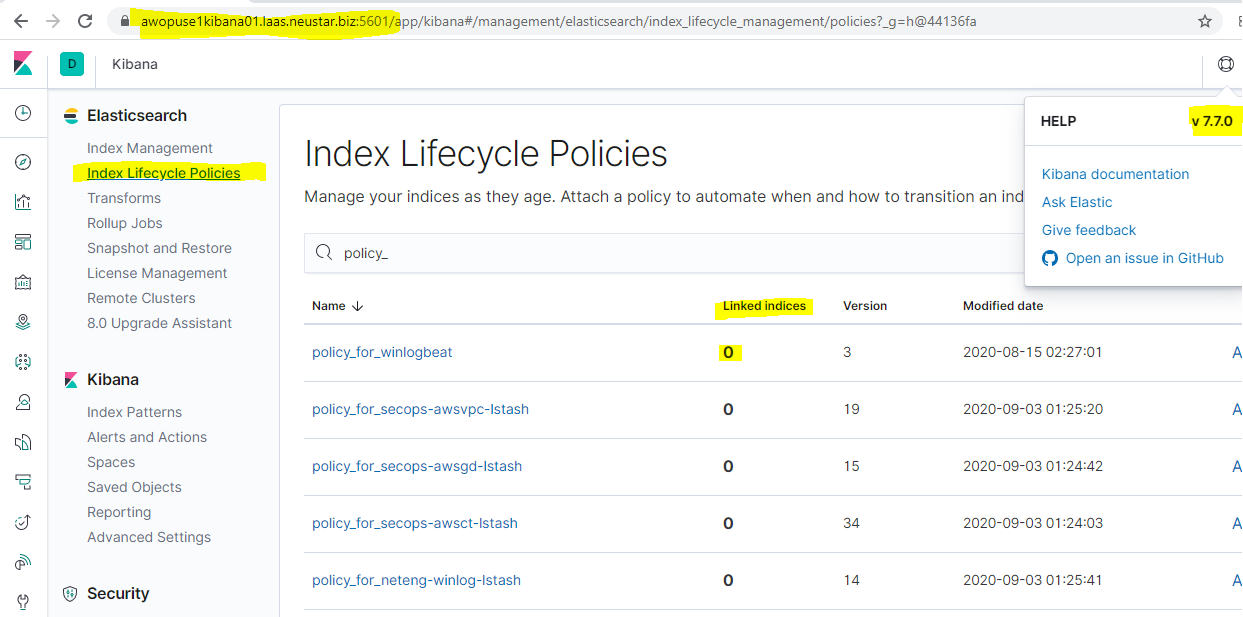
Above will let the shard move to Initializing status issue will be fixed.

1. **Linked indices not being updated:**

Login to <https://awopuse1kibana01.laas.neustar.biz:5601>/

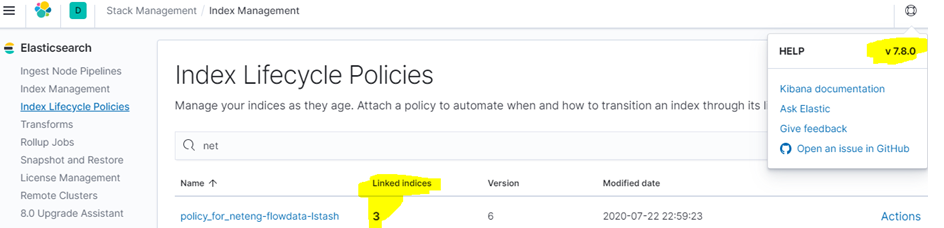
**Observation:** Observered that the Linked indices were not being updated correctlyto the respective index policy name, under

Kibana-UI > management > Index Lifecycle Policies > Linked indices



**Reason:** It is a bug identified in the **v7.7.0** and in the later version it is fixed which we validated.

See, sample screenshot from **v7.8.0** where this issue is already addressed.



## Create separate indices, ILM and aliase for each data type for syslog data source

Create separate indices for each data type along with aliases and Index life cycle management policies under on index for syslog data source.

**Index creation:**

Create indices for syslog data bases on event types(checkpoint,fortient and cisco) indices.

Below is the logstash output code for creating separate indices based on device type

output {

if [event][class] == "firewall\_checkpoint" {

elasticsearch {

hosts => ["https://masteres.laas.neustar.biz:9200","https://hotnode-es.laas.neustar.biz:9200","https://warmnode-es.laas.neustar.biz:9200"]

index => "neteng-syslog-lstash-checkpoint\_firewall-"

user => "elastic"

password => "elastic"

cacert => "/opt/eopsadmin/logstash/config/certificate/nsdev-cert-signed.crt"

ssl => true

#ssl\_certificate\_verification => true

ssl\_certificate\_verification => false

}

}

else if [event][class] == "firewall\_fortinet" {

elasticsearch {

hosts => ["https://masteres.laas.neustar.biz:9200","https://hotnode-es.laas.neustar.biz:9200","https://warmnode-es.laas.neustar.biz:9200"]

index => "neteng-syslog-lstash-fortinet-"

user => "elastic"

password => "elastic"

cacert => "/opt/eopsadmin/logstash/config/certificate/nsdev-cert-signed.crt"

ssl => true

#ssl\_certificate\_verification => true

ssl\_certificate\_verification => false

}

}

else if [message] =~ "%ASA-" {

elasticsearch {

hosts => ["https://masteres.laas.neustar.biz:9200","https://hotnode-es.laas.neustar.biz:9200","https://warmnode-es.laas.neustar.biz:9200"]

index => "neteng-syslog-lstash-cisco-asa-"

user => "elastic"

password => "elastic"

cacert => "/opt/eopsadmin/logstash/config/certificate/nsdev-cert-signed.crt"

ssl => true

#ssl\_certificate\_verification => true

ssl\_certificate\_verification => false

} }

}

}

All three indices are indexed into “neteng-syslog-lstash-\*” index

**Alias:**

Create the alias for indices by using below Api command

PUT neteng-syslog-lstash-checkpoint\_firewall-000001

{

"aliases": {

"neteng-syslog-lstash-checkpoint\_firewall -":{

"is\_write\_index": true

}

}

}

PUT neteng-syslog-lstash-fortinet-000001

{

"aliases": {

"neteng-syslog-lstash-fortinet -":{

"is\_write\_index": true

}

}

}

PUT neteng-syslog-lstash-cisco-asa -000001

{

"aliases": {

"neteng-syslog-lstash-cisco-asa -":{

"is\_write\_index": true

}

}

}

**ILM:-**

Create ILM policies for all the three indices as below

PUT \_ilm/policy/syslog-security-cisco\_asa\_policy

"policy" : {

"phases" : {

"warm" : {

"min\_age" : "0ms",

"actions" : {

"allocate" : {

"include" : { },

"exclude" : { },

"require" : {

"box\_type" : "hot"

}

},

"shrink" : {

"number\_of\_shards" : 1

},

"set\_priority" : {

"priority" : 50

}

}

},

"hot" : {

"min\_age" : "0ms",

"actions" : {

"rollover" : {

"max\_size" : "10kb",

"max\_age" : "5m"

},

"set\_priority" : {

"priority" : 100

}

}

},

"delete" : {

"min\_age" : "5m",

"actions" : {

"delete" : {

"delete\_searchable\_snapshot" : true

}

}

}

}

}

**Template:-**

Create a templete and add the ILM policy and rollover alias name that we have created above for each data type

PUT /\_template/template\_for\_syslog-security-cisco-asa

{

"index\_patterns" : [

"syslog-security-\*"

],

"settings" : {

"index" : {

"lifecycle" : {

"name" : "policy\_for\_syslog-security-cisco-asa",

"rollover\_alias" : "syslog-security-cisco-asa-"

},

"routing" : {

"allocation" : {

"require" : {

"box\_type" : "hot"

}

}

},

"unassigned" : {

"node\_left" : {

"delayed\_timeout" : "1m"

}

},

"number\_of\_shards" : "2",

"number\_of\_replicas" : "1"

}

}

}