**For access to linux server:**

Use Putty to logon to jump server

PuTTY Release 0.60

Host Name: UX\*\*\*\*\*\*\*\*

Port: 22

Followed by DST username and password (dt id and password)

**Do ssh to your linux machine**

[dt\*\*\*\*\*@ux\*\*\*\*\*\*\*~]# ssh root@10.xxx.xxx.xxx

**NOTE:**

[root@awdlinux ~]# cat /etc/system-release //to know about Linux OS

[root@awdlinux ~]# cat /etc/issue //to know the issue

[root@awdlinux ~]# uname –a or uname –m //to know the system architecture

[root@awdlinux ~]# arch //to know the system architecture

**Download latest ELK tool for your Linux version (Red Hat Enterprise Linux Server release 6.5 (x86\_64)**

-elasticsearch (elasticsearch-5.4.3.rpm)

-logstash (logstash-5.4.3.rpm)

-java8 (jdk-8u131-linux-x64.rpm)

-kibana (kibana-5.4.3-x86\_64.rpm)

-filebeat (filebeat-5.4.3-x86\_64.rpm)

-nginx (nginx-1.12.0-1.el6.ngx.x86\_64.rpm)

-GPG-KEY-elasticsearch

**For transferring the Installable/setup/files from windows to jump server using Secure ftp (Java Secure Channel Library)**

***In windows system***

Download JSch [**jsch-0.1.54.zip**](http://sourceforge.net/projects/jsch/files/jsch/0.1.54/jsch-0.1.54.zip/download) (link: [www.jcraft.com/jsch/](http://www.jcraft.com/jsch/))

Inside example folder, compile Sftp.java

Compile-> javac Sftp.java

Run-> java Sftp followed by details (host: UXP\*\*\*\*\*\*\*\* Username: <\*\*\*\*\*\*>) password: <\*\*\*\*\*\*\*>

*\*Leave the Kerberos username and password blank.*

Use ftp to send files to Linux machine from windows machine:

sftp> put local-path [remote-path] //To upload file to linux machine

eg. : put D:\Downloads\elasticsearch-5.4.3.rpm /\*\*\*\*\*\*\*/<directory-to-paste>

\*NOTE: for help on sftp> help or ?

Now all the Installable will be on jump server

**Copy these installable to linux machine from jump server use scp**

syntax: scp <src-file> user@ip-address:/<linux-machine-dest>

eg.: [dt\*\*\*\*\*@ux\*\*\*\*\*\*\* ~]$ scp logstash-5.4.3.rpm [root@10.xxx.xxx.xxx:/root/downloads](mailto:root@10.xxx.xxx.xxx:/root/downloads) followed by root password

**Do ssh to your linux machine**

[dt\*\*\*\*\*@ux\*\*\*\*\*\*\* ~]# ssh root@10.xxx.xxx.xxx

**To Install the installable:**

***Import elasticsearch PGP key***

rpm –import GPG-KEY-elasticsearch

***install elk***

[root@awdlinux ~]# rpm –ivh jdk-8u131-linux-x64.rpm

[root@awdlinux ~]# rpm –ivh elasticsearch-5.4.3.rpm

[root@awdlinux ~]# rpm –ivh kibana-5.4.3-x86\_64.rpm

[root@awdlinux ~]# rpm –ivh logstash-5.4.3.rpm

[root@awdlinux ~]# rpm –ivh filebeat-5.4.3.rpm

***To access kibana remotely***

[root@awdlinux ~]# rpm –ivh nginx-1.12.0-1.el6.ngx.x86\_64.rpm

go to /etc/nginx/

vi nginx.conf

user nginx;

worker\_processes 1;

error\_log /var/log/nginx/error.log warn;

pid /var/run/nginx.pid;

events {

worker\_connections 1024;

}

http {

include /etc/nginx/mime.types;

default\_type application/octet-stream;

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

access\_log /var/log/nginx/access.log main;

sendfile on;

#tcp\_nopush on;

keepalive\_timeout 65;

#gzip on;

include /etc/nginx/conf.d/\*.conf;

}

go to /etc/nginx/conf.d

vi kibana.conf

server {

listen 80;

server\_name "10.193.237.129";

location / {

proxy\_pass http://localhost:5601;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

***Important paths (directory structure)***

/etc/default/kibana.yml

/etc/default/logstash.yml

/etc/elasticsearch/

/etc/filebeat/

/etc/kibana/

/etc/logstash/

/etc/nginx/

/etc/nginx/

/etc/nginx/nginx.conf

/etc/nginx/conf.d/kibana.conf

/etc/init.d/logstash.conf

/etc/yum.repos.d/elasticsearch.repo

/etc/yum.repos.d/kibana.repo

/etc/yum.repos.d/logstash.repo

/usr/share/elasticsearch/

/usr/share/filebeat/

/usr/share/kibana/

/usr/share/nginx/

***set java PATH , JAVA\_HOME***

go to /usr/java/

export PATH=$PATH: /usr/java/jdk1.8.0\_131/bin

export JAVA\_HOME=/usr/java/jdk1.8.0\_131

***start elasticsearch***

go to /etc/elasticsearch

vi elasticsearch.yml

http.host: 0.0.0.0 (or http.bind\_host: 0.0.0.0)

network.host: 127.0.0.1 (or http.publish\_host: 192.168.10.10)

http.port: 9200

service elasticsearch start

*To check use command: $ curl localhost:9200*

***start kibana***

go to /etc/kibana

vi kibana.yml

server.host: “localhost”

server.port: 5601

elasticsearch.url: <http://localhost:9200>

elasticsearch.preserveHost: true

service kibana start

*To check use command: $ curl localhost:5601*

***start nginx***

To browser kibana in remote system

service nginx start

***Run the javaliveserver program***

[root@awdlinux elk]# javac ServerLog.java

[root@awdlinux elk]# nohup java ServerLog &

***start logstash***

go to /usr/share/logstash/bin/

either write or copy your config\_plugin here

|  |  |
| --- | --- |
| input {  beats {  port => 5044  add\_field => {  "input\_field\_added" => "success"  }  }  }  filter {  #grok match for timestamp match, used for replacing with log timestamp  grok {  match => [ "message", "%{TIMESTAMP\_ISO8601:tstamp} %{LOGLEVEL:log-level} " ]  } | #Replace the log timestamp with logstash @timestamp  date {  match => ["tstamp" , "ISO8601"]  target => "@timestamp"  #add\_field => { "debug" => "timestampMatched"}  #remove\_field => "tstamp"  }  }  output {  elasticsearch {  hosts => ["localhost:9200"]  index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"  codec => "rubydebug"  }  } |

[root@awdlinux bin]# nohup ./logstash -e 'input { beats { port => "5044" }} output { file { path => "/root/elk/file.log" codec=>"rubydebug"} elasticsearch { hosts => "localhost:9200" codec => "rubydebug" index => "logstash-%{+YYYY.MM.dd}" }}' &

***start filebeat***

go to /etc/filebeat/

vi filebeat.yml

paths:

- /root/elk/javalivelog/logFile.log

#output.elasticsearch:

#hosts: [“localhost:9200”]

output.logstash:

hosts:[“localhost:5044”]

service filebeat start

**NOTE:**

***Commands***

locate <filename> //To search file

rpm –qa |sort -n //To list all rpm installed in sorted

rpm –qa |grep –i <installable name> //To check Installable

rpm –e <package\_name> //To uninstall a package

netstat –nltp //Active connections only

netstat -anp |grep java //look onto the connections

su root -c "ps" //To run command as a different user

awk -F':' '{ print $1}' /etc/passwd // To see all users

export PATH=$PATH:/dst/home/dt216476/java/jdk1.8.0\_131/bin //set java PATH

export JAVA\_HOME= /dst/home/dt216476/java/jdk1.8.0\_131 //set JAVA\_HOME

***java path settings:***

first.sh (shell file):

#!bin/sh

export PATH=$PATH:/…./java/jdk1.8.0\_131/bin/

export JAVA\_HOME=/…./java/jdk1.8.0\_131/

To run .sh (shell to set environment variables) use: source ./first.sh

***Settings***

*elasticsearch.yml*

cluster.name: "myfirst-cluster"

node.name: "myfirst-node"

network.host: 127.0.0.1

http.host: 0.0.0.0

http.port: 9200

*kibana*.*yml*

server.port: 5601

server.host: "localhost"

elasticsearch.url: <http://localhost:9200>

elasticsearch.preserveHost: true

*filebeat.yml*

filebeat.prospectors:

- input\_type: log

- /root/elk/javalivelog/logFile.log

#output.elasticsearch:

#hosts: [“localhost:9200”]

output.logstash:

hosts: ["localhost:5044"]

*logstash.yml*

path.data: /var/lib/logstash

path.config: /etc/logstash/conf.d

path.logs: /var/log/logstash

*nginx.conf*

user nginx;

worker\_processes 1;

error\_log /var/log/nginx/error.log warn;

pid /var/run/nginx.pid;

events {

worker\_connections 1024;

}

http {

include /etc/nginx/mime.types;

default\_type application/octet-stream;

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

access\_log /var/log/nginx/access.log main;

sendfile on;

#tcp\_nopush on;

keepalive\_timeout 65;

#gzip on;

include /etc/nginx/conf.d/\*.conf;

}

*kibana.conf*

server {

listen 80;

server\_name "10.xxx.xxx.xxx";

location / {

proxy\_pass http://localhost:5601;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

***Create repository***

go to /etc/yum.repos.d and create 3 files

#vi kibana.repo

[kibana-5.x]

name=Kibana repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

#vi logstash.repo

[logstash-5.x]

name=Elastic repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

#vi elasticsearch.repo

[elasticsearch-5.x]

name=Elasticsearch repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

***how to run elasticsearch as a jboss***

give execute privileges to jboss and make elk chmod 777

chmod 555 /root

chmod 777 /root/elk

chgrp jboss /root/elk

The network.bind\_host setting allows controlling the host different network components will bind on. By default, the bind host will be anyLocalAddress (typically 0.0.0.0 or ::0).

The network.publish\_host setting allows controlling the host the node will publish itself within the cluster so other nodes will be able to connect to it. Of course, this can’t be the anyLocalAddress, and by default, it will be the first non-loopback address (if possible), or the local address.

The network.host setting is a simple setting to automatically set both network.bind\_host and network.publish\_host to the same host value.

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****